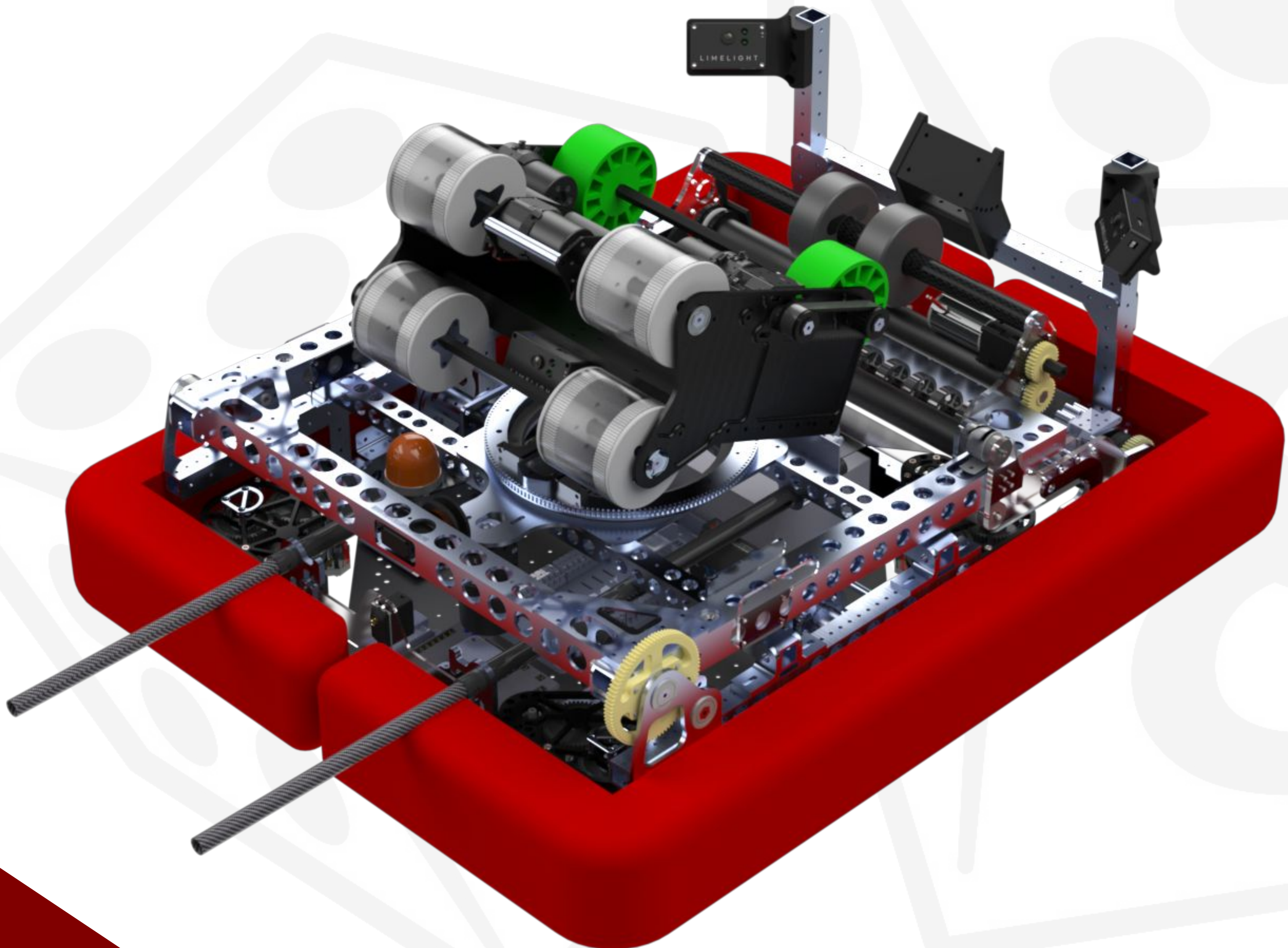
THE

High Rollers



2024 Technical Binder
“The Dealer’s Revenge”

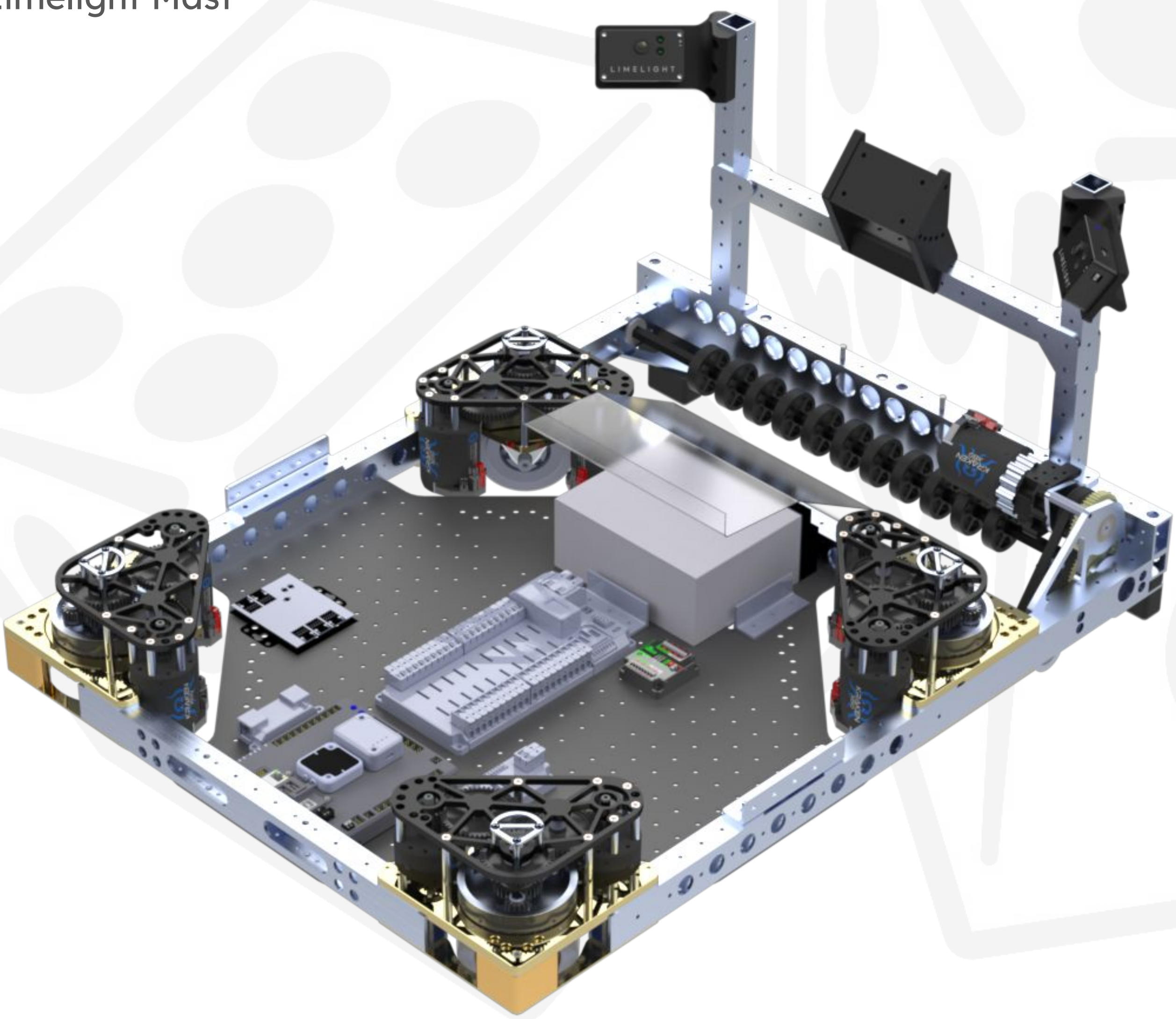
987

Table of Contents

	Page
Drivebase	**3-8**
Sectional Foreword	4
Frame	5
Swerve Module	6
Intake	7
Limelight Mast	8
Upper Structure	**9-14**
Sectional Foreword	10
Turret/Tilt	11
Shooter	12
Amp/Trap/Handoff Arm	13
Climber	14
Programming	**15-22**
Centralized Architecture	16
Turret Estimation	17
IDW Interpolation	18
Automatic Shot Decision-Making	19
Alternate Path Decision-Making	20
Sensor Usage	21
HRLogger	22

Drivebase

- Frame
- Swerve Modules
- Intake
- Limelight Mast



Recognizing the need for maneuverability and extra durability after prior events, we adopted a customized swerve drivebase. The drivebase also accommodates our under the bumper intake. It is also designed to accommodate an easy and quick bumper mounting solution.

Built around our custom swerve modules, the frame delivers extreme robustness and reliability while also providing an effective mounting solution for our under-the-bumper intake.

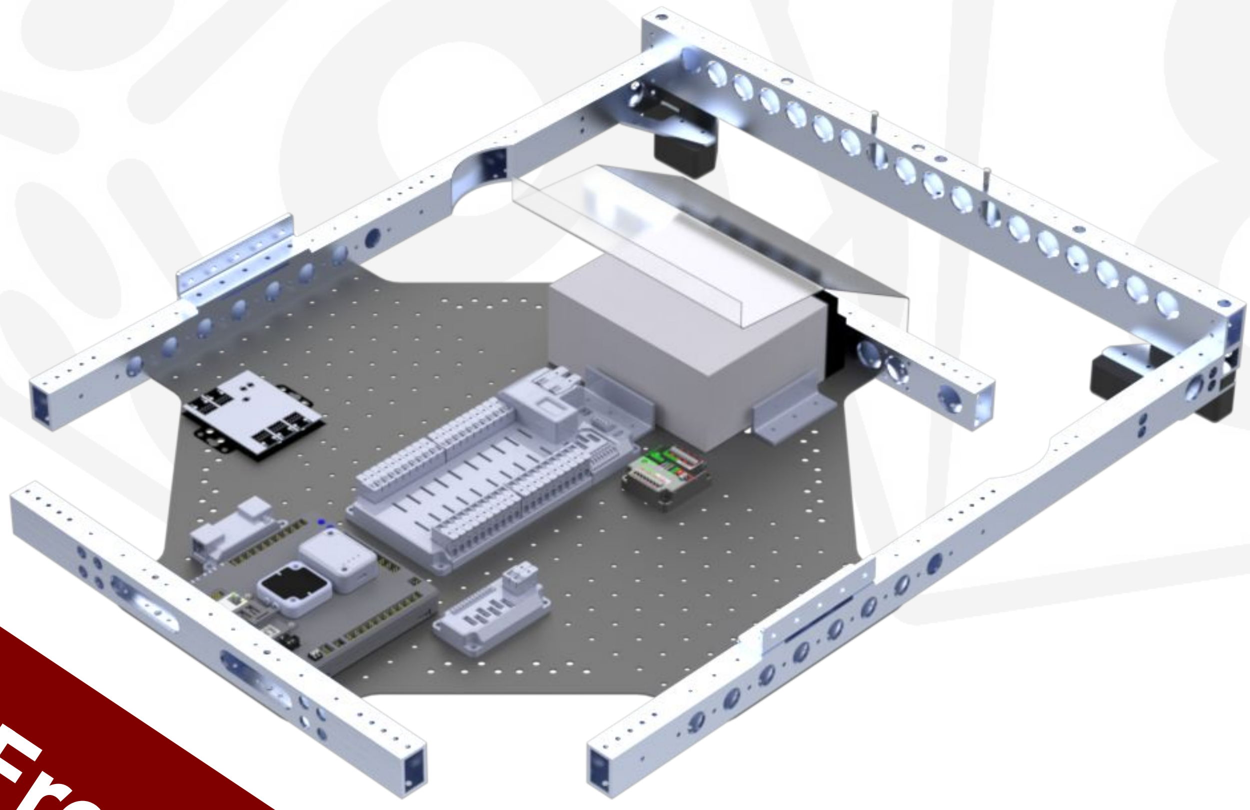
- Comprised of 1” x 1.5” x .125” Aluminum Tubes
- Bumper Supports strategically positioned off the side
- 32.5” x 26” frame dimensions
- Mounting point for UTB Intake

Front (Intaking Side)

- Aluminum 1”x 1” x .125” tube with 1” x 1.5” x .125” tube on top
- Custom 3D-printed intake funnels seamlessly integrate with the frame for efficient note collection and ensured protection for critical intake components

Bellypan

- Steel belly pan to keep robot COG as low as possible
- All robot electronics are mounted on the belly pan
- Front-biased battery to counter the weight of rear-mounted mechanisms



Frame

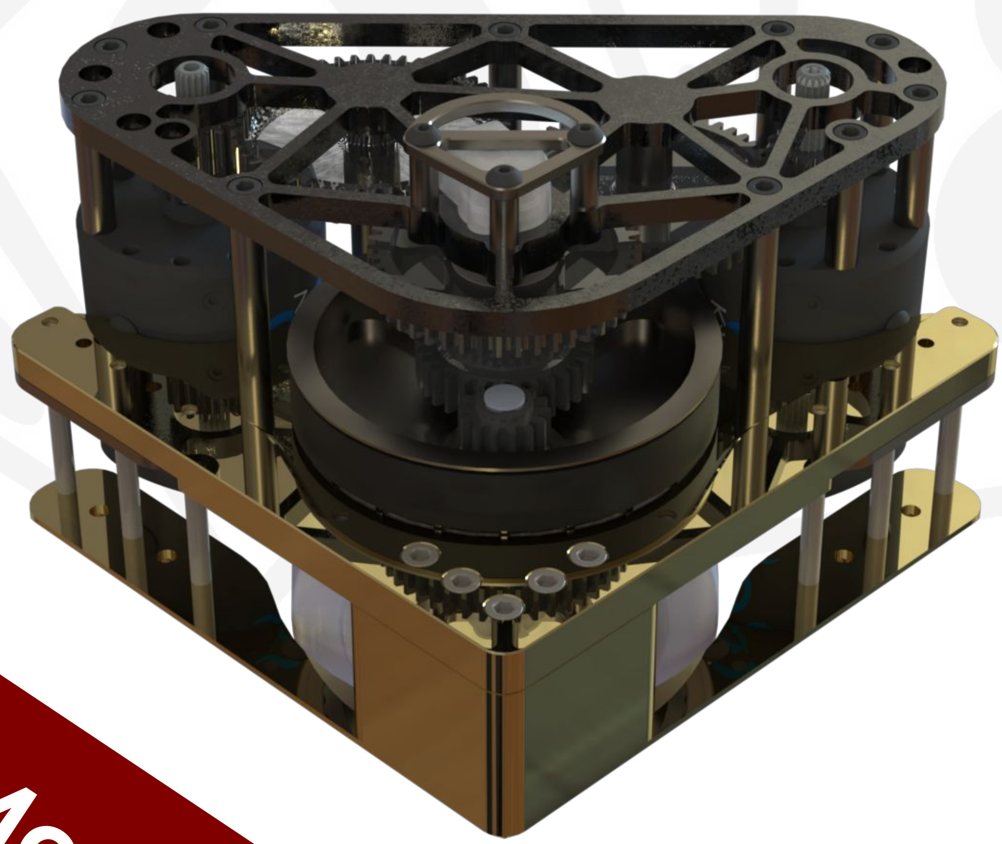
To optimize performance for the demanding speed and durability requirements of this year, we implemented a custom-designed module based on the SDS MK4i Module with minor profile changes to the main and bottom plate.

Plates

- Custom zinc-plated ASTM A36 steel main and bottom plate mitigates deformation observed from our 2023 Robot
- Anodized 7075 Aluminum Motor Plate

Gearing

- Custom double-stack gear (50T-30T)
- Interchangeable drive input pinions for selectable top speeds of 17.7ft/s (12-tooth motor pinion) or 23.6ft/s (16-tooth motor pinion)



Swerve Module

The main factor behind the decision to incorporate an under-the-bumper intake is to ensure that our intake mechanism is fully protected when interacting with other robots.

Pickup Specifications

- Intaking surface speed that exceeds robot max speed allows for on-the-fly pick up
- 1” counter-roller to “kick” notes off the ground efficiently dislodges notes from the tight spots, ensuring smooth intake
- Reliable 2” compliant wheels serve as the workhorse of the intake system, providing dependable note pickup

Design

- A custom 6061 aluminum motor spacer inspired from the WCP-1462 motor spacer allows for a 1:1 overall ratio using a Kraken X60 motor while also allowing the primarily driven hex shaft and the motor shaft to lie in the same axis of rotation
- Tapped mounting holes facilitate swift swapping of a spare and easy integration into the robot frame

Intake

To combat frequent pose loss during teleop caused by high-speed collisions leading to unreliable odometry data, we implemented a lightweight Limelight Mast. The two pose-updating Limelights significantly improves the accuracy and consistency of our robot's pose estimation throughout matchplay. The mast also provides a space to mount our pickup assisting Limelight (track notes).

Structure

- Made from 1” x 1” x .0625” VEX Versaframe Tubes
- Custom 3D-printed mounts designed to securely hold both the pickup and pose helper Limelights
    - The 3D-printed mount for the pickup assist Limelight allows for 10-degree incremental angle changes up to 40 degrees
- Pose helper Limelight mounts are strategically tilted upwards to ensure a clear field of view, eliminating potential obstructions from the robot’s subsystems.

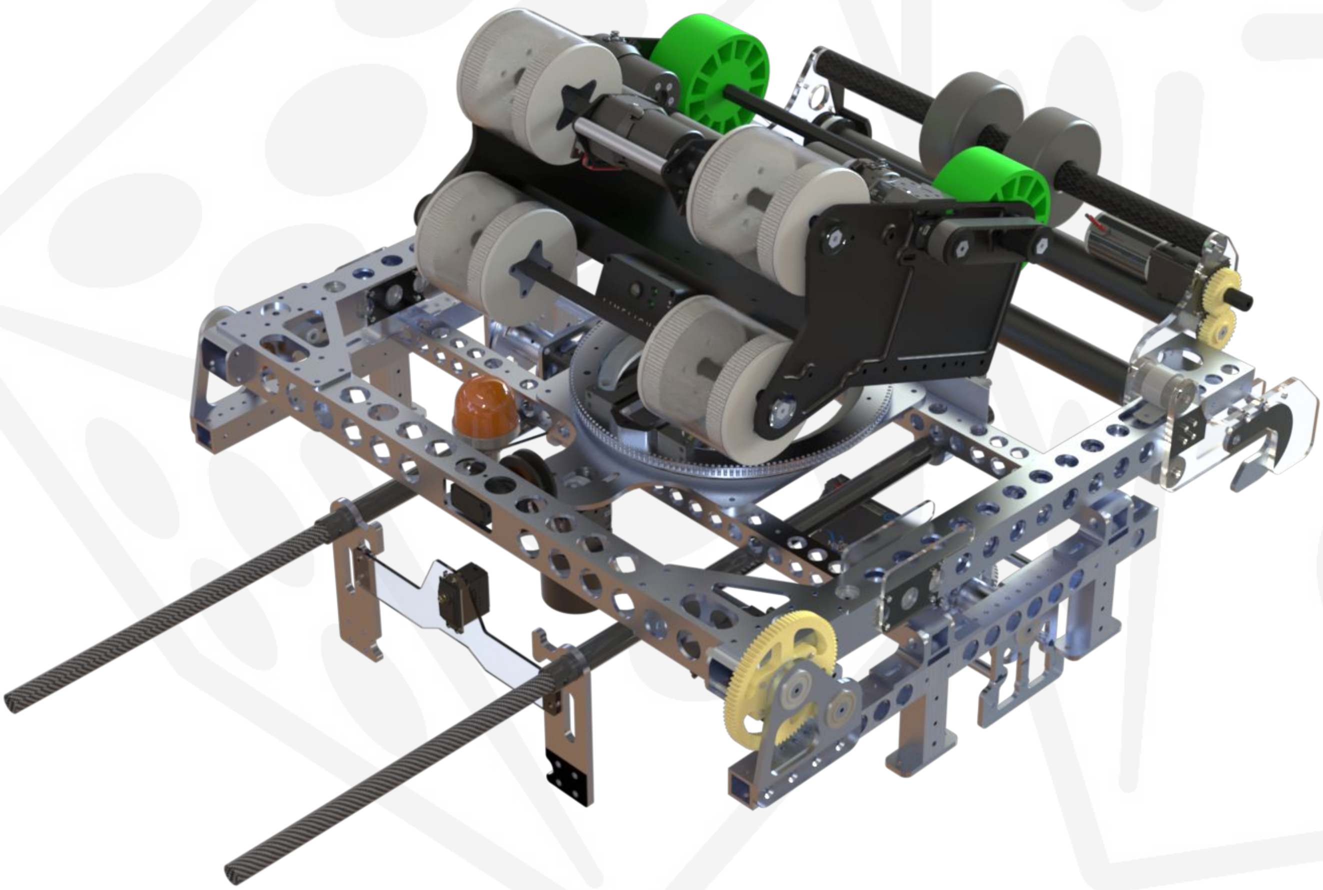
Cameras

- Two Limelight 3G cameras to combat pose loss caused by external interactions
- A separate Limelight 3 camera serves as the dedicated pickup assist (note tracking)

Limelight Mast

Upper Structure

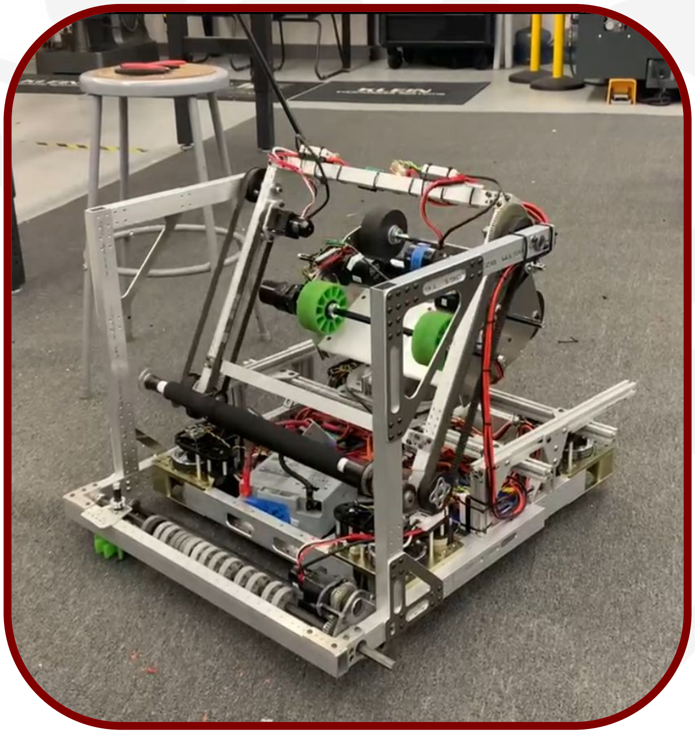
- Turret/Tilt
- Shooter
- Amp/Trap/Handoff Arm
- Climber

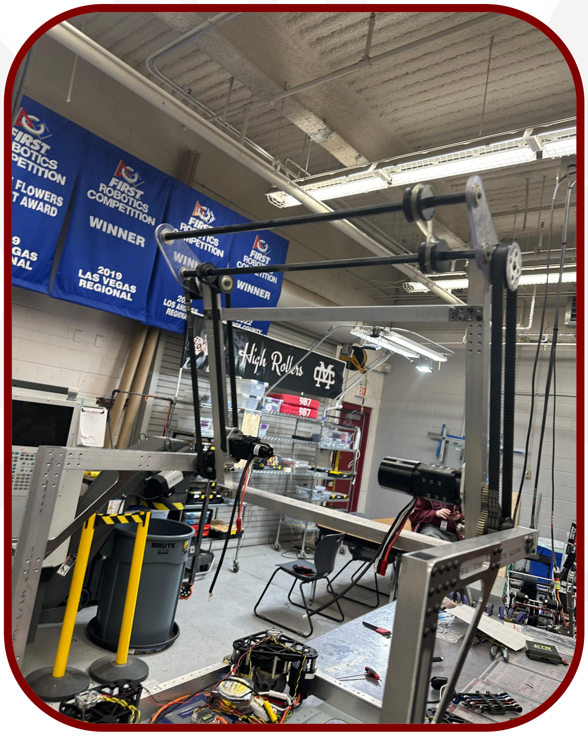


Aiming for versatility and full navigation under the stage, the upper structure is configured to achieve a maximum robot height of less than 26”. The installation also allows for easy maintenance of underlying electrical components by enabling the two crossbars, where the turret is mounted, to pivot for access. Additionally, it provides compact packaging between subsystems and allows for direct human loading from the source if needed.

During early build season, we identified key factors that will drive the design philosophy of the upper structure: **going under the stage and scoring the trap**. We then established the < 26” starting height constraint.

We came up with an iteration that allowed us to hand off notes from the intake whilst also scoring the amp, initially using a virtual 4 bar solution as shown below. However, unlike our final design, this iteration was too obstructive to allow a full 360-degree motion with the turret to which we concluded as extremely detrimental to our foreseeable robot performance.





Virtual 4-Bar Robot
Iteration

We decided to implement a turret this year in order to have more efficient amplification time usage and higher autonomous scoring potential. Our unique tilt system is very compact and packages well to the size of the turret bearing.

Tilt

- 20 lbs gas shocks (McMaster Carr 4138t532) stabilizes and locks tilt in place.
- A Neo-550 motor on a 100:1 Versa Planetary stack powers the tilt winch.
- The 0.75" hex profile on the winch allows for manual tilt as part of our pre-game setup.

Turret

- The custom 5mm HTD pulley for 9mm thick belts made from 6061 aluminum is bolted into a COTS bearing. Additional #10-32 threaded holes were also machined into the bearing itself to maintain the structural integrity of the subsystem after experiencing a mechanical issue post-SVR.
- A NEO Motor on a 5:1 MAXPlanetary Gearbox powers the turret.

Turret/Tilt

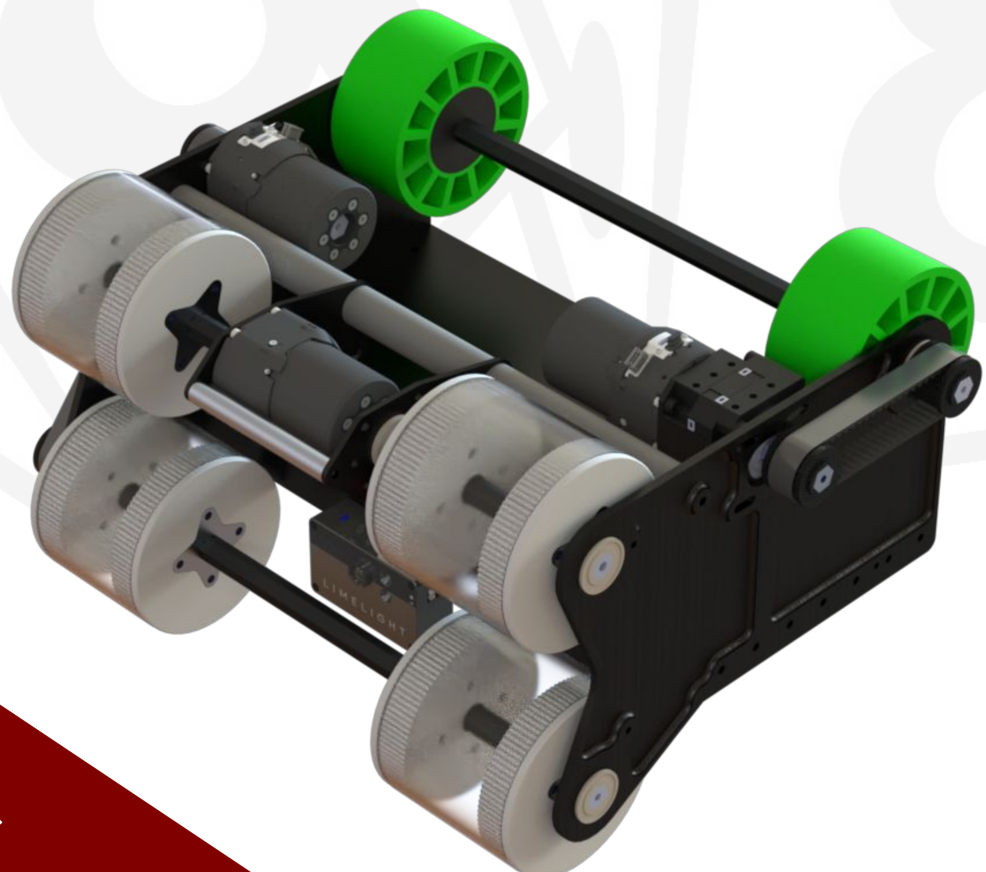
The shooter is configured and designed to maximize shooting range. We found that the vertical shooter wheel configuration had more range versus horizontal configurations during the prototyping stage of our robot. The whole shooter is powered by NEO Vortex Motors.

Important Programming Components

- 2 Integrated VEX ir sensors (276-2154) on the top wheel motor mounting plates and related software implementations prevent notes from launching while the shooter wheels are accelerating and allow for corrections in case of error during the note handoff process. In auto, these are used for on-the-fly adaptive pickup decisions.
- Pigeon IMU 2 located underneath the shooter base plate allows for accurate tilt angle readings.

Wheels

- Custom 4” polycarbonate shooter wheels with glued in 3D-printed hubs made out of polycarbonate filament.
- 4” 35A compliant wheels indexes and secures notes in place while in transit.
- 0.5” overall compression
- Top wheels take full advantage of the through-bore feature of the NEO Vortex motor, allowing for a more efficient packaging.



Shooter

Amp/Trap/Hand-off Arm

The arm consists of two connected telescopes that score the note into the amplifier and the trap. It also hands the note over from the intake to the shooter. While climbing, the arm is extended in order to place notes into the trap.

Extension

- Four 6.88lbs constant force springs (McMaster Carr, 9293k349), with two for each telescope, allows for an extremely fast one-time extension.
- Servo-powered 3D-printed spool, located in the middle portion of the cross bar, releases pins that secure the inner-stage, initiating the extension
- Delrin slider end caps to properly align and allow the inner stage to smoothly slide from the material's low friction properties.

Note Manipulation

- 3D-printed 4" passive rollers allow for precise trap access and scoring.
- Serpentine-driven rollers powered by BAG motor for note manipulation.

Our climb system is kept simple while being integrated to many subsystems. Even though past iterations allowed us to climb, their obstruction massively impacted our robots performance. One iteration was counterproductive, stopping the arm from scoring the trap without breaking the pivot gear. Our final iteration provided us with the most lightweight and least obstructive way to climb as well as score the trap.

Hooks

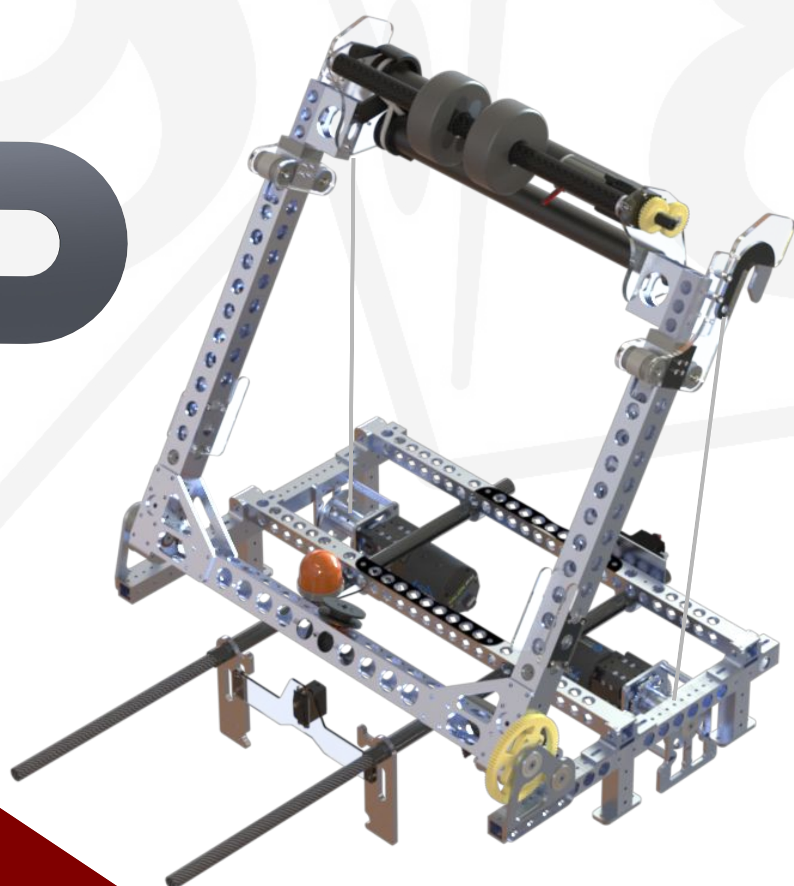
- Winch-driven rip-off hooks (integrated into the Amp Arm) made out of .1875" 6061 aluminum sheet lifts the whole robot.

Gearboxes

- 27:1 overall gear ratio on each winch gearboxes.
- Gearbox plates are also bolted directly to the belly pan for structural integrity.

Climb Stabilizing Forks

- Activated during climbing sequence to stabilize climb
- Schedule 80 PVC pipe outer stage
- Carbon-fiber extension
- Servo in the middle initiates extension as carbon fiber extension is then pulled by surgical tubings (not shown in render) tied to the rear servo mount plates.



Climber

Programming

- Centralized Architectures
- IDW Interpolation
- Turret Estimation
- Fully Automatic Shot Decision Making
- Alternate Path Decision Making
- Hardware Usage
- HRLogger

Centralized Architectures

- Our robot code is written in C++ to be as straightforward to the user while offering important features to excel in teleoperated and autonomous modes. We also made commitments to readability and ease of troubleshooting.

HRMotor

- HRMotor is our centralized Motor framework that supports all common FRC-specific REV and CTRE motors. It provides motor-specific and tuned starting-point controller constants for swerve drives, arms, and flywheels to make tuning less time-consuming.

HRControls

- HRControls is a solution we devised to centralize the way we manipulate our robot behavior, unifying control for autonomous and teleop periods. This largely came from the philosophy that we are not trying to make an autonomous robot, rather an autonomous driver. We identify then implement manipulability to key robot states (as variables). Acting as a state machine, We then change the values of these states in both autonomous and teleop periods. At its core, HRControls is a struct that contains all of our key robot states.
- *Top Level Example* :
    - *The **`goToNoteTracking`** state within the HRControls struct is a boolean value that tells the robot to track notes, whether in autonomous or teleop periods.*
        - *Teleop: When the driver presses the "B" button on the gamepad, set **`HRControls.goToNoteTracking`** to **true**.*
        - *Autonomous: During pre-programmed routines, the code sets the same **`HRControls.goToNoteTracking`** to **true** when note tracking is desired, and **false** when complete.*

Benefits

- Centralizing the way we manipulate core robot behaviors adheres to DRY Code (Don't Repeat Yourself) philosophy, allowing for increased readability by eliminating possible code repetitions which also leads to easy troubleshooting. We also know that if something works in teleop, it'll work in auto.

IDW Interpolation

- Crescendo presents a unique challenge: *as the distance to the goal increases, the scoring window shrinks, making long-range shots more demanding.*
- To address this, our robot employs an adaptive aiming strategy. We leverage adjustable tolerance windows for both pitch and yaw angles, measured in degrees. These windows define acceptable margins for error during the shot.
- Inverse Distance Weighting (IDW) plays a crucial role in determining the optimal size of these windows for any given range. Similar to interpolating on a graph, IDW intelligently analyzes data points collected through manual test shots at known field locations.
    - These test shots involve gradually increasing error until they miss the goal, creating reference points for the IDW algorithm. For instance, testing shots from the podium helps establish the robot's tolerance window for that specific distance.
- The beauty of IDW lies in its ability to consider any amount of data points while strategically weighting them based on the robot's current X and Y coordinates.
- Locations closer to a data point have a greater influence on the final tolerance window size.
    - This ensures the windows adapt seamlessly throughout the match, prioritizing pinpoint accuracy for long-range attempts while allowing for faster aiming at close distances.
- The addPointWithMirror function adds a point to the IDW table and successfully adds a mirror for it to the table for the opposite alliance that the data point was taken at as robot odometry is always referenced from the blue corner.

Datapoint example for a wing line shot:

```
ShotTable.addPointWithMirror(X:10.6, Y:0,
pitchTolerance:0.5, yawTolerance:0.5, shotSpeed:1);
```

Software Features

Turret Estimation

- To eliminate aiming with a controller joystick, our turret automatically aligns to the target whenever a note is detected by a VEX Line Tracking sensor. We employ basic trigonometry to accurately calculate the required aiming angle. Accounting for robot orientation, subtracting the robot's current rotation, ensures precise targeting regardless of the robot's overall heading.
- Example :

```
X = Robot Odometry X value;
Y = Robot Odometry Y Value;
targetX = Goal X position;
targetY = Goal Y position;
Omega = Robot Odometry Rotation in degrees;
deltaX = X - targetX;
deltaY = Y - targetY;
targetAngle = -(arctan(deltaX,deltaY)*(180/π)) - Omega;
```

- The code above would work, but can output angles larger than 360 degrees which would not be applicable for our turret, to solve that we made simple angle wrapping logic.
- Example :

```
while(targetAngle > 360){targetAngle -= 360;}
while(targetAngle < 0){targetAngle += 360;}
```

How do we tell the turret control motors this?

- We establish a relationship between encoder ticks and angle (in degrees). This allows us to translate desired target angles into motor commands the turret understands.
- We first physically rotate the turret a known number of degrees by hand. Simultaneously, we record the number of encoder ticks generated during this movement. By dividing the encoder ticks by the known number of degrees rotated, we establish a "ticks per degree" constant.
    - Once we have this constant, applying it becomes simple. We multiply the target angle, by the "ticks per degree" constant. This calculation translates the desired angle into the specific number of encoder ticks the motor needs to rotate. With this information, the motor receives a clear instruction to go to its intended position.

Fully Automatic Shot Decision-Making

- Shooting is a critical action in Crescendo, and our robot automates this process for optimal efficiency. Leveraging the previously mentioned turret estimation, the Limelight 3G camera seamlessly takes over aiming whenever the AprilTag on the target speaker is detected.

- A single boolean flag in the HRControls struct, aptly named "The Kill Button," acts as a master control for the entire shooting decision-making process. The robot continuously checks five key criteria:

    - ***Note Loaded***: Is a note present and ready to be fired from the robot?
    - **Shooter Ready**: Has the shooter motor achieved the required launch speed?
    - **Valid Target Detected**: Is a valid target (AprilTag) within the camera's field of view?
    - **Aiming Stability**: Have the yaw and pitch angles remained consistent for the past three program frames?
    - **Tolerance Met**: Do the current yaw and pitch values fall within the dynamic tolerance windows calculated by the IDW algorithm?

- If all five conditions are met, the robot takes action swiftly, likely surpassing human reaction time. It activates the handoff wheels at full power and fires the note. The "Kill Button" then gracefully transitions the robot back to the pickup state, allowing the operator to hold the button continuously for sustained autonomous shooting.

Alternate Path Decision-Making

- In Crescendo, the centrally located game pieces (notes) are prime targets for both alliances, creating opportunities to steal points from your opponent. With only five notes available, a single-note deficit can be decisive. To address this challenge, our robot prioritizes the ability to adapt its strategy on the fly.

Real-time Pickup Confirmation:

- During a pickup attempt, the robot continuously monitors two key data points for confirmation:
    - **Amp Roller Current Spike**: This indicates the amp roller engaging with a note.
    - **Shooter IR Sensors Activated**: These sensors detect a note entering the robot's shooter.
- This combination of checks offers exceptional reliability in determining successful pickup.

Data-Driven Decisions:

- The robot interprets the combination of these signals to make informed decisions:
    - **Note In Robot (Amp Roller Not Triggered, IR Sensors Activated)**: This confirms note acquisition, potentially indicating the amp roller's current threshold for detection needs adjustment. The robot then proceeds as normal.
    - **Note Entering Robot (Amp Roller Spike, IR Sensors Not Activated):** This signifies the note is mid-transit into the robot and it proceeds as normal.
    - **No Pickup Detected**: If neither signal activates, the robot waits briefly to account for potential delays due to factors like battery voltage.

Flexible Path Adaptation:

- If no pickup occurs after the brief wait, the robot runs an alternative path. A key advantage of our system, unlike others, is that our alternative paths seamlessly connect with the original intended path, enabling the robot to execute a single "fallback" maneuver and then resume its autonomous routine uninterrupted. This streamlines adaptation and minimizes time wasted on unsuccessful pickups.

Our robot uses numerous sensors to allow for smooth operation in both autonomous and teleoperated modes.

Drivetrain

- **Pigeon 2 IMU**: for autonomous heading control in both teleop and autonomous mode
- **Kraken X60 Integrated encoders**: track wheel rotations forming the foundation of our robust odometry system for accurate robot self-localization within autonomous tasks.

Shooter

- **Pigeon 2 IMU**: this IMU provides crucial absolute pitch readings, particularly useful for our winch system which leads to many variations on startup. Additionally, it calculates the chassis' relative yaw by comparing its own yaw data with the drivetrain IMU
- **2 VEX IR Sensors**: As briefly referenced in the shooter, we strategically positioned ir sensors to guarantee accurate note positioning within the shooter. One sensor acts as a failsafe, detecting notes that overshoot the first, while the other ensures precise placement for optimal launching. Those two work together to get perfect positioning every time.

Vision

- **Note Tracking Limelight 3 with Google Coral USB Accelerator** : Leveraging a pre-trained machine learning model, this limelight excels at detecting notes on the ground, allowing the robot to seamlessly center itself on notes in both teleop and autonomous.
- **2-Pose Estimating Limelight 3Gs**: Combining data from both of these limelights the robot is able to accurately center on the amp, source, and all 3 of the climb locations. These limelights also allow for constant updating of pose which is integrated into the MegaTag2 model.
- **Shooter Mounted Limelight 3G**: This limelight is used to target the AprilTag present on the speaker.

Sensor Usage

HRLogger

- A motor failure at Beach Blitz 2023 served as a wake-up call for our team. Initially misdiagnosed as a thermal shutdown, the culprit turned out to be a dead motor within a swerve module. This critical learning experience highlighted the need for a robust logging system to prevent similar catastrophic failures in the future.

- HRLogger is a powerful logging system designed to capture critical robot data throughout operation.

    - **Backend**: Continuously gathers data from various sensors and stores it in JSON files on the Roborio.
    - **Frontend**: Provides a user-friendly interface to visualize the recorded JSON data logs.

Easy Data Retrieval and Analysis

- Accessing logged data is simple. After a match, users can connect to the robot's WiFi or ethernet network and download the JSON files directly from the Rio.

Benefits of HRLogger:

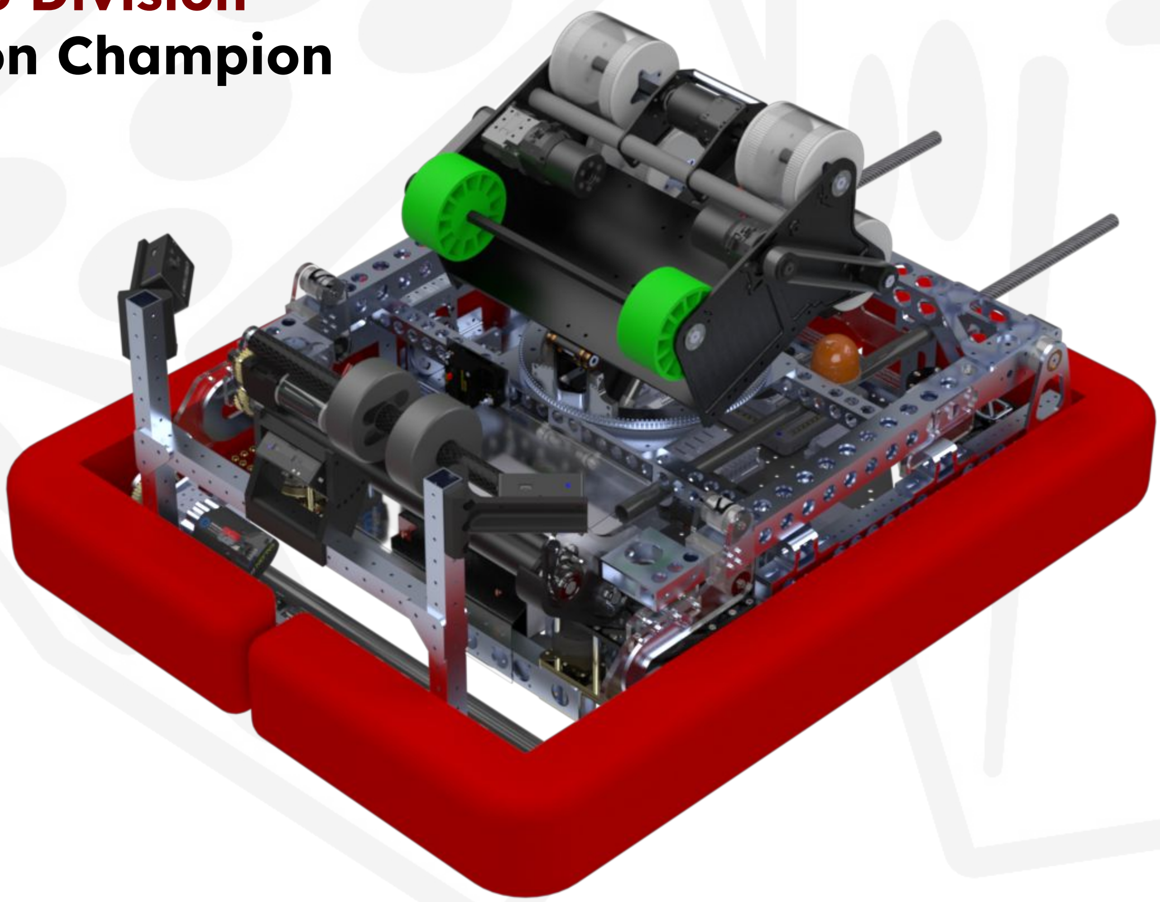
- **Wheel Slip Detection**: HRLogger's detailed logging helped us identify and address wheel slip at all three events this season. This allowed us to optimize acceleration profiles and maximize performance while minimizing slippage.
- **Proactive Thermal Management**: Thermal logging capabilities enable early detection of potential overheating issues within swerve modules, preventing motor failures before they occur.
- **Controller Input Logging**: HRLogger timestamps and visualizes controller inputs from both the driver and operator, allowing for post-match review and analysis. This resolves any confusion about robot behavior and clarifies whether an action originated from the robot or a controller misclick.

Custom Software Apps

Silicon Valley Regional
Autonomous Award

Las Vegas Regional
Excellence in Engineering & Regional Winner

FIRST Championship
Galileo Division
Division Champion



“It’s Not Enough!”

2024