Physics-Based Trajectory Optimization for Grasping in Cluttered Environments

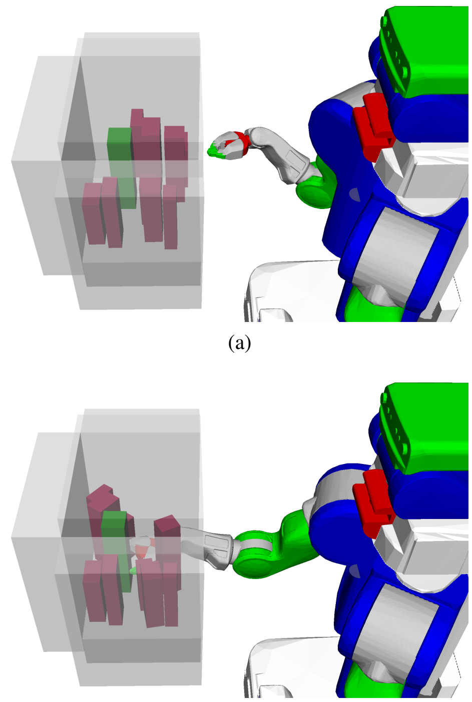
Nikita Kitaev Igor Mordatch Sachin Patil Pieter Abbeel

Abstract— **Grasping an object in a cluttered, unorganized environment is challenging because of unavoidable contacts and interactions between the robot and multiple immovable (static) and movable (dynamic) obstacles in the environment. Planning an approach trajectory for grasping in such situations can benefit from physics-based simulations that describe the dynamics of the interaction between the robot manipulator and the environment. In this work, we present a physics-based trajectory optimization approach for planning grasp approach trajectories. We present novel cost objectives and identify failure modes relevant to grasping in cluttered environments. Our approach uses rollouts of physics-based simulations to compute the gradient of the objective and of the dynamics. Our approach naturally generates behaviors such as choosing to push objects that are less likely to topple over, recognizing and avoiding situations which might cause a cascade of objects to fall over, and adjusting the manipulator trajectory to push objects aside in a direction orthogonal to the grasping direction. We present results in simulation for grasping in a variety of cluttered environments with varying levels of density of obstacles in the environment. Our experiments in simulation indicate that our approach outperforms a baseline approach that considers multiple straight-line trajectories modified to account for static obstacles by an aggregate success rate of 14% with varying degrees of object clutter.**

I. Introduction

In this work, we consider the problem of grasping and retrieving an object in a cluttered, unorganized environment such as a cluttered bookshelf or refrigerator shelf (Fig. 1). Grasping in such situations is challenging because it might be impossible to directly reach the target object while avoiding contact with multiple immovable (static) or movable (dynamic) objects in the environment. Grasp and manipulation planning in clutter should take into account physical interactions of the robot manipulator with these objects, so that the robot can move them out of its way without any undesirable consequences, such as objects toppling over the edge or getting crushed against a wall.

Physics-based simulation can provide useful cues about complex interactions between the robot and the environment. One possibility is to evaluate grasp approach trajectories by simulating them and selecting the most desirable trajectory. However, this might require evaluation of a large number of grasp approach trajectories to fully capture all possible outcomes in cluttered environments because of the large number of possible pairwise interactions. While some of these interactions can be pre-computed and cached [6], this comes at the price of making significant assumptions about the motion of the robot and objects in the environment.



(a)

(b)

Fig. 1: (a) The objective is to grasp a target object (shown in green) on a cluttered refrigerator shelf. In such situations, interactions with obstacles (shown in magenta) are unavoidable. (b) Our approach uses physics-based trajectory optimization to compute a grasp approach trajectory that moves obstacles out of the way to reach the target object.

In this work, we present a physics-based trajectory optimization method for planning grasp approach trajectories through cluttered environments. We consider the kinematic robot state in the trajectory optimization. Since the arm is fully actuated and is expected to move slowly, we do not consider dynamics of the arm in this problem. The configurations of the dynamic obstacles in the environment follow from rollouts of physics-based simulations. As a result, the optimization only needs to consider obstacles in the environment that would be impacted by the grasp approach trajectory.

We consider an optimization objective that is a weighted combination of costs that are relevant for grasping in cluttered environments. The gradient of this objective and physics-based dynamics is computed using roll-outs of the physics-based simulation. We take advantage of multi-core architectures to perform the gradient computation in parallel. For trajectory optimization, we extend a dynamic programming algorithm called iterative LQR (iLQR) [18]

Nikita Kitaev, Sachin Patil, and Pieter Abbeel are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA. Igor Mordatch is with the Department of Computer Science and Engineering, University of Washington, WA, USA.

that computes a linear control policy using a quadratic approximation of the value function around a given trajectory and iteratively refines the trajectory under forward executions of the computed policy until convergence to obtain a locally-optimal trajectory.

We present simulation results for grasping an object in a variety of environments and under varying degrees of clutter. Our approach naturally generates behaviors such as choosing to push objects that are less likely to topple over, closing the gripper when approaching clutter, recognizing and avoiding situations which might cause a cascade of objects falling over, and adjusting the manipulator trajectory to snake through clutter by pushing objects aside in a direction orthogonal to the grasping direction (Sec. VI). Our approach offers an improvement of 14% in the success rate of grasping in clutter as compared to a sampling-based straight-line grasp approach trajectories [6] at the expense of additional computation involved. Our experimental results are promising and we expect that significant computational speed-ups can be obtained on large-scale parallel architectures, such as those offered by cloud computing, to enable real-time planning for robotic grasping in cluttered environments.

II. Related Work

Robotic grasping has been extensively studied over the past four decades [2], [28]. Most prior work focuses on computing grasp quality metrics and planning grasps on a given target object. However, clutter poses a significant challenge for grasping in unstructured environments [14]. Prior work has addressed integrated perception and grasping in clutter where the objective is to grasp objects in an unorganized pile and place them in a bin [15], [23], but these methods do not specifically aim to grasp a single target object in clutter. Novel grasping mechanisms have been proposed for grasping a single object from a cluttered pile [21]. Leeper et al. [17] use a human operator for assistance to guide the robot through clutter with the objective of grasping a target object. In this work, we do not address the issue of perception and focus on autonomous grasping of a target object in a cluttered environment.

Prior work has explored the use of hierarchical approaches for robot motion planning amid movable obstacles [32], [34]. These methods use sampling-based planning to compute a high-level plan for the robot to get to a desired target location by moving obstacles out of the way and rearranging them. Such an approach has also been applied to the problem of grasping in clutter. Dogar et al. [7] propose a framework for push-grasping in clutter that plans a grasp approach trajectory to the target object by keeping track of a set of movable objects in the environment. Hauser et al. [10] consider the problem of removing the minimum number of movable obstacles for the robot to achieve its objective. Lindzey et al. [19] consider an extension where objects in the environment are pushed and rearranged by multiple robots. Kaelbling et al. [13] present an integrated task and motion planner that plans backwards from the objective of grasping a desired target object using a regression-based symbolic planner. Srivastava et al. [31] present a different approach to accomplish this by integrating a symbolic high-level planner with low-level kinematic trajectory optimization. Our approach can be integrated with a high-level planner to improve performance of grasping in highly cluttered environments where it might not be possible to reach the target object without explicitly removing or rearranging obstacles.

Prior work has studied the issue of manipulating objects by performing pushing operations [20], [24]. Berenson et al. [1] use a sampling-based planner that considers clearance from obstacles in the environment to plan a grasp approach trajectory in cluttered environments. Cosgun et al. [3] and King et al. [16] consider the problem of planning a series of push operations that would place an object at a desired target location. Recently, Dogar et al. [6] proposed a physics-based grasp planning approach that pre-computes and caches interactions of the robot gripper with obstacles in the environment. This enables fast evaluations of straight-line trajectories of the robot gripper to select the most desirable outcome for grasping in cluttered environments. However, this work does not consider inter-object interactions, which are significant in highly cluttered environments, and does not consider interactions of the manipulator arm with obstacles. Jain et al. [12] and extensions [25] assume no knowledge of the environment and use whole-arm tactile sensing with a reactive controller to reach into unknown, compliant environments with high degrees of clutter such as foliage. However, a reactive controller might not perform well in constrained spaces with rigid, movable obstacles.

Many robotic tasks that include locomotion and manipulation involve contacts with obstacles in the environment. Kinematic trajectory optimization methods have been proposed that avoid contact with obstacles in the environment [29], [30]. Recent work has focused on optimizing the execution speed of an existing trajectory under actuator limits and contact constraints [11], [26]. Erez et al. [9] and Mordatch et al. [22] perform dynamics trajectory optimization using a smooth contact dynamics model to facilitate gradient computations. In these works, the feasibility of the contact forces is included as a penalty term in the optimization cost. Posa et al. [27] use a direct, also known as collocation, optimization formulation that formulates the optimization problem in terms of the contact forces and the state of all objects in the environment. However, in highly cluttered environments, the optimization problem is very high-dimensional and enforcing the complementarity constraints arising out of contacts is challenging. Erez et al. [8] also use iterative LQR for trajectory optimization for humanoid robots, but this method optimizes over the joint angles and velocities of the robot and does not consider contacts with multiple, dynamic obstacles in the environment.

III. Problem Definition

We consider the problem of planning a robot arm trajectory to grasp a target object, as shown in Fig. 2. We consider a robot consisting of links $L_i, i = 1, \dots, N_{\text{links}}$,

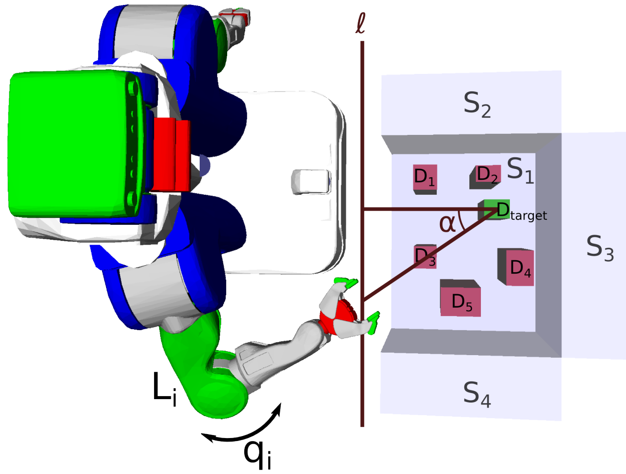


Fig. 2: The robot consists of links L_i with configuration defined by a vector of joint angles $\mathbf{q}$. The workspace contains immovable ($S_i,\ i = 1 \ldots N_s$) and movable ($D_i,\ i = 1 \ldots N_d$) obstacles. The objective is to grasp the target object D_{target}.

whose positions $L_i(\mathbf{q})$ are parametrized by a vector of joint angles $\mathbf{q}$. The environment is composed of a set of objects $\mathcal{O} = \{S_1, .., S_{N_s}, D_1, .., D_{N_d}\}$, where $S_j,\ j = 1, \ldots, N_s$ denote static obstacles in the scene (e.g. shelves or walls). The configuration of the dynamic obstacles are parametrized as $D_i(\mathbf{x}_i^t)$, where $\mathbf{x}_i^t \in SE(3)$ denotes the pose (position and orientation) of object i at time t. Let $\mathcal{X}^t = \{x_1, .., x_{N_d}\}$ be the set of object poses at time t. One of these objects is the object we want the robot to grasp, denoted as D_{target}. We are given the initial poses of the dynamic obstacles, $\mathcal{X}^1$. We also assume access to a physics-based simulator [33] that provides a smooth and differentiable dynamics function ϕ, which is used to calculate $\mathcal{X}^t = \phi(\mathcal{X}^{t-1}, \mathbf{q}^t)$. In this work, we also restrict ourselves to cases where all dynamic objects are boxes that rest on a single flat, horizontal surface and assume that the geometric and inertial properties of the objects are fully known.

Objective: The objective is to find a feasible grasp approach trajectory $\mathbf{q}^1, \ldots, \mathbf{q}^\Delta$ according to the evaluation criteria defined below, such that the initial robot configuration $\mathbf{q}^1$ is fully outside the work area, and the target object is in the pre-grasp zone of the robot gripper at time Δ. We assume that a motion planner can move the robot manipulator into this initial configuration prior to executing the trajectory.

Evaluation Criteria: The primary distinction between success and failure in this problem is the effect of the robot's actions on objects in the environment. We evaluate this by up-sampling the candidate trajectories and executing them in a physics simulation environment. Once the robot trajectory execution has finished, we continue the simulation for an additional segment of time to allow objects to come to rest.

We define a successful trajectory as one that moves the robot manipulator such that the robot can grasp the target object while all obstacles remain upright and on the working surface. We define partial success as when the robot is able to grasp the target object but tips over one or more dynamic objects in the process, without any objects falling off the edge of the working surface. All other behaviors are considered failures, which we divide into the following categories:

(1) Failure to grasp: The robot is not grasping the target object at the end of the trajectory. For example, the target object is pushed away from the robot, or the robot has a different object in its gripper instead.

(2) Objects falling over: At least one object falls over the edge of the working surface.

(3) Kinematic failure: The trajectory cannot be simulated to completion, either because the manipulator arm attempts to move through static obstacles or the dynamic objects get pushed into the static objects with excessive force.

IV. Baseline Approach

Following Dogar et al. [6], we consider a baseline approach that samples a number of straight-line trajectories at different approach angles α to the target (Fig. 2). However, instead of pre-computing quasi-static interactions, we use a full physics simulator to evaluate trajectories. This allows us to consider interactions between objects and interactions with the full manipulator arm instead of restricting ourselves to just the robot gripper.

Since all objects are assumed to rest on an horizontal surface, we consider trajectories where the gripper is horizontal and at a fixed height h above that surface. At this height level, we sample a series of approach angles $\alpha \in (\alpha_{\min}, \alpha_{\max})$. The starting gripper position lies on a line l that is outside the workspace and an end-gripper pose is chosen such that the target object can be grasped.

Given the gripper poses at the two endpoints of the trajectory, we use OpenRAVE [5] inverse kinematics solver to find corresponding joint configurations $\mathbf{q}_\alpha^1$ and $\mathbf{q}_\alpha^\Delta$. The IK step is not always possible for all angles α (e.g. when the approach angle is from the direction of a wall), in which case we discard the straight line trajectory. We use linear interpolation to create a full joint-space trajectory $\mathbf{q}_\alpha = \text{interpolate}(\mathbf{q}_\alpha^1, \mathbf{q}_\alpha^\Delta)$.

The trajectories $\mathbf{q}_\alpha$ may be infeasible due to collisions with static obstacles. We use a state-of-the-art motion planner [30] to find collision-free trajectories $\mathbf{q}'_\alpha$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{q}^{1,..,\Delta}} \ & \sum_t \sum_k w_k c_k^t(\mathbf{q}^t, \mathbf{q}^{t-1}) \\ \text{s.t.} \ & \mathbf{p}_{\text{gripper}}(\mathbf{q}^\Delta) = \mathbf{p}_{\text{gripper}}(\mathbf{q}_\alpha^\Delta) \\ & R_{\text{gripper}}(\mathbf{q}^\Delta) = R_{\text{gripper}}(\mathbf{q}_\alpha^\Delta), \end{aligned}$$

where $\mathbf{p}_{\text{gripper}}$ and R_{gripper} represent the position and rotation matrix describing the pose of the gripper (i.e. the constraints are chosen to leave the pregrasp position at time Δ unchanged). The objective is a weighted combination of cost terms c_k^t with weights w_k is the weight for k^{th} cost term. The optimization problem uses two cost functions: $c_{\text{static\_hull}}$ and c_{vel}, as described in Table I. The term $c_{\text{static\_hull}}$ penalizes penetration between robot links and static obstacles if it is closer than a distance d_{safe}. This term is a function of the convex hull of the robot link from consecutive timesteps to approximate the volume swept out by the link during

Collision costs

$$c^t_{\text{static\_hull}}(\mathbf{q}^t, \mathbf{q}^{t-1}) = \sum_{n=1}^{N_{\text{links}}} \sum_{j=1}^{N_{\text{static}}} \text{penetration}(\text{convhull}(L_n(\mathbf{q}^t), L_n(\mathbf{q}^{t-1})), S_j))$$

$$c^t_{\text{static}}(\mathbf{q}^t, \mathbf{q}^{t-1}) = \sum_{n=1}^{N_{\text{links}}} \sum_{j=1}^{N_{\text{static}}} \text{penetration}(L_n(\mathbf{q}^t), S_j)$$

$$c^t_{\text{dynamic}}(\mathbf{q}^t, \mathbf{q}^{t-1}, \mathbf{x}) = \sum_{n=1}^{N_{\text{links}}} \sum_{i=1}^{N_{\text{dynamic}}} \text{penetration}(\text{convhull}(L_n(\mathbf{q}^t), L_n(\mathbf{q}^{t-1})), D_i(\mathbf{x}_i))$$

Clutter behavior costs

$$c_{\text{force}}(\mathbf{q}, \mathbf{x}) = \frac{1}{2}\,||\mathbf{force}(\mathbf{q}, \mathbf{x})||^2$$

$$c_{\text{motion}}(\mathbf{x}) = \sum_i (\mathbf{p}_i(\mathbf{x}_i) - \mathbf{p}_i(\mathbf{x}^1_i))^2$$

$$c_{\text{upright}}(\mathbf{x}) = \cos^{-1}(R_i(\mathbf{x}) \cdot \mathbf{z})$$

Grasp constraints

$$c_{\text{grasp}}(\mathbf{q}, \mathbf{x}) = \text{distance}(\mathbf{p}_{\text{gripper}}(\mathbf{q}), D_{\text{target}}(\mathbf{x}))$$

$$c_{\text{grp\_horiz}}(\mathbf{q}) = \cos^{-1}(R_{\text{gripper}}(\mathbf{q}) \cdot \mathbf{z})$$

$$c_{\text{grp\_open}}(\mathbf{q}) = (q_{\text{gripper}} - q_{\text{gripper\_open}})^2$$

Regularization terms

$$c^t_{\text{vel}}(\mathbf{q}^t, \mathbf{q}^{t-1}) = ||\mathbf{q}^t - \mathbf{q}^{t-1}||^2$$

$$c^t_{\text{acc}}(\mathbf{q}^t, \mathbf{q}^{t-1}, \mathbf{q}^{t+1}) = \frac{1}{2}||\mathbf{q}^{t-1} - 2\mathbf{q}^t + \mathbf{q}^{t+1}||^2 \frac{1}{\text{dt}^2}$$

Trajectory matching costs

$$c_{\text{match}}(\mathbf{q}^t) = ||\mathbf{q}^t - \mathbf{q}^t_{\text{ref}}||^2$$

$$c_{\text{match\_vel}}(\mathbf{q}^t, \mathbf{q}^{t-1}) = ||(\mathbf{q}^t - \mathbf{q}^{t-1}) - (\mathbf{q}^t_{\text{ref}} - \mathbf{q}^{t-1}_{\text{ref}})||^2$$

Definitions

force: Vector of forces between all pairs of objects in the scene. Does not include internal forces.

penetration: Penetration depth between a pair of objects. Equal to 0 when the objects are a margin d_{safe} apart.

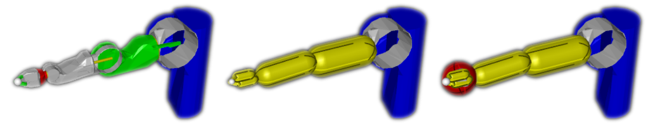
TABLE I: Cost terms used in the optimization formulation. The collision cost terms penalize collisions with obstacles, both static and dynamic, in the environment. The clutter behavior cost terms penalize forces, deviation from standing upright, and motion of dynamic obstacles. The grasp cost terms penalize deviation of the target object from the gripper pre-grasp pose. We also add regularization terms for smooth manipulator motions and trajectory matching costs for tracking a reference trajectory.

that time. The regularization term c_{vel} penalizes robot velocity, ensuring smooth output trajectories. We use sequential quadratic programming (SQP) [30], to solve this optimization problem starting with the initialization $\mathbf{q}_\alpha$. We used weights of $w_{\text{static\_hull}} = 80$ and $w_{\text{vel}} = 1$ in our implementation.

The performance of this baseline approach can be measured by evaluating all of the output trajectories $\mathbf{q}'_\alpha$ in a physics simulator, and selecting the value of α that yields the best performance based on the criteria in Sec. III.

V. Physics-based Trajectory Optimization

Our approach aims to improve on the baseline by considering the physical behavior of dynamic obstacles. We assume that they move according to rigid body dynamics, colliding with the arm and each other. Since the arm is fully actuated and is expected to move slowly, we do not consider dynamics of the arm in this problem. The objects start in the initial location and behave according to dynamics ϕ. The objects are unactuated and only move when colliding with the arm



(a) (b) (c)

Fig. 3: (a) Geometry of the robot manipulator arm. (b) Proxy capsule geometry fitted to conform to the arm geometry is used for planning. (c) The robot gripper is replaced with a sphere proxy for planning.

or other objects, hence ϕ couples trajectories of all objects. We formulate the following optimization problem:

$$\min_{\mathbf{q}^{1,\ldots,\Delta}} \sum_t \sum_k \sum_i w_k c^t_k(\mathbf{q}^t, \mathbf{q}^{t-1}, \mathbf{x}^t_i)$$
$$\text{s.t.}\quad \mathcal{X}^t = \phi(\mathcal{X}^{t-1}, \mathbf{q}^t), \quad \mathcal{X}^1 \text{is fixed}$$

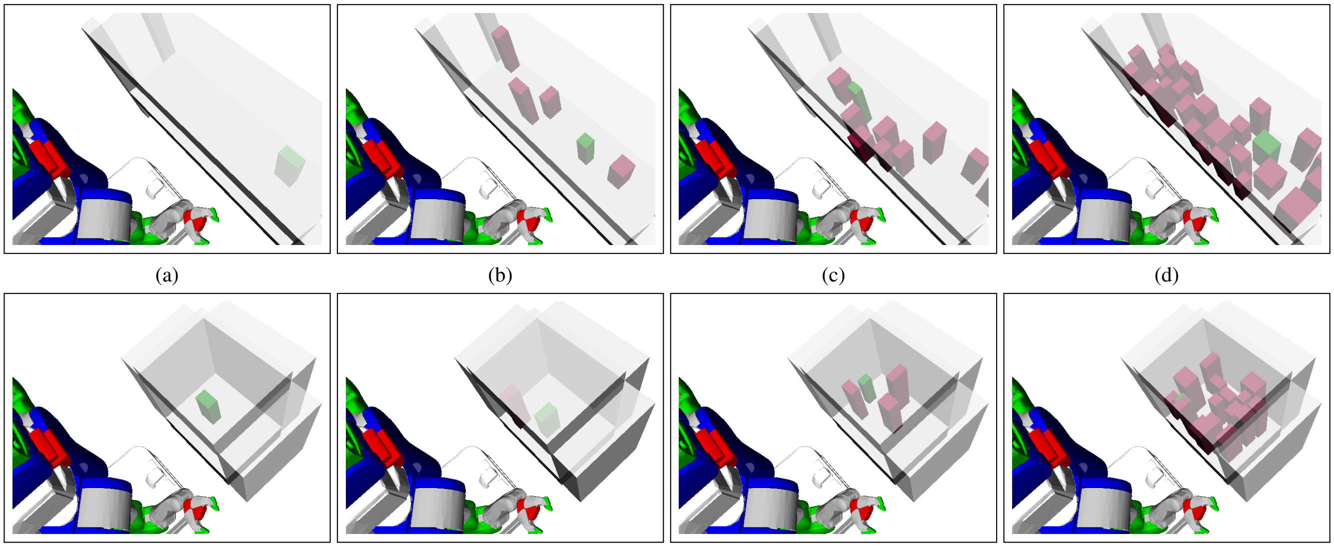
The above is a nonlinear control problem with controls $\mathbf{q}^{1,\ldots,\Delta}$ and can be solved with existing methods. We use the popular iLQR method [18], although other methods can also be applied. The full details of iLQR are outside the scope of this paper. Briefly, the method starts with an initial guess for a control trajectory $\mathbf{q}^{1,\ldots,\Delta}$ (which we can get from the previous section) and rolls out dynamics ϕ using this sequence of controls, resulting in object trajectory $\mathcal{X}^{1,\ldots,\Delta}$. Then a Linear Quadratic Regulator (LQR) problem is formed by making a linear approximation to dynamics ϕ and quadratic approximation to costs c_k around $\mathcal{X}^{1,\ldots,\Delta}$. The resulting standard LQR problem can be solved efficiently and exactly and yields an updated control trajectory $\mathbf{q}^{1,\ldots,\Delta}$. This process is iterated until convergence. In our experiments, we chose to cap the algorithm at 30 iterations, which we found to be sufficient for convergence.

To form the quadratic approximation to the costs c_k, we must calculate their gradients and Hessians, which we do analytically for all costs in Table I. To linearize the dynamics ϕ, we must calculate Jacobians $\frac{d\phi}{d\mathcal{X}}$ and $\frac{d\phi}{d\mathbf{q}}$, which we do with finite differencing of step size of 10^{-6}. For object dynamics simulation we use the MuJoCo [33] physics engine and its smooth contact model, which provides a differentiable and smooth dynamics function. To take advantage of smooth contacts, we replace the PR2 mesh model (Fig. 3a) with a proxy model made from primitive shapes such as capsules and spheres (Fig. 3b). Additionally, trajectories used for initialization will often have the robot gripper accidentally grasp obstacles. To avoid falling into a local minimum, the first 10 optimization iterations are run with the gripper replaced with a sphere proxy (Fig. 3c) that represents the convex hull of the gripper.

Cost Term	static	acc	force	motion	upright
Weight	10^3	10^{-3}	10	10^{-3}	1
Time	$t \leq \Delta$	$t \leq \Delta$	all	all	all

Cost Term	grasp	grp_horiz	grp_open
Weight	10^3	1	1
Time	terminal cost	terminal cost	terminal cost

TABLE II: Weights and timesteps for objective terms used in our approach



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4: **Scenes**: [a-d] Bookshelf and [e-f] Refrigerator shelf, with various densities of clutter objects. In both cases, the target object is highlighted in green.

Cost Terms: The problem uses the cost terms c_{static} and c_{acc} (Table I), which are similar to the terms used for the static collision avoidance. We add additional costs to penalize the failure modes described in Section III. The cost c_{force} prevents kinematically infeasible trajectories by penalizing contact forces for all robot links and obstacles in the environment. c_{motion} penalizes falling off the table and excessive shuffling, and c_{upright} penalizes tipping over. The terminal costs are c_{grasp}, which encourages the gripper to be close to the target object, $c_{\text{grp\_horiz}}$, which enforces that the gripper be horizontal, and $c_{\text{grp\_open}}$, which expresses that the gripper must be fully open at the last timestep.

The weights used for these cost terms, and the timesteps they were active for, are shown in Table II.

A. Initialization

For our approach, we intend to pass the trajectories q'_α generated by the baseline approach as initializations to the physics-based planner. However, these trajectories are often in heavy collision with dynamic obstacles.

To have better-conditioned physical behavior in the initializations, we run a pre-processing step that attempts to minimize dynamic obstacle contact. This is done using a cost c_{dynamic} (Table I), which assumes that the dynamic obstacle positions $\mathbf{x}_i$ remain constant and uses the same formulation as $c_{\text{static\_hull}}$. The costs $c_{\text{static\_hull}}$ and c_{vel} from the baseline approach remain to ensure smooth trajectories that don't collide with static obstacles.

B. Post-processing Step

When switching from the proxy PR2 model used in optimization to a mesh PR2 model used in the evaluation, slight collisions with static obstacles may appear. We therefore apply a final optimization pass to correct for this behavior. The costs are c_{static} from Table I, a penalty for deviating from the original trajectory c_{match}, and a smoothness term that penalizes deviating from the original velocity $c_{\text{match\_vel}}$.

VI. Experiments

A. Representative Environments

To showcase the behaviors that naturally fall out of our method, we constructed a series of environments that exhibit a variety of clutter conditions and object properties. A single approach angle α was used to construct the baseline kinematic trajectory for each run, which was then optimized with with our physics-based approach. To detect overfitting to the approximations used in the optimization, trajectories were simulated with a mesh-based robot model and a hard contacts (as opposed to the proxy model and soft contacts used in the planning step). We used the OpenRAVE simulation environment [5] with the Bullet Physics Engine [4] for simulating the dynamics. Figs. 5 - 8 show executions of trajectories that are computed using a baseline approach and our approach. The following behaviors were observed:

1) The physics-based optimization approach takes into account differences between the inertial properties of different objects, which lets it pick trajectories that avoid easy-to-topple obstacles (Fig. 5).

2) If an object's aspect ratio makes it easier to topple in one direction than another, our physics-based approach will learn to push it from the safer direction (Fig. 6).

3) The physics-based optimization approach is able to recognize the danger of triggering a cascade of falling objects. As seen in Figure 7, it finds a trajectory that results in less harmful object-object interactions.

4) For a highly cluttered scene (Fig. 8), the output trajectory from our approach moves the gripper from side to side as it moves through the clutter. This is similar to how humans "snake" their arms around obstacles and push them aside in a direction perpendicular to the grasp path.

5) In all of the scenes shown, the physics-based planner opens and closes the gripper at appropriate times as it moves through the clutter. This is an effective strategy to

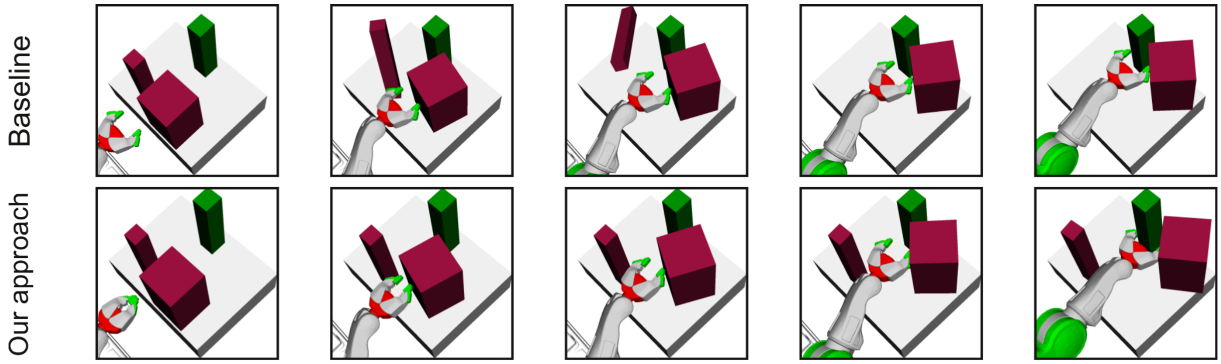


Fig. 5: **Large and small obstacles**: The robot must reach between two obstacles (red) to grasp the target object (green). The baseline approach (top) interacts with both obstacles, while our approach (bottom) only interacts with the larger, more stable obstacle.

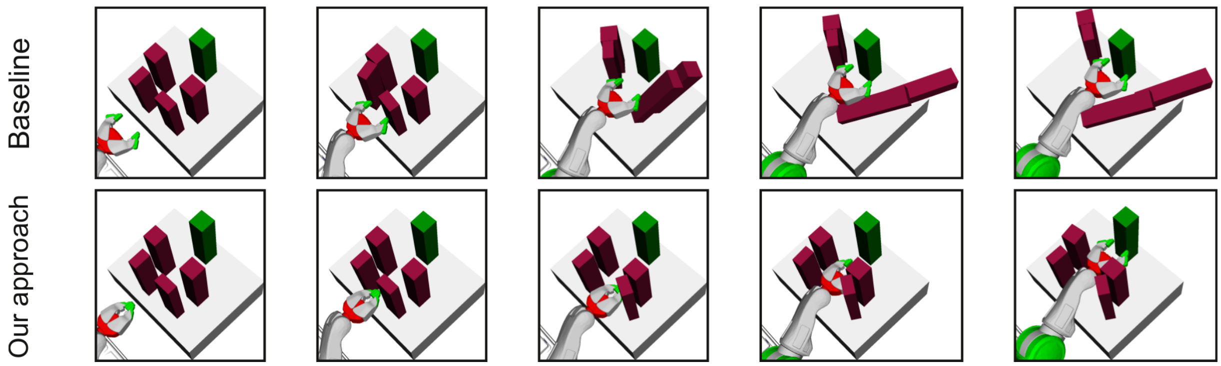


Fig. 6: **Pushing sideways**: The two obstacles in the lower-left are thin along one dimension, making it easy for them to topple in that direction. The baseline approach (top) topples one of these objects, while our approach (bottom) generates a more stable interaction by pushing on it sideways

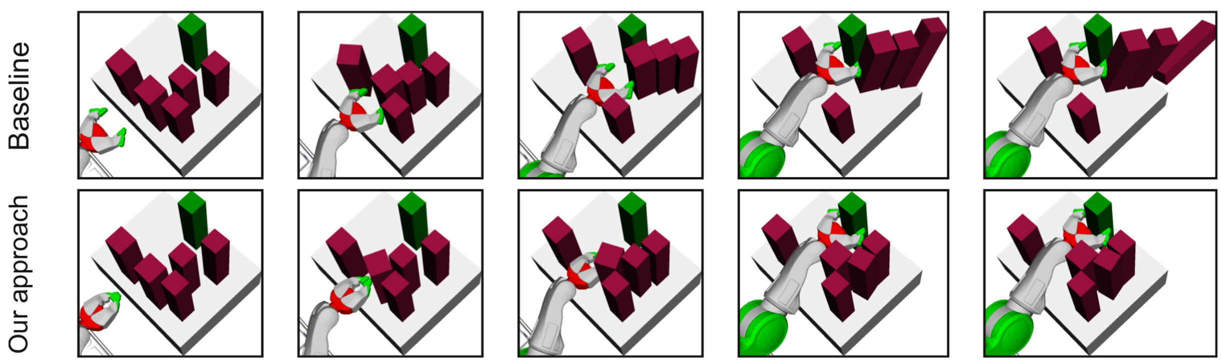


Fig. 7: **Cascade**: The baseline approach (top) triggers a cascade of falling obstacles in this scene. Our approach (bottom) generates a trajectory that does not trigger this cascade.

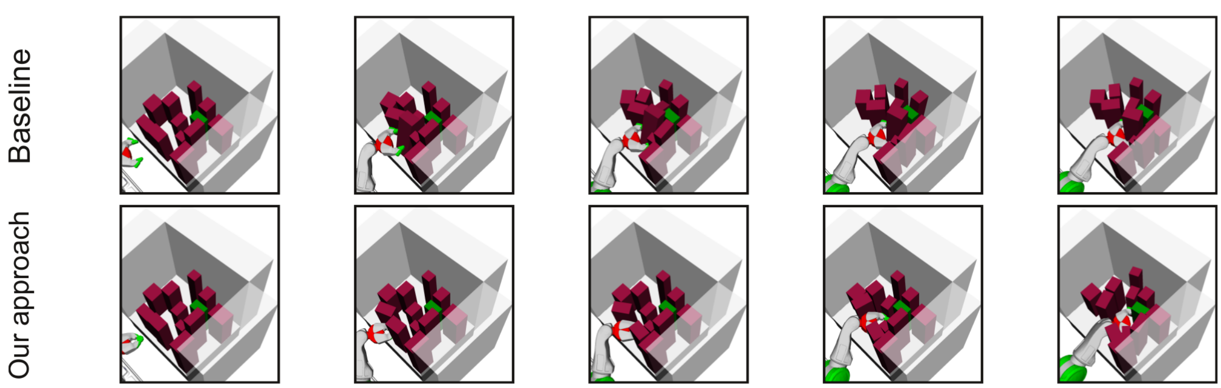


Fig. 8: **Snaking path**: The baseline trajectory (top) fails for this cluttered scene. The trajectory from our approach (bottom) alternates making contact with obstacles to the left and to the right of the gripper, pushing them to the side as the arm snakes through to grasp the target.

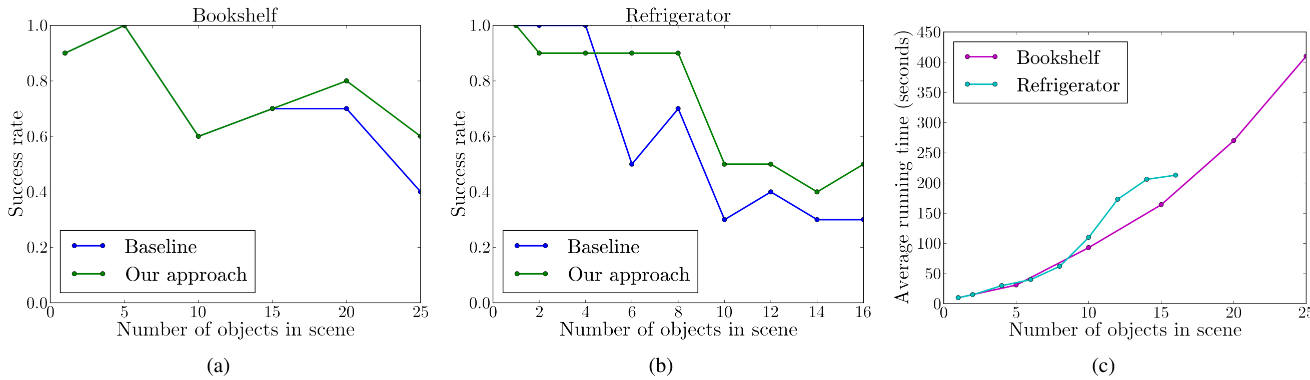


Fig. 9: (a,b) Comparison of success rates of both approaches for the bookshelf and refrigerator shelf environment on scenes with varying number of objects. (c) Average running time of our physics-based trajectory optimization on scenes with varying degrees of clutter.

avoid grasping the wrong object, while also pushing objects sideways instead of in the direction of the grasp.

B. *Quantitative Results*

We considered two types of environments: a cluttered bookshelf and a refrigerator shelf (Fig. 4). The bookshelf is challenging because there are no walls to keep the objects from falling off any of the four sides. The refrigerator shelf is more constrained, limiting range of motion for the robot arm and the degree to which objects can be pushed around.

Each environment is populated with a number of randomly-generated boxes. Their size, position, and mass density are selected uniformly at random within a defined range. In studying these algorithms, we are interested in comparing their performance across different levels of clutter. The bookshelf had 6 clutter levels: 1, 5, 10, 15, 20, or 25 objects (Fig. 4d). The refrigerator shelf had 9 clutter levels: 1, 2, 4, 6, 8, 10, 12, 14, or 16 objects (Fig. 4h). At each level of clutter, 10 scenes were generated at random, for a total of 150 scenes. For each scene, 14 choices of approach angle α were sampled. If an IK solution for an angle could be found, it was used to initialize the baseline approach. Our approach was run with the baseline trajectories as inputs.

The resulting trajectories were then executed within the OpenRAVE simulation environment, using the Bullet physics engine [4]. The success of each trajectory was scored automatically according to the classification in Sec. III. If any approach angle α led to a successful trajectory, the scene was marked as a success. If the best initialization led to only a partial success, the scene was marked as a partial success. In all other cases, the scene was considered a failure.

Effect of clutter density: The relationship between clutter level and success rate for the two scenes are shown in Figs. 9a and 9b, respectively. For clutter densities of 5 objects or less, the baseline approach was sufficient to generate success rates of near 100%. Starting with 6 objects, the scenes become more difficult for this class of algorithms. At the other end of the spectrum, the workspace is nearly completely filled with objects. Grasp approach trajectories can only be found for those configurations where the target object is near the front, which explains the low but nonzero success rates.

Comparison: Table III shows the success and partial success rates for each scene and algorithm. Our approach offers a 14% improvement in aggregate across 150 scenes in both bookshelf and refrigerator environments. In addition, our approach is able to naturally produce behaviors for grasping in cluttered environments.

Algorithm	Bookshelf	Refrigerator	Aggregate
Baseline	76.7% (8.3%)	65.6% (5.6%)	70% (6.7%)
Physics-aware	86.7% (10.0%)	83.3% (13.3%)	84.7% (12.0%)

TABLE III: Success rates comparing the two approaches. The cumulative success rate includes both complete successes and partial successes (shown in parentheses).

Computation time: Our approach was implemented in C++ and we run our experiments on a 2.4 Ghz 24-core workstation. Fig. 9c shows how the running time per initialization of our physics-based trajectory optimization method scales with increasing number of objects in the scene. The increase in running time is due to the large number of pairwise interactions between objects in the environment.

The computational bottleneck is the gradient computation of the objective and the dynamics, which is performed in parallel. The running times of other steps were small in comparison. Pre/post-processing took less than a second combined, while evaluating a number of trajectories in Bullet to select the best one for that scene took ≈ 1 second per evaluation.

Analysis of Failure Modes: We also classified all of the 918 trajectories returned by each algorithm according to the failure mode classification outlined in Sec. III. The results are shown in Table IV. For our data set, the primary failure mode was failing to grasp the target object, followed by kinematic infeasibility (such as generating excessive contact forces). Our physics-based approach was able to reduce both of these.

Outcome	Baseline	Our approach
Full success	36.9%	48.7%
Partial success	1.1%	2.2%
Failure to grasp	32.9%	27.2%
Objects falling over	1.4%	1.3%
Kinematic failure	27.7%	20.6%

TABLE IV: Trajectory outcome distribution for the two algorithms, across all trajectories generated for our dataset.

VII. Conclusion

In this work, we presented a physics-based trajectory optimization method that plans grasp approach trajectories for grasping in cluttered environments. Our approach considers interactions of the full manipulator arm with objects in the environment and naturally generates behaviors such as closing the gripper when approaching clutter and carefully pushing objects that are less likely to fall over or trigger a cascade. Our experimental results in simulation are promising and indicate that physics-based trajectory optimization can offer significant benefits in terms of grasp success rates compared to a baseline approach of straight-line grasp approach trajectories.

This work opens up several avenues for future work. The physics-based trajectory optimization incurs significant computational overhead. However, we can take advantage of large-scale parallel systems and resources such as cloud computing to make this feasible for real-time planning. We currently consider deterministic environments without uncertainty in actuation and collisions. However, our method also generates a linear feedback controller that can be applied to deal with non-determinism. We also plan to integrate our approach with a perception system for demonstration on a robotic platform.

Acknowledgements

This research has been funded in part by AFOSR-YIP Award #FA9550-12-1-0345, by NSF under award IIS-1227536, by a DARPA Young Faculty Award #D13AP00046, and by a Sloan Fellowship.

References

[1] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp Planning in Complex Scenes," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2007, pp. 42–48.

[2] A. Bicchi and V. Kumar, "Robotic Grasping and Contact: A Review," in *ICRA*, 2000, pp. 348–353.

[3] A. Cosgun, T. Hermans, V. Emeli, and M. Stilman, "Push Planning for Object Placement on Cluttered Table Surfaces," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011, pp. 4627–4632.

[4] E. Coumans *et al.*, "Bullet Physics Library," *Open source: bulletphysics.org*, 2006.

[5] R. Diankov, S. Srinivasa, D. Ferguson, and J. Kuffner, "Manipulation Planning with Caging Grasps," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2008, pp. 285–292.

[6] M. Dogar, K. Hsaio, M. Ciocarlie, and S. Srinivasa, "Physics-based Grasp Planning through Clutter," in *Robotics: Science and Systems (RSS)*, 2012.

[7] M. Dogar and S. Srinivasa, "A Framework for Push-Grasping in Clutter," in *Robotics: Science and Systems (RSS)*, 2011.

[8] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An Integrated System for Real-Time Model Predictive Control of Humanoid Robots," in *Int. Conf. on Humanoid Robots*, 2013.

[9] T. Erez and E. Todorov, "Trajectory Optimization for Domains with Contacts using Inverse Dynamics," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 4914–4919.

[10] K. Hauser, "The Minimum Constraint Removal Problem with Three Robotics Applications," *Int. Journal of Robotics Research*, vol. 33, no. 1, pp. 5–17, 2014.

[11] ——, "Fast Dynamic Optimization of Robot Paths under Actuator Limits and Frictional Contact," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[12] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in Clutter with Whole-Arm Tactile Sensing," *Int. Journal of Robotics Research*, vol. 32, no. 4, pp. 458–482, 2013.

[13] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical Task and Motion Planning in the Now," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 1470–1477.

[14] D. Katz, J. Kenney, and O. Brock, "How can Robots Succeed in Unstructured Environments?" in *RSS 2008 Workshop on Robot Manipulation: Intelligence in Human Environments*, 2008.

[15] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, Learning, and Exploiting Object Affordances for Autonomous Pile Manipulation," in *Robotics: Science and Systems (RSS)*, 2013.

[16] J. King, J. Haustein, S. Srinivasa, and T. Asfour, "Kinodynamic Reconfiguration Planning with Physics," in *IROS 2014 Workshop*, 2014.

[17] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for Human-in-the-loop Robotic Grasping," in *ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2012, pp. 1–8.

[18] W. Li and E. Todorov, "Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems," in *Int. Conf. on Informatics in Control, Automation, and Robotics*, 2004, pp. 222–229.

[19] L. Lindzey, R. Knepper, H. Choset, and S. Srinivasa, "The Feasible Transition Graph: Encoding Topology and Manipulation Constraints for Multirobot Push-Planning," in *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, 2014.

[20] M. T. Mason, "Mechanics and Planning of Manipulator Pushing Operations," *Int. Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.

[21] M. T. Mason, A. Rodriguez, S. S. Srinivasa, and A. S. Vazquez, "Autonomous Manipulation with a General-Purpose Simple Hand," *Int. Journal of Robotics Research*, vol. 31, no. 5, pp. 688–703, 2012.

[22] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant Optimization for Hand Manipulation," in *ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, 2012, pp. 137–144.

[23] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stuckler, A. Berner, J. Li, R. Klein, and S. Behnke, "Mobile Bin Picking with an Anthropomorphic Service Robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013, pp. 2327–2334.

[24] D. Omrcen, C. Boge, T. Asfour, A. Ude, and R. Dillmann, "Autonomous Acquisition of Pushing Actions to Support Object Grasping with a Humanoid Robot," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2009, pp. 277–283.

[25] D. Park, A. Kapusta, Y. K. Kim, J. M. Rehg, and C. Kemp, "Learning to Reach into the Unknown: Selecting an Initial Condition When Reaching in Clutter," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[26] Q.-C. Pham and Y. Nakamura, "Affine Trajectory Deformation for Redundant Manipulators," in *Robotics: Science and Systems (RSS)*, 2012.

[27] M. Posa, C. Cantu, and R. Tedrake, "A Direct Method for Trajectory Optimization of Rigid Bodies through Contact," *Int. Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[28] D. Prattichizzo and J. C. Trinkle, "Grasping," *Springer Handbook of Robotics*, pp. 671–700, 2008.

[29] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2009, pp. 489–494.

[30] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion Planning with Sequential Convex Optimization and Convex Collision Checking," *Int. Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[31] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined Task and Motion Planning Through an Extensible Planner-Independent Interface Layer," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[32] M. Stilman and J. Kuffner, "Planning among Movable Obstacles with Artificial Constraints," *Int. Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1295–1307, 2008.

[33] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A Physics Engine for Model-based Control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[34] J. Van Den Berg, M. Stilman, J. Kuffner, M. Lin, and D. Manocha, "Path Planning among Movable Obstacles: A Probabilistically Complete Approach," in *Algorithmic Foundation of Robotics VIII*, 2009, pp. 599–614.