

Motion Planning for Mobile Manipulators with Physical Contact in Uncertain Environment

Jessica Leu, Liting Sun, and Masayoshi Tomizuka

Abstract—Mobile manipulators often need to perform tasks under uncertainties. For instance, they need to navigate around moving objects and contact or manipulate unknown objects such as opening a door to enter a room. Such uncertainties may influence the robot dynamics and correspondingly its optimal operations. Hence, to successfully perform given tasks, mobile manipulators should be aware of such influence and actively explore possible uncertainties in the environment to integrate them in the motion planning module. For instance, a mobile manipulator trying to push an object in front of it (either a pushable door or a box) should reason about the possible dynamics of the object so that it can find the optimal strategy to handle it (either push forward or detour). To enable such capabilities, we formulate the uncertainty-exploring motion planning as a partially observable Markov decision process (POMDP) with an additional reward term to encourage uncertainty exploration. We also present a hybrid optimization algorithm, namely, the Hamiltonian Monte Carlo sampling with convex feasible set algorithm (HMCCFS), to solve the POMDP efficiently. The performance of the proposed framework and algorithm is verified via both simulations and experiments. The results show that the framework helps the robot to perform various tasks more successfully in the presence of different environment uncertainties.

I. INTRODUCTION

In recent years, mobile manipulators have become a popular choice for robotic applications thanks to their agility in navigation and manipulation. Different from base-fixed robot manipulators, mobile manipulators typically operate in environments with more uncertainties. The uncertainties can come from unobserved or moving obstacles [1], [2], [3], human movements [4], contacts with the environment [5], [6], etc. When a mobile manipulator (particularly a light-weight one (Fig. 1 [7])) interacts with the environment through contacts, on one hand, the uncertainties may influence the mobile manipulator’s dynamics and change its optimal actions correspondingly. On the other hand, such contacts provide a way for the robot to collect information about the uncertainties. In this work, we focus on motion planning frameworks of the mobile manipulators which can leverage the power of contacts to handle uncertainties.

Specifically, we consider general light-weight mobile manipulators whose dynamics might be changed due to contacts with the environment. Such uncertainty-induced dynamics change might be small enough to be compatible with the original planning module, and might also go beyond such that the original plan becomes dynamically infeasible to the robot. In either case, without explicitly considering these

All authors are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA {jess.leu24, liting.sun, tomizuka}@berkeley.edu

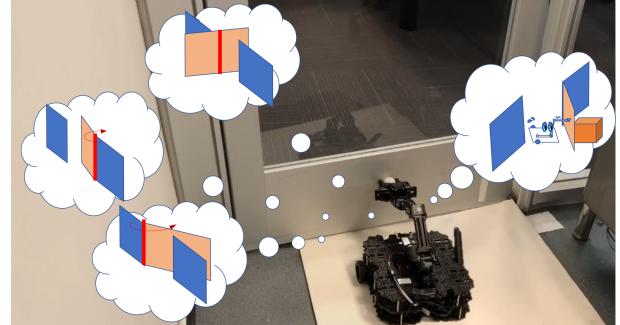


Fig. 1: A mobile manipulator tries to open a door and enter a room. Uncertainty exists since the robot does not know exactly whether this door is pushable or where the rotation axis of the door is.

uncertainties, the trajectories generated by traditional motion planners may no longer be optimal for the light-weight robot. For example, if a robot needs to pick an object without prior information about its center of mass, a motion planner not aware of such uncertainty may let the robot grasp the closest end of the object. This will result in an unstable grasp if the grasping point is far away from the center of mass. To explicitly handle uncertainties, one of the most common way is to replan or modify the trajectory as uncertainties occur [8]. The authors of [9] propose a method to adapt and modify the pre-planned trajectory online to deal with dynamic model discrepancies; while [10] carefully maps the uncertainty distribution in the state space and solves the planning problem.

Nevertheless, these passive methods may not be effective enough to handle uncertainties. For example, when a robot tries to push to open a door and enter a room, the dynamics of the robot will change significantly when it pushes on the rotation axis or some location that cannot generate enough torque to overcome the static friction. An even worse scenario is when an object behind the door blocks the door from being fully opened. This will cause failure since the robot will never be able to fully open the door no matter how it modifies the plan via re-planning. This is because the robot fails to actively explore and gain knowledge about the uncertainties. Several work has proposed to enable the robot to notify humans and get assistance [11], [12]. However, it is desired if the robot can actively deal with such uncertainties autonomously.

Hence, our key insight is that instead of passively adapt to uncertainties, mobile manipulators should leverage their ability of making contacts with the environment and actively explore and interpret environment uncertainties. In

other words, we need a motion planning formulation that enables robots to explore and collect information about the uncertainties so that they can find an optimal motion plan under such uncertainties. In the door example, if the robot can explore where it should push to open the door or realize that it can wiggle its way to pass the entrance without fully opening the door, it can successfully finish its task. Recently, this exploration idea has been demonstrated in human-robot-interaction scenarios for both autonomous driving and robot task planning [13], [4]. Inspired by these works, we apply and expand the idea to deal with environment uncertainties.

To equip mobile manipulators with the ability to explore uncertainties, we formulate a POMDP-based uncertainty-exploring motion planning problem where the robot is rewarded to explore and collect information in the presence of uncertainties. Our contributions are threefold:

- We propose a POMDP model for light-weight mobile manipulators in uncertain environment to formulate the uncertainty-exploring motion planning problem.
- We propose an algorithm, HMCCFS, that combines Hamiltonian Monte Carlo sampling (HMC) and the convex feasible set algorithm (CFS) to solve the POMDP.
- We implement the proposed method and demonstrate its success with both simulations and real-world experiments.

The remainder of the paper is organized as follows. Section 2 presents our proposed method. Sections 3 and 4 present the simulation and experiment results, respectively (video is publicly available at jessicaleu24.github.io/ICRA2021.html). Section 5 concludes the paper.

II. METHOD

In this section, we first formulate the uncertainty-exploring motion planning problem as a POMDP, and then propose an optimization method to solve the POMDP.

A. Uncertainties and the Robot Model

Robot-environment system and notation. Denote the observable state of the robot-environment system as $z_{all} = [z^\top, z_e^\top]^\top$, where z contains the robot state and z_e contains the environment state. For example, z_e can be used to capture the door open angle, the position of the observable obstacles, and the length and weight of the lifted portion of a chain. The robot and the environment are mutually influenced.

Denote the inputs of the robot as u . The robot dynamics at time k is shown below:

$$z_{k+1} = f_{dy}(z_k, u_k, z_{e,k}). \quad (1)$$

Similarly, the environment dynamics is a function of the robot input u and the environment internal characteristics, denoted by ϕ , i.e.,

$$z_{e,k+1} = f_e(z_{e,k}, u_k, z_k, \phi). \quad (2)$$

Hence, the robot dynamics can be re-written as:

$$z_{k+1} = f_{dy}(z_k, u_k, f_e(z_{all,k}, u_k, \phi)). \quad (3)$$

Uncertainty. We assume that the environment uncertainties is captured by its internal characteristic ϕ in Eq. (2). To do so, we rely on some prior knowledge. In the door example, if we know that the object blocking the entrance is a door, we have a prior knowledge a door can either be approximated as a second-order system with static friction, or as a wall if the door is blocked. But the exact model parameters such as the weight of the door, the static friction of the door hinge, the location of the door's rotation axis, etc. are captured by ϕ . In other words, ϕ contains continuous parameters such as the parameters of the physical system, as well as discrete parameters such as a Boolean parameter indicating whether the door is pushable. Parameters in ϕ are not directly observable and should be estimated by the robot to finish its task. For implementation, we discretize the environment parameter space to E discrete values, i.e., $\phi \in \{\phi_1, \phi_2, \dots, \phi_E\}$. We have the following assumption about the environment parameter:

Assumption 1 (The probabilistic model of the environment): The change of belief on the set of environment parameter, $\{\phi_1, \phi_2, \dots, \phi_E\}$, is a Markov process.

Although the robot does not have direct access to the environment parameter, ϕ , the belief of the environment parameter can be updated as the robot explores the uncertainties by making contacts with the environment.

B. Exploring and Planning

The mobile manipulator can update the belief over the environment parameter through contacts as follows:

$$b_{k+1}(\phi_i) \propto b_k(\phi_i) p(z_{e,k+1}|z_{all,k}, u_k, \phi_i), \quad (4) \\ \forall i=(1, 2, \dots, E).$$

Here we construct the conditional probability using our knowledge of the environment, i.e., $p(z_{e,k+1}|z_{all,k}, u_k, \phi_i) \propto \exp(-f_{\phi_i}(z_{e,k+1}, z_{all,k}, u))$. The function f_{ϕ_i} is determined by the environment dynamics and the environment parameter ϕ_i . The function first simulates the environment state $z_{e,k+1}$, then calculates the change from $z_{e,k}$ to $z_{e,k+1}$ and measures the consistency of the change with the expected behavior given ϕ_i . Since the environment dynamics with different ϕ_i expects different environment state outcomes, the more consistent the change is with the dynamic model, the higher the conditional probability is and so is the resulting belief at the next time step. The function f_{ϕ_i} (and also f_e in Eq. (2)) can be obtained by learning methods [13], [14] or be user-defined according the knowledge of the physical systems in the environment [15].

C. Planning with Exploration Under Uncertainty

With the previous belief update procedure, the uncertainty-exploring motion planning problem can be modeled as a POMDP [16] with states (z_{all}, ϕ) , action u , and reward $r(z_{all}, u)$ that the robot tries to maximize. The dynamic model of the POMDP is captured by Eq. (2) and (3). The observable state z_{all} can be used to update the belief of the environment parameter ϕ as introduced previously. We utilize the POMDP to formulate the motion planning problem that

drives the robot to explore and react to the uncertainties while reaching the goal. The planning problem is solved in a Receding Horizon Control (RHC) manner [17].

Notation. To formulate the RHC, we denote the planning horizon as H , the current robot state as $z(k)$, the input vector planned at time step k as $\mathbf{u}_k := [u_k^\top, u_{k+1}^\top, \dots, u_{k+H}^\top]^\top$, and the resulting state vector as $\mathbf{z}_{k+1} := [z_{k+1}^\top, z_{k+2}^\top, \dots, z_{k+H+1}^\top]^\top$. We obtain the dynamics, $\mathbf{z}_{k+1} = \mathbf{f}_{dy,z(k)}(\mathbf{u}_k)$, by concatenating the dynamic function (Eq. (3)) throughout the planning horizon.

Reward function. To drive the robot to explore and adjust to uncertainties while reaching the goal, we include three different terms in the reward function. At planning time step h ($h = k, k+1, \dots, k+H$), the first term is the state reward, $r_G(z_{all,h}, u_h, b_h)$, which encourages the robot to approach the goal with minimum input magnitude given different belief distribution. We also encourage the robot take actions that bring high information gain, therefore the second term, $r_I(b_{h+1}(\phi), b_h(\phi))$, calculates the change of the entropy over the belief of the environment parameter:

$$r_I(b_h(\phi), b_{h+1}(\phi)) = I(b_h(\phi)) - I(b_{h+1}(\phi)), \quad (5)$$

where

$$I(b_h(\phi)) = -\sum_{i=1}^E b_h(\phi_i) \log(b_h(\phi_i)). \quad (6)$$

Notice that

$$\sum_{h=k}^{k+H} r_I(b_h(\phi), b_{h+1}(\phi)) = I(b_k(\phi)) - I(b_{k+H+1}(\phi)). \quad (7)$$

Since $I(b_k(\phi))$ is a scalar, we denote

$$R_I(b_{k+H+1}(\phi)) := -I(b_{k+H+1}(\phi)). \quad (8)$$

The third term, $R_C(b_{k+H+1}(\phi))$, encourages actions that make the belief concentrate at the end of the plan so that the robot can be more certain about the uncertainties [18]. The function, $R_C(b_{k+H+1}(\phi))$, is proportional to the difference between the largest and the second largest belief. In other words, the larger the difference is, the higher the reward is, which indicates that the belief is more concentrated on the ϕ with the largest belief.

Constraints. The notion Γ_k defines the feasible set that satisfies the following assumption:

Assumption 2 (Constraint): At time step k , the state constraint Γ_k is non-convex and its complement is a collection of disjoint convex sets, i.e., each of the obstacle-region is itself convex.

Given j obstacles, the set Γ_k has the form:

$$\Gamma_k = \bigcap_j \Gamma_j = \bigcap_j \{\mathbf{z} : g_{j,k}(\mathbf{z}) \geq 0\}. \quad (9)$$

We assume that the constraint function $g_{j,k}(x)$ is a semi-convex function [19] that serves as a safety index with regard to the j th obstacle at time step k . For example, $g_{j,k}(x)$ can be the distance between a robot and the obstacle. The notion Γ_{max} is the input constraint.

The motion planning problem. With the above formulation, the problem has the following form:

$$\max_{\mathbf{u}_k} J(\mathbf{u}_k, b_k(\phi), z_{all,k}) \quad (10)$$

$$= \mathbb{E}_\phi \left[\sum_{h=k}^{k+H} r_G(z_{all,h}, u_h, b_h) \right] \quad (11)$$

$$+ R_I(b_{k+H+1}(\phi)) \quad (12)$$

$$+ R_C(b_{k+H+1}(\phi)), \quad (13)$$

$$s.t. \quad \mathbf{f}_{dy,z(k)}(\mathbf{u}_k) \in \Gamma_k, \quad (14)$$

$$\mathbf{u}_k \in \Gamma_{max}. \quad (15)$$

For simplicity, we will use J_k to refer $J(\mathbf{u}_k, b_k(\phi), z_{all,k})$ in the later sections.

D. Optimization Method

There are many ways to approximate the solution of a POMDP. Sampling-based algorithms in general are known to be fast and have better global optimality [20], [10]. Meanwhile, optimization-based algorithms [21], [13], although suffering from local optimality, can generate smooth trajectory and scale well in high dimension. Authors of [13] solve the self-driving car motion planning POMDP by a gradient decent method. A growing trend of solving optimization problem is to combine methods from different categories [22], [23]. This inspires the proposed method, HMCCFS, that combines a sampling-based algorithm with an optimization-based algorithm.

Notice that our motion planning problem is a non-convex and non-linear optimization problem. The many ways a robot contacts the environment under different uncertainty belief can be view as different homotopy classes of motion plans. The motion planning problem is also a high-dimensional planning problem. With these in mind, we proposed a hybrid method, HMCCFS, that combines Hamiltonian Monte Carlo (HMC) sampling [24] with the convex feasible set algorithm (CFS) [19]. HMC is one of the methods that can avoid local optima especially when the problem can have many different solutions from the distinct homotopy classes of the problem [25]. CFS is a fast optimization-based motion planning algorithm that solves non-convex problems by solving a sequence of convex sub-problems iteratively. The proposed algorithm has a hierarchical structure and is summarized as in Algorithm 1.

In Algorithm 1, given the current position, $z(k)$, the reward function J_k , and the constraints Γ_k and Γ_{max} , the HMC layer samples from different homotopy classes within a short period of time. To do so, we formulate a lower-dimensional problem that captures the critical type of movements instead of sampling in the original high-dimensional space. This is valid since we first focus more on finding the optimal homotopy class as opposed to finding the solution that generates smooth movements. The solution of this process, $\mathbf{u}^{(0)}$, is then used as a reference to convexify the original

Algorithm 1 HMCCFS

```

procedure HMCCFS( $z(k)$ ,  $J_k$ ,  $\Gamma_k$ ,  $\Gamma_{max}$ )
     $\mathbf{u}^{(0)} \leftarrow \text{HMC-sampling}(z(k), J_k, \Gamma_k, \Gamma_{max})$             $\triangleright$  Sample from different homotopy classes using HMC sampling.
     $J_{con,k} \leftarrow \text{convexify\_reward}(J_k, \mathbf{u}^{(0)})$                     $\triangleright$  Convexify the reward function with  $\mathbf{u}^{(0)}$ .
    while Stop criterion is not satisfied do
         $\chi^{(k-1)} = \bigcap_{j=1}^m \{\mathbf{u} : g_{j,k-1}(\mathbf{f}_{dy,z(k)}(\mathbf{u}^{(k-1)})) + \nabla^\top g_{j,k-1}(\mathbf{f}_{dy,z(k)}(\mathbf{u}^{(k-1)}))(\mathbf{u} - \mathbf{u}^{(k-1)}) \geq 0\}$ 
         $\mathbf{u}^{(k)} = \arg \min_{\mathbf{u} \in \chi^{(k-1)} \cap \Gamma_{max}} J_{con,k}(\mathbf{u}).$ 
    return  $\mathbf{u}^{(k)}$ 

```

reward function.

$$\begin{aligned}
J_{con,k}(\mathbf{u}) = & \\
& (\mathbf{u} - \mathbf{u}^{(0)})^\top \nabla_{\mathbf{u}\mathbf{u}} J_k(\mathbf{u}^{(0)}) (\mathbf{u} - \mathbf{u}^{(0)}) \\
& + \nabla_{\mathbf{u}}^\top J_k(\mathbf{u}^{(0)}) (\mathbf{u} - \mathbf{u}^{(0)}) + \|\mathbf{u} - \mathbf{u}^{(0)}\|_\Lambda. \tag{16}
\end{aligned}$$

We obtain the convexified reward $J_{con,k}$ shown above by approximating the reward function with regard to $\mathbf{u}^{(0)}$ using Newton's method. We also add a regularization term $\|\mathbf{u} - \mathbf{u}^{(0)}\|_\Lambda := (\mathbf{u} - \mathbf{u}^{(0)})^\top \Lambda (\mathbf{u} - \mathbf{u}^{(0)})$, where Λ is a weighting matrix, so that the final solution will stay in the same homotopy class as $\mathbf{u}^{(0)}$. With the convexified cost function and the constraints described previously, we can use CFS to obtain the final solution. In the k th iteration ($k = 1, 2, \dots$), CFS formulates a convex feasible set, $\chi^{(k-1)}$, by linearizing the constraints according to the previous solution $\mathbf{u}^{(k-1)}$. Then, a quadratic programming problem can be formulated and solved. The algorithm solves the problem iteratively and results in a sequence of $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}, \dots$. It is guaranteed in [19] that this sequence will converge to a local optimal, \mathbf{x}^* . In summary, the HMCCFS has two main features.

- HMCCFS has stochasticity due to the random sampling process in HMC so that HMCCFS is likely to find solutions in the optimal homotopy class under uncertainties.
- HMCCFS inherits the properties of CFS so that feasibility, smoothness, and convergence of the final solution are guaranteed.

III. SIMULATION RESULTS

A. Robot Model

The mobile manipulator, TB3O, used in this work is composed by a two-motor driven mobile platform (TurtleBot3), and a 4-DoF open-source manipulator (OpenMANIPULATOR). The robot has five links, including the base and four links of the arm. The configuration of TB3O can be determined by the following vectors: $\theta_w = [\theta_R, \theta_L]^\top$ are the angles of the left and right wheels; $\theta_m = [\theta_2, \theta_3, \theta_4, \theta_5]^\top$ are the angles of the arm joints; $z_p = [x_1, y_1, \theta_1]^\top$ are the x and y coordinates and the heading angle of the base. The robot state vector at time step k is denoted as $z_k = [z_s^\top z_s^\top]_k^\top$ where $z_s = [z_p, \theta_w, \theta_m]^\top$. The input vector is denoted as $u_k = [\alpha_R, \alpha_L, \alpha_2, \alpha_3, \alpha_4, \alpha_5]_k^\top$, which are the angular accelerations of the motors.

Denote the vector of motor angles to be $\Theta = [\theta_w^\top, \theta_m^\top]^\top$, a standard formulation of the robot dynamics is as follows:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) + \tau_{ext}(z_e), \tag{17}$$

where τ_{ext} is the torque caused by the external force applied to the robot's end-effector. Notice that τ_{ext} is a function of the environment state z_e , which is determined by the uncertain environment parameter ϕ (Eq. (2)). Therefore, Eq. (17) captures the robot dynamics change caused by the environment uncertainties.

B. Simulation Environment

In the simulation, a mobile manipulator is trying to enter a room from the entrance. The robot knows where the entrance is but do not know what type of entrance it is. In Fig. 2a, four different types of entrance are shown: a left-hand door, a revolving door, a right-hand door, and a entrance blocked by a light box (we assume this box is on a cart that only moves along the x -axis). These different types create uncertainties, and the robot will have to make multiple contacts with the environment to collect information and update its belief in each type so that it can plan optimally. To capture the uncertainties, we define the environment parameter as $\phi = [r_y, p_{rotate}]$, where r_y is the location of the rotation axis of the door and $p_{rotate} \in \{0, 1\}$ is a Boolean parameter that indicates whether the robot is pushing a door ($p_{rotate} = 1$) or pushing a box ($p_{rotate} = 0$). The environment state is the door open angle (or the location of the light box), i.e., $z_e = \theta_d$ (or $z_e = x_{box}$). We discretize the environment parameter space to four discrete values to match the four scenarios, i.e., $\phi \in \{\phi_1, \phi_2, \phi_3, \phi_4\}$. We initialize the robot to explore by contacting the center of the entrance.

C. Results

We test with two robot reward settings in the simulation and the results are shown in Fig. 2. In the first setting (Fig. 2b), the robot prioritize exploration of the uncertain types, i.e., the belief of different environment parameters; while in the second setting (Fig. 2c), the robot also considers the state rewards and tries to pass the entrance quickly. Comparing the two, we see the robot makes more contacts with the first setting and the beliefs are more concentrated on the correct type. However, sometimes the robot does not pass the entrance. In scenario 3, the robot continues to make short period of contact at different places. In scenario 4, the robot contacts the empty space between the box and the entrance.

Since the environment state does not change, the robot can be very sure that the scenario is a box because the environment state will change otherwise according to the environment dynamics. On the other hand, the robot successfully enters the room in all scenarios with a shorter time while running the second setting.

IV. EXPERIMENTAL RESULTS

A. Experimental Setups

Entering a room through a door. Like the simulation environment, the robot is trying to enter a room from the entrance. Here, the robot does not know whether the door is a left-hand door or a right-hand door. The door may also have a static friction that needs to be overcome at the beginning or it may be blocked by an object after a certain open angle. The environment parameter is $\phi = [r_y, p_{friction}, p_{push_able}]$, where r_y is the location of the rotation axis of the door and $p_{friction}, p_{push_able} \in \{0, 1\}$ are Boolean parameters that indicate whether the door has friction and whether it is pushable or not, respectively. The environment state is the door open angle, $z_e = \theta_d$. We have four scenarios: the robot can pass without adjustment, the robot needs to overcome friction, the robot can wiggle through the entrance, and the robot cannot enter. We initialize the robot to contact the center of the entrance.

Moving a battery chain. In this scenario, the robot is trying to move a battery chain (Fig. 4c) to a target location (The white area in Fig. 4c (10)). There are three different chains: a four-battery chain (Fig. 4c), a three-battery chain (Fig. 4b (middle)), and a chain that has one battery connected with two paper rolls (very light (Fig. 4b (left))). According to the environment dynamic model, the weight change will be different for different chains as the end-effector rises. The environment parameter is $\phi = [w, x_c, p_{pick_able}]$, where w is the weight of the chain, x_c is the center of mass of the chain, and $p_{pick_able} \in \{0, 1\}$ is a Boolean parameters that indicates whether the chain is pickable by the TB3O or not. The environment state is force applied at the end-effector, $z_e = F_{ee}$. The robot is initialized to first grasp the closest end of the chain.

In both experiments, the robot needs to contact and explore uncertainties while quickly complete its task.

B. Results

Some of the experimental results are shown in Fig. 3 and Fig. 4. (The video in the supplementary material includes more details of the experimental results.)

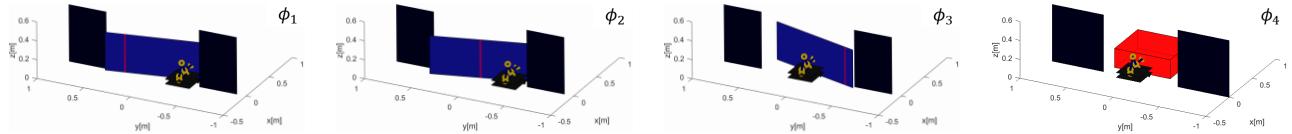
Entering a room through a door. The experimental result of scenario 2 is shown in Fig. 3a. The robot contacts the center of the door and encounters a static friction (Fig. 3a (1)). Since the reward of contacting the left and the right is symmetric, the robot randomly chooses to push the left (Fig. 3a (3)). However, contacting the left reduces the torque generated by the pushing action and the door still cannot be opened. Such an observation increases the robot's beliefs that the rotation axis is on the left or the door is not pushable. With the updated belief, the robot's optimal action is to

push the right side since such action can bring the maximum information gain and is more likely to open the door to enter the room (Fig. 3a (6)). This time the pushing action generates enough torque and the robot is able to open the door (Fig. 3a (7)~(10)). In scenario 3 (Fig. 3b), although the door is blocked midway and the robot soon believes that it is no longer pushable (Fig. 3b (2)), the robot also realizes that the open angle is large enough so it changes its strategy and wiggles through the entrance (Fig. 3b (3)~(5)). (Notice that once the door is opened, the robot can also detect whether the door is a left-hand door or a right-hand door by vision. Therefore, the robot knows it should navigate from the right side of the door after the door is blocked midway.) Results of scenario 1 and 4 are shown in the supplementary video.

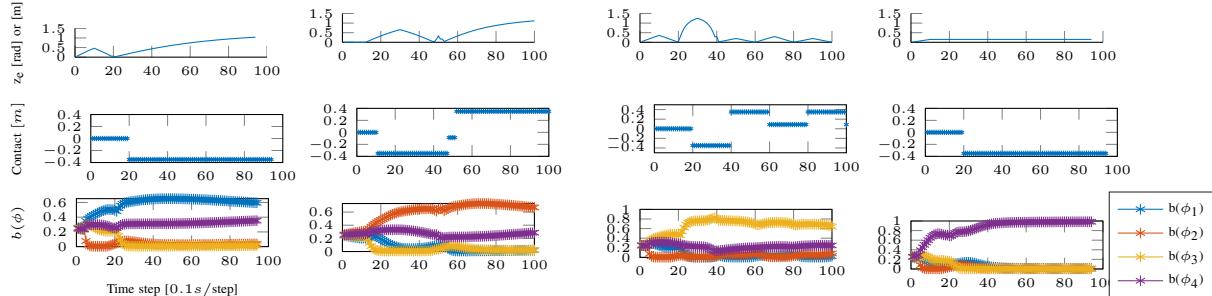
Moving a battery chain. The experimental result of scenario 3 (four-battery chain) is shown in Fig. 4c and the belief change in the experiment is shown in Fig. 4b (right). The robot starts with grasping and lifting the chain to a certain height (Fig. 4c (1)). Since all three chains will return the same measurement according to the environment dynamic model, the robot continues to rise the end-effector (Fig. 4c (2)). Now, the weight increases significantly and the robot realizes that the chain is heavy and the center of mass is far from the current grasping point, so the robot moves the grasping location closer to the middle of the chain (Fig. 4c (4)). In order to further explore whether the chain can be lifted, the robot again rises the end-effector higher (Fig. 4c (5)) and realized the chain is most likely the four-battery chain, which is too heavy to be completely lifted from the floor. Therefore, the alternative strategy has the biggest reward and the robot uses the base to push the chain to the target location (Fig. 4c (6)~(10)). In scenario 1 (battery-paper roll chain), the robot believes that the chain is light because the weight does not increase much as the end-effector rises (Fig. 4a (left)(2)), therefore it lifts the chain from the floor (Fig. 4b (left)) and moves the chain to the target location. In scenario 2 (three-battery chain), the robot first realizes the center of mass is far from the first grasping location (Fig. 4a (right)(II)), therefore it moves the grasping location and starts to lift the chain again. By utilizing the weight change information during lifting, the robot believes it can pick up the chain with a different pose (Fig. 4a (right)(IV)), Fig. 4b (middle)). Thus, the robot again picks the chain up and puts it at the target location.

V. CONCLUSION

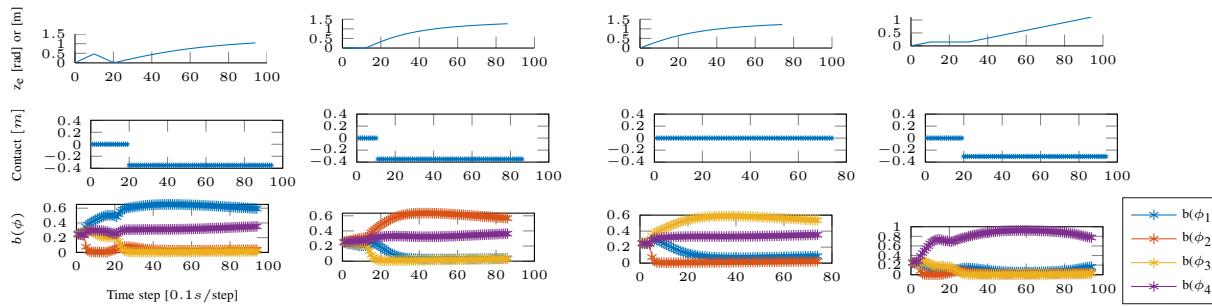
This paper presented a POMDP formulation that allowed light-weighted mobile manipulators to actively collect uncertainty information through physical contact and generate more feasible motion plans by explicitly integrating such information into motion planners. The planning problem encouraged the robots to take actions with high expected information gains as well as state rewards. Such a framework can efficiently help mobile manipulators to handle dynamic changes and possible failures caused by environment uncertainties. Both simulation and experiments were conducted to evaluate the performance of the proposed method. The



(a) Four different environments described by the four environment parameters in the simulation. (Red line indicates the location of the rotation axis.)

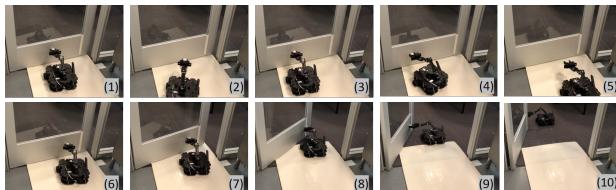


(b) Simulation result with robot prioritizing uncertainty exploration.



(c) Simulation result of robot exploring the environment while managing to pass the entrance.

Fig. 2: Simulation result of mobile manipulator in uncertain environments. (The x -axis of figures in (b) and (c) indicates time step. The first row in (b) and (c) shows the environment state z_e , the second row shows the contact point w.r.t. the y -axis, and the third row shows the change of the belief: $b(\phi_1)$ (blue), $b(\phi_2)$ (orange), $b(\phi_3)$ (yellow), and $b(\phi_4)$ (purple).)



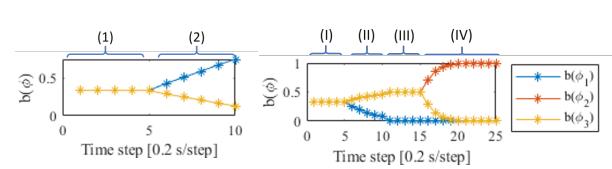
(a) Experimental result with TB3O overcoming the initial friction force and opens the door successfully.



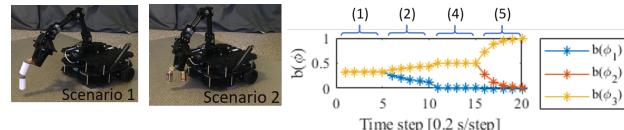
(b) Experimental result with TB3O navigating through the entrance when the door is blocked.

Fig. 3: Planning result of TB3O entering a room through a door.

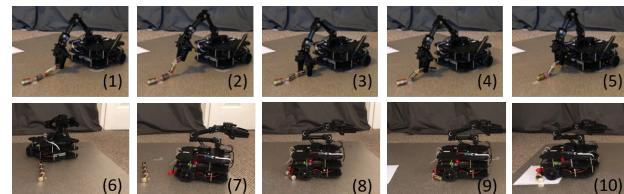
results showed that mobile manipulators with the proposed active exploration and belief update can handle environment uncertainties more flexibly and correspondingly complete tasks more efficiently. In the future, we will utilize not only physical contact but also other methods to explore uncertainties in more general application scenarios to further extend the capability of mobile manipulators.



(a) The belief change in scenario 1 (left) and scenario 2 (right).



(b) The grasping pose before transporting the chain in scenario 1 (left) and 2 (middle). The belief change in scenario 3 (right).



(c) Experimental result with TB3O moving a heavy four-battery chain (scenario 3).

Fig. 4: Experimental result of TB3O in moving-object scene.

REFERENCES

- [1] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.
- [2] L. Sun, W. Zhan, C.-Y. Chan, and M. Tomizuka, "Behavior planning of autonomous cars with social perception," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 207–213.
- [3] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.
- [4] A. Roncone, O. Mangin, and B. Scassellati, "Transparent role assignment and task allocation in human robot collaboration," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1014–1021.
- [5] M. Posa and R. Tedrake, "Direct trajectory optimization of rigid body dynamical systems through contact," in *Algorithmic foundations of robotics X*. Springer, 2013, pp. 527–542.
- [6] B. Saund and D. Berenson, "Motion planning for manipulators in unknown environments with contact sensing uncertainty," in *International Symposium on Experimental Robotics*. Springer, 2018, pp. 461–474.
- [7] "Turtlebot3 - robotis e-manual," <http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>, accessed: 2020-02-27.
- [8] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.
- [9] A. Vemula, Y. Oza, J. A. Bagnell, and M. Likhachev, "Planning and execution using inaccurate models with provable guarantees," *arXiv preprint arXiv:2003.04394*, 2020.
- [10] N. Meuleau, C. Plaunt, D. Smith, and T. Smith, "A pomdp for optimal motion planning with uncertain dynamics," in *ICAPS-10: POMDP Practitioners Workshop*. Citeseer, 2010.
- [11] M. Kwon, S. H. Huang, and A. D. Dragan, "Expressing robot incapability," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 87–95.
- [12] U. Ehsan, P. Tambwekar, L. Chan, B. Harrison, and M. O. Riedl, "Automated rationale generation: a technique for explainable ai and its effects on human perceptions," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019, pp. 263–274.
- [13] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 66–73.
- [14] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2555–2565.
- [15] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and machine theory*, vol. 37, no. 10, pp. 1213–1239, 2002.
- [16] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
- [17] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme," *The International Journal of Robotics Research*, vol. 29, no. 6, pp. 727–747, 2010.
- [18] H.-H. Li and W. J. Ma, "Confidence reports in decision-making with multiple alternatives violate the bayesian confidence hypothesis," *Nature Communications*, vol. 11, no. 1, pp. 1–11, 2020.
- [19] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning," *SIAM Journal on Control and Optimization*, vol. 56, no. 4, pp. 2712–2733, 2018.
- [20] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Pomdps for robotic tasks with mixed observability," in *Robotics: Science and systems*, vol. 5, 2009, p. 4.
- [21] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cassandra, "Solving pomdps by searching the space of finite policies," *arXiv preprint arXiv:1301.6720*, 2013.
- [22] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.
- [23] C. Park, F. Rabe, S. Sharma, C. Scheurer, U. E. Zimmermann, and D. Manocha, "Parallel cartesian planning in dynamic environments using constrained trajectory planning," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 983–990.
- [24] M. Betancourt and M. Girolami, "Hamiltonian monte carlo for hierarchical models," *Current trends in Bayesian methodology with applications*, vol. 79, no. 30, pp. 2–4, 2015.
- [25] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.