BakeBot: Baking Cookies with the PR2

Mario Bollini, Jennifer Barry, and Daniela Rus

I. INTRODUCTION

The creation of a robot chef represents a grand challenge. Accomplishing this requires advances in the fields of computer vision, task and motion planning, and manipulation. We present BakeBot, a first step towards a robot able to execute any recipe. BakeBot is a set of software programs and algorithms for the Willow Garage PR2 robot that enable it to bake cookies autonomously, from *mise en place* ingredient presentation through baking. We begin by discussing the challenges and opportunities posed by cooking tasks, then discuss the solutions we implemented with BakeBot and their performance, limitations, and possible extensions.

II. THE KITCHEN

The kitchen provides a rich array of tasks that range from pick and place through manipulation of semi-solid materials such as doughs and batters. The environment is semi-structured, allowing for many reasonable simplifying assumptions to be made while transitioning a robot control program from the block world or a simulation space. The inherent organization of a kitchen around a human-centric workspace, the relative similarity of kitchen tools and supplies to their peers (most spoons look similar, most bowls look similar, etc.), and the uniformity of kitchen tasks such as mixing, pouring, and placing into an oven provide an opportunity to apply the latest object recognition and motion planning theory in an environment that is familiar and controlled but able to provide additional challenges as the field progresses.

BakeBot bakes chocolate Afghan cookies, a recipe requiring the combination of flour, sugar, cocoa powder, rice krispies, and butter into a homogeneous batter followed by baking in an oven. These cookies are created in a kitchen environment consisting of two work surfaces, one for preparation of the ingredients and another to support a standard toaster oven (preheated to 320^oF). Both tables are immobile during the test, and constant distances are maintained between the robot's starting location, the preparation surface, and the oven. This eliminates the need for localization and navigation, neither of which is the focus of this project. Such capability has been demonstrated on the PR2 and we intend to include it in further generations of the system.

Six items are placed on the preparation table at the start of the baking process: four plastic ingredient bowls of various sizes and colors, a large plastic mixing bowl, and a metallic pie pan. The ingredients are pre-measured and distributed among the bowls on the table. The softened butter is placed in the mixing bowl, primarily to mitigate the risk of spills during manipulation, while the solid ingredients are distributed among the other bowls. The metal pie pan is greased with butter to prevent the cookies from sticking. Pre-measuring the ingredients and arranging them on the work surface dramatically simplifies the task of baking with a robot and is a reasonable simplifying step as it is commonly done for human chefs. The items are arranged in a 2 x 3 grid on the table, with the relative position of every item the same during each run. The absolute positions of the objects are not enforced, though a minimum separation is maintained between the objects.

The PR2 is modified with the attachment of a spatula to the right gripper. The spatula is rigidly attached to the gripper with bolts, eliminating the opening degree of freedom. This side-steps the need to grasp the spatula with the PR2 and to maintain a sufficient power grasp to control the spatula during aggressive mixing motions. The PR2 is covered in water-resistant surgical gowns that have been modified to fit the robot. These gowns protect the robot from splashes and particles. The left gripper is covered in waterproof plastic held in place with rubber bands which also help grab the bowls (as the fingertip rubber pads are covered with plastic).

III. BAKING ALGORITHM

The baking task was broken into subtasks with the intent of creating a library of parameterized motion primitives that can be applied to other recipes and different kitchen tasks. These subtasks are:

1) Locating the ingredients on the table
2) Collecting all of the ingredients into the mixing bowl
3) Mixing everything together into a homogeneous batter
4) Pouring the mixture into the pie pan
5) Locating the oven handle and opening the oven
6) Placing the cookie sheet into the oven

The action plan for cookie baking is currently a hard-coded hierarchical finite state machine composed of individual states representing motion primitives that are parameterized (with environmental and task-specific parameters) at runtime. The state machine was assembled to execute the recipe for chocolate Afghan cookies.

A. *Object recognition*

Object recognition was not the focus of this project. We utilized the ROS tabletop manipulation pipeline to detect the bowls on the table surface. The ingredients were identified by their relative position on the table. We experimented with

The authors would like to acknowledge Anne Holladay for her work on opening the oven door. All authors are with CSAIL, MIT.
{mbollini,jbarry, rus}@csail.mit.edu

Fig. 1. A pictoral timeline of the baking process, clockwise from top left: object recognition, grabbing the ingredients, pouring ingredient into mixing bowl, mixing the ingredients, pouring the batter into the pie pan, scraping the mixing bowl, opening the oven, placing pan into oven, closing oven.

HSB thresholding to differentiate between ingredients and found that it was successful distinguishing the colored bowls from one another but was not adequate for distinguishing between the flour and the sugar powders. We modified the tabletop manipulation package to allow us to detect and grasp objects across the entire tabletop surface.

B. Collecting ingredients

Collecting ingredients into the mixing bowl was accomplished by breaking the subtask into fine-grained components such as grasps, lifts, horizontal transits, and pours. We utilized the tabletop manipulation package for grasp planning and the OMPL planner to find inverse kinematic solutions that satisfied constraints in our Cartesian workspace. Ingredient collection was implemented as a hierarchical finite state machine within the overall system architecture. This compartmentalized individual actions into reusable primitives. The primary advantage of such an arrangement is that it allowed us to use a simple set of decision points to determine whether it was necessary to replan or to choose a different goal pose entirely.

C. Mixing

Mixing is the most manipulation intensive component of the system. It combines joint position control, planning inverse kinematic paths for Cartesian trajectories, and force-compliant control of the end effector.

The PR2 constrains the position of the mixing bowl by executing a grasp on the left of the bowl and maintaining it during the mixing motion. The grasp pose was chosen to maximize the mixing workspace.

The mixing was performed using the `ee_cart_imped_controller`, a ROS package released with Huan Liu, that allows for the execution of force/impedance trajectories in the Cartesian workspace. A trajectory was generated at runtime that pushed the spatula around the inside of the mixing bowl, maintaining constant contact force with the bottom and sides of the bowl. This proved successful in mixing all of the ingredients together into a homogeneous batter. The use of compliance allowed us to be robust to variations in the size and location of the mixing bowl and changes in the consistency of the batter while it is mixed together. Compliant trajectories were also used to scrape the batter off of the sides of the bowl and off of the spatula itself (using the rim of the bowl).

The mixing is performed open-loop with respect to the dough consistency and homogeneity, simply executing a pattern of compliant motions that guarantee that the ingredients will be thoroughly combined.

D. Pouring the final mixture

Pouring the final mixture from the mixing bowl into the pie pan utilized two motion primitives: the pouring primitive, parameterized for the mixing bowl and the pie pan, and the mixing primitive, parameterized for a horizontal scraping motion on the upturned mixing bowl. Balancing the workspaces of the compliant controller on the right arm (with the mixing spoon) and the joint position controller on the left arm (holding the upturned bowl over the pie pan) proved difficult and required considerable trial and error to find a solution that worked consistently. The result of this process is one giant cookie in the pie pan. We leave the cookie as-is because we feel that scooping individual cookies is beyond the scope of this project's manipulation and perception ambitions.

E. Putting into oven

The robot drives from the table to the oven by following an open loop path. The oven handle is located using cloud segmentation in front of the combined surface of the oven door and the hanging tablecloth. The robot localizes itself with this detection and modifies its position to maximize the probability of opening success. Once it has achieved a grasp on the oven handle, a force-compliant trajectory is generated to pull the handle backwards and down. This is effective in opening the oven as long as a quality grasp is achieved. If opening fails, a new grasp is planned and executed.

Once the oven is opened, the robot returns to the table, grasps the pie pan, and returns to the oven. The pie pan is inserted into the oven by planning and executing a path through a series of precomputed Cartesian waypoints while maintaining the horizontal pose of the pie pan. The oven is closed by executing a joint-space path with the left gripper, slowly impacting the oven door and forcing it to close. The oven is reopened after twenty minutes have elapsed.

IV. Conclusions and Future Work

Our goal is to create a robot chef that is able to execute recipes in standard kitchens. BakeBot is a first step towards this goal, demonstrating that a task-based hierarchical state machine combined with basic planning and manipulation are sufficient to accomplish the complicated task of baking cookies. BakeBot demonstrates another application of compliant control to execute trajectories that are in contact with unmodeled surfaces in the workspace. We are currently improving the reliability of the baking action and are starting work to better define the motion primitives to package them for use by high level hierarchical planners.