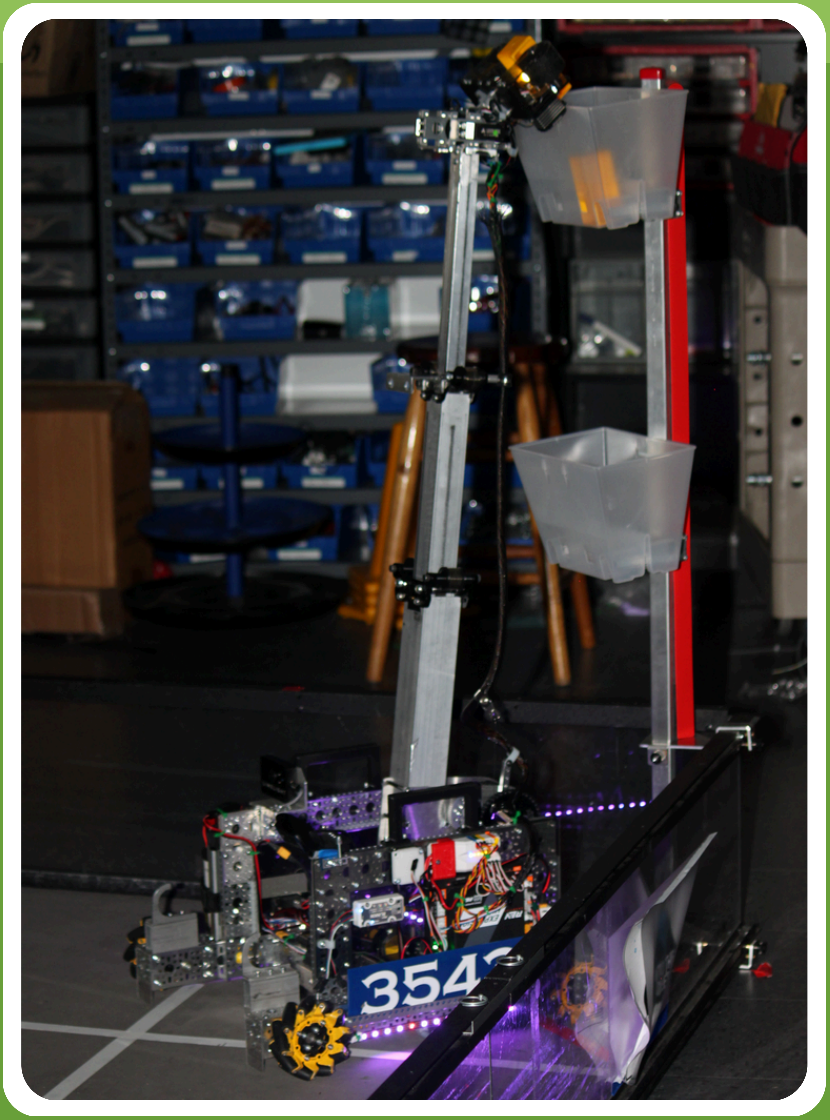
TITAN ROBOTICS FTC 3543



Presents...

ENGINEERING PORTFOLIO

2024-2025

Team Introduction

Hi! Welcome to our Engineering Portfolio! We're team #3543 Titan Robotics Club from International School of Bellevue.

Titan Robotics Club's original FRC team was formed in 2001 as a Senior Project, before expanding its reaches through our school to include FLL and FTC in 2009. At 15 years running, 3543 remains one of Washington State's oldest active FTC teams.

Our team is composed of 18 members. We have six 8th graders, three 9th graders, eight 10th graders, and one 11th grader. One-third of our team is composed of rookies this year, so we’re excited to take on this season with a good mix of veterans and energetic newcomers, consolidating our unique experiences and insights into 3543.

It fills us with joy and pride to introduce ourselves and our robot, Poseiden, for Into The Deep this year!

Our School

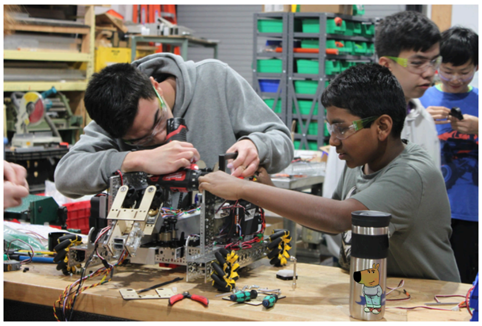
Our school is a 6-12 grade, choice/lottery-based college preparatory institution within the Bellevue School District. It boasts an enrollment of approximately 600 students, providing a focused and nurturing environment that fosters academic excellence and personal growth.

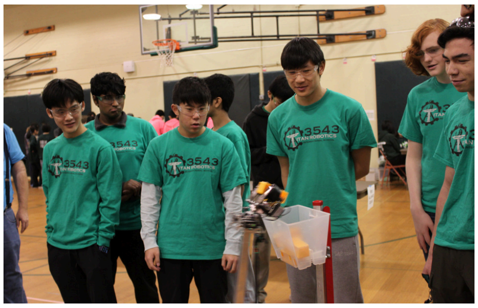
Our Club

Titan Robotics Club is a diverse family that creates a sustainable pipeline by including multiple teams through every grade level at our school. Titan Robotics Club provides the practical application to the theory taught in the classroom, allowing students to study science and technology outside of school while enjoying the fun of robotics! Although there are no traditional sports at International, we are proud to say that robotics is our sport, with nearly 1/6th of the school population participating in the club this year. Our **three FIRST Lego League (FLL)** teams introduce middle schoolers to the world of robotics and STEM. As these students progress to high school, many continue their involvement in the FIRST program by joining either our **FIRST Tech Challenge (FTC)** or **FIRST Robotics Competition (FRC)** teams. This allows them to build upon the skills and knowledge they gained in FLL and continue to explore the exciting world of robotics.

Our Team

We are a mix of different talents and passions united under FIRST. It is crucial to us that our robot is a result of everyone's work, from our mentors who guide us through building and programming, to our members who push themselves throughout the season to grow.





Team Member Profiles


Amos

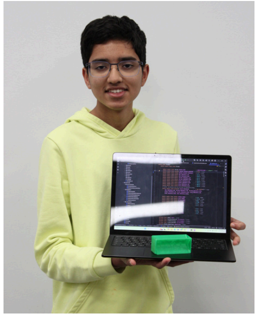

Avijit


Ashwin


Lucas


Daniel


Elliana


Aaryaman


Jason


Noah


Sam (Mentor)


Mike (Mentor)


Dave (Mentor)


Andy (Mentor)


Joseph (Mentor)


Jack


Anthony


Nidheesh


Hayden


Alex


Charlie


Benjamin


Ayan

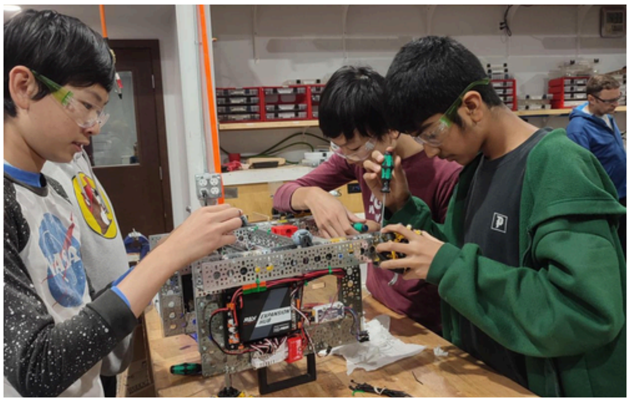

Peyton

School Interaction

Our Values

We always focus on the fundamental principles of FIRST, such as **gracious professionalism, cooperation,** and **growth**. We also nurture the core values of International School: **integrity, growth, collaboration,** and **creativity**.

We are always sure to support other teams, whether it be lending them parts at competitions or advising them online about how to troubleshoot and implement code.



Club Recruitment

We start at the 5th-grade barbeque, a special event where **prospective 5th graders** come to our school to eat lunch, to foster interest in those who will join the school in the future. We also advertise during our school's **Back to the Business Day, Club Fair**, **Bite Of IS,** and **interest meetings** where we are able to reach more of the student population, parents, and publicize the club more. For many people, **TRC is their first interaction** with robotics.

Training

Computer Aided Design

We hosted a student-led CAD training for Autodesk Inventor. This class started with basic application menus and commands, and then expanded to part modelling, assembly, and constraints.

Java Programming

We had mentor-taught beginner and intermediate Java programming classes. The beginner's class started with the basic fundamentals of programming and the language. The intermediate class went much deeper into the different modules of the TRCLib, our club's open-source library.

Robotics 101

Robotics 101 is intended to be experiential based learning that integrates CAD, fabrication, and programming. During each training, students learned about important concepts to help them succeed during the season, and they participated in hands-on projects, like building a swerve drive from scratch.

Safety Training

It is required that all members learn the names, whereabouts, procedures, and the many uses of each of the tools and machines in the workshop. This training covers things such as the importance of safety glasses and how to dress safely.

Team Sustainability

Foundation of Team Sustainability:

1) Financial stability (see Business Plan shown below)
2) Stable Mentor Base - Our core mentor base has been largely unchanged since the early days of the club
3) Stable build facility - We have secured a long-term build facility to enable the fabrication and assembly for FTC and FRC robots
4) Pipeline of interested students
5) Knowledge retention - See our wiki at ***wiki.titanrobotics.net***

FLL Connection

Older FTC and FRC students act as mentors for this program. They also assist in running and managing the club, giving them experience in project management. Through **student led mentorship**, FLL has been crucial in insuring younger students get valuable experience with programming and robotics before going into FTC. This creates a **sustainable cycle**, in which mentored younger students grow into mentors themselves.

FRC Connection

Through our special club structure, we are able to **share critical resources** with our FRC team, 492, such as our workspace and mentors, efficiently. Furthermore, resources such as our software library are used by both FTC and FRC, allowing us to find and reuse each others' capabilities. Some students participate in both programs, while others participate in one or the other while paying a single club due regardless.

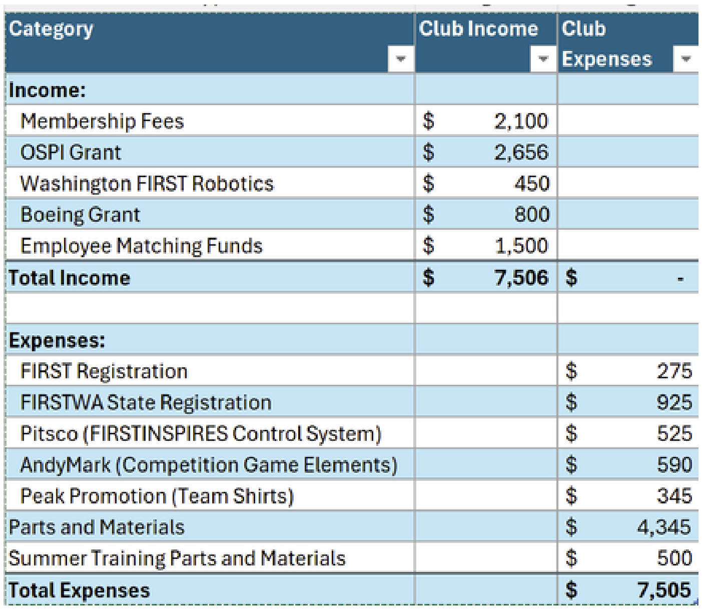
Business Plan

Titan Robotics Club has been able to **sustain itself over 24 years** due to consistent sponsors, grants, donations, and fundraising activities, which have brought stability to our FTC program finances. We have also been able to maintain a **financial reserve to ensure program viability**, even in the event of a loss of a key grant or donor.

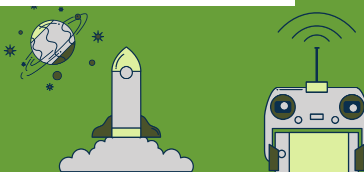
Perpetual Technologies is a generous donor that provides warehouse and meeting space offsite in Redmond. Both FTC and FRC meets offsite at the Redmond warehouse.

We work on a variety of funding sources, but a large amount of our **funding comes through grants** and/or company matching programs. OSPI, FIRST-WA, and Boeing have been long grant providers.

We also work to fundraise through our school and team community through mailing campaigns, FTC competitions, and smaller fundraisers.



Category	Club Income	Club Expenses
Income:		
Membership Fees	$ 2,100	
OSPI Grant	$ 2,656	
Washington FIRST Robotics	$ 450	
Boeing Grant	$ 800	
Employee Matching Funds	$ 1,500	
Total Income	**$ 7,506**	**$ -**
Expenses:		
FIRST Registration		$ 275
FIRSTWA State Registration		$ 925
Pitsco (FIRSTINSPIRES Control System)		$ 525
AndyMark (Competition Game Elements)		$ 590
Peak Promotion (Team Shirts)		$ 345
Parts and Materials		$ 4,345
Summer Training Parts and Materials		$ 500
Total Expenses		**$ 7,505**



Community Outreach

Learning from previous outreach events, this year we made sure to plan events as early as possible for good organization, and to make events as engaging as possible.

LMC Summer Camp

At the Little Master Club summer camp, kids were **introduced to the engineering design process** through examples from our previous robots. The kids even got a chance to drive a robot and learn about the different aspects of robotics. These concepts were applied to their very own engineering challenge that they worked on throughout the week.

Salvation Army

We showed off the robot at Salvation Army, and inspired the kids to do robotics in the future. We also taught a coding class to add to the excitement!

Elementary Schools

We were a popular hit at the Cherry Crest Family Engineering Night; kids loved driving and playing with the FRC bot! We also engaged in the local STEM community by visiting other elementary school Science Fairs.

Speaker Webinars

We invited different speakers in the STEM field to give webinars that we **hosted live**, and post on our YouTube channel.

Seafair Summer Festival

We demonstrated both FRC and FTC robots to the people of all ages at the 2023/24 Seafair Festival. Many people got a chance to learn about FIRST and drive the robot.

Strategy & Design

Subsystem Prioritization

During Kickoff, the first thing we did was deciding what functions we wanted the robot to have, and which ones were the most important.

	ID		Priority	Nice to have
Drivebase	1	Fast Robot	Nice to Have	1
Hint: Which obstacles can the robot go over?	2	Manuverable Robot	Must Have	
	3	Low Robot	Nice to Have	3
	4	Can Crab sideways	Must Have	
	5	Pushy Robot	Nice to Have	3
	7	High Ground Clearance	Nice to Have	2
	8	small form factor base	Nice to Have	3
	10			
Sample Intake	11	Pickup Samples	Must Have	
Hint: How do you want to pick up the samples/specimens?	12	Pickup Specimens	Must Have	
	13	manipulate specimen orientation	Nice to Have	3
	14	color detection/recognition	Must Have	
	15	detect multiple samples	Nice to Have	2
	16	retreive speciman from the wall	Nice to Have	1
	18	Detect Sample Pickup	Must Have	
	19	Auto-assist pickup	Nice to Have	1
	20	Long reach pickup	Nice to Have	2
		Short reach pickup	Must Have	
Sample Scoring	21	Pick up from one side/score from the other	Nice to Have	2
Hint: Where do you want to score the samples/specimens?	22	Scoring Lower Chamber	Nice to Have	1
	23	Scoring Upper Chamber	Nice to Have	3
	24	Scoring Low Basket	Must Have	
	25	Scoring High Basket	Nice to Have	2
	26	Score in Net Zone	Must Have	
	27	Push samples	Must Have	

TRC Open-Source Software Library

Our robot software is based on TRCLib, our own open-source software library with features such as Pure Pursuit drive, vector-based odometry, and abstraction layer for code compatibility between FRC and FTC robots. This library, which is **publicly available** on GITHUB, has been used by FTC teams around the world, such as Rise of Hephaestus in San Diego, Fixit in Victoria, Canada, and Violet Fusion in Florida.

The CAD files for each year's robot are also saved in a **CAD library**, serving as reference for future years' games. We also keep meeting notes and maintain a guide for the various tools available at our shop.

Design/Build Process

Digital Prototyping

Applying what we learned in training, we used Autodesk Inventor (CAD) to digitally model and assemble each of the subsystems. This allowed us to eliminate interferences, perform spatial integration, quickly iterate the design, and eliminate multiple fabrication cycles, which ultimately sped up the design process.

Rapid Prototyping

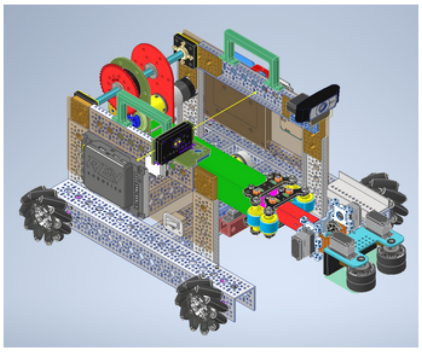
Once we are comfortable with the design, we use 3D printing, a laser cutter, or water jet cutter to **rapidly fabricate** parts using basic materials, which lets us build up a functional mechanism to test. Through this process we are able to further refine our designs.

Modular Design Approach

Concurrent with spatial integration, we also employ a modular approach to guarantee interchangeability. This is also to ensure that parts are accessible and aren't stacked on top of others during assembly. We have used this both in and out of competition to **quickly change out parts and upgrade subsystems**.

Ruggedization of the Parts

Although we prototype with wood and plastics, prior to competition, we update our parts by re-manufacturing out polycarbonate and aluminum parts where durability and strength are needed most.

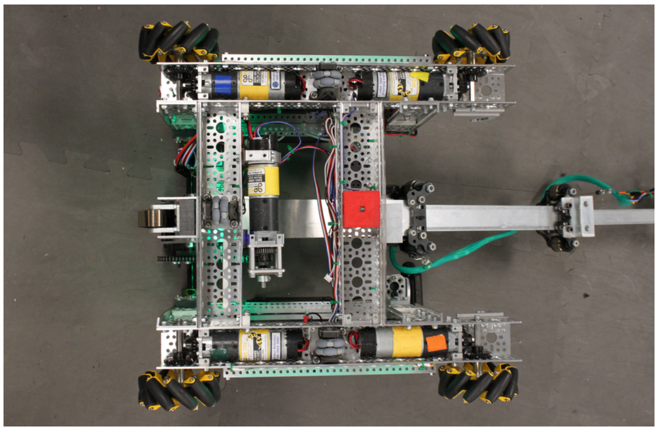


Drivebase & Climb

Mecanum Drive

We decided to use a mecanum drive base again this year. To achieve this, we used a classic **Gobilda strafer** base. The stock metric C-channels allow us to assemble a base that fits just within the allowed 18" square form factor. After we developed our subsystems, we adjusted the position of the cross C-channels to integrate with the game specific mechanism design. You can see that the drive motors are nested into the side channels, allowing the **Optii odometry mechanism** to mount in between the two motors at approximately the center of the beams. The **orthogonal odometry** mechanism is located at the center of the back cross beam. The pseudo H-frame has an open front to create space for the intake mechanism.

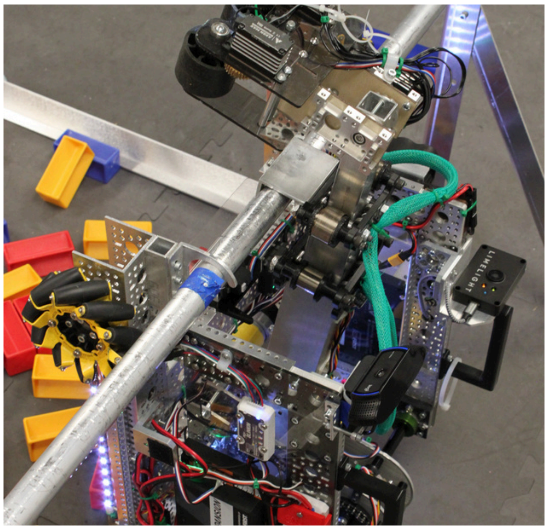
Final Drivebase Design



This is an image of our final drive base design, showcasing our changes to the beams to install subsystems, like odometry.

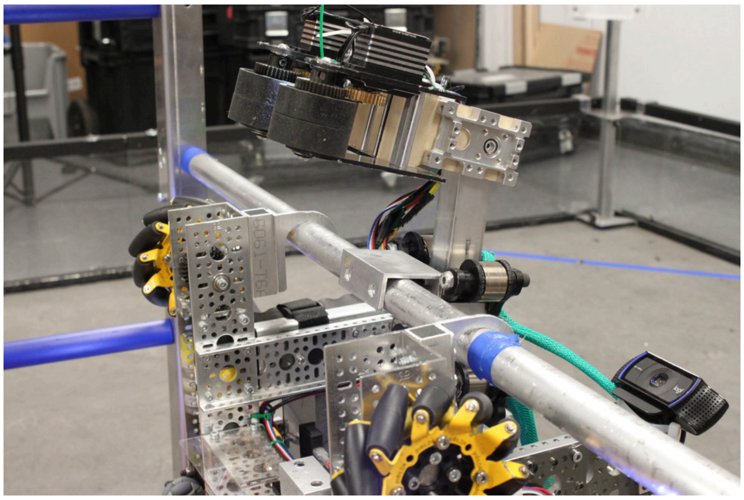
Climb Design

We added a hook to the top of our extender's first stage in order to lift our robot off of the mat, experimenting with correct placement to achieve the intended lift. We then use the elbow gearbox to rotate the robot base 90 degrees. Finally, we added a pair of fixed hooks on the front of the base so the robot would **stay in place even when power was cut.**



Final Climb Design

Following league 1 we evaluated climbing to a level 3 ascent. After initial brainstorming, we determined that level 3 would require a complex mechanism and similarly complex design work. Since level 3 ascent was low on our strategy priorities, we decided not to pursue it. Furthermore, during endgame, instead of hanging, (15 points,) we can score 2 samples, (16 points,) and possibly park in that same time.

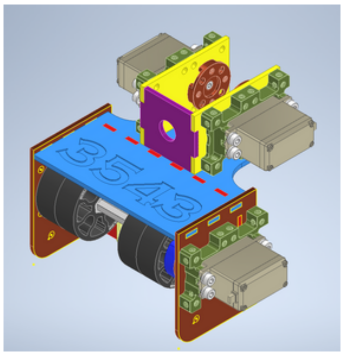


Intake & Wrist

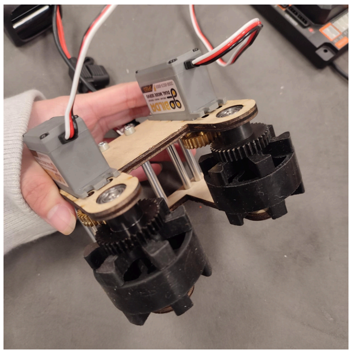
Intake Prototyping

This season, we started by dividing ourselves into multiple groups to brainstorm different intake designs. We decided the intake was a top priority, as it is essential for quick interactions with the game pieces. We came up with various designs including a claw, a “diving board” intake, and a beater intake. Mentors helped us think through potential issues and ways to make possible intake designs more efficient. Using CAD to speed up the design process, we quickly cut side panels to put together prototypes. We chose a beater as our intake because it is fast and was capable of picking up both samples and specimens quickly from multiple angles. We accommodated the specimen hook by adding a cutout in the side panels. During League 1, the intake missed the basket almost half of the time, so we changed the original **boot wheels to gecko wheels**, in order to eject the samples consistently.

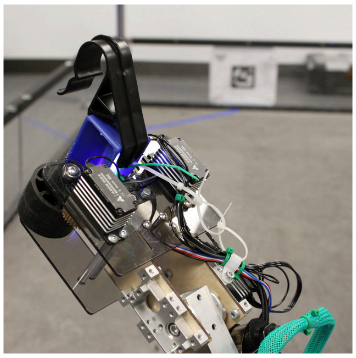
Claw Prototype



“Diving Board“ Intake



Beater Prototype



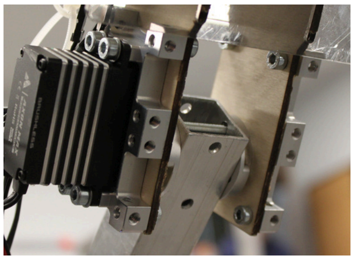
Final Intake

Differential Wrist

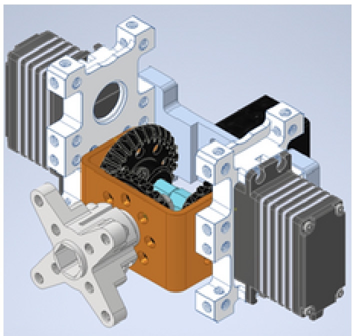
A large problem with our intake in our League 1 competition was that we could **only intake when the sample was perfectly aligned**, or else we were at risk of missing the sample.

We implemented a truly **unique** design using two Axon high torque servos to control the wrist. When both servos are turning in the same direction, it allows the end effector to rotate up and down 180 deg. When the servos are turning in opposite directions, it allows the wrist to rotate side to side 180 deg.

This was a **significant improvement** in the design between League 1 and League 2. By adding this additional degree of freedom, the operator is able to change the orientation of the wrist without needing to physically rotate the robot. This new change was incredibly beneficial, and key to our successes.



Wrist Version 1



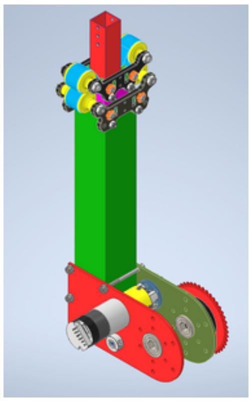
Differential Wrist

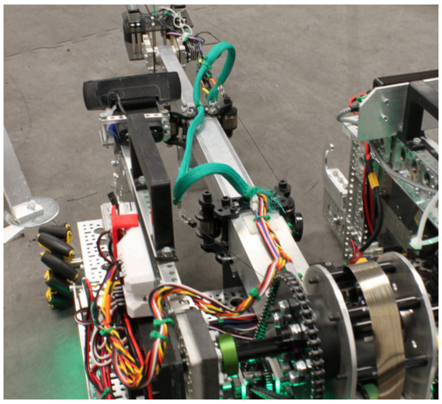
Extender & Elbow

Our arm design for this season is a extender that consists of 2 subsystems joined together to create our Extender Arm: Extender and Elbow. Our extender is a **2-stage telescoping** extender designed to reach the high basket with ease. We chose this design due to it's compact and sturdy construction. It uses **constant-force springs** to extend the telescope, and a winch to retract it.

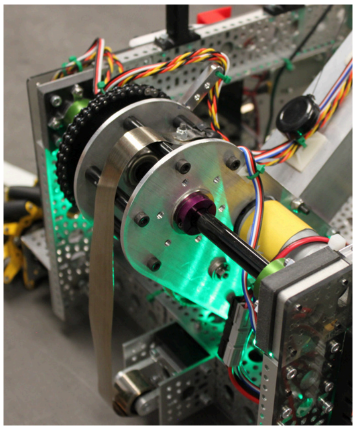
As with many design concepts, there were a series of inter-related trade-offs in the design to meet our design objectives. We traded between retraction speed (via gear ratios) and pull up force that affected our ability to climb in the end game. Similarly, we had to trade between the constant force spring strength and our ability to keep the telescope retracted in the unpowered condition.

One problem that we experienced was that friction of the telescoping tubes sometimes impacted our ability to fully extend the telescope. Because the extension was passive, we had no way to drive it to full extension. A concept that we evaluated was to replace the spring mechanism with a belt-driven extender. Because of the short turn around time between each of the competition events, we decided to keep our spring-loaded extender. If we advance past the semi-finals competition, we may revisit this decision. Otherwise, this redesign will be added to our priority list of summer projects to develop/optimize for use at some time in the future.

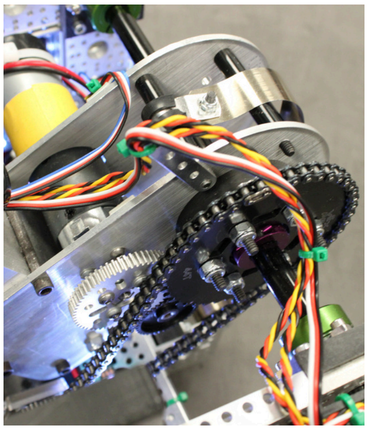




Elbow



Our elbow consists of two **custom-milled aluminum plates**, with offsets on the elbow in order for the extender to extend parallel to the ground when picking up and is designed for the **dead axle pivot**. Our elbow houses the motor for the extender winch, and it pivots itself using a gear over a dead axle to reduce backlash. This is all connected by chain to a stationary motor below. Putting these things together was challenging, so Avijit worked with Sam to understand the order that the parts should go together. They had to disassemble and reassemble sections of the elbow multiple times.



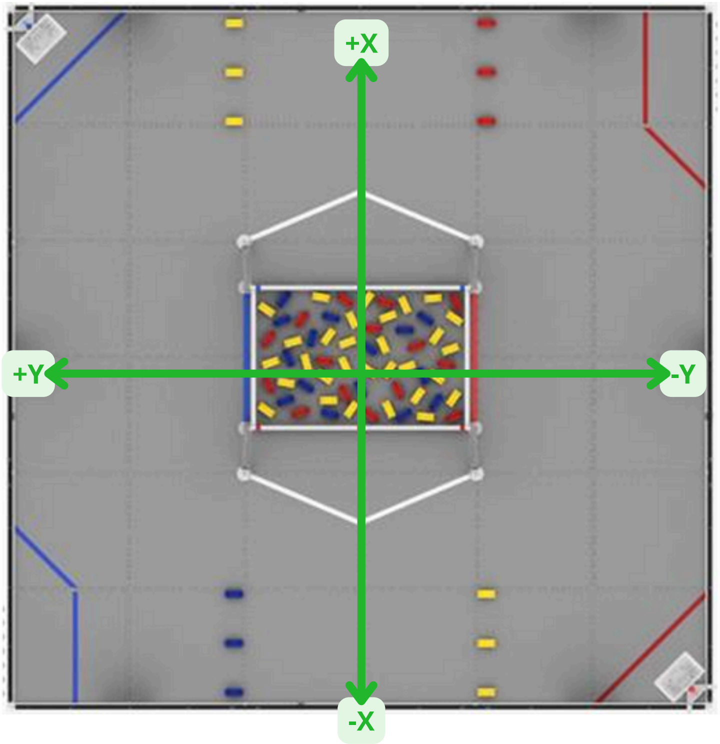
Odometry Background

How We Describe Robot Position

- Robot Position is described as absolute field coordinates in inches. However, for user accessibility, software is programmed in floor tile units. We convert them back to inches to be compliant with the rest of our library software.
- Positions are described as a triplet (x, y, heading).

What is Odometry?

- As the robot moves, we measure the change in position of robot from its previous position using data from odometry sensors (odometry pods and IMU) and integrate the changes to track the robot's absolute field location. This used to be done in our library running a sample rate of 50 Hz.
- This season, we use the Gobilda Pinpoint Odometry computer that can track the robot's position at a stunning sampling rate of 1500 Hz.



Why Do We Need Odometry?

- Pure Pursuit is a path following algorithm that requires path points described in absolute field coordinates and absolute odometry keeps track of the robot's absolute position guiding the Pure Pursuit algorithm.
- Easier to debug - We can print out the robot's position in our trace log so that if it executes a pure pursuit path incorrectly, we can check where odometry thought the robot was located.

How Do We Implement Odometry?

- **We used to use:**
  - Live-wheel-odometry: Taking data from the encoders on our drive train motors.
  - **But:** If any of the wheels slip—which is common when driving with mecanum wheels—the odometry gradually become inaccurate.

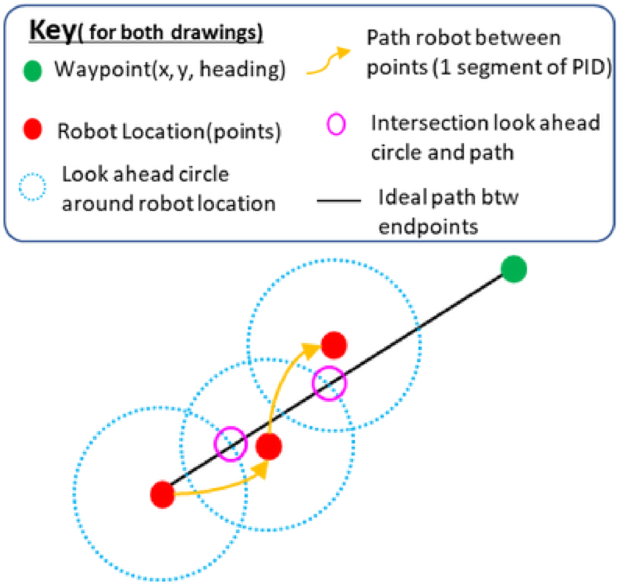
- **Solution: (Passive or "Dead Wheel" Odometry)**
  - We now use **unpowered Omni wheels** with encoders on springs
  - Now, even if the driving wheels slip or if the robot got hit, we still have these engaged on the ground tracking robot position accurately.

Pure Pursuit + Odometry

What is Pure Pursuit?

Pure Pursuit is a path following algorithm that uses the robot location from odometry to follow a curved path generated from path points. It uses a look ahead circle to intersect the path forming the look ahead target point which it endlessly pursuits while travelling along the path until we reach the final target at the end of path. To follow the path smoothly, our Pure Pursuit also employs **Motion Profiling** where it accelerates at the start of a waypoint to a specified max velocity and decelerates when approaching the next waypoint. This essentially slows down the robot on turns.

The Standard Pathway (Zig-Zag)



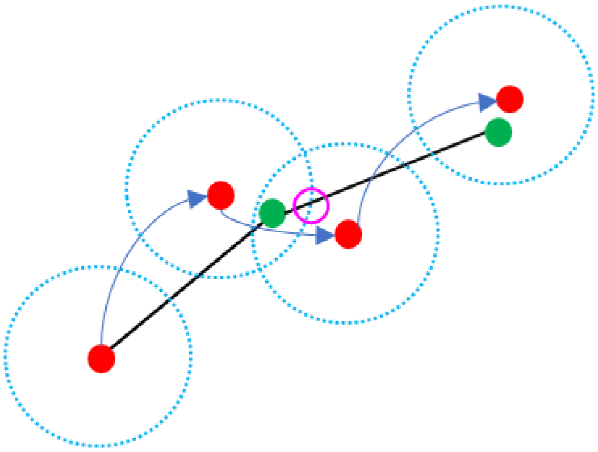
Downside to Our Old Algorithm

The radius of the look-ahead circle is generally about 6 inches and our old algorithm therefore relied on PID-controlled drives with targets of 6 inches or less. This caused the robot to move very slowly: because of drive error, the robot **would often overshoot** the intersection point and end up zig zagging around the path (i.e. oscillating along the path) instead of following it in a straight line.

Our Improved Algorithm

- To remedy these problems, we implemented v2 Pure Pursuit, which generates even **smoother paths**
- If the distance of the robot to the next point is more than the look-ahead circle radius, we bypass pure pursuit by setting a PID target to the next waypoint instead of the intersection between the look-ahead circle on a path. Once the distance is inside the radius, we switch back to traditional pure pursuit.
- This **solves both problems** that occur when the PID target is far away: PID controlled drive is a lot faster getting to the next waypoint -- it's more accurately following a straight line.

Our Improved Algorithm Path



Computer Vision

OpenCV

We are using the generic color blob detection OpenCV pipeline in our software library to locate samples of different colors. From the vision detection result, we calculate the sample angle, and employ Homography (*see Homography section below*) to calculate the real-world location of the detected samples.

LEDs

We currently have 7 LED Colors for assistance

Violet: Field mode orientation

White: Robot mode orientation

Aqua: No sample detected

Yellow: Yellow Sample picked up

Red: Red Sample/Specimen picked up

Blue: Blue Sample/Specimen picked up

Green: Detected April Tag

We can utilize LEDs to tell us what sample is detected or picked up when we aren't able to see it. This **can prevent penalties** from picking up the wrong sample color.

Camera

We use a **Logitech C920** camera that helps us with sample detection.

See *Technology*

OpenCV Pipelines

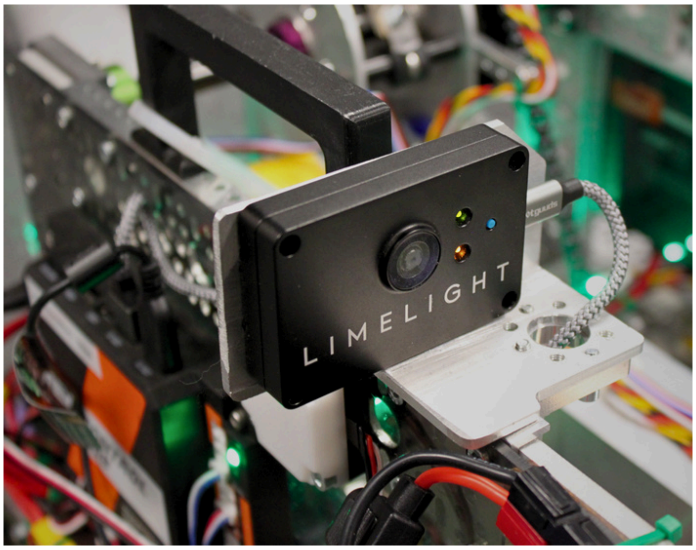
In order to look for an object's position, we use OpenCV **color blob** pipelines that then takes an image of what the webcam sees and applies a combination of color thresholding and morphological operations to narrow down on our desired subject.

Our pipeline for the Red Sample follows these steps:

1. Convert the color space from RGB to YCrCb
2. Color threshold filtering for the Red color
3. Find and filter contours (This is the blob part of the blob detection, removing stray pixels)
4. Criteria filtering (width, height, perimeter, area, aspect ratio, etc.)

Limelight

This year, we are using Limelight to detect game objects off the wall and for detecting AprilTag for **robot re-localization.**



Autonomous & Auto Assists

Context

In our practices, we realized that the drive team was having difficulty with scoring in the basket and the chamber. As a result, Aaryaman decided to create and refine multiple auto assists to make the scoring process easier.

Auto-Assists

Auto Climb:
This is our auto assist task that performs a climb during the endgame stage of a match. With the press of a button, the robot will climb on its own using pre-programmed code, performing different level ascents based on how many times the button is pressed.

Auto Pickup Sample from Ground:
This is our auto assist task that picks up samples from the ground. It uses vision to detect the location of the sample, turn the robot to it and extend the arm to pick up the sample. It can pick up from the spike marks as well as from the submersible.

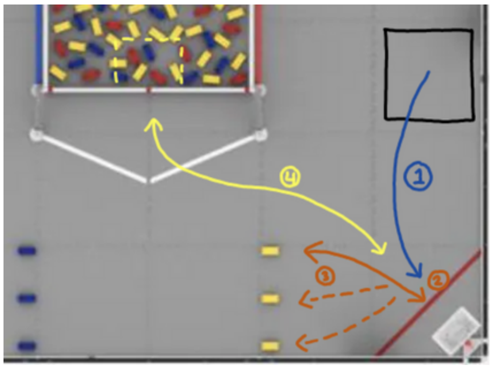
Auto Score Basket:
This is our auto assist task that scores a sample in the low or high basket. After the score height has been chosen, the robot will drive to the basket location on the field and move the elbow, extender and wrist to score the sample in the basket.

Auto Score Chamber:
This is our auto assist task that scores a specimen on the low or high rung. After the score height has been chosen, the robot will drive to the chamber location and move the elbow, extender and wrist to score the specimen on the selected rung.

Auto Pickup Specimen:
This is our auto assist task that picks up a specimen hung on the field wall. The robot will drive to the field wall where the specimen is hung. It will move the elbow, extender and wrist to pick up the specimen.

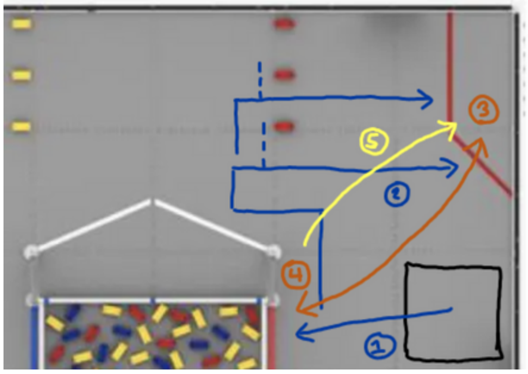
Basket Autonomous

1) Auto score pre-loaded sample.
2) Auto pick up sample from first spike mark using vision.
3) Auto score the sample to the basket.
Repeat step 2 and 3 for the second and third spike mark.
4) Go to the ascent zone and use vision to pick up a sample from the submersible. After scoring the sample, go back to the Ascent Zone and get a level 1 ascent.



Specimen Autonomous

1) Auto score the pre-loaded specimen.
2) Push two or three samples from the spike marks to the observation zone.
3) Auto pick up converted specimen from the field wall.
4) Auto score specimen on the chamber.
Repeat steps 3-4 for each specimen.
5) Park at the observation zone.



Technology

What is Homography

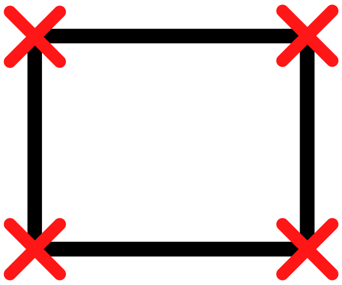
After we find objects with EasyOpenCV object detection, we can convert their pixel coordinates into real-world coordinates (x, y, angle), allowing us to navigate the robot towards it.

How it Works: The four corners of the camera screen form an inverted **trapezoid** on the ground in the real world due to 3D perspective. If we provide a mapping of the four screen pixel points to four real world floor points, we can use matrix math to calculate any pixel point on the screen to real world floor coordinates.

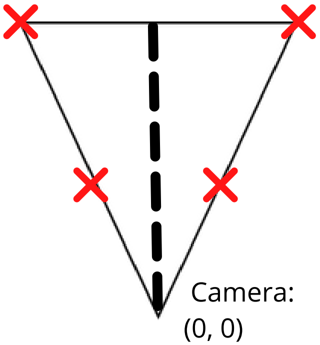
To Calibrate Homography:

1. Turn on the camera and put tapes on field where the four corners of camera screen are.
2. Measure the real-world coordinates of these corners relative to the camera.
3. Using our library, create a transformation matrix that maps the four screen corners to the four real-world coordinates.

Camera screen:



Real-world floor:



Applying Homography?

We can use vision with Homography in both Auto and TeleOp to **automate sample and specimen pickup**, saving us precious time.

Gravity Compensation

The force of gravity can be troublesome when rotating and extending the extender. This is why we **mathematically compensate** for the force of gravity in our code. This allows the extender to be easily controlled and hold its position. To set this value, we use an equation similar to the projected length below due to our offset.

Fitting Inside the Sizing Box

In this season, a new restriction is imposed on the robot not to exceed a specified floor space at all times. Since our extender arm needs to extend high enough to score into the high basket, it would exceed the floor space restriction if the arm was swung down on the floor fully extended. To solve this, we have implemented a **smart algorithm** for controlling the arm, such that it **will automatically retract the extender** as needed to make the projected floor space of the arm within restriction. The following shows the derivation using geometry and trigonometry calculations

https://github.com/trc492/Ftc2025IntoTheDeep/blob/main/TeamCode/src/main/java/teamcode/FtcTeleOp.java#L269

$$x = elbow\ angle$$

$$y = \max extender\ len$$

$$intake = intakelen \cdot \sin\left(\frac{currentPos - minPos}{maxPos - minPos} \cdot \pi\right)$$

$$\cos\left(x - \operatorname{atan}\left(\frac{offset}{y + intake}\right)\right) \cdot \csc\left(\operatorname{atan}\left(\frac{\text{offset}}{\text{y + intake}}\right)\right) \cdot \text{offset} = \text{maxLen}$$

$$\left(\cos(x)\cos\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right) + \sin(\mathrm{x})\sin\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right)\right) \cdot \csc\left(\operatorname{atan}\left(\frac{\text{offset}}{\text{y + intake}}\right)\right) \cdot \text{offset} = \text{maxLen}$$

$$\left(\cos(x)\cos\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right) + \sin(\mathrm{x})\sin\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right)\right) \cdot \frac{1}{\sin\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right)} \cdot \text{offset} = \text{maxLen}$$

$$\left(\cos(x)\frac{\cos\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right)}{\sin\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right)} + \sin(\mathrm{x})\right) \cdot \text{offset} = \text{maxLen}$$

$$\left(\cos(x)\cot\left(\operatorname{atan}\left(\frac{offset}{y + intake}\right)\right) + \sin(\mathrm{x})\right) \cdot \text{offset} = \text{maxLen}$$

$$\left(\cos(x)\frac{\text{y + intake}}{\text{offset}} + \sin(\mathrm{x})\right) \cdot \text{offset} = \text{maxLen}$$

$$\cos(x)\frac{\text{y + intake}}{\text{offset}} + \sin(\mathrm{x}) = \frac{\text{maxLen}}{\text{offset}}$$

$$\frac{\text{y + intake}}{\text{offset}} = \frac{\frac{\text{maxLen}}{\text{offset}} - \sin(\mathrm{x})}{\cos(x)}$$

$$\text{y + intake} = \frac{\text{maxLen} - \sin(\mathrm{x}) \cdot \text{offset}}{\cos(x)}$$

$$\mathrm{y} = \frac{\text{maxLen} - \sin(\mathrm{x}) \cdot \text{offset}}{\cos(x)} - \text{intake}$$