GUNN ROBOTICS TEAM

2023 Technical Binder

CONTENTS

THANK YOU TO OUR GENEROUS SPONSORS WHO MAKE EVERYTHING WE DO POSSIBLE

BOSCH

PG&E®

KLA FOUNDATION

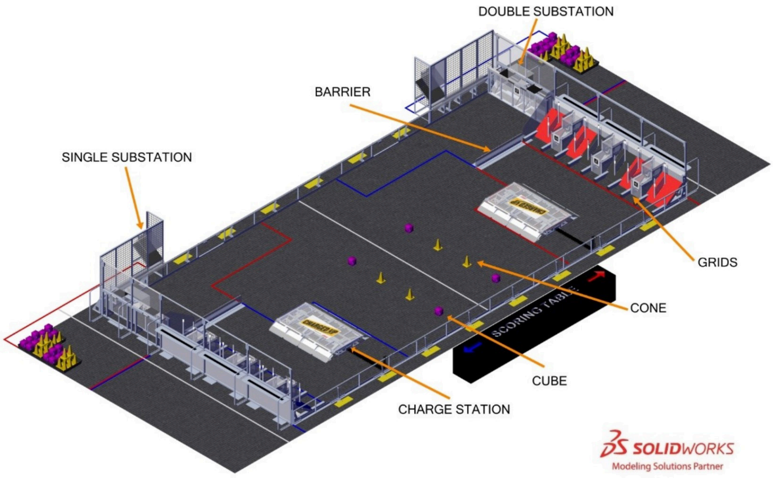
PALO ALTO
UNIFIED SCHOOL DISTRICT

FIRST®
CALIFORNIA ROBOTICS

PALO ALTO
Partners in Education
OUR EDUCATION FOUNDATION

GAME ANALYSIS

In the 2023 FRC season game, Charged Up, robots are tasked with scoring cones and cubes onto 3 levels in the grid. 8 game pieces begin on the ground, and the remainder must be retrieved from the alliance substations. During endgame, robots attempt to dock (drive onto the charging stations) and engage (balance the charging station).



MVP AND GOALS

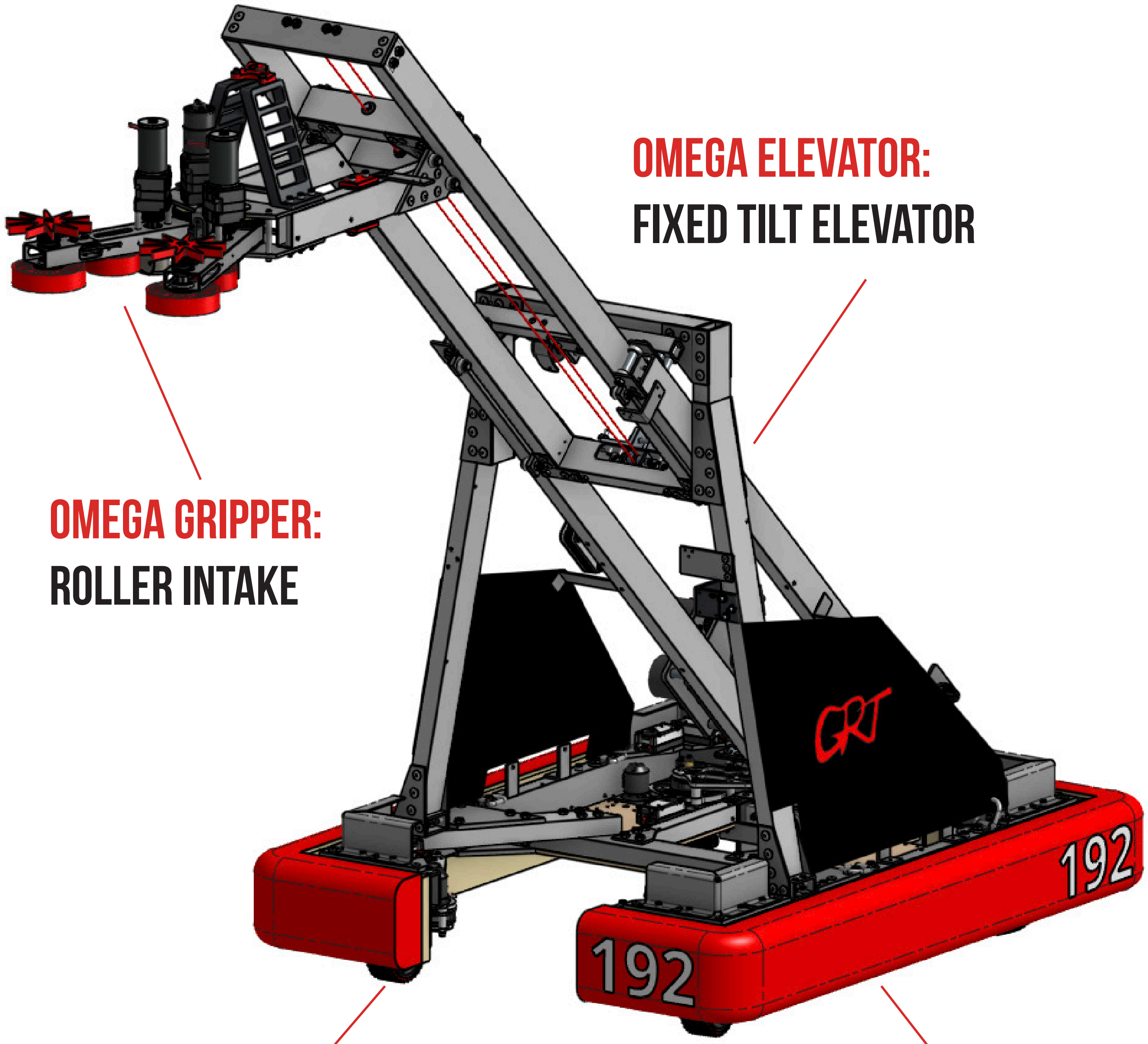
Our minimum viable product for this season was a robot that would be able to score game pieces in the hybrid row by pushing them into place. Our drivetrain also needed to be able to mount the charging station and balance, since engaging with the charging station is the greatest single instance of points in the game.

Of course we were shooting for more than just the MVP. Our target robot would be able to score both types of game elements on any valid grid location. If we failed to reach this target with the arm, we would aim to score both elements on the middle row. We decided to prioritize the ability to pickup cones since there are more cone slots in the grid. We aimed to develop a custom swerve drive for our robot, since the greater maneuverability is attractive to engage quickly on the charging station.

MECHANICAL

INTRODUCING

VERMILION



OMEGA ELEVATOR:

FIXED TILT ELEVATOR

OMEGA GRIPPER:

ROLLER INTAKE

CUSTOM SWERVE DRIVE

REVERSIBLE BUMPERS

DESIGN PROCESS

We decided to pursue 2 mechanisms for our Charged Up robot: an intake and an extension. Since in the past we have felt locked out of experimenting and exploring different designs, we opted to have 4 mechanism groups this year; we chose to develop 2 grippers and 2 elevators in parallel. For each mechanism, we went through a number of iterations to improve the design. About a month into build season, we made the decision to drop our 2 alpha designs and continue forward with the omega mechanisms.

TIMELINE

Kickoff Day — Mentor Design Review

WEEK 1 // *initial CAD & robot 0 prototyping* //

Robot 1 CAD // Mentor Design Review Complete

WEEK 2 *robot 1 cad* // *robot 1 machining*

WEEK 3 *robot 1 machining, manufacturing, assembly, wiring*

End Alpha Designs // Mentor Design Review — Parent Showcase

WEEK 4 *robot 1 testing, robot 2 cad* // *robot 2 machining* //

WEEK 5 *robot 2 machining, manufacturing, assembly, and testing*

WEEK 6 *controls testing and driver practice (and of course, hair dyeing!)*

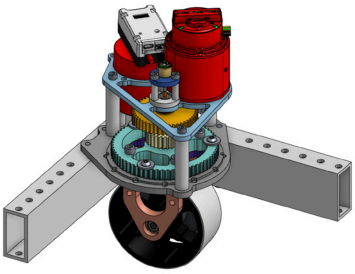
Off to Utah Regional and Monterey Bay Regional!

DRIVETRAIN

DELIVERABLES AND CONSTRAINTS

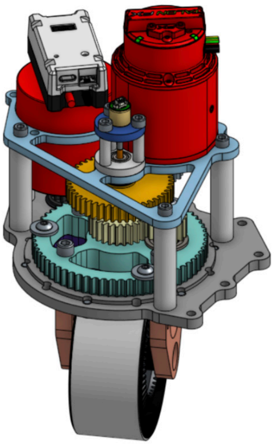
- Must be swerve
- Removable from underneath the robot
- Design ready and approved by the start of August
- Absolute encoder to eliminate encoder values drifting

CAD ITERATIONS



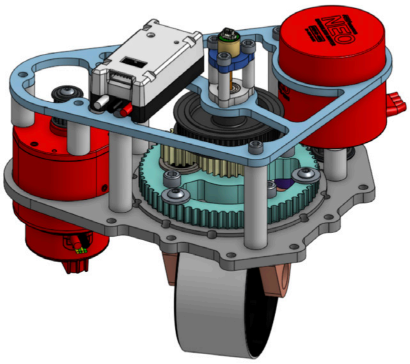
SWERVE 2022-23 - V1

- Removable corners to drop down
- CNC large pulley
- NEO for steering
- proved over the summer



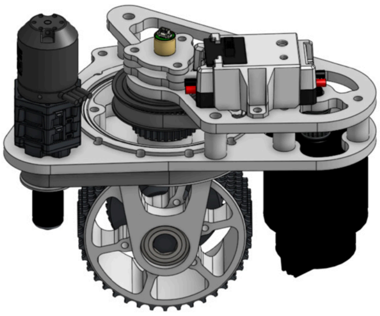
SWERVE 2022-23 - LINCOLN

- Tall and thin



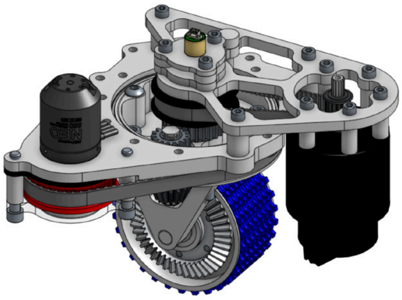
SWERVE 2022-23 - TAFT

- Wide
- Flipped drive motor



SWERVE 2022-23 - MADISON

- Very short
- First design to include a NEO 550
- NEO 550 powers a multi-stage planetary gearbox



SWERVE 2022-23 - KENNEDY

- NEO 550 powers the strain wave planetary gearbox
- Custom wheel

STRAIN WAVE PLANETARY GEARBOX

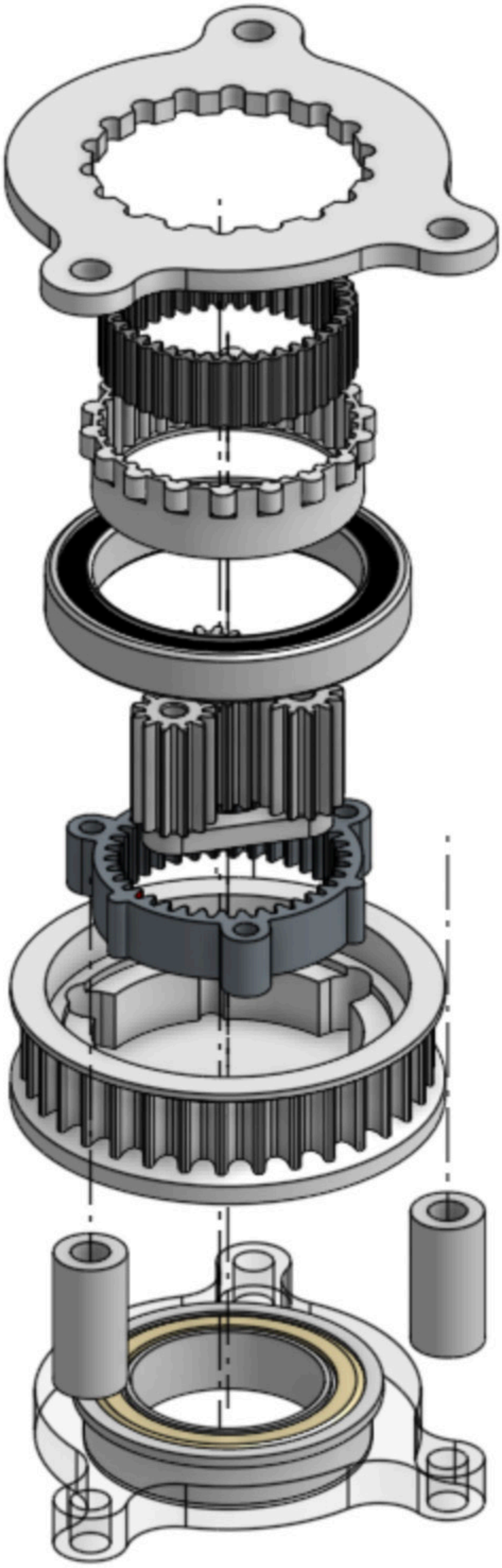
The emergence of Taft Swerve sparked a series of increasingly shorter designs with the goal to eliminate whichever part of the module was the tallest. We flattened the drive motor and gears, and moved the encoder to the lowest-possible position. Eventually, the tallest remaining part was the NEO 550. Neither the motor nor the planetary gearbox could get shorter as they are purchased from REV Robotics. The only way to make it shorter still was to create our own gearbox. It would need a reduction of at least 30:1, which would normally require at least two stages of a planetary gearbox. However, we decided to use a different gearing system, the strain wave gearbox: a reduction that would normally take two or more stages was achievable in just one. The gearbox has very low backlash and is impossible to backdrive.

PROTOTYPES

- PLA ring gear test melted
- PETG ring gears with steel planet gears test used plastic-dissolving lubricant
- Decided on metal ring gears with outsourced laser cut steel production
- Pressed the metal ring gear into a 3d printed pulley
- Laser cut plastic plates to hold a bearing and to connect the gearbox to the main plate
- Laser cut holder to keep the gears equally spaced

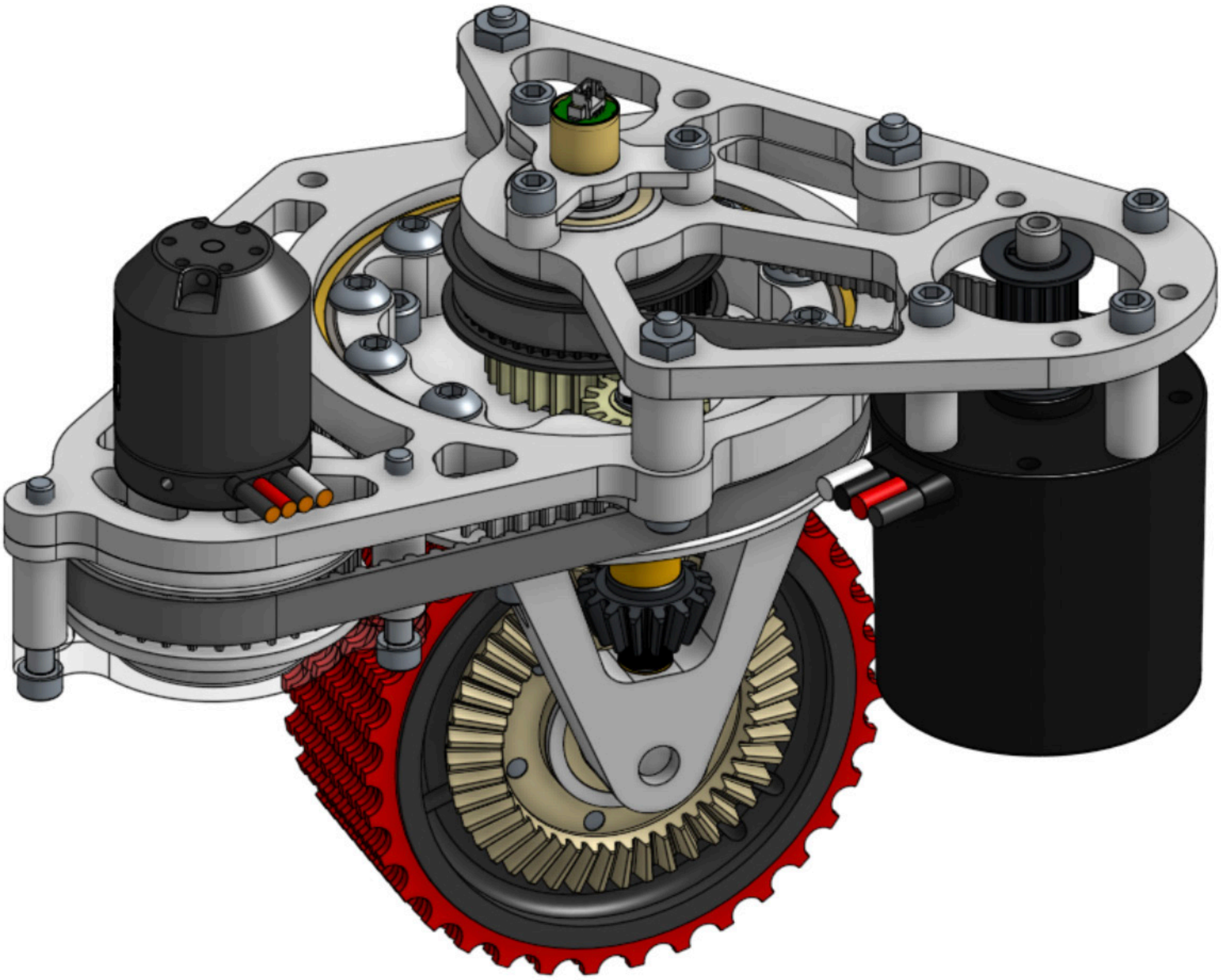
FINAL GEARBOX

- Two ring gears with the same pitch diameter but different tooth counts—one with 36 and one with 39—are constrained to the same rotational axis with a bearing
- Three 32DP 12t planet gears driven by another 32DP 12t pinion on a NEO 550
- Because the difference in teeth of the two ring gears is 3, the teeth perfectly line up in exactly 3 equidistant positions in the gearbox—the "strain wave"
- The planet gears spin around the gearbox, forcing the strain wave along with them
- When the strain wave moves one third of the way around the gearbox, the gears have moved only one tooth—giving the gearbox a 52:1 ratio
- One ring gear is fixed to the main plate while the other is fixed to a pulley that connects to the main steering pulley
- The NEO 550 drives a pinion which drives 3 planet gears which drive an imaginary strain wave, which causes the rings to rotate



FINAL MODULE DESIGN

SWERVE 2022-23 - SORENSON



- Purchased Westcoast Products wheels due to time constraints on custom wheels
- Smaller and thinner encoder holder for easy access to tighten the nut
- NEO motor instead of Falcon 500 due to supply chain issues
- Additional button head screws to retain the pulley
- Washers to hold the main bearing into the plate
- Module mounts to the bottom of the base instead of the top to make charging station engagement easier

GRIPPER

DELIVERABLES AND CONSTRAINTS

- Capable of shelf pickup for cones and cubes
- Capable of ground pickup for upright cones and cubes
- Tight grip on both cones and cubes
- Game elements fall vertically when released
- Fits within robot perimeter when attached to elevator

ALPHA GRIPPER: PNEUMATIC CLAMPING

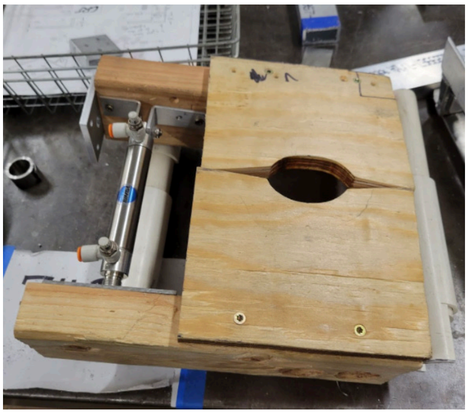
The clamping gripper picks up game elements by physically compressing them. Its large boxy shape perfectly holds the cubes, and the hole on top is able to clamp around the cone. We utilized pneumatics to actuate this compression, which avoided any motor stalling.

PNEUMATIC CLAMPING PROTOTYPES

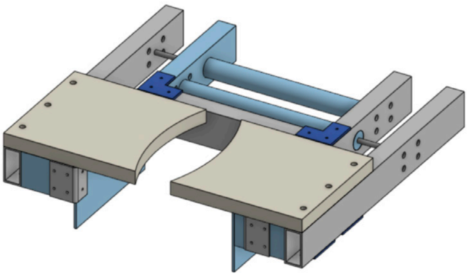
- Built around a single pneumatic cylinder
- Retracted pneumatic length: 6.5”
- Extended pneumatic length: 9”

Issues:

- Resistance between the 2 telescoping PVC pipes
- Heavy weight due to wood
- Low clamping force from resistance and weight
- Alignment issues because the sides of the clamp move together
- Low area of intake

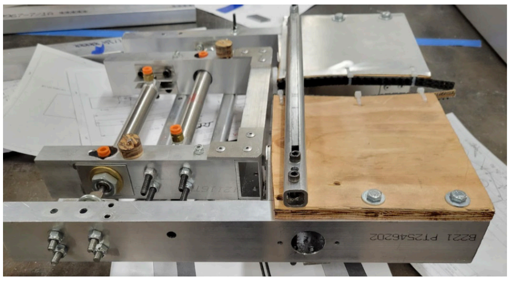


PNEUMATIC CLAMPING ITERATION I

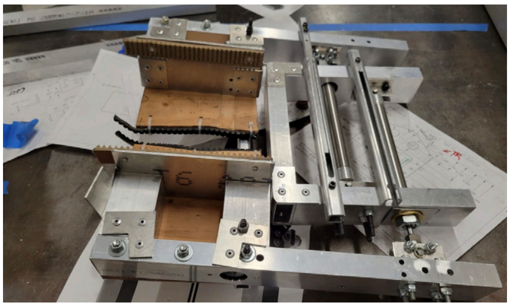


Goals for iteration I:

- Centered mounting bracket
- Increased grip strength
- Increased intake area



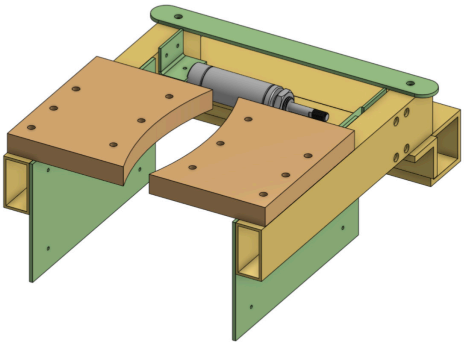
- Clamp built around a central frame housing 2 pneumatic cylinders
- Larger cone gripping region
- Roughtop on gripping surface
- Retracted pneumatic length: 8.5”
- Extended pneumatic length: 13.5”



Issues:

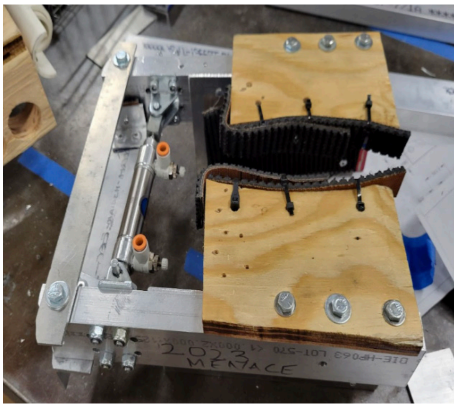
- Still too heavy, (10+ lbs)
- Didn’t fit in the chassis
- Binding gripper due to excessive friction between the rails

PNEUMATIC CLAMPING ITERATION II



Goals for iteration II:

- Rotational clamping to maximize intake area an fit in the chassis
- Dramatically reduced weight (<5lbs)
- 1 pneumatic for lower power draw
- Maintain a centered mount bracket



- Stonger grip strength
- Retracted pneumatic length: 6.5”
- Extended pneumatic length: 7.25”

Issues:

- Harder to pick up accurately
- One side of gripper sometimes sticks on the other or doesn’t close properly

OMEGA GRIPPER: ROLLER INTAKE

The roller intake is capable of intaking both game elements using one set of active rollers with compliant wheels and then clamping with spring force. The later iterations incorporate active release so pieces drop straight downward.

ROLLER INTAKE PROTOTYPES

- V-shaped wooden arms funnel in game pieces
- 2 rotating compliant wheels to intake
- 2 passive compliant wheels to maintain grip

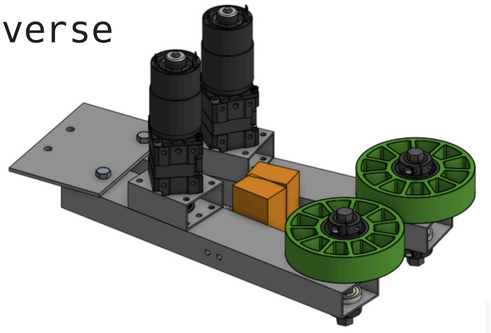
Initial tests proved the versatility. Managed a moderate grasp on cones and a strong grasp on cubes.

ROLLER INTAKE ITERATION I

- Back wheels replaced with wooden blocks
- Releases by running the motors in reverse
- Tried makeshift star rollers

Issues:

- Alignment of the arms
- Cube gets pushed without backing
- Spacing between the rollers

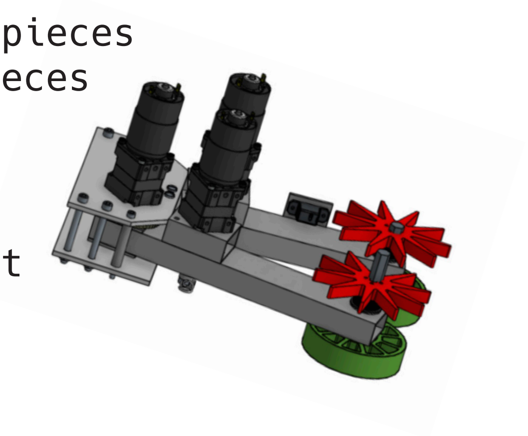


ROLLER INTAKE ITERATION II

- Added motor to open arms and release pieces
- Incorporated star rollers to catch pieces betterAdjusted spring force

Issues:

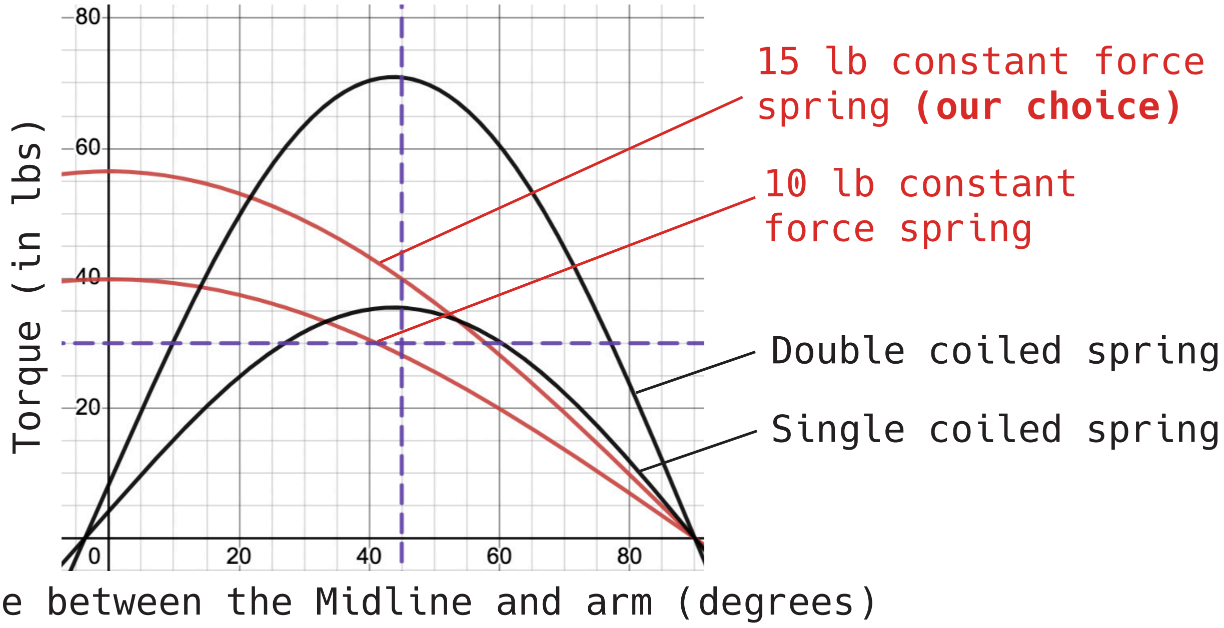
- Motor stalls against spring; could not quickly release the cube
- Mechanism weight
- Limited grip on game pieces



ROLLER INTAKE COMPETITION ITERATION

- Constant force spring solves motor stalling issue during quick release of the cube
- Back wheels fixed in place to provide more grip
- Color sensor added
- Reduced weight of the mechanism by slotting

ROLLER INTAKE SPRING CALCULATIONS



GRIPPER CONCLUSION

Although the alpha gripper (pneumatic clamping) successfully picked up game pieces in isolated testing, the mechanism required high driver precision when mounted to an elevator. The omega gripper (roller intake), on the other hand, required only a loose trajectory for a successful pickup. We opted for the omega gripper on our competition robot due to its higher driver lenience and faster cycle time.

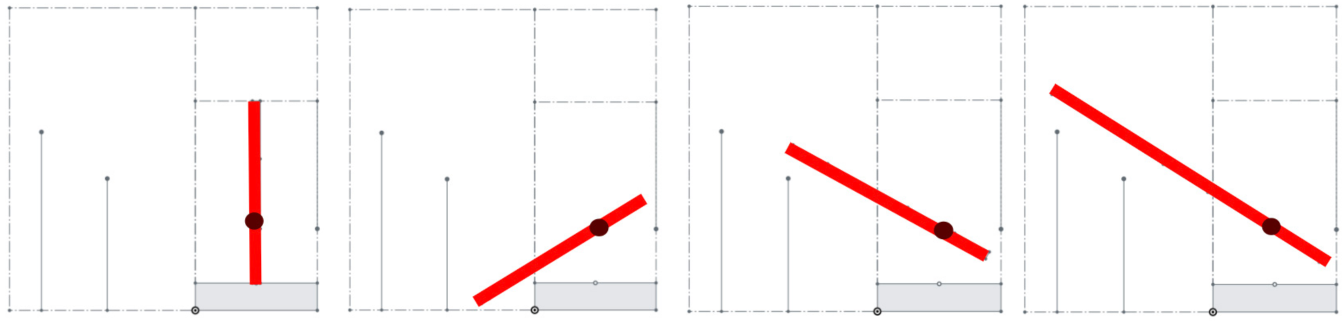
ELEVATOR

DELIVERABLES AND CONSTRAINTS

- Allows for both ground and shelf intake
- Able to reach all levels of the grid
- Retracted state within the frame perimeter height limit

ALPHA ELEVATOR: 180º PIVOT AND EXTENSION

The pivot elevator combines the benefits of a linkage with that of linear extension. Our design is able to fit within space requirements while still reaching every level on the grid. Two benefits of the design from the start were its significant range of motion, allowing precise control and alignment, and the possibility to intake on one side of the robot and outtake on the other.

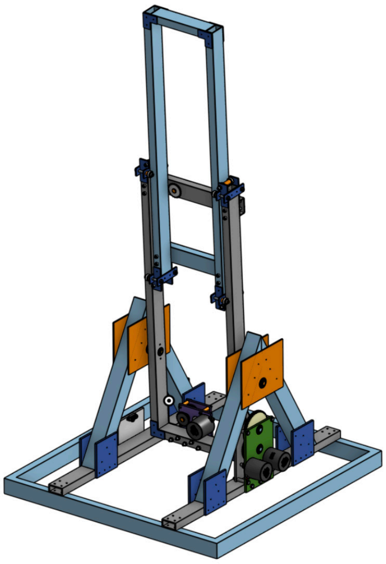


PIVOT ELEVATOR ITERATION I

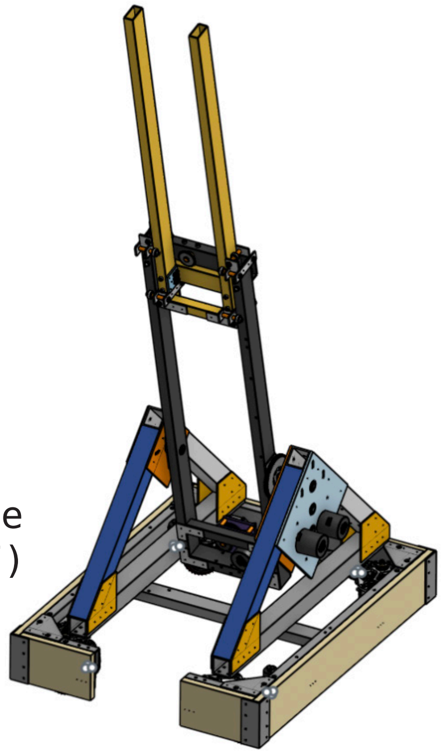
- Extension driven by chain and sprocket
- Triangular frames support pivot axles
- Pivot driven by chain and sprocket

Issues:

- Difficulty assembling
- Chain driving the pivot slipped
- Elevator chain misalignment and difficulty tensioning
- High power draw and torque at pivot



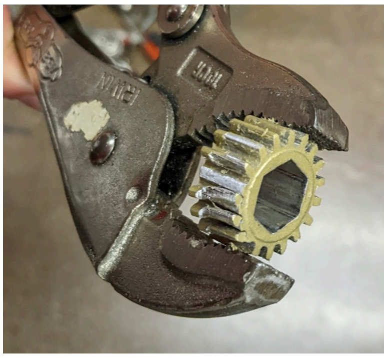
PIVOT ELEVATOR ITERATION II



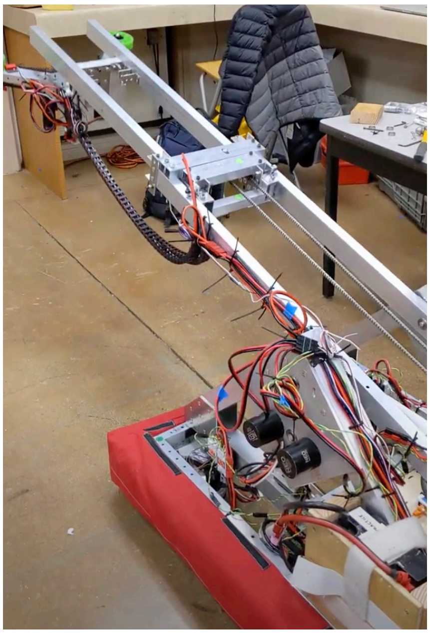
- Pivot triangles redesigned as right triangles with slotting for easier assembly
- Eliminated chain in favor of meshed gears driving pivot
- Properly aligned elevator chain and added a tensioner
- Added 10 lbs of counterweight to balance the torque on the pivot motor

Issues:

- With the arm fully extended, the torque sheared the teeth of our smallest (18T) gear
- When retracted, game pieces are still exposed and susceptible to being dropped or knocked out
- Changing angles required the grabber to operate while not level
- Even with counterweights, torque on the pivot is still huge
- The arm can bounce while pivoting because of gearbox backlash



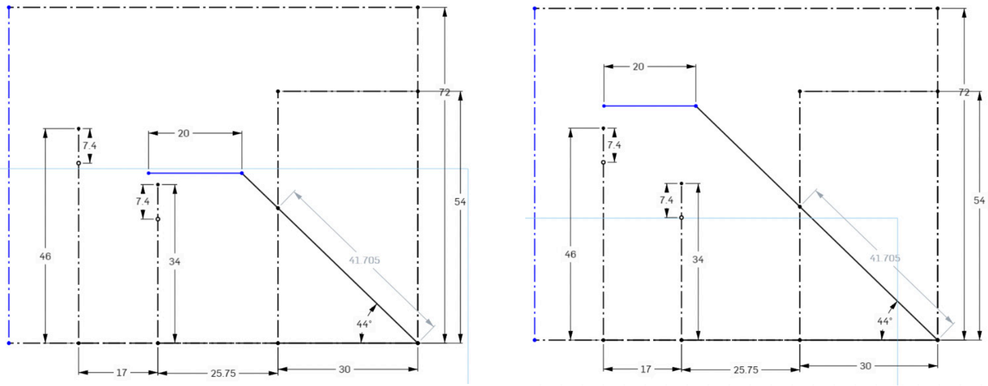
Gear damage (would require a steel replacement or larger gear)



Iteration II test

OMEGA ELEVATOR: FIXED TILT ELEVATOR

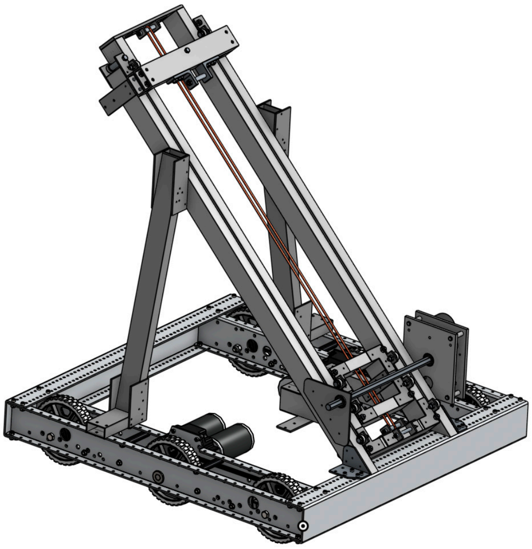
The fixed tilt elevator features a two-stage extension mounted at a fixed angle to achieve vertical and horizontal extension without a pivoting actuation. It can intake from the ground and shelf, and deposit at all three levels, with just a single actuation. The intake is kept level during extension and can be retracted into the base for transit. The elevator is powered by a winch system with cascade stringing.

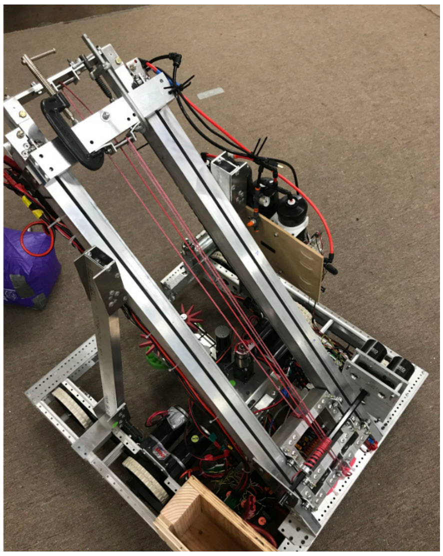


FIXED TILT ELEVATOR ITERATION I

Issues:

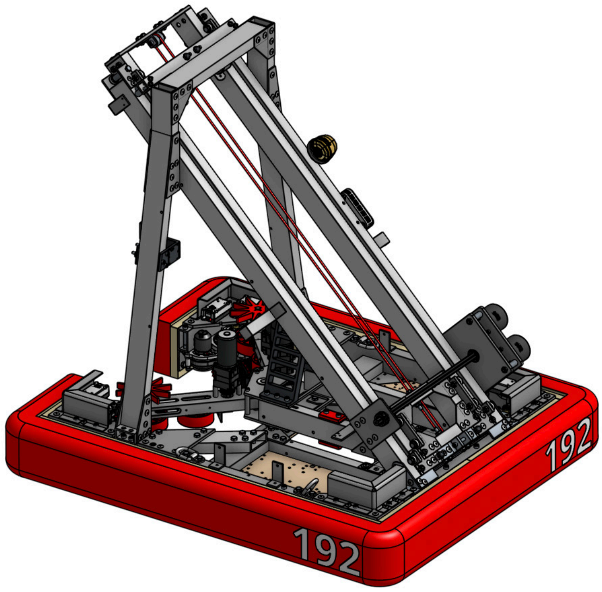
- Iteration I couldn’t reach the top pole
- The base cutout made for a very weak chassis
- High power draw
- Elevator slowly retracts on its own when it is not moving

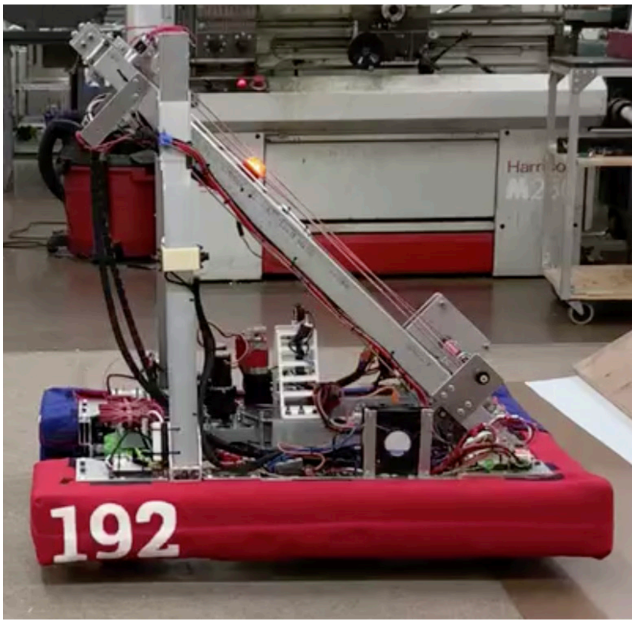


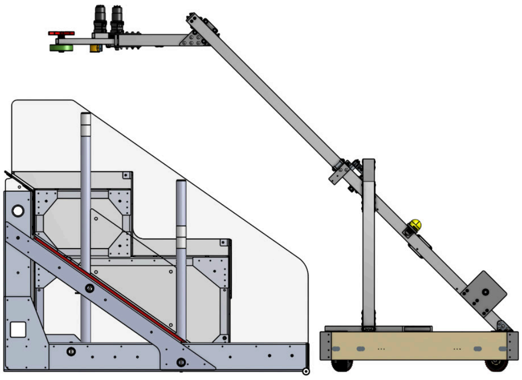


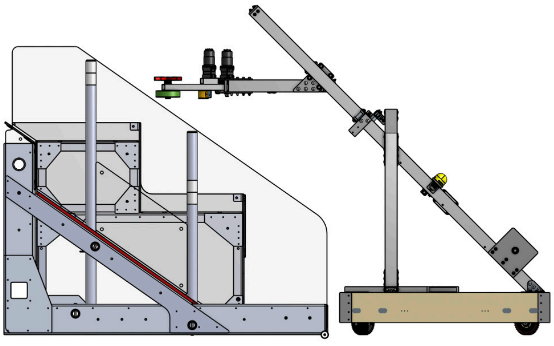
FIXED TILT ELEVATOR COMPETITION ITERATION

- Extended range of motion to ~61” to reach third level
- Added support beams to strengthen robot chassis and elevator against possible collisions during match
- Added constant force springs to assist the winch raising the elevator
- Added a downstring to retract the elevator









ELEVATOR CONCLUSION

Although the alpha elevator (180º pivot and extension) was promising and would meet our goals with another iteration or two, the omega elevator (fixed tilt elevator) was much closer to being competition ready at a crucial decision point. We opted for the omega elevator on our competition robot due to its simplicity and stability.

SOFTWARE

ELEVATOR (STATES)

When we chose our fixed tilt elevator design, we knew that the driver should not control the exact position of the elevator. The elevator has to move quickly to precise positions with accuracy unreasonable for a human. This led us to devise our state machine: The mechanism driver uses various buttons on their controller to move quickly between set states, named heights of the elevator, and our PID Controller does the work of moving the elevator carriage to the correct position. We didn’t want to remove the possibility of real-time driver adjustments to these states, so we also implemented a driver offset, where our driver can manually change the height of the elevator compared to the current set state. We also improved drivability with drop sequences.

STATES

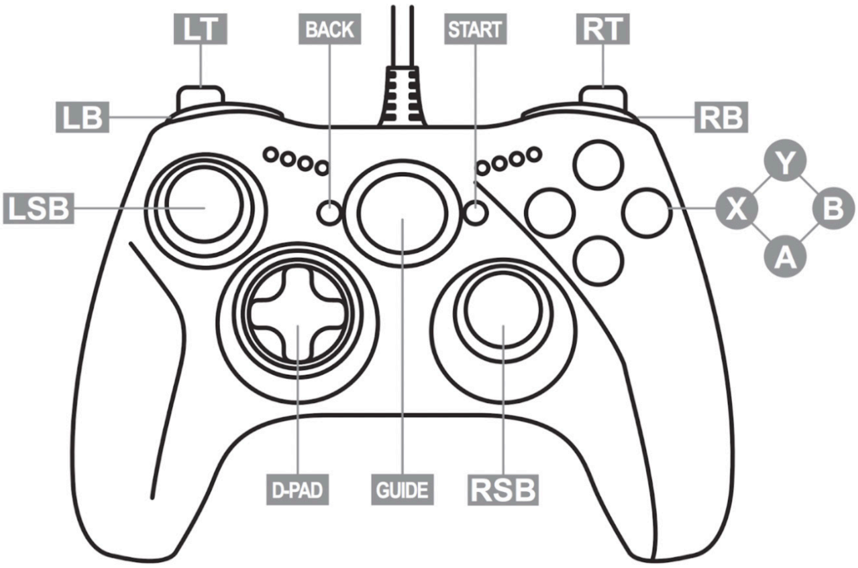
Ground: Ground is the home state of the robot, since the robot fits within the frame perimeter and the piece it is holding is protected. To avoid dragging, the ground state automatically becomes 5 inches taller when the robot detects a piece is being held.

Intaking States: The states shelf and chute are for intaking. We found that our mechanism was very unreliable in general when intaking at a raised position from the chute, so we do not use that state (instead we drop the piece from the chute onto the ground and intake from the ground). The shelf state works as intended, raising to just above the shelf height so our rollers can grab a piece.

Outtaking States: We have states for each height of grid pieces (hybrid, cone middle & high, cube middle & high). We also have an auxiliary state, cone middle drop, which is only used when initiating the drop sequence from cone middle. The elevator moves from cone middle down to cone middle drop, where the pole is already slightly inside the cone, before releasing its grip on the cone. This prevents the possibility of the cone falling behind the post and the cone snagging on the post when the robot moves to the grid.

CONTROL SCHEME

RB: Alternates between cube middle and cube high
LB: Alternates between cone middle and cone high
D-PAD: Top–shelf state
Right–hybrid state
Bottom–ground state
Left–resets offset
LS(vertical axis): Change offset
A: Initialize drop sequence
Y: Drop piece without drop sequence (seldom used)
RT: Run rollers to intake
LT: Run rollers to outtake (without drop sequence)

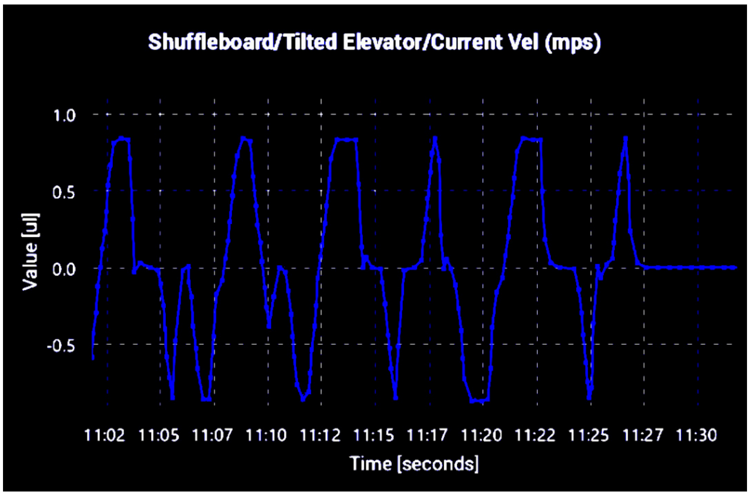


PID CONTROLLERS

PID controllers are a form of closed loop control, where sensors determine outputs which in turn change the value of the sensors (creating a “closed loop”). PID controllers calculate error between the current position and the goal position of a mechanism (in our case the height of our elevator). PID stands for proportional integral derivative, the three values from error that are each multiplied by a constant (your PID values).

PID CONTROLLERS CONT.

Velocity-based: Originally we thought a velocity PID would be ideal for our situation. Our Velocity PID used a library to create a trapezoid on the velocity/time graph, with its slope being a set max acceleration and its height being a set max velocity. The total area under the trapezoid is equal to the distance intended to travel. The PID control calculates error based on the difference between current velocity and the target velocity and uses the PID values to correct the error. The graph on the right shows multiple trapezoids (some upside down). The velocity-based PID jerked when stopping. The velocity was too high on the downslope of the trapezoid, and the intake was not slowing down early enough. Instead of complicating the problem further with a second PID for the latter half of the trapezoid, we decided to work on a simpler PID.



Position-based: Position-based PID initially seemed like the wrong solution for our problem. The main driving force of most PID controllers is the P value, and when changing states, the power given to the motors from the P value is massive, due to the large error incurred when switching states. This leads to extremely fast, and therefore jerky, starts, which would make the piece in our intake fall out. We decided to switch to a position-based PID when we discovered a ramp-rate feature on our motors, which acted effectively as a maximum acceleration (in reality it is capping the rate voltage being fed to the motor can increase). This led to a very smooth acceleration and deceleration, as well as a high velocity in between, thus solving our PID problem.

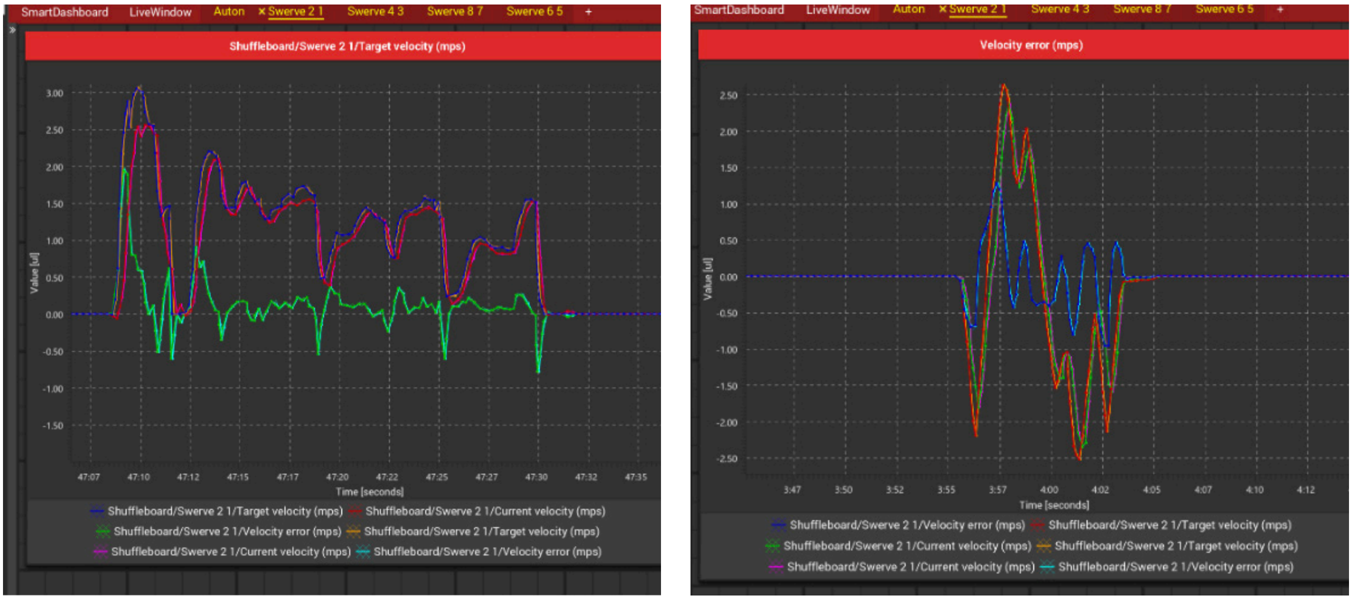
DROP SEQUENCES

We created drop sequences in order to reduce the number of buttons pressed by the mechanism driver. A drop sequence is started when the mechanism driver pressed the A button and consists of:

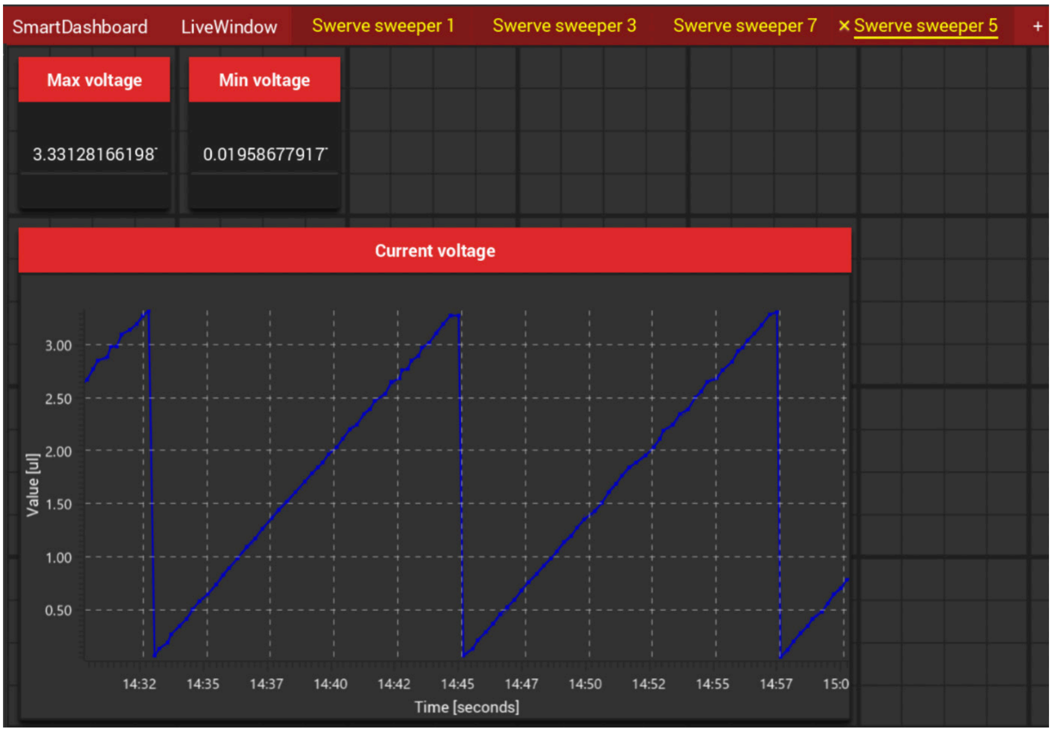
1. **Moving to an optional drop position**—only used with the middle cone state, ensures that the cone is completely above the peg before dropping
2. **Dropping the game piece**—opening the intake and/or running the rollers in reverse depending on the state the drop sequence was started in
3. **Reversing swerve**—move the entire robot back a few inches so that the intake doesn't hit any parts of the grid or any placed pieces on the grid on its way down
4. **Returning to ground**—the elevator moves back to the ground position, the drivers can move the robot during this time

Movements of the robot are locked for the drivers until the elevator returns to ground so that any accidental inputs do not disrupt dropping the cube and so that the drivers may hold down any inputs they plan to use just as the sequence ends.

SWERVE



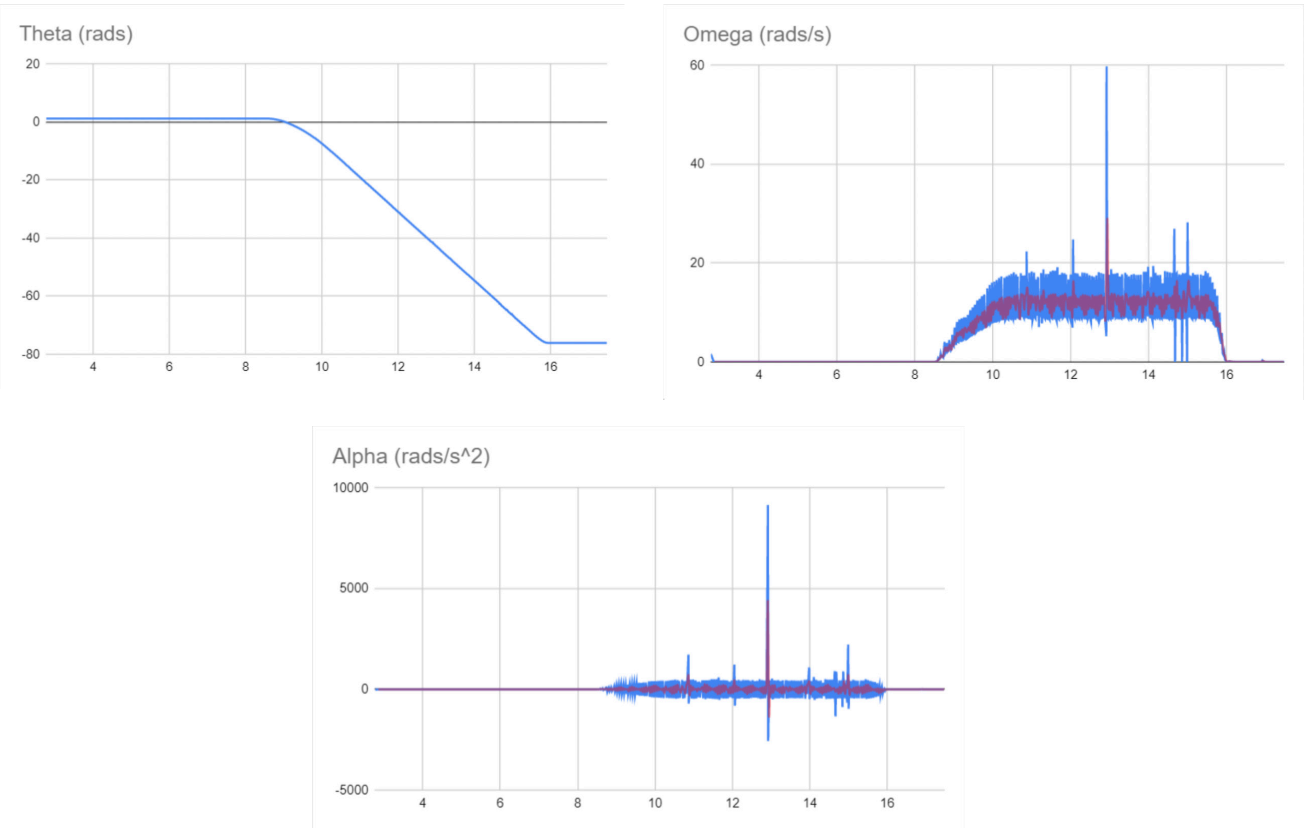
Tuning of PID for swerve drive motor



Max voltage 3.33128166198

Min voltage 0.01958677917

Compensation for encoder reading variation



Data collection and processing to develop a model of our system. Raw data is graphed in blue, a rolling average is applied to remove high-frequency noise, and the output is graphed in pink.

AUTONOMOUS

In autonomous, we plan for our robot to score points and navigate the field as accurately and as quickly as possible. This is done with our auton sequences. These sequences are a combination of several commands that are called in a specific order.

Our intake command runs the roller to intake a piece, and will stop when the limit switch inside the roller is pressed. Our drop command is a drop sequence that places a game piece. Because this command is also used during teleop, it allows the robot to run up against the grid dividers with its bumpers, and then drop the game piece. Drop command first lowers to a target height, which depends on the current place height. Then it calls a roller place command that widens to roller intake, subsequently dropping the game piece. Finally, the robot backs up a few inches from the grid, so that when the elevator lowers again, it does not collide with the cone or cube nodes on the grid. Elevator height command sets the requested target height for this drop command.

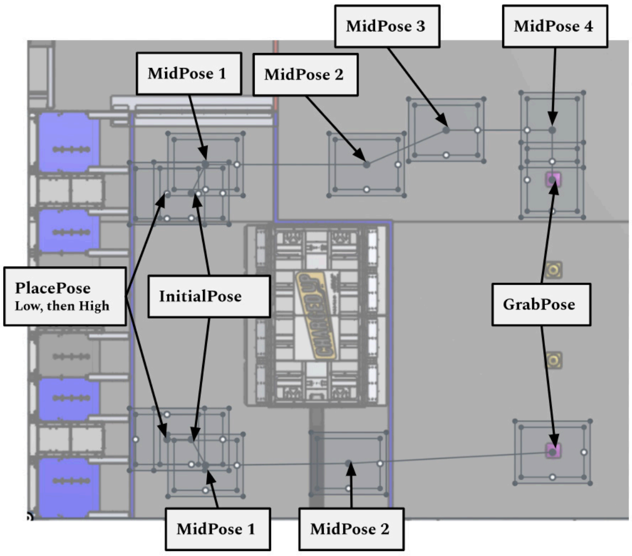
AUTONOMOUS SEQUENCES

We split up our auton paths into five separate sequences. Two paths on top, one balancing path in the middle, and two paths on the bottom. We stored the blue xy positions, and a boolean chooses whether to mirror these positions or not, based on which side our alliance is on. The top and bottom paths each have a 2 piece auton, and 1 piece auton.
In two piece autonomous, we first start 12 inches behind the grid dividers, before traveling forward against the dividers and placing our pre-loaded piece. In all of our auton sequences, the initial pose, and place poses are choosable, and will be determined as we communicate with our alliance. MidPose 1, as shown in the diagram below, is placed for the robot to avoid collision with the charging station, as it exits the community. For the top sequence, the robot will then turn 90 degrees, and pick up the second

AUTONOMOUS SEQUENCES CONT.

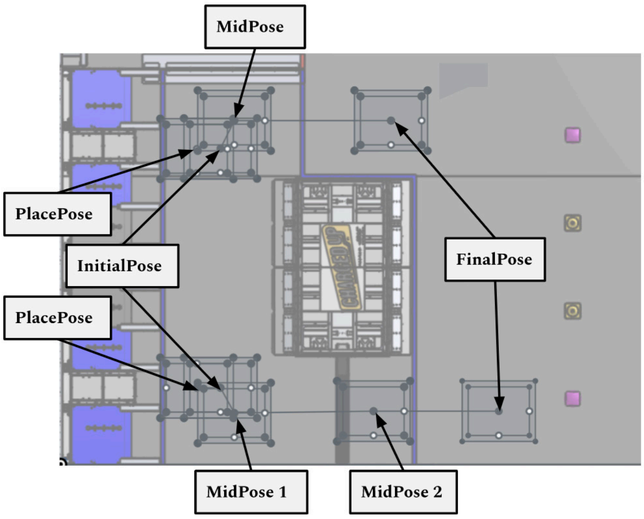
game piece at GrabPose. It will then retrace the MidPoses back into the community, before placing that game piece. We decided to grab the game piece from the top, as the 90 degree turn would take less time than a full 180 degree turn.

TOP AND BOTTOM 2 PIECE AUTON



For the bottom sequence, the only difference is that a 180 degree turn was chosen, as there is much more of a risk for bumping into the wall, or into another game piece. This also reduced the amount of bottom midposes needed.

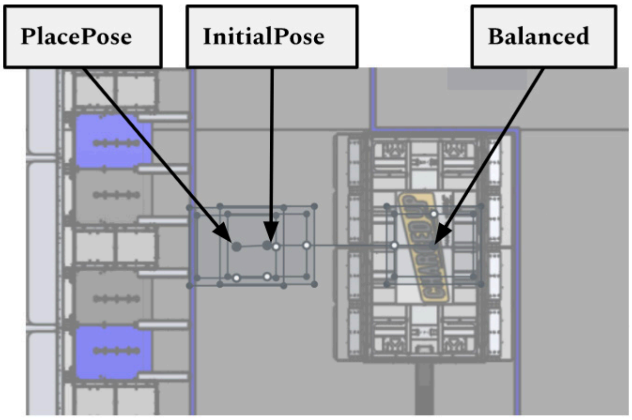
TOP AND BOTTOM 1 PIECE AUTON



One piece top and bottom auton was created due to time constraints after two piece auton was observed to take longer than 15 seconds. Two piece auton was also not making the full 180 degree turn to pick up the piece on the ground. In one piece auton, the robot would still place it's preloaded game piece after traveling from initial pose to place pose, but the robot would then taxi out of the community and stay there, instead of also going for another piece.

The Balance auton sequence also places its pre-loaded game piece, but then immediately transitions to balancing on the charging station. It drives up the charging station backwards, and balances under fifteen seconds.

BALANCING AUTON



AUTOBALANCING

Since the beginning of the season, it has been our goal to use the charging station to score points in both the autonomous period and the tele-operated period. To score the maximum number of points, we would need to balance on the charging station two times— once without human input of any kind. This meant we needed to thoroughly analyze the rotational behavior of the charging station and develop a control system that could balance our robot in a reliable and efficient manner.

We immediately started testing different control methods to maximize accuracy and minimize the time taken. We initially thought that a PID (Proportional, Integral, Derivative) controller would be enough to level the charging station by changing the drive speed as a function of the angle. However, results with this method were wildly inaccurate, resulting in the robot overshooting the balance point and sometimes even driving off the edge. As we figured out, the charging station had non-discrete rotation: the pitch of the charging station did not change proportionally as the robot moved across it.

This added another variable to our control system. We decided to factor in the change in pitch of the robot between sampling cycles as a metric of whether the charging station was rocking or not. This allowed our control system to account for how the charging station was rocking in addition to the error in pitch.

AUTOBALANCING CONT.

Using the pitch error and Δpitch, we developed three different balancing methods that bring a different approach to balancing. With the ‘DualPID’ method, we assign chassis speeds based on the sum of a pitch PID and a Δpitch PID, with the setpoints for both being 0º, meaning the robot is level and not rocking. With the ‘PIDswitch’ method, we separate balancing into ‘rough’ and ‘fine’ adjustment. Based on the current pitch and Δpitch, we choose the rough/fine PID such that the robot is able to give a little more power when it is trying to get onto the charging station and a little less power when it is trying to do the final tuning. Our last and most frequently-used method uses our understanding of the charging station’s behavior to create a chain of events. Upon getting onto the charging station, the robot drives up the slope at a slow, constant speed until the center of mass passes the fulcrum (identified by the pitch of the robot crossing 0.0º). The moment that happens, the robot goes into a fine adjustment PID period. This vastly reduces the amount of work the PID has to do and reduces the opportunity for overshoot to happen.