

Learning to Reach into the Unknown: Selecting Initial Conditions When Reaching in Clutter

Daehyung Park*, Ariel Kapusta, You Keun Kim, James M. Rehg, and Charles C. Kemp

Abstract—Often in highly-cluttered environments, a robot can observe the exterior of the environment with ease, but cannot directly view nor easily infer its detailed internal structure (e.g., dense foliage or a full refrigerator shelf). We present a data-driven approach that greatly improves a robot’s success at reaching to a goal location in the unknown interior of an environment based on observable external properties, such as the category of the clutter and the locations of openings into the clutter (i.e., apertures). We focus on the problem of selecting a good initial configuration for a manipulator when reaching with a greedy controller. We use density estimation to model the probability of a successful reach given an initial condition and then perform constrained optimization to find an initial condition with the highest estimated probability of success. We evaluate our approach with two simulated robots reaching in clutter, and provide a demonstration with a real PR2 robot reaching to locations through random apertures. In our evaluations, our approach significantly outperformed two alternative approaches when making two consecutive reach attempts to goals in distinct categories of unknown clutter. Notably, our approach only uses sparse readily-apparent features.

I. INTRODUCTION

Many manipulation tasks take place in environments with properties that are difficult to directly observe or indirectly infer. For example, dense foliage and a full shelf of a refrigerator consist of numerous objects that occlude the interior. Likewise, obscured locations behind structures can be reachable through gaps but not observable. Sensor limitations can also result in initial uncertainty about a particular environment, such as due to too little or too much light. In many of these situations, the robot can identify the category of the environment even though the specific details of the environment are unknown. Likewise, the robot can often detect openings through which it can reach. In this paper, we investigate a method that enables a robot to leverage its previous experience in environments with similar readily-apparent characteristics in order to improve its manipulation capabilities when confronted with a novel instance of a category of environment.

In particular, we focus on the problem of selecting an initial manipulator configuration when attempting to reach to a location in a cluttered environment. We assume that the robot only knows the goal location to which it should reach, the category of the environment, and the locations

D. Park, A. Kapusta, Y. Kim, and C. C. Kemp are with the Healthcare Robotics Lab, Georgia Institute of Technology, Atlanta, GA. *D. Park is the corresponding author {deric.park@gatech.edu}.

J. M. Rehg is with the Computational Perception Lab, Georgia Institute of Technology.

We gratefully acknowledge support from DARPA Maximum Mobility and Manipulation (M3) Contract W911NF-11-1-603, NIDRR RERC Grant H133E130037, and NSF Award IIS-1150157.

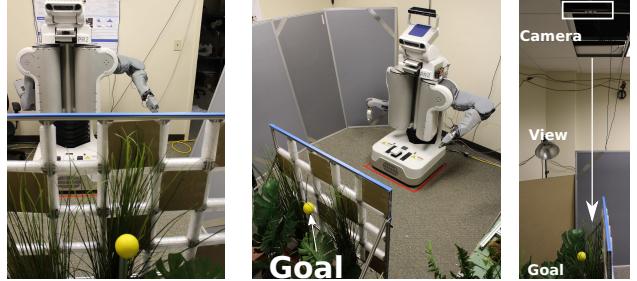


Fig. 1: Our method enabled a PR2 to select a base position, a torso height, an initial arm configuration, and an aperture to reach through in order to reach the yellow ball in the foliage. The label we used for this category of environment was *foliage-aperture-clutter*.

of apertures through which it can reach. We also assume that the robot has previously had the opportunity to learn by reaching into different instances of the same category of environment. Given a novel instance about which it has no specific information, the robot must decide how to configure its manipulator before using a greedy haptically-guided controller that we have previously presented [1]. We present our approach to data-driven selection, which we call learning initial conditions (LIC). We show that by intelligently selecting the initial configuration of its manipulator, a robot can greatly improve its chance of reaching the goal. We also show that if its first attempt fails, performing a second reach from another initial configuration, informed by its failure, results in a higher overall chance of success than other standard selection methods.

The selected initial manipulator configuration can influence the performance of a manipulation behavior in a variety of ways. And, predicting how the manipulator configuration will influence performance is challenging due to complex interactions among the controller, the robot’s body, and the environment. For example, configuring the end effector as a wedge and pointing it towards a gap can result in the end effector pushing through movable objects, while otherwise orienting it can result in the end effector pushing the movable objects along with it. Likewise, moving a link sideways increases the chances of it contacting an object, which might help guide the manipulator through the environment. Environments can also have dynamic elements that vary over time or due to the robot’s actions, such as a net or a fabric barrier. Given these complex interactions, we take a data-driven approach to probabilistically model the relationship between the initial configuration of the manipulator, the goal location, and the success of a reach for a given category of

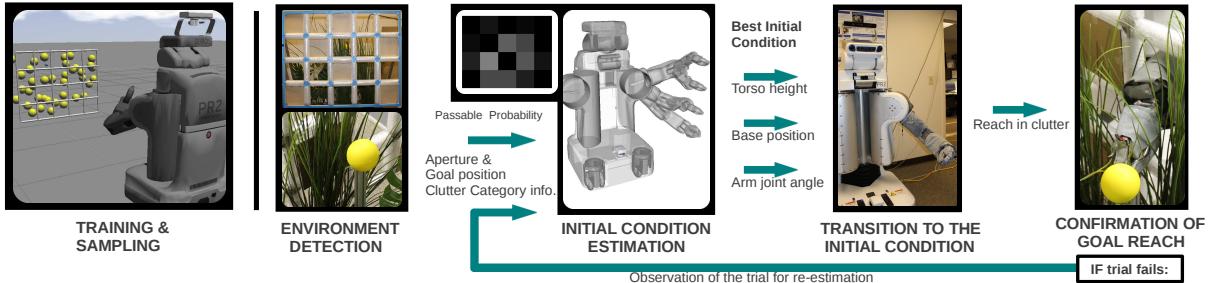


Fig. 2: Overview of the PR2 system, which uses LIC to reach into *foliage-aperture-clutter*. Training is performed in simulation prior to the real experiment. The goal and aperture locations were detected by external and head-mounted cameras, respectively. Then, LIC-1 selected the initial condition from the available base, torso, and arm configurations. If the first reach failed, LIC-2 was used. For the passable-probability map shown in the figure, whiter squares correspond with better apertures through which to reach.

environment. This has the benefit of making our approach straightforward to generalize to different robots, environments, and controllers.

We evaluated our approach in simulation using a 3 degree-of-freedom (DoF) planar arm. The environments consist of randomly placed rigid objects that were either fixed to the ground or movable. For example, a *cylinder-clutter* environment consists of upright cylinders, which has similarities to foliage. We also tested a narrow-passage environment, which is similar to a hidden tunnel. We found that, in both the first and second reach attempts, our approach outperformed the use of random or cost-metric initial configuration selection methods. In addition, we have demonstrated our approach using a Willow Garage PR2 robot with a 7 DoF arm in a real random-aperture environment. Our method has performed well in this common class of situations for which the robot can perceive openings through which to reach, which we refer to as apertures (see Fig. 1).

II. RELATED WORK

A number of previous studies have investigated the problem of learning policies that map the state of the world to a robot’s actions, and have used a variety of policy representations and learning methods such as those described in [2], [3], and [4]. In this paper, we use a predefined deterministic policy that takes the form of a controller that performs well when reaching in clutter [1]. At each time step, it uses model predictive control (MPC) to minimize the predicted distance between the robot’s end effector and the goal location while keeping predicted contact forces low. For this controller, our system learns initial conditions in the form of the manipulator’s initial configuration. In previous work, we assumed the category of a cluttered environment was known in order to rapidly classify incidental contact with the robot’s arm while reaching [5].

Our approach in this paper strongly relates to data-driven methods for trajectory generation. Jetchev and Toussaint showed how to find suitable trajectories from a trajectory database using high-dimensional situation descriptors ($\in \mathbb{R}^{791}$) with perfect information [6]. Dragan et al. use machine learning to select high-level attributes that are likely to be associated with an optimal trajectory, such as whether a trajectory should go to the left or right of an object [7]. Dey et al. proposed a framework to optimize control

sequences by learning contextual experience [8]. Berenson et al. presented an online framework for building a path library and producing a feasible path for a rapidly exploring random tree (RRT) [9]. This body of work emphasizes trajectory generation in fully known 3D environments. In contrast, we generate an initial condition that is likely to result in the success of a controller based on sparse readily-apparent features of highly-cluttered environments. We also provide evidence that these low-dimensional features can be highly informative.

Paolini et al. have pursued a strongly related approach to data-driven manipulation in clutter [10]. Our approach differs both in the task and the specifics of our probabilistic formulation. For example, we do not factor our probability distribution into two components, one relating perception to world state and the other relating world state to action outcomes.

In contrast to our approach, most research on manipulation in clutter assumes that detailed geometric information about the environment is available. Leeper et al. presented a method to help remote operators find collision-free poses for the end effector in clutter [11]. Researchers have also presented motion planners that use models of objects in the world to enable robots to rearrange clutter by pushing, grasping, and moving objects with their end effectors [12], [13], and [14].

III. LEARNING INITIAL CONDITIONS (LIC)

The starting configuration (i.e., initial condition) for a robot reaching into an unknown cluttered environment can significantly influence the overall chance of success. In our previous work we showed that greedy reaching in clutter with varying initial conditions is an effective method to achieve high rates of success (e.g., 91.4% of optimal given up to 5 reaches with distinct initial conditions) [1]. In this paper, we show that by intelligently selecting the initial condition for a reach based on readily-apparent features and past experience, a robot can achieve a high rate of success with substantially fewer reaches.

A. General Formulation

We propose a data-driven approach for selecting initial conditions that makes use of an experience library, $\mathcal{D} = \{(s_{-n}, x_{0-n}, f_{-n}), \dots, (s_{-1}, x_{0-1}, f_{-1})\}$. \mathcal{D} is a set of tuples of the form (s_i, x_{0i}, f_i) , where each tuple represents a

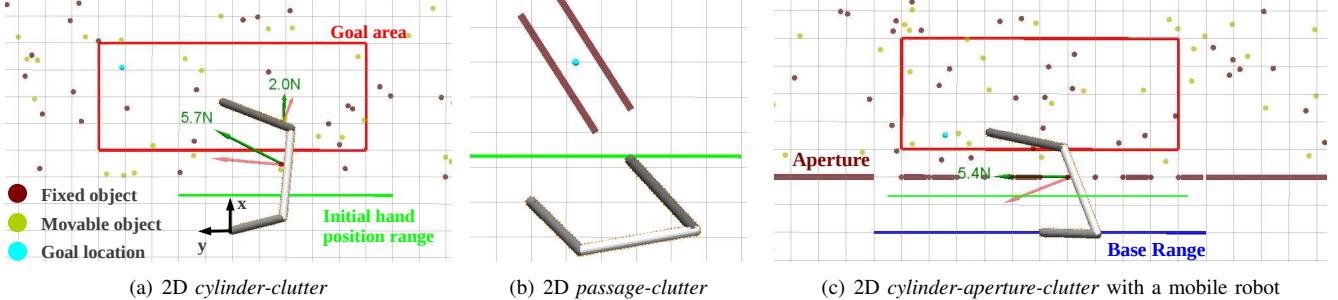


Fig. 3: Examples of two different categories of clutter with a three-link planar arm (grey and white). The arm starts from an initial condition on the hand range (green line) and reaches to a goal (cyan circle) in a goal area (red rectangle). The green and red arrows represent the contact force and the sensed normal component of this force on the arm’s surface. The example on the right also shows where the robot’s base can translate (blue line) and random apertures through which the robot must reach (gaps in the dark red line).

past reach attempt by the robot. s_i is a tuple that describes the specific situation encountered by the robot using sparse readily-apparent features. x_{0i} is the initial condition that was used by the robot. And, f_i is a truth value denoting whether the robot successfully reached its goal. Furthermore, we represent a situation as $s_i = (g_i, c_i, o_i)$, where g_i is the goal the robot attempted to reach, c_i is the category of the environment, and o_i represents other task-relevant observations the robot made prior to the attempt.

For our approach, we ideally wish to find a function m that maps the current situation, s , and past experience, \mathfrak{D} , to the initial condition x_0^* with the highest probability of success,

$$m(s, \mathfrak{D}) \rightarrow x_0^*, \quad (1)$$

Our data-driven method has two main steps: *learning* and *optimization*. The *learning* step builds \mathcal{F} , which is a non-parametric density estimate of $P(x_\infty = g|c, o, x_0)$, where x_∞ is the final state of the manipulator. That is $\mathcal{F}(s, x_0) \approx P(x_\infty = g|c, o, x_0)$. Notably, this density estimate is intended to generalize the contents of the experience library \mathfrak{D} to novel situations. In practice, we create a distinct model for each category of environment, c , and the content of our observation tuple, o , depends on the category. The *optimization* step searches \mathcal{F} to find \hat{x}_0^* , which is an initial condition with the maximal estimated probability of success, $\hat{x}_0^* = \arg \max_{x_0} \mathcal{F}(s, x_0)$, given the current situation, s .

We can summarize our method for approximating the function m as

$$\mathfrak{D} \xrightarrow{\text{Learning}} \mathcal{F}(s, x_0) \xrightarrow{\text{Optimization}} \hat{x}_0^*. \quad (2)$$

B. Reaching with a Mobile Base

Our formulation does not depend on the robot having a particular structure. For example, the initial condition, x_0 , for a highly redundant robot could represent all of the robot’s degrees of freedom. However, increasing the dimensionality of x_0 will typically make density estimation and optimization more challenging, including increasing the computation and training data required for good performance.

An important special case for robot reaching is when the robot has a mobile base that can translate the proximal end of a serial manipulator. If translating the robot and

the goal with respect to the environment does not change $P(x_\infty = g|c, o, x_0)$ we can describe the reaching task as being invariant to this translation. If we further require that during a reach the mobile base translates to a location and stops while the the serial manipulator reaches, we can use a more efficient approach to select the initial position of the mobile base and the initial configuration of the serial manipulator.

For this class of problem, we can make x_0 only represent the serial manipulator’s initial configuration and then learn \mathcal{F} with respect to the robot’s frame of reference. We can then perform the optimization step with respect to both translation of the mobile base, b_0 , and the serial manipulator’s configuration, x_0 , by finding the ordered pair (b_0, x_0) that results in the highest estimated probability of a successful reach. To do so, we appropriately translate the situation, s , into the mobile robot’s frame of reference. Specifically,

$$(b_0^*, x_0^*) = \arg \max_{(b_0, x_0)} \mathcal{F}(T_{b_0}(s), x_0), \quad (3)$$

where T_{b_0} translates s so that it represents the situation that would occur if the robot were to first translate to b_0 before reaching with its serial manipulator from configuration x_0 . As we demonstrate with our results, this approach can also work with reaching tasks that are only weakly invariant to translation. b_0 can also be selected from a limited set of translations, such as a set of translations that are more relevant to the task or for which the task is more invariant.

C. Making a Second Reach with LIC-2

The previous description provides a method with which a robot can select an initial condition, x_0 , when confronted with a wholly-novel situation, s , that consists of reaching to a goal, g , in an unknown environment of category, c , given various readily-apparent task-relevant observations, o . In this paper, we refer to the problem of learning initial conditions for this first reach as LIC-1 and the database used for LIC-1 as \mathfrak{D}_1 .

Due to the high uncertainty associated with this first reach (LIC-1), the robot will often fail. If it does fail to reach the goal, g , we would like to enable the robot to make another reach attempt to the same goal g in the same specific environment. While performing the first reach, we expect

that the robot will be able to make new readily-apparent observations that can enable it to do a better job selecting an initial condition for its second attempt.

We refer to the problem of selecting the initial condition, x_0 , for the second reach as LIC-2. For LIC-2, we could take a straightforward approach wherein we collect a new experience database, \mathcal{D}_2 , that solely consists of tuples describing second attempts to reach goal, g , each performed after a single failed attempt to reach g using the initial condition provided by LIC-1. In this way, LIC-2 could model causal dependencies between the first and second attempts and would be trained in the same manner as LIC-1, except for the use of \mathcal{D}_2 and additional observations. However, this would require having LIC-1 fully trained and collecting an entirely new experience database.

For this paper, we instead use a computationally favorable approximation for LIC-2 that simplifies training. We assume that a reach into a specific environment does not alter the environment in any way. This enables us to construct an experience database \mathcal{D}_2 from the same single reach attempts that we use to construct \mathcal{D}_1 . We do this by first having the robot attempt to reach a single goal in a specific environment from uniformly sampled initial conditions. Given our assumption that none of these reaches will change the environment, the probability of success for a particular reach does not depend on the reach attempts that preceded it. This enables us to create \mathcal{D}_2 by simulating the situation where the first reach fails and then a second reach is taken.

To simulate the first reach, we randomly select a failed reach from the set of failed reaches using equal probability across all of the failed reach attempts. Treating all possible first reach failures for a particular environment and goal as being equally likely has the advantage of not requiring a trained LIC-1. However, it introduces another approximation that is likely to reduce the performance of LIC-2, since LIC-2 would ideally be trained with respect to the actual distribution of failures resulting from LIC-1. We simulate the second reach by randomly selecting a reach from all of the reach attempts, whether they resulted in failure or not. To generate a useful set of experiences, we repeat this process for uniformly sampled goals over many environments, e_j , randomly drawn from the environment category, c , where the probability of drawing e_j is $P(e_j|c)$.

Once we have generated a set of simulated pairs of reaches, we append information obtained from the first reach to the observations available to the second reach. Each of the second reach attempt tuples in database \mathcal{D}_2 has an observation tuple, o_i , that includes additional observations obtained during the failed first reach. We always include the initial condition used in the failed first reach, x'_0 , as part of o_i , so that LIC-2 will know what was tried the first time. In addition, depending on the category of the environment, we include other observations, such as the final state of manipulator from the first reach, x'_∞ , which often corresponds to where the robot's end effector became stuck. Notably, rather than attempt to reconstruct a detailed model of the environment from information obtained during the

first reach, we instead use sparse, readily-apparent features to inform selection of the initial condition for the second reach. This works surprisingly well in practice.

Once we have generated \mathcal{D}_2 , the learning and optimization steps for LIC-2 are essentially the same as for LIC-1.

D. The Learning Step

For the *learning* step, we used a Gaussian process (GP) or K-Nearest Neighbor (K-NN) density estimator to generate \mathcal{F} from \mathcal{D} . For the GP estimation, we used an absolute exponential correlation model. For the K-NN estimation, we used a weighted Gaussian kernel. We performed these density estimates with the *scikit-learning* machine learning library in Python (<http://scikit-learn.org>).

One limitation of these estimators is that they can require a large amount of computation time during both learning and inference. To help overcome this issue, we sampled a fixed number of relevant local samples from the experience library, \mathcal{D} , using a K-NN local approximation algorithm prior to using the GP or K-NN density estimators, as described in [15]. In particular, we used this approach to reduce the number of samples when performing density estimation for LIC-2.

E. The Optimization Step

For the *optimization* step, we used the approximated probability distribution from the *learning* step as an objective function and performed the following constrained optimization:

$$\begin{aligned} & \underset{x_0}{\text{maximize}} && \mathcal{F}(s, x_0) \\ & \text{subject to} && q_{min} \leq IK(x_0) \leq q_{max} \\ & && x_0 \in \text{open space}, \end{aligned} \quad (4)$$

where q_{min} and q_{max} are minimum and maximum joint limits and $IK(x_0)$ is the inverse kinematics for the manipulator when the end effector is at pose x_0 . In addition, we constrained the end effector to be located in the open space outside of the clutter. For the simulated 3-link planar manipulator we used, x_0 defined a position and orientation for the manipulator, so there was no issue with redundancy. For the PR2, x_0 specified a pose of the end effector, which resulted in redundancy. For this paper, we handled this by pre-selecting a single arm configuration for each end effector pose, x_0 . To perform this constrained optimization, we used a bound-constrained minimization algorithm, L-BFGS-B [16] as implemented in SciPy (<http://www.scipy.org/>).

IV. EXPERIENCE LIBRARIES FOR EVALUATION

To build an actual experience library $\mathcal{D} = \{(s_{-n}, x_{0-n}, f_{-n}), \dots, (s_{-1}, x_{0-1}, f_{-1})\}$ with $s_i = (g_i, c_i, o_i)$, we need to define these tuples in detail. The details are especially important because, like most machine learning approaches, the effectiveness of LIC depends on using informative feature vectors that support generalization. The following list provides an overview of the parameters we used in our evaluation:

- **x_0 - initial condition:** For this paper, we defined x_0 to be the pose of the robot's end effector. For the 3-link

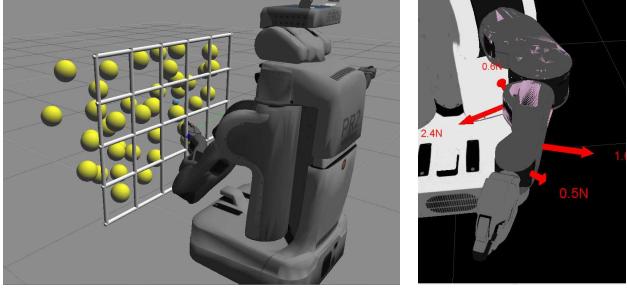


Fig. 4: **Left:** An example of *sphere-aperture-clutter* for the simulated PR2. **Right:** Visualized contact forces from simulated tactile sensors on the simulated PR2 arm. Red arrows represent contact forces normal to the arm’s surface.

planar arm x_0 was the position (x, y) and orientation θ of the end effector. For the PR2 robot, x_0 was the position (x, y, z) and orientation of the end effector with orientation represented as a quaternion (q_x, q_y, q_z, q_w) . For each class of environment, c , the allowed set of initial conditions can be different. For example, when presented with apertures, the robot is only allowed to select an initial end effector position that is at the center of an aperture, but it can select an end effector orientation from a continuous range.

- **g - goal:** We defined the goal, g , to be the ideal position of the end effector after a successful reach.
- **f - success or failure:** The truth value, f , denoted whether or not the end effector’s position was within a threshold distance of the goal, g , during a reach. Success did not depend on the end effector’s orientation.
- **c - category of the environment:** c represented the type of environment into which the robot must reach, such as a *cylinder-clutter*, *passage-clutter*, or *sphere-aperture-clutter* environment. For each simulated environment category in our evaluation, we defined a probability distribution from which we could draw environment instances. We also provided the correct category of the current environment to the robot prior to its first reach attempt.
- **o - task-relevant observations:** o varied based on the category of the environment, c , and whether the robot was making its first or second reach. For the first reach in our evaluations, o was the empty set, $o = \emptyset$. For the second reach in our evaluations, o consisted of information from the first reach attempt. Every second reach o included the initial condition used by the first reach, x'_0 . Depending on the environment category, o could also include the final position of the end effector in the first reach, x'_∞ , or an estimate of the final velocity of the end effector in the first reach, \dot{x}'_∞ .

In order to efficiently collect a large number of tuples, we had the robots perform reach attempts from pre-defined start configurations to task-space goals in simulation. First, we prepared a number of randomly cluttered environments, set \mathcal{V}_c , of category c . Next, we defined a discretized goal-set G in the environments and a start-condition set S in front of the robot. When simulating second reach data, we used all of the data we generated rather than randomly sampling from it with equal probability. We now provide details about

the data collection we used for our evaluation.

A. Data Collection for the 2D Testbed

We used two categories of randomly generated cluttered environments in a 2D testbed as described in Section V-A. Each environment included all planar and rigid objects with fixed sizes, masses, and friction coefficients.

- ***Cylinder-clutter*:** Using a uniform distribution, we randomly placed 40 movable and 40 fixed circular objects, each with a 0.01 m radius, in a $0.65 \times 2.4\text{ m}$ rectangular area, as shown in Fig. 3(a). A three-link planar arm attempted to reach from 21 initial conditions to 45 grid-distributed goals of size 5×9 in 20 different cluttered environments. We placed the goals on a horizontal, rectangular plane 0.4 m long and 0.8 m wide at 0.1 m intervals. The initial positions were on a segment, 0.8 m long, between the robot and the clutter, equally distributed at 0.1 m intervals. The initial orientations were equally distributed at 30° intervals. For sampling, we ran 18,900 reach attempts with 20 different clutter settings, 45 goal locations, and 21 initial conditions. For LIC-2, we used $o = \{x'_0, x'_\infty\}$.
- ***Passage-clutter*:** We randomly placed a fixed narrow passage with a 0.1 m gap between two 0.4 m long and 0.02 m wide walls, as shown in Fig. 3(b). The center positions of the passages were randomly selected on a horizontal segment 0.4 m long. The other properties were the same as the *cylinder-clutter*. The arm attempted to reach from 14 initial conditions to the middle point of the passage. We distributed the arm’s initial positions in the same way as in *cylinder-clutter*. The initial orientations were equally distributed at 45° intervals. For sampling, we ran 1,680 reach attempts of different clutter settings. For LIC-2, we used $o = \{x'_0, \dot{x}'_\infty\}$.
- ***Cylinder-aperture-clutter*:** A clutter environment was generated in the same way as the *cylinder-clutter* category. Then, fixed-width openings (apertures) were randomly placed in front of the environment (see Fig. 3(c)). The robot was given exactly one initial position for each aperture and initial orientations equally distributed at 45° intervals. We ran 7,200 reach attempts with 480 different clutter settings. For LIC-2, we used $o = \{x'_0\}$.

B. Data Collection for the 3D Testbed

In the 3D testbed described in Section V-A, we generated 40 random cluttered environments in a category, *sphere-aperture-clutter*, to obtain training data for a real experiment. The clutter contains 40 movable floating spheres in a $0.29 \times 0.4 \times 0.7\text{ m}$ rectangular parallelepiped in front of a simulated PR2. Each object had a 0.05 m radius and a 0.1 kg mass, and its velocity is decayed by a 0.5 exponential damping gain. Then, we put 20 square 0.2 m wide apertures between the clutter and the robot, as shown in Fig. 1 (Left).

The simulated PR2 tried to reach to 12 grid-distributed goals of size 4×3 in 40 different cluttered environments from 20 initial conditions. The goals were placed on a vertical, rectangular plane 0.6 m wide and 0.4 m tall at 0.2 m intervals behind a set of spheres. The initial positions were equally distributed on a vertical, rectangular plane of 0.8 m

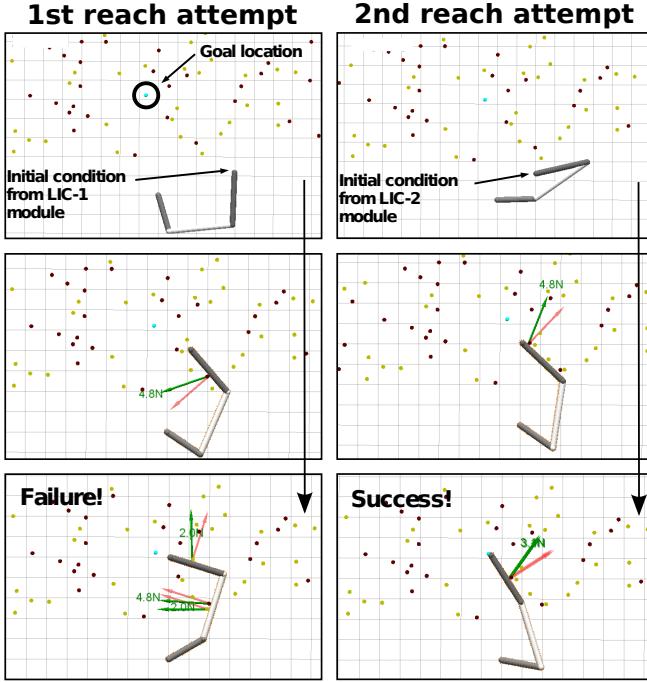


Fig. 5: Snapshots in time from two reach attempts in *cylinder-clutter*. **Left:** First reach attempt using LIC-1. **Right:** Second reach attempt using LIC-2.

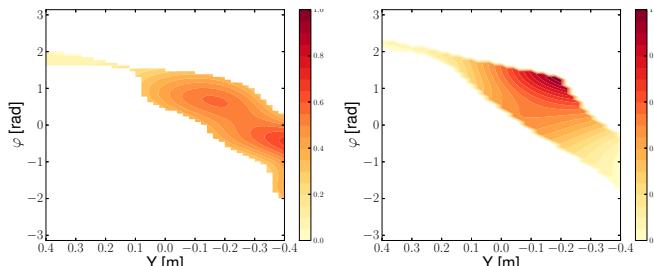


Fig. 6: Visualization of density estimates showing initial conditions that are more likely to succeed in red. The left density estimate from LIC-1 corresponds with the first reach attempt and the right density estimate from LIC-2 corresponds with the second reach attempt in Fig. 5. The initial condition is the end effector's position on the starting line and its orientation with respect to vertical (vertical is $\varphi=0$ rad). LIC-1 selected ($Y = -0.4$ m, $\varphi = -0.45$ rad) and LIC-2 selected ($Y = -0.190$ m, $\varphi = 1.216$ rad).

wide, 0.6 m tall, and 0.2 m intervals. To reduce the number of initial orientations, we selected one from each available configuration using a cost function C that assigns a lower cost to manipulator configurations with the end effector and forearm pointing towards the clutter.

$$C = 1/(\alpha \|x^T p_{(o_t o_w)}\| + \beta \|x^T p_{(o_t o_e)}\|), \quad (5)$$

where α and β are positive constants. x is a unit vector pointing from the robot to the clutter, $[1, 0, 0]^T$. $p_{(o_t o_w)}$ is a vector from the origin of the tip of the end effector to the wrist. $p_{(o_t o_e)}$ is a vector from the origin of the tip of the end effector to the elbow. We ran 9,025 reach attempts. We also used our approach from Section III-B to generalize the PR2's learning over translations parallel to the environment's surface, which it performed with its spine and wheeled base. For LIC-2, we used $o = \{x'_0\}$.

V. TESTBEDS FOR EVALUATION

We trained and evaluated our LIC method with two different simulation testbeds. Then, we demonstrated the performance in realistic clutter using a PR2.

A. Simulation Testbeds

We used a 2D simulation testbed with Open Dynamics Engine (ODE) (<http://www.ode.org/>). Fig. 3 shows the robot, which consists of a mobile base and a three-link planar arm, which has kinematic and dynamic properties similar to the properties of a human arm. The robot was controlled by a 1 kHz joint-space impedance controller, and tactile sensors were placed on the entire surface of each link with a density of 100 taxels per meter. A detailed description of this testbed can be found in [1]. In the scenarios depicted in Fig. 3(a) and 3(b), we restricted the base to a fixed point. In the scenario shown in Fig. 3(c) the base could move and we used our approach from Section III-B.

For 3D simulation, we used a simulated PR2 in Gazebo (<http://gazebosim.org>). The PR2, shown in Fig. 4, is a 32-DOF robot with two 7-DOF arms. We also simulated the arm with a 1 kHz joint-space impedance controller, and used simulated tactile sensors across the entire surface of the arm.

We used the quasi-static model predictive controller (MPC) from [1] to control all of the arms in this paper with a *don't care force threshold* of 5 N in 2D and 3 N in 3D.

B. Demonstration Testbed

For the proof-of-concept demonstration, we designed a real *foliage-aperture-clutter*, as shown in Fig. 1. 20 square apertures, 0.2 m wide, are randomly blocked and detected by a camera mounted on the PR2 head. A goal was also randomly placed and detected by an external camera. The objective was to reach the goal location by passing through the unobservable *foliage-clutter* behind the observable apertures. LIC selected the initial condition and the PR2 used MPC and fabric-based tactile sensors. Fig. 2 shows an overview of the experiment. LIC-1 estimated the probability of success of the available base, torso, and arm configurations given the goal and aperture locations. After selecting the initial condition, the PR2 moved to the location and then attempted to reach to the goal. If it failed, the PR2 tried again with LIC-2.

VI. EVALUATION

We tested our approach and compared it with random and cost-metric methods.

A. Strategies Used for Comparison

To evaluate our approach, we tested three strategies for initial condition selection:

- RND: This strategy randomly selected the first initial condition. If this reach fails it randomly selects the second condition. We assigned equal probability to all valid conditions.
- COST: This strategy selected the first initial condition from a cost-metric function that estimates a minimum-contact corridor between the hand and the goal. The selected condition is positioned closer to and orientated more

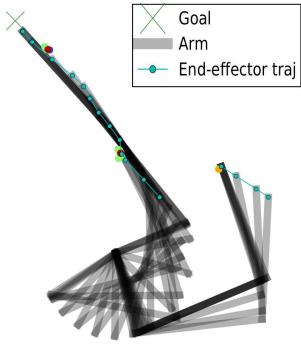


Fig. 7: An example of LIC-2 initiating a successful reach (left) into a still unknown part of the environment. LIC-1 initiated the failed reach on the right.

directly toward a goal than other conditions. If the first reach fails, this strategy selects the second best condition from the cost-metric function excluding the first condition's position and a local neighborhood around it.

- LIC: This strategy selects the first initial condition using LIC-1. If this reach fails, this strategy selects the second condition using LIC-2. Figures 5 and 6 illustrate the use of this approach.

B. Results

One advantage of LIC-1 is that it can improve performance when a robot reaches into an environment about which it has no specific geometric information. As illustrated by Fig. 7, LIC-2 can also result in the robot reaching into a still unknown part of the environment based on the results of the first reach.

We compared the performance of LIC to the random (RND) and cost-metric (COST) strategies. Fig. 8 shows that LIC with two different ML techniques, GP and K-NN, exhibited better performance than the RND and COST selection methods over thousands of reaches in two different categories of 2D cluttered environment. LIC-1 outperformed the first reach by the other strategies and outperformed the first and second reach made by the other strategies in the *passage-clutter* environment category. LIC-2 increased the overall success rate under all conditions, even though it only has the opportunity to make an attempt when LIC-1 fails.

Tables I and III show the numeric results for each initial condition selection method. The columns show the success rate of each strategy and the rows show the success rate of consecutive reach attempts. The fraction in parentheses gives the number of successful reach attempts out of the number of total reach attempts.

In the *cylinder-clutter* category, LIC had an overall success rate after two attempts of 8.45% compared to the RND and COST strategies success rates of 74.15% and 78.15%, respectively. In the *passage-clutter* category, LIC had a much greater overall success rate after two attempts of 83.95% compared to the RND and COST strategies success rates of 56.8% and 60.9%, respectively.

We also performed Student's *t*-tests on the resulting binomial distributions from the *cylinder-clutter* evaluation. Table II provides the *t*-test results between the pairs of tested approaches. We found significant differences between

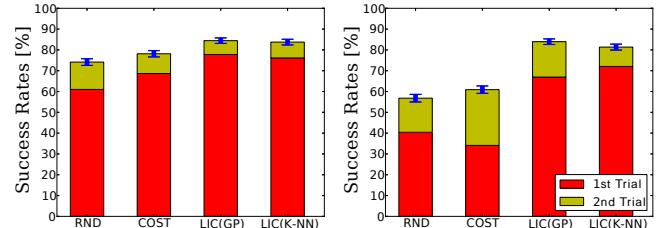


Fig. 8: Success rates of the methods we evaluated. Blue lines show 90% confidence intervals. **Left:** Comparison in *cylinder-clutter*. **Right:** Comparison in *passage-clutter*.

the performance of LIC and the other two strategies. In contrast, we did not find a significant difference between the performance of LIC with the two ML methods we used, GP and K-NN, for $\alpha = 0.05$.

We also evaluated our approach with *cylinder-aperture-clutter*, which requires a 3-link planar robot that can translate sideways to reach to a goal through random apertures (see Fig. 3(c)). Table IV shows that LIC had a greater overall success rate after two attempts of 77.8% compared to the RND and COST strategies success rates of 29.7% and 77.8%, respectively.

The PR2 robot reaching in *foliage-aperture-clutter* demonstrates that LIC can be applied to real robots reaching in 3D. Fig. 2 illustrates the operation of the PR2 reaching system. Fig. 9 shows an example of the PR2 successfully reaching through an aperture to a goal based on LIC. The PR2 successfully reached this goal after adjusting its base, torso, and arm configurations in that order.

VII. DISCUSSION

Although we focused on the problem of reaching in clutter, we expect that our approach could be generalized to other manipulation tasks. Other avenues for future work include extending our approach to more than two reach attempts, and performing more extensive evaluation of our approach with real robots reaching in real clutter.

Instead of detailed geometric models, our approach relies on probabilistic models strongly tied to the specific task of reaching. One challenge associated with our approach is the need to collect training data. As we illustrated with our PR2 demonstration, a plausible way forward is to use physical simulations to generate initial probabilistic models to inform the actions of real robots. As the real robot accumulates real-world experiences, it could then improve these initial models over time.

VIII. CONCLUSION

We have presented our approach to the problem of reaching into an unknown environment. The key to our approach is that readily-apparent aspects of an environment can usefully inform the first reach in the absence of detailed information about the environment's interior. For example, as we have shown, just knowing the category of an environment can enable a robot to select a better location from which to first reach into the unknown.

In addition, we've provided evidence that even simple observations from the first reach attempt, such as where

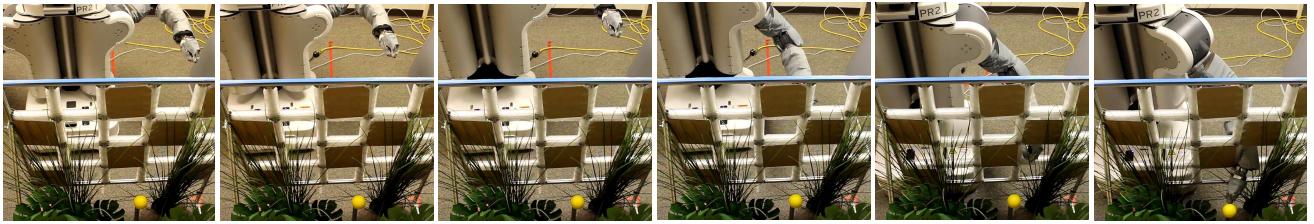


Fig. 9: Snapshots of from the PR2 reaching in *foliage-aperture-clutter*. It automatically adjusted its base, torso, and arm configurations using LIC to successfully reach the yellow ball.

TABLE I: Success rate of two consecutive reach attempts in *cylinder-clutter*. We use 200 random environments with 10 different random goal locations.

	RND	COST	LIC(GP)	LIC(K-NN)
1st attempt	Random	Cost metric	LIC1	LIC1
	61.0% (1220/2000)	68.65% (1373/2000)	77.8% (1556/2000)	76.15% (1523/2000)
2nd attempt	Random	Cost metric	LIC2	LIC2
	33.71% (263/780)	30.30% (190/627)	29.96% (133/444)	31.86% (152/477)
Total	74.15% (1483/2000)	78.15% (1563/2000)	84.45% (1689/2000)	83.75% (1675/2000)

TABLE II: Paired sample t-tests in *cylinder-clutter*. Each result describes degrees of freedom, *t*-test value, and *p*-value in order.

Pairs	Sample t-test result
RND-LIC(GP)	$t(3998) = -8.103, p = 7.059 \times 10^{-16}$
COST-LIC(GP)	$t(3998) = -5.125, p = 3.120 \times 10^{-7}$
RND-LIC(K-NN)	$t(3998) = -7.497, p = 8.038 \times 10^{-14}$
COST-LIC(K-NN)	$t(3998) = -4.520, p = 6.368 \times 10^{-6}$
LIC(GP)-LIC(K-NN)	$t(3998) = 0.605, p = 0.545$

the first reach started and where it ended, can beneficially inform a second reach attempt in the event of a failed first attempt. Our use of sparse, low-dimensional, readily-observable, task-relevant features distinguishes our approach from methods that attempt to make extensive observations in order to generate detailed models of the environment. Being able to perform well without having a detailed model of the environment gives robots the opportunity to efficiently act and also operate in environments that may not be conducive to detailed modeling, such as complex and dynamic cluttered environments, like a swamp or dense foliage blowing in the breeze.

REFERENCES

- [1] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, “Manipulation in clutter with whole-arm tactile sensing,” *International Journal of Robotics Research*, 2012.
- [2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robot. Auton. Syst.*, vol. 57, pp. 469–483, May 2009.
- [3] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proceedings of the twenty-first international conference on Machine learning*, ICML ’04, pp. 1–, ACM, 2004.
- [4] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, “Grasping pomdps,” in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 4685 –4692, April 2007.
- [5] T. Bhattacharjee, A. Kapusta, J. M. Rehg, and C. C. Kemp, “Rapid categorization of object properties from incidental contact with a tactile sensing robot arm,” 2013.
- [6] N. Jetchev and M. Toussaint, “Trajectory prediction: learning to map situations to robot trajectories,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML ’09, (New York, NY, USA), pp. 449–456, ACM, 2009.
- [7] A. Dragan, G. Gordon, and S. Srinivasa, “Learning from experience in manipulation planning: Setting the right goals,” in *Proceedings of the International Symposium on Robotics Research (ISRR)*, July 2011.
- [8] D. Dey, T. Y. Liu, M. Hebert, and J. A. Bagnell, “Contextual sequence prediction with application to control library optimization,” *Robotics*, p. 49, 2013.
- [9] D. Berenson, P. Abbeel, and K. Goldberg, “A robot path planning framework that learns from experience,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012.
- [10] R. Paolini, A. Rodriguez, S. S. Srinivasa, and M. T. Mason, “A data-driven statistical framework for post-grasp manipulation,” *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 600–615, 2014.
- [11] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, “Strategies for human-in-the-loop robotic grasping,” in *ACM/IEEE international conference on Human Robot Interaction*, 2012.
- [12] L. Y. Chang, S. Srinivasa, and N. Pollard, “Planning pre-grasp manipulation for transport tasks,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2010.
- [13] M. Dogar and S. Srinivasa, “A framework for push-grasping in clutter,” *Robotics: Science and Systems*, 2011.
- [14] M. Stilman, K. Nishiwaki, S. Kagami, and J. Kuffner, “Planning and executing navigation among movable obstacles,” *Advanced Robotics*, vol. 21, no. 14, pp. 1617–1634, 2007.
- [15] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair, “Gaussian process modeling of large scale terrain,” in *Robotics and Automation, ICRA ’09. IEEE International Conference on*, 2009.
- [16] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Trans. Math. Softw.*, vol. 23, pp. 550–560, Dec. 1997.