David Ihim, Abhyudit Manhas, AJ Arnolie
Professor Khatib
CS225A: Experimental Robotics
12 June 2023

CS225A Final Project Report: Robusser

Abstract/Introduction

Our project, titled **Robusser**, aimed to develop a robot that could ease the process of retrieving and washing used dishes in the same way a busser at a restaurant might. More specifically, given a room containing a variety of dishes, including both plates and cups, this robot would be able to easily navigate the room, locate and retrieve such dishes, rinse them in the sink, and then finally place them in the dishwasher to be cleaned. In terms of robot design for this task, we ultimately used a Franka Panda 7-degree-of-freedom robot arm with a 2-degree-of-freedom end effector grabber. This arm was elevated on a 3-degree-of-freedom mobile base such that it could maneuver throughout the room and reach both the tabletop on which the dishes are placed as well as the dishwasher down below. Some of the most significant challenges we encountered while developing this project included robot navigation and collision avoidance, grasping technique for plates and cups, and the integration of human interaction into the system.

Regarding the motivation behind this project, we, as a group, believe the field of domestic service robotics is one that already presents significant value and will become increasingly important in the near future. Automating and easing the process of cleaning up and washing dishes could save significant amounts of time for families, freeing up time for more important and less mundane activities. Furthermore, this project also presents significant value from the perspective of Assistive Care. Especially with nurses leaving the workforce at increasing rates, there is a definite need for technologies to support seniors and individuals with disabilities who currently rely on such nurses to complete basic activities of daily living. A technology like Robusser could automate the processes of cleaning and dishwashing, empowering those who might struggle to complete these tasks on their own to live more independent lives while simultaneously presenting an opportunity for companionship.

Final Implementation

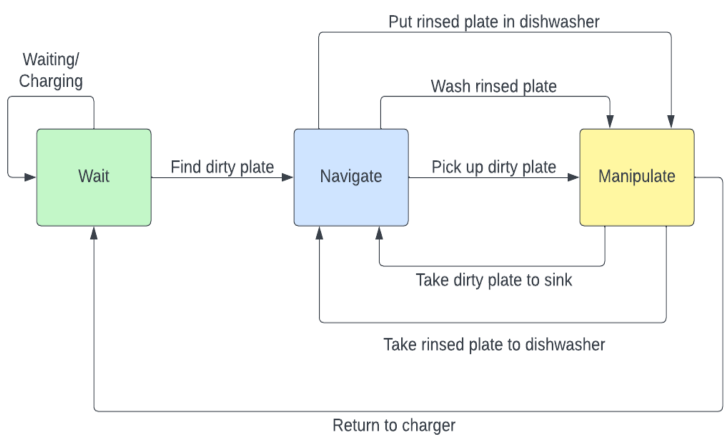

Figure 1. Control state machine for the robot.

Above, in **Figure 1**, we have provided a visualization of the state machine for our robot. Within this state machine, the robot operates under three main states: **WAIT**, **NAVIGATE**, and **MANIPULATE**. Consequently, the overall robot controller was divided into two control modes which encompass these three main states, one in which the manipulator is held static while the base is controlled to navigate the room (**BASE** mode), and one in which the base is held static and the 7-degree-of-freedom arm is manipulated with a position-orientation task controller (**ARM** mode).

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d(q, q_{obs})} - \frac{1}{d_0}\right)^2 & \text{if } d(q, q_{obs}) < d_0 \\ 0 & \text{if } d(q, q_{obs}) > d_0 \end{cases}$$

$$\Gamma_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} \eta\left(\frac{1}{d(q, q_{obs})} - \frac{1}{d_0}\right)\frac{q - q_{obs}}{d^3(q, q_{obs})} & \text{if } d(q, q_{obs}) < d_0 \\ 0 & \text{if } d(q, q_{obs}) > d_0 \end{cases}$$

Figure 2. Repulsive potential and resulting torques for obstacle avoidance. Here, $d(q, q_{obs}) = ||q - q_{obs}||_2$ and $q = [x, y]^T$, x and y being the base coordinates.

The **BASE** mode controller for the robot's mobile base was designed such that the robot could navigate autonomously using a repulsive potential field that prevented the robot from colliding with the table in the center of its designed environment (see **Figure 3**). In order to establish this repulsive potential field for the table, there were assumed to be small circular blobs (obstacles) distributed across the perimeter of the table. This controller was then structured such that joint space tasks used for base navigation as well as the maintenance of the gripper position were priorities, with the maintained joint postures of the arm operating in the null space of these tasks. Specifically, in the BASE mode, our command torques are:

$$\Gamma_{command} = \Gamma_{gripper} + \Gamma_{rep} + \Gamma_{base} + N^T_{base}\Gamma_{arm}$$

Additionally, when **ARM** mode was activated in the controller, the 7-degree-of-freedom manipulator was controlled using an operational space controller using input from the keyboard to modify both position and orientation along and about the X, Y, and Z axes as well as to open and close the gripper at the end-effector of the arm (see **Figure 4** for controls). In this controller, gripper position is controlled by a partial joint task on the two joints corresponding to the gripper fingers, then the arm's position and orientation are controlled using an operational space controller, and finally, the position of the base is maintained using a joint space controller which maintains the location of the base within the X-Y plane. In the ARM mode, the command torques are:

$$\Gamma_{command} = \Gamma_{gripper} + N^T_{gripper}(\Gamma_{posori} + N^T_{posori}(\Gamma_{rep} + \Gamma_{base} + N^T_{base}\Gamma_{arm}))$$

Thus for washing one dish, the robot:

1) switches to the NAVIGATE state and uses the BASE controller to reach the dish
2) switches to the MANIPULATE state and uses the arm controller to pick up the dish
3) switches back to the NAVIGATE state, and reaches the sink
4) switches to the MANIPULATE state, and “washes” the dish in the sink
5) switches back to the NAVIGATE state, and reaches the dishwasher
6) switches to the MANIPULATE state, and places the dish in the dishwasher
7) switches back to the NAVIGATE state, and goes back to its charging station
8) switches to the WAIT state, and waits for more dishes.

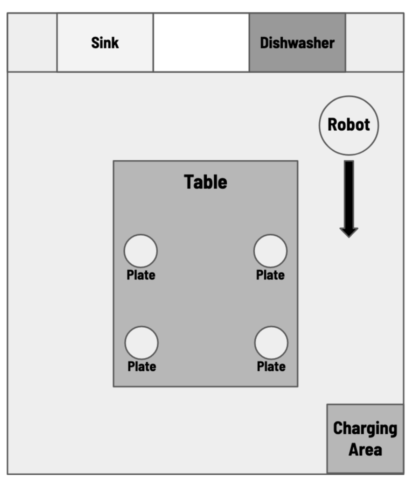

Figure 3. Room Layout for robot environment.

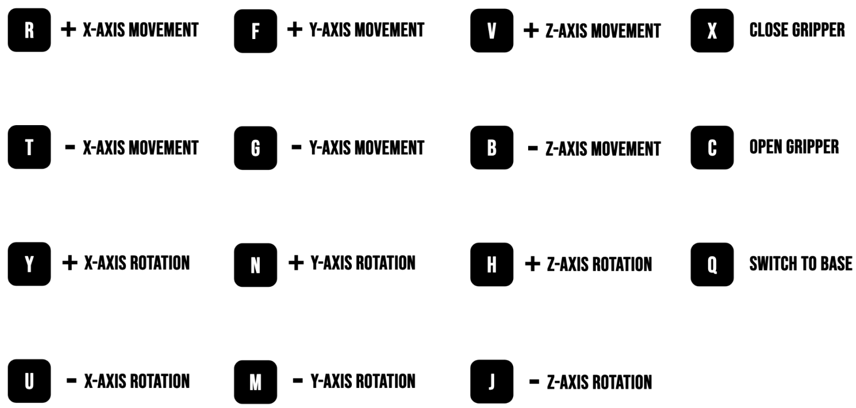

Figure 4. Keyboard controls for manipulator tasks.

Challenges

We certainly encountered a number of major challenges throughout the development of this project. We developed and tested reasonable solutions for the majority of these problems, though there are certainly improvements that could be made to these solutions given more time to iterate on this project. One of the earliest problems we encountered was that of establishing a method with which the robot could navigate around the environment without colliding into the larger objects in the scene such as the table in the center of the room. We initially considered either hand-specifying waypoints within the scene for the robot to travel to or implementing A* on a simplified map of the scene to have the robot generate trajectories for itself. However, we ultimately landed on using the repulsive potential field method for navigation as, for the most part, it would allow us to control the robot to travel to any point in the scene without any additional information, and it solved the problem of collision with the central table that would likely have still been an issue with either of the other solutions. From here, we spent significant time tuning the parameters of this repulsive potential field to ensure the stability of the system and avoid strange issues caused by interaction between the potential field and SAI's built-in collision system. We do believe this was the correct solution to use in this situation, though given another iteration, we might have spent more time tuning parameters to reduce the bouncing effect currently slightly visible at the edge of the bound of the potential field.

Another major problem we encountered and one that we struggled with for the majority of the project was the task of actually grasping dishes once we navigated to them. We initially did not expect to encounter major problems in this area. However, once we had developed a rudimentary system for manipulator control and tested it on the plates and cups we modeled, we found that these dishes slipped out of our gripper much of the time, even when we seemed to have a good grasp of these objects. We tried a number of fixes to this problem, first simplifying the collision meshes of the dishes and the individual fingers of the gripper to be primitive boxes. This seemed to improve our grip, but we were still experiencing slippage much of the time. From here, we tried modifying the inertial properties of the objects to make them less likely to rotate out of our grasp. Increasing rotational inertia did help significantly with grasping objects, but this created problems once we were grasping the object and wanted to move the robot while holding it. Another solution we used involved modifying the dynamic properties of the simulation itself. To this end, we significantly increased both the static and dynamic coefficients of friction for the simulator and set the collision restitution to 0. Ultimately, we tuned the system such that it was fairly consistent at grasping the plates and dishes we had designed. However, given more time, we might have tried other solutions including developing more specialized grippers for each of the dishes that we wanted to retrieve. Perhaps a gripper that had support on all four sides of a glass, for example, would be less likely to drop it.

Finally, we experienced a number of challenges related to integrating user interaction into the system. We initially had several ideas for interesting ways to promote user interaction with the system, from haptic interaction to first-person views of the manipulator using a VR headset. However, a variety of technical roadblocks with these systems along with significant time constraints created by some of the other problems discussed above largely prevented us from furthering the user interaction component of our project past the keyboard controller system that is discussed in sections above.

Results/Conclusion

Ultimately, this project was an incredible opportunity to learn more about the development process for robotic systems and to better understand many of the challenges that come with the development of such systems. Tasks such as grabbing and navigating that we might have originally viewed as fairly straightforward were far more complex to implement and get working in reality and each presented several unexpected challenges. Above all, this project made clear the extensive work necessary to develop seemingly “simple” robotic systems. Additionally, through our work on this project, we received very practical experience developing models in CAD softwares such as Onshape, building out robot systems using the SAI 2 software, and using Redis to interface between different components of the system. Finally, through our development of the various controllers we used to guide our robot through the several tasks necessary to complete the full “dishwashing” procedure, we gained a far more concrete understanding of many of the control concepts that we have been learning about both in Experimental Robotics as well as the Introduction to Robotics course. Below, we have included the links to both a raw successful run of the Robusser system as well as our edited project video.

Raw Dish Retrieval Video: https://youtu.be/ht0SKLGvn68

Full Project Video: **YouTube:** https://youtu.be/0aE7SLBCpgE

Google Drive: Robusser - CS225A Final Project Video.mp4