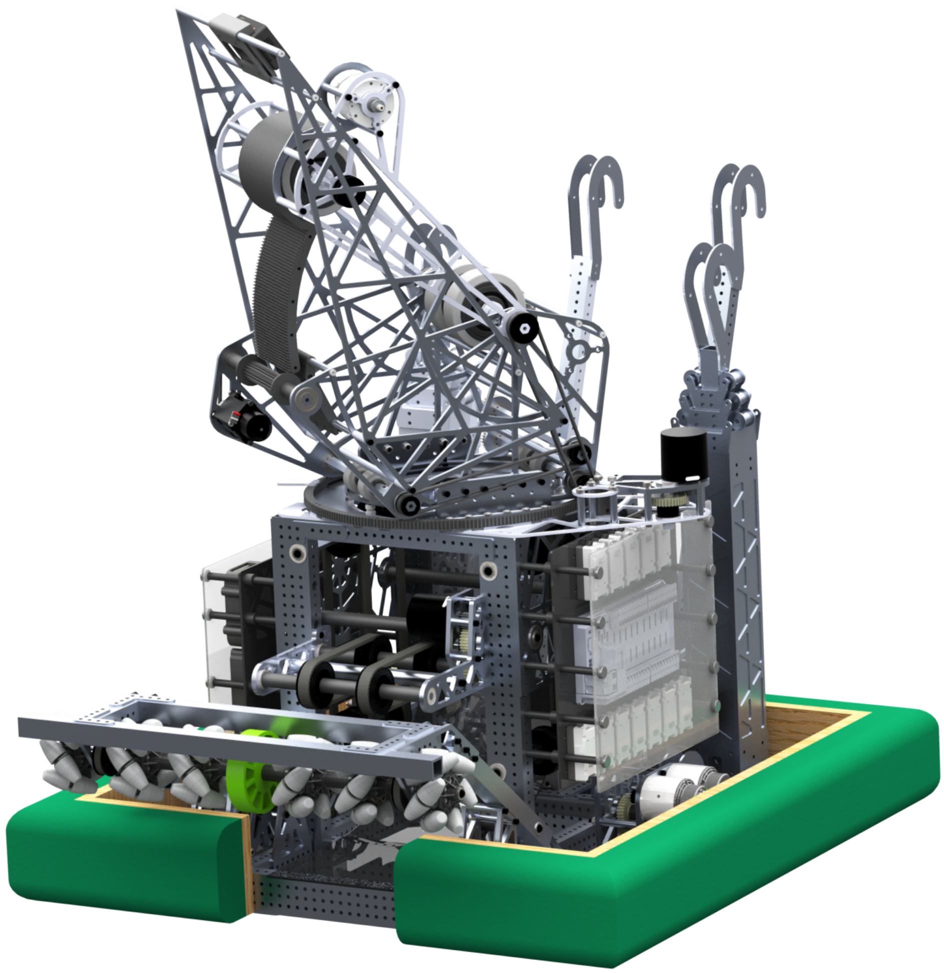
Paly Robotics

FRC 8 | 2022



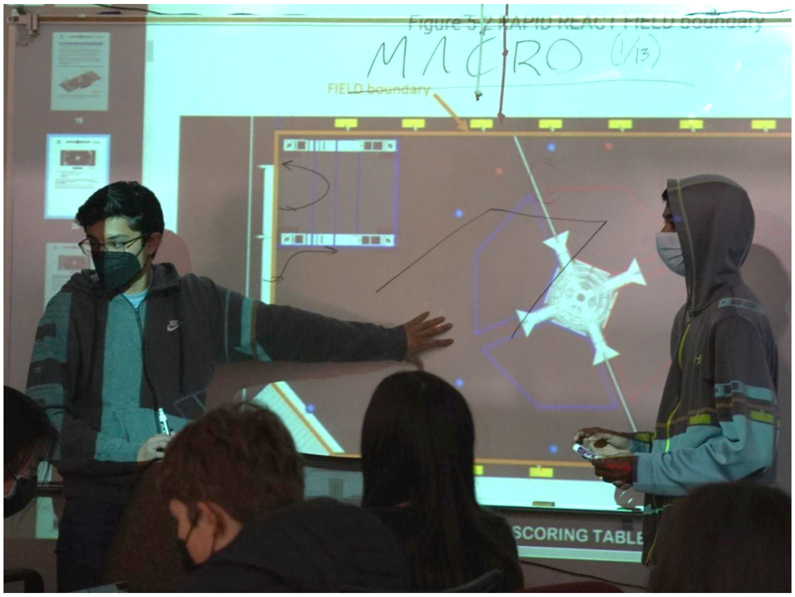
Alvaldi

Technical Documentation

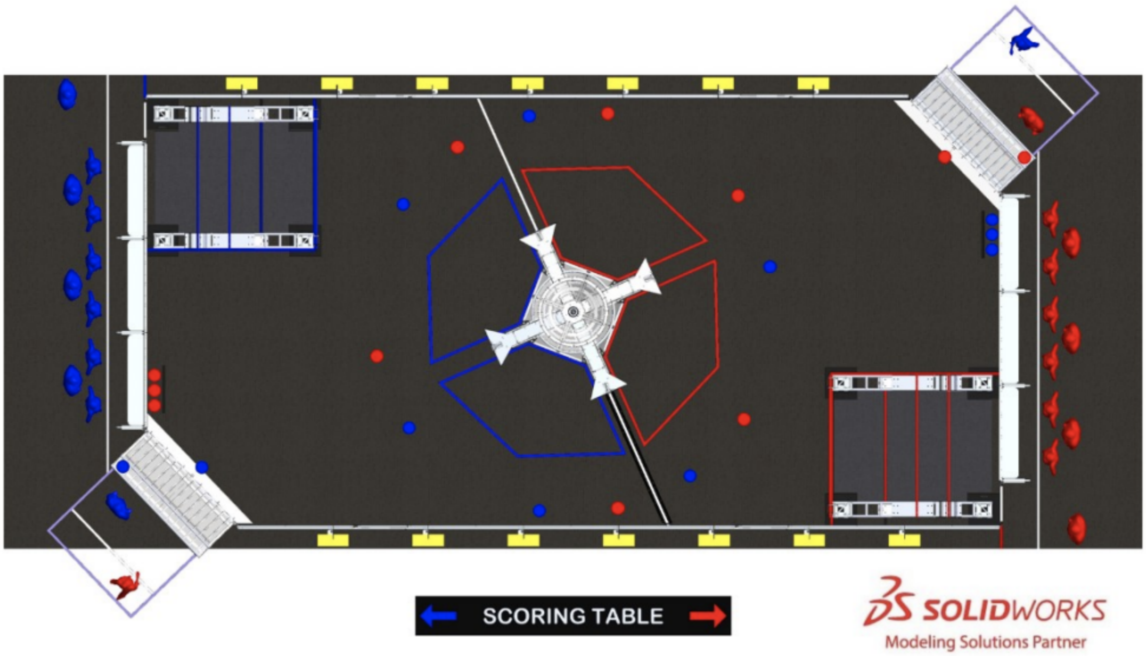
Rapid React

TABLE OF CONTENTS

STRATEGY



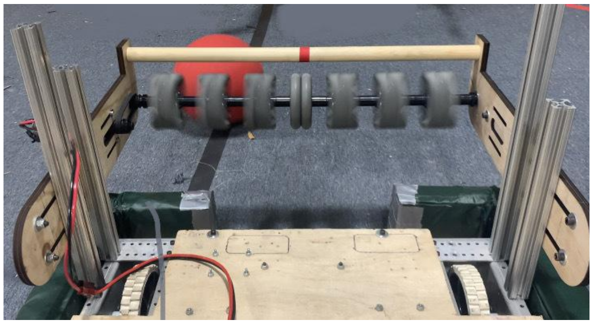
Rapid React



Goals

Needs	Wants	Wishes
• Robust Drivetrain • Score in upper hub from tarmac • Intake ball from ground • Intake with no jams • Mid-Rung Climb • 3 ball auto	• Score in upper hub from launchpad • High-Rung Climb • 4 ball auto	• Score upper hub while moving • Traversal-Rung Climb • 5 ball auto

Prototyping



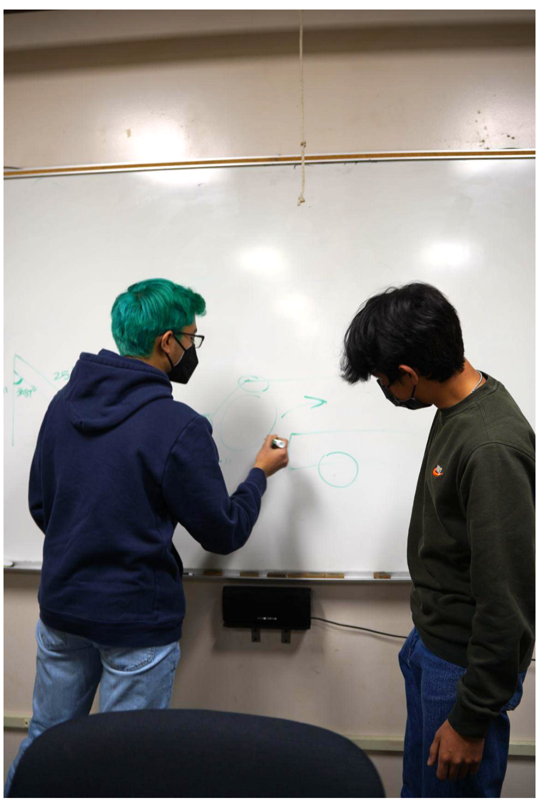
Intake: Tested two different configurations (split-bumper vs. over-the-bumper) and experimented with wheel type to find the best compression for intaking balls as well as to optimize center of gravity.

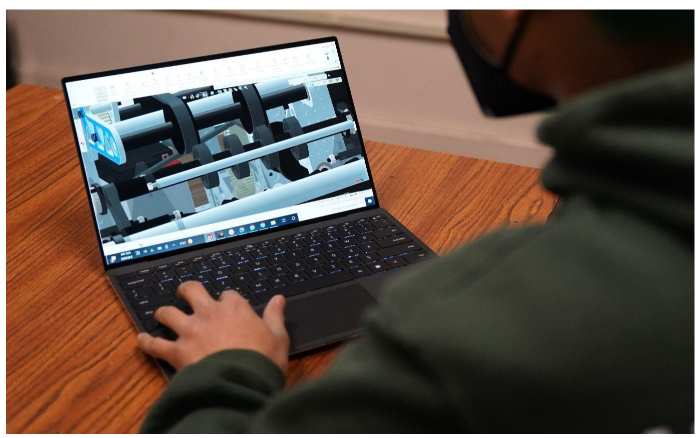
Single and Double Flywheel Shooter: Adjusted for ideal ball compression, shooter hood angle, ball spin and wheel type to identify peak velocity and shot arc.

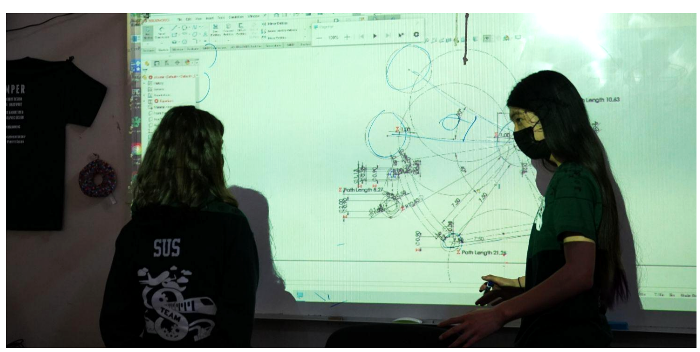
Hooks: Tested different active hook shapes using a combination of laser cut wood, 3D-printed gears, and bearings.

Indexer: Tested various materials (timing belts, omnis, polycord) to determine ideal compression, geometry, and to eliminate slippage while indexing balls.

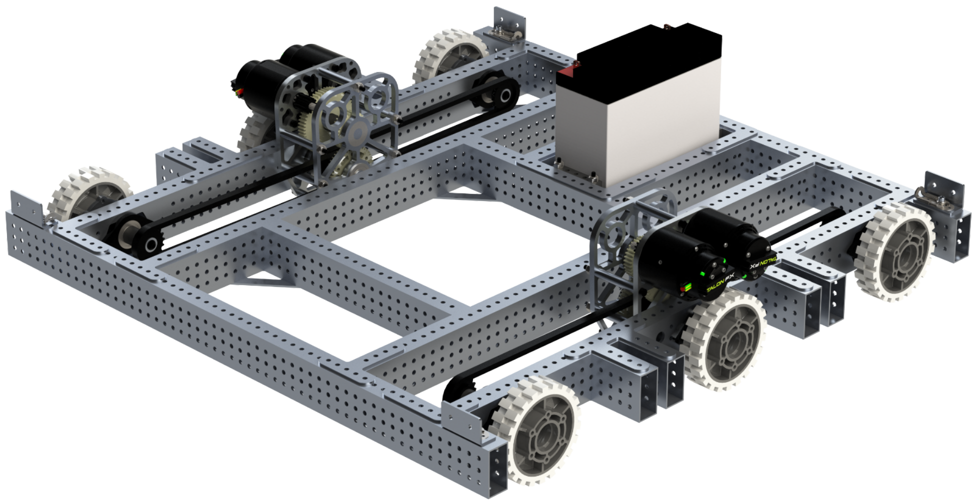
DESIGN





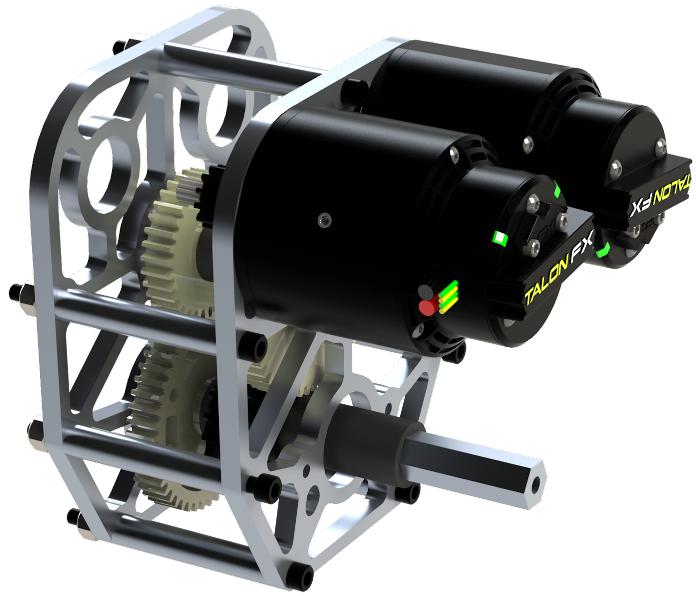


Drivetrain



- **Design Requirements**
    - Minimize weight by using 1/16” wall box beam and 25H chain
    - Convenient maintenance considerations
- **West Coast Drivetrain**
    - 28” x 28” frame perimeter to maximize space for mechanisms and modular hole pattern for seamless iteration
    - Six Andymark 4” hi-grip wheels for lightness and grip
    - 3/32” center drop for minimal scrub when turning
    - Chain tensioned using WCP bearing blocks and cams
    - Bumpers secured by snap-slide latches for quick and easy removal
    - 2 x 2 Falcon 500 gearboxes

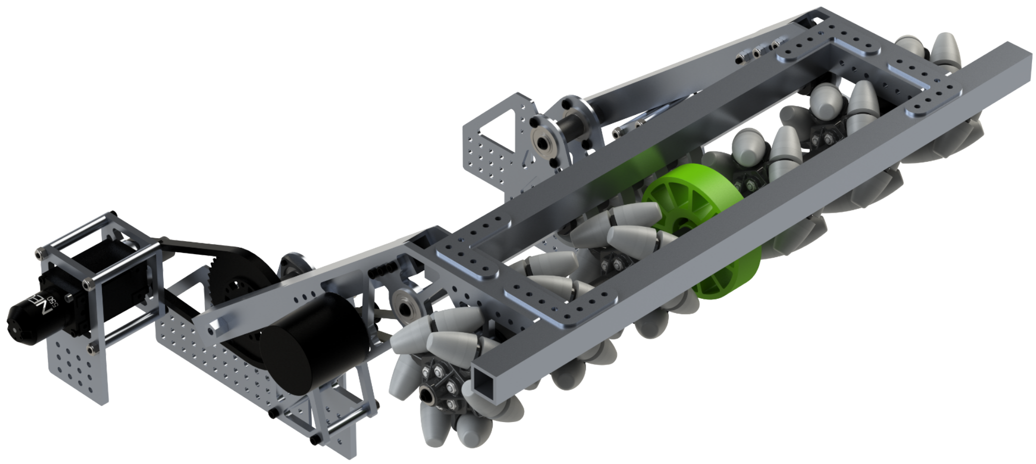
Drive Gearbox



- **Design Requirements**
  - Flexibility to change between multiple gear ratios by swapping pinions
  - Powerful enough to win pushing matches or cause wheels to slip and prevent stalling
  - Maximizes space within inner area of robot for indexer and other mechanisms

- **Flat 2 Falcon Gearbox**
  - 7:1 overall reduction
  - Drivetrain 15.91 ft/s theoretical max
  - Motors placed over wheel to reduce space

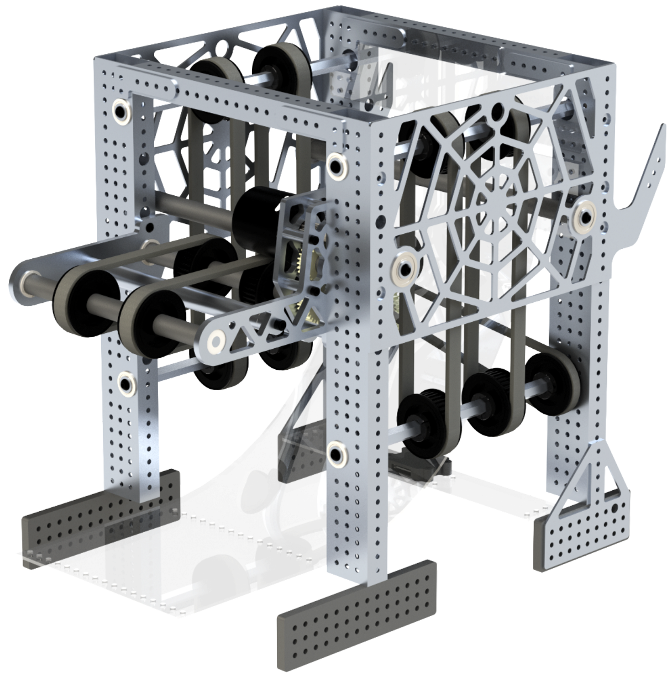
Intake



- **Design Requirements**
  - Fits within frame perimeter
  - Capable of quick deployment
  - Able to withstand lateral impacts against hub or other robots

- **Passive Mecanum Intake**
  - 4” mecanum wheels to vector balls inwards
    - 4” compliant wheel in the middle ensures the ball enters the indexer centered
    - Protected by metal tubing
  - 3:2 gear reduction
  - Second row of 2” compliant wheels ensures continuous ball compression
  - NEO 550 + Versaplanetary gearbox driving a sprocket of 225:1 reduction to stow when not intaking cargo and to avoid defense

Indexer



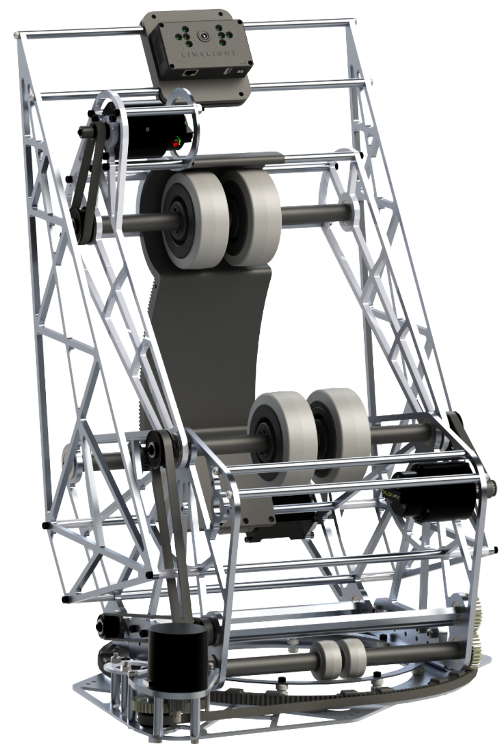
- **Design Requirements**
    - Ability to hold 2 balls at once
    - No jamming

- **Timing Belt Conveyor System**
    - 4:1 gear reduction
    - Timing belts create optimal ball compression
    - Curved polycarbonate sheet maintains compression and reduces slipping
    - Balls stored vertically/diagonally for lower center of gravity

- **Front L**
    - 2:1 gear reduction
    - Driven by separate gearbox from main column to ensure sufficient spacing between balls (thus reducing jamming)

Shooter

- **Design Requirements**
    - Able to shoot from multiple positions across the field
    - High accuracy + consistency
    - Capable of shooting two balls with limited drop-off

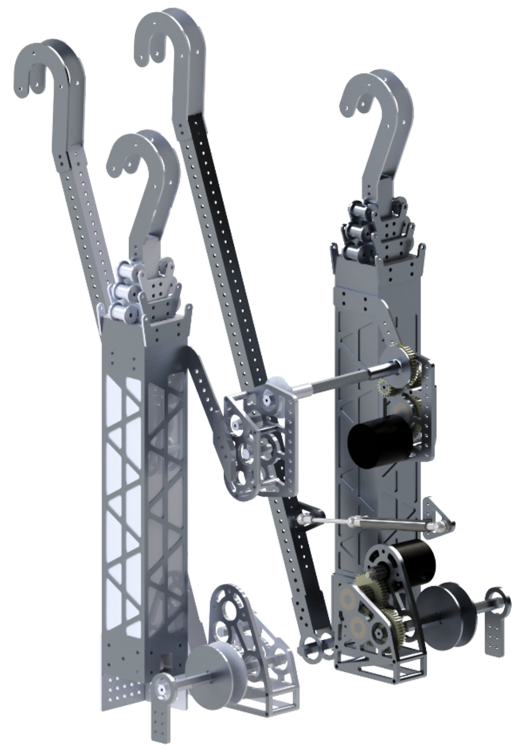


- **Double Flywheel Shooter**
    - Four 4” Colsons
    - Two Falcon 500 motors geared down to 9:4
    - Wheel Speeds: 2850 rpm (49.33 ft/s surface speed)
    - ¼” sheet aluminum with large pockets for light and rigid design

- **Continuous Hood**
    - 3D-printed rack and pinion driven by NEO 550s with a VersaPlanetary gearbox
    - Allows for shooting from any spot on the field

- **Turret**
    - 180 degrees of movement powered by NEO motor geared at 46:1
    - Shortens cycle time by reducing drive paths during auto and teleop

Climber



- **Design Requirements**
  - Robust, reliable, and minimal weight
  - Capable of reaching traversal rung
  - Unintrusive to ball path
- **3 Stage Continuous Telescoping Arms**
  - Max reach of 66”
  - Constant force springs and 3D-printed blocks create a smooth, vertical motion
- **NEO Gearbox**
  - 22.8:1 overall reduction
  - Climbs to mid rung height (60.25”) in 10s (theoretical max speed)
  - Kevlar strap pulls telescoping arm down and coils with ease
- **Swinger Arm**
  - Pivots on a point to allow for angle necessary for high and traversal rungs
  - NEO driven winch to retract swinger arm into frame perimeter
  - Gas shocks apply constant force to pivot swinger arms
- **Passive Hooks**
  - Pivot of hooks is collinear with the bar, preventing the bot from falling off while receiving downward forces
  - Allows quick alignment

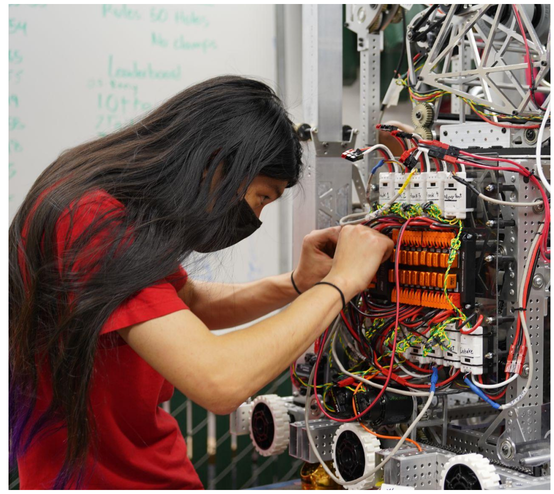
Electronics

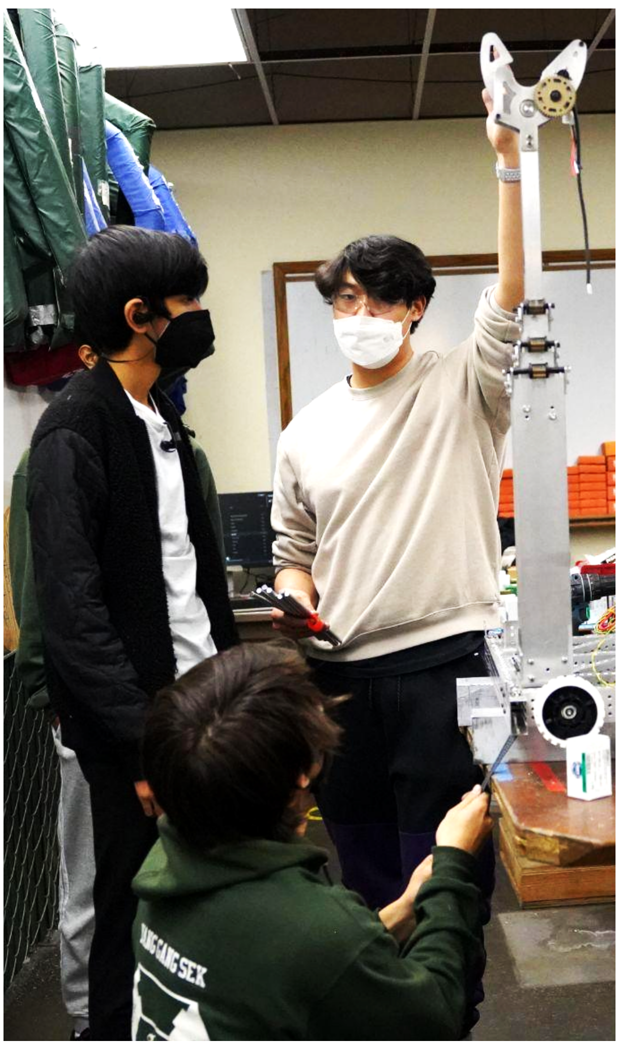
- **Design Requirements**
  - All electronics accessible for easy maintenance
  - Limited vertical space
  - Electronics protected from cargo, other robots, etc.

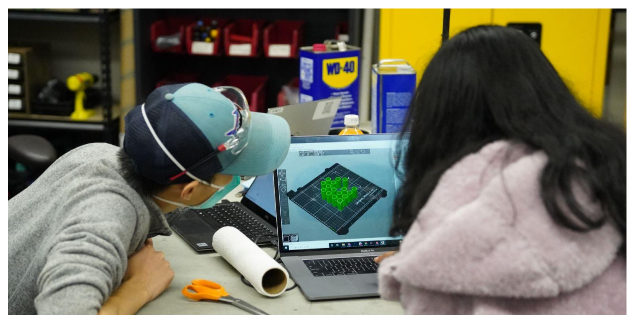
- **Split Vertical Electronics Boards**
  - Easily accessible, not blocked by mechanisms
  - Split into two boards, separated by related electronics components

- **Removable Covers**
  - Attachment with thumbscrews makes servicing and replacement between matches easy

ITERATION

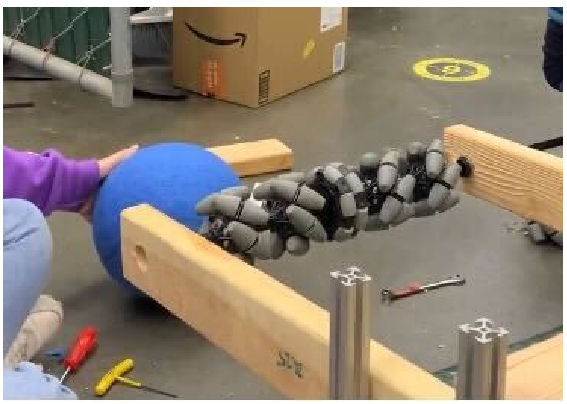


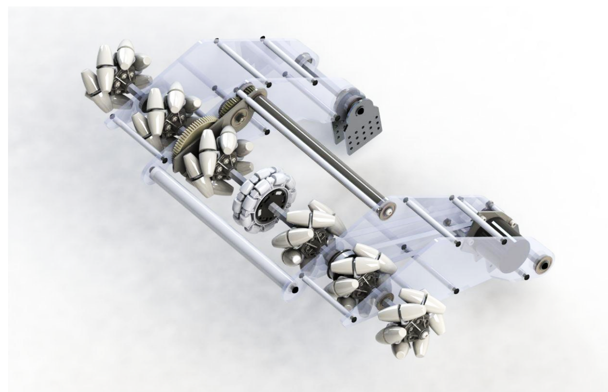




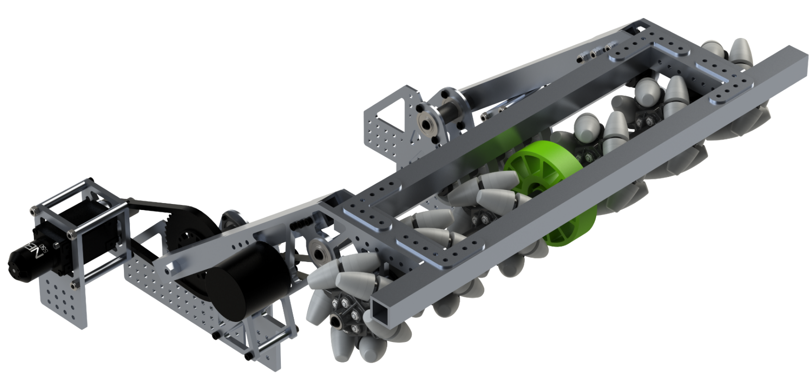
Intake

We started out by prototyping a 4” mecanum wheel intake with a centering omni wheel, and then integrated it into our robot design. However, when we tested it on the field, it would push balls away in some instances. Additionally, it was initially passive, but didn’t always deploy.

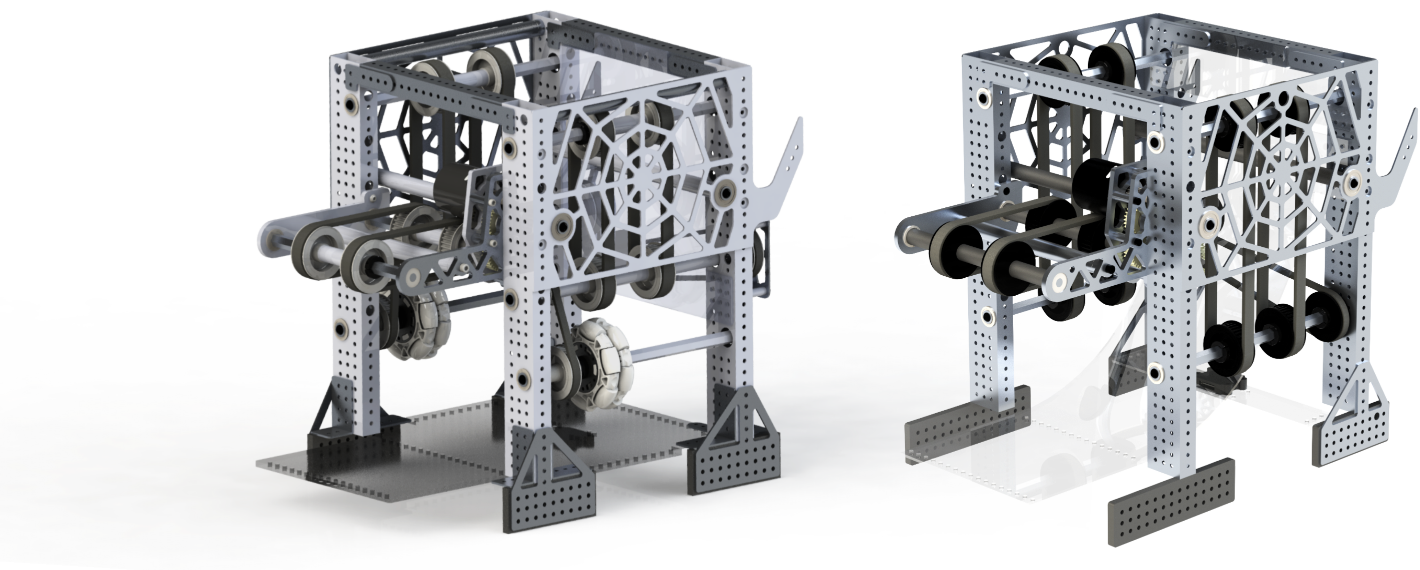




To address these concerns, we replaced the center wheel with a compliant wheel, which intakes balls consistently. We also decided to actuate our intake through the use of a motor, in order to control when it retracts.



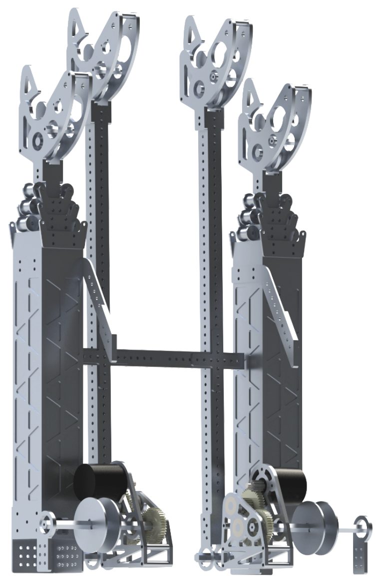
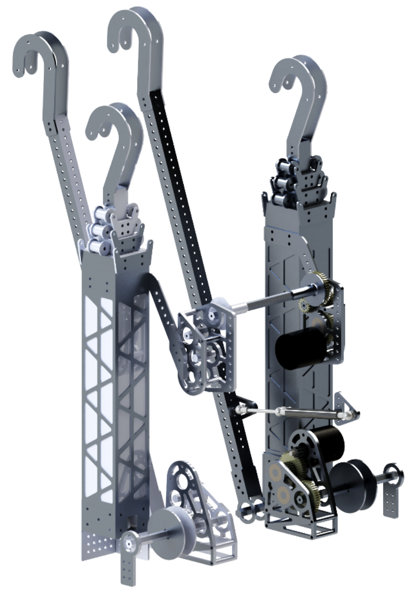
Indexer



Our indexer mechanism went through two iterations, the first of which was tested on our practice robot. The first iteration had issues with jamming and insufficient compression, resulting in a small dead zone, which we addressed by introducing a bent polycarbonate sheet.

Additionally, we removed unnecessary hardware like pulleys, an extra gearbox, and omni wheels to reduce weight and simplify the assembly process.

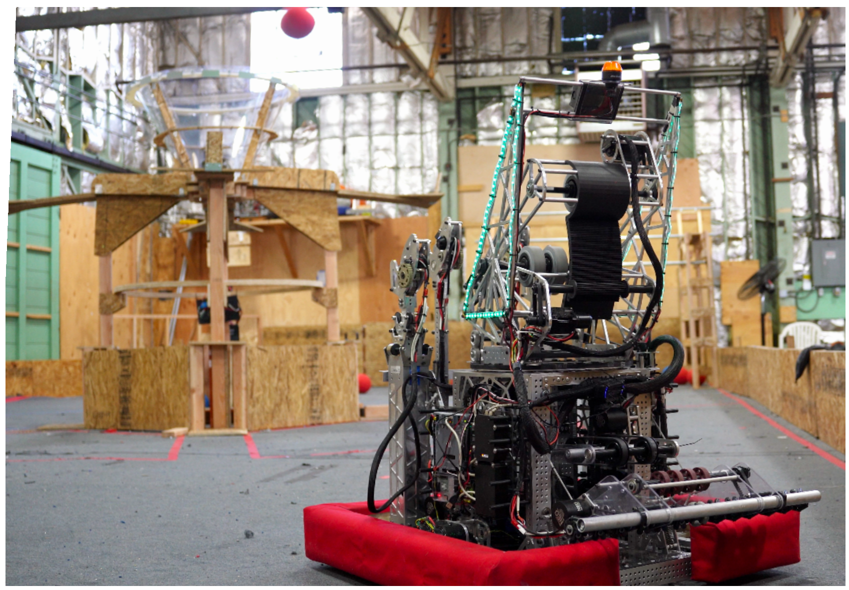
Climber

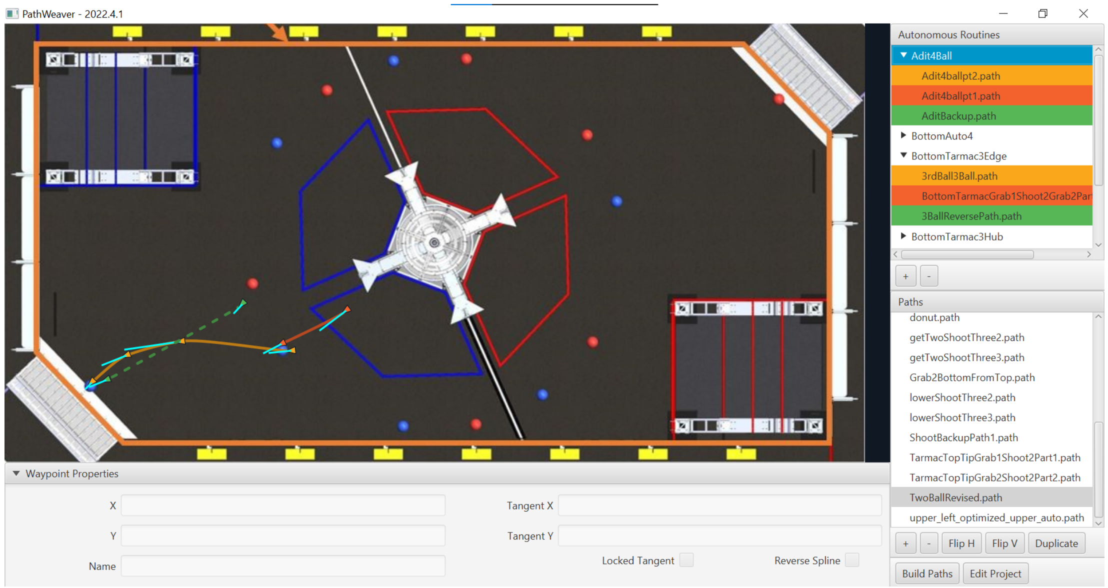
The first iteration of our climber used motor-driven clasping hooks and a passive swinger arm. However, this design was too reliant upon our robot's center of gravity, and also introduced a lot of swinging into our climbing routine.

In order to control the angle of our passive swinger arms, we implemented a gas shock and NEO-driven winch. We also exchanged our active hooks for passive ones to make aligning to the bar more efficient. As a result of all of these changes, we were able to cut our climb time by a full minute!

SOFTWARE



Autonomous Code



- **Path Following**
  - Drive paths created using the WPILib tool Pathweaver
  - Robot's velocity controlled through a trapezoidal motion profile
  - Current velocity and field position of the robot are recorded using a gyroscope and encoders
  - Drive controller uses the Ramsete algorithm to adjust the drivetrain's motor output based on error with target pose

- **Subsystem Routines**
  - Commands-based architecture allows us to run pre-programmed routines when predicates are reached

- **Vision**
  - Incorporates Limelight 2+ to detect vision targets through pixel and contour filtering
  - Provides us the distance to the target using the pitch detected by the Limelight, enabling our shooter to determine an appropriate target flywheel velocities and hood position.

Tele-Op Code

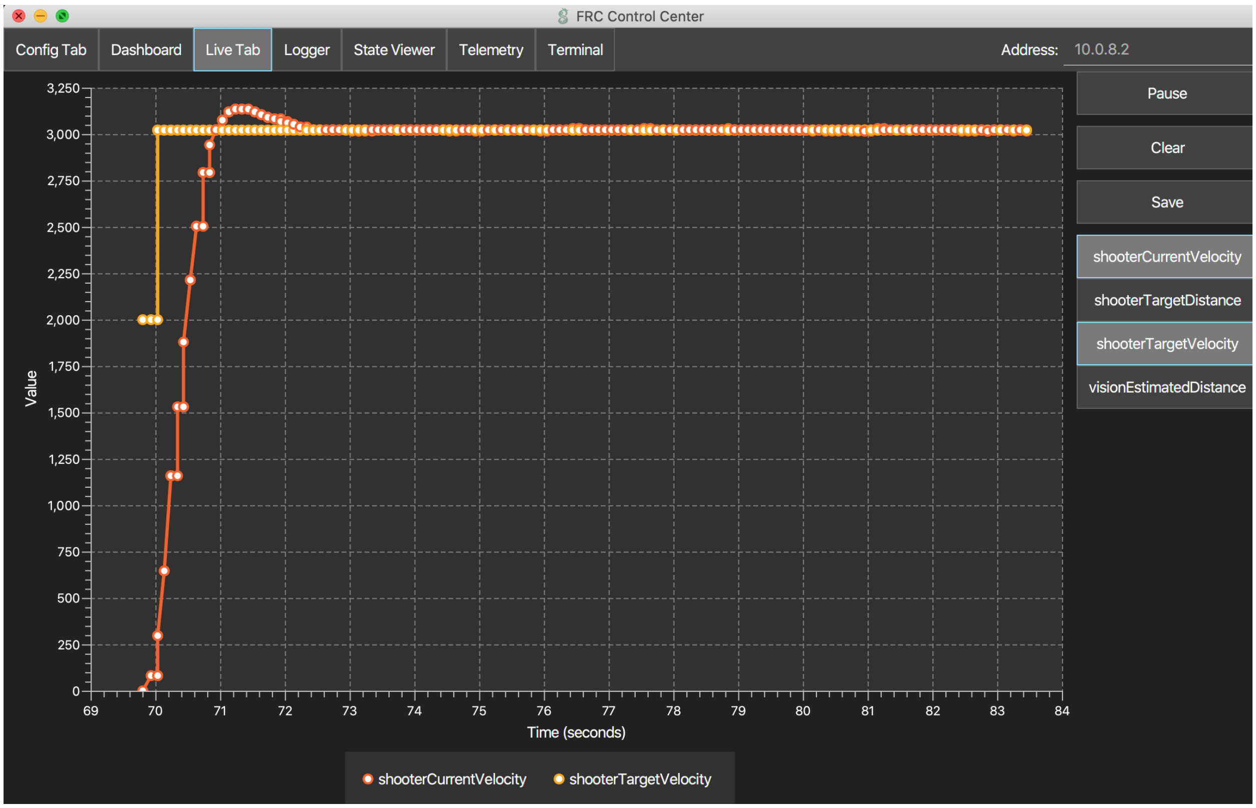
- **Intake, Indexer**
    - Supplies the motors with a tuned percent output value in order to feed cargo through the robot in a time-efficient manner while avoiding jams

- **Shooter**
    - Turret utilizes Limelight's detected yaw to target to adjust its position
        - When the Limelight is unable to detect the target, we use robot odometry (found using the drivetrain encoders and the gyroscope) to approximate our difference in our position from the last Limelight-detected position, allowing us to calculate the required turret angle to recapture the target, even when it is not visible
    - Interpolating tree map to automatically determine target flywheel velocities and hood position based on distance to the target

- **Climber**
    - Operator has manual control over the arm and hooks
    - Routines that toggle the hooks between opened and closed using buttons on the Xbox

- **Driver and operator feedback**
    - Xbox controller rumbles when the flywheels are at the desired velocity and the hood is at the desired position
    - LEDs on the sides of the robot provide the drivers with information about the turret alignment.

Control Center



- Custom application allows for quicker debugging and a faster workflow overall
- Streamlined UI that can display the robot's state
- Tab based architecture allows for future expansion and versatility, anything from creating autos to changing configs
- Live graphing of values for debugging, both quantitative and qualitative