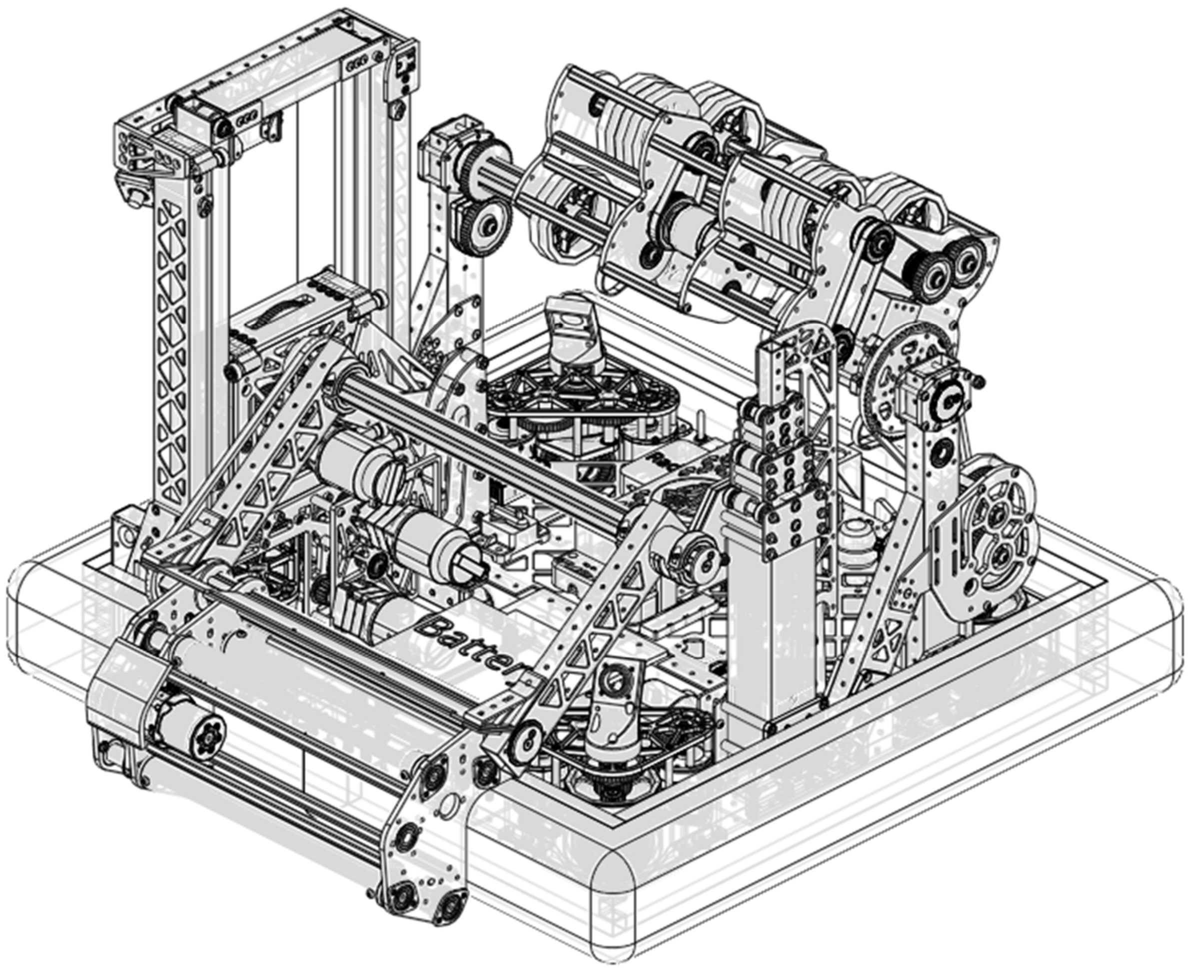
CRESCENDO SM

PRESENTED BY HAAS Gene Haas Foundation

PROUD MEMBER



PROTEUS

WOLVERINES

1757

2024 TECHNICAL BINDER

FORWARD

Hello, and let us welcome you to FRC Team 1757's 2023-2024 Season. We are coming off the teams most successful competitive season ever: in 2023 we not only won our first Ever Blue Banner at WPI Week 5, we also won Engineering Inspiration for the first time ever at the same event. Then 168 hours later in Springfield we did something we honestly never thought possible. After finishing Qualifications ranked 1 overall on Mier Division, we went on to Win the Division, Then After that we went on to Win the Overall New England District Championship when we defeated the Wilson Division Winners. Needless to say, we are clearly punching well above our weight and with almost 90% of our students returning we think the sky’s the limit this season.

Our season started in the fall of 2023, introducing a new class of 10 freshmen, sophomores, and juniors to the world of FRC. We showed off the robot at local town events, built a T-Shirt Cannon to raise school spirit at the prep rally, and hosted weekly technical seminars on everything from the engineering process to CAD, Electronics, Pneumatics, Mechanics, and everything in-between. We also worked on a radical redesign of the intake for our 2023 robot “Luxo” as a fun student driven effort to be more competitive at the off season events we planed on attending. We traveled to Tewksbury, MA in October to compete in the New England Robotics Derby where we – for the second year in a row – fell in the finals to take 2nd place. After NERD we then trekked to Manchester for River Rage 26 where we fell in the semifinal round. Satisfied with our Preseason development we piled into our classroom on a cold Saturday morning in January, eagerly anticipating this year's game. 5 CAD models, 8 shared Google Drives, ten weeks, 20 Weekend Build Sessions, 50 Zoom calls, 5799 lines of code, 104 git commits, 19,129 discord messages, and many, many cups of coffee later, we are proud to unveil our robot "Proteus" for the 2024 FRC Season.

Proteus’s namesake is one the ancient Greek gods of the sea known for his versatility, we designed the robot to be a fast nimble bot that could adapt to handle any aspect of this years game and as such we thought Proteus was a fitting name.

This year we continued our participation in the Open Alliance. We have found the Open Alliance teams and their open and timely build season updates so helpful to our team and we are thrilled to share our own progress and hopefully help other teams the same way the alliance has already helped us. If you want to learn even more about our robot and the design process, beyond what is contained in this manual, please visit our Chief Delphi Build Thread at https://www.chiefdelphi.com/t/frc-1757-wolverines-2023-2024-build-thread/442959

We hope you enjoy this brief look at the design process and technical details that went into this robot, and if you have any questions, look for one of our team members in the stands, in the pits, or on the field. We are always ready to share the knowledge we have gained and share a few hard-learned lessons we learned along the way.

SPRINGFIELD
OR BUST!

TABLE OF CONTENTS

GAME ANALYSIS

Every FRC season starts the same way; we gather together as a team, watch the kickoff stream, then hunker down and break down the game in back-to-back 8-hour build sessions. The hope is that by the time we walk out the door on Sunday night, we understand the game and know what we are doing.

Our process is heavily influenced on the format developed by Team 125 the neutrons which involves a systematic breakdown of the game.

This process begins with identifying every individual skill a robot would need to accomplish any part of the games tasks broken down as granular as drive forward to as complicated as being able to pick up game pieces off the floor without imput from the driver. A new record for this year, we identified **129** potentially useful robot skills that we might expect to either be barebones required or beneficial to the season. These robot skills aren't necessarily the *best* ways to do given tasks, so we then followed with which scoring objectives, and which skills complete which scoring objectives by number
This list is in no ways exhaustive, a lot of the sensor/indicator parts are generally helpful for entire bot, so aren't always reflected in the beneficial column

After this we identify all the ways the robot can score a point in the game. After we map out every way the robot can score we go back to our list of skills and determine which skills are needed to do each of the scoring objectives. Afterwards, we conduct time-based analysis, taking into account the maximum number of match points and the time needed to complete those tasks. Obviously, those with the largest point values and the least amount of time to complete will receive the biggest consideration during build season.

Based on our analysis, traps are very useful (representing 25% of a ranking point). We predict that a 6 piece auto is possible, but because the game pieces in the center it's very unlikely. We predict that, for a fast enough robot, a large number of cycles is very possible, meaning that the game piece RP is achievable in the absence of defense. Given the lack of large safe zones like last year, some of these numbers aren't as possible, for instance while our estimates put 22 cycles in teleop achievable, realistically it will not happen given any number of impedances.

This game will be fast paced! Here were our primary take aways. we came up with the following ideas:

- With fast cycles possible, and an open field, swerve is a no-brainer especially with modern enhancements to increase speed (such as the 16t pinion for Mk4is 4).
- Climb seems like a bare-minimum competitive requirement, so as we are competing in weeks 4,5, and if qualifying 6 and 8, we can expect robots to have some form of climber, but trap's point value is nonetheless important.
- a bare-minimum bot should have a subwoofer shot, and podium should be our must-have for competitive advantage
- the gamepiece may behave unpredictably in the air without spin, being able to control the spin on the note allows for variable shot and fine-tuned control of the gamepiece

Identifying Design Constraints

After the initial kick off weekend we spend approximately a week and half deep in the design identification phase to debate and settle on a overall robot Archetype

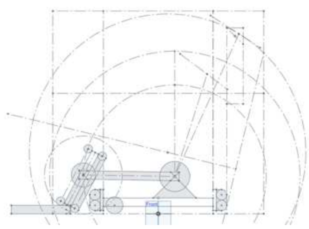
ARCHETYPE CONSIDERATIONS

If we wanted to be self-sufficient in earning ranking points, especially in endgame, earning the TRAP points is a necessity. As such, we realized a one-degree-of-freedom arm would not suffice. We then started to think about a two degree of freedom arm with a pivoting wrist in order to score into the trap, which would mean we would need a higher left in order to score.

◆ One degree of freedom arm, Quokkas style

◆ Two degree of freedom arm

can score trap as long as it picks itself up a considerable distance



◆ Team 148's 2023 Design: Pivot + Elevator + Pivot

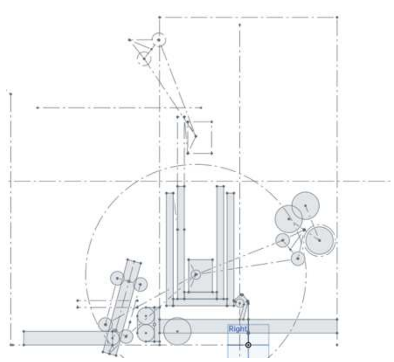
Can score trap, needs a secondary lift mechanism, unsure about stability at the end effector to creating repeatable shots. We then considered other archetypes such as the davidbot, or [Robot (CAD) in 12 hours 2](). While we liked its simplicity, we were unsure of the rigidity we can achieve with an elevator, the handoff, and adding another motor for indexing for the handoff

◆ Team 342's 2024 Design

It solved a few of our issues, most notably that the shooter was separate, allowing for more rigidity in the shot.
One of the most notable downsides was we would need to figure out how to rotate the arm 180 degrees about the carriage, while still maintaining the required rigidity

◆ Team 342 + Quokka's Hybrid Design

We then discussed taking the Quokkas design of a mixed shooter-intake combination, put onto the 342 archetype. While the arm would then not need to rotate a full 180 degrees, only 90 at most, it still has the variability from an elevator's rigidity, which we have never manufactured properly in the past.



◆ Team 2538's 2018 Design

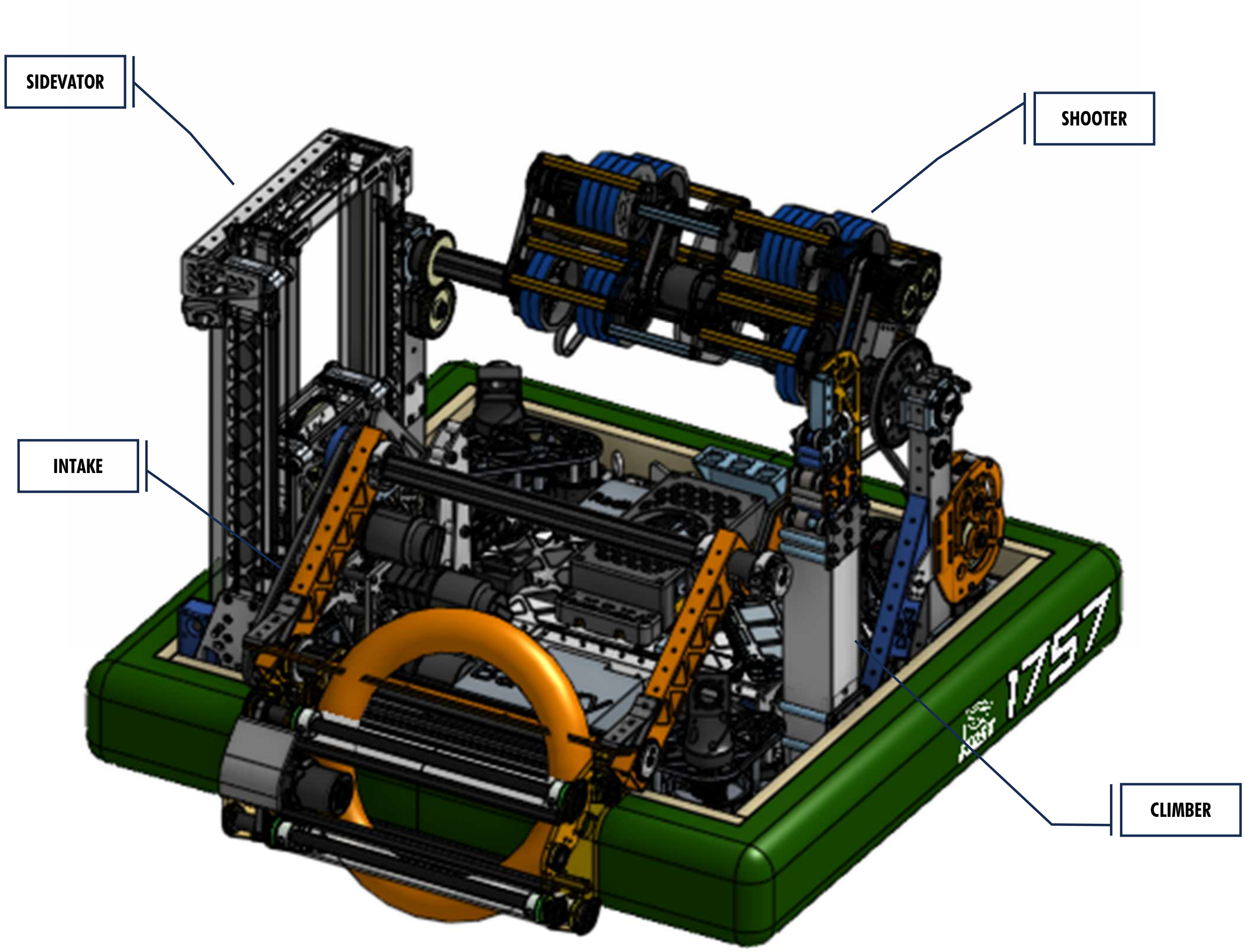
As we were looking at rigid elevator designs, we stumbled upon Team 2538's 2018 Design, which had a cantilever intake/shooter from a sideways elevator. This solved the problem of having to maintain a rigid elevator, while being able to rotate 180 degrees. Our technical captain then stayed up until midnight (by his own choice) to make a master sketch of what this would look like. It has ground intake, source intake, trap, intake is rotating about on an elevator, which passes off to a fixed position shooter. For amp scoring and for trap scoring it runs the intake in reverse.
There are fixed hardstops which allow for a static position to work well.
We also are looking at adding forks to balance the robot while it climbs.

After 2 weeks of discussion this is the Archetype, we moved forward with

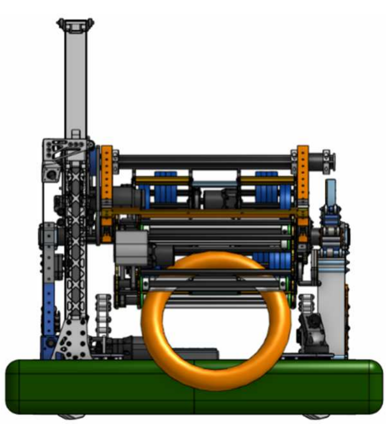
THE 1757 RAPID DEVELOPMENT MODEL

- Clearly Identify the design requirements of the system
- Design and Build a prototype that can be used to test design assumptions and Test
- Use what we learned from testing to develop a final design
- Fabricate final version and intergate into overall robot systems

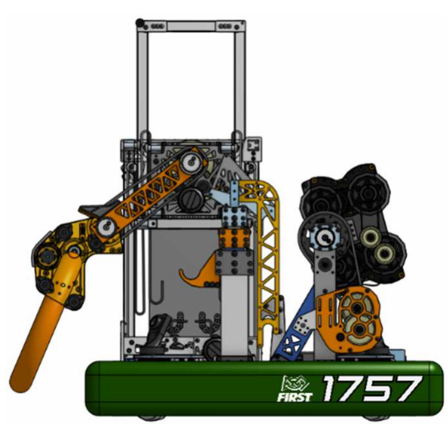
Final Robot Design



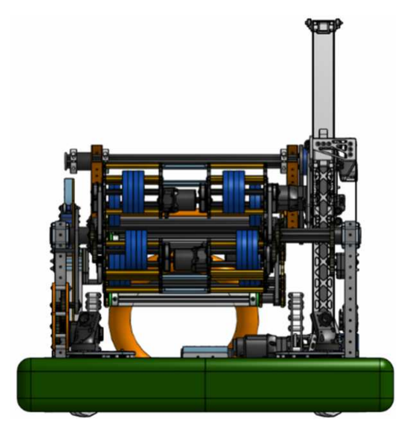
FRONT VIEW

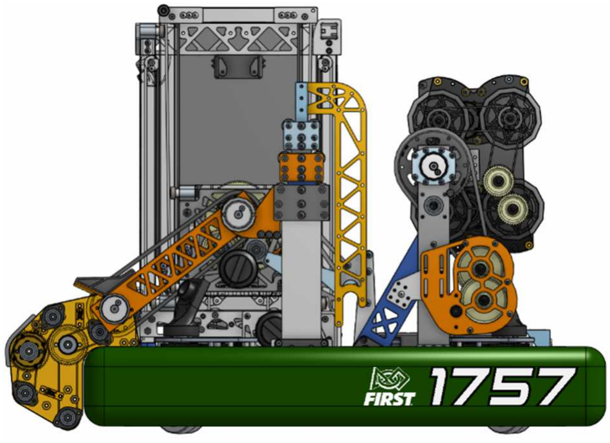


SIDE VIEW

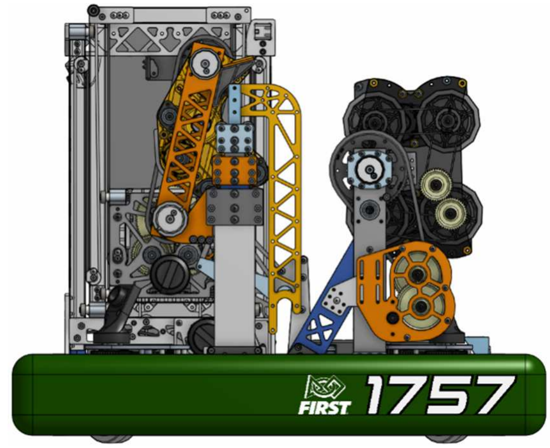


REAR VIEW

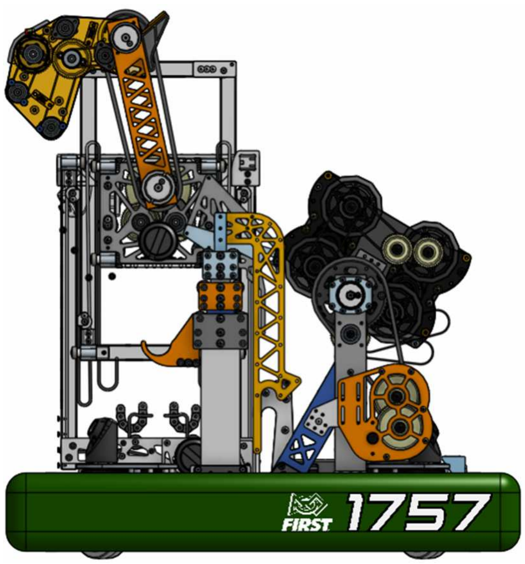




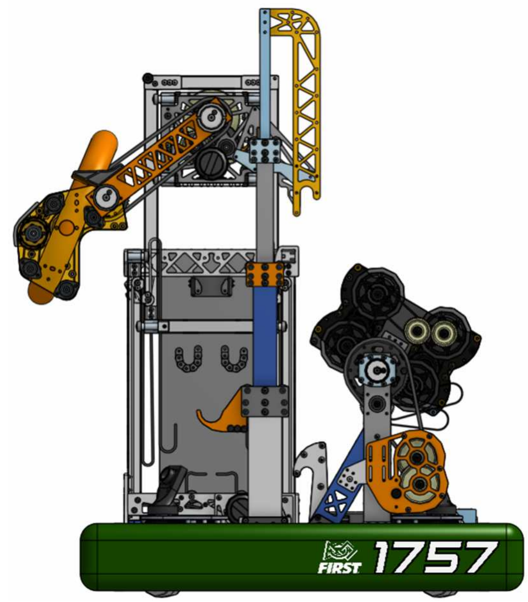
FLOOR PICKUP



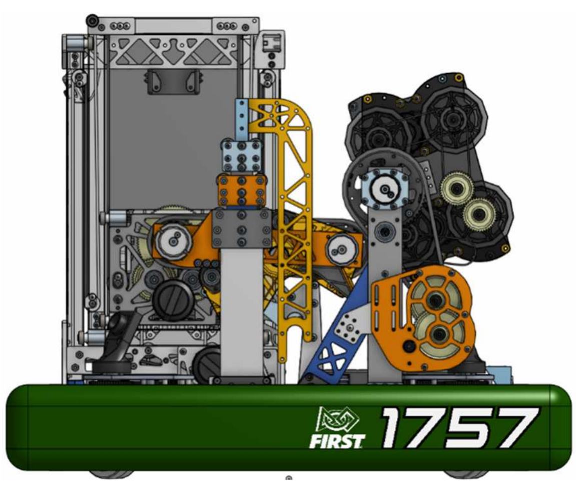
TRANSIT



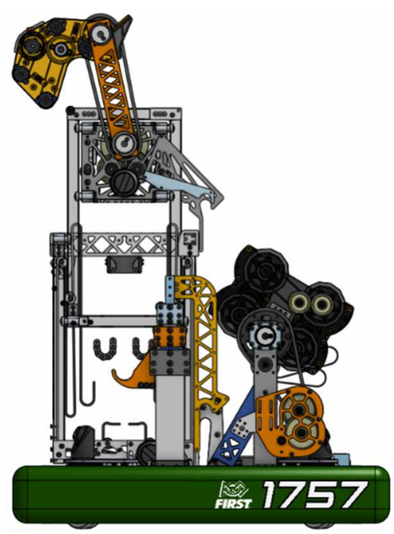
AMP SCORE



CLIMBING – STAGE 1



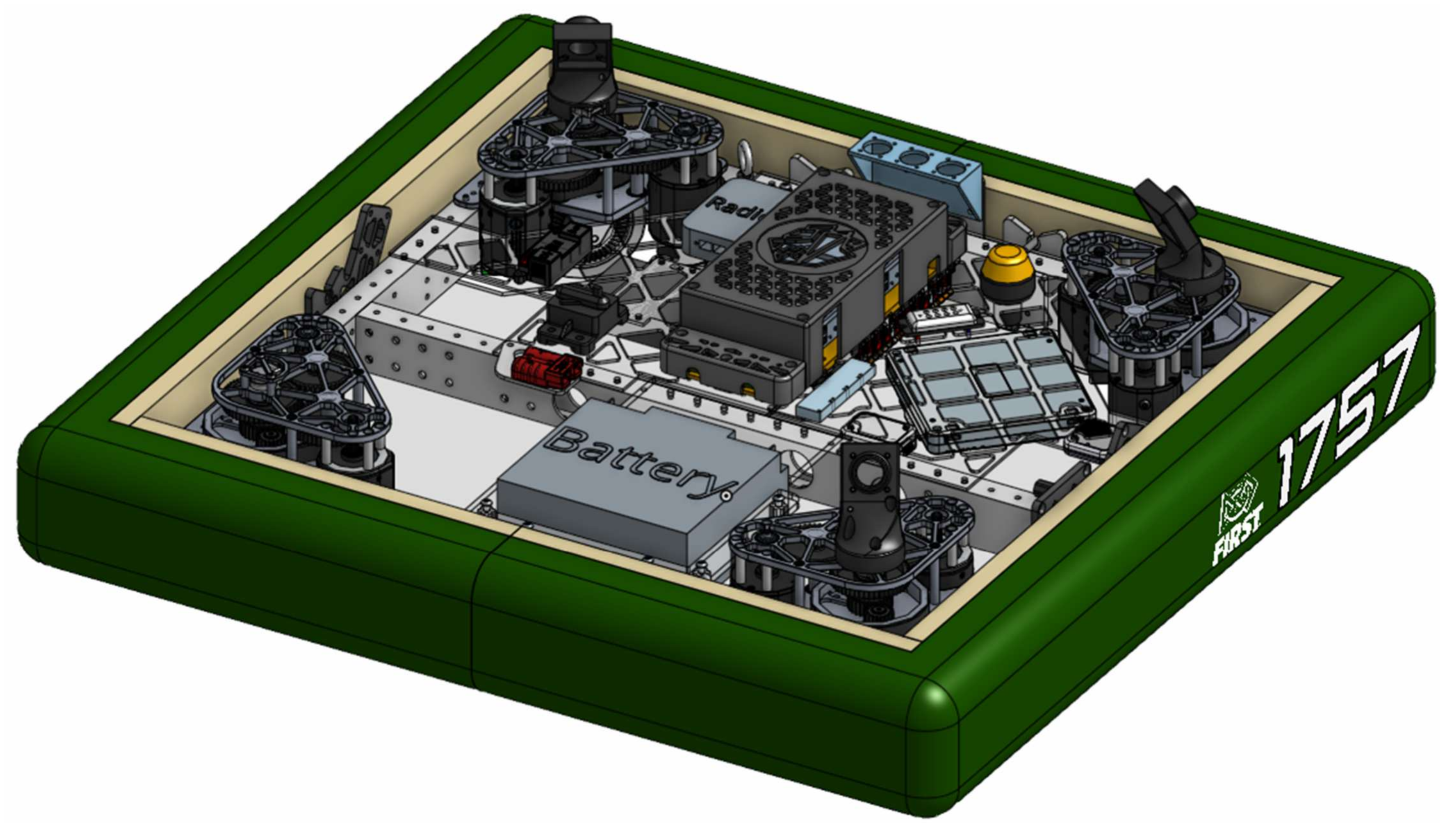
CLIMBING – STAGE 2



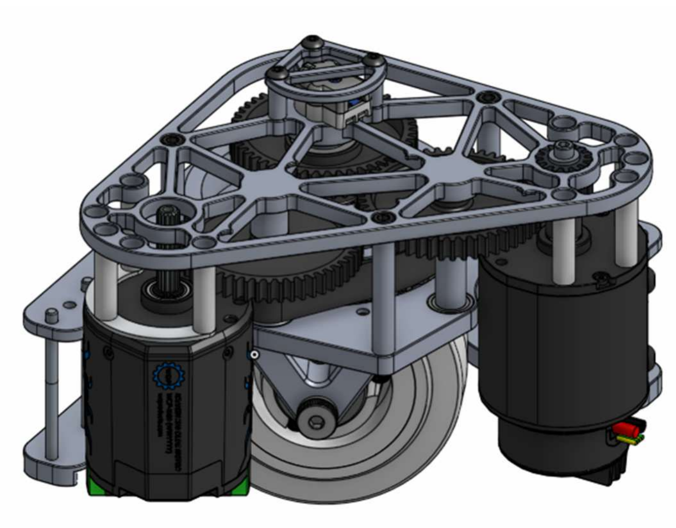
TRAP SCORE

ABOVE: Side View Images of the Robot Mechanism at Various Critical Points in its operation

MAJOR SYSTEM #1: DRIVE TRAIN

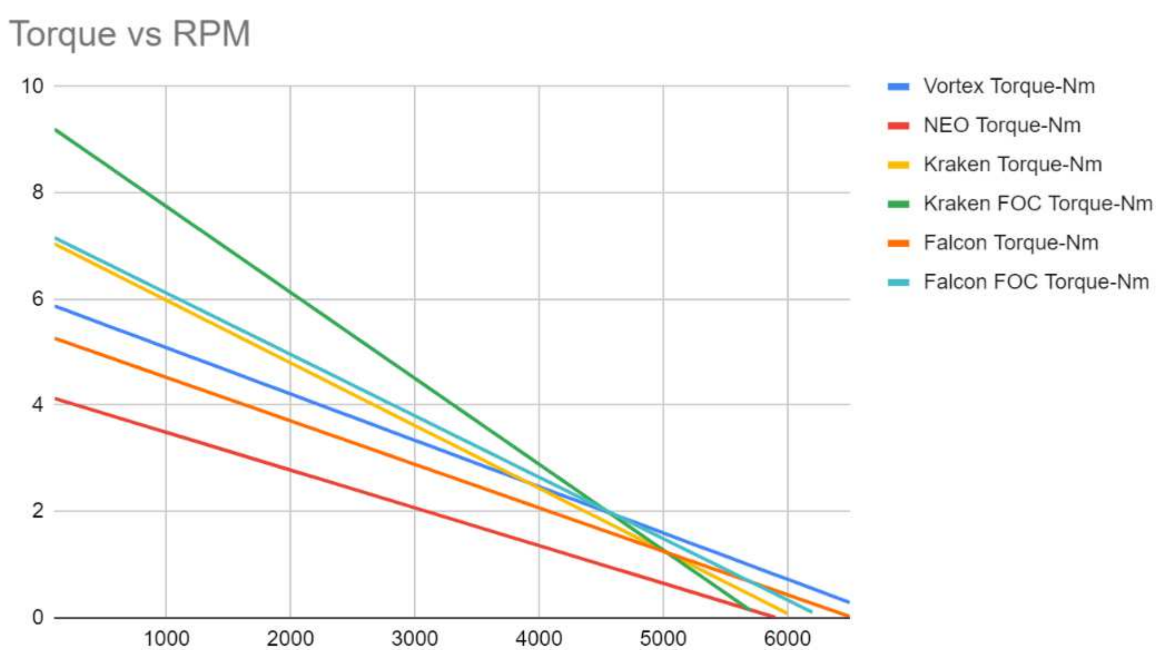


1.1 - SWERVE DRIVE MODULES



Proteus SDS MK4i Swerve Module Configuration	
Powerplant (Steering)	Falcon 500
Powerplant (Propulsion)	Kraken x60
Gearbox Configuration	L3
Overall Gearbox Ratio	6.12:1
Unadjusted Free Speed	17.1 ft/s
Drive Wheel	4" Coulson Caster

Proteus is our 3rd competition bot that has utilized a Swerve Drive for its drive train configuration. Each swerve drive modules utilize two motors, on ois used to power the propulsion (forward and back) of the Drive Wheel, and the second motor is used to adjust the rotation of the drive wheel. A Swerve drive system is holonomic meaning the robot can move in any direction relative to the field with changing the orientation of is chassis. New for this year we are taking advantage of the newly released Kraken X60 motors which are more powerful than the Falcon 500 motors we used on Skadi (2022) and Luxo (2023). We are still using Falcon 500 motors for the steering of the swerve module because we don't think the extra power provided by the kracken is necessary for that application. This year we have also upgrade the gearbox configuration from L2 to L3 increasing our overall free speed from 16.5 to 17.1 ft/s, we tried it on Luxo at some offseason events in the Fall and were extremely impressed. Finally, this year we moved away from the billet Aluminum wheels with replaceable nitrile treads in favor of Colson performance casters. The Colson wheels don't require nearly as much service and replacement, which is a significant quality of life improvement.



Above: Comparison of Torque vs RPM for Various Brushless motors commonly used in FRC Robots, as you can see the Kraken motors have far superior Torque performance at Low RPM, this means more power to get the robot moving faster (Graph: Courtesy of CTRE)

1.2 - ELECTRONICS SUBSYSTEM

Last year we moved to a completely underslung design for the electronics, this means that we have an electronics pan mounted on top of the drive rails and all the electronics are installed upside down underneath it. This allows for wide open access to the electronics systems when the robot is tipped over (very carefully) on its side. There is no need to remove robot components in order to access the electronics and the electronics are protected from any stray game pieces or getting snagged by another robot. This does leave the electronics exposed to the floor however and with our low ground clearance this could be a problem, to alleviate any potential disasters a 0.25" thick polycarbonate skid plate is mounted to the underside of the drive rails to protect the electronics bay. The Polycarbonate plate is easily removed by unscrewing a few 10/32 countersunk bolts.

Lesson Learned – Choose your Failure Mode!

The 0.25" Polycarbonate skid plate is secured by Countersunk 10-32 bolts threading into blind rivnuts that were installed into the bottom of the drive rail. The 10-32 bolts were often removed using a powered drill/driver for convivence/speed. Towards the end of the season we started to notice that the rivnut threads were starting to strip at an ever more alarming rate. The Aluminum rivnuts were much weaker than the Steel bolts and thus were causing the treads to strip after repeated installation and reinstallation. Striped out rivnuts are very time consuming to replace, unlike a striped bolt which is easy to replace. So this year we are installing Steel rivnuts in the drive rail and using aluminum screws to secure the skid plate. Now the screw is much more likely to fail before the rivnut.

Electronics System Major Components

- (1 ea) National Instruments - RoboRio 2
- (1 ea) REV Robotics - Power Distribution Hub
- (1 ea) CTRE Pigion 2 Accelerometer
- (1 ea) CTRE CANivore
- (1 ea) CTRE CANdel
- (2 ea) BrainBoxes – SW-015 5 Port Gigabit Switch
- (4 ea) AuduCam 120 FPS Global Shuter Webcams (For Computer Vision)
- (1 ea) Generic USB Camera (For Driver Vison)
- (4 ea) OrangePi OPi5 Compute Modules
- (4 ea) Kraken x60 Brushless Motors w/ Integrated Talon FX Motor Controller
- (8 ea) Falcon 500 Motors w/ Integrated Talon FX Motor Controller
- (3 ea) REV Robotics Flex brushless motor controller
- (3 ea) REV Robotics NEO Vortex thru Brushless DC Motor
- (1 ea) Open Mesh Access Point

1.4 - TESTING PORTS

Last year we added a convenient patch panel to the upper side of the robot to allow for quick access to essential data ports when we don't want to access the underslung electronics. This was an amazing quality of life improvement for the Drive team and the crew working on the bot in the pits. So great in fact that we didn't even build a new one we just ripped the old one right off our 2023 Robot Luxo and Installed it on Proteus

Patch Panel Slot 1 – USB Type A

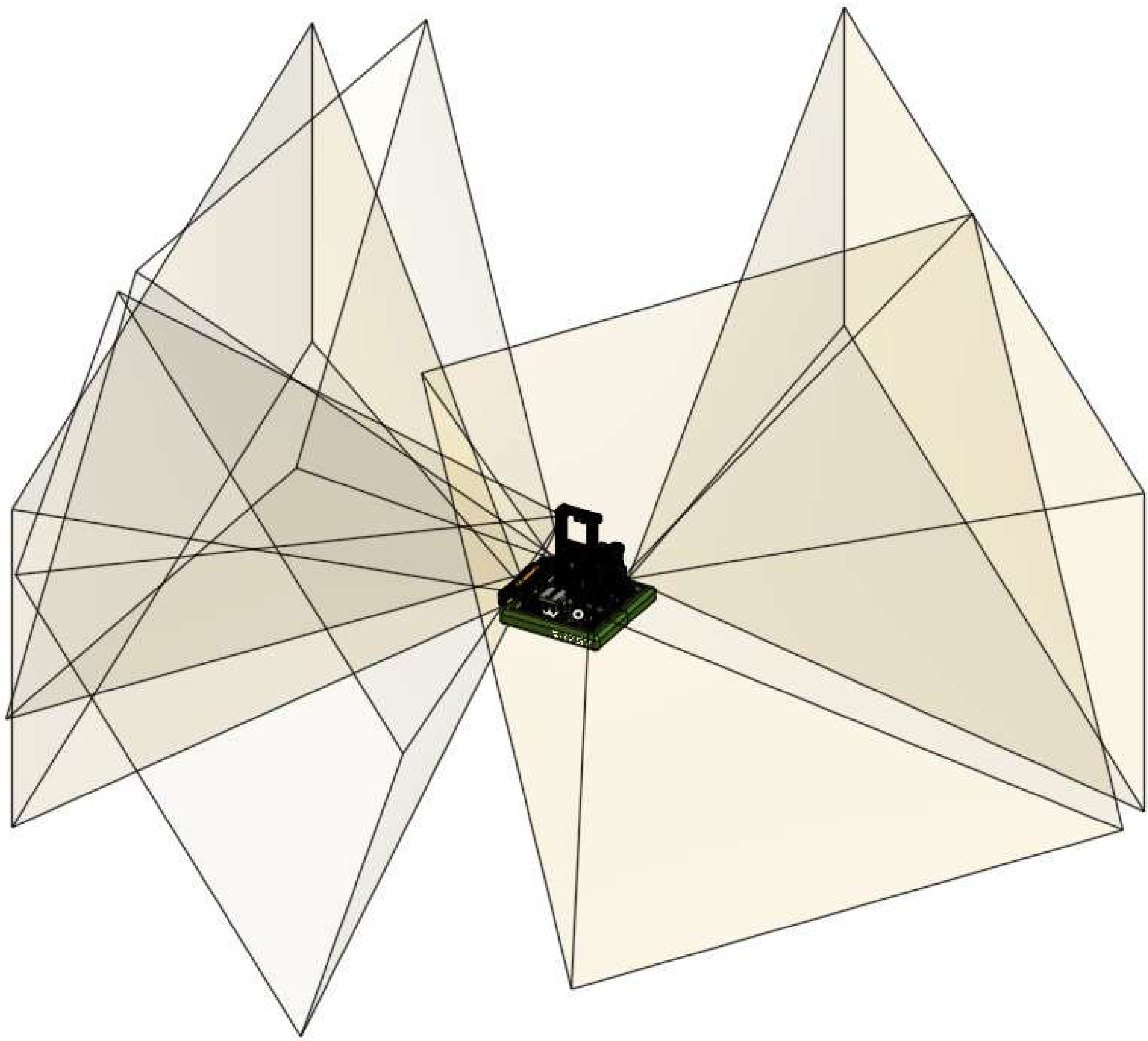
This slot connects to one of the USB Type A ports on the RoboRio. This typically has a USB flash drive plugged in. During a match all the system logs are copied to the USB drive. After a Match, the USB drive can be pulled and opened up in AdvantageScope on the debug machine for post-game analysis. It's our version of a Blackbox on an airplane.

Patch Panel Slot 2 – Push Button

By default, the majority of the motors on the robot are set to "Brake" mode, this means that when the robot is on the motors will resist rotation when not commanded to move, this means that if the robot disables halfway through its climb the robot would stay on the chain, they wouldn't back drive the motors and quickly drop back to the ground. However, this is very inconvenient when the robot is being loaded on and off the field and we have to adjust the robot to its starting configuration. This button allows the drive team to temporary disable brake mode on the motors and make what ever adjustments to the starting position that are needed

PATCH PANEL SLOT 3 – RJ45 CONNECTOR
This connects to the Ethernet Switch Via CAT5e for network access. Used for tethered connections to the bot during testing. Ethernet tethering is preferred, but we have encountered software reliability issues in the past.

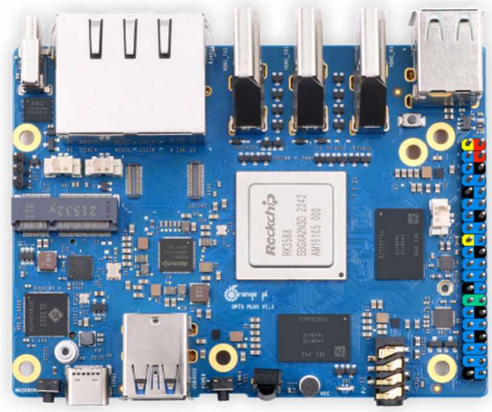
1.5 – CAMERA/VISON SYSTEMS



Above: Rendering of Camera FOVs

APRILTAGS AND VISON SYSTEM
This year we have gone all in on computer vision and robot pose estimation using Vision and AprilTags. In order to power this monumental software task we had to seriously upgrade the hardware we used to support it. In previous years we had made increasingly heavy use of Limelight vision cameras however we were very unsatisfied with their performance tracking the AprilTags which were introduced to the FRC field for the 2023 Game. Leaning heavily on developments we saw on other teams particularly 6328 Mechanical Advantage we transitioned this year to a new vision architecture. Instead of Limelight cameras we have installed ArduCam USB cameras on all four corners of the robot with each camera feeding into an OrangePi Compute Module for onboard vison processing and PoseEstimation. With this new system our robot knows its exact position and orientation on the field to a surprising degree of accuracy allowing us to automate a lot of processes such as driver alignment, and shooter trajectory with no input/ adjustment from the human operator.

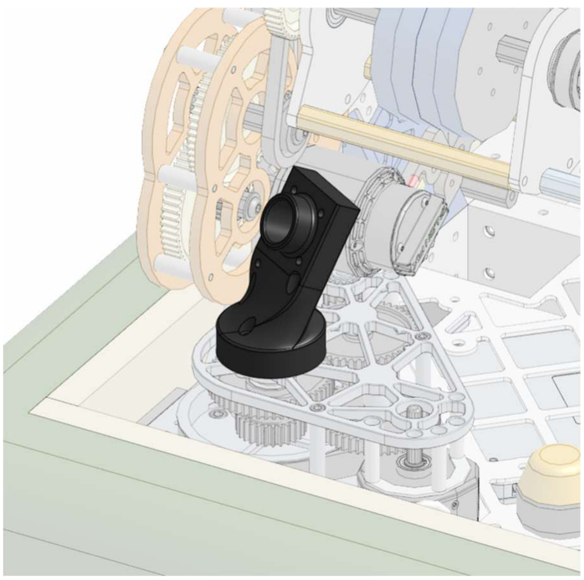
THE HARDWARE

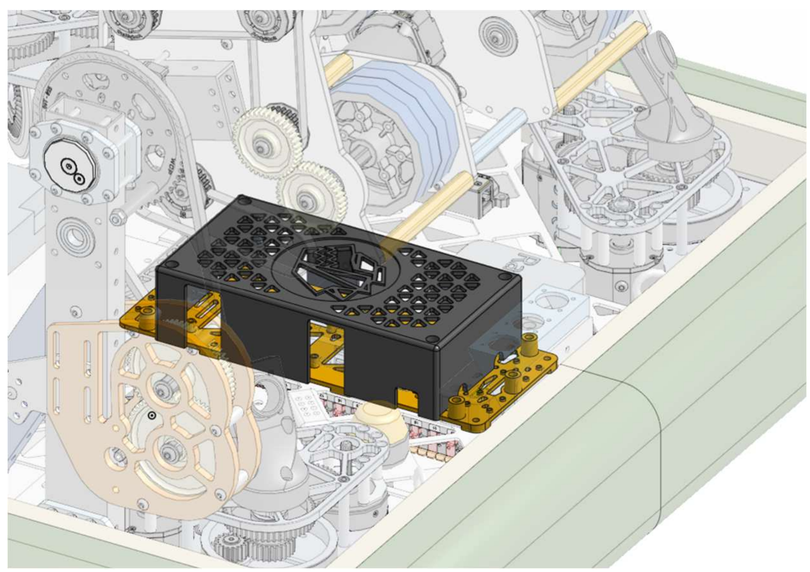


ARDUCAM USB
120fps Global Shutter USB Camera Board, 1MP OV9281 UVC Webcam with Low Distortion M12 Lens Without Microphones

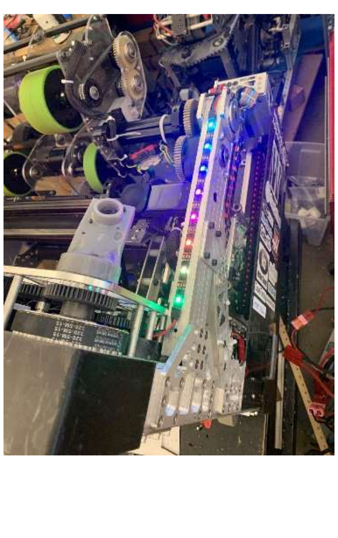
ORANGE PI 5 PLUS (32GB)
Rockchip RK3588 8-core 64-bit processor, 32GB LPDDR4X, M.2 slot supporting SSD and Wi-Fi 6/BT module

CUSTOM 3D PRINTED CAMERA MOUNTS AND COMPUTE MODULE ENCLOSURES





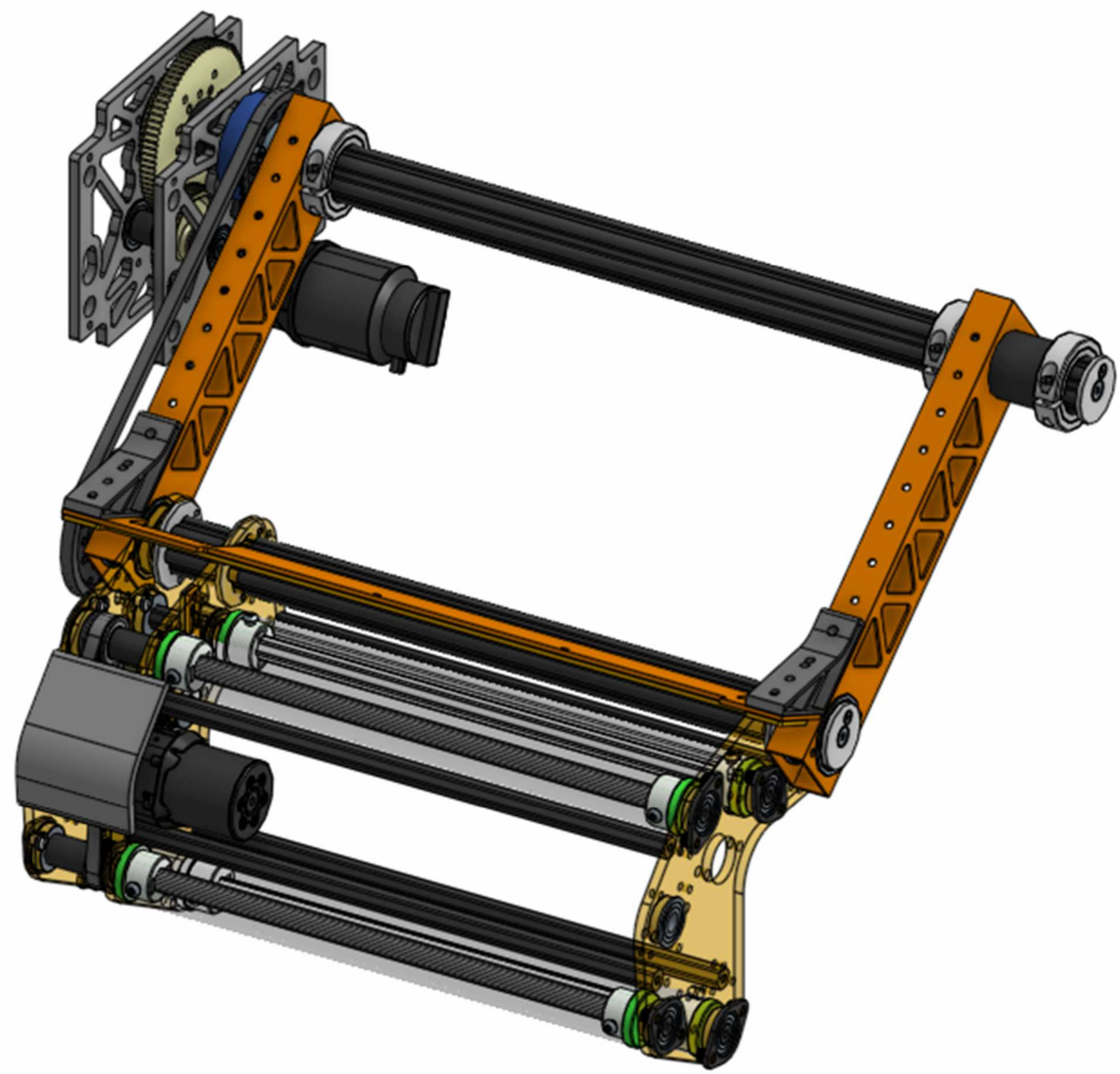
1.8 –ROBOT STATE INDICATORS



We Added LED RGBW Strips to the frame of our robot so we can easily signal to the driver what the robot thinks its current state is in for Example, if the driver tells the robot to align itself to the target the Lights Will Flash Yellow and remain Yellow until the Robots vison system says that the Shooter is locked on to its target, when the robot is locked on the lights turn green. This allows the Driver to use the Auto Aiming Features without taking his eyes off the robot itself. The second driver has a more detailed view of the state of various sensors on the bot and can call out any relevant info to the primary driver.

Just For Fun
Sometimes things are useful sometimes they are just for fun, we spend a few hours working on the code so that when the robot was on but not enabled the LEDS change to an animated Rainbow sequence. As soon as the match is over the robot displays Rainbow RGBs, does it serve any functional purpose…No, Ist it entertaining…Yes

MAJOR SYSTEM #2: INTAKE



Proteus's Intake is a true Multifunctional system. Not only does it pick the Notes up off the ground, it feeds them into the shooter, Scores them in the Amp and the Trap.

2.1- MOTORS & GEARBOXES

PIVOT MOTOR

The pivoting of the central Spline Shaft and the entire Virtual four-bar linkage is powered by a Falcon 500 Motor running though a custom gearbox contained with the inner most carriage of the sidevator

LESSON LEARNED – WHAT IS A VIRTUAL FOUR-BAR?

You may be asking what is a virtual-four bar, is it made up, is it the latest product from the metaverse. No a virtual four bar is just a fancy implementation of a basic mechanical principal Rev Robotics has provided a great explanation of the diffrences between Four-Bar and Virtual 4 Bar Mechansism at the following link https://docs.revrobotics.com/ftc-kickoff-concepts/powerplay-2022-2023/lifts

INTAKE MOTOR

The 4 intake rollers are all powered by a Neo Vortex thru Bore Motor. The Vortexes are new for this year and while this particular application did not necessarily require a thru bore motor it offered the most speed and power needed in the most compact form factor of any of the common FRC brushless motors on the market. The 4 intake rollers are connected to the drive motor though a series of HTD belts, pulleys and Gears

2.2 - ROLLERS

WCP ROLLER

Last year we were very pleased with the neoprene rollers and Vex VersaRoller hubs we used on the Intake for our Robot Luxo. They were however extremely difficult to assemble and service so this year we went with the much thinner Silicone rollers and Roller Hubs from WCP. These instead of relying on a 0.25” of compressible neoprene material for compression and grip the stickier silicone is only 0.125” in thickness

If we have a problem with the silicone losing its grip overtime we have plans to replace the silicone wrap on the rollers with cats tongue grip tape which we have seen many other first teams have repeated success with in the past two years.

2.3 - SENSORS

There are 2 lidar presence sensor switches built into the Intake, these sensors allow us to know where the note is as it makes its way through the intake. Because of our intended levels of automation on our robots we take a lot of time discussing and provisioning sensor requirements as part of the earlier deign phases, its always easier to remove a sensor you don’t actually need from the finished design than it is to cram one int there at the last minute when you realize you need it. This is our second year using these Lidar presence sensors from Polluo and we are quite satisfied with their performance even if they are a bit finicky to wire at first.

LESSON LEARNED – SIMPLIFY YOUR WIRES!

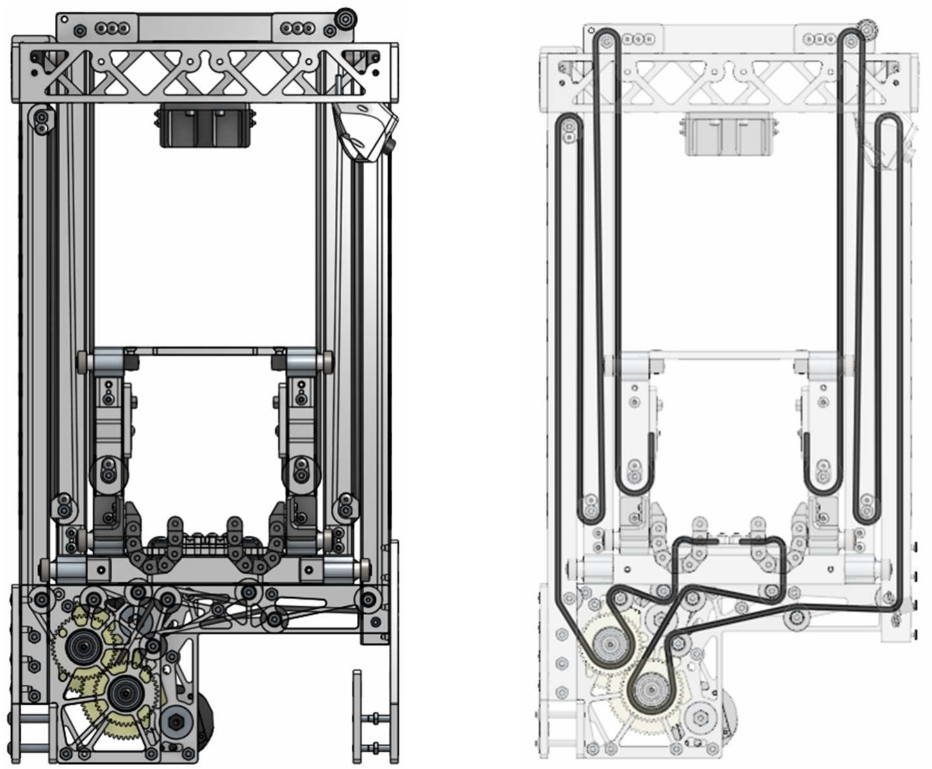
One of the biggest Achilles heels of any FRC robot is its wiring harness, wires need to carry signals from sensors and motors from where they are all the way back to the Brain of the robot, this involves running around in and through all the different sharp joints and pinch points of a robot, the more wires you have to run the more wires you have the potential to break so we like to reduce the amount of wires and how many moving joints they need to move through as much as possible. For example the line presense sensors for the intake: in order to wire these sensors to the rio the wire would need to run though 2 rotating joints and a sliding elevator.

But there is another option most of the Motor controllers that are used on the common FRC Motors (Neos, Falcons, krakens) have provisioning for limit switches to be wired directly to the motor controller, it usually takes some research on how each individual data port works but they usually appear as individualy addressed atribututes on the can bus. We make use of this to pass their sensor dat to the rio without

2.3 - THE CANTILEVER

The vertical up and down travel of the intake is controlled by the sidevator, this means that the entire intake acts as a large cantilevered arm off the side of the elevator when in motion. This required us to significantly reinforce the Spline Max Shaft that the whole intakes rotate on. While this is not ideal, it would have been too weight and space intensive to have a powered elevator on both sides of the intake. This is why all the gearing and motors are located on the inboard side of the intake relative to the side

MAJOR SYSTEM # 3: SIDE-VATOR



Left: side view of sidevator

Right: side view with timing belt paths isolated for illustration

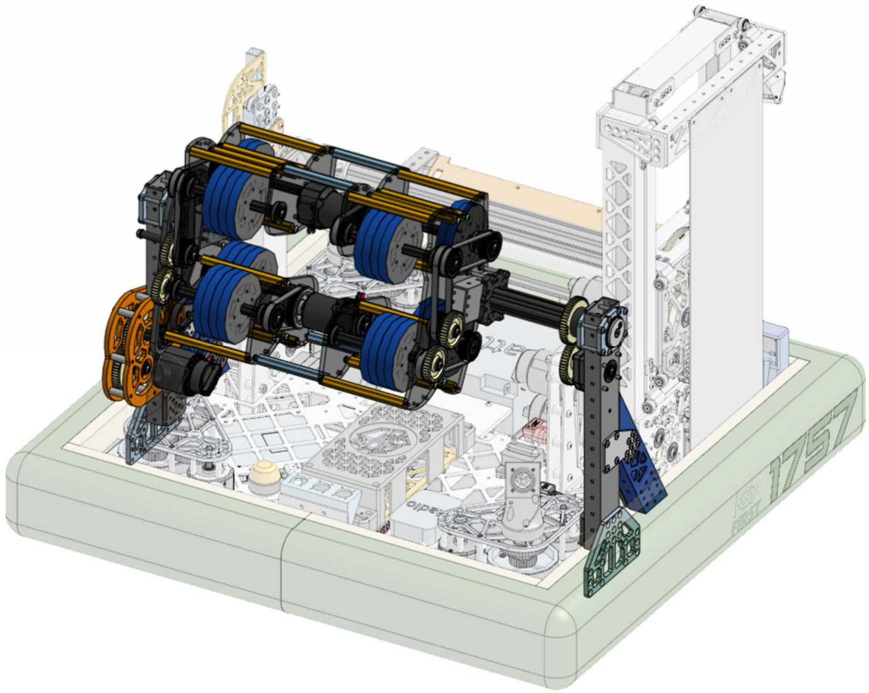
3.1 CONTINUOUS ELVATOR

We

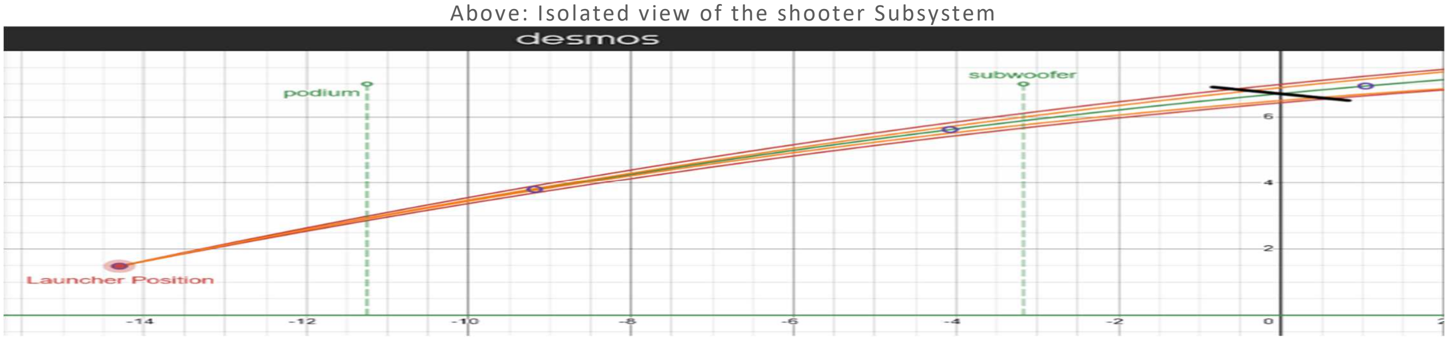
3.2 GEARBOX

Text

MAJOR SYSTEM # 4: SHOOTER



Above: Isolated view of the shooter Subsystem



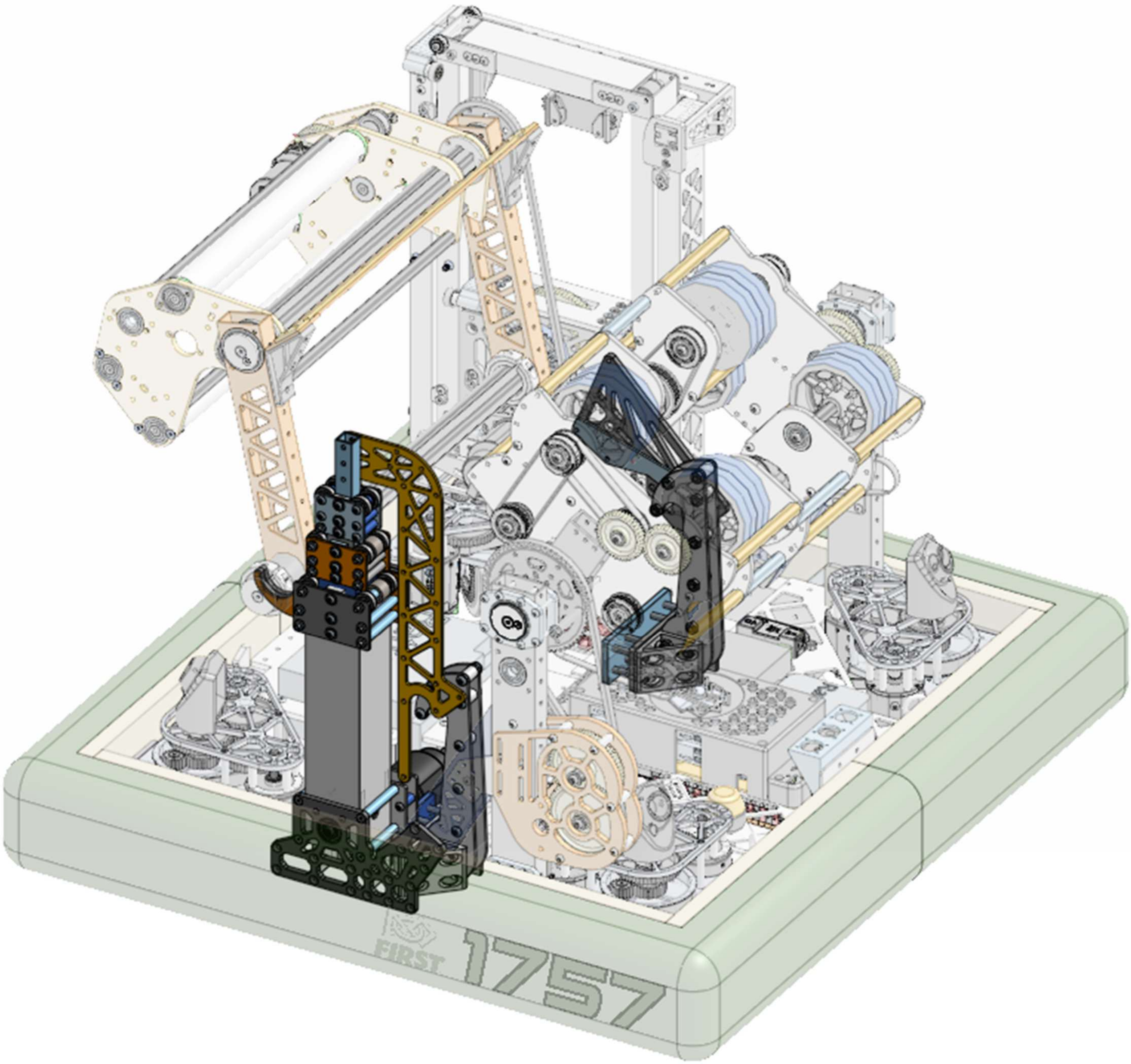
4.1 – ADUSTABLE ANGLE

We created a mathematical model of all of our potential shots to define the parabolic trajectory of the note coming out of the shooter. This model allowed us to not only make preliminary assumptions about flywheel speed and software tuning, it clearly showed us the range of motion our shooter need to be able to move through. It is aproximatly 55° starting parallel to the ground and rotating upwards. This will allow us to make shots anywhere from right up against the speaker to all the way out at the midline.

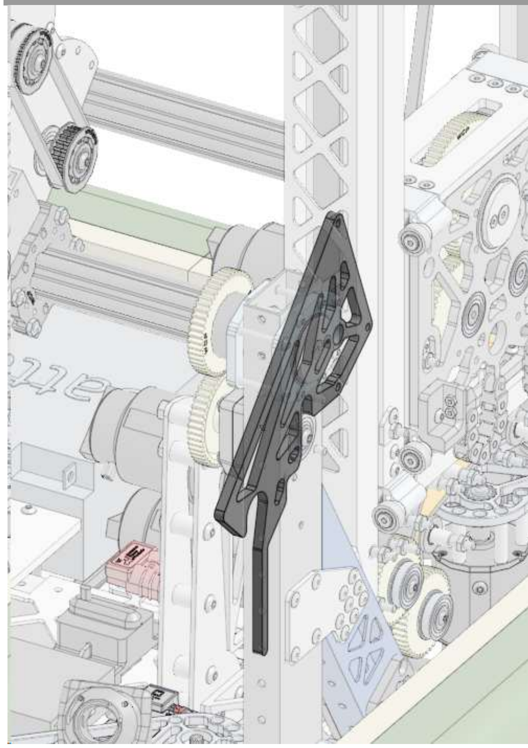
4.2 – INDEPENDENT FLYWHEELS

Obviously making shots from halfway across the field is a very ambitious goal, but we have made some very successful flywheel shooters in the past few years the big difference between those shooters and this one is that all of our previous is that they were shooting spherical game pieces while this year's game piece was toroidal. We knew we needed to develop a prototype very quicky to answer 2 very important questions: First, can a flywheel shooter launch a toroidal game piece with a stable flight path, and second what type of spin has a better impact of flight stability is in above and bellow like with a sphere or is left to right more beneficial to flight stability. No one had a ready answer to this on the internet so we knew we would have to experimentally validate this ourselves. Luckily the answer to the first answer was a simple yes, but we still spent a whole afternoon evaluating the second question. In the end we determine that despite the different shape, top and bottom spin differential in the flywheels were still the most effective at stabilizing the flight path of the note. For the final robot we start with the estimated speeds and launch angles based on our mathematical model and then experimentally verify those values to create an adjusted trajectory calculation. All the trajectory and flywheel speed adjustments are done by the robot in code, the Driver only commands the robot to Aim and it takes care of the rest. There is always a chance that something could go wrong, so the second driver has the ability to apply a global "fudge" factor to the trajectory calculation in the middle of a match if they notice shots are off in a consistent manner.

MAJOR SYSTEM # 5: CLIMBER

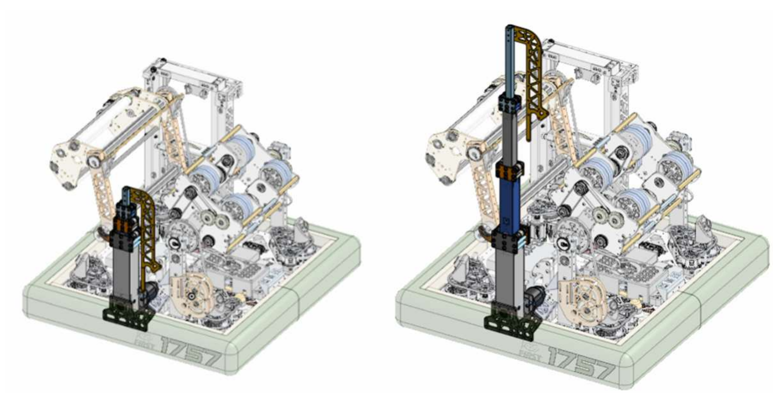


5.1 -



5.2 – ACTIVE HOOK – CLIMBER

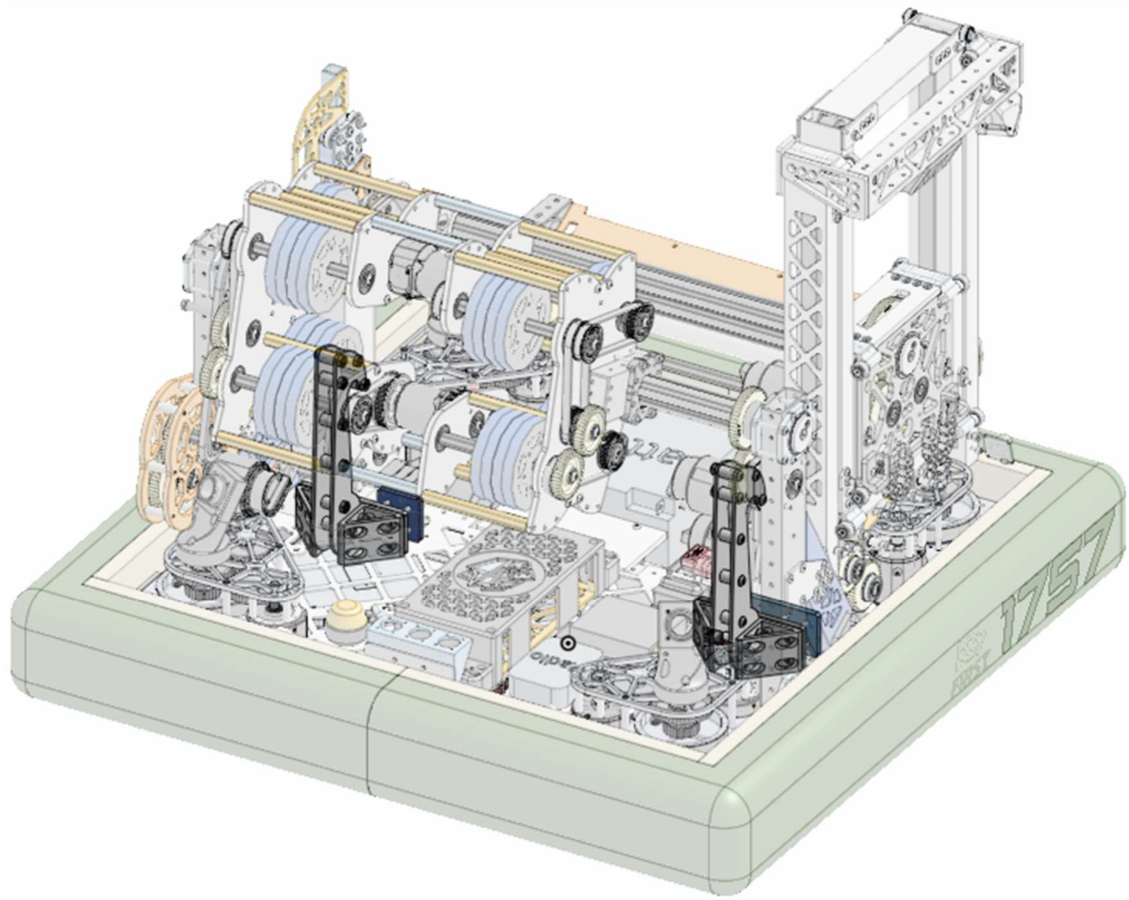
We Knew that we needed to climb with two points, but given our packaging factors and the way the intake worked we could not put a second sidevator on the opposite side of the intake it would simply be too much weight on the bot, but most the climbing power already comes from the powerful and rigid sidevator system. So we added this second climber on the other side. It is essentially an Andymark Climber-in-a-box at its core, however we realized that we needed more stages than the Andymark unit comes with, so after a carful design study of how the Andymark design works we managed to reverse engineer an additional outermost stage to incorporate with our climber.



Left: Isolated view of the Climber - Retracted
Right: Isolated view of the Climber - Extended

The climber uses constant force springs which are always trying to extend the telescoping sections, this is restrained by a loop of cable anchored to the uppermost stage and a winch housed inside the outermost stage. As the winch spools out the cable the telescoping sections extend, as the winch is pulled in the telescoping sections collapse into each other. This is the second Robot we have used this system on having previously used a similar system on our 2022 robot Skadi.

5.3 - PASSIVE HOOKS



Above: Isolated View of the Passive Hooks

These spring-loaded passive hooks latch onto the chain so that we can raise the sidevator again to score a note in the trap, They also ensure that the robot is positively engaged on the Chain at the end of the match. Since the Climb score is judged 5 seconds afte the end of the match there is the possibility given the weight of the robot that the motors could backdrive and slowly lower the robot to the ground post match before the climb is scored, as long as they engage the hook these hooks ensure that that wont happen.

The passive hooks are mounted directly to the center drive rail of the robot ensuring they are securely anchored to one of the stronges parts of the robot and their proximity to the robots center of gravity reduces the chances of a bending moment being formed relative to the bots rotation in space.

SOFTWARE

SOFTWARE: OUR DEVELOPMENT ENVIRONMENT

WPILib

The perineal stalwart, we still rely on core elements of WPIlib for robot communications and debugging. WPILib's new Logging features have greatly enhanced our Debugging capabilities

RobotPy

We have found that students have a lot easier time learning python then they do Java or C++ so with the growing support for RobotPy we migrated our Codebase from Java to Python in 2020.

GitHub

Without Github our level of remote work and collaboration just wouldn't be possible.

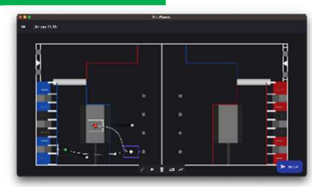
AdvantageScope

WE LOVE ADVANTAGE SCOPE! Not only does it log *everything* but does it in away that is intuitive and easy to review. No more searching though 10000 lines of log files to find the one piece of information we need. Huge thanks to team 6328 for building such a great tool.

PhotonVision

We are using PhotoVision as our native development framework for Computer vision due to its growing wide support inside the FRC community. It does not include native support for RobotPy however so as an offseason project our lead programmer wrote a custom wrapper for PhotonVivion so it can work inside our RobotPy environment

PathPlanner



This is our Second full season using PathPlanner as part of our autonomous software stack. The simple to use graphical interface allows us to quick generate new autonomous sequences and deploy them in a matter of minutes

LESSON LEARNED – ENSURING CODING SUCCESS!

One of our code policies is to have at least two other students of the team review the code before being merged into main. This has two main pros/cons

- **Pro:** multiple students on the team know whats going on with code, not a single source of failure
- **Con:** if one student has a bunch of back to back, individual complete changes, they have to wait for other students to completely understand to use on the bot

The con part has hurt us especially nearing competitions last year, where we just had one student who made a lot of last-minute integration changes to fix the bot, but we had a “magic branch” which defeated the purpose of main. More students this year have stepped up, and we hope that by just finishing faster these on-the-fly changes are mitigated

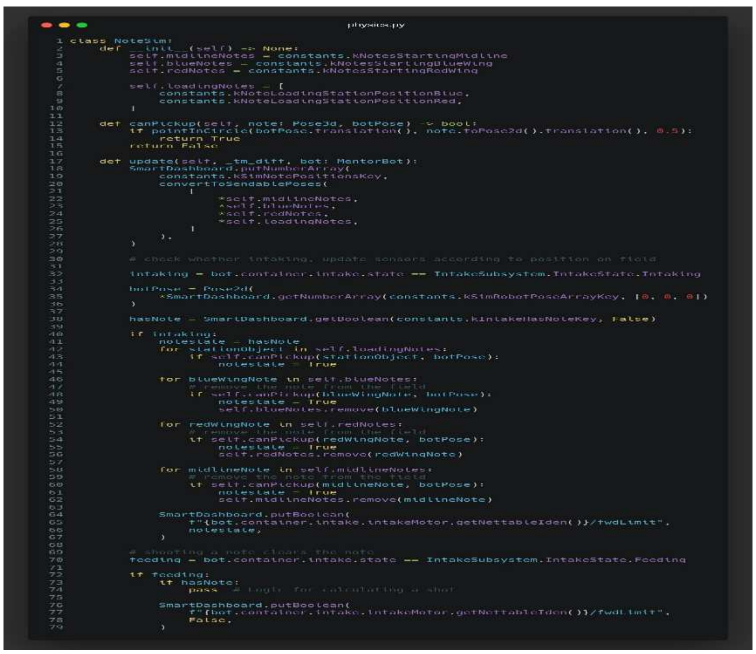
SOFTWARE: DRIVE

Using our same custom swerve base for this year, the drive subsystem is pretty box-standard. A few of our major optimizations include utilizing second degree swerve kinematics to increase efficiency as well as compensating for a rotating and moving robot Intake:

A Brief Tangent

Two years ago our lead programmer had a new idea for drive control, an absolute relative drive. The common swerve drive control method was to have a field relative translation for the bot, and a robot relative rotation. What this meant is a left input on the rotation axis would result in the robot rotating to the left at a constant speed. A translation action was not affected by rotation but instead was in "field relative" space. The difference of absolute drive is that the rotation is also field relative. A left input on the rotation stick will yield the robot turning to face left. This year we expect this type of robot control to be very important for drivers when they have to be able to turn to specific positions for collection and scoring on swerve drives. You can see this in action in any one of our videos from last year. Having fixed controlled rotation will allow for precise driver input and less fiddling with controls when cycle time is very important. The drivers have also experimented with alternate driving methods on swerve to get used to interesting control schemes such as a curvature drive, standard tank drive, standard field relative drive, and full robot relative drive.

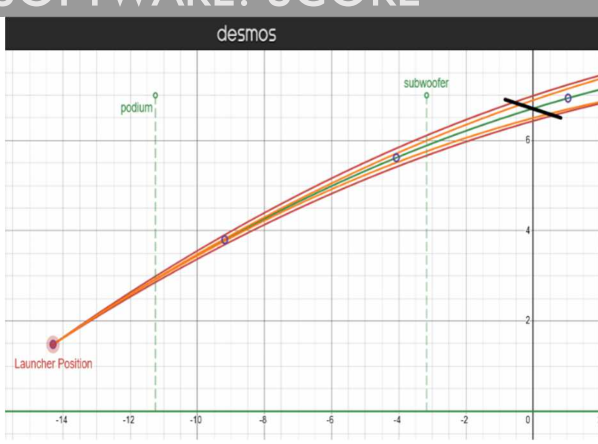
SOFTWARE: INTAKE



```
class NoteSim:
    def __init__(self) -> None:
        self.midlineNotes = constants.kNotesStartingMidline
        self.blueNotes = constants.kNotesStartingBlueWing
        self.redNotes = constants.kNotesStartingRedWing

        self.loadingNotes = [
            constants.kNoteLoadingStationPositionBlue,
            constants.kNoteLoadingStationPositionRed,
        ]

    def canPickup(self, note: Pose3d, botPose) -> bool:
        if pointInCircle(botPose.translation(), note.toPose2d().translation(), 0.5):
            return True
        return False

    def update(self, _tm_diff, bot: MentorBot):
        SmartDashboard.putNumberArray(
            constants.kSimNotePositionsKey,
            convertToSendablePoses(
                [
                    *self.midlineNotes,
                    *self.blueNotes,
                    *self.redNotes,
                    *self.loadingNotes,
                ]
            ),
        )

        # check whether intaking, update sensors according to position on field

        intaking = bot.container.intake.state == IntakeSubsystem.IntakeState.Intaking

        botPose = Pose2d(
            *SmartDashboard.getNumberArray(constants.kSimRobotPoseArrayKey, [0, 0, 0])
        )

        hasNote = SmartDashboard.getBoolean(constants.kIntakeHasNoteKey, False)

        if intaking:
            notestate = hasNote
            for stationObject in self.loadingNotes:
                if self.canPickup(stationObject, botPose):
                    notestate = True

            for blueWingNote in self.blueNotes:
                # remove the note from the field
                if self.canPickup(blueWingNote, botPose):
                    notestate = True
                    self.blueNotes.remove(blueWingNote)

            for redWingNote in self.redNotes:
                # remove the note from the field
                if self.canPickup(redWingNote, botPose):
                    notestate = True
                    self.redNotes.remove(redWingNote)

            for midlineNote in self.midlineNotes:
                # remove the note from the field
                if self.canPickup(midlineNote, botPose):
                    notestate = True
                    self.midlineNotes.remove(midlineNote)

            SmartDashboard.putBoolean(
                f"{bot.container.intake.intakeMotor.getNettableIden()}/fwdLimit",
                notestate,
            )

        # shooting a note clears the note
        feeding = bot.container.intake.state == IntakeSubsystem.IntakeState.Feeding

        if feeding:
            if hasNote:
                pass  # logic for calculating a shot

            SmartDashboard.putBoolean(
                f"{bot.container.intake.intakeMotor.getNettableIden()}/fwdLimit",
                False,
            )
```

Using our full field positioning The intake is a complicated mechanism, and the most varied in state machines. It has two requirements from a software side: quick intaking and positional awareness. The first of which is completed mechanically, and executed with a velocity control on the motor itself, the second of which with positional awareness involves utilizing the two presence sensors on the intake. We calculated that, at the speeds we want to run the motor, the note has a variation of 6in due to just the timing of the roboRIO, to compensate for this there is a secondary positioning sequencing that slowly moves the note until a presence sensor doesn't detect, at which point the location of the note is known and it can move to a safe position. This is also used in amp and trap scoring to hold a known position.

SOFTWARE: SCORE



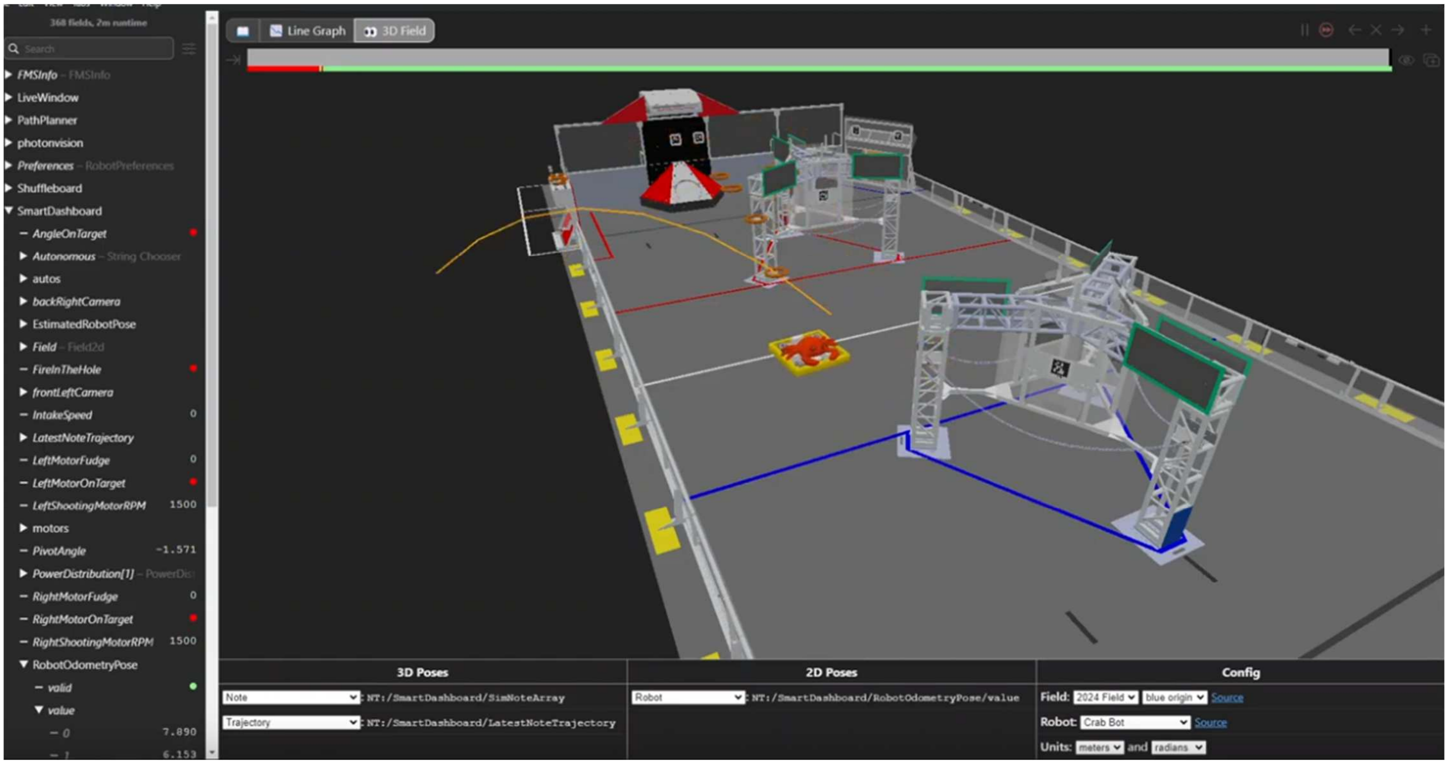
we calculate an automatic velocity-compensated shot. An iterative solver compensates for velocity of the robot, giving the static-shot target for a moving robot with a calculated time of flight.

We first used an graphing program to create a theoritcal model of the shot to develop our software algorithm then after testing with the actual shooter prototypes A tested angle compensation is given due to the note having some angle variation from the angle of the shooter on exiting. An adjustable constant spin is put onto the note based on the difference in each side of the note. Motion magic is used to give a controlled and time-optimized movement of the attitude adjustment

SOFTWARE: AUTONOMUS

We are utilizing pathplanner to quickly build autonomous pathing sequencing. Because of the ease of control for each subsystem, having target states and knowing whether each subsystem has completed its task, waiting for actions to finish before continuing on paths is simple to do.

SOFTWARE: SIMULATION



We historically get our bot done criminally late in the game. simulation and code helps us stay ahead and ensure that the time between robot being mechanically and electrically done, and when the bot can be competition-viable is reduced as much as possible. Each motor is individually simulated from CTRE and incorporated to give a 3 dimensional representation of the bot's pieces from what their motor encoder readings give. This allows the majority of the systems to be tuned within simulation because of accurate modeling of cad-measured inertia values Vision is also simulated based on the field of view of each camera, and variation is added to ensure the robustness of the vision system.

SOFTWARE: VISION

4 OrangePi 5+s run PhotonVision, and give us a 3 dimensional representation of the tags on the field. Using a kalman filter each of these pose estimates is fed back into a global positioning system. This is used to both help with auto aim shooting and vision-assisted autonomous. A note camera also detects notes on the field and returns a 3 dimensional position of the note relative to the robot, allowing automatic intaking

ENGINEERING TEAM

Technical Captain

Luke Maxwell*

Business Captain

Sean Tao*

Technical Vice-Captain

Landon Bayer

Business Vice-Captain

Claire Peng

Programing Lead	*Design Lead*	*Open Alliance Lead(s)*	*Outreach Lead*	*Media Lead*	*Finance Lead*
Ivan Cal	**Jacob Kaplan**	**Liam McWeeney* & Baili Jiang***	**Nina Pappas**	**Rachel Qu**	**Divya Siddi**

STUDENT MEMBERS

Henry Marsland	Lauren Buza	Elias Mukdissi*	Julian Saroufim
Kevin Bai	Yuhan Chen	Elizabeth Lowney	Anthony Yang
Anya Jiang	David Confoey	An Pham	Isabella Cai
Adrian Wong	Marshall Atkins	Vincent Milinazzo	Ethan Lee
Duanmu Jiang	Alex Theofilou	Constantina Flevarakis	Erik Cerulli
Adrianna Cirillo	Owen Monahan	Ella Xu	Celina Yau

WHS Faculty Advisors

James Looney & Sean Geary°

Senior Mentors

Dwight Meglan **Chris Aloisio°** **Steve Harrington°**

Mentors

Anthony Gelsomini **Manny Barros°** **Mark Holthouse** *Mentor Emeritus* **Charley Marsland °** **Amber Maxwell**

Eric Yamaguchi° **Sean Lendrum °**

Westwood Robotics, Inc
Board Of Directors

Steve Harrington° *President*

Catherine Tseng *Treasurer*

Manny Barros°

Cheten Gopal *Clerk*

Jack Tseng

**Denotes High School Seniors* *°Denotes FRC Alumni*

SPONSORS

Cardathea	Westwood High School	Gene Haas Foundation	Medtronic
Act Fasteners	Intuitive Foundation	IRA Motor Group	Teradyne
Deadham Savings	Needham Bank	Schneider Electric	Poirier
Comella’s		Print Master	
McWeeney Family	Peng Family	Gopal Family	Gelosimi Family

VENDORS

Andymark	West Coast Products	Vex Pro	Swerve Drive Specialties
Discord	Google Work	Onshape	Robo Promos
McMaster-Carr	The Home Depot	Cross the Road Electronics	MS Visual Studio
Python	GitHub	RobotPy	Java

1757

Special thanks to everyone who makes this team possible Including

Our Students
Our Mentors
Our Teachers
Our Parents

Our Amazing Sponsors

Westwood Public Schools
Timothy Piwowar - Superintendent
Allison Borchers - Asst. Superintendent
Cindy Cox - Building Use Coordinator
Tom Carey - Director of Facilities
Ron Smith - Asst Director of Facilites
Amy Davenport - Principal
Aishleen Marcus - Assistant Principal
Nicole Haberman - Assistant Principal
Tom Millett - Science Dept. Chair
WHS Janitorial/Facilities Staff
WHS Library Staff

#FRC1757
@WWRobotics1757