Shelf Buddy

Authors: Ludi Cao, Esther Jang, Bhumika Kapur: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*— ShelfBuddy is an assistive robot that helps users grab objects from high shelves. When going grocery shopping, reaching objects on a shelf can be a very difficult task, especially for those with disabilities. ShelfBuddy makes grocery shopping an easier and a more accessible process by allowing users to focus on just looking for their items. A user can point to an object on a shelf using a laser pointer, and ShelfBuddy will be able to grab the object from the shelf and place it in the user's basket.**

***Index Terms*—AprilTags, Autonomous Navigation, Computer Vision, Control, Laser Point Detection, Linear Slides, Robotics**

1 INTRODUCTION

Grocery shopping is intended to provide people with the option of selecting their own goods and thus be a seamless and enjoyable experience. Unfortunately, it is still a difficult task for people with disabilities as many objects are located on higher shelves that are out of reach. Current solutions are also inconvenient or require other people's assistance.

ShelfBuddy is an autonomous robot system that will retrieve items on higher grocery shelves for users, ensuring that grocery shopping is a personalized experience for everyone. The user first selects items by pointing a laser on the object they want. Then, after detecting the laser point, ShelfBuddy will navigate to the object on the shelf, grab the object, and deliver the item to the user's basket. This solution allows users in need to have more control over their grocery shopping experience and improves their accessibility.

Our current solution applies to shopping medical containers typically found at a pharmacy. We identify that shoppers in this environment are more likely to require assistance. In addition, the testing inputs would be more controlled for us to test. The requirements for our solution were chosen with the intention of making ShelfBuddy a user-friendly system. Our primary requirements are the following:

1. The overall success rate of our system, which is the entire process of detecting the item being pointed at, retrieval of the object, and navigation to and from the shelf and basket, should be 97.5%

2. Requirement 1 depends on the success rate of three subsystems. The first requires the success rate of the navigation to and from the shelf and basket. We would like this value to be 98.5%.

3. The second subsystem in which we would like to measure its success rate is the detection of laser pointer. We would like the robot to correctly detect an object pointed with a laser with a success rate of 99%

4. The third subsystem to measure is the process of retrieving the item on the shelf, and holding on to the item when sending to the basket. We would like the success rate to be 99.5%

5. The robot should travel at a speed of 0.5 m/s.

6. The latency for grabbing an object when at the shelf should be 3 seconds.

7. The latency of processing a snapshot of the shelf for the laser point is 1 second.

8. The distance between items on the shelf is roughly 2 inches.

9. The dimension of the shelf is 3ft $\times$ 4ft $\times$ 1ft

10. The robot will retrieve items of sized around 3 inches and weighing a maximum of 1 pound.

11. The robot should be user-friendly in terms of space occupancy within the grocery store and ease of use.

2 DESIGN REQUIREMENTS

Requirement 1: The average number of items a customer would purchase per visit at a popular pharmacy such as CVS is 4 [1]. Our robot would want to hit the benchmark of successfully meeting the user's shopping needs 9 times out of 10 visits. We believe this would meet user satisfaction levels. On average 10 visits at the store would require 40 runs, so at lest 39 of them should be successful. This corresponds to a success rate of $\frac{39}{40} = 97.5\%$. To test this requirement, we plan to run 40 complete trials, where after executing the program the robot first follows the user, scans through the shelf to find the pointed object, successfully retrieves the object and places it to the grocery basket. We expect the entire process to succeed 39 times.

Requirement 2: Based on the overall success rate from *Requirement 1*, we determined that the success rate of the navigation process is 98.5%. This success rate is slightly lower than the expected success rate based on the overall success rate. We determined that navigation would have a poorer success rate because of the fact that compared to the other subsystems, navigation involves more moving components and has to deal with the complexity behind

traversing across an area.

Requirement 3: Based on the overall success rate from *Requirement 1*, the success rate for laser detection is 99%. This is the anticipated success rate of the subsystem based on the overall rate, and we decided that this was reasonable given that laser detection is not as complex as navigation but also not straightforward enough to have a lower error rate.

Requirement 4: Based on the overall success rate from *Requirement 1*, the success rate for object retrieval is 99.5%. This is the highest accuracy among the three sub systems which contribute to the overall success rate. We believe that given successful navigation and item detection, the robot would be positioned in a straight angle that should be easy to grab the object, so there is few room left where error could happen.

Requirement 5: To provide user-friendly assistance, ShelfBuddy should travel at a speed that is close to the average speed a user can travel in a wheelchair, which is 0.79m/s. [2] This would ensure that using ShelfBuddy would take roughly the same amount of time as the user travels.

Requirement 6: The latency of grabbing an object is 3 seconds because while it is slightly slower than the average latency of the time taken for a human to grab an object, it is still not unreasonably long for the user's experience.

Requirement 7: The latency of processing a laser pointer on an object will be 1 second. Research shows that the average time a human takes to process visual stimuli is .25 seconds [3]. Thus, a one second laser processing time is within reason in comparison to the amount of time it would take a person to detect the laser.

Requirement 8: Typically, two items on a shelf are on average tightly packed on a grocery store. However, we would be operating under the assumption that the gap is 2 inches in order to feasibly achieve our grabbing requirements. This is still as minimum of an increase from the real world scenario.

Requirement 9: Our shelf mirrors a typical shelf found in a pharmacy store in most dimensions. However, because the actual height would unfortunately introduce too much complexity for our project to manage in the short time frame, we chose to limit our height to 3 feet. Our project is hence more of a proof of concept for this idea that would be blown into larger proportions in the real world.

Requirement 10: Our use case of the robot would be to retrieve medical containers typically seen at a CVS pharmacy. The items are around 3 inches and weigh at most 1 pound.

Requirement 11: Our use case is within a grocery store with other shoppers and carts that the robot will have to share narrow aisle space with. The robot should hence cause minimal interference in order to succeed in our use case. It should also be simple to use since its intention is to improve a shopper's experience and hence not cause unnecessary inconvenience.

3 ARCHITECTURE OVERVIEW

The CAD model (Figure 9, page 12) and informational flow chart of our system (Figure 10, page 13) are shown in the appendix. Our system is roughly split into three sub systems: the drive train and navigation algorithm, detection and item recognition, as well as the linear slide and retrieval system. We would use the Jetson Xavier as the main computing board, and the Arduino as the controller of the motor. Computer vision algorithms would be run on the Jetson Xavier, and the information would be passed to the motor drivers through the Arduino connected with a USB cable.

3.1 Wheel Chassis Design

The mechanical design for the wheel chassis is described in Figure 1.

The wheel chassis is driven by four omni-directional wheels placed at diagonal directions. These omni-directional wheels come in packs of two which give the robot much more stability and less drift. Each pair of wheels is controlled by its own DC motor, where it's mounted to aluminum extrusions that form the base of our system. The chassis base is an octagonal shape such that the four sides at the corners connecting to the wheels are shorter than the four sides on the side. On top of the chassis we plan to laser cut a board that mounts to the extrusions where electronic components such as the Jetson Xavier, the Aruduinos, the motor drivers, the power, etc, can fit compactly onto.

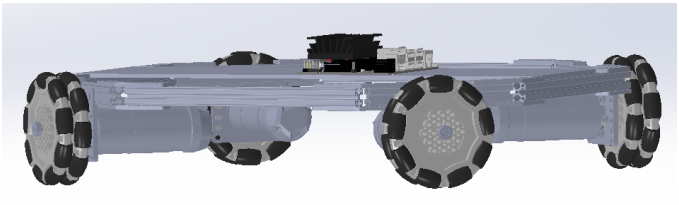

Figure 1: Wheel Chassis

3.2 Computer Vision

We will use an Intel RealSense D435 Depth Camera, which has high resolution and depth sensing technology. The camera will allow our robot to travel throughout the drugstore, follow the user, and navigate to the shelf to retrieve the object.

3.2.1 Navigation:

We will use AprilTags, a visual fiducial system Figure 2, for navigation. The tags will be printed on normal paper and will be attached to our shelf and the user's basket. Using the AprilTag detection software, our robot will follow the user's basket throughout the store. The AprilTag detection software outputs the tag information and its precise 3D location in relation to the camera Figure 3. This information will allow our robot to navigate accurately and efficiently.

3.2.2 Laser Recognition:

When the user indicates that they are pointing to an object, the robot will scan the closest shelf for the laser pointer. The robot will detect the laser using the laser pointer algorithm, which will output the location of the laser point in an image, if there is one, and travel towards the shelf to that object.

3.2.3 Item Detection:

After the object that the laser pointer is pointing at has been detected, the robot will need to retrieve the item. This will be done using the edge detection algorithm, which will output the location of the edges of the item relative to the camera.

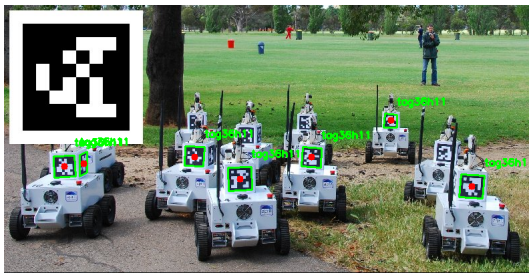

Figure 2: AprilTag example.

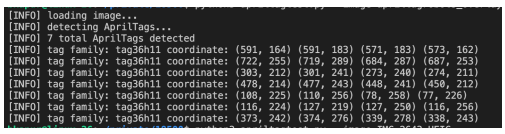

Figure 3: Detection algorithm example output.

3.3 Retrieval System Design

3.3.1 Claw Gripper End Effector

We will be using a PWM servo-powered claw from Amazon for our end effector. The servo would be connected to a PCA9685 PWM servo driver which connects to the Jetson Xavier using I2C pin connections.

3.3.2 Z-Axis Linear Slide System

We will be using a motorized pulley-powered linear slide system for the z-axis linear slide system. We would have 2 mirrored, parallel linear slide systems mounted on both sides of the chassis and powered by 2 DC motors (1 each respectively). The motors would be connected to a motor driver and then to an Arduino.

4 DESIGN TRADE STUDIES

4.1 Wheel Selection Trade-offs

We have a few options of wheel types for our system drive train. To meet *Requirement 2*, the robot should move in different directions with high precision, since navigation involves following the user in one direction, and moving towards the shelf in a potentially perpendicular direction. The robot should also be able to position itself parallel towards the shelf from any angle, since the retrieval process requires high precision from the direction of the wheel base. Therefore, the choice of using normal wheels would be restricting, since the robot would then constantly perform turning which might accumulate high inaccuracy.

We then decide from two popular omni-directional wheel choices: omniwheels vs mecanum wheels. This would require our system to have four DC motors for each wheel. Although this is incovenient, we believe this stills brings greater advantage than using normal wheels connected to 2 DC motors, as it saves us from putting significant cost in directional control. Both wheels are configured and can be programmed in a way that travels in eight directions. The main advantage of mecanum wheels is that they have greater traction to the floor, which helps in terrains where the road conditions are less than ideal. However, this is not useful to our use case environment, and its cons of being more costly, heavier, and less accurate in directional control are far more problematic.

As for wheel size, there are 60 mm diameter choices and 90 mm diameter options that the vendor provides. Smaller wheel sizes provide more torque and large wheel sizes provide more speed. Since the robot is travelling on a smooth (low friction constant) and flat (no vector from weight acting against) floor, we anticipate that the torque inserted on the wheels would not be a main concern. However, to satisfy *Requirement 5*, we would want to optimize the speed of the robot if we can. Hence, we decide to use the 90 mm diameter wheels. As later verified in section 4.2, this decision would work well with our choice of motors and would provide the speed that would meet the requirement.

4.2 Motor Selection Trade-offs

We would select the appropriate motor to satisfy *Requirement 5*. We refer to equations

$$w = 2\pi f \tag{1}$$

$$v = rw \tag{2}$$

Combining equations (1) and (2), we derive

$$\begin{aligned} v &= \frac{D}{2} \cdot 2\pi f \\ &= \pi D f \\ &= \pi \cdot 0.09m \cdot \frac{RPM}{60s} \end{aligned}$$

The weight of our system would include sub components as follows: (Length of extrusions is justified in section 4.3, where we derive the weight)

Item	Weight(kg)
Wheel	$0.07 \times 8 = 0.56$
Extrusion	9m $\rightarrow$ 2
Electrical components	1.5
Claw and object weight	1.5

Total: 5.56 kg + 6 $\times$ motor weight

Since our system incurs much weight, we would estimate a 25% reduction from max speed of motor. Hence, the speed of our robot should be

$$\begin{aligned} &\pi \cdot 0.09m \cdot \frac{RPM}{60s} > \frac{0.5m/s}{0.75} \\ &\Rightarrow RPM > 141.47\text{rpm} \end{aligned}$$

Hence, the robot would require an RPM of at least 141.47 rpm, so we couldn't use another common standard of 100 rpm motors. We also want to limit the weight of the motors, as higher weight would cause greater reduction in max speed. If the gain in rpm is small, then the overall speed requirement might not be met.

The vendor that we order from provides a couple of DC motor options, including a planetary gear motor with a 40:1 gear ratio, a planetary gear motor with a 20:1 gear ration, a spur gear motor with a 40:1 gear ratio, as well as a spur gear motor with a 20:1 gear ratio. The 40:1 gear ratio motors provide 150 rpm at stall torque of 4.2 Nm, and the 20:1 gear ratio motors provide 300 rpm at stall torque of 2.1 Nm. Although an rpm value of 150 rpm would theoretically meet the speed requirements according to calculations, the requirements would barely be met under the ideal case. We are also using the maximum specs to compare, and maximum specs might not be achieved during experiments. The cost of using a 20:1 gear ratio motor is that its stall torque is reduced by half. However, given that each motor weighs at most 0.5 kg, the total weight would be at most 8 kg. With the assumption that the robot travels on a flat and friction-less surface, the robot would require the torque to provide a force such that $F = ma$ to start from the stopped state to the start state. Assume $a = 0.25m/s^2$. The output shaft length is 0.04 m, so the force that can be applied by one motor at least $2Nm/0.04m = 50N \approx 5kg$. We power 4 DC motors in total, so the max weight of our robot could be 20 kg. Note that most of the numbers we use to estimate the minimum torque load are worse than worst case situations. The conclusion is that the stall torque specs for either motor should work for our system.

Initially our team decided to choose the spur gear motors with 20:1 gear ratios followed from the calculations above. We also chose the spur gear motors over planetary motors because our system isn't complex enough to justify the usage of planetary motors, where the advantage is prominent during high loads. The spur gear motors are also lighter, cheaper, and perform equally if not better in efficiency. Unfortunately, the 20:1 gear ratio motors are out of stock. The only feasible option is using the planetary gear motor with a 20:1 ratio, with a slight increase in 20 dollars for our budget.

4.3 Motor Driver Trade-offs

We noticed the Capstone Inventory had L298N based motor controller, and the model is a popular choice among similar projects. Our motor requires 12 V and has a maximum output power of 15 W. This means that the motor needs at least 15 W / 12 V = 1.25 A. The maximum allowable current of an L298N is 2 A, which is too close to the lower bound that our motor needs. To err on the safe side, we decide to use the BTS7960 DC stepper motor drivers, as the maximum allowable current is 43 A

4.4 Wheel Chassis Trade-offs

We designed the separate mechanical components in SolidWorks to ensure these components are physically compatible with each other. Our initial design is shown in Figure 4. Compared to the current iteration, the main difference for the new design is size. To satisfy *Requirement 9*, I was initially worried that having a base size much smaller than the height of the reaching claw would cause the robot to be unbalanced, especially since the claw would extend horizontally further than the front of the robot. The previous size of the design was roughly 70 cm $\times$ 70 cm, where the 4 longer extrusions were 55 cm long. However, after preliminary calculations, we

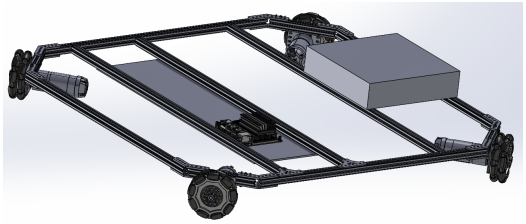

Figure 4: First chassis design

realize the initial concerns are completely redundant: The weight of the motors themselves are 0.45 kg × 6 = 2.7 kg. From *Requirement 1* our item weight is 0.45 kg. Additional components that might reach in front of the robot include an Intel Realsense camera weight around 0.072 kg, and the weight of the claw itself. Assuming the claw extends in front of the robot base for 20 cm, and according to lever theory in physics, we estimate there is sufficient normal force from the ground that would balance the robot and prevent it from "tipping" forward, so we should leave this out of concern.

Therefore, our goal is to make the chassis base as small as possible, since a smaller size means smaller weight and more power efficiency. We would want to mount the linear shaft on the side of the base, which is 1.5 cm × 4 = 6 cm in length. The side would also include the length of the motor and the size of its mount. We also include the margins of connecting brackets, which take up roughly 8cm. Hence, I designed the four longer extrusions to be 20cm in length, which should sufficiently fit these components without wasting too much extra space. To double check, we would also want to fit all the electrical components on the board mounted on top of the extrusions, including a Jetson Xavier Board, two Arduinos, a breadboard, a power supply, motor drivers, and other miscellaneous. From rough calculations a board size of 8 cm and the length across the base should be sufficient. Note that the extrusions at the corners which mount to the wheels are designed to be of length 12 cm, since the diameter of the wheels are 9 cm.

The top down design for the current chassis is shown below in Figure 5

4.5 Battery Selection Trade-offs

The nominal voltage of our motors is 12V. We decide to use rechargeable batteries, which could be of material NiMH or LiPo. We looked at two reasonable models correspondingly, the "Tenergy NiMH Battery Pack 12V 2000mAh High Capacity Rechargeable Battery", as well as the "Ovonic 11.1V 5000mAh 3S 50C-100C LiPo Battery", two of which most suited our needs. Including the charger, the NiMH battery is around 10 dollars more expensive than the other option.

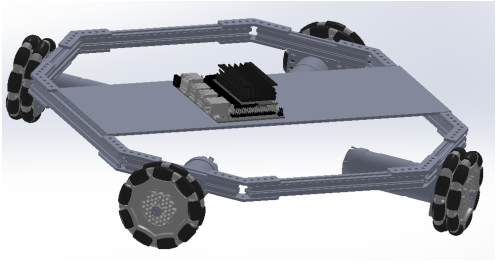

Figure 5: Current chassis design

We first calculate the energy needed for our robot. For power consumption using the NiMH battery, given the maximum rated power of the motors is 15 W, the maximum current each motor draws would be 1.25 A. The wheel chassis which includes 4 motors connected in parallel would draw 5 A current. With the assumption that the motors controlling the linear slide system are not consistently turning, we estimate an average of 6 A current drawn throughout one run. With a 2000 mA h capacity, this would mean that a single charge would only last for 2 / 6 h = 0.333 h ≈ 20 min, which is inconvenient for consistent testing. It also violates *Requirement 11*, as the robot would have to be constantly recharged, and a user-friendly system should not only operate within a short amount of timeframe. In addition, the specs state that a recharging rate of 1 A maximum is recommended. Therefore, the battery needs to charge itself for 2 hours, which worsens the problem of a short duration per charge. We then consider the power consumption associated with the usage of the LiPo battery. Most LiPo rechargeable batteries come at a standard size of 11.1 V. We decide this 1 V difference would not significantly impact performance. Given the maximum rated power of the motors is 15 W, the maximum current each motor draws would be 15 W / 11.1 V = 1.35 A. 6 motors connected in parallel would draw 7.5 A current. The wheel chassis which includes 4 motors connected in parallel would draw 5.4 A current. In a similar fashion, we estimate the average current drawn through a run would be 6.5 A. With a battery of 5000 mA h capacity, a single charge would last for 5 / 6.5 h = 0.769 h ≈ 46 min. A charge rate of 1C is the recommended value for LiPo batteries [4], which corresponds to a current of 5 A. Therefore, it would take one hour to fully charge the battery. This is a significant improvement in timings compared to the NiMH battery. Therefore, we decide to use the 11.1 V LiPo batteries as our power source.

4.6 End Effector Trade-offs

We wanted to select an end effector that would be able accessible for us in terms of money and availability while still emulating modern solutions. At first, we explored using a vacuum suction gripper end effector as this was the most common solution being used in current industry for robots similar to our intended use case. Using a vacuum suction gripper would have also given us the ability to have a larger margin of error for getting bounds of an object and gripping items of various shapes and sizes. Unfortunately, such vacuum suction gripper parts were unavailable to us as they were either too costly for our budget or would not have shipped in our tight time frame. Hence, we pivoted our end effector design choice to using a claw gripper, as this was the second most popular end effector option we found in our research.

When researching claw gripper options, the most popular choice that was accessible for us and would meet our use case was a servo-powered claw. We found that we could either purchase an off-the-shelf part for out of the box use,

or we could design and custom cut or print a claw. Based off our research, custom designing our own claw had the advantage of potentially improved gripping strength and the ability to grab larger objects. However, we determined that the time cost of this approach was significantly larger than its added value as there was not much existing information regarding it to work off of. As a result, we opted for purchasing an off-the-shelf servo claw part that would meet our requirements. Since choosing to purchase the part returned an extra week of time that would have been spent building and testing the claw part, we decided to allocate this time towards integration. This is a highly beneficial investment of time because our project has so many subsystems that are required to integrate completely for the system to work and will thus be one of the most complicated, time-consuming, and failure-prone parts of our project.

To satisfy our *Requirement 10*, we had to find a claw gripper that has a maximum gripping width of at least 3 inches. We also needed a servo that would power the claw to provide a gripping force and have a coefficient of friction with the object as defined in the following:

$$\begin{aligned} F_{grip}\mu &\geq (0.4536\text{kg})(9.8\frac{\text{m}}{\text{s}^2}) \\ &\geq 4.4463\text{N} \end{aligned}$$

To find the part that would satisfy these requirements, we considered many vendors. We first looked at Robotshop.com as there were many claw gripper options from various brand-name robot part vendors. However, we had a lot of difficulty finding a part that could satisfy our width requirement and had a promising design. Hence, we looked to Amazon.com instead as even though the vendors may have been slightly more off-brand, we knew that the part would arrive with faster shipping time and allow us to spend more time testing to compensate for this concern.

We found 2 servo claws that seemed to meet our *Requirement 10*. Both were from the same vendor, and one had a larger linear gripping block (claw A) while the other had a more rounded claw design (claw B). Based on its specifications, claw A has a link radius of 43mm. This means that for claw A to grab an object of width 3 inches, a single claw side must open $\theta = 62.38°$. Similarly, based on its approximate specifications, claw B has a link radius of 100mm. This means that for claw B to grab an object of width 3 inches, a single claw side must open $\theta = 23.92°$.

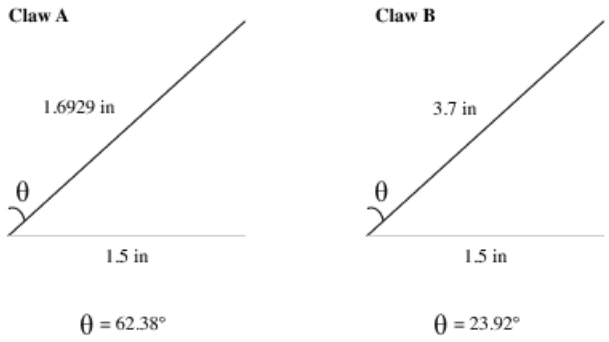

Figure 6: Claw Angle Calculation

The claw is made out of an aluminum alloy and assuming our products are made out of a paper-based material, we can use the static coefficient of friction between aluminum and cardboard of 0.43 [5]. This means that we can find approximately find the torque, τ_{servo}, of the servo needed using following:

$$\begin{aligned} F_{grip}sin(\theta)r &= \tau_{servo} \\ F_{grip} &= \frac{\tau_{servo}}{sin(\theta)r} \\ \frac{\tau_{servo}}{sin(\theta)r} &\geq \frac{4.4463\text{N}}{\mu} \\ \tau_{servo} &\geq \frac{4.4463\text{N}sin(\theta)r}{0.43} \\ \tau_{servo} &\geq 10.34\text{N}sin(\theta)r \end{aligned}$$

This means that claw A will require a servo with a torque of at least $10.34\text{N}sin(62.38°)(0.043\text{m}) = 0.394\text{Nm}$ Similarly, this means that claw B will require a servo with a torque of at least $10.34\text{N}sin(23.92°)(0.1\text{m}) = 0.419\text{Nm}$ The claw kits are from the same vendor and come with a PWM servo of torque 1.96Nm which is more than sufficient for meeting our requirements. We currently plan on purchasing both claw options and manually testing each claw to determine which shape is best suited for gripping objects in our use case, given that both otherwise satisfy our end effector requirements.

4.7 Linear Z-Axis Actuating System Trade-offs

For our linear z-axis actuating system, we made several alternative design considerations with a focus on meeting the height aspect of *Requirement 9*. The industry standard choice would have been to use a link arm robot, but we decided that this would introduce unnecessary complexity to our project since we only need actuate on the z-axis and would not use the extra degree of freedoms offered. Another alternative was to use a linear guide rail system, but we did

not consider this choice as it would have occupied too much vertical space. This would have caused our robot to pose as a larger interference to others, especially when not in use, and hence did not meet our *Requirement 11*. Hence, we decided to use a cascading linear sliding system and ultimately ended up deciding between a pulley powered linear slide system or a cascading rack and pinion system. We ended up selecting the pulley powered linear slide system as we wanted to reach a large height which would require several sets of rack gears and would introduce unnecessary complexity and cost.

The total mass of the lifted system is $m_{extrusions} + m_{slideparts} + m_{camera} + m_{claw_system} = 0.44\text{kg} + 0.1\text{kg} + 0.3\text{kg} + 0.2\text{kg} = 1.04\text{kg}$. This means that the lift has to be able to handle a force of $(1.04\text{kg})(9.8\frac{\text{m}}{\text{s}^2}) = 10.19\text{N}$. We also wanted to have 2 parallel sliders to keep the system balanced. Hence, each motor and pulley pair that we chose to control the system had to be able to handle at least 5.095N of force. The motor we chose had $4.2Nm$ of stall torque and we also opted for a pulley with smaller radius of 60mm, which results in a force of $\frac{4.2\text{Nm}}{0.06\text{m}} = 70N$ which is more than enough force.

4.8 Computer Trade-offs:

When choosing the computer for our project we considered two main options, the Jetson Xavier and the Raspberry Pi. Both computers were in the ECE inventory, so price was not a concern. Some members of our group were familiar with the Raspberry Pi, while none of us had experience with the Xavier NX. In terms of specifications, the Raspberry Pi has a BroadCom VideoCore VI GPU, 2-4GB of RAM, a 40 GPIO pin header and a variety of ports. [6] In contrast, the Xavier NX has a 384-core NVIDIA Volta™ GPU, a 7-way visual processor, and 8GB of memory.[7] We chose to use the Jetson Xavier due to the extra memory, and the fact that all our research concluded that the Jetson Xaiver was superior when it came to AI applications. Since we plan to use Computer Vision in our project, we wanted to select a computer which would best support this.

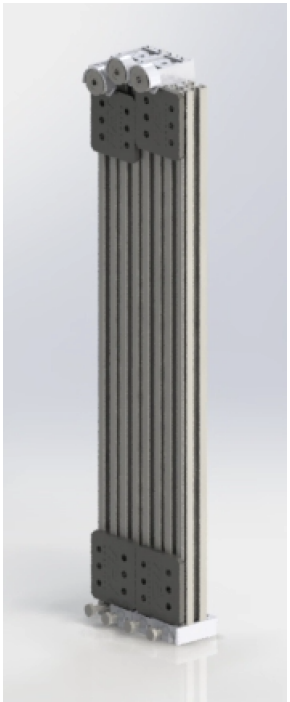

Figure 7: CAD of one side of linear slide part

4.9 Camera Trade-offs:

We mainly considered two cameras for our project, an Intel Realsense or the Arducam UC-698 along with a depth sensor. Both cameras were available to us at no cost, and the distance sensor would also be inexpensive. The Arducam has a 75 degree FOV, 4k@30FPS frame rate. In comparison the Intel Realsense 435 has an 85 degree depth FOV and 90 FPS depth frame rate, 69 degree RGB FOV and 30 FPS RGB frame rate. The FOV and frame rate of both cameras were similar, but the main difference between the two cameras was the depth sending capability of the RealSense. Given that our robot must approach the shelf and move the claw horizontally to retrieve the object, we wanted to select the camera that would provide us with the most accurate depth information. Thus, we chose the Intel Realsense for our system.

5 SYSTEM DESCRIPTION

5.1 Hardware Connections

As explained in the design trade studies, we decide to use the BTS 7960 DC stepper motor drivers due to the current requirements from our motors. One disadvantage that comes along with this choice is that only one motor can be connected to this motor driver. Along with the two motors from the actuating system, we would require 6 motors in total, which require 6 motor drivers. Each motor driver consists of 2 PWM pins for controlling motor speed and 2 output signal pins for controlling motor direction that we would want to connect to the Arduino. One Arduino has 6 PWM pins and 8 other digital pins. Therefore, one Arduino can connect to 3 of the motor drivers, and the visual layout is shown in Figure _. Note the green coloring corresponds to the PWM pins and the orange coloring corresponds to the EN-R and EN-L pins which control motor direction. We would need 2 Arduinos to connect all the motors.

Communincation between the two Arduinos and the Jetson Xavier would be through USB cables. A USB C would be connected from the Jetson Xavier to the Intel Realsense. The Jetson Xavier has 4 USB ports, which suffices our needs.

The PWM servo for the claw will be connected to a servo controller which connects to the Jetson Xavier with

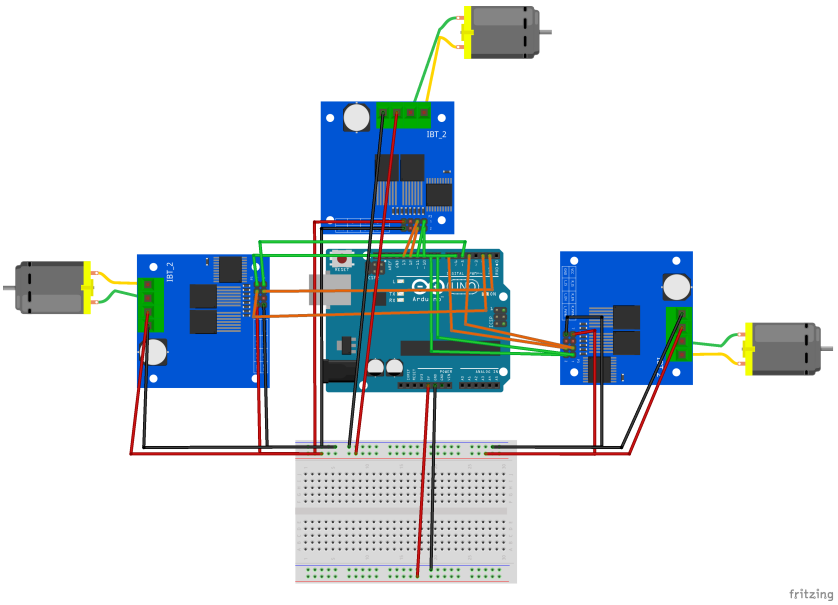

Figure 8: Hardware connections for Motor and Motor Drivers

I2C.

The Jetson Xavier and the DC motors require 12V, and the Arduino is powered with 5V. Conveniently, there are two 5V output pins on the Jetson Xavier which we can hook to the Arduino, so a buck converter is not needed.

5.2 Software Communication

Our system requires communication between camera vision information such as depth and coordinates of the object, as well as the motor control on the Arduino. Since all of us on the team are very familiar with Python, we decide to use Python as the language to send serial data between these models. There is a package called PySerial which is a convenient tool to read data from the Arduino.

5.3 Computer Vision

5.3.1 Navigation:

For navigation we will be using AprilTags. We chose to use AprilTags because the Tags are easy and free to generate and print. The detection software is also highly accurate, within 4 centimeters of the actual position when the camera is 2 meters away from the item. [8] The detection software is originally written in Java, but we will be using a version written in Python for simplicity.[9] The code will use the apriltag library available in Python, which is a library that allows for apriltag marker detection. The algorithm initializes a detector which then searches the input image for apriltags based on the family of tag provided. When a tag is found, the four corners of the tag are identified. This algorithm will allow the robot to navigate throughout the store and follow a specified user throughout their trip.

5.3.2 Laser Detection:

The laser detection algorithm will detect whether or not the laser is present in the current frame. The algorithm will be implmented in Python using the following steps [10]:
1) Convert the image from RGB to HSV color space. This will be done using OpenCv function.
2) Separate the image into the three different components (Hue, Saturation, and Value)
3) Apply a threshold, which will be determined by testing, to each component of the image, which will filter out the least bright parts of the image
4) Determine if the laser is in the frame based on the remaining HSV values. The exact threshold for the laser value will be determined by experimentation.

5.3.3 Edge Detection:

We will be using the Canny edge detection algorithm to detect the edges of our selected items. We chose the Canny edge detection algorithm after researching the different edge detection options and comparing the advantages and disadvantages for each. We compared Canny to Sobel, Zero Crossing, and Laplacian of Gaussian algorithms. From our research we found that the Canny algorithm was most accurate, especially in noisy conditions. The steps of the algorithm, which are implemented in Python, are as follows [11]:
1) Smooth the image using a Gaussian filter
2) Find the X and Y gradients of the image
3) Compute the direction of the edge using the gradients
4) Apply non maximal suppression to trace the edges
5) Use thresholds to finalize the edges

6 PROJECT MANAGEMENT

6.1 Schedule

See figure 11 on page 14 for the schedule.

6.2 Team Member Responsibilities

Ludi Cao is primarily responsible for building the drivetrain and implementing its navigation to and from the shelf.

Esther Jang is primarily responsible for building and programming the linear slide system and setting up the claw gripper. She will also spend an extra week starting with the overall system integration.

Bhumika Kapur is primarily responsible for the computer vision aspect of the project. She will be configuring the camera, writing the AprilTag, edge, and laser detection algorithms

6.3 Budget

See page 11 for the bill of materials for our various subsystems.

6.4 Risk Management

We made sure to budget enough time for integration of our subsystems as there is a big risk of our entire system failing if integration does not succeed. Hence, we set aside 3 weeks in our time schedule and also had Esther spend a week starting on integration to allocate even more time to this task.

In terms of the potential design risks, we have the concern that the robot would not be able to detect the laser pointer when it starts beside the user. A strategy to mitigate this risk is to add a constraint for the maximal starting distance between the shelf and the robot. We believe this is the most convenient solution and does not add unrealistic assumptions to our use case, since the aisle width between shelves in a grocery store is typically limited as well. However, if shortening the distance does not improve the performance significantly, a sub-ideal approach would be to restrict the background colors of the items we test on. Intuitively, shining a laser pointer on a dark background should improve detection rate.

Another concern would be the claw not able to grab the object firmly enough to withhold it from dropping when travelling to the basket. We discuss that wrapping the claw with material of high coefficient of friction would mitigate the issue. In the worse case, since we still have room in our budget, we can order a more powerful claw, as through research claws vary significantly in price, and hopefully a claw of better built quality would have improved performance.

Finally, since the navigation algorithm is complex and involves communication with each subsystem, there is a risk that the navigation algorithm would have a higher failure rate for a sub-procedure. Possible mitigations would be to include additional sensors that

7 RELATED WORK

IAM Robotics Swift: An autonomous manipulation robot that can autonomously pick items specified in a fulfillment order from shelves. A shelf can be up to 7 feet tall, and the robot can scan through an aisle of shelves. The robot commonly uses a vacuum gripper end effector that can hold up to 15 lbs. It also uses 3D mapping-based computer vision for its autonomy control. [12]

Amazon Picking Challenge: The company Amazon used to host a robotics competition which involved robotics being able to pick objects off a shelf. The winning robot of 2017's competition used an end effector that was a combination of a suction gripper, claw, and sliding mechanism. It also trained itself on models of objects to guide its computer vision detection. [13]

8 SUMMARY

We hope to create an efficient and robust system that will make grocery shopping accessible and convenient for everyone. So far, we have spent most of our time finalizing our design and hope to begin implementing our solution soon.

9 Bibliography

1. https://www.businessinsider.com/typical-cvs-shopper-demographic-urban-genx-earning-high-income-2021-9
2. https://pubmed.ncbi.nlm.nih.gov/18247253/
3. https://backyardbrains.com/experiments/reactiontime
4. https://www.rcgroups.com/forums/showthread.php?2798973-How-long-to-charge-a-3s-11-1v-5000mah-Lipo
5. https://hypertextbook.com/facts/2005/aluminum.shtml
6. https://magpi.raspberrypi.com/articles/raspberry-pi-4-specs-benchmarks
7. https://developer.nvidia.com/embedded/jetson-modules
8. https://roboticsknowledgebase.com/wiki/sensing/apriltags/
9. https://www.pyimagesearch.com/2020/11/02/apriltag-with-python/

10. https://bradmontgomery.net/blog/tracking-a-laser-pointer-with-python-and-opencv/

11. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.6312rep=rep1type=pdf

12. https://www.iamrobotics.com/products/swift/

13. https://www.nature.com/articles/s42256-019-0031-6.pdf

Table 1: Bill of Materials for Chassis

Description	Model #	Manufacturer	Quantity	Cost @	Total
HD Hex Motor 40:1 Spur Gearbox	REV-41-1301	Rev Robotics	4	$30.00	$120.00
15mm Extrusion - 1m - 90° Ends	REV-41-1017	Rev Robotics	2	$12.00	$24.00
90mm Omni Wheel - 2 Pack	REV-41-1190	Rev Robotics	4	$25.00	$100.00
Slim Shaft Collar - 10 Pack	REV-41-1629	Rev Robotics	1	$8.00	$8.00
15mm Plastic 135 Degree Bracket - 8 Pack	REV-41-1310	Rev Robotics	2	$5.00	$10.00
15mm Metal Flat HD Hex Motor Bracket V2 - 4 Pack	REV-41-1486	Rev Robotics	1	$5.00	$5.00
15mm Metal Bent HD Hex Motor Bracket V2 - 4 Pack	REV-41-1487	Rev Robotics	1	$5.00	$5.00
M3 Nut - 100 Pack	REV-41-1126	Rev Robotics	1	$5.00	$5.00
M3 x 16mm Hex Cap Screws - 100 Pack - 4 Pack	REV-41-1360	Rev Robotics	1	$11.00	$11.00
DC Stepper Motor Driver	BTS7960	DORHEA	1	$27.55	$27.55
Ovonic 11.1V 5000mAh 3S 50C-100C LiPo Battery	Lipo Battery	Ovonic	1	$27.99	$27.99
HTRC LiPo Battery Charger	B3AC Pro	HTRC	1	$22.24	$22.24
					$354.84

Table 2: Bill of Materials for Linear Slides and Claw

Description	Model #	Manufacturer	Quantity	Cost @	Total
HD Hex Motor 40:1 Spur Gearbox	REV-41-1301	Rev Robotics	2	$30.00	$60.00
15mm Metal Flat HD Hex Motor Bracket V2 - 4 Pack	REV-41-1486	Rev Robotics	1	$5.00	$5.00
15mm Metal Bent HD Hex Motor Bracket V2 - 4 Pack	REV-41-1487	Rev Robotics	1	$5.00	$5.00
15mm Linear Motion Kit V2	REV-45-1507	Rev Robotics	3	$12.00	$36.00
5.5mm Nut Driver	REV-41-1119	Rev Robotics	1	$6.50	$6.50
2mm Allen Wrench	REV-41-1377	Rev Robotics	1	$1.00	$1.00
15mm Extrusion - 1m - 90° Ends	REV-41-1017	Rev Robotics	2	$12.00	$24.00
Small Pulley Bearings - 10 Pack	REV-41-1368	Rev Robotics	1	$11.00	$11.00
1.2mm UHMWPE Cord - 10M	REV-41-1162	Rev Robotics	1	$6.00	$6.00
15mm Metal HD Inside Corner Bracket - 4 Pack	REV-41-1688	Rev Robotics	1	$10.00	$10.00
15mm Plastic Lap Corner Bracket - 8 Pack	REV-41-1321	Rev Robotics	1	$5.00	$5.00
60mm Pulley - 4 Pack	REV-41-1345	Rev Robotics	1	$8.00	$8.00
Mechanical Claw BigClaw Robot Gripper w/ 335MG Servo	B089KMK954	LewanSoul	1	$33.00	$33.00
LewanSoul Mechanical Claw BigClaw Robot Gripper	B08Q7XZVR4	LewanSoul	1	$23.00	$23.00
Adafruit 16-Channel 12-bit PWM/Servo Driver	PCA9685	Adafruit	1	$15.00	$15.00
DC Stepper Motor Driver	BTS7960	Teyleten Robot	1	$10.50	$10.50
					$259

Table 3: Bill of Materials for Miscellaneous Items

Description	Model #	Manufacturer	Quantity	Cost @	Total
SanDisk 32GB MicroSDHC Memory Card	SDSDQ-032G-A11M	SanDisk	1	$7.50	$7.50
Shipping From Vendor			1	$14.50	$14.50
Intel RealSense	D435	Intel	1	$299.00	Borrowed
Jetson Xavier NX		Nvidia	1	$399.00	Borrowed
Arduino Uno	A000066	Arduino	2	$23.00	Already Owned
					$25

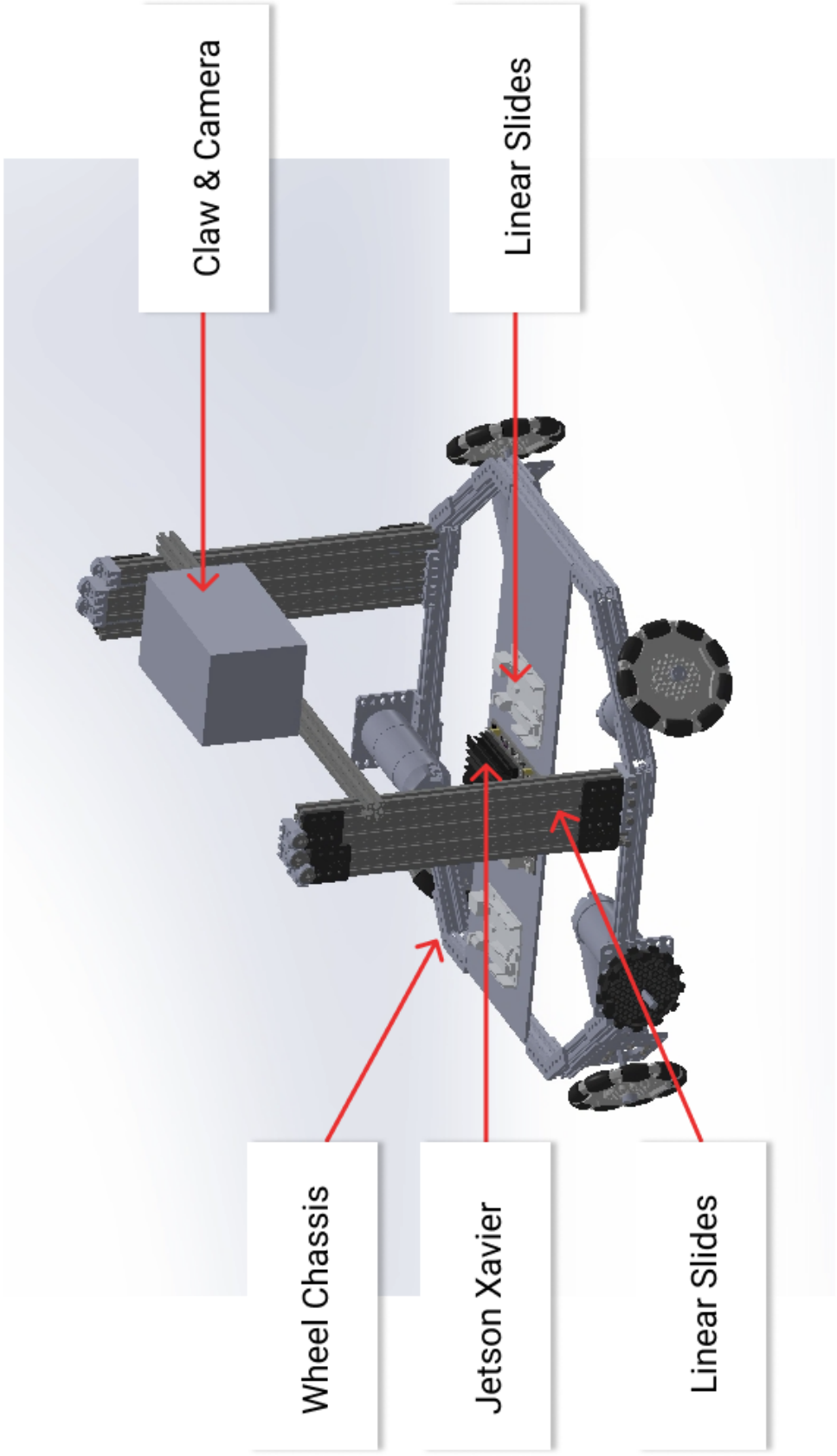

Figure 9: Full CAD of System

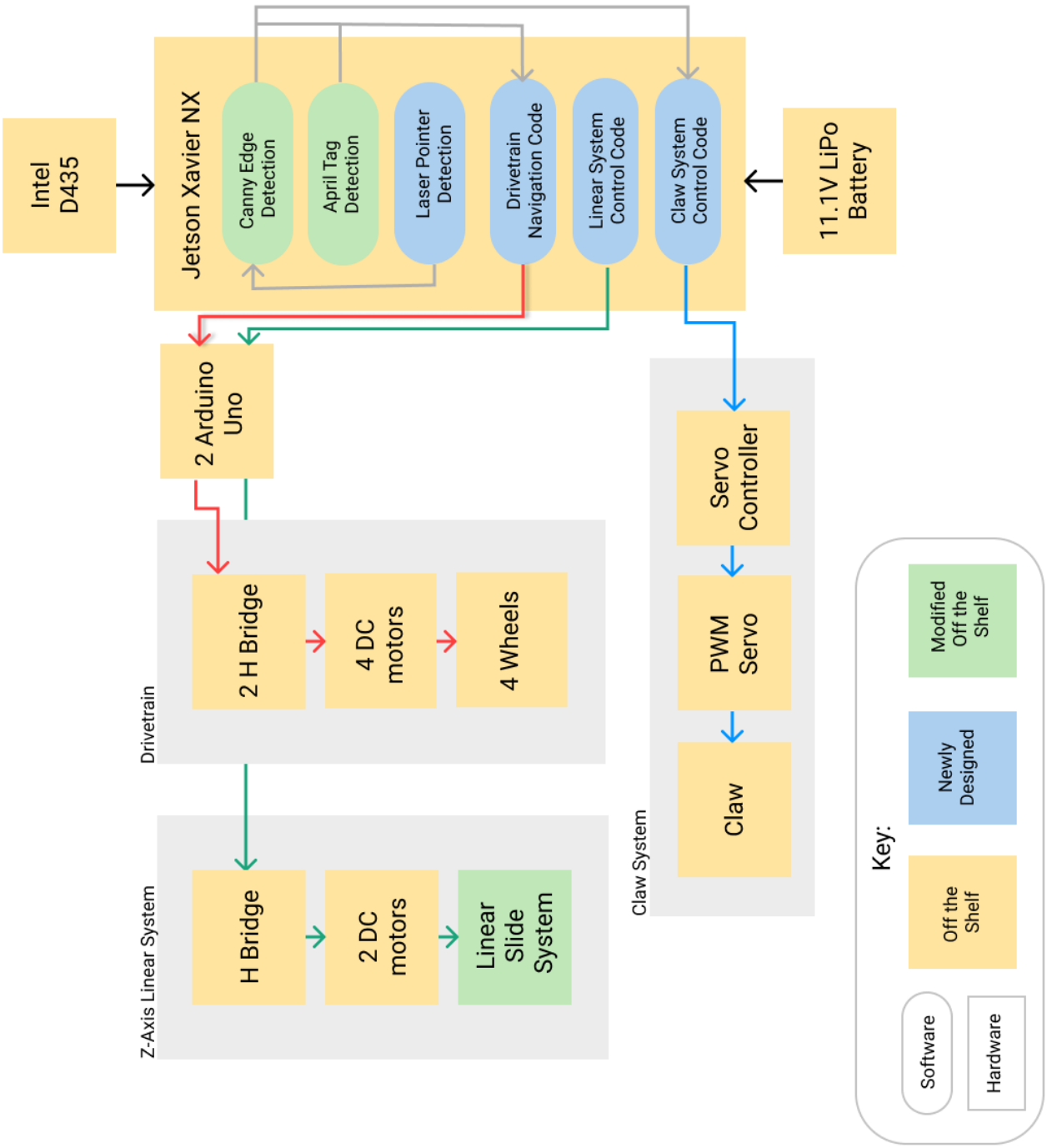

Figure 10: Block Diagram of System

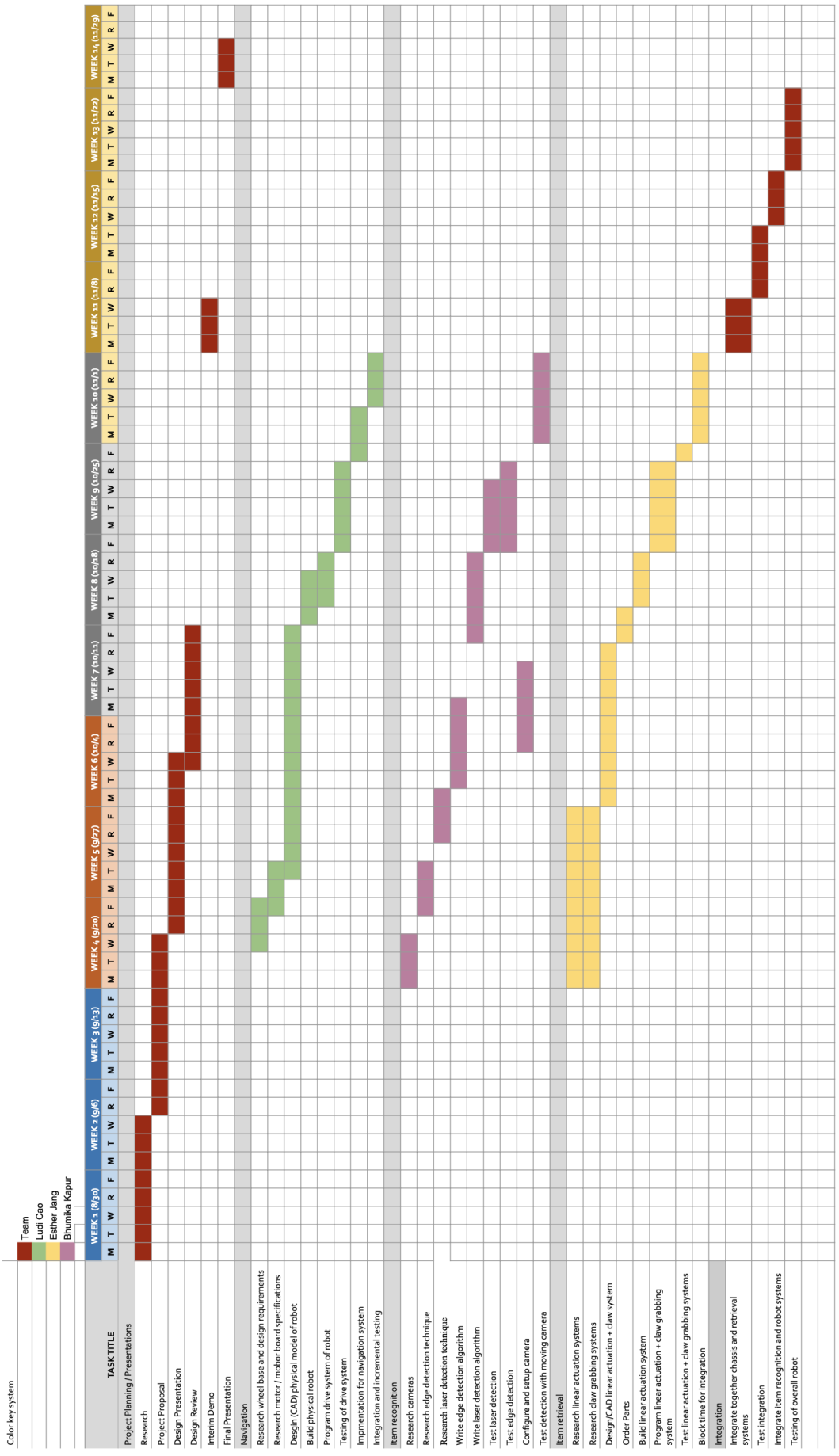

Figure 11: Gantt Chart