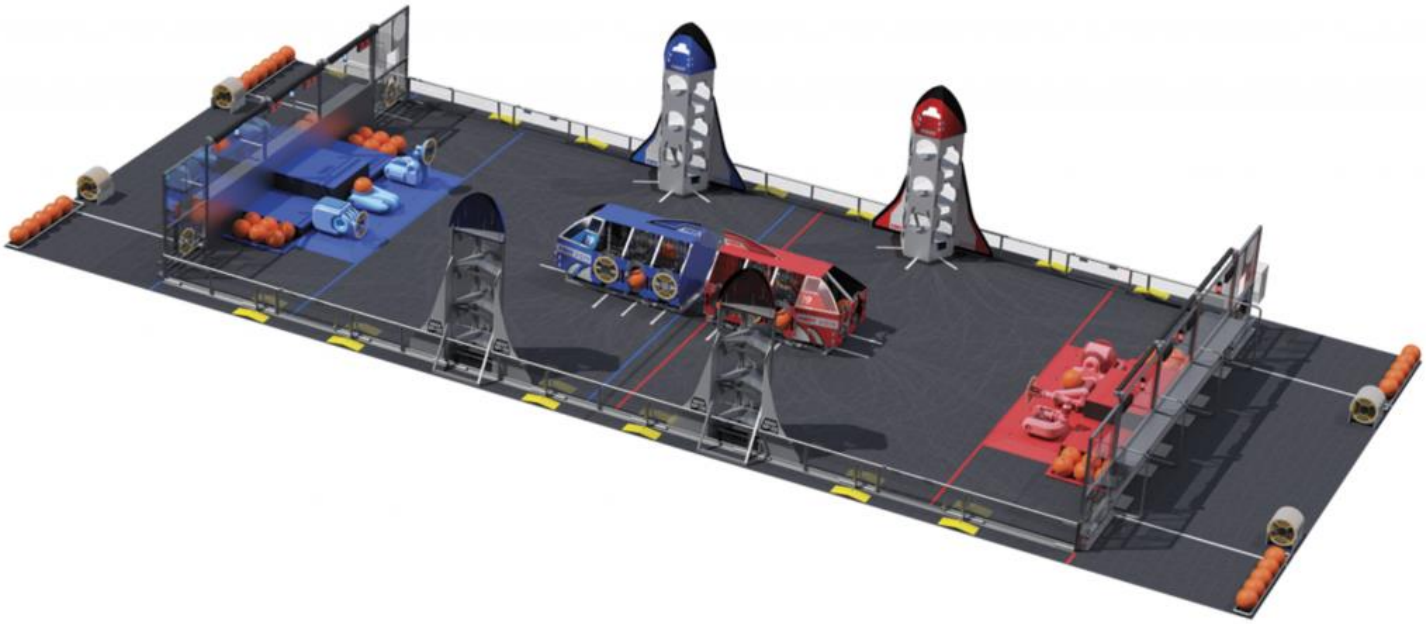
FRC Team 980 ThunderBots

A community robotics team for Burbank/Glendale/North LA County

2019 Engineering Notebook

Game Strategy



Our strategy is to utilize our autonomous capabilities and effective manipulator design to score as many points as possible. We start on Habitat Level 2, where we also return to climb at the end of every match.

In qualifications, we focus on completing one of the two rockets by filling it with both hatches and cargo to gain the Ranking Point.

In eliminations, we shift our focus to the cargo ship and lower rocket levels to maximize the total number of points scored.

This strategy is complemented by our defensive abilities, which we also use to prevent other teams from scoring.

DESIGN REQUIREMENTS

The robot SHALL:

- Be robust and repairable
- Move with enough power to push other robots when playing defense
- Be able to climb to Level 2 at the end of a match

The chassis SHALL:

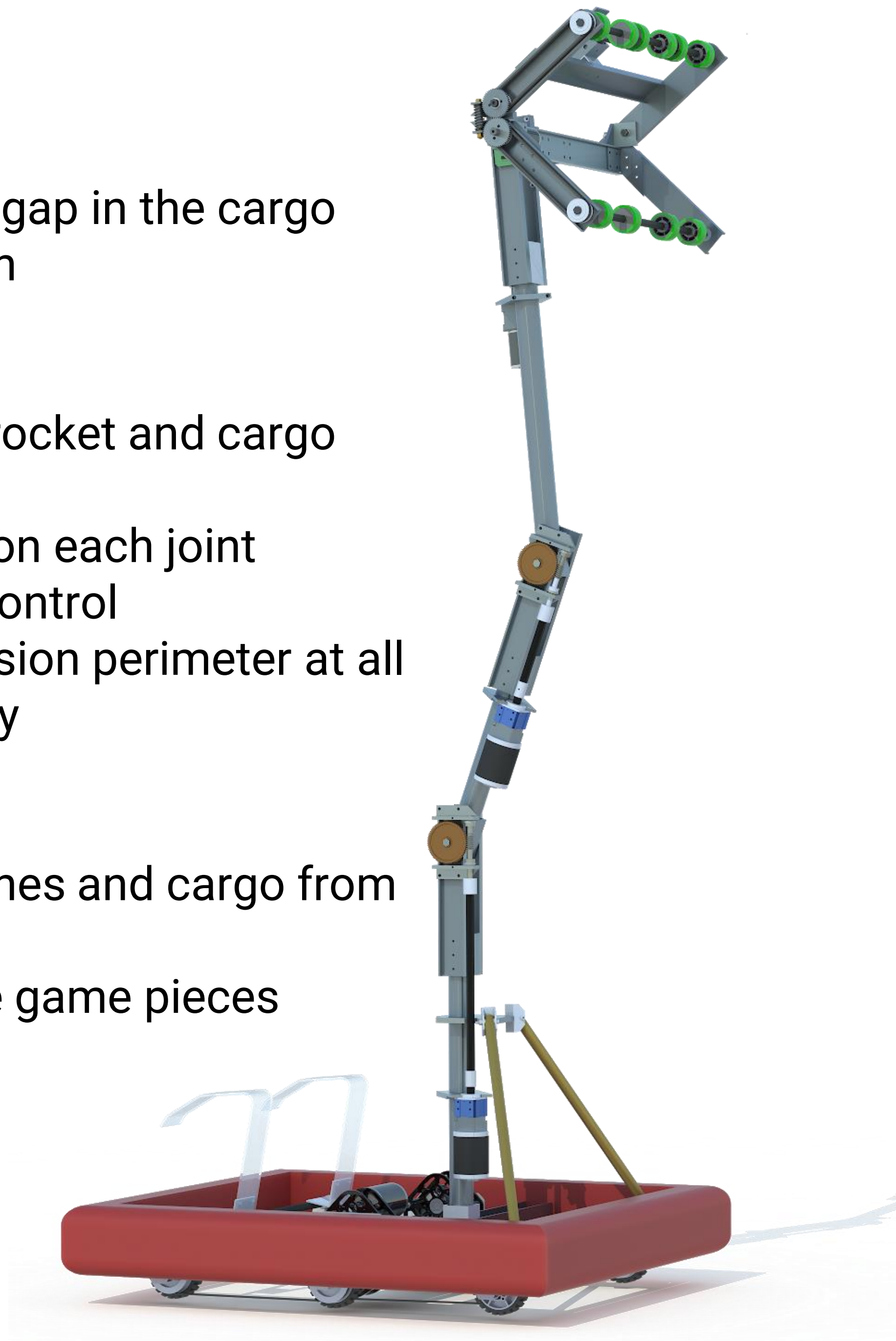
- Be able to fit under the gap in the cargo ship and loading station

The arm SHALL:

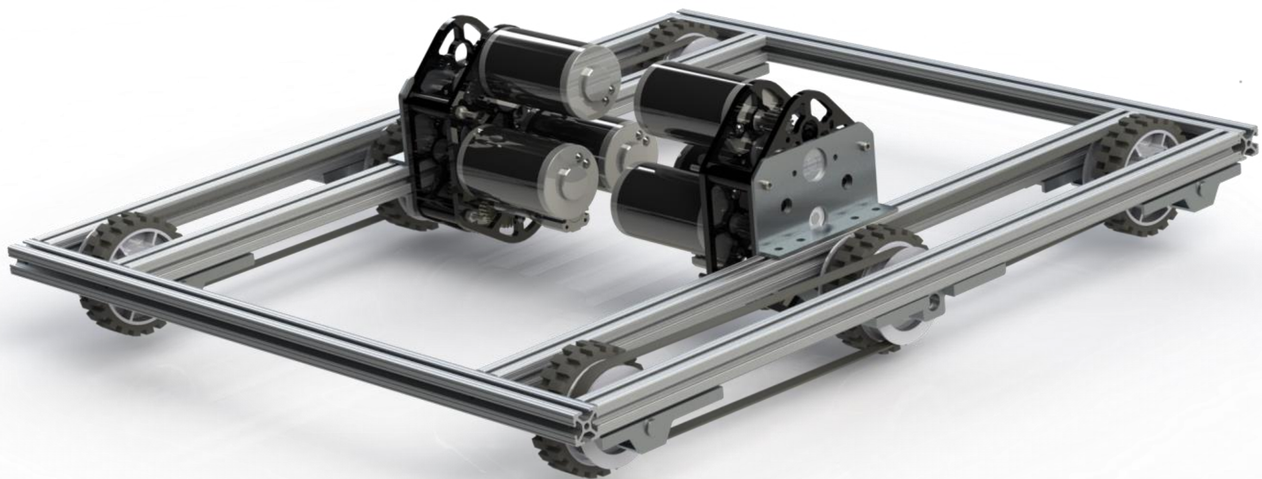
- Be able to reach most rocket and cargo ship positions
- Be able to safely position each joint without direct human control
- Fit within the 30” extension perimeter at all times during match play

The end effector SHALL:

- Be able to pick up hatches and cargo from the floor
- Be able to quickly cycle game pieces

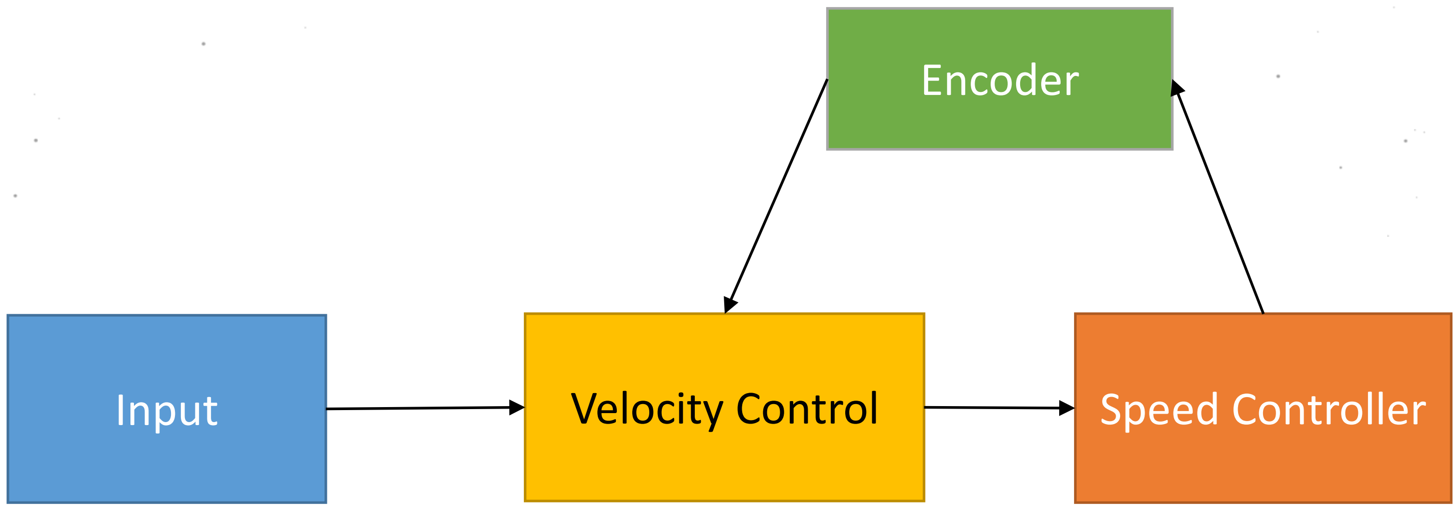


Drive System
Design



- Six-wheeled drive train powered by 6 CIM motors
- Adjustable axle heights in order to maintain ideal traction
- Two-speed drivetrain allows for high-speed scoring and high-power defense

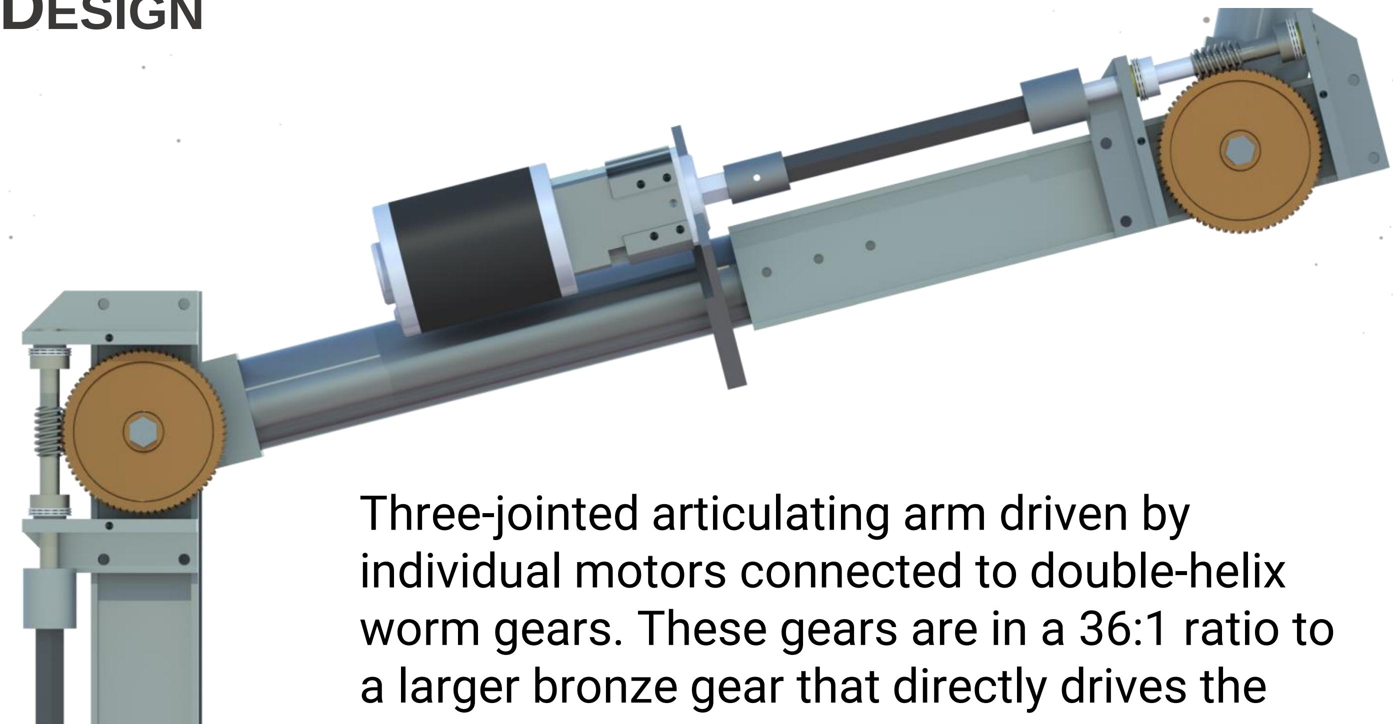
Drive System
Velocity Control



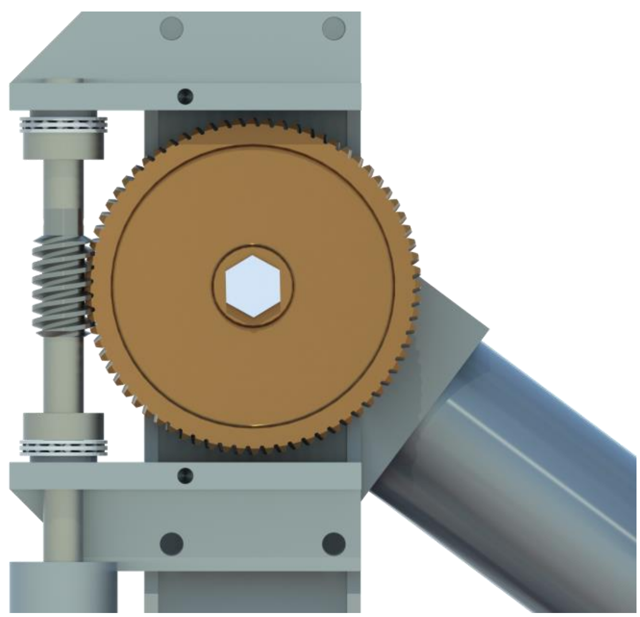
This year, we have implemented a PID-based velocity control algorithm.

Our driver controls request each side of the drive base to run at a specific velocity. Using feedback from the encoders, the algorithm regulates and adjusts the amount of voltage sent to ensure we remain at the desired velocity.

Robot Arm
Design

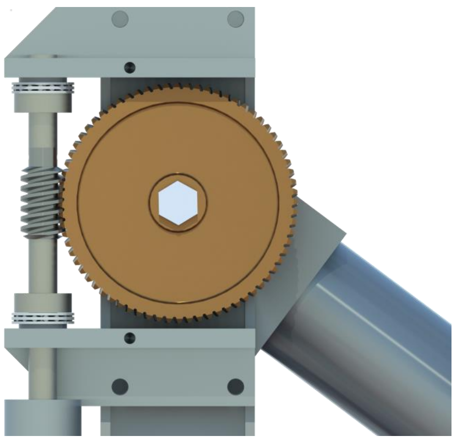


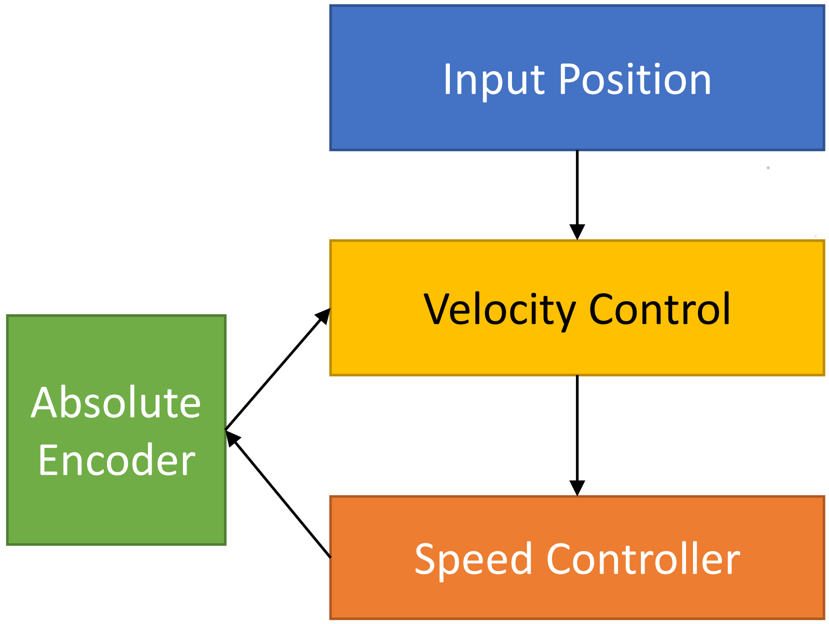
Three-jointed articulating arm driven by individual motors connected to double-helix worm gears. These gears are in a 36:1 ratio to a larger bronze gear that directly drives the joint.



- Double-helix worm gear allows for arm to remain stationary without motor assistance
- Driven by a CIM and Planetary reducer

Robot Arm
Positional Control





To ensure optimal performance, the arm is always controlled autonomously.

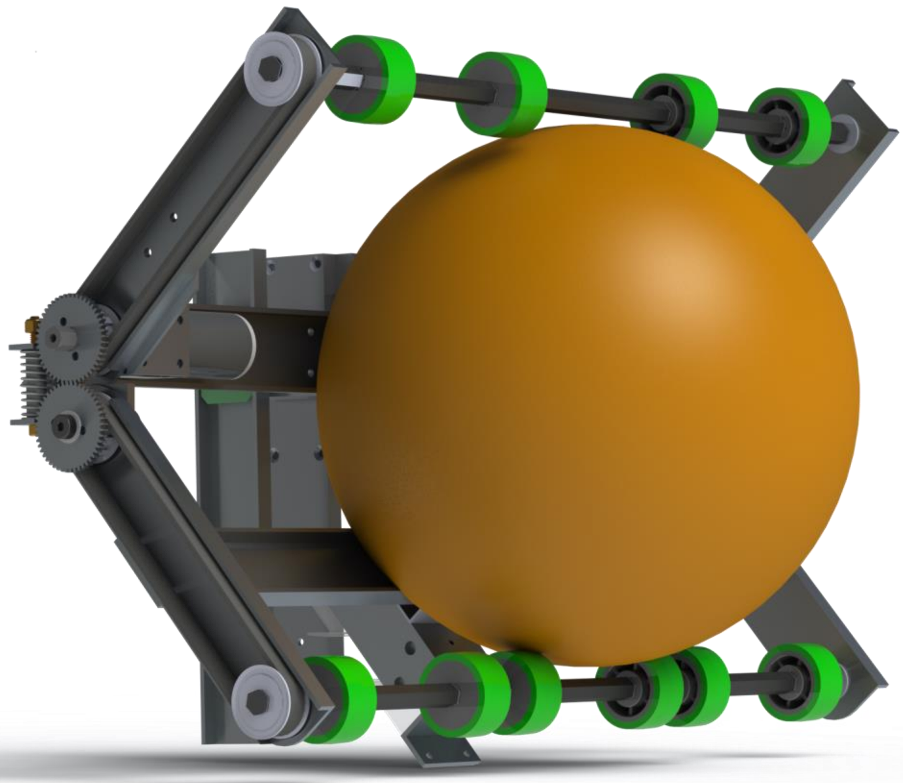
Each joint has an absolute encoder attached that we read the angle of via our Rioduino coprocessor. We additionally calculate the velocity of each joint and use it in a custom velocity control algorithm that moves the joint in a controlled manner.

We have a suite of encoder positions programmed for each joint. When the operator or an autonomous routine specifies a position, we move the three joints simultaneously to reach the desired angles.

Robot Arm
Positions

- Cargo and hatch delivery
- Top, middle and bottom rocket levels + cargo ship
- Floor hatch and cargo pickup

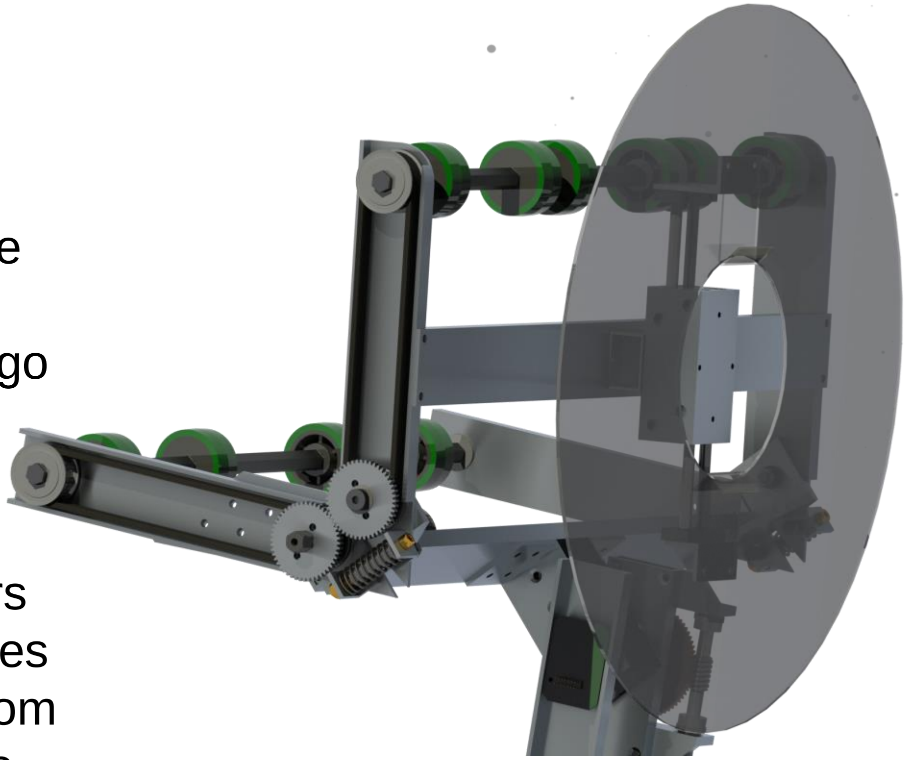
End Effector
Cargo Pickup



- Flexible intake using compression spring mechanism allows capture of cargo with tolerance for varying levels of inflation
- 22” long hex shaft gives greater range of capture and allows for easy depositing
- Intake mechanism driven by BAG motor which powers gear and belt system

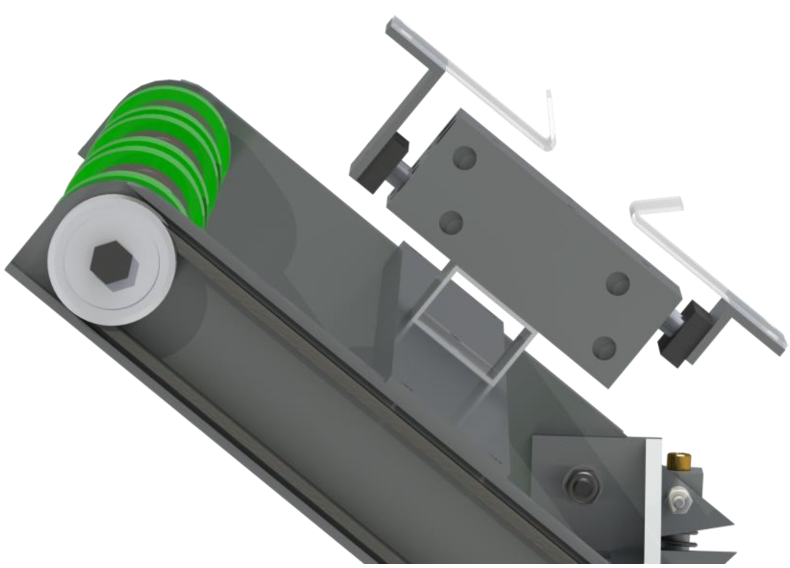
END EFFECTOR

HATCH PICKUP



- Located on underside of end effector, does not interfere with cargo pickup

- Uses two air cylinders with bent Lexan pieces to pick up hatches from ground as well as the stations

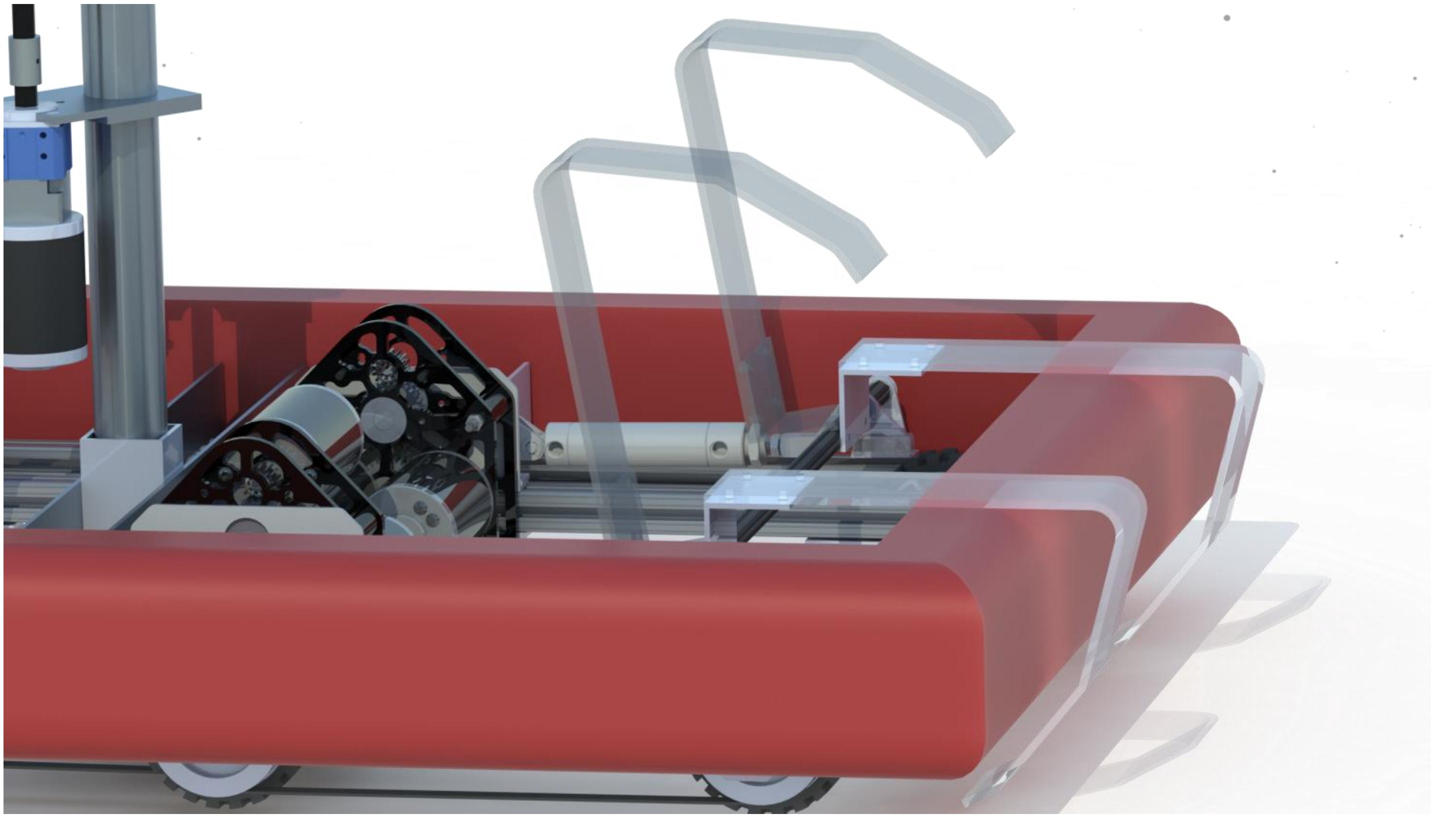


- Lexan hatch pickup pieces easy to modify and replace

- Went through several iterations to maximize ground pickup success

- Pneumatic system ensures secure pickup, transfer, and placement throughout the game

CLIMBING

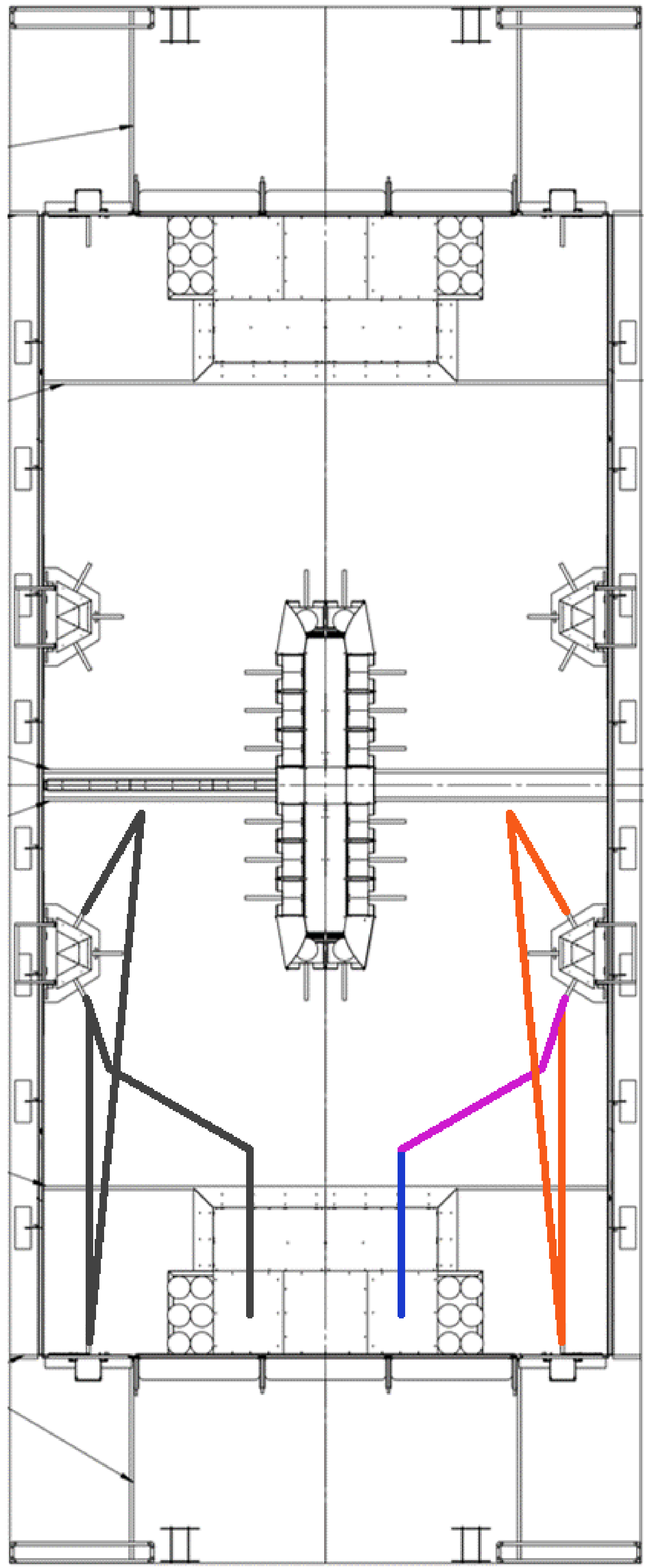


Lightning XVII doesn't climb… It *flies.*

Modeled after trophy trucks, our flying mechanism is comprised of two Lexan skid plates that constantly deflect our robot's forward motion upward and onto Habitat Level 2.

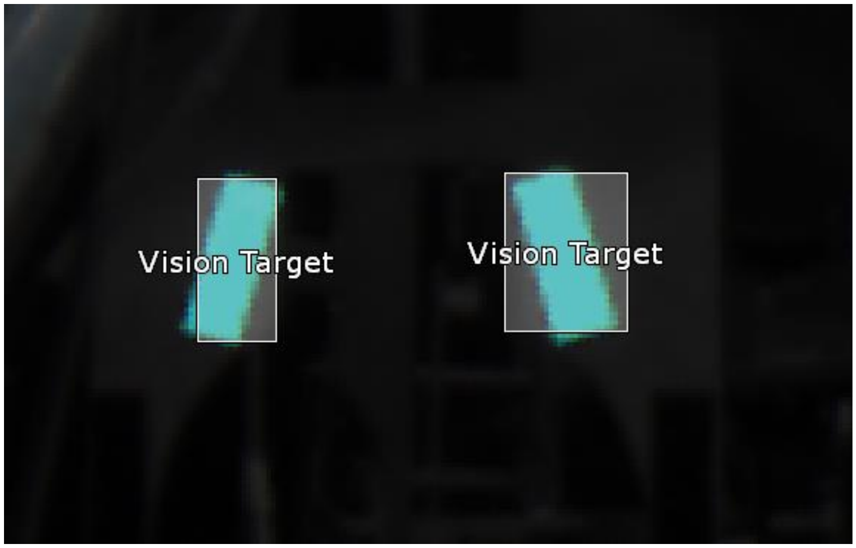
Sandstorm Strategy

Our robot autonomously navigates the Sandstorm, using its onboard sensors to deliver two hatches to the Rocket.



- We start by driving off Level 2 at 10ft/s, then use our IMU to stop once our robot is oriented level with the carpet
- We navigate to the Rocket and the Loading Station using our encoders and IMU for odometry
- When we approach a loading or scoring zone, we use one of our Pixy cameras to line up with the vision targets

VISION SYSTEM



- A Pixy2 camera is used to detect vision targets
  - We interface with it via our Rioduino coprocessor

- Our algorithm finds the two largest segments of retroreflective tape and calculates their width and center position

- We then feed this data into our automated navigation system, which steers the robot towards the targets