Report for CS 6751, Spring 2017
Grasping and Prehensile Manipulation Project

Vighnesh Vatsal

Sibley School of Mechanical and Aerospace Engineering
Cornell University

1 Literature Review

Grasping end-effectors are one of the most commonly employed manipulation tools in robotic arms. Most literature on grasping-based manipulation [11] considers the scenario in which the robotic arm firmly grips the object, immobilizing it and making it a part of the arm. To place the object in its desired goal configuration is then considered to be a path-planning problem in the C–space of the robotic arm [3], which has been augmented with the grasped object.

On the contrary, humans tend not to manipulate objects in their environment with only pick-and-place style grasping. For instance, to move a box across a table, one may slide it over the surface by pushing [4]. This form of motion is termed as *prehensile* manipulation. In this class of motion, the object is not assumed to be an extension of the manipulator. Based on the geometry of the object and the robot's end-effector, we can identify configurations that either completely immobilize an object (grasps), or constrain it in such a way that it cannot escape to infinity, but is not completely immobilized (cages). The full geometric framework for these conditions is presented in [12].

Broadly, we can distinguish between control and planning for manipulation tasks. Planning methods take the full plan of action into account, generally simulating the whole process offline to check for conditions of feasibility and optimality [9] of the motion. Control-oriented approaches tend to be real-time, sensor feedback-based methods that aid in the performance of the motion at an implementation level [5], generally ensuring local optimality. Task Frame representational approaches [10], and methods involving friction and physics modeling [4] tend to fall into this category.

This project is aimed at developing a manipulation planner for a robot arm that uses the motion primitives of grasping, caging, and pushing, along with arm translation and rotation. Thus far, most prior work on planning has assumed the grasping primitive to be true, thus reducing the problem to finding a path in C–space of the robot. One such set of algorithms distinguishes between *transit* paths, where the object is not in contact with the robot, and *transfer* paths, where the contact is achieved [1]. In the interim of switching between these two modes, it is assumed that an immobilizing grasp can be found. Work specific to prehensile motions (caging, pushing) includes efforts to geometrically model a pushing motion to facilitate a grasp [8], and caging-specific tree-search based algorithms [6]. There have also been efforts to incorporate object physics, friction and mechanics modeling for highly dynamic motion plans that allow for the object to move as an independent object during the plan [7]. Finally, there are task-space based planners that assume a quasi-static motion of objects, while taking into account the uncertainty in their position, implemented using an RRT-based search [2].

I aim to extend the state of the art by generating a planner that can switch between pushing, caging and grasping based on the environment and goal for a particular prismatic object. At present, tree-search and probabilistic roadmap [9] based approaches are being explored where the switching condition emerges as a consequence of a suitable search heuristic cost function.

2 Technical Contribution

2.1 Motion Primitives

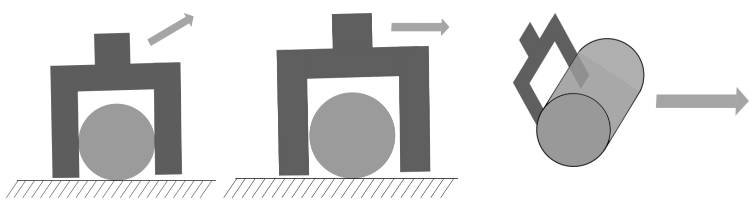

Fig. 1. The three motion primitives in the manipulation task with a two-finger gripper and a pipe: grasping, caging, and pushing

The manipulation task to be performed in this work is for a two-finger end effector-based robotic arm to take an object from an initial state to a specified goal state. To accomplish this, the robot makes use of three manipulation primitives (Fig. 1):

- *Grasping:* As mentioned in Section 1, this is the most commonly employed manipulation primitive in robotic arms. It involves a full force closure on the object, rendering it immobile and effectively converting it into an extension of the end-effector. It allows for the full 6-degrees of freedom (DoF) translation and rotation of the object.
- *Caging:* Caging involves constraining the object such that it is unable to move arbitrarily far away from the end-effector. However, it is free to move within the geometric confines of the end effector. Specifically in the case of a two-finger gripper caging a pipe about its diameter, the pipe is free to move in the axial direction, and free to rotate about any axis. The only constraint is that its center of mass must translate along with the gripper on a flat surface.

– *Pushing:* Pushing differs from caging in the way that the gripper is unable to rotate the object. It can only slide it along a flat surface while preserving orientation.

2.2 System Description

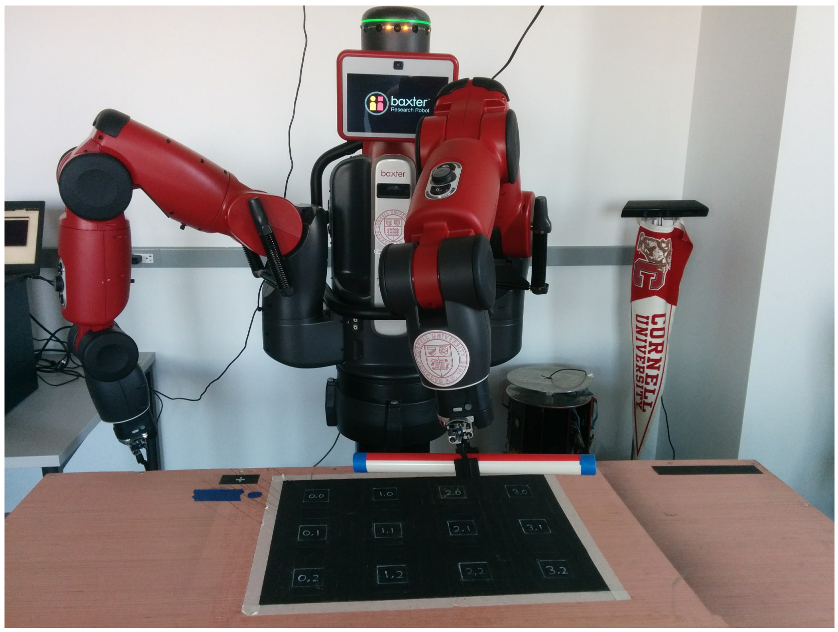

Fig. 2. Baxter robot system setup for the manipulation task

The setup for demonstrating the proposed prehensile manipulation planner consists of the following components: a Baxter robot (Fig. 2), an overhead Kinect camera, a flat table, a raised board, and a marked PVC pipe to be manipulated.

There are underlying simplifying assumptions in the structure of this setup:

– The PVC pipe is specifically chosen because of its prismatic shape for ease of grasping, pushing and caging it by the Baxter robot's gripper. We are not investigating the effect of object affordances in the present planning scheme.
– All motions are performed in a quasi-static manner. This assumption also implies that there is sufficient friction present between every relevant surface to prevent dynamic motions such as rolling of the pipe while pushing.
– There are no obstacles present in this environment.
– The positions the center of mass of every object apart from the pipe are static.
– The pipe always remains coplanar with the flat table surfaces.

2.3 Planning Framework

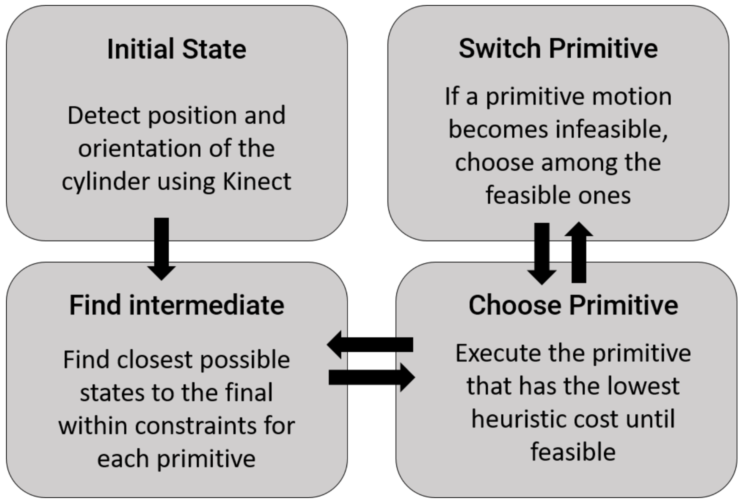

Fig. 3. Planning framework followed for the manipulation task

The pipe is a rigid body, whose state $s \in SE(3)$. It can lie on a series of flat planes such with fixed Z coordinate and prescribed limits on the X and Y coordinates. The problem statement is then to take the pipe from $s_i \in SE(3)$ to $s_f \in SE(3)$ using a sequence of movements of the robotic arm in its C-space. The structure of the problem imposes that only the (x, y, z) and θ (rotation about the Z axis) coordinates of the pipe are variable. These are computed using RBG-sensing of the end-points of the pipe with a known radius. Since planning is performed in the C-space, it is beneficial to formulate the problem in terms of the robot's end-effector. We can use the same four coordinates for this purpose: (x, y, z, θ), but this time of the end-effector. Each primitive imposes constraints on the motion of the robot end-effector:

- Pushing requires that θ and z remain constant, while x and y are variable, normal to θ of the pipe.
- Caging requires z to be constant, all other coordinates are variable
- Grasping imposes no extra constraints

The Open Motion Planning Library (OMPL) allows us to specify initial and final positions for the end-effector in Cartesian space and generates motion plans in the C-space for Baxter's arm.

Once the initial state of the pipe is known, intermediate states $s_t \in SE(3)$ is found that is as close as possible to s_f within the constraints for each primitive, e.g. if the plan involves pure translation in all three directions, the intermediate state for caging or pushing would be $(x_f, y_f, z_i, \theta_i)$, while that for grasping would be $(x_f, y_f, z_f, \theta_i)$

Kinect gives position, orientation of pipe. Then specify end goal position and orientation. Then run three parallel PRM* trees with motion primitives, apply motion with lowest cost based on heuristic. Switch between motions when one cost becomes lower than others, or other two become intractable, like pushing or caging when lifting up. Testing remains to be done.

References

1. Alami, R., Laumond, J.P., Siméon, T.: Two manipulation planning algorithms. In: WAFR Proceedings of the workshop on Algorithmic foundations of robotics. pp. 109–125. AK Peters, Ltd. Natick, MA, USA (1994)
2. Berenson, D., Srinivasa, S.S., Kuffner, J.J.: Addressing pose uncertainty in manipulation planning using task space regions. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. pp. 1419–1425. IEEE (2009)
3. Brock, O., Kuffner, J., Xiao, J.: Motion for manipulation tasks. In: Springer Handbook of Robotics, pp. 615–645. Springer (2008)
4. Chavan-Dafle, N., Rodriguez, A.: Prehensile pushing: In-hand manipulation with push-primitives. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 6215–6222. IEEE (2015)
5. Dafle, N.C., Rodriguez, A., Paolini, R., Tang, B., Srinivasa, S.S., Erdmann, M., Mason, M.T., Lundberg, I., Staab, H., Fuhlbrigge, T.: Extrinsic dexterity: In-hand manipulation with external forces. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on. pp. 1578–1585. IEEE (2014)
6. Diankov, R., Srinivasa, S.S., Ferguson, D., Kuffner, J.: Manipulation planning with caging grasps. In: Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on. pp. 285–292. IEEE (2008)
7. Dogar, M., Hsiao, K., Ciocarlie, M., Srinivasa, S.: Physics-based grasp planning through clutter (2012)
8. Dogar, M.R., Srinivasa, S.S.: Push-grasping with dexterous hands: Mechanics and a method. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. pp. 2123–2130. IEEE (2010)
9. Kavraki, L.E., LaValle, S.M.: Motion planning. In: Springer handbook of robotics, pp. 139–162. Springer (2016)
10. Prats, M., Sanz, P.J., Del Pobil, A.P.: A framework for compliant physical interaction. Autonomous Robots 28(1), 89–111 (2010)
11. Prattichizzo, D., Trinkle, J.C.: Grasping. In: Springer handbook of robotics, pp. 955–988. Springer (2016)
12. Rodriguez, A., Mason, M.T., Ferry, S.: From caging to grasping. The International Journal of Robotics Research 31(7), 886–900 (2012)