CRC Kinetic Subteam Report

Kinetic Subteam

Fall 2021

Contents

1 Introduction

This report was written by the Kinetic subteam of Fall 2021, with the purpose of documenting the design process of a combat robot and our experiences such that future CRC members have something to refer to as well as a way to learn about past mistakes.

Within the Fall 2021 semester, we aimed to design a high-kinetic-energy 12-lb combat robot, seeking to build it for competition in the following semester. Although our focus was designing a good robot for winning the competition, we also prioritized learning and having fun along the way. The competition we plan to attend is the Norwalk Havoc Robot League (NHRL) competition happening on March 26, 2022.

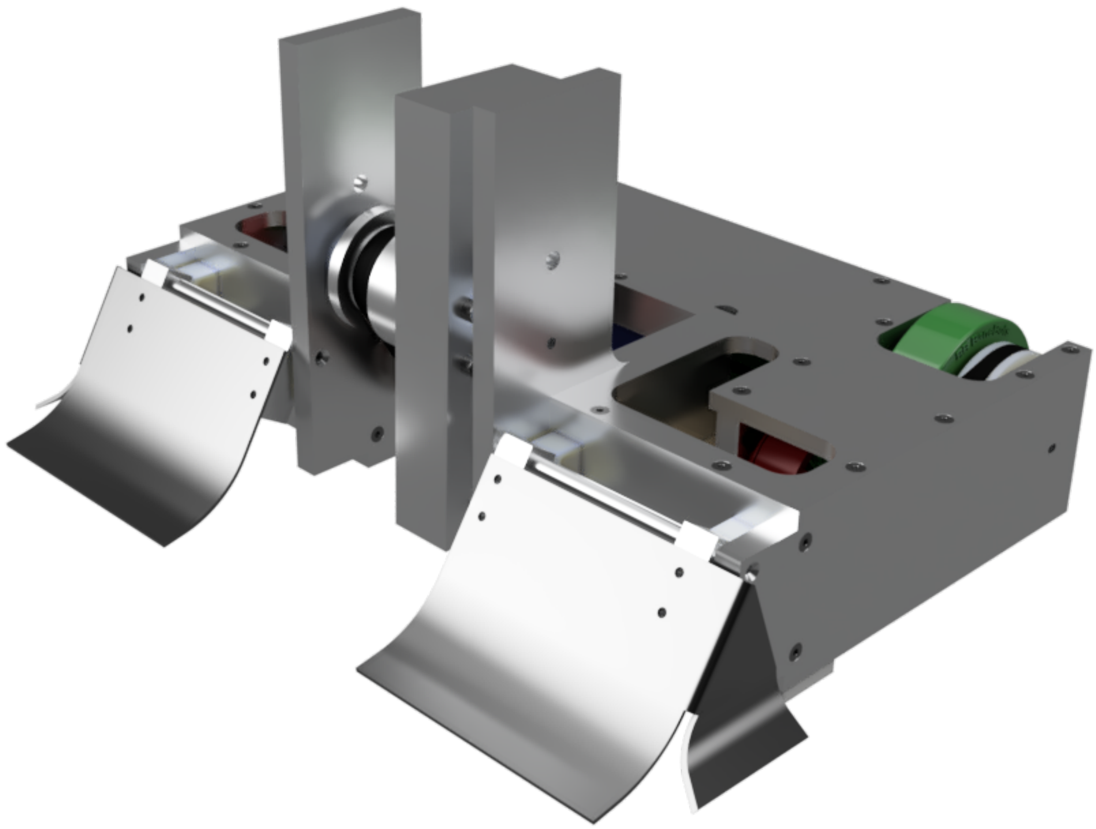

Figure 1: Final CAD of our combat robot Richard

2 Method

Since our goal is to do well at competition, our design process emphasized simplicity and robustness over creativity. We chose to design for manufacturing on manual mills and lathes because we wanted to get more hands-on experience and save money. Also, making a robot with a weapon concept that has worked in the past would improve our chances in competition. However, we strongly avoided copying previous designs. Instead, we performed extensive analysis and research to understand, improve, and perfect our design.

Our design constraints include a weight limit of 12 lb and a volume limit of 30"x30"x24". The competition rules also limit the maximum voltage for the robot's electrical system to 72 Volts.

2.1 Subsystem Assignments

To help allocate work in a more organized and effective manner, we assigned roles based on robot subsystems: Powertrain, Chassis, and Weapon. The goal here was to give everyone a clear focus and to promote specialization: if any of us were looking for tasks, our subsystem would be the first place to find them, and we would build deeper knowledge in smaller areas. However, in the absence of work in their own subsystem, anyone could help anyone. This inclusivity helped keep the team cohesive and moving forward. Subsystem assignments and responsibilities are outlined below:

Weapon: Isaac Newcomb, Sofie Halpern

- Design the weapon of the robot, considering feasibility, stress limits, and manufacturing capabilities as well as competition flexibility ie. repairability/replaceability and durability or lifespan
- Relies on the chassis for mounting, and relies on the electronics/drivetrain for proper functioning
- Collaborate with Powertrain to select the weapon motor and Electronic Speed Controller (ESC)

Chassis: Erhunmwunse Eghafona, Ricky Wang, Anna Boese

- Design the Chassis while considering manufacturing methods and joining elements
- Integrate chassis with the weapon system and allow room for electronics, keeping in mind the drivetrain type

Powertrain: Mohammad Ali Moghaddasi, James Courtenay, Spencer Hurst

- Select compatible electronic components appropriate for powering and controlling the robot's drivetrain and weapon
- Design the drivetrain: Deciding on 2-/3-/4-wheel drive, indirect or direct drive, fast or punchy style, invertibility, acceleration, and time to travel across the arena.
- Design and/or select mounts for the electronics (e.g. motor mount).
- Collaborate with Weapon to select the weapon motor and ESC

With these subsystems, we were able to efficiently complete the design of our combat robot.

2.2 Logistics

To make progress on our robot design, the Kinetic subteam held weekly 3-hour subteam meetings, organized and led by Ricky Wang to ensure that we had a clear game plan for each session and that we did not get off topic for too long. Once we split up into our subsystems, we scheduled subsystem work sessions as needed. We decided that it was easier to plan the subsystem work sessions by the week rather than having a recurring time, as the workload for each subsystem differed week by week. Our work was structured around our hard deadlines, some of which were set by Cornell's academic calendar, while others were set by the leads on the team to ensure that the subteams presented their progress to the team every few weeks.

We set up internal deadlines and a schedule for our subteam to follow each week. As we progressed through the fall semester, we deviated from our original schedule and smaller deadlines, since this was the team's first time having a full and legitimate design process. We kept on track for the early stages but fell apart later in the process, so we should use this year's actual timeline (of how things panned out) as a model for setting up our timeline in future years.

Hard Deadlines

New members onboarding	October 25th
Preliminary Design Review	Saturday November 13th
Second Design Review	Saturday December 4th
Study Period and Exams	Wed December 8 to Sat December 18
Last day of fall semester	Saturday December 18

Internal Deadlines

Initial parts list and initial computer-aided design (CAD)	Friday November 12th
Complete Simulations and Optimization	Saturday November 20th
Finalized Parts List and CAD	Saturday November 20th
Parts Drawings and Manufacturing Plan	Friday December 3rd

Schedule

Week of October 3	Select general design
Week of October 10	[Fall break] Weapon design research
Week of October 17	Drivetrain+Electronics research, assign subsystems
Week of October 24	Chassis research and initial equations
Week of October 31	New member onboarding, initial parts list and initial sketch(es)
Week of November 7	Initial parts list and initial CAD, start simulations
Week of November 14	Finalize parts list, CAD+simulations
Week of November 21	Start parts drawings and manufacturing plan
Week of November 28	[Thanksgiving break] Wrap up parts drawings and manufacturing plan

2.3 Choosing a weapon type

To start off our design, we made a Pugh Decision Matrix to choose between a few types of weapon that we were interested in (Fig. 2).

Pugh Matrix - Kinetic

Combat Robotics @Cornell

Problem/Situation: What type of combat robot should the Kinetic subteam make?

		Alternatives							
		1	2	3	4	5	6		
Criteria	Weight	Full-body spinner	Pneumatics Fork	One pneumatic spike in center	Pneumatics pusher	Vertical spinner	Horizontal spinner	Totals	Rank
Manufacturability	5	-1	-1	0	-1	1	1		
Affordability	5	0	-1	-1	-1	1	1		
Maneuverability	2	0	1	1	1	-1	-1		
Repairability (between matches)	3	-1	0	1	0	1	0		
Armor Toughness	2	1	0	0	0	0	-1		
Reversability / Self-right	3	0	1	1	1	-1	0		
Weapon Effectiveness	4	0	-1	-1	-1	1	1		
	Totals	-6	-9	-1	-9	12	10		
	Rank	4	5	3	5	1	2		

Figure 2: The Pugh Decision Matrix comparing robot types

A full body spinner would have been too hard to manufacture and repair; any pneumatic design would have been too expensive and minimally effective; and a horizontal spinner was a close runner-up but seemed more vulnerable, requiring a large open area and not allowing for significant wedge-based protection. By elimination, we decided to design a vertical spinner.

2.4 Deciding on our general design

We wanted to make sure everyone on the subteam was on the same page when it came to the robot design, both its features of the robot and the design reasons for them. To allow all subteam members to gain a deep level of understanding of our general robot design, we utilized a research-heavy approach in our early design phase. In the first week of the subsystem research, we focused on the weapon design. The next week, we discussed the powertrain design, and finally, we worked on the chassis.

In preparation for each design session, the subteam completed 1-3 hours of independent research using our combat robotics guides ([1] and How to Build a Combat Robot), popular forums (Ask Aaron and even some Combat Robotics subreddits), Youtube videos by established hobbyists (including Just 'Cuz Robotics), and articles that we found when Googling some key words. In order to have less overlap with the findings each member brought to the work session, we made an effort to consult different resources.

For each work session, we started by giving each member a turn to share their main findings about the subsystem of discussion. This naturally transitioned into comparing the information from the different resources and coming to a consensus of how we should design each main component of the subsystem. We placed a large focus on making whiteboard sketches to elaborate on our ideas and consider conflicting ideas. Once we had sets of ideas to choose from, we began to add in the physical considerations involved in the subsystem. Some of these considerations include center of mass, calculating an optimal length of a component, and dependencies that some components have with others. We left each of these three work sessions with an in depth sketch or list of ideas that describe the subsystem, for the details and CAD to be made once we split up into subsystems.

2.5 Subsystem work

After splitting into Powertrain, Chassis, and Weapon, each subsystem added 1-2 work sessions during the week to make progress before our weekend subteam meetings. This enabled us to utilize subteam work sessions for ironing out issues, getting one another's insight, and subsystem integration.

3 Subsystems

3.1 Powertrain

The Powertrain subsystem consisted of the electronics for powering-up the robot, and drivetrain for maneuverability and accelerating of the robot. We decided on a 2-wheel invertible drivetrain with the wheels on the back of the motor, skidding on two static teflon square pieces at the front of the robot. Our goal was to be fast in speed rather than "punchy & pushy" since vertical spinners operate better under a higher momentum impact boosted by a high speed.

3.1.1 Pre-PDR

Since the goal within the first couple months was to design a powertrain system for a 3-lb robot, in the preliminary selection of parts, almost every component was very specific to a 3-lb robot. The corresponding parts list and cost analysis were compiled in a Google Sheet document [2]. An important highlight of the original 3-lb design and parts list that was different than that of the 12-lb's was the direct drive 22MM DC GEARMOTOR WITH ALUMINUM GEARBOX from Botkits. According to our research, this motor seemed to be promising within the beetle-weight (3-lb) class. After the team raised the funds to switch to 12-lb robots, all the electronics needed to be re-specced.

3.1.2 Post-PDR

Since the powertrain subsystem is very weight-dependent, for the conversion from 3-lb to 12-lb, almost all components needed to be re-selected and a new drivetrain needed to be designed. Here, the different sections are explained based on category.

Drivetrain Power Transmission & Gear/Pulley Ratios

The Drivetrain was determined to have a gear ratio of 3:1 through two pulleys to increase the torque and decrease the speed of the motor, $\sim$6000 revolutions per minute (RPM) transmitted to the wheels, $\sim$2000 RPM, to allow for better initial startup, control, and traction. A 0.7" diameter 3D printed Onyx pulley transmits the power from the motor to the 2.1" diameter pulley of the same material that actuates the wheel.

Design of the Pulleys

The mounting of the pulleys (shown as white in Figure 3) to the wheel was a simple matter of a direct bolt through the pulley, through the wheel, nutted on the other side of the wheel. While the team decided to go with a 3D printed pulley since a Nylon/Onyx material pulley would be able to withstand , the belt was decided ahead of time so the pulleys could be made to fit the belt effectively. Knowing the approximate 3:1 gear ratio needed to decrease the motor speed to a reasonable value, the diameters of the pulleys were chosen to fit the geometry of the chassis and still allow for invertibility of the robot. Additionally, the distance between the pulleys was designed to be adjustable with slots in the mounts of the drive shaft through the wheel's hub. The material chosen for this 3D printing was Onyx because it is relatively strong and of high quality according to Rapid Prototyping Laboratory of Cornell, where we placed our no-charge (due to being a project team) order of our 3D-printed parts. As we had the width of the wheel and the width of the belt provided from research and the width of the pulley dependent, but now also ascertained from the size of the belt, we could decide on the screws needed for the wheels to allow for a proper mounting of the wheel to the pulley. The screws decided upon were 40mm long and Medium-Strength Steel-Nylon insert Lock-nuts.

The Motor used 16 mm long black oxide steel socket head screws to ensure that the motor's own pulley was secured to the motor and spin with it at the motors respective RPM.

Design of the Wheel System

The Wheel Spacers are similarly 3D-printed (shown as yellow in Figure 3) and are used to ensure that the connected wheel/pulley apparatus doesn't move along the axle of the shaft (axially). The Shaft is statically mounted between the wheel mount and side plates, through a wheel-hub and a pulley-ball-bearing allowing the motor to directly rotate the wheel itself through the pulley/belt instead of rotating the axle. This was decided upon because the shaft therefore becomes an extra level of support on the sides and because it prevents another potential breakpoint by not needing several extra bearings which could be easily broken by a side swipe.

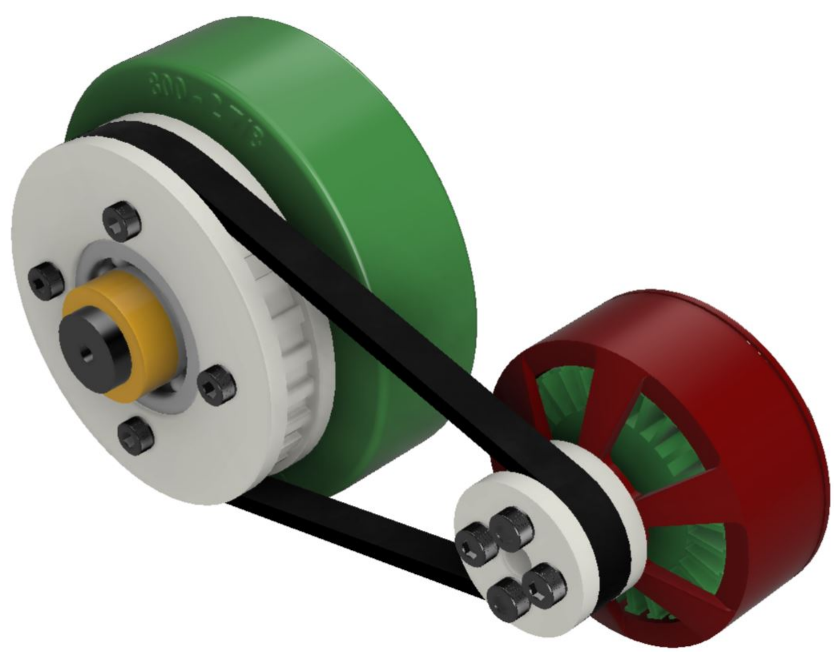

Figure 3: Drivetrain System of two teeth-pulleys driven by a timing belt transmitting the power from the motor to the wheel. Green is the wheel, red is the brushless motor, the two white components are the pulleys, bolts through the pulleys are in gray, 3D printed spacers in orange, and the dark gray aluminum metal shaft through the wheel and pulley and spacer.

Mounting of the Motor and Wheel

Different mounts were made for the motor and the wheel (Fig. 4). Aluminum was used as the material for these parts to ensure a solid support that can stay rigid under high-impacts from robots/obstacles.

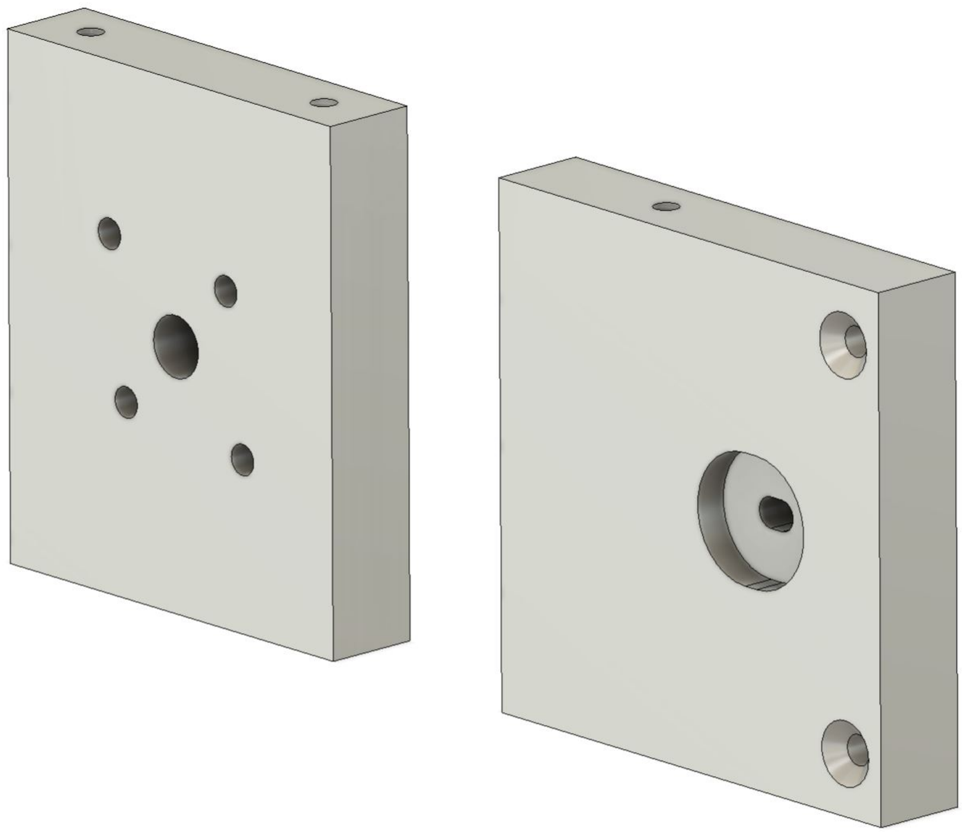

Figure 4: The Motor mount CAD can be seen on the left. The Wheel mount CAD is on the right.

Choosing motors

Our search for motors began with brushed, since it is a much simpler direct drivetrain system due to an addition of a planetary gearbox on the brushed motor, which increased torque and the handling of the stress of the motor shaft by converting to a larger-size drive shaft. However, for a 12-lb weight class robot, these robots can be quite expensive. Hence, we decided to switch gears and look more into brushless motors, which are cheaper and lighter. The only complication that this brings into the system is the need for an indirect drivetrain system, which was satisfied by pulleys as explained in the 3.1.2 Section. After deciding between brushed/brushless, the most important parameters to pay attention to are the operating voltage of the motor, current rating, stall current/torque, and speed in RPM.

Using online motor calculators and calculators we implemented ourselves, after an extensive search of brushed/brushless motors, we selected BangGood's Racerstar Racing Edition 4114 BR4114 380-400KV 4-8S Brushless Motor For 600 650 700 800 RC Drone FPV Racing. This is also the motor that

Draconid, a 12-lb robot at NHRL used for their drive system. The calculation proved that this motor would be able to carry our robot's weight, and accelerate enough to travel across the arena side-to-side within a desirable low amount of time (∼2 seconds) with an average speed of 5624 RPM, equivalent of ∼15 mph using our 2.75" diameter wheels [3]. Figure 5 was taken from Just 'Cuz Robotics implemented calculator [4]. Another online calculator we used was Team Tentacles [5]. We also implemented Team Tentacle's source code in MATLAB, which can be seen in Section 6.2.

Robotics Drivetrain Calculations	Input	Calc	TO USE THIS SHEET: FILE > MAKE A COPY, DO NOT REQUEST EDIT ACCESS					Per Motor		Team Tentacle Calculator
			Assumed 2WD is 25% weight			20% Added		Imperial	Metric	
Wheel/Bot Info	Wheel Diam (in)	Co. Fr	Robot Eff. Weight (lb)	Num Wheels	Num Drive Motors Total	Weight Per Wheel (lb)	Max Tire Force (lbf)	Min Torque To Spin (ozf-in)	Min Torque To Spin (Nm)	
	2.875	0.9	12	2	2	3.600	3.240	74.52	0.526	

				Metric	Imperial				Good	Theoretical Speed @ Chosen Ratio		Good	Team Tentacle Values	
Brushless Motor Info	Rated Voltage	Stall Current	kV (RPM/Volt)	Est. Stall Torque (Nm)	Est. Stall Torque (ozf-in)	Bot Voltage	Max RPM	1.5X Ratio	Chosen Ratio	(ft/s)	(m/s)	(mph)	RPM/Volt	oz-in/amp
	14.8	23	400	0.477	67.579	14.8	5920	1.654	3.000	24.755	7.545	16.878	400.0	2.938

				Metric	Imperial				Good	Theoretical Speed @ Chosen Ratio		Good	Team Tentacle Values	
Brushed Motor Info	Rated Voltage	Stall Current	Free RPM @ Rated V	Stall Torque (Nm)	Stall Torque (ozf-in)	Bot Voltage	Max RPM	1.5X Ratio	Chosen Ratio	(ft/s)	(m/s)	(mph)	RPM/Volt	oz-in/amp
	18.0	130	19600	0.466	66.000	18.0	19600	1.694	2.600	94.567	28.824	64.476	1088.9	0.508

Unit Conversion Calculator			
From		To	
1	ozf-in	0.00706	Nm
1	Nm	141.600	ozf-in
1	ft-lbf	0.73760	Nm
1	ft-lbf	192.000	ozf-in
1	in	25.400	mm
1	mm	0.03937	in
1	lb	0.45360	kg
1	kg	2.205	lb
1	kg-cm	13.890	ozf-in

Lipo Voltages		
Cell Count	Nominal	Full Charge
1	3.7	4.2
2	7.4	8.4
3	11.1	12.6
4	14.8	16.8
5	18.5	21.0
6	22.2	25.2
8	29.6	33.6
10	37.0	42.0
12	44.4	50.4

Figure 5: Brushless Motor Calculations (N/A Brushed calculations as our motor is brushless)

Another calculation done to ensure the stall torque of the motor is sufficient to startup our robot was implemented using Newton's second law of motion and equations of kinematics as

$$\tau_{stall} = FoS\mu N\frac{r}{R}, \tag{1}$$

where τ_{stall} is the stall torque of the motor, FoS is the factor of safety $1.5 - 2.0$, μ is the coefficient of friction ~ 0.9, N is the normal force supported by each wheel, in this case $N = \frac{mg}{2}$ with mg being the weight of the robot, r is the radius of the wheel, and R is the gear ratio.

Electronics

The electronics included of 3 ESCs (2 for the drive motors, 1 for the weapon motor), power-switch (turns the robot on and off), 4S LiPo (4 cells, equivalent of 14.8 volts) 4000 mAh 30 C battery, 16 American Wire Gauge (AWG) wires, transmitter and receiver (i.e. Radio Control system), and the three motors (2 for drive, 1 for weapon). A parts list of all the powertrain parts were crafted including links to vendors' websites [6]. A wiring diagram of the entire electronic system is shown in Figure 6. Note that the ESCs include battery elimination circuits (BECs) which decrease the voltage to 5V to be able to safely power up the receiver without blowing it up. ESCs connect to the receiver by three wires power, signal, and ground. It is important to know that only one power (5V) wire from the 3 ESCs needs to power up the receiver. Therefore, two of the ESC power lines will be trimmed to comply with that.

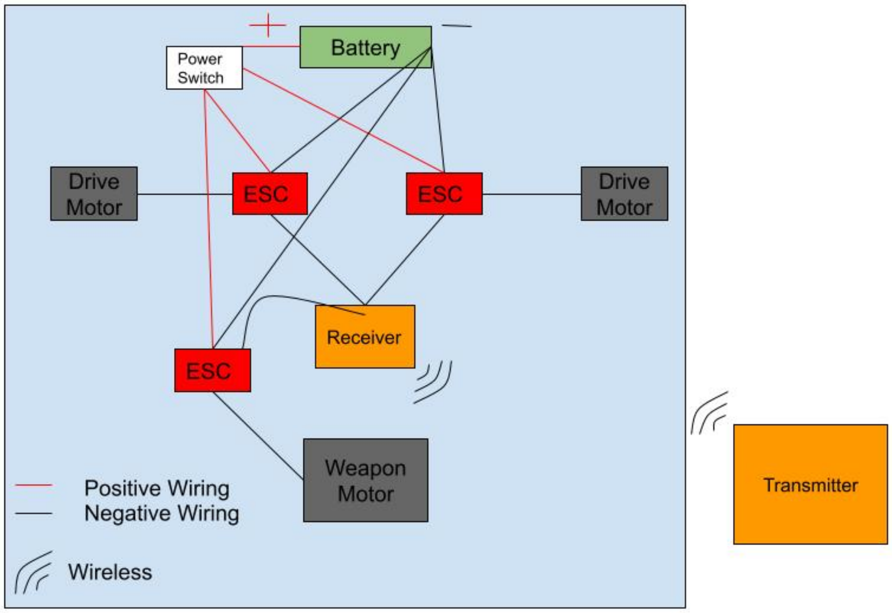

Figure 6: Schematic of the electronics wiring. The positive sign represents the positive lead of the battery, and the negative represents the negative lead of the battery

Battery Discharge Calculations

When selecting the electronics, all parts have to be capable of handling the voltage across them and the drawn current from the battery. In our case, the supplied voltage from the battery was 14.8 volts, and the maximum discharge rate of the battery was calculated as 120 amps. This was determined using a multiplicaiton of the C-rawting and the capacity in mAh.

$$4000mAh \times \frac{1A}{1000mA} \times 30C = 120A. \tag{2}$$

This means that combining all of our electronics (ESC, motor, etc.) we can only draw up to 120 Amps of current. This was sufficient since the continuous current draw of the ESC was 60 A, while the the motor load current was 25.5 A, lower than that of the ESC. Hence, a maximum of 120 A will be drawn by the 2 ESCs, which is actually highly unlikely to happen, considering motors will pull a maximum of 51 A.

Battery Run Time in the Competition

To calculate how long the battery will last, the 4000 milliampere-hours was divided by the average/continuous current drawn from the drivetrain and the weapon subsytems. It was asuumed that 80% of the drive's 60 Amps (which is 48A assuming motors are running at full 51Amps the entire time, with ESCs using a total of 9Amps), plus having the weapon motors running at 50% of 55 Amps (which is 27.5A using half of the max current of the motor), then

$$4000mAh \times \frac{1A}{1000mA} \times \frac{60minutes}{1hour} \times \frac{1}{48A + 27.5A} = 3.2minutes, \quad (3)$$

which is 12 seconds over the competition constraint of 3 minutes.

3.1.3 Final Status

After the Critical Design Review, the only changes made to the Powertrain subsystem were due to the unavailability of the selected items on the manufacturer's website. ESCs and motors had to be reselected for the drivetrain, however, we were able to find similiar parts to what was previously picked. The new ESC is a 60A rather than a 40, since it was the only one we were able to find that would be compatible to the Banggood motors. The new motor also had a slightly lower kv ($RPM/suppliedvoltage$ of 340-380 kv rather than the original 400 kv. This only slightly decreases the drivetrain speed, but it offers an increased torque. Therefore we are happy with the choices we made.

3.2 Chassis

3.2.1 Wedge

Launching an opponent into the air is more valuable than sliding an opponent across the arena. In order to launch an opponent upwards, the weapon needs to hit from below — and in order to hit from below, we need the opponent to be lifted off the ground a bit first. A wedge accomplishes this, and it has some defensive advantages, too.

Horizontal spinners can be easily destabilized if their weapon gets tilted off-axis. A wedge exploits this vulnerability by gently lifting the opponent's blade, taking a glance below.

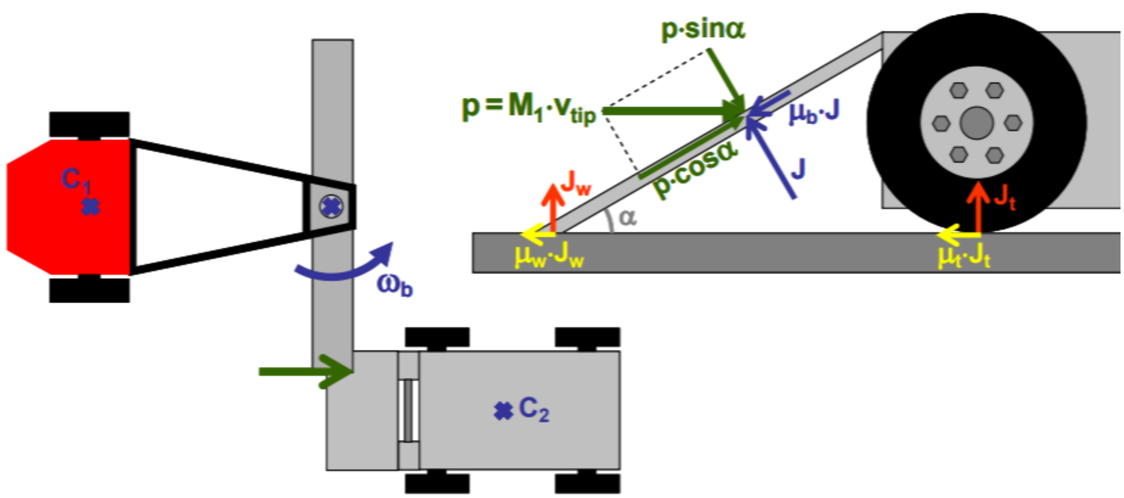

Figure 7: Wedge vs Horizontal Spinner. 'p' is the momentum of the horizontal spinner. 'J' is the reaction impulse. Subscripts 'w' and 't' represent the wheel and tire respectively.

According to the figure above, by minimizing the friction coefficient, horizontal impulse is converted into vertical impulse and there's an smaller impact on the robot. A steel wedge was utilized due to it's resistance to dents which serves to minimize friction throughout the battle. Additionally, a low horizontal spinner can cause significant damage to a standard wedge. This can be countered by having a low-angle wedge, however, a smaller wedge-angle correlates to a weaker wedge. We found that having a curved wedge with side guards provides adequate defense to low spinners while not sacrificing strength.

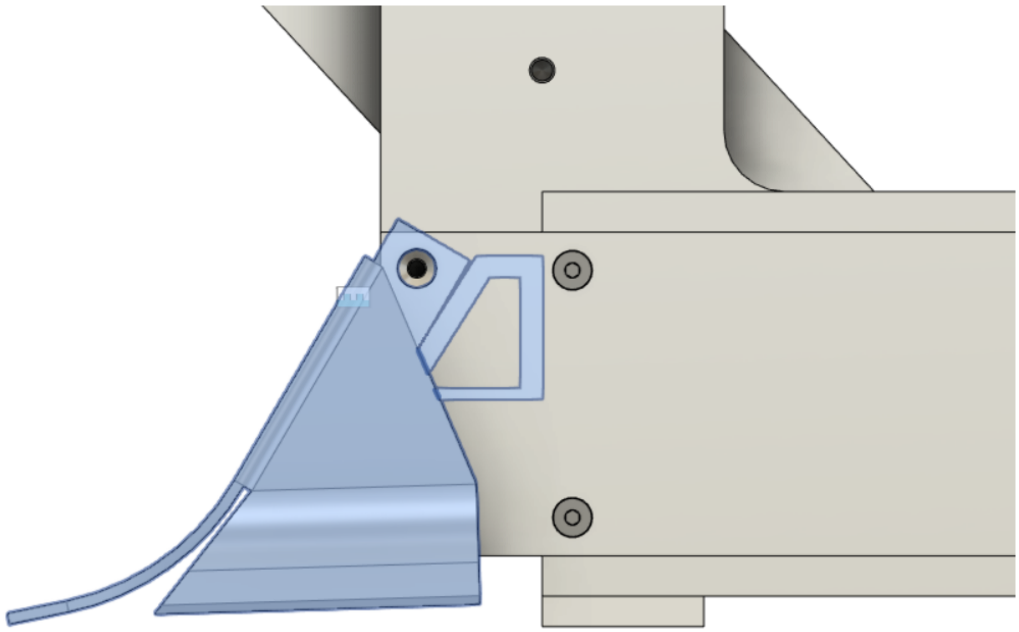

Figure 8: Angle Limiter and Wedge. While the wedge design choices were finalized 'Pre-PDR', the final CAD Design wasn't made until 'Final Status'. Notice the trapezoidal piece behind the wedge acts as an angle limiter that prevents the wedge from rotating backwards.

3.2.2 Chassis Material

There are 3 types of armor typically found among combat robots, namely traditional, ablative, and reactive. We decided to go with a traditional type of armor (hard material) since it is simple and works well. Aluminum 6061 was chosen as the material for the chassis frame's material as well as any additional armor (such as wheel guards) since it has good impact and yield strength while being easily machinable and not too heavy (as compared to steel).

3.2.3 Chassis Width

Vertical spinners are known to be very strong and destructive in the combat robotics field, but it is also commonly known that they are very unstable. Since vertical spinners carry a very fast spinning object, it builds up angular momentum in an unfavorable direction. When the vertical spinner robot tries to turn, gyroscopic effects cause it to tip onto one side, negatively impacting maneuverability. Upon research, we found that there was an equation (Eq. 4) to calculate the critical turning speed and, in turn, adjust the robot design such that the critical turning speed was a good value for a quick and maneuverable robot [1]. The critical turning speed is essentially the maximum allowable turning speed for the robot to stay completely on the ground.

$$\omega_{y,critical} = \frac{mgd}{I_{zz}\omega_z} \tag{4}$$

Where $\omega_{y,critical}$ is the critical turning speed in rad/s, m is the mass of the robot in kg, g is gravitational acceleration in m/s^2, d is half of the width of the robot in m, I_{zz} is the moment of inertia of the spinning weapon in kgm^2, and ω_z is the rotational speed of the spinning weapon in rad/s.

Based on Equation 4, it becomes clear that assuming the same weapon inertia and speed as well as the same weight class, the only chassis-related parameter that can be changed to improve the critical turning speed is the width of the robot, which corresponds to d (half-width). Thus, we opted to make a wider than longer robot to prevent negative gyroscopic effects.

The values of the weapon moment of inertia and the weapon spinning speed were provided as $0.001142kgm^2$ and $1311.09rad/s$, respectively. Then, based on Equation 4 and assuming a desired critical turning speed of at least 240 degrees in one second, the value of d comes out to be 4.75 inches, meaning the distance between the two wheels should be 9.5 inches.

3.2.4 Screws

Something that was noted from our resource findings was that the orientation we put our screws in matters. This is because screws shear much more easily than they break due to tension forces (Fig. 9).

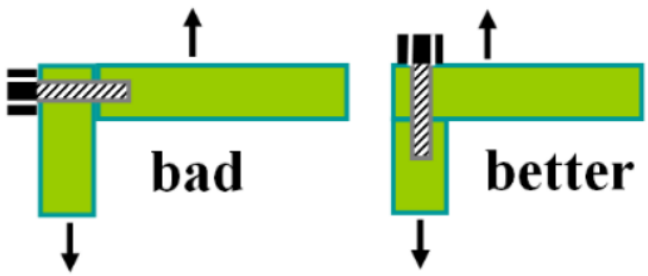

Figure 9: Demonstration of how putting in screws such that they are more susceptible to shear stress is undesirable. [1]

Based on this information, the frame was designed such that the screws on the side of the robot are oriented perpendicular to the sides, so that the robot handles direct hits from the side better since the joining screws would not be put under shear stress. The reasoning for not focusing as much on the orientation of the screws on the front and back of the frame is that the wedges handle hits from the front and hits from the back will be rare.

In addition, the type of screws to use was determined to be countersunk screws instead of button head screws because we learned that button heads could provide opportunities for an opponent's weapon to catch onto them and fling our robot around. By switching to countersunk screws whose heads are flush with the chassis frame, opponents would be much less likely to catch onto screw heads and therefore less likely to land lucky hits.

3.2.5 Invertibility

Considering the volatile nature of combat robot fights, it is very likely for the robot to end up flipped upside down. This is why we had to design for invertibility, otherwise the robot would be stuck and that would result in a default loss. We did this by making "rabbit-looking ears" on the frame. These would provide the points of contact needed (in addition to the wheels) to drive the robot in a flipped state without the vertical spinner colliding with the floor.

3.2.6 Weight Reduction

The bot ended up being overweight by three pounds, so to not exceed the weight limit we decided to scrap any extra armor or guards (iterations of which can be examined in the Appendix) after simulations proved that the frame could hold up well by itself (Fig. 10).

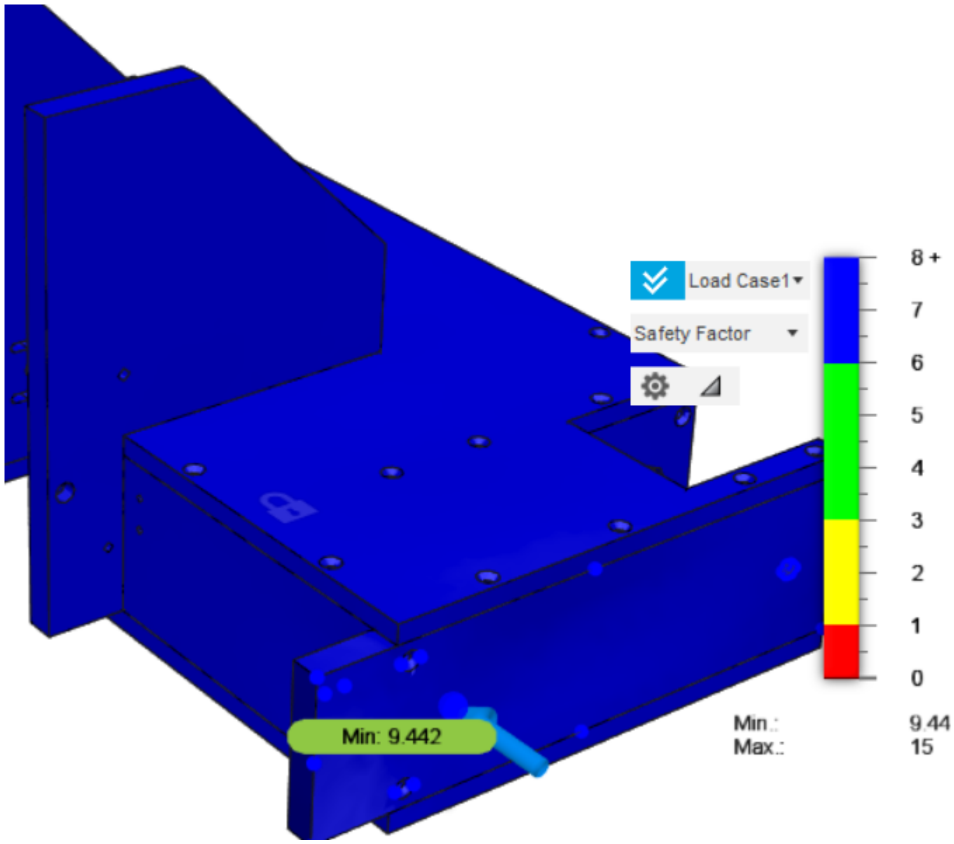

Figure 10: Simulations showed that the wall of the chassis frame itself is able to handle impacts without guards

Yet more weight had to be cut off, so slots and fillets were made within less important areas of the frame and the overall width of the robot was reduced. Reducing the width of the robot in turn reduces the distance between the two wheels, leading to less maneuverability. Since the width was reduced to the point that the distance between the two wheels changed from 9.5 inches to 8 inches, the new critical turning speed becomes approximately 200 degrees in one second based on Equation 4, just over 180 degrees per second.

3.3 Weapon

The weapon subsystem enables the robot to deal damage. As such, we decided to minimize our vertical spinner's spin-up time and maximize its kinetic energy while not drawing too much current and not weighing more than our fair share of the 12-lb limit.

3.3.1 Pre-PDR

Weapon Blade

There are multiple kinds of blade that work on vertical spinners, but the two most prominent types are bars and disks. During the general weapon design work session (before we split up into our subsystems), we opted for a bar spinner design; while disks can be more stable, they require CNC or waterjet machining. What's more, bar spinners are easier to machine on a manual mill — which means they're cheaper and will give us more chances to machine parts.

The same motivating factors helped us decide how many protrusions (or "teeth") to have around the weapon's circumference. We found that the goal was to minimize the number of teeth: this way, for a given rotational speed, there would be more time between blows (and therefore a larger contact area with an opponent). Limiting our weapon to having one or two teeth also allows each tooth to be thicker and stronger, whereas more teeth would have to share the weapon material, becoming thinner and weaker. The most obvious choice, then, is a single-tooth bar. However, this geometry requires a counterweight wider than the blade and is therefore harder (but still possible) to machine. We decided to postpone the decision between one or two teeth; we wanted to get feedback from the team at PDR.

During our work sessions, we often brought up the appeal of spikes at the end of our spinner which could catch onto a part of our opponent and help cause more damage. These extra spikes at the end would also add more complexity and excitement to our design, making the weapon look more interesting than a rectangular bar. Even though we desired spikes at the end of the spinner, we decided they are less of a priority and that it was more important for us to start simple: we would figure out the basics of the weapon subsystem first, and if we had room for more weight later on, we would design small spikes using scrap steel in the machine shop. We decided that the weapon blade would be made out of Aluminum 6061, since it is strong, lightweight and affordable. We chose the dimensions to be 0.5"x2"x6", as this seemed to be a reasonable size, with the intention of modifying the size if necessary during subsystem integration.

Defining the Subsystem

Now that we had solidified the blade's design, we needed to break down the rest of the subsystem into its parts, figuring out each part's purpose and determining if we would buy it off the shelf or design it in CAD. We settled on the following breakdown:

- Weapon Shaft → manufacture
- On weapon shaft:
  - Blade → manufacture
  - Pulley → depends on motor type
  - Ball Bearings → purchase
  - Screws → purchase
- Elsewhere:
  - Motor → purchase
  - Belt → purchase

Selecting a Motor Type

Next, we discussed the desired features of our weapon motor. We sketched two possible belt-driven designs: one using an inrunner motor (Figure 11) and one using an outrunner motor (Figure 12). The inrunner version would be heavier and have more parts, but it seemed more robust and customizable: the motor could be protected by flexible couplings and hidden within the chassis; the pulley could have support from both sides and be purchased rather than machined by hand.

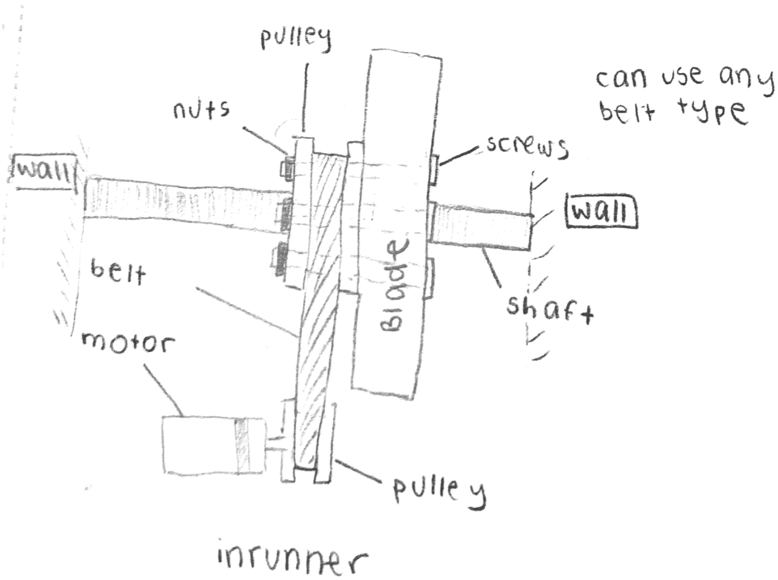

Figure 11: A sketch of a potential inrunner weapon design (not shown: flexible collars or support from both sides)

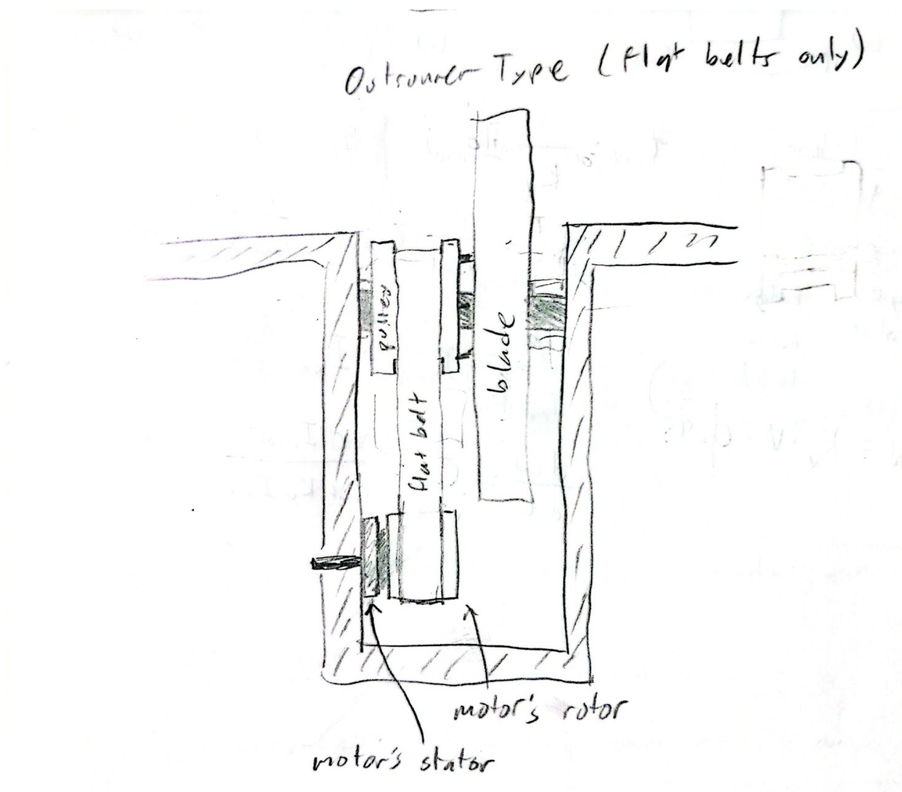

Figure 12: A sketch of a potential inrunner weapon design (not shown: flexible collars or support from both sides)

The outrunner version would allow us to have a more compact weapon drive system, reducing the subsystem's component count and total weight. Since an outrunner would require us to place the belt on the motor housing, we worried that the belt could slip off the motor during a battle. The outrunner motor would also need to be mounted outside of the chassis, making it more vulnerable than the inrunner motor option.

Another concern with the outrunner motor design is that the motor would only be supported on one side. Outrunner motors are typically designed to support propellers, which usually handle axial loads more than radial loads and don't provide mounting options for their far sides. This poses a risk to the motor: the belt could experience sudden jerks when the weapon hits an opponent, potentially damaging the motor or bending the frame. With these considerations in mind, we initially selected an inrunner motor design. With an inrunner motor, we planned to use a v-belt between two v-belt pulleys to minimize potential derailing.

Since we were contacting Georgia Tech's combat robotics team about some questions concerning simulations, we also asked them for their insight on motor selection. From their email response, we learned that our concerns with the outrunner design were unwarranted. Georgia Tech exclusively use outrunner motors for their weapons and get good results in our weight class. Intrigued, we decided reconsider outrunner motors. We went into the Preliminary Design Review with both motor options, seeking input.

3.3.2 Post-PDR

Mechanical Tweaks

After hearing more at the Preliminary Design Review (which Georgia Tech attended), we decided to follow Georgia Tech's advice and switch to an outrunner for our weapon motor. This decision led us to switch from a v-belt to a flat belt, replacing the rear pulley simply with the outer housing of the motor. The other feedback we received during the PDR was to ensure that if we add spikes to our weapon blade, we keep them small enough that the weapon doesn't get stuck in the opponent.

We did not receive any feedback with regards to the number of teeth for our blade, so we still needed to decide whether to include one or two. Since we have seen vertical bar spinner robots succeed with both one and two teeth, we felt that there was no obvious choice of one option over the other. Realizing that a one-tooth blade would require Fusion 360 scripting to design, we decided to pursue the two-tooth option for simplicity. This also allowed us to focus on the other components in the subsystem.

In order to transmit power from the weapon motor to the blade, we opted to bolt the pulley and blade together, allowing them to rotate together about the weapon shaft. This also allowed us to reduce the number of bearings to a total of two, rather than having four (two on the pulley and two on the blade). This also eliminated the need for a rotating shaft, because the torque would be transmitted through the screws holding them together rather than through the shaft.

Motor Selection

Originally, we planned to select our motor by calculating the desired specs for our weapon. However, after attempting to derive equations relating the blade's moment of inertia to desired torque and getting almost nowhere, we realized that there were too many unknown variables for us to come up with a list of specs for our motor.

Changing tacks, we searched for motors used by existing 12-lb vertical spinners, expecting to derive specs using those motors as a starting point. After searching the NHRL former competitor list and Georgia Tech's Wiki page to find similar robots to ours, we were only able to find the information for one outrunner motor used in a 12lb spinner. After learning more about this motor and how it worked on Melani, the 12lb horizontal spinner made by Georgia Tech, we decided to use their weapon motor as a starting point [7]. The exact model of the motor was out of stock, but we found an alternative with similar specs from Hobbyking. Using a MATLAB script made by Ricky, we found that our weapon would have a slightly higher speed and energy than desired. The equivalent weapon kV (and therefore the speed of the blade) is easy to refine by adjusting the size of the weapon pulley, and we were able to make this adjustment to make the alternative motor meet our needs. [8].

Isaac: it would be great if you can add our calculations and an explanation of the weapon energy from weapon meeting 3 in this section

Pulley and Belt Design

We only needed to design one pulley to complete the weapon drive system because the other "pulley" is the outrunner motor's housing. Since our selected motor would produce a higher speed than desired with no reduction, we decided to gear down with a reduction ratio of 0.8. Since the motor's housing has a diameter of 42mm, the pulley diameter is:

$$42mm \times 0.8 = 33.6mm \tag{5}$$

$$33.6mm * \frac{1in}{25.4mm} = 1.322835in \approx 1.3in \tag{6}$$

Since we use inches in our CAD, we converted this diameter to inches and rounded to the nearest tenth, leading to a pulley diameter of 1.3 in. After finalizing this diameter, we searched McMaster and other similar vendors for a flat pulley of this size. The prices seemed unreasonable, and as this pulley would be used with a flat belt, the geometry would be easily manufacturable, so we chose to design our own pulley. It should be noted that if we used a v-belt or timing belt, that it would be much more difficult to design our own pulley and manufacture it in the machine shop.

After deciding on the smaller diameter for the pulley, we determined the width and larger diameter of the pulley to give it an h-shape. The larger diameter had to be big enough to ensure the belt would not slip off the pulley, but not too large to be unnecessarily massive (since we have the weight limit to keep in mind). The width of the smaller section of the pulley was determined to be slightly larger the belt width - we added 1/16" to allow for a small amount of movement but not enough to result in slipping. The other dimensions of the pulley did not require calculations or have dependencies on other dimensions in the subsystem, so we aimed to minimize additional weight while also airing on the side of caution to prevent the pulley from being too flimsy.

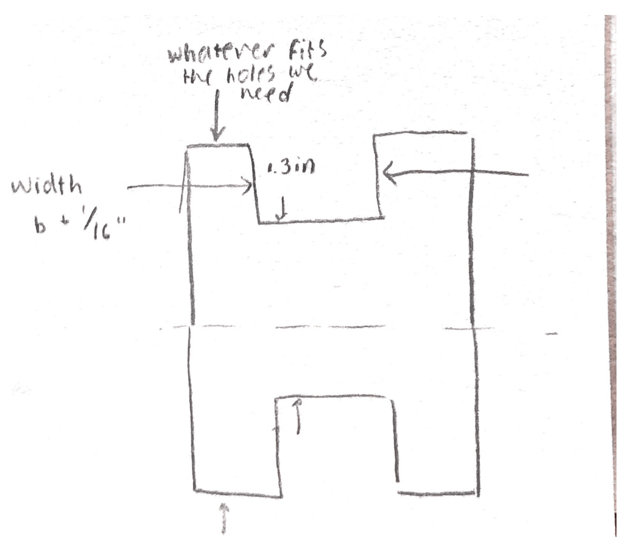

Figure 13: Original sketch of pulley geometry

The final pulley dimensions could not be determined until we calculated the belt width. The width of the motor housing is 18mm (or around 0.7 inches), so a 1/2" wide belt gives a balances between reducing costs while also still being robust. Since the motor is delivering a large amount of power to the weapon, we did not want to make the belt too thin.

The other parameter needed for the belt is the belt length. Belt length is dependent from efficiency of power transmission, so we realized that a longer belt would only cause the size of the robot (and thereby the weight) to increase. As such, we aimed to minimize the length of the belt. We first expected to purchase an off the shelf belt, as custom length belts from vendors like McMaster can be pricey. Unfortunately, we were unable to locate any 1/2" belts in the length range we needed. Most belts we found were either way too expensive, not the correct belt width or length by a long shot (for example, the belts were around 0.1 in wide), or the belts were affordable but did not include dimensions online. If it is possible to find the dimensions of replacement belts for vacuum cleaners and other similar appliances, and those dimensions are in the range of what's required for our robot, this can be a good way to get a cheaper belt, since these replacement belts were a lot more affordable than the ones we found on McMaster. Since we did not have luck with non-custom belts online, we had the flexibility to get any length belt, rounded to the nearest inch. Minimizing the length of the belt while also ensuring a little space in between the motor housing and pulley (and rounding to the nearest inch) led to a 9 inch long belt. [9]

One concern we had with minimizing the belt length was interference between the weapon motor and the blade, as the blade could potentially hit the motor while spinning if not placed sufficiently far away from the motor. To solve this issue, we made the pulley thicker on one side that the other, so the motor is not in the same plane as the blade.

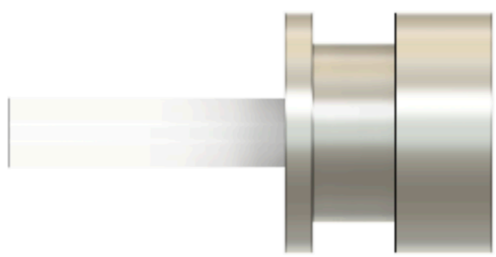

Figure 14: Final geometry of pulley. Changed from original design to prevent interference with weapon blade

ESC Selection

When we selected our weapon, Hobbyking had a recommended ESC (Electronic Speed Controller) at the bottom of the webpage, so we intitially planned to use that ESC to simplify our subsystem work. However, when finalizing our CAD, we learned that the ESC was out of stock on the website and that the model was not available on other websites.

While the motor specs suggested an ESC with the capacity to deliver 70-80A, our subsystem did not have any knowledge on how to select an ESC, so we consulted the Riobotz guide to learn what each parameter means for the ESC and how it relates to the motor specs. After learning that our ESC's normal current draw should be higher than the motor's maximum current draw, and that the ESC needs to be compatible with our 4S LiPo battery, we found a few options at various price points with slightly different specs. Ultimately, we chose an ESC that has a slightly lower current draw than the recommended range on the motor's specs (by 10-20A), because our selected weapon ESCs were nearly half the price of the ESCs that can deliver the recommended current draw. [9]

Purpose: To display multiple appropriate options for the ESC for the weapon motor

Info about how to select ESCs (from Riobotz, pages 145-146)

1) Select hobby ESC (electronic speed controller) with a higher voltage than motor. ESC is usually given in terms of cells, so you can divide your motor's voltage by 1.2 to give an equivalent cell rating. Then, choose a controller that has a higher cell rating.

2) Find an ESC with a higher current rating than the robot's current draw (use reverse current rating). **I am pretty sure they mean the current draw of the motor(s) connected to this ESC**

*Continuous current rating is approx ¼ max

Motor Info								
Part name with Link	**Description**	**Supplier/Vendor**	**kV**	**Max current**	**Normal current draw**	**Max Voltage**	**Recommended ESC rating**	
PROPDRIVE v2 4238 750KV Brushless Outrunner Motor	Outrunner motor	Hobbyking	750	55A	13.75A	17V	70-80A	
ESC Info								
Part name with Link	**Description**	**Supplier / Vendor**	**kV**	**Max current**	**Normal current draw**	**Voltage range**	**Price**	**Comments**
Turnigy 100A Speed Controller w BEC [Battery Elimination Circuit]	-	Hobbyking	-	120A	100A	5.6V-26V	~~$48.72~~ $14.54	Out of stock. UBEC mode: 3A/5V out (not sure if this is necessary)
70A Drone ESC	-	Amazon (Dilwe store)	-	75A	70A	7.4-25.9V	$38.59	BEC Output: 5V / 5A. Says it's compatible with a 2-7s LiPo
70A Aircraft ESC	-	Amazon (RBSD)	-	80A	70A	7.4-22.2V	$48.19	SBEC (Switching BEC) - for "power hungry electronics". BEC: 5.5V / 5A. Says it's compatible with a 2-6s LiPo
60A hobbyking	-	Hobbyking	-	80A	60A	7.4-14.8		SBEC (Switching BEC) - for "power hungry electronics". BEC: 5.5V / 5A. Says it's compatible with a 2-4s LiPo
						7.4		
Examples of ESCs that would not work								
80A for RC Truck ESC	-	Amazon (GloGLOW Store)	-	400A	80A	7.4-14.8V	$39.59	Supported Motor Type: Brushed (instead of **brushless**) Motor (540/550/775 motor)
80A for RC car	-	Amazon (Dilwe Store)	-	480A	80A	7.4-11.1V	$42.49	Product Applicable Battery: 2-3s Lithium Battery -- **not enough voltage**. Also product pictures say not to put a "burden on the car"? Not sure if this is just saying that the ESC will not put a burden or if it can't handle a load...

Figure 15: Screenshot of the ESC selection guidelines and options, with the chosen ESC highlighted in lime green

Mounting and Joining Elements

To mount the weapon motor, we needed to attach its stator to one of the weapon side plates with four screws. As such, we added the necessary hole pattern to the right plate, including a central hole to keep the rotor out of contact with the wall. Also, to allow belt tension adjustments, we designed slotted holes for the screws and oversized the central hole.

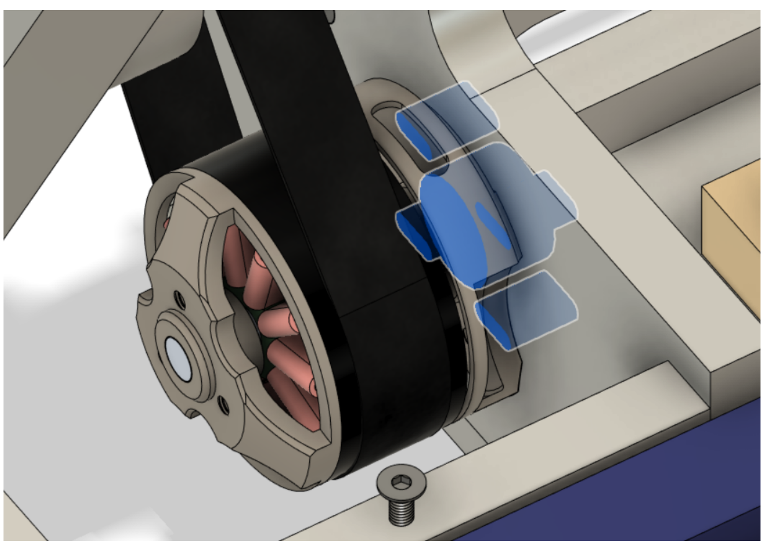

Figure 16: The weapon motor is mounted to the right weapon plate. The mounting holes are highlighted in blue.

To hold the main part of the weapon assembly in place, the weapon axle extends 1/8" into the

3/8"-thick plates that sandwich it (Figure 17). This way, the shear forces from blows are transferred to the plates without relying on the shear strength of the M4 mounting screws.

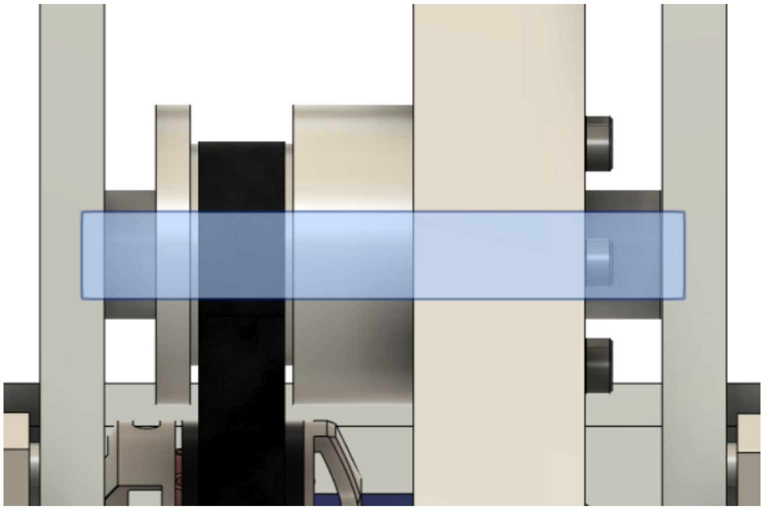

Figure 17: The weapon shaft is highlighted in blue and extends slightly into the weapon walls.

To stabilize the weapon on its shaft while allowing high-speed rotation, we needed to use two bearings for each rigid object: one on each side. If the pulley and blade were separate, that design would require a total of four bearings. However, we decided to directly attach the pulley to the blade using six M4 screws. This simplification allowed us to use a total of two bearings: one on the left of the blade and one on the right of the pulley. To fully constrain the system, the blade and pulley are kept from sliding axially by two 3D-printed spacers (Figure 18).

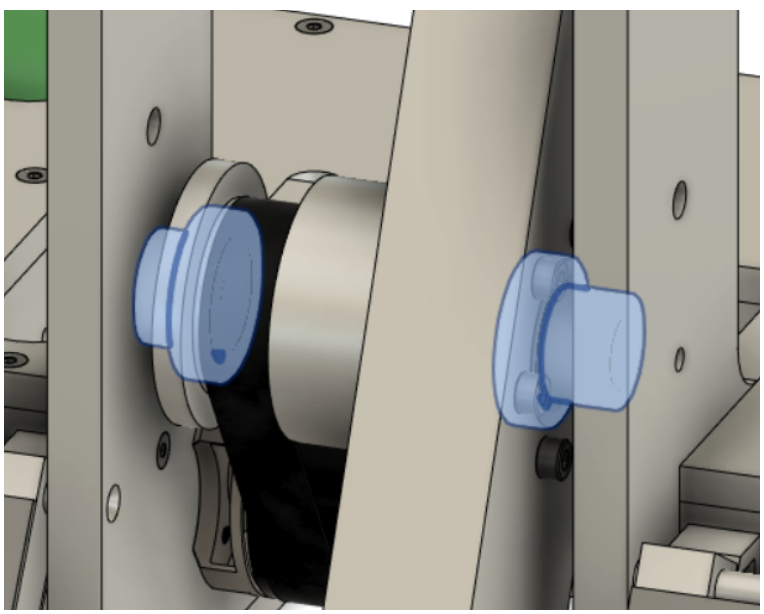

Figure 18: The bearings (towards center) and spacers (towards walls) are highlighted in blue.

3.3.3 Final Status

After the Critical Design Review (CDR), we did not make any changes to the weapon subsystem. Although the robot was overweight, we realized that all of the components in the subsystem needed to stay the same and that components in other subsystems could be modified to achieve our 12 lb weight limit. Even though the size of our components did not change, the positioning of the weapon motor (with respect to the blade and pulley) was moved from horizontal to diagonal to allow the size of the chassis to decrease without causing the motor and pulley to interfere with one another. As of January 2022, we still have not implemented the spikes at the end of our weapon. We may design and manufacture spikes and see if they help the weapon cut into our opponent during the testing phase, as we intend to use scrap metal from the shop or other project teams for the spikes, if we make them.

In summary, the weapon subsystem consists of a two tooth rectangular bar spinner powered by an outrunner motor, which is connected to the blade using a H-shaped pulley and flat belt. The blade is mounted in the "ears" of the chassis, while the motor is mounted in the center cavity of the frame.

4 Training

4.1 Newbie Fundamentals

The various workshops hosted through the fall semester included design fundamentals, electronics, weapon design, documentation, materials, joining elements, and motors and transmissions. Lots of material was covered that included everything from the general design of the bot, such as the rock-paper-scissors structure, to the intricacies of the electronics that can power it, such as radio transmitters, servos, and solenoids. The first meeting composed of setting up our Fusion 360 and connecting into our shared work spaces.

Following these workshops, Shubham Mathur, the fundamentals organizer, would give a presentation on more mechanical engineering-specific topics, such as machining practices, various tools, schematics, and Fusion 360 basics. Our Fusion 360 training began with the very basics, simply downloading the correct version and getting access to the Combat Robots Cornell workspace, but as the semester progressed we began making sketches and models of our chosen BattleBot Bronco and learning how to use joints to make a full assembly. Fusion 360 tutorials were also given to us to learn tips and tricks on our own time.

4.2 Subteam Specific

On a subteam specific level, we spent a lot of time getting caught up with the current design of the bot, learning why certain decisions were made: for example, the type of weapon, the type of motor, the type of pulley to run the motor, etc. After learning the ins and outs of our bot, we were given specific tasks which further increased our understanding of the robot and how the subteam functions as a whole. In the chassis subsystem, the focus was on learning CAD fundamentals as soon as possible, understanding the strengths of different materials, figuring out different ways to conjoin parts, and adjusting parts based on available manufacturing methods. Within the Powertrain subsystem a good amount of time was dedicated to understanding the parts chosen in addition to the equations used to understand whether their output was sufficient to move the bot.

5 Reflection

5.1 Design Process

The design process was good in the ideation phase. Although it took some time since it was ensured that everyone had a say, it made sure that everyone was on the same page and had a deep understanding of the general robot design that we would go for.

However, the design process felt a bit slow. Not only did we start a more rigorous design process a bit after the start of the semester, but also reading ground was not covered efficiently since people were not assigned different resources for research. To fix this slow timeline, the design process could be started earlier or more meetings can be held during the more rigorous parts of the design process.

Another problem we faced in the design process was that there were too many free variables. Due to the fact that there were so many parts of the robot that depended on each other, we got lost in terms of which part of the robot to design first. Eventually, we realized that we just had to choose some variables and set them to some value(s). By doing so, it created a starting point in the design and we were able to proceed from there.

5.2 Lessons learned

Presentation Context

Every subteam on CRC participated in design reviews, including the PDR. A big problem we realized within this design review was that we did not cater our presentation to people not on CRC. We had assumed that everyone in the audience knew about combat robots and what the goals of the competition were to some extent, so we brushed past the "context" parts of the presentation too quickly. This caused our audience to have less of an understanding on the goals that CRC wanted to reach, which made it harder for them to ask good questions or provide feedback.

Simulations

Due to the slow design process and a lot of lost time, we were unable to perform rigorous simulations on the CAD of the robot in the end. Only one or two Fusion simulations were done at the end of the semester, when any and all simulations should actually be done alongside the CAD of the robot to determine any necessary modifications within the design phase.

5.3 Recommendations for future projects

Subsystems

Although we created subsystems so that everyone had work to do, there would be no point if there were not enough tasks within the subsystem. This is why everyone on a subsystem should be assigned a specific part of the project that they can work. For example, the chassis subsystem did this by splitting up parts of the chassis into the wedges, the frame, and the wheel guards. All three members of the chassis subsystem were then able to constantly have something to work on.

It should also be noted that some subsystems' workflow happens before others, like Powertrain. In these cases, it should be made sure that people on other subsystems "float" and help the earlier subsystems.

On the design process

In a lot of ways, we made design decisions based on what was more simple so that we could figure out the design of all the other components (to prevent us from lingering on one or two components that could be made more complicated but would require attention that held back the other parts of the robot). While this was the right decision this year, since we had a lot of new territory to discover with making our first robot with the thorough design process, in future years it would be wise to explore some of those more complicated designs (for example, a one tooth blade). In order to prevent the more complicated components from holding back our timeline, we can split the subsystems up even more so team members focus only on a few components, allowing us to dive deeper into a complex (but not overly complex) and exciting design. That said, it is recommended to take the approach of "starting simple" and avoid options that are unnecessarily complicated.

Something that also needs to be figured out is if FMEA is useful. This should be done at the beginning of every design process. If it is, it should be added to the plans of the project and done at some point. The issue that we had with FMEA in the design process was it was never definite. We weren't really sure if we were going to do it or not, and, in the end, we didn't perform FMEA when it could have potentially been useful with more consideration.

Since our ideation process went well, we recommend that it is ensured everyone gets to pitch in during the idea generation phase. Even just making sure that each person has a whiteboard marker implicitly allows people the equal chance to share ideas. If you notice someone does not have much to share (maybe they didn't find any new information than what was already discussed), it is helpful to give them a topic to look deeper into so they have something unique to share with everyone. It is also helpful to check for understanding while communicating ideas — and to teach everyone about what is being discussed.

Component selection

The Job of component selection is tough because you are juggling expedience, quality, and price (which is also determined by expedience). One of the major issues which occurred during our year was an issue with buying form Banggood. The motors we bought from them were originally supposed to arrive to our school by Mid February at the latest upon buying but because they were being sent from China the items were delayed by over a month.

Lesson Learned 1: Give 2-3 months leeway if you are buying motors from China. Especially if it is an essential part that is hard to replace.

Solution: The solution we came to was having a member directly call the company and explain how the delay was unacceptable. In response they offered a full refund (presumably including shipping) should it not arrive in time for the competition. They also are looking into US based warehouses as to whether there are any that could be rerouted from a closer warehouse. If a future team faces this same problem, we recommend kindly but firmly explaining to said company that the delay is unacceptable and then ask what other options they have to get the thing here in time.

Lesson Learned 2: Shipping costs add up quickly.

Solution: Limit the number of vendors if you are buying items. Buying them in bulk allows you to save on shipping costs as those will increase to a substantial amount over time. We tried to use companies that shipped slower but cheaper too, in order to help further save on costs like using Clark instead of McMaster which ships quickly but costs far more as a result.

CAD

In our process, we did not have many issues with having to fix joining elements that don't work out, but one more recommendation for future projects is to absolutely make sure that all screws and fasteners are included in the CAD for the final design review. Forgetting to add a few or not being positive about which you are using can cause you to have to circle back and fix many things in the CAD. We recommend having team members critically review the CADs of other subsystems (to get an outside perspective), or even have Sportsman review our CAD and we review theirs, at or right after each design review so we can catch mistakes quickly. While it would be great to get people who are outside of CRC to take a look at our CAD too since they may be able to catch mistakes more easily than us (as an outside perspective looking at a CAD for the first time), it is a little unreasonable to expect someone to look at a design in that much depth for a project they are not a part of.

5.4 Fundamentals Training

For newbie training, one of the largest learning priorities for kinetic team members for future years would be an increased emphasis on Fusion 360. The majority of the team's time throughout the year will be CADing the robot. As such devoting a full session to ensuring no downloading issues, loading up the shared Fusion 360 space, how to move within said space, good practices (proper encapsulation of parts into file system), hot-keys (how to make screw holes quickly and easily), working on practice problems/cads, should become the first and should be a mandatory attendance meeting. Even if someone is already experienced in this topic, it is good for everyone to have a review and will enable everyone to start with the same base understanding of Fusion.

References

[1] Marco A Meggiolaro. *RioBotz Combat Tutorial*. RioBotz. Rio de Janeiro, Brazil, 2009.

[2] Kinetic.Powertrain - Mohammad Ali Moghaddasi. *Powertrain Parts List and Cost Analysis of our 3-lb robot*. URL: https://docs.google.com/spreadsheets/d/1Sa74Eq-L_iRkhP0jizSuoDHAQRpuHOaf1VM_76OlbJA/edit#gid=1762537304.

[3] Kinetic.Powertrain - Mohammad Ali Moghaddasi. *Powertrain drivetrain speed calculations*. URL: https://docs.google.com/spreadsheets/d/19lR2sViIkuDyvllqSbQAf0LIXIDXdkAjpXFmjn1vrIQ/edit#gid=0.

[4] Just 'Cuz Robotics. *Drivetrain Calculations*. URL: https://docs.google.com/spreadsheets/d/1TuYqQjuKuVg45yB0CH9xKMNMJVlubS5YVwWn0h92HVA/edit?usp=sharing.

[5] Team Tentacle. *Motor Calculations*. URL: http://runamok.tech/squid/newtorquecalc.htm.

[6] Kinetic.Powertrain - Mohammad Ali Moghaddasi. *Powertrain Parts List and Cost Analysis of our 12-lb robot*. URL: https://docs.google.com/spreadsheets/d/1Sa74Eq-L_iRkhP0jizSuoDHAQRpuHOaf1VM_76OlbJA/edit#gid=151526841.

[7] Georgia Tech RoboJackets. *Melani - 12lb Horizontal Spinner*. URL: https://wiki.robojackets.org/Melani.

[8] Kinetic.Weapon - Ricky Wang. *Weapon Motor Speed Calculations and Reduction Ratios*. URL: https://drive.google.com/file/d/1yFGwNkkFp--eojLax8S7MLuLt5r6SiPh/view?usp=sharing.

[9] Kinetic.Weapon - Sofie Halpern. *Specs and Calculations: Weapon Belt and ESC*. URL: https://docs.google.com/spreadsheets/d/1gx4_NmnsH-7PWO9SzGikZShwzBI-9qm0hX_GOI_vvl4/edit?usp=sharing.

6 Appendix

6.1 Guards

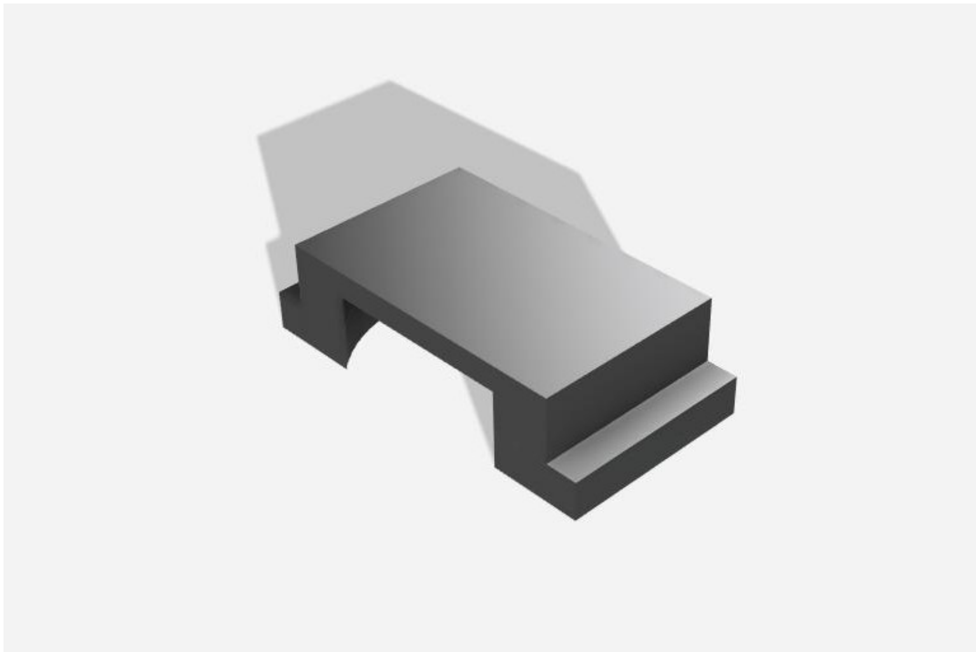

Figure 19: Original Wheel Guard Design

Figure 20: Second Wheel Guard Design

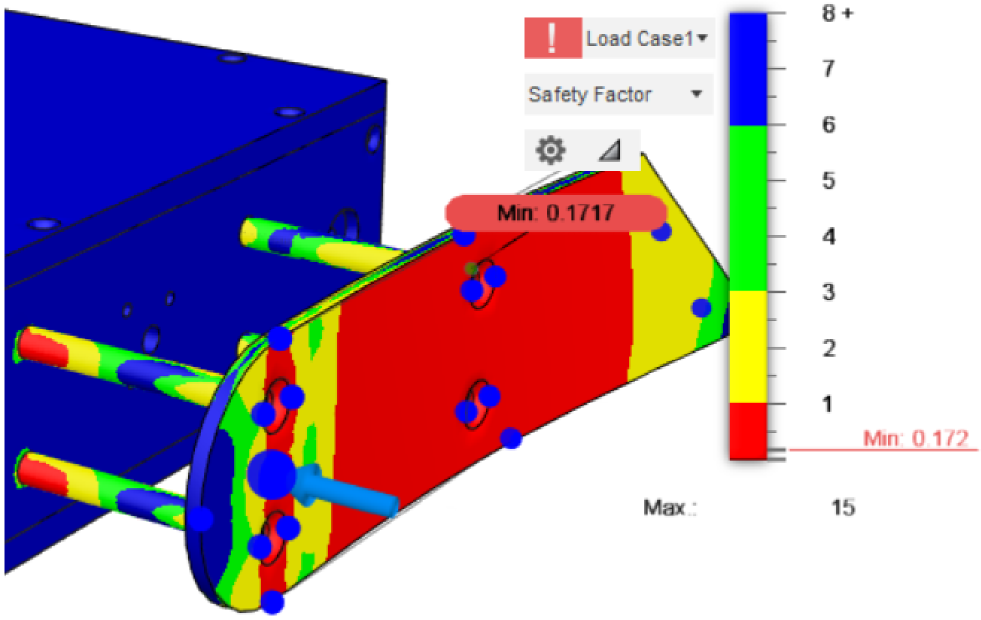

Figure 21: To approximate a dynamic impact force on the standoff guards, a 1000lbf static load was simulated. It can be seen that the standoff guards fail almost completely.

Figure 22: Third Wheel/Wall Guard Design

6.2 MATLAB Code for Motor Speccing

```
% input parameters
botweight=12; % lbf
wheelweight=6; % weight supported by wheels (lbf)
motorsperside=1; % motors per side of robot
wheeldia=2.875; % inches
gearing=3; % gear reduction ratio
armres=0; % resistance (ohms), INPUT 0 IF NOT KNOWN
kv=400; % voltage constant (RPM/volt)
maxv=14.8; % operating voltage
stall=23; % max amperage
cf=0.9; % friction coefficient
side=16; % arena side length (ft)
interval = .01; % interval used for discrete approximation

% calculated parameters
if armres==0
    armres=maxv/stall; % resistance (ohms)
end
kt= 141.61*.85*1/(kv*2*pi/60); % estimated torque constant (oz-in/amp)
nlrpm=maxv*kv; % no-load rpm
botspeed=nlrpm/gearing/60*pi*wheeldia/17.6; % topspeed (MPH)
corner=sqrt(2)*side; % corner-to-corner length (ft)
maxpush=wheelweight*16*cf; % effective pushing force (ozf)
mass=botweight; % bot mass (lbm)

% variables
accel=0;
timetospeed=0;
disttospeed=0;
timetoside=0;
timetocorner=0;
speedside=0;
speedcorner=0;
v0=0;
vf=0;
dist=0;
amps=0;
rpm0=0;
rpmf=0;
force_avail=0;

% calculation
% Discrete approximation loop.  Using the selected motors torque characteristics
% loop through the rpm and amperage calculations and accelerate the motor for the
% interval, accumulating distance traveled and time elapsed until the end RPM equals
% the no-load rpm.  Account for traction limit and motor stall.
while (round(rpmf) < nlrpm)

    % determine the torque available for acceleration
```

```
    amps = (-1*(rpm0/kv-maxv))/armres;
    force_avail = amps*kt*motorsperside*2*gearing/(wheeldia/2); % ozf

    % if the torque is greater than the maximum pushing force, use the
    % max pushing force - this accounts for wheelspin

    if (maxpush < force_avail)
        force_avail = maxpush;
    end

    % Determine the acceleration, A=F/M

    accel = force_avail/16*32.174/mass; % ft/s^2

    % now accelerate using the available torque for the interval

    vf = v0+accel*interval; % ft/s
    dist = v0*interval+(accel*interval*interval)/2; % ft
    rpmf = ((vf*60*12)/(pi*wheeldia))*gearing; % RPM

    % accumulate the results

    timetospeed=timetospeed+interval;
    disttospeed=disttospeed+dist;

    if ((disttospeed >= side)&&(speedside==0))
        timetoside = timetospeed;
        speedside = vf*.682;
    end
    if ((disttospeed >=corner)&&(speedcorner==0))
        timetocorner = timetospeed;
        speedcorner = vf*.682;
    end

    % set up the next iteration
    % cut off the loop once we are no longer accelerating noticeably -or-
    % we are at our max speed or RPM.

    if ((vf-v0 < .01) || (vf*.682 >= botspeed))
        break
    end

    v0=vf;
    rpm0 = rpmf;

end

% Calculate the time to finish a 60 and 45 foot box rush if
% disttospeed < distance of rush

if (disttospeed < side)
    speedside = vf*.682;
    timetoside = timetospeed+(side-disttospeed)/vf;
```

```
end
if (disttospeed < corner)
    speedcorner = vf*.682;
    timetocorner = timetospeed+(corner-disttospeed)/vf;
end

% Check for sufficient torque and factor of safety
% if (kt*stall*gearing < maxpush*wheeldia/2)
%     disp('Not enough torque!')
% else
    FOS=kt*stall*gearing/(maxpush*wheeldia/2);
    fprintf('Torque Factor of Safety of %.2f\n',FOS)
% end

% output
fprintf('Effective Pushing Force: %.2f lbf\n',maxpush/16)
fprintf('Time to Top Speed: %.2f sec\n',timetospeed)
fprintf('Distance to Top Speed: %.2f ft\n',disttospeed)
fprintf('Top Speed, Side-to-Side: %.2f mph\n',speedside)
fprintf('Time to Side: %.2f sec\n',timetoside)
fprintf('Top Speed, Corner-to-Corner: %.2f mph\n',speedcorner)
fprintf('Time to Corner: %.2f sec\n',timetocorner)
```

6.3 MATLAB Code for Weapon Spin-up

```
% Calculator for Kinetic Weapon using Brushless Motor
% input parameters
botmass= 12; % lbm
voltage= 11.1; % applied voltage (V)
kv= 520; % RPM/V
res= 0.016; % resistance (Ohms)
MOI= .0338; % moment of inertia (kgm^2)
gearing= 2; % gearing/pulley ratio
maxamp= 100; % max continuous amp (A)
maxP= 2000; % max continuous power (W)
e= .85; % efficiency
interval = .01; % interval used for discrete approximation

% calculated parameters
botmass= botmass/2.2; % kg
nlrpm= voltage*kv; % no-load rpm
kt= e*1/(kv*2*pi/60); % estimated torque constant (Nm/A)

% variables
rpmf= 0;
rpm0= 0;
apms= 0;
torq= 0;
alpha= 0;
w0= 0;
wf= 0;
timetospeed= 0;

% calculation
% Discrete approximation loop until 90% of no-load RPM is reached.
% Amperage is based on current RPM
% Torque is based on amperage and gearing/pulley ratio
% Newton's 2nd Law and rotational kinematics are utilized
while (round(rpmf) < .9*nlrpm)
    amps = (-1*(rpm0/kv-voltage))/res;
    if (amps>2*maxamp)
        amps=2*maxamp;
    end
    torq = amps*kt*gearing;
    alpha = torq/MOI; % angular acceleration (rad/s^2)
    wf = w0+alpha*interval; % rad/s
    rpmf = (wf*60/(2*pi))*gearing; % RPM
    timetospeed=timetospeed+interval;
    if (rpmf-rpm0 < .01)
        break
    end
    rpm0 = rpmf;
    w0 = wf;
end
E = 0.5*MOI*wf^2; % energy (J)

% output
```

```
if (E>60*botmass)
    fprintf('Sufficient Energy! FOS of %.2f\n',E/(60*botmass))
else
    fprintf('Insufficient Energy, need %.2f J\n',60*botmass)
end
fprintf('Final Weapon Energy: %.2f J\n',E)
fprintf('Final Weapon Speed: %.2f rad/s or %.2f RPM\n',wf,wf*60/(2*pi))
fprintf('Spin-up Time: %.2f s\n',timetospeed)
if (amps<.8*maxamp && amps*voltage<maxP)
    disp('Safe to use')
elseif (amps>=maxamp || amps*voltage>=maxP)
    disp('Unsafe to use')
else
    disp('May be risky to use (shorter lifetime)')
end
```