NULL ROBOTICS

FTC 8103

2022-23 Engineering Portfolio

Our Team

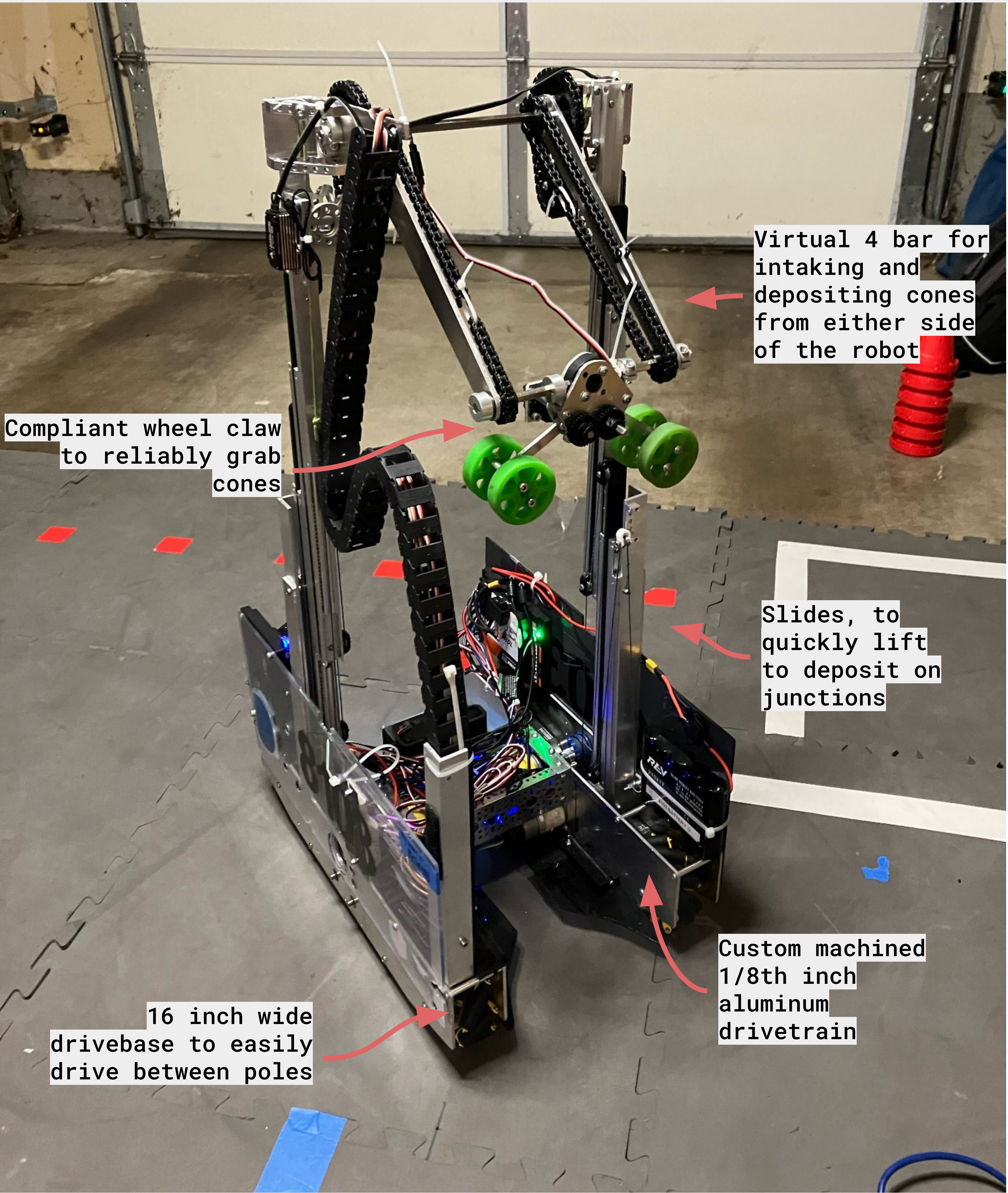
We are team 8103 Null Robotics, the varsity robotics team from Eastside Preparatory school.

Our overall team goal is to create a program where students can be inspired to learn more about science and technology. We aim to cultivate an environment where more experienced team members mentor newer members through engineering projects so that they can develop the skills necessary to contribute to the team. Furthermore, regardless of skill level, we continuously push the boundaries of what our robot is capable of and create new challenges for every team member.

Subteams

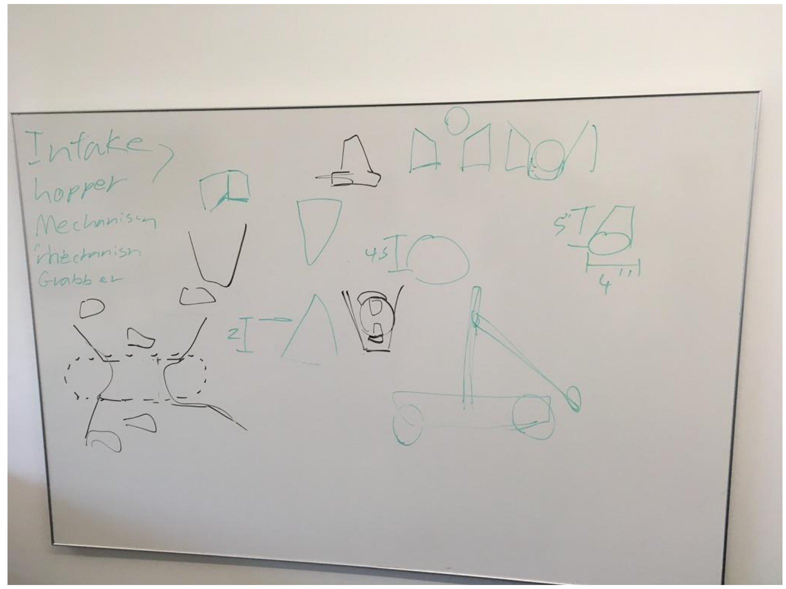
DESIGN / HARDWARE The hardware team designs and fabricates physical parts for our robot. Our goals are to make simple, reliable, high scoring designs that put minimum load on drivers and programmers.	SOFTWARE The software team works in tandem with the hardware team to ensure our robot can complete all game objectives efficiently.
OUTREACH The outreach team ensures we are assisting our 2 sister teams: 15203 Undefined Robotics and 9884 Void Robotics. It also makes sure we are engaging with other FTC teams.	DOCUMENTATION The documentation team makes sure we are following the engineering design process. It is also responsible for ensuring we test our designs and use the feedback to improve

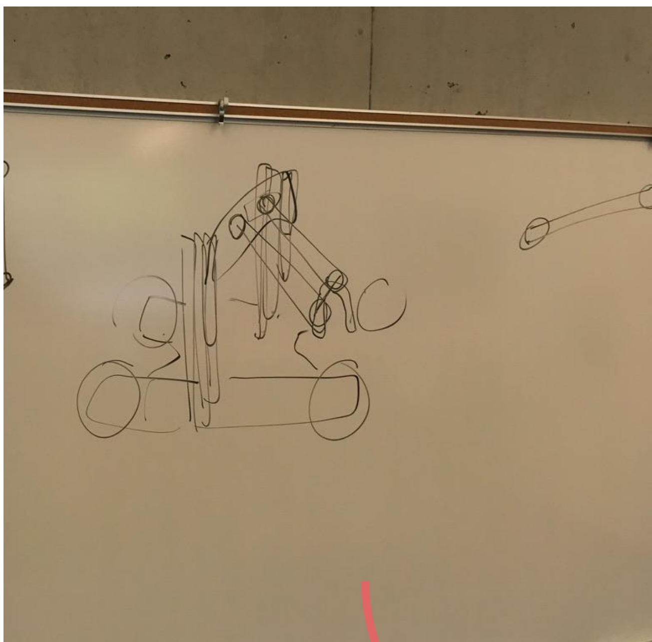
Our Robot


Virtual 4 bar for intaking and depositing cones from either side of the robot
Compliant wheel claw to reliably grab cones
Slides, to quickly lift to deposit on junctions
Custom machined 1/8th inch aluminum drivetrain
16 inch wide drivebase to easily drive between poles

Brainstorming and CAD

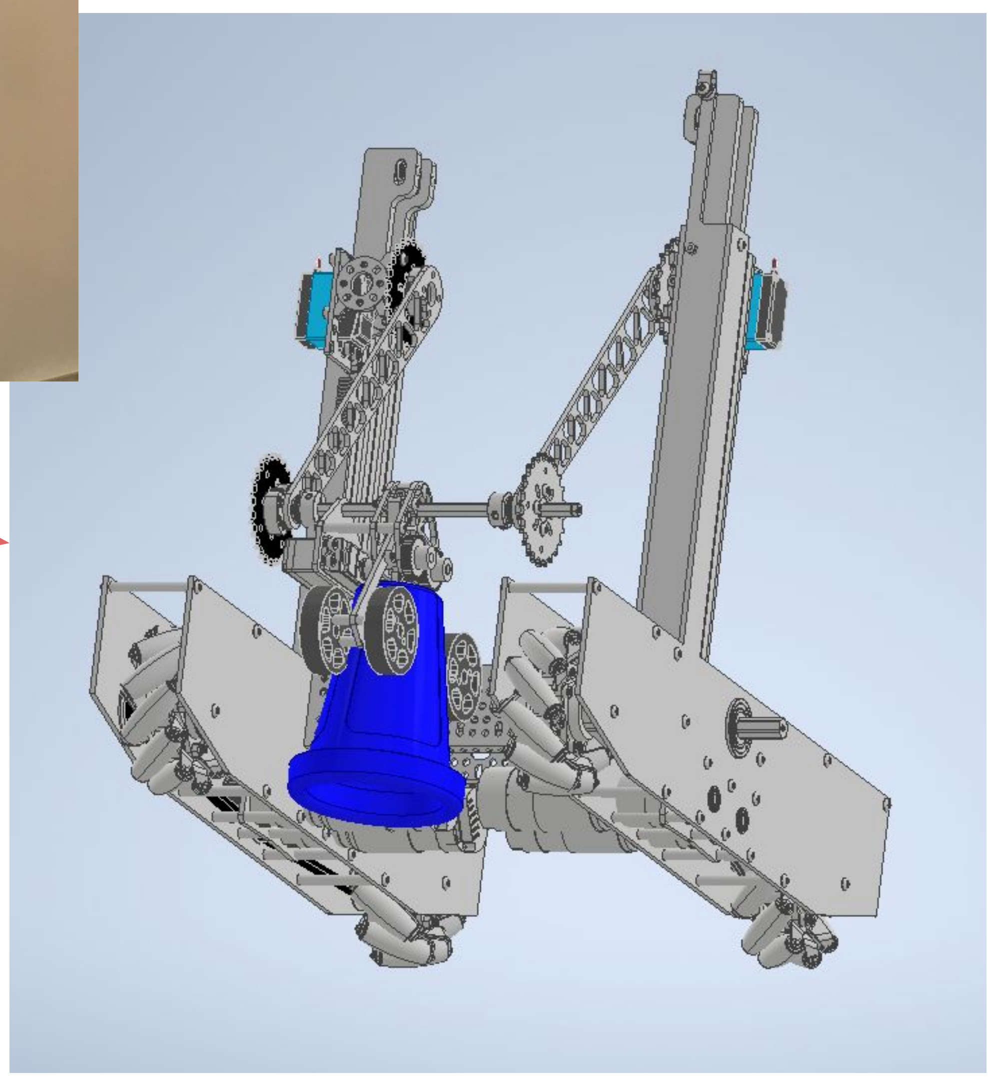
After first watching the game reveal, we discussed several possible designs.
We eventually decided on our core design of a claw on a virtual four bar on slides. This ensures we can intake and deposit on both sides of the robot. We discussed possible claw designs and ways to align cones.





We sketched our design, and translated it into an Autodesk Inventor CAD model.

Our design team maintains this CAD model, keeping it up-to-date with any design changes to the robot. The use of CAD allows us to produce custom parts and to make sure our designs will work before building them

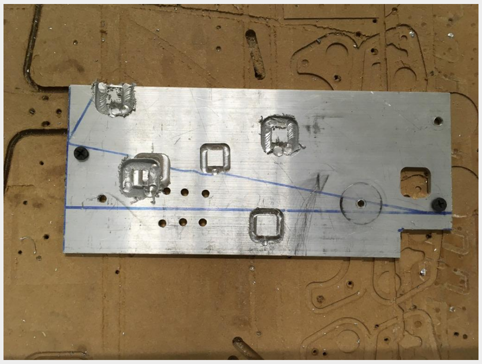


Fabrication - Aluminium

The core structure of our robot is made out of aluminum, which gives us sufficient strength while being relatively light.

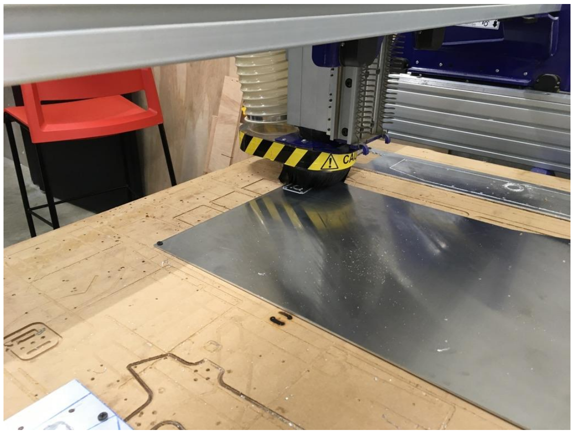
This, year, we were able to learn from our mentors how to use our school's ShopBot cnc. While it is not initially meant for aluminum, we were able to tune feeds and speeds that cut aluminum reliably, but slowly.

This was a valuable learning experience for our members, and we made sure to pass it on to new students so our team can continue to use this capability in the future.

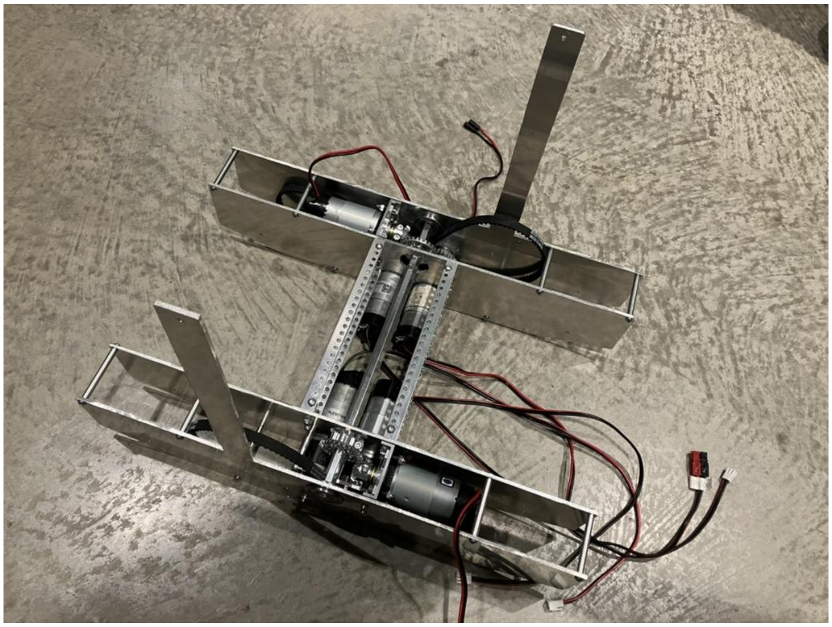


A test piece we used to tune feeds and speeds. As you can see, it took us several tries to find settings that worked well

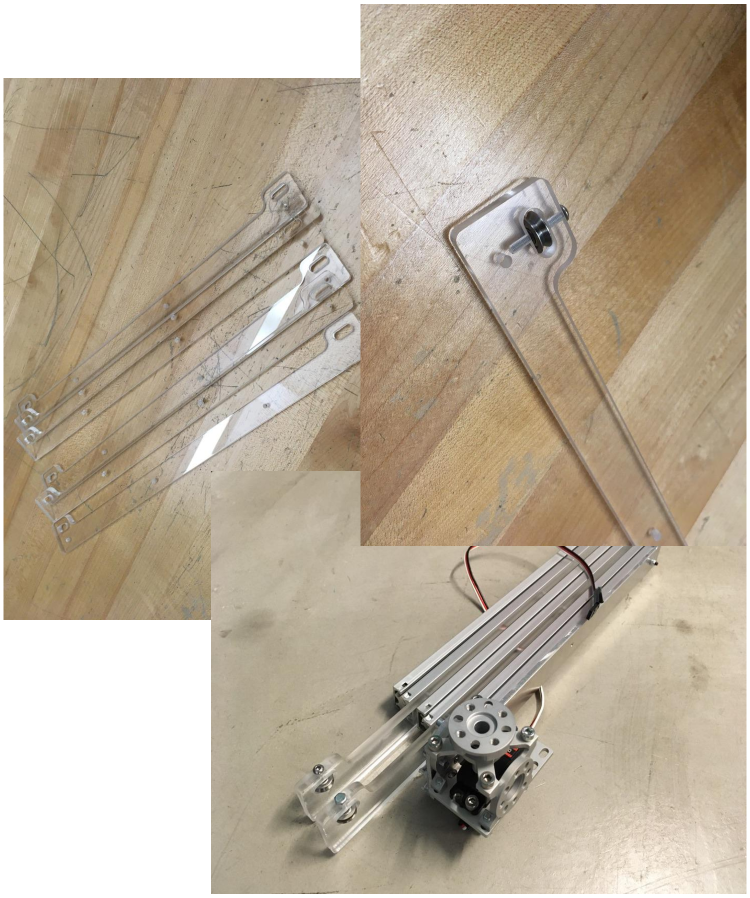
Example



The ShopBot allowed us to cut out several parts, including our main drivetrain plates, as shown here.



Fabrication - Plastic

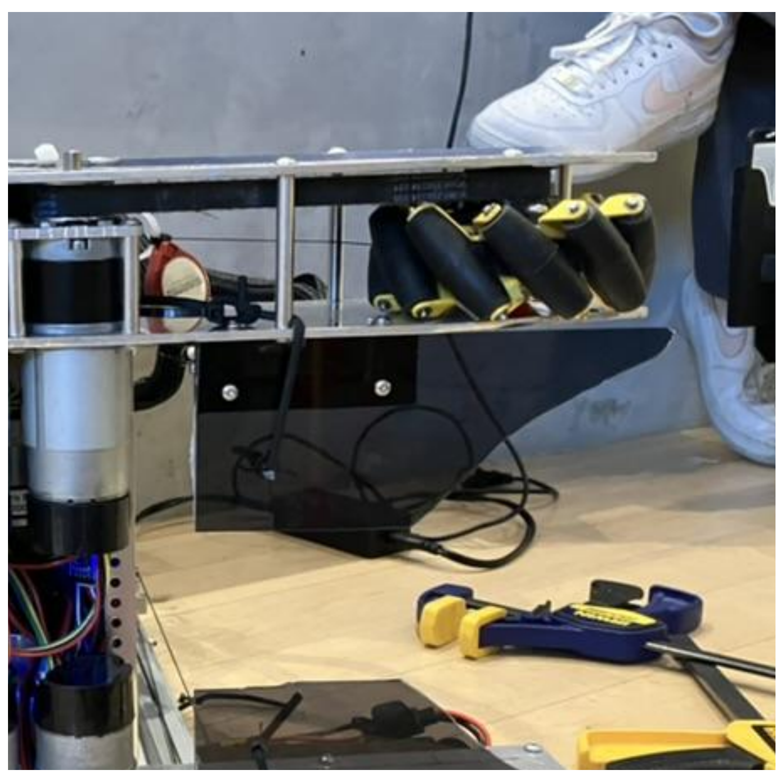


Slide Inserts

We used laser-cut quarter inch acrylic for the stages of our linear slides. The nice thing about this material is that it is easy to drill and tap, allowing us to use m3 screws to retain the bearings for our slide string. We eventually replaced these with delrin, as the acrylic was cracking

Cone Guides

We used polycarbonate for our cone guides, in order to keep them lightweight but sturdy. For the hinges of our cone guides, we used delrin, which allows them to smoothly pivot with low friction.



Camera Case

We used 3d printed PETG and laser-cut acrylic for our camera case.

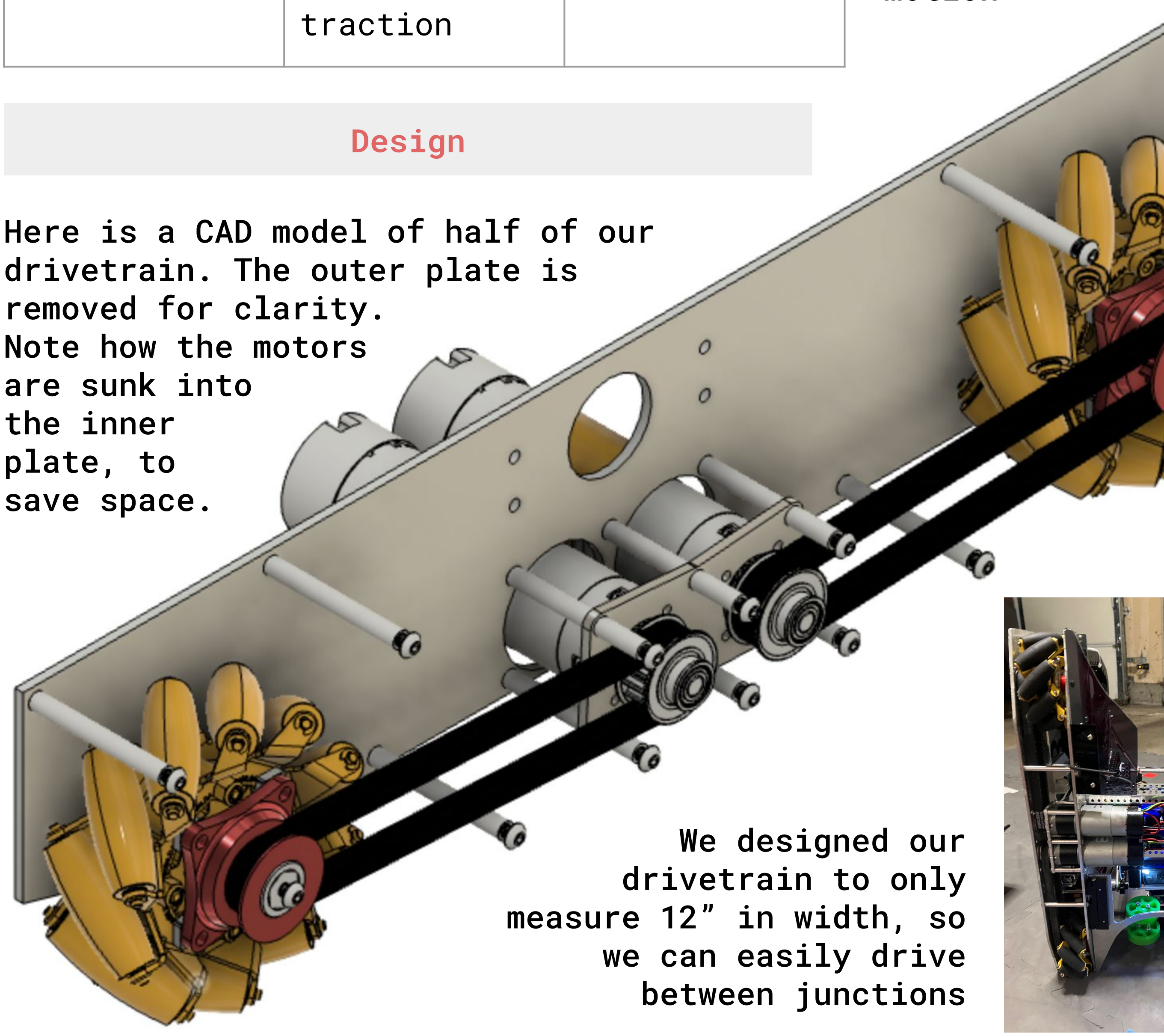
Drivetrain

Drivetrain	Pros	Cons
6 wheel drive	Good traction and acceleration	Cannot move in any direction
Mecanum	Can move in any direction	Poor Acceleration and traction
Swerve	Can move in any direction, with full traction	Very complicated to implement

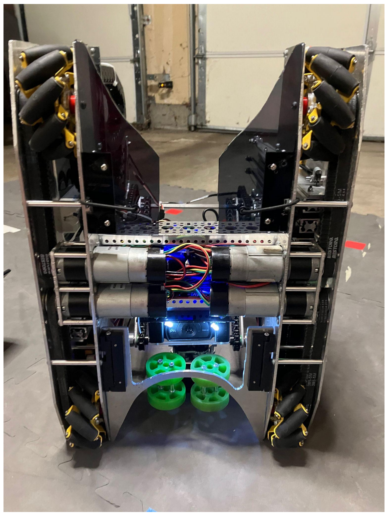
We discussed several possible drivetrains for this year's game. We knew we wanted a maneuverable drivetrain to navigate junctions. After considering the pros and cons of each, we decided on a mecanum drivetrain as it provided us a simple way to get holonomic motion

Design

Here is a CAD model of half of our drivetrain. The outer plate is removed for clarity. Note how the motors are sunk into the inner plate, to save space.

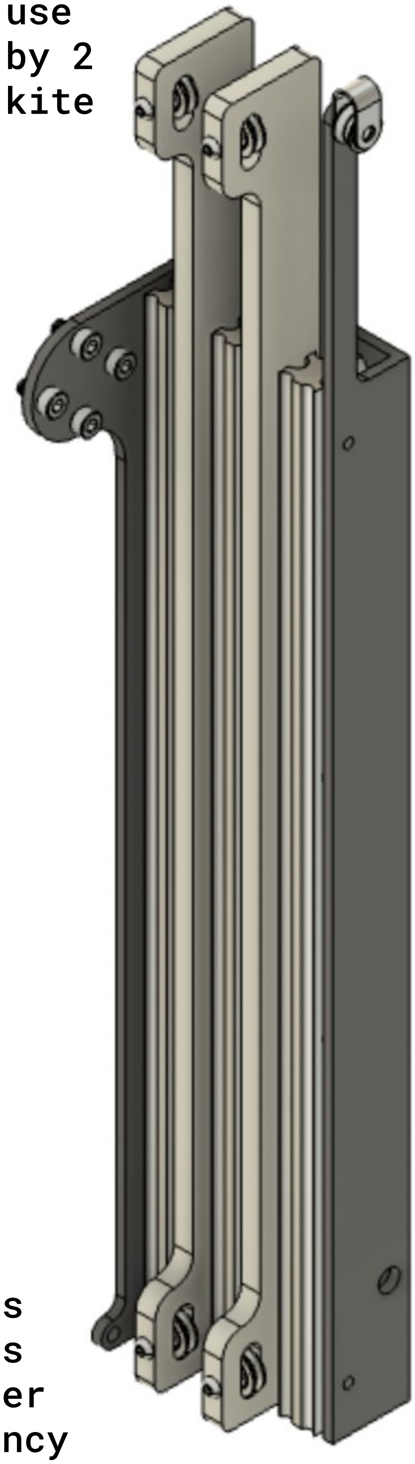


We designed our drivetrain to only measure 12" in width, so we can easily drive between junctions



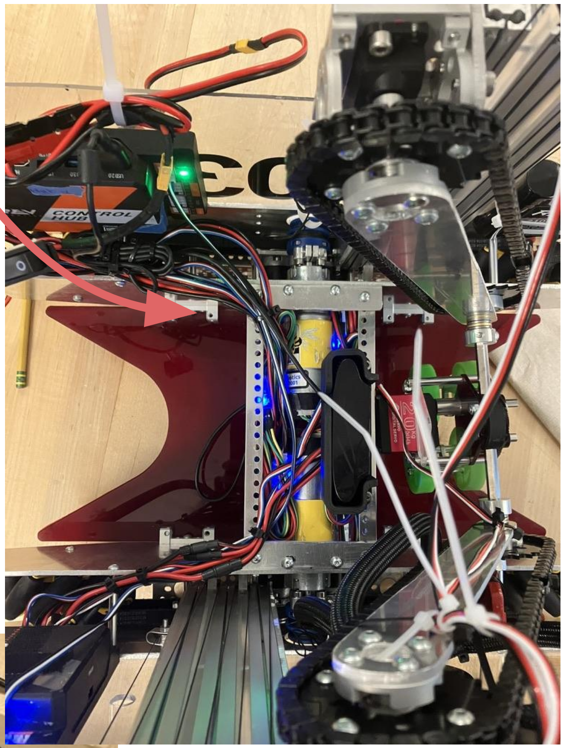
Lift

Our lift allows us to quickly and reliably raise cones to the low, medium, and high junctions. We use 2 sets of cascading Misumi linear slides, driven by 2 motors. We drive them with 3d printed spools and kite string.

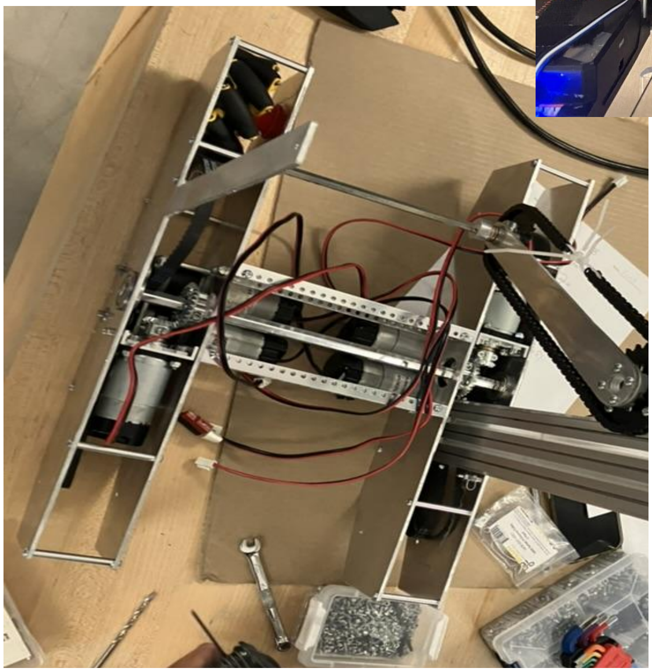


Improvements

In our first design, we used 2 1:1 motors and 2 sets of bevel gears to drive our string spools

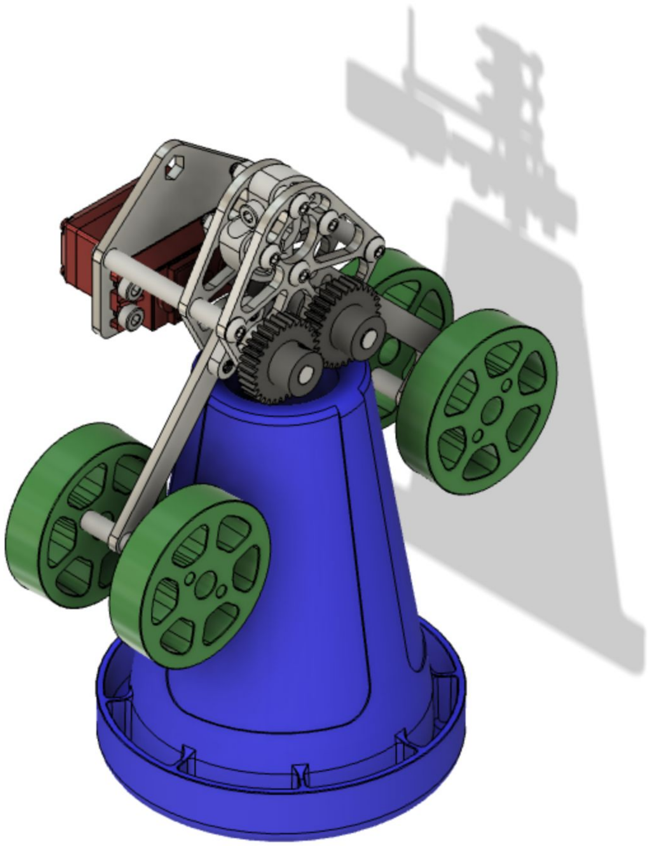


However, these performed very poorly, often lacking the power needed to raise the arm. We underestimated the efficiency losses of bevel gears at high speeds.



Our new design uses 5.2:1 geared motors directly, for higher torque and efficiency

Claw

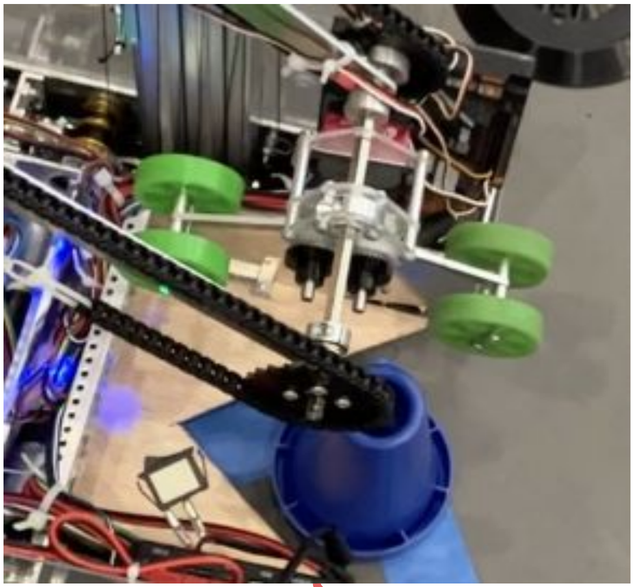


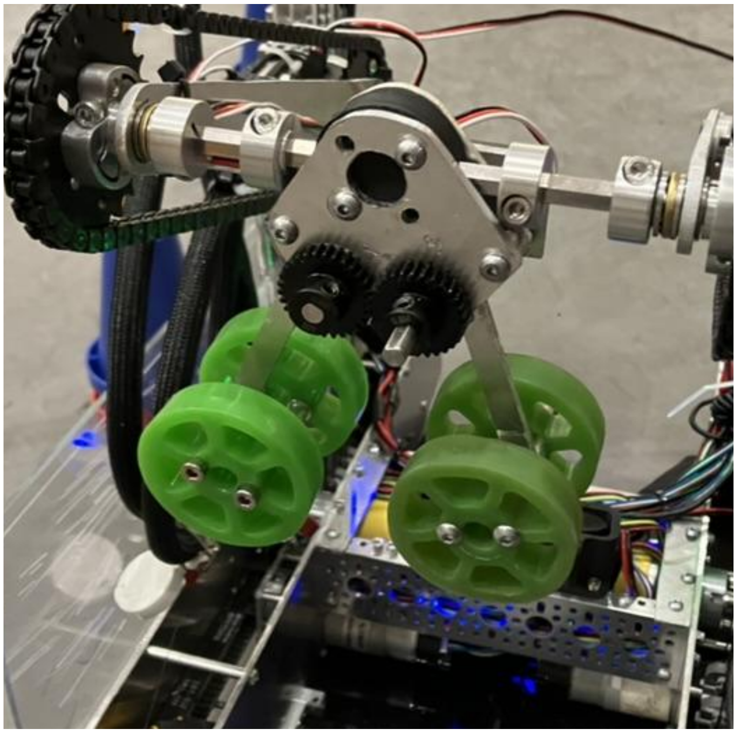
The goal of our claw is to reliably grab and release cones. To do this we use 2 arms with compliant wheels. The compliant wheels provide grip and allow the claw to work even if the cone is slightly misaligned. The claw is driven by a REV 20kg servo.

Our claw is constructed out of 1/8th inch aluminum plate and M4 standoffs. This is the same construction style we use for several parts of our robot, including the drivetrain

Improvements

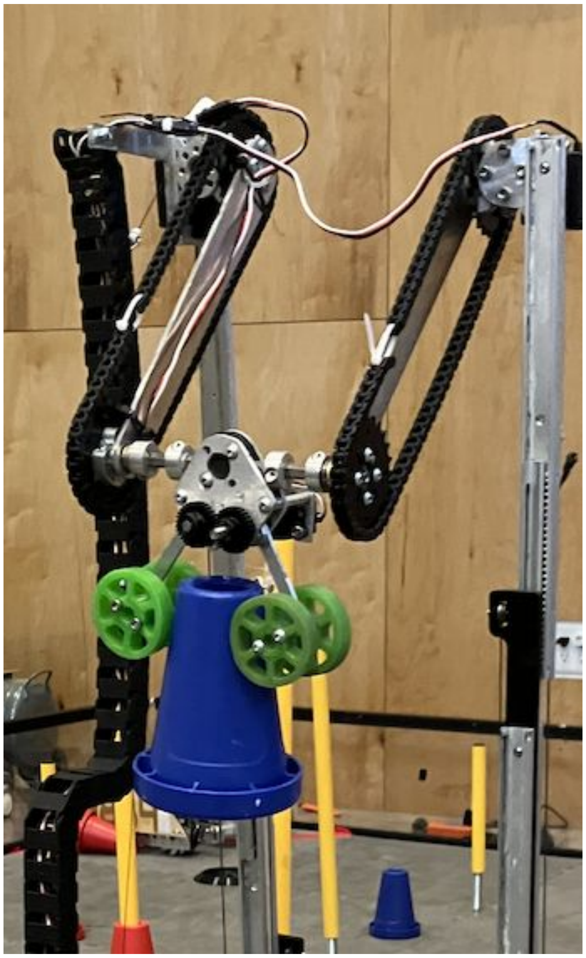
Our initial claw was made of acrylic and mounted in line with the robot. This alignment meant we could not easily grab cones near the wall.





We then rotated the claw 90 degrees as shown on the right. We also replaced the acrylic, which was cracking, with aluminum

Virtual 4 Bar



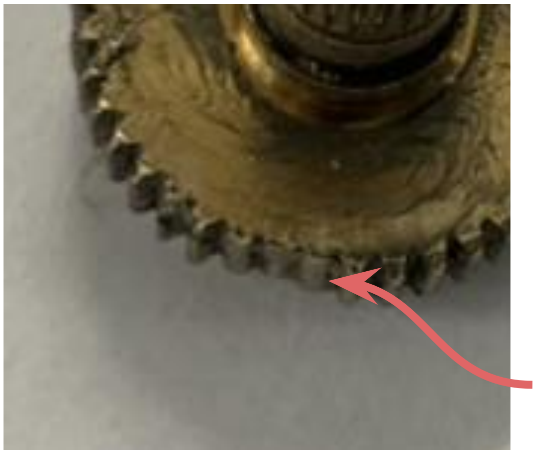
Our primary design goal for our arm is to be able to quickly reach to both sides of the robot for intaking and outtaking. Throughout this motion, we want our claw to remain level. To keep it level, we use a virtual 4 bar mechanism made with plastic chain

$$F = ma = \left(540g * 9.81\frac{m}{s^2}\right) = 5.297N$$

$$T = Fr = (5.297\text{N})(232 * 10^{-3}\text{m})$$

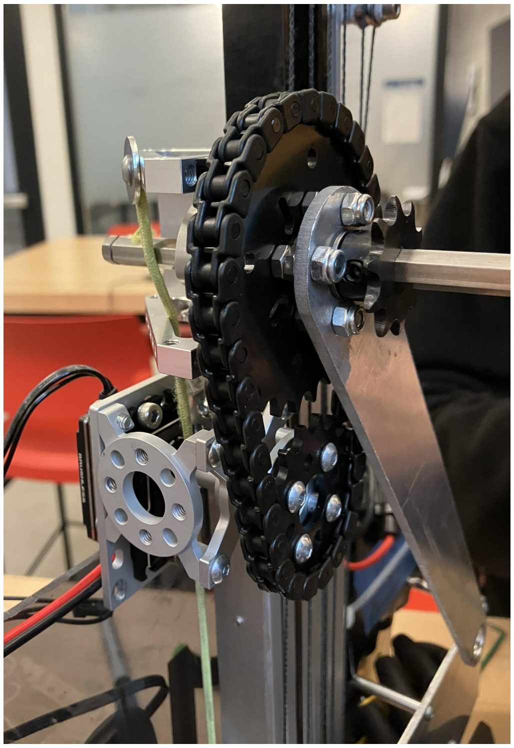
$$= 1.229Nm = 12.532kgcm$$

When designing our 4 bar, we calculated the torque needed to hold our claw and a cone against gravity. We chose 2 20 kg-cm servos as they give us a good safety margin above this torque.



Our first lift design, despite having sufficient torque, sometimes broke servos. Upon inspection, we found that is was the output gear of the servos breaking under shock load.

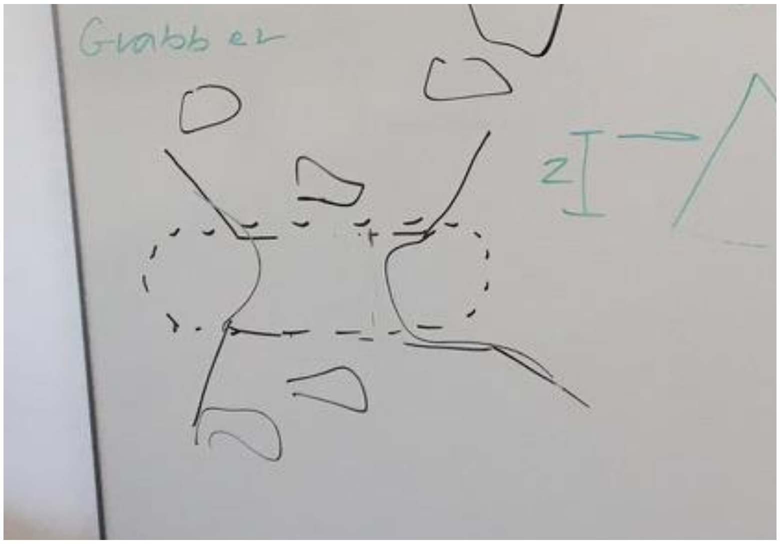
Improvements

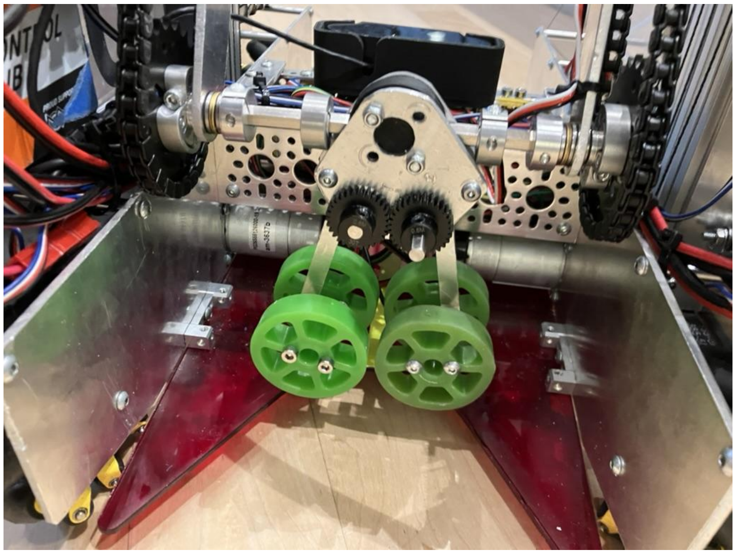


Our new design, shown on the right, isolates our servo from shock load using two sprockets and a length of plastic chain. The plastic chain's stretch limits the impact on the servo's output shaft.

As an added precaution, we replaced our brass-geared servos with titanium-geared ones

Cone Guides



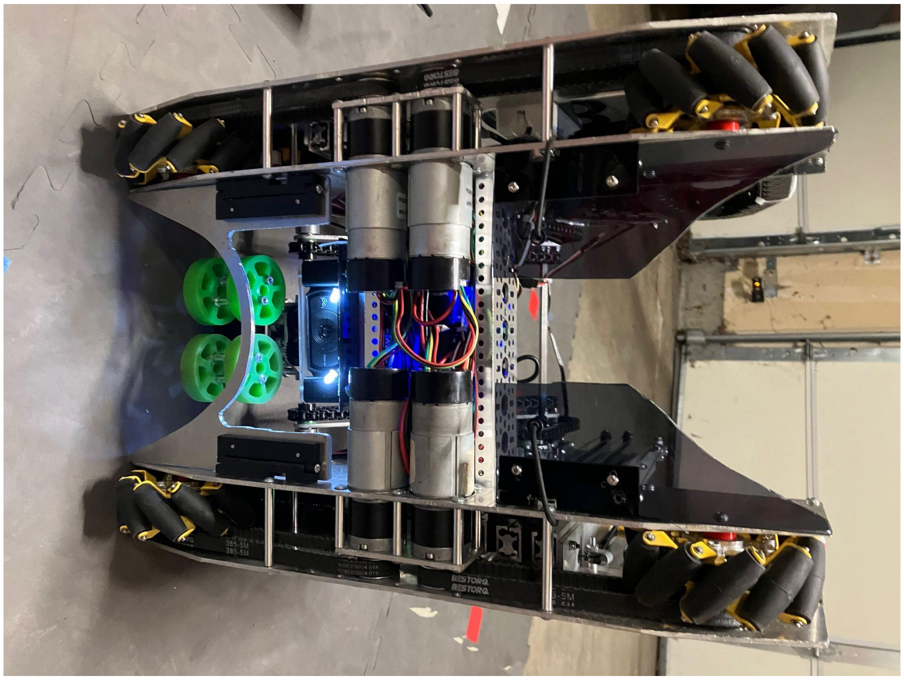


From our first design, we knew we wanted a way to quickly align cones so our claw can reach them

Our first design used a single acrylic plate on the base of our robot that would align with the lip of cones

After our first competition, we noticed problems with our design.

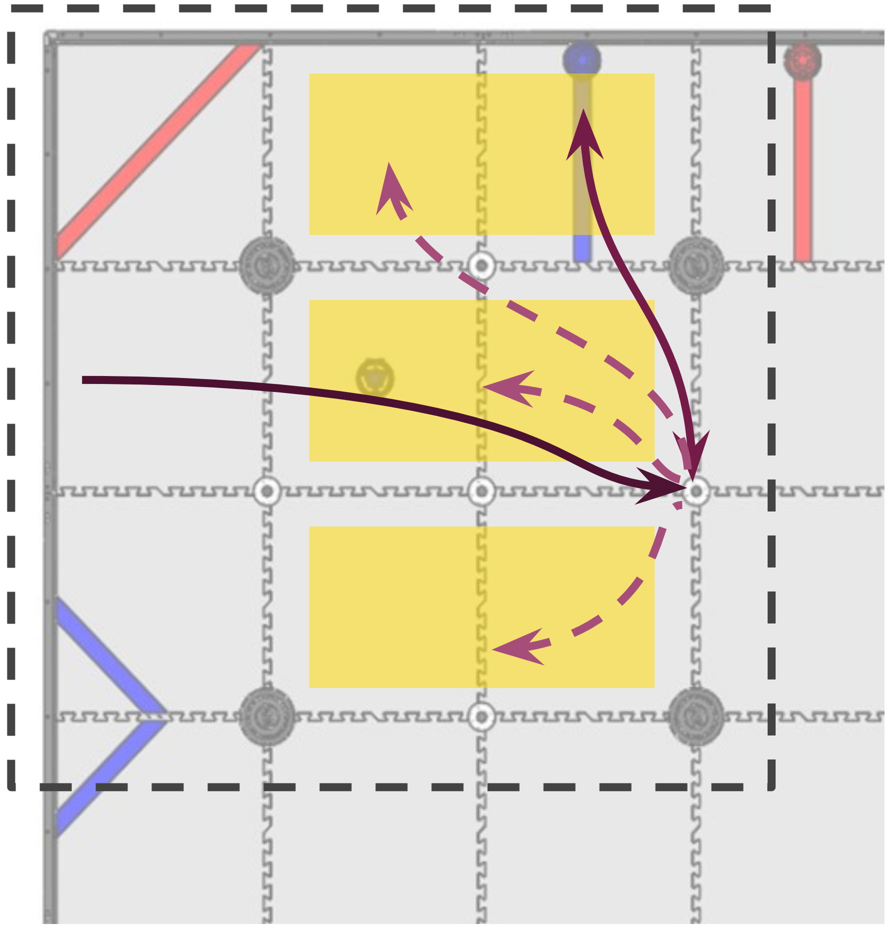
1. We could not drive over the low junction due to the low height of the plate
2. The Acrylic was weak and began to crack
3. Aligning with the lip of cones led to the potential to knock them over



Our new design uses 2 polycarbonate wedges, raised above the cone's lip. The wedges are hinged and sprung downward, so that we can trap the cone under the guides but they lift as we raise our lift. This new design allowed us to easily drive straight between the substation and the nearest high junction

Autonomous Strategy

Having a high-scoring, reliable autonomous is very important this year, to gain points for our alliance as well as to increase our tie-breaker score.



Our strategy in autonomous is to:

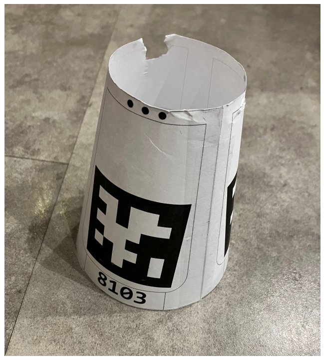
- **read our custom signal sleeve**
- **place our preloaded cone on a high junction** (6 points)
- **4 times:**
  - **grab cones from the nearest stack and place them on a high junction** (5 points x 4 = 20 points)
- **park in the designated signal zone** (20 points)

Our 5 scores cones are then double counted in autonomous for a total of 72 autonomous points

This autonomous stays fully on one side of the field so as not to interfere with our alliance partner

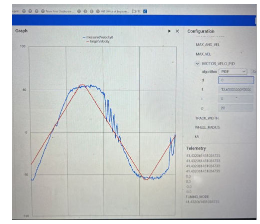
Autonomous Software

Signal Sleeve



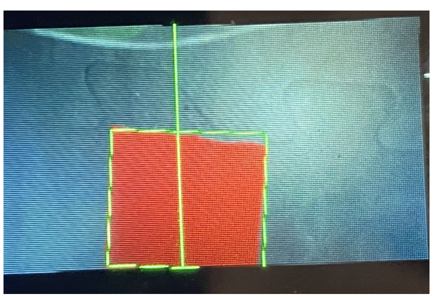
For our signal sleeve, we use aprilTags, a library that allows us to create “tags” that we can easily recognize with a webcam and assign unique numerical ids to.

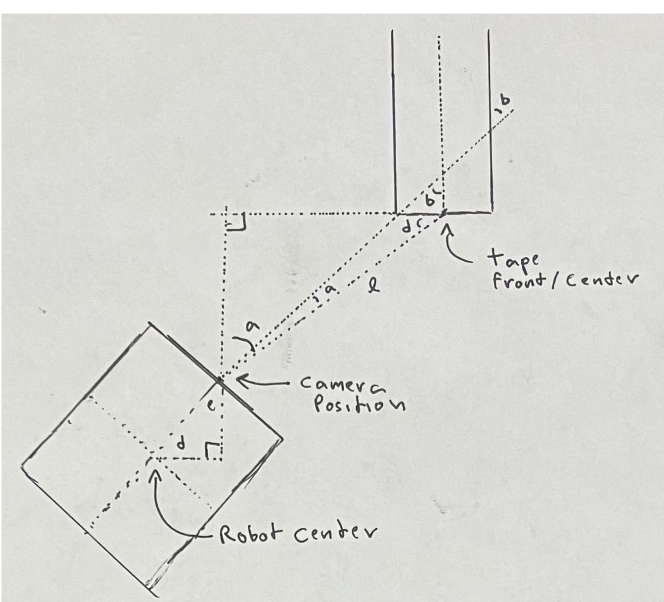
Robot Motion



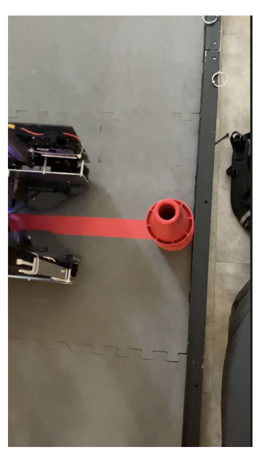
For motion during autonomous, we used the Roadrunner library. This library allows us to control the acceleration of our robot, so we minimise wheel slip. It also allows us to follow the complex robot paths seen on the previous page.

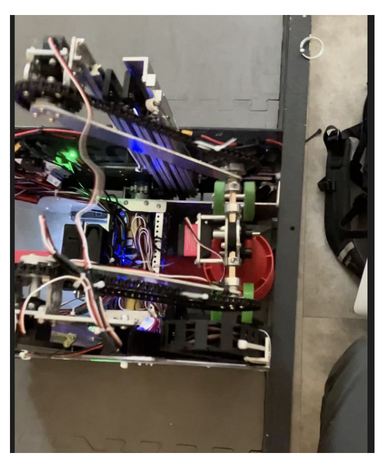
Tape relocalization





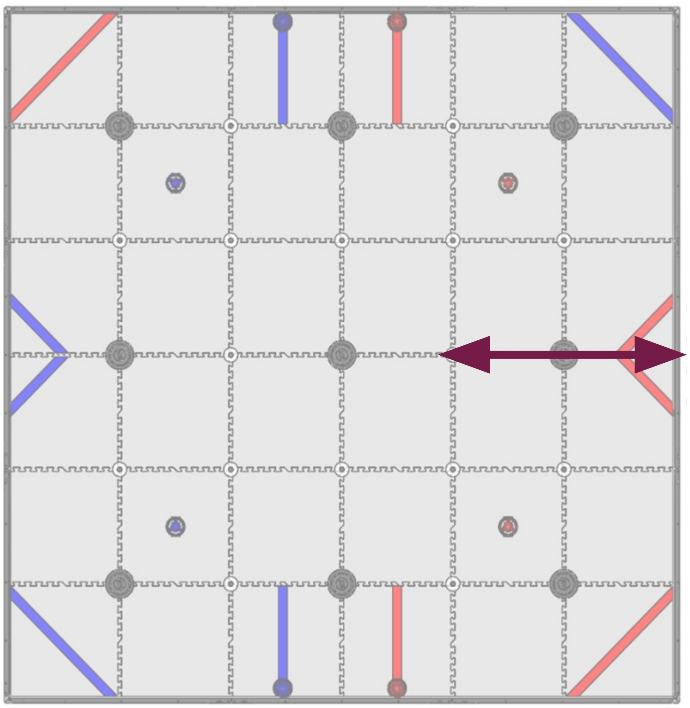
We use computer vision (opencv) to relocalize our autonomous on the tape each time we grab a cone. This ensures we maintain an accurate position estimate throughout the autonomous period and can consistently align with the junctions.



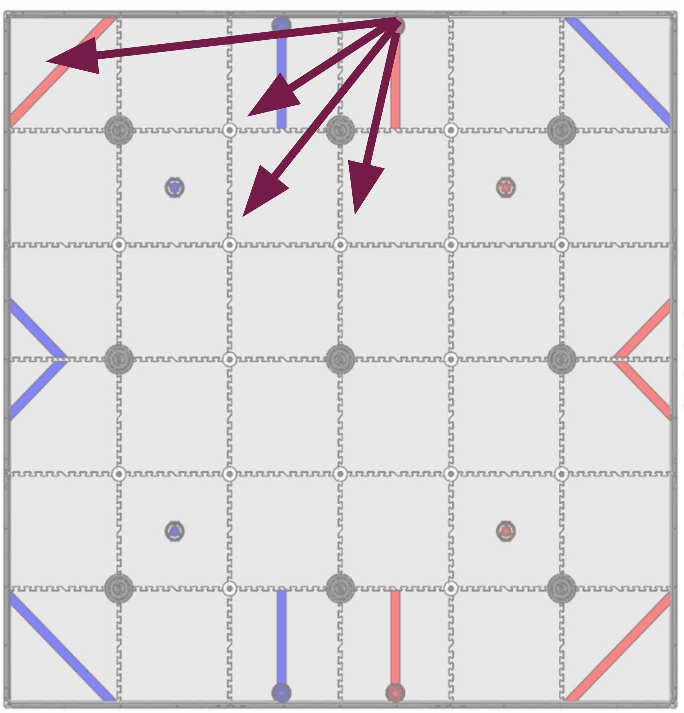


TeleOp Strategy

The main ways we gain points are from junction scoring, possession, and circuits. We have 2 main strategies we can run, depending on how our robot compares to our alliance partner's.



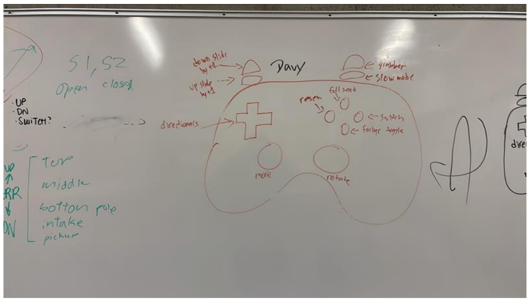
The simplest way is to grab cones from the human player in the substation and score them on a high goal. This gets us 5 points per cone scored. We will run this strategy when we are much faster than our alliance partner and want an fast way to get points.

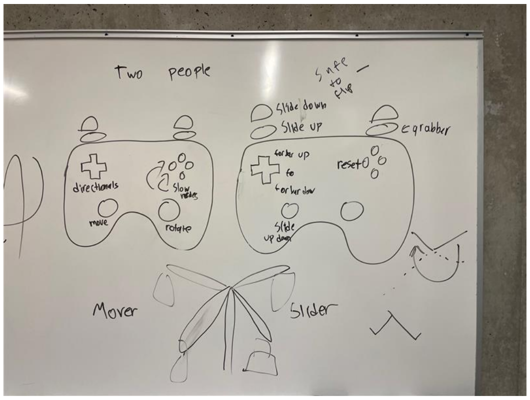


If we are working with our alliance partner to attempt a circuit, we can cycle cones from the stack to the back end of the circuit path. We would run this strategy when our alliance partner can cycle cones from the terminal faster than us and we want to support them by helping to get a circuit and possessing poles.

TeleOp Software

Controls





Our main goal TeleOp programming is to minimize the work for our drivers so they can focus on driving and strategy. We do this by automating as much as possible. To that end, we use a preprogrammed array of positions for the ground, low, medium, and high junctions. We also have preset positions for our four bar that we use for intaking and outtaking. Finally, we make use of macro buttons, like (Y) which brings the lift up and puts the fourbar into cone-drop position.

As shown on the left, we went through multiple possible control schemes to find what felt best for the drivers.

Improvements

After our first meet, we reviewed match videos and decided on a number of improvements to Teleop software that allowed us to score cones faster

Problem	Solution
Hard for driver to control slides, 4 bar, and claw.	2 - driver control removes strain from the main driver, allowing them to move with fewer pauses
Hard to place on ground junctions	Better pre-programmed heights
Longer cycle times because our robot was unable to drive over the ground junction	Placed our guides on hinges to allow them to swing upwards when driving over junctions