Beachbots MQP

A Major Qualifying Project

submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfilment of the requirements for the

degree of Bachelor of Science

By: Sebastien Casimir, Christopher Grier, Everett Johnson, Benjamin Seibert, Sean Tidd

Date: May 5th, 2020

Report Submitted to:

Professors Nicholas Bertozzi and Brad Miller

Worcester Polytechnic Institute

Table of Contents

Table of Figures

Table of Tables

Authorship

All members of this team collaborated and contributed to the project and report equally. Sections of the report were divided among the members of the group to work on individually. Each section was written by an individual team member and then edited by the remaining team members. After individual edits, the entire team edited together, thus making each section a representation of our collective work.

Abstract

Beaches across the globe are being polluted by human trash. While there are solutions to remove pollution from beaches, they mostly entail large machines that are not environmentally friendly and require a human operator. The goal of this MQP is to develop a robotic solution to clean these beaches effectively. This project aims to develop building blocks for later MQPs including the development of one small robot, a base station with a beach mapping system, and the communication methods between them.

Chapter 1: Introduction

Currently, five trillion pieces of unwanted and harmful plastic debris are floating in the oceans around the world (McCarthy, 2018). This debris often washes up on beaches endangering the wildlife and humans that interact with these beaches. The impact of beach pollution is not just an environmental hazard, but also an economic threat due to reduced tourism. Tourism with respect to beaches is heavily influenced by the cleanliness of the area. If trash is continually removed from these beaches, it could result in a decreased threat to wildlife and a reduction of an economic threat.

While there are solutions to this problem, they mostly entail large machines that are not environmentally friendly and require a human operator. An example of this is the Surf Rake, a large machine that is dragged behind a tractor, raking debris into a sifting mechanism. This system requires a human operator to drive the machine over each inch of the beach that needs to be cleaned. The advantage of a system like this is its ability to clean all forms of trash, large or small. In addition, these systems are efficient for trash collected per hour, some being able to clean up to nine acres an hour (SURF RAKE SPECIFICATIONS, 2019). However, a system like this is not environmentally friendly and can remove some of the natural debris from the beach. A more common form of beach clean-up is a volunteer-based service. This activity is more environmentally friendly as people pick up the garbage by hand and no harm is done to the beach ecosystem because only the trash is removed, not the important natural debris. Whether it is volunteers picking up trash by hand or tractors raking over sand, a human is heavily involved in the process, which is labor intensive and inefficient. In some cases, the desired outcome is the complete removal of debris. However, on beaches not near highly populated areas, this extra

work is detrimental to the environment.

The goal for this project is to have a group of robots work together to pick up trash on the beach. This goal will be completed by developing a scalable robotic workforce that will use computer vision to recognize trash and custom collection mechanisms to sift through the sand. In addition, a large robot will be used to map the beach and organize the robotic workforce. In a future implementation, the large robot will be able to collect trash and either empty independently or have an operator unload the trash. A successful implementation of this system would result in a reduction in the carbon footprint and the need for human labor in relation to past solutions. This system could be in theory applied to locations other than beaches including parks, and recreational locations.

The problem that this project seeks to address is the difficulty to clean beaches. Pollution of coastal environments degrades and destroys unique beach habitat used by humans, animals, and plants. Accurately cleaning beaches is important as litter can affect both the health of those in the ecosystem and tourist activity. This robotic system aims to precisely remove particular types of trash compared to the larger more indiscriminate machines used today. This project focuses primarily on public beaches since these types of beaches have heavy traffic and are easily accessible.

Chapter 2: Background

2.1 Introduction

Beaches across the globe are being polluted by human trash. This trash can harm the environment and the creatures that inhabit it. Cities and states combat beach pollution by spending upwards of several million dollars a year on all beaches by utilizing either a large workforce of volunteers or heavy machinery (Parker, 2018). These solutions are often ineffective and not environmentally friendly. A more effective solution must be found to remove beach pollution. The problem that this project is trying to address is the difficulty to clean beaches. Pollution of coastal environments degrades and destroys unique beach habitat used by humans, animals, and plants. Achieving this is important as litter can affect both the health of those in the ecosystem and tourist activity. This project focuses primarily on public beaches since these types of beaches have heavy traffic and are easily accessible.

2.2 Effect on Tourism

Every year, Americans take more than nine hundred million trips to coastal areas and spend about $44 billion annually during these trips (US EPA 1996). A survey concluded that two characteristics of beaches that visitors valued most were that the beach was clean of debris and the quality of the water (NOAA, n.d.). Therefore, unsanitary beaches can result in a loss of interest from potential tourists in turn resulting in a loss of tourism revenue for the state.

2.3 Effect on Health

The polluted waters not only harm tourist interest, but also their health through many waterborne diseases. Numerous health concerns, such as gastrointestinal problems, directly associated with beach use result in millions in medical bills. In Orange County, California, there occur roughly 36,778 gastrointestinal (GI) episodes per year from polluted beaches (Dwighta et al, 2005). With GI treatments costing approximately $36 each, Orange County residents pay over $1.3 million per year on these treatments alone. Lacerations are also a threat to public safety on beaches. Lacerations to beach goers can result from sharp objects being left in the sand and stepped on. A study at Lorne beach, Victoria showed that out of the 211 beach injuries during the summer holiday period, 37% were lacerations and beach litter directly caused 19% of the lacerations (Grenfell, Ross, 1992). Beach pollution is a direct threat to public health and needs to be addressed promptly.

2.4 History of Beach Pollution

Concerns over beach pollution began in the 1960's as anecdotal reports of seabirds and marine life being entangled, and ingesting plastics started to surface. Beach litter specifically came under scrutiny in 1972 when a researcher debunked the notion that beach users were responsible for most of the litter (Bergmann, Gutow, & Klages, 2015). Most of the litter actually comes from processing plants. Following this study, Doctor Anthony Cundell, a researcher who built on the work of Bergmann assessed the amount of litter washing ashore over one month. These and similar early studies found how easily litter can travel between beaches and the dangers of this high accumulation rate (Bergmann, Gutow, & Klages, 2015). Outside of the scientific realm, environmental groups began hosting beach clean-ups, such as the International

Coastal Cleanup. One of the most helpful procedures when cleaning these beaches is documentation. The director of the Plastic Soup Foundation, Maria Westerbos says, "a cleanup only makes sense when people take photos of everything they find — so we can investigate where it's coming from". Therefore, beach cleanup events like the International Coastal Cleanup have volunteers go to beaches and manually pick up trash. In some cases, they also record the items they collect on data cards (International Coastal Cleanup, 2019).

Currently, the most common type of beach cleaning is known as ‘grooming’ and involves removing litter as well as natural debris. While this cleaning can be done manually, it is more typical to see large machinery utilizing rakes or sieves to separate sand from debris. One such machine, the Surf Rake weighs 3,800lbs and can clean up to nine acres an hour (SURF RAKE SPECIFICATIONS, 2019). Although this system is capable of quickly cleaning a large area, it also clears all the natural debris, such as sea glass and driftwood, known as wrack. The wrack is a source of food and habitat for many organisms, and its removal could affect the ecosystem’s food chains leading to the loss of species (Zielinski, Botero, & Yanes, 2019). In addition, wrack removal is assumed to alter the dynamics of sediment accumulation and retention, leading to the loss of sandy beaches (Zielinski, Botero, & Yanes, 2019). Aside from the physical damage to the beach ecosystem, the large tractor required to move this Surf Rake contributes a large carbon footprint. A Nebraska OECD (Organisation for Economic Co-operation and Development) tractor test found that a typical tractor at 16mph used 8.01 gal/hour of fuel (Hanna, n.d.). This figure translates to roughly 2 mpg, a mere 8% of the average mpg of a car. While the carbon footprint from the many cars driving to the beach will be much greater, any reduction in carbon usage is important.

2.5 Causes of Pollution

Ocean pollution is the introduction of toxic materials such as plastic, oil, chemicals, agricultural waste, and industrial waste into the ocean. There can be several causes of ocean pollution, but the leading causes include sewage, toxic chemicals from industries, nuclear waste, thermal pollution, plastics, acid rain, and oil spillage (US EPA 2019).

Trash and other solid materials that reach rivers, bays, and oceans eventually wash up on beaches. It includes plastic bags, bottles and cans, cigarette filters, bottle caps, and lids (US EPA 2019). Other sources include people at the beach leaving behind their trash, and fishermen losing or discarding fishing nets and lines in the ocean.

One of America's most widespread and costly problems is excess nitrogen and phosphorus in the air and water. Nitrogen and phosphorus support the growth of algae and aquatic plants, which provide food and habitat for fish, shellfish and smaller organisms that live in water (US EPA 2019). However, when too much nitrogen and phosphorus enter the environment, usually from a wide range of human activities, the air and water can become polluted, resulting in serious environmental and human health issues, and affecting the economy (US EPA 2019). Animal manure, excess fertilizer applied to crops and fields, and soil erosion make agriculture one of the largest sources of nitrogen and phosphorus pollution in the country.

2.6 Major Stakeholders

Many different groups are interested in cleaning up beaches, such as local governments, tourists, wildlife, and private beach cleaning companies. Local governments and tourists are a symbiotic group of stakeholders as local governments rely on revenue from tourism and tourists care about the aesthetic and public safety of beaches. Some of Massachusetts' $220,000 beach

budget is allocated to the maintenance and cleaning of beaches (Commonwealth of Massachusetts, 2019), but the state also relies on volunteer efforts to keep the coastline clear, such as Coastsweep (Coastsweep, n.d.). Wildlife is also a stakeholder as trash and debris can be toxic to marine life such as sea turtles, birds, and mammals (Impacts of Mismanaged Trash, nd). Private beach cleaning companies, such as Sandman Beach Cleaning, are a stakeholder as they rely on dirty beaches for their business.

2.7 The Solution

The goal of this project is to create robots that can clean beaches of small debris. A single human operator will be required to set up the robots on the beach as well as collect the trash from them at the end. While the robots are cleaning, they will need little to no interaction from the operator. Ideally, the target beach will be void of beachgoers as there are no plans for active avoidance of humans. They will be able to traverse clear beaches with minor pollution and be able to map the beach terrain to ensure the entire beach was cleaned. Beaches with severe pollution, debris in the water, or debris that is larger than a soda can will not be cleaned by this project. Beaches with softer sand will be the focus because it is easier to remove trash on soft sand than hard sand. The robot system will be able to identify trash with reasonable accuracy while also being able to retrieve it on its first attempt most of the time. Natural debris on the beach such as branches and shells will be avoided. The robot system should be scalable so that additional robots can be added to the group with little effort. In the future, this project can be continued by other MQPs to incorporate new robots to pick up different types of trash.

Chapter 3: Methodology

The objectives for this project are shown in Appendix D.

3.1 Mechanical Design

3.1.1 Sand Locomotion Selection and Design

The first task was to design a system that would be able to drive on sand. Tank treads were determined to be the best system for driving in sand for several reasons. Using treads provides higher traction on sand than wheels (Hughes 2013). Treads also slip less than wheels due to the amount of contact they have with the ground. Wheels are more likely to sink in the soft sand than treads if they are too narrow.

During testing of the first design, the stock tread design proved to be ineffective, as sand would stick in between the ribs of the treads. This would clog the system and prevent movement. To solve this issue, a different tread design was implemented, like the treads used on the Moonraker robot shown in Figure 1 (Markhor, 2017).

Figure 1: Moonraker Tread System

These treads used spaced C channel cut offs on a chain to allow sand to slip between the large crevices while moving. However, the system that was designed for this project proved to be ineffective at driving due to how the chain was connected to the C channel. Through more research, a solution was found where the treads were completely 3D printed and connected by bolts. The treads have teeth that fit into the drive sprocket's holes, and a connected gear drives this sprocket to the drive pinion with a 50:11 gear ratio. The tread design and gearing can be seen in Figure 2 and 3, respectively.

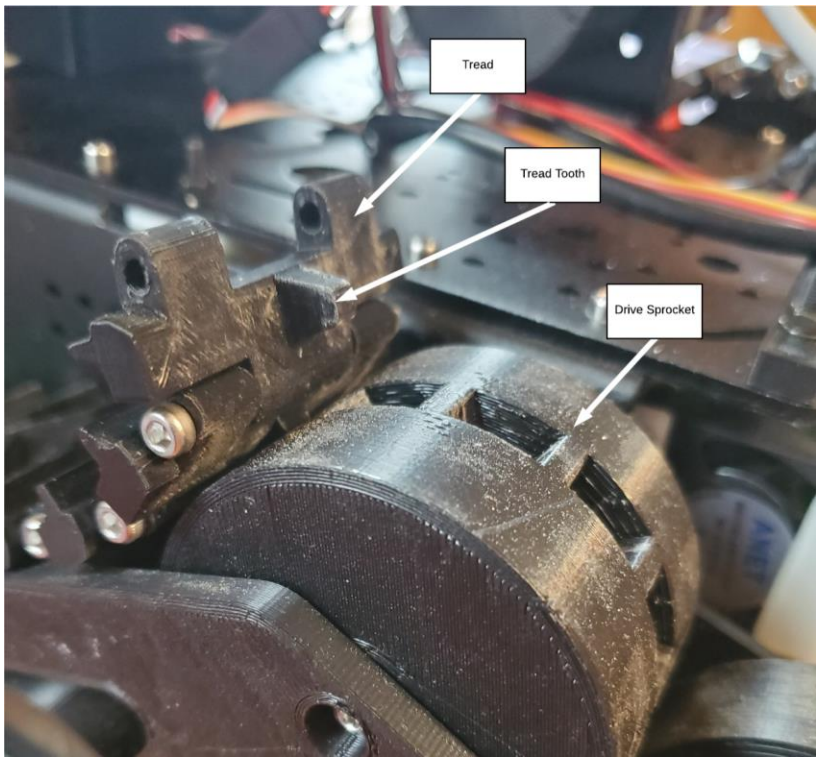


Figure 2: Tread and Drive Sprocket

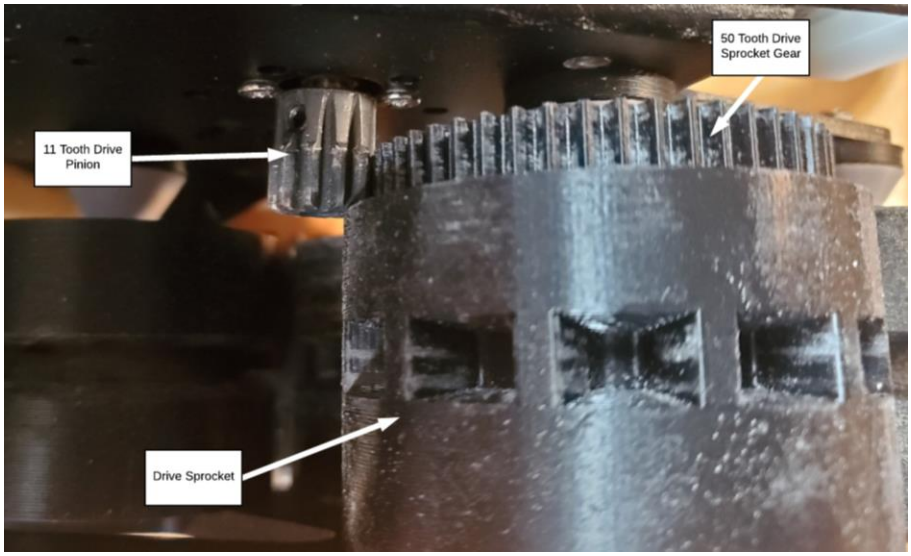


Figure 3: Drive Pinion and Sprocket

The sprocket and runner wheels have captive bearings that allow for smooth rotation. The final feature of this drive train is an external plate which connects the runners and drive sprocket adding rigidity to the system. This system was based on the Thingiverse model and this

implementation can be seen in Figure 4 (Bryant87, 2017). An assembly of the system can be seen in Appendix F.

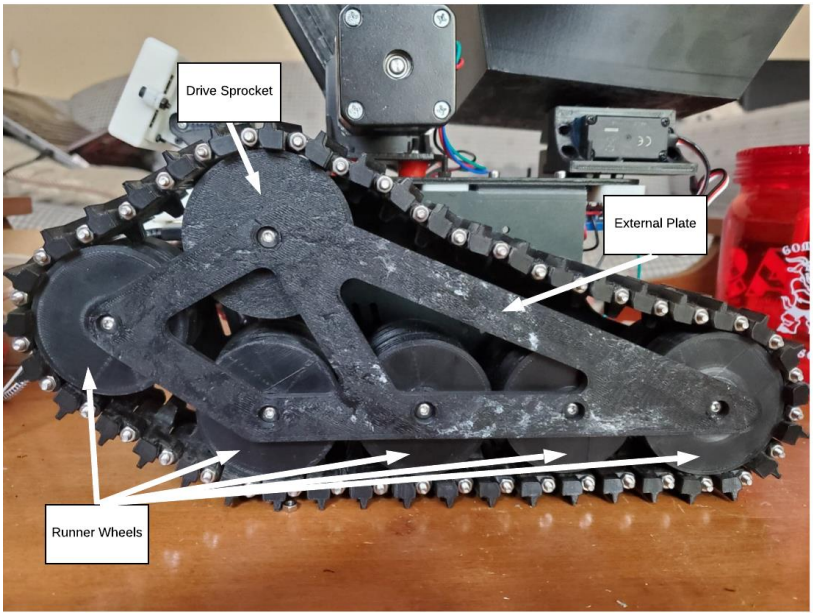


Figure 4: Drive Train

The tread design changed over the course of the project due to testing with the robot driving in sand. The design shown in Figure 5 was created to displace more sand than previous designs and elevate the robot. With these factors, the drive train sits on top of the sand, not allowing sand to clog the drive train.

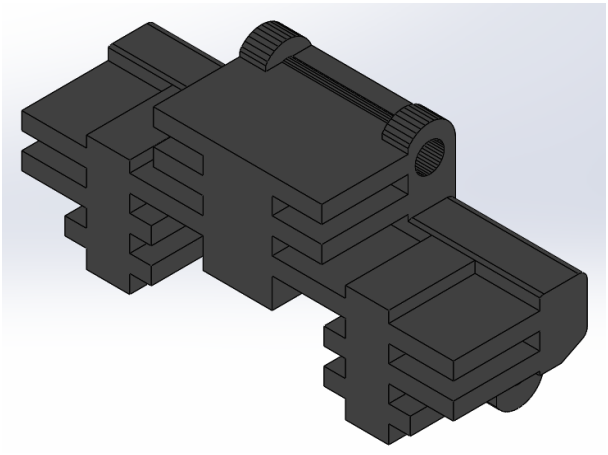


Figure 5: Tread Design

3.1.2 Robot Body Selection

Due to the challenges of designing two robots from scratch, robust robotic platforms were searched for as a starting point for the design. Requirements for both necessary platforms are listed below:

- Reasonably priced
- Can drive in beach environments
- Have space for additional motors and sensors

The Smallbot was found from a selection across various websites, first, due to robotic systems smaller than 300mm being prevalent and inexpensive. The chosen platform can be seen in the figure below.

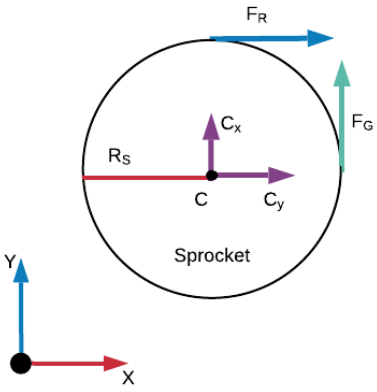
Figure 6: Smallbot Platform

This robot has a full aluminum body with plastic treads held together with small metal rods. The specifications claim that the robot can bear 3kg of payload, but due to later modifications, it ultimately could support 5kg. The many holes and tie down points on the top of the body allow for easier modular design. The included motors specifications are shown in the figure below. A single motor has a maximum torque of 0.93N-m, and a no-load speed of 100rpm. To calculate the torque requirement of the drive train, the force analysis and FBD

shown in Figures 7 and 8 were used. See assumptions related to this force analysis in Appendix E. The estimated force on the treads (F_R) was derived from the weight of the robot, assuming that the treads have perfect adhesion to the ground, they will not experience more force than the weight of the robot.

Given:
R = 11 / 50
R_S = 29 mm
F_R = 20.4 N

Legend:
F_R : Resistive Force on Sprocket from treads
F_G : Force on Spocket due to motor
R_S : Radius of Sprocket
R : Gear Ratio
C : Sprocket Axle
C_y: Force on C in the Y axis
C_x: Force on C in the X axis

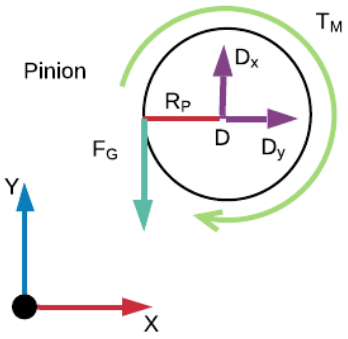


Equations of Equalibrium:

$$\sum M_C = 0 = F_G(R_S) - F_R(R_S)$$
$$F_G = 20.4N$$

Given:
Max Motor Torque (MMT) = 0.930Nm
R = 11 / 50
R_P = 6 mm

Legend:
T_M : Torque of the Pinion Motor
F_G : Force on Spocket due to motor
R_P : Radius of Pinion
R : Gear Ratio
D : Sprocket Axle
D_y : Force on D in the Y axis
D_x : Force on D in the X axis



Equations of Equalibrium:

$$\sum M_D = 0 = F_G(R_P) - T_M$$
$$T_M = 0.12Nm$$

Figures 7 and 8: FBD of Drive Train

A required torque of 0.12 Nm was found, allowing the motors to be driven at 87 rpm. This speed was chosen because of the resulting position along the speed and efficiency curve in addition to being at the torque requirement. This value was extracted by using the motor data graph below in Figure 9, where it is represented by the black dashed line. This allows the motor to be driven at a reasonable efficiency while remaining near its maximum speed.

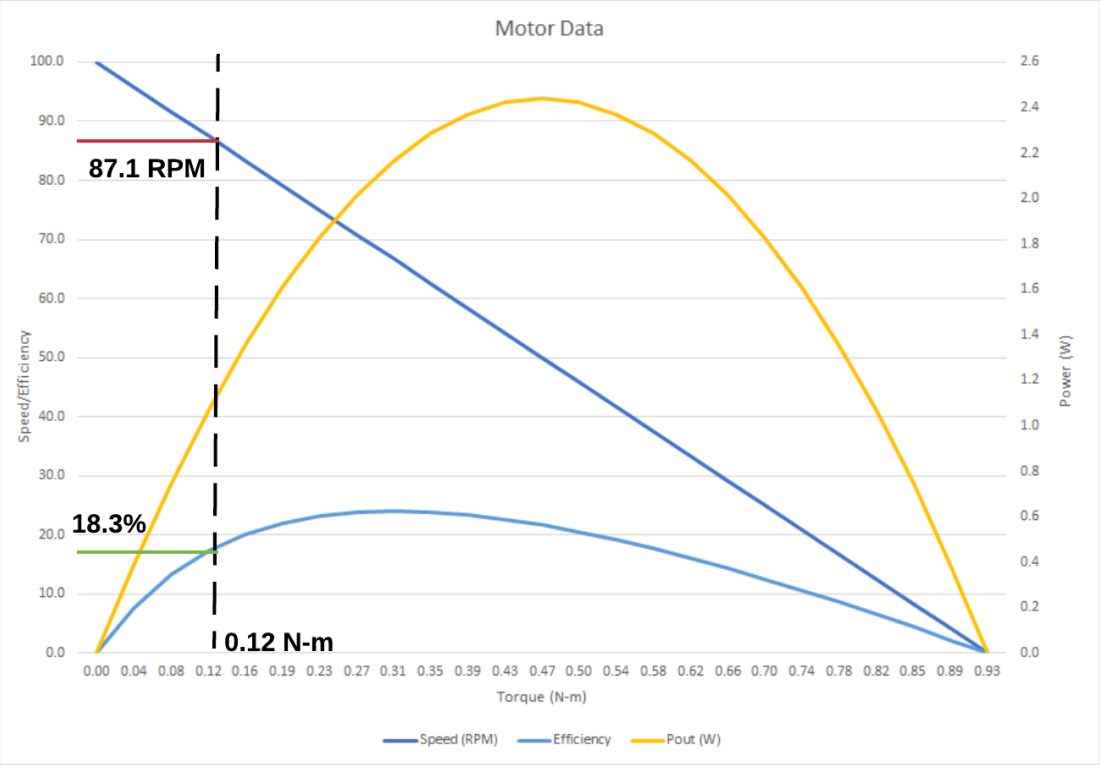


Figure 9: Motor Data for Smallbot Drive Motors

The Basebot posed a larger problem as platforms significantly larger than the 300mm Smallbot cost considerably more. Thus, instead of buying a platform including drive train and motors, a garden cart was chosen. This cart is pictured below in Figure 10.

Figure 10: Garden Cart

In addition to the lower price point, this cart came equipped with 10” wheels and Ackerman steering. To add these components to another base it would cost a considerable amount of money and time. Due to the larger wheels, this cart will not require the use of treads to traverse through sand.

3.1.3 Collection Mechanism

For the final design, a metal claw gripper was purchased and modified, shown in Figure 10.

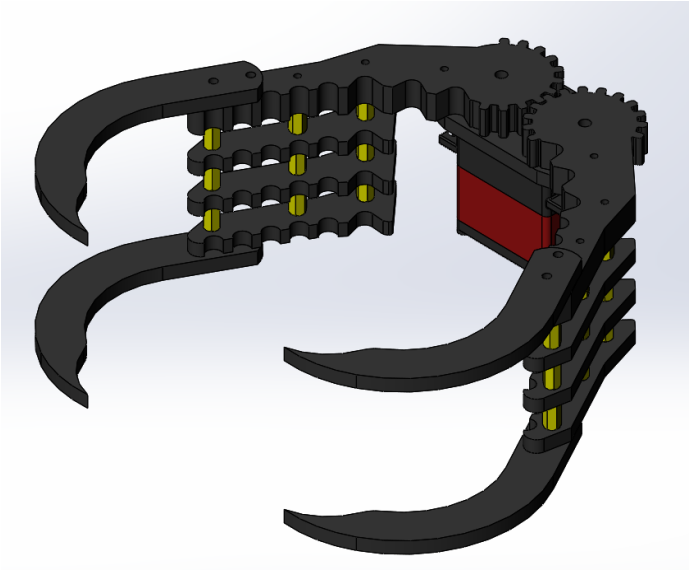
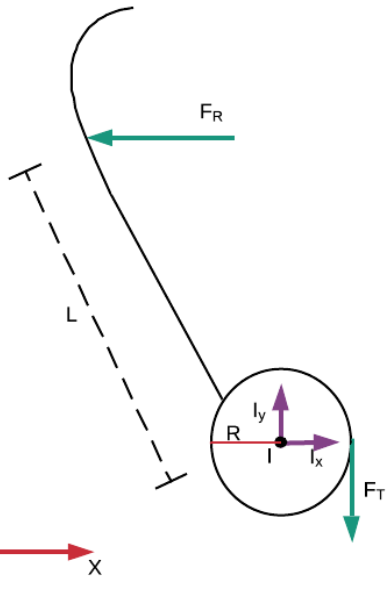


Figure 11: Claw Gripper

This claw utilizes a 30 kg-cm (2.9 N-m) servo motor to actuate the geared finger assembly to grab the trash. The gripper can exert approximately 18 N of force at the tip of its fingers shown in Figures 12 and 13. The value was found from performing force analysis on the gripper and the assumptions for this can be seen in Appendix E.

Given:
L = 0.092 m
R = 13mm

Legend:
L : Length of gripper finger
F_R : Resistive Force at the point of contact for finger
F_T : Force from other finger
R : Radius of Gear
I: Servo axle
I_y: Force on I in the Y axis
I_x: Force on I in the X axis

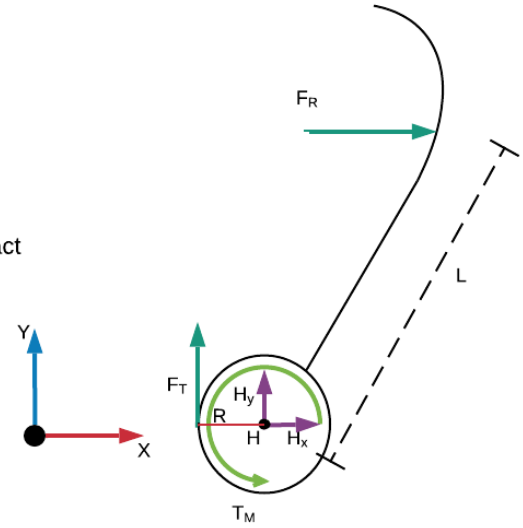


Equations of Equalibrium:

$$\sum M_I = 0 = F_R(L)cos(30) - F_T(R)$$
$$F_T = 6.13F_R$$

Given:
L = 0.092 m
T_M = 2.94 Nm
R = 13 mm

Legend:
L : Length of gripper finger
F_R : Resistive Force at the point of contact for finger
F_T : Force from other finger
R : Radius of Gear
H: Servo axle
H_y: Force on H in the Y axis
H_x: Force on H in the X axis



Equations of Equalibrium:

$$\sum M_H = 0 = T_M - F_T(R) - F_R(L)cos(30)$$
$$F_R = 18.4N$$

Figures 12 and 13: Gripper Force Analysis

Each finger is separated from each other by three standoffs, which allows the fingers to move through sand more easily than other collector designs. The aluminum construction is also superior to the other considered ABS plastic designs. The gripper mechanism exists at the end of a 2DOF arm. This arm contains a geared stepper motor for the shoulder joint and a 2.0 N-m servo as an elbow joint.

The figure below shows a basic force analysis of the arm with derived torque requirements of 2.14 Nm and 1.15 Nm for the shoulder and elbow motors, respectively. The requirements are met by maximum torque outputs of the arm motors being 4 Nm and 1.96 Nm for the shoulder (stepper motor) and elbow (2.0 N-m servo) joints, respectively. The F_{Gg} force was derived from the weight of the gripper (252g) and F_P from the weight of full soda can (400g). The force F_{Ge} was found from the weight of the elbow motor (60g) and weight of the link L_2 (54g). The weight of a can is variable in the beach environment, as the can could have an

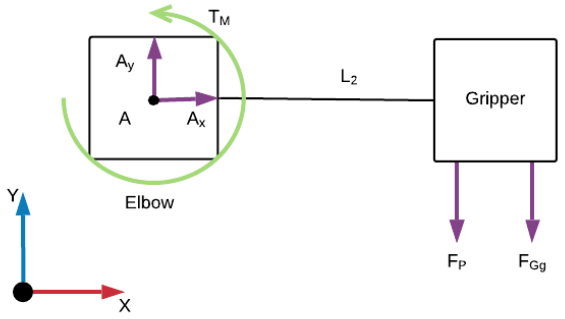
amount of sand or water inside of it. Due to this, the elbow motor was chosen due to its safety factor of 1.7. The force analysis for the collector arm is shown below in Figures 14 and 15. The force analysis shown in Figures 14 and 15 are determined using the Assumptions listed in Appendix E.

Given:
Elbow Motor Max Torque: 1.96 Nm
L_2 = 0.180 m
F_P = 0.252 kg * 9.81 m/s^2 = 2.472 N
F_{Gg} = 0.400 kg * 9.81 m/s^2 = 3.924 N

Legend:
L_2: Link 2 Length
F_P: Force due to the weight of the payload
F_{Gg}: Force due to the weight of the Gripper and weight of L2.
T_M: Torque of the Motor
A: Elbow axle
A_y: Force on A in the Y axis
A_x: Force on A in the X axis

Equations of Equalibrium:

$$\sum M_A = 0 = T_M - L_2(F_P + F_{Gg})$$
$$T_M = 1.15Nm$$

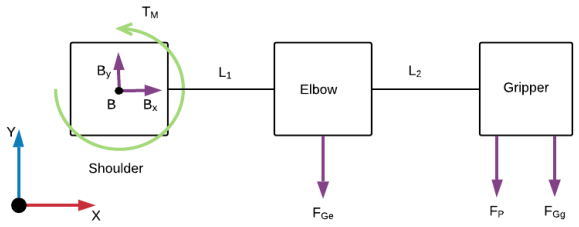


Given:
Shoulder Motor Max Torque: 4 Nm
L_1 = 0.130 m
L_2 = 0.180 m
F_{Ge} = 0.114 kg * 9.81 m/s^2 = 1.118 N
F_P = 0.252 kg * 9.81 m/s^2 = 2.472 N
F_{Gg} = 0.400 kg * 9.81 m/s^2 = 3.924 N

Legend:
L_1: Link 1 Length
L_2: Link 2 Length
F_{Ge}: Force due to weight of Elbow motor and weight of L_1
F_P: Force due to the weight of the payload
F_{Gg}: Force due to the weight of the Gripper and weight of L_2.
B: Shoulder axle
B_y: Force on B in the Y axis
B_x: Force on B in the X axis

Equations of Equalibrium:

$$\sum M_S = 0 = T_M - L_1(F_{Ge}) - (L_1 + L_2)(F_P + F_{Gg})$$
$$T_M = 2.14Nm$$



Figures 14 and 15: Force Analysis of Collector Arm

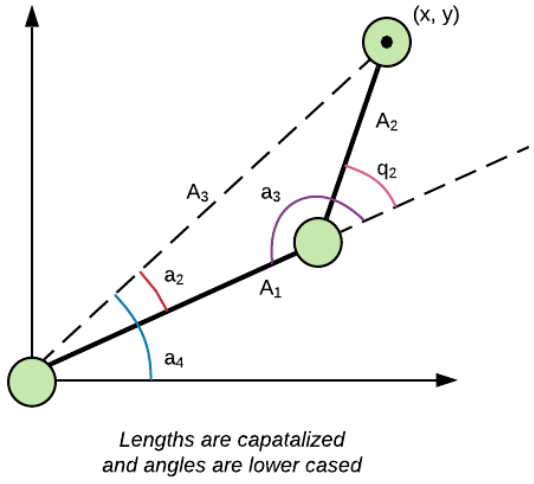
The forward kinematics of the arm were calculated by recording the Denavit–Hartenberg (DH) parameters of the arm system. There are four parameters recorded for each joint: d, θ, r, α. The symbol d is the offset along the z-axis to the common normal. Θ is the angle about the z-axis from the old x-axis location to the new x-axis position. The symbol r is the length of the common

normal. Finally, α is the angle about the common normal from the old z-axis position to the new z-axis location. The DH parameter table is shown below. $\theta_1$is the angle of the shoulder and $\theta_2$is the angle of the elbow.

Table 1: DH Parameters

d (meters)	θ (degrees)	r (meters)	α (degrees)
0.0	θ_1	0.130	0.0
0.0	θ_2	0.180	0.0

The inverse kinematics were then calculated using the geometric approach. With the inverse kinematics, it was possible to set the end effector to any point in the task space and calculate the required angles to get there. The equations to get the joint angles $q_1$and q_2 from a given (x, y) coordinate from the end effector are shown below.



$$A_1 = 0.13 \qquad A_2 = 0.18 \qquad A_3 = 0.31$$

$$(A_3)^2 = (A_1)^2 + (A_2)^2 - 2A_1A_2cos(a_3)$$

$$a_3 = cos^{-1}\left(\frac{(A_1)^2+(A_2)^2-(A_3)^2}{2A_1A_2}\right)$$

$$q_2 = 180 - a_3 \qquad a_4 = tan^{-1}\left(\frac{y}{x}\right)$$

$$a_2 = cos^{-1}\left(\frac{(A_1)^2+(A_3)^2-(A_2)^2}{2A_1A_3}\right)$$

$$q_1 = a_4 - a_2$$

$$a_4 = a_2 + q_1$$

Figure 16: Inverse Kinematics

3.1.4 Trash Storage Design

To store trash on the robot, a bucket was designed and fitted to the back of the robot. The design for this bucket is shown below in Figure 17.

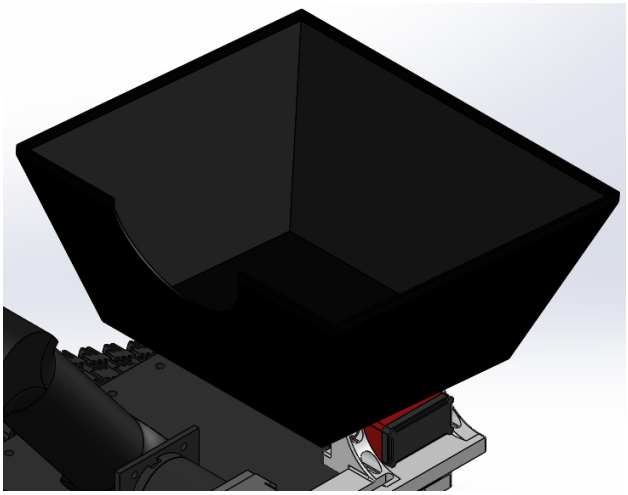


Figure 17: Smallbot Bucket Design

For this robot to dump the contents of the bucket, a servo is attached underneath that allows for easy manipulation and dumping. An arch is cut from the front bucket wall to allow the arm easy access to place objects in the bucket. The bucket has a volume of 0.0014 cubic meters. This is approximately enough to comfortably hold 3-4 soda cans. Provided below in Figure 18 is an image of the bucket holding five cans of soda.

Figure 18: Smallbot Holding Cans

The bucket servo has a torque requirement of 1.1 Nm, when the bucket contains four full soda cans which is the max capacity of the bucket. Figure 19 demonstrates how the servo empties a bucket. There is an arm attached to the servo that controls the bucket's movement.

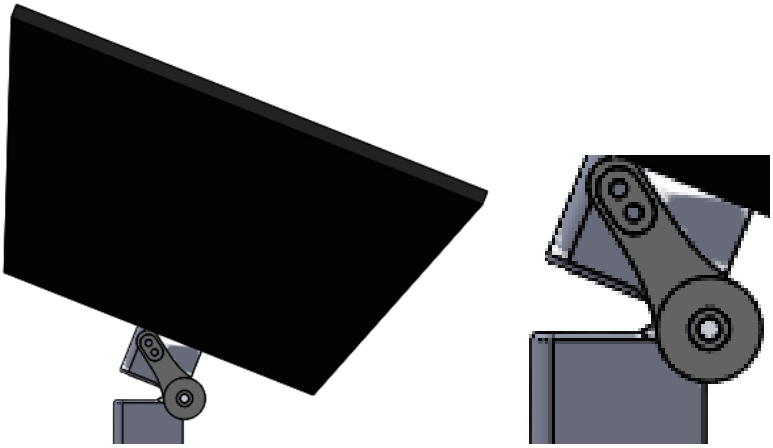
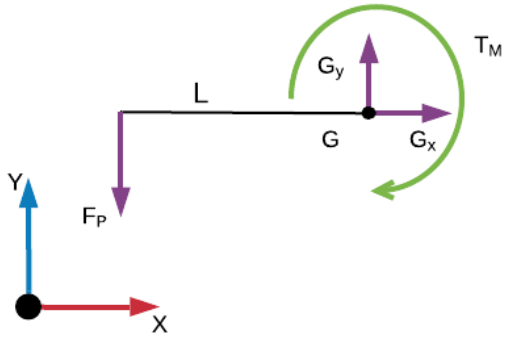


Figure 19: Servo Bucket Mechanism

The calculation is made below in Figure 20. The 2 kg value in the F_P variable is the weight of four soda cans (0.4kg) rounded to the nearest kilogram. The force analysis shown in Figure 20 is determined using the assumptions listed in Appendix E.

Given:
Max Servo Torque = 1.96 N-m
L = 0.056 m
$F_P = 2kg * 9.8m/s^2 = 19.6$ N

Legend:
L: Link 1 Length
F_P: Force due to the weight of the bucket
T_M: Torque Requirement for the motor
G : Servo Axle
G_y: Force on G in the Y axis
G_x: Force on G in the X axis



Equations of Equalibrium:

$$\sum M_G = 0 = F_P(L) - T_M$$

$$T_M = 1.1Nm$$

Figure 20: Force Analysis of Bucket

3.1.5 Camera Mount

When placing the camera on the robot, it was necessary that the camera tilts to allow for camera calibration and manipulation. A 3D printed part was designed to hold a servo that would tilt the 4-ounce camera. With the entire design assembled, the servo can hold and move the camera to any required position within its range. The design for this is shown in Figure 21 below.

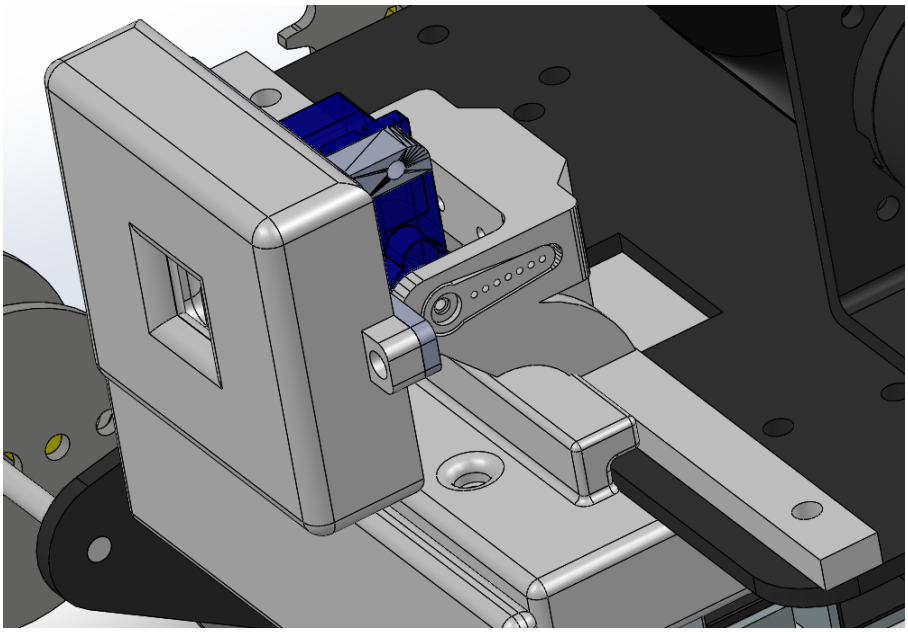


Figure 21: Camera Mount

3.2 Electrical System

3.2.1 Microprocessor Selection

Raspberry Pis were initially intended for all robots, as they would have enough power to handle OpenCV, ROS, and most future project expansions.

A ZED stereo camera was required for the Basebot, which would handle the mapping of the beach. The ZED camera is optimized for a Jetson board, so it was decided to replace the Raspberry Pi on the Basebot. The Jetson is much more capable of handling the stereo camera and mapping while also handling the management system.

3.2.2 Electrical System Design

The first step for the electrical design was determining the layout and components needed for the system. Figure 22 depicts a simplified block diagram of the components and their

connections in the system. As the figure shows, the small sensors are powered by the Raspberry Pi's 5V and 3.3V sources. The UBEC (universal battery eliminator circuit) is a device that converts high-voltage (such as 25V from a 6S LiPo) to lower voltages and provides a stable step down from the 11.1V source to the 5V 0.7A requirement to power the Raspberry Pi. The drive motors of the robot are powered and controlled by a dual h-bridge circuit taking power from a small voltage regulator.

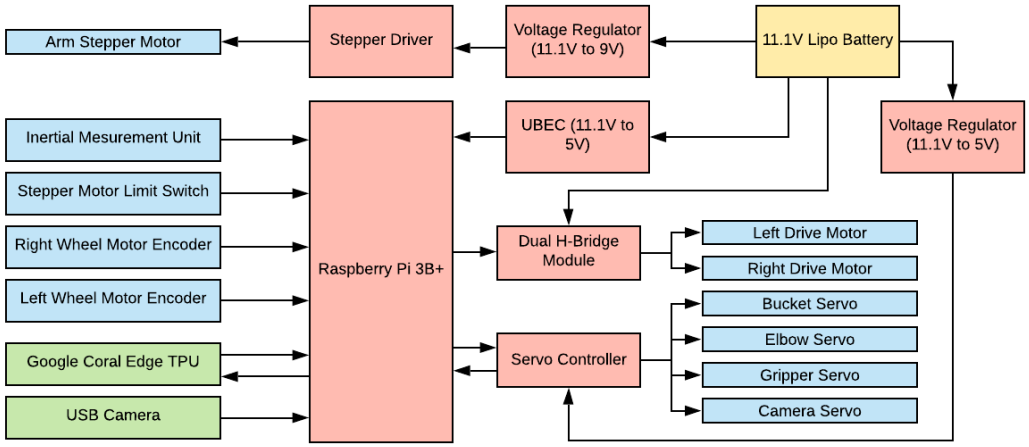


Figure 22: Smallbot Electrical Diagram

In the table below, all electrical elements in the Smallbot are documented with their power requirements. The max and idle draw columns were gathered from specification sheets and through testing. The power columns were derived using $P = I * V$.

Table 2: Power Requirements

Component	Amp Draw (A)	Idle Amp Draw (A)	Max Power (W)	Idle Power (W)
Raspberry Pi 3B+	2.50	0.70	12.50	3.50
Camera	0.40	0.40	2.00	2.00
H-Bridge	0.04	0.01	0.18	0.07
Camera Servo	0.50	0.40	2.50	2.00
DC Motor	1.20	0.50	10.80	4.50
Bucket Servo	3.50	0.40	17.50	2.00
Stepper Motor	1.68	0.40	8.40	2.00
Elbow and Gripper	1.80	0.10	9.00	0.50

Servo				
Coral Edge TPU	0.90	0.50	4.50	2.50
Stepper Motor Driver	0.04	0.04	0.20	0.20
Servo Controller	0.03	0.03	0.15	0.15
IMU	0.10	0.00	0.35	0.00

Using this data and the battery data from the three cell, 8000 mAh LiPo battery that is powering this system, the battery life of the robot can be derived. The maximum estimated run time of the system is approximately four hours while running idle and one hour at full load. To increase this time, a recharging station could be made to lengthen this time. In addition, 3D printed cases were fabricated for all components including the Raspberry Pi and USB camera.

3.2.3 Sensors

There are a few sensors dedicated to perception and positional data between the Basebot and the Smallbot. They were picked based on the requirements of low price, availability, and functionality.

The StereoLabs ZED stereo camera was purchased for $350. It has the ability to perceive its surroundings in 3D up to 20 meters. The ZED camera also supports ROS and has premade software available. The Basebot will use the ZED camera to scan the beach and divide it into zones.

A standard RGB camera was also purchased for the Smallbot. The BSG13MP camera module is $100. It provides USB connection and 75 degrees of vision. It supports Linux and Raspberry Pis. The BSG13MP is used for the Smallbots computer vision to locate trash for pick up.

The GY-521 IMU was purchased for $1. It contains both an accelerometer and a gyroscope with three axes of rotation. The IMU sends data using I2C to the Raspberry Pi. The

IMU will be utilized by the Smallbot to get the current angle the robot is at along its Z-axis and turning the robot to a specified angle.

Hall effect encoder sensors are attached to the drive motors. By using the encoders in conjunction with the IMU data, the robot can estimate its position.

3.3 Computer System

3.3.1 Task Management

Another task was designing how the robots would communicate with one another. Using ROS over a wireless network was determined to be the optimal architecture that would allow for easier project expansion in the future. In addition, all members of the project were familiar and comfortable working with ROS. The Basebot manages the Smallbots by assigning them tasks to accomplish, update their maps, and give them their position. There are three different types of tasks:

- Clean Task: contains the zone to be cleaned.
- Dump Task: sent when the Smallbot is ready to dump its container of collected trash.
- Avoid Task: contains instructions to avoid hazards.

Avoid tasks have the highest priority, so when a new avoid task is received; current progress on the previous task will be paused. Avoid tasks will be sent in cases not already covered by A* with the occupancy grid. Clean tasks have the lowest priority, so all higher priority tasks must be completed before they can be worked on. This management system can be expanded to manage multiple Smallbots.

3.3.2 ZED Mapping and Zones

In addition to managing the Smallbots, the Basebot will also map the beach with a ZED stereo camera. The Basebot is equipped with NVidia's Jetson Nano which has more power than a standard Raspberry Pi 3B+. This extra processing power will allow us to run SLAM (Simultaneous Location and Mapping). The map generated will be divided into sections that will act as the current operating field. The size of these sections will be about 10 square feet. These sections will be further divided into what will be called zones. The zones will be a strip in a section where a Smallbot will search for trash. These zones will be sent in cleaning tasks to the Smallbot. The ZED camera outputs a 3D map of what it can see up to 50 ft away. Due to a decrease in accuracy further from the ZED camera, the full operating distance of 20 meters was not used. This map will be updated in the management system so all robots will know where they are.

3.3.3 ZED Camera Testing

The ZED camera comes with an API that can create a 3D mesh or Fused Point Cloud from images it captures. Creating these can be a long process which would hold up code, however the API also comes with an asynchronous method of generating the mesh or point cloud in a separate thread. A mesh will be used as it is easier to generate and contains more geometrical information, which will be easier to divide into traversable sections. The mesh generated is of all images captured so far, not just the current image. The Mesh can easily be filtered to only contain traversable triangles, or triangles in a certain area, which will allow for zone generation of the beach immediately in front of it.

To test this, tutorials were followed and then scripts were created following a similar procedure as the tutorials. A class was written that filters and manages the original mesh to obtain the traversable area, which is used to create work zones. ROS was used to test both the custom scripts and filters.

3.3.4 Cleaning Process

Once the robots are initialized on the beach, the Basebot will begin mapping with the ZED camera. The section it is currently occupying will be cleaned first. It will divide the current section into workable zones and then assign these zones as tasks one at a time to the robots that are working for it. Once all the zones are cleared in the section of the beach, the system will move to the next unexplored section and repeat the process. During this time, the Basebot is continually mapping and monitoring the locations of the robots working on it. Once the entire beach is mapped and cleaned, all robots will shut down. This flow of events is shown visually in Figure 23.

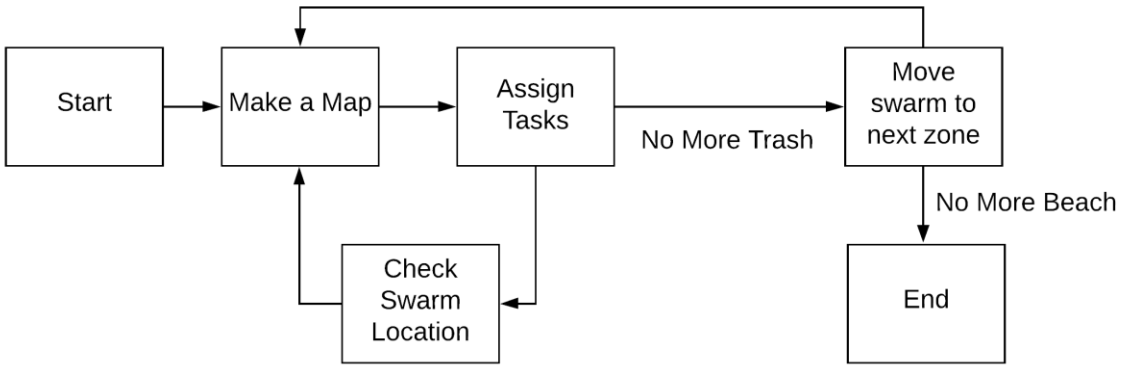


Figure 23: Cleaning Process Flow Chart

3.3.5 Smallbot Main Process

The Smallbot will request information from the Basebot, such as what its next task is, its position, and if it needs to avoid something. These tasks are then sent to a task identifier class to be handled by the proper task managers. During a clean task, the main task the Smallbot will be executing, the Smallbot will drive to the first coordinate in the zone. This information is available in the task object as well as its type and priority. Once in the zone, the Smallbot will generate coordinates in a serpentine path to thoroughly cover the area. If any trash is detected by the computer vision while navigating to a coordinate, the Smallbot will attempt to collect it. If the Smallbot's trash bin becomes full, it will save the execution of the clean task and request a dump task. Otherwise, the Smallbot will continue along the path until completed, request a new clean task from the Basebot, and repeat the process.

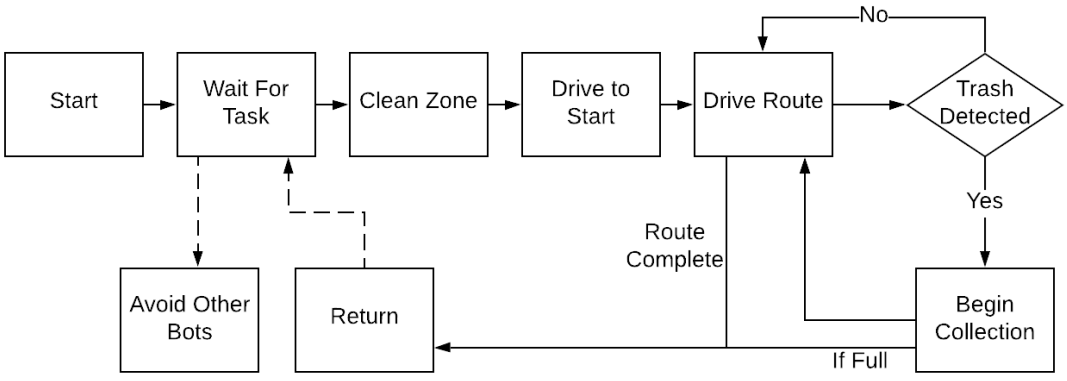


Figure 24: Smallbot Main Process

3.3.6 Smallbot Machine Learning and Computer Vision

For the Smallbot to be able to tell what objects/trash needs to be picked up, it used machine learning. Specifically, the Smallbot collected frames from the camera when instructed

and used a Tensorflow object detection model to identify trash. To assist with the computational power needed to run one of these models, a Google Coral Edge TPU was plugged in via USB to the Raspberry Pi. Adding this component increased the frame rate that the camera can use to identify objects on the beach and decreased the amount of CPU usage on the Raspberry Pi. This allowed the Raspberry Pi to focus on other computationally heavy processes, including but not limited to ROS and localization.

To train the model Google Cloud Platform's AutoML training platform was used. The pipeline for this training is provided below in Figure 25.

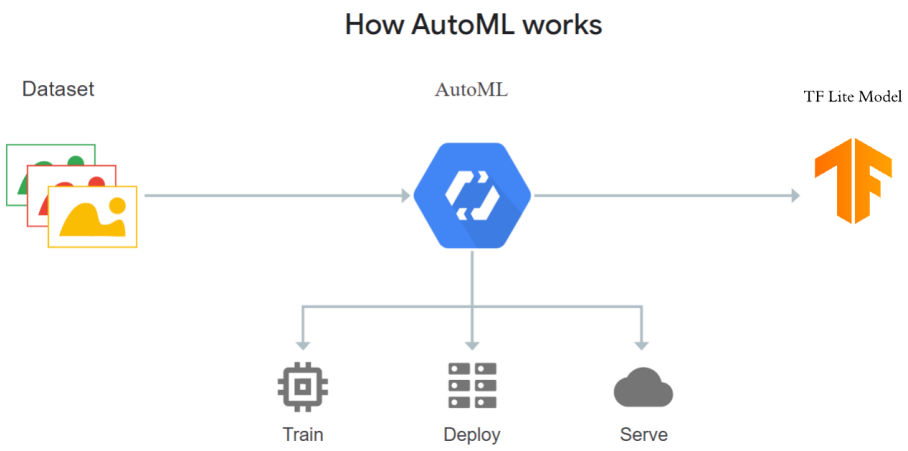


Figure 25: AutoML Pipeline

The first step to train the model was labeling the training data. Using the built-in tools provided in AutoML, the large dataset of soda can images was labeled. Some of the labeled images are shown below in Figure 26. The model was trained on 428 images and tested on 49 images. These images contained various orientations, lighting conditions, and environments. There were also many kinds of graphical images on each can prevent the model from identifying only certain kinds of cans. Photos were also taken where the can was partially submerged in the sand and some where they were crushed.

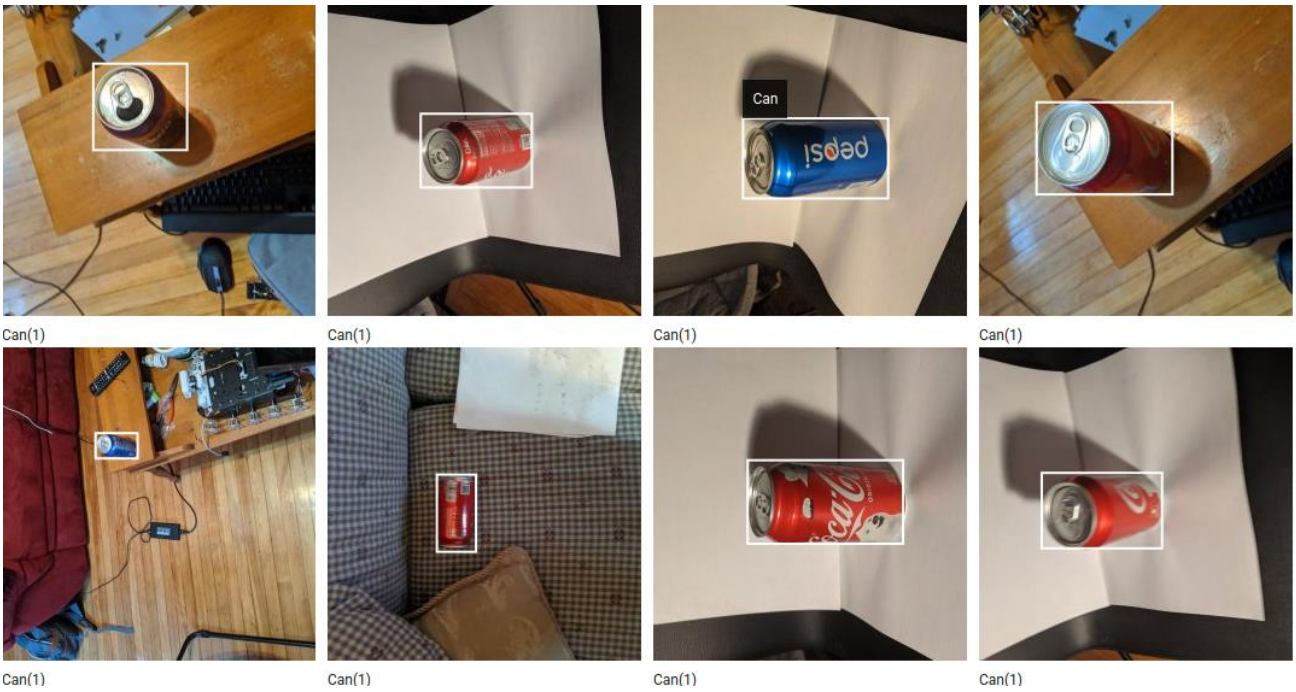


Figure 26: Example Labeled Cans

After the model had concluded training, the team was able to evaluate the model and produce Figure 27. This figure was generated with a confidence interval of 0.5 and an Intersection over Union threshold of 0.5. An IoU threshold is used to measure the overlap between the predicted bounding box for an object and the actual bounding box for it that was given during the labeling stage. The closer the predicted bounding box values are to the actual bounding box values, the greater the intersection and the greater the IoU value. In this figure, the team was also able to determine the recall and precision percentages of the model based on a given confidence interval. Due to the numerous definitions for these terms, the definitions in this context are given below:

Recall - A high recall model produces fewer false negatives.

Precision - A high precision model produces fewer false positives.

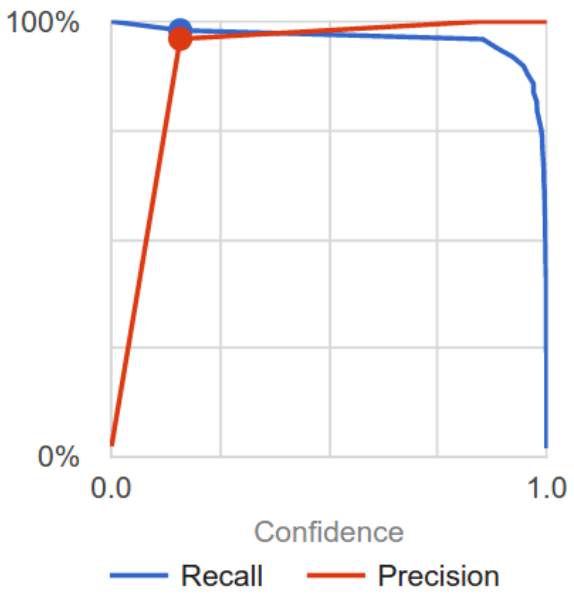


Figure 27: Recall and Precision Percentage

Figure 27 shows a visualization of the recall and precision percentages for the differing confidence levels on a sample of 428 images. The X-axis shows the confidence of the model. Confidence was lower on the left and higher on the right. The Y-axis is the percentage of each of the variables.

As can be seen from the graph over time there is a strong drop off in the recall percentage as the confidence increases to 1.0. This is because the more confidence there is the less false negatives that result from the model. There is also a strong increase in precision between 0.0 and 0.25. This is because when there is less confidence about the model there is less precision in the results.

Since the model was able to identify objects effectively, the model was then exported to the Raspberry Pi to be used with the Google Coral USB Accelerator pictured below in Figure 28.

Figure 28: USB Accelerator

Testing for this product with the model is shown below in section 4.1.10. After testing, it was determined that the coral does in fact improve the framerate that was received. With the Coral attached, the frame rate was approximately two frames per second and was .5 frames per second without it. One issue that the group ran into was the inability for the edgetpu library to work in ROS. To fix this issue, the team was able to establish a socket connection between a Python 3 script running the video code and a ROS Node. This socket connection allows the transfer of a video information between the script and the node.

After it was confirmed that the Machine Learning model could work on the smallbot, being able to use this information to track and pick up a can is vital to the success of this project. To complete this task, the team focused on tracking in one axis at a time. The first axis that was worked on was the pitch of the camera. This was implemented using the servo attached to the camera mount. Once this axis was concluded to work effectively, the next axis was then implemented on top of the previous. The next axis was the roll of the robot. This was implemented by having the robot rotate left and right to align with the closest can. This was probably the most challenging axis since the drive system has varying degrees of power based on which way the robot was turning. Once the robot was able to consistently track in these two axes, getting the robot to move towards the cans was the next challenge. Due to the lack of an

extremely high frame rate it was much easier to only move in two axes at one time instead of all three. This meant that anytime the robot centroid strayed too far from the can threshold it would stop moving forward and rotate to fix its position inside of the threshold. It was determined that this was the optimal and fastest way to navigate towards cans. To determine how close a can was the area of the bounding box surrounding the can was used. If that area was within a certain threshold it was then determined that a can was in the optimal location to be picked up. The robot then would drive forward a set distance to place the can in range of the arms reach. A common problem that was faced by this movement was that if the robot moved too much it would run over the can. This was solved by adding a front plate to the robot. When the robot determined that it was able to pick up a can it would also trigger the robot to stop what it was doing until it had collected the can. Once this process was completed, it would go back to finding more cans. If there were no cans present, then the robot would stop moving and in later implementations be able to go back on its path.

Chapter 4: Experimental Evaluation

4.1.0: Unit Tests

The following tests show the fundamental tests that were performed to validate the systems design. To test the Smallbot's movement, the robot's driving speed, turning accuracy, and sand locomotion ability were tested. To test the Smallbot's arm ability the force of the gripper, arm accuracy, and arm lifting power were tested. In addition, the bucket functionality, the machine learning model with and without the Edge TPU accelerator, ROS communication, and ZED functionality were tested.

4.1.1: Driving Speed of Smallbot

From the calculations in the previous chapter, this robot is estimated to drive at approximately 0.105 m/s at the desired motor speed of 75 rpm. This speed will be influenced by the weight and friction the treads apply to the drive sprocket. To test the speed of the Smallbot, it was driven 300 millimeters and timed using a stopwatch. Ten trials were used to determine the actual drive speed of the robot. These trials are shown below in Figure 29.

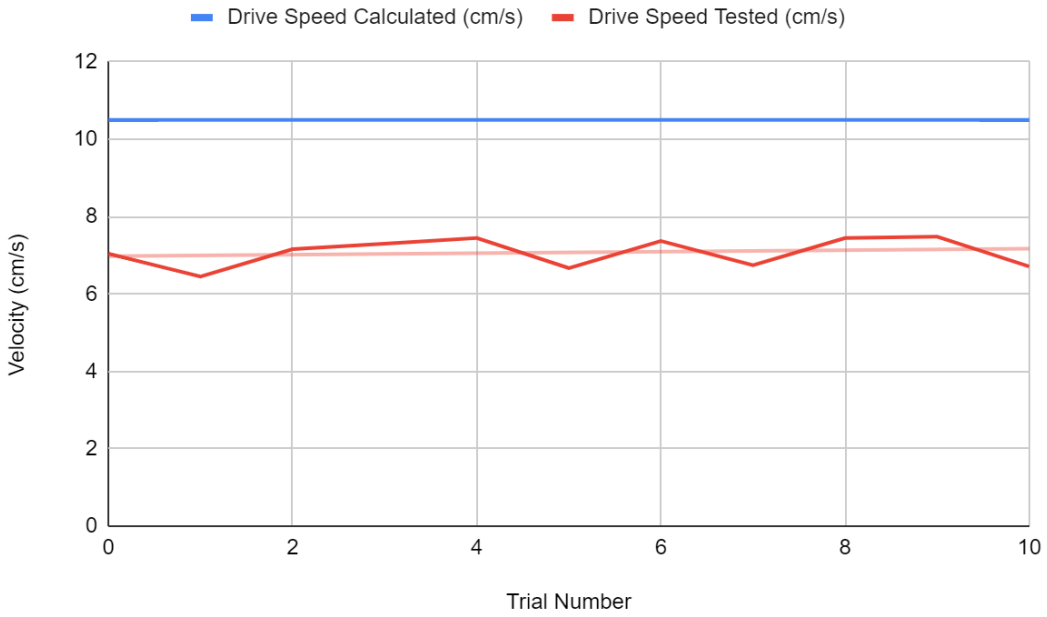


Figure 29: Calculated Speed Vs. Tested Speed

Figure 29 shows a comparison between the calculated speed of the robot and the tested speed of the Smallbot. The axes show velocity of the robot in meters per second and the trial number. Two lines are displayed on this chart. In blue is the calculated drive speed of 0.105 m/s. The red line varies as it goes between the trials, but the trend line averages approximately 0.070 m/s.

4.1.3: Turning Accuracy of Smallbot

Due to the nature of driving in sand, accurate turning is critically important to navigating the beach. The turning accuracy was tested by attaching a marker to the robot and measuring the arc created while turning to varying angles. Twenty trials were performed where the robot turned to both 45 and 90 degrees. On average, the turning accuracy was off the target angle by less than 7 degrees. The data for this test is shown in Appendix C Table 5.

4.1.4: Sand Locomotion of Smallbot

To test the Smallbot's ability to drive in sand, a small sand pit was constructed. This sandpit was large enough for the robot to be able to rotate and drive short distances. During the development of the drivetrain, early designs became clogged with sand and could not move. The final design of the tread proved to be able to rotate continuously and drive linearly without clogging.

4.1.5: Gripping Force of Smallbot Gripper

The gripper was estimated to output 18 N of force at the tip of the fingers. This force was tested by attaching a spring mass scale to the end of the plastic extender, and as the gripper is closed, the peak force was recorded. This peak force was halved as the force was only recorded from one side of gripper assembly. The tested force falls short of the expected force due to high friction in the gears of the gripper.

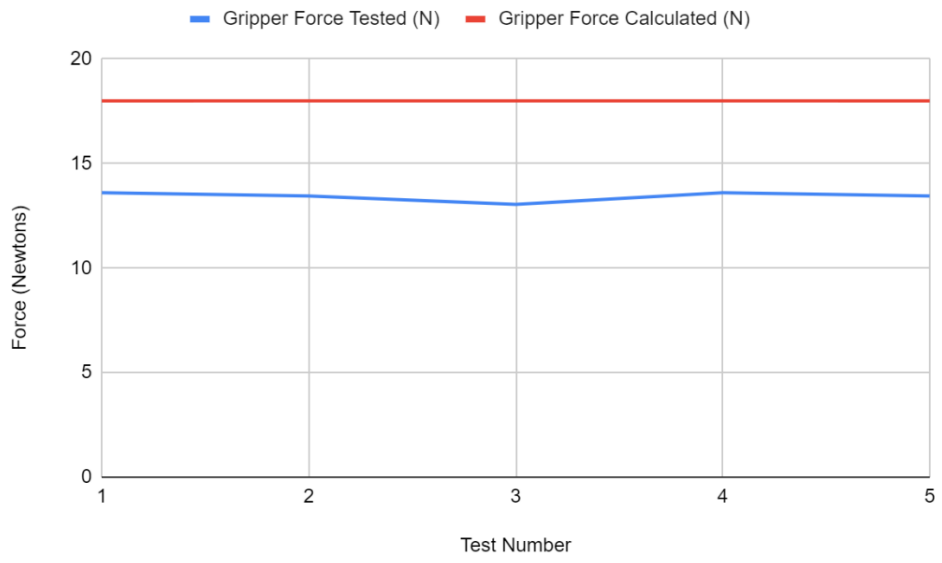


Figure 30: Calculated Gripper Force Vs. Tested Gripper Force

In Figure 30, the calculated gripper force (in blue) is compared with the tested gripper force (in red). The X-axis shows the test number. The Y-axis shows the force in Newtons that the Smallbot's gripper applied to the force sensor. The pale red line represents the trend line,

4.1.6: Smallbot Arm Accuracy

The arm's position accuracy is critically important to being able to accurately pick up the trash. Thus, the kinematics of the arm were calculated, and the high resolution of the stepper motor allows for precise control of the shoulder joint. To test how accurate the arm is when given an end effector position to go to, the end effector position was measured with a set of calipers and compared with the desired position. This testing allowed for the error in the arm to be tested.

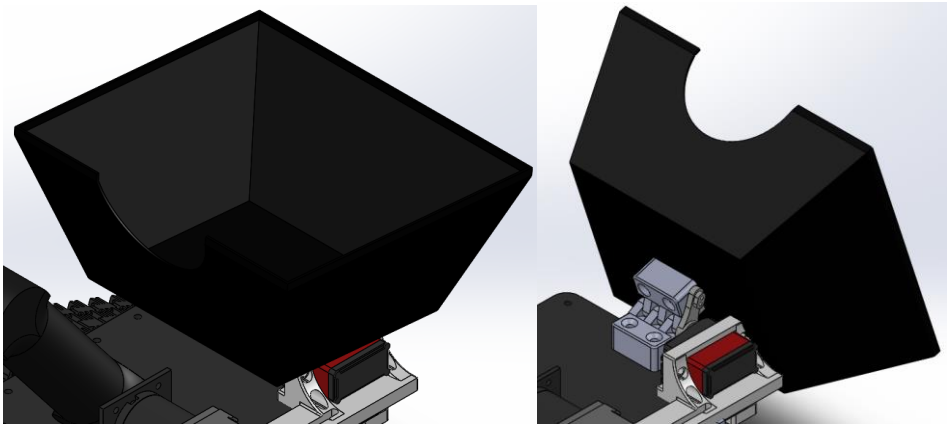
4.1.7: Arm Max Lift Weight

The maximum lift weight of the arm was determined through torque calculations with the arm motors and link lengths resulting in a lifting capacity of 1.1kgs. This is well above the weight of a full soda can, averaging a weight of 0.4kg.

In the testing, four different cans with various weights were used. Can 1 weighed 100g, can 2 weighed 200g, can 3 weighed 450g, and can 4 weighed 650g. The elbow servo was able to lift all but the last can; however, the servo was still able to hold its position. This last can was filled with both sand and water, and weighed more than the average full can, so this is an extreme scenario that the robot would likely not have to lift.

4.1.8: Smallbot Bucket Testing

In order to test the accuracy of the buckets position the team isolated the buckets movement without weight. The requirements for the bucket movement are purely if the Bucket can go between two states. The first is the standard position where the bucket is level on top of the robot. The second is the bucket tilted all the way backwards at a 90-degree angle. A visualization of these two states are shown below in Figures 31 and 32.



Figures 31 and 32: Bucket position 1 (left) and Bucket position 2 (right)

Upon testing, the bucket was able to move to these two positions with reasonable accuracy (+/- 2 degrees) as well as empty all cans from the bucket. To test the hold weight of the bucket, numerous filled soda cans were placed inside of the bucket. The bucket would then need to move into the state described by Figure 32 shown above. If the bucket were able to tilt all the way without stall, then the bucket's functionality would be successfully tested. The bucket was able to complete this test by lifting the weight that was provided to it to the desired location.

4.1.9: Smallbot Machine Learning Recall and Precision

To test the machine learning model, the model was run on a variety of test images. Two of these test images are provided below in Figures 33 and 34. To clarify, the images pictured in

Figures 33 and 34 are completely new images the model has never seen before. For this model to work, it needs to be able to correctly identify soda cans. The team predicts that the model will identify cans often. Figures 33 and 34 were some of many images that were tested. All of which returned the correct outcomes.

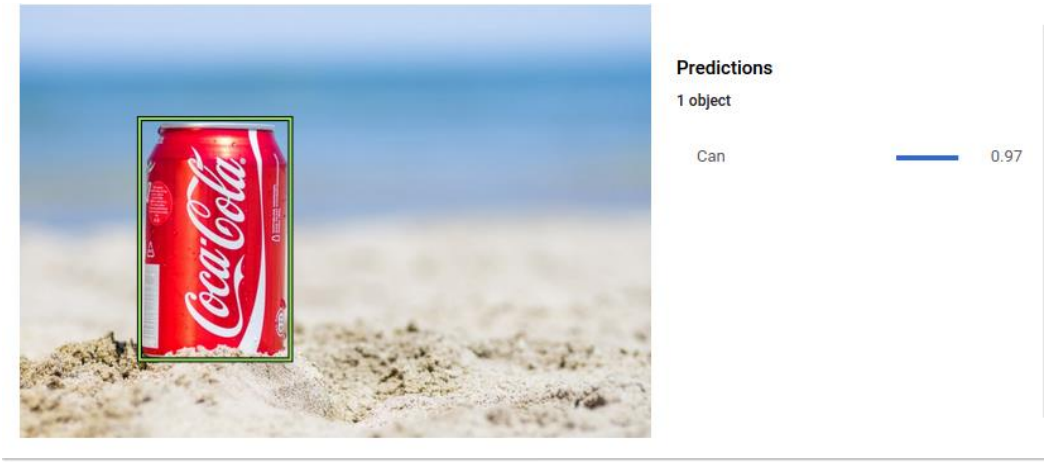


Figure 33: Machine Learning Test Image 1

Figure 33 shows the output from the model after being run on the image to the left in the figure. On the right of the figure is the predictions that were made about the image, including the number of objects and what each object is, along with how confident the model is that the object is what it identified it as. In this case, the model identified that there was one object, which was a can, and it was 97% confident that this object is a can. Another feature that this figure provides is a bounding box on which object it deemed to be a can. In this case, it properly identified the soda can pictured.

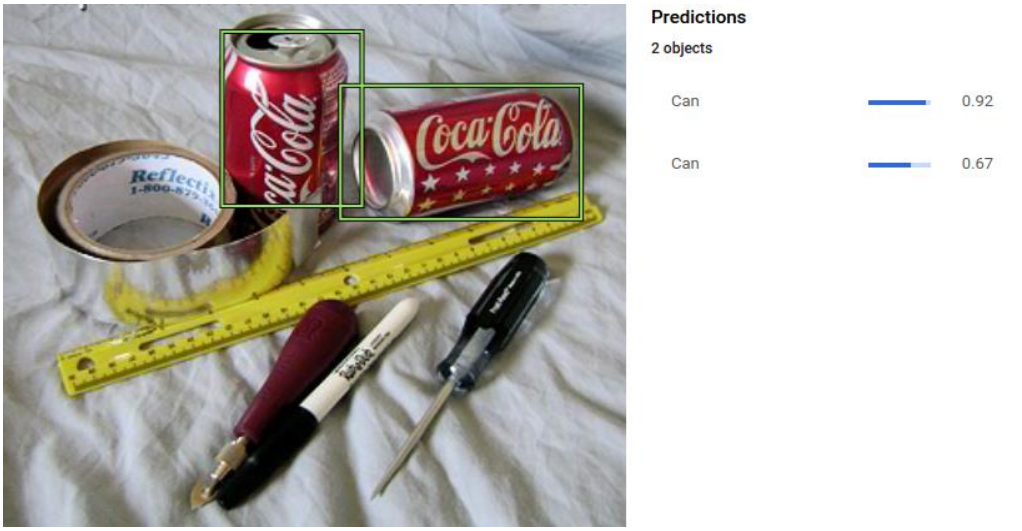


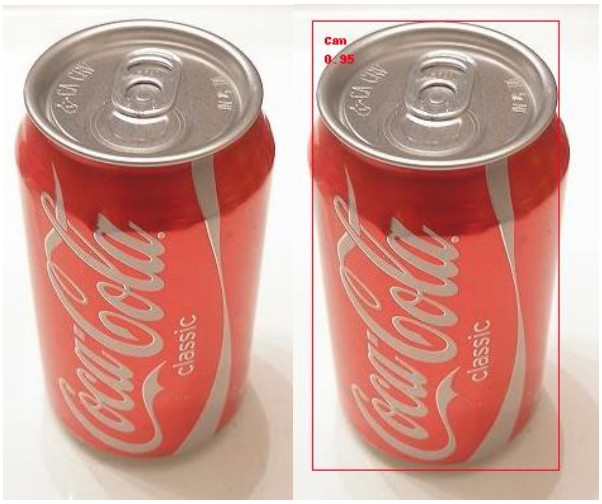
Figure 34: Machine Learning Test Image 2

Figure 34 also depicts an output from the model. This time the model identified that there were two objects in the figure. Both objects were cans. It identified one of these cans at 97% and the other at 67%. Even though the model was less confident, it still identified the proper items.

4.1.10 Smallbot Edge TPU Accelerator Testing

After the completed testing of the model off the Raspberry Pi. The team then decided that it was time to test on the Raspberry Pi with the Google Coral Edge TPU Accelerator. The team first tested the ability to run using the accelerator. This was in fact true, as it was able to receive the correct outcome from the terminal execution of the example code from the coral website (Google-Coral, 2020).

The second test that was completed was to make sure that it was working on a known image of the can. To test that the model ran on the Raspberry Pi using the detect_image.py file from the tutorial as shown previously. The results from this are pictured below in Figures 35 and 36.



Figures 35 and 36: Unprocessed and Processed Soda Can Image

Figure 35 shows the unprocessed image. This is the input image to the model. Figure 36 shows the processed image output from the model. As shown in Figure 36, there is a bounding box around the can, and a label telling us it is a can, along with the probability that it is a can, which in this case is 95%.

4.1.11 ROS Testing

To test ROS and the custom communication system, task objects were created that would be parsed and sent as service messages through ROS from the Basebot to the Smallbot. Each part of the task was sent in a ROS message. When the Smallbot receives this message, it is parsed back into a task object. This testing performed as expected.

4.1.12 ZED Testing

To test the ZED camera's functionality, data was taken from local areas around WPI and tested the mesh outputs and occupancy grid generation. Overall, this works as intended. The ZED was able to move to any new position and orientation and it will be able to identify new

zones and create cleaning tasks for those zones. This test ensures that the entire system could clean an entire beach when it is fully implemented.

4.2.0: Integration Tests

The following tests integrated more than one part of the system to test integral functionality of the system. Basic ROS communication with the custom messages was tested with and without the ZED camera mapping system. In addition, the Smallbot's ability to correctly identify and collect a can.

4.2.1: Between Smallbot and Basebot

To test the basic communication between the Smallbot and the Basebot, different tasks were simulated to send to the Smallbot from the Basebot. In the first test, cleaning tasks were created and the Smallbot kept requesting them indefinitely. The Smallbot would increment a counter in the cleaning manager to simulate it cleaning the zone. Once the counter reached a threshold, it marked the task as complete and requested a new cleaning task from the Basebot. This test functioned just as expected.

In the next test, the same system was kept with the cleaning tasks, but also periodically added avoid tasks. These functioned the same way as cleaning tasks with a counter. However, when an avoid task is added to the Smallbot, it takes priority and interrupts the current cleaning task. This will ensure that immediate instructions can be sent to the robot that will interrupt the current cleaning task. Once an avoid task is complete, the Smallbot resumes the past cleaning task. Periodically, a new avoid task is sent to the Smallbot. In addition, when working on a task,

different velocities were sent to ensure that the robot could follow commands while working on a task. This test functioned as expected.

4.2.2: Incorporating the ZED camera

The next test of robot communication was to incorporate the ZED camera. The ZED, as part of the Basebot node, continuously mapped the world in front of it and generated an occupancy grid of this area. From this visible area, it determined what zones are in its field of view and created a cleaning task for each one. This was then set as the Basebot's list of cleaning tasks. Once the Smallbot connected, it began cleaning the tasks sequentially. Path traversal was also tested, where waypoints were added in the zone. This path looked like a zigzag through the zone starting at the bottom and ending at the top. Once the end was reached, it marked the task as complete. To simulate moving, a simulated position was tracked that incremented towards its goal every step of the Smallbot. Once the simulated position is within a threshold of the goal position, then it marks that position as reached and tries to reach the next goal position. Follow the link in Appendix A to view this simulation in its entirety. The green rectangle is a zone and the purple sphere is the simulated position. This process will repeat until all zones are completed in front of the ZED.

This test was repeated while moving the ZED from its initial position, changing the visible zones. The Basebot was able to update its visible zones and create new cleaning tasks to send to the Smallbot. In addition, this test had several Smallbots simulated as part of the system working on cleaning tasks. This test worked as expected, proving that the system would be sufficient to clean an entire beach.

4.2.3: Computer Vision Tracking

In order to test the computer vision tracking, cans were put in front of the robot's path in various different locations in relation to the smallbot, and counted the number of times the robot was able to successfully pick up the can and the number of times it was not. In order to make sure this procedure worked over 30 different can placements were tested within the Robots path. The robot was able to pick up 28 of these cans. Two of the cans were not detected initially, but with a slight nudge, the robot was able to detect the cans just fine. The suspected reason for this failure is the machine learning model not being trained on enough images of the bottom of the can.

Chapter 5: Discussion and Results

The testing proved that this robotic system is feasible for full scale testing. Each component of the system had varying successes and failures; thus, they have been broken out into sections. The original objectives of the project are shown in Appendix D.

5.1 Mechanical Results

Most of the mechanical goals were reached. The Smallbot could drive reliably, as shown in the driving speed unit test. The actual driving speed proved to be slower than the calculated speed, which could be due to the friction in the drive train. This friction could be reduced by using machined aluminum parts instead of 3D printed parts and lubrication could be added to the gears and treads. Sand locomotion could not be fully tested due to the COVID-19 pandemic; however, a small test in a sand pit showed that the tread design has the potential to work. The Smallbot could turn in place but this does not confirm full driving capability.

The collection mechanism was fully implemented and shown to work in integration testing. The gripper's finger design allowed it to move through sand without resistance in the testing sand pit. On an actual beach, sand density is highly variable, and more testing needs to be done before this design can be proven to work on any beach. The arm design also performed well in the unit testing. The mechanical limit of this system was its inability to lift a can that was completely full of sand and water weighing 650 grams. While this is an extreme case compared to a full can of soda weighing 400 grams, this proves that a stronger elbow joint servo should be used for any edge cases.

Storage and dumping of the trash were the easiest of the mechanical systems and was successful. The system was able to store multiple cans and dump them even when the combined

weights of the cans were 1.5 kg. This shows that the system can reliably store and dump three full cans of soda.

Aside from the Smallbot, the Basebot was not fabricated in this iteration of the project. The computer science requirements were fully tested; however, the electro-mechanical components were not. Future

5.2 Electrical Results

Our original electrical objectives were achieved. The electrical system was implemented on a perforated board allowing for a clean design and less chance of wiring error. Ideally, this system should have a PCB and be tested in simulation to confirm that all parts are fully powered. Each electrical component was tested to ensure that the subsystem it controlled operated correctly. The Basebot electrical system was not implemented into the system as it was easier to test the ZED camera without it.

5.3 Computer Science Results

The computer science aspect of the project had successful and incomplete objectives. To start, the ROS communication was thoroughly tested and determined to be a success. Zones were communicated as tasks with multiple Smallbots connected to the Basebot without any system errors. The management system is the backbone of the project and as such was one of the first parts completed.

The ZED camera was the next successful portion of the computer science aspect of this project. The system was successfully able to have the ZED camera create a 3D mesh of the beach and then create an occupancy grid of the visible area and identify work zones for the Smallbots to clean. However, this was not tested on a real beach. It is unknown if the system will act

differently in a real environment or how accurate simultaneous localization and mapping will be on a full beach. There are no obvious issues with the system being unable to do so, but until the Basebot is fabricated this will not be certain.

Due to the Basebot not being fabricated, the system was unable to move forward with the trash exchange between the Smallbot and Basebot. This objective is a goal of future teams as they develop the Basebot. This also affects the final objective of returning to the start position. The full mechanical system should be fabricated before this feature is implemented in the management system.

Originally the Smallbot was going to detect trash with OpenCV, but after minor testing early in the project, it was decided that cans were unable to be identified consistently with traditional computer vision methods. Instead, machine learning was implemented into the system. This included generating a Tensorflow Lite model as described in the methodology section. This method ultimately worked better than traditional computer vision methods and completed the goal.

Chapter 6: Conclusion

In the project's present state, the system can clean a beach of cans. The Basebot can maintain multiple Smallbots, mapping a beach, and dividing it into workable zones to be sent as tasks to each of its Smallbots. The Smallbot can communicate with the Basebot, and moving in its environment, correctly identifying a can, moving to the can's position, and collecting it. This project is a proof of concept of a multi robot system that can clean a beach of human debris. This project laid down the foundations for future teams to continue developing the system and integrate further functionality.

In the future implementations of this project, the system should look towards fabricating the Basebot, upgrading Smallbot positional sensors, collecting additional pieces of trash other than soda cans, developing a stand-alone method for communication, and improving the scalability of the current system. With these improvements, this system should be able to accomplish the mission that was set out to complete.

Bibliography

Baker, A. (2003, March 4). Desperately Seeking Survival. Retrieved October 3, 2019, from http://content.time.com/time/magazine/article/0,9171,393799,00.html.

Bergmann, M., Gutow, L., & Klages, M. (2015). *Marine anthropogenic litter*. doi: 10.1007/978-3-319-16510-3

COASTSWEEP. (n.d.). Retrieved from https://www.mass.gov/coastsweep.

Commonwealth of Massachusetts. (n.d.). FY 2019 Final Budget. Retrieved from https://malegislature.gov/Budget/FY2019/FinalBudget.

Deutsche Welle. (n.d.). Plastic pollution: Do beach cleanups really make a difference?: DW: 20.12.2018. Retrieved October 3, 2019, from https://www.dw.com/en/plastic-pollution-do-beach-cleanups-really-make-a-difference/a-46196975.

Google-Coral. (2020, January 27). google-coral/tflite. Retrieved May 8, 2020, from https://github.com/google-coral/tflite/tree/master/python/examples/detection

Grenfell, R.D., & Ross, K.N. (1992). How dangerous is that visit to the beach? A pilot study of beach injuries. *Australian family physician*, *21 8*, 1145-8 .

Hanna, M., & Petersen, D. (n.d.). Fuel efficiency factors for tractor selection. *Farm*

Energy. Iowa State University. https://store.extension.iastate.edu/Product/Fuel-Efficiency-Factors-for-Tractor-Selection-Farm-Energy-PDF

Hughes, J., Marisol, & James. (2013, November 11). Wheels vs Continuous Tracks: Advantages and Disadvantages. Retrieved January 16, 2020, from https://www.intorobotics.com/wheels-vs-continuous-tracks-advantages-disadvantages/

H.Dwighta, R., M.Fernandezb, L., B.Bakerc, D., & H.Olsone, B. (2005, April 22). Estimating the economic burden from illnesses associated with recreational coastal water pollution-a case study in Orange County, California. Retrieved October 3, 2019, from https://www.sciencedirect.com/science/article/pii/S0301479705000289.

Impacts of Mismanaged Trash. (2017, May 23). Retrieved from https://www.epa.gov/trash-free-waters/impacts-mismanaged-trash.

International Coastal Cleanup. (2019, September 21). Retrieved October 4, 2019, from https://oceanconservancy.org/trash-free-seas/international-coastal-cleanup/.

Join The Wave. (n.d.). Retrieved October 3, 2019, from https://oceanconservancy.org/trash-free-seas/international-coastal-cleanup/volunteer/.

Markhor: Robotic Mining Platform. (n.d.).

Menon, R. (2018, December 6). Plastic pollution threatens tourism dependent Bali. Retrieved October 3, 2019, from https://www.downtoearth.org.in/blog/environment/plastic-pollution-threatens-tourism-dependent-bali-62408.

National Economics Program. (n.d.). NOEP Coastal Economics. Retrieved October 3, 2019, from https://www.oceaneconomics.org/Market/coastal/coastalEcon.asp.

NOAA. (n.d.). Is Marine Debris Impacting Your Beach Day, and Your Wallet? Retrieved from https://response.restoration.noaa.gov/marine-debris-impacting-your-beach-day-and-your-wallet.

Parker, Laura. “Beach Clean-up Study Shows Global Scope of Plastic Pollution.” *National Geographic*, 10 Oct. 2018, www.nationalgeographic.com/environment/2018/10/greenpeace-beach-cleanup-report-highlights-ocean-plastic-problem/.

RC Speed Tank by Bryant87. (2017, January 11). Retrieved from https://www.thingiverse.com/thing:2024364

United States Environmental Protection Agency. (2019, June 6). LEARN: What Affects Beach Health. Retrieved October 3, 2019, from https://www.epa.gov/beaches/learn-what-affects-beach-health.

United States. Environmental Protection Agency. Office of Water. (1996). *Liquid Assets: A Summertime Perspective on the Importance of Clean Water to the Nation's Economy*. *Liquid*

Assets: A Summertime Perspective on the Importance of Clean Water to the Nation's Economy. United States Environmental Protection Agency, Office of Water.

USB Accelerator. (n.d.). Retrieved from https://coral.ai/products/accelerator/

Stickel, B. H., & Jahn, A. (2012). THE COST TO WEST COAST COMMUNITIES OF DEALING WITH TRASH, REDUCING MARINE DEBRIS. Retrieved from https://www.coastal.ca.gov/publiced/coordinators/WestCoastCommsCost-MngMarineDebris.pdf

Zielinski, S., Botero, C. M., & Yanes, A. (2019). To clean or not to clean? A critical review of beach cleaning methods and impacts. *Marine Pollution Bulletin*, *139*, 390–401. doi: 10.1016/j.marpolbul.2018.12.027

Appendix

Appendix A: Project Presentation Video

Link to video: https://www.youtube.com/watch?v=hBx1OiOC5TI

Appendix B: Beachbots MQP Wiki

Link to website: https://sites.google.com/view/beachbotsmqp/home

Appendix C: Unit Testing Data

Table 3: Gripper Test Data

Test Number	Gripper Force Tested (N)
1	13.6
2	13.45
3	13.05
4	13.6
5	13.45

Table 4: Drive Speed Data

Trial Number	Drive Speed Tested (cm/s)
1	70.4
2	64.5
3	71.5
4	72.9
5	74.4
6	66.6
7	73.7
8	67.4
9	74.4
10	74.8

Table 5: Smallbot Turning Test

Test Number	Attempted Angle (Degrees)	Actual Angle (Degrees)
1	90	83
2	90	99
3	90	90
4	90	100
5	90	91
6	90	80
7	90	86
8	90	90
9	90	97
10	90	102
1	45	38
2	45	51
3	45	58
4	45	53
5	45	55

Appendix D: Initial Objectives

Mechanical Objectives

- Collector to pick up trash. Trash will be of soda can size or smaller
- Collector will filter out sand from trash and debris
- Collected trash will be dumped into the Basebot
- Both Smallbot and Basebot will store trash
- Form of locomotion to move through sand while also not sinking

Electrical Objectives

- Smallbots will operate with Raspberry Pi 3 B+ with the default camera
- Basebot will operate with a Jetson Nano with a ZED Stereo Camera

Computer Science Objectives

- Use ROS to communicate information between Basebot and Smallbots
- Basebot will map field with stereo camera, then divide the field into zones
- Zones will be sent as Tasks to the Smallbot to clean
- Smallbots will use OpenCV to detect trash to collect
- Smallbots return to the Basebot when the trash container is full
- Once the beach is cleaned, robots will return to starting location for retrieval

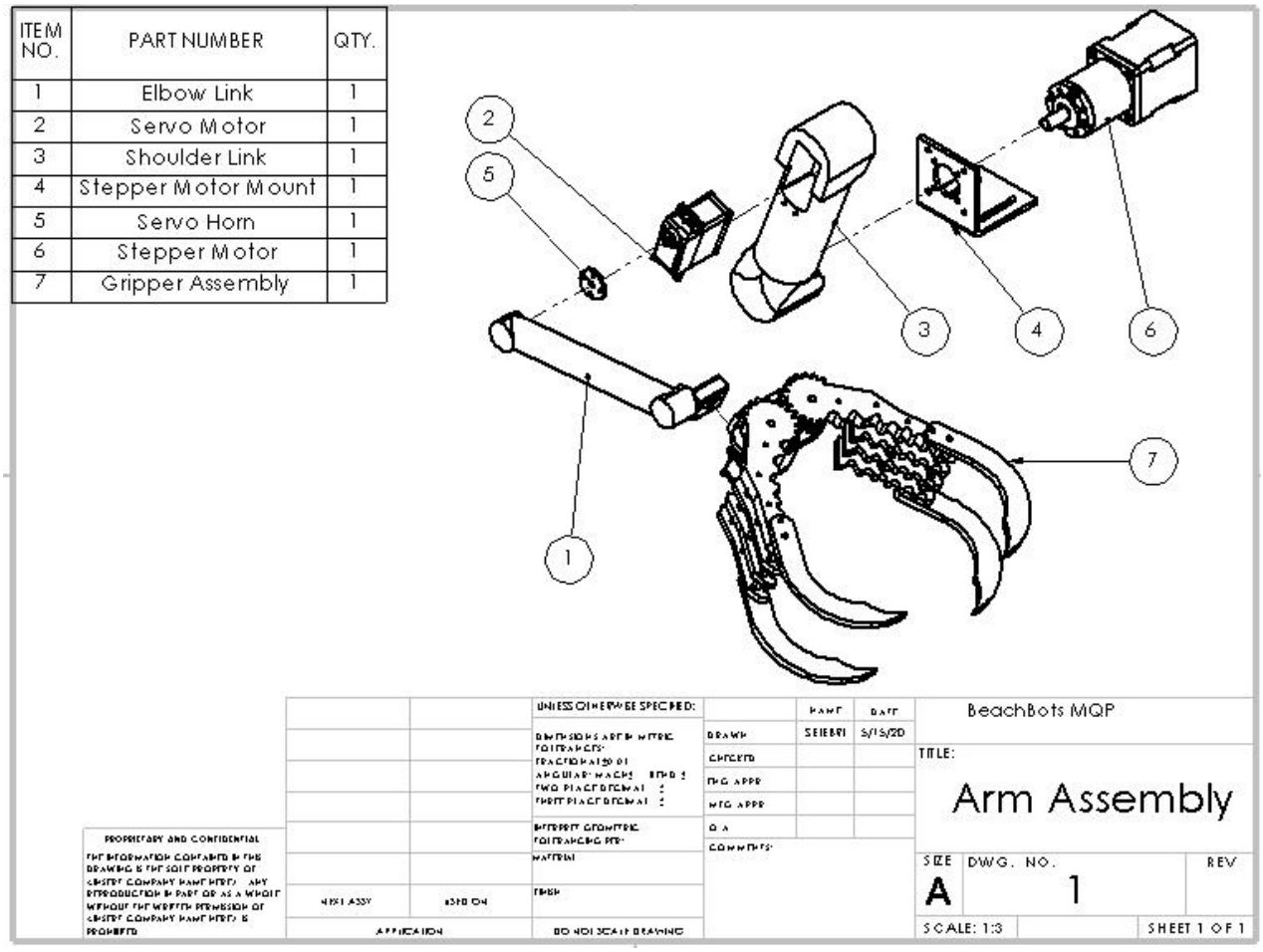
Appendix E: Force Analysis Assumptions

- All links are rigid bodies
- Frictionless joints and links
- Force of gravity on links is negligible

- Gears have perfect contact at a point

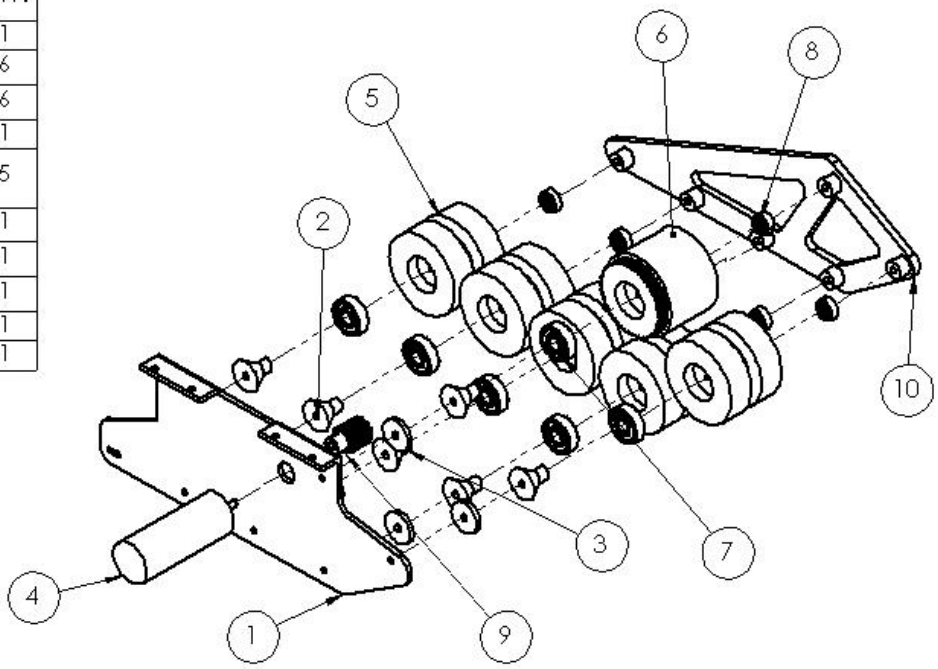
Appendix F: Exploded Assemblies Views

In addition to these drawings, the full Solidworks assembly of the robot can be found on the wiki in Appendix B.



Figures 37: Arm Subassembly

ITEM NO.	PART NAME	QTY.
1	Side Panel	1
2	Disc Spacer	6
3	Cone Spacer	6
4	Motor	1
5	Runner Wheel Sub Assembly	5
6	Geared Sprocket	1
7	608 Ball Bearing	1
8	624 Ball Bearing	1
9	Pinion	1
10	External Plate	1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN METRIC
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING

	NAME	DATE
DRAWN	SEIBERT	5/16/20
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		

COMMENTS:

NEXT ASSY | USED ON | APPLICATION

BeachBots MQP

TITLE:
Drive Train Sub Assembly

SIZE A | DWG. NO. 2 | REV

SCALE: 1:3 | SHEET 1 OF 1

Figures 38: Drive Train Subassembly