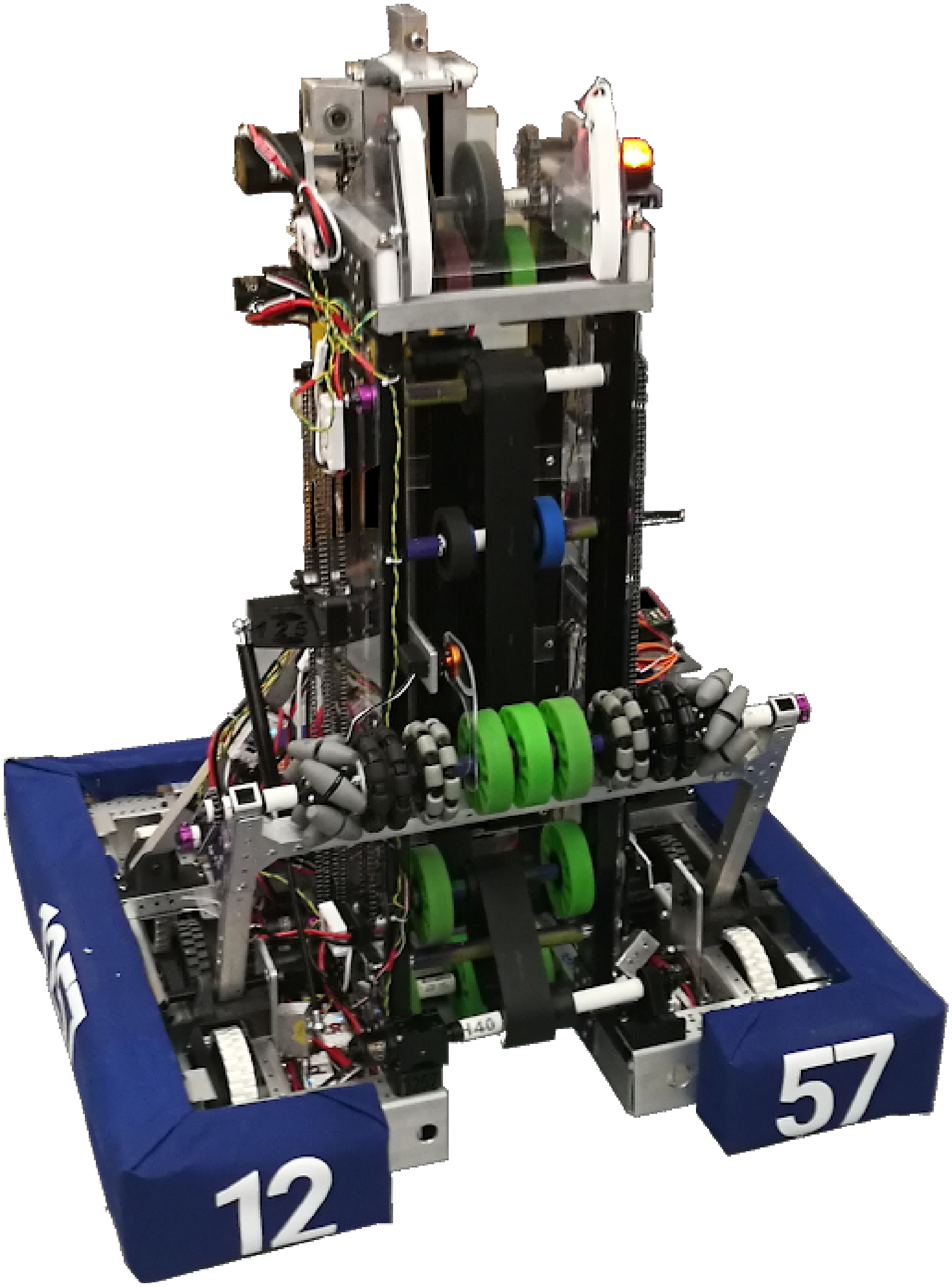


FRC Team 1257 Parallel Universe

2019-2020 Technical Documentation

Last Updated 05 March 2020

Table of Contents

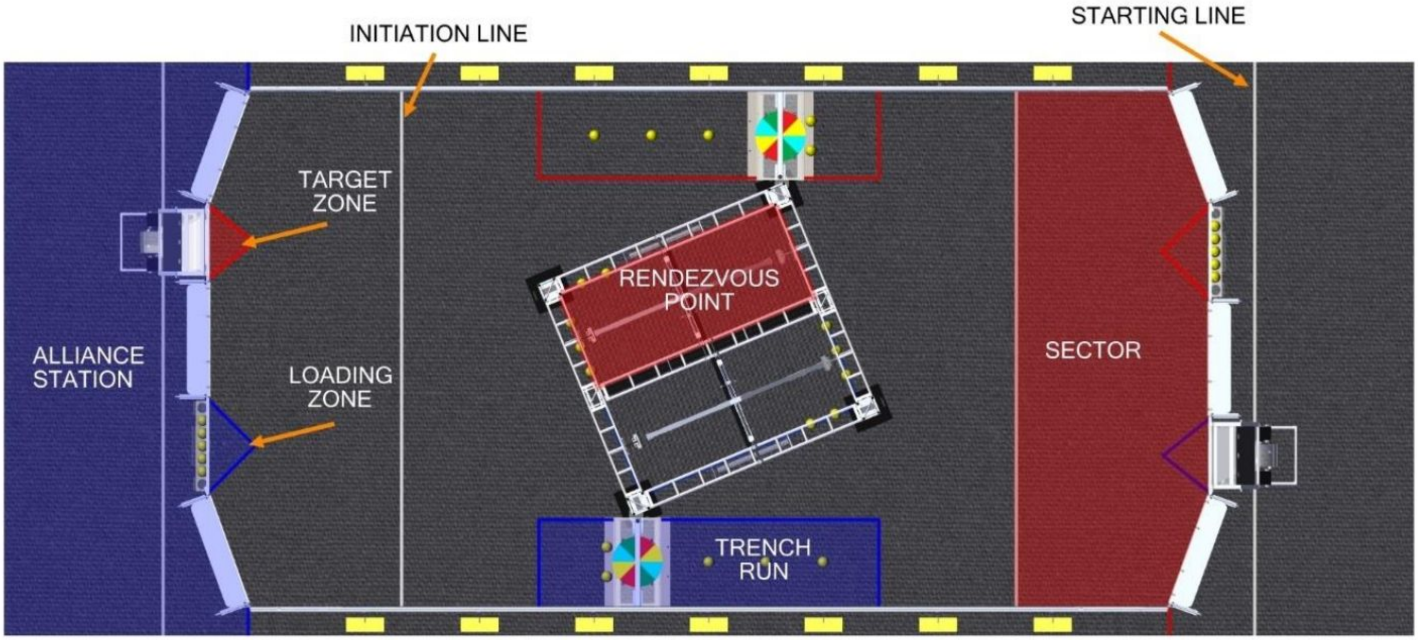
Strategic Design

Scoring and Complexities

After the rules were read during the team-wide kickoff, consensus had to be established across the team regarding outlining every possible way to score. For each possible scoring method, three things were noted to assist in weighing them against one another: the value it provides in match, the difficulty of building a mechanism to do this on the robot, and the time that said task would take out of the match. For each task, the score value was provided in the rules, the difficulty (on a scale from 1 to 10) was subjectively reasoned about by the team until a consensus was reached, and the time was estimated by the team. From these conclusions, each scoring method was prioritized on a need/want basis.

The following conclusions were agreed upon:

Task	Value	Difficulty	Time	Need/Want?
Initiation Line (Auto)	***5 pts***	***1***	***1-2 sec***	***Need***
Bottom Port (Auto)	2 pts	3-7	rest of auto	Want
Outer Port (Auto)	4 pts	6-7	rest of auto	Want
Inner Port (Auto)	6 pts	10	rest of auto	Want
Bottom Port (Teleop)	***1 pt***	***2***	***20-30 sec (5 PCs)***	***Need***
Outer Port (Teleop)	2 pts	6	25-35 sec (5 PCs)	Want
Inner Port (Teleop)	3 pts	8-9	25-35 sec (5 PCs)	Want
Rotation Control	10 pts	3	10 sec	Want
Position Control	20 pts	4	10 sec	Want
Parking	***5 pt***	***1***	***5 sec***	***Need***
Hanging	***25 pts***	***7-8***	***20 sec***	***Need***
Level	***15 pts (alliance)***	***9***	***40 sec***	***Need***
Operational	***1 RP***	***9-10***	***40 sec***	***Need***
Energized	1 RP	7-8	whole match	Want



Near vs. Far

The term *archetype* (with respect to FRC) is one coined by 1257 strategists when laying a framework for strategic design, picklisting, and match strategy and is defined as the ideal robot to accomplish a set of objectives. 1257's way of organizing strategy involves splitting the total set of objectives that can be accomplished in a match into ones that can be performed by different types of robots (for example, in the 2019 season, the most effective w9ay to split robots was into rocket bots, cargo ship bots, and defenders). Given the massive roles cycles play in match strategy, 1257 aimed to define archetypes based on what affected cycles the most (and therefore from where robots could score). For this reason, the team considered the question of whether to build a robot that scores Power Cells by lining up next to the Power Port or shooting from afar.

The benefits of scoring from nearby include the following:

- the Target Zone is protected by G10 and G11, making it easier to reliably score without being concerned about defense
- shooting from a distance involves having a reliable way to line up with the Power Port (nearby, a robot can just align itself with it)

The benefits of scoring from a distance include the following:

- by decreasing the distance from the Loading Bay to where a team would score Power Cells, a team can halve its cycle time
- there may be a lot of defense throughout the field that would need to be maneuvered to line up next to the Power Port, but shooting from a distance can avoid this problem

Eventually, the team decided to build a robot that could score from nearby, because:

- the insurance against defense that comes with scoring from nearby is worth the extra few seconds every cycle (5 Power Cells over a long time is better than 0 over a short time)
- it is a lot easier to line up with the Power Port when scoring from nearby

High vs. Low

Given that 1257 had the choice to place Power Cells in the Bottom Port or the Outer and Inner Ports from inside the Target Zone, the team weighed each one against the other.

The benefits of scoring inside the Bottom Port are that:

- it is easier to build a robot that can place Power Cells in a lower goal
- during the match, it is easier to place Power Cells in a lower goal than it is to score them into a higher goal

On the other hand, the benefits of scoring inside the Outer and Inner Ports are that:

- building a component to store and score Power Cells is the hardest part of the mechanism anyway (and will be necessary whether going for a lower or a higher goal)
- the scoring differential between a higher goal and a lower goal is massive (it is at least double, depending on how many Power Cells enter the Inner Port)

Given the above factors, the team chose to build a robot that could score into the Outer and Inner Ports because:

- the scoring differential justified the gap in difficulty between a lower and a higher goal
- the steps required to build a robot that could score in the higher goal are only marginally more than those required to build one that could score in the lower goal

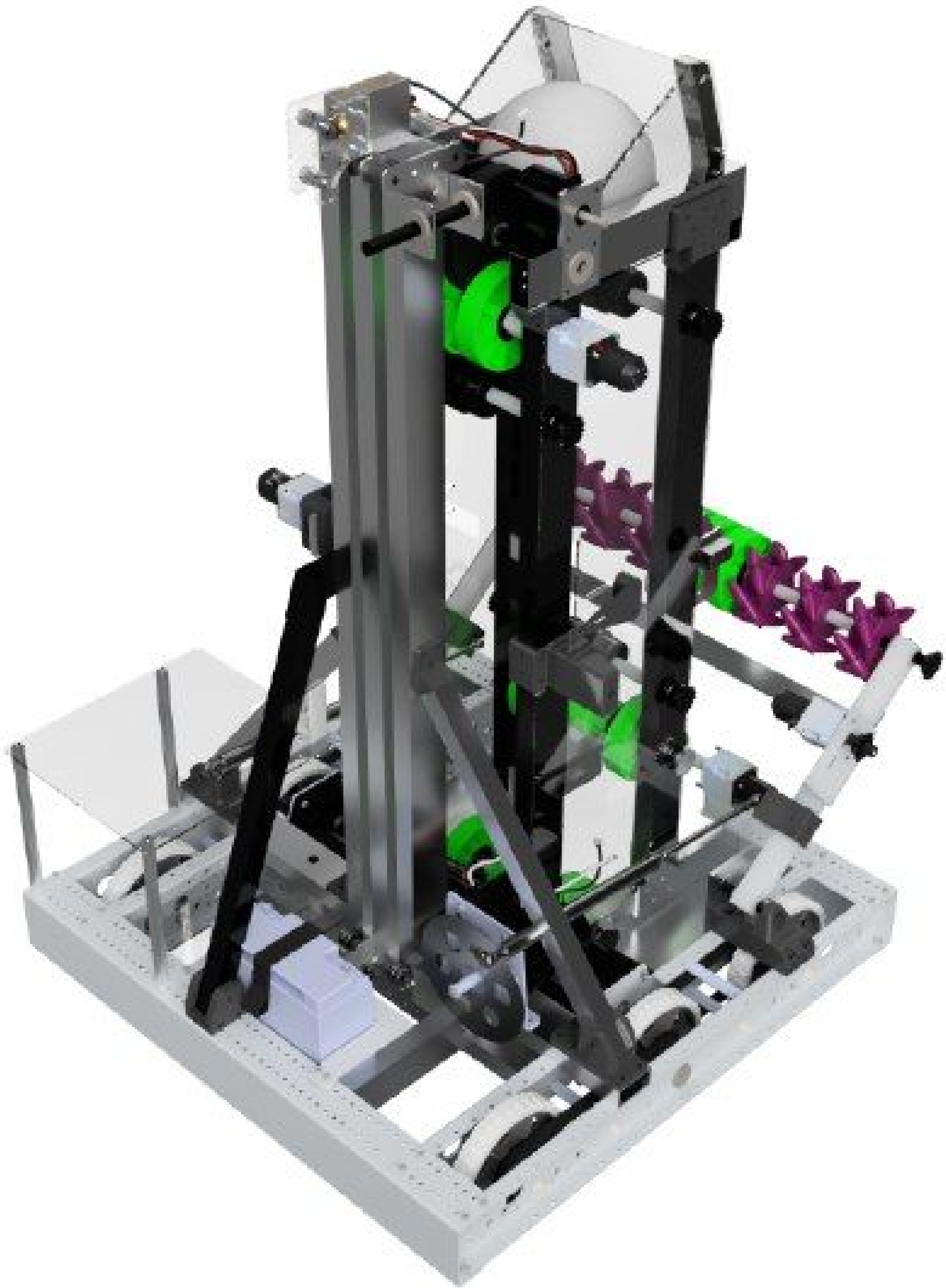
Final Decision

The team decided that if given time, a mechanism for the Control Panel (which was noted as a want on the table, as the scoring capacity for the Control Panel is rather limited if teams cannot activate Stage 2) could be created.

However, given the above decisions, the team’s final decision was to build a robot that

- could score Power Cells in the Outer and Inner Ports from the Target Zone
- could reliably climb in the Endgame and keep the Generator Switch level (these were also noted as a need in the table)

Robot Design



Drivetrain

The drivetrain allows our robot to maneuver around the field through teleoperated or autonomous control.

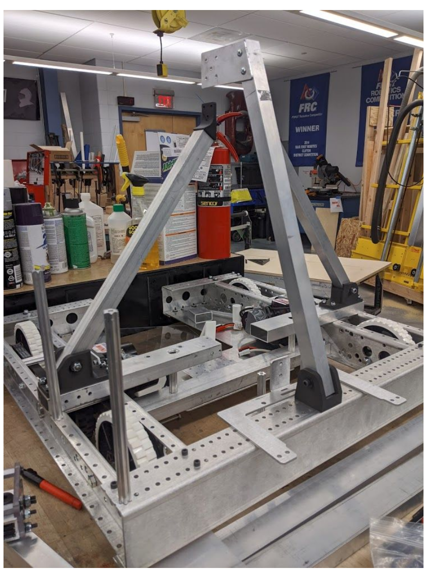
- Standard drop-center tank drive from the AndyMark kit chassis

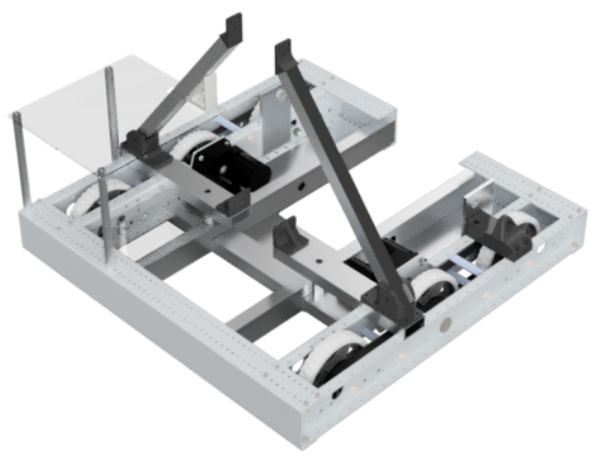
Features

- Controller using arcade drive with two sticks on an Xbox One Controller
- Closed loop velocity drive for more precise teleoperated control
- Autonomous Capability:
    - Two cascaded PID controllers to drive a set distance while maintaining a 0° heading
    - PID controller on the heading from the NavX board to turn to a specific angle
    - Path following with a Ramsete unicycle controller to follow WPILib generated trajectories.

Reasoning

- Simple, so we can rely on it and focus on other mechanisms
- Easy to control and provides good performance during defense





Intake

The intake is responsible for intaking retaining and transferring power cells to the indexer.

- Uses a SPARK MAX Motor Controller with a NEO 550 Brushless Motor connected to a roller with mecanum, omni, and compliant wheels to center balls/fuel

Features

- Uses a servo to keep the intake within the frame perimeter until the match begins, which is when the servo is disengaged

Reasoning

- To improve cycle times, mecanum and omni wheels were used to vector power cells into our indexer, decreasing the accuracy and alignment required to intake a power cell

Indexer

The indexer manages a maximum of 5 power cells in a linear vertical fashion within the robot. Once the robot is in position, it sends the power cells into the shooter for scoring.

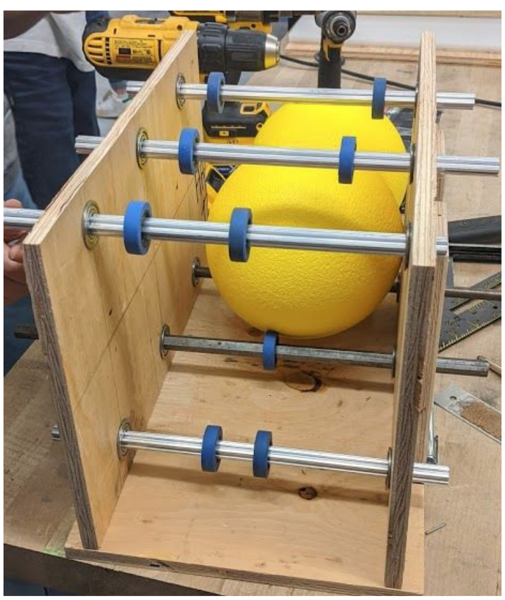
- Uses a SPARK MAX Motor Controller with a NEO 550 Brushless Motor to control the top stop wheels that regulate when the power cells enter the shooter’s control.
- Uses three SPARK MAX Motor Controllers with three NEO 550 Brushless Motors to control each conveyor belt on the front and back sides indexer
    - Design inspired by 341’s robot and 25’s robot in the 2012 season and First Capital Ri3D 2020

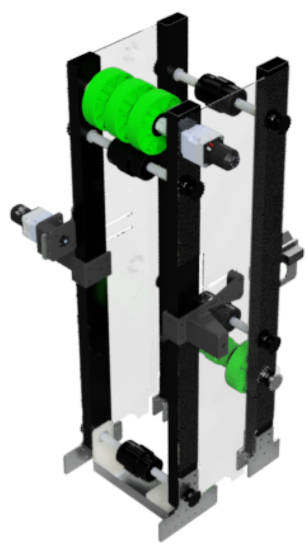
Features

- Uses two break beam sensors at the bottom of the indexer to detect approaching power cells and automatically run the indexer
- Uses the REV Color Sensor v3 at the top to detect the distance from the nearest object and ensure the stored power cells do not run through the stop wheels at the top

Reasoning

- From First Capital Ri3D 2020, we realised that helical hoppers and similar designs would not work or be difficult to implement due to the power cell’s high friction. We opted to use a dual sided conveyor like the above designs to mitigate this.





Shooter

The shooter enables our robot to score by launching power cells into the power port's high goal.

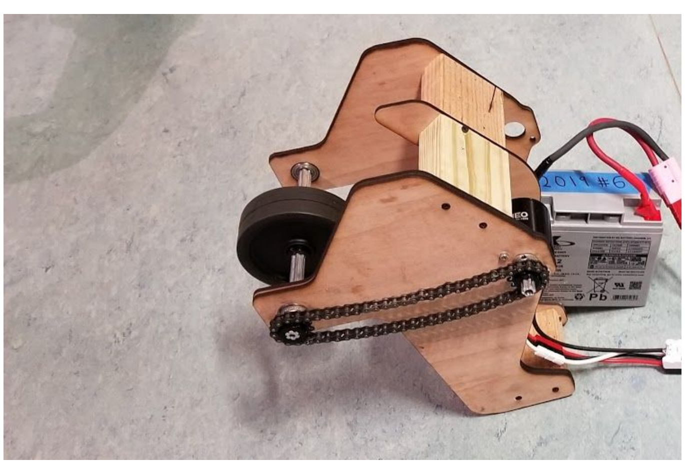
- Uses two SPARK MAX controllers with two NEO brushless motors for high power & space efficiency
- Shoots at a predetermined angle and speed that is calibrated so the robot is within the protected zone while scoring

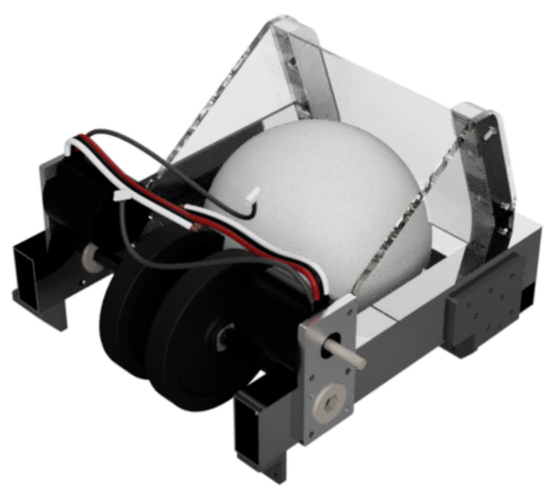
Features

- Closed-loop control velocity PID controller combined with feedforward velocity control to achieve an accurate and consistent shooting speed
- Checks if the shooter is within the desired range and disables sending balls to the shooter if the check fails

Reasoning

- This was our first year designing a shooter. We decided to because we reasoned that designing and making the shooter was not the hard part; rather, the indexer and intake would be
  - These would already be on a low bot without a shooter.
  - In addition, after the first few prototypes were made, we realised that the shooter needed some small adjustments from the first prototype
- Initially, we had only one NEO, but the flywheel was slowing down too much between shots, so we added another one.
- We decided to shoot from closer range due to it being a protected zone.
  - This also helps making control much less difficult and our robot more consistent, as we're at more or less the same place every cycle.





Climb

The climber is responsible for enabling the robot to hang on the generator switch by raising a hook up to the bar.

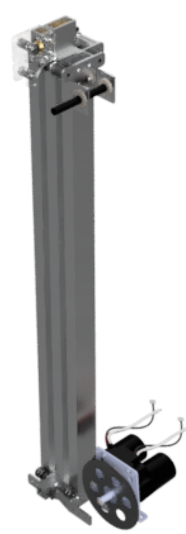
- Uses two SPARK MAX motor controllers with two NEO 550 Brushless motors to raise and lower a hook
- A servo connected to a disk brake locks the climb in place once it gets to the target height

Features

- Closed loop velocity control with the controller to achieve smoother and more precise operator control
- PID controller on encoder position to automatically move our climb mechanism to the top
- Minecraft golden sword duct-taped onto the hook (important)

Reasoning

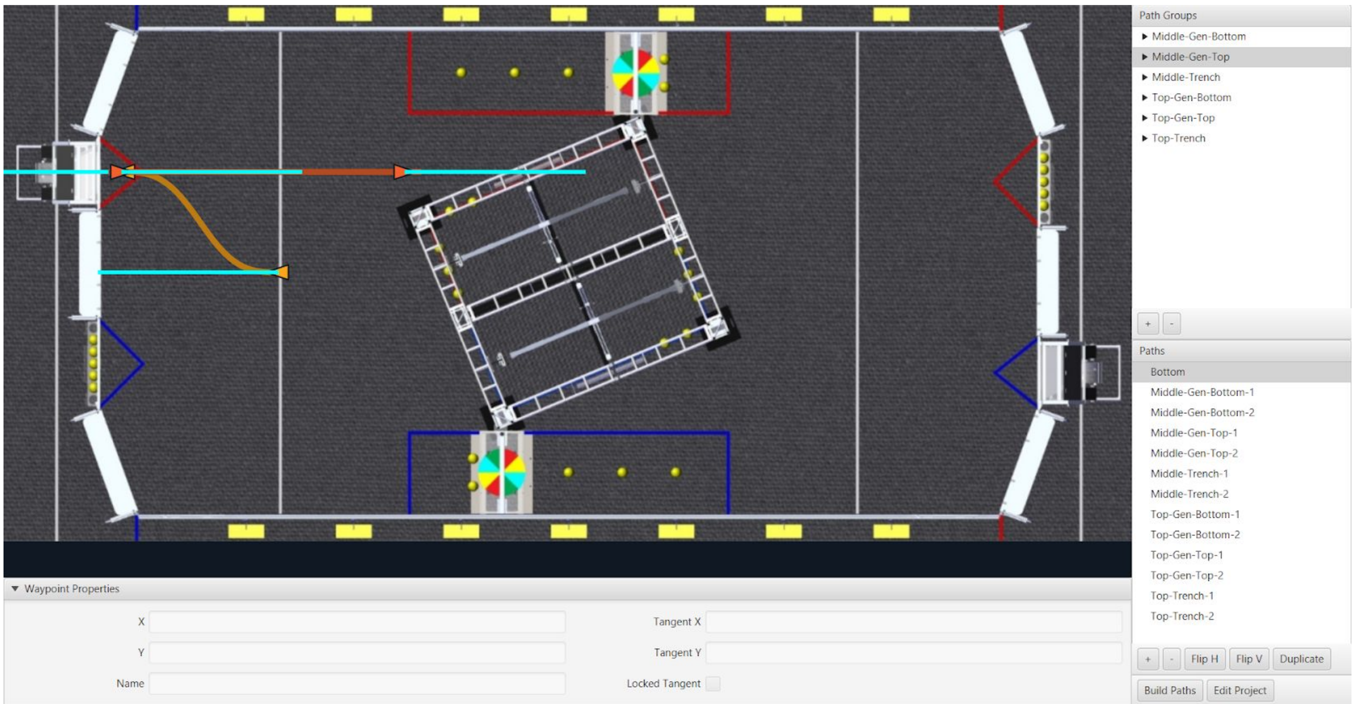
- Climb was determined to be extremely vital; this was confirmed in week 0/week 1
- Went with a simpler climb; we didn't worry about fitting under the trench run/control panel, so we could just have a one-stage elevator with a hook on it
- The sword has Sharpness V, as it helps cut through the air and improves aerodynamics
    - To counteract gold's fairly low durability, we also put Mending on it so we can repair it if it begins to break.
    - We decided not to go with diamond for cost reasons.



Autonomous

- Top priorities: maximizing score potential with preload and *setting up* for teleop intaking

- Included routines based off of WPILib trajectories
  - Separate paths for starting in the center, left, or right side of the field
  - Separate paths for different ending locations depending on our alliance strategy
  - Paths optimize scoring efficiency by running actions in parallel such as revving up the shooter while driving towards the goal
  - We visualized the generated trajectories and debugged them before sending them to the robot by creating a simple visualization app that draws the trajectories on top of the field
- Did *not* consider intaking extra power cells
  - Above our current level of control



WPILib Pathweaver program used to plan paths for autonomous

Motor Choices

- This year, our original design was to go full brushless with NEOs and NEO 550s*
  - The brushless motors offer a more compact design while obtaining more efficiency
    - Higher Power Delivery
    - Higher Speed
    - Much smaller than comparable brushed alternatives
  - *Due to troubleshooting issues, we temporarily had a BAG on the intake, but designed this as a fallback for if we would need to switch back.
    - In general, any NEO 550 can be swapped to a BAG by simply switching it from the gearbox and setting the Spark MAX to brushed mode with no encoder.

GitHub Links

Below we have links to all of our robot code for this year. Feel free to browse around and use this code as a reference!

2020 Robot Code - https://github.com/FRC1257/2020-Robot

Training: https://github.com/FRC1257/robotics-training

Vision Library - https://github.com/FRC1257/snail-vision