CRITICAL DESIGN REVIEW

Amazon Picking Challenge

Team Members (Team PLAID):
Matt LAUER
Michael BECK
Akshay BHAGAT
Che-Yen LU
Jin ZHU
Mentors:
John DOLAN
Dimitrios APOSTOLOPOULOS
Advisor:
Max LIKHACHEV

December 15, 2016

Abstract

This report summarizes Team PLAID's progress on the development of an autonomous picking and stowage system for warehouse-like environments, for the Fall 2016 semester. This system is being designed in accordance for the competition rules for the 2017 Amazon Picking Challenge.

The report begins with the project description and use case followed by the system-level requirements. The system functional and cyber physical architectures describe how the system meets the aforementioned requirements.

Next, the current system status (as of FVE Encore) is reported along with the project management tools that were used to plan and track the team's progress. The report then proceeds to analyze the strong and weak points of the current system implementation and to identify areas for refinement.

The last part of the report comprises lessons learned, conclusions, references and appendices.

Table of Contents

The CDRR consists of the following components:

2. Project Description

Team PLAID's objective is to develop an autonomous robot that can pick and stow various objects in a warehouse-like environment. This project is a continuation of team HARP's project, which was an entry to the 2016 Amazon Picking Challenge.

Although high-speed picking robots have appeared in the industry for years, these robots can't be widely used. Robots still have great difficulty in picking up arbitrary objects when they operate in open, unknown environments or under uncontrolled conditions, such as lighting variation and occlusions.

Moreover, picking and stowing items in pods is the last link remaining to be automated in Amazon's warehouses. After 2009, Kiva robots changed the face of Amazon warehousing, allowing shelves to be moved to workers without human intervention. However, manpower is still required in order to pick and place items in the designated boxes. With our robot, the operational cost of a warehouse can be reduced. No air-conditioned facilities or restrooms are required anymore. The costs associated with human resources, management, payroll,etc. are also reduced.

3. Use case

Joe is the owner of a large ecommerce corporation. He is having trouble competing with other companies and is looking for a way to reduce the recurring cost of managing his warehouses.

Joe decides to purchase a Plaid Picking Platform in order to reduce the cost of labor at his warehouse. This 30system includes a robotic arm, a custom shelf, and an actuated shelf frame. Joe assigns a technician to setup the robot.The technician clears the robot's workspace, and Joe instructs his employees not to enter the robot's workspace while it is working. In the course of about one half hour, Joe's technician has connected the robot to the warehouse server and power. The robot is ready for picking and stowing.

Shelves are placed inside of the actuated frame and the warehouse server orders the robot to pick or stow various items. First the actuated frame unfolds the shelf. The robot then accept the order and picks and stows the appropriate item without any human intervention. For picking, the robot will search the specified bin for the desired item and move items that may be occluding the desired item. Once the item is located, the robot will grasp the item and place it in a tote that is mounted on the robot. In the case of stowing, the robot will search for and pick desired items from a tote and place those items on the shelf. After either picking or stowing the robot will update the warehouse server with the new location of any item that has been moved.

When Joe decides it is time for maintenance, he can instruct the robot to return to the home position and execute a complete stop.f

The efficiency of Joe's warehouse has been increased, which will help him save money and compete with other companies.

Figure 1 is a graphical representation which shows how the robot arm would pick items from the shelf after the actuated frame has been unfolded.

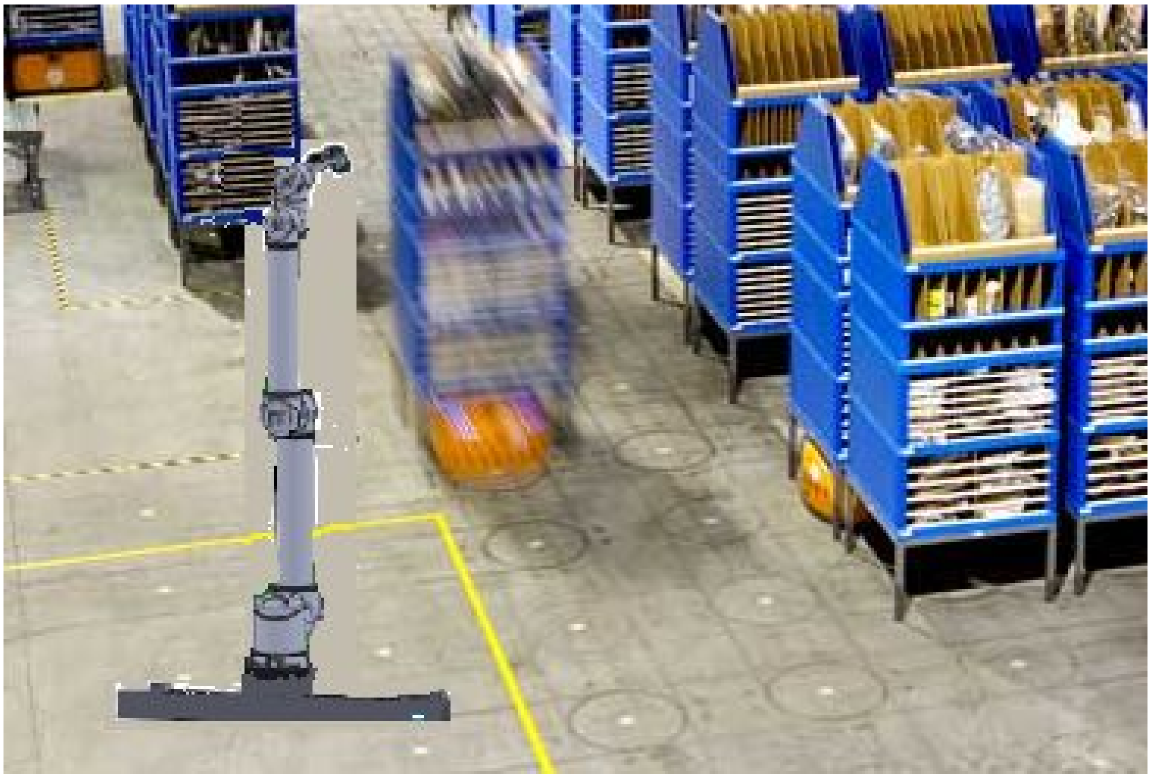

Figure 1. Use case CAD model representation

4. System-level Requirements

4.1 Functional Requirements

The functional and performance requirements for this project have been determined in accordance with the goal of creating a competitive pick and stow robotic system for the 2017 Amazon Picking Challenge. The current complete set of functional, performance, and non-functional updates can be seen below. These requirements reflect the new rules [1] laid out by Amazon for the 2017 Challenge which we're released in late November. Where requirements have changed as a result of these new rules they are explained in the table row labeled "Changes".

M.F.1	Hold all challenge items on a fabricated shelf
Changes	The new rules specify that competing teams are to fabricate their own shelves for item stowage. This is a change from using the kiva pod which was previously provided by Amazon.

M.F.2	Accept user orders in the form of a JSON file

M.F.3	Identify and localize challenge items

M.F.4	Pick items from the fabricated shelf

M.F.5	Place picked items into a user-specified tote
Changes	The new ruleset specifies that the items must be placed into one of three specified totes as specified within the user order.

M.F.6	Generate an item location report after the user order has been completed

4.2 Performance Requirements

M.P.1	Interpret pick and stow object orders with 100% accuracy

M.P.2	Achieve at least 12 successful picks within a 15 minute time frame
D.P.1	Achieve at least 10 successful picks within a 15 minute time frame, with half of the picked items being "generic"
Changes	The mandatory metric reflects what would have been considered a "perfect run" (grabbing all items within the time frame) in the 2016 Challenge. Previously this requirement specified that only 8 picks be achieved within the same time frame. Because a large part of the challenge for previous competitions was working within the kiva pod, an administrative decision has been made that the difficulty should scale accordingly. Generic items are currently outside of the scope of work for this course, however their inclusion inside of this desirable requirement matches the 2017 rules and may be accomplishable with outside aid.

M.P.3	Drop no more than 1 item for every 6 items successfully picked inside of a test run
D.P.2	Drop 0 items inside of a test run
Changes	Similar to M.P.2 a decision was made to scale the difficulty of this metric due to the new ruleset allowing for item storage inside of a custom shelf instead of the kiva pod. The old metric specified that 1 item for every 4 picked could be dropped.

M.P.4	Store and pick items from no less than 2 and no more than 10 bins, which occupy a 125 cm x 5000 cm^2 (height x floorspace) or smaller volume
Changes	These measurements are inherent to the new competition environment and rules as outlined by Amazon for the 2017 Challenge.

M.P.5	Capable of lifting items weighing up to 2 kg
Changes	This weight reflects the heaviest possible weight for an item as specified by Amazon for the 2017 challenge.

M.P.6	Generate an output file which reports locations for items still within the shelf bins with 100% accuracy

M.P.7	Autonomously identify 95% of non-occluded items within a shelf bin

M.P.8	System grasper capable of gripping and maintaining its hold on 90% of competition items
D.P.8	System grasper capable of gripping and maintaining its hold on 100% of competition items

M.P.9	Drop picked items into the order bin from a height of no more than .3m from the bottom surface of the order bin

M.P.10	Minimize path planning computation time to less than 2 seconds per path.
Changes	This is a new metric which has been defined by the project's sponsor (the Search-Based Planning Lab).

4.3 Non-Functional Requirements

M.N.1	Cost no more than $5000 excluding any contributions from the Searched-Based Planning Lab
Changes	The new system design is beyond the project scope budget of $5000. The Search-Based Planning Lab is considering providing additional funding in order to meet the new design costs.

M.N.2	Be reliably assembled/disassembled and transportable through conventional mailing services to the competition location

M.N.3	Adapt to a variety of lighting conditions including different lux ranges and types of light

M.N.4	The entire system must remain within a 2.5 m x 2.5 m workspace with the exception of the end effector
Changes	This requirement is laid out by Amazon in the new competition rules.

M.N.5	The system must have a reliable emergency stop

M.N.6	The system must perform in a modular fashion
Changes	This is a new metric which has been defined by the project's sponsor (the Search-Based Planning Lab). This metric allows for easier debugging for the various subsystems.

5. Functional Architecture

In order to successfully meet the performance and non-functional requirements under the 2017 competition rules, the functional architecture comprises input handling, perception, path planning, grasping, and shelf subsystems as shown in Figure 2. The new rules require teams to design their own storage system. The shelf subsystem represents this new component in the functional architecture.

The black box model of the system takes in a work order, picks or stows items based on the work order and outputs an item report.

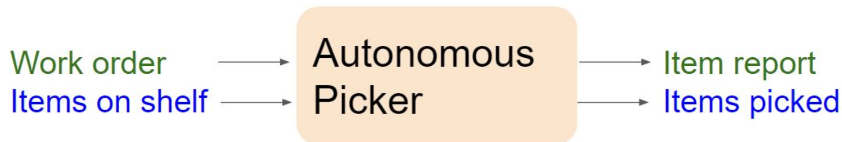

Figure 2. System Black Box Model

The input/output subsystem is responsible for communication between the system and the human user. Input information is a work order, and the output information is a list of items picked by the robot.

The shelf subsystem unfolds at the beginning of a pick task and folds back upon conclusion of the same. This function makes perception and grasping easier.

The Perception subsystem represents the function of localize the shelf and identify items in each bin. Object recognition and pose estimation are the two main functions for perception subsystem. Accuracy in this section is vital for correct pickings.

The Path Planning subsystems accepts information from perception and plans the path to pick and place the item. Planning takes the item location information from perception and gives the arm a trajectory to follow. The robustness and speed of planning is important for how reliable the system is and how fast the system can run.

The Grasping subsystem picks an item from the designated bin at a designated angle, based on the information from perception, such as the item pose and surface normals. Grasping tools or strategies could be different for different items.

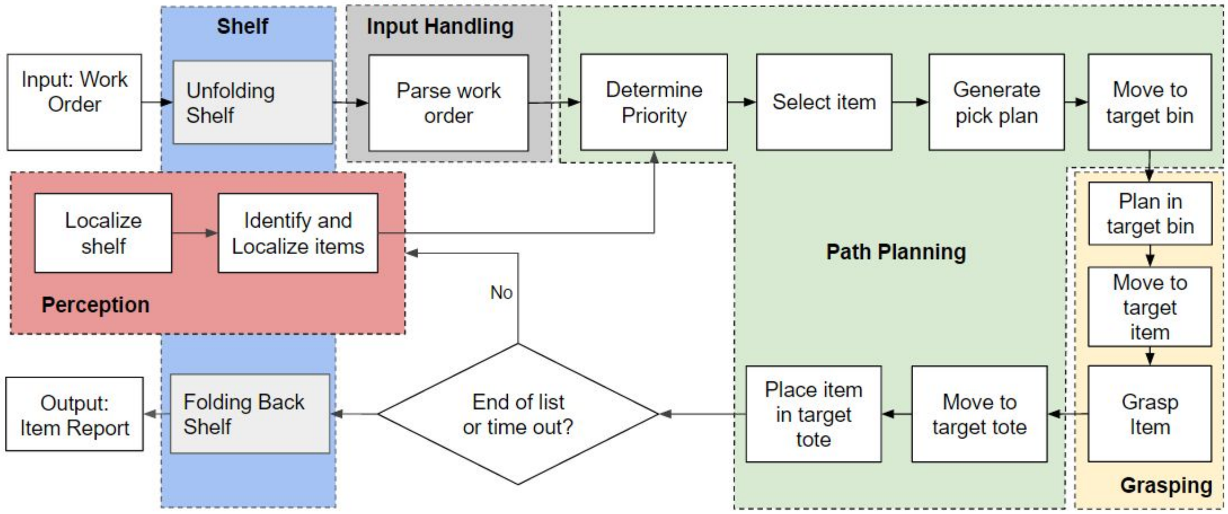

Figure 3. Functional Architecture

The functional architecture marks the flow of material and information within the system. An Item Location File will be given at the beginning of a task. This JSON file defines which items are in which bin. The Order File defines the items need to be picked from an order. The robot parses the work order to determine which items need to be picked and which bin each item is in. After the robot finishes, it will output an Item Report, which list the items that were picked.

The shelf will be unfolded at the workstation at the beginning of a task for easier vision and grasping. The cameras on workstation will look at each bin top down, identify items, and determine item pose. At the same time, the camera on end effector will localize the shelf and then identify and localize the items in the lowest bin.

Based on the confidence score from the perception subsystem and the “pickability” of the item, an item priority list is generated. The item on the top of a priority list will be selected and a path to the target bin will be generated. The end effector follows the path and moves to the item. Then the end effector grasps the item, moves to tote, and places the item safely in the tote.

After picking each item, the robot will check if it has reached the end of the item list or the end of competition time. If neither is true, the robot will repeat the process. Otherwise, the robot will fold the shelf back to original form and report to user which items it has picked.

6. Cyber-physical architecture

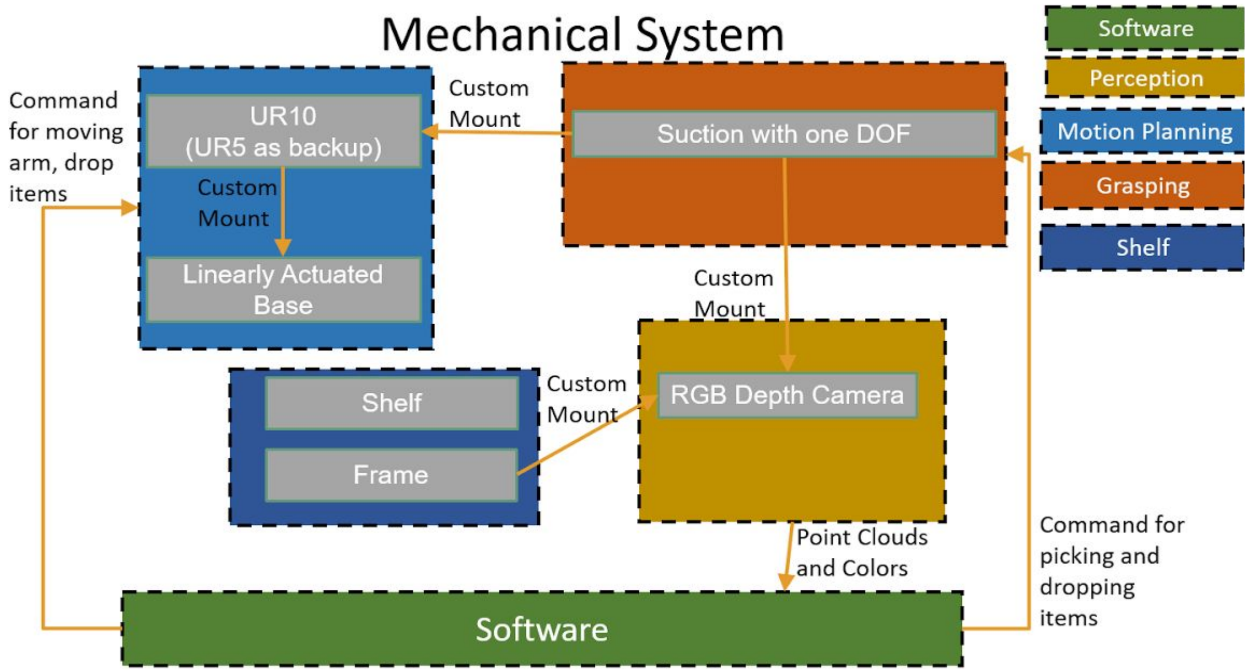

Figure 4. Mechanical System (Low budget).

This report includes two separate Cyber-Physical Architectures based on the level of funding we are able to acquire in the next few months. The low budget and high budget architectures are shown in Figure 4 and Figure 5 respectively. In the case of the mechanical portion of the Cyber-Physical Architecture, the primary difference is that the custom shelf frame will not be actuated if no additional funding is secured. If enough funding is available, we intend to construct an actuated frame that can fold and unfold the shelf with rotary actuators that would receive commands from the software system of the robot. All other parts of the architecture remain the same and you may refer to either Figure 4 or Figure 5 to understand the remaining description of the mechanical system.

Excluding the shelf, hardware for this architecture has been broken into three main subsystems that correspond to the systems seen in software. The software system is shown in green and will be explored in more detail later.

The section in light blue is the hardware that is relevant to motion planning, which includes the UR10 robotic arm as well as the linearly actuated base on which the arm will be mounted. The software system will be able to issue commands to the arm and base.

In red, the grasping subsystem includes a custom made end effector with one degree of freedom on the tip of the suction head. The software system will be able to issue commands to engage suction as well as receive pressure feedback from the suction head.

The yellow section is the perception hardware, which will be four RGB-D cameras. Three of the cameras will be mounted on the shelf frame, and one will be mounted on the end effector. The cameras will be sending a stream of point clouds and color data to the software system.

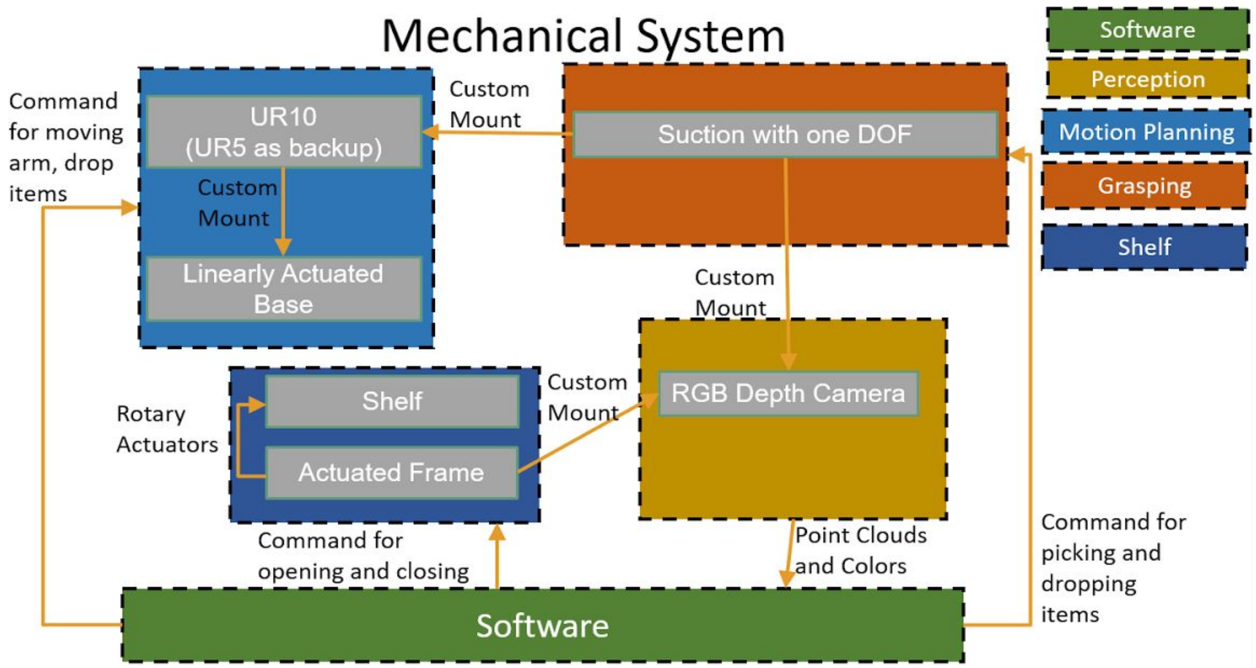

Figure 5. Mechanical System (High Budget).

As for software portion of cyber-physical architecture, the green box in Figure 5 has been broken into five software subsystems. Figure 6 reflects the change in mechanical portion and shows the software portion in detail. Each dashed box represents a subsystem and corresponding rosnode. Also, all subsystems are capable of communicating with each other through rosservice, which makes communication interfaces unified and easy to maintain.

The state machine and central control is the section in green. High-level flow control and exception handling is implemented in this block. Well-designed system control guarantees robustness and performance.

The motion planning service is represented in the light blue boxes. This subsystem can be used as a service by other subsystems. The Moveit ROS package and SBPL planner are also used to facilitate the development of planning.

The perception subsystem is represented in yellow. The Kinect_bridge is used to connect the kinect driver and the ROS environment. PCL 1.7 is used to manipulate point cloud data.

The grasping subsystem is represented in orange. This subsystem subscribes to the pressure sensor reading and sends commands to control suction power over rosserial. It is also responsible for deciding on grasping surfaces and grasping strategies.

The shelf subsystem is represented by dark blue sections. The system control subsystem can operate rotary motors to fold or unfold the shelf. The planning scene in simulation will also reflect the change in real world.

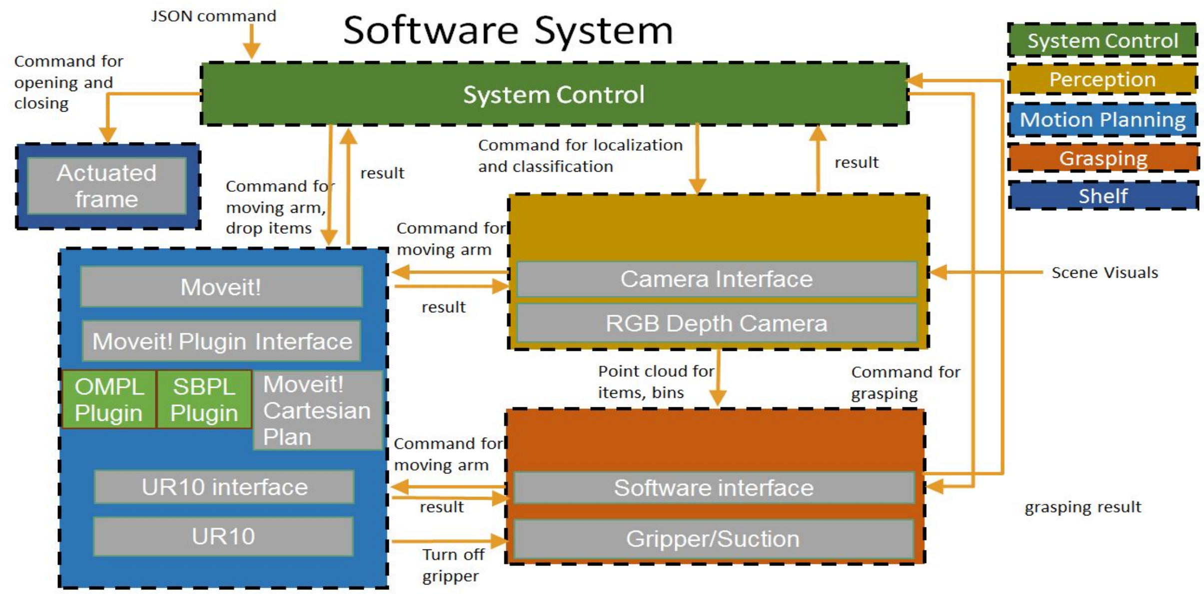

Figure 6. Software System(High Budget)

7. Current System Status

7.1 Targeted Requirements

Mandatory functional requirements 3, 4, 5, and 6 were all targeted for the Fall Validation Experiment (these correlate to identifying and localizing shelf items, picking the items, placing the items in a tote, and outputting a file with item information). The focus of honing on on these targets was to produce a minimally-viable product (MVP), which would resemble the entire system operation operating at a low level. More specifically the following operations were targeted: performing path planning to place the end effector in and out of bins, implementing grasping controls including use of the system vacuum feedback, identifying target items within bins and pass simple pose information to other subsystems, and demonstrating overall system integration through the state machine.

Working towards an MVP allowed the team to make sure that each subsystem was being designed such that it could interact with each other subsystem through the state machine. Producing an MVP also allows for working towards optimizing system features which are already implemented. Optimization is an important part of this project and needs to begin as early as possible in order to produce a competitive overall system for the competition.

Specific metrics for targeted Fall requirements included identifying 3 items and determining correlating simple pose information regarding their location within the kiva pod, and picking at least 2 out of the 3 items within a 5 minute window. The process of picking encompassed moving the end effector over an item, engaging suction in order to grip the item, moving the end effector over the tote, and placing the picked items inside of the tote. Determinations for where to move the end effector within the shelf were to be made through use of the vision system, which was to pass pose information to the grasping subsystem through the state machine. At the end of the picking run the system was also to output item identification information and bounding boxes to a desired folder.

7.2 Overall System Depiction

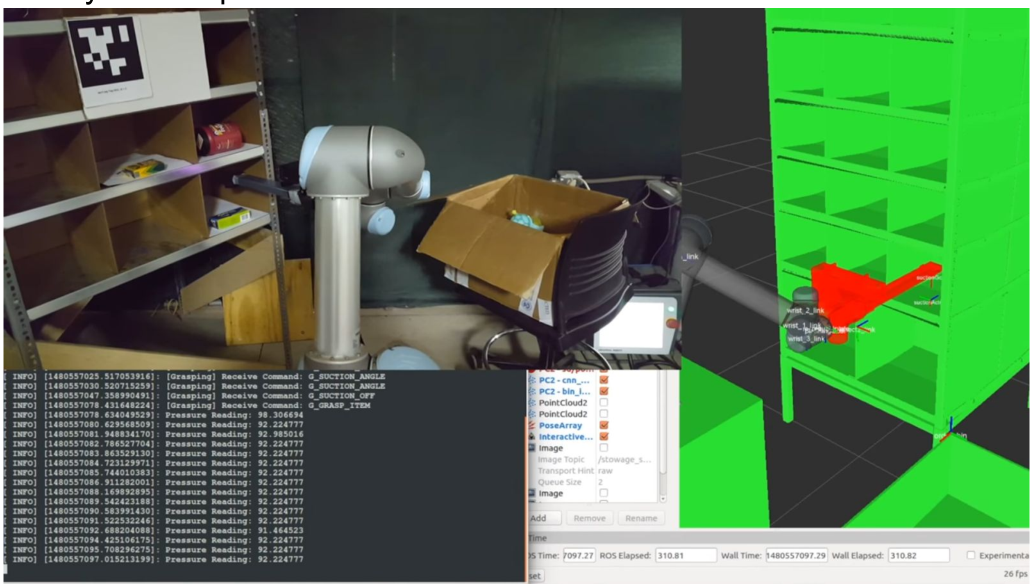

Figure 7. Full System Depiction

Figure 7 is a composite image which shows that multiple subsystems are working correctly in tandem to perform a successful pick. On the right of Figure 7, the collision model for the planning subsystem is shown. In the lower left corner, pressure feedback from the grasping subsystem is being printed to the terminal. To execute a pick, the grasping system uses data from the perception system to send movement requests to the motion planning system. Higher level decisions are made by the robot's state machine. While the subsystems are implemented in a simple way, their integration is complete. Each subsystem is explored in more detail below.

7.3 Current Subsystem Status

7.3.1 Subsystem Status-Software

The software subsystem in this project leverages resources from RI faculty and last year's team. Due to requirements for flexibility and maintainability, a new overall software architecture has been implemented.

The software architecture consists of four ROS nodes: system control, perception, motion planning, and grasping. Nodes have been divided based on their functionalities. The number of ROS nodes should be minimized without damaging system flexibility and the potential to run subsystems in parallel.

The system control node which issues commands to the other four nodes. Each node parses the commands and executes the desired task. Since those commands are blocking calls, the system control node will wait until a result is returned. For the motion planning node, the task is pretty simple. It moves the arm to the specific position requested by the client. The request could either specify the bin number or the arm pose. As for the perception node, it is responsible for shelf localization, item classification and item pose estimation. Finally, the grasping node take point clouds of bin and items as inputs to generate grasping poses. After generating poses, it will use the motion planning service to move the end effector to the grasping position and grab the target item. Figure 8 is the flow chart for the picking task. The dashed line boxes stand for ROS nodes, and solid line boxes represent services or functionalities nodes provide.

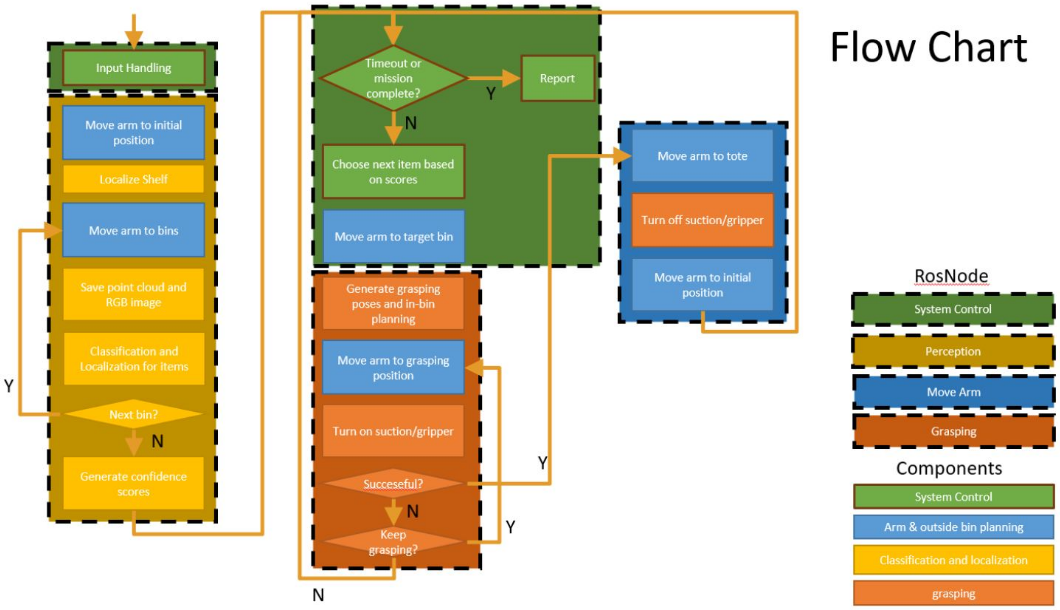

Figure 8. Software Flow Chart:Picking

7.3.2 FVE Evaluation-Software

The software subsystem was demonstrated to be fully functional in a picking scenario. The end effector pauses in front of each bin to gather point cloud and image data. The system control moves the end effector to the top of target item based on the height and centroid location supplied by the perception subsystem. Pressure sensors are used to detect a successful grasp. If the pressure sensors are unable to sense a successful grasp the arm keeps moving down until it establishes suction.

Due to the stop-motion criteria for the robotic am, results can be disastrous if the state machine doesn't detect the anticipated pressure sensor reading. Unexpected conditions should be taken into consideration to enhance the robustness of the state machine and allow it to fail "gracefully".

7.3.3 Subsystem Status-Motion Planning

The motion planning subsystem is currently on schedule and is meeting or on track to meet a few major goals.

The first goal is that planning should act as a generic service for other subsystems. Right now there are a few generic planning calls, such as move to tote, move to bin, and move to home, that can be requested by any subsystem and function without any major issues. As the project moves forward, there may be additional generic plan types to be implemented, so an easy framework has been made for adding more. All requests also receive a response that indicates a whether the plan was successful or not.

With the help of experience graphs, planning has also become more robust within bins. Experience graphs are a way of modifying a heuristic for a search based planner with the goal of reducing the number of expanded states during search, which in turn reduces planning time. Since the Amazon Robotics Challenge has a tightly constrained and mostly static environment, experience graphs are a useful tool. In the past, a relatively rigid set of poses needed to be reached in a specific order to enter bins, but experience graphs has made the system more robust. Even fairly difficult plans, such as planning from one bin to another can be found reliably in a few seconds without the need to specifically plan to certain waypoints outside the relevant bins. This will lead to more generic and simple state machine in the future, but has not yet been used in any demonstrations on the physical robot. Figure 9 shows a visualization of some of the cached trajectories that were used to verify the usefulness of the experience graph planner. New trajectories will need to be acquired once the new shelf is fully designed. This process should be simplified by using a short trajectory capturing script that was written for the original experience graph.

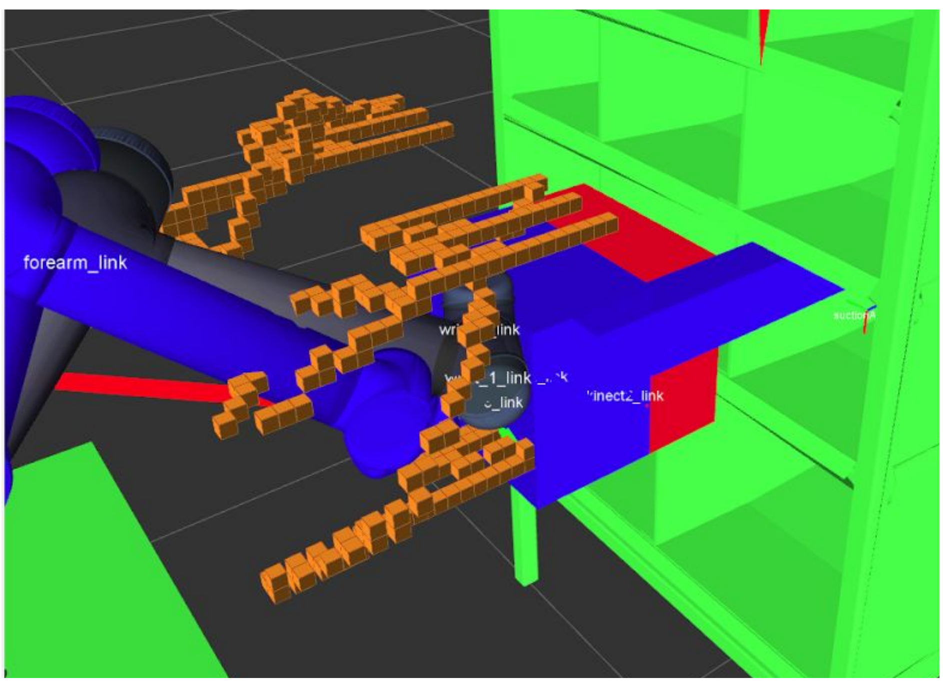

Figure 9. The cached trajectories of an experience graph.

The motion planning subsystem also experiences some jerky motion during grasping attempts. These issues are caused by a series of short motion requests to increment the gripper downwards, and the jerkiness becomes most apparent when the vision system has estimated an item pose in the wrong area. This poor estimation may cause grasping to increment down slowly towards the bottom of the bin, which takes a long time and exacerbates jerkiness. The reason this issue is brought up here in planning, is that the solution lives in planning. By implementing motion planning with the ROS actionlib package there may be a convenient way to make downward planning within a bin smooth. This has not been implemented yet.

7.3.4 Modeling and Testing-Motion Planning

So far modeling for motion planning is done entirely using ROS and RViz. This model includes all the important collision models such as the mount, the UR5 arm, and the shelf. Since the motion planning system uses these collision models to generate plans in the real world, it is important that these models reflect the real world as accurately as possible. On a few occasions we witnessed unexpected motion plans and had to update the models accordingly.

For testing the system, there are three types of tests that have been conducted. The first is a test that checks the configuration space of the arm. Using a script from last year's team, we are able to feed in a list of relevant end effector poses to the an inverse kinematics program that determines how many of the listed poses were reachable. This type of test helps us evaluate the effects of new hardware by giving us an accurate measure of the arm's configuration space. This type of test will be critical for finalizing the design of our new shelf.

Another type of test for planning is a planning time test. This type of test characterizes the amount of time needed to generate a plan. Some slight modifications to the planning server have been made in order to record in a text file how long it took to generate a plan. The values in the file can then be plotted and interpreted easily.

7.3.5 FVE Evaluation-Motion Planning

For FVE, the planning subsystem was fully functional, but had jittery motion while attempting to grasp items. During Encore, this jittering was reduced, but still appeared when the vision system gave an incorrect item pose estimation. The reduction in jitter shaved about one minute off of total test time, which demonstrates the system's ability to act quickly during competition.

Besides jitter, planning acted well during FVE and Encore. Plans were generated quickly and the success of plans was reported accurately. During FVE and Encore, each other subsystem was able to send commands to planning, which demonstrates the desired generic service structure.

7.3.6 Subsystem Status-Grasping Software

The software component of the grasping subsystem is fully integrated into the rest of the software. This means that grasping is able to request motion plans, receive commands from the state machine, receive pose estimation data from perception, and finally control the vacuum that is used to grasp items. Now that these basic integration steps have been taken, the grasping subsystem is ready to be refined for more complex grasping techniques.

7.3.7 Subsystem Status-Grasping Hardware

The gripper has been designed with the following goals in mind: having a narrow and lightweight but sturdy profile, possessing tight tolerances for the 1-DOF within a 0-90 degree range, and being composed of easily replaceable parts. To that end an aluminum C-channel has been chosen for the gripper frame, which allows for easy access to hosing which will be strapped to the inside of the channel and provides sturdy support while remaining lightweight. The gripper suction head and some of the linkages have been 3D-printed, allowing for a cheap and quick way to make small modifications as necessary. The remaining linkages are to be milled from aluminum or be composed of laser-cut Delrin. Shoulder bolts have been used for the pivots. The design was developed with the aid of the RI Machine Shop.

7.3.8 Modeling and Testing-Grasping Hardware

Modeling for the gripper linkages was accomplished through the use of SolidWorks as seen in Figure 10. Mates were created within the design assembly which allowed only for free movement of the 1-DOF pivot. This simulation allowed the linkages to be adjusted until the suction head moved in the desired manner within the model.

Testing was accomplished through the use of a DC power source. The linear actuator was powered in both the forward and reverse polarity to check for the pivot tolerances.

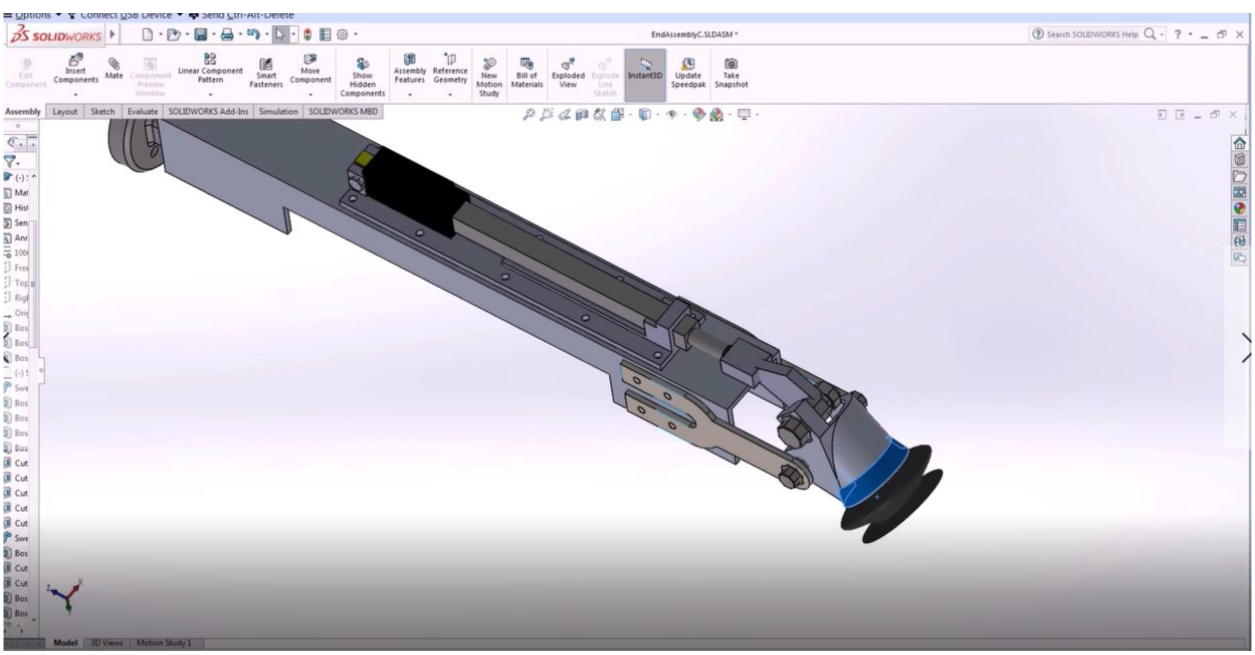

Figure 10. Gripper SolidWorks Assembly

7.3.9 FVE Evaluation-Gripper Fabrication

For the first Fall Validation Experiment gripper controls were implemented through rosserial. The first FVE demonstrated that the gripper mockup had poor tolerances and allowed for too much play within the pivot for a given angle.

The gripper linkages and suction head were redesigned for the FVE encore to improve tolerance issues, as seen in Figure 11. With the new changes the system demonstrated that it could move to any user-specified with a few degrees through ROS.

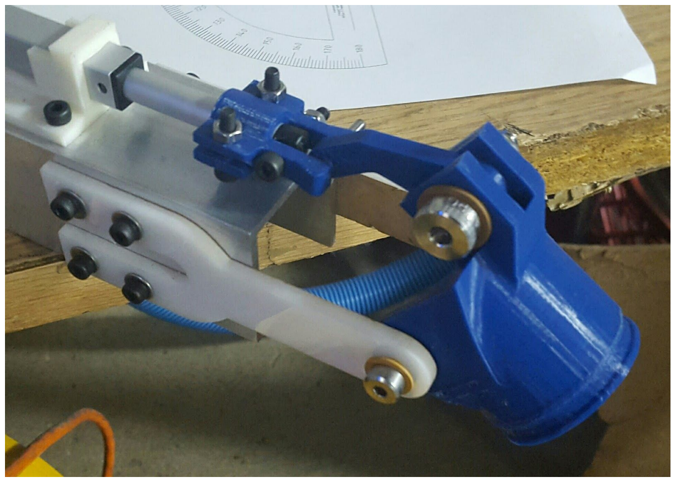

Figure 11. Fabricated gripper prototype for FVE Encore

7.3.10 Subsystem Status-Perception

The current system implementation of the Perception subsystem is shown in Fig 12, below.

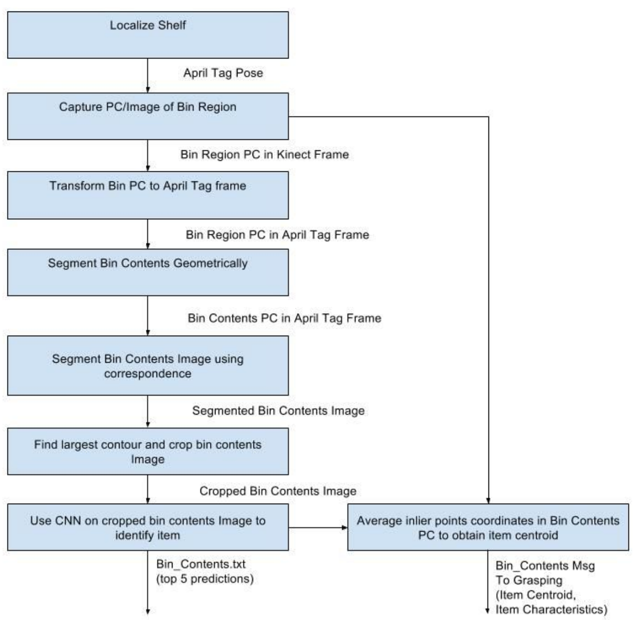

Figure 12. Perception subsystem flowchart

The perception system is behind schedule in several areas. The major areas are:

1. Item Segmentation from the Point Cloud must be extended to multiple touching items.
2. The shelf and kinect setup is position invariant. Full pose invariance is preferred, so as to deal with perturbation of the shelf.
3. The item identification needs to be replaced by a more robust model that will account for the surprise items as well.
4. Utilities have to be developed to calibrate and measure the transforms of all kinects as required in the new shelf design.

The areas of refinement and further work for the spring semester are shown in table 1, below under desired system status column.

Table 1. Current and Desired System Status

Vision System Functions	Current System Status	Desired System Status
Get Bin Point Cloud/Image	Position Invariant April Tag Based Segmentation	Pose Invariant April Tag Based Segmentation
	Dead-on View of Adjacent Bin	View over Gripper of Bin Straight Ahead
		Multiple Views/Dense Point Cloud
Get Item Image	Single Item/Contour Based	Multiple Touching Items/Bounding Box
Get Item Point Cloud	Geometric Based Single Item	Multiple Item Correspondence Based
Identify Item		
CNN-RGB Identification	Using last years trained model	As per performance Requirements
CNN-Depth/Shape Invariant Identification		
PERCH		
Item Pose Determination		
PERCH		
CNN-RGB based Pose Estimation		
Implemented	Required	Desired/Time Permitting
Doesn't Fullfill Performance Req.		

7.3.11 FVE Evaluation-Perception

For FVE, the perception subsystem was integrated with the state machine, however was not conveying any details of item pose to the planning and grasping subsystems. For the FVE Encore the perception system was providing a basic pose estimate to the grasping subsystem that consisted of item centroid and item characteristics.

The item identification for the FVE and FVE Encore was done using a makeshift method, that entailed resizing the cropped item image and feeding it to team HARP's superpixel classifier to generate the top 5 item predictions. This method was quite unreliable leading to up to 66 percent wrong classifications as it was purely based on dominant item color and thus required the item color to be largely homogenous.

7.3.12 Subsystem Status-Shelf Design and Fabrication

A new component for this project is the design and fabrication of a shelf in order to hold the challenge items. After communications with the team sponsor (SBPL) a conceptual idealization for the shelf manifested, as seen in Figure 13. This idealization focuses around moving from an upright bin position to a flat laid out bin position during the challenge runtime.

The realization of this positioning would provide significant benefits to the path planning, grasping, and vision systems. For the design to meet the requirements of the challenge the bins must return to their original positioning before the end of the competition runtime, and the shelf cannot be touching the outer supporting frame at the beginning or end of the challenge run.

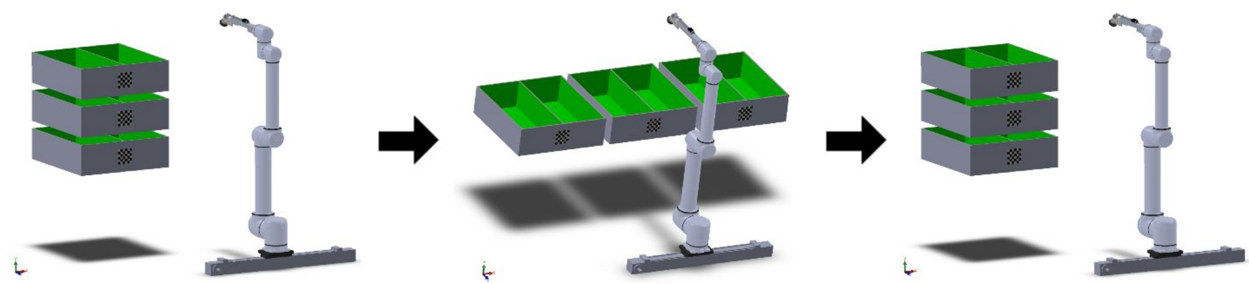

Figure 13. Shelf design idealization

The current shelf design (based on the idealization) utilizes rotary actuators in order to fold out bins onto a supporting frame, as seen in Figures 14 and 15. The design is not able to realize a completely parallel layout for the bins, however it does allow for a top-down approach to each bin which is the largest benefit of the idealization. The design also features three extra sensors which can aid in perception runtime and robustness. The middle section of the shelf is hollowed out as much as possible to allow space for the arm to access the bottom bins. The first priority for shelf fabrication will be to fabricate a non-actuated wooden mockup in order to verify the design viability.

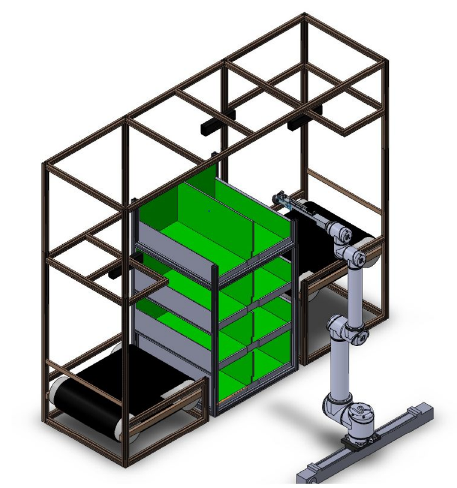

Figure 14. Shelf design - starting position (representative of competition run start/end)

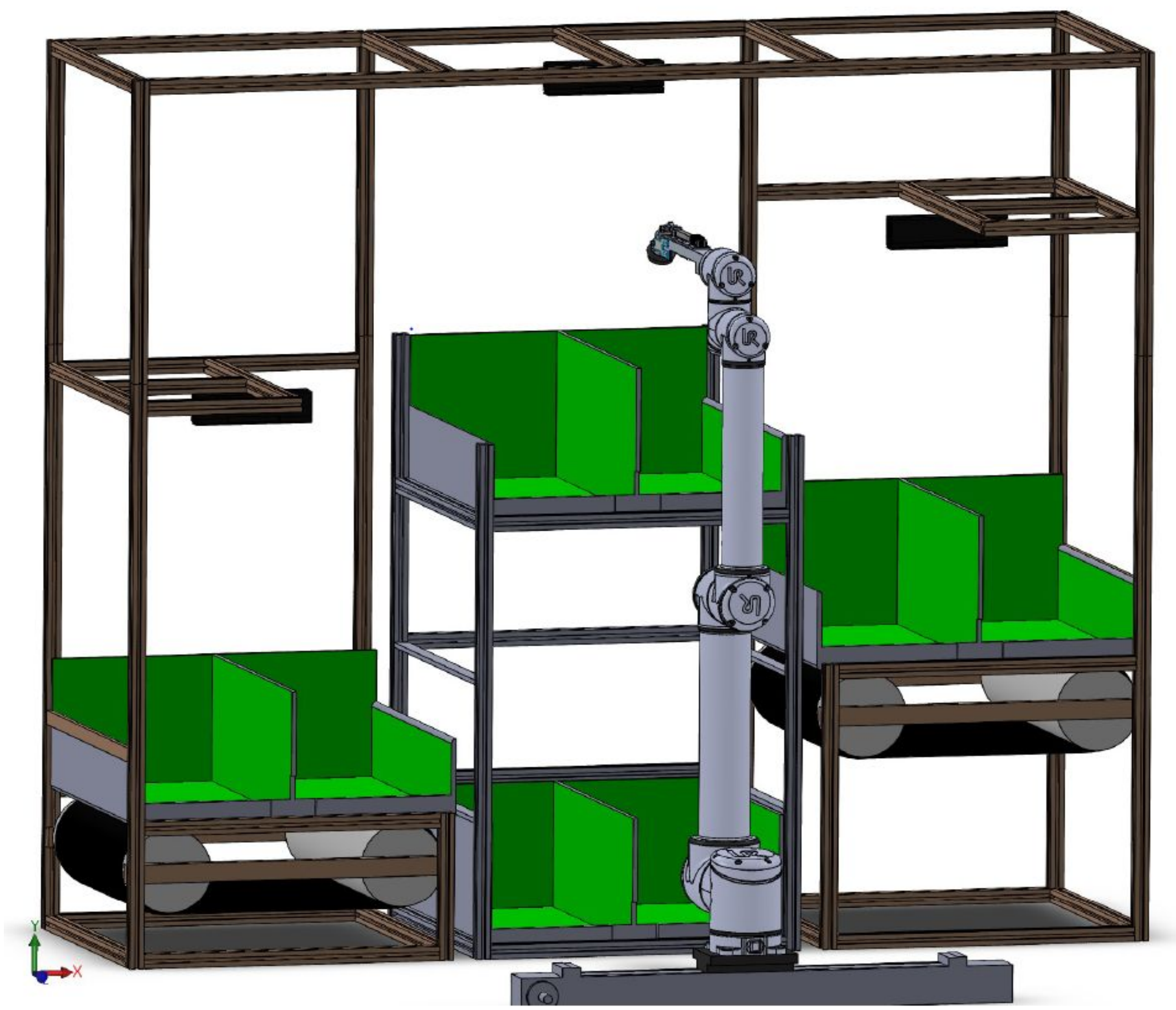

Figure 15. Shelf design - unfolded position (representative of competition runtime)

7.4 Strong / Weak Points

The strongest portions of the system currently include the following: established subsystem modularity and state machine integration, solid path planning and grasping controls, and a good overall system design (as validated through feedback from SBPL). Subsystem modularity continues to prove to be instrumental by allowing for quick identification of the origins of system errors, with the state machine providing clear feedback for the system performance (it is currently very rare for there to be confusion as to whether an error is being caused by the grasping or perception subsystems for example). Path planning and grasping controls are continually performing as expected and have demonstrated easy modification and clear code readability.

The current weak points for the system include the following: "graceful" failure strategies have not been implemented within the state machine, the vacuum generator for the grasping system is taking longer than anticipated to be procured, and the perception system is not performing to the standards which were previously outlined by the project schedule. In order to tackle these issues the team is allocating extra work time over Winter Break. The state machine modifications are manageable and simply require time for testing and code revision. The vacuum system is currently being researched with aid from outside distributors and is anticipated to be procured by mid-January. The perception system requires higher granularity in its design specifications in order to stay on tasks and to have more quantifiable metrics. This granularity will be accomplished through the development of detailed flow charts and UML diagrams, which are to be finished before the start of the Spring semester.

8. Project Management

8.1 Work Breakdown Structure

The Work Breakdown Structure is divided by subsystem, ash shown in Figure16. The shelf subsystem is newly added based on the 2017 new rule. Individual work packages are assigned to individual task holders.

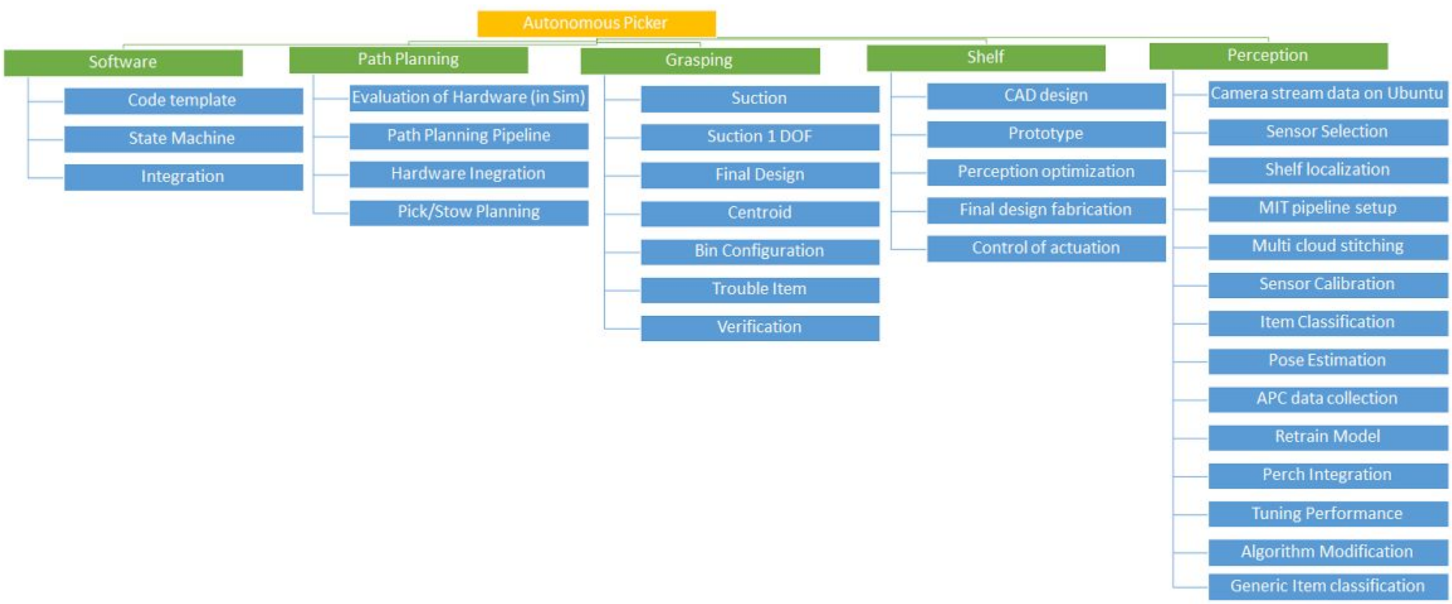

Figure 16. Work breakdown structure by subsystem

This WBS helped us to look at priorities and dependencies. For example, software integration will be dependent on functional subsystems. Perception shelf localization can be made easier by putting apriltags on the shelf. These kind of dependencies were taking into account for making the milestones and schedules.

We have a more granular four level WBS for our own use. Based on fall semester, a well defined work package makes assigning task holders easier. Listing out the details for each work package also helps to see some dependencies that wouldn't be apparent in a higher level WBS.

Laying out work packages also helps us to define the scope we are aiming at for SVE and the picking challenge. The main considerations are manpower and budget. Taking perception as an example, generic item classification is out of scope for SVE, but we need to do it for the competition. Also, actuating the shelf is not included in SVE, since the expense would be beyond the $5000 budget and this task would be dependent on funding. We still included these packages in the schedule to help ourselves stay organized, but they will not become a part of PR or SVE.

8.2 Schedule Status

The high level milestones for 2017 Spring Semester are shown in Table 2. A more granular schedule will be made based on our progress during the winter break.

For the past 2016 Fall Semester, everything was on schedule except the perception subsystem. This was mainly due to our team's lack of experience with computer vision and the work was not well planned without a project manager.

Table 2. Schedule for Spring Semester

Milestone	Date
Perch Integration Shelf mockup	January
Item Pose Estimation via RGB data Integrate Final Design Gripper	Early February
Planning Hierarchy Established	Early March
Grasping Algorithms for Final Design Gripper New shelf fabricated	Mid March
All sub-systems Integration and testing	Late April
Spring Validation Experiment	Early May

We will catch up with perception progress by putting extra effort during the winter. This includes familiarizing with tools such as Caffe, Point Cloud Library, and Perch. Understanding of the tools and go through the learning curve can help us to be more efficient next semester. Also, familiarization with basic concepts and tools can give us a better sense of how long each task in the vision subsystem would take, and help us be more organized.

In addition, we plan to get Perch integrated over winter without having other subsystems having dependency issues. Perch will handle the non-deformable object recognition, and let us to run the whole state machine with better computer vision accuracy in the beginning of the Spring Semester. The first two to four weeks for Spring Semester will be mainly devoted to getting perception subsystem up to speed.

8.3 Test Plan

8.3.1 Milestones

The milestones of progress reviews for this project have been determined in accordance with the goal of creating a competitive robotic system for the 2017 Amazon Picking Challenge. Since the perception sub-system is running behind the schedule now, we will focus on it in the beginning of spring semester. Table 3 below shows milestones for each progress review in details.

One issue with these milestones is a lack of concrete numbers for vision deliverables. At this we are not sure what requirements are reasonable, which is a problem by itself. To solve this problem, Team Plaid will be devoting time over the Winter break to bring perception up to speed and make more specific goals for the Spring semester.

Table 3. The milestones for progress reviews

Progress Review 7		
Date	Late January 2017	Test Method
Deliverables	Integrate Perch algorithm to estimate object pose	Demonstration
	Control linear base via ROS	Demonstration
	Gripper trade study based on 2017 items	Presentation
	Update mount and gripper in planning scene	Demonstration
	New Shelf mockup	Demonstration

Progress Review 8		
Date	Mid February 2017	Test Method
Deliverables	Image data collection for 2017 APC items (turntable)	Demonstration
	Final Design for end effector	Demonstration
	Optimize centroid strategies for 2017 items	Demonstration

Progress Review 9		
Date	Late February 2017	Test Method
Deliverables	Retrain Caffe model for APC 2017	Demonstration
	Integrate final design end effector with arm	Demonstration
	Implement full planning hierarchy	Demonstration

	Develop grasping strategies for items contactin bin wall	Demonstration

Progress Review 10		
Date	Mid March 2017	Test Method
Deliverables	Improve accuracy by tuning Caffe model parameters	Demonstration
	Fabricated new shelf	Demonstration
	Grasping strategies for crowded bin/blocked items	Demonstration
	Control final design end effector via ROS	Demonstration

Progress Review 11		
Date	Early April 2017	Test Method
Deliverables	Multiple point cloud fusion	Demonstration
	Grasping different grasping strategies based on item	Demonstration
	Whole system integration	Demonstration

Progress Review 12		
Date	Mid April 2017	Test Method
Deliverables	Improve system robustness (enhance state machine to show fault recovery capability)	Demonstration
	Improve perception pipeline by point cloud fusion	Demonstration

8.3.2 Test Plan

The outline for the Spring Validation Experiment is specified in Tables 4, 5, and 6 below. The inclusion of experiment metrics which demonstrate “graceful” failure are being considered for future inclusion into the SVE.

Table 4. Working area and equipment requirements

Picking Scenario for APC	
Date: 04/25/2017 **Location:** NSH Level B Fence/High Bay **Testing Area:** 3m x 2.5m working area	**Testing Equipment Needed:** ● 1-DOF Suction Gripper/Final Design Gripper ● 4 RGB-D Cameras (Realsense SR300 / Kinect 2.0) ● UR10 w/ Linear Actuator ● Fabricated Shelf and Platform Framework ● 3 Totes ● 2 Workstations ● 32 APC 2017 Items

Table 5. SVE test process steps

Picking Scenario for APC
Test Process: 1. The shelf will be populated with 32 items from the APC 2017 dictionary. 2. The system will be given a JSON file reflecting the correct item bin locations, as well as the desired tote for each item. 3. The perception system will recognize items in each bin and report the results to the workstation. 4. The perception system will localize itself and the shelf to the robot arm platform. 5. The perception system will detect the item of interest determined by the workstation and recognize its pose to find a valid suction/gripping surface. 6. The path planner will move the UR10 outside the desired bin or over the desired item surface. 7. The grasping system will implement a strategy based on the item to attempt to grasp the item on a predefined surface, communicating with path planning to move the arm. 8. The suction system will adhere to the item surface. 9. The path planner will move the item over the desired tote. 10. The grasping system will disengage suction and drop the item into the tote. 11. Repeat steps 5-10 till time is up or all tasks are completed.

Table 6. SVE quantitative goals

Quantitative Goals

1. **Pick up at least 12 items and drop them inside their target totes within 15 minutes, dropping no more than 2 items to the floor**

2. **Drop items into the totes from no more than .3m from the bottom of the totes**

3. **Generate an item report in the form of a JSON for the items remaining on the shelf, with 100% accuracy for item bin locations (excluding any dropped items)**

Figures 17 and 18 below show an example of the testing setup. The system will reside inside of a taped area to demonstrate that it is fitting within the 2.5m x 2.5m working area. The arm will pick items from the shelf and place them into a specified tote within the timeframe.

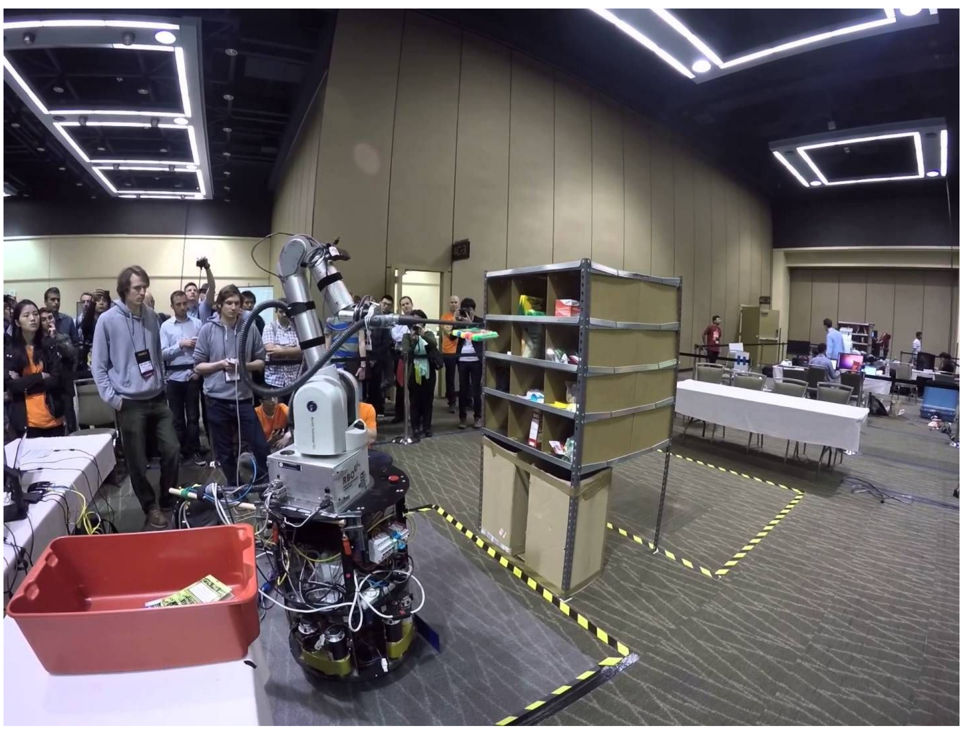

Figure 17. SVE example environment - Team RBO 2015 picking [2]

Figure 18. SVE example environment - Team RBO 2015 placing item in tote

8.4 Budget

The total budget for the MRSD Project course is listed in Table 7. The costs pertaining to the competition including the transport of the robot to Nagoya, Japan have not been considered as part of the budget for the MRSD Project Course. The items yet to be purchased have been marked in red. The funding deficit shall be managed by securing funding from the SBPL lab and/or cost cutting as described in our risk reduction plan.

Table 7. Budget/ Projected Total Spending

Category	Details	Quantity	Cost
Perception	Intel RealSense Camera	2	$298
	USB Hub	1	$40
Path Planning	Linear Base System	1	$2,588
	UR5/UR10	1	$0
System Control	Server Rack and Flatbed Comp.	1	$1,000
	Workstation(GPUs)	2	$0
Grasping	Firgelli L16	1	$0
	Suction System	1	$3,400
	Power Box (Task 12)	1	$0
	Prototypes	1	$200
Shelf	Slider Mount	1	$400
	Shelf Frame	1	$800
	Shelf Actuators	4	$400
	LEDs & Diffusers	1	$100
Competition	Material Transport		
	Power Adapter		
	Spares		
Total			**$9,177**
Funds Remaining			**($4,177)**

The big ticket items for the project are listed in Table 8. The entries in red are the purchases yet to be made.

Table 8. Big ticket Items

BIG TICKET ITEMS	Linear Base	$2,588
	Vaccuum	$3,400
	Shelf	$1,700
	Server Rack and Computers	$1,000

The table in Table 9 shows the current budget status on a subsystem basis. All expenditures listed here have already been made. The red entry shows the funds remaining from the MRSD budget, today.

Table 9. Current Budget Status

CURRENT BUDGET STATUS	Perception	$300
	Path Planning	$2,558
	System Control	$0
	Grasping	$0
	Total	$2,858
	Funds Remaining	$2,142

8.5 Risk Management

Team PLAID identified 4 major risks ID 1, 2, 3, 4 in the Preliminary Design Review. For the Critical Design Review the team added two more risks ID 5, 6.

In the course of the Fall semester, Team PLAID mitigated three risks, ID 2, 3, 5 and is currently tracking three risks, ID 1, 4, 6.

Table 10. Risk Management Table

ID	Risk Title	Type	Description	Likeli hood	Impact	Mitigating Actions
1	Gripper Design Pivot	Techical Project Schedule	APC 2017 items are significantly different requiring novel gripper design and testing not realizable in time window	4	4	>Design/ Purchase Alternate Gripper >Manufactured Best-Guess Gripper
2	Different Platform Secured	Techical Project Schedule	UR5 is secured for project,APC	0	0	>Design easy to port code
3	Platform Availibility	Techical Project Schedule	UR10 Platform Sponsor does not provide platform	0	0	>Contact Alternate Sponsors >Secure UR5 platform
4	Generic Item Detection	All	Scope of generic items too large to implement	4	4	>Integrate PERCH >Get MSCV team
5	Data from Realsense Camera Unreliable	Techical Project Schedule	Data does not stream in the ROS driver for SR300 Realsense camera	0	0	>Write ROS wrapper node >Use Kinect Sensor
6	System Cost Over Budget	All	System cost to be competitive in APC requires custom shelf design and more expensive vacuum system	4	5	>Design Passive Shelf >Secure Funding from SBPL

Our treatment of risks is listed below:

Risk 1: The item list for APC 2017 has not been released yet. Purchasing alternate grippers is not possible because of the status of our budget. We have fabricated our best guess 1-DOF suction gripper as part of our risk reduction plan. The gripper will be tested on the new item list in the Spring Semester and the risk shall be appropriately written of or escalated.

Risk 2: Platform availability is no longer an issue. We have received a confirmation towards the procurement of our UR10 platform. This risk has been retired.

Risk 3: Securing a different platform is no longer a risk as mentioned above. We performed risk mitigation for the same from the beginning of the semester by ensuring that the planning code was easy to port to the UR10 platform. This risk has been retired.

Risk 4: Although the scope of generic item detection and pose estimation remains large, we have partially mitigated this risk by the addition of 1 MSCV student to the perception team. We plan to further mitigate this risk by integrating PERCH into the perception pipeline early next semester.

Risk 5: The team was unable to obtain usable data from the Intel Realsense camera. We have mitigated this risk by choosing to use the Kinect Camera for the competition. This risk has been retired.

Risk 6: The projected expenditure is already almost twice the MRSD project course budget. The team plans to mitigate this risk by securing additional funding from the SBPL Lab or by designing a passive shelf for the competition.

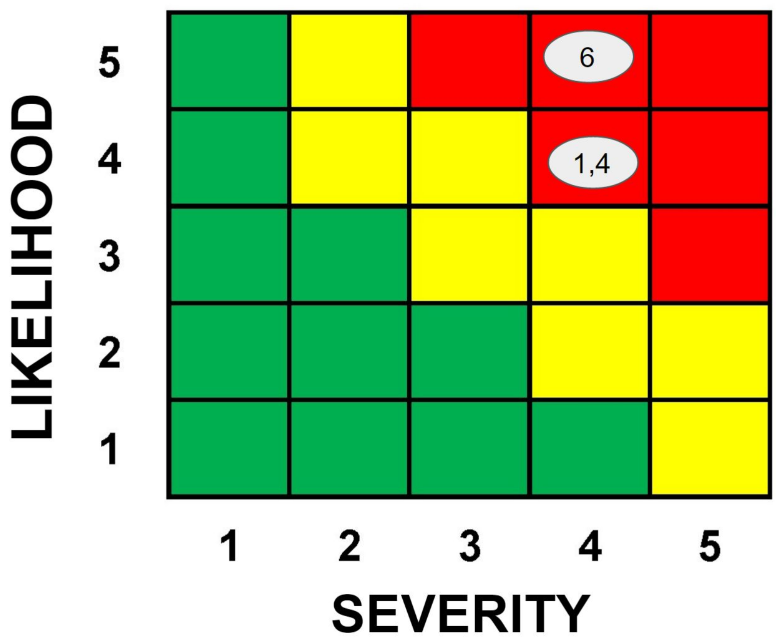

Figure 19. Risk Likelihood-Consequence Table tracking current risks

9. Conclusions

9.1 Key spring semester activities

According to the WBS and PR goals, there are several key activities in spring semester. Once new item list is received, grasping capability of 1 DOF end effector will be tested for every item in the list. If coverage doesn't meet our expectation, new gripper need to be designed and fabricated before the end of March. Grasping strategies will also depend on new item list. Instead of using centroid top down strategies, new grasping strategies will be developed based on the characteristics of new items before the end of April.

Based on the new APC rule, we have freedom to design shelf on our own. Planning scene of new shelf and its prototype will be implemented before the end of January. Optimization of perception regarding to shelf will start engaging in the end of February.

As for perception subsystem, Perch Algorithm will be integrated before the end of January to complete the perception pipeline. By using Perch, item pose estimation and basic item classification is available for our system. Since Perch is the foundation of perception in our system, it will be the first priority to accomplish. Also, Caffe is the standard technique for image classification. Understanding and training Caffe model will be the following key tasks after Perch. Well-trained Caffe model should be ready in the end of February.

Although planning is fully integrated in our system and works well, the performance could still be improved by training experience graph. Experience graph will be trained in the end of February. Also the whole planning pipeline will be evaluated and optimized by tuning planner hierarchy.

9.2 Lessons Learned

Our team has learned a few important lessons over the semester that will help us complete our work in the Spring. The first lesson is that we needed to have a distinct project manager role. Until recently, no one has a had a good view of the entire system and that has hurt us in many ways. In the past when it came time to integrate and make plans there was frequently a disconnect between teammates that would take time to resolve. To remedy this, our team has appointed Mike as our project manager in order to maintain a unified vision of the project and make sure all subsystems stay on track.

Another lesson we have learned is to give more conservative estimates for the time a task will take, especially if it is the first time doing such a task. Underestimating timing causes a lot of friction and leads to stressful nights before demonstrations that should be avoided at all costs. In a similar vein, we also need to make better use of our scheduling tools from Systems Engineering. At some points during the semester we went out of touch with our existing plans which made getting back in synch difficult. If we had used those tools to their fullest potential, there would have been less management turmoil.

The last lesson we learned is that a good way to make a perfect system is to make an imperfect one first. There were a few occasions where learning valuable lessons on a weak prototype would have saved much more time than trying to make something perfect on the first try. We are now committed to a process of prototyping and revision that should lead to faster and better designs and executions.

10. References

1. Amazon. Amazon robotics challenge. Retrieved December 15, 2016, from https://www.amazonrobotics.com/#/pickingchallenge
2. Team RBO, Amazon Picking Challenge 2015.