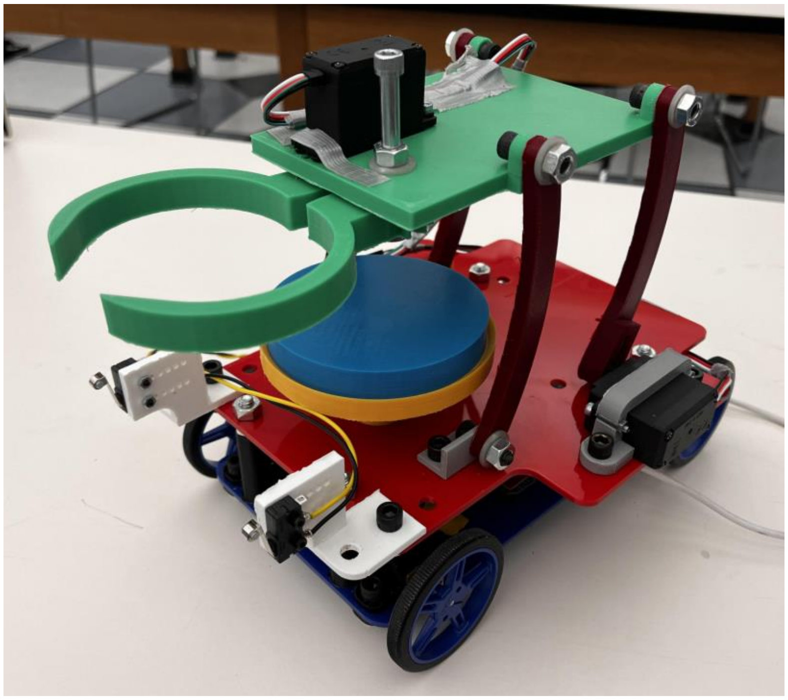
MAE 162 Final Report

Prof. YenChi Chang
TA Kevin Lai, Jason Liu

Group 5
Skyler Coleman
Andrew Gautier
Nathan Ge
Getty George V
Karen Su

MAE 162E: Mechanical Product Design-II
Spring 2024



Mechanical & Aerospace Engineering Department
University of California, Los Angeles

June 14, 2024

Abstract

This design report details the development, controls, and manufacturing process of an innovative autonomous “burger transporter” robot. Initially, both high and low-level design requirements are established. Of the three proposed concepts, the four-bar mechanism was selected for further development. The report provides a comprehensive design overview, including detailed subsystem specifications. Critical calculations are performed to ensure optimal performance, covering slope clearance capabilities, tractive force, power and torque requirements, four-bar dimensioning, trajectory planning, and servo specifications. Additionally, the report provides an in-depth analysis of the control design.

The subsequent section covers project execution details, describing the product fabrication processes in detail. It includes a comprehensive breakdown of material and labor costs, as well as project planning specifics. The robot's performance is evaluated against predefined metrics to verify its capability and compliance with design requirements. The report concludes by affirming that the four-bar mechanism design successfully meets all system requirements, demonstrating the robot's effectiveness in transporting burgers autonomously.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description	Units
G	Grashof Index (Four Bar)	m
V	Validity Index (Four Bar)	m
s	Shortest Member (Four Bar)	m
l	Longest Member (Four Bar)	m
p	1st Other Member (Four Bar)	m
q	2nd Other Member (Four Bar)	m
μ	Coefficient of friction between rubber and plywood	-
N	Normal force	N
L	Wheelbase	m
L_c	Center of Mass distance from rear wheel axel	m
h_c	Center of Mass height off the ground	m
N	Normal Force	N
g	Gravitational Acceleration Constant	m/s^2
m	Total mass of robot	kg
P	Power	W
F	Force	N
F_{mg}	Total Weight Force	N
r	Radius	m
τ	Torque	N-m
β	% weight distribution between rear and front wheels	-
d	Distance	m
t	Time	s
V	Velocity	m/s
a	Acceleration	m/s^2

1. Introduction

The task at hand is to create a fast food delivery system meant to retrieve a two-piece burger from the kitchen and deliver it to the customer in the restaurant. The end use of this product is to replace servers in fast-food restaurants and automate the food delivery process. For the lab model, the robot must pick up the correct patties determined by a dice roll. It must then follow a path through an obstacle course consisting of a ramp and a height barrier to drop off the burger at a counter. The client also expects the robot to pass a drop test, be marketable as a product, and cost less than $500 to build. A common example of a similar existing robot is Airpuria's "ServiPlus" which is capable of delivering plates from the kitchen to the customer, following a certain path each time. The vision and navigation system of the ServiPlus is similar to the current goal and can be learned from, however, the group's design is more useful as it can pick up and drop off the order without human intervention.

Figure 1: The ServiPlus, one of many food delivery products available on the market [1]

Starship Technologies has been a leader in autonomous food transportation with their small, self-driving robots designed for last-mile delivery of groceries and restaurant foods. Since its founding in 2014, Starship has completed over 6 million deliveries in several countries in Europe and 50 university campuses in the United States, like UCLA. Rather than line following, Starship robots use an array of cameras and sensors to map the world around them and effectively navigate on sidewalks. The technology focuses on precision and safety, utilizing machine learning algorithms to avoid obstacles and deal with challenges in urban environments such as construction and congested gatherings. The robots also have a manual override if unexpected circumstances cause the autonomous system to fail. Starship's robots have strongly influenced the food delivery industry by providing an environmentally friendly solution to delivery driver shortages. Several similar food delivery systems have been deployed since the spread of Starship, such as Coco, which is a human-operated robot delivery company based in Santa Monica.

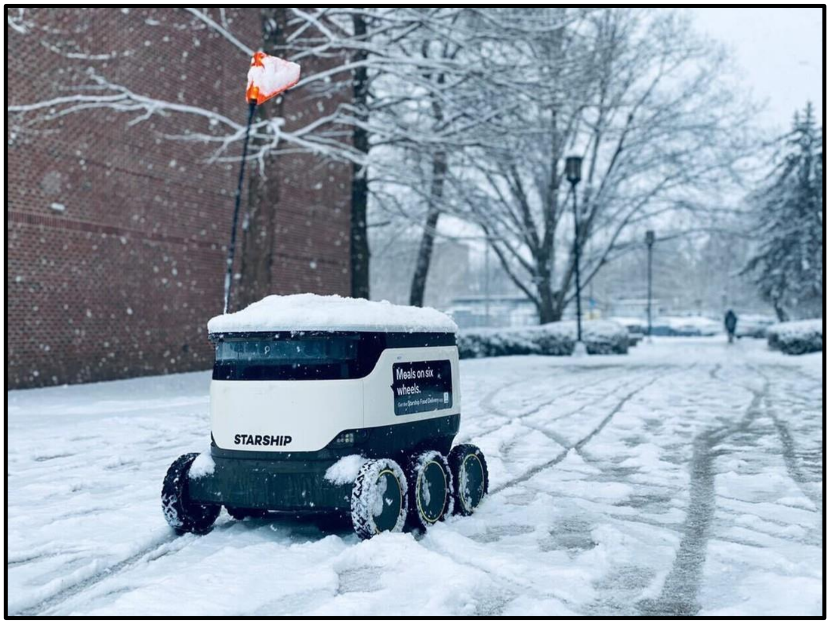


Figure 2: Starship robots traverse through difficult environments to autonomously deliver food [2]

The team aims to have the hamburger robot detect the dice roll by sight, carry at least three burger components, and complete the task within two minutes. The robot will be built from scratch using laser cutting, 3D printing, and hand assembly.

2. Design Description

2.1 Design Concept Developments

In this section, our top three concepts we believe can meet customer requirements and budget constraints are presented and shown in detail. These concepts were generated by first individually coming up with ideas and then discussing with the group and the TAs to determine features from different designs that the group wanted to keep and combine.

Table 1: High-Level Design Requirements (HLDR)

HLDR*	Description	Comment
1	The robot shall follow a black line through the entire course	Basic Task
2	Robot shall be capable of autonomously transporting two 3" diameter disks in tandem.	Basic Task
3	The robot shall be capable of retrieving the two disks from two locations in a specified order from a platform 14.75 cm tall.	Basic Task
4	The robot shall be capable of dispensing the disks in a stack on a platform 14.75 cm tall.	Basic Task
5	The robot shall be capable of transporting the disks up and down a slope of 10 degrees.	The robot will navigate up and down a hill on the track.
6	The robot shall stop for a dynamic obstacle	Basic Task
7	The robot shall survive a drop test	Basic Task
8	The robot shall cost at most 500 dollars.	Hard cost
9	The robot shall recognize dice to retrieve the correct disks	Bonus task
10	Robot shall be capable of carrying 3 disks	Bonus task

Table 2: Low-Level Design Requirements (LLDR)

LLDR	Description	Comment
1	The robot shall have a combined intake and outtake system to retrieve and put back discs onto designated platforms.	Combining the intake and outtake system will simplify the amount of moving parts on the robot and ensure consistency.
2	The robot shall have a pedestal to hold the discs once it has picked them up	This will allow the robot to transport two parts at a time as is desired
3	The robot shall complete the task promptly	The goal is 2 minutes from start to finish
4	The robot shall use a Raspberry Pi as a microcontroller	The Raspberry Pi allows the team to use a camera and Python programming as required in LLDR 4
5	The robot shall be programmed in Python	Python has a wide variety of easy-to-use libraries and more helpful error messages, so it will be able to accomplish more with it than C++
6	The robot shall have an attached camera to attempt to read dice rolls	The Raspberry Pi can easily attach to the camera, and Python has many libraries to perform computer vision tasks
7	The robot shall not use any sensors that utilize an analog output	The Raspberry Pi does not have an ADC, so trying to use these sensors will pose a challenge. Digital counterparts to analog sensors are numerous and readily available.

2.1.1 Rokusho Four-bar

The first concept attempts to greatly simplify the mechanical design of the robot by combining the horizontal and vertical motion of the grabbing arm into a single mechanism. The four-bar is a well-studied mechanism that involves two sets of parallel members joined with pin joints. The lengths of the members can be adjusted to change the motion and trajectory of the mechanism given a rotary motion input at one of the joints. At the opposite end of the driven joint of the mechanism will be a claw that can open and close to retrieve the cylinders. When retracted, the claw will rest above a pedestal, allowing the robot to store multiple cylinders at a time. Beneath this mechanism is a simple drive train that will use tank steering to maneuver. This concept was individually generated and then modified with help from a TA. It was chosen because of the novelty of its four-bar design. Looking at the HLDR and LLDRs set, the four-bar design is capable of achieving all of the requirements. The intake and outtake system is the same with just the motor reversed, which will ensure consistency in grabbing and dispensing. Additionally, the camera and sensors designed for this robot satisfy the LLDRs set.

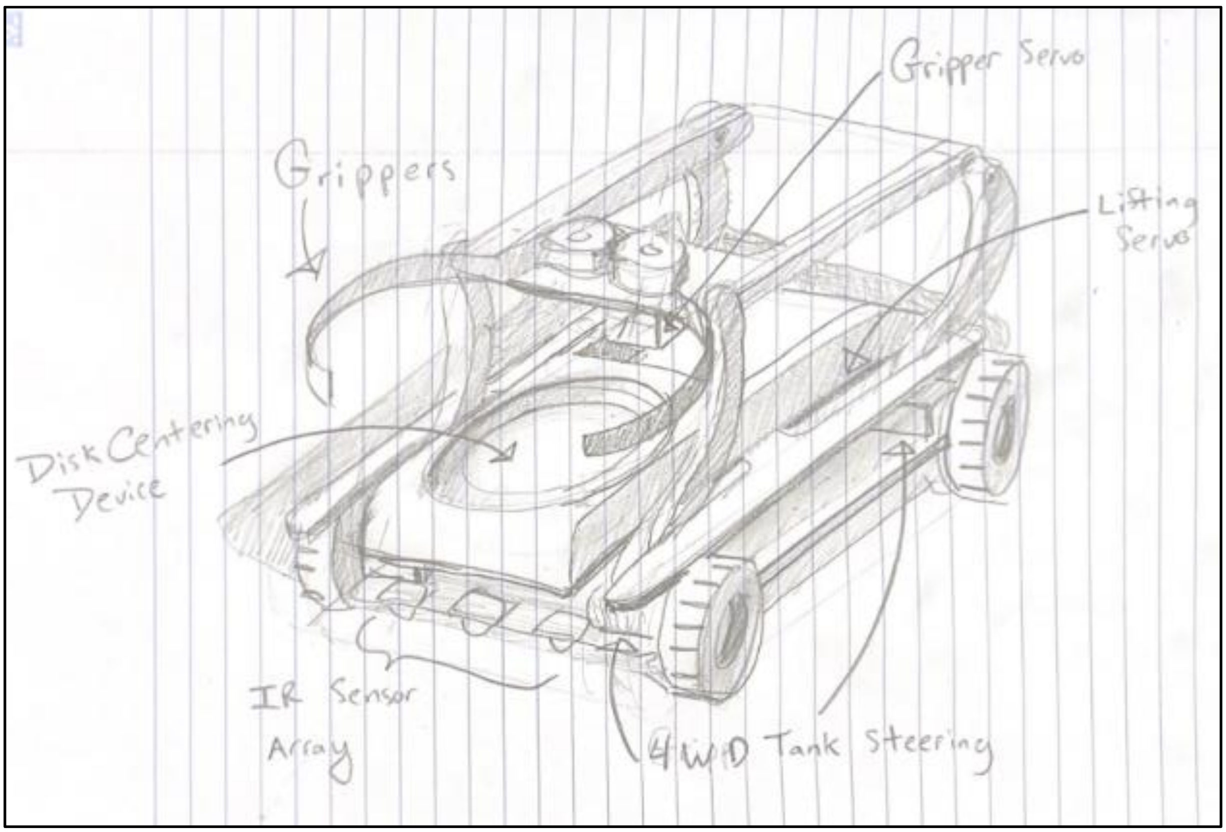


Figure 3: Concept Sketch of concept 1 (four-bar)

2.1.2 Double arm

This robot uses a pair of arms to pick up and store the cylinder pieces. The arms would extend from opposite sides of the robot, and each arm would grab a cylinder and flip over, storing it in the robot's main body. Once they have been stacked, a pushing mechanism would push the stack onto the pedestal. Notably, the two arms can be tucked inside the robot at the beginning and end of the task, allowing it to pass the HLDR of fitting through the gate. This concept was generated individually and modified as a group. It was chosen because the mecanum wheels were determined to likely result in high reliability for the positioning of the claw. This design does not meet the LLDR of a combined intake/outtake system, meaning that it requires more motors and moving parts. This design also does not meet the LLDR of using tank treads, which means it will be more complex in terms of mechanics and programming. The extra complexity means that this design is less likely to work. All other HLDRs and LLDRs could be met.

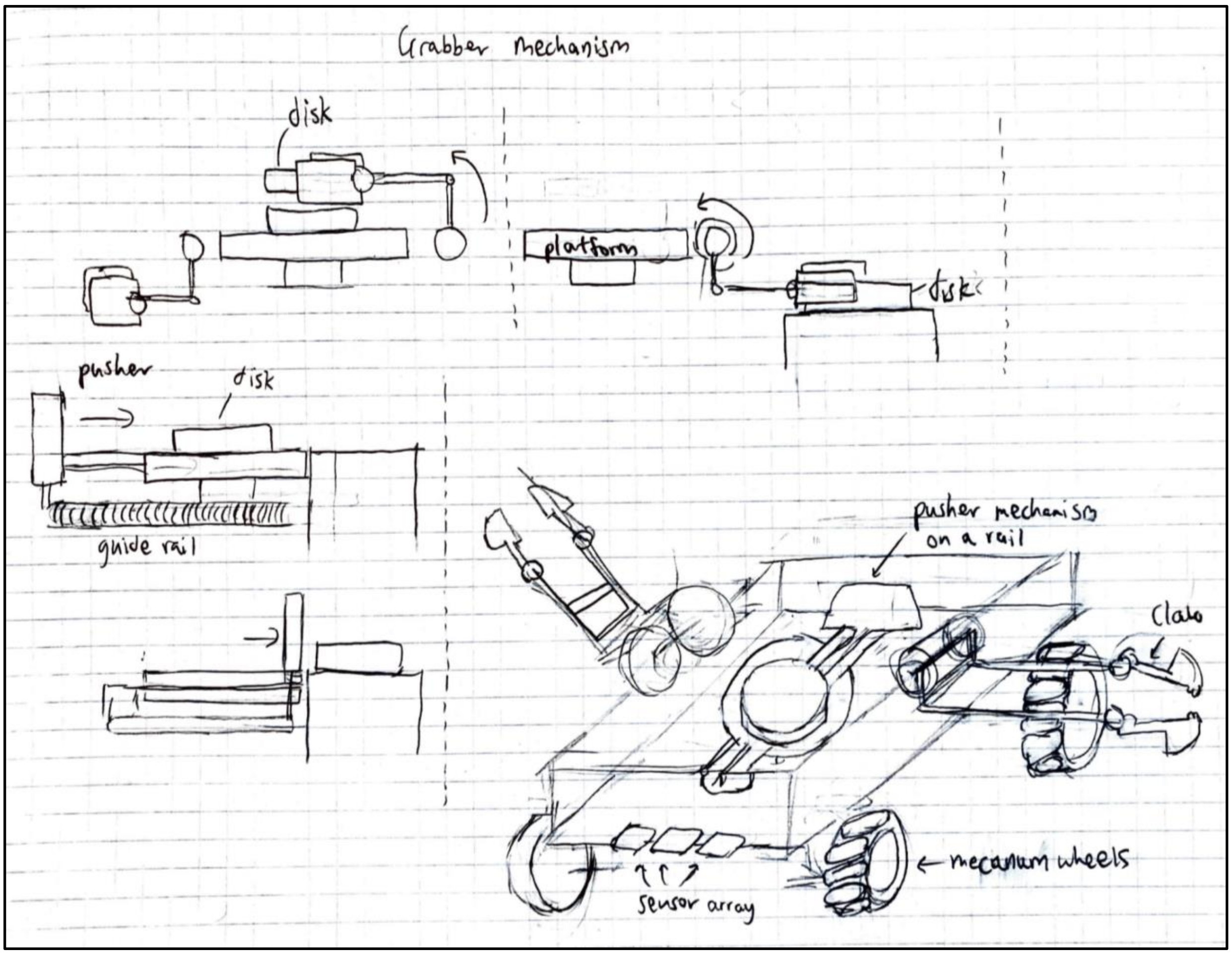


Figure 4: Concept Sketch of Concept 2 (double arm)

2.1.3 Roller Intake

The final robot concept utilized a roller and belt system similar to what can be found on an assembly line. Each belt would be equipped with fins to grab the cylinders which would then get secured by the rest of the belt. Once taken in, the robot could take in another cylinder with the same method and stack the cylinders with an internal mechanism. Finally, a conveyor belt that is under the disk base can output the disks from the other side of the robot. This concept was inspired by existing intake robots and it was chosen because of various YouTube videos demonstrating that this is a common method for robots to intake disks. In regards to the HLDR and LLDRs, this design should have no trouble achieving all of the HLDRs. However, this design doesn't achieve LLDR 1 which requires the intake and outtake system to be the same mechanism. Thus, the roller intake design might not be as simple and consistent as desired because of the increased amount of moving mechanisms. Note that tank treads could be used to meet the LLDR, but they are not pictured.

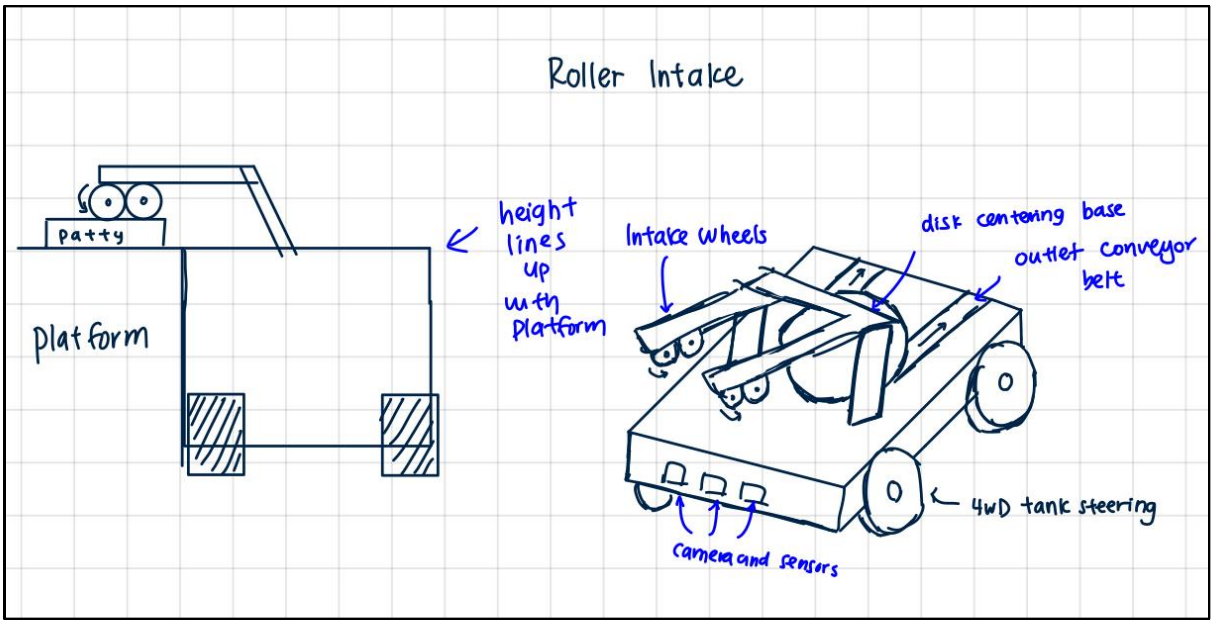


Figure 5: Concept Sketch of Concept 3 (roller intake)

2.1.4 Pairwise Comparison Chart and Objectives-Tree

Table 3: Pairwise Comparison Chart

						# of Designers / Scale						
						5						
			Compared To									
	Is this Factor Used? (1 = YES, 0 = NO)	Factors	Complexity	Aesthetics	Robustness	Durability	Weight	Novelty	Complexity (P)	Maneuverability	Total	Weight
Asset	1	Complexity (M)	5	0	5	1	0	0	0	4	15	0.08
	1	Aesthetics	5	5	5	5	5	3	5	5	38	0.21
	1	Robustness	0	0	5	0	0	0	0	0	5	0.03
	1	Durability	4	0	5	5	0	0	1	5	20	0.11
	1	Weight	5	0	5	5	5	0	5	4	29	0.16
	1	Novelty	5	2	5	5	5	5	4	5	36	0.20
	1	Complexity (P)	5	0	5	4	0	1	5	5	25	0.14
	1	Maneuverability	1	0	5	0	1	0	0	5	12	0.07
Aaron Aoyama, MAE-162B, UCLA, Fall 2010									Total Sum:		180	1.00

Table 4: Objectives Tree

		Design Scores			Weighted Design Scores		
Design Objective Factors	PCC Weight	Rokusho Four Bar	Double Arm	Roller Intake	Design 1	Design 2	Design 3
Complexity (M)	0.08	9	6	5	0.75	0.50	0.42
Aesthetics	0.21	8	5	6	1.69	1.06	1.27
Robustness	0.03	8	6	4	0.22	0.17	0.11
Durability	0.11	7	6	9	0.78	0.67	1.00
Weight	0.16	9	7	4	1.45	1.13	0.64
Novelty	0.20	10	9	6	2.00	1.80	1.20
Complexity (P)	0.14	5	8	7	0.69	1.11	0.97
Maneuverability	0.07	7	7	7	0.47	0.47	0.47
	1.00			Total Score:	8.05	6.89	6.08

The Objective Tree entries were decided on as a group based on what the group felt were important considerations based on the group's design abilities and the expectations of the customer. "Complexity (M)" is mechanical complexity and represents how complex the moving parts of the robot are. "Aesthetics" represents the general appearance. "Robustness" is the ability to tolerate different or unexpected situations. "Durability" represents the ability of the robot to withstand damage, namely from the drop test. "Weight" is the mass of the robot, which should be low to reduce the motor power needed. "Novelty" is how unique the design is. "Complexity (P)" is programming complexity and represents the difficulty associated with the coding. "Maneuverability" is the robot's ability to swiftly move through the course.

2.2 Design Overview

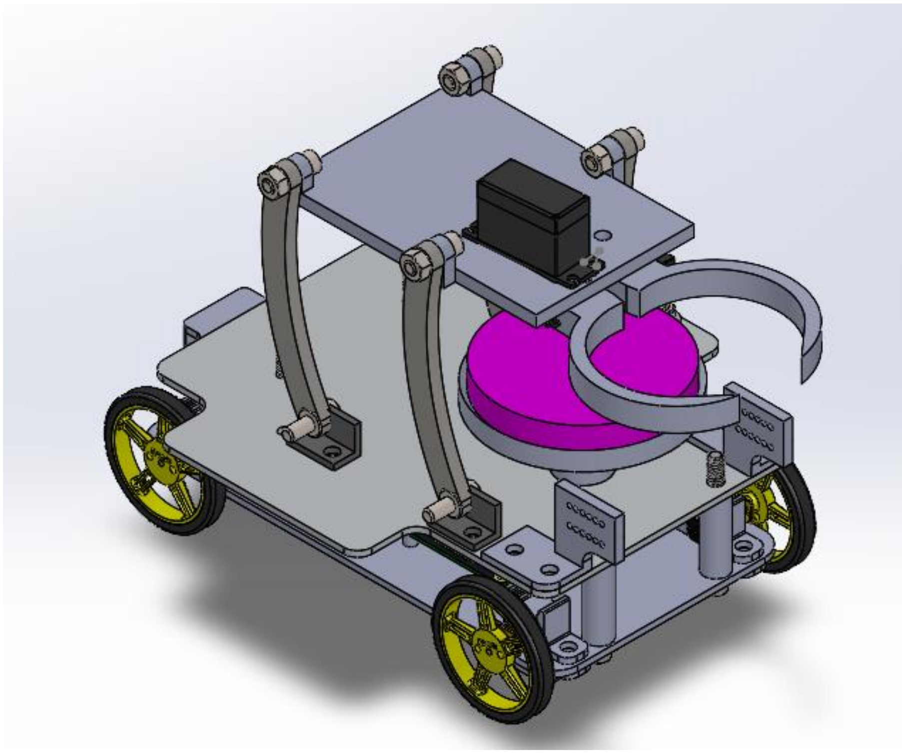


Figure 6: CAD Assembly of the Robot

Our robot is broken into four main parts: the drivetrain, the grabber mechanism, the disk catcher, and the limit switches. The drivetrain is two acrylic laser-cut pieces that are separated by 3D-printed spacers and bolts. The bottom layer is where we mount all of the electrical hardware, motors, and wheels. The top layer houses our grabber mechanism, the disk catcher, and the limit switches. The grabber mechanism consists of a four-bar that's separated by a 3D-printed plate. The reason we chose to go with a four-bar design is because it's able to achieve vertical and horizontal distance at the same time. This allowed them to create a compact robot and have more space to add the disk catcher in the middle of the robot body. This plate between the four-bar links has two geared claws and a servo mounted onto it. The way this claw is designed allows for the placement of the disk on the platform to be forgiving as the claw can center the disk very well.

When the robot drives toward the platform and the limit switch at the front of the robot is pressed, the four-bar servo turns so the entire grabber mechanism shifts forward. Our controls then go through a series of actions to open the claw, grab the disk, and bring it back to the center of the disk catcher. This process repeats two more times to grab three disks. During transport, the claw will hold onto the topmost disk to ensure disc stability. Once the robot reaches the final destination, with the activation of the limit switch, the grabber mechanism will go down to the bottom disk and grab all three disks at once to place them onto the platform.

This design was created to have a simple yet robust robot. The mechanical components of this robot were easy to maintain and hard to break. This allowed our team to fine-tune the software side very well and prioritize testing the line following and grabbing sequence.

2.3 System Specifications

The robot has dimensions of 19.89cm (height) by 23cm (length) by 14.71cm (width), which fits the required dimensions of 20cm x 25cm x 30 cm. The drive train consists of four 100:1 Micro Metal Gearmotors with a rated torque of. Each motor is controlled by an MP6550 Single Brushed DC Motor Driver. A 13 IR sensor array module is used for line tracking. The operating voltage of the IR sensor is 2.9 V to 5.5 V. The output format is analog voltages. This line tracker was chosen for its large amount of sensors and its length as it can span across the thickness of the black line. The wheel diameter of 6cm was chosen as it allowed our robot to clear the 10-degree slope.

The grabber mechanism is dimensioned such that it perfectly arrives at the height of the platform which is 14.76cm high. The geared claw mechanism in its closed state ensures secure contact on multiple points of the disk. This provides a strong grip on the disk while it's traveling down to the disk catcher. The grabber mechanism was designed to grab onto 3-in disks. The disk catcher is created at a specific height and width to ensure it can hold the 3-in disks and stack 3 of the disks. The limit switches at the front of the robot are used to also detect where there is a dynamic object and continue running when the object is removed.

The entire robot cost $451.34 which is below the high-level requirement of $500. It takes about 1 minute and 15 seconds for our robot to complete the entire course which achieves our lower level requirement of 2 minutes.

On the software side, a Raspberry Pi 3 was implemented to allow us to upload software to the robot remotely and get data logs after each run. This streamlined the process of troubleshooting as we could know what the sensors on our robot were reading at all times.

2.4 Mechanical Systems

Our robot has four main mechanical systems. The drivetrain houses all of our electrical hardware, battery packs, and motors.

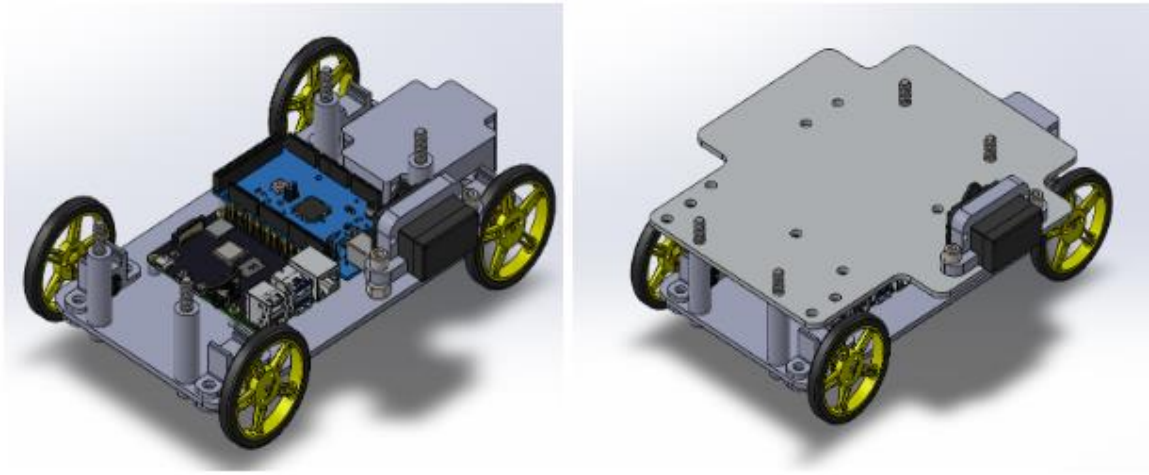


Figure 7: CAD Assembly of the Drivetrain

There is a four-bar grabber mechanism that can grab onto the disks from the platform and bring it back inside the body of our robot.

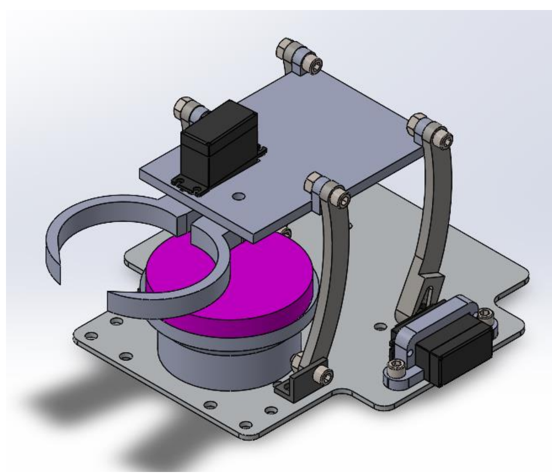


Figure 8: CAD Assembly of Grabber Mechanism

There is a disk catcher piece that is placed at the end of the four-bar trajectory. Lastly, there are limit switches at the front of our robot that control the distance between the platform and our robot and tell the robot when to actuate and grab the disk.

2.5 Control Systems

The control systems of the robot consist of all our sensors that are attached to the robot and the motors we have as well. There is an IR sensor array, two limit switches, two servo motors, and four DC motors for the wheels. The sensors and how they affect the rest of the motors will be discussed more in Section 5.3 of this report.

3. Subsystem Design Description

3.1 Structural Subsystem

3.1.1 Design Requirements

Structural Integrity

- **Material Selection**: Use lightweight yet durable materials such as wood or high-strength plastics to balance weight and strength.
- **Load-Bearing Capacity**: The structure should support the robot's components, payload, and additional forces during operation.
- **Impact Resistance**: Design for resilience against shocks, vibrations, and impacts during operation and abnormal conditions (i.e. a drop test)

Chassis Design

- **Geometry**: Ensure the chassis is geometrically optimized for stability and weight distribution while remaining compact
- **Modularity**: Design the chassis to be modular for easy assembly, maintenance, and upgrades.
- **Mounting Points**: Provide standardized mounting points for sensors, controllers, and actuators.
- **Ease of manufacturing:** Design simplicity to employ cost-effective and cheap manufacturing techniques such as 3d-printing and laser cutting

Drive Base

- **Wheel Alignment**: Ensure proper alignment for all four wheels to maintain straight-line motion and minimize wear.
- **Ground Clearance**: Determine appropriate ground clearance, wheelbase, and track width based on ramp clearance capability

Power and Wiring Management

- **Battery Placement**: Secure and accessible battery placement for balance and easy replacement.
- **Cable Routing**: Design pathways for safe and efficient routing of wires and cables to avoid interference and damage and allow for easy troubleshooting

3.1.2 Subsystem Description

The structural base of the robot consists of two .13-inch thick laser-cut acrylic plates. 4 bolt-spacer columns separating the two acrylic plates provide ample space to mount 4 drivetrain motors microcontrollers, and wire routing to the bottom plate. The top acrylic plate houses mechanical components (four-bar linkage and grabber mechanism), and the batteries for accessibility reasons. Modular separation of disc collection and locomotive functionality enables parallel development and ease of troubleshooting.

All mounting points adhere to the 1/4"-20 screw standard for simplicity.

3.1.3 Subsystem CAD Models and Engineering Drawings

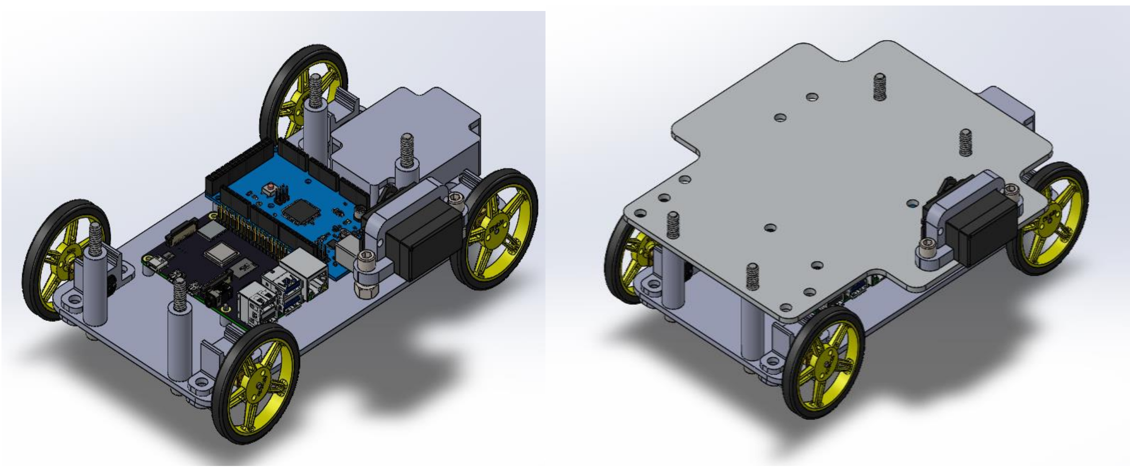


Figure 9: Drive base assembly with and without top acrylic cover

3.2 Mechanical Subsystem

3.2.1 Design Requirements

Mechanical Integrity

- **Material Selection**: Use high-strength, lightweight materials like aluminum alloys, stainless steel, or carbon fiber to ensure durability and performance.
- **Load Capacity**: Design the mechanism to handle the maximum expected load without deformation or failure.
- **Wear Resistance**: Select materials and finishes that minimize wear and extend the lifespan of moving parts.

Actuation

- **Actuator Type**: Choose an appropriate actuator (e.g., servo motor) based on required force, speed, and control precision.
- **Placement**: Position the actuator to optimize leverage and efficiency, minimizing power consumption while maximizing grip strength.

Linkage Design

- **Link Lengths and Ratios**: Calculate the link lengths/ratios to achieve the desired motion
- **Joint Design**: Ensure joints are robust and provide smooth, frictionless movement.
- **Tolerance and Clearances**: Maintain tight tolerances to ensure precision but allow enough clearance to prevent binding or excessive friction.

Gripper Design

- **Jaw Geometry**: Design the gripper jaws to match the shape and size of the discs being handled and accommodate a range of disc positions relative to the grabber
- **Grip Force**: Ensure the grip force is sufficient to hold objects securely without falling out
- **Opening Width**: Define the gripper's maximum and minimum opening widths to accommodate various disc sizes.
- **Movement Precision**: Ensure the mechanism provides precise and repeatable movements for consistent performance.

Integration and Mounting

- **Space Constraints**: The design must conform to spatial constraints within the system and application environment.
- **Assembly**: Design for easy assembly, disassembly, and maintenance.

The following design requirements were recognized but not implemented with the understanding that our robot would not see commercial use

Safety and Reliability

- **Overload Protection**: Incorporate overload protection mechanisms (e.g., torque limiters, slip clutches) to prevent damage in case of excessive force.
- **Safety Features**: Include safety features such as emergency stops and failsafe positions to protect operators and equipment.

Control: Integrate precise control mechanisms (e.g., feedback sensors, controllers) to achieve accurate positioning and gripping force.

3.2.2 Subsystem Description

The platform housing the grabber claw mechanism is sandwiched between four-bar linkages. Only the left side four-bar linkage is directly motorized. The input link, driven by a servo motor, rotates to move the coupler and output links, translating to the simultaneous horizontal and vertical motion of the disc retrieval/placement assembly.

The claw mechanism is driven by a singular servo directly attached to one claw piece. The two claw pieces are connected via gears in a 1:1 gear ratio integrated into their design. As the servo rotates one claw piece, the other moves in a mirrored motion, enabling the opening and closing of the claw.

Coordinated motion between the claw and the four-bar linkage enables disc retrieval and placement to and from the disc platform

All joints within this subsystem consist of thru-holes joined via 1/4"-20 head screws and nuts. While this joint design may lack smoothness, it maintains uniformity in hardware usage and meets our operational requirements effectively.

3.2.3 Subsystem CAD Models and Engineering Drawings

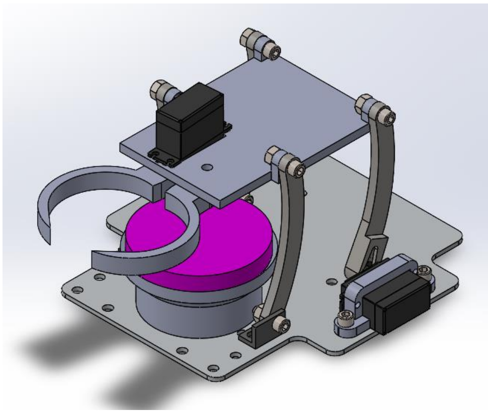


Figure 10: Disc retrieval/placement assembly

3.3 Electrical Subsystem

3.3.1 Design Requirements

Power Supply and Distribution

- **Wiring**: Use appropriate gauge wires for power distribution to minimize voltage drop and heat generation.
- **Connectors**: Select robust, reliable connectors that are easy to connect/disconnect and are rated for the current they will carry.
- **Power**: Choose an appropriate power delivery system (e.g., batteries, AC power) based on the robot's operational environment and duration.
- **Voltage and Current Requirements**: Ensure the power supply meets the voltage and current needs of all components, including motors, sensors, and controllers.

Embedded Systems

- **Microcontroller/Processor**: Choose a microcontroller or processor with adequate processing power, memory, and I/O capabilities to handle all tasks.
- **Communication Protocols**: Use reliable communication protocols (e.g., SSH) suitable for remote code revisions and data logging.

Sensors and Actuators

- **Compatibility**: Ensure all sensors and actuators are compatible with the control system's voltage, current, and communication protocol requirements.
- **Precision and Accuracy**: Select sensors and actuators with the required precision and accuracy for the application.
- **Mounting and Placement**: Strategically place sensors and actuators to optimize performance and accessibility.

3.3.2 Subsystem Description

One 5V battery powers A Raspberry Pi 3. This Raspberry Pi, in turn, supplies logic power to an Arduino ATMega 2560 microcontroller through its 5V port. The Arduino distributes logic power to all sensors and actuators on the robot, including limit switches, an IR sensor array, and motor drivers. A separate 5V battery source is dedicated to providing motor power to servos and motor drivers. This separation ensures that the power-hungry actuators do not interfere with the logic power delivery, meeting our power requirements effectively.

Importantly, all electronic components were specced to operate on our 5V power delivery standard.

The Raspberry Pi functions as an intermediary between the laptop and the onboard Arduino microcontroller. It connects to the laptop via SSH communication protocol and to the Arduino via serial communication. This setup allows for remote code uploads to the Arduino, facilitating rapid code iteration, and enables data logging from the Arduino to the laptop for analysis.

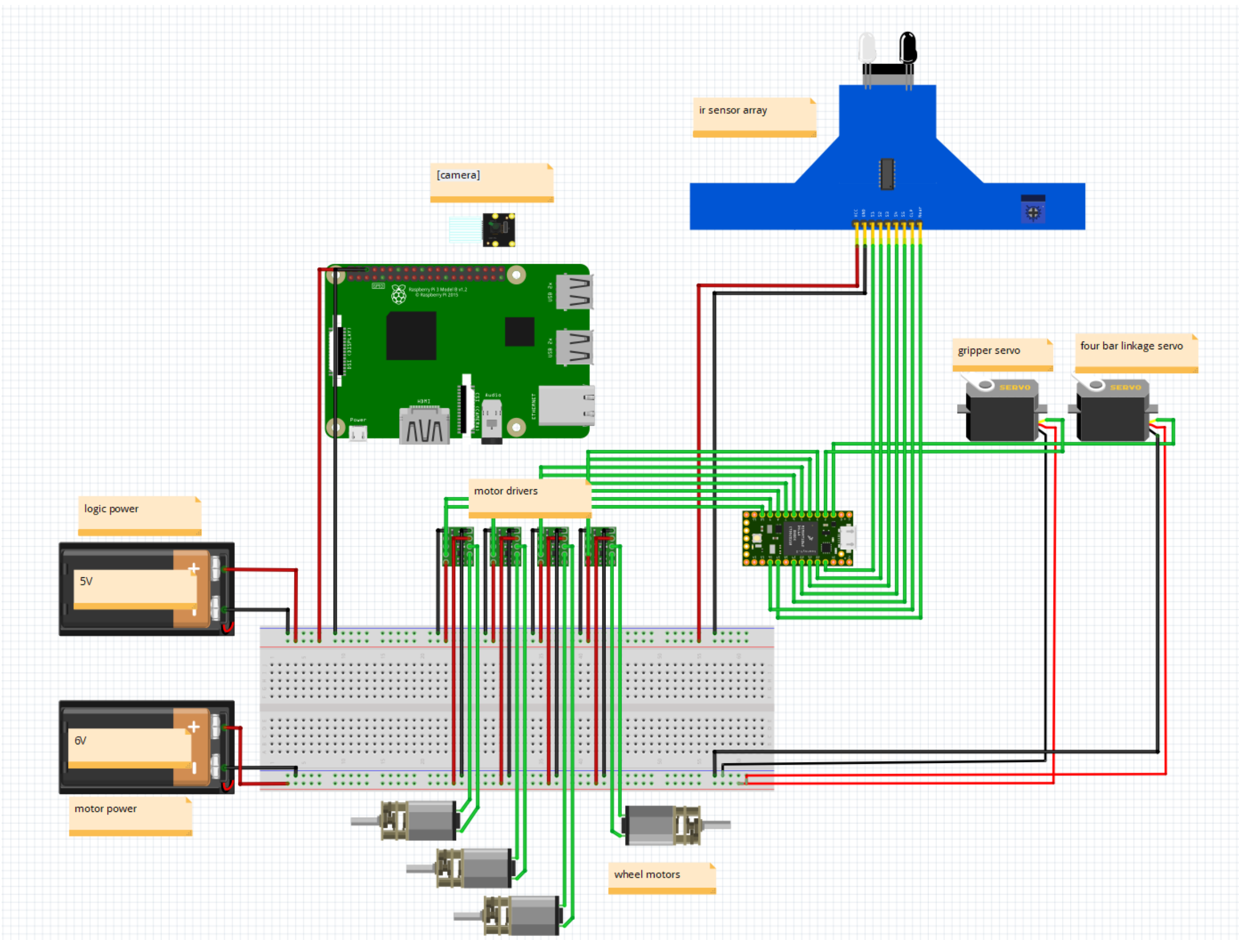


Figure 11: Wiring Schematic

4. Design Analysis

4.1 Preliminary Calculations

4.1.1 Total Weight

Table 5: List of Components and Weights in Pounds

Component	Weight (lbs)
Four Bar Mechanism	0.13
Drivetrain Plate (2x)	0.60
Wheel (4x)	0.80
Wheel Motors (4x)	0.78
Camera and Camera Mount	0.074
IR Sensor	0.028
Four Bar Servo	0.63
Gripper Mechanism (Double Claw)	0.16
Gripper Plate	0.043
Gripper Servo	0.63
Arduino	0.104
Battery	0.313

Total Weight of Robot: 4.92 lbs

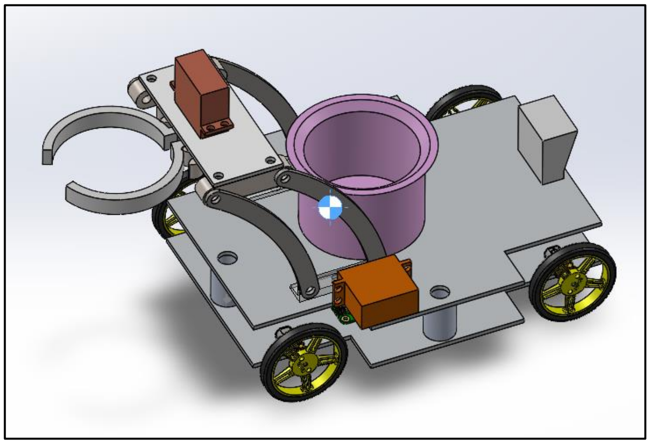


Figure 12: Center of Mass of Robot

4.1.2 Tractive force

Our wheels are made of rubber and the course is made from plywood. The lowest coefficient of friction between rubber and plywood found online is 0.7, so that value will be used [4]. The required minimum coefficients of friction for rear-wheel drive, front-wheel drive, and all-wheel drive are calculated below.

$$\mu_{FWD} = -L * sin(\theta)/(h_c * sin(\theta) - L_c * cos(\theta)) = 0.407$$
$$\mu_{RWD} = L * sin(\theta)/(h_c * sin(\theta) - L_c * cos(\theta) + L * cos(\theta)) = 0.785$$
$$\mu_{AWD} = tan(\theta) = 0.268$$

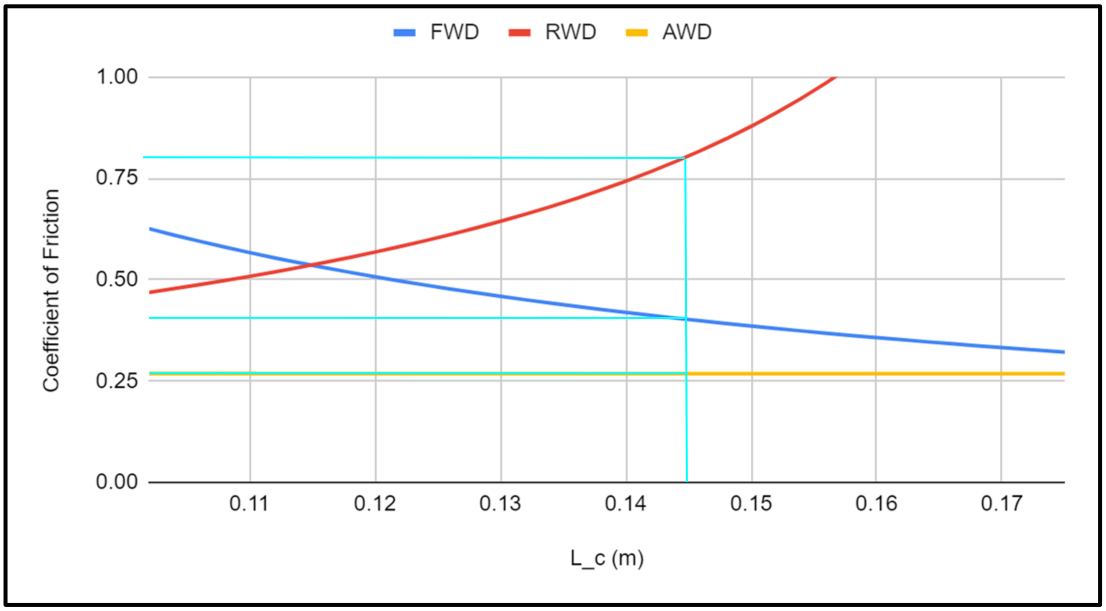


Figure 13: Friction coefficients for three drive trains at the COM distance of $L_c = 0.144m$

The normal force between the front and rear tires and the ground are calculated below.

$$N_f = (F_{mg} * L_c * cos(\theta) - F_{mg} * h_c * sin(\theta))/L = 8.61N$$
$$N_r = (F_{mg} * (L - L_c) * cos(\theta) - F_{mg} * h_c * sin(\theta))/L = 4.46N$$

The total tractive force is calculated below using the normal forces on the wheels and their coefficient of friction on the wood surface.

$$F_t = \mu(N_f + N_r) = 9.80N$$

The component of the gravitational force acting down along the slope is given as:

$$F_{g,slope} = mgsin(\theta) = 5.67N$$

Since the tractive force is greater than the force of gravity down the slope, it is confirmed that the cart will successfully climb the slope without slipping.

4.2 Move Profile

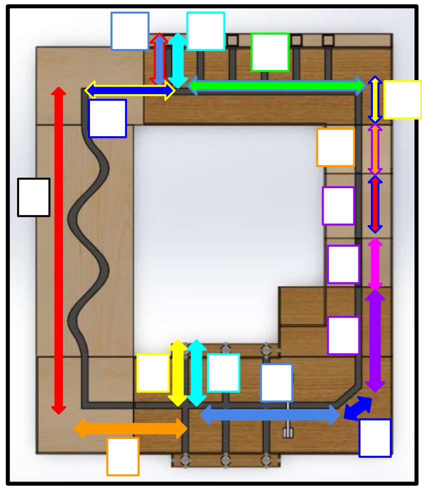


Figure 14: Move Profile Sectioning

4.2.1 Maximum Velocities along Paths

Table 6: Path Segment Length and Velocity

Path Segment	Segment Length (cm)	Time Spent on Segment (s)	Velocity (cm/s)
1	210.75	16	13.171875
2	78	5	15.6
3	35	3	11.66666667
4	35	3	11.66666667
5	115	8	14.375
6	25	2	12.5
7	80	6	13.33333333
8	40	6	6.666666667
9	50	4	12.5
10	40	6	6.666666667
11	29	4	7.25
12	148	10	14.8
13	35	3	11.66666667
14	35	3	11.66666667
15	65	6	10.83333333

β (% weight distribution on front wheels) and the coefficient of friction (μ)

$$\beta = L_c/L - (h_c/L * tan(\theta)) = 0.66$$

μ_{wheel} = 0.75

4.2.2 Move Profile and Velocity Estimates

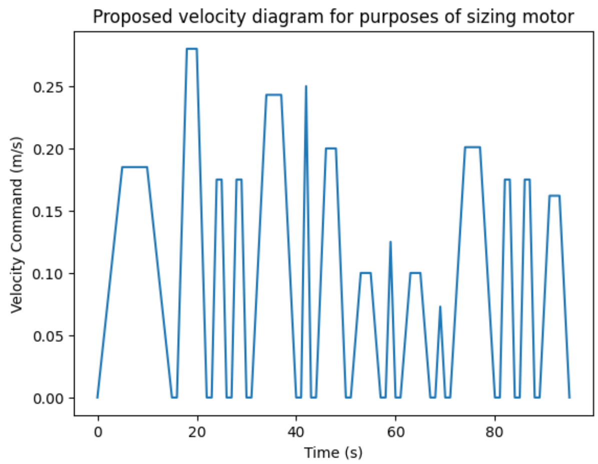


Figure 15: Move Profile Plot

Table 7: Estimated Velocity & Acceleration

Path Segment	d_{tot} (m)	t_{tot} (s)	V_{max} (m/s)	a_{max} (m/s^2)
1	2.1075	16	0.198	0.037
2	0.78	5	0.234	0.140
3	0.35	3	0.175	0.175
4	0.35	3	0.175	0.175
5	1.15	8	0.216	0.081
6	0.25	2	0.250	0.250
7	0.80	6	0.200	0.100
8	0.40	6	0.100	0.050
9	0.50	4	0.250	0.125
10	0.40	6	0.100	0.050
11	0.29	4	0.145	0.073
12	1.48	10	0.222	0.067
13	0.35	3	0.175	0.175
14	0.35	3	0.175	0.175
15	0.65	6	0.163	0.081

Accelerations are found by rearranging Newton's equation of motion to solve for acceleration, given the time estimates and distances for each segment. These derived equations are given below for triangular and trapezoidal velocity profiles, assuming that all accelerations, decelerations, and constant speed sections last for the same time interval.

$$a_{triangle} = d_{tot}/t_{tot}^2$$

$$a_{trapezoid} = d_{tot}/2t_{tot}^2$$

4.3 Drive System Power and Propulsion Torque Requirements

Propulsion forces

$$F_{prop} = F_i + F_w + F_f + F_{rol}$$

Table 8: Estimated Propulsion Force for Each Path Segment

Path Segment	F_i (N)	F_w (N)	F_f (N)	F_{rol} (N)	F_{prop} (N)
1	34.8	0	2949.3	138.2	3122.3
2	131.9	0	2949.3	138.2	3219.4
3	164.4	0	2949.3	138.2	3251.9
4	164.4	0	2949.3	138.2	3251.9
5	76.0	0	2949.3	138.2	3163.5
6	234.9	0	2949.3	138.2	3322.4
7	94.0	0	2949.3	138.2	3181.5
8	47.0	1600.4	2904.5	136.1	4688.0
9	117.4	0	2949.3	138.2	3205.0
10	-47.0	-1600.4	2904.5	136.1	1393.2
11	68.1	0	2949.3	138.2	3155.6
12	62.6	0	2949.3	138.2	3150.1
13	164.4	0	2949.3	138.2	3251.9
14	164.4	0	2949.3	138.2	3251.9
15	76.3	0	2949.3	138.2	3163.9

The following equations were used to compute the theoretical power and torque requirements along each segment of the track:

$$P_{req} = F_{prop} x\, V_{max}$$

$$\tau_{prop} = (F_{prop} x \frac{wheel\ dia}{2}) \frac{1}{\eta_{drive\ system\ efficiency}}$$

Table 9: Estimated Maximum Required Propulsion Power and Motor Torque along Each Segment

Path Segment	*Pprop (W)*	*Tprop (N-m)*
1	616.904	84.453
2	753.347	87.079
3	569.089	87.958
4	569.089	87.958
5	682.128	85.566
6	830.600	89.864
7	636.295	86.052
8	468.802	126.801
9	801.241	86.688
10	139.322	37.684
11	457.568	85.353
12	699.321	85.204
13	569.089	87.958
14	569.089	87.958
15	514.127	85.576

The selected drive system employs 4 separately controlled wheels, allowing for a tank drive subsystem. This doesn't greatly affect the amount of torque required to drive the device as each motor is directly linked to the wheel. The biggest thing that would affect the torque required by the drivetrain is the radius of the wheel that is used. The current design uses a 6cm diameter motor, meaning that if 1 N of force was required then 6cm*N of force would have to be applied.

Required motor torque (Treq < Tmotor)

$$\tau_{req} = \tau_{prop,max} = 126.801 N * m$$

4.4 Delivery Sub-System Calculations

The proposed delivery system utilizes a four-bar mechanism upon which a gripper mechanism is attached. The four-bar provides simultaneous vertical and horizontal movement but requires calculations to ensure that the proposed mechanism can reach the cylinder goals, store them without interference, and place them back. Additionally, a servo motor that is sufficiently powerful enough to drive the mechanism has to be calculated.

4.3.1 Driving servo calculations

A four-bar linkage was imagined with a mass representing the claw and payload at its very end. A wide range of four-bar mechanism states was considered by changing the angle of the input linkage. For all these states, the motor torque required to keep the mechanism static was calculated. From this, the required motor power can be calculated depending on how severe an angle is required.

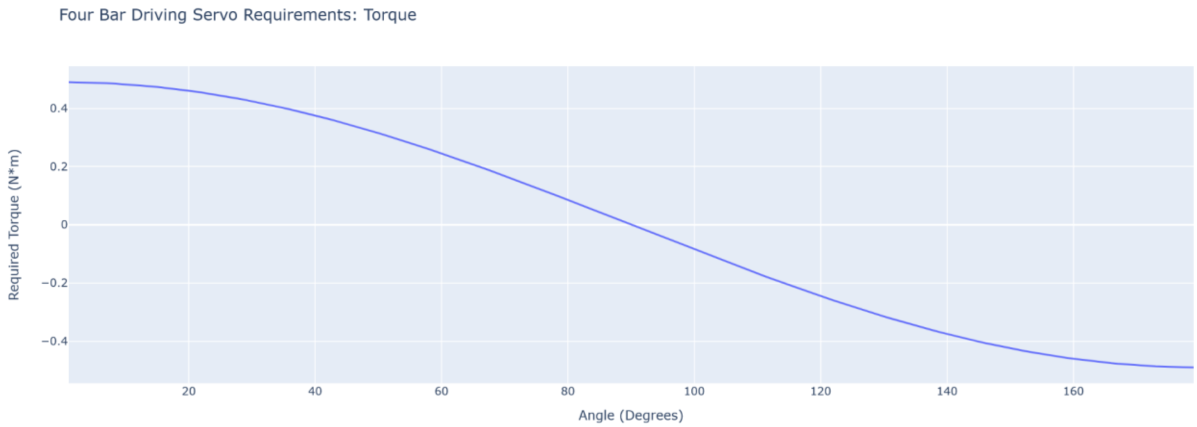
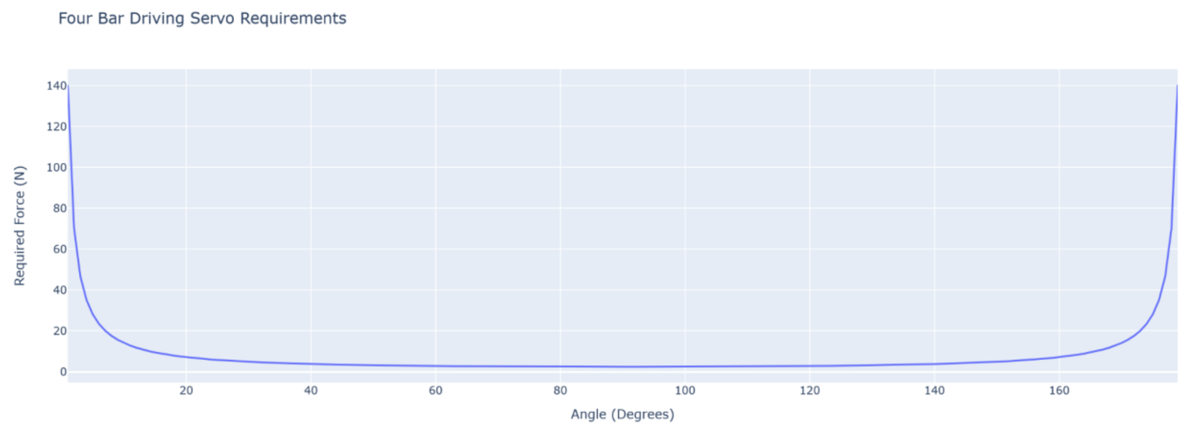


Figure 16: Motor torque required to hold the four-bar static over a range of four-bar angles



\

Figure 17: Radial Load required to hold the four-bar static over a range of four-bar angles

4.3.2 Four-Bar Trajectory Confirmation

Once a configuration of the four-bar mechanism was proposed, it was first tested using an online simulator to confirm that it theoretically possessed all the characteristics it needed. The simulation provided an animation of the mechanism that confirmed that the path taken by the mechanism was circular and that the floating link stayed parallel to the ground link at all times.

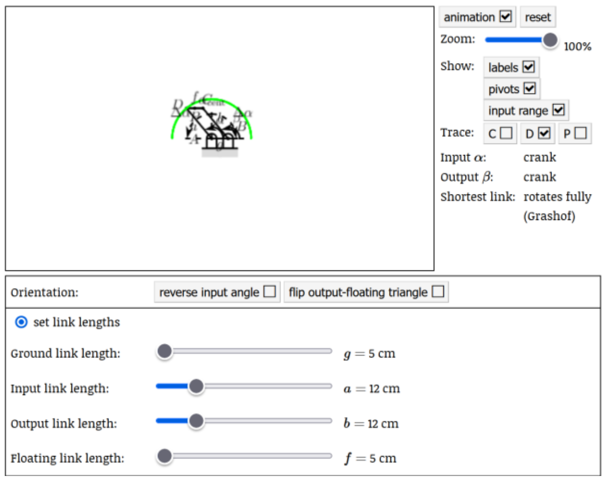


Figure 18: Calculated Four Bar Lengths Through an Online Simulator [3]

4.3.3 Validity and Grashof Index

The four-bar mechanism has several basic calculations that can be performed to predict the mechanism's validity as well as some of its characteristics. The validity index compares the length of the longest linkage to the sum of the lengths of the other linkages. If the longest linkage isn't longer than the sum then the four-bar design is possible. The Grashof index indicates if the shortest linkage can fully rotate or if it can only reciprocate. When that analysis was performed on the proposed design, it was found that the design was possible and would produce a Grashof linkage.

Grashof index:	$G = s + l - p - q = 0 \text{ cm} = 0$
Validity index:	$V = l - s - p - q = 10 \text{ cm} \geq 0$

5. Control System Design

5.1 Drivetrain

We selected the 100:1 Micro Metal Gearmotor MP 6V from Pololu to power the wheels of our robot. It was selected using the torque and speed specification using the drive system power and torque specifications above to have an FOS of at least 2. These motors were driven by a MP6550 Single Brushed DC Motor Driver Carrier that was also from Pololu. The Pololu motor driver was specified to provide enough current to the driving motors with a large factor of safety. While the motors were meant to be driven with 6V a 5V supply was selected since current was the limiting factor in motor operation.

Figure 19: 6V Micro Metal Gear motor

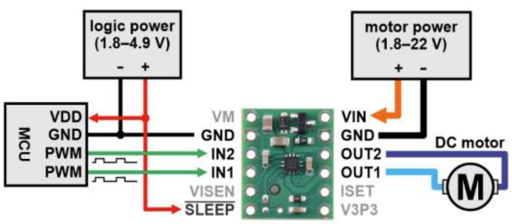


Figure 20: Wiring Diagram of MP6550 Motor Driver

Each motor driver required two PWM signals to modulate the direction and power of current flowing to the DC motors. In our control code, this was handled by a motor object that had helper functions to convert a desired drive percentage to PWM commands. The control of the drive motors was completely open-loop and tuned from empirical operations.

5.2 Food Retrieval/Delivery Mechanism

For dropoff/retrieval, the robot uses IR sensors to know which intersection to deliver the juice box. The robot counts T-junctions for retrieval and the right junctions for dropoff. Junction detection and their associated state transitions allow the robot to activate the desired retrieval/collection routine. Once the robot turns into the correct lane for dropoff/retrieval with a hard-coded turn program, the robot uses line following to traverse to the desired platform. Using two limit switches placed in front of the robot, the robot correctly aligns to the wall for accurate disc collection/retrieval. Once both limit switches are activated and alignment is ensured, the robot runs preprogrammed coordinated movements of the four-bar linkage and claw mechanisms to collect or drop discs. The robot proceeds to the backwards line following before it turns out of the branch.

*Refer to section 13.3 for code

5.3 Sensors and Theory of Operation

There are two inputs into the control system. IR Line Sensors and limit switches. Their model and signal conditioning are described below.

5.3.1 IR Line Following Sensors & Signal Filtering

QTRX-MD-13A was selected as our IR sensor array. It has 13 sensors 8mm apart and spans a distance of 10 cm or the entire width of the robot.

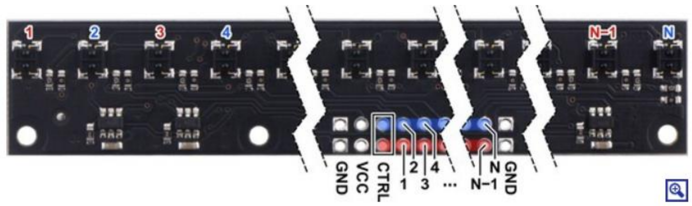


Figure 21: Pinout Diagram of the IR Sensor Array

Each sensor outputs an analog reading between 0 and 1023 bits. Before the robot begins nominal operation the IR sensor is calibrated on a dirty section of track. The minimum reflectance (black track) is recorded as the upper analog reading and the maximum reflectance (white track) is recorded as the lower bound reading. Then analog readings are mapped to a value between 0 and 1000 where 0 is the lower bound reading and 1000 is the upper bound. Figure 16 shows how these readings change over time when crossing over a T junction of black tape.

ssr1	ssr2	ssr3	ssr4	ssr5	ssr6	ssr7	ssr8	ssr9	ssr10	ssr11	ssr12	ssr1
35	4	8	5	853	997	1000	1000	969	19	0	0	0
68	28	31	104	858	993	994	987	931	14	0	0	0
0	16	43	170	854	980	984	985	928	28	0	0	0
4	0	18	146	818	967	975	968	931	47	0	0	4
0	0	8	203	824	961	967	965	909	16	0	0	0
0	0	6	161	793	950	963	965	931	66	0	0	0
68	51	72	190	830	976	975	974	920	74	0	0	0
114	123	86	201	832	987	998	992	933	95	0	2	4
39	20	60	175	709	991	990	992	941	52	0	0	0
85	80	66	161	522	995	1000	994	950	177	187	259	295
968	991	983	1000	982	991	1000	1000	993	992	1000	1000	997
354	355	345	483	948	967	980	979	948	404	354	365	371
0	10	18	172	940	984	998	992	918	0	0	0	0
0	0	0	106	956	982	992	990	911	0	0	0	0
14	8	33	146	942	978	980	970	890	0	0	0	0
39	30	68	166	938	978	980	976	881	0	0	0	0
24	4	39	146	933	965	965	965	838	19	0	0	0
82	74	72	208	916	950	959	948	825	16	0	0	0
66	86	64	217	929	959	957	954	853	45	0	0	0

Figure 22: Readout of IR Sensors at a T Junction

After the inputs of the line-following sensors are read they undergo some signal conditioning for both the line-following controller and junction detector.

Line following
A weighted average of all the analog sensor readings is taken to determine the position of the center of the black line in relation to the robot. We've found that a value of 600 is around the center of the robot. A basic PID controller is used to determine the controller output as described in the base_controller object. This control output biases current to the left or right wheels which steers the robot towards the center of the line. For more detailed logic see Section 13.3 of the Appendix.

Junction Detection
To properly control our robot it must recognize left, right, and T junctions. During a control cycle, the leftmost and rightmost IR sensors are read to see if there is black tape that shoots off to the left or right of the robot. These values are written to a circular buffer which is then processed by the return_junction function. The return_junction function returns an enum of what the robot is currently seeing. It's been designed to be more sensitive to T junctions and less sensitive to left and right junctions since we've had issues with the robot detecting turns instead of T junctions. The logic of the return_junction function is in the main file in section 13.3 of the Appendix.

5.3.2 Limit Switches

Figure 23: SPDT Limit Switch

Two limit switches were wired to the Arduino Mega to detect objects and the wall of the course. All the pins on the Arduino Mega come with configurable pull-ups so only a digital pin and Gnd needed to be wired to the switches. In States 0 & 22, if any of the switches were pressed the robot would stop. The switches are also used to transition from state 22 (line following until a wall) to dropping off our grabbing a disk.

5.4 State Diagram & Overall Code Description

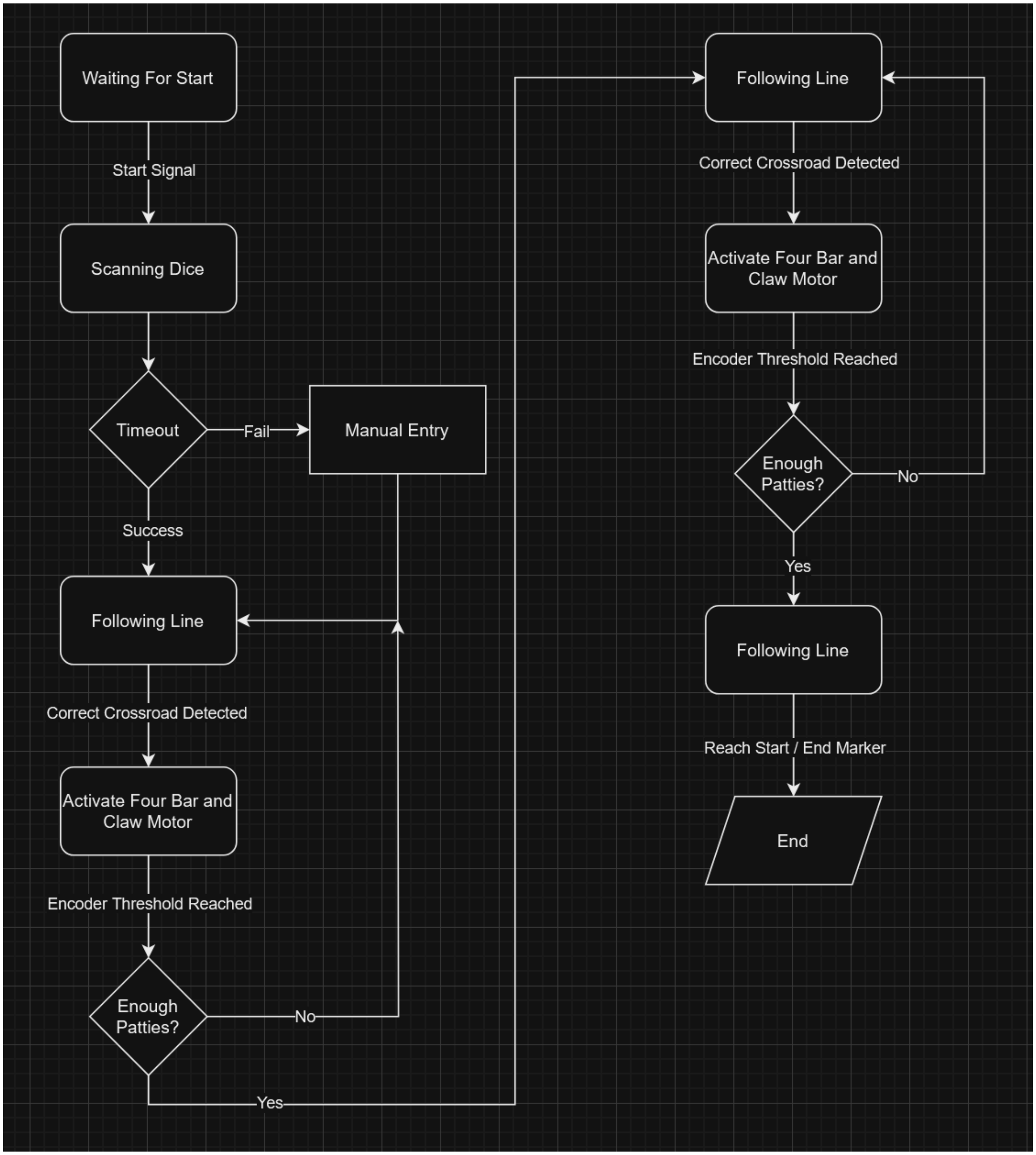


Figure 24: State Diagram

The state diagram was created to provide a structure for the software component of the project. It features a lot of the fundamental, high-level logic that allows the system to operate which is much easier for humans to read than code. The system uses multiple timeouts and logic breaks to account for issues that may arise during operation, allowing for continued operation instead of just stopping in a failed state.

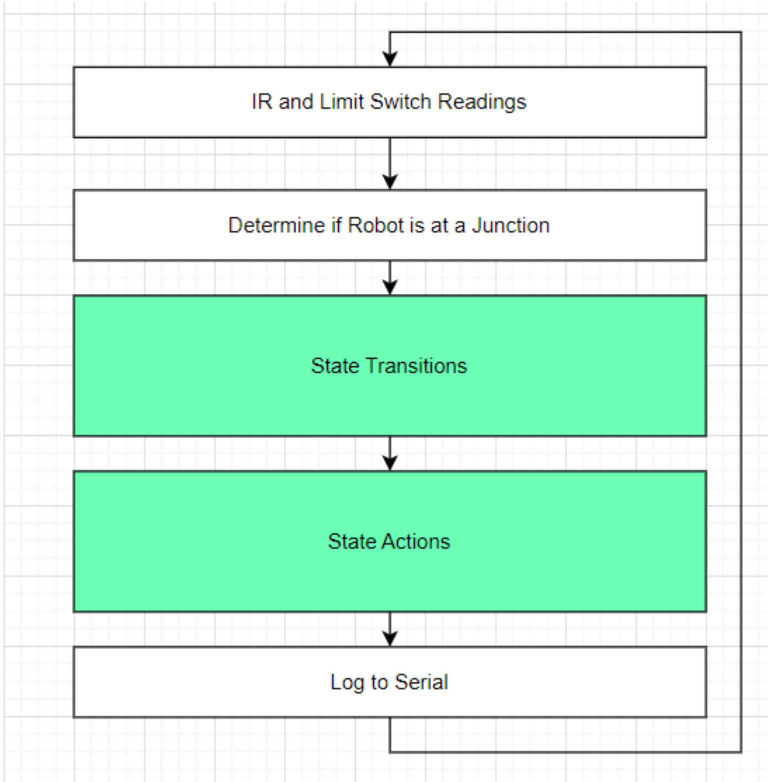


Figure 25: State Transition Diagram

6. Product Fabrication

6.1 Drive Systems

The baseplate - the primary structure of the drivetrain - was cut out of acrylic using a laser cutter. Four 3-d printed motor mounts were fastened to the baseplate using 1/4"-20 head screws and nuts. The accompanying electronics hardware was all mounted onto the holes of the bottom acrylic plate. We also chose to have a two-layer drivetrain so there was a second laser-cut acrylic piece mounted with 3D-printed spacers. All 3D printed parts on our robot are PLA material with 20% infill. These settings ensured a strong enough part while having a quicker print time.

Throughout the design and assembly process, we realized there were additional mounting holes needed as we didn't add on the Raspberry Pi or the limit switches till later in the design cycle. However, when we went to drill the holes, due to the brittle nature of acrylic, we ended up cracking a piece of the acrylic off. This forced us to completely recut the acrylic piece. We definitely should have been more careful about putting excessive force onto the piece as we didn't account in the timeline to recreate some of the major components of our robot.

Another oversight was the impact of weight on the handling of the robot. Although weight provided improved traction, we were unable to clear the ramp at max power. We had to swap to lighter battery sources and remove redundant hardware to remedy this issue.

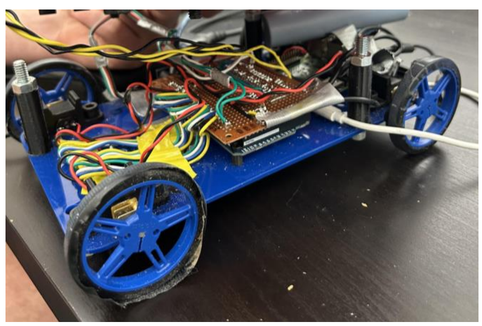


Figure 26: Electronic Components Mounted on Drivetrain Plate

6.2 Four Bar Mechanism

The four-bar mechanism was all 3D printed with holes designed so bolts could be used to mount the 3D printed parts onto the acrylic piece. 3D-printed L brackets were used to mount the four bars to the acrylic plate. When we were assembling the four-bar system, we unknowingly put too much force tightening the bolt to the nut and ended up deforming one of the L brackets. This led us to recreate the 3D-printed L bracket so it was longer and had more mounting points.

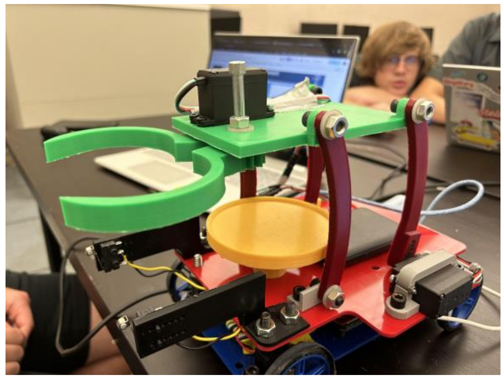


Figure 27: Grabber and Four Bar Mechanism Assembled

6.3 Grabber Mechanism

The grabber mechanism features a plate that connects to the four-bar as well as two geared claws. All of these components are 3D printed. One of the claws is directly connected to the servo and the other claw is attached with a bolt and nut. A common issue that we had was that the 3D printed pieces had a tolerance and made it hard to fit the bolts and servo attachments into the holes. We had to file down the holes to allow for a thru-hole as the bolt is meant to rotate inside the hole.

6.4 Final Product Pictures

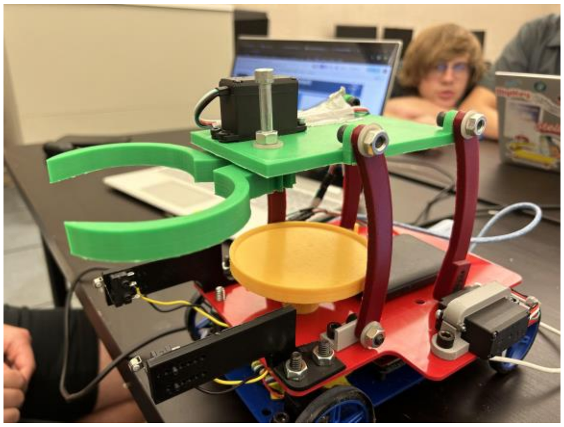


Figure 28: Grabber and Four Bar Mechanism

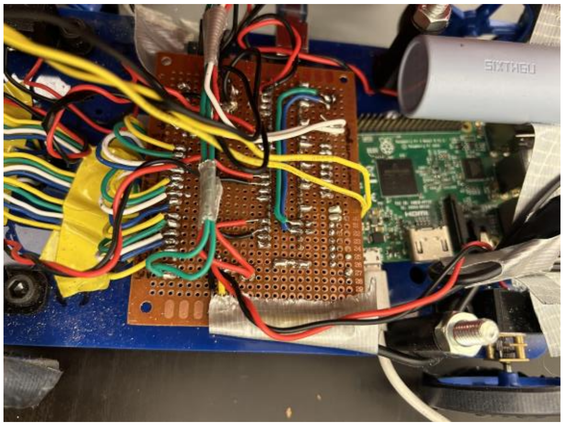


Figure 29: Electronics Configuration

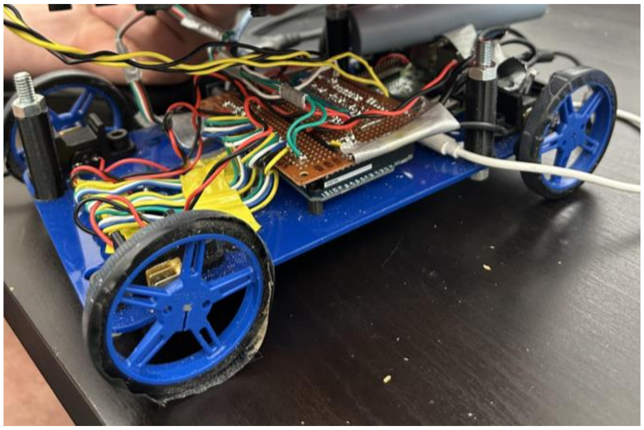


Figure 30: Electronic Components Mounted on Drivetrain

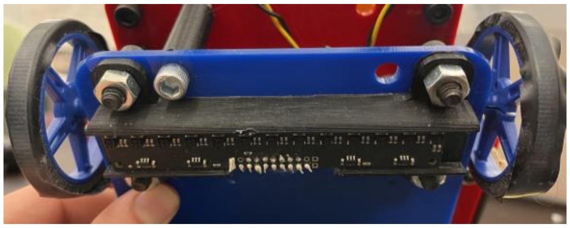


Figure 31: IR Sensor Mounted on Bottom of Car

7. Product Performance Testing and Evaluation

7.1 Run Times

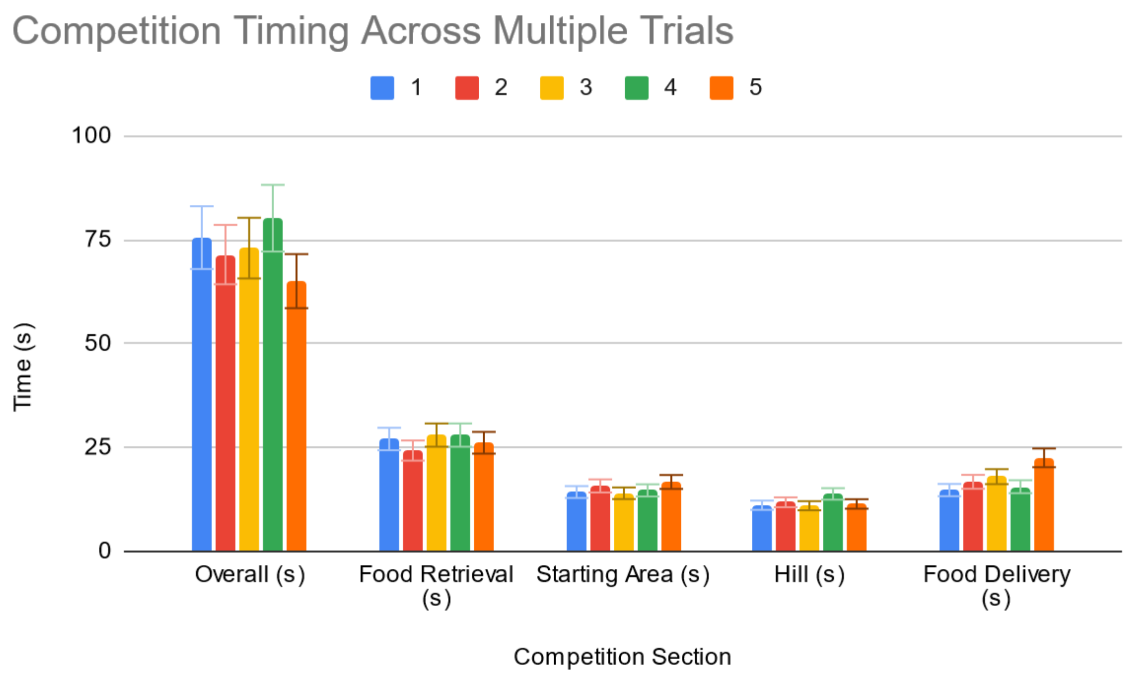


Figure 32: Graph of robot performance in different sections of the competition over five different trials. Error bars are also added to show the variance in each data set, where each range of the error bars excludes 10% of the data.

7.1.1 Food Retrieval

Food Retrieval (s)	27.06	24.28	28.62	28.00	26.16

7.1.2 Starting Area

Starting Area (s)	14.30	15.76	14.18	14.69	16.72

7.1.3 Mt. Bruin Hill

Hill (s)	11.12	11.83	11.29	13.84	11.43

7.1.4 Food Delivery

Food Delivery (s)	14.77	16.75	18.38	15.58	22.52

7.2 Overall Performance

Overall (s)	75.53	71.44	73.47	80.20	65.05

Overall the robot's performance is very high, resulting in a high speed of the robot as it traverses throughout the course. The robot has a very fast drivetrain owing to the high mechanical advantage of the gears for the wheel motors. Additionally, the grabber mechanism is very simple, allowing the servo to quickly grab and stow the patties. The combination of these two characteristics means that the overall speed of the robot is fast. Looking at the error bars from the error analysis it can be seen that the robot is fairly consistent in its performance. The uncertainty in the performance arises mainly as a result of wheel slip and inconsistencies in the track surface. These can cause the robot to overturn due to the tank steering method or cause the robot to catch and drift somewhat off the line. Additionally, differences in the placement of the patties for pickup can cause the times to change a significant amount.

8. Work Breakdown Schedule

8.1 Work Breakdown Schedule Diagram

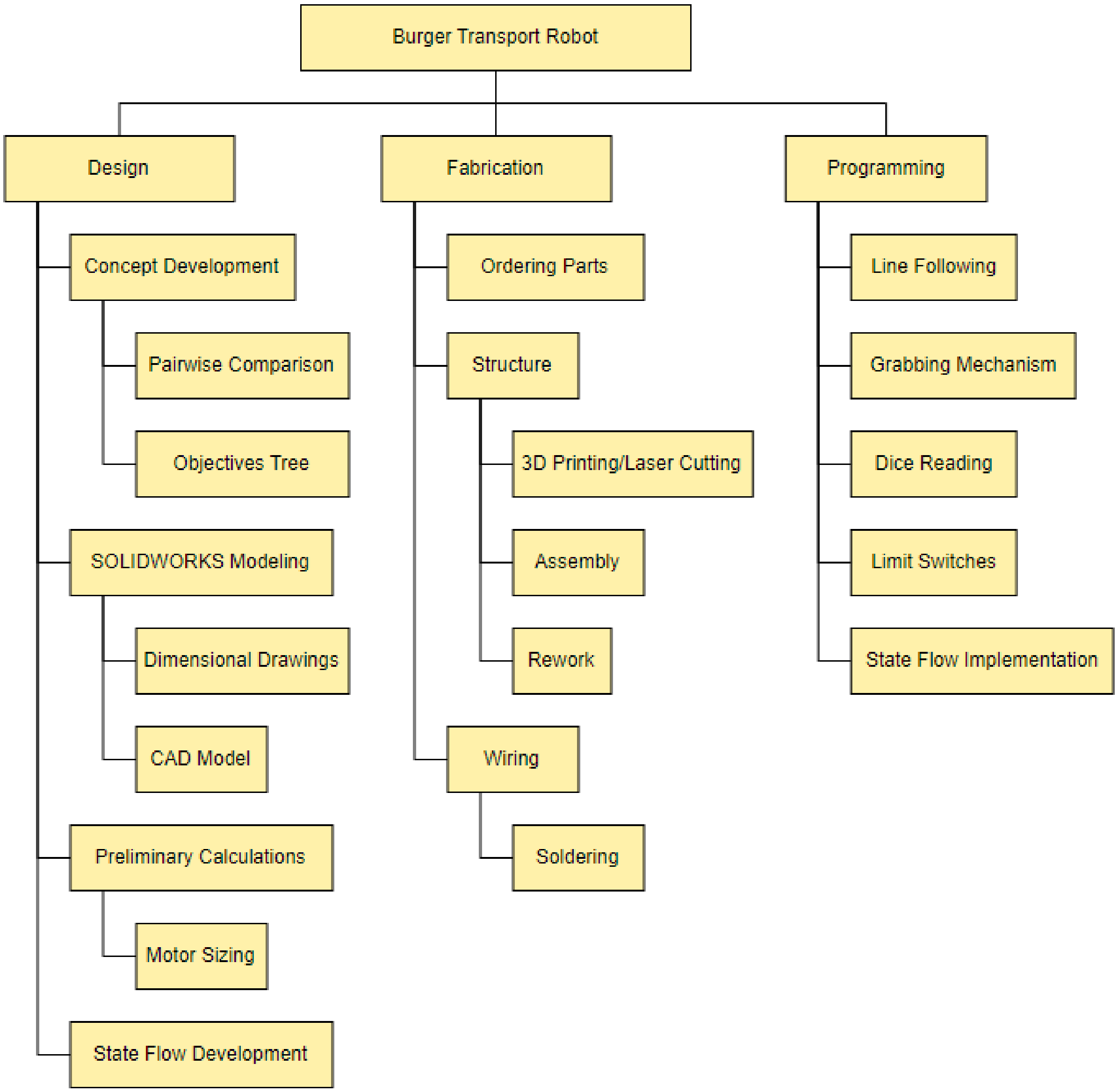


Figure 33: Work Breakdown Schedule Diagram

8.2 Work Breakdown Schedule Dictionary

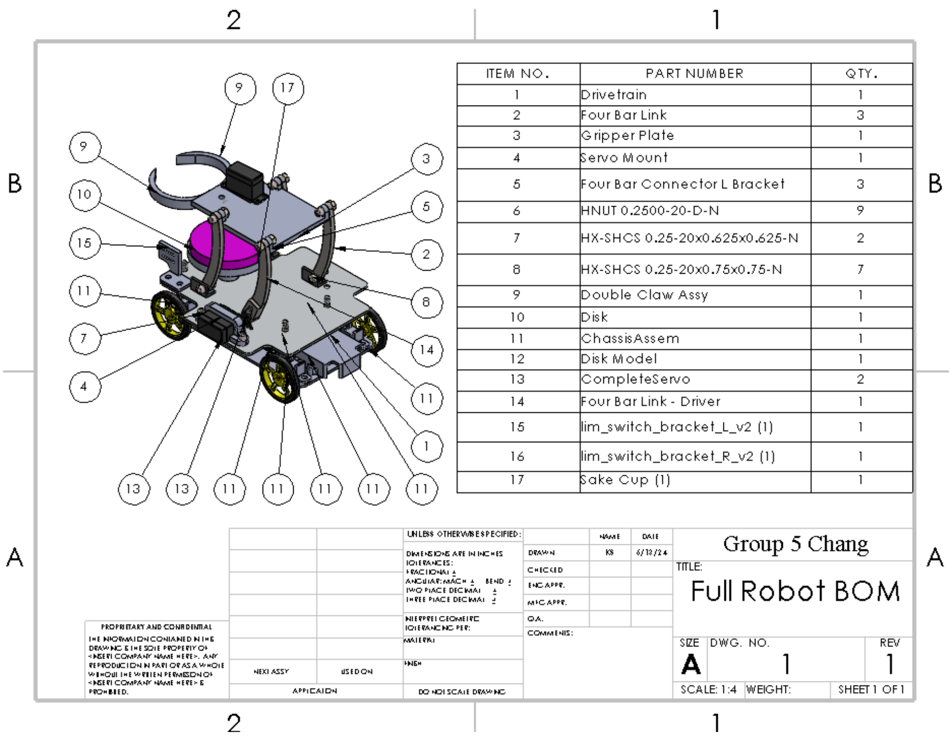
Table 10: Work Breakdown Schedule Dictionary

Item	Description
Concept Development	Brainstorming and hand-drawing potential concepts for a disk-retrieval robot
Pairwise Comparison	A chart used to rank the objectives of the robot by importance
Objectives Tree	A chart is used to choose which proposed design best aligns with the objectives weighted by importance as determined in the pairwise comparison
SolidWorks Modeling	The use of SolidWorks as a tool to simulate various factors in the design of a robot
Dimensional Drawings	The creation of SolidWorks drawings to determine how high the chassis must sit and how long the four-bar links must be to reach the counter
CAD Model	The creation of a full-scale model of all components of the assembled robot on SolidWorks
Preliminary Calculations	Calculations related to the velocity of the robot throughout the course as well as traction on the hill with consideration of the mass of the robot as determined by the SolidWorks model
Motor Sizing	Calculation of motor requirements for the robot to climb the hill
State Flow Development	Creation of a state flow diagram that represents an outline of the code to be implemented
Ordering Parts	Determination of the parts desired for purchase, drafting and submission of order request forms to the UCLA MAE department, correspondence with the staff, and picking up orders from the logistics center
3D Printing/Laser Cutting	Fabrication of custom parts in the maker space to be used for the structure of the robot
Assembly	Assembling the ordered parts with the custom fabricated parts to create the full structure of the robot
Rework	Redesign and fabrication of certain structural elements in the robot as issues arise regarding interference and stability
Wiring	Setup of electrical components of the robot
Soldering	Soldering of wires to solidify connections to sensors
Programming	Development of all code required for the robot to navigate throughout the course and complete the task

Line Following	Creating and tuning a PID controller that keeps the robot centered on the black line using the IR sensors
Grabbing Mechanism	Coding the four-bar servo and the claw servo in conjunction to accomplish the task of picking up and dropping off the disk
Dice Reading	Development of a function using camera data to read the dice rolls
Limit Switches	Coding of the limit switches to stop the robot when an obstacle is reached
State Flow Implementation	Implementation of the logic used for the robot to navigate the course and complete the various tasks when required

9. BOM and Cost Analysis

9.1 Assembly Drawings/BOM



ITEM NO.	PART NUMBER	QTY.
1	Drivetrain	1
2	Four Bar Link	3
3	Gripper Plate	1
4	Servo Mount	1
5	Four Bar Connector L Bracket	3
6	HNUT 0.2500-20-D-N	9
7	HX-SHCS 0.25-20x0.625x0.625-N	2
8	HX-SHCS 0.25-20x0.75x0.75-N	7
9	Double Claw Assy	1
10	Disk	1
11	ChassisAssem	1
12	Disk Model	1
13	CompleteServo	2
14	Four Bar Link - Driver	1
15	lim_switch_bracket_L_v2 (1)	1
16	lim_switch_bracket_R_v2 (1)	1
17	Sake Cup (1)	1

Figure 34: Assembly drawing and BOM

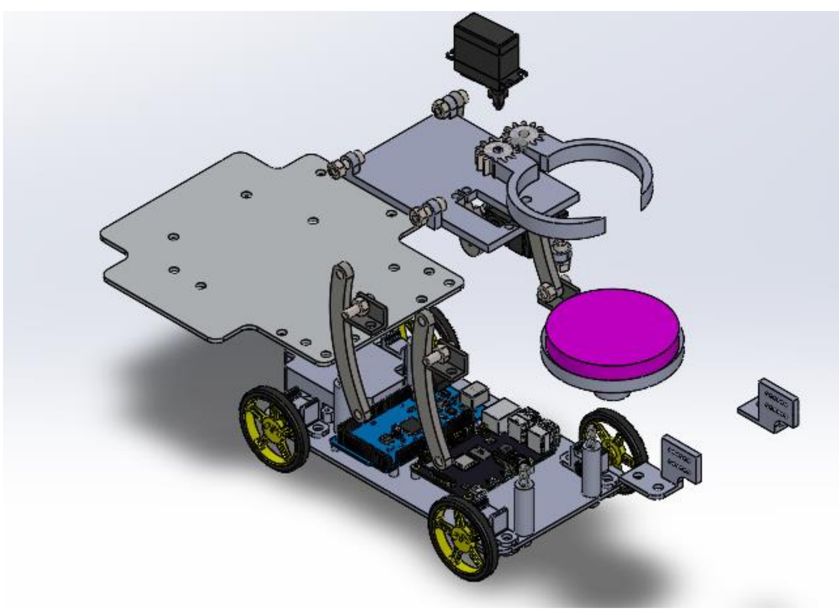


Figure 35: Exploded View of the Robot

9.3 Final Cost Analysis

In this section, the free components are listed first, followed by the components that were paid for. For the free components, the source is listed.

9.3.1 Material Costs

Table 11: List of free materials used

Part	Manufacturer/Material	Quantity	Cost	Source
Arduino Mega + cord	ELEGOO	1	0	TA donation
Mini Portable Charger+cord	Miadi	1	0	Group member donation
Mini Portable Charger	SIXTHGU	1	0	Group member donation
Chassis base	Acrylic	1	0	Lasercut
Chassis top	Acrylic	1	0	Lasercut
Chassis Spacer	PLA	4	0	3D printed
Four-bar long L bracket	PLA	1	0	3D printed
Four-bar short L bracket	PLA	1	0	3D printed
4 bar link	PLA	3	0	3D printed
4 bar servo link	PLA	1	0	3D printed
Gripper plate	PLA	1	0	3D printed
Left Claw	PLA	1	0	3D printed
Right Claw	PLA	1	0	3D printed
Disk Holder	PLA	1	0	3D printed
Camera Holder	PLA	1	0	3D printed
Servo Mount	PLA	1	0	3D printed
Wheel Motor Mount	PLA	4	0	3D printed
IR Sensor Mount	PLA	1	0	3D printed
Limit Switch Mount L	PLA	1	0	3D printed
Limit Switch Mount R	PLA	1	0	3D printed
Arduino Spacer	PLA	3	0	3D printed

Table 12: List of purchased materials

Part	Vendor/Manufacture	Unit Cost	Quantity	Total Cost
Wheels x2	Pololu	5.75	3	17.25
Wheel motor	Pololu	18.35	5	91.75
Single DC Motor Driver	Pololu	4.95	5	24.75
Servo for gripper	Pololu	11.95	1	11.95
Servo for four bar	Pololu	22.95	2	45.90
Raspberry Pi Pico Cable	Amazon	6.99	1	6.99
Rechargeable Battery Pack	Pololu	15.15	1	15.15
5V voltage regulator	Pololu	12.95	1	12.95
5V battery pack	Amazon	19.99	1	19.99
Large IR Sensor Array	Pololu	22.90	1	22.90
JR Connectors (M)	Pololu	4.25	2	8.50
JR Connectors (F)	Pololu	2.75	1	2.75
22 AWG Wire	Amazon	14.99	1	14.99
Limit Switch	Pololu	1.79	3	5.37
Raspberry Pi Camera Board	Adafruit	29.95	1	29.95
Steel Hex Nut M3 x100	McMaster-Carr	2.62	1	2.62
Socket Head Screw M3x0.5mm 16mm Long x25	McMaster-Carr	$5.31	1	$5.31
Nylon Plastic Washer for 1/4" Screw x25	McMaster-Carr	$7.54	1	$7.54
Socket Head Screw 1/4"-20 3/4" Long x50	McMaster-Carr	$11.38	1	$11.38
Steel Hex Nut Grade 5 1/4"-20 x100	McMaster-Carr	$8.95	1	$8.95
Socket Head Screw 1/4"-20, 5" Long x5	McMaster-Carr	$8.50	1	$8.50
Socket Head Screw 1/4"-20, 2-1/4" Long x10	McMaster-Carr	$5.97	1	$5.97
Socket Head Screw M2.5x0.45mm 14mm Long x50	McMaster-Carr	$7.52	1	$7.52
Steel Hex Nut M2.5 x 0.45 mm x100	McMaster-Carr	$2.22	1	$2.22
Total taxes	$37.95			
Total shipping	$26.05			
Grand Total	**$451.34**			

The sum of expenditures came out to $451.34, which is less than the budget requirement of $500. Note that there are various items in the material costs that are not in the BOM such as the camera and various extra parts, which were not included in the final iteration of the robot.

9.3.2 Labor Costs

This project lasted for 20 weeks. During the first 15 weeks, each of the 5 group members spent approximately 2 hours per week on the robot. For the final 5 weeks, much more time was spent. For the full duration of the project, the average time per week of each group member is estimated as 4 hours. Using the average engineering salary of $36 per hour, this comes out to $14,400 in labor costs.

10. Design Requirement Satisfaction

Table 13: HLDR Satisfaction Table

HLDR*	Description	Satisfied?
1	The robot shall follow a black line through the entire course	Yes. With PID control we were able to get accurate line following at a fast speed
2	Robot shall be capable of autonomously transporting two 3” diameter disks in tandem.	Yes. The disks stack on top of each other on the pedestal and are held in place by the grabber.
3	The robot shall be capable of retrieving the two disks from two locations in a specified order from a platform 14.75 cm tall.	Yes, but the robot can have trouble navigating the junctions near the pickup spot
4	The robot shall be capable of dispensing the disks in a stack on a platform 14.75 cm tall.	Yes. The robot can grab the bottom disk and drop off the entire stack together.
5	The robot shall be capable of transporting the disks up and down a slope of 10 degrees.	Yes. The robot is programmed to speed up once it reaches the ramp so that enough power is delivered to climb the hill.
6	The robot shall stop for a dynamic obstacle	Yes, the robot touches the obstacle with the limit switches and immediately stops.
7	The robot shall survive a drop test	Yes. Parts were securely fastened and chosen to be thick enough to withstand damage
8	The robot shall cost at most 500 dollars.	Yes, not including labor costs. Total cost of parts came out to $451.34
9	The robot shall recognize dice to retrieve the correct disks	No. We focused time on the required tasks and did not have time for this bonus task. If we had more time, we would have explored dice reading programs to implement.
10	The robot shall be capable of carrying 3 disks	Yes. We designed the four bars to be tall enough to stack a third disk.

11. Conclusion

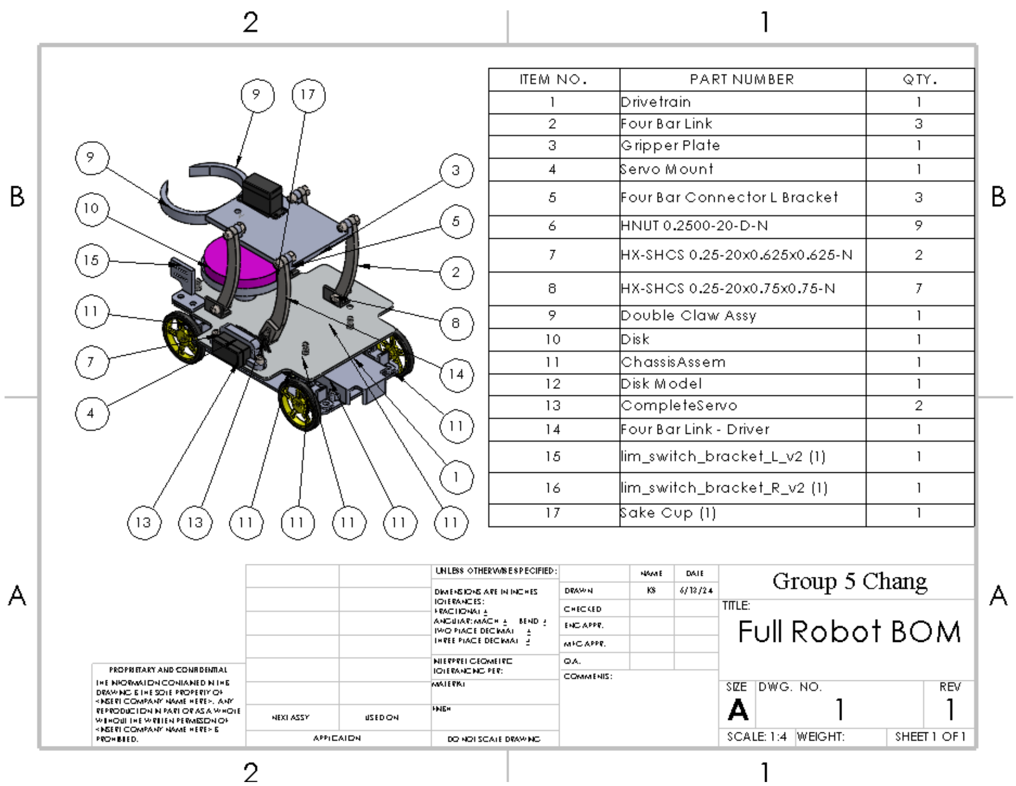
The final design of the food-transporting robot features a durable and compact structure built from laser-cut acrylic and 3D-printed components. It uses a four-bar linkage to achieve the vertical and horizontal motion needed to intake and dispense burgers with a geared claw mechanism. The robot includes a drivetrain with four gear motors, 13 IR sensors for line following, and two limit switches to aid in positioning. This design allows our robot to complete the delivery task within two minutes. Resource allocation allowed us to keep our project within budget. Regarding project management, it became clear that early and continuous communication within the team and with TAs was important to move the project forward efficiently. Testing took longer than expected and we were left with less time. We learned to make a more rigid schedule of deadlines in the future so that all steps of the project are given an adequate amount of time to complete. This project emphasized the importance of integrating mechanical simplicity with robust software control in engineering projects. The experience we gained from this project will be helpful for our future endeavors in robotics and automation.

12. References

[1] ServiPlus: https://airpuria.com/products/servi-plus-the-service-robot; accessed Feb. 1st,, 2024
[2] Starship: https://www.starship.xyz/; accessed Feb. 28th, 2024
[3] Four Bar Simulator: https://dynref.engr.illinois.edu/aml.html; accessed Feb. 28th,, 2024
[4] Coefficient of friction: https://www.echemi.com/community/what-is-the-coefficient-of-friction-for-rubber-on-wood_mjart2204131570_131.html; accessed Feb. 28th, 2024

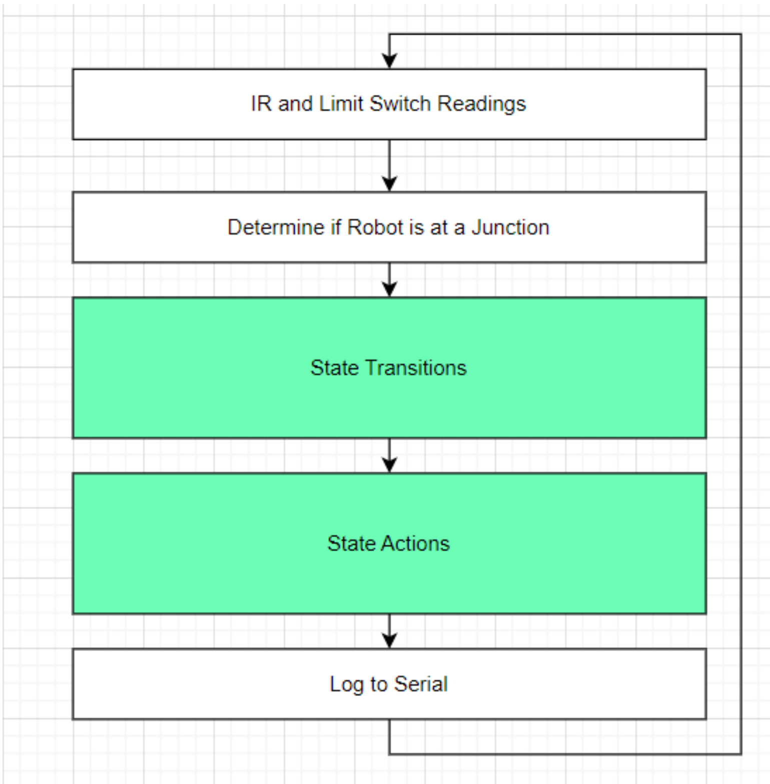
13. Appendix

13.1 Engineering Drawings



ITEM NO.	PART NUMBER	QTY.
1	Drivetrain	1
2	Four Bar Link	3
3	Gripper Plate	1
4	Servo Mount	1
5	Four Bar Connector L Bracket	3
6	HNUT 0.2500-20-D-N	9
7	HX-SHCS 0.25-20x0.625x0.625-N	2
8	HX-SHCS 0.25-20x0.75x0.75-N	7
9	Double Claw Assy	1
10	Disk	1
11	ChassisAssem	1
12	Disk Model	1
13	CompleteServo	2
14	Four Bar Link - Driver	1
15	lim_switch_bracket_L_v2 (1)	1
16	lim_switch_bracket_R_v2 (1)	1
17	Sake Cup (1)	1

13.2 Stateflow Charts



13.3 Code

https://github.com/gettyv/ROKUSHO

Main Routine
https://github.com/gettyv/ROKUSHO/blob/d444f2a53ab4f90fa68349161edad69b2deab038/src/main.cpp