MRSD Project Course Team D
Human Assistive Robotic Picker
Entry for the 2016 Amazon Picking Challenge

December 17th 2015

Abhishek Bhatia
Alex Brinkman
Feroze Naina
Lekha Mohan
Rick Shanor

Abstract

Amazon has automated their warehouses by using robots to move storage shelves. However, they still require human intervention to pick each object from the shelf bin and place it into the shipping box which is error prone and expensive. Our primary goal is to solve this problem by developing a robot system that can automatically parse a list of items, identify desired items on a shelf, and pick and place them into the order bin. We have partnered with Professor Maxim Likhachev and the Search Based Planning Lab to compete in the 2016 Amazon Picking Challenge.

Our system, the Human Assistive Robotic Picker (HARP), consists of perception, gripping and platform sub-systems. The perception system identifies items of interest based on their known geometric models. The PR2 robot platform, outfitted with a suction gripper, picks up small household objects from the twelve shelf bins. This semester we have validated the individual subsystems by achieving desired perception accuracies and demonstrating PR2 pick-and-place task planning in simulation. Next semester's primary focuses will be on integration and testing.

This report gives the details of current status and technical analysis of Team Harp's progress toward competing in the 2016 Amazon Picking Challenge.

Table of Contents

1. Project Description 4
2. Use Case 5
3. System-Level Requirements 6
3.1. Functional and Performance Requirements 6
3.2. Non-functional Requirements 8
5. Cyberphysical Architecture 11
5.1. Physical Architecture 11
5.2. Software Architecture 12
6. Current System Status 14
6.1. Fall Semester Targeted Requirements 14
6.2. System Description 14
6.2.1. Platform Subsystem Description 15
6.2.2. Perception Subsystem Description 16
6.2.3. Suction Subsystem Description 16
6.3. System Modeling, Analysis, and Testing 17
6.3.1. Platform Modeling, Analysis, and Testing 17
6.3.2. Perception Modeling, Analysis, and Testing 19
6.3.3. Suction Modeling, Analysis, and Testing 21
6.4. Fall Validation Experiment Summary 23
6.5. Semester Evaluation and Conclusion 23
7. Project Management 25
7.1. Work Breakdown Structure 25
7.1.1. Perception Work Breakdown 25
7.1.2. Grasping Work Breakdown 25
7.1.3. PR2 Control Work Breakdown 26
7.1.4. High-Level Behavior Work Breakdown 26
7.2. Milestone Schedule, 2016 26
7.3. Test Plan 27
7.4. Budget 28
7.5. Risk Management 28
8. Conclusions 30
8.1. Lessons Learned 30
8.2. Key Spring Semester Activities 30
9. References 31

1. Project Description

Rapid growth in the worldwide market for warehouse automation and control systems is being driven by the global boom in e-commerce. Amazon is able to quickly package and ship millions of items to customers from a network of fulfillment centers all over the globe. Amazon sells 306 items per second and has 96 fulfillment centers across the United States. This wouldn't be possible without leveraging cutting-edge advances in technology.

Amazon's cutting edge automated Kiva shelves remove much of the walking and searching for items in the warehouses. However, commercially viable automated item selection and packaging in unstructured environments still remains a difficult challenge. In order to spur the advancement of these fundamental technologies, Amazon organized the first Amazon Picking Challenge (APC) at ICRA 2015. This competition aims to stimulate academic and industry interest in more generalized pick-and-place robotic systems with the ultimate of goal of automating the item retrieval task in Amazon's order fulfilment process.

We are developing the Human Assistive Robotic Picker (HARP) as an entry to the competition 2016 APC. The goal of HARP is to enhance warehouse automation. Such a system would ensure faster processing of packaging and delivering items. HARP will require highly-sophisticated features, such as item identification and manipulation strategy in order to operate in dynamic environment and perform the core functions of item retrieval and item stowage. We will utilize intelligent perception, robust autonomous decision-making, a capable manipulation platform, and an innovative suction system. Team HARP is developing algorithms around the PR2 robot platform to achieve this pick-and-place warehouse task. The PR2 research platform will be used courtesy of the Search-based Planning lab at CMU.

2. Use Case

John's final project demonstration for the MRSD fall validation experiment is due tomorrow. While running his final tests, the primary drive motor burnt out. With limited amount of time and no spares left, John thought he was out of luck. As a last resort, he logged onto amazon.com to check how fast he could receive spare parts. Fortunately for John, Amazon recently implemented the Human Assistive Robotic Picker in its fully autonomous warehouses. John places his order. The order is dispatched to a collection of robots in the warehouse. First, Kiva shelves autonomously drive from storage to their place in the order queue. This is where HARP comes into play.

HARP performs the task of **grabbing** items off shelves and **boxing** them for John. First, HARP **parses** John's order and **determines** the item of interest. Next, sensors **perceive** the position and orientation of the drive motor on the shelf. Then a robotic arm strategically **grabs** the item off the shelf. Finally, HARP **places** the motor into the customer's box.

In less than thirty minutes, Johns motor is out for delivery. Hours later, the box arrives on John's doorstep, just in time to impress the MRSD professors before the demo.

Figure 1: John makes a purchase on Amazon. HARP identifies, grabs, and boxes his item. It is out for delivery in under 30 minutes.

3. System-Level Requirements

The functional, nonfunctional, and performance requirements are driven by the primary objective of creating a pick-and-place robot to compete in the 2016 Amazon Picking Challenge. In reading through the requirements, it is useful to understand the types of items we hope to pick-and-place. The item list from the 2015 Amazon Picking Challenge is shown in figure 2. All requirements are identified as mandatory in order to compete in the challenge in accordance with the competition rules and specifications. From the figure below, it is evident that objects are of different shapes, sizes and transparency.

Figure 2: Items from the 2015 APC

3.1. Functional and Performance Requirements

The functional requirements were written from analyzing the pick and place task. The performance requirements were produced by analyzing the operation of the top three teams during the competition last year. Our goal is to be competitive with these teams by successfully picking three items off the shelf in twenty minutes. Throughout the design process, the performance requirement metrics have shifted as we have learned more about the technical aspects of this problem. Specifically, accuracy requirements for the perception system have decreased. However, requirements for the grasping subsystem have proportionally increased such that our major functional goals are still met.

FR1	Accept order list from user
PR1	Interpret work order with 100% accuracy
Description	The JSON format order list is processed.

FR2	Autonomously determine positions and orientations of target items on shelf
PR2	Autonomously identify object with 50% accuracy
Description	The position and orientation are calculated by the perception module using state-of-the art algorithms. The pose must be determined in order to acquire the objects. Shelf contains up to three items from the item list, non-occluding.

FR3	Accurately determine item grasp position
PR3	Autonomously determine suction grasping surface on 90% of attempts
Description	The perception module outputs position of end-effector for optimal grasping.

FR4	Autonomously picks item from shelf bin
PR4	Autonomously picks item of known pose from shelf bin on 75% of attempts
Description	The kinematics planning is done to pick up the items from the shelf.

FR5	Autonomously places item in order bin
PR5	Autonomously places 90% of picked item in order bin from a height of no more than .3 meters
Description	Once the item is picked, the robot drops it off it in order bin.

FR6	Must follow the dimensional constraints set by Amazon Picking Challenge
PR6	Acquire items from bins located at a max height of 1.86m and minimum height of .78m Acquire items from a .27m x .27m shelf bin Be able to lift items up to .5kg mass
Description	The items and shelf units specified by the Amazon Picking Challenge rules add constraints to our design.

FR7	Does not drop or damage items during grasping from shelf bin or during transportation to order bin.
Description	During robot operation, the robot should not allow items to fall down. The robot should not deform the items in any way. This ensures we are only adding value.

PR7	Acquire at least 3 items in under 20 minutes
Description	Maximize the number of items successfully picked and placed in the given time.

3.2. Non-functional Requirements

Nonfunctional requirements are driven by both the MRSD course and requirements set forth to compete in the Amazon Picking Challenge rules. Specifications, such as the dimensions of the robot work area and of the Kiva picking shelf are shown in figure 3 below.

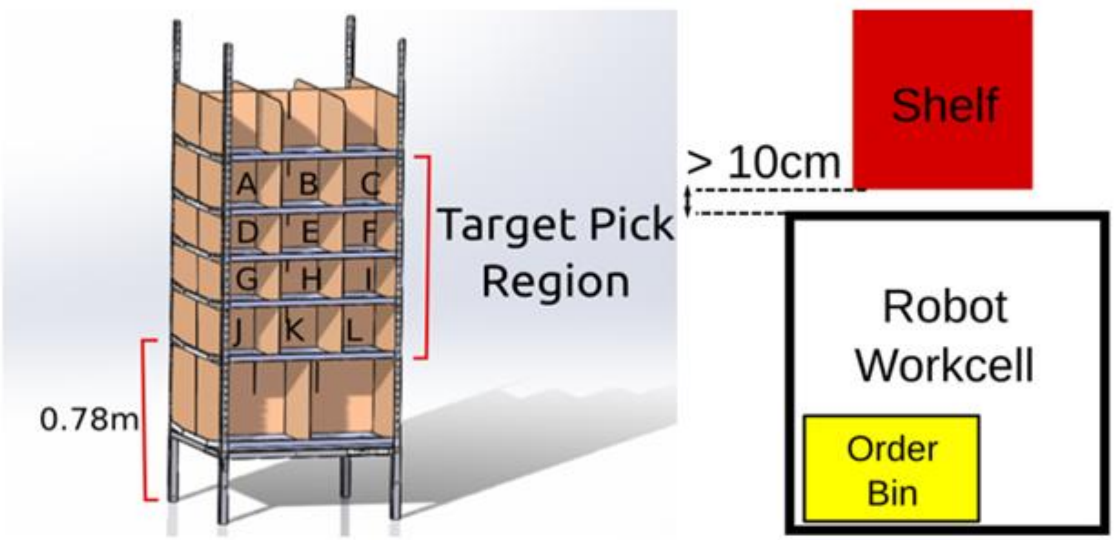

Figure 3: The picking shelf (left) and the robot work area (right)

NF1	Cost no more than $4000
NF2	Be completed by May 1st, 2016
Description	MRSD project requirement.

NF3	Transportable or available at ICRA 2016
Description	The robot should be capable of being disassembled and reassembled easily. Alternatively, the robot platform must be available for use at the ICRA competition in Sweden, Stockholm in May 2016.

NF4	Perception robust to lighting between 320-500 lux
Description	The robot's perception system should operate reliably under different lighting conditions and changes in physical geometry. This is because of the possible variations in test environment and competition environment.

NF5	Be available for testing at least 1 day per week
Description	Algorithms must be tested on the real platform every week to ensure consistency with simulation model. This is a desired requirement which aided in the selection of a suitable robotic platform to develop our system around.

NF6	Start and stay within a 2m by 2m boundary (except the end effector)
Description	The competition rules state that the robot should stay within the 2m x 2m work cell and only the end effector can reach into the shelf. The shelf is at least 10cm away from the work cell area.

NF7	Have an emergency stop
Description	The Amazon Picking Challenge requires a stop button to halt the manipulator platform in case of accidents. This is a safety requirement.

4. Functional Architecture

The functional architecture, shown in Figure 4, can be into four functional areas.

Input Handling: The robot autonomously parses the items in the list to generate an item plan. The input handling function uses an algorithm to select small and easy to grasp items and places them in the beginning of the list followed by larger items and items with no definite shape.

Perception: The perception function is responsible for scanning the shelf, scanning individual bins, determining item pose and providing the system with sufficient data to plan the manipulator trajectory to grasp the item from the shelf bin and place it in order bin.

PR2: PR2 (Platform) function takes the item pose data as input, focuses on kinematics of the path plan to determine the best trajectory (shortest distance and collision free), and moves the manipulator arms to best grasp position. Further, platform function also determines the reverse path to move the arm towards the order bin.

Suction: Suction function decides on the best grasp strategy and orients the end-effector with respect to the object pose. Once the suction arm is close to the object, grasping function switches on the suction mechanism and grasps the object.

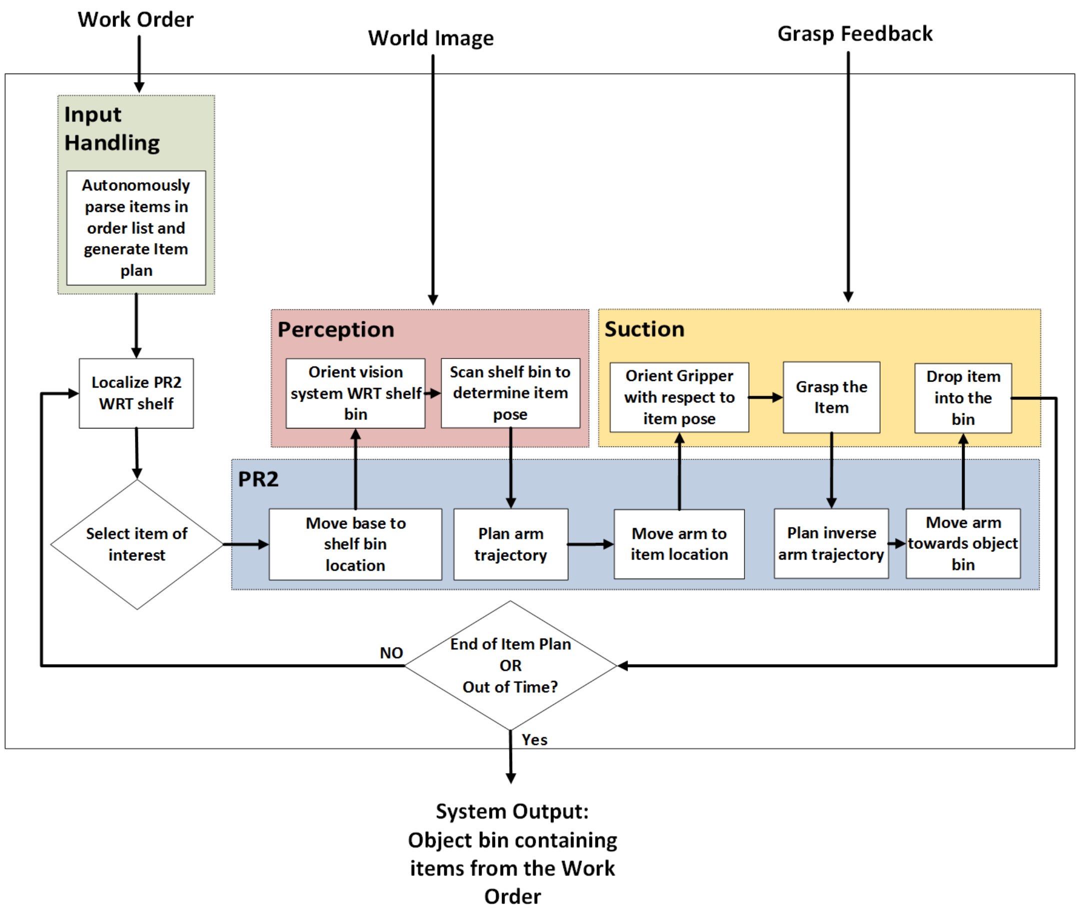

Figure 4: Functional Architecture

The functional architecture highlights the interaction between the four main functional areas. The user passes a Javascript Object Notation file which includes the item contents of all bins on the shelf and the desired order items. The robot generates an item plan to grasp the maximum number of objects in the desired timeframe. To do so, we plan to have an algorithm in place that assigns different weights such as ease of grasping, points associated etc. to each object. Finally input handling subsystem generates a work order that attempts to maximize the overall score.

Next, the PR2 localizes itself with respect to the shelf. Once ready, PR2 begins with grasping and dropping operation on a per item basis. The details of the target item is passed to perception, where the vision sensors scan the shelf to determine item pose and passes the information further to the PR2. PR2 plans the trajectory and moves the base and manipulator arm to the item location.

Once the arm is outside the shelf bin, the grasping system grasps the item using a suction gripper. The robot then moves the base closer to the order bin location and places the object into the order bin. The system re-iterates this loop of grasping and dropping until it has either picked up all items in the dictionary or the time limit has been reached. Once out of loop, robot moves back to the resting area.

5. Cyberphysical Architecture

The cyberphysical architecture is divided into a separate physical architecture and software architecture. The physical architecture describes the interaction of the hardware components and the software architecture describes the actual flow of data and synergy between different blocks.

5.1. Physical Architecture

The physical architecture, shown in figure 5, graphically depicts the interaction of PR2 robot with perception and suction subsystems and various components. A Two Quad-Core i7 Xeon is built onboard the PR2 that runs on Ubuntu 12.04 and ROS Groovy. This computer takes care of the state controller including PR2’s motion planning and arm planning. The gripper subsystem consists of the suction mechanism which is controlled by the ROS state controller through an Arduino microcontroller and relays. The pressure sensor is used to sense the grasp status.

The perception sensor is the Kinect2. The Kinect2 has a much better resolution and depth accuracy than the Kinect and is thus worth the difficulty of adding a networked laptop to the system. However, ROS Groovy does not have the drivers to support the Kinect2. Thus, a separate laptop that runs Ubuntu 14.04 and ROS Indigo and supports USB 3.0 manages the perception algorithms.

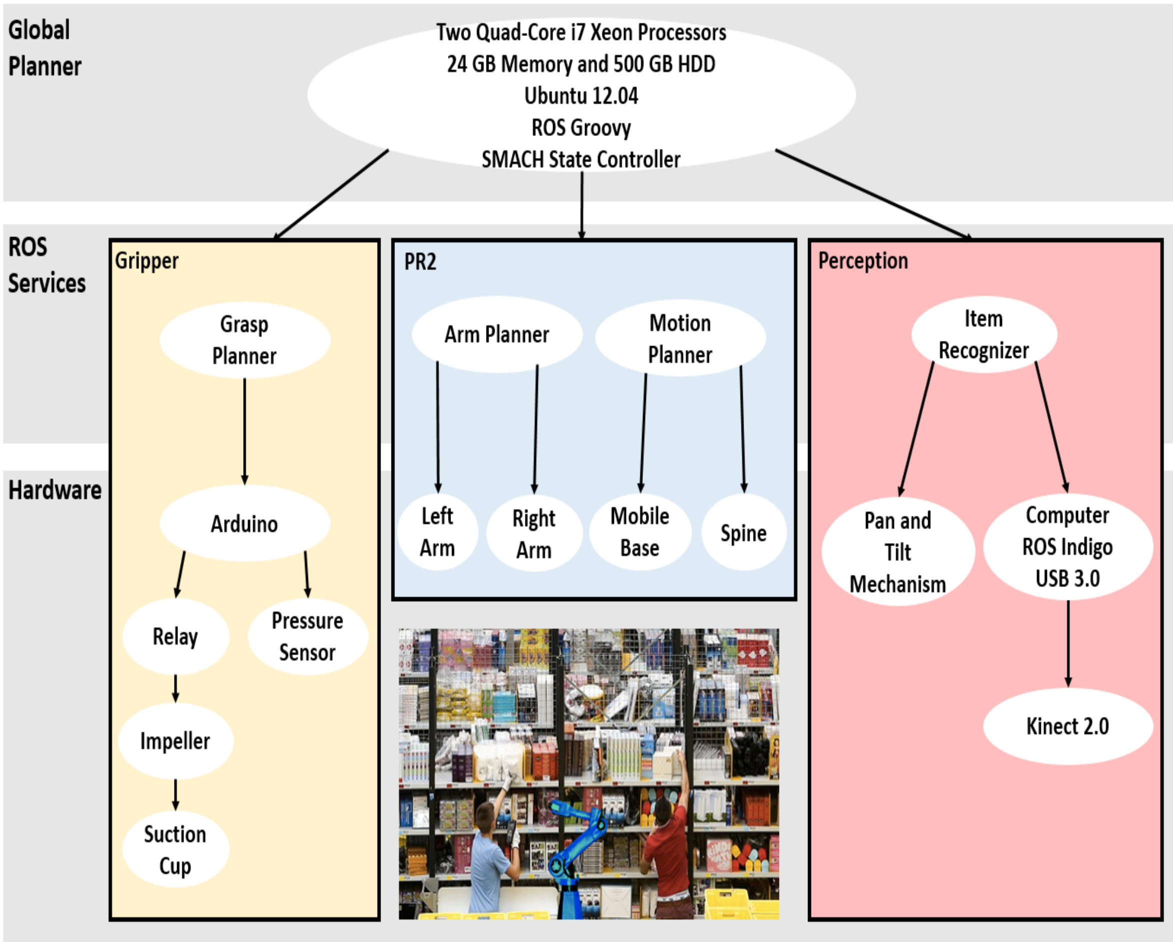

Figure 5: Physical Architecture

5.2. Software Architecture

The software architecture diagram shown in Figure 6 explains the control and feedback mechanisms necessary in order to achieve the desired system functions. The user input, in the form of a text file, is given to the master ROS controller, which begins the SMACH based state machine. In the 'input-handling' state, the master ROS controller passes in an item of interest, which also includes the specific bin number. The robot then aligns the Kinect2 and captures data. This raw data is passed back (over USB 3.0) to the laptop where vision processing happens. The item recognition pipeline then determines the position and orientation of the item on the shelf. Specifically, using image segmentation techniques, bag of words classifiers, and known shelf geometry, the approximate item location is determined. Using this data, the point cloud depth data is down sampled to the region of interest. Using ground truth 3D object data, the known geometry is fit to the depth data acquired by Kinect2. Algorithms available in Point Cloud Library are used to simplify the vision task. This results in a position and orientation of the item of interest relative to the PR2. A ROS wrapper is implemented outside the vision pipeline that takes inputs from the ROS state controller, passes it to the vision pipeline and further takes the vision output (item position and pose information) and passes it to the ROS state controller.

The desired end effector position is passed to the PR2 (Platform) state. The motion planner calculates the path required to move the base to shelf bin and also keeps track of the spine control, moving the spine up and down as necessary. The arm planner creates a series of actuator commands, which are required to position the arm relative to the item. Error collision checks are performed to ensure that the PR2 will not intersect with the shelf. Position feedback, supplied by encoders and other sensors, verifies that the trajectories are executed properly. Once this occurs, the final position of the arm is sent to the ROS state controller.

Finally, arm and item positions are sent to the grasping mechanism. Using trained methods of item acquisition, unique to each item, a grasping plan is generated. A microcontroller is responsible for low level commands. Tactile feedback (from a pressure sensor), indicates successful grasp. Once the item is acquired, the ROS controller receives a grasp success signal from the grasp controller. The motion and arm planner repeats, moving the base closer to the object bin and item from the shelf bin to the order bin. This cycle repeats until all items from the input text file have been acquired.

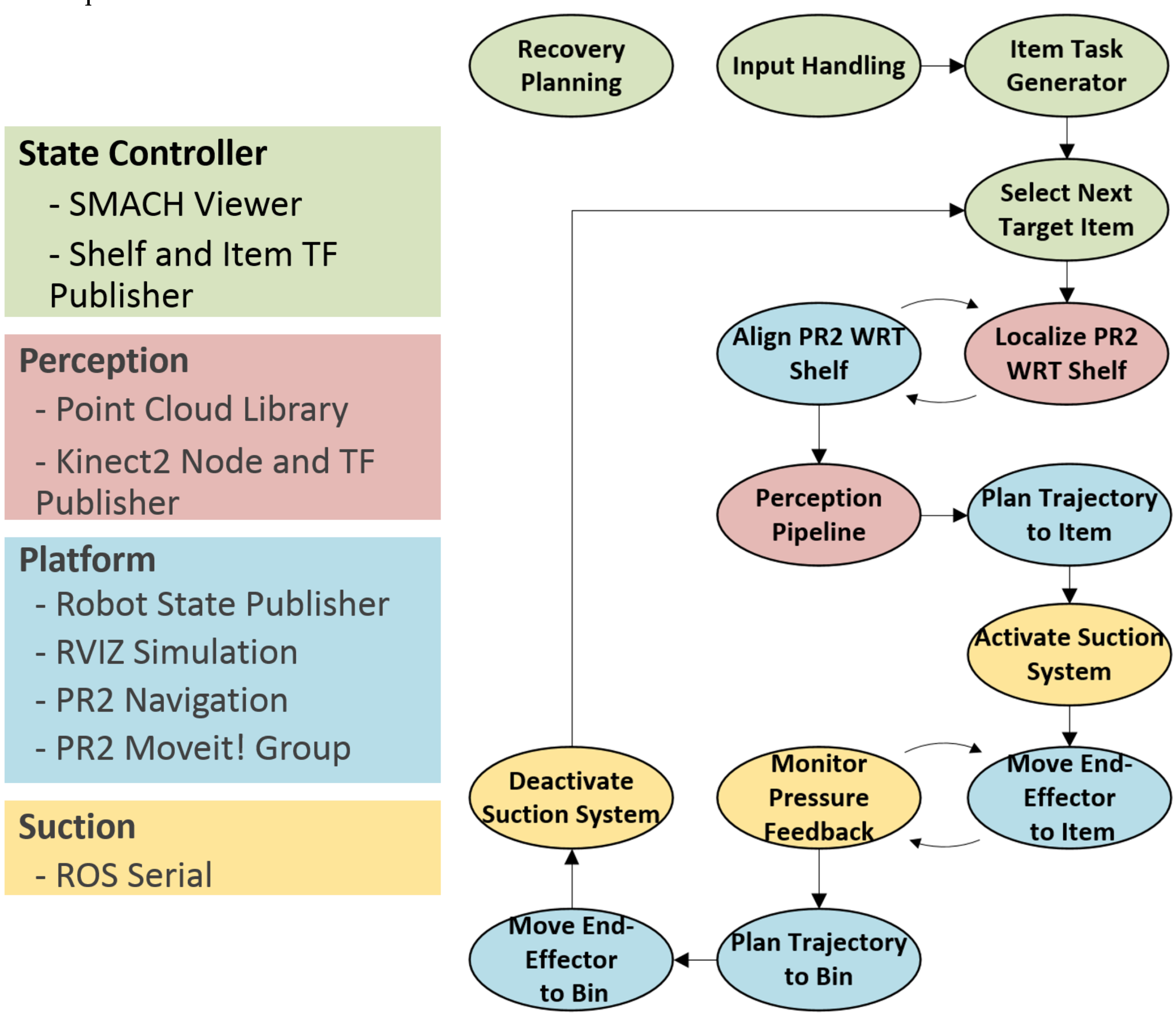

Figure 6: Software Architecture

6. Current System Status

6.1. Fall Semester Targeted Requirements

This semester, team HARP focused on developing key subsystems with the goal of focusing on integration next semester. We targeted the following requirements to validate at the fall validation experiment.

Perception

- FR2: Autonomously determine positions and orientations of target items
- PR2: Autonomously determine positions and orientations of target items with 50% accuracy
- NF4: Be robust to lighting between 320-500 lux

Grasping

- PR6: Be able to lift items up to .5kg mass
- FR7: Does not drop items
- FR8: Does not damage items

Platform

- FR4: Autonomously picks item from shelf bin
- PR4: Autonomously pick item of known pose from shelf bin on 75% of attempts
- FR5: Autonomously places item in order bin
- PR5: Autonomously place 90% of picked item in order bin from a height of no more than .3 meters
- PR6: Acquire items from a .27m x .27m shelf bin
- PR6: Acquire items from bins located at a max height of 1.86m and minimum height of .78m

By validating the perception, grasping, and platform subsystems independently, next semester can be focused entirely on integration and testing.

6.2. System Description

The pick-and-place actions are being developed around the PR2 robotic platform. This system, shown in figure 7, consists of several major components. The platform is the PR2 itself, a two armed mobile manipulator with omnidirectional base and a pan-and-tilt head. The perception system consists of a head-mounted Kinect2. Finally, the grasping and manipulation consists of custom designed suction cups held by the PR2's two finger gripper.

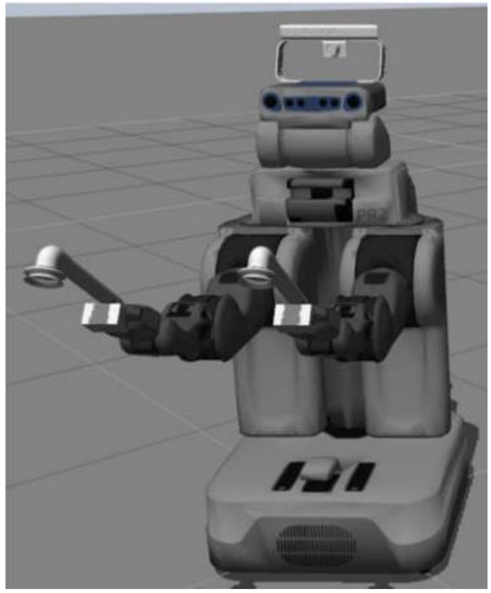

Figure 7: The PR2 with head mounted Kinect and two finger gripper

6.2.1. Platform Subsystem Description

We are using the Willow Garage PR2 robot platform for our project. It has two integrated computers with running Core i7 processors with 24GB RAM. It has two 7 DOF arms with a two finger gripper each. The maximum payload capacity at the end-effector is 1.8kg. It has an omni-directional base. The torso can be extended to reach up to the second-from-the-top row of shelf bins (1.55m). The platform is fully integrated with ROS and runs on Ubuntu 12.04 LTS with ROS Groovy. These components are called out in Figure 8.

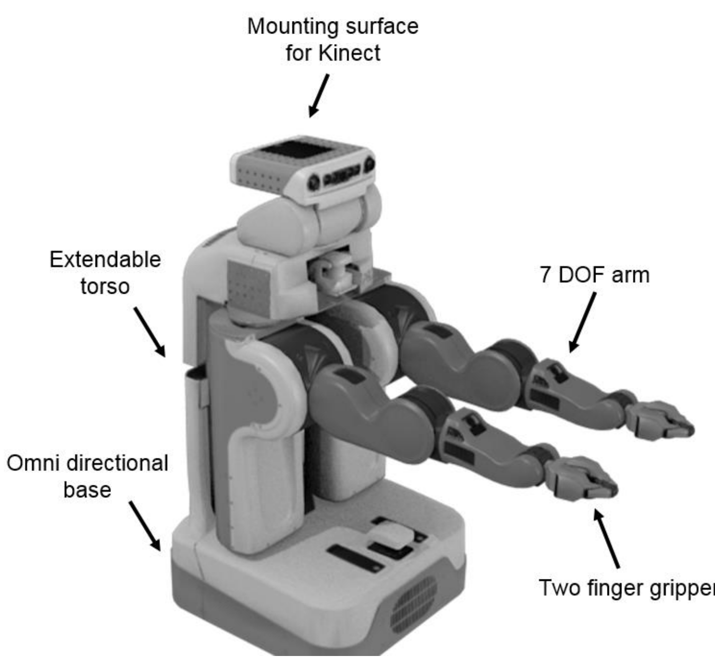

Figure 8: The PR2 Robotic Platform

6.2.2. Perception Subsystem Description

The perception subsystem uses ground truth .stl models generated from Rutgers University to identify item. The overall perception pipeline is shown in Figure 9. First, the Kinect 2.0 image is grabbed using the libfreenect2 tool. The image has to be converted from a depth map to a point cloud. The Point Cloud library is used to analyze the shelf data. First, the shelf is filtered out from the depth cloud based on the known geometry of the shelf with respect to the location of the sensor. Next, Euclidian segmentation is used to isolate each item on the shelf. A wrapper function ensures that the number of clusters identified is equal to the number of items expected on the shelf. Finally, items are identified and their pose is estimated using an Iterative Closest Point algorithm. Every item in the scene is compared to every ground truth model of expected items on the shelf. Matches are scored based on a custom metric counting inliers and outliers. A matrix is created and the shelf score is maximized based on each item-ground truth pairing.

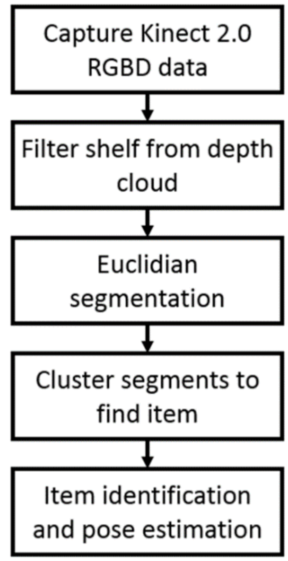

Figure 9: Vision Pipeline

6.2.3. Suction Subsystem Description

Initial trade studies showed suction systems would be far superior to traditional grippers for this task. After prototyping several solutions, we determined a high flow system was required to deal with imperfect seals of porous items. A custom suction cup gripper was designed which can be held by PR2's two finger gripper. The gripper shown is capable of acquiring all objects from the 2015 amazon picking challenge list as shown in Figure 10.

Figure 10: Suction Gripper

All the electronics for the suction system are packed into the enclosure shown in Figure 11. A custom PCB holds an Arduino, reads up to four analog pressure sensors, and controls two AC relays. The box contains inlet power connector, two output plugs to connect two vacuums, two serial connectors to attach up to four sensors, and status LED's. Pressure sensors installed on the vacuum hose detect when a drop in pressure has occurred, indicating that an item has been acquired. This subsystem communicates with the main computer over ROS serial and is controlled by the main state controller.

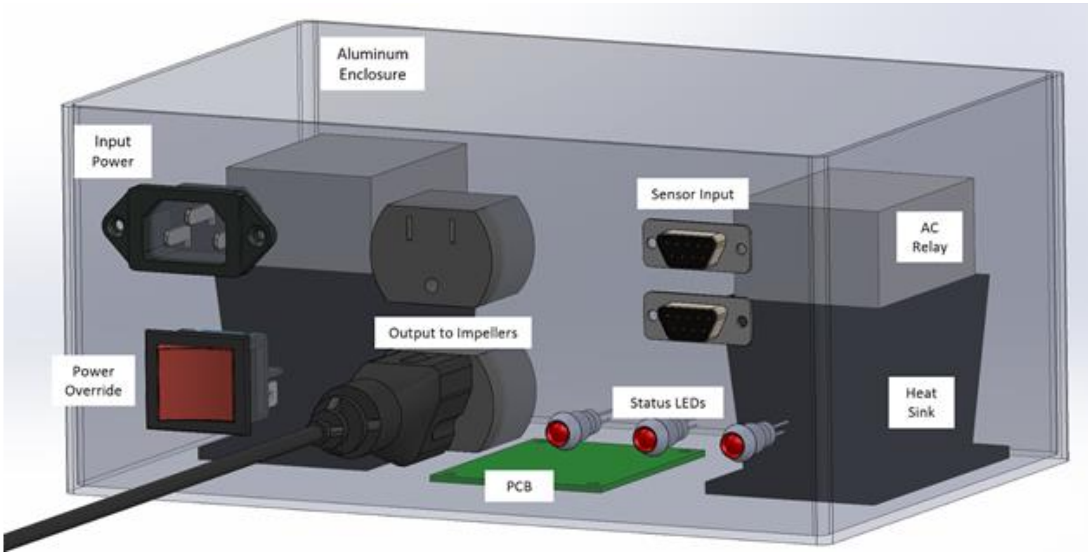

Figure 11: Suction Electronics Enclosure CAD

6.3. System Modeling, Analysis, and Testing

6.3.1. Platform Modeling, Analysis, and Testing

The PR2 platform is initially tested using the Gazebo simulator, allowing us to verify that our path planning is working correctly. The shelf is visualized inside RViz using display markers. The pose of the bottom left edge of the shelf bins are published as transforms in ROS. Figure 23 shows the PR2 and shelf simulated in RViz.

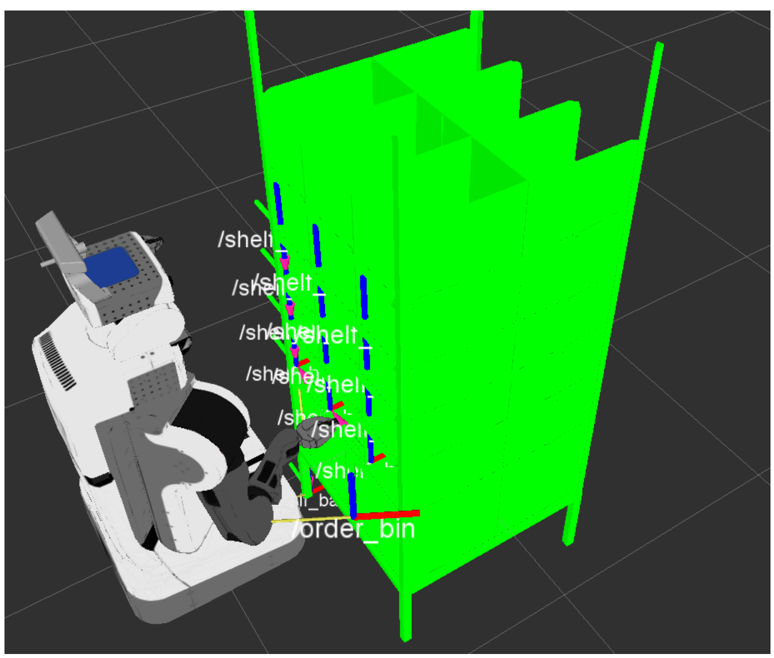

Figure 12: PR2 and Shelf Visualized in RVIZ

The MoveIt! package is used for planning and executing arm trajectories. It allows motion planning libraries to be easily swapped using plugins. Currently, we are using the OMPL plugin for path planning. We will eventually be switching to the search based planner plugin being developed by SBPL as it is expected to be faster. For the base controller, we are using proportional control.

An executive smach state controller is used to define tasks as states. The smach library allows us to easily visualize and debug the state machine. Figure 13 shows the state machine during run-time. Each state is represented as a node. They are connected with pointed arrows denoting the transition condition. The current state is highlighted in green.

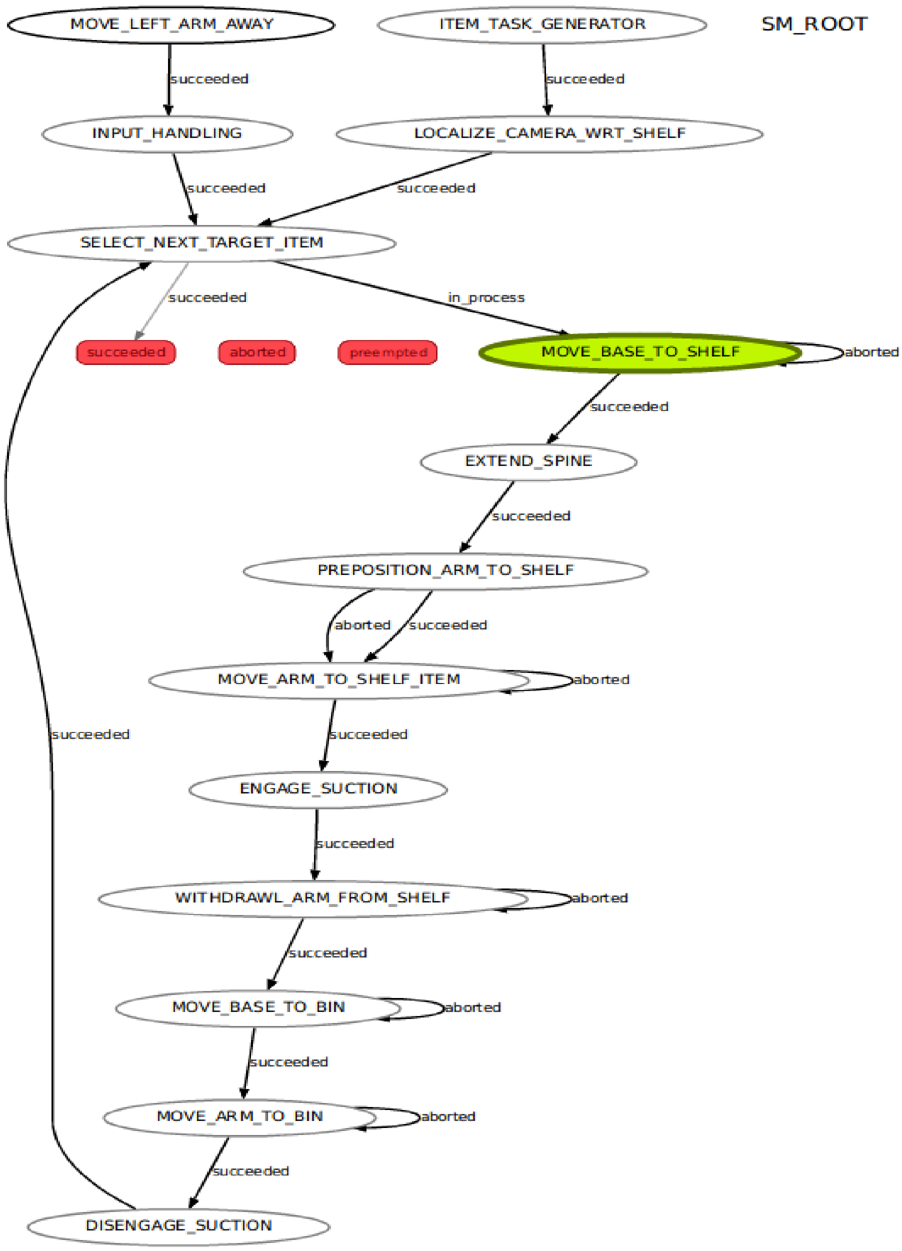

Figure 13: Executive SMACH State Controller

6.3.2. Perception Modeling, Analysis, and Testing

First, vision algorithms were developed around a huge database provided by Rutgers University gathered during last year's competition. This allowed for quick prototyping before we had the Kinect 2.0 running and before we had acquired our own set of test items. Using this dataset, a 1000 image test was run. The algorithm accurately predicted the location of the item on the shelf to within 3 cm on 56% of attempts. The error distribution is shown in figure 14. The images from the database had been collected using the original Kinect. This performance indicated the need to transition to the more accurate Kinect 2.

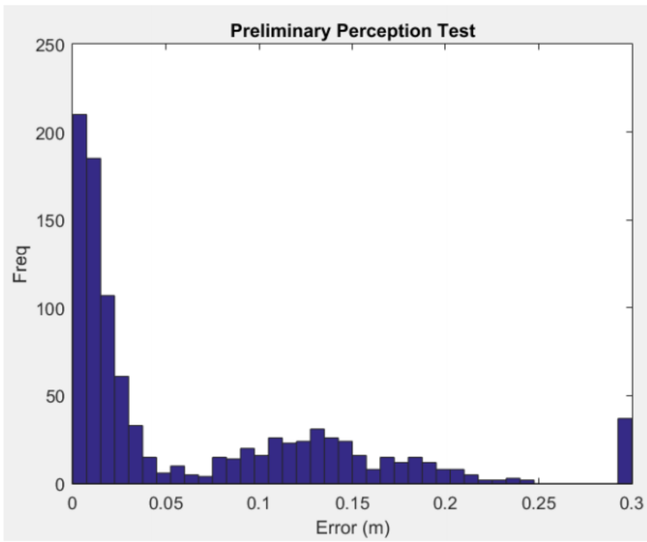

Figure 14: Perception results on the Rutgers dataset

After getting the vision pipeline running on real time data, a test setup was developed simulating a typical shelf. This configuration, shown in figure 15, allowed to test configurations in real time. A command line tool was developed in order to input item contents into the perception algorithms.

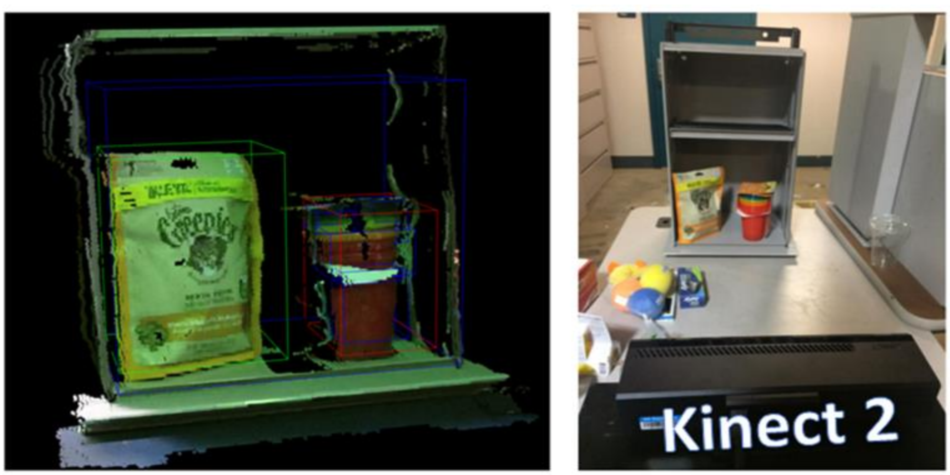

Figure 15: Kinect 2 test setup

For the fall validation experiment, we showed that the vision system has an accuracy of above 50% for one, two, and three item shelves. In addition, by implementing algorithms that only use depth data, the system is robust to changes in lighting conditions. Figure 16 shows examples of the input and output of the perception system. The kinect2 captures a shelf image, predicts item location, and projects the ground truth model into the scene.

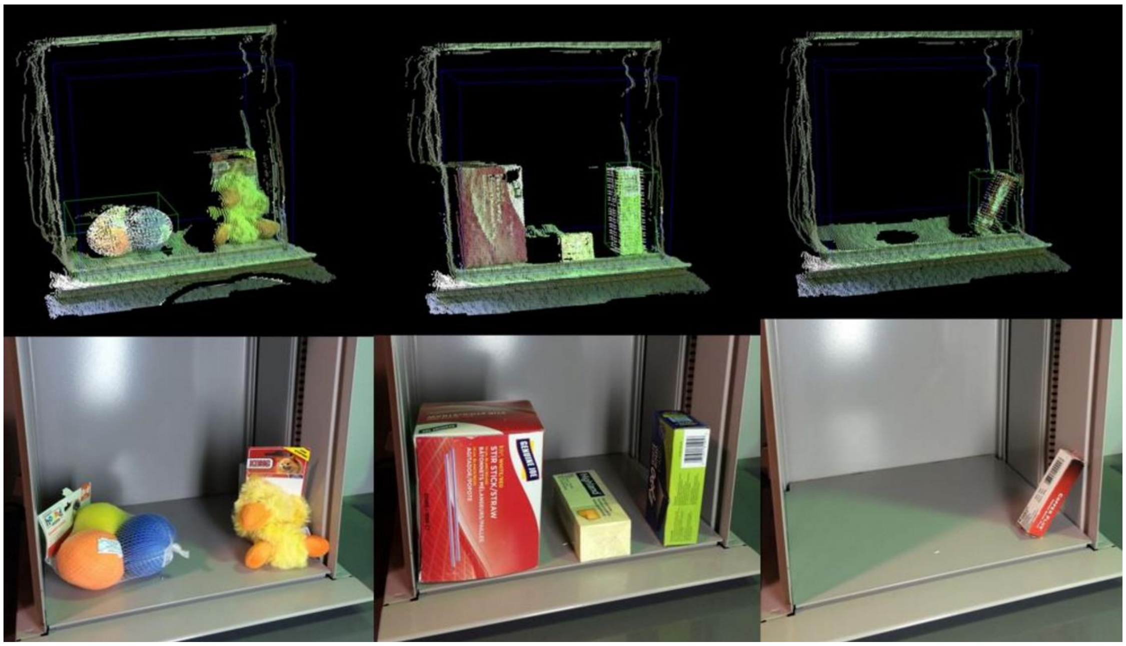

Figure 16: Successful perception output

A series of tests was run on two and three item shelves in order to better understand not only how accurate our perception system was but also understand where it fails. Accuracies are given in Table 1. One item identification is almost perfectly accurate, and two item was very good as well. Accuracy decreases noticeably when we move to three item shelves. There are three main ways the perception algorithms can fail. First, the clustering can fail. When this occurs, either one item is not found because the depth data is sparse or that two items are clustered together because of their proximity on the shelf. Second, the perception algorithms can misidentify items when the least squares matching does not correctly match ground truth with the captured scene data. Finally, ICP can fail to identify item pose.

Table 1: Perception System Analysis

	Two Items	Three Items
Correct Tests	22	13
Incorrect Tests	3	12
Accuracy	88%	52%
90% CI	72%-97%	34%-69%
Clustering Failure	1	3
ID Failure	0	6
Pose Failure	2	3
Average Run Time	15.66	19.93

This testing allowed for us to draw many useful conclusions about the perception subsystem. Several items are especially problematic, including the duck toy as well as specular items such as safety glasses. A list of suggested algorithm improvements was generated based on the results of this testing.

Failure Mode: Duck, Stanley tools, glasses, outlet plugs
Improvement: Process of elimination, ID these last. Skip shelves that contain 2 or more of these items
Failure Mode: Clustering Failure
Improvement: Use color or surface normal information
Failure Mode: Pose Estimation Failure
Improvement: Rewrite PCL ICP algorithm to minimize custom cost function, not just minimize the number of inliers
Failure Mode: Minimal depth data
Improvement: Take depth images from multiple perspectives (only possible once we transition to robot testing)
Improvement: Speed up algorithm by down sampling data, plus possibly break ICP loop after very good score achieved

6.3.3. Suction Modeling, Analysis, and Testing

In order to control the vacuum system, a custom PCB was designed, shown in Figure 17. The board holds an Arduino Nano. This Arduino reads up to four analog sensors (including two pressure sensors), display the status of the circuit via three status LED's, and triggers two relays which power two impellers used for a suction gripper.

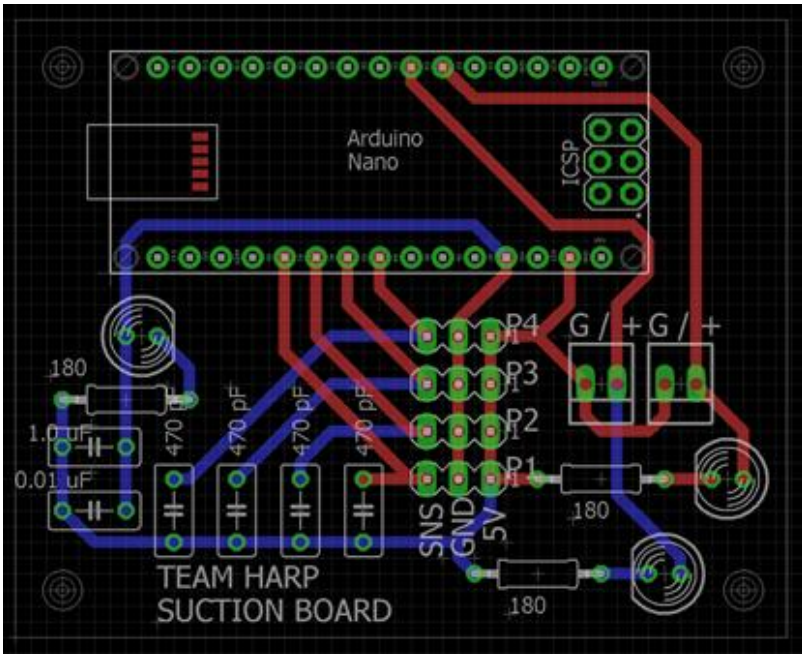

Figure 17: Custom PCB for Suction System

After completing the electronics box, the suction data was analyzed and a custom ROS node was written to monitor the state of the system. Preliminary results of the suction filter are shown in Figure 18. The raw pressure data (blue) is very noisy. Even after applying a rolling average filter, the signal to noise ratio is fairly small. In the test below, four items are picked up by the gripper. However, the filter only accurately detects three of four pressure drops in the example below.

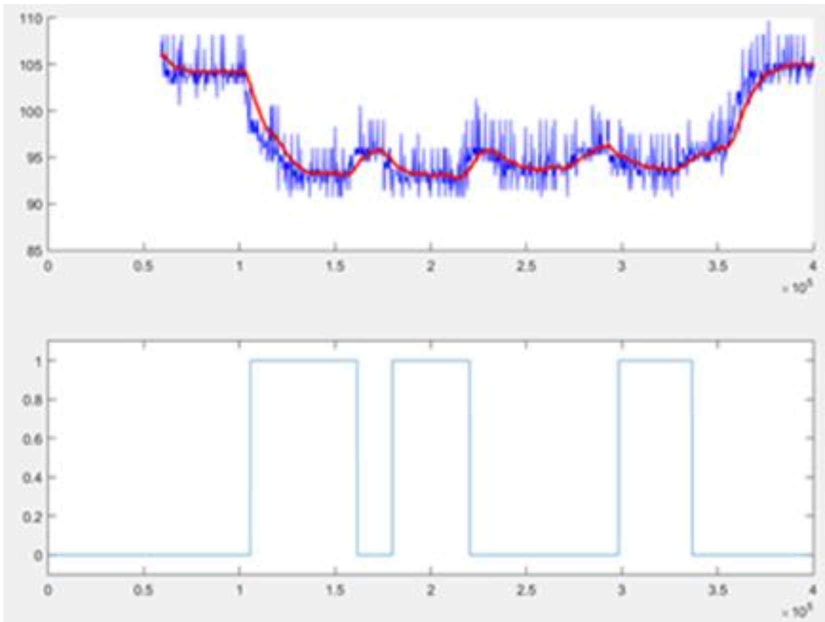

Figure 18: Raw and filtered suction values (top) and Grasp status prediction (bottom)

We are currently looking into finding a more sensitive pressure sensor with a better signal to noise ratio.

6.4. Fall Validation Experiment Summary

During the fall validation experiment, all targeted requirements were met. In addition, during the fall validation experiment encore, we transitioned from running in simulation to running the system on PR2. Table 2 highlights the outcome of the Fall Validation Experiments.

Table 2: Fall Validation Experiment Summary

Experiment	Expected Outcome	Requirements Validated	Outcome
Perception	The perception system predicts item pose on the shelf, shows results on a graphical user interface for analysis	FR2, PR2, NF4	FVE - Showed above 50% accuracy on 12 test cases FVE Encore - Showed 88% accuracy on two item cases and 52% on three item cases Note: NF4 verified through inspection, algorithm invariant to lighting.
Hardware in the Loop Test	The PR2 simulation runs, moving to each of the 12 order bins. At each bin, the simulation pauses and waits for an item to be grasped by the user	PR4, FR4, PR4, FR5, PR5, PR6	FVE - Showed successful results in simulation, pressure sensor detected item pickup on all attempts. Demonstrated pickup of .5kg item
PR2 State Machine Test	The PR2 runs, moving to each of the 12 order bins. At each bin, the robot pauses and waits for an item to be placed onto the suction cup	PR4, FR4, PR4, FR5, FR7, FR8	FVE Encore - Showed gripper holding items in dynamics cases. Increased confidence in transition from simulation to robot.

6.5. Semester Evaluation and Conclusion

The major strengths of our project include

- Developed PR2 simulation environment
- Flexible and reliable gripping solution
- Baseline perception performance

- Each subtask performs well in isolation

This semester, the upfront learning curve to run on the PR2 was fairly large. The PR2 in the Search Based Planning Lab still runs ROS Groovy. This led to compatibility issues with MoveIt! as it is no longer updated for ROS Groovy The documentation and example code were written for the latest version of MoveIt! which runs on ROS Hydro and above. We had issues with implementing collision avoidance in MoveIt!. Recently, we found sample code that implements collision avoidance in the outdated version and will be working on implementing it for our project during winter break.

The decision to use a suction based gripper simplified the grasping problem greatly. The gripper can pick up items of all shapes and sizes. In addition, there is room for error in the gripping approach trajectory. Finally, our design has shown the capability to pick up all but one of the items from the 2015 Amazon Picking Challenge list. By simplifying the grasping problem, the team can focus on improving the other subtasks.

The learning curve for Point Cloud Library and 3D item recognition was quite steep. At first, a lot of time was spent just reading research papers and testing out many different algorithms. It took several weeks to just get a pipeline up and running. Now that the vision algorithm works well, it will be possible to tweak and fine tune the algorithm next semester to increase performance.

The three major strength listed above highlight that the subtasks identified as critical at the beginning of the semester perform well. Independently, they far exceed the baseline performance metrics. We expect that each subsystem's performance will decrease after integration due to unforeseen issues related to the interplay between each subtask.

The major weaknesses of our project include

- Subtasks have yet to be integrated
- Tasks are likely to get more complicated with the announcement of the 2016 rules

By design, our team focused our efforts on developing each of the major subsystems in the 2015 Fall semester. As a result, little development was devoted to system level integration. The challenges and risks we face in the 2016 Spring semester build upon each subsystem and require precise subsystem interaction which will be harder to mitigate. To adequately address this weakness, we are leaving ample time for system integration activities and ensuring the designs are thoroughly analyzed before executing.

When running the experiment on the actual PR2 platform, we noticed that the base had jittery motion and was drifting with time. The jittery movement was a bug due to network latency affecting how the transform tree is published. The drifting will be solved when we localize the base with the respect to the shelf.

Our decision to build a system capable of competing in the Amazon Picking Challenge is a great opportunity but also exposes our project to changing requirements. A large known weakness of our project is that our requirements are based on outdated 2015 APC rules. While we have attempted to anticipate the changes in 2016 competition requirements, we will have to act fast once the final rules are released to revise our requirements to align with the competition rules

7. Project Management

7.1. Work Breakdown Structure

The Work Breakdown Structure is organized by subsystem. Figure 19 shows a graphical depiction of the Work Breakdown Structure. Tasks in green show tasks that have been completed, tasks in yellow are ongoing, tasks in red indicate progress is running behind schedule, and tasks in white have yet to be started.

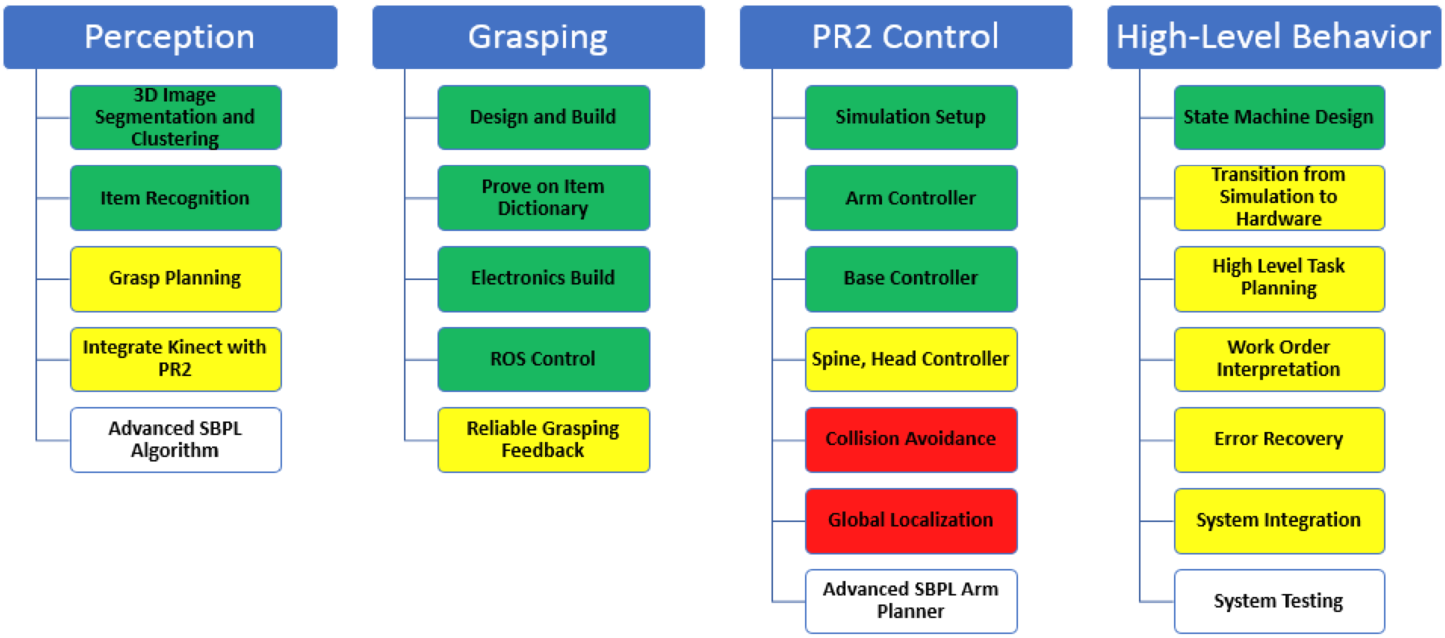

Figure 19: Work Breakdown Structure

7.1.1. Perception Work Breakdown

The isolated tasks of the perception system were completed for the Fall Validation Experiments in December 2015. The image segmentation and image recognition tasks have been validated in isolation. Work has been started on both simple and advanced grasp planning to find approach vectors normal to the items so the suction end effector can pick up the item. The work to integrate the Kinect with the PR2 has been started. To integrate the perception system, the Kinect mounting locations.

7.1.2. Grasping Work Breakdown

The grasping subsystem is almost fully completed. The vacuum system, electronics, and end effector have been designed and built for the Fall Validation Experiment. The only task left to improve upon is the sensing feedback from the suction system. The system looks for a pressure drop to provide feedback if the target item has been picked up by the end effector. This feedback is used to advance the state controller or adjust the location of the end effector to pick up the target item. Procuring and validating a higher resolution pressure sensor should finish this task.

7.1.3. PR2 Control Work Breakdown

To develop on the PR2, simulation and computation compatibility issues have been overcome. Additionally, the basic arm, base, and spine controllers have been wrapped into interfaces used by the state controller. The two main tasks left for 2016 are the implementation of collision avoidance in the ROS MoveIt! Planner and global localization. Currently, the control of the PR2 has neglected collision objects in the environment to allow more focus on system and state controller development. Going forward, collision avoidance will become increasingly important as we begin to work with the shelf and actual items. The second main task is localization. The PR2 is equipped with several laser sensors in addition to the Kinectv2, all of which each can be used to find the shelf and localize the PR2 with respect to the picking shelf. Finalizing an approach and implementing it will be the focus in early 2016.

7.1.4. High-Level Behavior Work Breakdown

To date, high level behavior has been focused on developing the state controller and building error recovery functionality specific to the Fall Validation Experiment. Going forward, interpreting the input item list to generate the high level task plan will be one of the first task necessary to begin running full system tests. Beyond that, the results of the perception and grasping efficacy will be used to make high level plans to maximize the system performance for the Amazon Picking Challenge. Finally, significant work will go into improving system performance once each of the subsystems are integrated and validated.

7.2. Milestone Schedule, 2016

The milestone schedule served as an invaluable tool during the Fall 2015 semester to keep our team on track and focused. Great care was taken to define and agree upon the milestone schedule for the 2016 Spring semester shown in Figure 20.

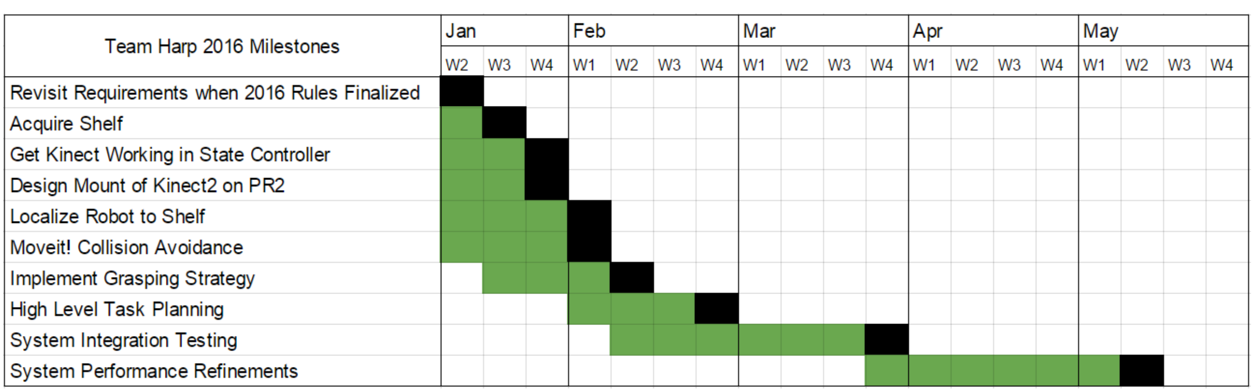

Figure 20: Fall Milestone Schedule

7.3. Test Plan

A progress review schedule (table 3) identifies technical deliverables leading up to the SVE. The first tasks will be to revisit the current requirements once the 2016 Amazon Picking Challenge rules are released to ensure our system capabilities are aligned to the demands of the competition. The Kinect integration work to get the perception algorithm integration into the state controller and the mounting locations planned will be complete by the end of January. Getting the perception system integrated onto the PR2 will enable focus on the localization strategy which will be completed shortly thereafter. Finally, the last critical task will be to implement collision avoidance and implement the grasp planning. Afterwards, the focus will be on debugging any problems that arise with system integration and improving system performance. The following Table 3 shows the biweekly demonstrations we plan to make for the spring 2016 semester.

Table 3: Progress Review Schedule

Progress Review 6 – 1/21/2016	Demonstrate collision avoidance in simulation and updated system requirements
Progress Review 7 - 2/4/2016	Demonstrate global localization and collision avoidance on PR2
Progress Review 8 - 2/18/2016	Demonstrate perception system working on the PR2
Progress Review 9 - 3/3/2016	Demonstrate ability to pick 1 items off a table
Progress Review 10 - 3/17/2016	Demonstrate ability to pick 1 items off a shelf (defined pose)
Progress Review 11 - 4/7/2016	Demonstrate ability to pick 1 items off a shelf (random pose)
Progress Review 12 - 4/21/2016	**SVE**: Demonstrate ability to pick 3 items in 20 minutes
Progress Review 13 - 5/10/2016	Demonstrate ability to pick 5 items in 15 minutes

The spring validation experiment will be to fulfill the input order to the best of the system's capabilities. The target items will be presented to the robot and the robot will autonomously determine the high level tasks planning required to retrieve as many items from the input list as possible. The robot will attempt to perceive the target item among the background items off each bin, move the suction end effector to pick the item, and place the item in the order bin without damaging or dropping the item. The system will demonstrate its ability to fulfill all functional and nonfunctional requirements in this experiment.

7.4. Budget

The budget including spent to date is shown in Figure 21.

Item Description	Cost
Suction Prototype	$ (109.42)
Shop Vac (Qty 2)	$ (245.58)
Electronics	$ (224.52)
Kinect 2	$ (140.00)
Item Dictionary	$ (50.00)
Total Cost	$ (769.52)
Remaining Funds	$ 3230.48

Figure 21: Project Budget

Our spending has been comprised mainly of suction system components. Future expenses include raw materials for the boost platform for elevating the PR2 and other miscellaneous expenses related to building the competition shelf and procuring the items out of the dictionary.

7.5. Risk Management

The top risks are with the perception system performance and gripper design. Figure 22 shows the most significant risks faced going forward along with a matrix showing the consequence and likelihood of each risk.

	Risk	Description	Type	Cause	Consequence	Mitigation
A	Perception system viewing angle	View from PR2 may be insufficient to generate grasp vector	Technical	Mounting Limitations on PR2 Algorithm sensitive to perspective angle	System cannot pick certain items	Detailed analysis of PR2 Mounting design Leave time for algorithm refinements
B	Depth sensor cannot detect specular items	Irregular items may not match well to Ground Truth Models	Technical	Ground Truth model Sparse Item in odd configuration	System cannot pick certain items	View shelf from multiple angles Add color features to algorithm
C	Gripper design may be insufficient for new 2016 rules	A stowage task may make current design ineffective	Technical	Requirements change	Suction end effector redesign	Have several designs in mind anticipating rule change
D	Perception algorithm does not scale on crowded shelf bins	Perception algorithm runs in $O(n^3)$ for n items	Technical	ICP algorithm requires large amount of computation	May not be competitive	Parallelize algorithm Use SBPL algorithm

Figure 22: Top Risks for 2016

The perception viewing angle is the largest risk going forward. The problems we face stem from the limited mounting locations available to use on the PR2. The most obvious mounting location is on the PR2 head where mounting surfaces are available. The downside to a head mounted Kinect is the limited view into the lowest shelves. There is the possibility of a sternum mounted location seen in Figure 23 but we will need to determine if this location is feasible with the Kinectv2 and that it will provide the appropriate viewing angle.

Figure 23: Possible Kinect Mounting Configuration

The next risk is the difficulty of the current implementation of the perception algorithm to classify and estimate pose of specular objects. This is a known shortcoming in the perception algorithm. Once the final 2016 item dictionary is released, the approach and fix for this shortcoming will be decided. One possible approach is the inclusion of pixel color features in the classification which currently only looks at depth data.

The third risk stems from alignment between the gripper design and the Amazon Picking Challenge rules. The current design works great but the geometry is specific to the picking task. If the final rules include a stowage task or other task not included in our current design considerations we will need to change the design to redress this.

Finally, the fourth risk is the computation time of the perception algorithm. The current algorithm takes cubic time with number of items on the shelf. For shelves containing three or fewer items, this should not be a big issue but for five or more items this computation time could be too long. For now, mitigation plans are put on hold until the latest competition rules are announced. We may be able to improve the run time by implementing the iterative closest point feature matching in parallel which should run in linear time with number of distinct items.

8. Conclusions

8.1. Lessons Learned

Know when to take personal growth roles vs. productivity roles
As students, we are eager to learn and grow our skill through exposure to new concepts and cross-curriculum work. As deadlines approach, we must meet deadlines so there must be a delicate balance between personal growth and productivity within any academic project.

Be wary of working with new or unsupported hardware or software
A lot of the problems we encountered this semester were with software and hardware compatibility. The Kinectv2 is a good example of this. The specs of the Kinectv2 are good enough to motivate our use of the system but the recent introduction means there is little support and we have to sink countless hours debugging low level problems.

Know when to cut your losses
Plans are great to keep efforts in line with the task that need to be done but sometimes our best efforts result in not progress. Knowing when to stop, regroup, and approach a problem in a different way has been important for us to achieve our successes this far.

The best resource is your MRSD peers, TAs, and professors
Our fellow MRSD student come from diverse backgrounds and offer a vast resource of experience and hard-earned lessons. Whether it is asking to borrow a spare part or sensor or getting feedback on a design, the other teams have been great at lending a helping hand. Additionally, the TA's, professors, sponsors, and lecture material offer a wealth of information that have helped us greatly.

8.2. Key Spring Semester Activities

All of the planned milestones have been laid out in the work breakdown structure and in the milestone schedule sections. To reiterate, the key activities for 2016 will be a revision of the system requirement once the 2016 competition rules are announced, development and refinement of the global localization strategy, implementation of the grasping strategy, integration of the perception algorithm and hardware on the PR2, and finally the system level performance testing. These four critical activities will be crucial to our success both in the MRSD program and in the Amazon Picking Competition in 2016.

9. References

The following is a list of technical references consulted for the execution of this project.

Reference	**Link**
Point Cloud Library	http://pointclouds.org/
OpenCV	http://opencv.org/
SMACH	https://github.com/ros/executive_smach
ROS	http://www.ros.org/
Willow Garage	https://www.willowgarage.com/
MoveIt!	http://moveit.ros.org/
OMPL	http://ompl.kavrakilab.org/
Gazebo	http://gazebosim.org/
Amazon Picking Challege	http://amazonpickingchallenge.org/