Assistive Intent Recognition and Manipulation

Final Report

Team H: Intent Bot

I-Chen Jwo
Jiahong Ouyang
Karsh Tharyani
Liz Yang
Ting-Che Lin

Project Sponsor
Prof. Henny Admoni

The Robotics Institute
Carnegie Mellon University

May 07, 2018

Abstract

Assistive technologies are still blooming. When it comes to addressing the needs of the handicapped, in our case people with upper-body dysfunctions, assistive technologies, today, usually use brain-computer interface(BCI) or electromyography(EMG) to extract the nerve impulses of the user, and in turn, once they have determined what the user wants, either through an external actuator or an actuator attached to the handicapped himself/herself, they perform the manipulation task. Intent Bot is yet another attempt at doing so, however with the variation that, instead of using the above modes(BCI or EMG), it combines the user's speech and gaze in order to interpret the intent of the user and commands a manipulator to pick and place the intended object. Unlike other modes used to interface with assistive technologies, Intent Bot requires little calibration and is entirely non-intrusive with regard to the user's anatomy. In this work, we present the requirements of the system(Intent Bot), the current system status, the system's architectures, lessons drawn from the initial prototyping, and the conclusions drawn from our progress so far. Additionally, in the end, we speculate the future work which can improve the functionality of the system overall.

Contents

1 Project Description

Many types of orthopaedic or neuro-muscular impairments can impact upper body mobility. These include amputation, paralysis, multiple sclerosis, and many other impairments. The impact on upper body mobility can range from full upper body paralysis to limited functionality in hands for grasping. The U.S. Census Bureau has indicated that more than 8.2% of the U.S. population or 19.9 million Americans suffer from upper body limitations [3].

An upper body mobility can have a severe effect on the quality of life of the patient. It can prevent the patient from performing everyday tasks such as picking up a cup or opening a door. Patients with severe mobility impairment issues can often teleoperation a robot arm to assist with daily tasks [4]. However, the current form of teleoperation requires significant cognitive and physical effort from the user [6]. This often causes the teleoperation task to be time-consuming and exhausting.

Shared autonomy has been shown to be able to decrease operator-fatigue and reduce the time taken per task [5]. Furthermore, novel methods through non-verbal inputs such as eye gaze have been shown to augment shared autonomy capabilities and their effects [6]. Motivated by these results, the project aims to create an assistive robot that would aid a user in everyday tasks by predicting his/her intent through gaze and speech. We believe that our intuitive system would allow for greater independence to patients with limited mobility.

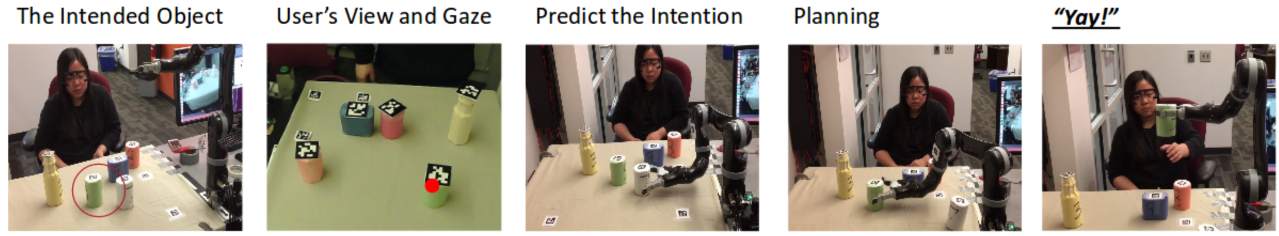
2 Use Case

War hero, valiant, veteran, Captain Jean Rodgers walks into her favourite Italian restaurant. She is here to meet an old friend from the academy.

As Jean enters the restaurant, she is greeted by the waiter who escorts her and her friend to a table. When they reach the table, the waiter hands Jean a pair of sleek glasses along with the menu. her friend is curious about the robotic arm along with the depth camera attached to the table top, and about the glasses which Jean holds.

Jean lost the strength and stability in her forearms due to a bullet wound in the War. She explains how the bot in front of them, called the Intent Bot, assists her at manipulating objects. She can gaze at an object, which is lying on the table and which she wants to pick, and at the command of her voice the robotic arm will pick and place it next to her. Her friend is simply baffled at this. Jean demonstrates. First, she wears the pair of glasses which, essentially, track Jean's pupils. Next, she gazes at the can of soda in front of her, and says, "Pass me that soda-can!" the robotic arm reaches for the soda-can, which weighs about 400 grams and is around one arm-length away, and places it near Jean's right arm. Her friend is inquisitive. He is surprised about how the robotic arm, despite all the food and the tableware in its way, identified the soda-can, and without dropping it or disturbing any other object, placed it at the exact location where Jean had intended. Although there were many soda-cans on the table, it identified the exact soda-can which Jean intended. As they continue dining, the waiter approaches their table to refill their glasses with water. The Intent Bot was reaching for the can on the table and was about to collide with the waiter when Jean presses with her foot the emergency-stop button. The bot comes to a complete standstill. It is an emergency feature of the robot meant for just this kind of scenario. The waiter resets the manipulator arm and the Intent Bot reactivates.

It is the end of a wonderful evening as they bid each other goodbye. It was an amazing evening and everything that Jean could wish for. Figure 2.1 gives an idea about the use case scenario described above.



Step 1:
User with upper body dysfunction is sitting on a table for lunch and wants the green soup can; she stares at it for 5 - 10 seconds

Step 2:
The Intent Bot records and tracks the gaze of the user and sends this intention to the manipulator

Step 3:
The manipulator plans in the cluttered environment to give to the user the desired object

Figure 2.1: Graphical Representation Intent Bot

3 System-Level Requirements

The system-level requirements of the assistive robot have changed significantly for the final system. The main changes were the addition of the capability for the intent bot to remove clutters and the exclusive use of gaze information for intent recognition. From our Fall Validation Experiment, we have realized that the gaze information was sufficiently informative to identify the object that the user intents for. Furthermore, we also realized that the intent recognition system can be more intuitive if the gaze information was used exclusively. As such, we have removed the system requirements regarding predicting intention from speech and have added system requirements for predicting intention from gaze only. Additionally, we also expanded the system requirements of the manipulation subsystem to include the requirements for the clutter removal. Finally, for M.P.7 we are required to improve the system so it will recognize the user's intent more intuitively than the system presented in the Fall Validation Experiment. This requirement is added because the speech-and-gaze-based recognition system requires the users to fix their gaze at the object they intend the robot to pick up. However, the gaze fixation requirement is very obtrusive and cognitively taxing. So we believe that by making Intent-Bot more intuitive, the user's experience will improve drastically.

3.1 Performance Requirements

3.1.1 Mandatory Performance Requirements

M.P.1 Will recognize the user's intent correctly 80% of the time
The robot should correctly recognize the user's intent all the time. Due to technical challenges, a more conservative requirement is set.

M.P.2 Will recognize the user's intended object from speech correctly 90% of the time

M.P.3 Will update user's intention 15 times per second
For a responsive system, the intent bot will update its most current intent prediction at the rate of 15 times per second.

M.P.4 Will pick up the intended object 60% of the time in a simple case
For a frustration-free experience, the robot must pick up the correct object all the time. However, technical challenges forced us to have a more conservative requirement. This requirement is for a simple case when there is only one obstacle object(defined in the Manipulation Subsystem).

M.P.5 Will pick up the intended object 33% of the time in a complex case
A complex case is a close to an ideal representation of the real-world where there are more objects on the table and the intended object is blocked by a number of other objects. The complex case is one wherein there are three obstacle objects between the manipulator and the intended target object.

M.P.6 Will stop with the emergency stop input 100% of the time
To ensure the safety of the user and those around him/her, the robot should be stopped anytime the user judges the robot to be dangerous.

3.1.2 Desirable Performance Requirements

D.P.1 Will recognize the user's intent correctly 90% of the time

D.P.2 Will correctly place/bring(to user) the object 80% of the time

D.P.3 Will update user's intention 30 times per second

D.P.4 Will pick up the intended object 80% of the time

D.P.5 Will identify and pick up the correct object in a clustered environment where objects are at least 10 cm apart
For the system to function in the real-life scenario, it must be able to pick up objects in a cluttered environment. This has been refined from 5 cm to 10 cm as the fingers are themselves about 5 cm thick.

D.P.6 Will complete the pick and place task within 1 minute
To integrate seamlessly into a patient's life, the task must be completed as soon as possible. The 1-minute mark is a good compromise due to technical challenges.

D.P.7 Will pick up objects that weight up to 1 kg
Common items in everyday pick and place task are usually below 1 kg in weight.

3.2 Non-Functional Requirements

3.2.1 Mandatory Non-Functional Requirements

M.N.1 Will be more intuitive to use than the joystick/teleoperated control
For the system to benefit the patients, it must be more intuitive to use than the manual system they are using right now.

M.N.2 Will cost less than $5000 to develop
This is a budget constraint imposed by the MRSD project.

M.N.3 Will be robust to varying lighting conditions
Lighting in the real word will vary due to many conditions

M.N.4 Will be more intuitive to use than the system presented in the Fall Validation Experiment
The intuitiveness will be measured through both subjectively and objectively metrics. The users will judge subjectively which system is more intuitive and the user's engagement time with the system will also be analyzed. We believe that the system that requires less user engagement time and is ranked more intuitive by the user will be more intuitive.

3.2.2 Desirable Non-Functional Requirements

D.N.1 Will function robustly in non-laboratorial condition
To assist the user in real life, the system must function robustly in the non-controlled and non-constrained environment

D.N.2 Will be portable and be untethered
The system must be portable and untethered so it could follow the user to whichever location the user chooses.

4 Functional Architecture

The functional architecture of the Intent Bot is as shown in Figure 4.1. The functions can be broken down serially into namely Intent and Scene Observation, Interpretation, and Object Manipulation.

4.1 Environment Observation

The function of Environment Observation is to preprocess raw data input into a usable form by the manipulation and intent prediction subsystem.

4.2 Intent Prediction

4.2.1 Assumptions for Intent

At this point, a particular assertion needs addressing for the system to work seamlessly, and without any hiccups. It has been shown in prior research that when a user is intending to pick an object, he/she looks at it [8]. This is why the gaze is such an important measure to address and identify the user's intent.

4.2.2 Intent Recognition

The function of intent recognition is to extract the intent information from the gaze recognition subsystem and the environment observations to output the target object's ID.

The function of gaze recognition is to achieve a probabilistic map of the gaze's location. The input is the image of eyes and the image of the environment. The output of this part is the instantaneous intent location of the image of the environment.

4.3 Manipulation

Moving down the functional architecture, once the intent has been recognized, the Interpretation outputs the pose of the intended object. The pose is sent as an output to the Object Manipulation subsystem, whose task is to localize the object and plan a path to reach and, later, place the object based on the user's intention. In addition to bringing the object to the user, the subsystem is also responsible for identifying and removing the clutter in the environment.

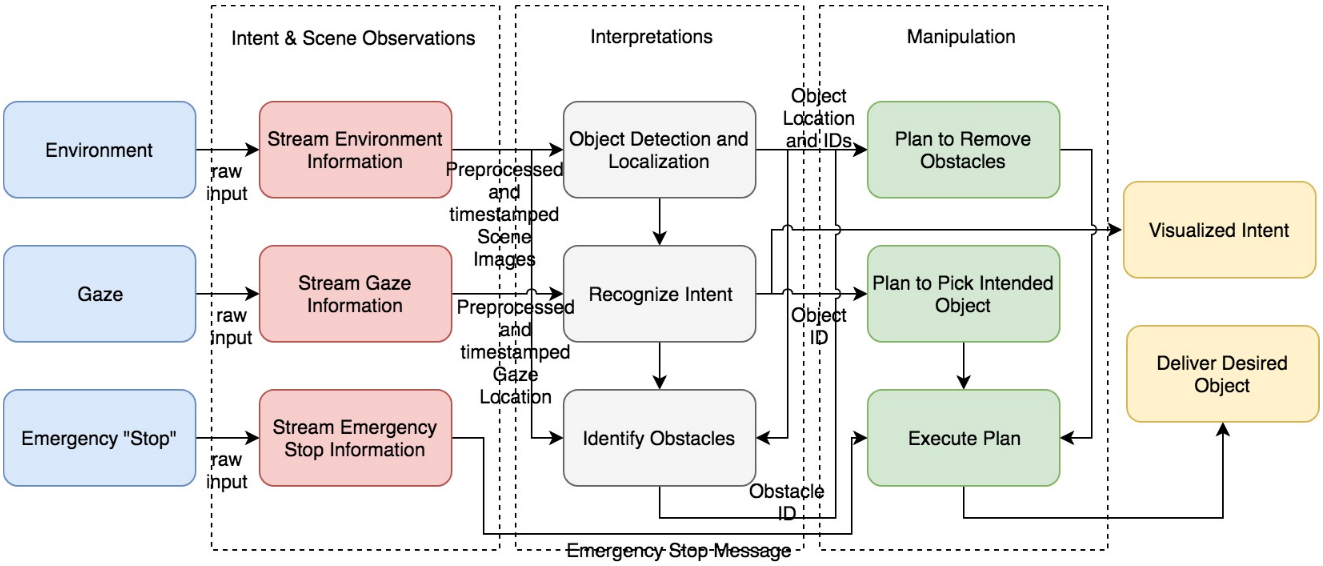


Figure 4.1: Intent Bot's Functional Architecture

5 System-Level Trade Studies

There are several subsystems that were used for the Fall Validation Experiment but weren't used in the Spring Validation System. They are included here for completeness, although very briefly.

5.1 Speech and Language Subsystem

Speech and Language Subsystem's main function is to recognize and understand the speech command of the user. To recognize a user's speech, we need to gather the sound waves generated by the user through a microphone. This sound waves will be processed through a speech recognition module to obtain the transcript of the speech. The first trade study will be about the type of microphone that will be used and the second trade study will be about the transcription software. To understand the speech command, the obtained transcription would be processed by a natural language processing module. This module would output the desired action and the object to be manipulated. The third trade study will be about this module.

5.1.1 Speech Acquisition System

Microphones come in many different types and forms. However, by limiting the microphones for speech recognition purposes we can narrow the microphones down to the following three forms: headsets, desk microphones, and built-in microphones. The microphone headset was chosen due to its accuracy and ease of use.

5.1.2 Speech Recognition

There are several different ways to construct a speech recognition system. The system could be constructed using external speech APIs, using open-source speech recognizer, or self-trained neural network. Since high accuracy and low time to setup is paramount for our system to function properly, we have decided to use external speech APIs even though it is not a free solution.

5.1.3 Command Parsing

Commands in the English grammar are known as imperative sentences. This group of sentences has a specific grammatical structure that we could take advantage of. By obtaining the action verb and the direct object, we would know the action that the user wants to be performed and the object that will be affected. Syntactic parsing is a very well established research area in natural language processing. We, again, chose external natural language processing APIs for our syntactic parser due to the ease of setup and accuracy.

5.2 Gaze Tracking

There are two ways to do the gaze tracking: one is using the gaze tracking glasses and the other one is using free-head gaze tracking.
For the gaze tracking glass, there is already a concise, accurate, and commercialized product with which we can get the gaze position in the world frame. It achieves a gaze accuracy of 0.60 degrees and a precision of 0.08 degrees. The time latency is 5.7ms, which is not sensible to humans. So, this product is an ideal choice during the setup process.
For the free-head gaze tracking, projective-Geometry method is used to estimate the gaze position. Compared to gaze tracking with glasses, this technology will surely give more freedom to the users and will be more comfortable to use. But for now, the free-head gaze tracking achieves the accuracy of 1 degree, which is 60% lower than that of glasses. This method is, therefore, less precise for our intent recognition system.

	Gaze Tracking Glasses	Free Head Gaze Tracking
Cost	5	2
User experience	3	5
Accuracy	5	3
Total Score	13	10

Table 5.1: Trade Study for Gaze Tracking

5.2.1 Intention Prediction

The intention prediction part will do the analysis of the user's intention based on the information of gaze tracking. There are two ways of doing so. One is using the AprilTag to get the bounding box for each object and decide the intention by the time that gaze locates in the box, but it's not robust of various lightening condition and doesn't include gaze pattern, which is not really intuitive.
The other way is using a neural network to directly get the target ID based on both image and gaze information, but it may require advanced computation resources. The table 5.3 shows the trade study about the intention prediction, which shows that using the neural network is a more appropriate way to solve the problem.

	Weight	AprilTag	Neural Network
Intuitiveness	4	3	5
Computation Resources	3	3	5
Robustness	5	3	4
Accuracy	5	5	4
Total Score	5	3.6	4.4

Table 5.2: Trade Study for Intention Prediction

5.3 Manipulation

5.3.1 Manipulator

The Manipulator is expected to pick up objects found in everyday use, particularly, on the table. Thus, repeatability and payload weight are the most important factors for this project. The displacement error caused by the manipulator needs to be as low as possible so that the manipulator is able to reach the exact position. Also, the manipulator is expected to finish the task under a minute, thus the operation speed should be taken into consideration, as well. Besides that, the working range should be comparable to human reach. Kinova[10] and ROBOTIS[11] are the best choices for the project over the uArm[9] or Mover6[12] as was shown in the Conceptual Design Review, and hence the Mico Kinova arm was chosen.

	Weight	uArm Swift Pro	Mico Kinova	ROBOTIS	Mover6
DOF	3	9	12	12	12
Payload Weight	5	10	15	20	5
Speed	4	8	12	8	8
Working Range	4	4	16	8	8
Repeatibility	5	10	15	20	10
Cost	4	4	3	3	2
Connectivity	1	4	3	3	3
Total Score	5	2.45	3.8	3.7	2.4

Table 5.3: Trade Study for Manipulator

5.3.2 Gripper

Since the gripper will be able to pick up objects in daily life and shall complete the task within a minute, payload weight and speed become essential. Additionally, all the objects come in different shapes, so adaptability is also critical for the gripper. Besides all the performance factors, the weight of the gripper itself shall not cause too much burden on the manipulator. The 2-Finger Gripper was the final choice for our project.

5.3.3 Grasping Information

As we explained above, the manipulator needs the information like the size and the shape of the target object, so that the manipulator can grasp the object. We have two ways to solve that problem. One way is to use AprilTag to access the stored object information in a database. This option is very convenient but is limited to the objects that are stored in the database. The second way is to use computer vision method to segment and identify the object from the environment. The problem, here, pertains to the robustness of the algorithm and also to the situation in which the objects are clustered or blocked. The comparison of these two methods is as shown in the Table 5.4. In conclusion, for grasping information, we decided to use prior knowledge first to establish the whole system.

	Weight	AprilTag	Segmentation
Accuracy	5	5	4
Shape Variation	4	2	4
Complex Environment	3	4	3
Complexity	3	5	3
Total Score	5	4.0	3.6

Table 5.4: Trade study for grasping information

6 Cyberphysical Architecture

6.1 Intent Prediction

In the intent prediction system, the main subsystems are gaze tracking, environment perception, and intent prediction subsystem.

In the gaze tracking subsystem, Pupil-Lab gaze-tracking glasses are used to locate the gaze. There are three cameras. Two of the cameras faces towards the pupils, and the other faces towards the environment. From the image of the first camera, we will locate the pupils, and analyze the direction. After calibration, the location of the gaze is mapped to the world-frame images. The gaze information is then published to a topic where the intent prediction subsystem would use to predict the user's intent.

In the environment perception subsystem, one RGBD camera locates the objects in the environment. The object detection relies on AprilTags. This information is used for the manipulator to do planning. The world frame camera is also published in a topic that the intent prediction subsystem subscribes to.

In the intent prediction part, the gaze information and the world frame camera are preprocessed and fed into a trained neural network. The neural network will use the gaze and world frame camera information to predict the object that the user wants. Once the intention of the user is clear, the manipulator server is called to grasp the target.

6.2 Manipulation & Planning

In Figure 6.1, one can observe that the Manipulation Package is responsible for locating the objects, planning for the intended object, and also removing the obstacles in the way. It utilizes the *planInClutter* API which was developed by the team to decide which object is to be removed before planning to the target object. It also uses the PrPy's CHOMP Planner to optimize the trajectory output by the CBiRRT Planner. In order to localize the objects, April Tags have been put on top of the objects locations of which are detected using a Microsoft Kinect Sensor. The pose of the object is in 2D and does not take into consideration the depth or height of the object.

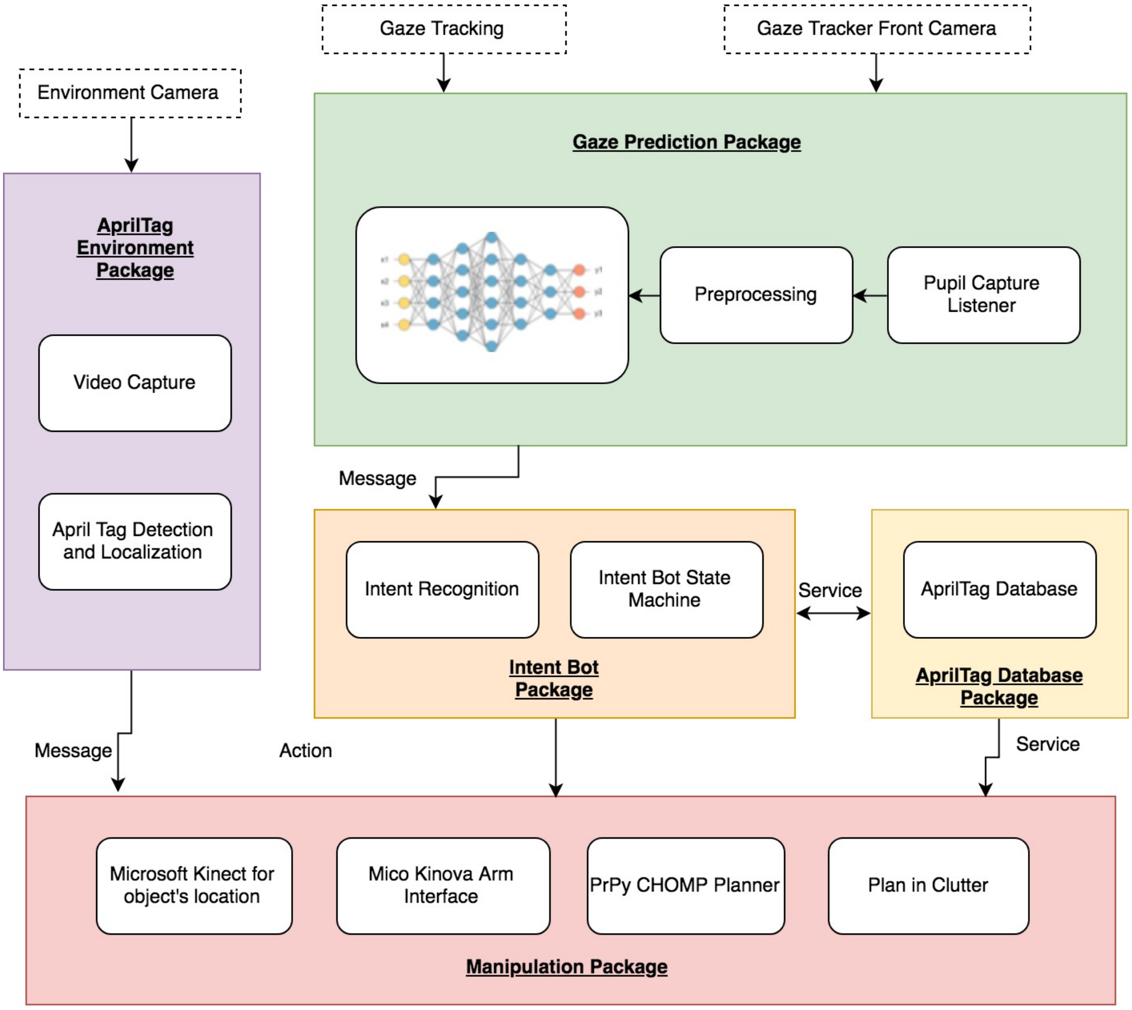


Figure 6.1: Intent Bot's Cyberphysical Architecture

7 System description & evaluation

7.1 Subsystem Descriptions

7.1.1 Overall System Depiction

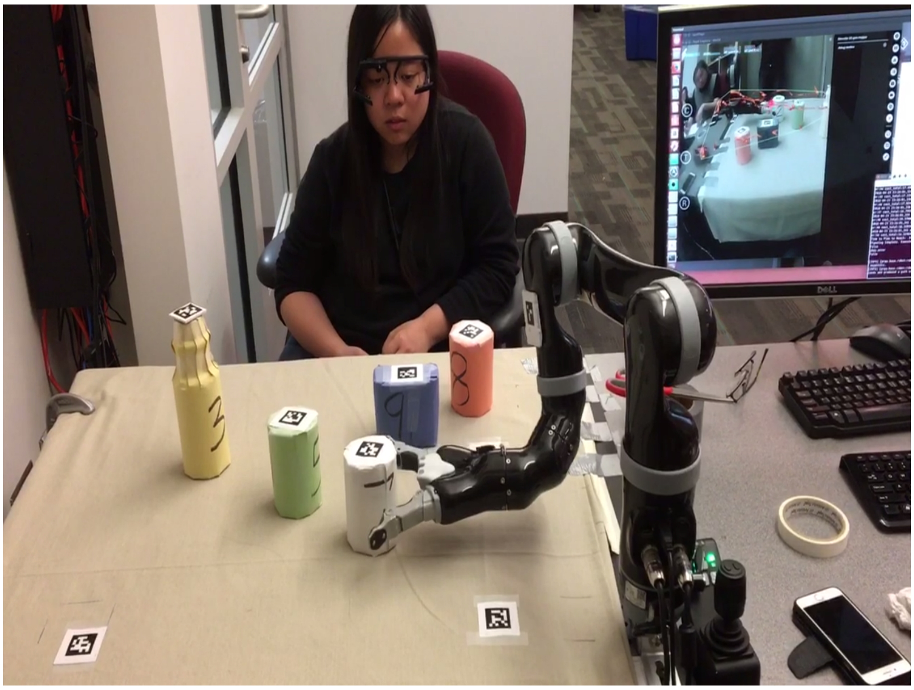


Figure 7.1: Overall System Depiction of the Intent-Bot

Figure 7.1 is the depiction of the overall system which was shown during the Spring Validation Experiment of the intent-bot. The gaze tracker glasses are worn by the user, the Kinova Mico arm is shown in the background, the Kinect mounted overhead.

When the Shift Key is pressed, the gaze subsystem records the gaze of the user and classifies the object that the user is looking at. The output of the classifier is fed to the manipulator. confidence. Then the manipulator server is called and plans to reach the target for the user.

7.1.2 Gaze Tracking Subsystem

In the gaze tracking subsystem, Pupil-Lab gaze tracking glasses are used to locate the gaze. Through calibration, the location of the gaze is mapped to the world-frame images, allowing us to know which item the user is gazing at. A neural network is used in the gaze subsystem to analyze the user's intention in real-time by outputting the color of the intended object. The robot and the operator perform share-autonomy here to decide when to press the key to stop the analysis of the intention.

The following figure shows the window we use to monitor the gaze position and the environment.

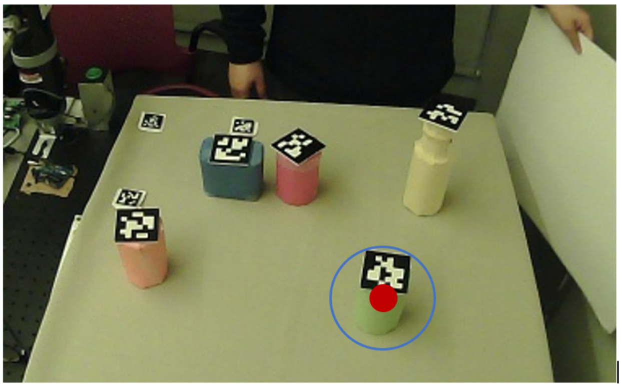


Figure 7.2: Test Window

7.1.3 Intent Recognition Subsystem

The intent recognition subsystem will extract the user's intention using the gaze tracking subsystem. The user's intention will be updated at every frame. Then when the user presses the shifting bottom, the last 30 predictions will be aggregated and the model prediction will be the final intent prediction.

The network is a pretrained AlexNet. The input of the AlexNet is the 224 × 224 image cropped around the position of gaze. The output of the pretrained AlexNet was changed to 6 according to the requirement of our system. We fine-tuned the network for classification. It is predicting which object the user is gazing at in real time.

The beginning and the ending of the prediction process is controlled through the keyboard. With the space key, the system begins to predict, and with a Shift key, the system stops prediction and talk to the manipulator to go for the target.

7.1.4 AprilTag Detection Subsystem for Object Localization

The Kinect Sensor is used to detect the position of the objects for the manipulation subsystem.

First, top-view image frames are read from Kinect. Second, AprilTags that are on the top of each object and the position markers on the table are detected. Third, the 2D position relative to the positions of the position markers of each object is predicted. Finally, the subsystem publishes position for each object to the manipulator subsystem. The image that is captured by the Kinect is shown in figure 7.3.

The current code includes the z-orientation of the tags on the table, as well. However, due to erroneous orientations, this was not incorporated in the current code-base.

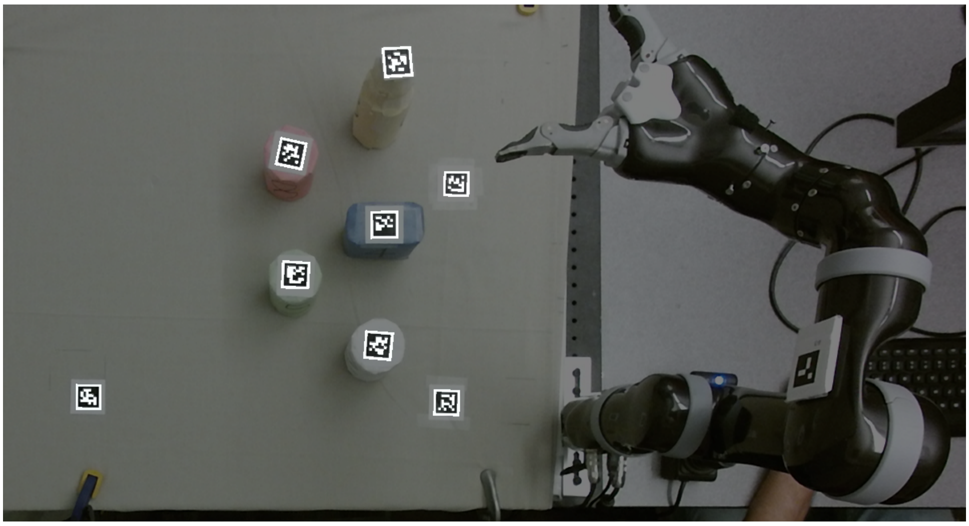


Figure 7.3: Image captured by Kinect

7.1.5 Manipulation Subsystem

The Manipulation Subsystem has to meet the following requirements:

1. Identify the objects in its environment
2. Reach for the intended object in clutter
3. Avoid all by removing or dodging other objects except the target object
4. Will hold the object
5. Will be be an action server for the Intent node
6. Will communicate with the sensor above to identify where the objects are.

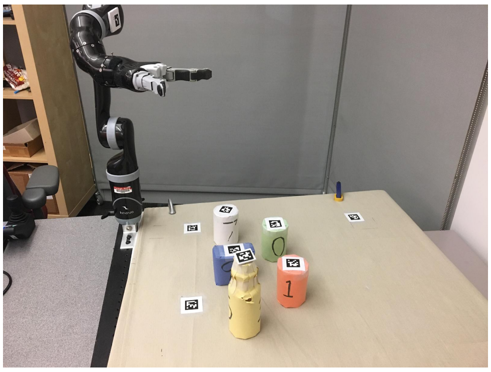


Figure 7.4: Kinova Mico: manipulator arm for Manipulation Subsystem

Figure 7.4 shows the Kinova Mico Manipulator mounted on a bench. The major additions to the system during the Spring Semester have been in the optimization of the plans and trajectories generated by the manipulator and its ability to plan in a cluttered environment.

7.1.6 CHOMP Optimization

A plan to a certain object is generated only when the manipulator loads it in its simulated environment as shown in Figure 7.5. One can observe the various objects on the table which are perceived by the manipulator. A plan can be optimized by an optimizer. For the same, we are using CHOMP.
The CHOMP Planner stands for Covariant Hamilton Optimization for Motion Planning. The advantage of this optimization over a Regular RRT with Path Shortening is that it tries to emerge with a path which is approximately on a straight line joining the end-effector to the object. Thus, a plan generated by the TSR(Task Space Region) based plan which is accompanied by a Constrained Bi-Directional RRT(Rapidly Exploring Random Trees) can be further optimized to obtain acceptable and smooth trajectories. Figure 7.5 shows the approximated links of the manipulator when the plan is optimized.

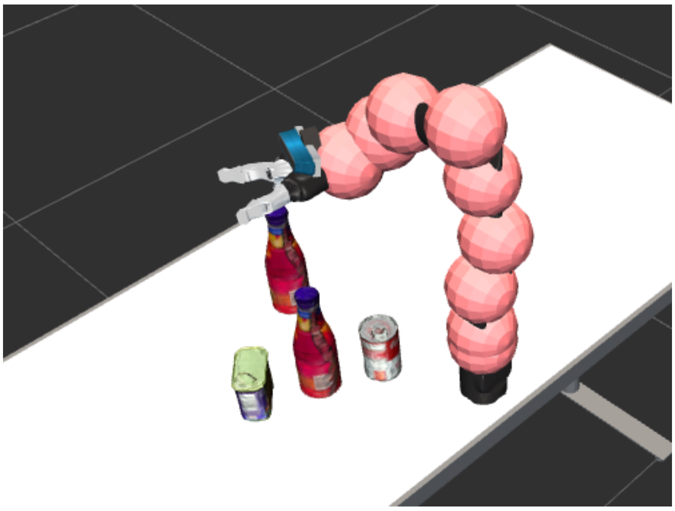


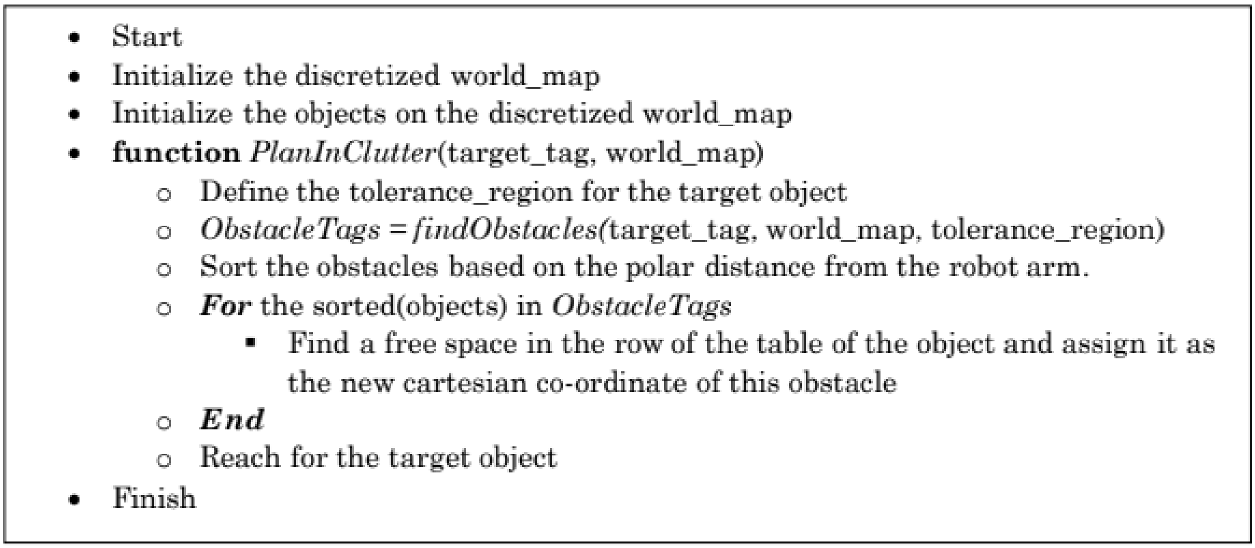
Figure 7.5: CHOMP Planner's approximation of the joints and links

7.1.7 Planning In Clutter

Planning in a cluttered environment is done by removing the obstacle objects out of the way by choosing a particular heuristic. The essential features of the heuristic of planning in clutter(which we are using) are:

- Identify which tags or objects are the obstacles based on a polar coverage.
- Once these objects are identified assign a new place on the table where these objects should be moved.

The reason for choosing of this heuristic is that the planner usually fails or takes longer to plan in the case where there are many obstacles in between the robot base and the target object. Additionally, this also solves the problem of solving for any impossible configuration required to grab the object based on the task space region for grasping the object. Figure 7.6 shows the pseudocode for planning in clutter and 7.7 is a graphical representation of the heuristic.



```
• Start
• Initialize the discretized world_map
• Initialize the objects on the discretized world_map
• function PlanInClutter(target_tag, world_map)
    o Define the tolerance_region for the target object
    o ObstacleTags = findObstacles(target_tag, world_map, tolerance_region)
    o Sort the obstacles based on the polar distance from the robot arm.
    o For the sorted(objects) in ObstacleTags
        ▪ Find a free space in the row of the table of the object and assign it as
          the new cartesian co-ordinate of this obstacle
    o End
    o Reach for the target object
• Finish
```

Figure 7.6: Pseudocode for planning in clutter

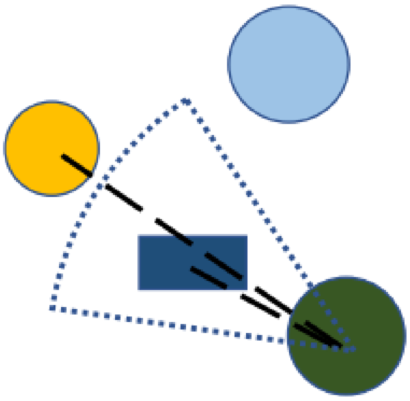


Figure 7.7: Blue dotted region shows the region of interest. This must be free of objects in order to grab the bottle(yellow) from the robot's base(green)

7.2 Modeling, analyze and Testing

7.2.1 Intent Prediction

When the user wants one of the objects in the workspace, he would gaze at the object, and give a command containing the objects. Under such assumption, given the intent, the gaze would not add more information for speech, so they are conditional independent.

The native model uses the position of gaze to crop the world image. Use AlexNet as a classifier.

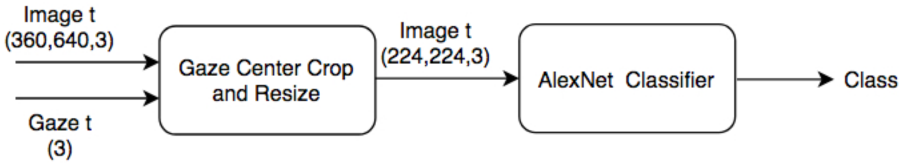


Figure 7.8: native model

The baseline model uses AlexNet as a feature extractor. Use LSTM to process the gaze sequence. Simply concatenate gaze and image features and feed to the MLP.

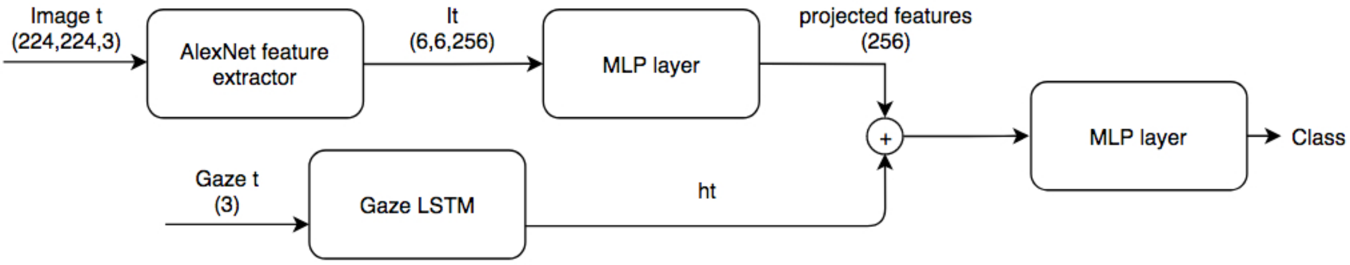


Figure 7.9: baseline model

The spatial attention model adds spatial attention layer to assign different weights to each region based on the baseline model.

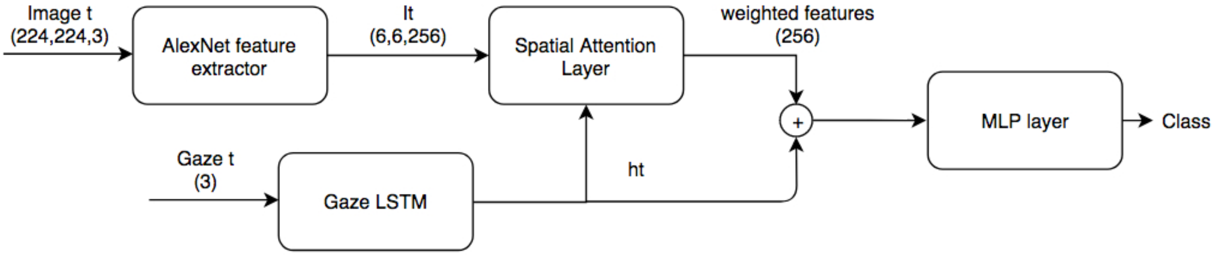


Figure 7.10: Spatial Attention Model

The multiple attention model adds temporal attention layer to assign different weights to each frame based on the spatial attention model.

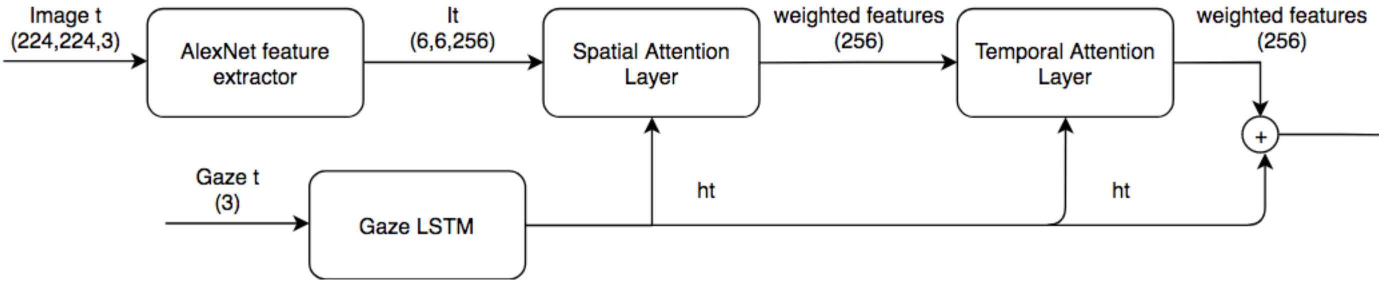
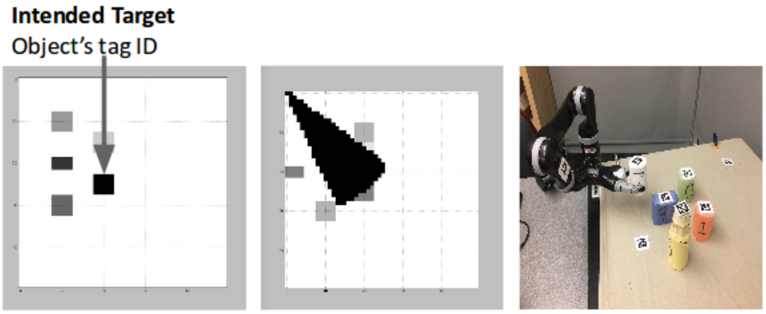


Figure 7.11: Multiple Attention Model

In the final implementation, we are using the first model for the intent prediction. The system is taking the output of the softmax layer in the last 30 frames, and using the sum of the 30 frames to make the prediction. The class with the largest score should be the prediction.

7.3 Planning in Clutter(testing)

In order to test that the planning in clutter was working several tests as shown in Figure 7.12 were conducted. This helped ensure that the planner was working in edge cases and in both the simple and the complex environment.



Step 1:
Make a Footprint map of the objects

Step 2:
Find obstacle-objects in the Target's way; their new locations

Step 3:
Action! Plan to intended object using TSR and CBiRRT

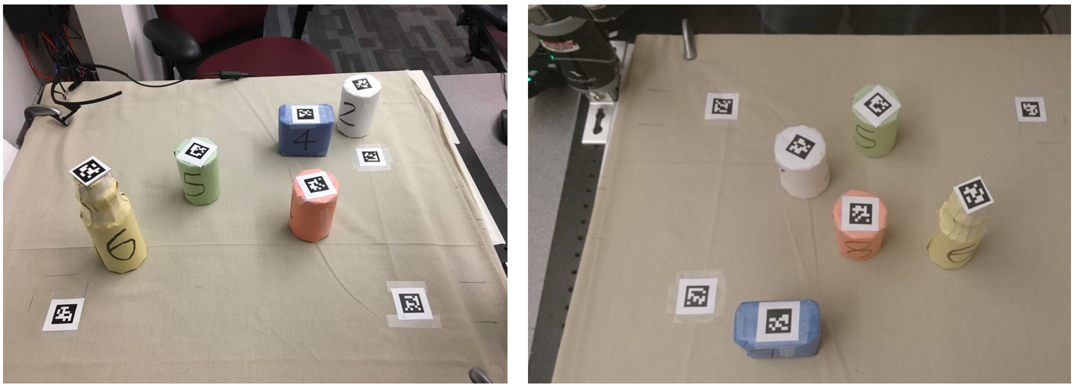
Figure 7.12: Testing for Planning in clutter

7.4 SVE Performance Evaluation

Summarized below are the performance criteria for the SVE:

- **Scenario**

  - **Simple Environment:** Tolerance Half-Angle of the obstacle sector as shown in Figure 7.6 is 10 degrees and there is only one obstacle in this zone. Besides, it shouldn't have the occlusion of objects from the front view. An simple scenario is shown in 7.13(a). In SVE, we will run the simple environment for 5 times.
  - **Complex Environment:** Tolerance Half-Angle of the obstacle sector as shown in Figure 7.6 is 20 degrees and there are three obstacles in this zone. Besides, it shouldn't have the occlusion of objects from the front view. An complex scenario is shown in 7.13(b). In SVE, we will run the complex environment for 3 times.



(a) (b)

Figure 7.13: (a) An example of simple scenario(The orange one is the obstacle and the green one is the target). (b) An example of complex scenario(The orange, green and white are the obstacles and the yellow one is the target).

- **Criteria**
  - The intention prediction should have the success rate of more than 0.7 (5/8).
  - The whole system success rate of the simple scenario should be more than 0.6 (3/5).
  - The whole system success rate of the complex scenario should be more than 0.45 (1/3).
  - The total time for the manipulator to bring the object should be less than 3 minutes in the simple scenario.
  - The total time for the manipulator to bring the object should be less than 5 minutes in the complex scenario.

Figure 7.14 shows a graphical representation of the performance statistics, while Table 7.1 shows the numbers associated with the performance statistics of Intent Bot.

In SVE, the intention prediction part achieved the success rate of 5 out of 6. The failure case was the confusion of white and yellow, which are really similar. We

achieved whole system success 4 out of 5 and 2 out of 2 in simple and complex scenario separately. The failure case was because of some unknown error in OpenRave. All the 6 cases finished within the time limit.
In SVE encore, the intention prediction had 100% accuracy. The whole system held the success rate of 3 out of 5 and 1 out of 3 for the simple and complex case, due to some error of controller and OpenRave and the accident problem of Kinect. All the 8 cases finished within the time limit.

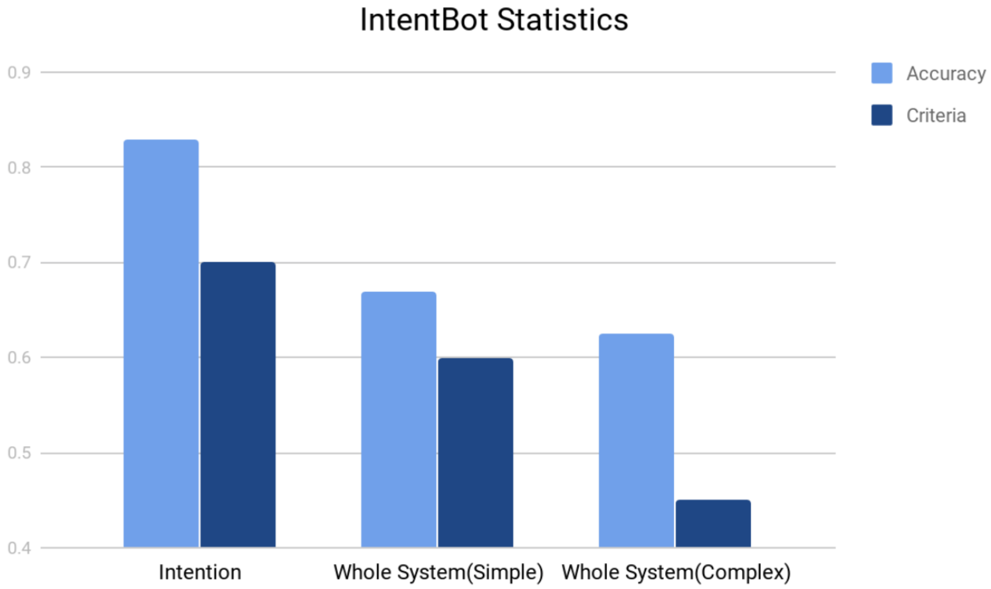


Figure 7.14: SVE Performance Statistics

Test	Statistics	Criteria
Intention Prediction	82%	70%
Whole System (Simple Scenario)	67%	60%
Whole System (Complex Scenario)	62%	45%

Table 7.1: SVE Performance Statistics

7.5 Strong and Weak Points

7.5.1 Strong Points

1. **Real-time prediction of intention subsystem.** - For the intention subsystem, we used a neural network to predict the target object in real-time. With share-autonomy between the robot and the operator, the subsystem can output the target as soon as it gets the intention.

2. **Intuitiveness of the intention subsystem.** - In the system from last semester, the user need to gaze at the object he wants for around 10 seconds, but in our current system, the user can use their gaze more intuitively and only 2 seconds of gaze is enough to show the intention.

3. **Short Planning time of manipulation subsystem.** - Comparing to the system from last semester, we substituted the previous planner with CHOMP to achieve more reasonable planning trajectory and faster planning speed.

4. **Remove obstacles first when planning to the target.** - For some cases, it's hard for the manipulator to plan to the target object as there are some objects in the way. To find the valid planning, the manipulator will first remove the obstacles objects and then reach out for the object.

7.5.2 Weak Points

1. **Intention prediction is not based on the gaze sequence pattern.** - We first tried to decide the user's intention based on the gaze sequence pattern, but the sequential model didn't work really well, thus our current system is only based on the gaze position for each frame.

2. **The intention subsystem is not fully automatic.** - For the current system, it needs the human operator to press a key to decide when to stop the analysis of the intention. It can be improved by using user's voice as a stop signal to make the system automatic.

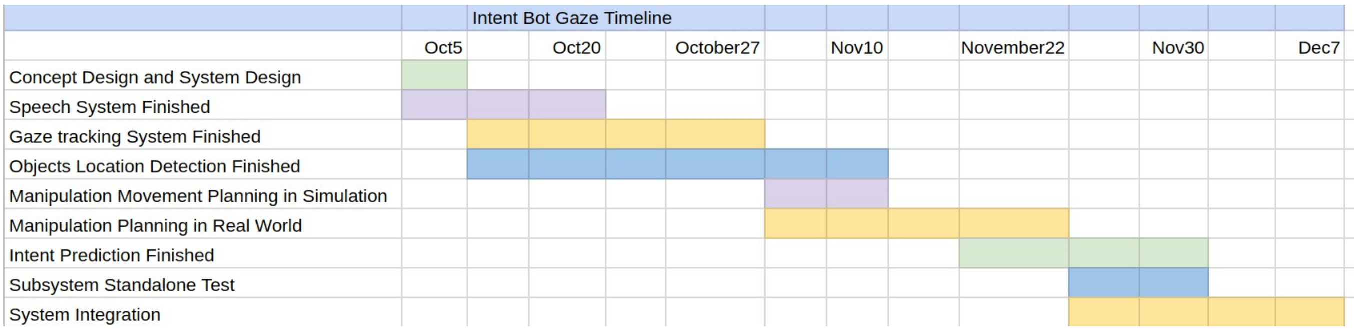
3. **Errors in the controller from time to time.** - Sometimes, the manipulator subsystem might face the problem of the controller, which lead to some failure cases of opening fingers after reaching the objects.

4. **Some unknown errors in OpenRave.** - The OpenRave package is used in manipulator subsystem and it will pop some unknown errors from time to time to make the manipulator subsystem got stuck.

8 Project Management

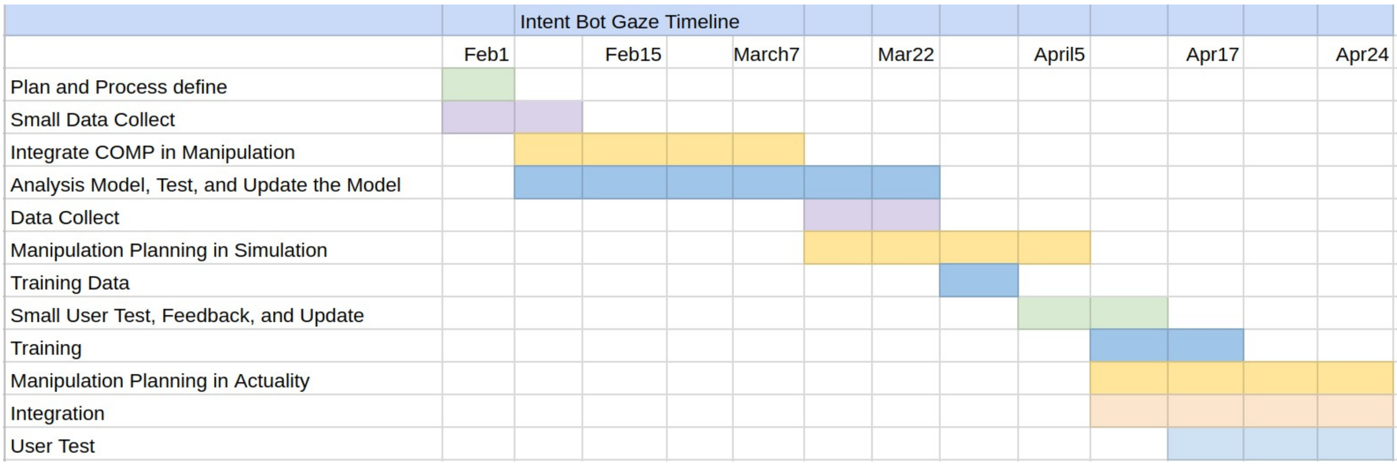
8.1 Schedule

Given below is our schedule for the whole project. In the first semester, we mainly worked on the implementation of the basic function of the system. We used a native way to implement each of the functional requirement. In the second semester, we mainly worked on the improvement of each subsystem. We made each subsystem more robust by updating the algorithms.



		Intent Bot Gaze Timeline											
	Oct5		Oct20		October27		Nov10		November22		Nov30		Dec7
Concept Design and System Design	🟩												
Speech System Finished	🟪	🟪	🟪										
Gaze tracking System Finished		🟨	🟨	🟨	🟨								
Objects Location Detection Finished		🟦	🟦	🟦	🟦	🟦	🟦						
Manipulation Movement Planning in Simulation						🟪	🟪						
Manipulation Planning in Real World						🟨	🟨	🟨	🟨				
Intent Prediction Finished									🟩	🟩	🟩		
Subsystem Standalone Test										🟦	🟦		
System Integration										🟨	🟨	🟨	🟨

Figure 8.1: Gantt for Fall Semester



		Intent Bot Gaze Timeline											
	Feb1		Feb15		March7		Mar22		April5		Apr17		Apr24
Plan and Process define	🟩												
Small Data Collect	🟪	🟪											
Integrate COMP in Manipulation		🟨	🟨	🟨	🟨								
Analysis Model, Test, and Update the Model		🟦	🟦	🟦	🟦	🟦	🟦						
Data Collect						🟪	🟪						
Manipulation Planning in Simulation						🟨	🟨	🟨	🟨				
Training Data								🟦					
Small User Test, Feedback, and Update									🟩	🟩			
Training										🟦	🟦		
Manipulation Planning in Actuality										🟨	🟨	🟨	🟨
Integration										🟧	🟧	🟧	🟧
User Test											🟦	🟦	🟦

Figure 8.2: Gantt for Spring Semester

Failures:

- The major failure was that we wrongly estimated the performance of our network. Exploring different models and tried different models tool much longer time than we expected.

Successes:

- It was lucky that we listed the failure of the intent prediction model as a great risk, so we implemented the backup plan at an early time.
- We tried to make each subsystem independent to each other when implementing and testing, so each part could be implemented in parallel, which saved a lot of time.

8.2 Budget Status

Of the estimated budget which was shown in the Conceptual Design Review, we have only spent 370$ as most of the apparatus was provided to us by our sponsor. Hence, we have a total remaining allocated budget of 2150$(2520 - 370) and an additional MRSD budget of 2630$.

Part Name	**Budget ROM ($)**
Mico Kinova Arm	0.00
Custom PC	2603.00
Gaze Tracker	0.00
Kinect	0.00
Microphone Headsets	20.00
AprilTag	0.00
Objects	100.00
Google API Subscription	100.00
Emergency Stop Button	300.00
Total Expense	**2823.00**

Table 8.1: Total Expense of the Intent-Bot System

Part Name	**Expense ($)**
RAM	335
SSD	238
GeForce	1000
Motherboard	382
Power Supply	119
CPU	449
Misc(tower case, connectors, contacts)	80
Total Expense	**2603.00**

Table 8.2: Breakdown of Custom PC

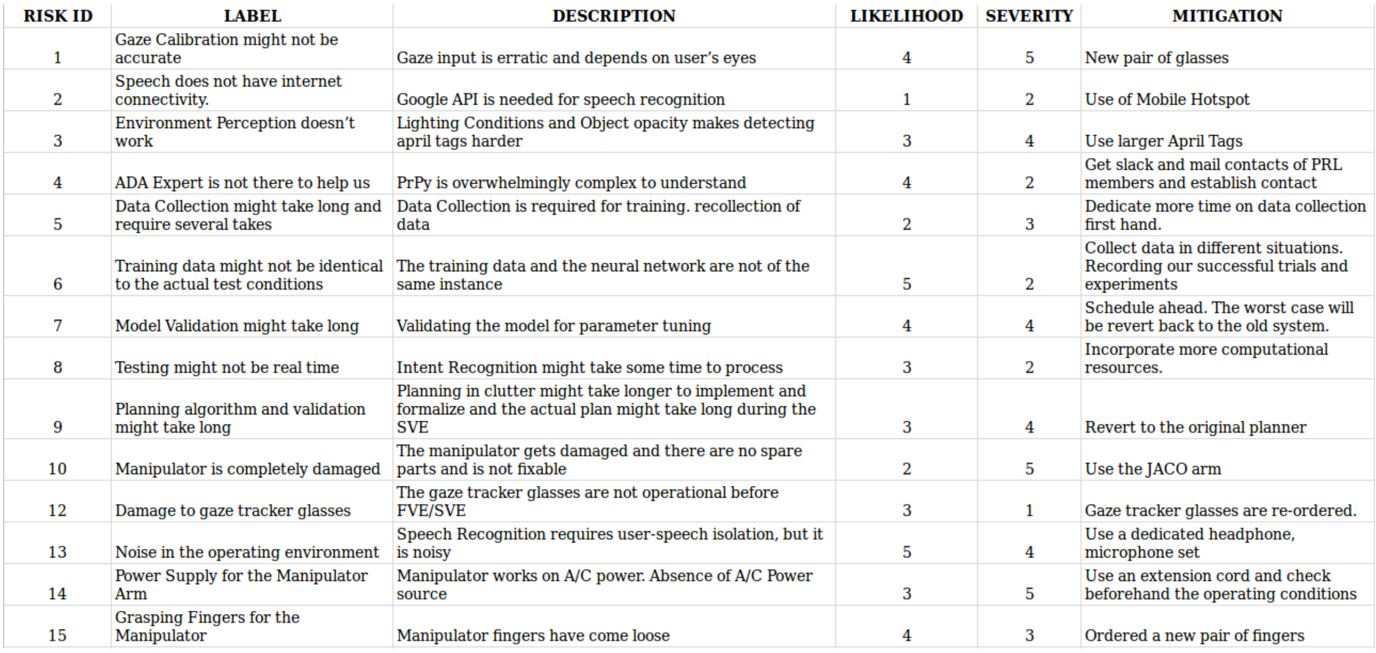
Please refer to the link **here** to get details of the parts used.

- Successes:
  - Our expenditures can be re-used by future students.
  - We did not overspend and were able to get the items that we needed in time for our project.
- Failures:
  - We should have foreseen what we would need and should have bought the GPU in advance so that we would haven't had to struggle at the very last moment.

8.3 Risk Management

Figure 8.3 was the most recent Risk Management Table. We think we did a decent amount of forecasting and post-mishap risk management. Although our anticipation of risks has been fairly straightforward there are some risks which are difficult to forecast.

- Successes:
  - Risk ID 5, 6, and 7: We anticipated these risks and turns out we had to ultimately switch to a simple backup plan for making the gaze intuitive.
  - Risk ID 1: During our final run in the SVE Encore, the Kinect had crashed and we had to reboot the entire computer. We were worried that we would have to calibrate I-Chen again and that it might take us more time as this had once happened during the Fall. However, when we reboot the system, the gaze was still accurate and that was all because of the new pair of glasses which retain calibration unless they have been taken off(which, gladly, was not the case).
  - Risk ID 15: Just a day before the SVE Encore, we came to know that the finger had been damaged. Apparently, the springs of the finger had come out. Since we had faced a similar issue during the Fall semester, and that we had ordered an extra pair of fingers we were prepared for this. Although the finger got fixed the night before the Encore, had it not been, we would have simply replaced it.
  - Risk ID 9: Planning was crucial to the project and we wanted to make sure that the plan times were low, and the manipulator faced as fewer errors as possible. We overcame this issue by putting it in the risk and this lead to more unit tests to ensure that planning was almost always successful.
- Failures:
  - Kinect Crashed During the Encore: We had never encountered this problem before and thought that the Kinect might never fail. This was a grave mistake that we made. We could have anticipated this by taking a more functional perspective than a hardware perspective for Risk Management.



RISK ID	LABEL	DESCRIPTION	LIKELIHOOD	SEVERITY	MITIGATION
1	Gaze Calibration might not be accurate	Gaze input is erratic and depends on user's eyes	4	5	New pair of glasses
2	Speech does not have internet connectivity.	Google API is needed for speech recognition	1	2	Use of Mobile Hotspot
3	Environment Perception doesn't work	Lighting Conditions and Object opacity makes detecting april tags harder	3	4	Use larger April Tags
4	ADA Expert is not there to help us	PrPy is overwhelmingly complex to understand	4	2	Get slack and mail contacts of PRL members and establish contact
5	Data Collection might take long and require several takes	Data Collection is required for training. recollection of data	2	3	Dedicate more time on data collection first hand.
6	Training data might not be identical to the actual test conditions	The training data and the neural network are not of the same instance	5	2	Collect data in different situations. Recording our successful trials and experiments
7	Model Validation might take long	Validating the model for parameter tuning	4	4	Schedule ahead. The worst case will be revert back to the old system.
8	Testing might not be real time	Intent Recognition might take some time to process	3	2	Incorporate more computational resources.
9	Planning algorithm and validation might take long	Planning in clutter might take longer to implement and formalize and the actual plan might take long during the SVE	3	4	Revert to the original planner
10	Manipulator is completely damaged	The manipulator gets damaged and there are no spare parts and is not fixable	2	5	Use the JACO arm
12	Damage to gaze tracker glasses	The gaze tracker glasses are not operational before FVE/SVE	3	1	Gaze tracker glasses are re-ordered.
13	Noise in the operating environment	Speech Recognition requires user-speech isolation, but it is noisy	5	4	Use a dedicated headphone, microphone set
14	Power Supply for the Manipulator Arm	Manipulator works on A/C power. Absence of A/C Power source	3	5	Use an extension cord and check beforehand the operating conditions
15	Grasping Fingers for the Manipulator	Manipulator fingers have come loose	4	3	Ordered a new pair of fingers

Figure 8.3: Risk Management

- Unknown OpenRAVE Error: During the night before our SVE and SVE Encore, one of the failure cases was that the objects were not even added to the scene. This was an OpenRAVE Error while adding kinbodies to the environment. We should have put OpenRAVE in the Risks and could have tested more to ensure that this did not happen again.
- Controller Failures and Fingers Don't close: Despite the occurrence of this failure, the team did not get enough time to modify the controller of the manipulator and make changes to it. Since the fingers closed most of the time, we did not include this in our Risks. We were mistaken when, once again, the fingers did not close during one of our simple cases in the SVE.

9 Conclusions

9.0.1 Key lessons learned

The following are the key lessons learned:

- **The naive combination of speech and gaze information is more accurate, but not intuitive**
  During the testing of the intent recognition subsystem, we realized that the system requires the users to fixate their gaze on the item they intend the robot to grasp for their intention to be registered by the system. However, such requirement is very disruptive to the users. During the interaction, the users have to stop whatever they are doing and focus their gaze on the item. Drawing from this, we were successfully able to incorporate the gaze of the user as a singular modality to convey intent. However, we also noticed that this was slightly inaccurate as compared to the combination of speech and gaze. Nonetheless, it was cognitively less taxing than our previous system.

- **Importance of simulation**
  During the Spring Semester, we observed that the research work increased dramatically which caused us to not work on the robot and respect the time of the senior lab-members. Hence, we decided to test in the simulation and only once the planning in the manipulator had succeeded in simulation that we could use it on the real robot. This made code development faster and more efficient.

- **The importance of unit testing and constant integration**
  At the end of the FVE, all the team members decided to install the ADA repository of the manipulator on their system in order to conduct user-studies. This removed the reliability of the manipulator on a single user. Although the manipulator, later, did not become a part of the user-study this was still useful in case we wanted to.

- **The importance of modularised code**
  Most of the team members have little or no experience in writing lengthy, but maintainable pieces of code. However, this semester, we started modularising the code by maintaining the same functionality of the code. Most of the scripts were now broken down into functions or encapsulated into classes. This made the code more retractable and it was easier to find bugs.

- **Integration can be seamless**
  We observed that the integration can be seamless if the Cyber-Physical Architecture is solid and is modularised. We had a ROS Structure in place since the beginning of the last semester which ensured that even though the subsystems were being developed independently we could easily integrate them at the last moment.

9.0.2 Future Work

- **Bringing objects to the user maintaining an orientation**
  One cannot simply expect that while a soup-can which is being brought to the user should spill on the table. Given the current system, the manipulator generates a path which can cause the object to turnover. Thus, future work in order to constrain the orientation of the grasped object will be required.

- **Studying gaze and other modalities for intention**
  A better study in order to understand the various modalities which can convey the user's intent in an intuitive manner should be conducted. Although we have limited ourselves to gaze and speech, capturing the head motion of the user or other forms of modalities can turn out to be more expressive and easier to grasp.

- **Conducting focus-group studies on people with disabilities and upper-body dysfunctions**
  One of our team-members had suggested that we take a Product Designing and Development approach in order to identify the needs of the user. However, due to time constraints, this did not come through. However, if such a research is conducted beforehand it will be really useful for the developers in the future in order to make a more robust system that meets the requirements of the end-user.

- **Incorporating the head's pose-detection**
  While our system was intended to feed the user, it is not able to do so. The major challenge lies in finding the pose of the user's head, specifically the mouth's, with respect to the end-effector of the robot. This feature will be a nice addition to the project.

References

[1] Time-of-Flight and Kinect Imaging, http://campar.in.tum.de/twiki/pub/Chair/TeachingSs11Kinect/2011-DSensors_LabCourse_Kinect.pdf

[2] Time-of-flight camera, https://en.wikipedia.org/wiki/Time-of-flight_camera

[3] Americans with Disabilities: 2010, https://www.census.gov/newsroom/cspan/disability/20120726_cspan_disability_slides.pdf

[4] Assistive Robotic Manipulation through Shared Autonomy and a Body-Machine Interface, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4737957/

[5] Shared Autonomy via Hindsight Optimization for Teleoperation and Teaming, SharedAutonomyviaHindsightOptimizationforTeleoperationandTeaming

[6] Predicting User Intent Through Eye Gaze for Shared Autonomy, http://hennyadmoni.com/documents/admoni2016aaaifs.pdf

[7] depth camera, http://www.cs.cmu.edu/afs/cs/academic/class/15869-f11/www/lectures/19_depthcamera.pdf

[8] Admoni, H. (2016). Nonverbal communication in socially assistive human-robot interaction. AI Matters, 2(4), 9-10.

[9] uArm-Swift-Specifications-en, http://download.ufactory.cc/docs/en/uArm-Swift-Specifications-en.pdf

[10] Kinova-Specs-MICO2-6DOF-Web-170512-1, http://www.kinovarobotics.com/wp-content/uploads/2015/02/Kinova-Specs-MICO2-6DOF-Web-170512-1.pdf

[11] Robotis Manipulator, http://www.robotis.us/robotis-manipulator-h/

[12] Mover6 6DOF http://www.robotshop.com/en/mover6-6dof-robot-arm.html