

A Probabilistic Model for Planar Sliding of Objects with Unknown Material Properties: Identification and Robust Planning

Changkyu Song and Abdeslam Boularias¹

Abstract—This paper introduces a new technique for learning probabilistic models of mass and friction distributions of unknown objects, and performing robust sliding actions by using the learned models. The proposed method is executed in two consecutive phases. In the exploration phase, a table-top object is poked by a robot from different angles. The observed motions of the object are compared against simulated motions with various hypothesized mass and friction models. The simulation-to-reality gap is then differentiated with respect to the unknown mass and friction parameters, and the analytically computed gradient is used to optimize those parameters. Since it is difficult to disentangle the mass from the friction coefficients in low-data and quasi-motion regimes, our approach retains a set of locally optimal pairs of mass and friction models. A probability distribution on the models is computed based on the relative accuracy of each pair of models. In the exploitation phase, a probabilistic planner is used to select a goal configuration and waypoints that are stable with a high confidence. The proposed technique is evaluated on real objects and using a real manipulator. The results show that this technique can not only identify accurately mass and friction coefficients of non-uniform heterogeneous objects, but can also be used to successfully slide an unknown object to the edge of a table and pick it up from there, without any human assistance or feedback.

I. INTRODUCTION

Pre-grasp sliding manipulation of objects is a useful skill that is necessary when an object is too thin relative to the size of a robotic hand to be directly grasped from a flat surface. This skill is also useful for the execution of tool-use grasps where the orientation of an object needs to be modified before picking up the object. For example, a pre-grasp sliding manipulation approach that was recently developed [1] consists in performing a sequence of non-prehensile actions such as side-pushing and top-sliding to relocate an object to the edge of a table so that part of the object sticks out of the table. The object could then be grasped by a robotic hand without colliding with the table.

Most existing methods for pre-grasp sliding manipulation require the existence of predefined geometric and mechanical models of the target object [2]–[7]. Such models are necessary for selecting appropriate pushing forces and ensuring the balance of the object when it is moved to the edge of the support surface. In this work, we relax this assumption and consider the general problem of side-pushing an unknown object toward a stable goal configuration.

The proposed approach consists in first identifying the mass distribution of the target object, and its coefficients of friction with the support surface, from observed motions and

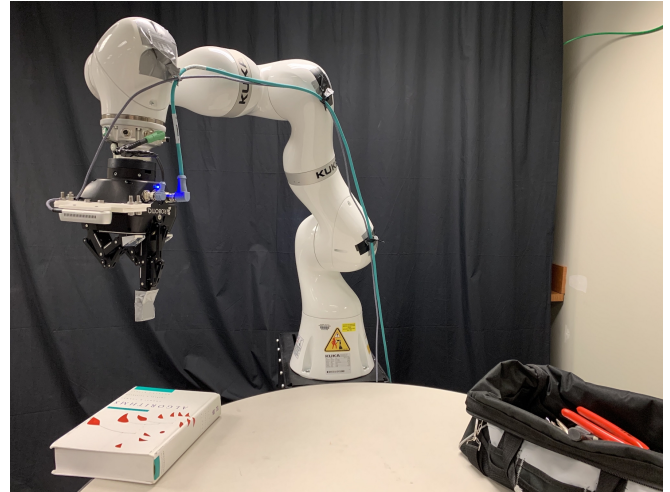


Fig. 1. Robotic setup used in the experiments. In this scenario, the robot pushes a book into a desired final pose through consecutive pushes. The choice of contact points and pushing directions is non-trivial as it depends on the mass distribution and the friction coefficients of different parts of the object, which are unknown *a priori*.

rotations of the object. While there are many techniques for identifying mechanical properties of objects from data [2], [8]–[15], existing works identify only the friction parameters, and assume that the mass distribution is known. Moreover, several existing methods assume that the bottom surface of the object has a uniform homogenous friction everywhere. In this work, we consider a grid-based representation of the target object. The object is modeled as a large set of small cuboids that are attached to each other. The mass and coefficient of friction of each cuboid can be different from the others. The analytical gradient of the object’s motion is then used to learn the mechanical properties of each cuboid.

The problem of identifying the mass distribution and friction coefficients of an unknown object is under-determined. From the small number of data points collected by the robot by pushing the object, there is a large subspace of possible mass and friction values that can be identified and that explain the observed motions equally well. This problem is particularly pronounced in quasi-static motions, where the mass of the object cannot easily be disentangled from its friction with the support surface. To solve this problem, the full spectrum of possible mass and friction values should be retained. Therefore, our method returns a probability distribution of the mass and friction parameters that can be used to deal with this problem of uncertainty. In a nutshell, we perform several gradient-descent searches

¹The authors are with the Department of Computer Science of Rutgers University, Piscataway, New Jersey 08854, USA. {cs1080, ab1544}@cs.rutgers.edu

starting from different random initial points, and retain all the local minima. A soft-max function is then applied on the values of the local minima to obtain a probability distribution on possible mass and friction values.

The obtained distribution is used, in a second phase, for robust probabilistic planning. Planning here refers to the problem of finding a stable goal configuration and a sequence of side-pushing actions that displace the object to the goal while preserving its balance. Probabilistic planning consists in searching for actions in a *multiverse* simulation, wherein next states are sampled from various models of mass and friction of the object, with different probabilities. The sequence of actions that has the highest success probability is then selected and executed. The proposed technique for probabilistic model identification and manipulation planning is evaluated both in simulation and on real objects and robot. The results clearly confirm the advantage of using this technique to safely manipulate unknown objects.

II. RELATED WORK

The *mechanics of pushing* was explored in several past works [2], [8]–[15]. Notably, Mason [8] studied the rotation and translation of an object pushed at a single point contact. Lynch and Mason [9] also presented an algorithm for stable pushing when objects remain in contact with an end-effector. Yoshikawa and Kurisu [15] proposed a regression method for identifying pivot points of a pushed object. A similar setup was considered in [10] with a constraint to ensure positive friction coefficients. The *limit surface* is a convex set of all friction wrenches that can be applied on an object. It is generally approximated as an ellipsoid [11], or a higher-order polynomial [12]–[14]. An ellipsoid approximation was also used in a planar sliding model to simulate the motion of a pushed object and to perform a *push-grasp* [2]. In contrast with our method, these works identify only the friction parameters, and assume that the mass distribution is known. Moreover, the present work focuses on safe planning when the identified mass and friction models are uncertain due to the small number of actions used to identify the models in a quasi-static regime. Several *natively differentiable physics engines* have been proposed recently [16]. For example, a combination of a learned and a differentiable engine was used to learn to predict motions of planar objects [17]. Differentiable physics engines were also used in a planning algorithm to manipulate tools [18]. A differentiable contact model was used in [19] to allow for optimization of several locomotion and manipulation tasks. A differentiable physical simulation, formulated as a Linear Complementary Problem (LCP), was also recently proposed [20].

III. PROBLEM SETUP AND NOTATIONS

We revisit the problem of pre-grasp sliding manipulation presented in [1]. In this problem, the robot's task consists in grasping a rigid object placed on a table. The object cannot be easily grasped, so the robot slides it to the table's edge where part of it sticks out of the table and becomes exposed. The robot can then easily grasp it from there. In

this work, we consider the same problem, but for objects with unknown shape and material properties. The problem then consists in 1) selecting pushing actions to explore the object and gather data points about it, 2) inferring the mass and friction distributions of the object from the observed motions, 3) selecting a stable goal configuration x_T^d from which the object can be grasped, and 4) planning a sequence of side-pushing actions that displace the object from its current pose x_0 to the desired final pose x_T^d .

The object is represented as a set of small cuboids. The object is thus divided into a large number of connected cells $1, 2, \dots, n$, using a regular grid structure, wherein each cell i has its own local mass and coefficient of friction. The object's pose x_t at time $t \in [0, T]$ is given as a vector $x_t = [p_{x,t}^1, p_{y,t}^1, \theta_t^1, \dots, p_{x,t}^n, p_{y,t}^n, \theta_t^n]^T$, where $(p_{x,t}^i, p_{y,t}^i)$ is the i^{th} cell's 2D position on the surface, and θ_t^i is its angle of rotation. The object's generalized velocity at time t is denoted as $\dot{x}_t = [\dot{p}_{x,t}^i, \dot{p}_{y,t}^i, \dot{\theta}_t^i]_{i=0}^n$. The object's mass matrix \mathcal{M} is a diagonal $3n \times 3n$ matrix, where the diagonal is $[\mathcal{I}_1, \mathcal{M}_1, \mathcal{M}_1, \mathcal{I}_2, \mathcal{M}_2, \mathcal{M}_2, \dots, \mathcal{I}_n, \mathcal{M}_n, \mathcal{M}_n]$, \mathcal{I}_i is the moment of inertia of the i^{th} cell of the object, and \mathcal{M}_i is its mass. $\mathcal{I}_i = \frac{1}{6}\mathcal{M}_i w^2$ where w is the width of a cuboid. μ is a $3n \times 3n$ diagonal matrix, where the diagonal is $[\mu_1, \mu_1, \mu_1, \mu_2, \mu_2, \mu_2, \dots, \mu_n, \mu_n, \mu_n]$. μ_i is the coefficient of friction between the i^{th} cell of the object and the surface.

An external generalized force denoted by F is an $1 \times 3n$ vector $[f_x^1, f_y^1, \tau^1, \dots, f_x^n, f_y^n, \tau^n]^T$, where $[f_x^i, f_y^i]$ and τ^n are respectively the force and torque applied on cell i . External forces are generated from the contact between the object and a fingertip of the robotic hand used to push the object. We assume that at any given time t , at most one cell of the object is in contact with the fingertip.

A ground-truth trajectory \mathcal{T}^g is a state-action sequence $(x_0^g, \dot{x}_0^g, F_0, \dots, x_{T-1}^g, \dot{x}_{T-1}^g, F_{T-1}, x_T^g, \dot{x}_T^g)$, wherein (x_t^g, \dot{x}_t^g) is the observed pose and velocity of the pushed object, and F_t is the external force applied at time t , as defined above. A corresponding simulated trajectory \mathcal{T} is obtained by starting at the same initial state \hat{x}_0 in the corresponding real trajectory, i.e., $\hat{x}_0 = x_0$, and applying the same control sequence $(F_0, F_1, \dots, F_{T-1})$. Thus, the simulated trajectory \mathcal{T} results in a state-action sequence $(x_0^g, \dot{x}_0^g, F_0, x_1, \dot{x}_1, F_1, \dots, x_{T-1}, \dot{x}_{T-1}, F_{T-1}, x_T, \dot{x}_T)$, where $x_{t+1} = x_t + \dot{x}_t dt$ is the predicted next pose. Velocity \dot{x}_t is a vector corresponding to translation and angular velocities in the plane for each of the n cells, it is predicted in simulation as $\dot{x}_{t+1} = V(x_t, \dot{x}_t, F_t, \mathcal{M}, \mu)$. The goal is to identify mass distribution \mathcal{M} and friction map μ that result in simulated trajectories that are as close as possible to the real observed ones. Therefore, the objective is to solve the following optimization problem,

$$(\mathcal{M}^*, \mu^*) = \arg \min_{\mathcal{M}, \mu} \text{loss}(\mathcal{M}, \mu), \quad (1)$$

$$\text{loss}(\mathcal{M}, \mu) \stackrel{\text{def}}{=} \sum_{t=0}^{T-2} \|x_{t+2}^g - (x_{t+1}^g + V(x_t^g, \dot{x}_t^g, F_t, \mathcal{M}, \mu)dt)\|_2.$$

Since x_t is a vector containing all cells' positions, the loss is the sum of distances between each cell's ground-truth pose and its predicted pose, which is equivalent to the average distance (ADD) metric as proposed in [21].

IV. DETERMINISTIC MASS AND FRICTION IDENTIFICATION

In [22], it has been shown that the gradient of the loss function in Equation 1 with respect to mass parameters \mathcal{M} and friction map μ can be computed analytically, and used to search for ground-truth values (\mathcal{M}^*, μ^*) by using the gradient-descent method. The algorithm proposed in [22] consists in initializing (\mathcal{M}, μ) with arbitrary positive values, and repeatedly use (\mathcal{M}, μ) to generate a simulated trajectory $(x_0^g, \dot{x}_0^g, F_0, x_1, \dot{x}_1, F_1, \dots, x_{T-1}, \dot{x}_{T-1}, F_{T-1}, x_T, \dot{x}_T)$, which will then be used to update (\mathcal{M}, μ) as follows:

$$\begin{aligned}\mu &\leftarrow \mu + \sqrt{\alpha_{\text{rate}}} \sum_{t=0}^{T-2} \text{Diag}(x_{t+2} - x_{t+2}^g), \\ \mathcal{M} &\leftarrow \mathcal{M} + \alpha_{\text{rate}} \sum_{t=0}^{T-2} \text{Diag}(x_{t+2} - x_{t+2}^g),\end{aligned}\quad (2)$$

wherein α_{rate} is a predefined step-size. While this approach is shown to return an accurate map of the frictional forces on the surface of a homogeneous object, it cannot disentangle the friction from the mass if all the data points correspond to quasi-static motions. In other terms, the product mass \times friction of the identified mass and friction of each cell in the object is close to its ground-truth value, and it is accurate enough to predict the translation and rotation of the object under various pushing actions. However, it cannot tell which part of the product mass \times friction is mass and which part is friction. For example, if the ground-truth friction force at some cell of the object is $\mathcal{M}_i^* \times \mu_i^* = 10g$ N, where $g = 9.8$, then the update equations 2 can typically identify a value for \mathcal{M}_i and for μ_i such that $\mathcal{M}_i \times \mu_i \approx \mathcal{M}_i^* \times \mu_i^* = 10g$ N and so that the translation and rotation of the object can be predicted accurately in quasi-static motions on a flat surface. But the identified individual values for mass and friction can be very far from the ground-truth values. For example, any pair of values from the set $\{(1kg, 98\frac{N}{kg}), (98kg, 1\frac{N}{kg}), (2kg, 49\frac{N}{kg}), (20kg, 4.9\frac{N}{kg}), \dots\}$ can be returned by the deterministic identification algorithm proposed in [22].

V. PROPOSED APPROACH

While the exact value of the mass of each cell of the object is relatively irrelevant in quasi-static motions, that is not the case when the object is pushed to the edge of the table and where one needs to predict if it remains stable there or if it drops, which is necessary for pre-grasp sliding manipulation. In the following we propose a solution to this problem by first presenting a probabilistic model identification approach that returns a distribution on joint mass and frictions maps, instead of single point estimates. We then show that the probabilistic model can be used for robustly selecting stable goal configurations and planning pre-grasp sliding manipulation actions.

A. Probabilistic Mass and Friction Model

We generalize the deterministic model presented in Section III into a probabilistic one. The probabilistic model is given by a set of mass and friction maps denoted by $\Theta = \{(\mathcal{M}^{(1)}, \mu^{(1)}), (\mathcal{M}^{(2)}, \mu^{(2)}), \dots, (\mathcal{M}^{(K)}, \mu^{(K)})\}$ and a probability measure $P : \Theta \rightarrow [0, 1]$. Each $(\mathcal{M}^{(k)}, \mu^{(k)})$ corresponds to one of K sampled models, and is defined as explained in Section III. Thus, $\mathcal{M}^{(k)}$ is a diagonal matrix, where the diagonal corresponds to the moment of inertia and the mass of each cell of the object, and $\mu^{(k)}$ is a diagonal matrix, where the diagonal corresponds to the coefficients of friction between each of the n cells of the object and the support surface. We assume that: $\forall k \in \{1, \dots, K\}, \forall i \in \{1, \dots, n\} : \mu_i^{(k)} \in [0, \mu_{\text{max}}]$, and $\mathcal{M}_i^{(k)} \in [0, \mathcal{M}_{\text{max}}]$ where μ_{max} and \mathcal{M}_{max} are given upper bounds. We will explain in the following how models in Θ are sampled and how their probabilities are computed.

B. Inference of Probabilistic Models

The proposed inference algorithm consists in collecting a real-world trajectory $(x_t^g, \dot{x}_t^g, F_t)_{t=0}^T$ of ground-truth object poses and velocities that result from a sequence of selected exploratory pushing forces $(F_t)_{t=0}^T$. Different models $(\mathcal{M}^{(k)}, \mu^{(k)})$ of mass and friction distributions are then returned by the algorithm, with different probabilities. Each one of the models is obtained by upper-bounding the mass and the friction with a different limit, and using the gradient-descent method to search for values of mass and friction that minimize the simulation-to-reality gap.

The main steps of this search process are explained in Algorithm 1. The set Θ of sampled models is initially empty. The first pushing action is selected arbitrarily as no prior information about the object's mass or friction distributions is available. The magnitudes of all the pushing actions are small to avoid moving the object out of the robot's workspace. At each iteration of the algorithm, an action is selected by calling the subroutine `ActionSelection` presented in Algorithm 2, which will be explained in the next subsection. The returned action F_T is applied on the object. The next pose and velocity of the object $(x_{T+1}^g, \dot{x}_{T+1}^g)$ is extracted from recorded depth images after removing the background.

The inner loop of the algorithm (lines 7-27) consists in setting different upper bounds for mass and friction and optimizing $(\mathcal{M}^{(k)}, \mu^{(k)})$ to produce simulated trajectories of the pushed object that are as close as possible to the recorded ground-truth. In the first half of the total number of iterations K , the upper bound of the mass per cell is fixed to $\mathcal{M}_{\text{max}} \in \mathbb{R}$ whereas the upper bound of the friction is gradually increased at each iteration. In the second half, the upper bound of the friction is fixed to $\mu_{\text{max}} \in \mathbb{R}$ whereas the upper bound of the mass per cell is gradually increased at each iteration. Other combinations of mass and friction upper bounds can also be considered as well. The provided upper limits are the same for every cell in the object, whereas the mass and friction maps learned by the algorithm are highly heterogenous, as will be shown in the experiments.

At each iteration k of the inner loop, the state of the object in simulation (x_0, \dot{x}_0) is reset to the ground-truth initial state (x_0^g, \dot{x}_0^g) (line 13). Both the mass matrix $\mathcal{M}^{(k)}$ and the friction matrix and $\mu^{(k)}$ are initialized to half their maximum values (lines 14-15). In the next steps (lines 17-25), $\mathcal{M}^{(k)}$ and $\mu^{(k)}$ are iteratively used to predict in simulation the trajectory of the manipulated object under the same forces $\{F_t\}_{t=0}^T$, and updated so that the difference between the predicted pose x_{t+1} is as close as possible to the observed ground-truth x_{t+1}^g . The sum of the differences over all time-steps is denoted by $loss^{(k)}$. The mass and friction matrices are updated by using the stochastic gradient of the loss (lines 21-22), wherein $Diag(x_{t+1} - x_{t+1}^g)$ is a diagonal matrix that contains $(x_{t+1} - x_{t+1}^g)$ in the main diagonal, in accordance with the definitions of \mathcal{M} and μ provided in Section III. The mass and friction are increased or decreased depending on the signs in the error vector $(x_{t+1} - x_{t+1}^g)$, which corresponds to the reality gap.

Finally, the models identified from the full sequence of pushing actions are weighed based on their respective accuracies, using a soft-max function (line 30). The soft-max operator returns a probability distribution over competing models of mass and friction distributions. Each of these models corresponds to a local minimum of the reality-to-simulation loss, obtained by gradient-descent. Despite the fact that most of these models achieve a very small loss in predicting the motion of the object on a flat surface, they vary significantly in the way they attribute the observed motion to the mass or to the friction coefficients. While this difference is not consequential on a flat surface, it is important for predicting the balance of the object when it is pushed all the way to the edge of the support surface.

C. Active Exploration

The data used to identify the material properties of the manipulated object is actively collected by the robot. Algorithm 2 summarizes three different strategies that can be followed for selecting the pushing forces. First, a finite set \mathcal{S} of candidate forces is generated by randomly sampling contact points from the object's contour, along with pushing directions and force magnitudes. The simplest strategy consists in returning a randomly selected force $F_T \in \mathcal{S}$. Another strategy selects the action that is most different from the previously executed forces $\{F_0, \dots, F_{T-1}\}$, to increase the diversity of the data and to poke different parts of the object. The third strategy consists in simulating the effect of each force $F \in \mathcal{S}$ on each possible (mass, friction) model from the set of identified models Θ . The action that leads to the highest disagreement between the models is then selected.

D. Robust Planning with Inferred Models

After a set Θ of possible (mass, friction) models is identified in the exploration phase by using Algorithm 1, Algorithm 3 is called to plan a new sequence of pushing actions that displace the object into a desired final pose x_T^d . We start by generating a sequence of intermediate poses, i.e., *waypoints*, by using the rapidly-exploring random tree

Algorithm 1: Inference of Probabilistic Mass and Friction Models with Differentiable Physics

Input: Learning rate α_{rate} ; Total number of pushing actions $nb_actions$; Upper bounds μ_{max} and \mathcal{M}_{max} ;
Output: A set Θ of mass and friction maps, and a corresponding probability distribution P ;

- 1 Sample a random contact point from the object's contour, and use the robot's fingertip to push the object from the sampled point with a small random force F_0 ; Record the observed new state (x_1^g, \dot{x}_1^g) ;
- 2 $\Theta \leftarrow \emptyset$;
- 3 **for** $T = 1, nb_actions$ **do**
- 4 $F_T \leftarrow \text{ActionSelection}(\Theta, \{F_0, \dots, F_{T-1}\})$;
- 5 Apply F_T ; Record the observed new state $(x_{T+1}^g, \dot{x}_{T+1}^g)$;
- 6 $\Theta \leftarrow \emptyset$;
- 7 **for** $k = 1, K$ **do**
- 8 **if** $k \leq \frac{K}{2}$ **then**
- 9 $\mu_{max}^{(k)} \leftarrow \frac{2k}{K} \mu_{max}$; $\mathcal{M}_{max}^{(k)} \leftarrow \mathcal{M}_{max}$;
- 10 **else**
- 11 $\mu_{max}^{(k)} \leftarrow \mu_{max}$; $\mathcal{M}_{max}^{(k)} \leftarrow (\frac{2k+1}{K} - 1) \mathcal{M}_{max}$;
- 12 **end**
- 13 $(x_0, \dot{x}_0) \leftarrow (x_0^g, \dot{x}_0^g)$; $x_1 \leftarrow x_1^g$;
- 14 $\mathcal{M}^{(k)} \leftarrow \frac{1}{2} I \mathcal{M}_{max}^{(k)}$;
- 15 $\mu^{(k)} \leftarrow \frac{1}{2} I \mu_{max}^{(k)}$;
- 16 $loss^{(k)} \leftarrow 0$;
- 17 **for** $t = 0, T-1$ **do**
- 18 $\hat{x}_{t+1} \leftarrow V(x_t, \hat{x}_t, F_t, \mathcal{M}^{(k)}, \mu^{(k)})$;
 ▷ Predicting velocity
- 19 $x_{t+2} \leftarrow x_{t+1} + \hat{x}_{t+1} dt$; ▷ Predicting next pose
- 20 $loss^{(k)} \leftarrow loss^{(k)} + \|x_{t+1} - x_{t+1}^g\|_2$;
- 21 $\mu^{(k)} \leftarrow \mu^{(k)} - \sqrt{\alpha_{rate}} Diag(x_{t+1} - x_{t+1}^g)$;
- 22 $\mathcal{M}^{(k)} \leftarrow \mathcal{M}^{(k)} - \alpha_{rate} Diag(x_{t+1} - x_{t+1}^g)$;
- 23 $\mu^{(k)} \leftarrow \arg \min_{\mu' \in [0, I \mu_{max}^{(k)}]} \|\mu' - \mu^{(k)}\|_\infty$;
- 24 $\mathcal{M}^{(k)} \leftarrow \arg \min_{\mathcal{M}' \in [0, I \mathcal{M}_{max}^{(k)}]} \|\mathcal{M}' - \mathcal{M}^{(k)}\|_\infty$;
 ▷ Projecting the gradients
- 25 **end**
- 26 $\Theta \leftarrow \Theta \cup \{(\mathcal{M}^{(k)}, \mu^{(k)})\}$;
- 27 **end**
- 28 **end**
- 29 **for** $k = 1, K$ **do**
- 30 $P(\mathcal{M}^{(k)}, \mu^{(k)}) = \frac{e^{-\frac{1}{\tau} loss^{(k)}}}{\sum_{i=1}^K e^{-\frac{1}{\tau} loss^{(i)}}}$;
- 31 **end**

(*RRT**) algorithm. Our algorithm then generates a sequence of pushing forces $(F_t)_{t=0}^{T-1}$ that track the waypoints. At each step in the sequence, a force F_t is selected and optimized so that, when executed, the resulting pose of the object is as close as possible to the nearest waypoint. There are several heuristics that can be used to select a force F_t that moves the object to the nearest waypoint x_{target} at time-step t . In this work, we use the following strategy. We start by computing the expected center of mass of the object as, $\hat{x} = \sum_{k=1}^K P(\mathcal{M}^{(k)}, \mu^{(k)}) \frac{\sum_{i \in \{1, \dots, n\}} (\mathcal{M}_i^{(k)}) (p_{x,t}^i, p_{y,t}^i)}{\sum_{i \in \{1, \dots, n\}} \mathcal{M}_i^{(k)}}$. We then find a cell i from the object's outer envelope as $i = \arg \max_{i \in \{1, \dots, n\}} \frac{(\hat{x} - x_{target}) \cdot ((p_{x,t}^i, p_{y,t}^i) - \hat{x})^T}{\|\hat{x} - x_{target}\|_2 \|(p_{x,t}^i, p_{y,t}^i) - \hat{x}\|_2}$. In other

Algorithm 2: ActionSelection($\Theta, \{F_0, \dots, F_{T-1}\}$)

```

1 Create a set  $\mathcal{S}$  of random candidate forces applied on
  random contact points in the object's contour;
2 if SelectionMode = random then
3    $F_T \leftarrow \text{RandomSample}(\mathcal{S})$ ;
4 end
5 if SelectionMode = mostDifferent then
6    $F_T \leftarrow \arg \max_{F \in \mathcal{S}} \min_{t \in [0, T-1]} \|F_t - F\|_2$ ;
7 end
8 if SelectionMode = mostDistinctive then
9    $D_{max} \leftarrow 0$ ;
10  foreach  $F \in \mathcal{S}$  do
11     $D \leftarrow 0$ ;
12    foreach  $(\mathcal{M}^{(i)}, \mu^{(i)}), (\mathcal{M}^{(j)}, \mu^{(j)}) \in \Theta$  do
13       $D \leftarrow D + \|V(x_0, \dot{x}_0, F, \mathcal{M}^{(i)}, \mu^{(i)})$ 
14         $- V(x_0, \dot{x}_0, F, \mathcal{M}^{(j)}, \mu^{(j)})\|_2$ ;
15    end
16    if  $D > D_{max}$  then
17       $D_{max} \leftarrow D$ ;  $F_T \leftarrow F$ ;
18    end
19  end
20 end
21 return  $F_T$ ;

```

terms, i is the outer cell that is most aligned to the axis $\hat{x} - x_{target}$. The direction of the pushing force is always selected as the opposite of the surface normal at the contact point. The next step consists in simulating the pushing force and predicting the next state of the object (line 12) according to each (mass, friction) model in Θ . The expected gap (or cost) of the simulated action is the weighted average of the distances between predicted poses, according to different models, and the target pose x_{target} , wherein the weights are the probabilities of the various models. This process is repeated by perturbing the location of the contact point, and retaining at the end the one that yields the minimum gap from the target pose.

E. Application: Pre-Grasp Sliding Manipulation

The proposed probabilistic model identification technique is used in Algorithm 4 to perform pre-grasp sliding manipulation of unknown objects. The algorithm integrates the previous ones to solve the problem of pushing an unknown object to the edge of a table and grasping it safely from there. We assume that the goal region is known. First, we start by using Algorithm 1 to explore the object and identify a set Θ of possible (mass, friction) models and their probabilities P . We then search for a configuration in the goal region that satisfies two criteria: 1) the object can be grasped without colliding with the support surface, and 2) the object is stable with a sufficiently high probability. To this end, a large number of object poses in the goal region are sampled and verified. Sampled poses that do not allow for a collision-free form-closure grasp are eliminated. Also, for each sampled pose, we use the identified mass models to

Algorithm 3: Robust Planning with Inferred Models

```

Input: A set  $\Theta$  of mass and friction maps, and a
  corresponding probability distribution  $P$ ;
Output: A sequence of pushing forces  $(F_t)_{t=0}^{T-1}$ ;
1 Find a set  $\mathcal{X}_{waypoint}$  of waypoints by calling
   $RRT^*(x_0, x_T^d)$ ;
2  $\dot{x}_0 \leftarrow \mathbf{0}$ ;  $t \leftarrow 0$ ;
3 while  $\mathcal{X}_{waypoint} \neq \emptyset$  do
4    $x_{target} = \arg \min_{x \in \mathcal{X}_{waypoint}} \|x - x_t\|_2$ ;
5   repeat
6      $gap_{min} \leftarrow \|x_0 - x_T^d\|_2$ ;
7     repeat
8       Select cell  $i$  from the contour of the object;
9       Choose the direction of the force  $(f_x^i, f_y^i)$  as the
        opposite of the object's surface normal at cell  $i$ ;
         $F^i \leftarrow [0, 0, 0, \dots, f_x^i, f_y^i, 0, \dots, 0, 0, 0]$ ;
         $gap \leftarrow 0$ ;
10      foreach  $(\mathcal{M}^{(k)}, \mu^{(k)}) \in \Theta$  do
11         $gap \leftarrow gap + P(\mathcal{M}^{(k)}, \mu^{(k)}) \|x_t + \dot{x}_t dt +$ 
12           $V(x_t, \dot{x}_t, F^i, \mathcal{M}^{(k)}, \mu^{(k)}) dt - x_{target}\|_2$ ;
13      end
14      if  $gap \leq gap_{min}$  then
15         $gap_{min} \leftarrow gap$ ;
16         $F_t \leftarrow [0, 0, 0, \dots, f_x^i, f_y^i, 0, \dots, 0, 0, 0]$ ;
17      end
18      until timeout;
19       $\dot{x}_{t+1} \leftarrow V(x_t, \dot{x}_t, F_t, \mathcal{M}, \mu)$ ;  $x_{t+1} \leftarrow x_t + \dot{x}_t dt$ ;
20       $t \leftarrow t + 1$ ;
21      until  $\min_{x \in \mathcal{X}_{waypoint}} \|x - x_t\|_2 \leq \epsilon$ ;
22       $\mathcal{X}_{waypoint} \leftarrow$ 
         $\mathcal{X}_{waypoint} \setminus \{\arg \min_{x \in \mathcal{X}_{waypoint}} \|x - x_t\|_2\}$ ;
23 end

```

predict in simulation the stability of the object. The failure probability is computed by adding together the probabilities of all mass models where the object is predict to fall from the table. Only goal configurations that ensure the stability of the object with a probability higher than $1 - \epsilon$ are selected. The robust planning algorithm 3 is then used to select a sequence of pushing actions to displace the object to the selected goal configuration.

VI. EXPERIMENTS

We report here the results of experiments on pre-grasp sliding manipulation both in simulation and using a real robot and objects to evaluate the proposed approach. A video of the experiments is attached as a supplementary material.

A. Experimental Setup

The experiments are performed on both simulated and real robot and six objects: a book, a hammer, a box, a ranch, a snack, and a spray gun. The simulation experiments are performed using the physics engine *Bullet* and models of the robot and objects. In the real setup, we used a *Kuka* robot with a *Robotiq* 3-finger hand and an RGB-D camera.

B. Compared Methods

The proposed algorithm is compared against the following methods. Planning with Uniform Mass Distribution plans each object's trajectory assuming

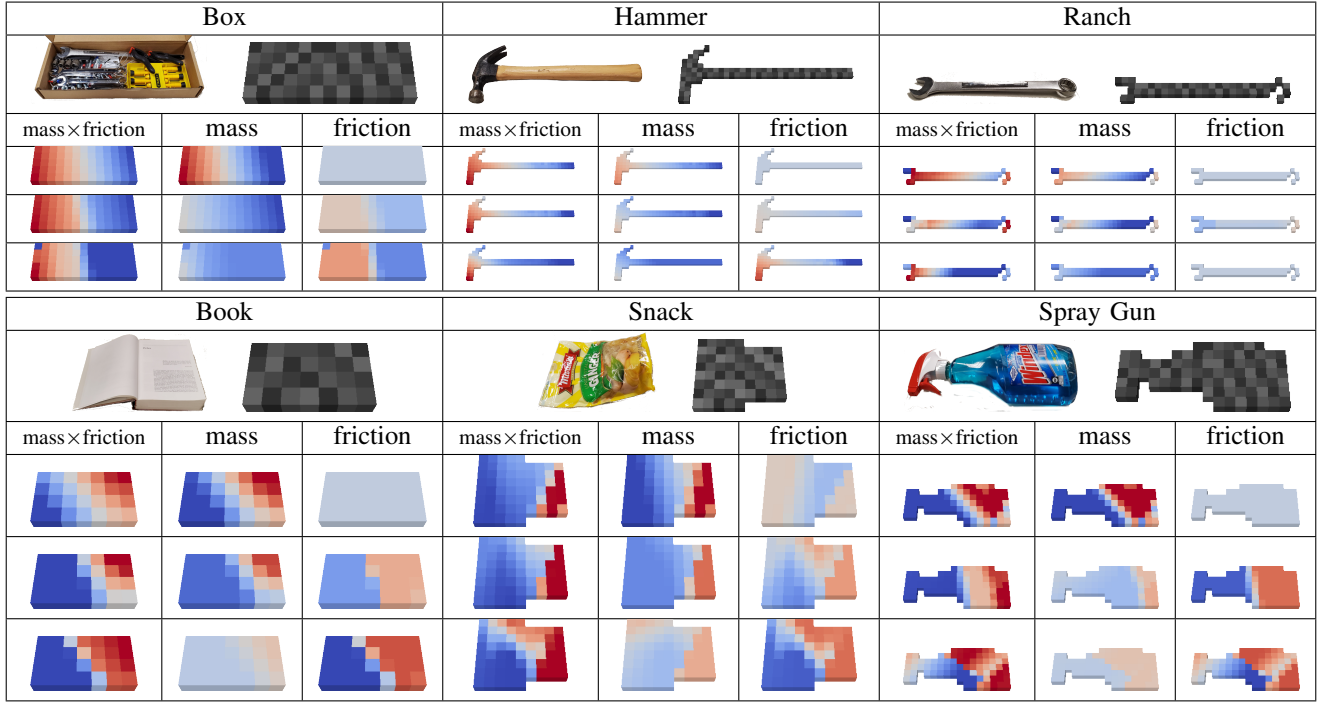


Fig. 2. Cuboid representations of real objects, along with samples from distribution P of joint models returned by Algorithm 1 using a real robot to collect data. Red indicates higher values. Note how different combinations of mass and friction maps can result in the same frictional forces.

Algorithm 4: Pre-Grasp Sliding Manipulation of Unknown Objects

Input: Point cloud of an object’s upper surface; Maximum mass and friction $m_{max}, u_{max} \in \mathbb{R}$; Desired goal region \mathcal{G} ; Failure probability threshold ϵ ;

- 1 Create a 3D shape \mathcal{S} of the object by projecting its upper surface down on the support surface;
 - 2 Decompose the 3D shape into a grid of n small cuboids;
 - 3 Let x_0^g be the vector of ground-truth positions and rotations of the n cuboids at time 0; $\dot{x}_0^g = \mathbf{0}$;
 - 4 Use Algorithm 1 to manipulate the object and identify a set Θ of possible joint mass and friction models and their corresponding probabilities P ;
 - 5 Sample a goal configuration $x^d \in \mathcal{G}$ where 1) a form-closure grasp of the object is available, and 2) the probability that the object remains stable under gravity, according to the identified mass distribution, is higher than a threshold $1 - \epsilon$;
 - 6 Use Algorithm 3 to obtain a new sequence of forces $(F_t)_{t=0}^{T_{\text{execution}}}$ to push the object to x^d ;
 - 7 Execute the sequence $(F_t)_{t=0}^{T_{\text{execution}}}$ with the robotic hand;
 - 8 Apply a form-closure grasp to pick up the object;
-

it has a uniform mass and friction. Thus, each cuboid in the object has the same mass and coefficient of friction as the other cuboids. Deterministic Planning with Identified Mass Distribution uses Algorithm 1 to learn the material properties from the same data as our method, but samples one model of mass and friction $(\mathcal{M}^{(k)}, \mu^{(k)}) \sim P$, and plans the object’s entire trajectory based on the sampled model. All the results of this method are averaged over ten samples of $(\mathcal{M}^{(k)}, \mu^{(k)}) \sim P$. Note

that each of the sampled models is already locally optimal and yields accurate predictions of the sliding motions. Because ground-truth mass and friction are known in the simulation experiments, we also compare the proposed algorithm with Planning with Ground-truth Mass Distribution which provides an upper bound.

C. Model Identification Results with Real Robot and Objects

Figure 2 shows qualitative results obtained from Algorithm 1 and five pushing actions per object. Each object is represented by a set of cuboids shown in the first rows. The number of cells per object varies from 28 to 88 depending on the size of the object. Notice that models of the same object have different mass and friction maps, however, they produce similar frictional forces given by the product of the mass with the friction at each cell. This ambiguity justifies the need for probabilistic reasoning about the models to avoid pushing the object into unbalanced configurations.

D. Balance Prediction with Real Objects

Figure 5 evaluates the accuracy of the proposed algorithm in selecting goal configurations based on the stability of the object. We manually placed each of the six real objects in 10 random poses on the edge of a round table where the form-closure grasp of the object is available. And we recorded whether the objects fell or stayed on the table in each pose. Planning with Uniform Mass Distribution and Deterministic Planning with Identified Mass Distribution returned any pose that is predicted to be stable in simulation using the identified or provided models. The proposed algorithm returns any pose where the object is predicted to remain balanced with a probability

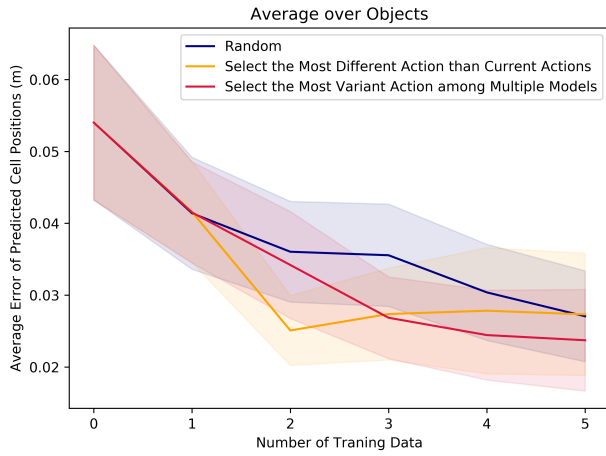


Fig. 3. The average predicted cell position error (in meters) as a function of the number of training data. The results indicate that active action selection does not offer any clear advantage over a random strategy in this task.

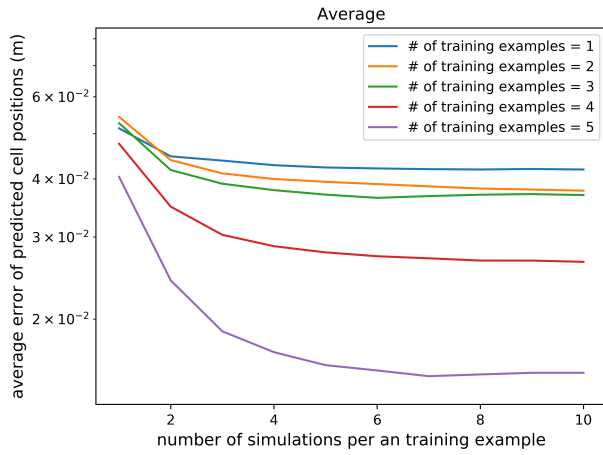


Fig. 4. The average predicted cell position error (in meters) with different numbers of training examples.

higher than the threshold of 0.90. The reported results are averaged over 100 independent trials per object. The proposed probabilistic method obtained the highest success rate, because it considers many possible mass models, with different probabilities, and selects the safest goal poses. The other methods select one mass model per trial for predicting the balance of the object. Again, note that all the mass and friction models here are obtained from the same identification process, and they are all local minima of the loss function with virtually similar objective values.

E. Pre-grasp Sliding Manipulation

1) *Simulation Experiments:* Each of the objects is placed in 100 random rotations at the center of a round table with a radius of 0.5m. After identifying the models of each object from five pushing actions, the goal is to select a sequence of pushing actions that displace the object into graspable goal configurations at the edge of the table. To make the experiments more challenging, we added a control noise to

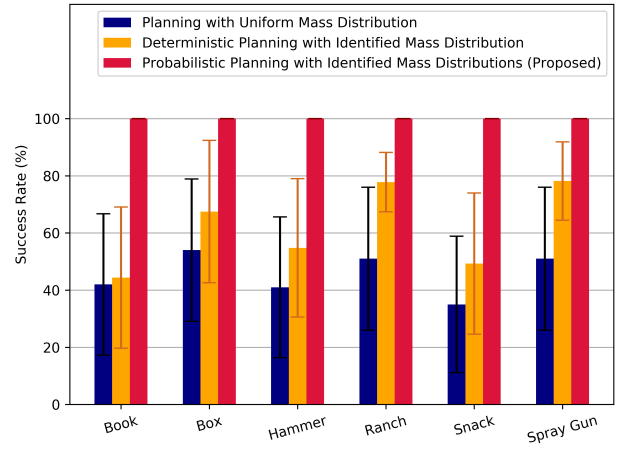
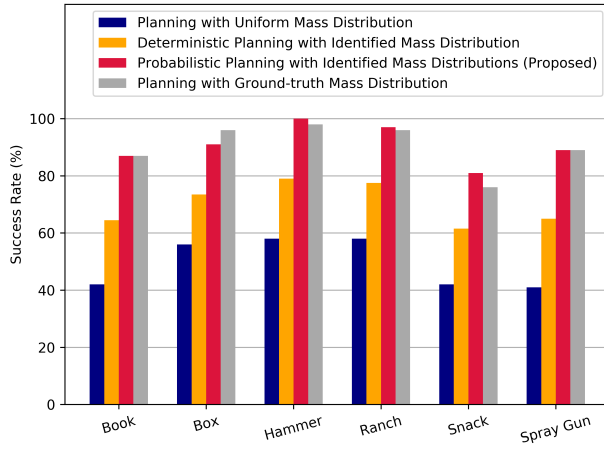


Fig. 5. Percentages of stable goal configurations selected by different methods using real robot and objects

the simulator so that planning does not perfectly match with execution. Figure 6 (a) shows that in all of the 100 trials with random initial object poses, the proposed method successfully placed the object in graspable and balanced poses. *Planning with Uniform Model Identification* failed mostly because it chose unstable poses as a goal due to the fact that most objects do not have a uniform mass. *Deterministic Planning with Identified Mass Distribution* also dropped the object when it chose inaccurate mass maps, which results in selecting unstable goal poses. As shown in Figure 4, all the identified mass models perfectly predicted the motions of the objects while they were entirely on the surface because they were used jointly with inaccurate friction maps that compensate for the mass in the product mass \times friction, as shown in Figure 2. The proposed method achieves the highest success rate, with a slightly higher number of pushing actions than other methods, as shown in Figure 6 (b).

2) *Real Robot Experiments:* We repeated the same experiment as the previous one using the real robotic setup with a hammer set in 10 different initial poses on the table. The *Kuka* robot repeatedly observes the object's pose on the table and pushes the object toward pre-grasp poses, using Algorithm 4. Once the object reaches a goal pose, the robot grasps the object by inserting its fingers around it and aligning the approaching direction to the surface normal of the contact point. Figure 7 shows the percentage of successful grasps resulting from the entire process of Algorithm 4. The lower success rate of *Planning with Uniform Mass Distribution* and *Deterministic Planning with identified Mass Distribution* mostly comes from unstable goal pose selections. The *Uniform Mass Distribution* shows much lower success rate than simulation results. This is because reachable object poses with the real robot are more limited than in simulation, and plans obtained from inaccurate mass distributions lead into pushing the object out of the workspace. The proposed method was able to avoid those situation by considering many plausible models simultaneously and choosing actions that are most guaranteed to succeed.



(a) Percentage of successful grasps

Method	# of Actions
Planning with Uniform Mass	6.05 (± 1.76)
Deterministic Planning	5.69 (± 1.96)
Probabilistic Planning (Proposed)	6.27 (± 1.82)
Planning with Ground-truth Mass	5.46 (± 1.44)

(b) Average number of executed actions

Fig. 6. Simulation results on pre-grasp sliding manipulation

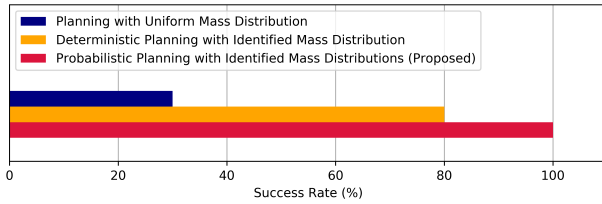


Fig. 7. Real robot results on pre-grasp sliding manipulation

VII. CONCLUSION

Manipulating heterogeneous objects safely requires accurate models of mass and friction maps. In this work, we proposed a new method for identifying such models automatically from data collected by using a robotic manipulator. When the data is limited and collected from quasi-static motions, it is not possible to disentangle mass from friction. Therefore, the proposed approach returns a probability distribution over several plausible models. We have also shown that the inferred probability distribution can be used to plan side-pushing actions that displace an object into the edge of a support surface while keeping it balanced. The object can then be grasped easily from the edge. Future work includes investigating other probabilistic mass and friction representations, as well as inferring models of objects in highly cluttered scenes where contact and collisions between different unknown objects may occur.

REFERENCES

- [1] K. Hang, A. Morgan, and A. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2019.
- [2] M. R. Dogar and S. S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2123–2130, 2010.
- [3] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. S. Srinivasa, "Object search by manipulation," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4973–4980.
- [4] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-Based Grasp Planning Through Clutter," in *R:SS*, July 2012.
- [5] J. E. King, M. Klingensmith, C. M. Dellin, M. R. Dogar, P. Velagapudi, N. S. Pollard, and S. S. Srinivasa, "Pregrasp manipulation as trajectory optimization," in *Robotics: Science and Systems*, 2013.
- [6] M. R. Dogar and S. S. Srinivasa, "A framework for push-grasping in clutter," in *Robotics: Science and Systems*, 2011.
- [7] —, "A planning framework for non-prehensile manipulation under clutter and uncertainty," *Autonomous Robots*, vol. 33, no. 3, pp. 217–236, Oct 2012.
- [8] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.
- [9] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 533–556, 1996.
- [10] K. M. Lynch, "Estimating the friction parameters of pushed objects," in *1993 International Conference on Intelligent Robots and Systems*, ser. 1993 International Conference on Intelligent Robots and Systems, 12 1993, pp. 186–193.
- [11] R. D. Howe and M. R. Cutkosky, "Practical force-motion models for sliding manipulation," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 557–572, 1996.
- [12] J. Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason, "A convex polynomial force-motion model for planar sliding: Identification and application," in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, 2016, pp. 372–377.
- [13] J. Zhou, Y. Hou, and M. T. Mason, "Pushing revisited: Differential flatness, trajectory planning, and stabilization," *I. J. Robotics Res.*, vol. 38, no. 12-13, 2019.
- [14] J. Zhou, M. T. Mason, R. Paolini, and D. Bagnell, "A convex polynomial model for planar sliding mechanics: theory, application, and experimental validation," *I. J. Robotics Res.*, vol. 37, no. 2-3, pp. 249–265, 2018.
- [15] T. Yoshikawa and M. Kurisu, "Identification of the center of friction from pushing an object by a mobile robot," *Proceedings IROS '91: IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, pp. 449–454 vol.2, 1991.
- [16] J. Degraeve, M. Hermans, J. Dambre, and F. Wyffels, "A differentiable physics engine for deep learning in robotics," *CoRR*, vol. abs/1611.01652, 2016.
- [17] A. Kloss, S. Schaal, and J. Bohg, "Combining learned and analytical models for predicting action effects," *CoRR*, vol. abs/1710.04102, 2017.
- [18] M. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *Proc. of Robotics: Science and Systems (R:SS 2018)*, 2018.
- [19] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 43:1–43:8, July 2012. [Online]. Available: <http://doi.acm.org/10.1145/2185520.2185539>
- [20] F. de Avila Belbute-Peres and Z. Kolter, "A modular differentiable rigid body physics engine," in *Deep Reinforcement Learning Symposium, NIPS*, 2017.
- [21] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds., 2013, pp. 548–562.
- [22] C. Song and A. Boularias, "Identifying mechanical models of unknown objects with differentiable physics simulations," 2020, Under review for the 2nd Annual Conference on Learning for Dynamics and Control (L4DC 2020), Berkeley, California.