Hybrid Differential Dynamic Programming for Planar Manipulation Primitives

Neel Doshi and Alberto Rodriguez
Dept. of Mechanical Engineering – Massachusetts Institute of Technology
<nddoshi, albertor>@mit.edu

***Abstract*— We present a differential dynamic programming algorithm for planning and closed-loop control of manipulation primitives with frictional contact switches. Executing these primitives is challenging as they are hybrid, under-actuated, and stochastic. Our approach addresses this by planning a trajectory over a finite horizon, considering a small number of contact switches, and generating a stabilizing controller. We evaluate our framework in a simulation study for two primitives, planar pushing and pivoting, and find that we can plan pose-to-pose trajectories from most configurations with only a few hybrid switches (1 to 2) and in reasonable time (1 to 5 s). We further demonstrate that our controller stabilizes these hybrid trajectories on a real pushing system.**

I. Introduction

Seemingly complex manipulation tasks are often composed of a sequence of simpler behaviors. Motivated by this observation, researcher often segment tasks into *manipulation primitives* such as grasping, pulling, pushing, etc. These primitives can then be used to facilitate planning and control for robotic manipulation; however, defining primitives, planning within a primitive, and scheduling primitives are all areas of active research.

This work proposes a computationally efficient planning and control framework based on Differential Dynamic Programming (DDP) for primitives of moderate complexity. Specifically, we develop an approach for executing manipulation primitives with underactuated frictional dynamics that supports a small number of hybrid contact switches. We find that switching contact formations within a primitive increases its expressiveness, which can potentially reduce the total number of primitives needed and ease their scheduling. Specifically, our experiments show that:

- The ability to select and switch contact locations is key to the success of a primitive.
- Only a few (1-2) contact location switches are needed to converge from most initial configurations.

Finally, we show that our framework can plan and control hybrid trajectories on a physical planar pushing system.

II. Related Work

In this section we summarize some related work on manipulation primitives and DDP.

Manipulation primitives Primitives can simplify planning manipulation tasks. Woodruff et al. [1] propose a framework where each contact formation is a different primitive and use this to execute dynamic motions with a fixed primitive-schedule on a physical system. On the other hand, Toussaint et al. [2] use a few expressive primitives to realize diverse set of behaviors; however, this approach is only verified in simulation. Our framework balances these approaches and is similar to that of Hou et al. [3], who develop controllers for two moderate-complexity primitives and demonstrate pose-to-pose re-orientation on a physical system.

Differential dynamic programming Differential dynamic programming is an iterative trajectory optimization method that leverages the temporal structure in Bellman's equation to achieve local optimality. Original developed by Jacobson and Mayne [4] for smooth, unconstrained systems, it has since been extended in many ways, including for systems with linear input constraints [5]. Relevant to our work, Pajarinen et al. [6] consider DDP for planar pushing, and Yamaguchi et al. [7] use DDP to plan for graph dynamical systems.

III. Hybrid Differential Dynamic Programming

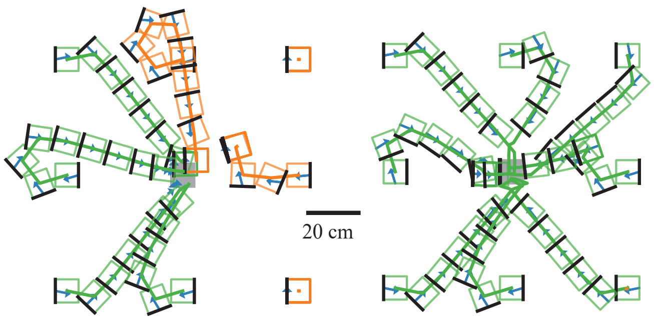
Our algorithm extends input-constrained DDP [5] to systems with hybrid switches. We use DDP to (1) enumerate and rank all feasible mode sequences and to (2) optimize the trajectory and feedback law associated with the best mode sequence. In addition to initial state and input trajectories, we allow the user to specify the maximum number of hybrid switches (N_{switch}) and the set of hybrid modes ($\mathscr{M}$).

We first build a depth $N_{\text{switch}}+1$ tree of trajectories that enumerates all feasible hybrid possibilities. We use input-constrained DDP with a small iteration limit to optimize each edge (trajectory) in the tree and approximate associated cost. Second, we select the branch (a sequence of connected N_{switch} edges) with the lowest total cost, and set the mode schedule to that of the selected branch. Finally, we use DDP to optimize the state trajectory and control law associated with best branch. The hyper-parameters of our algorithm are N_{switch}, $\mathscr{M}$, the planning horizon (N), and the maximum number of DDP iterations during tree generation (N_{iter}).

IV. Simulation Studies

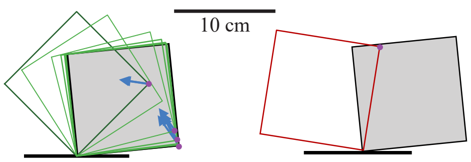
We use our algorithm to plan pose-to-pose trajectories for planar pushing and pivoting. The planner chooses both a sequence of the contacts from $\mathscr{M}$ and optimizes the continuous motion variables. We show a number of trajectories that highlight different aspects of our approach below.

Planar Pushing We present trajectories for available contact sets of dimension one and three from eight representative initial conditions (Fig. 1). The goal is the origin with zero

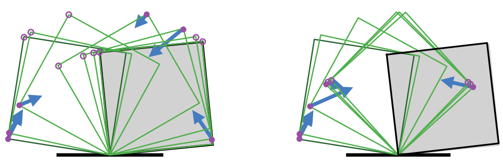


(a) Left pusher (b) Left, top, & right

Fig. 1. Trajectories for pushing from eight representative initial conditions. The goal is a solid gray square, the pusher force is drawn with blue arrows, the left side of pusher is shown in black, and successful (unsuccessful) trajectories are depicted in green (orange).



(a) Different initial conditions



(b) Different number of palms

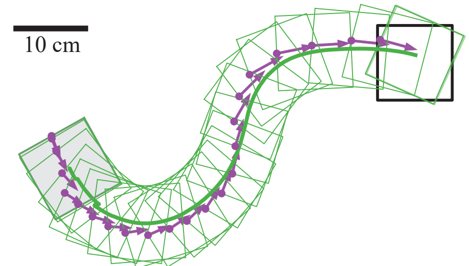
Fig. 2. Trajectories for planar pivoting from two representative initial conditions. The goal is the black-outlined solid gray square, the contact forces are drawn with blue arrows, and available corner-contacts are marked with purple circles with active contacts filled in. Successful and unsuccessful trajectories are shown using the same color scheme as Fig. 1.

orientation. We observe that with only the left contact (Fig. 1a), the algorithm finds solutions for initial conditions that are to the left of the goal. With three available contacts (Fig. 1b), the algorithm finds trajectories that converge to the goal from all initial conditions. It is usually only necessary to select the best contact; however, we see a hybrid switch for a trajectory in Fig. 1b. Finally, the mean planning time is 0.40 and 0.70 s for one and three available contacts, respectively.

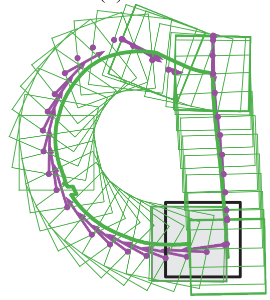
Planar Pivoting Sample trajectories for available contact sets of dimension one, two, and three from two initial conditions are shown in Fig. 2. The goal is at 10° with zero angular velocity. The ability to reason about contact switches is important for pivoting – we cannot pivot from 80° to 10° with only a single active contact (Fig. 2a). Moreover, the planner finds different solutions with two or three available contacts (Fig. 2b). Finally, the mean planning time is 0.67, 3.12, and 7.30 s for the trajectories where planner considers one, two, and three available contacts, respectively.

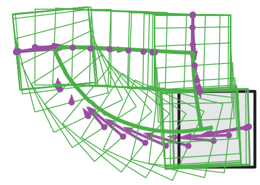
V. Experimental Results

We experimentally validate our framework for planar pushing with a industrial robotic manipulator (ABB IRB 120). The object rests on a flat plywood surface and is moved by a metallic rod attached to the robot. Fig. 3 shows



(a) No switches: $x_0 = (0.5, 0.15, {-\pi}/2)$





(b) One switch $x_0 = (\,0, 0, \pi)$

(c) Two switches $x_0 = (\,0, 0, {-\pi}/2)$

Fig. 3. Three example closed-loop pushes with (a) no contact switches, (b) one contact switch, and (c) two contact switches. The object pose and Cartesian trajectory is shown in green. The pusher location (force) is indicate by a purple circle (arrow). The light-gray box is the initial condition, and the black-outlined box is the goal.

executed trajectories from challenging initial conditions with zero, one, and two contact switches.

VI. Conclusions

We introduce a hybrid DDP algorithm for dynamical system with frictional interactions and discontinuous switches. We demonstrate the ability to plan and control over finite horizons while reasoning about contact switches for planar pushing and pivoting. Moreover, we execute and stabilize planned trajectories on a physical pushing system. However, we find that the final errors are larger for more complex push trajectories. We believe this is due to slipping between the pusher and the slider that is unaccounted for in both the planner and controller, and will address this in future work.

References

[1] J. Z. Woodruff and K. M. Lynch, "Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4066–4073.

[2] M. Toussaint, K. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning." in *Robotics: Science and Systems*, 2018.

[3] Y. Hou, Z. Jia, and M. T. Mason, "Fast planning for 3d any-pose-reorienting using pivoting," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1631–1638.

[4] D. H. Jacobson and D. Q. Mayne, *Differential dynamic programming*. North-Holland, 1970.

[5] D. M. Murray and S. J. Yakowitz, "Constrained differential dynamic programming and its application to multireservoir control," *Water Resources Research*, vol. 15, no. 5, pp. 1017–1027, 1979.

[6] J. Pajarinen, V. Kyrki, M. Koval, S. Srinivasa, J. Peters, and G. Neumann, "Hybrid control trajectory optimization under uncertainty," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5694–5701.

[7] A. Yamaguchi and C. G. Atkeson, "Differential dynamic programming for graph-structured dynamical systems: Generalization of pouring behavior with different skills," *IEEE-RAS International Conference on Humanoid Robots*, no. November 2016, pp. 1029–1036, 2016.