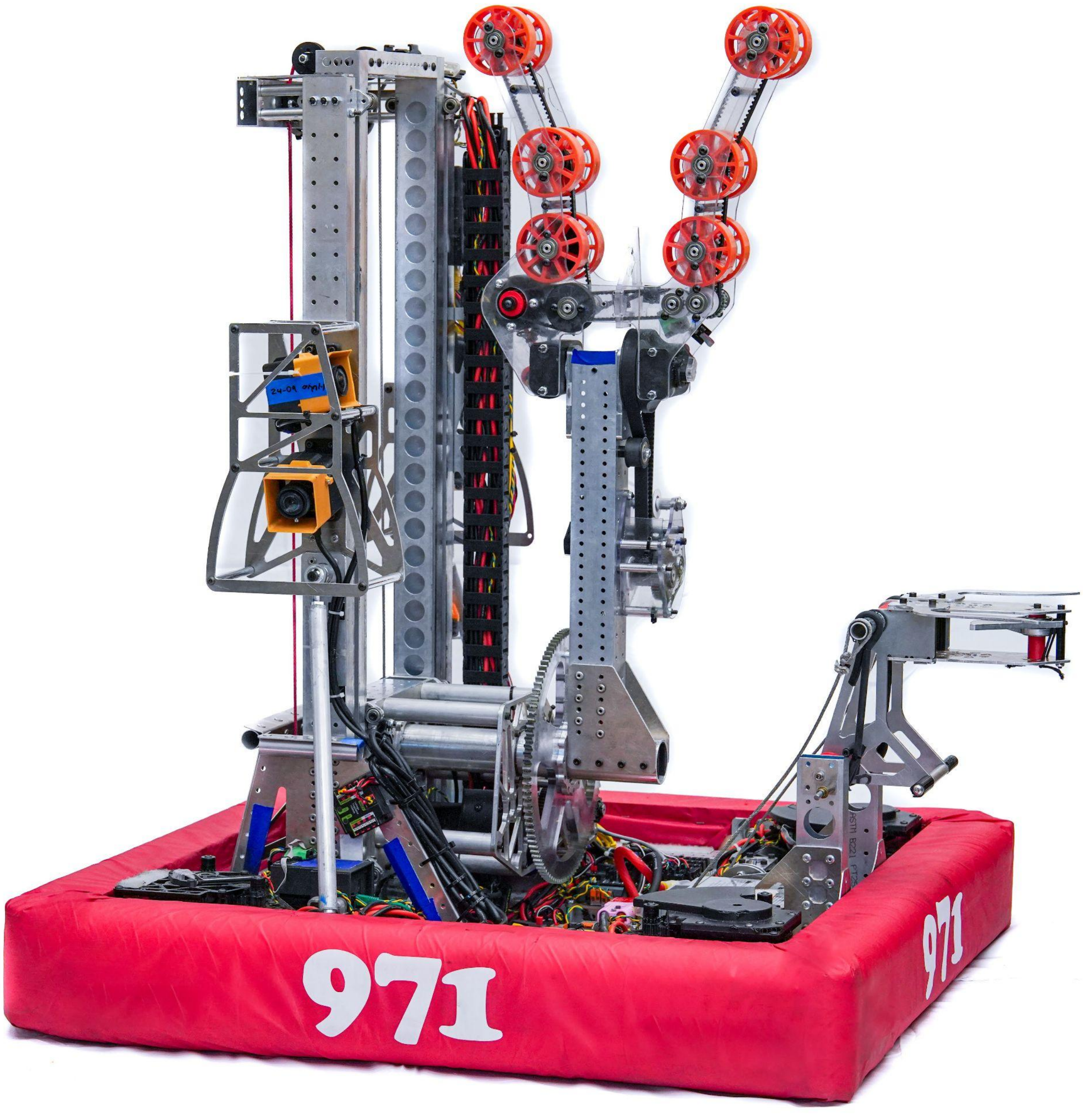
SPARTAN ROBOTICS



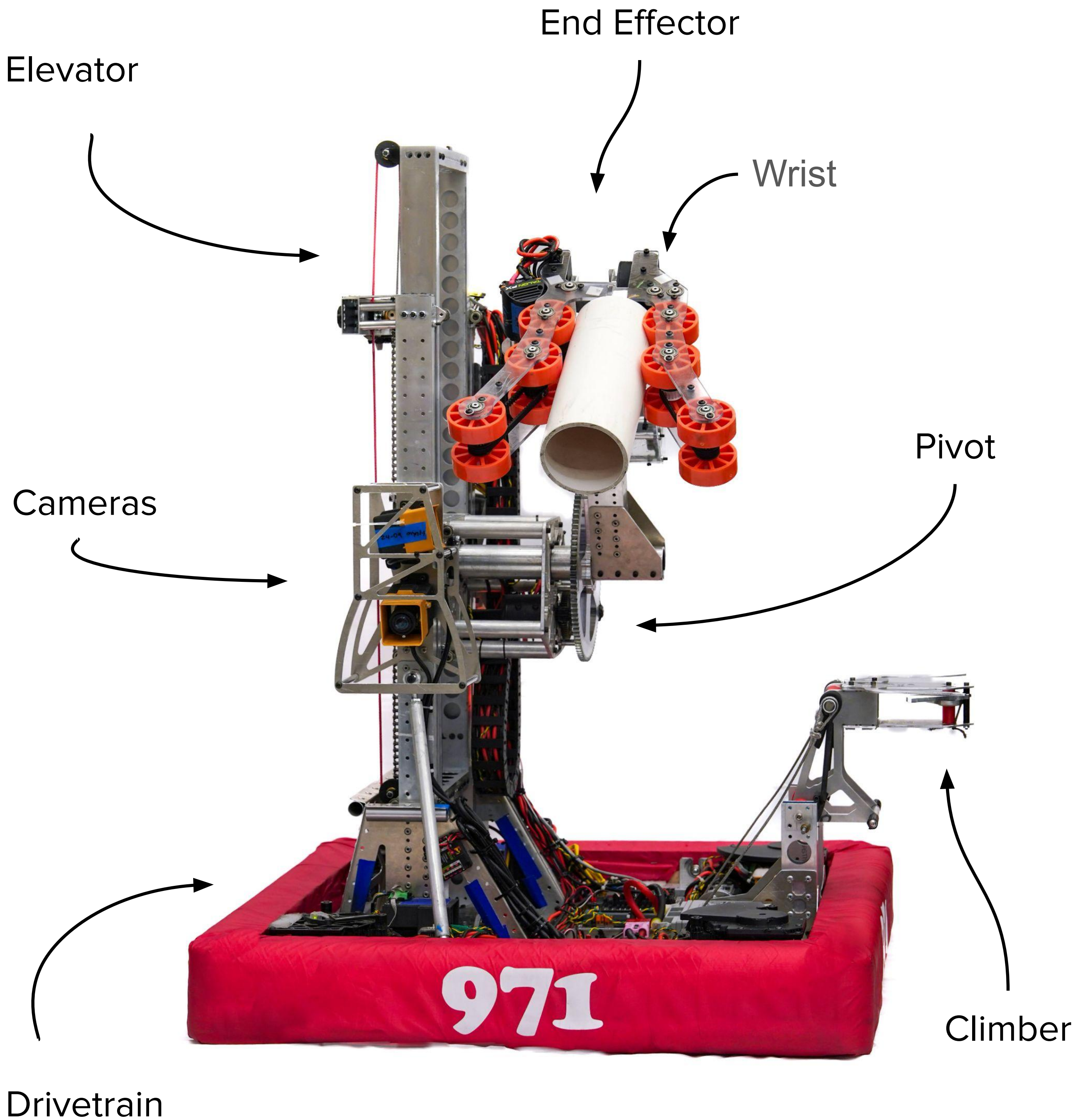
THE FIDDLER · FRC 971

Technical Documentation 2025

Table of Contents

Table of Contents

Robot Overview



Strategy Analysis

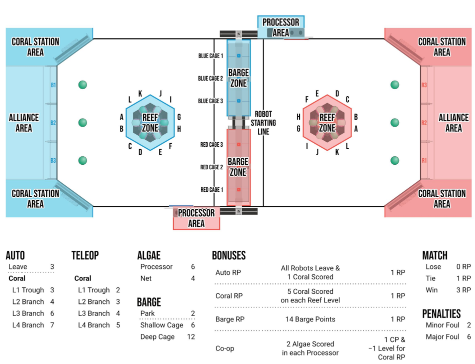
Primary Objective: Win a regional event and the compete at the Championship by executing a robot design that prioritizes accurate coral scoring.

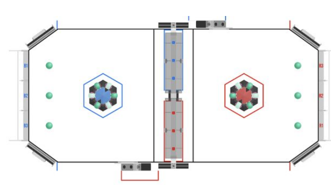
Prioritized Items

- Execute a consistent L4 coral auto
- Score coral in all levels efficiently and accurately, prioritizing L3 and L4
- Score algae in processor and barge
- Execute a fast deep climb
- Obtain at least 3 RP every single match

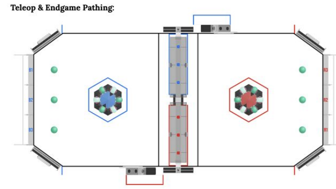
Obtaining Ranking Points

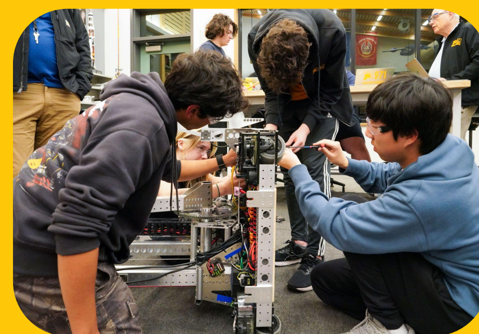
- Auto: ensure we have a consistent coral-scoring auto and utilize our ACE Team to guarantee all alliance partners get mobility.
- Coral: score coral efficiently in all levels to achieve 5 game pieces on each level.
- Barge: design and execute a consistent deep climb every match and ensure all alliance members park.
- Win!

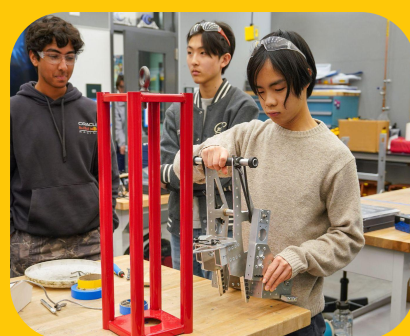




Teleop & Endgame Pathing:







Robot Design

Subsystem Breakdown

Drivetrain

- Fast X2i swerve drivetrain with traction wheels
- Maneuver swiftly between HPS and reef, avoiding defense
- Covered to protect electronics from falling coral and cages

Elevator

- Two-stage GreyT Cascade elevator
- Efficiently transport arm to score at all levels on the reef and pickup coral from the ground

Pivot + Arm

- Execute a 360+ degree range of motion
- Swiftly assume a wide range of scoring and pickup positions

Wrist

- Dead axle wrist mechanism with 220-degree range of motion
- Lightweight and allows accurate L4 scoring

End Effector

- Compliance wheels in a claw-style end effect
- Effectively manipulate coral and algae game pieces

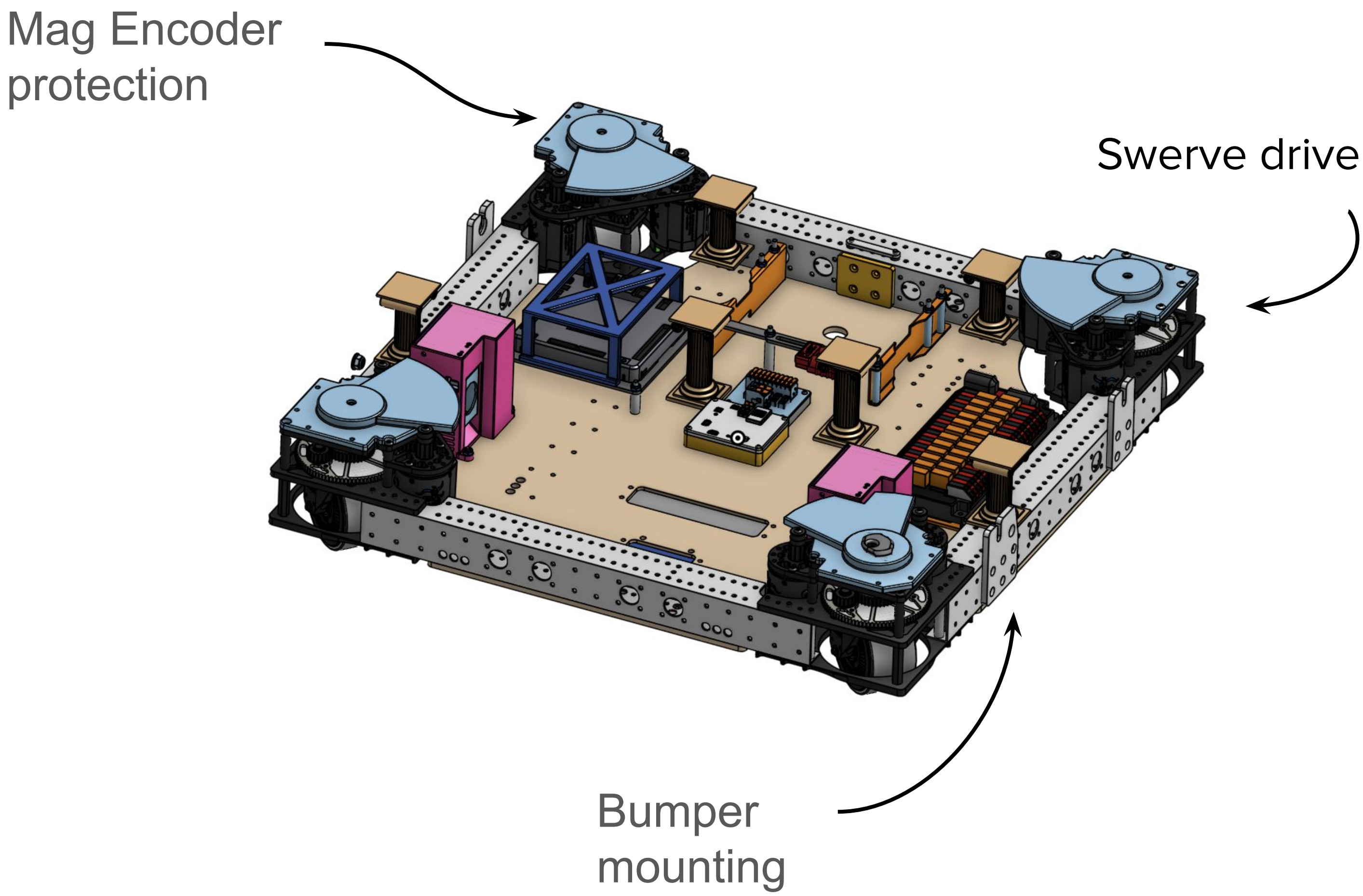
Ground Intake

- Two rollers guide coral into end effector
- Implemented after both regional events

Climber

- Latching hook design to elevate the robot off the ground

Drivetrain



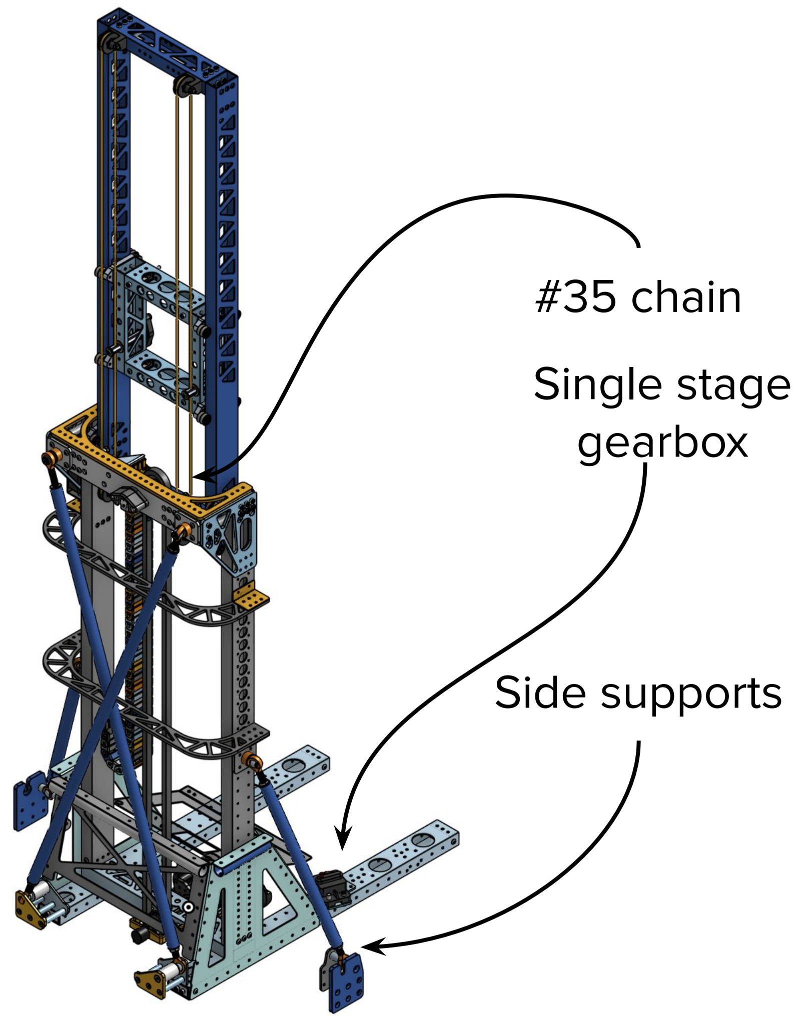
Features

- First season with swerve
  - WCP X2i modules
  - X1 10t Drive Ratio
- CAD - configurable tube lengths
  - 29in parallel distance

Manufacturing + Assembly

- Tubes and belly pan cut on CNC router
- Various 3D printed parts and covers to protect electronics

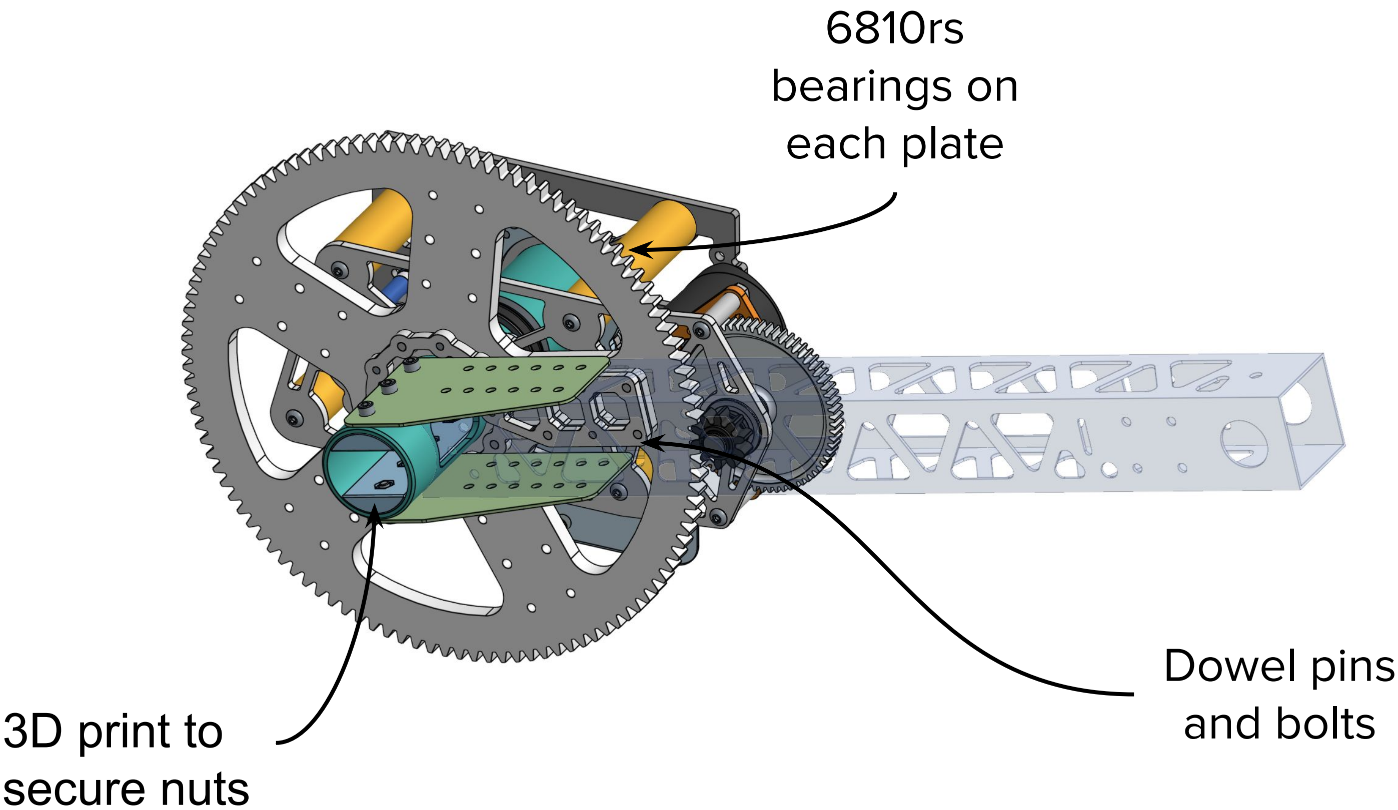
Elevator



Features

- 2 kraken x60s power 7.75:1 gearbox
- #35 chain to handle weight of carriage
- 2 stage cascading elevator that extends to ~6 feet
- 0.75” OD tube side and back supports provide extra stiffness when scoring
- 10 turn pot and 2 mag encoders tracks elevator height and provides redundancy from chain slipping
- Cross plates provide stiffness at key scoring positions
- Full extension in ~1 second

Pivot + Arm



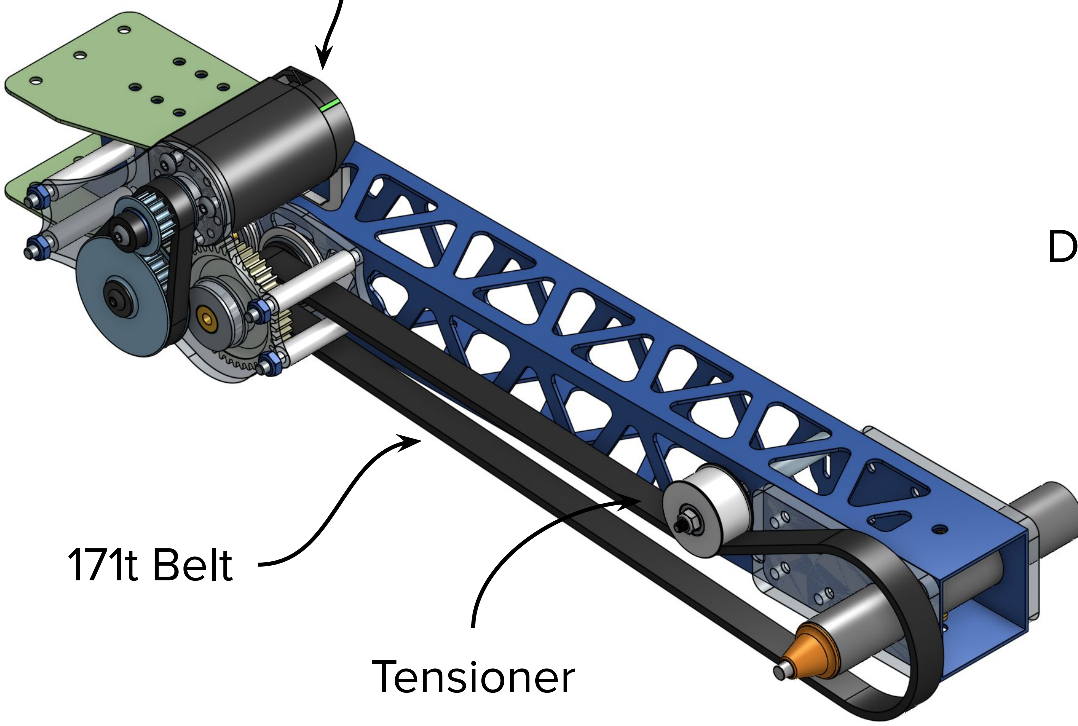
Features

- 110 tooth polycarbonate 10DP gear, forces calculated using Lewis Equation
- 1” OD tube spacers with tube nut inserts to create a rigid carriage
- 6810rs bearings and large tube shaft for rigidity
- 1:96 gear reduction to be able to run Kraken at peak efficiency and exert ~33 Nm of torque
- Combination of dowel pins and bolts to adequately transfer load from the gear to the arm

Electronics

- IGUS for wiring through elevator
- IGUS rests against lexan sheet mounted to the elevator
- A 10 turn pot and mag encoder driven by a belt off the pivot shaft to measure pivot angle

Wrist



Design Process

- Originally 3DP live axle mount
  - Switched to dead axle
- Shifted Gearbox
- Increased to 15mm wide belt
- Added tensioner
- Moved kraken to clear pivot
- Added idler to increase ROM
- Removed tensioner blocks

Features

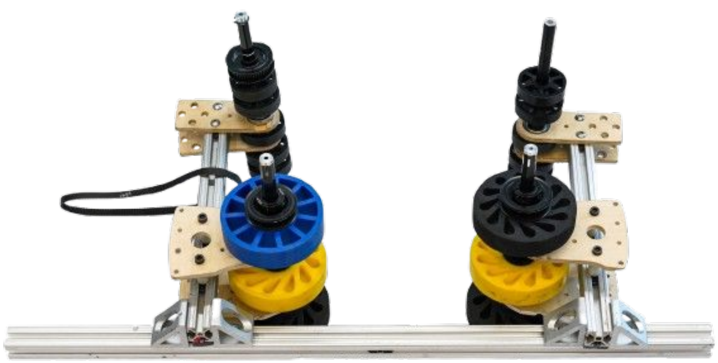
- 171t 15mm belt
- Tensioner + idler system to maximize ROM
- 220° End Effector ROM
- 1:28 Gear Reduction for 5.3 Nm of torque
- 180° rotation in less than 150 milliseconds

End Effector Prototyping

Goal

Converge on a single mechanism design that can:

- Intake coral from the human player station and ground at a wide range of angles
- Intake algae to remove from the reef and score in the processor and barge
- Score coral in L3 and L4 very efficiently + accurately



Compliance Claw

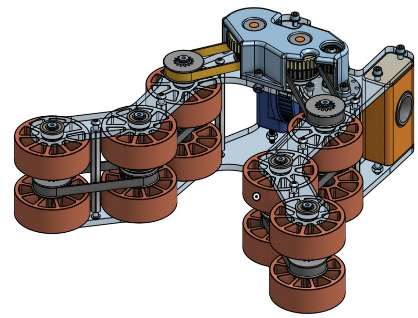
Suction Testing

Indexing Intake

End Effector Revisions

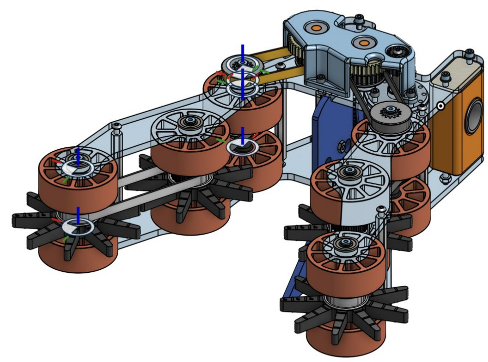
Rev 1:

- Initial test of a funnel with a wrist. Issues with indexing coral due to high compression and too wide of an angle.



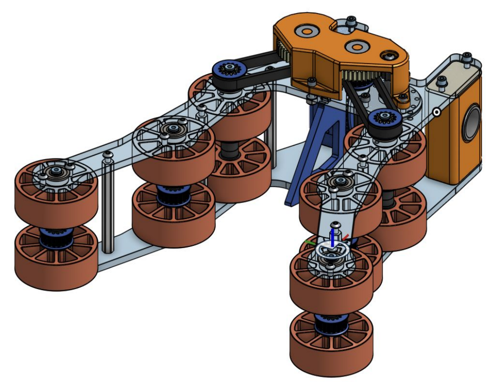
Rev 2:

- Narrow angle was promising, but the star wheels restricted our ability to intake algae.



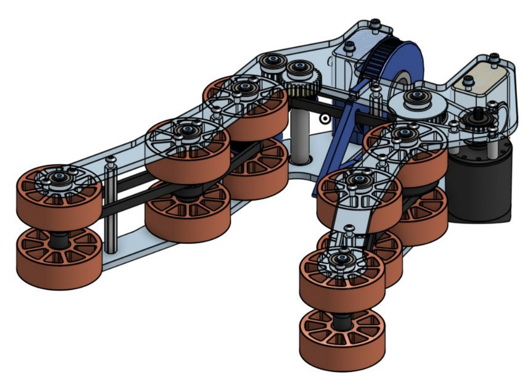
Rev 3:

- We removed the star wheels, but the gearbox cover continued to break during drive practice.

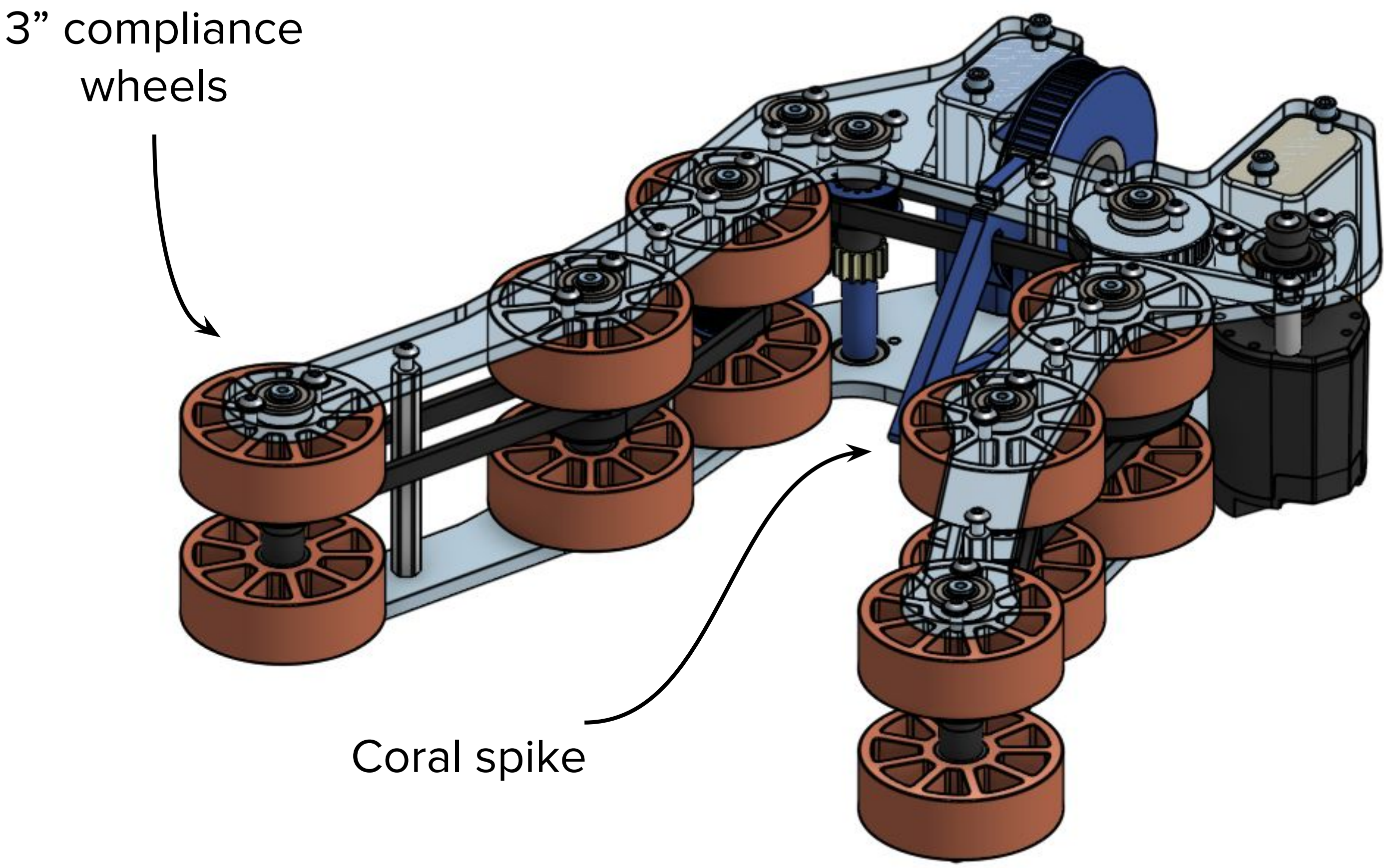


Rev 4:

- We moved the gearbox inside to increase ROM and durability.



End Effector Current Specifics

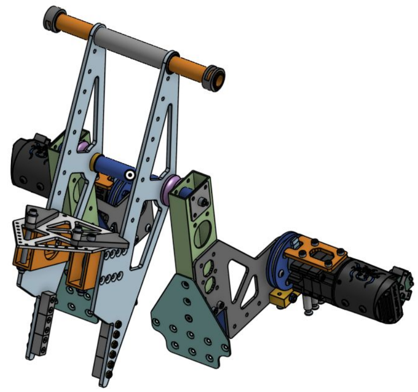


Features

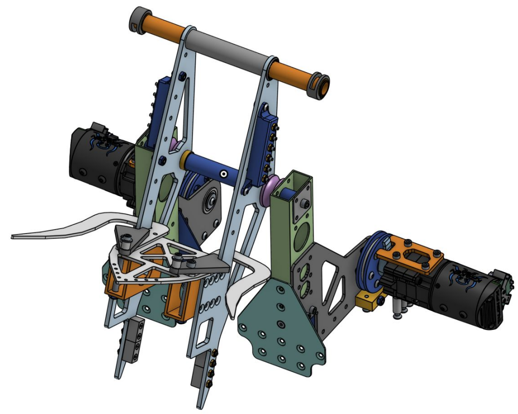
- Minor fixes from previous iteration ➔ greater ROM, better retention of algae, more torque
- Uses WCP SRPP for increased weight savings and durability
- 3” compliance wheel + custom 3D printed pulley stackups
  - Center-to-center distance determined by 80-20 prototyping
- Spike to help secure coral in end effector during transport
- Powered by Kraken X60

Climber Revisions

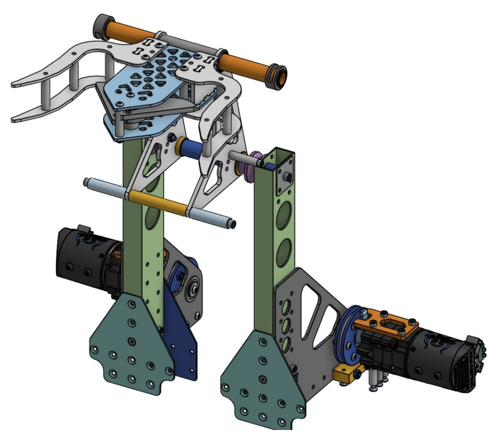
Rev 1:
- Difficult to latch because of the angle



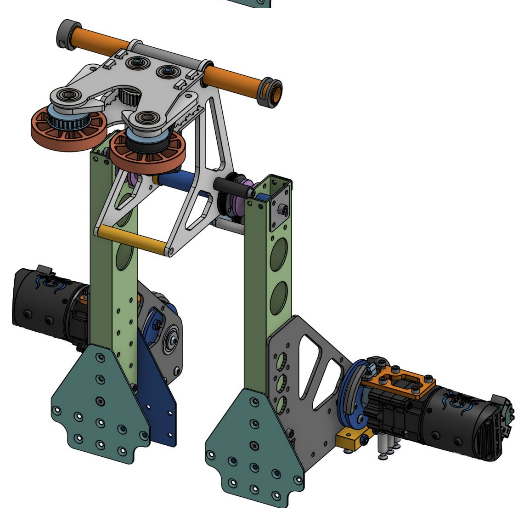
Rev 2:
- Added guiding plates, more magnets
- Optimized angle from this general design



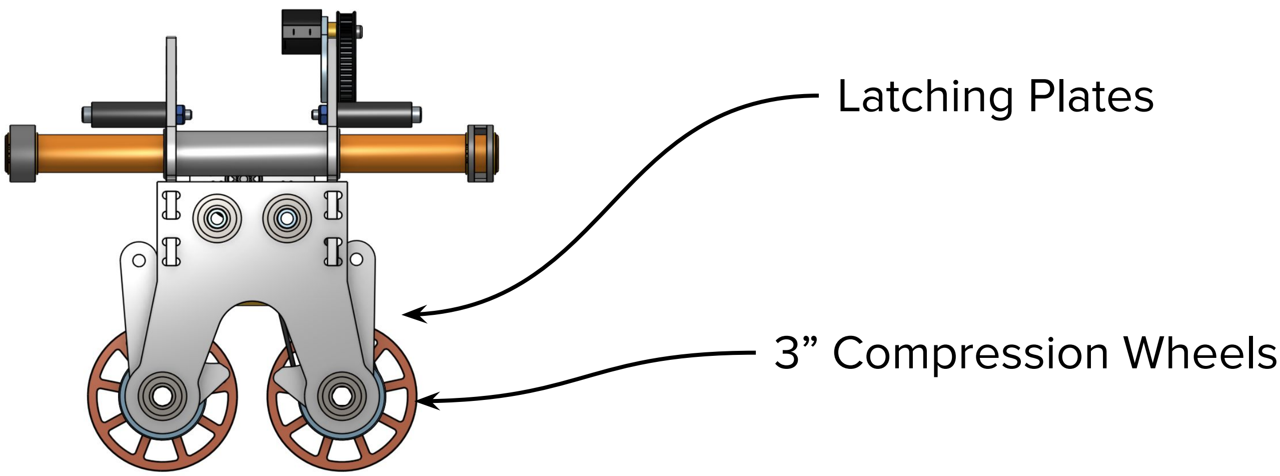
Rev 3:
- Increased the height of the pivot point to avoid slip
- Alternate latch in case slip was still a problem

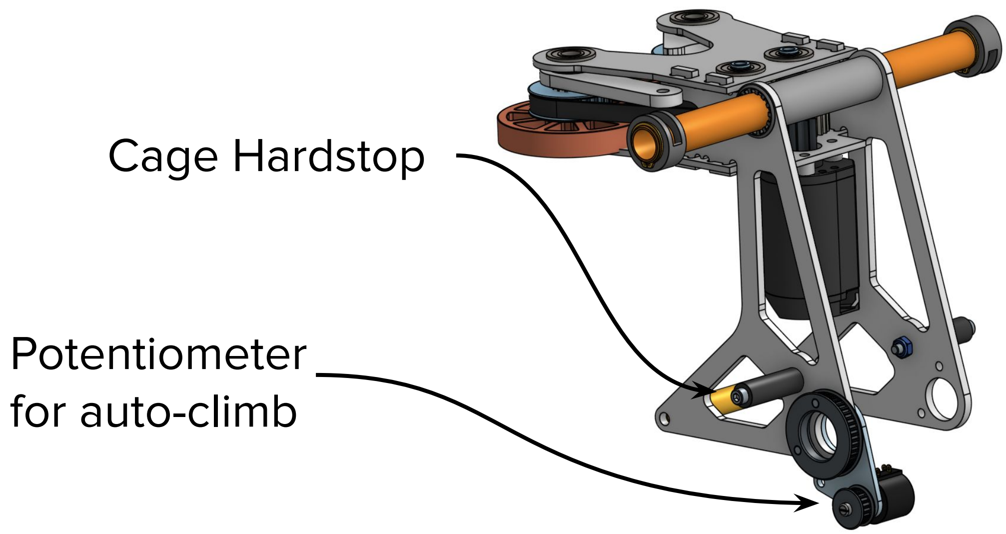


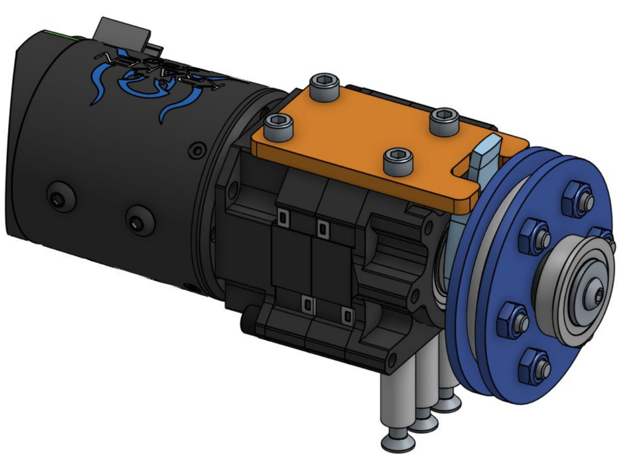
Rev 4:
- Switched to wheeled climber design
- Kept same pivoting mechanics as previous rev



Climber Current Specifics



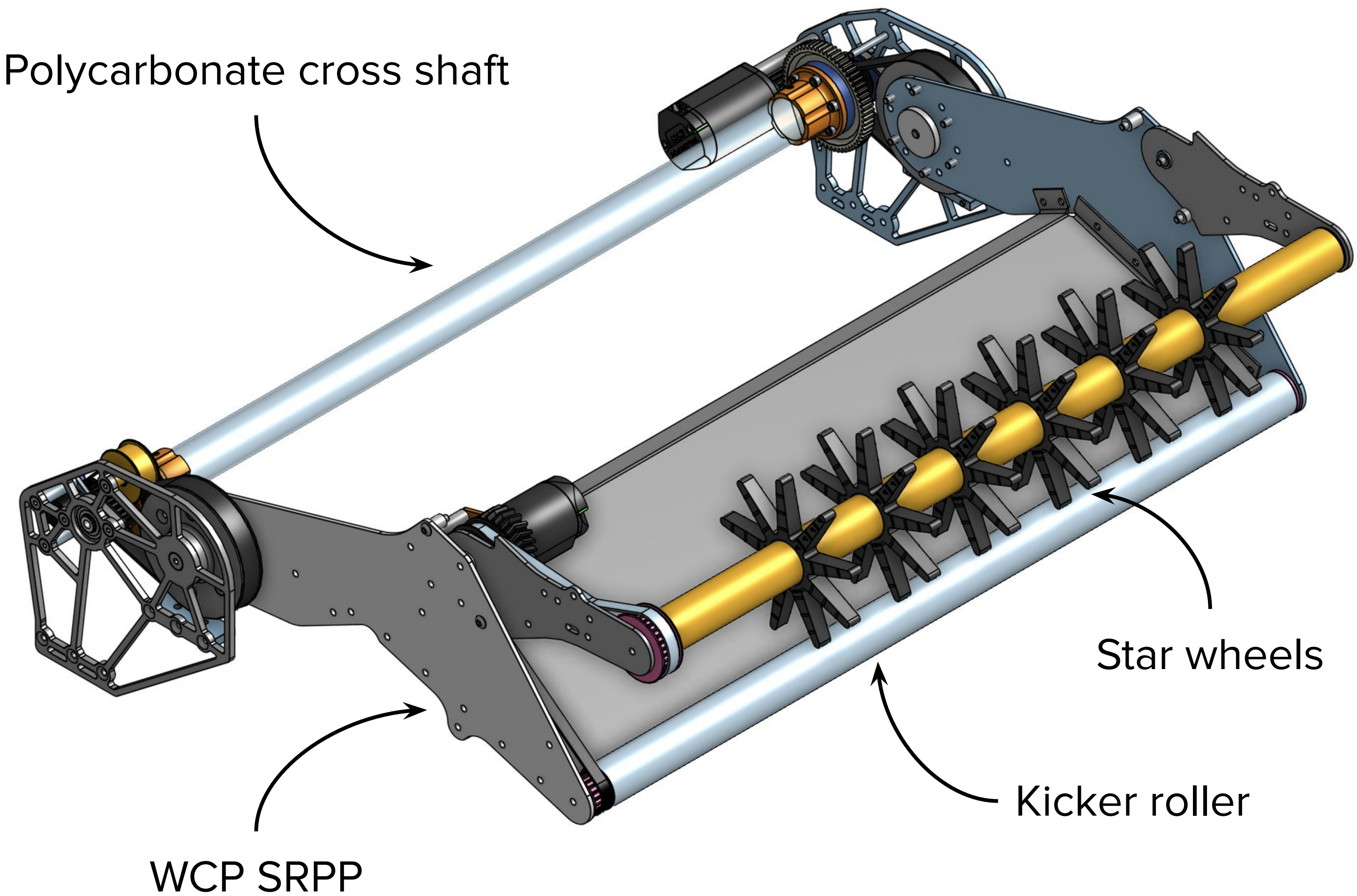




Gearbox:

- Two 45:1 MAX Planetary Gearboxes (one per side)
- Reversible ratchets to prevent falling down

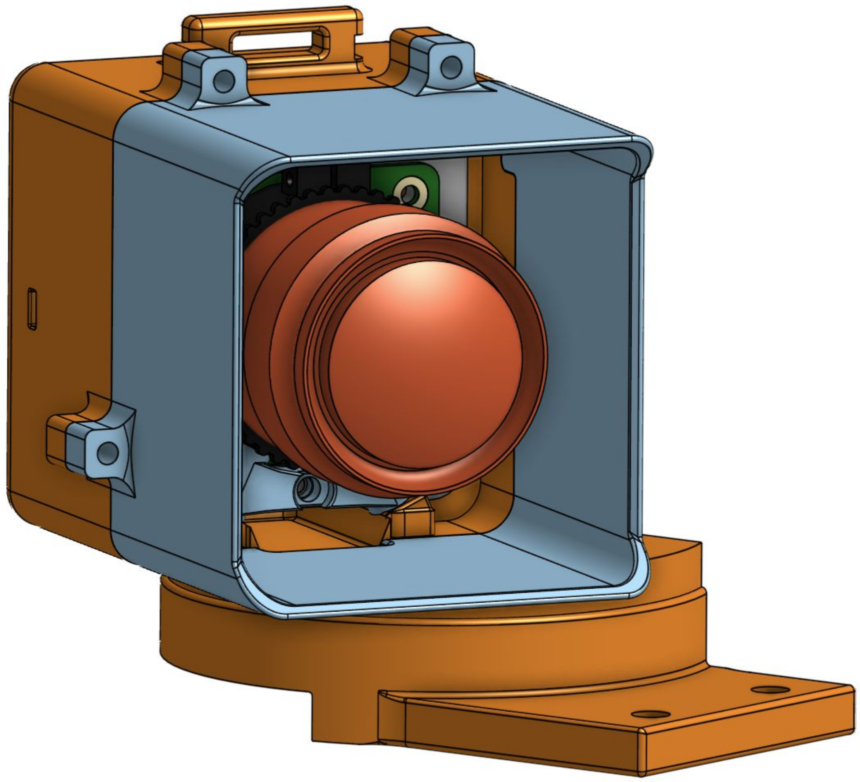
Ground Intake



Features

- Implemented ground intake before Champs
- Motor powered slapdown intake with a polycarbonate cross shaft
- Uses WCP SRPP for increased weight savings and impact resistance
- 5” star wheels brings coral in and kicker roller helps pull it off the ground
- Top roller connected to rest of intake with bushings and surgical tubing tensions it

Cameras

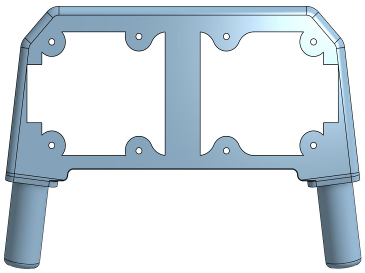


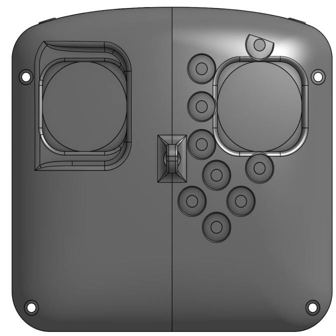
Cameras mounted to swerve modules/drivetrain tubes

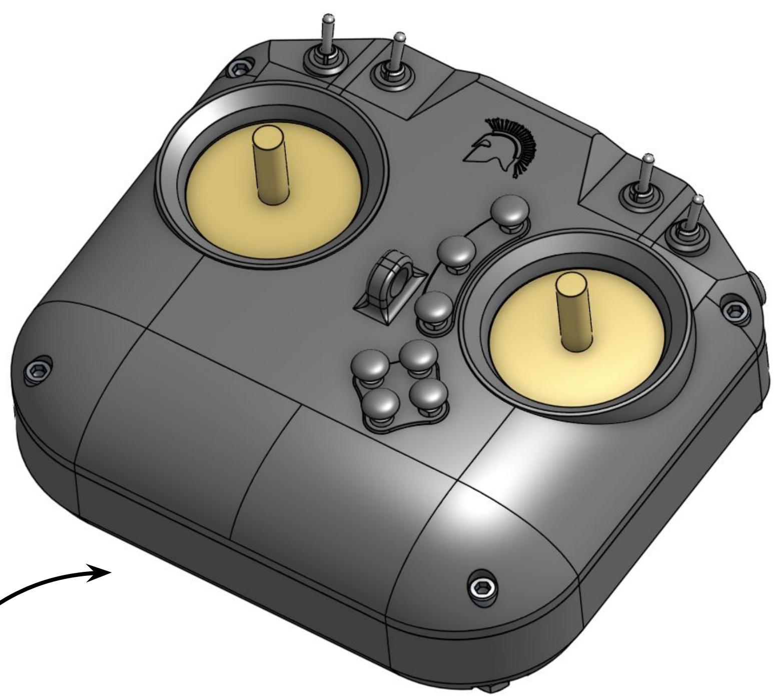
Specifics

- Raspberry Pi Global Shutter Camera
- Every two cameras connect to an NVIDIA Jetson Orin NX 8GB
- Two cameras allow for reef apriltag detection

Joystick Controller



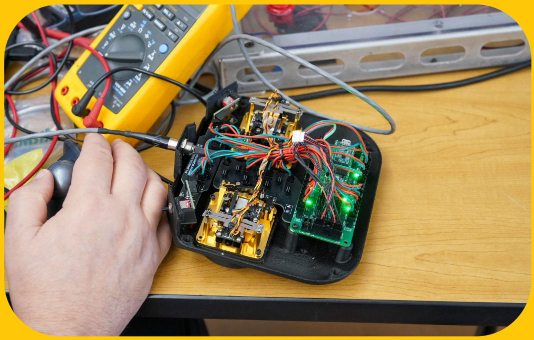
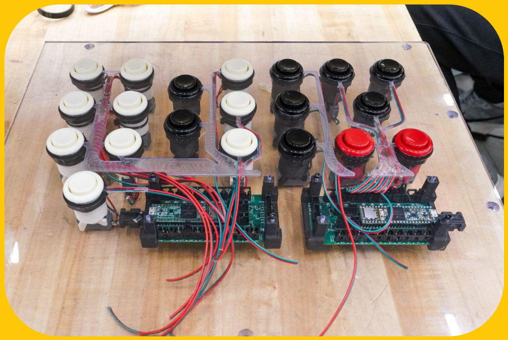




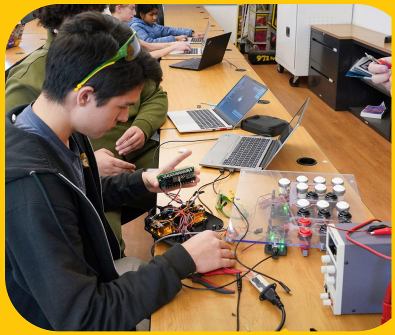
Controller Specifics

- Radiomaster Ag01 Gimbals
- 13 buttons/switches
- 3 Custom PCBs

Current iteration



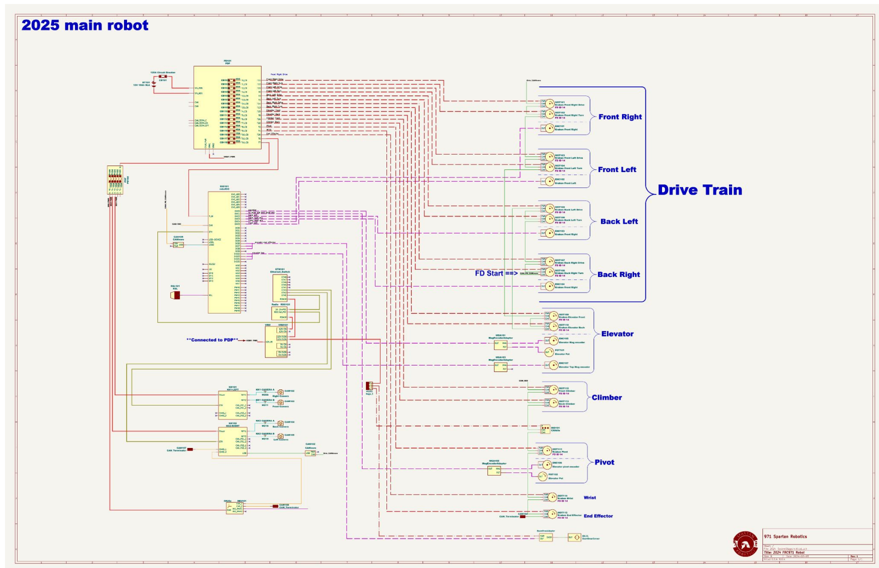


Electrical

Robot Wiring

Planning

- Created a full robot schematic with all ports of PDB, roborio, sensors, orins, motors, etc.
- All subsystems are represented
- Measured lengths in CAD to understand how long wires needed to be



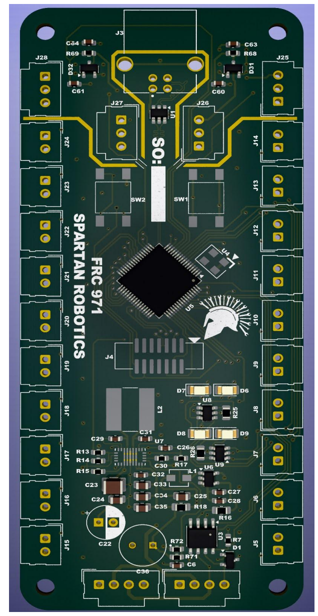
Wiring Process

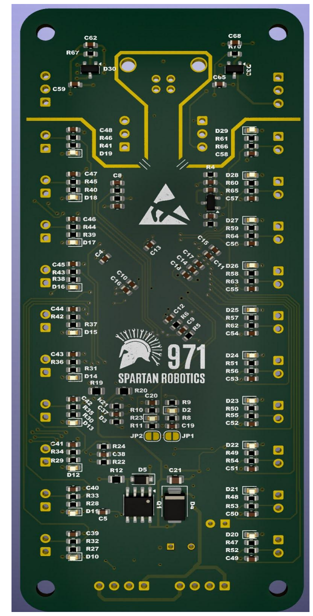
- Repairability:
  - Connectorized at many joints for easy hot swapping
  - Standardized pinout for simplified testing
- Cutting and crimping wires:
  - Used the system diagram to know what cables go where
- 2 CAN busses: one CAN-FD to orins for swerve motors and one for everything else to roborio
- Routed through the IGUS cable chain to protect electrical components

Custom Boards

Design

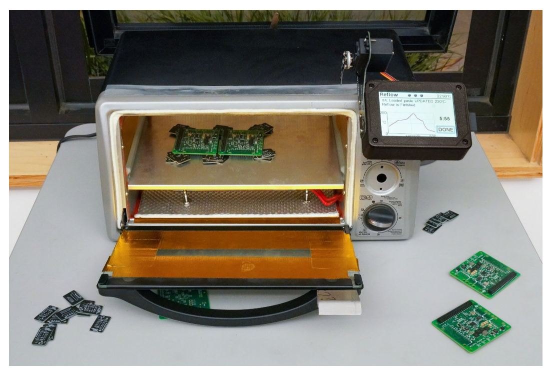
- Identify a recurring problem or a feature
- Find all the features and build requirements
- Design in KiCad
- Build & Test





Assembly Process

- Used both solder paste masks plus reflow oven and hand soldering
- Made 50+ boards total
- 6 students involved
- Total boards manufactured:
    - Spartan Board
    - Optical Encoder Adapter
    - SRX Encoder Adapter
    - Potentiometer Adapter
    - CAN Terminator Board
    - Beam Break Board
    - IMU-2X V3 Board
    - Teensy 3.5 Button Board
    - MIPI to HDMI Board
    - Mini Power Distribution Board
    - NX Adapter Board
    - NX Indicator Board
    - Time of Flight board
    - 3 Controller Boards

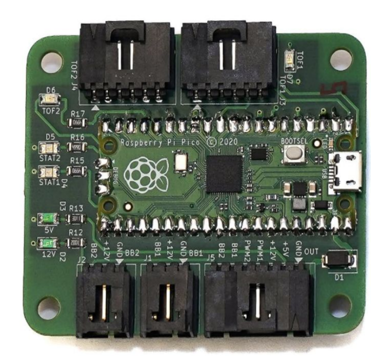


Custom Board Inventory

Teensy 3.5 Button Board:

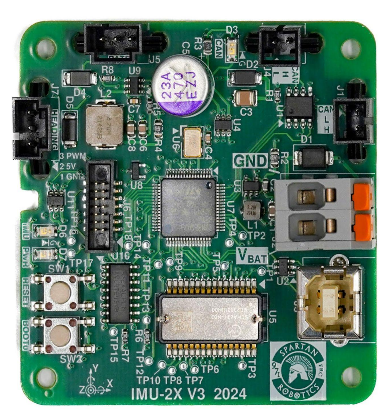
- Uses the Teensy 3.5 microcontroller
  - Not sold anymore - new STM32 based board coming soon
- Three of these boards are used in our custom driver station
  - 1 in our custom controller for getting data from Ag01 gimbals
  - 2 are used for button inputs

Time of Flight:



- Takes in 2 beam breaks and 2 TOF controllers
  - TOF controller measures the ‘time of flight’ of light that is reflected and gives us the distance
- Takes the output of the TOF sensor and turns it into something the roborio can understand

IMU Board-2X V3:

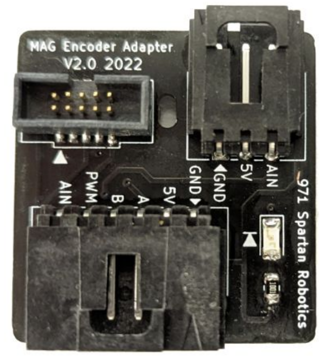


- Used for localization
- Transfers IMU (position) data to one of the orins (communicates using CAN-FD)
- Uses a high accuracy, relatively low range IMU combined with a high range, lower accuracy IMU

Custom Board Inventory

Mag Encoder Adapter:

- Combines the digital and analog inputs of both the mag encoder and potentiometer into one connection that goes back to our Spartan Board



Shaft Encoder Adapter:

- Takes the insecure and weak connector on the optical shaft encoder and turns it into our standardized molex connector system

Potentiometer Adapter:

- Mounts to the malformed face of the potentiometer
- Instead of it being when we break one of the potentiometers we have to desolder it, we can just unclip the connector and swap it

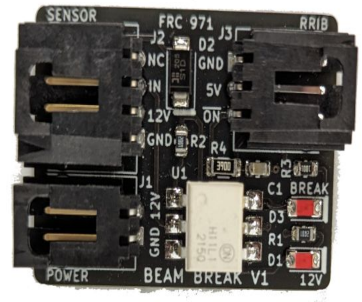
Spartan Board V2.0:

- Placed directly onto the roborio
- Converts the non-locking connectors into our molex connector system, allowing for a firm connection

Custom Board Inventory

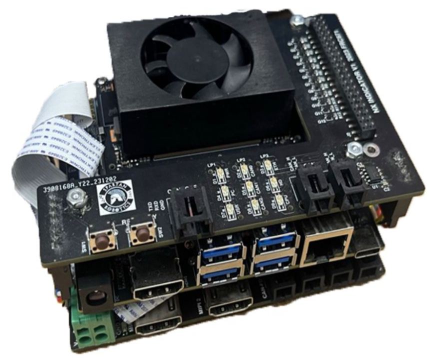
Beam Break Adapter:

- Powers the beam break sensors on the robot
- Takes 12V from the VRM and sends it up to the beam break
- Outputs the digital signal from the beam break
- There isn't a beam break on the robot, but the wire is routed and the board is ready in case we add one to the end effector later



NX Adapter Board:

- Goes underneath the Seeed Studio J401 NX interface board
- Takes in power and CAN
- Has HDMI ports to connect cameras

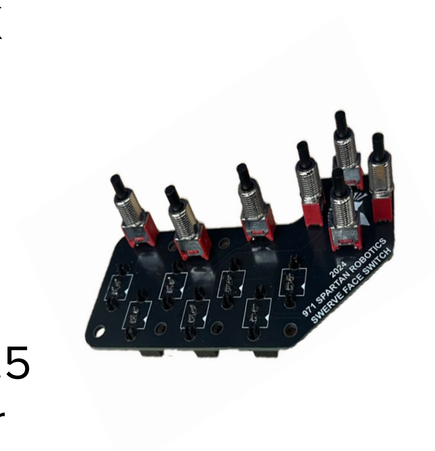


NX Indicator Board:

- A collection of LEDs to show status
- A CAN transceiver for the NX internal CAN interface
- A serial debug interface and reset and download buttons to enable flashing the NX

Controller Boards:

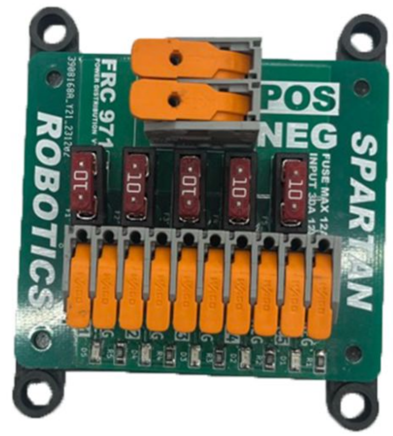
- 3 different boards in controller
- Converts buttons and switches to molex connectors
- Wires go from the connectors to a Teensy 3.5 Button Board inside of the custom controller



Custom Board Inventory

Mini Power Distribution Board:

- Plugs into a 20A slot on the PDP
- Distributes power into 5 ports, each with a 10A fuse
- Powers the IMU and VRMs

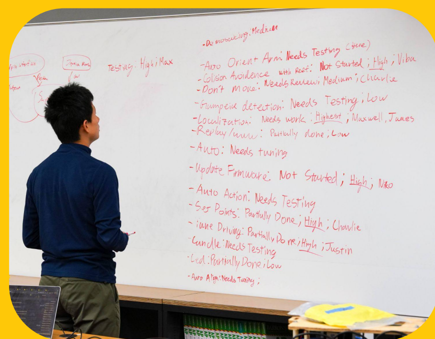


MIPI to HDMI Board:

- Converts the MIPI cable coming from our cameras to an HDMI port
- The HDMI cables then connect to the orins
- This board is within all camera 3D-printed casings, so the MIPI cable or the board is never exposed and is less likely to be damaged

Kraken Power/Can Adapter Boards:

- Converts the terminals for CAN and power to anderson connectors (for power) and molex connectors (for CAN)
- Easily accessible
- Easy to swap out wires when necessary



Software

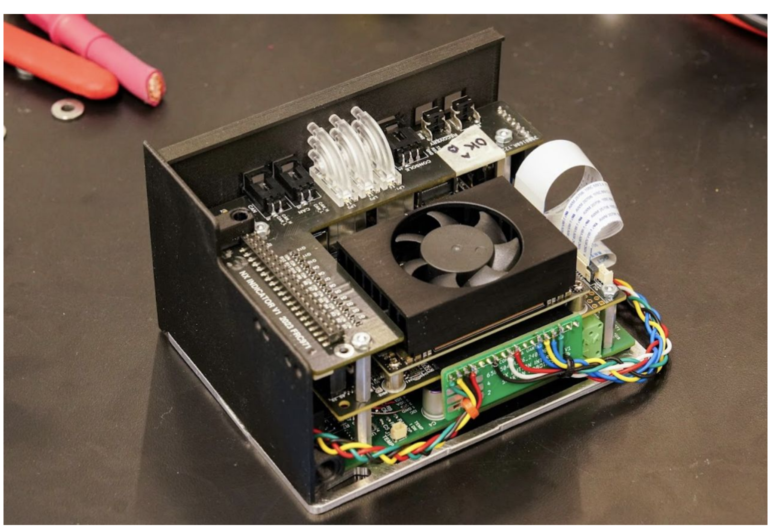
Orins

Why we use Orins

- In the past, we dealt with consistency issues on the RockPis used to run our vision processes
- We also dealt with a lack of compute, which was limiting what we could do on them

Coding the Orins

- The NVIDIA Orin modules were the solution we landed on, containing GPU processing and pcie expansion which enabled us to connect via CAN to our IMU-2X
- We had to patch the kernel and create a custom Operating System image which works with our code
- We then wanted to rewrite the apriltag detection library we used into CUDA code which ran on the GPU. Giving us a significant speed up and allowing us to run detections on 1456x1088 images at 60fps
- Finally, using our custom boards, we've read in the data from 2 cameras, running 2 detection processes which shared the resulting pose estimates with the Localization orin



A complete Orin stack

Swerve Control

Linearizing LQR:

Theory

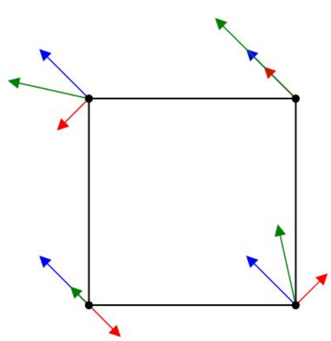
- Swerve physics are nonlinear ➔ regular linear controllers like the LQR don't work globally
    - However, they do approximate well at each time step if we linearize the physics

Implementation

- Solve Inverse Kinematics based on joystick goal to give ideal module position
- Solve LQR using those positions for input(motor current)
- Linearize Dynamics using the google's Ceres functor
- Since we're already linearizing, we can use nonlinear costs and linearize those two
    - nonlinear cost on slip to reduce wheel burnout

Orin Control

- In previous years we've reached the cpu limit on the roborio
- This year we run our swerve controller on NVIDIA Jetson Orins, which control the motors via CANivore
    - Still has enable signal from roborio



Pose Estimation

Odometry: "Dead Reckoning"

- For our swerve controller to work properly we need a good estimate of our robots state
  - Essentially our odometry
- Tried two implementations for this - a simple average on the swerve encoders and a nonlinear kalman filter on the sensors of our robot.

Naive Estimator:

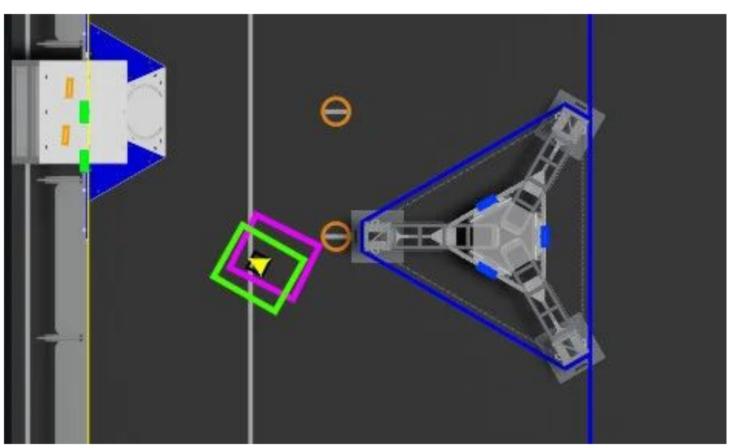
- Calculates the effect that each modules velocity has on the velocity of the robot, and averages them.

Extended Kalman Filter:

- Uses a mechanical model of how the swerve drive moves in order to estimate the velocity of the robot.

Global Localization:

- Global Localization is essentially the process of estimating your location relative to the field coordinate system
- This year we tried multiple approaches, eventually deciding on a linear kalman filter, which blends together our vision with our Dead Reckoning odometry



Object Detection/Vision

Apriltag Detection:

- We wrote a custom library which runs our Apriltag Detection fully on the GPU
- This lets us achieve speeds up to 240fps on the full resolution of 1456x1088 images from our cameras
- We then use the pose from our four cameras to estimate our position

Object Detection:

- We gathered image data of coral using our own cameras
    - Used autodistill to label the images
- Trained a Yolov8 model to generate bounding box of coral from a camera image
- Projected a ray onto the floor to calculate the distance of the coral. (We assume a y-offset for the coral)

Autonomous

Path Generation:

- Uses an algorithm based on “Lipp, T., & Boyd, S. Minimum-time speed optimisation over a fixed path” to optimize our control given our own swerve physics model over a given path (in this case a spline)

Path Following:

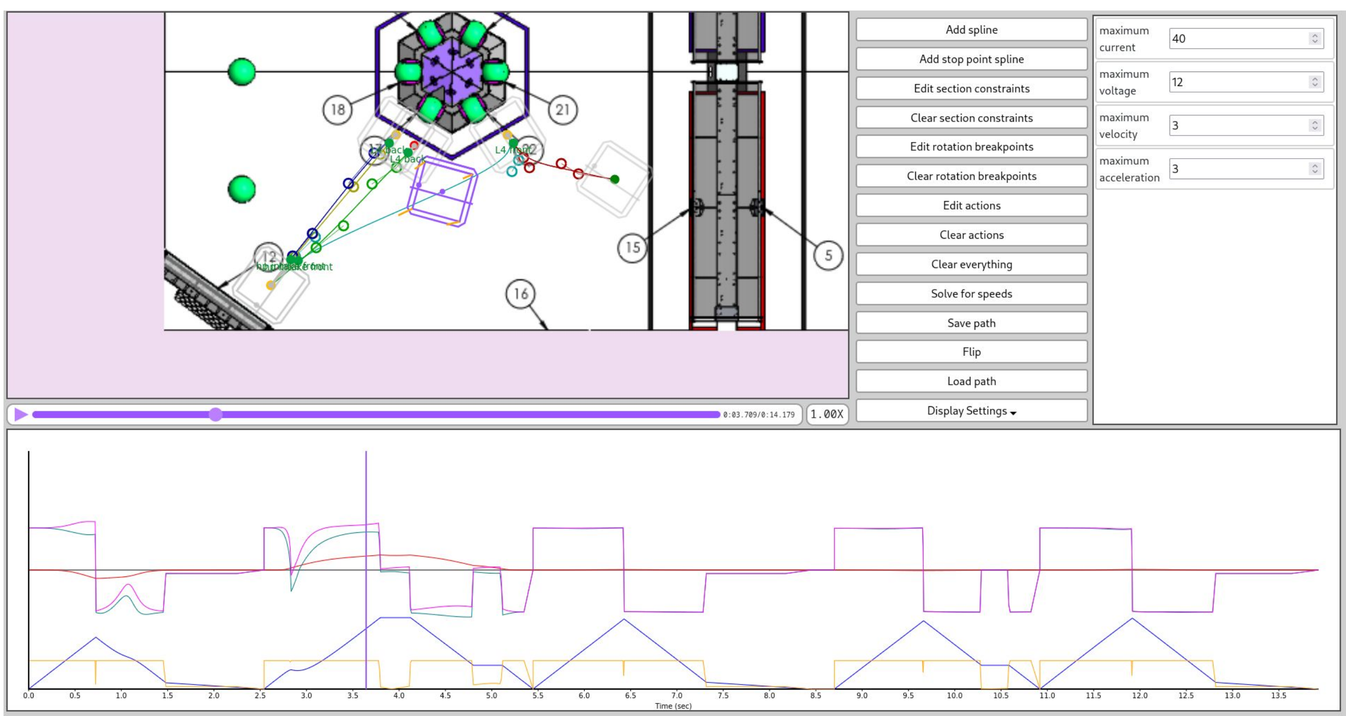
- PD loop added to velocity output(given from above)
- also has a loop with the error, 5 timesteps in the future to help with 2nd order dynamics(acceleration)

Auto Align:

- Auto align to reef button
- PD loop to control for velocity based on current pose and goal pose
- Get current pose from localizer
- Calculate goal pose based on
    - Closest apriltag pose (transformed to robot coordinates)
    - Theta of the robot vs the tag to avoid unnecessary turning (superstructure can score on either side of the robot)
    - Constant offsets for placing left/right and maintaining proper distance from reef

SplineUI

- Based on previous tank SplineUI in python
- How it works:
  - Six-point splines are used to define the path of the robot
  - Movement optimization code calculates the fastest path between points (limits on voltage, current, velocity, and acceleration)
  - Actions can be placed along the path, setting superstructure goals directly from the UI
  - Paths are automatically flippable (top to bottom of field) and the robot automatically accounts for which alliance it's on when performing autos.



3-piece auto in the UI

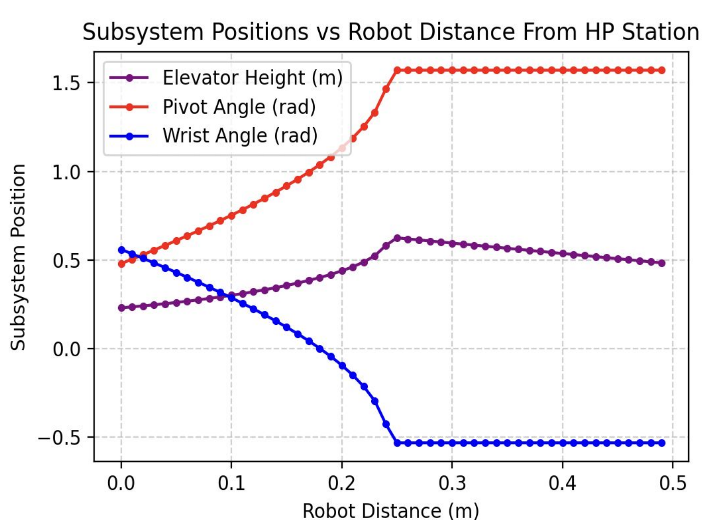
Superstructure

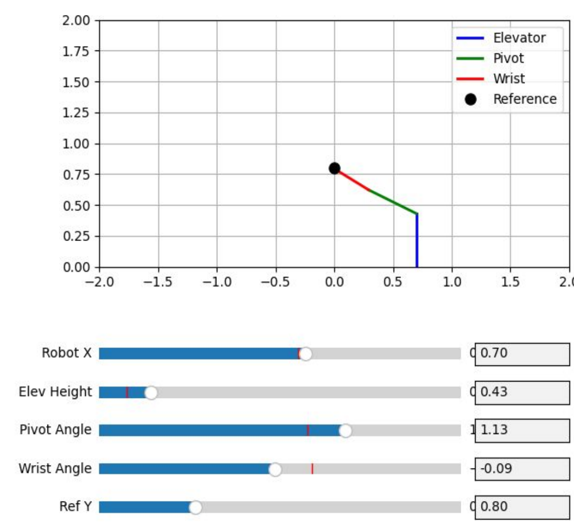
Multiple subsystems/DOFs

- Motion profiles written in python for all subsystems
- Must properly control the Elevator, Wrist, and Pivot
- Optimal control (LQR) is used along with a Kalman Filter based position estimation in order to accurately control each subsystem
- Superstructure logic to avoid collisions with reef
- Precise control using potentiometer and absolute encoder data estimated using kalman filter

Inverse Kinematics

- Inverse kinematics: process of obtaining joint angles from known coordinates of end effector
- By positioning the end effector using localization data, we are able to calculate HP intake and reef scoring set points



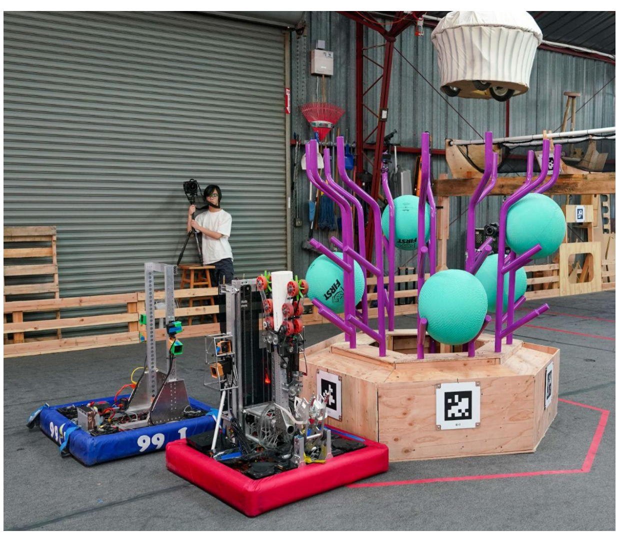


Competition

Drive Practice

Goal: Dedicate at least 15 hours per week to drive practice starting three weeks after kickoff.

- Software drive practice: tuning autos, debugging, tuning localization and auto-align.
- Driver practice:
    - Perfecting efficient coral scoring
    - Optimizing human player coral feeding
    - Emphasizing communication between our driver and operator
    - Practiced call-outs on and off the field
    - Attempted to re-create a competition environment to make sure they would be heard
    - CG control
    - Defense and avoiding obstacles



Scouting

971 Scouting App

- Tracks cycle times based on when buttons are pressed
- Simpler UI allows scouts to focus more on the game, and less on where to press

Scouting Team

- Pairing up scouts to increase our data accuracy
- 80% data scouts, 20% note scouts
- Prediction games: students are incentivized to scout accurately so they can use their data to correctly predict future match outcomes

Tableau

- All students can view all scouting data in tableau, and see how it is used to make informed choices in strategy and our picklist