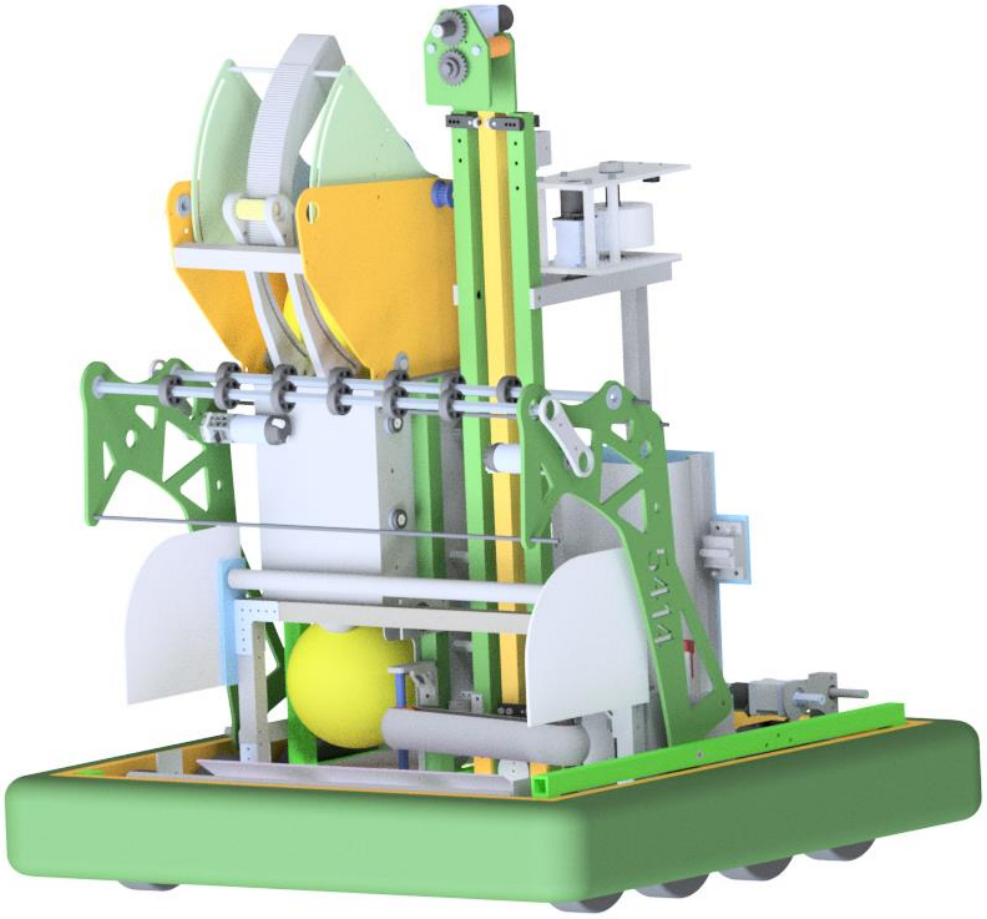
Pearadox

— Robotics Team 5414 —



StarScraper

Technical Binder

Table of Contents

Summary

Analysis:

Our breakdown and analysis of this year's game flew into full motion just minutes after we finished watching the *Infinite Recharge* reveal video. We studied every aspect of the game, going in depth into scoring, penalties, and more.

We realized off the bat that the autonomous period was an important strategic asset, as every power cell scored then is worth double points. This led us to begin plotting out several feasible routes and strategies for this period, each with small variations to the order in which we shot, collected, and handled power cells. We recognized some of them as far more difficult to pull off than others, but these riskier options often came with greater possible rewards.

As the Teleoperated Period (or TeleOp) is most of the game, we of course, put most of our thought into our strategies here. Our most central debate was in the tradeoff of making long-distance shots to reduce cycle

times at the potential cost of decreased accuracy. We performed calculations to try and estimate which would be more viable, but as everything was purely theoretical, it was hard to make a single decision. We also factored the advancement between Stages and manipulation of the Control Panels into our TeleOp strategy. We tried to figure out whether it would be worth it to include a device for spinning the Panels on our robot, and how much time going to interact with them would take out of our cycles.

The Endgame and climbing were the last, but certainly not least, of what we analyzed. Our climb was a staple on our robot last year, and we wanted to thoroughly research what strategies could be effective for climbing this year. Would we want to bring other robots with us? Would we want to move along the bar once we've climbed? Climbing awards, a handsome number of points, especially if multiple robots pull off a balanced climb, so we wanted to make sure climbing was something we could do well.

Priorities:

After our kickoff we placed design priority on climbing and having a reliable shooter. Our reasoning behind this is to maximize the amount of points possible. The climb (our number one priority) is able to get 40 points with just a single robot that climbs and balances which is more than enough to give us an advantage in our matches and opens up the possibility for a ranking point if an alliance member is able to climb as well. Then we decided to prioritize a consistent shooter because this would opens lots more scoring options and allows us to generate points throughout the match as well as open opportunities for the color wheel to be scored.

Everybot:

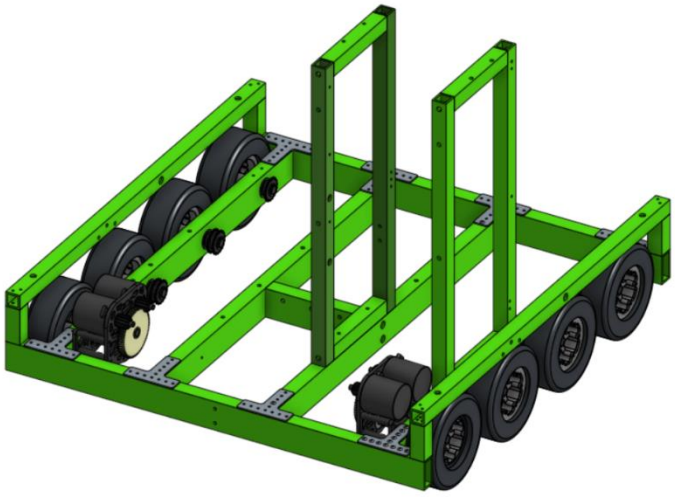
As part of our team's strategy, we created a subsystem specific to building the Everybot. The goals of this project were to give some new team members experience building an FRC robot from start to finish and to learn how to help other any alliance members that may use an Everybot. By building the Everybot, we know what parts are likely to break and parts that we can adapt to work better for potential alliance members. We are also able to use this robot for driving and defense practice.

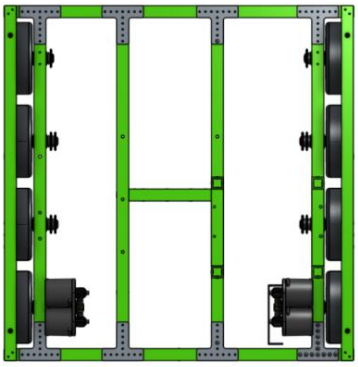
During competitions, some of the students who worked on building the Everybot will be able to assist teams who need help building or repairing their robots. We have also put together an Everybox of parts for the Everybot that includes pieces which break often.

Design- Drive Train

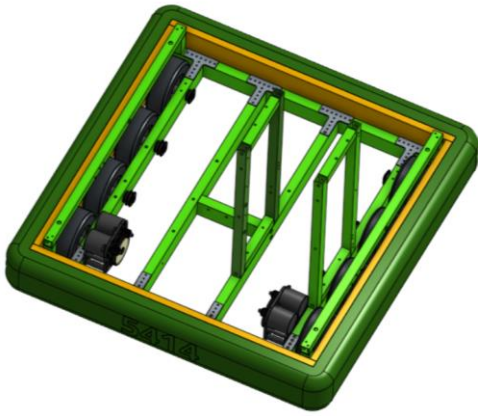
- **Design Goals**
  - Reducing impact from berms
  - Sturdy
  - Reduce gearbox size
  - Have a stable robot
- **Design Goals**
  - Pnuematic Tires
  - 2x1in 1/8in Tubing
  - Flipped Gearboxes
  - H support in center of bot
  - NEOs

Design Progression





We had one design with the only real changes being mounting holes for other subsystems and including the ball tower frame. We used flipped gearboxes so that the gearboxes wouldn't take up much of the robot. Base used pneumatic wheels that can easily go over the berms. Base used Neos that are stronger and lighter than cims.



Design - Intake

- **Design Goals:**
  - Touch it own it
  - Lightest as possible while still being effective
  - Over the bumper
  - Raise and lower in less than .5 seconds
- **Implementation**
  - ¼" Lexan
  - 5:1 Ultraplanetary gearboxes with Neo550 on rollers to make as short as possible as to not interfere with the ball path
  - 81:1 2 stage VersaPlanetary on a neo550 to raise and lower

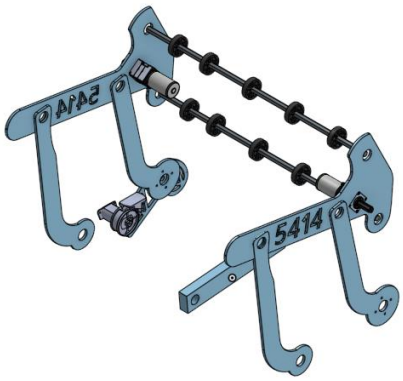
- **Retractable window shades on the top and bottom to funnel balls in and keep them in.**
- **#35 Chain driven up and down to reduce the possibility of slipping**

Design Progression

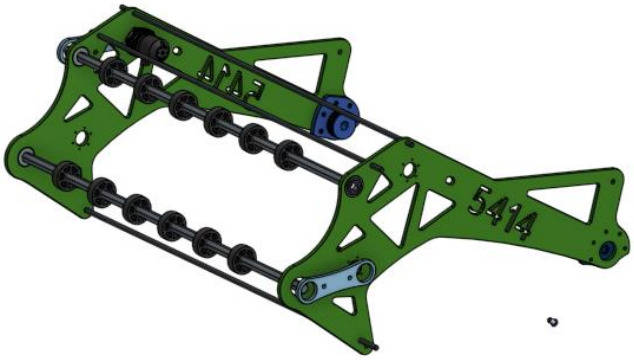
First Design:

We did a lot of prototyping to discover dimensions and what a possible ball path would end up looking like. Early prototyping revealed we would need to have some arm that would go down and in front of the bumpers. In later prototyping we discovered many possible dead zones in the ball path. To fix these dead zones we shortened the roller motors by using UltraPlanetary gearboxes. And adjusted where the window shade would sit on the bumper when down.

At first we thought a four bar design would be the best way to go as it would take up small amount of space in the robot and raise and lower parallel so we didn't have to worry about one side coming out of sync of the other side. We eventually decided on the one point of rotation instead as it was much simpler and provided everything we needed.



Final Design:



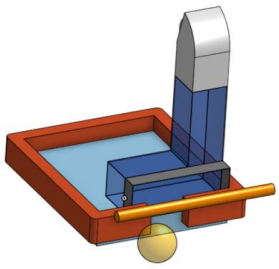
Design - Ball Hopper

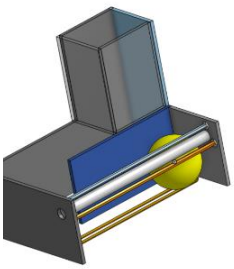
- **Design Goals/Tasks**
  - Our task was to transport the power cells from intake to the ball tower/shooter.
  - No jamming
  - Be able to fit in limited space given
  - Prevent power cells from escaping
  - Integrate well with intake and ball tower/no interferences
- **Implementation**
  - 2 1”x 1”x 24.5” and 2 1”x 1”x 8” aluminum bars
  - 1 Poly belt and 2 crowned pulleys

- One of the crowned pulleys have a bearing inside because we wanted a dead axel
- 1/8” Lexan for floor of hopper and ball tower
- 2 2”x 6.25”x 1/4" pieces of lexan on the sides for intake’s window shades
- 10:1 Versaplanetary attached to bag motor

Design Progression

Original Idea:

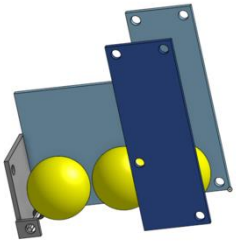




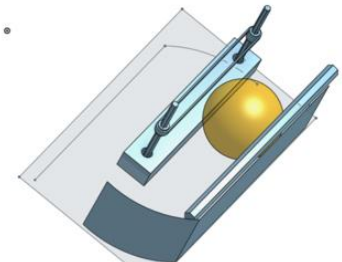
We were first given the idea to make a similar design to these but not an exact replica and inspired our first few designs.

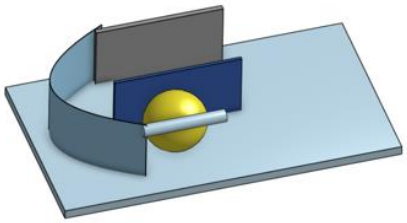
First Designs:

In these designs, they had a similar concept to have them go in a curve and then into ball tower.



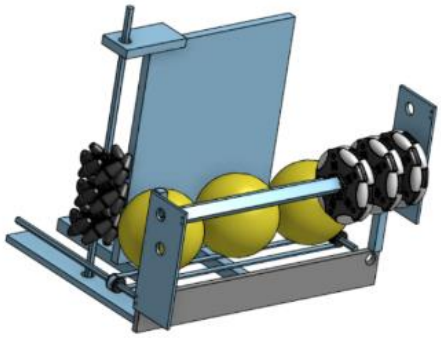
2020 echnical Binder

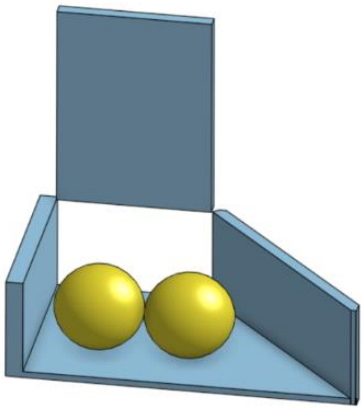




Second Designs:

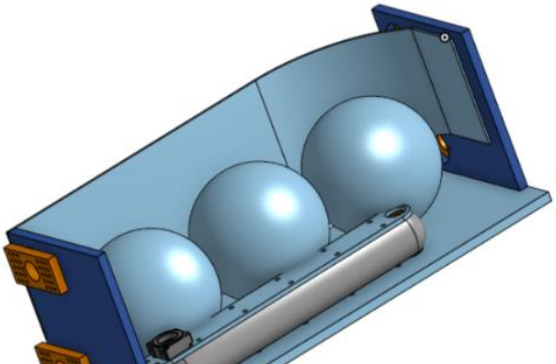
We experimented with different amou
positions of omni wheels and mecum
We found that many power cells get
when transitioning from the belt, to t
mecanum wheels. The other design w
shape and gravity-fed. There were su
be belts that run in opposite direction
sides. There were no jams, but it too
of space

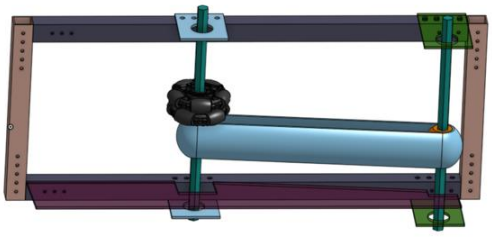




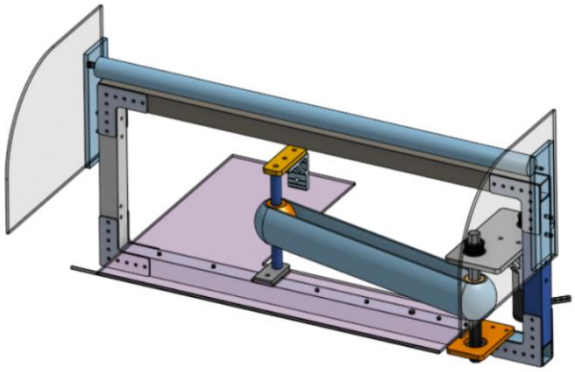
Third Design:

We started using Poly belt, having the belts run along the side. We added adjustable plates on the side to adjust the position of the shafts that will collect power cells. We meant to have intake bring the balls about 8" above the bumper, but we decided to not have that.





We removed the sides and roof of our design because we realized intake was going to be our sides and roof. We added an omni wheel above the belt and angled our belt. Lexan was added to act as a bridge between intake and ball hopper.



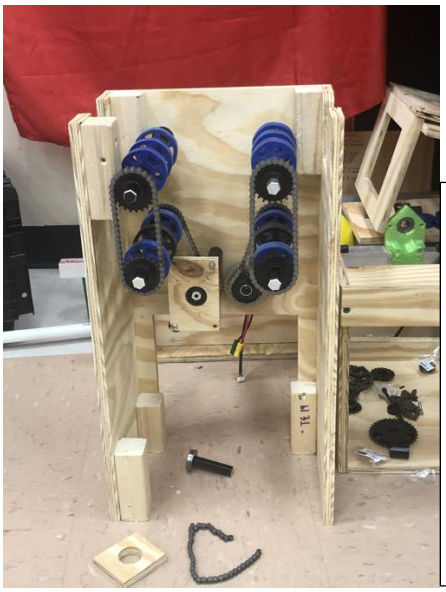
We focused more on mounting our motor and the hopper to the base and along ball tower. We kept the ramp but added a floor for ball tower and added side plates to prevent power cells from escaping the robot.

Design - Ball Tower

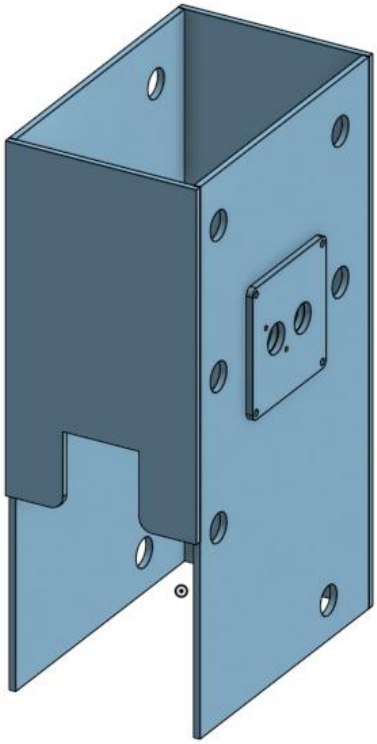
- **Design Goals**
  - Automatic Ball Indexing
  - Be as light as possible while still providing rigidity to the shooter
- **Implementation**
  - 1/16” Lexan for body of tower
  - 10:1 Versaplanetary attached to Bag motor
  - Poly belt and crowned pulleys for ball movement
  - Gears to change directions of input rotation
  - Superstructure
    - 4 1” x 1” solid aluminum bars
    - 4 1/8” 1” x 1” aluminum tubing for support

Design Progression

First Design

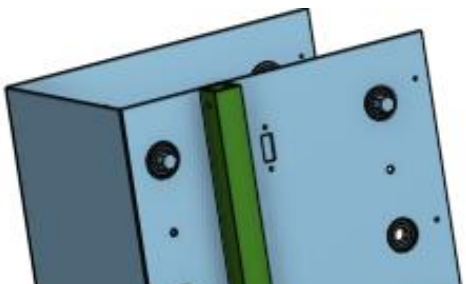


The first design involves 12 2” compliant wheels that is made of plywood. Many problems were caused by the lack of design concept and planning. The gearbox hit cut the balls, the chain also cut the balls, and there was too much compression.



The second design involves a poly belt system that allows a continuous flow of movement without having multiple chain linkages. The gearbox, gears, and chain have all now been moved to the outside to prevent unnecessary contact with the ball.

Final Design



Technica

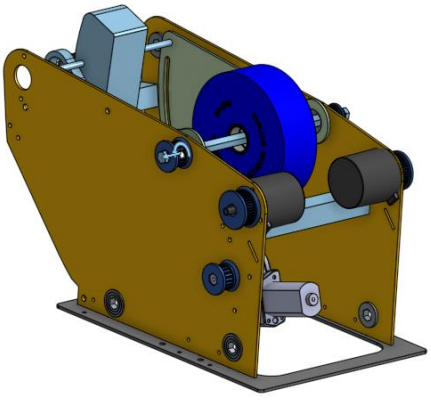
Final design involves the same poly belt system but rather than making it out of ¼” Lexan, 1/16” Lexan was being used instead to save

Design - Flywheel Shooter

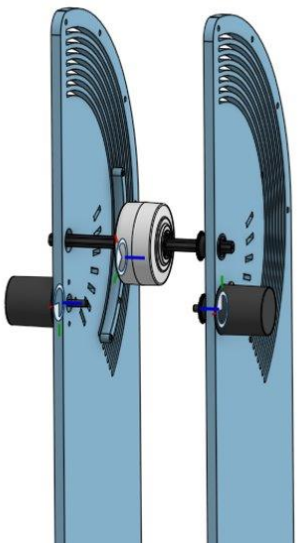
- **Design List**
  - **Accurately score from:**
    - **Initiation Line into Inner Goal**
    - **Far Side of the Trench into the Outer Goal**
    - **Scoring Zone into the Outer Goal**
    - **Varying distances by using vision integration**
  - **Smallest Possible Footprint to Avoid Integration Issues**
  - **Light Weight Design to keep our Center of Gravity Low**
- **Implementation**

- Uses a rack and pinion style hood to change the angle of our different shots with a 20.2:1 ratio to enable fine-tuned angles powered by a Snowblower motor to prevent unwanted changes in angle
- Uses two NEOs with a 1:1.7 pulley ratio to power our flywheel
- 6” Colson wheel as the flywheel
- Implements a varying compression arc to avoid power loss but still gain extra power at the end of the ball path

Design Progression



First created prototype made to allow us to test different compressions as well as different exit angles.Final design which allows us to utilize an adjustable hood and maintain consistent shots



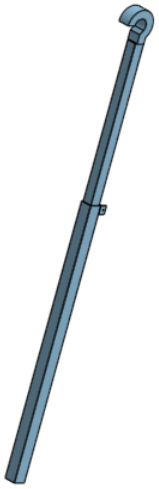
Design - Climber

- **Design Goals**
  - Be as small, compact, and light as possible
  - Be able to adjust our position on the switch after we climb.
  - Climb as quickly as possible to maximize cycles for the shooter
- **Implementation**
  - Uses two constant force springs rated for about 6 pounds of force each, allowing the elevator to reach the up position in the up position.
  - Uses a winch powered by a Mini CIM with a 36:1 CIM sport gearbox to lift the robot.
  - The bar traversal mechanism and the hook are easily interchangeable depending on the capabilities and needs of our alliance partners.

- Uses a Neo 550 with a 25:1 Versa Planetary gearbox to power a wheel that supports the robot on the switch so the robot's position can be adjusted.
- Uses a ratcheting system to prevent the winch from back-driving once the robot is powered off ensuring that the robot stays suspended from the switch after the match.
- Uses a servo with a cam attached that can disengage the ratcheting leaver if it becomes necessary to reverse the climber.

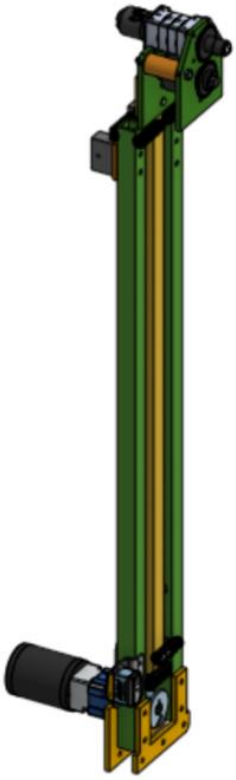
Design Progression

"Lightsaber Climb"



This was one of our earlier ideas for a climbing mechanism. It has a smaller footprint than our final one, but it is more difficult to build and integrate.

"Elevator Climb"



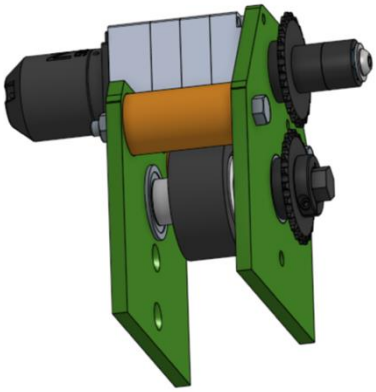
This is our final elevator design. It uses constant force springs to power the elevator on its way up, and a winch to pull the robot up, or the elevator back down if necessary. Some of the interesting design features include the bar traverser, the hook, and the ratcheting system.

"Buddy Climb"

We spent some time trying to develop a mechanism that would lift our alliance's robots in addition to our own. However, we decided that many robots would be able to climb on their own, and that due to the 12" horizontal extension limit we would be limiting our height and possibly our capabilities by attempting to be small enough to lift other robots. Although it didn't end up being used, we learned a lot from attempting this, including integration limitations and possible strategies for approaching this game.

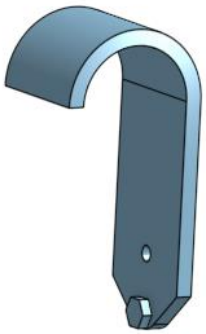
Key Parts of Final Design:

Bar Traversal

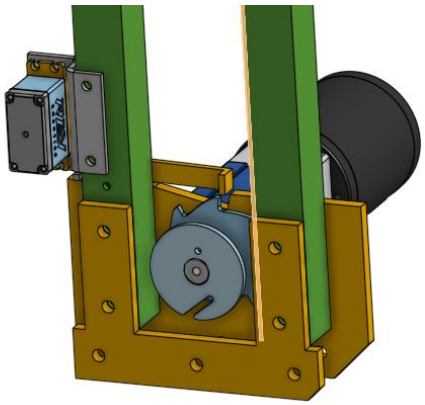


This allows our robot to move along the generator switch and help our alliance balance the switch for more points. It is easily interchangeable with a hook if we decide that it isn't necessary for a match. It consists of side plates and custom-made standoffs, and is driven by a NEO 550 brushless motor with a two stage 25:1 VersaPlanetary gearbox.

The hook is the other possible attachment to the elevator, making it versatile in terms of its capabilities. The hook is bent out of ¼" aluminum and is useful when we feel that bar traversal isn't necessary for a match.



Ratcheting System

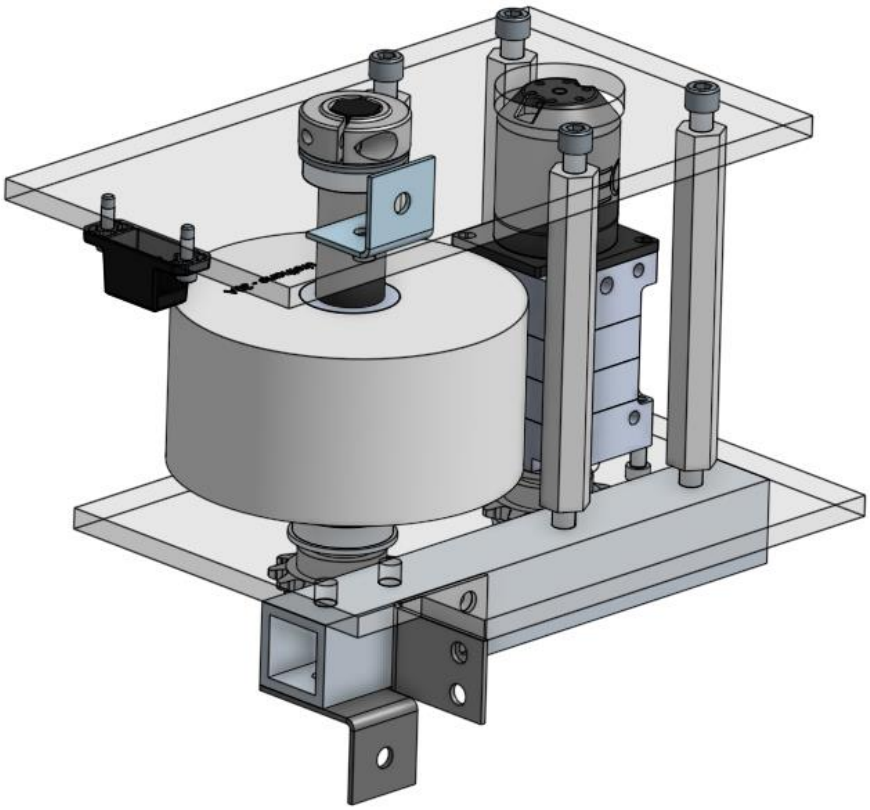


After our prototyping, we realized that our winch motor back driving was a completely feasible issue, which would cause the robot to fall back down from the generator switch. It consists of a machined arm, a servo, and an attachment to the winch. The servo can engage the arm with the winch, thereby only allowing the winch motor to spin in one direction when it is engaged.

Design – Control Panel

- **Design Goals**
  - Be able to withstand the full force of the robot
  - Be able to use a sensor to detect current color
  - Be able to spin the control panel at a safe speed
  - Be able to easily mount with other subsystems
- **Implementation**
  - Uses a top ¼" Lexan plate to provide support and mounting for color sensor
  - Uses a Neo 550 and a REV Color Sensor to accurately spin the wheel according to its current color
  - Uses a 20:1 gear ratio in order to spin the Control Panel at a high but safe speed while providing the most traction
  - Uses a Durasoft wheel to have enough tolerance to mitigate the damage that might be caused by driving into the control panel at full speed
  - Uses a motor inside the assembly to save space and easier mounting

- **Progression**



Electrical

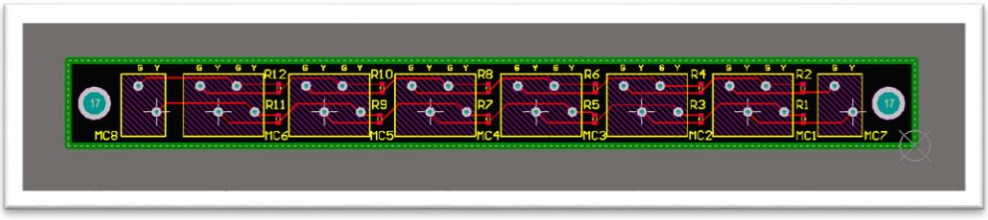
• Part Selection

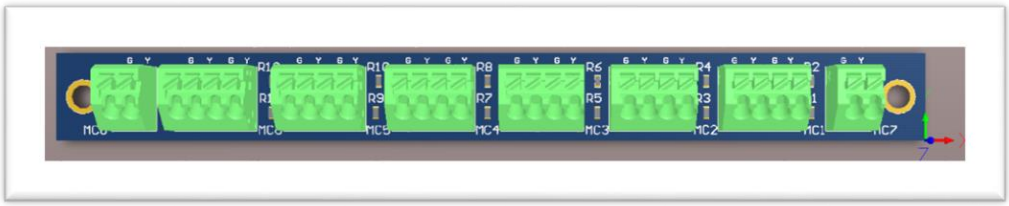
This year, none of our subsystems required any pneumatics, so we were able to use mainly standard FRC components from the kit of parts.

However, to pick motor controllers, we had to accommodate for the other subsystems. For example, many of our motors were REV Robotics' NEOs or NEO 550s with internal encoders, so we had to use Spark MAXes. In the end, we had to use eleven Spark MAXes, two Victor SPXes, and two Talon SRXes.

We also designed a new printed circuit board meant for CAN with improvements from last year's, where resistors have been added to help ease troubleshooting and continue to clean up CAN wiring. We plan to have this implemented by Week 5, Pasadena.

• CAN Bus





The CAN Bus Connector PCB is intended for easy and cleaner wiring as it centralizes the location, which we used as the Power Distribution Panel this year by 3D printing amount that rests on top, as opposed to having wires be next to motor controllers. The PCB is only meant to connect the motor controllers' CAN, as it has sixteen slots per side. We used DigiKey terminal blocks to help secure the CAN, and the bottoms of the blocks connect to the traces on the PCB which then connect to each other, forming a complete circuit.

A difference that we implemented this year is the use of resistors. We identified a problem last year that if we weren't to use all sixteen PDP slots, the PCB could be rendered useless. By placing resistors in between terminal blocks, we can resolve the problem in addition to an added benefit of being able to troubleshoot mistakes more effectively.

- **Integration**

Integration of electronics this year was challenging in the aspect that we had limited, far apart space. We had a couple of options to begin with, including under the ball hopper or on the ball tower. After discussion and tests, we decided to mount our board of motor controllers and the PDP on a vertical post. This gave the benefit of a lesser chance of a fuel cell accidentally hitting the motor controllers and PDP versus having it face outwards on the ball tower. We mounted our other vital electronics on a board on the ball tower for easy wiring to the PDP.

Programming

• Peariscope

This year, instead of using a Limelight or another "black box" vision system like PixyCam, we decided to make our own vision system, called Peariscope. The vision system consists of a Raspberry Pi Camera 5MP (Gen 1), a Raspberry Pi 3B+, and some green 3W star PCB LEDs. We are running our own software on the WPILib-provided Raspberry Pi image. In order to view the stream and post data computed from the video, we use a combination of NetworkTables and CameraServer, both of which are part of WPILib.

We decided to use Python to program our Peariscope. The choice of this language is due to the ease of development and speed. Even though CPython is very slow compared to C++, OpenCV and NumPy are C++ and C bindings, respectively, which allows for near-C/C++ level performance in Python. Our program consists of retroreflector detection, contour filtering, and morphological operations, and it outputs the properties, such as the size and position, of the retroreflectors.

The Raspberry Pi 3B+, with more performance of the Pi 3+ Compute Module used in the Limelight, allows us to process higher-resolution footage at an acceptable framerate. The green 3W star LEDs are run at around 2/3 max power so that they aren't too bright, but still bright enough to allow us to detect the target across the field. We selected the Raspberry Pi Camera Gen 1 because it was relatively affordable while being easy to use with the Pi, without being as overkill as the Gen 2 camera is.

- ## Control Loops

For the flywheel shooter, we are using a feedfoward-only loop for control. We discovered for just shooting from a couple of preset positions, there was no real need for full PIDFF control. The flywheel hood is controlled by a bang-bang controller for angle control.

All the other subsystems do not require fine control, so we used basic percent output control.

Summary

Autonomous

- score 3 starting balls into the outer port with some shots scoring into the inner port and move off of initiation line
- can take in balls from other teams and score them as well

Tele-Op

- can score into the outer goal from anywhere between the scoring zone to the far trench
- can climb and self balance as well as traverse across the bar to balance with a teammate
- can collect balls from the loading zone and off the floor

Drive Capabilities

- 115 pounds
- 8 pneumatic wheel drive, high traction
- Max speed = 13.4 ft/s