2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Planning and Control for Dynamic, Nonprehensile, and Hybrid Manipulation Tasks

J. Zachary Woodruff and Kevin M. Lynch

***Abstract*— In this paper we propose a method for motion planning and feedback control of hybrid, dynamic, and nonprehensile manipulation tasks. We outline five subproblems to address this: determining a set of manipulation primitives, choosing a sequence of tasks, picking transition states, motion planning for each individual primitive, and stabilizing each mode using feedback control. We apply the framework to plan a sequence of motions for manipulating a block with a planar 3R manipulator. We demonstrate preliminary experimental results for a block resting on the manipulator with a desired goal state on a ledge outside of the robot's workspace. The planned primitives reorient the block using a series of fixed, rolling, and sliding contact modes, and throw it to the goal state.**

I. Introduction

People and animals can effectively manipulate objects of many shapes, sizes, weights, and materials using a variety of primitives such as grasping, pushing, sliding, tipping, rolling, and throwing. In contrast, most robots manipulate objects by pick-and-place. There is good reason for this: once a firm grasp is established, the robot can reliably control the motion of the part without needing to continuously sense the state of the part or correct for modeling uncertainties. Most manipulation primitives mentioned above are more sensitive to uncertainties in part state, geometry, mass, friction, and restitution, and to the robot's own control errors. Nonetheless, restricting robots to only grasp objects artificially limits the set of tasks that they can accomplish. Leveraging a larger set of manipulation primitives is crucial for robots to reach their full potential in industrial automation, exploration, home care, military, and space applications.

A. Background

While manipulation primitives exist for manipulating several objects simultaneously, for simplicity we will focus on the case of a single rigid object. We define manipulation primitives according to the number and types of contacts the object makes with a robot and its (rigid) environment. Contacts are classified according to whether they are sliding or fixed/rolling, and contacts with a robot are further classified according to the control law the robot implements at that contact (e.g., position control, force control, hybrid position/force control, compliance control, etc.).

J. Zachary Woodruff, and Kevin M. Lynch are with the Neuroscience and Robotics Lab (NxR), Northwestern University, Evanston, IL 60208 USA. `jzwoodruff at u.northwestern.edu, kmlynch at northwestern.edu`. Kevin M. Lynch is also affiliated with the Northwestern Institute on Complex Systems (NICO).

This material is based upon work supported by the NSF Graduate Research Fellowship Program under grant No. DGE-1324585.

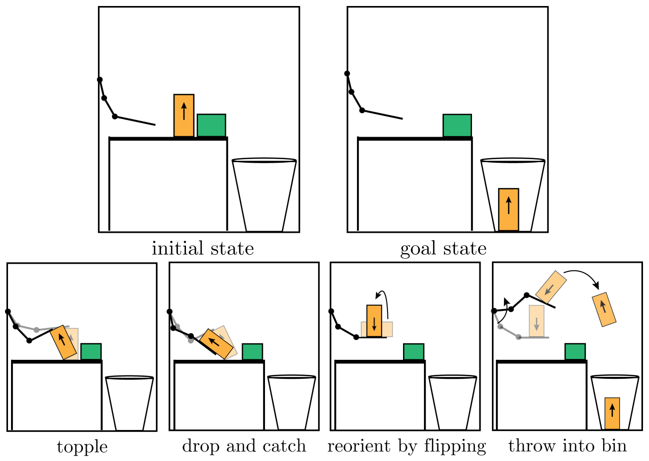

Fig. 1. An example of a manipulation task of picking up a block off a table and throwing it into a bin. This task involves multiple dynamic nonprehensile primitives.

Consider the planar example of a block and 3R manipulator shown in Figure 1. The block is initially at rest on the table with a desired goal state in the bin to the right. Another object, which should not be disturbed, is between the block's initial position and the goal configuration. The figure illustrates one possible solution to the manipulation task, consisting of a sequence of primitives. The controlled toppling primitive consists of one-point rolling between the block and the table while the robot applies a hybrid position-force controlled fixed contact to the top of the block, to control the internal force toward the rolling contact while controlling the orthogonal velocity. Once the block passes the unstable equilibrium point, the robot releases the block, letting it topple by gravity. The robot quickly moves underneath the block and "catches" it. The next primitive is a two-point, fixed-contact "dynamic grasp" carry, followed by a free flight phase of the block (a throw). After catching the object, the robot executes a dynamic grasp carry, followed by a phase where the object is in one-point rolling contact with the manipulator, followed by a free flight phase.

In summary, the manipulation sequence consists of a set of primitives punctuated by transitions:

controlled topple + free topple + catch + dynamic grasp + free flight + catch + dynamic grasp + rolling (controlled)+ free flight.

Each unique primitive is assigned an index i, and the dynamics governing each primitive are different, as the coupling of the manipulator controls to the object through the contacts, and the possible contact forces applied by the environment, are different. We define the manipulator controls to be $\mathbf{u} \in \mathbb{R}^{n_u}$. In coordinate parameterizations of the configurations of the object and the manipulator, the configuration of the object is $\mathbf{q}_{\text{obj}} \in \mathbb{R}^{n_{\text{obj}}}$, and the configuration of the manipulator is $\mathbf{q}_{\text{m}} \in \mathbb{R}^{n_{\text{m}}}$. The total system configuration is defined as $\mathbf{q} = [\mathbf{q}_{\text{m}}^{\mathsf{T}} \ \mathbf{q}_{\text{obj}}^{\mathsf{T}}]^{\mathsf{T}}$, and the state of the system as $\mathbf{x} = [\mathbf{q}^{\mathsf{T}} \ \dot{\mathbf{q}}^{\mathsf{T}}]^{\mathsf{T}}$. The dynamics during each primitive can then be written

$$\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}, \mathbf{u}).$$

In other words, manipulation is a hybrid system.

Each primitive i has a domain of applicability $\mathcal{D}_i$, i.e., a region in the state-control space where the dynamics of that primitive describes the evolution of the system[1]. The state-control space has $(2n_{\text{m}} + 2n_{\text{obj}} + n_u) = d$ dimensions, and this space is partitioned by the different manipulation primitives. In the example in Figure 1, $n_m = n_u = 3$ and $n_{\text{obj}} = 3$, so the state-control space has $d = 15$ dimensions. A full-dimensional subset of this state-control space corresponds to no contacts: the object is in free flight and the manipulator moves freely. We could call this the "free flight" primitive, a primitive that is always part of a throw. For other manipulation primitives, the state and control constraints implied by active contacts reduce the dimension of the domain of applicability. The "free topple" primitive in Figure 1 is twelve-dimensional assuming that the rolling vertex on the block is prespecified. It can be parameterized by the nine robot states and controls, the angle and rotational velocity of the block, and the position of the vertex on the table. The "controlled topple" primitive is nine-dimensional assuming the endpoint of the robot is in contact and that the rolling vertex is prespecified. This mode can be parameterized by the three control freedoms, the internal configuration and velocity of the robot, the angle and rotational velocity of the block, and the positions of the vertex on the table and the robot on the object. Of course there are also inequality constraints on the states and controls for each manipulation primitive, but they generally do not reduce the dimension of the domain of applicability.

In theory, the entire state-control space could be partitioned into manipulation primitives of different dimensions. Boundaries between these primitives, where a manipulation plan can transition from one primitive to the other, are described by one or more equations that are simultaneously satisfied. For example, the boundary between the twelve-dimensional "free topple" space and the nine-dimensional "controlled topple" space occurs where the conditions of both primitives are simultaneously satisfied, i.e., the controlled topple conditions are satisfied, but the contact force at the robot contact is zero.

Thus we can think of manipulation planning as planning a sequence of manipulation primitives, such that the initial state of the system is in the domain of applicability of the first primitive and the goal state of the system is in the domain of applicability of the last primitive. If the goal is given as m constraints on the final state of the object, then the goal is specified by a $(d - m)$-dimensional subset of the state-control space, which may span more than one manipulation primitive. The manipulation problem can be broken into the following subproblems:

1) **Primitive characterization.** Given descriptions of a robot, object, and the environment, derive a set of contact modes and determine contact constraints, dynamics, and the domain of applicability for each mode.
2) **Primitive sequence planning.** Choose a sequence of N primitives to transition through such that the first primitive contains the initial state and the final primitive contains the goal.
3) **Transition state planning.** Choose a sequence of transition states $\{(\mathbf{x}_p, \mathbf{u}_p) \mid p = 1 \ldots (N-1)\}$, such that each transition lies on the boundary of sequential primitives.
4) **Planning the individual manipulation primitives.** Derive N motion plans, one for each primitive, so that they connect at their transitions in the state-control space, and such that the first primitive begins at the initial state of the system and the last primitive ends somewhere in the $(d - m)$-dimensional goal region.
5) **Stabilizing the individual primitives.** Derive feedback controllers to stabilize the motion plans for different primitives.

In practice, it is not possible to explicitly construct the full partitioning of the state-control space. This problem is harder than explicitly calculating a mathematical representation of configuration space obstacles, a problem that researchers in motion planning have purposely avoided for years. It may be possible, however, to define a (small) library of manipulation primitives, their domains of applicability, and the state-control transition equations between them, such that the domains of applicability cover a good percentage of the state space of interest.

In this paper we first develop a set of steps to address the subproblems above for solving hybrid, dynamic, and nonprehensile manipulation problems. We begin to explore this approach to manipulation planning using the example of a three-degree-of-freedom manipulator and a planar block shown in Figure 2. This builds upon our past work developing individual manipulation primitives such as rolling and pushing using 1-joint dynamic robots [1]. We demonstrate the motion plans experimentally for a block resting on the manipulator with a desired goal state on a ledge outside of the robot's workspace. The planned primitives shown in Figure 6 manipulate the block using a series of rolling and sliding contacts and throw it to the goal state.

B. Paper outline

Section II reviews related work on which this paper builds. Section III gives a general outline to plan for dynamic, nonprehensile, and hybrid manipulation tasks, and Section IV applies the framework to a specific example of flipping up

[1]It is well known that problems in rigid-body frictional mechanics, such as those in this paper, are subject to ambiguity issues, i.e., the same system state $\mathbf{x}$ and controls $\mathbf{u}$ can result in more than one possible $\dot{\mathbf{x}}$. In this paper, we set this possibility aside.

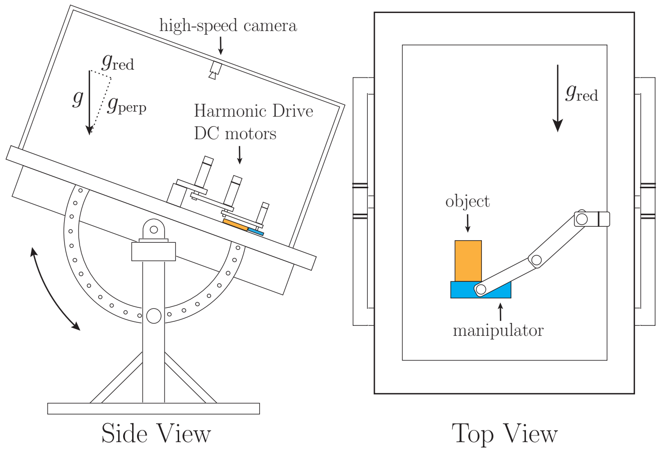

Fig. 2. Our experimental setup consists of an inclined air hockey table with a planar 3R robot driven by brushed DC motors with harmonic drive gearing, and an OptiTrack s250e 250 Hz camera. The angle of the table allows 2D dynamic manipulation experiments in reduced gravity ($0.4g$), and the camera system gives feedback on object positions.

a block at rest on a manipulator and balancing it. Section V describes the experimental setup, and presents preliminary results of planning and executing a block manipulation task.

II. Related Work

A. Hybrid manipulation planning

We use the classical hybrid automata formulation for modeling the manipulation problem. Goebel et al. [2] provide a tutorial on modeling hybrid system dynamics, analyzing stability, and designing stabilizing controllers. Johnson et al. [3] present a hybrid dynamical system model that provides existence and uniqueness guarantees for systems with common approximations such as rigid bodies and plastic impacts.

Motion planners for hybrid systems must reason about trajectories that pass through different contact modes. Trinkle and Hunter [4] extend the dexterous manipulation planning problem to consider rolling and slipping. Erdmann [5] analyzes the task of two palms manipulating a part. Given information about the object, and a desired start and end configuration, the planner determines a set of nonprehensile motions to reorient the part to a goal position. The hybrid planning problem is further developed by Yashima [6] and Miyazawa [7] using randomized motion planning to plan dexterous and graspless manipulation tasks, respectively. Additionally Maeda [8] uses graph-based methods for planning graspless manipulation.

Numerous works have addressed hybrid planning for dynamic manipulation tasks. Furukawa et al. demonstrate dynamic, prehensile, robotic manipulation by tossing a foam cylinder up and catching it [9]. Srinivasa et al. [10] address a dynamic flip-up problem to find motions that tip a block while maintaining a rolling contact with a flat manipulator that can move in a vertical plane. Pekarovskiy et al. [11] calculate optimal batting trajectories for a planar object on an air table and deform them online to send the object to desired goal states.

Some recent works have moved away from the hybrid automata model to formulations that do not treat modes separately. Tassa and Todorov [12] use the method of stochastic complementarity to smooth the discontinuous dynamics of hybrid systems allowing them to be solved by more classical optimization methods. Posa et al. [13] use direct methods and the complementarity formulation to plan motions for dynamic systems with impacts and Coulomb friction.

B. Trajectory control

Trajectory control involves the design of feedback controllers to stabilize dynamic systems about desired trajectories. For linear systems this is often done with LQR control, and many nonlinear systems are stabilized using a time varying LQR controller about a linearized trajectory [14]. Cimen [15] surveys the State-Dependent Riccati Equation (SDRE) control method which parameterizes nonlinear dynamics into a linear structure with state-dependent coefficient matrices. Posa et al. [16] use sum-of-squares methods for computing Lyapunov certificates to show stability and design controllers for rigid-body systems with impacts and friction. Numerous methods and references for the control and stabilization of hybrid systems can be found in [2]. Our hybrid formulation has similarities to those in locomotion research, but we are not generally interested in periodic trajectories (like gaits) which prevents us from achieving cycle-wise stability.

III. Hybrid planning and control formulation

The following is an outline of the hybrid, dynamic manipulation problem we address in this paper. We assume a manipulator and an object interacting in a 2D environment. Given a description of a manipulator and an object, the initial state of the system $\mathbf{x}_{\text{o}}$, and the desired final state $\mathbf{x}_{\text{f}}$, we find controls $\mathbf{u}(t)$ and stabilizing feedback controllers $\mathbf{u}_{\text{fbk},i}(\mathbf{x}, t)$ that bring the system to the desired final state through a set of N contact modes $\{i_p \mid i_p \in \mathcal{I},\ p = 1 \dots N\}$ and transition state-control pairs $\{(\mathbf{x}_p, \mathbf{u}_p) \mid p = 1 \dots (N-1)\}$, while satisfying system dynamics and contact constraints of each mode, transition constraints between modes, and state/control inequality constraints.

A. Primitive characterization

The first step we take to solve the manipulation planning problem is to determine a set of manipulation primitives. Rather than enumerating all possible contact modes, we choose $\mathcal{I}$ desired contact modes from the full set of possible block/manipulator/environment contacts, and then determine the necessary contact and transition constraints for planning through and transitioning between them.

Each mode is defined by a set of contacts that either slide, roll, or remain fixed along the surface of the object. The configuration constraints can be expressed by the function $\phi(\mathbf{q}) = 0$, and the velocity constraints can be expressed as c_i Pfaffian constraints of the form $\mathbf{A}_i(\mathbf{q})\dot{\mathbf{q}} = 0$ where $\mathbf{A}_i(\mathbf{q}) \in \mathbb{R}^{c_i \times (n_{\text{m}} + n_{\text{obj}})}$. These constraint forces and the friction properties determine the total force at the contact. The set of all m_i constraints in each mode i reduces the d

dimensional state-control space to $d_i = d - m_i$. The set of all state-control pairs that satisfy the constraints for a given mode is the domain of applicability $\mathcal{D}_i$.

Each contact mode $i \in \{1 \ldots \mathcal{I}\}$ has a set of corresponding dynamics $\dot{\mathbf{x}} = \mathbf{f}_i(\mathbf{x}, \mathbf{u})$. We define the boundaries between modes by the guard sets $\mathcal{G}_{jk}(\mathbf{x}, \mathbf{u}) = 0$ that represent a transition from mode j to mode k. A guard set is empty if no feasible transitions exist between modes $(j, k) \in \mathcal{I} \times \mathcal{I}$. Transitions include a reset map $\mathbf{x}^+ = (\mathbf{q}, \dot{\mathbf{q}}^+) = \mathcal{R}_{jk}(\mathbf{q}, \dot{\mathbf{q}}^-)$ that maps the pre-transition state to the post-transition state. This reset map is the identity function for transitions that do not involve impacts. For transitions with impacts, it encodes the instantaneous velocity change.

B. Primitive sequence planning

The purpose of the primitive sequence planner is to choose the contact mode order for the motion plan. Given a mathematical description of a hybrid system, the initial state $\mathbf{x}_\text{o}$, and desired final state $\mathbf{x}_\text{f}$ (or m constraints on the goal region), determine the number of contact modes N, and the mode order $\{i_p \mid i_p \in \mathcal{I},\ p = 1 \ldots N\}$ connecting the initial and final states.

We first compile a transition map using the information about the hybrid system derived in Section III-A. This map represents the topology of the state-control space with the $\mathcal{I}$ modes as the nodes, and the feasible transitions $\mathcal{G}_{jk}$ as the edges. The map includes information on whether transitions are smooth or have impacts that cause state discontinuities according to the reset map $\mathcal{R}_{jk}$. It also has information on how many additional constraints are gained or lost when a transition occurs which can give insight into how robust a transition is to state uncertainty. An example of such a transition map is shown in Figure 3 for the block-manipulator system analyzed in Section IV.

Reaching a transition point by moving along the state axis depends on the continuous evolution of the system state, whereas transitions along the control axis can happen instantaneously. For example, we cannot catch an object in free flight until the object and manipulator are in contact, but we can instantaneously release a nonprehensile contact by accelerating away from an object. We can therefore classify transitions as controlled, partially controlled, or state determined, depending on the constraints that are active on the guard set[2].

To create the mode sequence, we first choose initial and final modes with domains of applicability $\mathcal{D}_i$ that contain the given initial and final states $\mathbf{x}_\text{o}$ and $\mathbf{x}_\text{f}$. These modes are not necessarily unique because a given state can be in more than one contact mode. In the simplest case, the desired motion can be achieved within a single contact mode, and in that case $N = 1$ and the high-level mode planner is finished. For the more general case, a motion planning algorithm can be used to plan a sequence of modes through the transition map that connects the initial and final states.

[2]If control rate limits are applied to the manipulator, then all transitions must evolve over time, but we assume direct acceleration control.

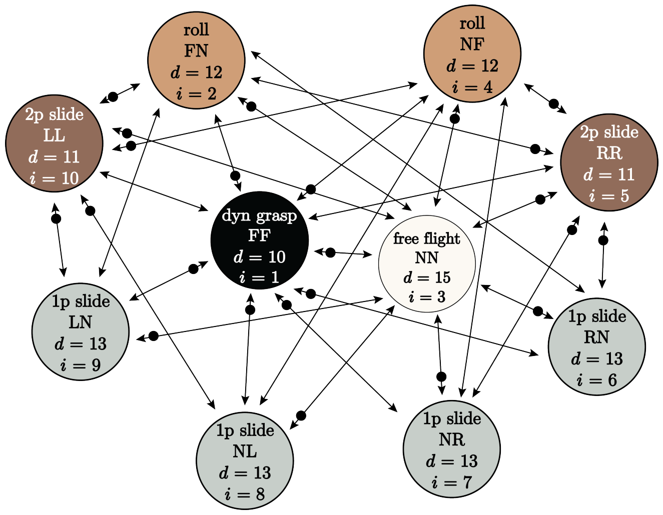

Fig. 3. Diagram showing the transition map for the block-manipulator system in Figure 4. The map only considers two of the corners of the block in contact with one surface of the manipulator. This results in 10 of the 25 total contact modes for the block with one edge of the manipulator. Each node shows the mode name, the contact state at each of the two contacts, the number of free dimensions, and the mode number i. The two letters for each contact represent whether the left and right contact are fixed (F), not in contact (N), sliding left (L), or sliding right (R). The total number of dimensions are the six block states, six manipulator states, and the three controls resulting in a fifteen-dimensional system. Free flight is the only unconstrained mode, and all other modes must satisfy contact and velocity constraints which reduce the dimension. The black dots on some edges indicate that the transition requires an impact to occur.

C. Transition state planning

Once a set of contact modes is chosen, the transition state-control pairs between them must be determined. Given the hybrid system description and the contact mode order $\{i_p \mid i_p \in \mathcal{I},\ p = 1 \ldots N\}$, pick transition states $\{(\mathbf{x}_p, \mathbf{u}_p) \mid p = 1 \ldots (N-1)\}$, where $\mathcal{G}_{i_p i_{p+1}}(\mathbf{x}_p, \mathbf{u}_p) = 0$. For $p = N$, no transition occurs, but we set the desired "transition state" $(\mathbf{x}_p, \mathbf{u}_p)$ as the desired final state of the system $(\mathbf{x}_\text{f}, \mathbf{u}_\text{f})$.

Determining the transition state is difficult because the union of domains of applicability is a lower-dimensional subspace of the full state-control space. Therefore we must have a method to choose transitions that lie on the guard set $\mathcal{G}_{jk}(\mathbf{x}, \mathbf{u})$. This problem is related to sample-based motion planning for robots with pose constraints [17].

D. Single-mode motion planning

With the transition points chosen, a motion planner determines a set of controls for each mode that brings the object and manipulator from the initial state in that mode to the desired transition state out of the mode. Given a hybrid system description, the mode order, and transition states $\{(i_p, \mathbf{x}_p, \mathbf{u}_p) \mid p = \{1 \ldots N\}\}$, the task is to find a time t_p and set of controls $\mathbf{u}(t)$ for $\{t \mid t_{p-1} \leq t \leq t_p\}$ that brings the system from $(\mathbf{x}_{p-1}, \mathbf{u}_{p-1})$ to $(\mathbf{x}_p, \mathbf{u}_p)$ while satisfying dynamics $\mathbf{f}_{i_p}$.

This can be done using a variety of motion planners such as direct and indirect optimization or sample-based methods.

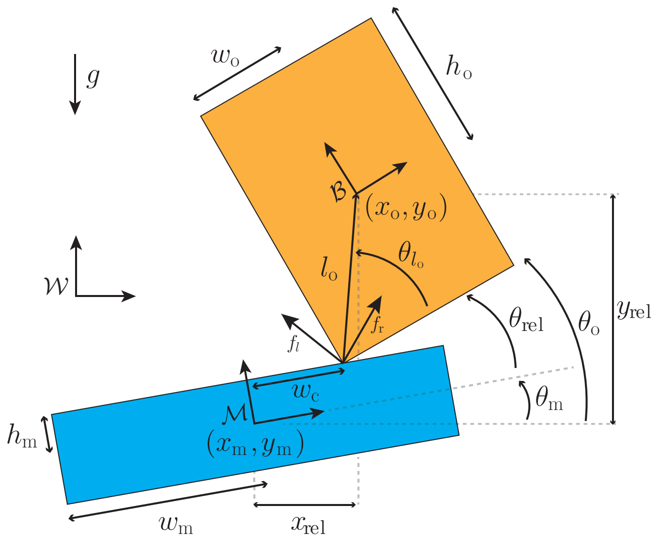

Fig. 4. Diagram showing the manipulator (blue) and the block (orange), relevant measurements and parameters, and the world $\{\mathcal{W}\}$, body $\{\mathcal{B}\}$, and manipulator $\{\mathcal{M}\}$ frames.

The motion planner needs to account for the different system dynamics in each of the modes, and ensure that the desired trajectory does not result in undesired mode transitions. Some modes are underactuated which raises important issues in trajectory planning and control [18] [19].

The primitive sequence planning, transition state planning, and single-mode motion planning can be implemented as an iterative process. If the planner fails at any point, it can return to previous steps and calculate new transition states or mode sequences. This could allow algorithms that are guaranteed to approximately search the space of all possible solutions eventually, either for completeness or optimality. This iterative method can be further extended to reason about hybrid trajectories that are more robust to uncertainty in system state, modeling parameters, and controls.

E. *Primitive stabilization*

Once each of the individual mode plans has been determined, the output is a dynamically feasible trajectory $(\mathbf{x}(t), \mathbf{u}(t))$ for $\{t \mid t_{\mathrm{o}} \leq t \leq t_N\}$ from the initial state $\mathbf{x}_{\mathrm{o}}$ to the desired final state $\mathbf{x}_{\mathrm{f}}$. This trajectory passes through contact modes $\{i_p \mid i_p \in \mathcal{I},\ p = 1 \dots N\}$ and transition states $\{(\mathbf{x}_p, \mathbf{u}_p) \mid p = 1 \dots (N-1)\}$.

The final step is to develop a feedback controller $\mathbf{u}_{\mathrm{fbk},i}(\mathbf{x}, t)$ to stabilize the motion plan about deviations from the trajectory in each mode. Some examples are receding horizon or direct state feedback controllers. The controller commands within each mode i_p must be consistent with the constraints of the current contact mode to avoid causing an unwanted transition between modes. We define $\mathbf{u}_{\mathrm{des}}(t) = \mathbf{u}(t) + \mathbf{u}_{\mathrm{fbk}}(\mathbf{x}, t)$ as the desired control with feedback, and $\mathbf{u}_{\mathrm{proj}}(t) = \mathcal{P}_i(\mathbf{x}(t), \mathbf{u}_{\mathrm{des}}(t))$ as a projection that maps the desired control back to the set of controls consistent with maintaining the current contact mode i.

IV. Block and manipulator example

The general hybrid planner and control system outlined in Section III will now be applied to the problem of a rectangular block and manipulator in contact and moving in a 2D plane. The block is initially at rest on the manipulator, and the desired final state has the block resting on a ledge rotated 180 degrees. We use a set of five primitives to accomplish the goal and these are shown in Figure 6. We chose these primitives because they demonstrate controlled, partially controlled, and state determined transitions between multiple contact modes, as well as feedback control within the rolling balance contact mode. We assume the manipulator's acceleration is directly controlled.

The block-manipulator system is shown in Figure 4. The specific contact mode in the figure is the block rolling about the left contact (mode $i = 2$), but the system description is valid for all modes. All angles and positions are measured with respect to the world frame $\{\mathcal{W}\}$ unless otherwise stated. A body frame $\{\mathcal{B}\}$ is attached to the center of the object, and a manipulator frame $\{\mathcal{M}\}$ is attached to the center of the manipulator. The manipulator's pose is represented by $\mathbf{q}_{\mathrm{m}} = [x_{\mathrm{m}}, y_{\mathrm{m}}, \theta_{\mathrm{m}}]^{\mathsf{T}}$, and the object's pose is represented by $\mathbf{q}_{\mathrm{o}} = [x_{\mathrm{o}}, y_{\mathrm{o}}, \theta_{\mathrm{o}}]^{\mathsf{T}}$. The configuration and velocity of the system are denoted as $\mathbf{q} = [\mathbf{q}_{\mathrm{m}}^{\mathsf{T}}, \mathbf{q}_{\mathrm{o}}^{\mathsf{T}}]^{\mathsf{T}}$ and $\dot{\mathbf{q}} = [\dot{\mathbf{q}}_{\mathrm{m}}^{\mathsf{T}}, \dot{\mathbf{q}}_{\mathrm{o}}^{\mathsf{T}}]^{\mathsf{T}}$, respectively. The full state of the system is defined as $\mathbf{x} = [\mathbf{q}^{\mathsf{T}}, \dot{\mathbf{q}}^{\mathsf{T}}]^{\mathsf{T}}$. The variable $\mathbf{q}_{\mathrm{rel}} = \mathbf{q}_{\mathrm{o}} - \mathbf{q}_{\mathrm{m}}$ describes the relative position and orientation between the manipulator frame and the object frame. The variable $\theta_{\mathrm{rel}} = \theta_{\mathrm{o}} - \theta_{\mathrm{m}}$ is the relative orientation between the object and manipulator. The side lengths of the object and manipulator are denoted by the half-widths and half-heights $w_{\mathrm{o}}, h_{\mathrm{o}}, w_{\mathrm{m}}$, and h_{m}. We assume a Coulomb friction cone model at each contact represented by f_ℓ and f_{r}, with a friction coefficient μ.

A. *Primitive characterization*

In this section we derive some of the $\mathcal{I}$ possible contact modes for the block-manipulator system, along with the mode dynamics and contact and transition constraints. Even for this relatively simple system there are 25 contact modes considering only the block and the top surface of the manipulator. There are many similar modes where relative motion is in the opposite direction causing forces on the other edge of the friction cone, or the corner(s) in contact are different, so we define five classes of contact modes which are shown in Figure 5. A detailed transition map of the contact modes for two corners of the block with one edge of the manipulator is shown in Figure 3. In our motion plan we use dynamic grasp, sliding, free flight, and rolling primitives. For dynamic grasp $(i = 1)$ the block follows the manipulator as long as the forces that must be applied to the block to have it follow the manipulator's trajectory are inside the wrench cone available from the contacts. For sliding regrasp $(i = 5)$ the manipulator accelerates beyond this limit to cause relative motion between the part and the object [20] which is the focus of our previous work. Free flight dynamics $(i = 3)$ involve no contact so the object

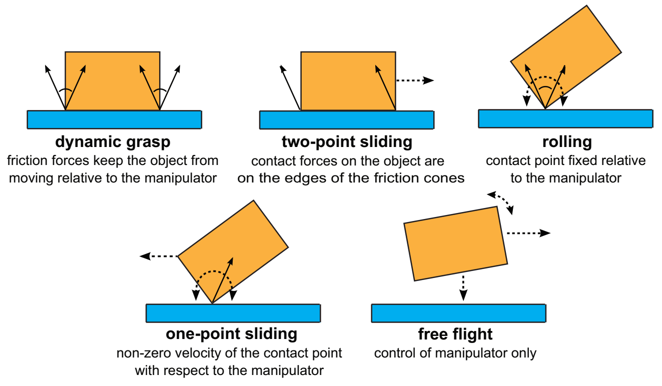
dynamic grasp
friction forces keep the object from moving relative to the manipulator
two-point sliding
contact forces on the object are on the edges of the friction cones
rolling
contact point fixed relative to the manipulator
one-point sliding
non-zero velocity of the contact point with respect to the manipulator
free flight
control of manipulator only

Fig. 5. Diagram showing the different classes of contact modes of the block-manipulator system. The frictional forces are shown by solid arrows and relative motion is shown by dashed arrows.

follows a parabolic trajectory. We derive the rolling mode $(i = 2)$ constraints and dynamics below.

We first define additional parameters shown in Figure 4 before deriving the constraints. We denote the contact location distance along the x direction of the manipulator frame $\{\mathcal{M}\}$ as w_c. The distance ℓ_o represents the diagonal length from the contact corner of the object to its center, and θ_{ℓ_o} is the angle between the base of the object and ℓ_o. The following kinematic constraints keep the block and manipulator in contact.

$$\begin{aligned} x_\text{o} - \ell_\text{o}\cos(\theta_\text{o} + \theta_{\ell_\text{o}}) &= x_\text{m} + w_\text{c}\cos(\theta_\text{m}) - h_\text{m}\sin(\theta_\text{m}) \\ y_\text{o} - \ell_\text{o}\sin(\theta_\text{o} + \theta_{\ell_\text{o}}) &= y_\text{m} + w_\text{c}\sin(\theta_\text{m}) + h_\text{m}\cos(\theta_\text{m}). \end{aligned} \tag{1}$$

Taking the derivative of these constraints yields the Pfaffian constraints $A_2(\mathbf{q})\dot{\mathbf{q}} = 0$, where

$$A_2(\mathbf{q}) = \begin{bmatrix} 1 & 0 & -h_\text{m}\cos(\theta_\text{m}) - w_\text{c}\sin(\theta_\text{m}) & -1 & 0 & -\ell_\text{o}\sin(\theta_\text{o} + \theta_{\ell_\text{o}}) \\ 0 & 1 & -h_\text{m}\sin(\theta_\text{m}) + w_\text{c}\cos(\theta_\text{m}) & 0 & -1 & \ell_\text{o}\cos(\theta_\text{o} + \theta_{\ell_\text{o}}) \end{bmatrix}. \tag{2}$$

We assume that the manipulator is directly acceleration controlled so we can choose desired $\mathbf{u} = [\ddot{x}_\text{m}, \ddot{y}_\text{m}, \ddot{\theta}_\text{m}]^\mathsf{T}$. The motion of the manipulator results in frictional constraint forces at the contact which we denote as f_ℓ and f_r for the left and right edges of the friction cone as shown in Figure 4. Due to the contact constraints and the given contact location with the manipulator w_c, the full fifteen-dimensional state-control space can be represented by an eleven-dimensional subspace. We choose to represent the system as $\mathbf{x}_\text{roll} = [\mathbf{q}_\text{roll}^\mathsf{T}, \dot{\mathbf{q}}_\text{roll}^\mathsf{T}]^\mathsf{T}$, where $\mathbf{q}_\text{roll} = [\mathbf{q}_\text{o}^\mathsf{T}, \theta_\text{m}]^\mathsf{T}$. We then use the constraints in (2) to derive expressions $f_\ell = g_\ell(\mathbf{x}_\text{roll}, \mathbf{u}_\text{m})$ and $f_\text{r} = g_\text{r}(\mathbf{x}_\text{roll}, \mathbf{u}_\text{m})$ which map given state-control pairs to friction cone forces.

We then change coordinates and assume we can directly apply the controls $\mathbf{u}_\text{roll} = [f_\ell, f_\text{r}, \ddot{\theta}_\text{m}]^\mathsf{T}$ which includes the two forces at the contact and the rotational acceleration of the manipulator. Summing the forces and moments acting on the block from f_ℓ and f_r leads to the following control-affine dynamic equations for the object accelerations:

$$\ddot{\mathbf{q}}_\text{o} = \begin{bmatrix} -\frac{\cos(\theta_\mu - \theta_\text{m})}{m_\text{o}} & \frac{\cos(\theta_\mu + \theta_\text{m})}{m_\text{o}} \\ \frac{\sin(\theta_\mu - \theta_\text{m})}{m_\text{o}} & \frac{\sin(\theta_\mu + \theta_\text{m})}{m_\text{o}} \\ \frac{\ell_\text{o}\sin(\theta_\text{m} - \psi - \theta_\mu)}{j_\text{o}} & \frac{\ell_\text{o}\sin(\psi - \theta_m - \theta_\mu)}{j_\text{o}} \end{bmatrix} \begin{bmatrix} f_\ell \\ f_\text{r} \end{bmatrix} + \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix}, \tag{3}$$

where $\theta_\mu = \arctan(\mu)$ is the angle the friction cone makes with the contact normal, $\psi = \theta_\text{o} + \theta_{\ell_\text{o}}$ is the angle of the block centerline ℓ_o in the world frame, m_o is the mass of the object, j_o is the rotational inertia of the object, and g is the gravity acting on the block.

For rolling dynamics we have analyzed a subspace $\mathbf{x}_\text{roll}$ of the total state-space $\mathbf{x}$. As long as the applied frictional forces at the contact $[f_\ell, f_\text{r}]^\mathsf{T}$ are greater than zero, the block is neither slipping on the manipulator nor breaking contact.

B. Primitive sequence planning

We now plan the sequence of modes to move the block from its initial state on the manipulator to the goal state on a ledge rotated 180 degrees. Because the ledge is outside of the manipulator's workspace it is clear that the plan will require a throw to get the object to the desired goal state. A transition map between the various contact modes for the block manipulator system is shown in Figure 3. We manually choose the following mode sequence to achieve the goal:

dynamic grasp + rolling (balance controller) + rolling (roll left controller) + catch + sliding + catch + dynamic grasp + free flight.

In this paper the catch is not so much a primitive as it is a way to transition between primitives while allowing some uncertainty. Although not necessary to achieve this specific goal, the balanced rolling mode is included to demonstrate an example of real-time feedback control during the motion.

C. Transition state planning

The transition into rolling is chosen to have the block near its unstable equilibrium with the manipulator at rest to increase the chance of a successful balance. The transition out of the balance mode is chosen to have the block near the bottom of the workspace to allow more distance to accelerate the block during the throw. The sliding transition is chosen to quickly reposition the block on the manipulator, and the dynamic grasp to free flight transition is chosen so the block will reach the goal state along its free flight trajectory.

D. Single-mode motion planning

With the mode order and transition states chosen, the next step is to determine trajectories in each mode that join the initial state, the $(N-1)$ transition points, and the final state. In this paper we manually generated manipulator motions that followed fifth-order polynomials, and used simulation to verify the block trajectory and check that the contact constraints were not violated. To achieve the desired goal rotation of the object, we first rotate the block by 90 degrees before throwing it. The initial dynamic grasp motion rotates the block up to a goal angle for the rolling (balancing) mode.

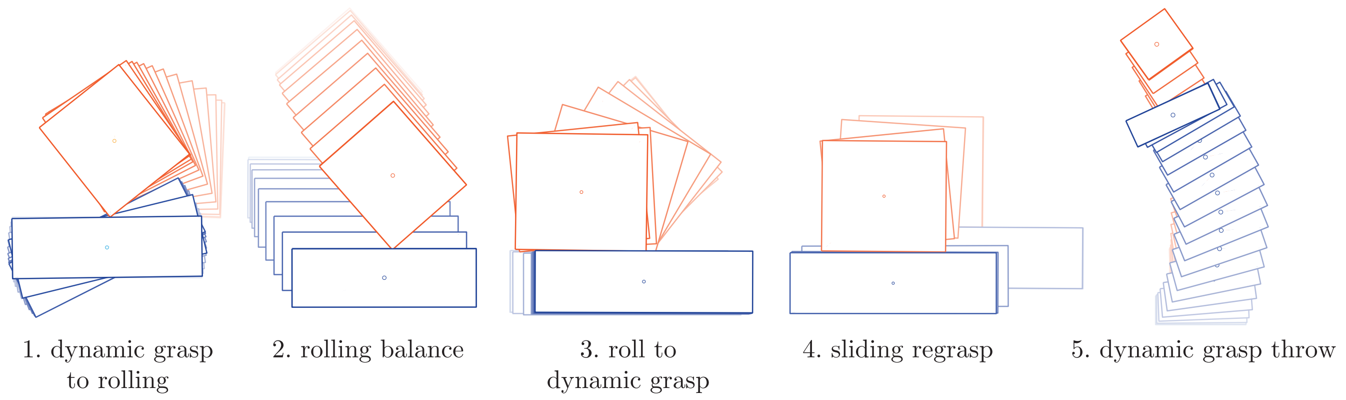

Fig. 6. Diagrams showing the five motion primitives planned and used in the experiment. The figures are generated from experimental trajectories of a block (orange) and manipulator (blue) transitioning through multiple contact modes.

The rolling with the balance controller brings the object to a new position, then accelerates to the right causing the object to roll to the left. A catch allows the block to come to rest on the manipulator before sliding along the surface of the object to reposition it. Another catch allows the block to come to rest before using dynamic grasp and then free flight to throw the block to the goal state. Diagrams of these motion plans generated from experimental data are shown in Figure 6. We have implemented more automated planning methods such as sequential quadratic programming for optimizing trajectories in different modes but that method was not in the experiment in this paper.

E. *Primitive stabilization*

Feedback can be used to account for sources of modeling error and increase the reliability of motion plans. For dynamic manipulation tasks the set of possible feedback methods is constrained by computation time limitations. Control loops running at a high frequency (1000 Hz in our case) provide only a small window to perform calculations and adjust the open-loop motion plan based on system feedback. For this reason we chose to use a linearized LQR controller. The nonlinear dynamics for the block can be approximated as a linear system in a small neighborhood of the trajectory. By linearizing the dynamics about the desired trajectory calculated in Section IV-D, we can create a time-varying state-feedback controller that stabilizes desired motion primitives about the nominal trajectory.

The output of the LQR feedback controller is a set of controls $\mathbf{u}_{\text{fbk}} = [f_{l,\text{fbk}}, f_{\text{r,fbk}}, \ddot{\theta}_{\text{m,fbk}}]^{\mathsf{T}}$. The desired controls are then $\mathbf{u}_{\text{des}}(t) = \mathbf{u}(t) + \mathbf{u}_{\text{fbk}}(\mathbf{x}, t)$. We use a projection method $\mathbf{u}_{\text{proj}} = \mathcal{P}_i(\mathbf{x}, \mathbf{u}_{\text{des}})$ that maps invalid commands to controls that will not cause an unwanted mode transition. For rolling mode ($i = 2$), the projection maps negative contact forces f_ℓ, f_{r} less than zero to zero to satisfy unilateral contact constraints, and saturates infeasible manipulator accelerations. If the nominal trajectory is far enough from the boundary of feasible controls, then small perturbations about the trajectory will be recoverable with the LQR control output.

V. Block manipulation experiment

A. *Experimental setup*

The experimental setup consists of a 3-DOF robot arm that moves in a plane parallel to the surface of an inclined air hockey table. A diagram of the experimental setup is shown in Figure 2. Experiments are conducted at 40% full gravity by inclining the table at 24 degrees with respect to horizontal. Each link is actuated by a brushed DC motor with harmonic drive gearing and current controlled using Junus motor amplifiers. The 1000 Hz motion controller runs on a PC104 embedded computer running the QNX real-time operating system. Vision feedback is given by a 250 Hz IR Optitrack camera. Desired trajectories and experimental results are transmitted between the PC104 and a PC running MATLAB using a TCP/IP connection.

B. *Experiment*

The block is initially at rest on the manipulator, and the desired final state has the block resting on a ledge rotated 180 degrees. We use a set of five primitives to accomplish the goal. The initial dynamic grasp motion rotates the block up to a goal angle for the balancing mode. The controlled rolling primitive brings the object to a new position then accelerates to the right causing it to roll to the left. The manipulator then slides along the surface of the object to reposition it and then throws it to the goal state. Diagrams of these primitives generated from the actual experiment are shown in Figure 6. A set of images from the experiment are shown in Figure 7 and a video is attached in the supplemental media.

The experiment successfully uses different primitives to transition between contact modes during nonprehensile and dynamic manipulation tasks. It also demonstrates the use of a feedback controller during the manipulation task to stabilize the desired trajectory. Most of the planned motions were reliable over multiple experiments, but the basin of attraction of the balancing controller is relatively small. In the future we plan to improve the reliability of the rolling mode controller, develop feedback controllers for additional modes, test the

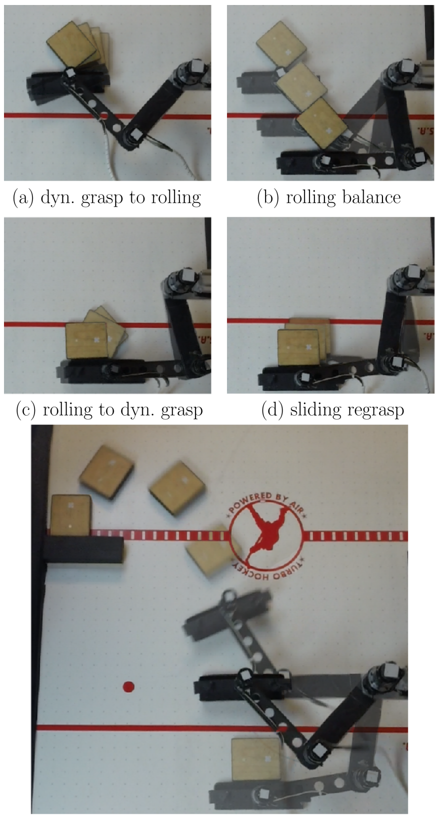
(a) dyn. grasp to rolling (b) rolling balance (c) rolling to dyn. grasp (d) sliding regrasp (e) dynamic grasp throw

Fig. 7. Images from the experiment showing the motion executions.

repeatability of the experimental results, and analyze tracking performance in individual modes and at the transitions.

VI. Conclusions

In this paper we proposed a method for motion planning and feedback control of hybrid, dynamic, and nonprehensile manipulation tasks. We outlined five subproblems—primitive characterization, sequence planning, picking transition states, planning individual motion primitives, and stabilizing individual modes—and then demonstrated the framework by manually planning a sequence of motions for manipulating a block. The motions were demonstrated experimentally with a planar 3R manipulator and block on an inclined air hockey table. Future work will focus on automating the process of generating primitives, choosing mode sequences, planning within single modes, and stabilizing them. We also plan to implement additional real-time nonlinear feedback controllers such as sequential action control [21] to improve the reliability of planned motions. For contact modes where feedback control is impractical we will develop methods to explicitly estimate and manage uncertainty. Future experiments testing these plans will be analyzed in more depth.

Acknowledgment

We would like to thank Adam Barber for his help in developing the experimental setup and Paul Umbanhowar, Jian Shi, and Nelson Rosa for their support and feedback while developing this work.

References

[1] K. M. Lynch and M. T. Mason, “Dynamic Nonprehensile Manipulation: Controllability, Planning, and Experiments,” *The International Journal of Robotics Research*, vol. 18, no. 1, pp. 64–92, jan 1999.

[2] R. Goebel, R. Sanfelice, and A. Teel, “Hybrid dynamical systems,” *IEEE Control Systems*, vol. 29, no. 2, pp. 28–93, apr 2009.

[3] A. M. Johnson, S. A. Burden, and D. E. Koditschek, “A hybrid systems model for simple manipulation and self-manipulation systems,” *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1354–1392, 2016.

[4] J. Trinkle and J. Hunter, “A framework for planning dexterous manipulation,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1991, pp. 1245–1251.

[5] M. A. Erdmann, “An Exploration of Nonprehensile Two-Palm Manipulation,” *The International Journal of Robotics Research*, vol. 17, no. 5, pp. 485–503, 1998.

[6] M. Yashima, Y. Shiina, and H. Yamaguchi, “Randomized manipulation planning for a multi-fingered hand by switching contact modes,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2. IEEE, 2003, pp. 2689–2694.

[7] K. Miyazawa, Y. Maeda, and T. Arai, “Planning of graspless manipulation based on rapidly-exploring random trees,” *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, vol. 2005, pp. 7–12, 2005.

[8] Y. Maeda and T. Arai, “Planning of graspless manipulation by a multifingered robot hand,” *Advanced Robotics*, vol. 19, no. 5, pp. 501–521, jan 2005.

[9] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa, “Dynamic Regrasping Using a High-speed Multifingered Hand and a High-speed Vision System,” *IEEE International Conference on Robotics and Automation*, pp. 181–187, 2006.

[10] S. Srinivasa, M. Erdmann, and M. Mason, “Using projected dynamics to plan dynamic contact manipulation,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3618–3623.

[11] A. Pekarovskiy, T. Nierhoff, J. Schenek, Y. Nakamura, S. Hirche, and M. Buss, “Online deformation of optimal trajectories for constrained nonprehensile manipulation,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 2481–2487.

[12] Y. Tassa and E. Todorov, “Stochastic Complementarity for Local Control of Discontinuous Dynamics.” *Robotics: Science and Systems*, 2010.

[13] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, jan 2014.

[14] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, ser. Dover Books on Engineering. Dover Publications, 2007.

[15] T. Çimen, “State-Dependent Riccati Equation (SDRE) control: A survey,” *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 17, no. 1 Part 1, pp. 3761–3775, 2008.

[16] M. Posa, M. Tobenkin, and R. Tedrake, “Stability Analysis and Control of Rigid-Body Systems With Impacts and Friction,” *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1423–1437, jun 2016.

[17] D. Berenson, S. S. Srinivasa, and J. Kuffner, “Task Space Regions: A framework for pose-constrained manipulation planning,” *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.

[18] K. M. Lynch, “Underactuated robots,” in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds. Springer-Verlag, 2015, pp. 1503–1510.

[19] K. M. Lynch, A. M. Bloch, S. V. Drakunov, M. Reyhanoglu, and D. Zenkov, “Control of nonholonomic and underactuated systems,” in *The Control Handbook*, W. Levine, Ed. Taylor and Francis, 2011.

[20] J. Shi, J. Z. Woodruff, and K. M. Lynch, “Dynamic in-hand sliding manipulation,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2015, pp. 870–877.

[21] A. R. Ansari and T. D. Murphey, “Sequential action control: closed-form optimal control for nonlinear and nonsmooth systems,” *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1196–1214, 2016.