16-662 Final Report:
Cooperative Manipulation

Clare Cui, Brad Factor, Maitreya Naik, Angad Sidhu, Logan Wan
`ccui, bfactor, mjnaik, assidhu, lwan`

1 Problem Definition

This project was initially proposed by Professor Maxim Likhachev of the Search-Based Planning Lab (SBPL). Our team has demonstrated the overall task of developing planning and coordination strategies for two mobile manipulators that can jointly accomplish a single manipulation task. In this case, we are working with a General Dynamics RoMan Platform robot equipped with a JPL Robosimian arm, as well as a Willow Garage PR2 bimanual humanoid robot. This task has many far-reaching consequences, and arguably one of the most pertinent robotics challenges of our time.

For example, we can consider a common scenario in the home: watching television. Often, when we watch television for extended periods of time, we tend to get thirsty or hungry. In the modern year of 2017, we still need to physically move ourselves into the kitchen, where people typically store beverages and snacks, and move back to the television. We plan on developing the technology and demonstration that will enable a $500,000 robot to fetch your snacks and drinks for you. The future is so close, that we can figuratively taste it.

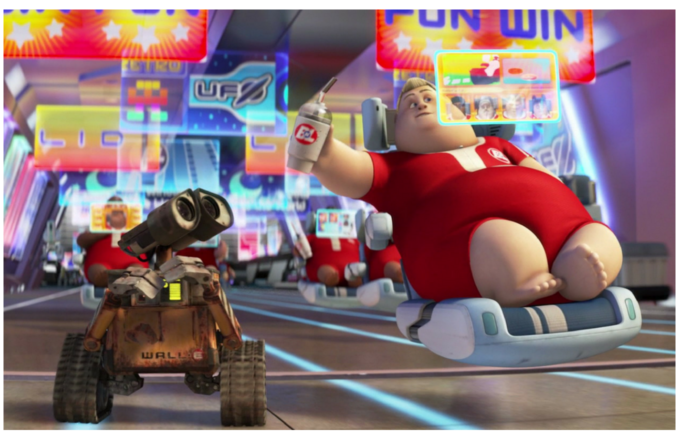

Figure 1: A future we can all get behind

This task involves the integration and engineering of multiple aspects of a robotics system. For the RoMan platform, this involvement is extensive. Since much of the current knowledge of the platform went away with a lab member to JPL, much work needed to be done to both learn how to get it working as well as integrate new features for a National Robotics week demo. For the PR2, this work involved less integration work, and more developing the necessary scripts and software for enabling the handoff demo.

For the robot platforms, the following needed to be accomplished:

1. Perception of the grasping object
2. Grasp generation for the object

3. Inverse kinematics planning to reach the object
4. Additional planning to move the object and place it at a desired location
5. (PR2 Only) Planning where to move to both grasp and drop off an object

2 Related Works

This type of problem is relatively well studied in the context of human-robot handoffs. The extension of this problem to robot-robot handoff does not appear to be researched as much. One such paper that deals with human-robot handoff is from Sid Srinivasa's group here at CMU, titled "Perception and Control Challenges for Effective Human-Robot Handoffs" [1].

This paper discusses (namely) the challenges for a human and robot to handoff items, as well as new technological breakthroughs that enable the enhanced perception that allow for the task to happen. In this paper, the authors introduce a handoff framework that enables more reliable handoffs that don't rely on humans for more complex path planning; rather, the robot is able to more proactively reach for and grasp an object during handoff.

Figure 2: Example human-robot handoff

Another paper that discusses a very similar task is "A simple mechanism for improving performance in multi-robot constrained-space foraging tasks" by Maya Mataric and her lab at USC. This paper deals with a task called "bucket brigading", where a group of robots deals with handling resources from one point to another while dealing with space constraints. They were able to demonstrate the feasibility of their algorithm and approach via simulation.

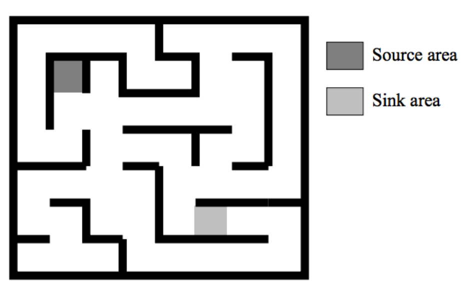

Figure 3: Example resource-constrained environment for handoffs

Similar to the work from Sid's Personal Robotics Lab, Aaron Edsinger from MIT CSAIL also worked on the human-robot handoff problem. However, instead of dealing with the perception and planning of the object, it deals with the human-robot interaction aspect of the task. Thus the robot is able to infer the intent of the task via gestures, and deal with the handoff task appropriately.

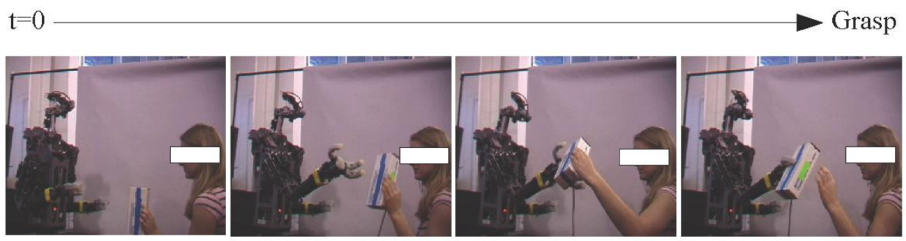

Figure 4: Example handoff task

3 Approach and Implementation

3.1 System

The whole system consisted of PR2 and RoMan executing in concert the whole process of moving an item from a table in front of RoMan to a table on the other side of the room. This consists of the user selecting an item on a web interface. This fires of a message to RoMan to pick that item. Once the RoMan successfully moves the item from the table in front of him to the table next to him, it fires a message over to PR2. PR2 comes over and moves the item the last leg to the serving table on the other side of the room. In Figure 5, the setup of the final demonstration is shown.

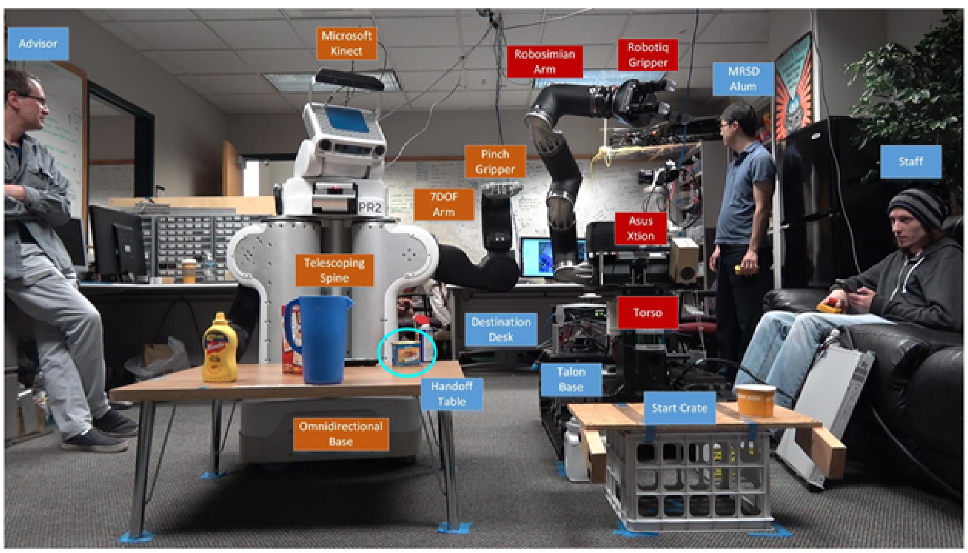

Figure 5: Setup of the final demonstration.

3.2 PR2

3.2.1 Hardware

We had to consider the gripper structure to decide on a pregrasp and grasp that was feasible and stable for most objects, as well as repeatable for either of the arms to execute. Hence, for each object, we had a pregrasp and grasp pair on both sides of the object and a top-down, which turned out to be more stable for object pickup and drop.

3.2.2 Software

For PR2, we addressed the issue of perception with the PERCH[4] algorithm and code base. Originally, the PR2 was perceiving AR tags on simple blocks to calculate potential grasps. This AR tag perception was reused to generate a collision model of the desk where PR2 was to start from, and return to, finally putting the object down. For PR2, we addressed the issue of perception with the PERCH[4] algorithm and code base. To generate the database, we

Figure 6: PR2 Robot

scripted a rosnode to use the ROS transform lookuplistener to find the transform from the perceived object in base_footprint to right_wrist_roll_link which is the right arm wrist joint. Next, we put the PR2 in mannequin mode in order to move and lock the arm in place. Then we use the object to base transform and the wrist joint to base transform, to get the wrist joint pose in the object pose frame according to simple homogenous transformations. We store the translation and quaternions we get for each pre-grasp and grasp into a YAML that another script could load into the ros param server.

To connect with the PR2 pipeline, we wrote a ROS node that subscribed to a topic that the UI published to when a request came through for an object. Upon receiving the request, a set of grasps was retrieved from the rosparam server. After receiving the set of grasps, they were looped through to determine the best grasp to use, given the current object pose. This was determined by calculating the shortest Euclidean distance between the current position of the wrist and its pre-grasp position.

Finally, the motion planning for navigation was accomplished by using the MoveIt planner by PhD students, Sung Kim[6] and Tae-hyung Kim.

3.2.3 Challenges

There were a few issues that we experienced in working with PR2. One of the interesting problems was working with tflistener, which had a function for finding the transformation between two frames. For some reason, the argument order for working with PR2 was different than the one used for Roman, in that the arguments were flipped for the two robots. Another challenge experienced was getting correct grasps for the objects. Since we obtained the grasps manually through the mannequin mode, it was, at times, difficult for PR2 to get to the grasp positions that we had created due to error in the PR2 system. After repeatedly getting new grasps, we were finally able to get a set that worked well in demo. Finally, we also had an issue with PERCH, where we were not able to determine why it was not recognizing a request for a specific object. The reason for this turned out to be latency in initialization on PERCH's part, so we had to put in a one second sleep after the request was sent to get PERCH to accept it.

3.3 RoMan

3.3.1 Hardware

The hardware consists of a TALON robot base which requires a functioning NAVBox for autonomous navigation. For a manipulator we had a 7 DoF JPL Robosimian arm with harmonic drive actuators and a Robotiq 3 fingered gripper. The gripper also allowed us to adjust the grip strength and get a wide and pinch grip of objects. In order to perceive the object we had a Asus Xtion RGB-D camera.

3.3.2 Software

Our software pipeline for RoMan was very similar to PR2. We perceived the pose of the object we needed to grasp using PERCH[4]. This allowed us to find the pose of the object with only three DOF (X,Y,Yaw). This makes a few (reasonable) assumptions for our demo; objects needed to be placed rightside up and must be placed onto a flat (and parallel relative to a base reference frame) table. Once we get a pose of an object from PERCH[4] we can then generate a grasp by picking from a database of pre-recorded grasps of each object. These grasps were manually recorded in advance and were chosen to be robust to small errors in pose estimation and to be as consistent as possible. The general approach was to use "top-down" grasping as that best suited RoMans specific arm and gripper. It also significantly helped reduce the chance of knocking over the object.

We then leveraged the ROS MoveIt framework to generate a plans to various end effector poses. We ended up only including a small amount of information in our planning scene like the table from which RoMan is picking the object and the table to which RoMan is placing the object. The planner also had a reasonable collision model of the arm and gripper itself so as to be intelligent enough to avoid self-collision. Unfortunately the planner knew nothing about the object itself which is why when planning to the grasp we included a "pregrasp". This "pregrasp" pose was generated by taking the grasp pose from the database and translating back by a small constant amount along the x axis of the end effector (which points perpendicular to the palm of our gripper). We would then plan to our grasp pose and actuate our gripper. While this doesn't guarantee we won't knock over or move objects we are attempting to grasp, for our specific environment it turned out to be a simple but robust strategy.

RoMan would then plan from the grasp pose to a target table pose. This was a fixed pose and again we used a "predrop," "drop," "predrop" scheme to help avoid knocking over the object when we moved RoMan's arm out from the workspace.

3.3.3 Challenges

The largest systemic challenge was the lack of consolidated knowledge base and the departure of a significant resource. This resulted in a long, steep learning curve in figuring out how to start up and operate RoMan. Substantial effort was applied to attempting to operate the robot through the RoMan Viewer application without success even following support from General Dynamics engineering. The most recent demo code base was established around the use of a April Tags, while our demo was restricted to using object recognition through PERCH, substantially reducing the utility of the code. Although the Robotiq gripper was highly reliable, the Robosimian arm and torso actuator tended to overheat and lose etherCAT client connection. The complexity of the robot was mirrored in the complexity of the startup procedure, necessitating logging into three to four computers and starting numerous screen sessions for each. In their initial state, the available planners returned success regardless of whether the plan was computed or the trajectory was executed. Unlike PR2, Roman does not have a mannequin mode, requiring nonintuitive manual control to set desired poses.

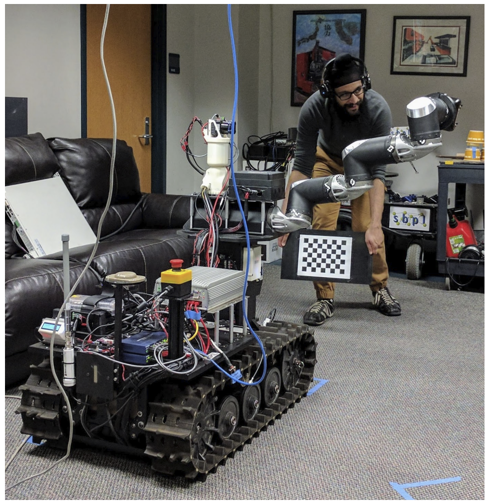

Figure 7: Roman Perch Calibration

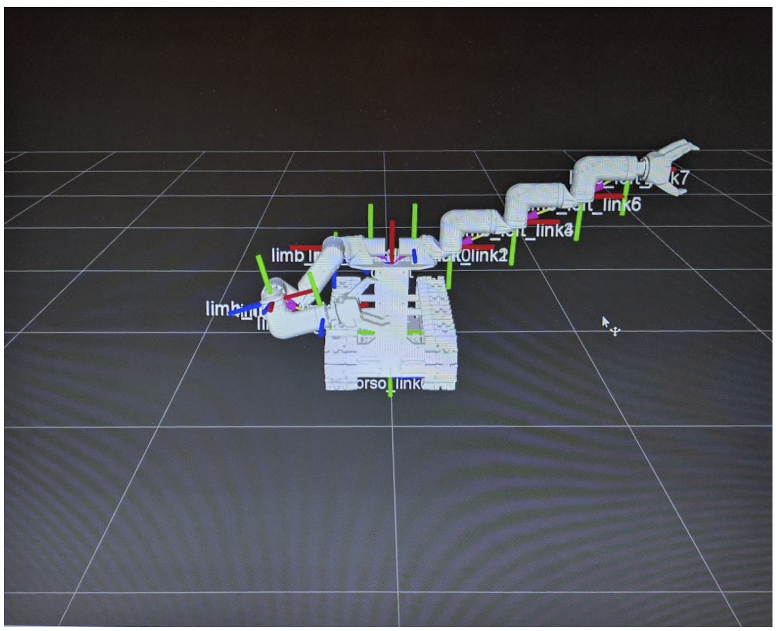

Figure 8: Roman in rviz

4 Results

We successfully were able to complete the demo in time for National Robotics Week. We found RoMan had a success rate of nearly 90% of detecting and moving the object from the table in front of it to the table next to PR2. Unfortunately there were some issues with the PR2 where it would shutdown halfway while executing a plan. This could be resolved just by soft stopping and soft starting the PR2 and it would continue its current plan. Unfortunately this happened throughout our demos which caused it to fail relatively frequently. The goal was to allow for a handoff between both robots instead of having an intermediate table however because of the issues we ran into both on the RoMan, and later on with the PR2, we were unable to meet this ultimate goal. Also because of timing and other challenges we were only able to use one class of object (Spam) to transport from one table to the other. Similarly we didn't bother implementing the logic at the top level which coordinated the robots as we were not often very succesful and manual intervention and restarting was a much more convenient approach.

5 Work Division

At a high level, Clare and Maitreya worked on PR2 and Angad, Brad, and Logan worked with the finicky Roman. We also had some assistance from various members in the SBPL and the breakdown is showed in 9 and 10.

Task	Owner
Integrated Perch into the picking demo	Clare, Maitreya, Venkat (SBPL)
Wrote script to collect grasps in mannequin mode	Clare, Maitreya
Created a grasp database for 4 objects	Clare, Maitreya
Wrote script to return grasps requested by demo pipeline	Clare, Maitreya
Navigation pipeline to and from handoff table	Sung, Tae-Hyung (SBPL)
Compute joint trajectories to object	Sung, Tae-Hyung (SBPL)
Grasping and placing object	Sung, Tae-Hyung (SBPL)

Figure 9: PR2 Work Distribution

Task	Owner
RoMan bringup	Brad, Ricky, Logan
Integrate Perch into the picking demo	Ricky, Venkat (SBPL)
Determined grasps for 2 objects	Brad, Ricky, Logan
Script for establishing grasp and pregrasp locations	Ricky, Brad
Multi-robot communication	Logan
Add collision objects to planning scene	Ishani (SBPL)
Robot action state machine	Brad, Ricky, Logan, Ishani (SBPL)
Compute joint trajectories to object	Brad, Ricky, Ishani (SBPL), Andrew (SBPL)

Figure 10: Roman Work Distribution

6 Final Video

The final video can be seen at: `https://www.youtube.com/watch?v=ReEUrb_i_ng`

References

[1] Vincenzo Micelli, Kyle Strabala, Siddhartha Srinivasa *Perception and Control Challenges for Effective Human-Robot Handoffs*. RSS 2011 RGB-D Workshop, 2011.

[2] Esben H. Østergaard, Gaurav S. Sukhatme, Maja J. Mataric *A simple mechanism for improving performance in multi-robot constrained-space foraging tasks*. AGENTS '01 Proceedings of the fifth international conference on Autonomous agents, 2001.

[3] Aaron Edsinger, Charles C. Kemp *Human-Robot Interaction for Cooperative Manipulation: Handing Objects to One Another*. 16th IEEE International Conference on Robot & Human Interactive Communication, 2007.

[4] Narayanan, Venkatraman and Likhachev, Maxim *PERCH: perception via search for multi-object recognition and localization*. Robotics and Automation (ICRA), 2016 IEEE International Conference

[5] Berk C. Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, Aaron M. Dollar *The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research*

[6] Sung Kyun Kim, CS, CMU, SBPL `https://www.cs.cmu.edu/~sungkyuk/`