Technical Binder: Gillbert

2025 FIRST Robotics Competition Season

REEFSCAPE

3128 aluminum narwhals

Table of Contents

Analysis

Game Analysis and Strategic Decisions

Immediately after Kickoff, we realized the vast number of strategic opportunities presented to us with Reefscape.

Strategic Questions

- Questions:
  - What will the optimal structure of a playoff alliance be?
  - How viable is defense at the opponent Coral Human Player Station?
  - Will robots collide with each other while climbing?

Robot Skills/Functional Requirements

- We created a list of priorities based on our initial analysis of the game, leading to a list of overall season goals:
  - A low center of gravity
  - Modular
  - Easily repairable
  - Consistent performance (reliable mechanically and wiring-wise)
- From this priority list, we created functional requirements that our technical departments would use to aid robot design

<table>
<tr><th></th><th>Canadian Pacific</th><th>San Diego</th></tr>
<tr><td rowspan="7">Movement</td><td colspan="2">Drive omnidirectionally</td></tr>
<tr><td colspan="2">Smooth movement (<<1% chance of DCing)</td></tr>
<tr><td colspan="2">Able to move from Human Player Station to Reef Protected Zone within 3 seconds</td></tr>
<tr><td colspan="2">Fully replace wheels in 7 minutes</td></tr>
<tr><td colspan="2">Low CoG</td></tr>
<tr><td colspan="2">Can align with branch in 2-3 seconds</td></tr>
<tr><td colspan="2">Can align with Human Player Station in 1-2 seconds</td></tr>
<tr><td rowspan="3">Auto</td><td colspan="2">Can leave the robot starting line autonomously</td></tr>
<tr><td colspan="2">Can drive to trough and score preloaded Coral into a Reef autonomously</td></tr>
<tr><td></td><td>Can score preloaded Coral and HPS Coral into trough/L4 autonomously</td></tr>
<tr><td rowspan="2">Intaking</td><td colspan="2">Robot can intake Coral from Human Player Station in ~1 second</td></tr>
<tr><td colspan="2">Robot can intake Algae from ground</td></tr>
<tr><td rowspan="3">Control</td><td colspan="2">Robot can knock off Algae from the Reef without DCing</td></tr>
<tr><td colspan="2">Coral and Algae does not fall out of robot when colliding with defending robots</td></tr>
<tr><td colspan="2">Coral does not fall into the robot in a place that it cannot be removed from</td></tr>
<tr><td rowspan="5">Scoring</td><td colspan="2">Robot can extend and score Coral from inside robot into L1 in 1 second (first placements 12/12, second placements 8/12)</td></tr>
<tr><td colspan="2">Robot can extend and score Coral from inside robot into L2 in 2 seconds 19/20 times</td></tr>
<tr><td colspan="2">Robot can extend and score Coral from inside robot into L3 in 2 seconds 19/20 times</td></tr>
<tr><td colspan="2">Robot can extend and score Coral from inside robot into L4 in 2 seconds 19/20 times</td></tr>
<tr><td colspan="2">Robot can score Algae into Processor in 1 second 20/20 times</td></tr>
<tr><td>Climb</td><td>Can align and clear the ground (5/5) of the time on the Deep Cage within a time frame of 15 seconds</td><td>Can align and clear the ground (5/5) of the time on the Deep Cage within a time frame of 15 seconds</td></tr>
</table>

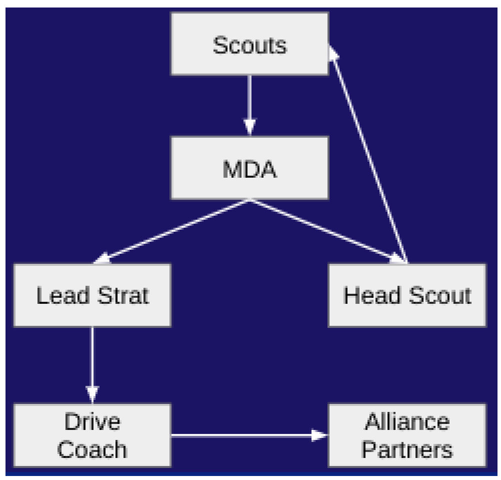
Technical Recommendations

- The hypothetical 2 point difference between barge and processor scoring of algae is too miniscule to be a necessary robot capability.
  - Decided on a ground algae intake with a stick to knock algae off the reef instead of an algae manipulator on the elevator.
- Extremely short cycle paths make driving efficiency a high priority.
  - Decided on using a swerve drive to reduce time for robot rotation and increase movement speed.

Match Strategy and Simulation

Match Strategies

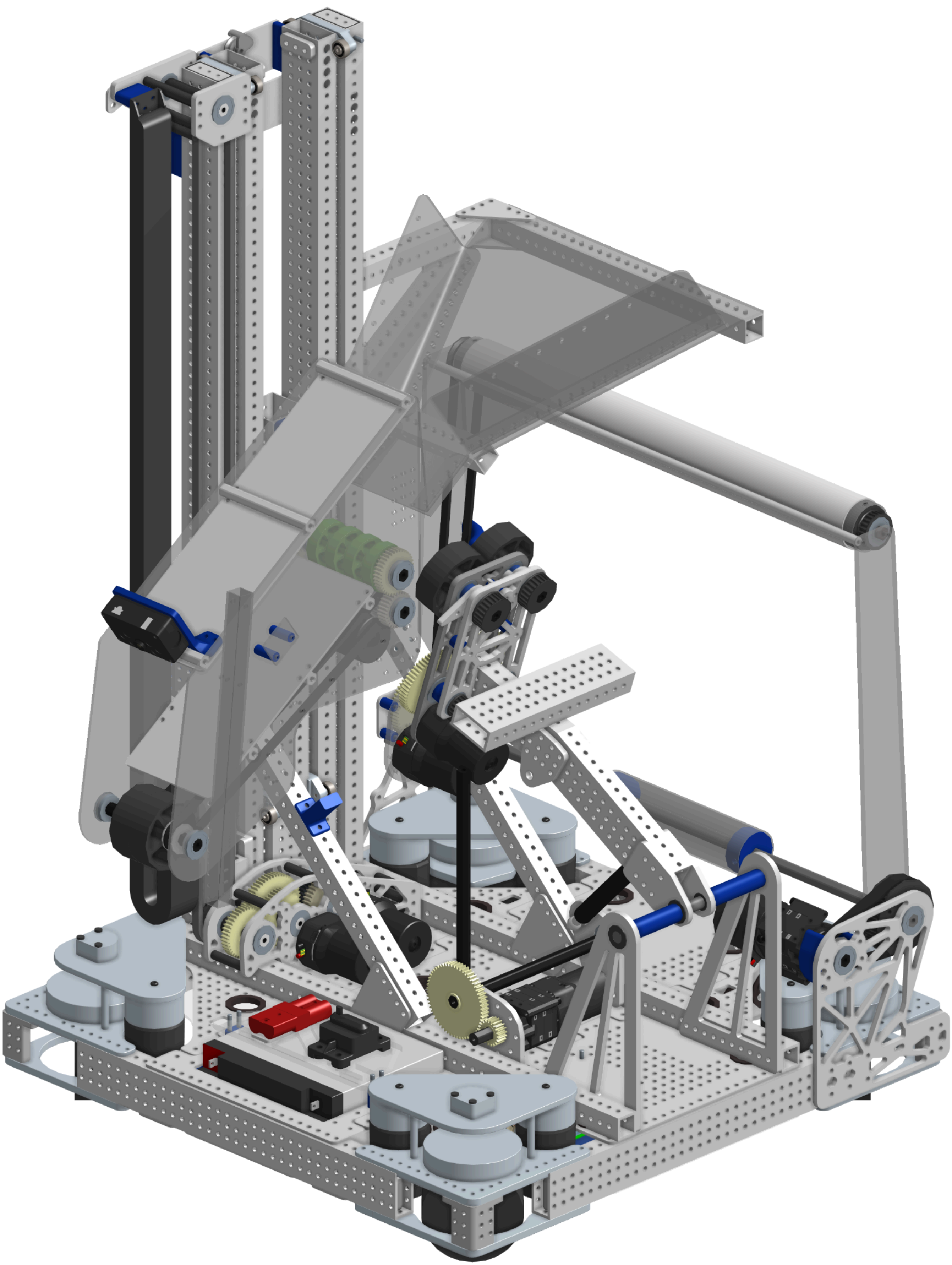
- We have done many activities to improve our understanding of how to create match strategies.
  - Cycle Analysis
    - Drew the paths that robots may take to score during a match.
    - Helped us understand how to efficiently allocate scoring roles and coordinate autos with teammates.
  - Paper Games
    - Practiced writing match strategies for randomly generated alliances based on hypothetical robot archetypes.
  - Pre-regional Events
    - Scouted Week 0 competitions virtually.
    - Helped test our scouting scouting systems, gain valuable scouting practice, and learn insights on how Reefscape matches will play.



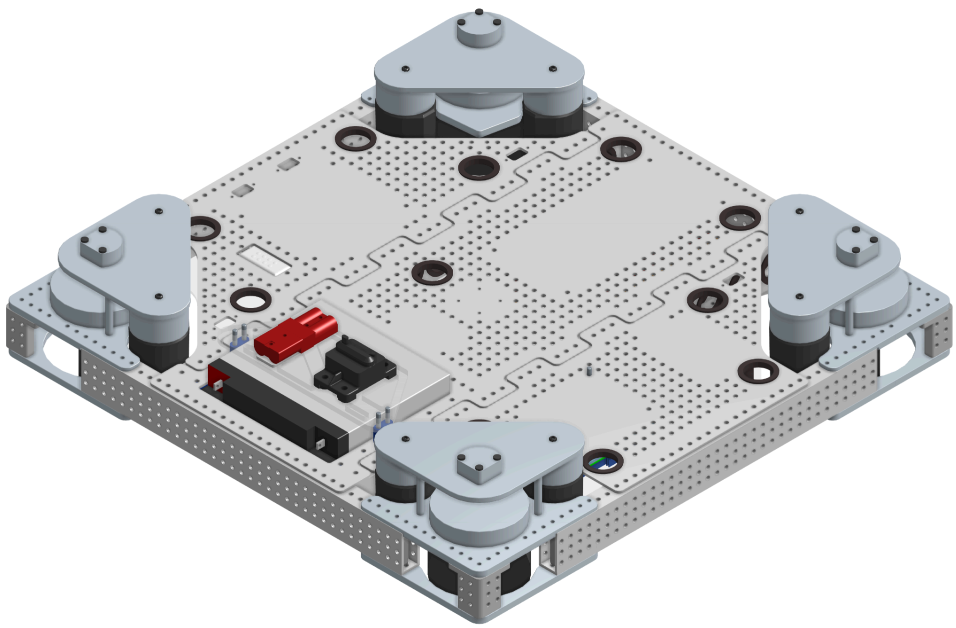
Scouting Systems

- <u>All match strategies are derived from robot performance data gathered by scouts</u>
  - We utilize our scouting system - "Manta" - of three applications to gather, store, and analyze data effectively. <u>*(see scouting systems whitepaper for more information)*</u>
  - Leadership within Strategy (Lead Strategist, Match Data Analyst, Head Scout, Drive Coach) analyzes data and forms match strategies with it.
- We have also created a scouting alliance at all of our regionals to give other teams access to our data visualization and collection tools, in return for greater data redundancy and accuracy.
  - Currently working with 1 other team at Canadian Pacific Regional

Robot Design



Drivetrain



Swerve drivetrain quickly maneuvers the robot around the field

MK4i Swerve Modules

- L2 gear ratio (15.5 ft/s)
- 4” Vex Grip Lock wheels
- Kraken x60s for drive and steering

Chassis

- 26.5” x 26.5” frame
- Brainpan for easy electrical access
- Tapped steel plates for mounting

Bumpers

- 4 segments for easy removal
- 1/2” HDPE backing
- Layered Foam to absorb impacts

Elevator

The Elevator raises the Manipulator for swift scoring on all Reef levels.

Elevator

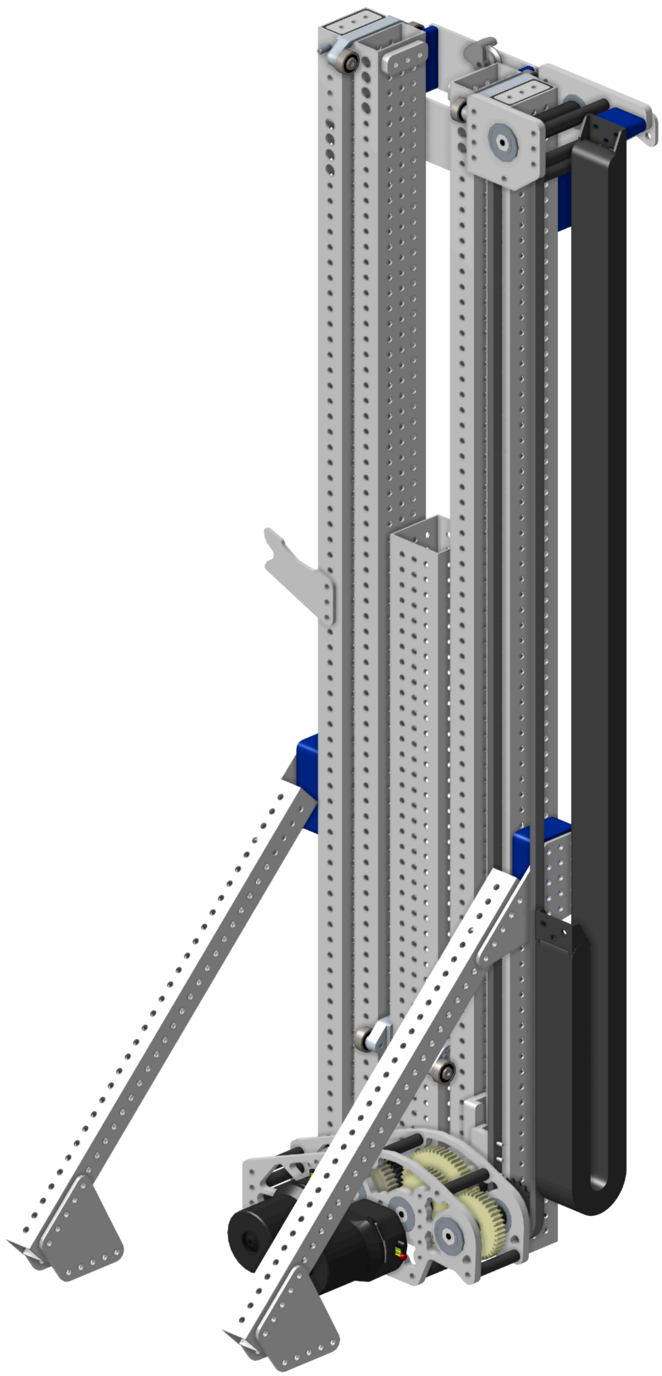
- 2 Vortexes geared 12:1
- 2 x 1 x 1/16” frame supported by A-Frame and tensioned Dyneema
- Open top design allows carriage to extend above crossbars.

Carriage

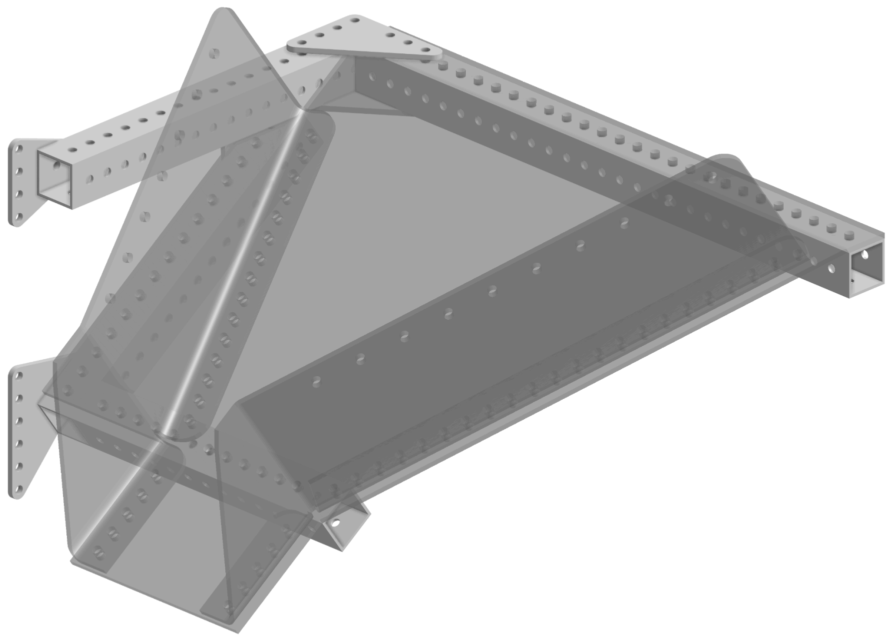
- Moves full range of motion in under 1 second
- 2 x 2 carriage allows for elevator to fit in between swerve modules

Electronics

- Wires run up through aluminum channels for protected routing
- WCP cable chain on side to enable smooth wire motion



Funnel



The Funnel serializes Coral from the Station and feeds it into the Manipulator

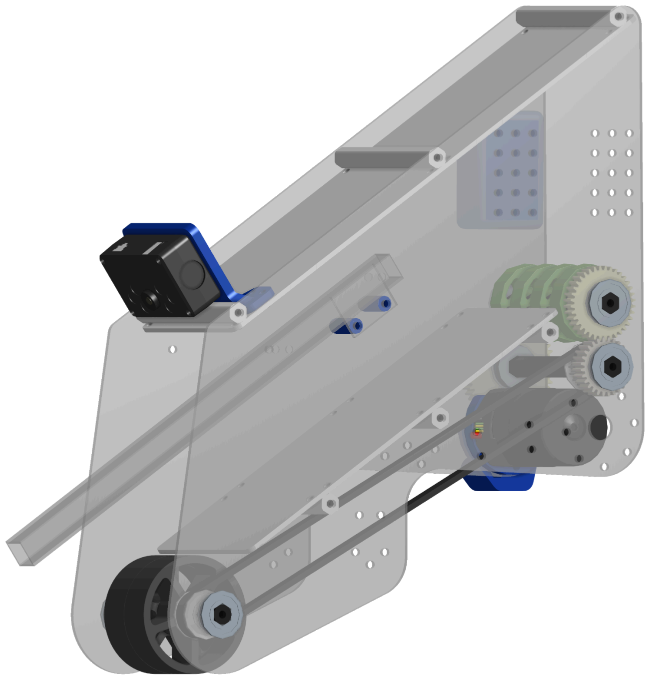
Frame

- 1/16” boxtube for lightweight yet stable construction
- Attached to elevator to remove unnecessary supports
- Bent polycarbonate gusset for support

Funnel

- Bent 1/4” polycarbonate
- Compliant 1/8” HDPE to help with centering
- Completely passive design to reduce weight

Manipulator



The Manipulator receives Coral from the Funnel, and scores it on all Levels.

Rollers

- 2″ Andymark wheels for intaking
- 4″ Thrifty wheels for outaking
- Wheels geared opposite for sensor-less game piece retention

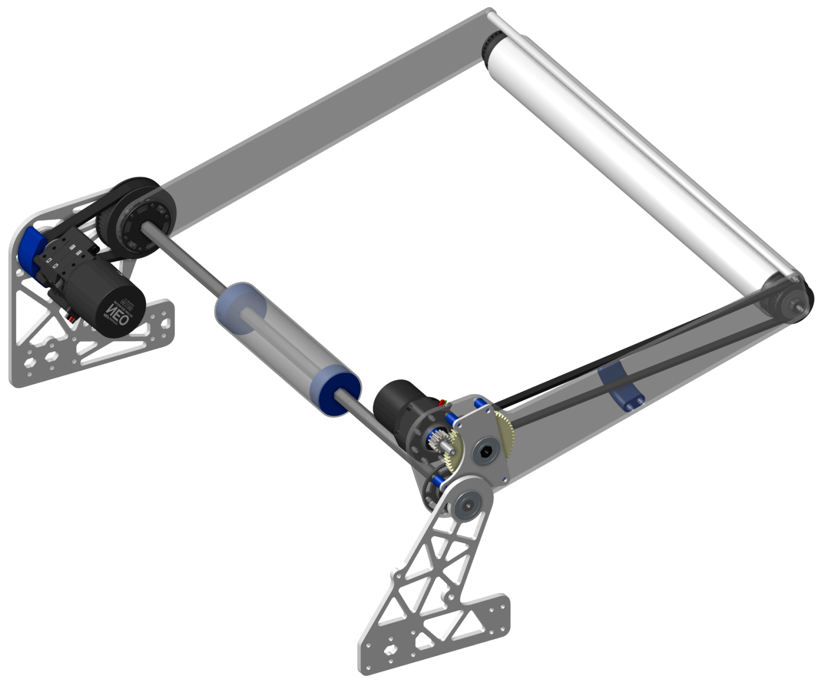
Algae Removal

- 3/4″ polycarbonate tubing
- Surgical tubing for passive deployment
- Polycarbonate latch secures knocker inside robot at the start of the match

Construction

- 1/8″ delrin side plates
- 1/8″ aluminum scab plate for added support
- Limelight 3 for reef auto-align

Intake



The Intake gets Algae from the ground and scores it in the Processor

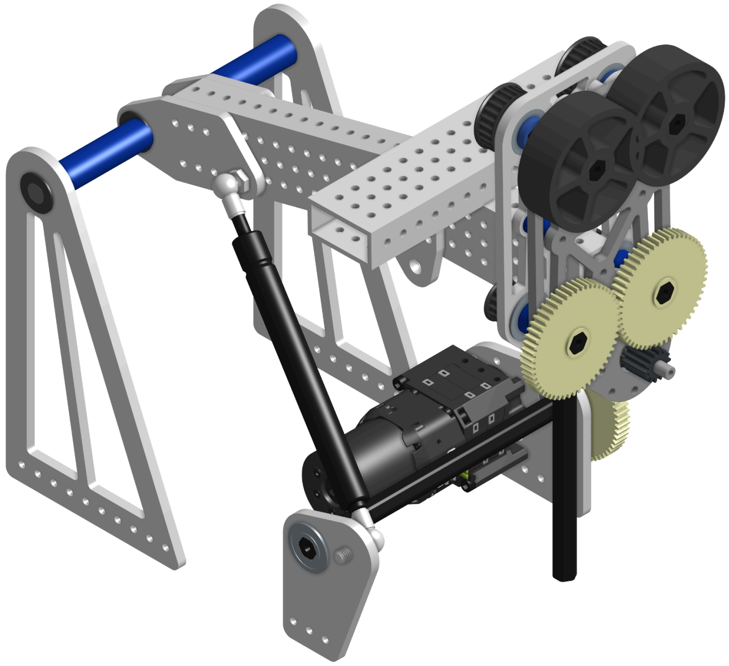
Roller

- 1 Vortex geared 5:1
- 2" OD 1/8" wall polycarbonate tube with silicone material
- Dead-axle pulleys eliminates need for stand-offs

Pivot

- 1 Vortex geared 48:1
- 1/4” aluminum pocketed side plates
- Belt-driven pivot to reduce maintenance

Climber



The Climber pulls down on the cage to lift the Robot off the ground. Dyneema rope around a hex jackshaft winches down the Climber in ~0.5 seconds

Pivot

- 1 Vortex geared 45:1 to prevent backdriving
- Dyneema wound around hex to winch head down
- Head passively raised with 30 lb gas shock
- Steel dead axle to support robot weight

Head

- 1 vortex geared 6:1
- 3” compliant wheels for repeatable and fast alignment
- Passive one-way hooks to lock onto the cage
- Silicone tubing on 2x1 to prevent slippage

Software

State Machine

Subsystem State Machines: Each subsystem is a state machine consisting of different mechanisms with unique states and a transition map for transitions between states.

NAR_Subsystem

- Interface that specifies the functionality and expectations each subsystem we create on our robot should have

NAR_PIDSubsytem

- Our custom subsystem class that controls PID subsystems and has specified other quality of life additions

Position/Velocity/Voltage Based Subsystems

- Classes shaped around their specific purpose
- Position -> elevator, pivot, climber
- Velocity -> shooter
- Voltage -> rollers, manip
- Titled as a mechanism

Finite state machine subsystems

- A class that includes all necessary mechanisms to form a complete functioning subsystem on our robot
- Includes all states for each mechanisms and respective transitions between

Transitions + transition maps

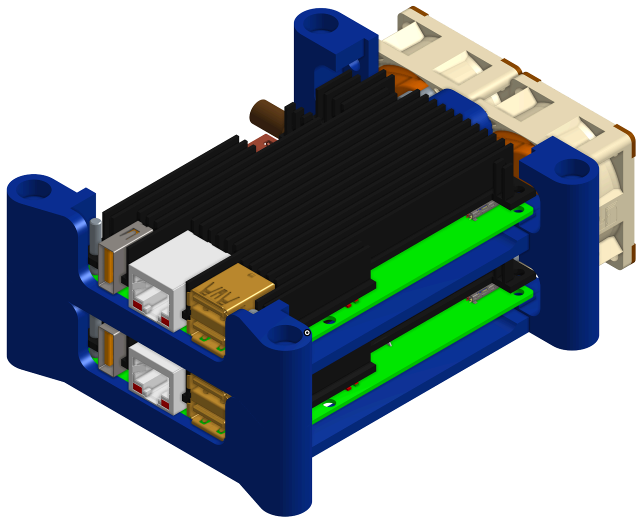
- Maps a “state” of the subsystem
- Transitions allow the subsystem to go from state to state in proper fashion

Robot states + state machine
Subsystem States + state machine
Subsystem mechanism

Default and exclusive toggling paradigm

- Certain states, called exclusive states, are only able to be accessed after reach a state before hand, known as default
- The pairing of these default -> exclusive states allow a clear visual of how our robot functions

Vision and Odometry



Vision Setup:

- Two OV2311 cameras running off an Orange Pi 5 with photonvision:
- Two cameras on the swerve drive with 125° and 150° FOV M12 lenses to maximize April Tag visibility
- Encased in aluminum heat sink
  - Increases time to throttle by ~5 minutes
- Have two large Noctua fans dynamically cooling the boards
  - Increase time to throttle to over ~30 mins

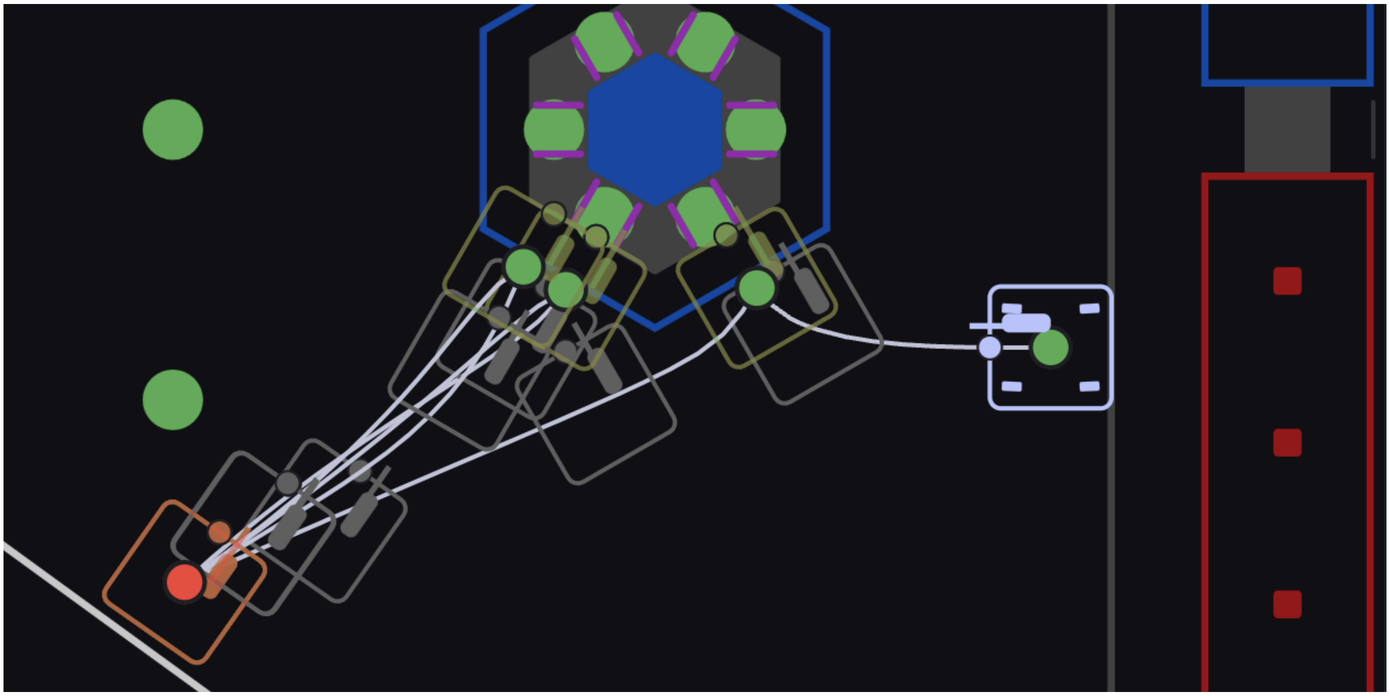
Pose Estimation:

- **Tag Filtering -** filter out tags based on ambiguity, distance, and ID
  - Ambiguity - number from 0-1 representing pose inaccuracy; reject any above 0.3
  - Distance - filter out tags at far distances ~4.0 meters and beyond
  - ID - reject tags discovered through testing that are detrimental to our positions
- Input stabilization with gyro - rely on gyro values to correct vision input estimates
  - Inputs experimentally determined to be unreliable
  - Filters inputs, and utilizes robot angle to calculate resulting data

Pose Utilization:

- **Auto-alignment** to field elements (reef, HP station) - uses estimated pose to line up to field elements autonomously
- Autonomous mode - pose estimates used to increase accuracy in auto
  - Guarantees consistency no matter the field conditions

Autonomous Control



System Check

- Single button press to <u>run multiple checks for each subsystem</u>
- Used to <u>efficiently test all systems in between matches</u> to ensure functionality
-

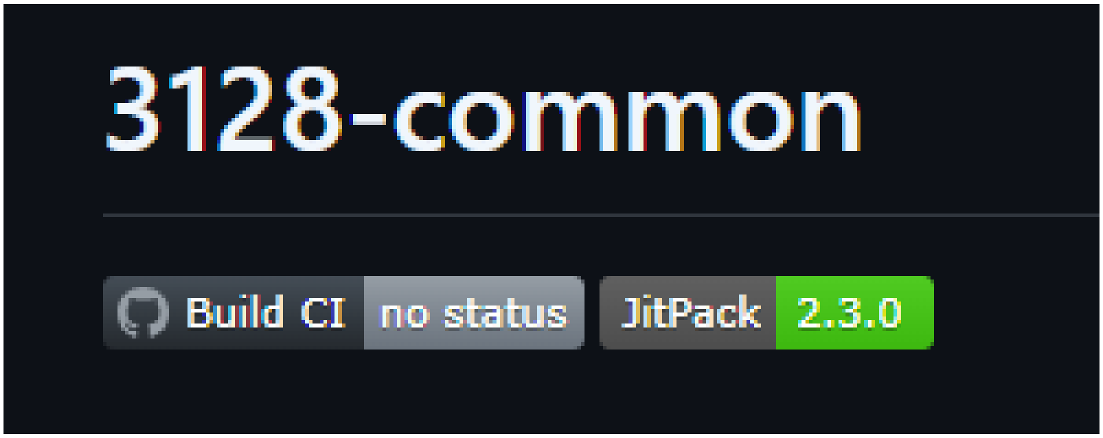
Auto Movement

- Auto-alignment to reef scoring elements and the human player station
- Finds the closest reef or human player station and automatically travels and aligns to it for scoring or picking up pieces

Algae / Coral/ Reef Detection

- Utilizes <u>limelight with a custom ssd-mobilenet</u> to detect <u>both corals and algae.</u> This was developed to be compatible with the yolo and ssdmobilenet configuration for redundancy. Objects are filtered by confidence, area, and position of the object. <u>Trained on colored images under various light conditions to minimize overfitting of kernels to specific luminance values</u>.
- To align with the center of the reef to score coral, we utilized another <u>custom-trained model for accuracy</u>. To improve generalization, hues were altered along with various orientations and lighting conditions.
- In case of underfitting/ low generalization, we employed the usage of a secondary <u>Python script</u> to filter the coral rods based on the color, and fine-tuned HSV values during calibration to ensure consistency.

Common Repository



- Github repository https://github.com/Team3128/3128-common
- Repository to store code reused each year
- Published using jitpack
- Similar usage as other venderdep libraries, ie. Phoenix library and REVLib
    - Easy updating and download
    - Changes can be distributed to all users easily and efficiently
- Centralized code structure
- Functionalities:
    - Swerve Base
    - Subsystem Templates:
        - NAR_PIDSubsystem (Custom PID Subsystem implementation)
        - FSM Subsystem Base
        - Position Subsystem Base
        - Velocity Subsystem Base
        - Voltage Subsystem Base
    - Transitions and Transition Maps
    - Control Systems (PID, Feed Forward, Trapezoidal PID, etc.)
    - Motor Wrappers
    - Shuffleboard Interface
    - Narwhal Dashboard Java Web Socket
    - Vision Processing
    - SysID
    - Controllers (XBox, Button Boards, etc.)

- **Submodules:**
    - Git submodules are used for efficient access to the common repository within a separate repository for our robot code. Allows for easy access to view and change code within the Common Repository

Narwhal Dashboard

Narwhal Dashboard is a custom web-based dashboard made to be a faster, more customizable, and flexible version of the WPILib dashboards.

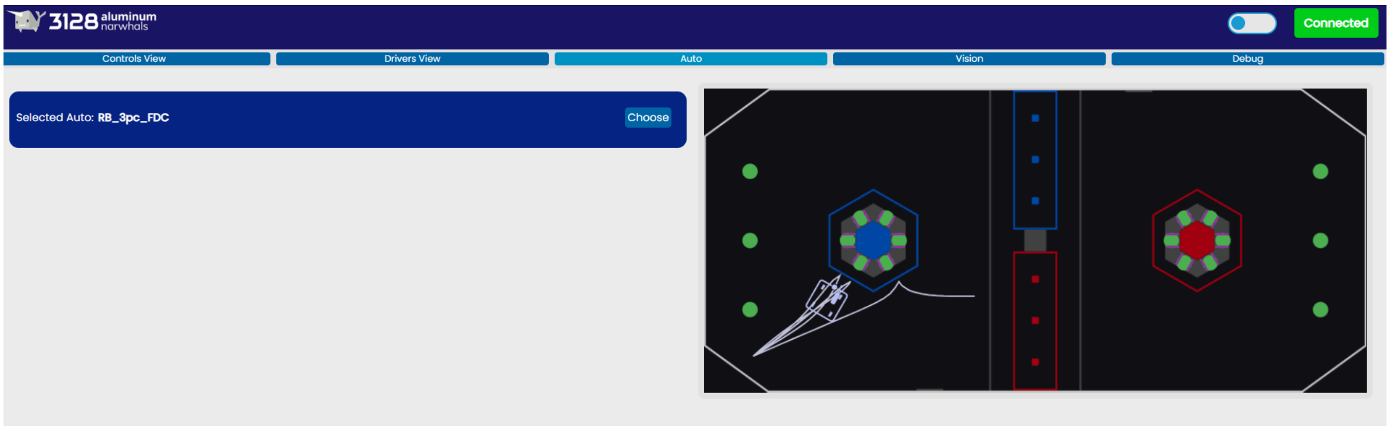
Purpose:

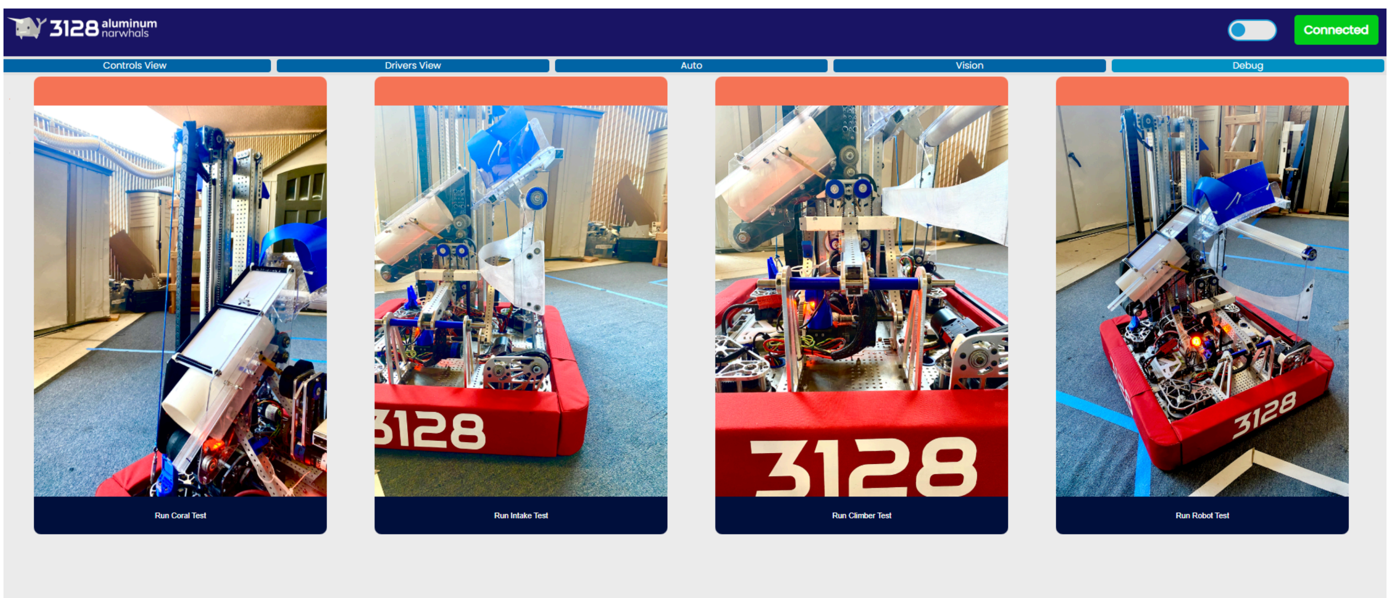
- Offload data processing from robot to driver station computer
- Useful tool for drivers and programmers during matches and while debugging and testing

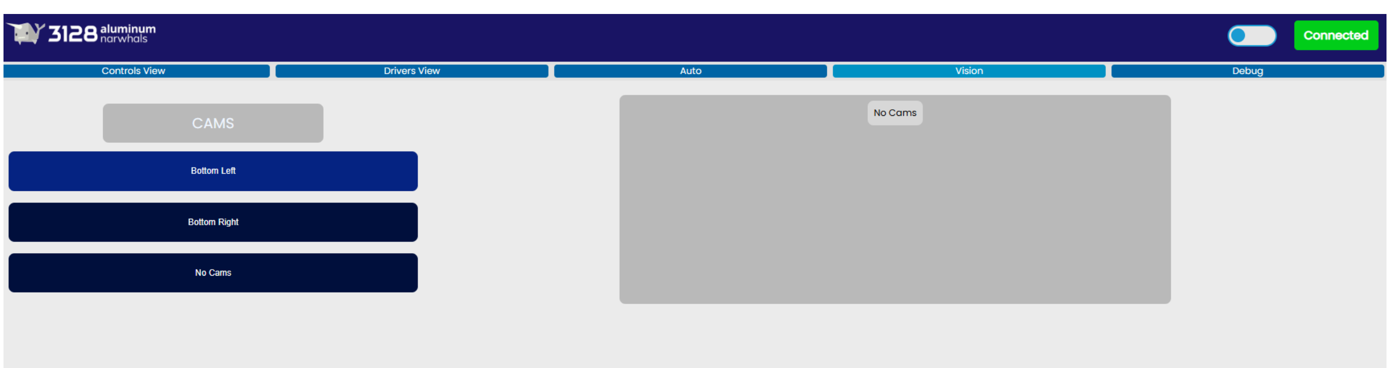
Implementation:

- Currently written in HTML/CSS/JS in the React JS framework
- Connects to the RoboRIO via a Java web socket server to revive .json file of tagged information from Java robot repository
- Compatible with robot inputs through annotations processor
- Modularity allows for flexibility and ease of testing

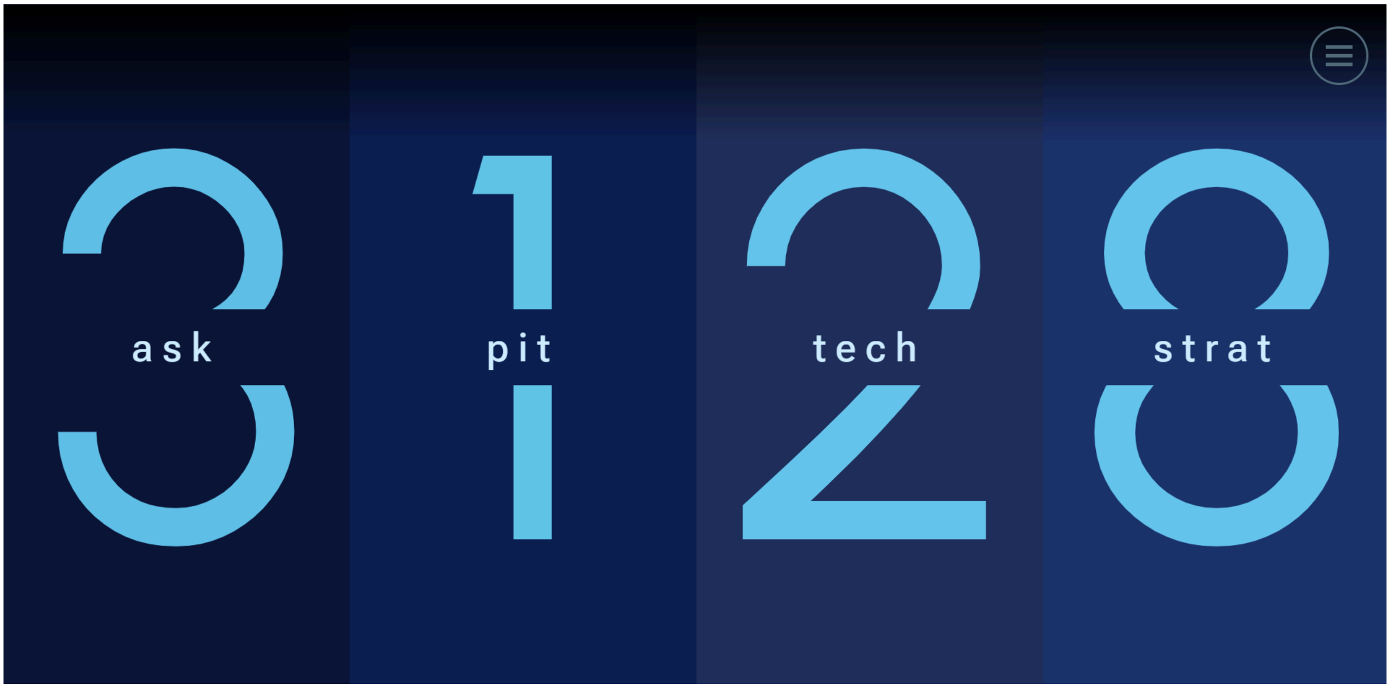

3128 aluminum narwhals
Connected
Controls View
Drivers View
Auto
Vision
Debug
0:00s


3128 aluminum narwhals
Connected
Controls View
Drivers View
Auto
Vision
Debug
Selected Auto: RB_3pc_FDC
Choose


3128 aluminum narwhals
Connected
Controls View
Drivers View
Auto
Vision
Debug
Run Coral Test
Run Intake Test
Run Climber Test
Run Robot Test


3128 aluminum narwhals
Connected
Controls View
Drivers View
Auto
Vision
Debug
CAMS
Bottom Left
Bottom Right
No Cams
No Cams

Odyssey



Overview:

- Hub for our extra projects and for organizational purposes

NARAsk:

- Search engine to filter all documentation on existing code and information in 3128 Controls department including Common, Narwhal Dashboard, Vision systems, Autonomous Controls, Github Practices, etc.
- Built from pure JavaScript
- Proprietary search algorithm and database

NARPit:

- Online battery log for efficiency organization accessible by all members
- Download after each match and competition for analysis on battery optimization

NARTech:

- Online interactive technical binder

NARStrat:

- Our team's scouting systems for match-data analysis and predictions