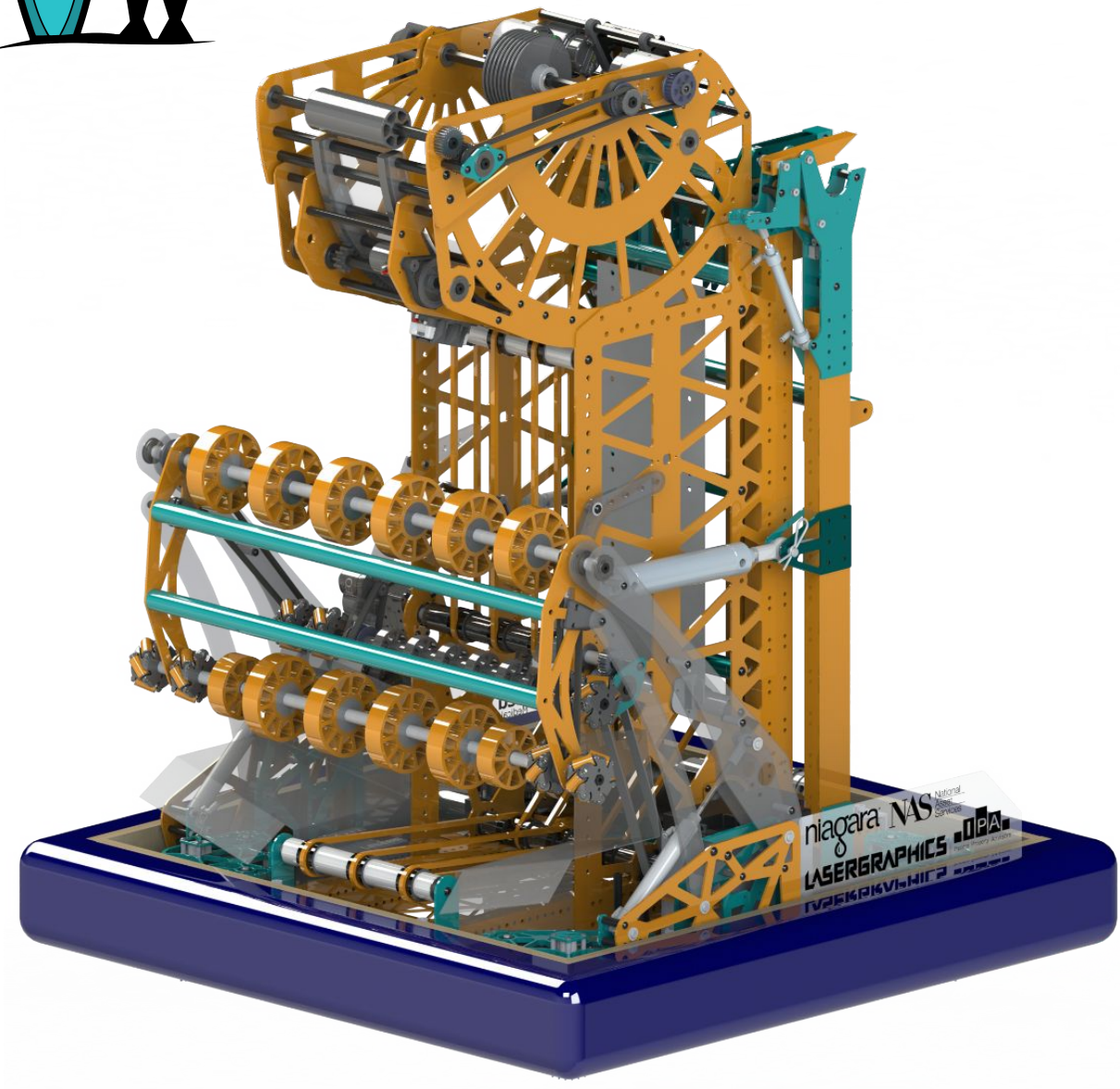
2022 Technical Documentation



ECLIPSE

“

I've been a mentor on the team since the 2014 off-season and I’m blown away by everything we have accomplished in the last 7 years. Our program is quite incredible–just keep spreading the message of FIRST in the community, especially in places and with populations where we don't have reach right now, and keep driving towards excellence.

- **Saikiran Ramanan**

”

Table of Contents

I. Design Process

II. Mechanical Design

III. Programming

I. Design Process

Game Strategy

In years prior, we've watched the kickoff livestream in-person as a team. This year was different. As it's hard to maintain social distancing when having a team-wide discussion, this season's strategy was planned out on a virtual meeting so we can have an open discussion without any risks.

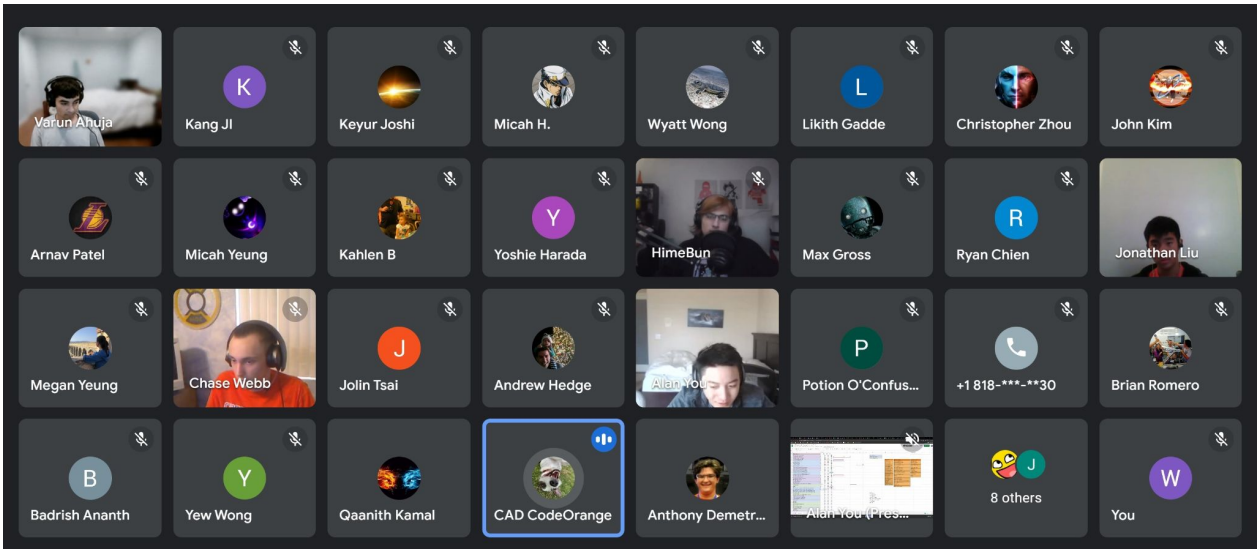
To plan out overall season strategy, we start by first reading through the entire game manual as a team so **everyone**–no matter if you're a new student or veteran–can **contribute** to the discussion. Then, we start creating a list of our goals for the season. Our goals will determine what strategy we want to employ on the field.

Goals:

1. Win Champs
2. Win subdivision
3. Win Chairman's
4. Win regional
5. Seed high
6. Win matches

Strategies:

1. Win RP
2. Score points
3. Fast cycles
4. Drive under defense
5. Climb high



Ranking Specifications

From out list of goals, we then ask **what** exactly our robot needs to do to achieve these goals. These **what**'s are our robot specifications. It's important in this step to only ask **what** and disregard any **how**'s. Implementation of our specifications is a design problem which we will figure out when we start prototyping and CADding.

After a long list of specifications are compiled from open discussion, we move through each specification individually and sort each as a **Wish**, **Prefer**, or **Demand**

Wish	Prefer	Demand
Score in lower hub in teleop	Climb to high rung	Score in upper hub in teleop
Score in lower hub in auto	Be able to move from mid to high rung	Score in upper hub in auto
Climb to low rung	Be able to move from high to traversal rung	Climb to traversal rung
Be able to move from low to mid rung	Score from far launch pad	Climb to mid rung
Score from far zone	Score from anywhere inside tarmac	Climb directly to mid rung
Intake from terminal during teleop	Score from center zone	Intake cargo from ground during teleop
No swinging	Aim/Shoot from everywhere where we can shoot	Intake cargo from ground during auto
Score 2 cargo at once	Strafe	Be able to move from mid to traversal rung
Push through defense	Don't carry more than 2 cargo	Score from close launch pad
	Score quintet	Score from fender
	Clear hopper	Driver camera
	Score above defense	Allow an alliance partner to climb with us on same rung
Key	Upside-down electronics	Allow an alliance partner to climb with us on different rur
Field Actions	Score while moving	Allow 2 alliance partners to climb with us on different rur
Game Piece Actions	Use easily-replaceable COTS parts	Easily repairable
Construction Actions	Modular	Protection against controlling cargo by accident
	Intake bouncing ball	Drive fast
	Fast climb	Auto aiming
	Reversible climb	Easily accessible electronics
	Shoot at a different angle from drivebase	Easily accessed hopper
		Untippable
		Carry 2 cargo
		Only score alliance colored cargo into hub
		Taxi from tarmac during auto
		Eject opponent colored cargo
		Human playing scores into upper hub during auto
		Intake multiple balls without jamming
		Hold 1 cargo
		Score quickly
		No jamming
		No deadzones
		Drive over cable protector
		Robust
		Shoot accurately
		Drive around defense
		Secure battery
		Easily-accessible battery (no removing bumpers)
		Testable
		Easily-removable bumpers
		Robust bumpers
		Subsystems aren't dependent quality of game pieces/fie
		Retain climb at T=-5
		Self-ratcheting climber (no setup pre-match)

Wish: while beneficial, not worth time and effort

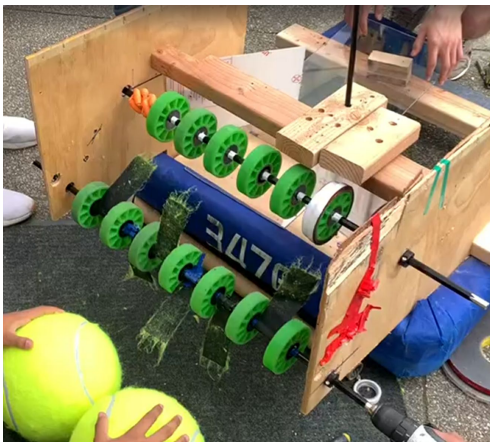
Prefer: worth looking into if we have extra time and resources

Demand: absolutely must be on the robot

Prototyping

Next, our team splits into smaller subsystem groups to prototype ideas following our wish, prefer, demand list. This year, prototyping groups also acted as cohorts to contain possible exposure to the coronavirus.

Subsystem groups brainstorm ideas and build prototypes with wood and scrap material. It is important to collect **data** and document **functionality** through video in this step so designers can look back to determine subsystem specifications



Intake prototype

Hopper prototype

Shooter prototype

CAD Design

Every season, our goal is to finish the detailed modeling of each subsystem by week 3. This year, we finally kept close to this deadline!

The CAD Design phase is a constant back-and-forth between the CAD subteam, prototyping group, and mentors. Based off of testing, the prototype group relays information to the CADder assigned to the group. Once an initial design have been modeled, the CADder presents the design to mentors and teammates in an **open discussion**. Any feedback is quickly implemented and another round of review begins.

Once a design has been approved, drawings are created to turn each 3D part into a 2D diagram to be manufactured by the manufacturing subteam.

Even while their designs are being manufactured, CADders often start designing improvements based on information released by other teams or new information from continued prototyping. **Iteration** and **improvement** is extremely important to our team.

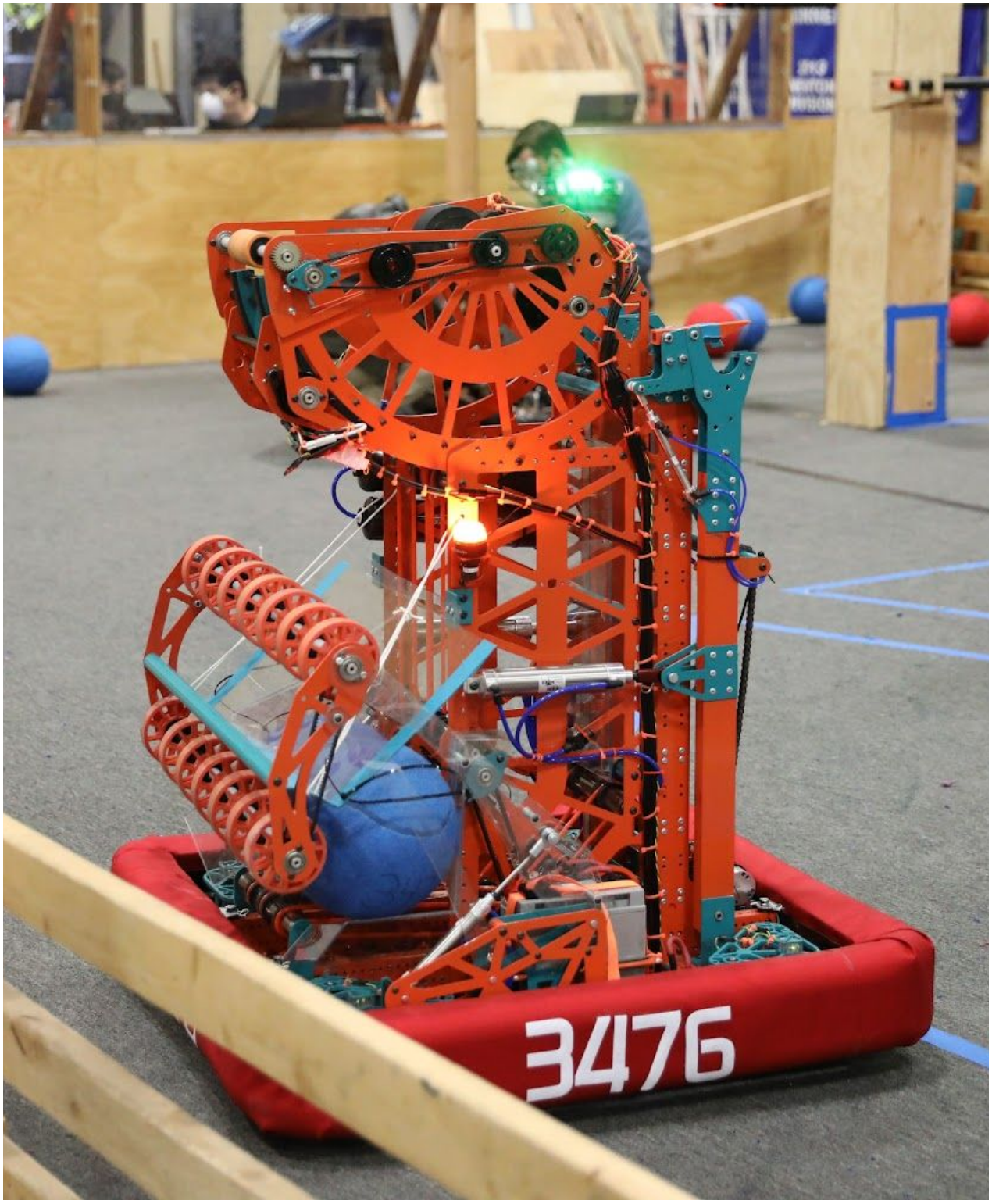
Fabrication

The manufacturing subteam is the largest subteam on our team and works to bring the robot model on the computer into reality.

With hundreds of parts to be manufactured, **organization** and **quality** is key. The manufacturing lead ensures each part is made to the tolerance specified on the drawing and completed parts are properly labeled for powder coating and assembly.

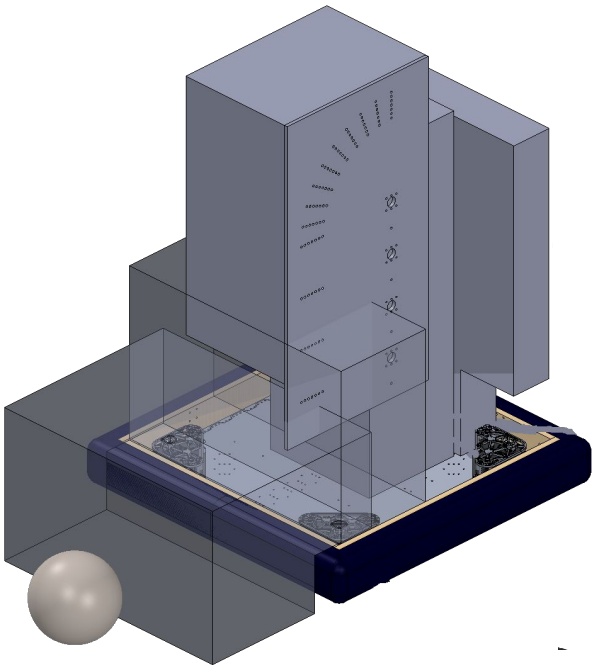
Our team uses **basic machinery** like bandsaw, chop saw, bench grinder, and belt sander as well as **advanced machinery** like mill, lathe, laser cutter, CNC mill, and CNC router to manufacture our robot. One of our sponsors, Applied Medical, also allows us to use their Water Jet to cut large sheet metal plates.

Unique parts like camera mounts or custom pulleys can be created using our Prusa, Ender, and Markforged 3D printers. The Markforged printer allows us to print with a carbon-fiber/nylon filament for load-bearing parts.

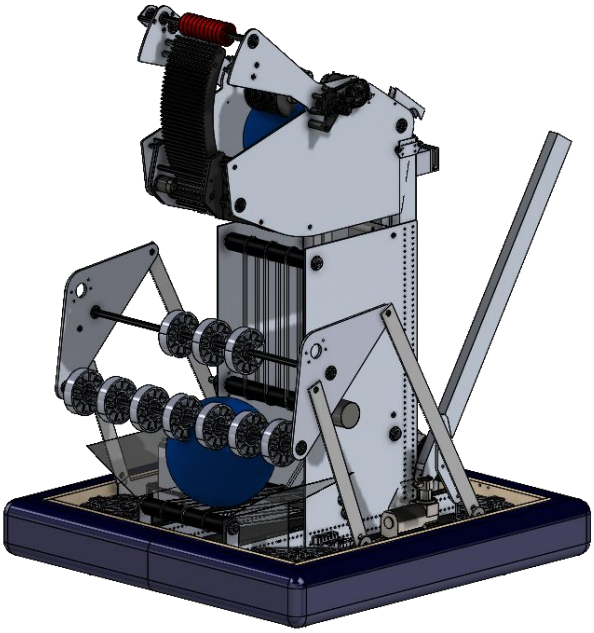


II. Mechanical Design

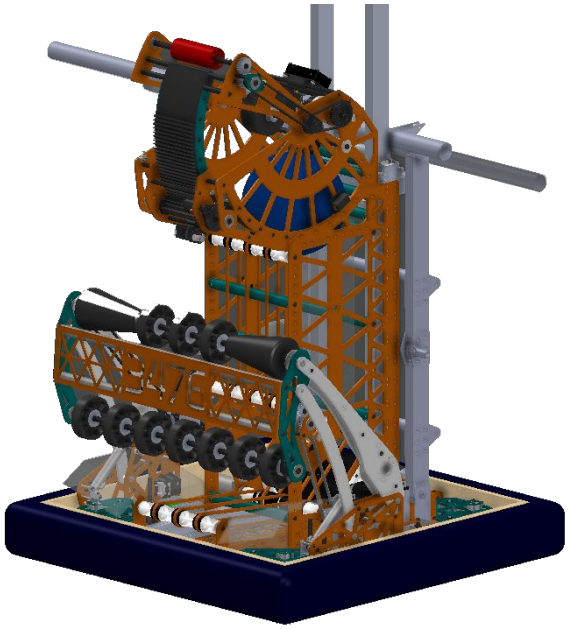
Day 7 We start with block CAD to get a basic idea of subsystem size



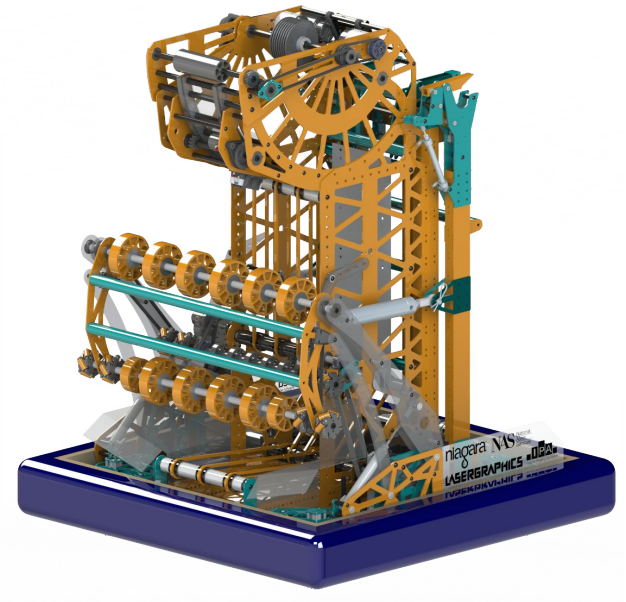
Day 14 Early subsystem designs based on prototyping



Day 19 Nearly done! Climber was the hardest subsystem this year

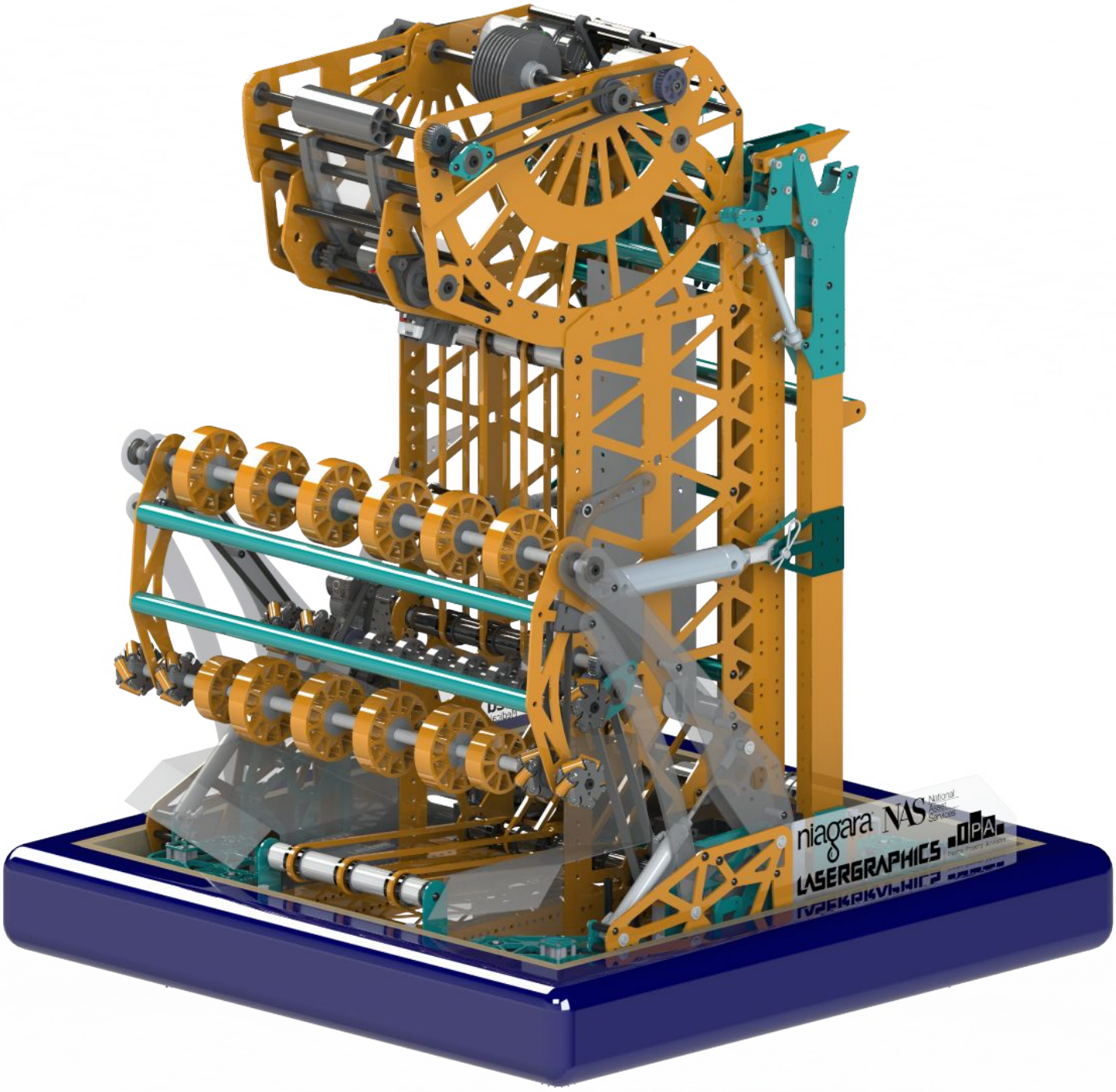


Day 100 Refinement and iteration is the name of the game. Our CAD is never finished.

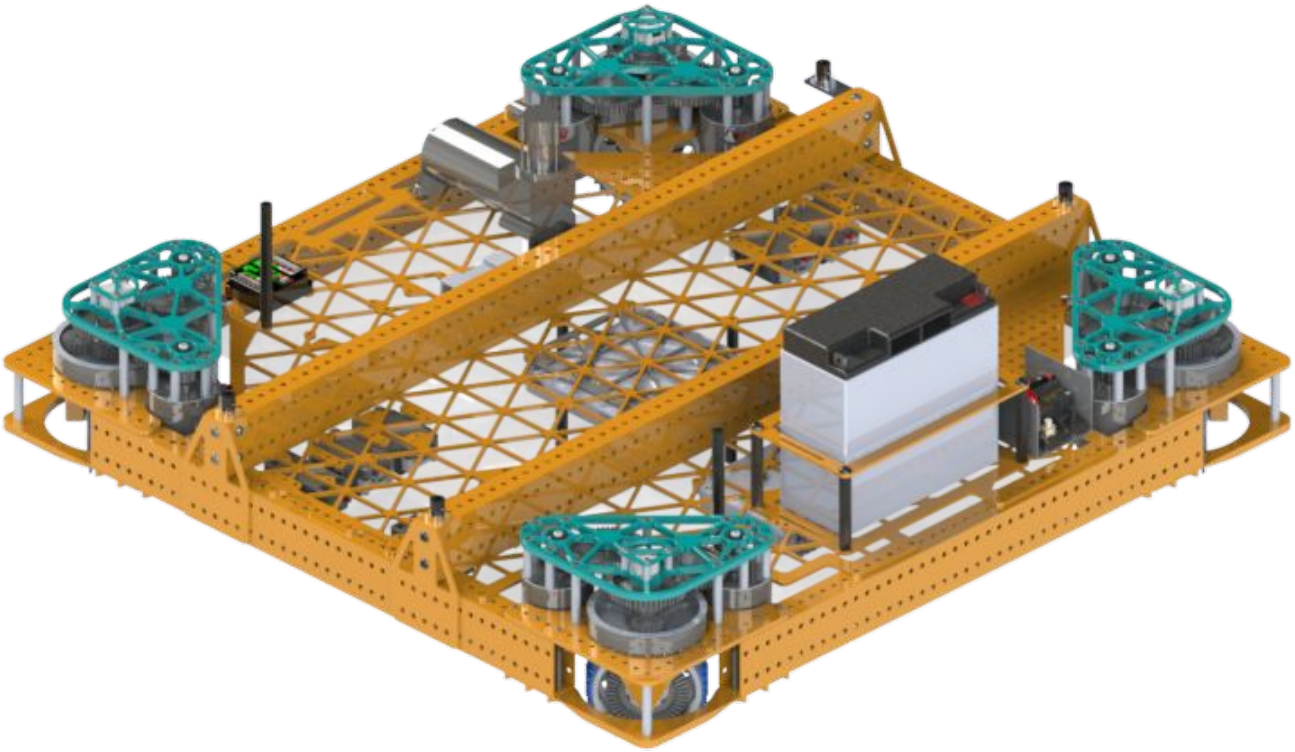


Final Design

Dimensions: 29.5 X 29.5 X 45.9in, 125lb



Drivebase



Frame

- 29.5in X 29.5in
- 1/16in box tubing drive frame
- 1/8in box tubing cross rails for hopper and climber mounting

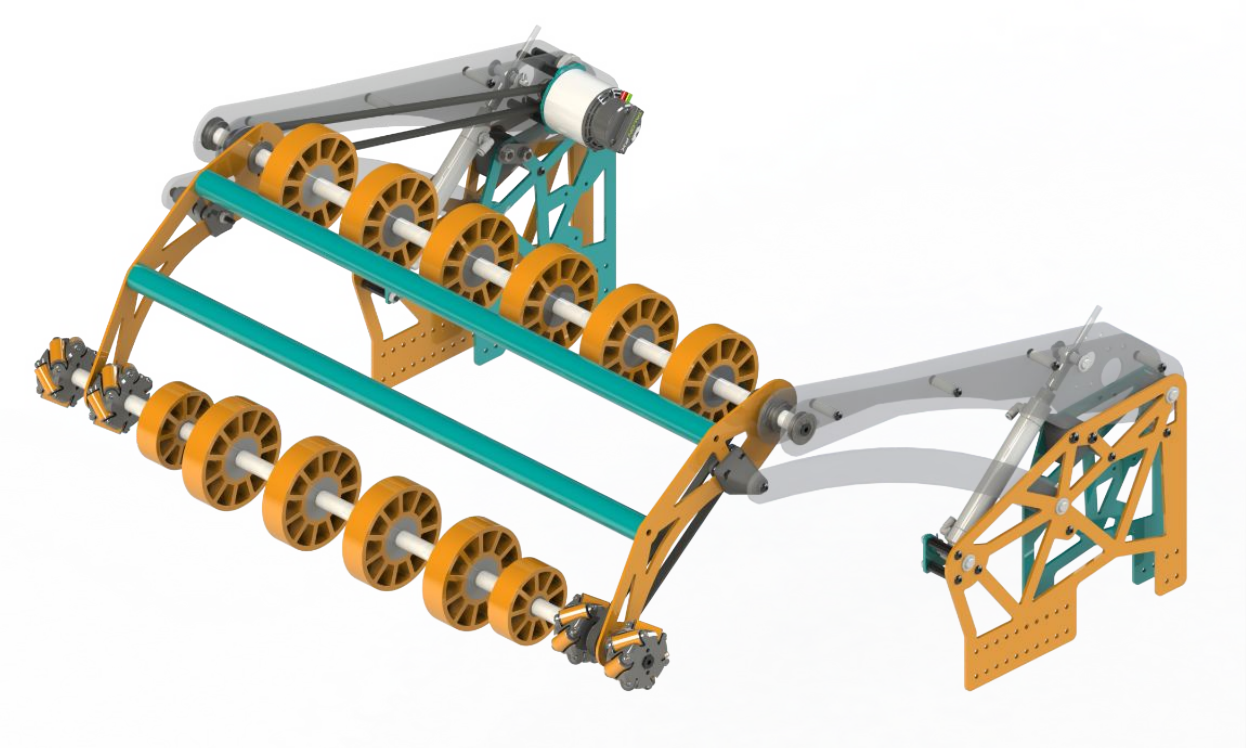
Swerve Drive

- Four SDS MK4i swerve modules driven by 8 Falcon 500s
- 3.5in traction wheels

Hybrid Electronics Mounting

- RoboRIO, motor controllers, and PDH are mounted in upside-down orientation on the bottom of the belly pan for protection against bouncing cargo
- PCH, VRM, compressor, pneumatic controls, and ethernet system are mounted on topside of belly pan for easy access

Intake



Four Bar Linkage

- Four bar design for compact and vertical stow position to save more horizontal space for other subsystems
- Mounted with cotter pins for quick replacement in case plates become deformed

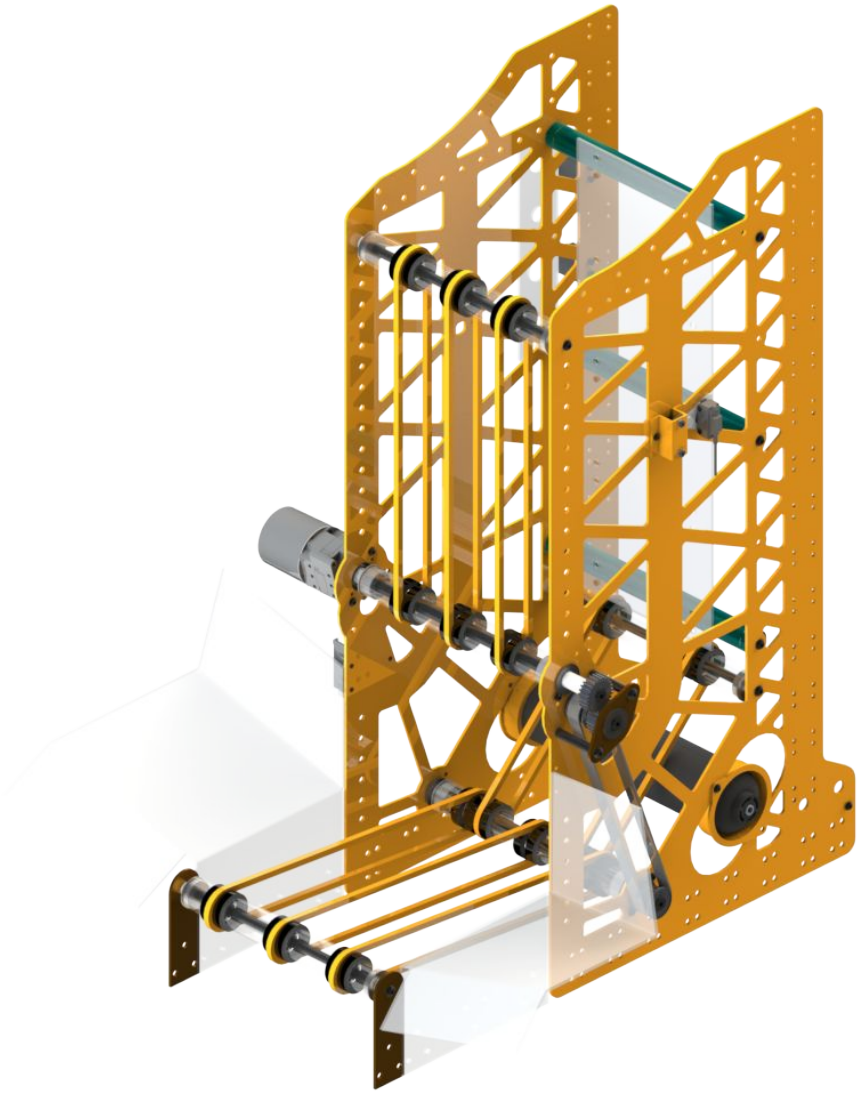
Shock Absorbing Design

- Shock loads from intake colliding with robots and field elements mitigated with flexible polycarbonate four bar linkages
- Pneumatic acts as a gas spring in “out” position to comply to shape of cargo

Rollers

- Two rollers made from spaced 4in compliant wheels with mecanum wheels on sides to center cargo
- Driven by Falcon 500 geared 1:2

Hopper



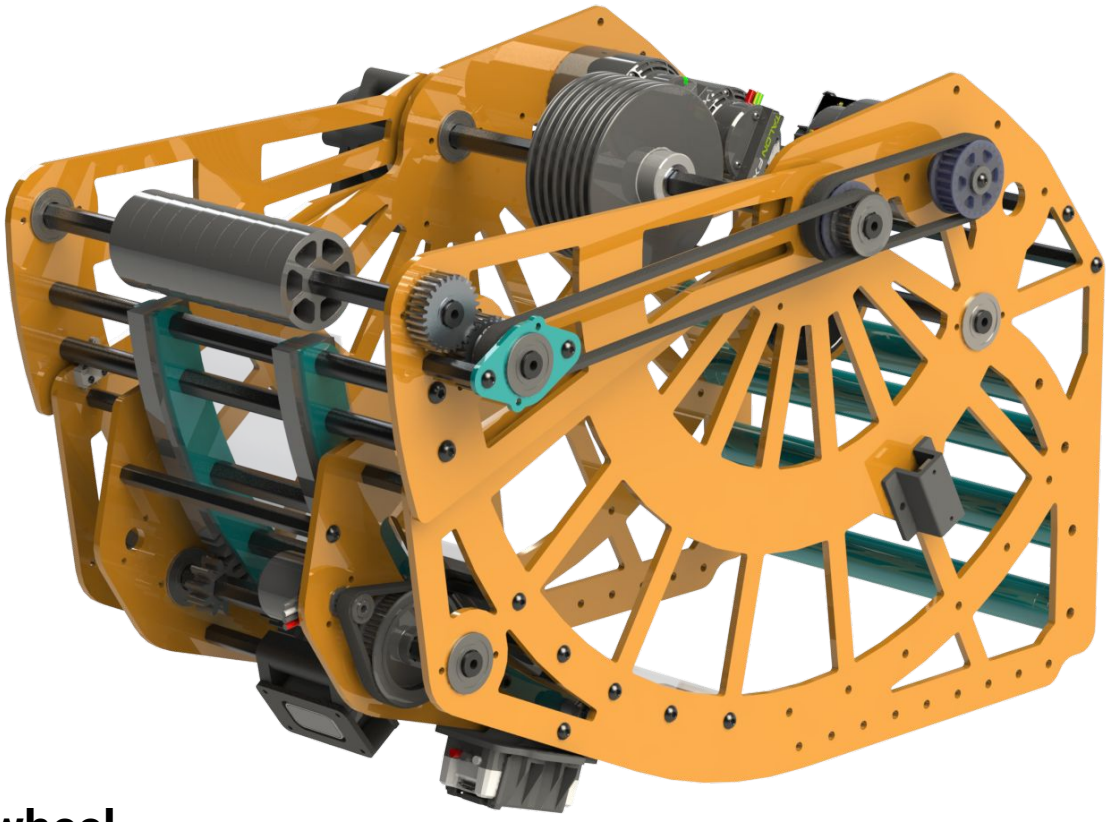
Polycord Rollers

- Polycord grips and compresses cargo from three contact points to keep the cargo controlled and centered
- Roller assembly powered by NEO geared 1:3

Sensors

- Limelight ensures correct cargo color
- Breakbeam sensor ensures robot never controls more than two cargo

Shooter



Flywheel

- 4in “Self Cleaning” wheel is semi-compliant to grip cargo
- Driven by two Falcon 500s geared 1:1
- 1.25in cargo compression

Hood

- 40 degree hood range from 90 degrees to 50 degrees to allow us to shoot from fender to alliance wall
- Custom-printed 10DP rack and pinion for less wear
- Driven by NEO 550 geared 1:5
- 2in top roller to minimize backspin geared 1:1 off flywheel shaft

Feederwheel

- 4in Andymark Stealth Wheel to feed cargo only when flywheel is up to speed to ensure consistent shooting
- Driven by Falcon 500 geared 1:2

Climber

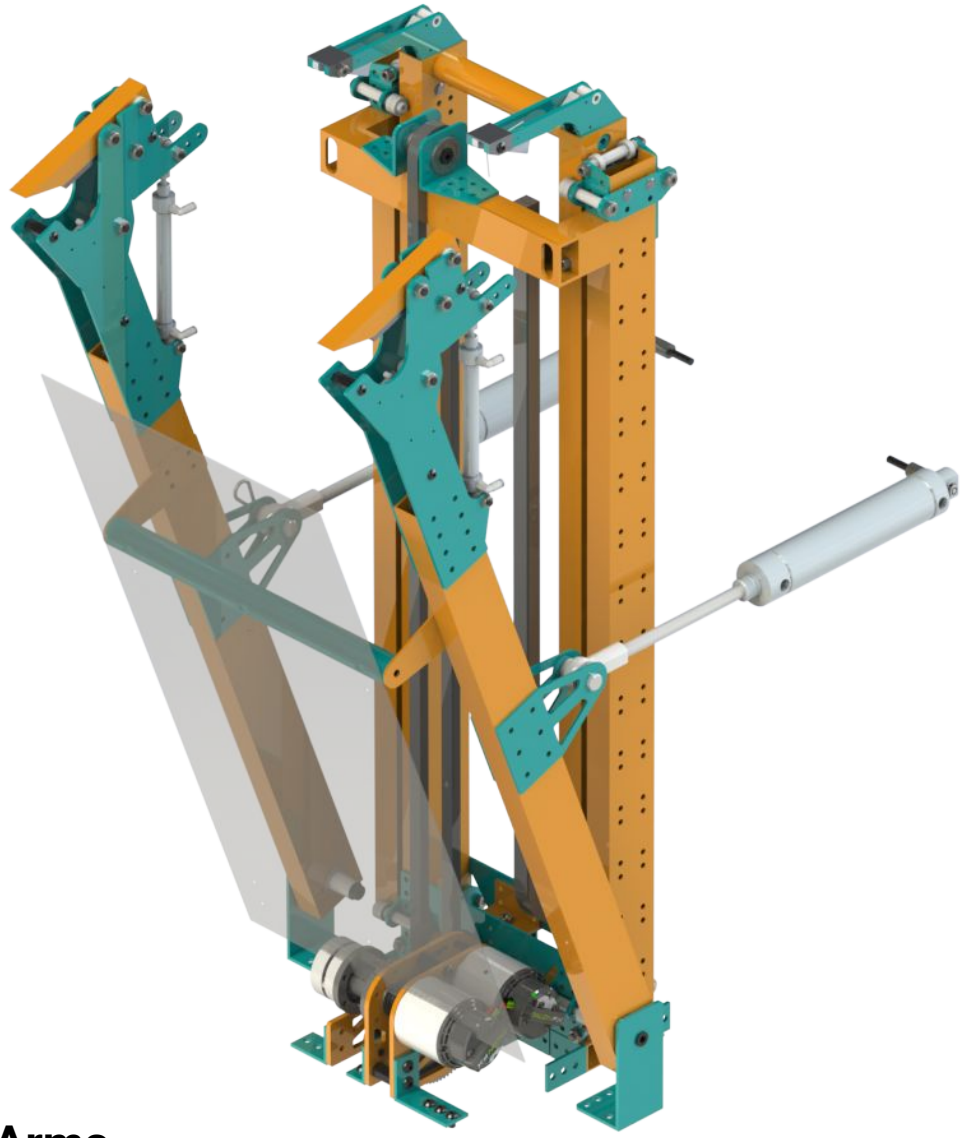
Elevator

- One stage elevator that extends 27in in 0.95 sec
- #35 chain drive
- Spring-loaded top latches to grasp bar even when swinging

Gearbox

- Robust and compact two stage gearbox powering #35 chain to drive elevator
- Driven by two Falcon 500s geared 1:10.1
- Pneumatic friction brake to lock climber elevator in position between rungs

Climber (cont.)



Pivot Arms

- Large, 1.5in bore, 5in stroke air cylinders mounted on both sides to tilt robot downwards to reach next rung
- Easily removable with 0.5 round shaft and cotter pin attachment points due to high shock loads

Claws

- Over the center linkage that does not apply load on driving pneumatics; claws can open and close under high load
- Each claw easily holds the weight of a swinging robot
- Contact switches to determine open/close states

Outtake

Roller

- 2in Thriftybot compliant wheels
- 2.5in custom 3D printed mecanum wheels to center cargo towards hopper
- Driven by NEO 550 geared 1:3

Wrong Color Ejection

- Limelight detects cargo color as it enters intake
- If correct color, outtake spins in direction of the hopper
- If wrong color, outtake spins in opposite direction to spit balls out of collection area

T=0 Climber

When the round ends, **pneumatics reset back to their default state**. Using this principle, our post-round climb is driven entirely by pneumatics. Thanks to team 6328 for giving us this idea!

At the end of the round, a grapple claw attached to a rope is fired by tensioned surgical tubing that was constrained during the match with a pneumatic pancake cylinder. After the grapple grabs onto the traversal bar, a much larger pneumatic cylinder pulls on the rope to lift our robot off the ground.

This climber would have been extremely beneficial to our match strategy because it's entire climb time occurs **after the match ends**, giving our drivers extra time to get in a few more cycles. The climber is also very low profile as it only takes up a tiny, 1in-wide area on the traversal bar. This allows for an extremely quick and simple **triple** traversal, which would be extremely valuable at competition. We finally designed, built, and tested this climb in the week before World Championships, but it was **made illegal** literally a day after we finally got it working.

So while we're not using it on our robot, we felt like it was a cool concept so it's here in our technical documentation!

T=0 Climber (cont.)

Eligibility

- QA 134: over extension after match doesn't draw penalties
- QA 154: head ref has discretion over legality of this climb
- QA 166: solidifies QA 134 ruling by making it globally applicable and specifically affirms that over-extension after the match does not affect eligibility for hangar points
- Precedence set by 2412 at PNW District Championships
    - Cleared by inspection for match play
    - Not penalized for overextending after match
- **Made illegal by Team Update 21**

Grapple

- 6in wide grapple hook
- Spring latches
- Fiberglass rods act as guides as grapple travels to bar to ensure accuracy

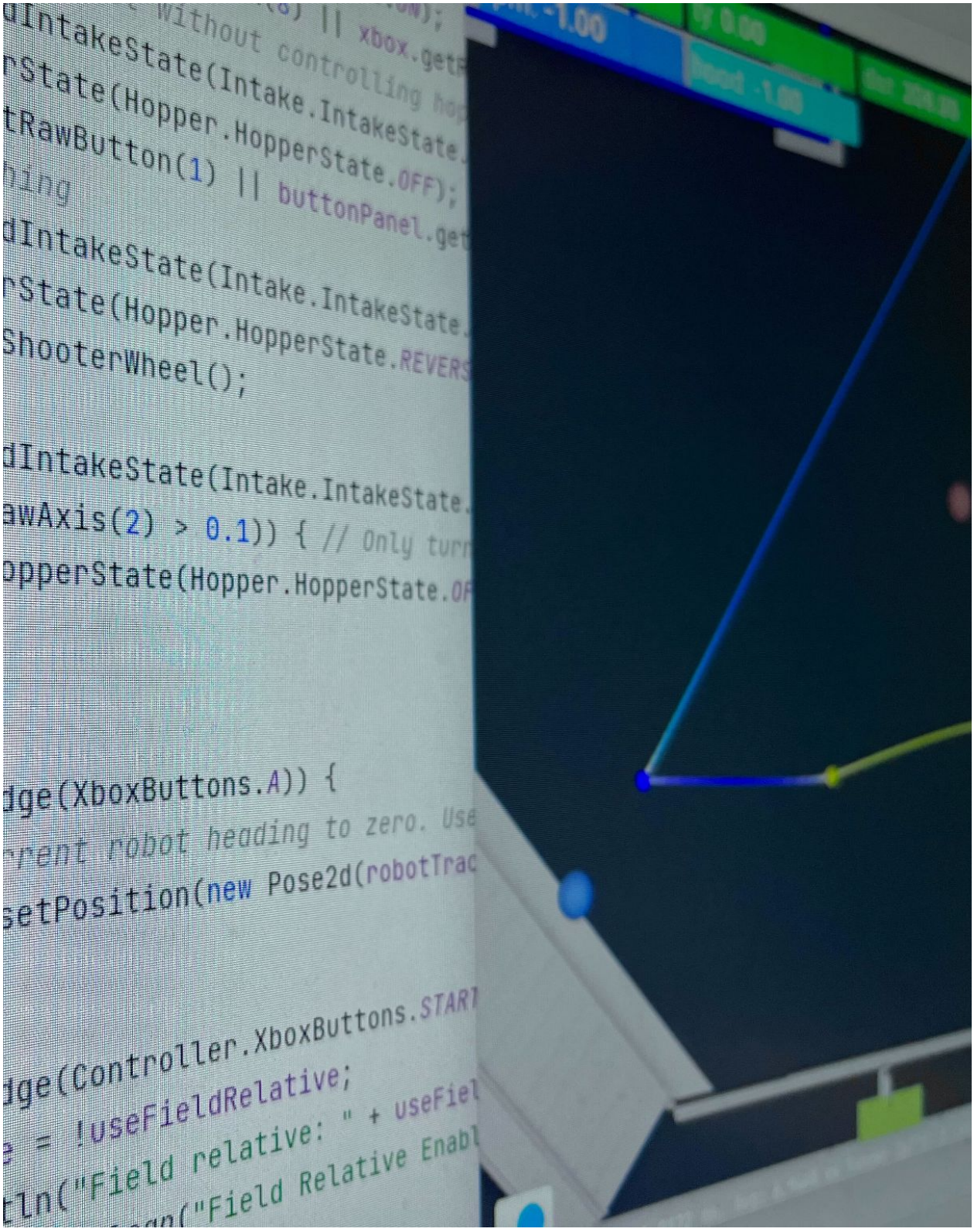
Launcher

- Grapple fired to traversal bar with tensioned surgical tubing
- Pneumatic pancake cylinder engages during match to constrain grapple hook from firing
- When robot is disabled, pancake cylinder disengages and grapple hook is fired

Lift

- Pneumatic cylinder with 2in Bore, 12in Stroke pulls on rope attached to grapple hook to pull robot 5 in off ground
- When robot is disabled, cylinder pulls up slowly with help from flow control valve

III. Programming



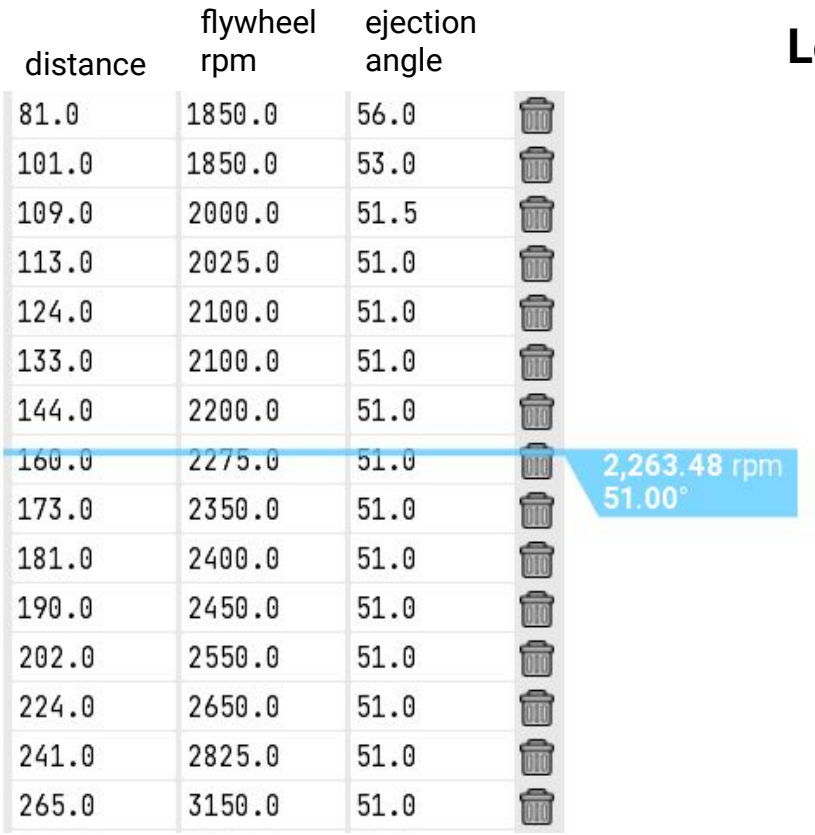
General Shooting

Aiming powered by Vision + Odometry

- Aim times of ≤1s
- Able to solve for the full pose of the robot using vision data & the gyro angle
- Sensor fusion with wheel odometry filters out noise from vision
- Latency Compensation allows for higher resolution tracking allowing us get useful vision data from anywhere on the field (even from alliance station wall)
- Cleanly falls back to robot odometry when vision is not visible/obstructed

Stop and Shoot

- Iteratively predicts stopping position using current robot velocity and acceleration, allowing the robot to aim towards correct goal position even before it gets there.
- Optimized to be simple and fast for drivers by taking control of robot movement and aiming in one press of a button.



Lookup table

- Used to determine flywheel speed & hood angle
- Utilizes the distance approximation from the limelight & robot tracker with error of < 5in at 200in
- 19 different shooter setpoints
  - Linear interpolation is used to determine the shooter configurations between multiple setpoints.
- Updatable in real-time with live feedback on the current configuration
  - Config is uploaded when typing stops and takes effect immediately–even while the robot is enabled and driving

Shooting and Moving

Shooting while moving

- Will automatically calculates a “fake” goal position to shoot at in order to make it into the real goal.
- Achieves this by modeling the ball flight based on robot velocity, and position.
- During our testing we came up with the following system of equations to figure out where to aim based on the robot position and velocity

$$\vec{F} = \vec{P} + t \cdot \vec{V} \qquad\qquad t = a \cdot norm(\vec{F})$$

Where:
F = Position of the Fake Goal
P = Position of the robot (relative to the goal)
t = Time of Flight of the ball
a = a scalar value for our time of flight (technically our time of flight is a piecewise function with a linear bit at the beginning, but any continuous function works for the way we solve it)

We then use successive approximation to solve this system of equations to obtain a “fake” goal position. We then pass this fake goal position into our static shot pipeline to have the robot aim and shoot at that fake goal position.

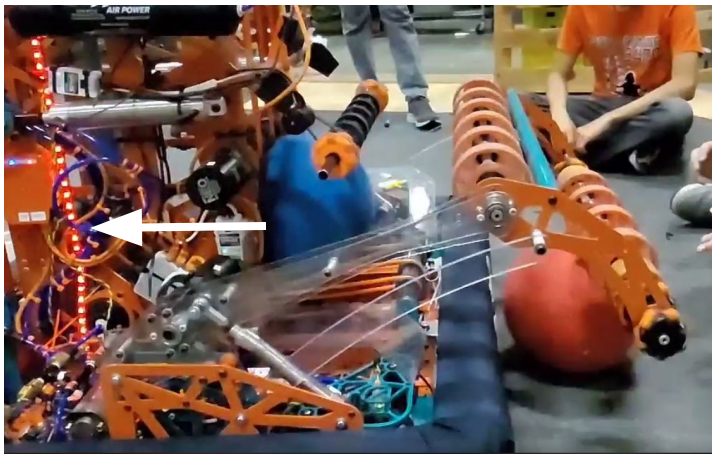
- All of this happens with the click of one button.

In this image, the robot aims slightly right of the goal in order to compensate for the leftward motion.

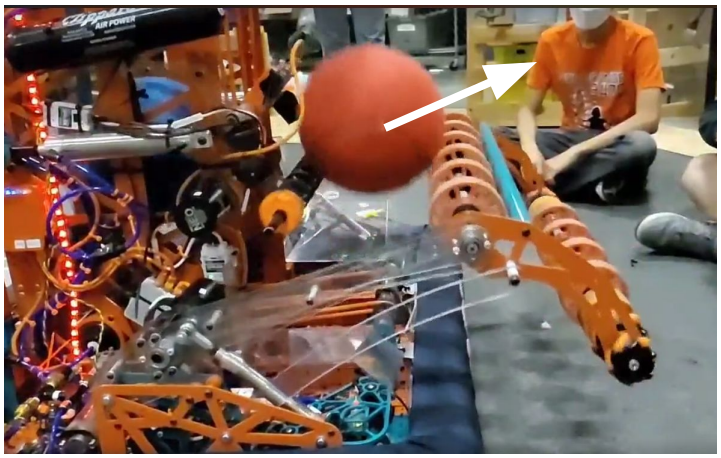
Automatic Ball Ejection

Ejecting wrong colored balls

- Uses intake mounted limelight in order to detect balls of opposite color.
- In order to ensure that color detection works in all lighting conditions, a white LED strip illuminates the balls as they enter the robot.
- Uses outtake mechanism to quickly eject balls, allowing for groups of balls to be easily sorted through.



The blue ball is accepted into the robot



The red ball is detected as a ball from the opposing alliance and is ejected from the robot automatically

3 Ball Penalty Protection

- A hopper mounted break beam sensor allows for the detection of excess balls in the hopper.
- When 3 balls are detected in the robot, it automatically ejects a ball through its hood without driver intervention.
- This ensures that we do not obtain a G403 violation as a third ball would be ejected from the robot quickly enough to be considered only momentary control.

One Click Climb

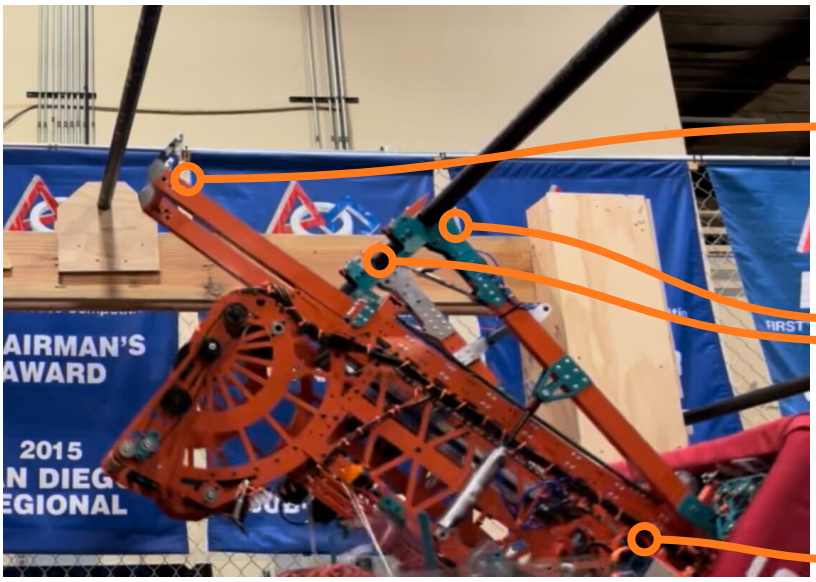
Removes human error as the roborio is in control of climb through its duration.

Fully automatic, sensor-based climb

- Two contact switches on each pivot arm ensure that we're locked onto the bar before continuing onto the next step.
- Encoders are used to make sure that the arms are extended to the correct lengths, at the correct time.
- The NavX2 gyro is used to ensure we're at the correct part of our swing before continuing to the next step.

Driver Activation

- Drivers hold down the climb button while climbing.
- If the drivers release the button the climb is paused. It is resumed once the drivers rehold the button.



Encoders ensure that the elevator arm is at the correct height

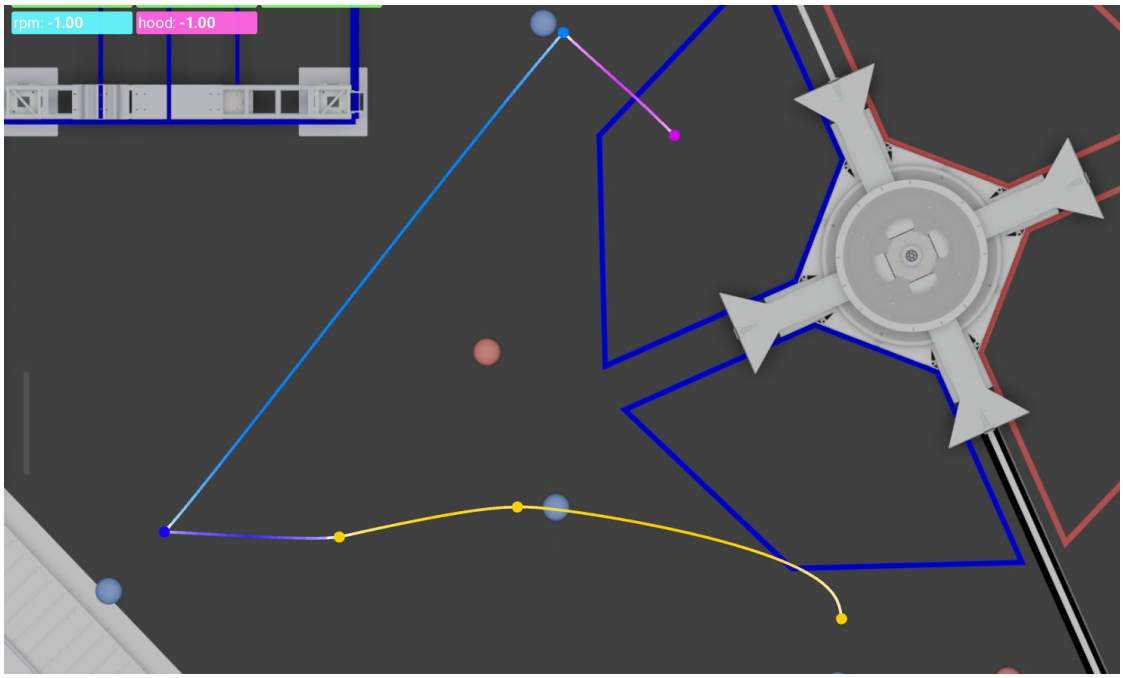
Switches ensure that we're latched onto the bar

The NavX2 gyro that we're in the correct part of our swing to continue

Autonomous

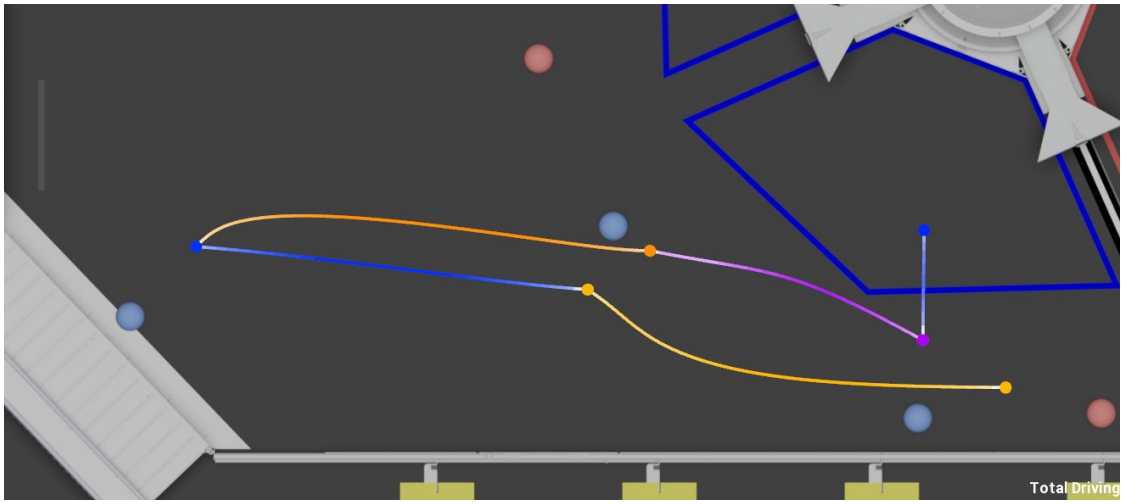
6 ball auto

Our highest scoring auto. Only used when our alliance partner's autos won't intersect the path.



5 ball auto

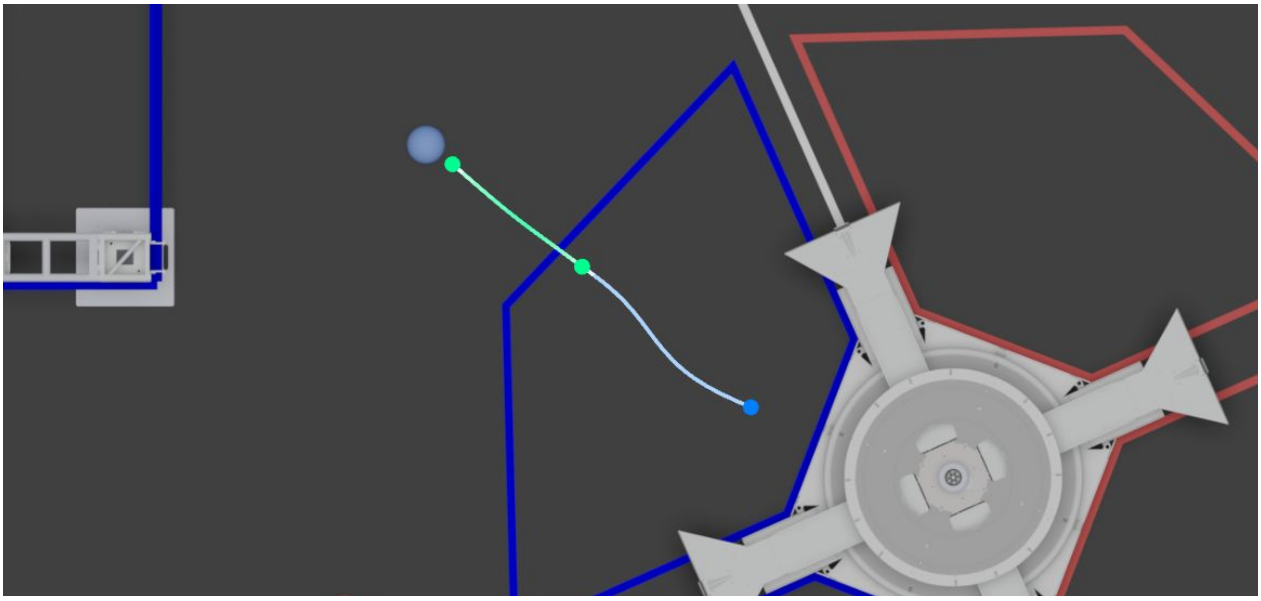
Our main auto. Designed to be extremely reliable and will be our goto for most matches. At the end it positions ourselves in the ideal spot for the start of teleop



Autonomous (cont.)

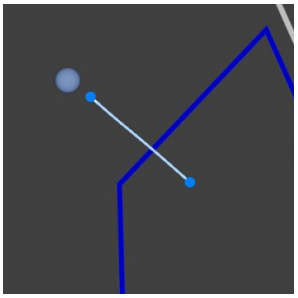
Buddy Auto (3 ball)

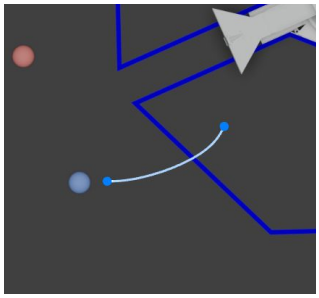
This Auto is designed for if one of our other alliance partners has a reliable 5 ball auto. It will take a ball of the bumpers of the 3rd robot and then shoot the 2 that it has (1 from the other robot + one preload). It then drives to the 3rd ball and shoots that.

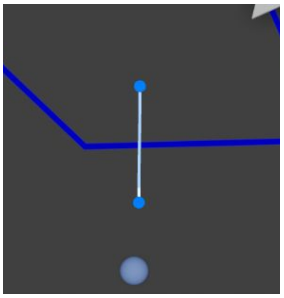


2 Ball Auto

These autos are designed to be used in case our other alliance members have longer autos that they want to run.



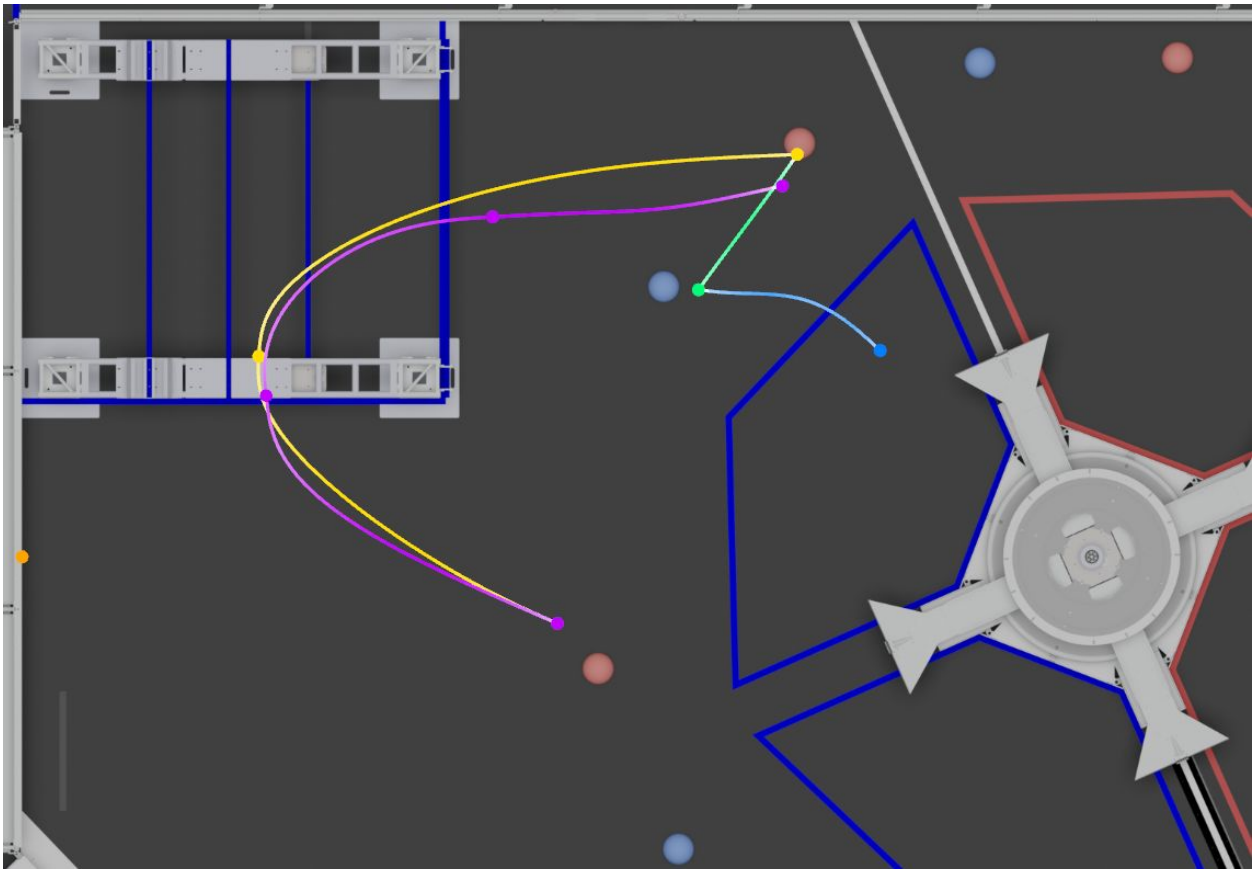




Autonomous (cont.)

2 Ball + 2 Hide

In the case where another robot on our alliance has a reliable 5 ball, we have a 2 ball auto that hides the opponent cargo on our side of the field in the hanger. This makes it so that our opponents have to travel slightly further to get their cargo, giving us a slight advantage. We also travel through the hanger zone to ensure that a third robot has room to back up and get their taxi points.

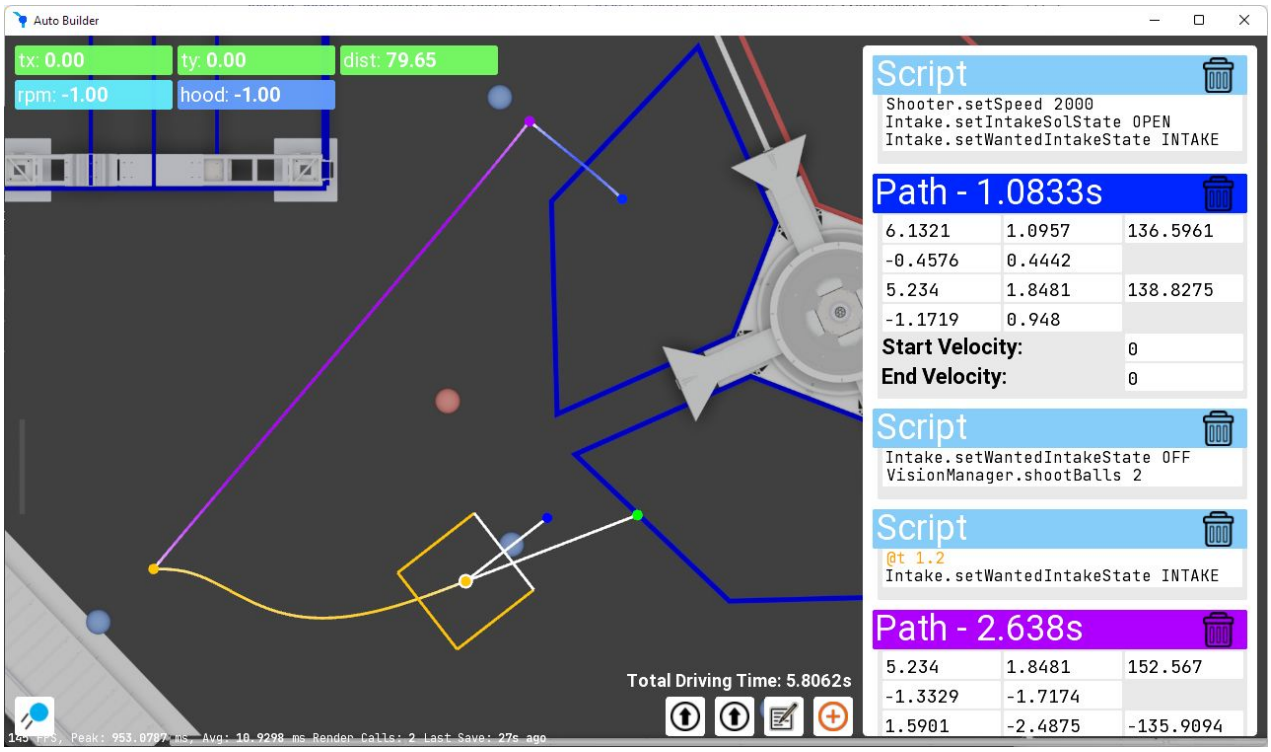


Auto Builder GUI

A fast, intuitive, open-source[1] GUI for quickly designing and iterating autonomous routines for FRC.

Features

- Drag and drop GUI for designing robot autos.
- Supports holonomic (swerve, etc.) & differential drivebases.
- Built in scripting blocks that allow you to execute ANY robot method with no extra configuration.
- Iterate without deploying code. This tool uses networktables to upload autos directly to the robot reducing a normally 20-30s + deploy time to less than 3s!
- It utilizes the same path planner as the robot which allows us to precisely control the robot path and know the exact duration of each path segment. Autos working first try were a common occurrence.
- Real-time feedback. Odometry is streamed to the gui in real time and the path the robot drives is visible in the GUI. This allows us to detect where errors are occurring and quickly correct them!
- Developed from start to finish by students.

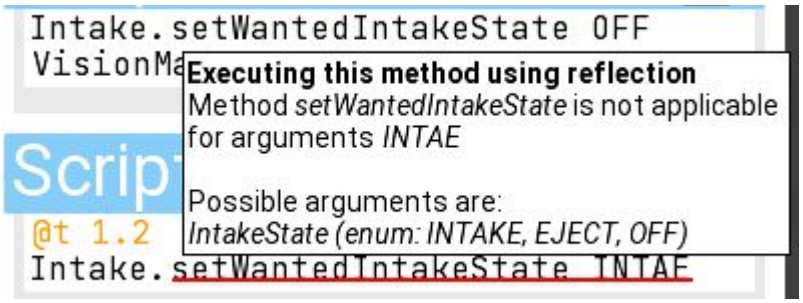


1. https://github.com/FRC3476/AutoBuilder

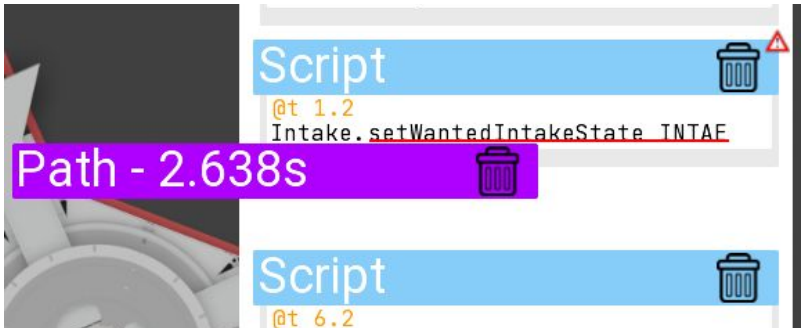
Auto Builder GUI (cont.)

Intuitive Design

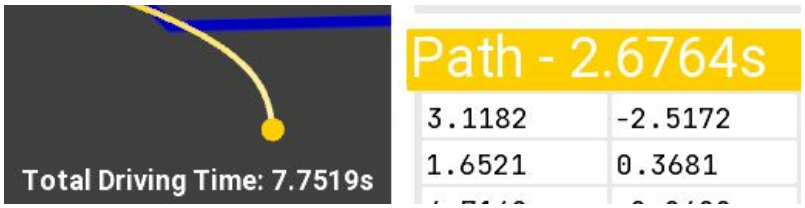
- Full undo & redo support
- Paths are connected. Dragging the end of one path will drag the beginning of the next path.
- Adding/Deleting points is as easy as right clicking.
- Error checking & syntax highlighting for scripts.



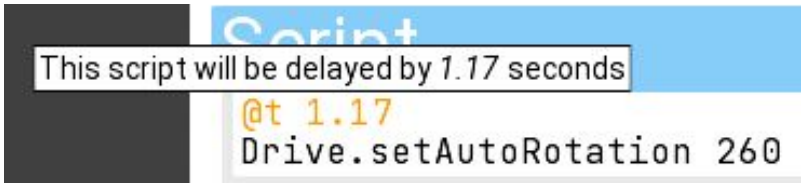
- Drag & drop to change the order of paths around.



- Driving time is calculated programmatically and easily visible.



- Easily schedule asynchronous scripts to run while the robot is driving.




Western Digital
3476
TAP MAGIC


ORDE ORANGE 3476
3476


CAUTION


FIRST ROBOTICS COMPETITION
FRC
SAN DIEGO REGIONAL
NOMAD

Some Final Remarks

Wow. It's been a really crazy few years but we're finally back to doing what we love most. After a full year of virtual meetings in 2021, our team had two years of students to train for the 2022 season. But our new students stepped up to the occasion; this is the first year that we've finished the robot in a timely matter without needing to rush (as much) before competition!

We'd like to thank our parents, mentors, and generous sponsors for giving us the opportunity to build this robot. None of this could have happened without them.

And we'd like to thank you, the reader, for being interested in our robot! If you'd like more information about us, please visit our website: "teamcodeorange.com" There, you can find our previous robots, helpful guides, CAD files, and our Github repository.

Thanks,
Code Orange, FRC Team 3476