Final Design Review: WALL-E

MAE 322

Team FRYday

December 15, 2023

Team members: Ayanna SMITH, Calvin NGUYEN, Jasper WALDMAN

Jimmy TRAN, Kazuki TOJO, Luc HARBERS

Rihan SAJID, Sarah FRY, Shayan ALAVI

Instructors: Professor D. NOSENCHUCK, A. GAILLARD, G. NORTHEY

Role/Subteam	Member(s)
Project lead	Kazuki Tojo
Hardware lead	Jasper Waldman
Software lead	Jimmy Tran
Drivetrain	Jasper Waldman, Luc Harbers
MedKit deposition	Sarah Fry, Shayan Alavi
Electronics	Calvin Nguyen, Jimmy Tran, Rihan Sajid
Components	Ayanna Smith, Kazuki Tojo

Table 1: Roles and subteams, along with the responsible team members.

FDR Section	Member(s)
Executive Summary	Kazuki Tojo
Introduction	Kazuki Tojo, Jasper Waldman
SaRR Specifications	Sarah Fry, Jasper Waldman, Luc Harbers
SaRR Operational modes	Jimmy Tran
Project Management	Kazuki Tojo
Weight & Mass Analysis	Calvin Nguyen, Rihan Sajid
Static & Dynamic Analysis	Ayanna Smith, Shayan Alavi
Torque, Energy & Performance analysis	Ayanna Smith
Control System	Jimmy Tran, Calvin Nguyen
Engineering Drawings	Rihan Sajid, Sarah Fry
Test Results	Sarah Fry, Jimmy Tran
Conclusions and Further Work	Sarah Fry, Luc Harbers

Table 2: Team members' contributions to the FDR.

1 Executive Summary

The idea of a Search and Rescue Robot (SaRR) is inherently strongly tied to the idea of automation. The necessity to search and rescue a victim implies the hazardous nature of the environment and its conditions. Where possible, it is sensible to minimise the number of human rescuers put in similar danger, and the implementation of a SaRR achieves this goal.

A SaRR must be able to navigate through obstacles as it maneuvers towards the victim, and ultimately improve the odds of rescue. The latter can be achieved through different approaches, from providing a line of communication with the rescuers to ideally recovering the victim to a safe location. It should also be mentioned that autonomous operation is a significant element of a SaRR, as there may be areas without accessibility to a remote operator.

In this project, WALL-E—a functional prototype of a medical kit delivery SaRR—was developed using a tread design with a front lip. This approach has three key advantages. First, the front lip provides a strong grip on elevated surfaces, allowing WALL-E to reliably climb over obstacles. Second, the treads allow for a zero turning radius, excellent for precise maneuvers through irregular and unpredictable terrain. Finally, unlike a wheel design, treads allow WALL-E to maintain a low center of gravity, offering crucial stability. This ensures that WALL-E does not tip over and potentially drop the medical kit that must be delivered to the victim.

WALL-E was controlled manually for initial testing, and the program was refined over several weeks to achieve full autonomy, meeting all requirements of the MAE 322 Course. WALL-E has successfully been rigorously tested through multiple contiguous runs of the full obstacle course, achieving this in one continuous autonomous mode. The obstacle course consists of an up/down ramp with light alignment, a 3 ft wide chute with two bends and a 12" wall separated in 6" increments. As for the delivery of the medical kit, WALL-E successfully detects, maneuvers towards and deposits the item for a "trapped victim" by a light source.

The project was also completed $254.67 under the allotted budget of $750.00.

2 Introduction

The vision was simple: keep the design as simple as possible with the sole goal of reliably and efficiently maneuvering through the obstacle course. As we considered numerous candidate designs, emphasis was put on minimising mechanisms to reduce potential sources of failure. The quote, "If it isn't broken, don't fix it", best summarises our approach to the project. Ultimately, the decision was made to implement a tread design with a raised front lip. This design addressed our two main concerns of the SaRR's competency with wall breaching and its general grip throughout the obstacle course. These have consistently proved to be two of the most common issues in the past.

Numerous approaches have been taken, to varying degrees of success, in the design of robots for retrieval and obstacle-breaching applications. Three examples were found from the literature to analyse the approaches that experts have taken in the past. The robot by Wolf et al. [1] (*Fig. 1*) presents a relatively simple design

Figure 1: A mobile search and rescue robot. Taken from Figure 1 of "Design and control of a mobile hyper-redundant urban search and rescue robot" by Wolf et al.
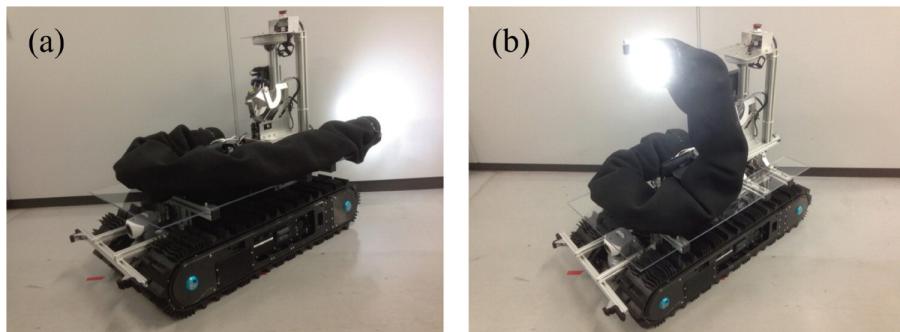

Figure 2: A SaRR with a snake robot on a mobile base. Taken from Figure 4 of "Development of a separable search-and-rescue robot composed of a mobile robot and a snake robot" by Kamegawa et al.
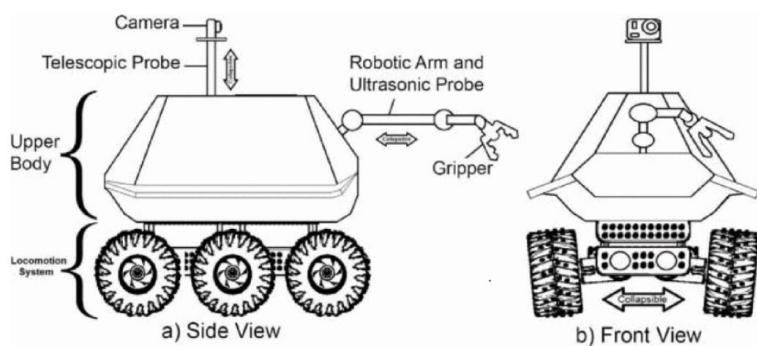

Figure 3: A mobile search and rescue robot. Taken from Figure 2 of "Design and implementation of a semi-autonomous mobile search and rescue robot: SALVOR" by Denker & İşeri.

(a) First stage.   (b) Second stage.   (c) Third stage.
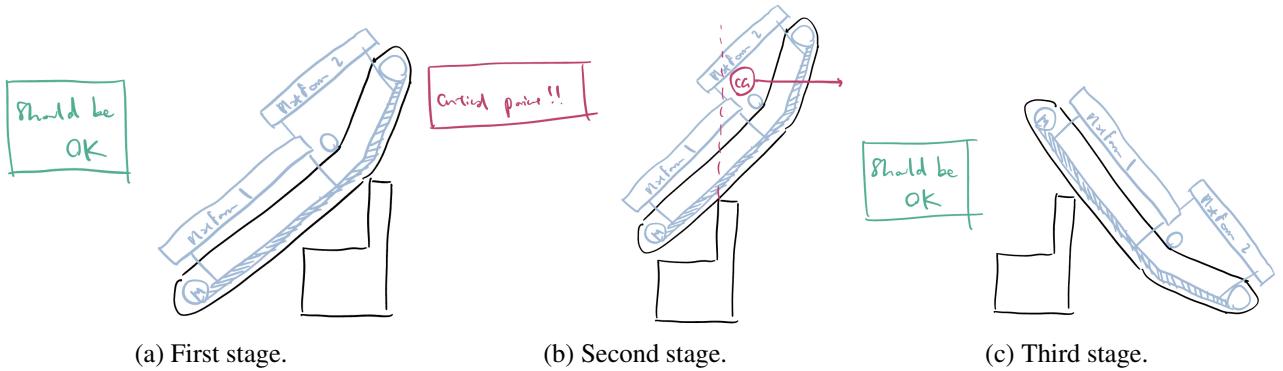
Figure 4: Three critical stages of wall breaching.
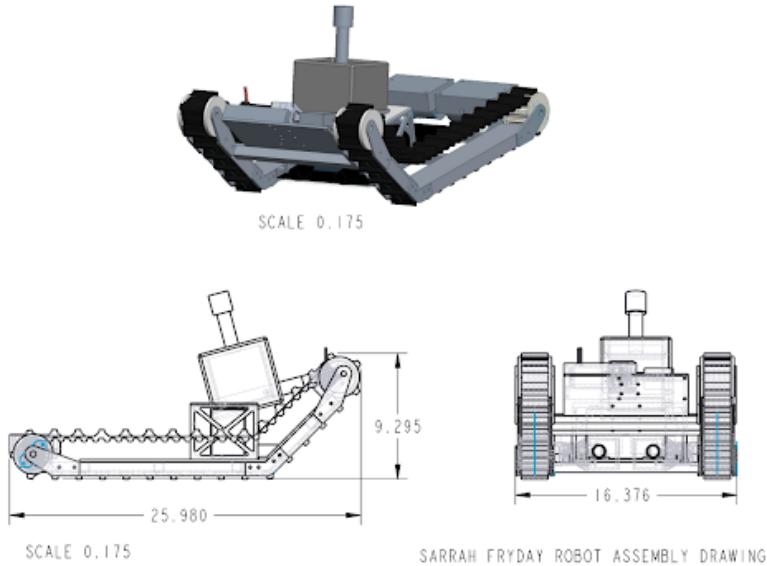
Figure 5: CAD drawing of the full, final robot.

with four robust wheels and a solid chassis. The internal components are covered and difficult to see but there are sensors on top and some complex mechanical elements such as suspension. This was likely necessary for operation on rough terrain. It also has a snake-like arm mechanism, presumably to reach deeper into inaccessible areas. Figure 2 shows a combined robot, with a snake robot mounted on a mobile base, developed by Kamegawa et al. [2]. The mobile base is of a horizontal tread design with a solid chassis, which sensors mounted in a high position for navigation. The snake robot is able to detach and maneuver independently into inaccessible areas. Finally, Denker and İşeri [3] present a 6-wheel approach (*Fig. 3*) in their paper, where the wheels are similar to those of Wolf et al. Whilst the internal components are hidden by a protective shell, a camera protrudes from the top and a robotic arm, with an ultrasonic probe and a gripper, reaches forward for intricate tasks. It should be highlighted that all of these designs possess certain common features: simplicity, robustness, wheels/treads with large teeth for good grip, protruding sensors and a slender mechanism to perform intricate tasks in tight areas.

In terms of wall breaching, three critical stages of the process were identified. First, and arguably the most important of all, is the "catching" of the first 6" step of the wall. We deemed it of utmost priority that the robot could reliably grip onto the first step and pull itself up the wall, and a robust solution was found by setting the
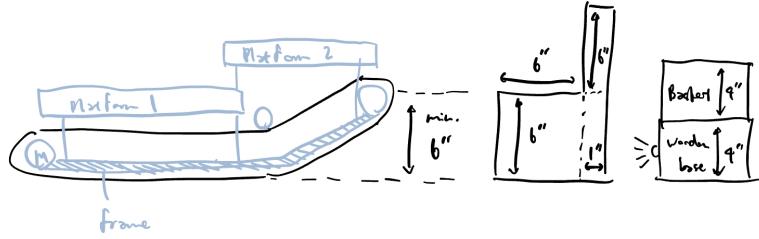
Figure 6: Preliminary side view drawing of the SaRR, adjacent and in comparison to the wall, the light source and the MedKit basket.

raised front lip to a minimum height of 6" and using treads of large teeth (*Fig. 4a*). Second, at the moment the back of the robot loses contact with the top of the first 6" step, it must be ensured that the centre of gravity (CG) of the body is to the right of the wall so as to allow the SaRR to fall forward instead of backward into an unrecoverable orientation—this is visualised in Figure 4b. CG analysis of the full CAD model (*Fig. 5*) and real-life testing confirms that the SaRR design satisfies this critical requirement. Finally, it was necessary that the SaRR's body was longer than the height of the 1 ft wall to ensured a controlled, upright descent after peaking over the wall (*Fig. 4c*).

With regards to the general grip, the skid geometry was designed to ensure a large area of contact to apply downward force on the treads. However, the large contact area also raised concerns for frictional forces and torque sufficiency, especially during wall breaching. This issue was mitigated by implementing slippery UHMW polyethylene material for the skids. Furthermore, to prevent derailment of the treads due to strong side ward forces during rotation, the skids incorporated U-shaped side lips.

Other aspects of consideration included the width and height of the robot, as well as the MedKit deposition mechanism. The SaRR could not be too wide so as to hinder its ability to safely navigate through the chute. Therefore, brushless motors were purchased externally to significantly minimise the body width whilst allowing for direct drive of both wheels. The height had to be low enough so as to pass underneath the overhang bar above the wall, although the low height of the tread design was more advantageous with regards to CG and stability. It also allowed for low positioning of the light sensors, which was necessary to reduce signal disruption from glare and reflection. Likewise, this height was optimal for the MedKit deposition mechanism, as it was on the same level as the MedKit collection basket. The alternative to a tread design was a wheel design, where there would be a minimum wheel radius of 6" to get over the first step of the wall, which would have required a significantly higher position of the robot platform. A larger wheel would have also required more torque from the motors.

With all the above factors in consideration, the preliminary design (*Fig. 6*) was created. The final CAD model, from several different angles, is shown in Figure 5. The team's emphasis on simplicity is reflected in the small number of moving parts and basic geometry of the frame and chassis. Robustness is also highlighted in the rigid structure of the chassis, constructed with hollow aluminium square tubes. The team strongly believed that it was of critical importance to lay a solid foundation from which to build the rest of the SaRR off of.

The simplistic design, in particular the small number of moving parts, also minimises weight and provided size flexibility to optimise for the specific obstacle course.

With regard to existing standards and codes, the team focused on addressing only the requirements necessitated by the MAE 322 SaRR Course. All design decisions were based on specific situational conditions and demands. However, upon researching industry and government-level search and rescue robot requirements, the team found that our simple approach to design, based on only the specific situational purpose of a robot, is largely reminiscent of the design and standard-generation process in more critical, real-life applications.

3 Specifications

3.1 SaRR Specifications

3.1.1 Weight

- Battery: 0.75 lbs

- Motors: 2x Neo Motor (1.876 lbs), 2x Versa Planetary Gearbox (1.48 lbs)

- Electronics: 1.5 lbs

- Drivetrain: 20 lbs

3.1.2 Size

- Length: 25.980"

- Height: 9.295"

- Width: 16.376"

- Wheel Diameter: 2.400"

3.1.3 Course Estimations

- Time to complete course: $1 \min 40 \, \mathrm{s}$ in autonomous mode; $\approx 30 \, \mathrm{s}$ in manual mode

- Max Speed: $0.762 \, \mathrm{m \, s^{-1}}$ / 1.705 mph

- Estimated Range: From energy calculations with a large factor of safety overhead, an estimated 2 repetitions can be done of the course if done fully autonomous. However, in practice, the LiPo battery charge can hold for at least 10-15 runs of the course fully autonomously before experiencing dips in "nominal" voltage (range where the torque output and rpm of motors is relatively consistent), and consequently performance.
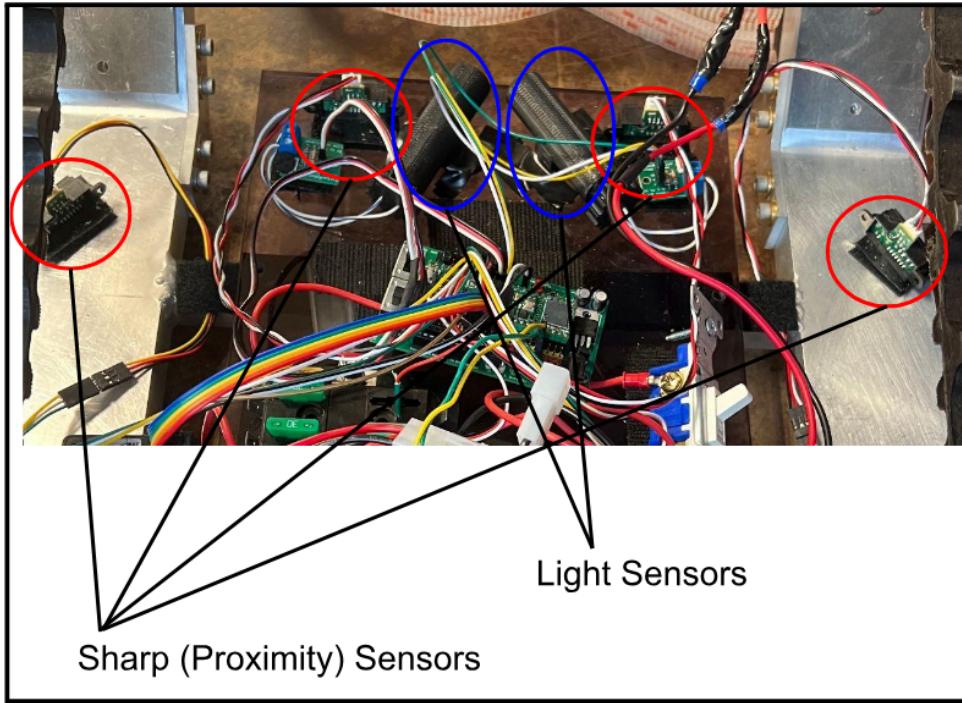
Figure 7: Diagram of Sensor Placement

3.2 SaRR Electrical: Sensor Configuration and Wiring

In consideration of our design priority - cost and simplicity - our team decided to stick with the same sensors as the sample robot. Although these sensors are capable of only basic functionalities, they can work reliably in the low-light conditions required of the task at hand. Additionally, the benefit of using the in-house sensors, is that they are in-house - meaning our group would not need to purchase any further sensors, minimizing the potential uncertainty of extra costs, sensors not working as they should, and the simple fact that we can easily just grab another sensor from the storage room if needed. Specifically, our SaRR employs the usage of 2 light sensors and 4 sharp sensors. The light sensors work by hooking up photoresistors - resistive elements that can vary resistance with light exposure, to a simple logic chip; on the other hand, the sharp sensors work by reflecting and detecting emitted infrared light.

As shown in the Figure 7, the SaRR has 2 forward-facing light sensors pointed in a 'cross' configuration (right sensor directed leftwards, and left sensor directed rightwards). Ultimately, the robot will need to detect a light source, but also navigate towards it - the 'cross' configuration allows for a wider sweep, or 'field-of-view,' such that the sensors can acquire a light source in the vicinity and move towards it whilst balancing out the light sensor values. As will be the case for most of our navigation code, "balancing sensor values" entails measuring sensor values from each pair of respective sensors, calculating the difference between them, and if this difference exceeds a tolerance threshold, turn left or right accordingly. To stop in front of objects, such as the wall, or the medkit basket, two forward-facing sharp (proximity) sensors were utilized to detect whether the SaRR is close enough, and aligned to the object. In the wall case, the SaRR will execute the wall climbing code, and in the medkit basket case, the medkit arm will be deployed to release the payload. Lastly, two

side-facing sharp sensors were also used to avoid wall collisions inside the chute. These sensors were angled slightly forward: from testing, it was shown that this configuration allowed for more 'preemptive' navigation in the sense that the robot would be able to detect changes in the path ahead earlier. In addition, this would also eliminate some corner cases, such as when the SaRR may be perfect in the middle but oriented at an angle - in which the logic may think the robot is aligned correctly because the side sensor values are still balanced, leading to a more oscillatory search behavior inside the chute.

Most of our electrical parts were reused from the sample robot (switch, fuse box, ground terminal, sensors, Teensy 4.1). Consequently, the wiring remained largely the same. Figure 8 presents an overall wiring schematic.

3.3 SaRR Operational Modes, Autonomy and Control System

Our SaRR was able to achieve 'full autonomy' - meaning that our robot was able to navigate the MAE322 obstacle course end-to-end with no manual input. The obstacle course involved traversing a ramp, a 3 ft wide chute, a wall consisting of 2 steps and a drop, and lastly, navigating towards a medkit basket to deploy a payload. Our team was able to achieve this by breaking up the course navigation logic into two main branches - which were toggled between depending on the measured photosensor values. In the darker parts of the course where no light was present, the code would default to tasks such as navigating the chute and traversing the wall. On the other hand, when a light source was detected, the code would switch to either navigating the ramp or finding the medkit basket. This segmentation allowed the team to upload a singular script that could handle both cases at the same time without a need to switch between autonomous operational modes. The robot was also able to differentiate between different tasks within the 2 branches by storing global booleans that 'check-marked' certain tasks from the course being completed. For example, to stop in front of the medkit basket, our team's robot was equipped with 2 forward-facing sharp sensors. However, upon approaching the ramp, the proximity values would spike in response to the ramp's incline, potentially mistaking it for the medkit basket. To solve this problem, we added a check for whether the robot had gone over the wall: this gave the robot enough information to decide whether or not to deploy the medkit.

The bulk of our navigational autonomy hinged on balancing sensor values between a pair of 2 cross-directional photosensors, 2 side-facing proximity sensors, and 2 forward-facing proximity sensors. If the difference between two sensors crossed a tolerance threshold, the robot would be instructed to move in the opposing direction to rebalance the readings. For example, for the chute navigation, if the right-facing proximity sensor value was higher than the left-facing proximity sensor, the robot would turn left until the value difference was back within tolerance. A high-level block diagram of our robot's full logic can be seen in Figure 9.
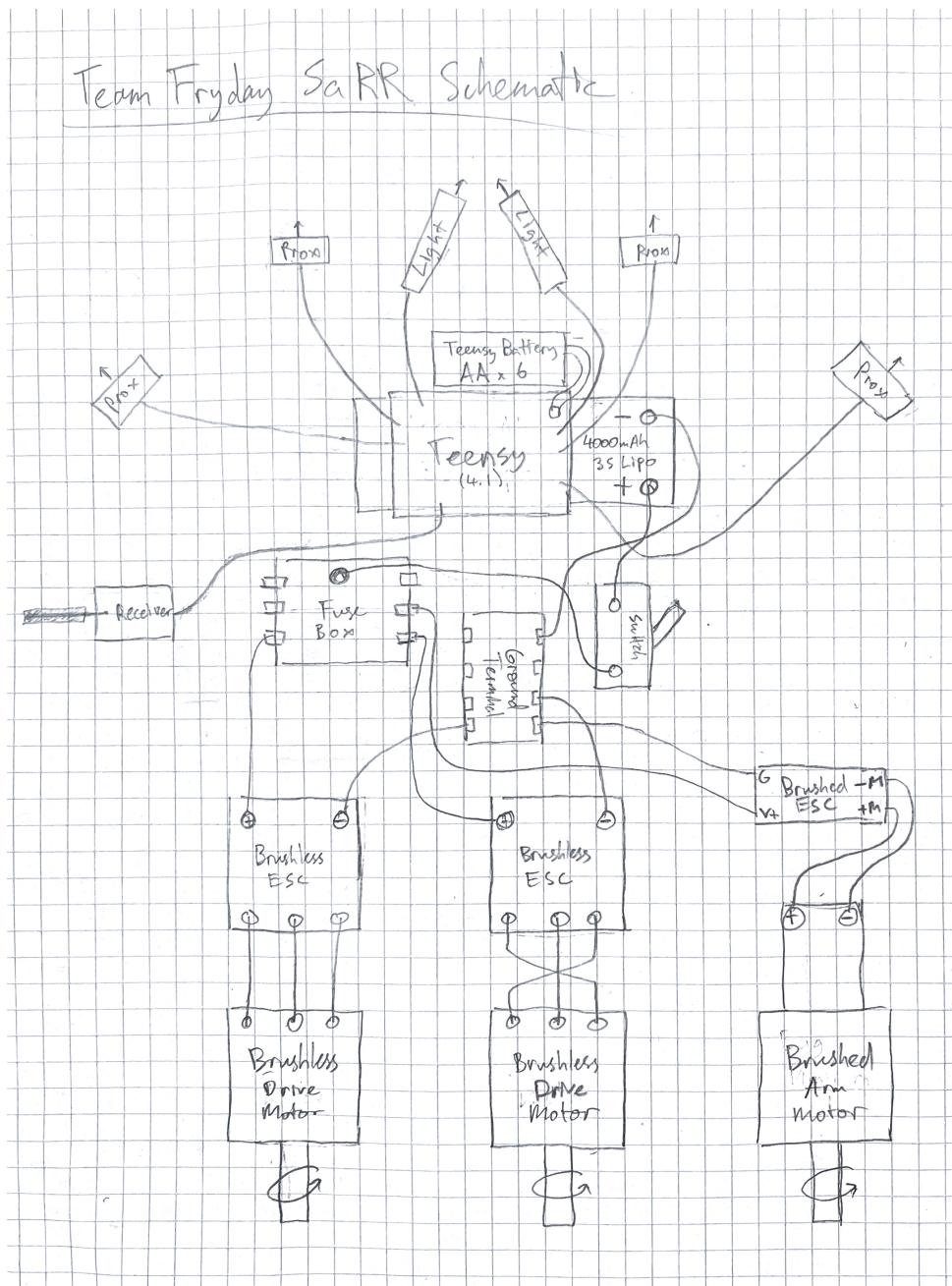
Figure 8: Wiring Schematic of Team Fryday's SaRR, Wall-E. It should be noted that the lines only refer to general electrical connections. For example, the connection between a proximity sensor and the teensy will require power, ground, and signal lines, whereas a line in between the switch and the fuse box is simply a singular power cable.
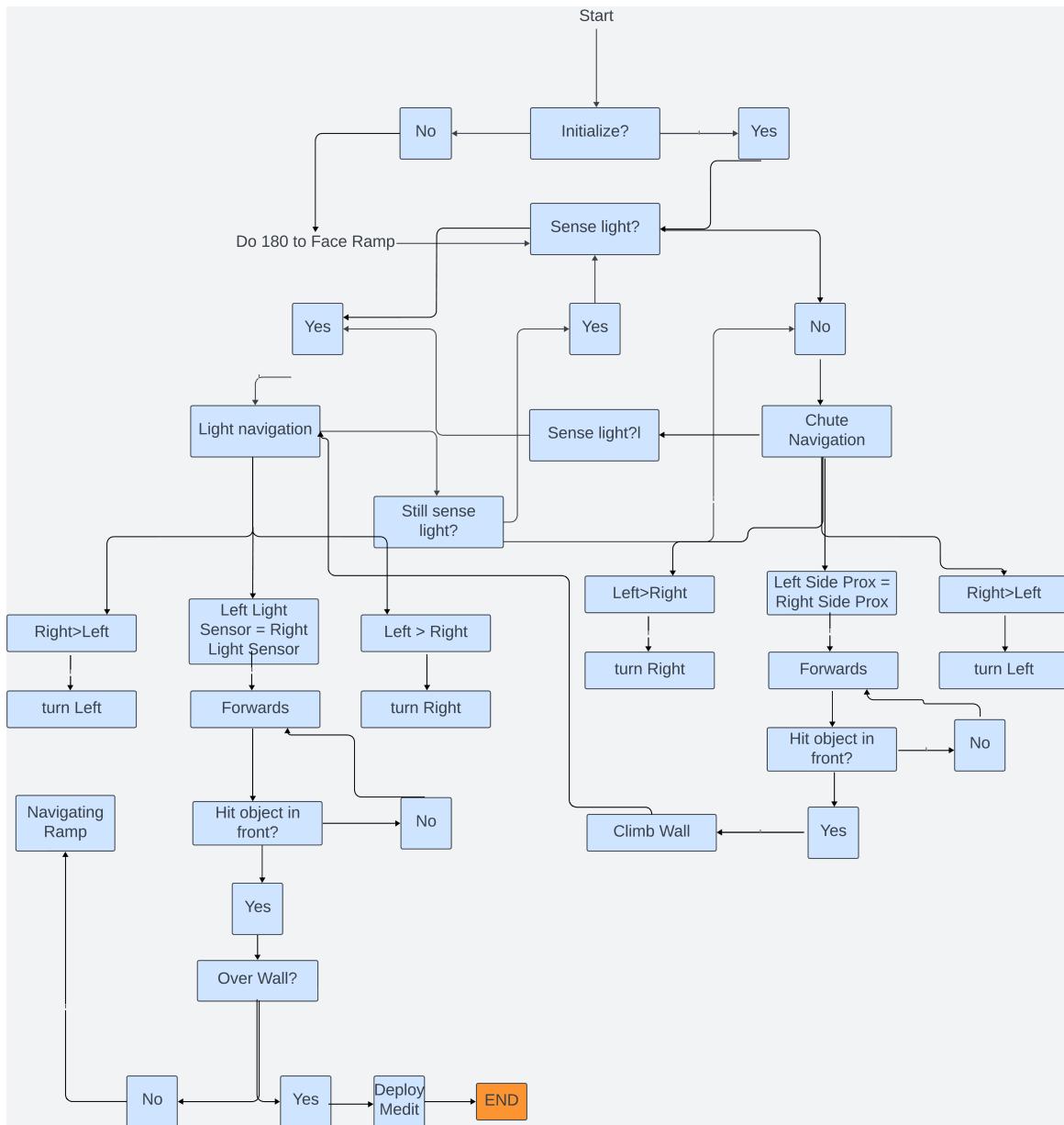
Figure 9: High-Level Logic Block Diagram
Full Code: https://github.com/jt7347/SaRR_Fryday

4 Project Management

4.1 Team Roles and Responsibilities

The team, of nine members, was structured into four subteams, with a project lead, a hardware lead and a software lead to ensure steady progress towards scheduled objectives and ultimate completion.

The project lead was Kazuki Tojo, the hardware lead was Jasper Waldman and the software lead was Jimmy Tran. The project lead overlooked the higher level aspects, ensuring that the project progressed and was completed within time, financial, logistical and technical limitations. The project lead constantly kept these wide range of requirements in consideration, steering the team if prioritisation in a certain direction became necessary. The hardware lead overlooked the design and manufacturing of the SaRR to satisfy its technical requirements. The software lead overlooked the implementation and testing of the SaRR's autonomous code to achieve full autonomy. As the SaRR's requirements were inherently intertwined with each other, the three leads worked to strike a delicate balance to satisfy each with minimal time, financial and logistical costs.

The team was divided into four subteams: drivetrain, MedKit deposition, electronics and components. The drivetrain subteam, consisting of Jasper Waldman and Luc Harbers, was responsible for ensuring the completion of the drivetrain, including but not limited to the robot frame, treads, wheels and belts, whilst meeting requirements such as dimensions, weight and centre of gravity (CG). The MedKit deposition subteam, consisting of Sarah Fry and Shayan Alavi, was responsible for developing the mechanism to deposit the MedKit. The electronics subteam, consisting of Calvin Nguyen, Jimmy Tran and Rihan Sajid, held responsibility for the light sensors, proximity sensors and code implementation to ensure reliable and efficient autonomous navigation through the course and to the "trapped victim." The components subteam, consisting of Ayanna Smith and Kazuki Tojo, was responsible for identifying and purchasing components, such as stock materials and motors, that satisfied technical requirements whilst remaining within the budget.

Despite these assignments, the team operated fluidly. Due to the nature of the project, each subteam had periods of concentrated work and other periods with less tasks. As such, whilst each subteam took lead in their respective tasks, the whole team worked dynamically to assist in various aspects of the project.

The roles and responsibilities of Team FRYday are available in Table 1. The contributions of each member to this FDR are available in Table 2.

4.2 Tasks and Milestones

A Gantt chart was used to visually organise the tasks and timeline of this project. Figure 10 shows a cropped section of the chart that displays the tasks, dates, members involved and hours estimated. In total, the project was estimated to take a total of 852 hours. The full Gantt chart can be seen in Figure 32 of Appendix A.

In comparison, Figure 11 displays the key tasks and their timeline, upon completion of the project. Whilst the CAD model was prepared slightly in advance of the expected date, the drivetrain manufacturing was executed slightly differently to what was initially planned. The initial plan was to have a first iteration of the

No	Task	Start	Days	End		Hours	#Members	Total Hours	Luc	Kazuki	Jasper	Calvin	Shayan	Sarah	Rihan	Ayanna	Jimmy
0	**Demonstrate Drive Train (Open-Loop)**	**25-Sep**	**5**	**29-Sep**	==												
1	Lab x4	25-Sep	5	29-Sep	x	12	9	108	x	x	x	x	x	x	x	x	x
100	**Demonstrate Autonomous Navigation to Light Source**	**2-Oct**	**4**	**6-Oct**	==												
101	Lab	2-Oct	4	6-Oct	x	3	9	27	x	x	x	x	x	x	x	x	x
200	**Preliminary Design Review**	**16-Oct**	**8**	**24-Oct**	==												
201	Executive Summary	19-Oct	1	20-Oct	x	1	1	1		x							
202	Introduction	21-Oct	3	24-Oct	x	2	1	2		x							
203	Specification	21-Oct	3	24-Oct	x	3	9	27	x	x	x	x	x	x	x	x	x
204	Detailed Design and Analysis	21-Oct	3	24-Oct	x	3	9	27	x	x	x	x	x	x	x	x	x
205	Project Management	20-Oct	1	21-Oct	x	5	1	5		x							
300	**Manufacture 1st Iteration of Drivetrain (DT)**	**25-Oct**	**16**	**10-Nov**	==												
301	Rough CAD model (to set dimensions)	25-Oct	2	27-Oct	x	6	6	36	x		x	x			x	x	x
302	Cardboard model (for course testing)	28-Oct	1	29-Oct	x	2	4	8					x		x	x	x
303	Refined CAD model	28-Oct	3	31-Oct	x	3	2	6	x		x						
304	Manufacturing	1-Nov	6	7-Nov	x	6	6	36	x		x	x			x	x	x
305	Components purchasing	25-Oct	4	29-Oct	x	2	2	4		x						x	
306	Electronics mounting	8-Nov	2	10-Nov	x	4	5	20				x	x	x	x		x
307	Assembly	8-Nov	2	10-Nov	x	3	9	27	x	x	x	x	x	x	x	x	x
308	Med kit mechanism (MKM) CAD model	25-Oct	16	10-Nov	x	4	2	8						x	x		
309	TRAVERSE RAMP AND CHUTE	10-Nov	0	10-Nov	x	2	9	18	x	x	x	x	x	x	x	x	x
400	**2nd Iteration of DT**	**11-Nov**	**6**	**17-Nov**	==												
401	Identify areas for improvement DT	11-Nov	1	12-Nov	x	2	9	18	x	x	x	x	x	x	x	x	x
402	Make adjustments to DT	13-Nov	3	16-Nov	x	8	7	56	x	x	x	x			x	x	x
403	Manufacture MKM	11-Nov	5	16-Nov	x	6	2	12					x	x			
404	TRAVERSE WALL (Open Loop)	17-Nov	0	17-Nov	x	2	9	18	x	x	x	x	x	x	x	x	x
500	**3rd Iteration of DT**	**18-Nov**	**13**	**1-Dec**	==												
501	Identify areas for improvement DT	18-Nov	2	20-Nov	x	2	9	18	x	x	x	x	x	x	x	x	x
502	Make adjustments to DT	21-Nov	9	30-Nov	x	6	7	42	x	x	x	x			x	x	x
503	Install MKM	18-Nov	2	20-Nov	x	3	2	6					x	x			
504	Identify + make adjustements to MKM	21-Nov	9	30-Nov	x	8	2	16					x	x			
505	AUTONOMOUS OBJECT PLACEMENT	1-Dec	0	1-Dec	x	2	9	18	x	x	x	x	x	x	x	x	x
600	**Final Iteration of DT**	**2-Dec**	**8**	**10-Dec**	==												
601	Identify areas for improvement	2-Dec	1	3-Dec	x	2	9	18	x	x	x	x	x	x	x	x	x
602	Make adjustments	4-Dec	3	7-Dec	x	6	9	54	x	x	x	x	x	x	x	x	x
603	FULL-COURSE AUTONOMOUS TEST	8-Dec	0	8-Dec	x	3	9	27	x	x	x	x	x	x	x	x	x
604	Identify + make adjustments	9-Dec	0	9-Dec	x	6	9	54	x	x	x	x	x	x	x	x	x
605	FULL-COURSE AUTONOMOUS TEST ver2	10-Dec	0	10-Dec	x	3	9	27	x	x	x	x	x	x	x	x	x
600	**Final Full-Course Rehearsal**	**11-Dec**	**0**	**11-Dec**	==												
601	SaRRah Fry will slay the course	11-Dec	0	11-Dec	x	3	9	27	x	x	x	x	x	x	x	x	x
700	**Final Presentation**	**8-Dec**	**7**	**15-Dec**	==												
701	FDR preparation (more details closer to date)	8-Dec	6	14-Dec	x	8	9	72	x	x	x	x	x	x	x	x	x
702	Team FRYday will slay the presentation	15-Dec	0	15-Dec	x	1	9	9	x	x	x	x	x	x	x	x	x
	Project Duration	**2-Oct**	**74**	**15-Dec**				852									

Figure 10: Cropped section of the Gantt chart used to organise the project.

12

Key Tasks	Hours	Timeline	Completion	Member(s)
Initial CAD design	100	28-Oct	Yes	Jasper Waldman, Kazuki Tojo, Sarah Fry, Shayan Alavi
Drivetrain manufacturing	500	20-Nov	Yes	Jasper Waldman, Luc Harbers, Ayanna Smith, Rihan Sajid, Sarah Fry, Jimmy Tran, Calvin Nguyen, Shayan Alavi, Kazuki Tojo
Manual control	20	20-Nov	Yes	Jimmy Tran, Calvin Nguyen, Rihan Sajid
Wall breaching (autonomous)	20	20-Nov	Yes	Jimmy Tran, Luc Harbers, Kazuk Tojo
Medkit mechanism	100	28-Nov	Yes	Sarah Fry, Shayan Alavi
Light navigation (autonomous)	20	28-Nov	Yes	Jimmy Tran, Sarah Fry, Shayan Alavi
Chute navigation (autonomous)	20	29-Nov	Yes	Jimmy Tran, Luc Harbers, Jasper Waldman
Ramp navigation (autonomous)	50	4-Dec	Yes	Jimmy Tran, Jasper Waldman, Kazuki Tojo
Static/dynamic analysis	50	8-Dec	Yes	Ayanna Smith, Rihan Sajid, Calvin Nguyen
Full autonomy	100	9-Dec	Yes	Jimmy Tran, Sarah Fry, Jasper Waldman, Kazuki Tojo
TOTAL	980			

Figure 11: Post-completion reflection of key tasks and their timeline.

drivetrain ready for 10-Nov, followed by a three more iterations for a final completion date of 7-Dec. However, we chose to prioritise robust manufacturing practices and achieved WALL-E largely in one iteration, reaching completion almost three weeks in advance on 20-Nov, before Thanksgiving break. After Thanksgiving break, the team first achieved autonomous navigation through each individual obstacle, before completing full-course autonomy on 9-Dec.

4.3 Risk Handling

Due to the team's strong emphasis on front-loading the project, there was sufficient buffer for unpredictable circumstances such as part-breaking and delivery delays. The team rapidly reacted to such circumstances, fully aware that procrastination could lead to delays on the scale of days or weeks. Perhaps the only challenge we did not initially account for was the difficulty of autonomous navigation over the ramp, which seemed simple but required precise tuning of autonomous parameters to prevent the robot from falling over the edge. Unlike chute or light navigation which implemented proximity and light sensors, there was no simple way to detect the edge of the ramp.

4.4 Cost Analysis

The team was allotted a budget of $750.00. We initially estimated a realistic spending of $500.00, accounting for purchasing mistakes and component malfunctions. The final cost was $495.33, very close to the estimated spending and $254.67 under the project budget. The full breakdown of the costs is presented in Table 3.

Item	Costs
Unidirectional drone ESC x2	$43.98
NEO brushless motor x3	$144.00
Architectural 6063 Aluminum U-Channel (8 ft. length)	$70.32
Slippery UHMW Polyethylene Bar (5 ft. length)	$52.25
Architectural 6063 Aluminum Rectangular Tube (6 ft. length)	$16.62
Architectural 6063 Aluminum 90 Degree Angle (4 ft. length)	$22.97
Clear Scratch- and UV-Resistant Cast Acrylic Sheet (12" x 12")	$15.30
Clear Cast Acrylic Sheet (12" x 24")	$21.13
Zinc-Plated Steel Pan Head Phillips Screw (Pack of 100)	$8.44
Heat-Set Inserts for Plastic (Pack of 5)	$6.57
18-8 Stainless Steel Hex Drive Flat Head Screw (Pack of 100)	$13.77
Bidirectional RC ESC x2	$79.98
Total	**$495.33**

Table 3: Full breakdown of project costs.

4.5 Overall Management Approach

The team certainly approached this project with an emphasis on trust and delegation. The team understood the scale and complexity of the requirements of the project, as well as the resultant necessity to divide the workload and respectively specialise in certain aspects with major demands. The project and technical leads served not as figures of authority but rather as strings that connect the various areas of delegation, ensuring that even progress was made to fulfill the numerous technical and scheduling requirements of the project.

The team met every week, usually after one of the Monday or Wednesday lectures. The project lead usually led the meeting, but it was ensured that everyone was given the opportunity to voice their thoughts before moving on from any agenda. Input from the whole team was vital to ensure a range of perspectives. It also allowed the team to identify issues as early as possible, often before they even became problems—prevention was always preferred to intervention. However, when an issue did inevitably arise, the responsible subteam took the lead in devising and delegating a solution plan. The significance of regular meetings was understood to avoid cramming before deadlines. This was particularly important, as the nature of the project meant that, in many situations, cramming was not an option. Purchase orders were sometimes delayed, and seemingly insignificant issues occasionally revealed severe consequences and required hours of fixing.

Whilst all measures possible were taken to prevent minor problems from evolving into critical issues, it was inevitable at times. In the event of a critical issue, a meeting, led by the project lead and technical lead, was held with the entire team, to raise ideas for potential solutions. The pros and cons of each idea were be discussed and the group agreed on a final solution, optimal for the time, financial and technical demands at the time.

Throughout the project, a Slack channel was used for constant communication between all team members. Slack was initially identified as the optimal method of communication due to its functionality to create multiple specialised channels, as well as to communicate directly with specific team members whilst keeping everyone

else in the loop. In reflection, the Slack channel was integral to the success of Team FRYday.

5 Detailed Design and Analysis

5.1 Drawings

Engineering drawings of all parts of WALL-E are presented in Appendix A. These are Figures 23, 24, 25, 26, 27, 28, 30 and 31.
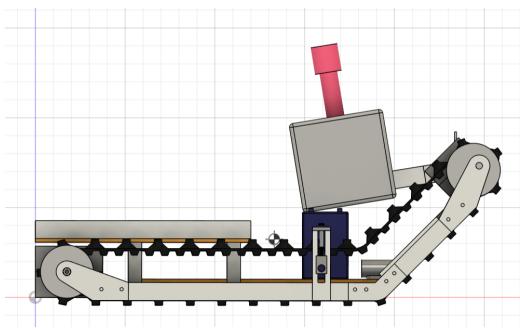
5.2 Weights and Centres of Mass

Mass Calculations:
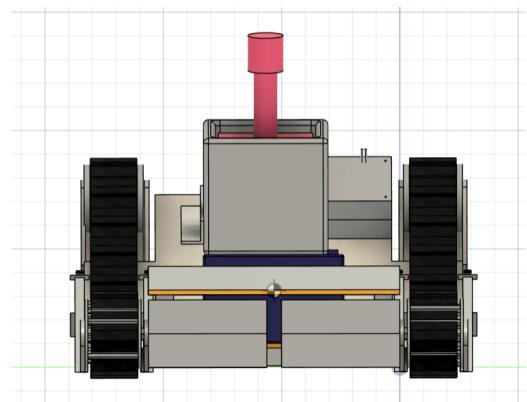
Name	Material	Volume(inch^3)	Density(lbs/inch^3)	Amount	Mass (lbs)
Treads	Rubber	37.3	0.035	2	2.611
Skid	Al6063	25.4	0.0979	2	4.973
Front wheels	PVC	1.03	0.048	4	0.19776
chassis bar	Al 6063	1.89	0.0979	4	0.740124
Short chassis bar	Al 6063	0.386	0.0979	4	0.1511576
Platform base	Acrylic	23.2	0.0428	1	0.99296
Front plate	Al 6063	16.4	0.0979	1	1.60556
Battery	Specific	N/A	N/A	1	0.75
Gussets	Al 6063	0.481	0.0979	4	0.1883596
Back ears	Al 6063	0.99	0.0979	4	0.387684
Motor + gearbox	Specific	N/A	N/A	2	3.330236
Motor mounts	Polyurethane	15.95	0.04335	2	1.382865
Medkit arm	Al 6063	2.1	0.0979	1	0.20559
Medkit basket	PLA	56.09	0.0515	1	2.888635
Medkit coupler	Nylon	4.15	0.0415	1	0.172225
Medkit	Wood	N/A	N/A	1	2.2
Medkit motor + gearbox	Specific	N/A	N/A	2	1.323991
Medkit motor mount	Polyurethane	9.52	0.04335	1	0.412692
Tread wheels	Nylon	9.4	0.0415	2	0.7802

Total Mass: 25.3 lbs
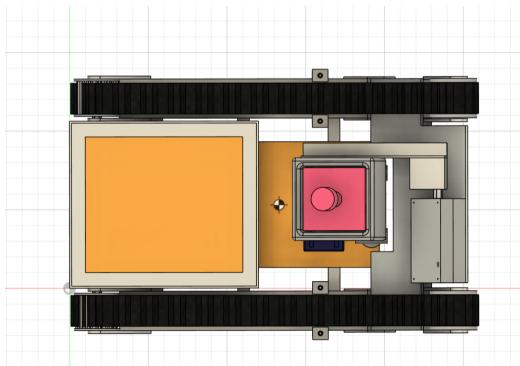
The Center of Mass, from 4 different views, is presented in Figure 12.
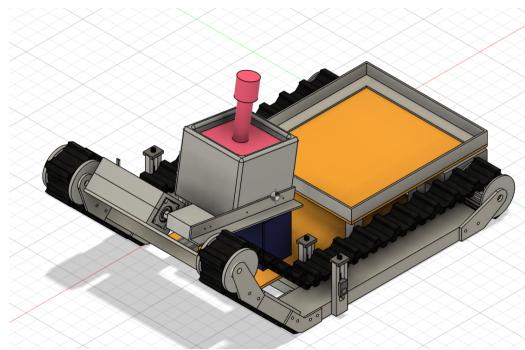
(a) Side view of COM.

(b) Front view of COM.

(c) Top down view of COM.

(d) Diagonal view of COM.

Figure 12: Center of Mass of robot.
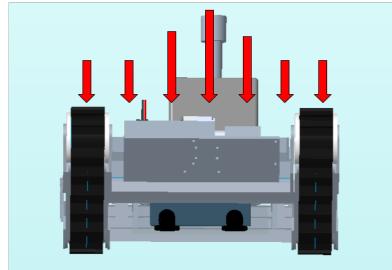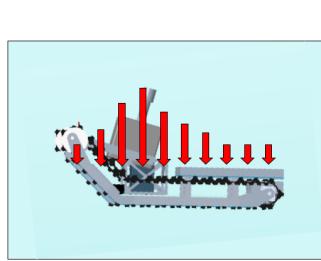
Figure 13: Weight distribution throughout the SaRR.
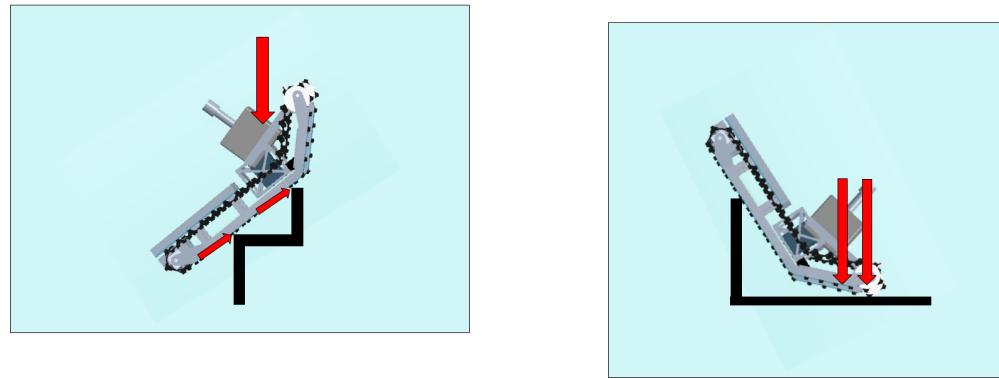
Figure 14: Forces involved during climbing and falling off wall.
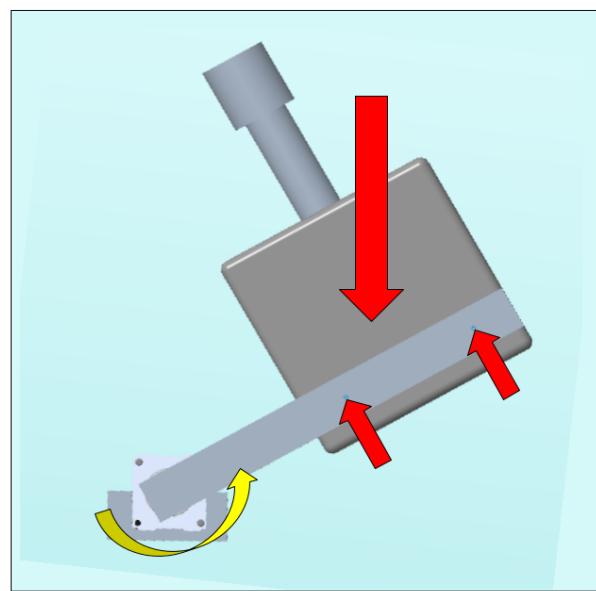

Figure 15: Medkit mechanism forces.
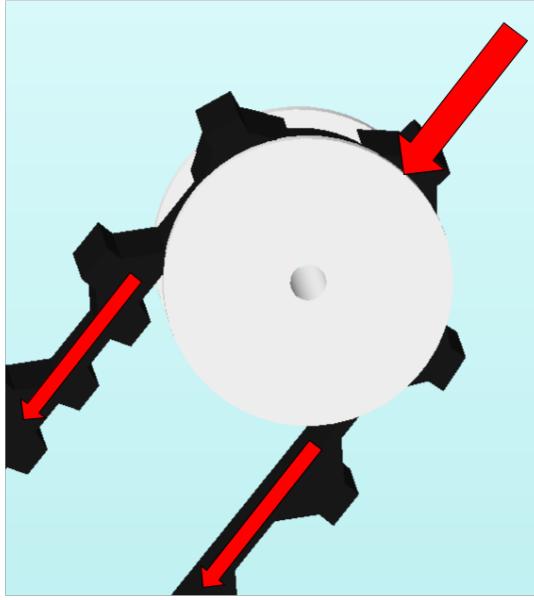
Figure 16: Forces applied on the treads.

5.3   Design Loads

The weight of the robot is spread out throughout the body of the robot, however, more weight is concentrated in the center of the robot due to the placement of the electronics and the Medkit (*Fig. 13*). This causes some shear stress where the electronics platform is welded to the outer frame, however, the weld is strong enough to overcome this force and not break. Once the robot begins to climb the stairs, the center of mass of the robot has to be accounted for in order to avoid it tipping over. The battery and Medkit mechanism were placed further in front of the robot to move the center of mass near the front of the robot, and therefore avoid it tipping over.

The biggest concern with forces comes from the robot hitting the ground once it breaches the wall. The weight of the robot falling is concentrated in the angled front of the robot (*Fig. 14*). This puts a lot of pressure on the bend in the outer frame. The aluminium is strong enough to withstand this impact for the first couple of occasions. The structural integrity of the outer frame is a possible concern when the robot is tested to take this impact many times.

The Medkit mechanism primarily deals with shear stress along the Medbasket, as the downward force of the weight of the Medkit and the basket is countered by the rotation of the motor, causing shear stress along the basket (*Fig. 15*). The screws that connect the basket is designed to absorb most of this stress, however, the basket does begin to crack over time.

Lastly, the wheels for the treads deal with great forces due to the treads pulling against them (*Fig. 16*). The movement and tension of the treads applies great force to the wheels as they rotate. This was an issue when testing with PLA plastic, as the plastic wheels would crack and break. These issues were mitigated once nylon wheels were incorporated instead of the PLA.
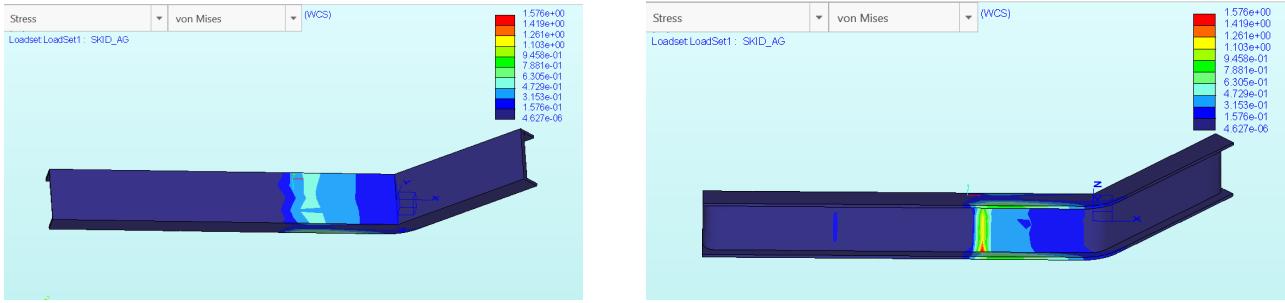
Figure 17: Stress on skid if made of complete Aluminum 6063
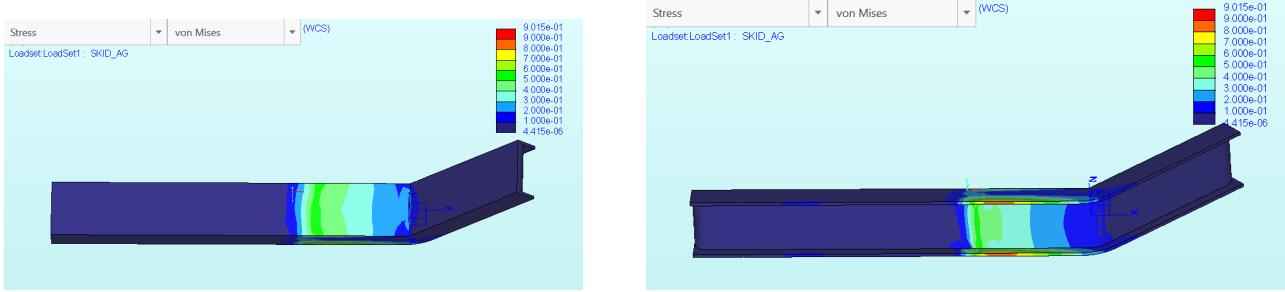
Figure 18: Stress on skid if made of complete Polyurethane

5.4 Structural Elements

The skids are the main components of our robot. They form the chassis as well as provide structure to the treads. They also hold the entire weight of the robot and thus deal with the most stress. Our initial thoughts for the material of the entire skid was aluminum 6063, however, we knew this would make the skid heavy as well as create more friction on the tread. Our other option for the materials was polyurethane, which would still keep shape, and also be lightweight.

Performing a structural analysis with the weight of our robot in the normal position showed that the highest points of stress on the polyurethane skid (*Fig. 18*) overall are lower than the highest points of stress on the aluminum 6063 (*Fig. 17*). However, specifically, the highest point of stress for the aluminum 6063 is on the middle part of the inner skid, with stresses between 1.1 ksi and 1.6 ksi. On the polyurethane, the highest point of stress is in the middle part of the outer skid border, with stresses between 0.7 and 0.9 ksi.

From this, we decided to use both aluminum 6063 and polyurethane for the skids. The outer part of the skids would be made of aluminum 6063 to keep strength while the inner part of the skids would be made of polyurethane to keep the skids lightweight and reduce the stress on the inner skid.

Next was simulating the structural analysis on the skid during the wall section (*Fig. 19*). After climbing the wall, the front of our robot drops from a height of about 20 inches. Simulating this would show the points of the skid that have high stress and thus, might fracture. The highest point of stress on the skid after the drop occurs at the section where the skid is bent. A large part of this bend on the outer skid experiences a stress between 0.3 ksi and 0.5 ksi, with the largest stress exactly at the bend being 0.9 ksi. The inner skid as well as
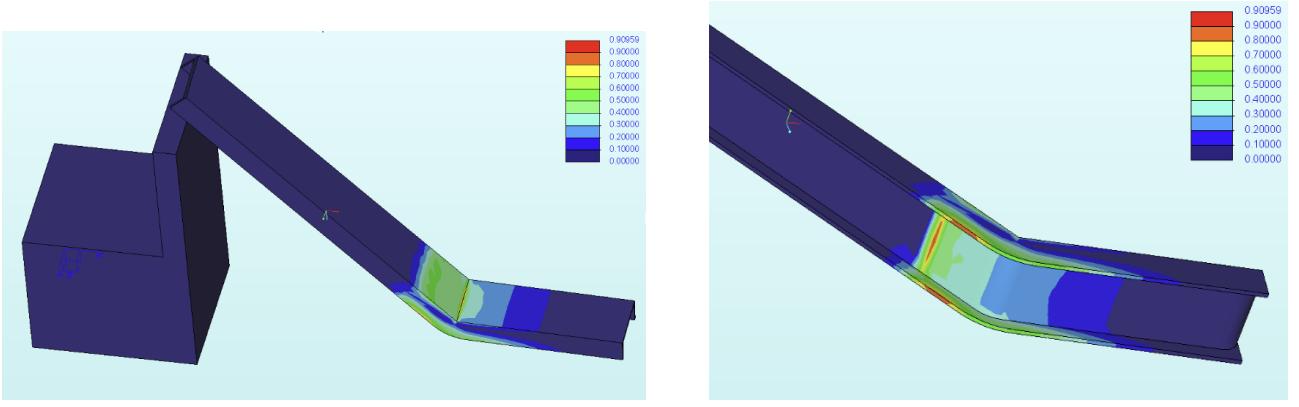
Figure 19: Stress on skid during drop from wall

the border of the outer skid experience the highest points of stress.

The highest stress the skid experiences is still considerably low. Given this, and that the tread is made of a type of rubber, we believe that the tread will absorb most of the force from the drop and the skid shouldn't fracture.
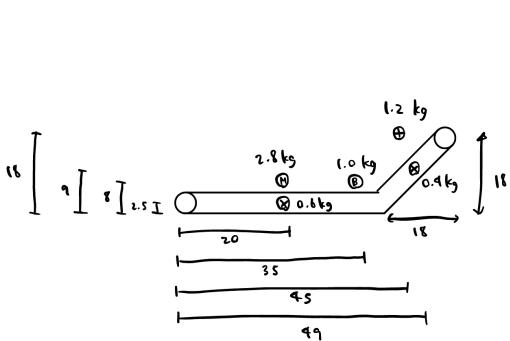
5.5 Dynamic Analysis

We conducted a preliminary analysis in order to determine whether our initial concept was feasible. Our main subject of analysis was getting over the wall, as we determined that it was the most technically challenging part of the course. We created sketches to understand how the robot would interact with the wall (*Fig. 6*), which allowed us to get general dimensions of the frame.
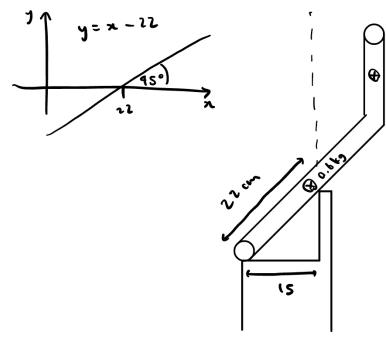
From this, we recognized that the front lip of the robot must at least be greater than the height of the first step, 6". This would allow the robot to easily mount the first step, and get started on the climb.

Using free body diagrams, with rough dimensions and estimated masses, we then considered the dynamics as the robot drove over the wall. There is a critical point as the robot drives over the wall when its rear loses contact with the bottom step. If at that point, the center of gravity is in front of the top step, it will make it over the wall. If the center of gravity is behind, the robot will fall back onto the top step, and may fall backwards in the worst case. A MATLAB code was written to simulate the CG based on the positioning of components within the robot. Different scenarios were experimented, such as front-wheel drive and back-wheel drive, and it was determined that there was sufficient component-positioning flexibility in achieving the necessary CG position. We also looked at the case where the robot has made it over the wall and is on the way down. In that scenario, if the robot is too short, it may tip end-over-front as the front of the robot falls to the ground. Keeping the robot long enough will make it so there is a point of contact with the ground and the wall until the robot is completely over. This is demonstrated in Figure 4.

The design tenet we extracted from this analysis was to keep the center of gravity forward and low. With these facts in mind, we then considered if it was feasible with this design to put the center of gravity in an optimal position. To do so we estimated locations and masses of the heaviest components. We considered the

(a) Mass Components and Rough Dimensions.

(b) Center of Gravity at Critical Point.

Figure 20: Static Analysis with Estimated Masses.

two sections of the frame, the motors, the battery, and the MedKit in Figure 20.

5.5.1 Motor Analysis

There was a large number of possible motors that we could have used for the drivetrain of our robot. The initial requirements for our motor were to have a high torque and power. A high torque allows the robot to drive through all sections of the course while holding its load and the load of the MedKit, especially over the breach wall which requires the most torque out of the course. The torque of the motor contributes to the power, of which high power means the SaRR will move quickly through the course. Out of a desire to keep the battery small, we sought to avoid a scenario in which the motor would have to take in a voltage higher than 12 V. We looked into DC motors as they are usually low in weight and cost for their high speed. The key factor in choosing a DC motor was its greater efficiency than an AC motor, as well as the fact that DC motor does not require an inverter, keeping the electronics simple.

Our initial choice was to use the motors that were used for our lab robots - the Mini CIM motor. The advantage to using this motor was that we knew it could work and we knew how it worked, so we could avoid spending time experimenting with a new motor and it possibly not working for multiple reasons (can't hold the load, too small, etc.). However, since it's important that the center of gravity (CG) of our robot stays in the front and the motors have to be placed in the back, we decided to use the NEO Brushless motor. The NEO motor is a much smaller motor than the CIM, meaning it also has less weight so the CG stays at the front. Another advantage to the NEO motors is they make the design of our drivetrain simpler. With the NEO motors, we don't need to have a pulley system connecting the motor and the wheel like we did for the CIM, we can instead just have the motor, gearbox, and wheel as a direct-drive system.

As previously stated, the breach wall would require the most torque from the robot so the estimations started with that. The angle the robot is at while riding the wall is about $52°$. The maximum weight of the SaRR is estimated to be 10 kg, and the additional MedKit of 1 kg makes the total max load 11 kg. Gravity (g) is 9.81 m/s^2. The coefficient of friction is 0.5 as this is the minimum required COF for flooring surfaces as stated by OSHA and the American Society for Testing and Materials. The radius of the wheels is 1.2 in. The force equations are then:
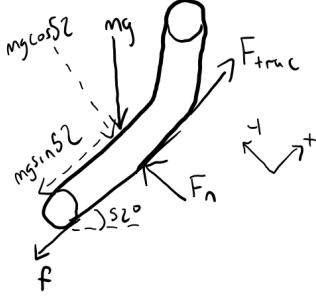
Figure 21: Forces acting on the robot while climbing up the wall

$$\sum F_x = 0 = F_{trac} - mgsin(52) - mgsin(52) * f$$

$$F_{trac} = (11kg)(9.8m/s^2)sin(52) + (11kg)(9.8m/s^2)sin(52)(0.5) = 128N$$

$$F_{trac} = T_{req}/r$$

$$T_{req} = rF_{trac} = (1.2in(0.0254m/1in))(128N) = 3.9Nm$$

With our required torque of 3.9 Nm, we then calculated the gear ratio we would need for the motor based on the max power. Looking at the NEO motor curve, at max power, the torque is about 1.35 Nm with a speed of about 2900 RPM.

$$GearRatio = T_{req}/T_{max} = 3.9Nm/1.35Nm = 2.9 = 3:1$$

$$OutputSpeed = Speed_{max}/GR = 2900RPM/3 = 967RPM$$

Using these specs, it can be seen that a 3:1 gear ratio would be best, however for our robot, this creates a speed that is too high. Thus, we had to increase the gear ratio to decrease the speed and make the wheels more controllable with the circuitry. Through testing, we found a 16:1 gear ratio to work best.

$$16:1; \ 3.9Nm/16 = 0.25Nm$$

$$OutputSpeed = 5400RPM/16 = 338RPM$$

Using this gear ratio, the torque required of the motor is 0.25 Nm which, looking at the motor curve, gives the max motor speed to be about 5400 RPM. The max output speed (speed of the wheel) is then about 338 RPM. Adding a safety factor of 2, to account for possible transmission or other losses, gives a required torque of 7.8 Nm. With this safety factor, the torque required of the motor is about 0.5 Nm, which gives a motor speed of 4700 RPM, which is then an output speed of 294 RPM or 0.938 m/s. Through driving the course, the actual max speed was shown to be 0.762 m/s which shows that there were more losses somewhere in either the
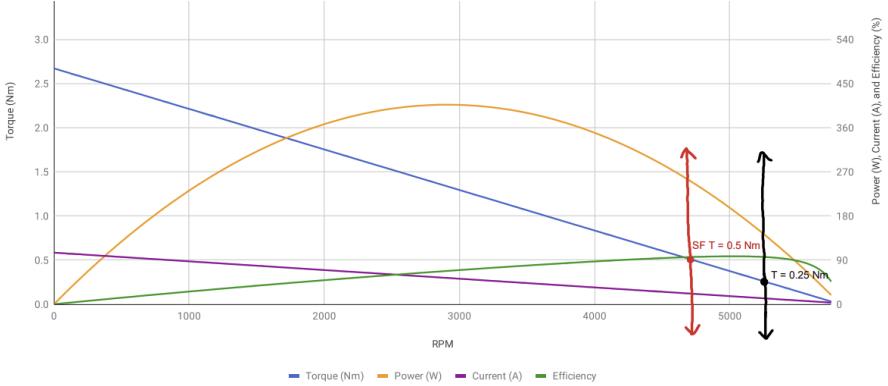
Figure 22: NEO Brushless Motor Curve displaying torque line for motor with 16:1 gearbox and safety factor torque line

mechanisms or circuitry.

A minor setback to using the NEO brushless motor was that the shaft had to be shortened to fit into the versaplanetary gearboxes we had. When trying to shorten one of the shafts, this resulted in the inner part of the shaft that connects to the motor breaking, of which we had to purchase another motor.

5.5.2 Battery Analysis

Our initial plan was to use the 12V, 84 Wh LiFeP04 Battery from lab for our SaRR for simplicity. However, the NEO brushless reversible ESCs purchased with the NEO motors were programmed to work specifically with LiPO batteries. LiPO batteries typically come in cells: a #S battery (where # denotes a number typically ranging from 1-6) denotes how many 3.7V cells are stacked in series. The RC Car ESCs had a piece of code to recognize the number of cells present when plugging in the battery and subsequently cut off the power at a certain voltage threshold to protect the battery and circuit. However, since the lab battery's voltage sat right between the cutoff threshold between a 3S and 4S battery (3S when low charge, and 4S when high charge), the ESCs would think the battery is a depleted 4S battery, and end up cutting the power. To solve this problem, we switched to a 4000 mAh 3S LiPO battery.

To determine if the battery will work with our SaRR, we calculated the total power our SaRR will use going through the course. We used a SaRR mass of 11 kg (SaRR max weight + MedKit weight), the max speed of the motors which is about 3800 RPM or 12 m/s, a motor efficiency of 40% and a drivetrain efficiency of 70%. First, we calculated the energy per acceleration at 100% theoretical efficiency using the kinetic energy equation: $KE = 1/2mv^2$. This gives us an energy of 792 J per acceleration. Next, we calculated the energy the SaRR would require at each section of the course:

$$\sum E = E_{nav1} + E_{ramp} + E_{chute} + E_{wall} + E_{nav2} + E_{dropoff}$$

Navigating to the ramp takes 1 acceleration: $E_{nav1} = 792J$. We estimated the elevation of the ramp to be no more than 1 ft and the ramp has an acceleration and a climb: $E_{ramp} = 1/2mv^2 + mgh = 792J +$

$(11kg)(9.81m/s^2)(1ft * 0.3048m/ft) = 825J$. The chute has three sections, so estimating that after each section, the SaRR has to turn then move forward. Rotations take 1/2 an acceleration: $E_{chute} = 1accel + 2(1rot + 1accel) = 792J + 2(396J + 792J) = 3168J$. The wall has an acceleration and a climb, however because movement up the wall may not be fully straight, we added a factor of three to the overall energy: $E_{wall} = 3(1/2mv^2+mgh) = 3(792J+(11kg)(9.81m/s^2)(0.3048m) = 2475J$. While testing, the navigation to the goal typically took 1 acceleration and about 6 rotations: $E_{nav2} = 792J) + 6(396J) = 3168J$. The MedKit dropoff involves the arm, which needs to raise and then go back: $E_{dropoff} = 2(1/2mv^2 + mgh) = 2(1/2(1kg)(0.12m/s)^2 + (1kg)(9.81m/s^2)(0.3048m)) = 6J$.

$$\sum E = 792J + 825J + 3168J + 2475J + 3168J + 6J = 10434J$$

Since this is the energy at 100% efficiency, we have to divide this energy total by the motor and drivetrain efficiencies. We also multiply this by a safety factor of 3:

$$\sum E = 3 \cdot \frac{10434 \text{ J}}{0.4 \cdot 0.7} = 112kJ$$

Converting this, our SaRR's maximum energy requirement for one run of the course is about 31 Wh, which is within the range of the battery that has 44 Wh. It is optimal that this max energy is well within the range. We calculated this using large safety factors and some overestimations to ensure that if there were larger-than-normal consumptions of energy - either due to friction, motor efficiency, or internal losses in the motors, gearboxes, or circuitry - our SaRR would still be able to drive through the course at least once. In actuality, through testing we've found that the SaRR's energy consumption from the battery is quite low. Whereas with the maximum energy calculated the SaRR can run the course fully autonomously about two times, physically the SaRR has been able to go through the course fully autonomously at least 10-15 times before experiencing dips in voltage.

6 Engineering Drawings

Engineering drawings of all parts of WALL-E are presented in Appendix A. These are Figures 23, 24, 25, 26, 27, 28, 30 and 31.

7 Test Results

Full test results are presented in Table 5. It should be noted that time duration values in the table were taken from the final, smoothest autonomous test run for greatest accuracy and relevance to the final demonstration.

Course Component	Quantitative Observations	Qualitative Observations
Forward Driving	Max speed:0.762 m/s, 1.705 mph	Robot does not reach this speed during autonomous course operation for accuracy and navigation purposes. Side note: the ESCs used were designed for remote-control cars - the reverse drive speed is roughly half the frequency of forward drive, as RC cars rarely find the need to drive as fast in reverse. Consequently 'max' forward speed shown to the left is around 67% of the theoretical max to match the max reverse speed of the electronic speed controllers.
Ramp Light Signal Acquisition	Time: 4 s	This stage involves a hard-coded 180° turn, that depending on battery charge level, could occasionally undershoot or overshoot this value.
Light Navigation to Ramp	Time: 19 s	Bang-Bang control was used throughout the course. Speed was reduced in early phases of the course to reduce overshoot and error.
Ramp Traversing	Time: 20 s	When the speed was too low through this region of the course, the robot would occasionally fail to scale the peak of the ramp.
Chute Navigation	Time: 16 s	Any gap between the end of the chute and the start of the wall caused the robot to veer off course.
Wall Breaching	Time: 7 s	Safe and reliable wall breaching was impossible before the MedKit mechanism was physically implemented, as the weight of the subsystem moved the CoG forward enough to allow breaching.
Light Navigation to MedKit	Time: 30 s	To address initial problems with differentiation between this phase and light navigation to the ramp, a condition was used in the code to track whether wall breaching had yet occurred
MedKit Deposition	Time: 4 s	System passed on first test and continued to function reliably. A new, stronger basket was printed when the first iteration cracked.
Autonomous Operation	Full Course Time: 1 min 40 s	The most prominent issue in autonomous testing was rebooting of the Teensy due to battery circuit disconnection after wall breaching impact. To address this, a 3D printed battery casing was printed to prevent shifting of the batteries.
Manual Operation	Full Course Time: 30 s	Time to complete course manually depends heavily on skill of driver

Table 5: Full test results of WALL-E.

8 Conclusions and Further Work

8.1 Final Product Performance Assessment

This semester, Team FRYday met all requirements laid out in the MAE 322 Course. Two weeks in advance of the Final SaRR Demonstration, WALL-E had autonomously traversed each section of the entire SaRR course successfully, including the initial light signal acquisition and navigation, ramp ascent and descent, chute proximity navigation, stair and wall breaching, light navigation to the MedKit, and MedKit deposition. A week ahead of the Final Demonstration, WALL-E successfully completed multiple autonomous, contiguous runs of the SaRR course, demonstrating both its reliability and robustness through its repeated success and ability to withstand the extensive testing it underwent. We also accomplished the aforementioned goals using $254.67 less than our allotted budget. Team FRYday believes we owe our success to a consistent dedication to maximal design simplicity to reduce points of failure and devote greater resources to iteration and testing.

8.2 Viability for Deployment

Based on this assessment, Team FRYday endorses the viability of WALL-E as a working prototype for a SaRR model, suitable for industry-level manufacturing and deployment. The tread-based drivetrain with a front lip makes WALL-E unique in its tight turning radius, robustness to impact, low center of gravity, and ability to surmount obstacles quickly and efficiently. The simple rotational motored deposition mechanism ensures both the security of the cargo during transport and its controlled, precise delivery. At this point, Team FRYday has completed all planned design, analysis, simulation, manufacturing and testing work for the prototype produced in the MAE 322 course. Thus, our directions for future work focus on the next steps required for large-scale deployment.

8.3 Next Steps

To this end, future work on WALL-E would build upon and extend the current robustness and reliability of the MAE 322 product, both in terms of hardware and software. Current ideas for increased hardware robustness include implementing higher-quality sensors for increased reliability and precision, adding spring-loaded shock absorbers for the entire electronics plate, building shock-protective and waterproof housing for all electronic components, and adding tensioners to the treads to prevent gradual loosening of the fit over time with material wear. On the software side, robustness and reliability could be improved through adding and programming in encoders and accelerometers for better understanding of the robot's kinematics, and implementing a battery voltage monitor to compensate for level of battery potential by adjusting torque with PWM output. Additional ideas include replacing hard-coded course elements (i.e. 180° turn for initial signal acquisition) with more autonomous solutions (i.e. turn until light intensity reaches a certain threshold), and substantiating more extensive edge case navigation and failure recovery protocols in the case of a major disruption to the robot's movement or navigation.

8.4 Acknowledgements

A   Appendix
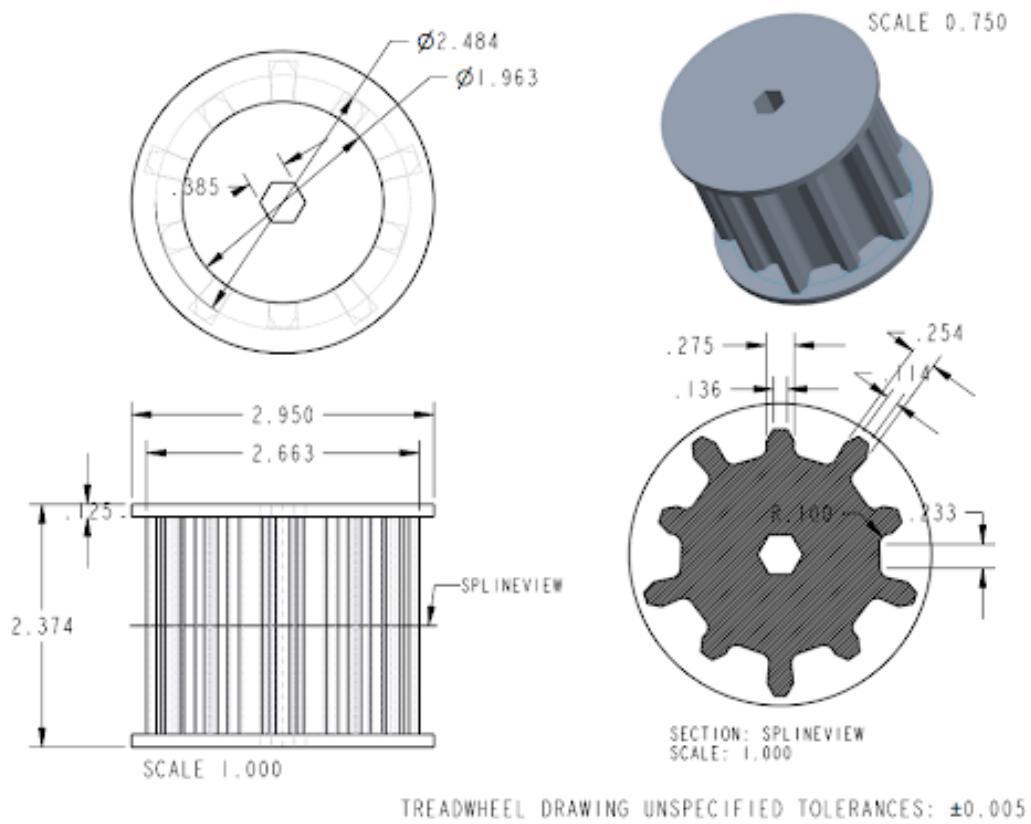
Figure 23: Engineering drawing of the back wheel.
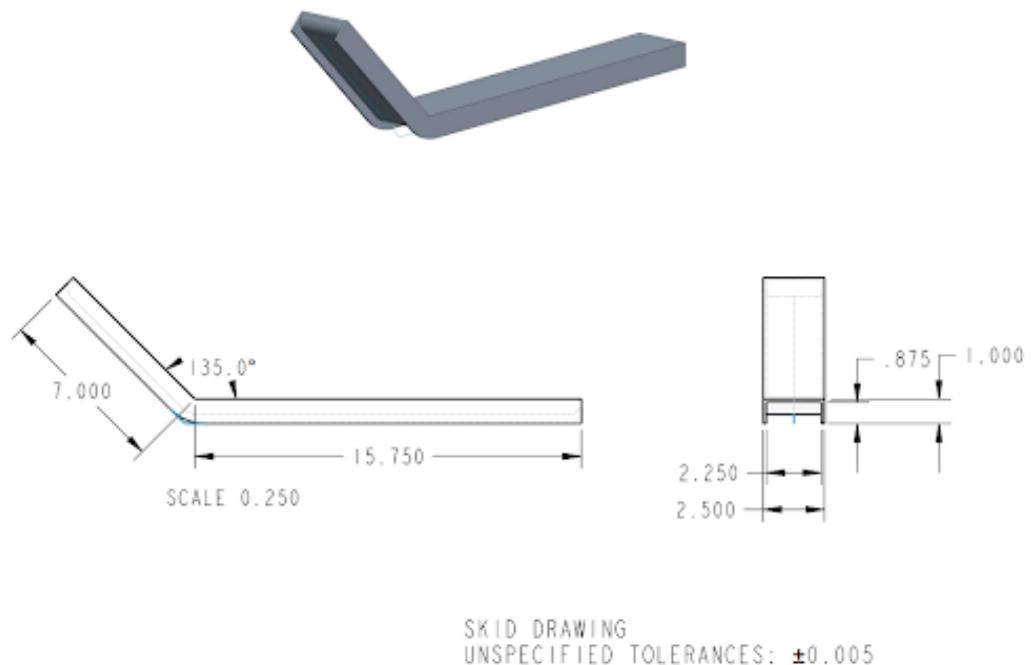

Figure 24: Engineering drawing of the skid.

SECTION: A
SCALE: 0.750

1.300 1.144 .885

CROSS SECTION B   .711   CROSS SECTION A

R1.156

2.301   1.704 2.311   1.704 3.750

SECTION: B
SCALE: 0.500

Ø.250X2   R.200   Ø.159X2

SCALE 0.500

.300

MOTOR MOUNT DRAWING
UNSPECIFIED TOLERANCES: ±0.005   1.425

5.000
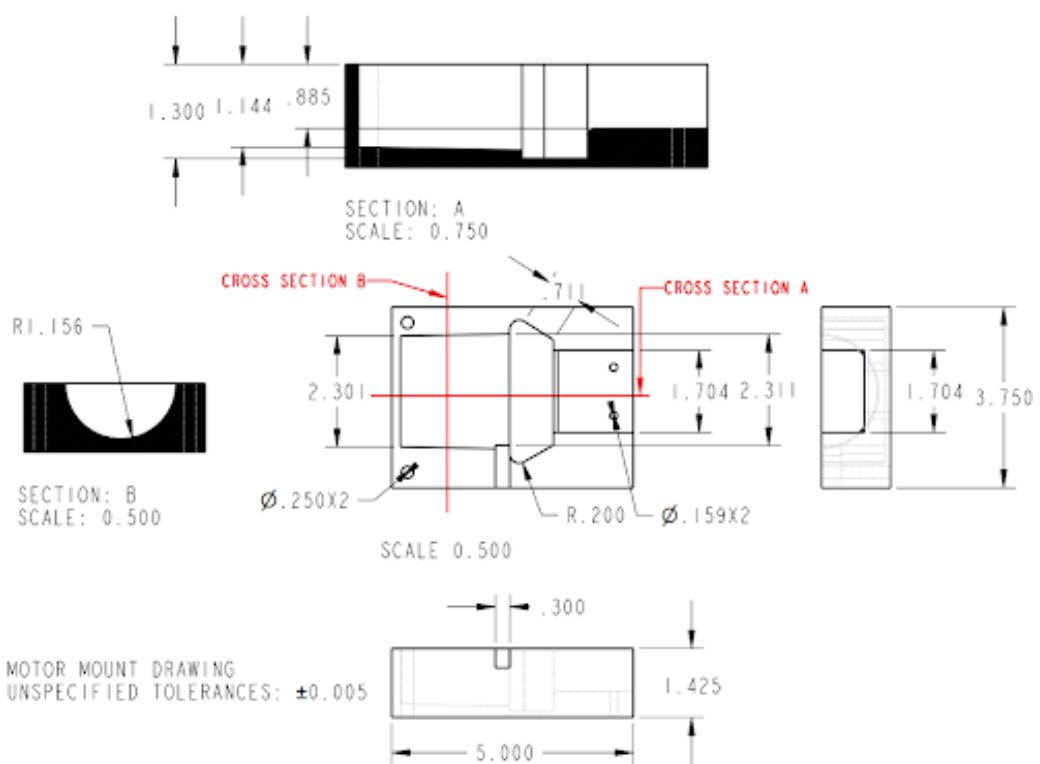
Figure 25: Engineering drawing of the main motor mount.

SCALE 0.750

Ø.375

2.300   3.000

SCALE 0.750

FRONT WHEEL DRAWING
UNSPECIFIED TOLERANCES: ±0.005

2.000

2.500
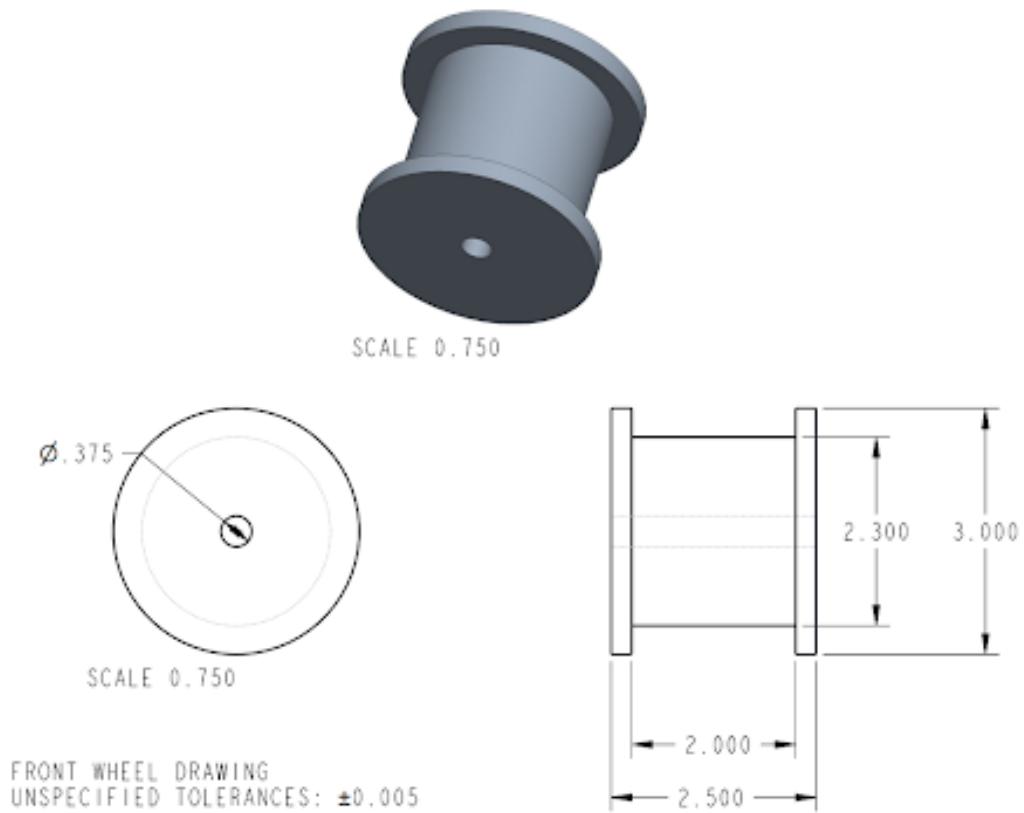
Figure 26: Engineering drawing of the front wheel.

SCALE 1.000

GUSSET DRAWING
UNSPECIFIED TOLERANCE: ±0.005

SCALE 1.000
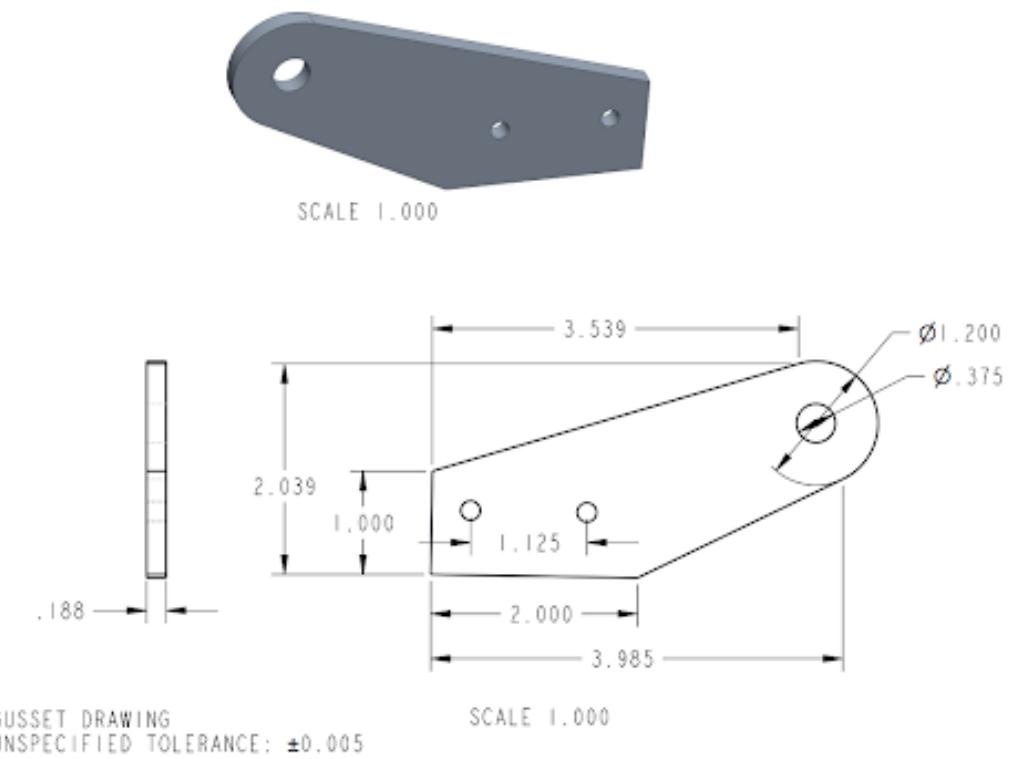
Figure 27: Engineering drawing of the gusset.

SCALE 0.750

SECTION: A
SCALE: 0.750

MEDKIT MOTOR MOUNT DRAWING
UNSPECIFIED TOLERANCE: ±0.005

CROSS SECTION A
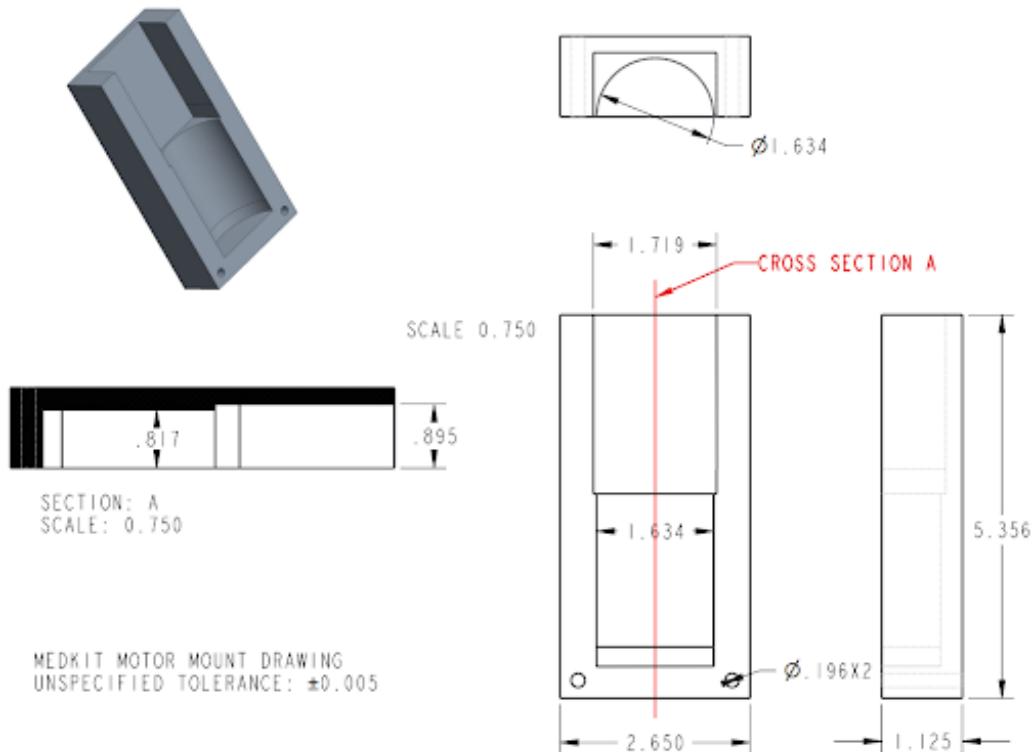
Figure 28: Engineering drawing of the Medkit motor mount.
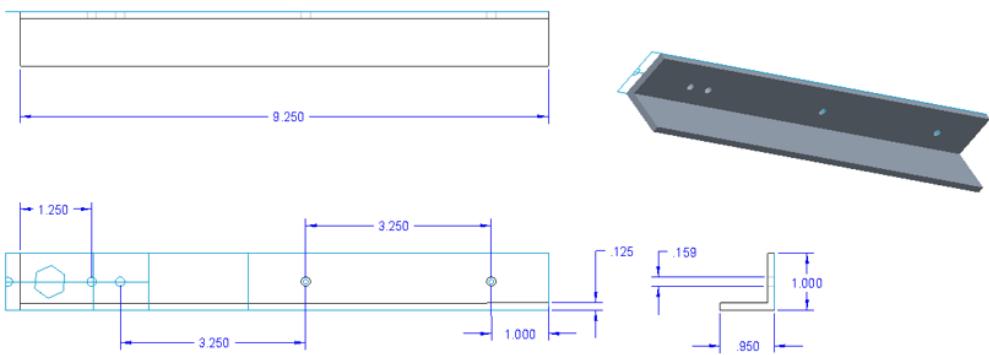
Figure 29: Engineering drawing of the Medkit arm.
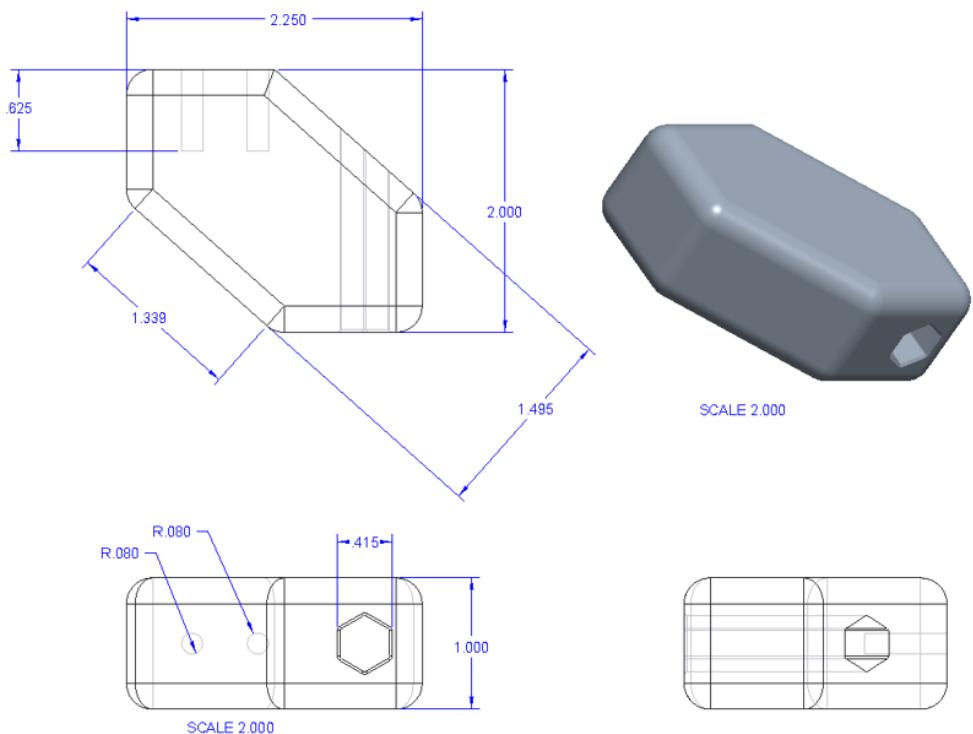
Figure 30: Engineering drawing of the Medkit arm coupler.
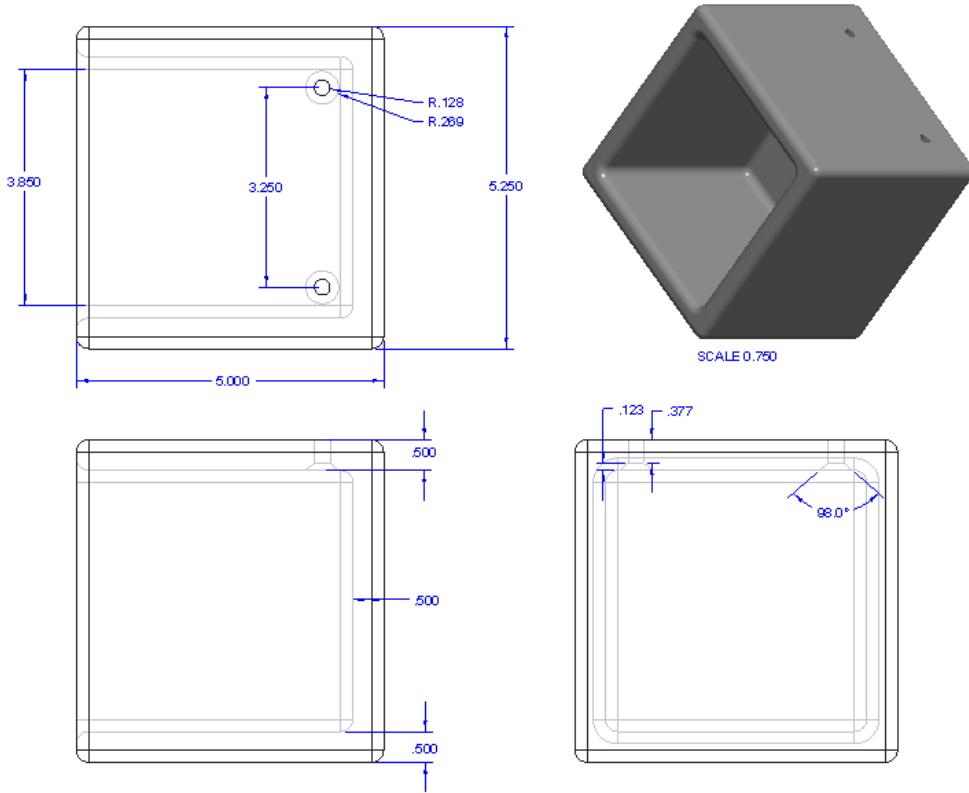
Figure 31: Engineering drawing of the Medkit basket.
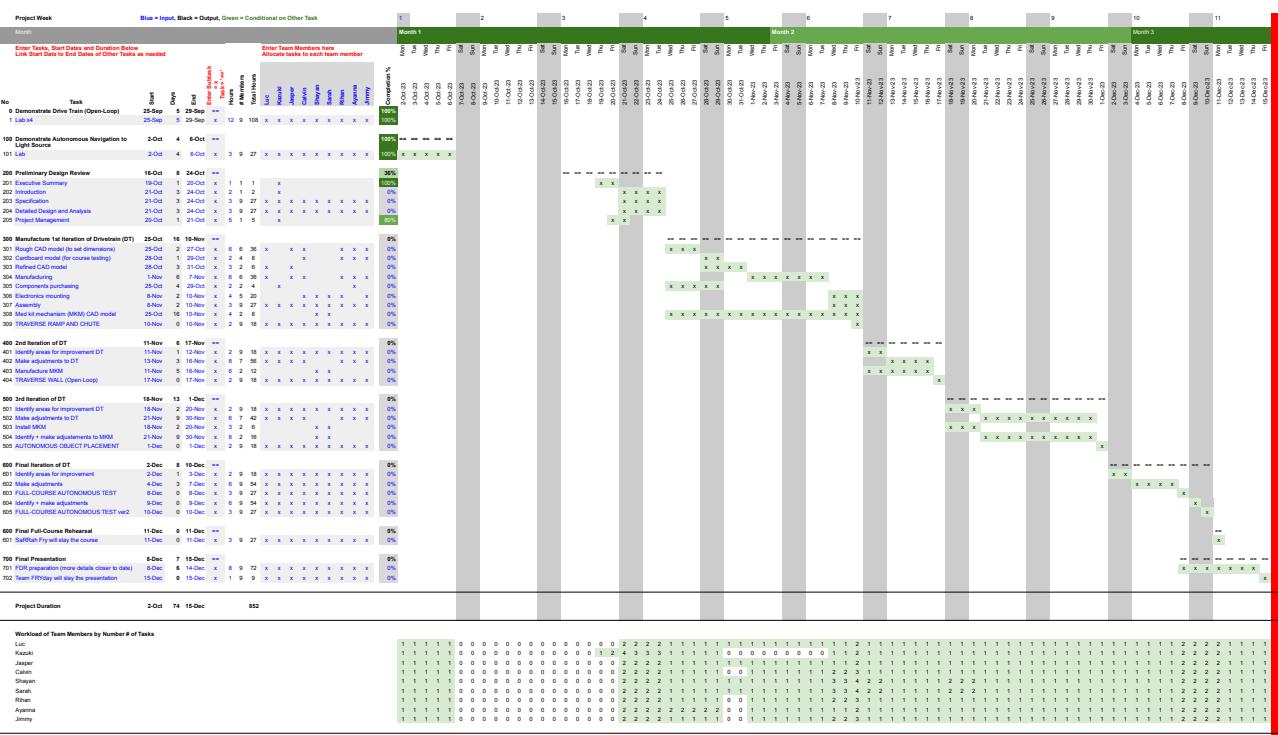
Figure 32: Full Gantt chart used to organise the project.

References

[1] A. Wolf, H. H. Choset, B. H. Brown, and R. W. Casciola, "Design and control of a mobile hyper-redundant urban search and rescue robot," *Advanced Robotics*, vol. 19, no. 3, pp. 221–248, Jan. 2005, Publisher: Taylor & Francis _eprint: https://doi.org/10.1163/1568553053583652, ISSN: 0169-1864. DOI: `10.1163/1568553053583652`. [Online]. Available: `https://doi.org/10.1163/1568553053583652` (visited on 10/27/2023).

[2] T. Kamegawa, T. Akiyama, S. Sakai, *et al.*, "Development of a separable search-and-rescue robot composed of a mobile robot and a snake robot," *Advanced Robotics*, vol. 34, no. 2, pp. 132–139, Jan. 2020, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01691864.2019.1691941, ISSN: 0169-1864. DOI: `10.1080/01691864.2019.1691941`. [Online]. Available: `https://doi.org/10.1080/01691864.2019.1691941` (visited on 10/27/2023).

[3] A. Denker and M. C. İşeri, "Design and implementation of a semi-autonomous mobile search and rescue robot: SALVOR," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Sep. 2017, pp. 1–6. DOI: `10.1109/IDAP.2017.8090184`. [Online]. Available: `https://ieeexplore.ieee.org/abstract/document/8090184` (visited on 10/27/2023).