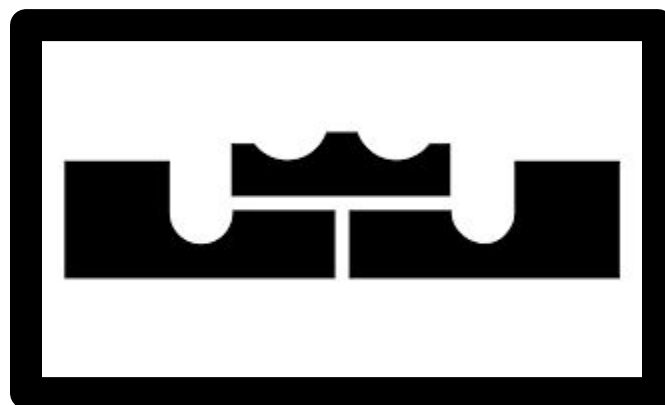


**A notebook documenting the process of
designing and creating our robot**



Start Date: May 20th, 2025
Team Number: 8115x



ver Page	1
Table of Contents	2
Notebook Setup	8
Biographies / Team Introduction	9
Design Process	10
Field Outline	11
Game Rules	13
Game Strategy	16
DESIGN 1: Early Season Robot	19
Drivetrain - Goals and Constraints	20
Breakdown - Robot Breakdown	21
Drivetrain - Type of Drivetrain	23
Drivetrain - Wheel Size	25
Drivetrain - Type and Wheel Size (Decision Matrix)	27
Drivetrain - Gear Ratio and Speed	28
Drivetrain - Concepts	29
Drivetrain - Select Design pt 2	30
Drivetrain - Assembly	31
Drivetrain - Programmed Controls	32
Intake - Goals and Constraints	33
Intake - Types of Intakes	34
Intake - Select Design	36
Intake - Programmed Controls	37
Scoring - Trying to Score on All Goals	38
Scoring - Mechanism - All Goals (AG)	39
Scoring - AG Select Design	40
Lift - Lift Design Brainstorm	41
Scoring - AG Select Design (cont.)	42
Scoring - AG Lift Assembly	44
Scoring - Lift Programmed Controls	46
Storage - Funneling Balls and Storing	47
Storage - Motor + Mechanisms	49
Scoring Mechanism	50
Aligner - Type of Intake	52
DESIGN 2: Under Goal Robot	53
Reasons for Changing Our Bot	54
Game Strategy 2	55
Long Goal Mech - Goals and Constraints	56
Long Goal Mech - Type of Mech	57
Long Goal Mech - Select Design	58

Ball Storage - General Overview	60
Ball Storage - Select Design	61
Ball Storage - Problem 1	62
Ball Storage - Problem 2	63
Ball Storage - Backboard Material Selection	64
Ball Storage Mech - Assembly	65
Long Goal - Goals and Constraints (Scoring)	66
Scoring Mechanism - Select Material	67
Scoring Mechanism - Problem 1	68
Intake - Select Design (cont.)	69
Aligner - Goals and Constraints	70
Intake - General Overview	71
Intake Application	72
Match Loader - General Overview	73
Match Loader Application	74
Odometry - General Overview	75
Autonomous - Match Autonomous (Odometry Setup)	76
Autonomous - Match Autonomous (IMU Code)	77
Autonomous - 4 Ball Long Goal Auto	78
Tournament Preparation (Driver Practice)	79
Tournament Preparation (Auto Testing)	80
Auckland V5RC #6: Tournament	81
Planned Improvements	82
Competition Issues 2	84
Auckland V5RC #10: Tournament	85
DESIGN 3: Front to Back Scoring Robot	86
Game Strategy 3	87
Drivetrain - Improvement 1	88
Match Loader - Improvement	89
Intake Design - Goals and Constraints	90
Intake - Type of Mech	91
Intake Design - Select Design	93
Ball Storage - Design Modification	95
Intake - Test Solution Goals	96
Ball Storage - Improvement 1	97
Ball Storage - Improvement 2	98
Central Goal Aligner	99
Descoring Mechanism - Improvement	100
Central Goal Aligner (cont.)	101
Descore Hooks - Goals and Constraints	102

Descore Hook - Select Design	103
Descore Hook - Assembly	104
Descore Hook - Test Solution Goals	105
Descore Hook - Programmed Controls	106
Intake - Programmed Controls	107
Autonomous - Solo Winpoint	108
Autonomous - 7 Ball Wing	109
Auckland V5RC #7: Tournament	110
Planned Improvements	111
Game Strategy 3 (Updated)	112
Match Loader Refinements	113
Aligner Refinements	114
Aligner - Design Options	115
Aligner Design - Select Design	116
Wall Stake Mech - Assembly	117
Descore Wing and Hook - Select Design	118
Descore Hook - Assembly (v2)	119
Auckland V5RC #8: Tournament	120
Competition Issues 2	121
DESIGN 4: Nationals Robot (Front to Back Roller)	122
Game Strategy 3 (Nationals)	123
Intake - Type of Mech	124
Intake Design - Select Design	125
Intake - Assembly	127
Drivetrain - Improvement	128
Long Goal Aligner - Modification	129
Middle Goal Scoring - Type of Mech	130
Middle Goal Scoring - Select Design	131
Central Goal Descoring Mechanism	132
Intake - Test Solution Goals	133
Middle Goal Scoring - Test Solution Goals	134
Mogo Holder - Type of Holder	135
Intake Design - Select Design (Anti-Jam)	136
Intake - Assembly (Sprockets)	137
Middle Goal Scoring - Test Solution Goals (v2)	138
Intake - Backboard Material Selection	139
Wing Aligner	140
Autonomous Path Following Algorithm	141
Distance Sensors - Test Solution Goals	145
Autonomous - Skills Auto (96 Route)	146

Middle Goal Aligner - Brainstorm	147
Middle Goal Aligner - Design Matrix	148
Middle Goal Aligner - Iteration 1	149
Middle Goal Aligner - Iteration 1 Eval	150
Middle Goal Aligner - Iteration 2	151
Middle Goal Aligner - Iteration 2 Eval	152
Middle Goal Aligner - Iteration 3	153
Middle Goal Aligner - Iteration 3 Eval	154
Weight Distribution - Evaluation	155

Section

Subsection | Page #

Title

Headers underlined
Descriptions below

Signifies empty space

Identify the Problem

Brainstorm Solutions

Select and Plan

Build and Program

Test Solution

Daily Notebook

Guide

Design Process

Notebook Outline

Table

Category #1	Description
Category #2	Description

Image

Our notebook is divided into 7 color-coded sections, 5 allocated to the design process and 2 for outlining notebook procedures and daily build sessions. Formatting the notebook in this manner isn't only helpful from a reader's perspective, but also helps us write the notebook efficiently, ensuring that each task completed in each build has a corresponding section to be orderly inputted into.

Team Introduction

We are Team 8115X, a select group of four high school students who are extremely passionate about the VEX Robotics Competition. All of our members have around 1-3 years experience in VEX and most have received awards at a high level. This season, we aim to build upon our progress and serve as an example to younger teams as it is our last year competing in VEX.

Jonathan Gong



- Grade: 12
- Experience: 3 years
- Roles: Builder, Drive, Programming

Eric Qin



- Grade: 12
- Experience: 3 years
- Roles: Builder, Drive Coach, Scout

Maique Pang



- Grade: 12
- Experience: 3 years
- Roles: Builder, Alliance Drive Coach, Scout

Dominic Xia



- Grade: 12
- Experience: 6 months
- Roles: Builder, Scout, Notebooker

Team Roles

Role	Member(s)
Builder	All members
Designer	Jonathan Gong, Eric Qin, Dominic xia
Programmer	Jonathan Gong
Notebook	Everyone
Driver	Jonathan Gong
Drive Coach	Eric Qin
Alliance Drive Coach	Maique Pang
Scout	Eric Qin
Match Strategy	Maique Pang

Goal

We will list the steps of the engineering design process so that we can adhere to them this season.

Engineering Design Process:

Define the problem 

Brainstorm, Diagram, or Prototype 

Select and Plan 

Build a model 

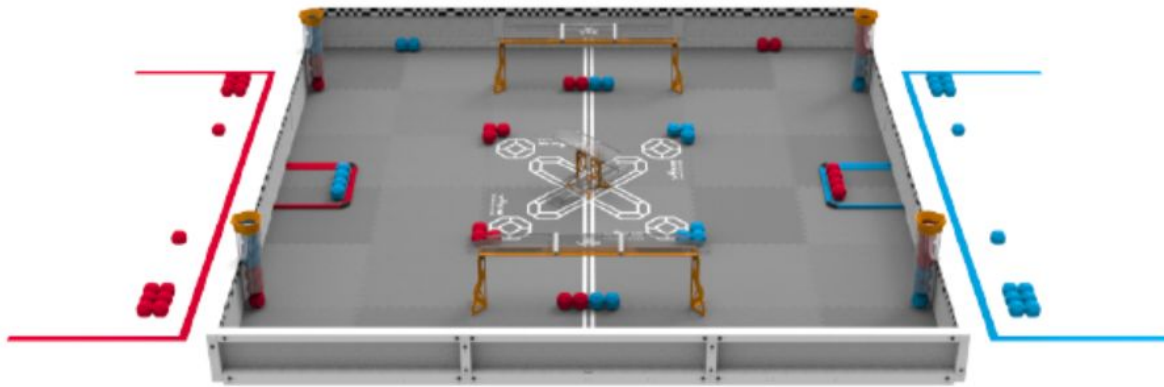
Evaluate and Improve our design 

We will write daily notebook entries summarizing our work during our team meetings and connecting it to the corresponding step in the engineering design process. We believe this industry-proven process will keep us organized and ensure we are always focused on our goals.

Goal

We will analyze the Push Back field so we can better understand this year's game.

VRC Push Back Field



Field Elements

Ring

- 88 total
 - 44 red
 - 44 blue

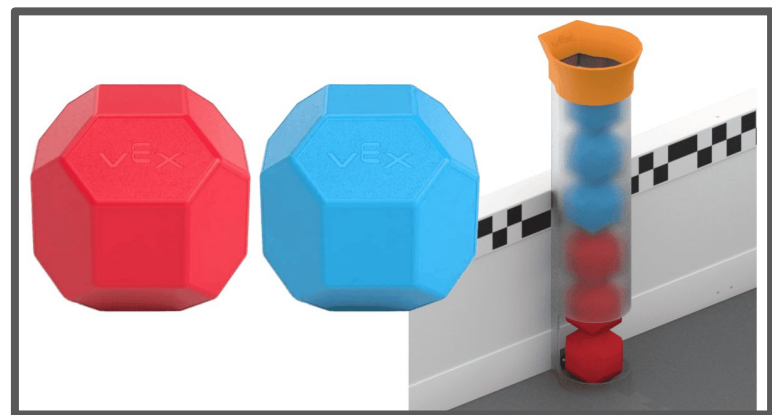
How they are used?

- 1 Preload per Robot (2 per Alliance)
- 12 Match Loads per color, held by the Drive Team
- 36 per color start on the field in fixed locations
- 12 balls start inside the Loaders

Size & weight

- Edge-to-edge width: 3.23 in (82 mm)
- Corner-to-corner: 3.84 in (97.6 mm)
- Mass: 40 ± 4 g

Diagrams

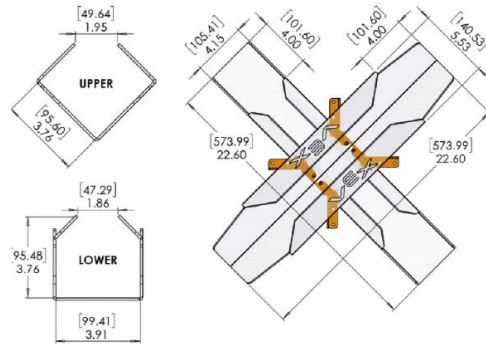


Center Goal

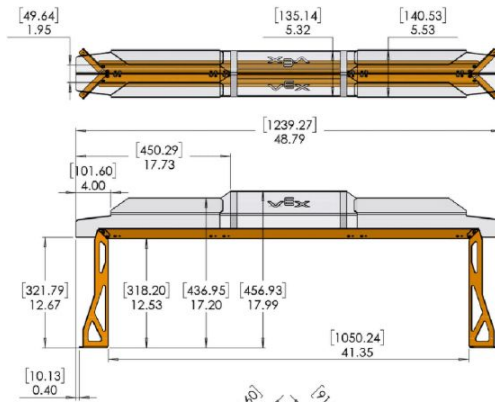
- 2 total (Upper Center Goal + Lower Center Goal)
- Each holds multiple Blocks (no fixed capacity; Blocks sit in a small trough-like rail)
- Located at the center of the field

Diagrams

Center Goal Specifications



Long Goal Specifications

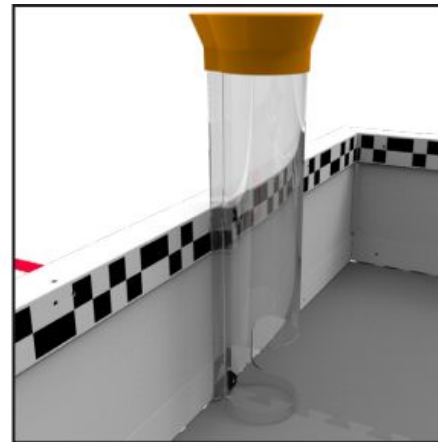


Long Goals

- 2 total
- Overall length: about 48.8 in (15 balls can fit)
- Control Zone: Area between the white tape lines.

Match Loaders

- 4 total (2 red, 2 blue)
- Each Loader can hold Match Load Blocks fed by the Drive Team
- Located beside each Alliance Station on the field wall
- Height: about 21.3 in
- Exit opening: roughly 3.9 in wide, (sized for one Block)



Parking Barrier

- 2 total Park Zones
- 1 red, 1 blue
- Flat, open area — no bars or tiers.
- 3 Sides
- Used for endgame Parking scoring



Goal

Prior to developing game strategies, we must understand important rules that may affect our match plans. Important sections below are highlighted.

SG1

Outlining the legal robot starting position

1. Robots must contact the barrier that defines their Alliance's Park Zone.
2. Robots may start with one Preload Block.
3. Robots must be stationary and may not begin in contact with Goals or Loaders.
4. Robots must not contact any Blocks other than their preload

SG2

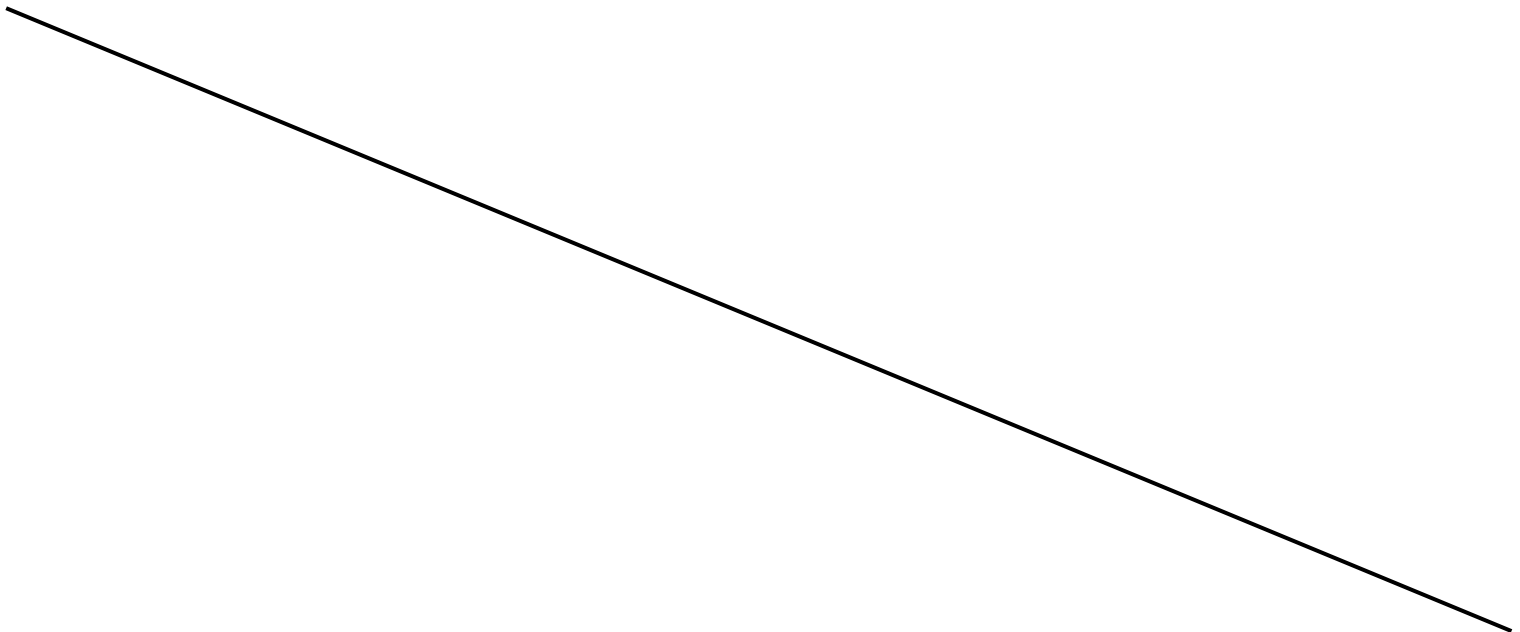
Guidelines for horizontal expansion

1. Robots may not exceed the standard 18" × 18" × 18" limit at any time.
2. Expansion beyond this size is not allowed (except for VEX U, which doesn't apply here).
3. The Robot can never be larger than 22" wide or 22" long (must always be able to fit within a 22"x 22" square horizontal footprint).

SG3

Allowance of vertical expansion

1. Once the Match begins, Robots are allowed to expand vertically.
2. Maximum height: No part of the Robot may ever be higher than 22" at any time during the Match.
3. Another way to think about it: throughout the Match, the Robot must always be able to fit inside a 22" × 22" × 22" cube.



SG4

Teams may not remove Blocks from the Field on purpose.

1. Any Block that leaves the Field (even by accident) is given to the same-color Drive Team to use as a Match Load.
2. After a team has three Matches with SG4 Violations, every future SG4 Violation becomes a Major.
3. Removing three or more Blocks in one Match is an automatic Major.

SG5

Outline of preloads

1. The preload must be contacting one robot of the same alliance color as the preload
2. The preload must be contacting the same robot as another preload
3. The preload must not be contacting or encircling a stake or any other scoring objects

SG6

Rules for Handling Blocks

1. Robots may carry, push, or plow unlimited Blocks.
2. Expansion limits still apply — all Blocks being held count toward your Robot's total footprint.
3. Blocks cannot cause you to break SG2 (horizontal expansion) or SG3 (vertical height).

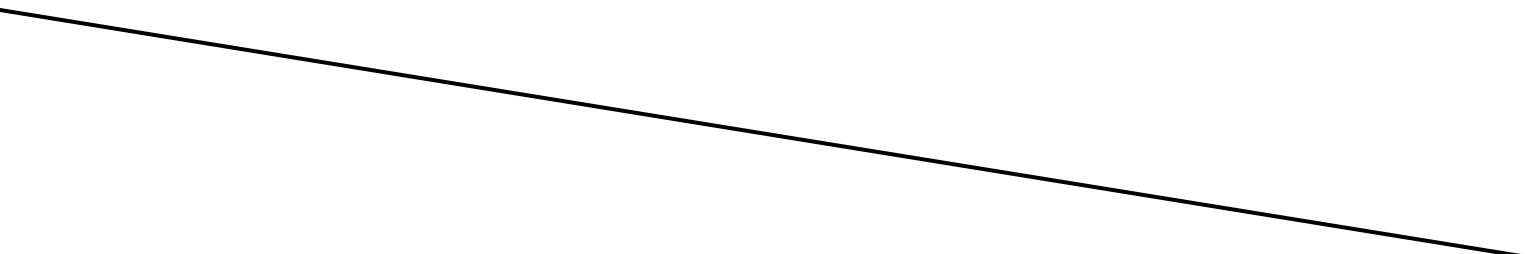
SG7

Rule disallowing crossing the autonomous line

1. During the autonomous period, robots may not contact foam tiles, scoring objects, or field elements which are on the opposing alliance's side of the autonomous line

SG8

Outline of robot interactions near the autonomous line

1. If both Robots engage with the same Block at the Autonomous Line, normal gameplay judgment is used.
 2. Incidental Violations of <SG7> will not be penalized and will not automatically award the Autonomous Bonus to the opponent as described in <GG13>.
 3. Intentional or abusive contact still triggers GG13 / GG14 / GG15 Violations.
- 

SG9

Rule for Match Loads

1. Drive Team Members may add Match Load Blocks into either Loader next to their Alliance Station.
2. Match Loads can only be introduced during Driver Control.
3. Illegal Match Loads may become Match Affecting depending on how many happen and the score difference.

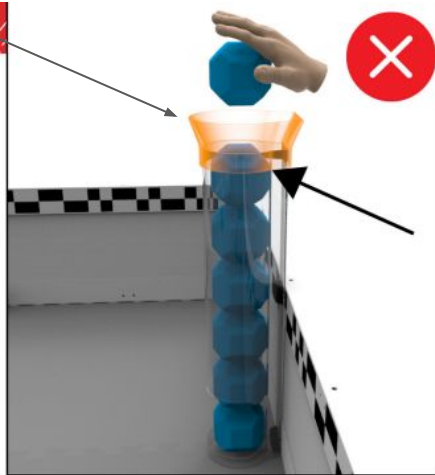


Figure SG9-1: A Block is partially within the orange portion of the Loader, so it would not be legal to add a Block to this Loader.

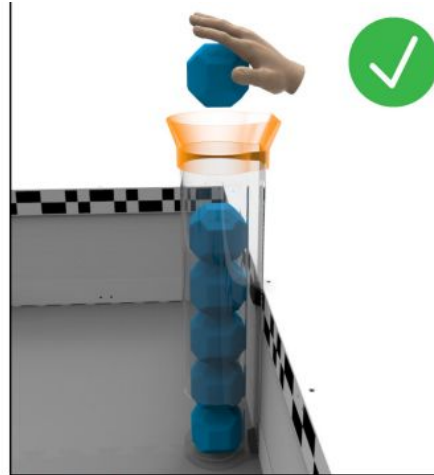


Figure SG9-2: There are no Blocks within the orange portion of the Loader, so it is legal to add a Block to this Loader.

SG10

Protection of alliance wall stakes

1. Robots may not touch Blocks that are inside enclosed sections of Long Goals.
2. Reaching into the top of a Goal is allowed but risky — you won't get "benefit of the doubt" in collisions.
3. Goalkeeping (blocking a Goal by sitting in the opening) is not allowed and results in a Violation.

SG11

Protected Park Zones (Endgame)

1. In the last 20 seconds, Robots may not touch the opponent's Park Zone.
2. Robots may not touch opponent Robots that are partially or fully inside their Park Zone.
3. Intentional or match-affecting actions become Major Violations.
4. Indirect contact can still violate SG11.

Goal

We will develop strategies for this year’s game to prepare for and understand the best solution implementations going forward.

Scoring Prioritization

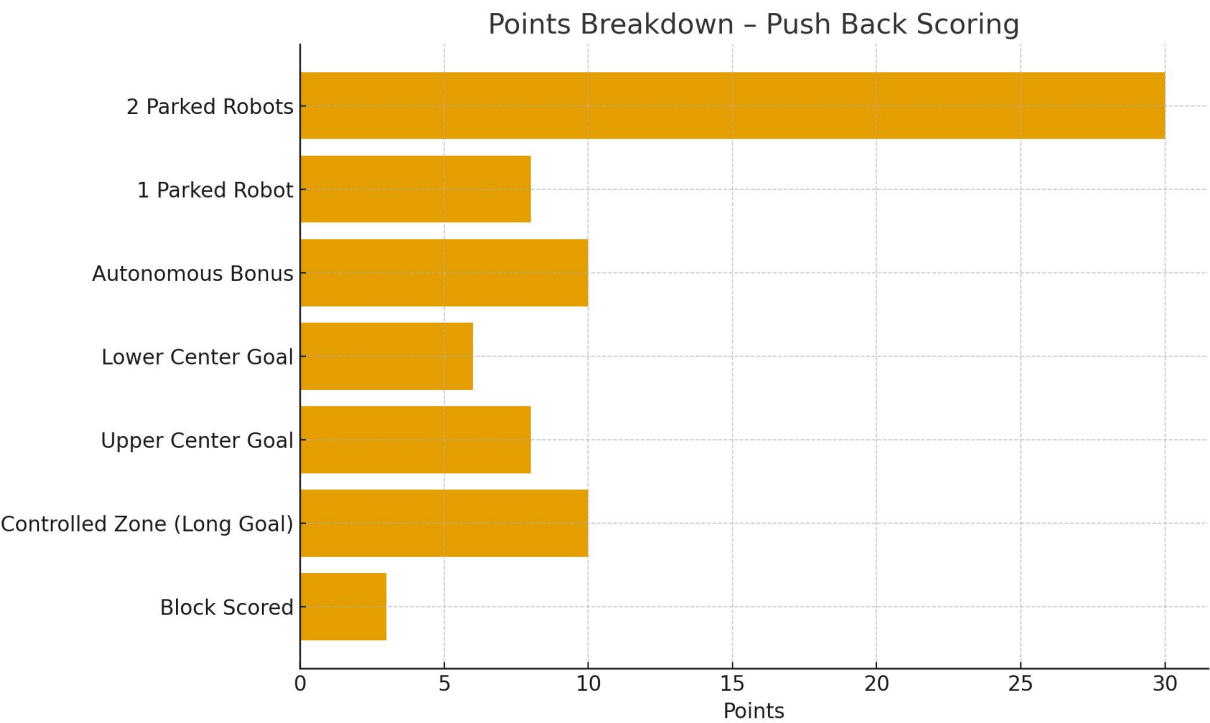
Scoring method points:

- 3 points for each Block scored
- 10 points for each Controlled Zone in a Long Goal
- 8 points for a Controlled Center Goal – Upper
- 6 points for a Controlled Center Goal – Lower
- 10 points for winning the Autonomous Bonus
- 8 points for 1 Parked Robot
- 30 points for 2 Parked Robots

Observations:

- Controlling zones in a Long Goal offers some of the highest single scoring values and becomes a major strategic focus.
- Parking is a significant endgame swing: two robots parked can shift a close match by 30 points.
- The control zones for both center goals and long goals are worth more than scoring individual balls.

Autonomous Bonus	10 Points
Each Block Scored	3 Points
Each Controlled Zone in a Long Goal	10 Points
Controlled Center Goal - Upper	8 Points
Controlled Center Goal - Lower	6 Points
1 Parked Alliance Robot	8 Points
2 Parked Alliance Robots	30 Points



Offensive Strategies

Score Blocks in Long Goals

Pros	Cons
- Easy to cycle quickly - Long Goals hold many Blocks, making them great for fast dump/push bots - Low precision required compared to Center Goals - Strong early scoring to build point advantage (control zones).	- Opponents can descore or take control of zones - Controlling zones is harder as it requires more blocks - Requires a more complex design as the long goals are higher off the ground

Score Blocks in Center Goals (Upper & Lower)

Pros	Cons
- Center Goals are contestable, so winning them matters strategically - Scoring on center goals can pressure opponents and draw them into the middle	- Requires more precision and mechanism control

Prioritising Control Zones

Pros	Cons
- 10 points per controlled zone – extremely high-value scoring - Can swing the end of a Match even if Block count is equal	- Hard to do under strong defense - Opponents can fight for zone control by placing their color last

Defensive Strategies

Push Opponent's Blocks Out of Scoring Position

Pros	Cons
- Lowers opponent scoring without needing precision - A Simple bot can perform this effectively	- Must avoid SG4 Violations (Blocks leaving the Field) - Can unintentionally improve opponent positioning if done poorly

Contest Center Goals (Blocking & Interference)

Pros	Cons
- Easy strategy to execute - Prevents opponents from obtaining control zones	- Risk of entanglement or SG10 violations - May sacrifice your own scoring opportunities

Deny Long Goal Zone Control

Pros	Cons
- Each controlled zone is worth 10 points, so denying these is high-impact	- Robots can't be everywhere at once - zones are long and spread out

Body-Blocking Park Zones

Pros	Cons
- Prevents opponents from parking for 8 or 30 points - Disrupts endgame strategy	- Must retreat before the last 20 seconds to avoid SG11 penalties - Robots with strong drivetrains may push through anyway

Design 1: Early Season Robot

Goal

We will identify design objectives for our drivebase to ensure we spot and prevent deficits in efficiency across a variety of parameters we will outline.

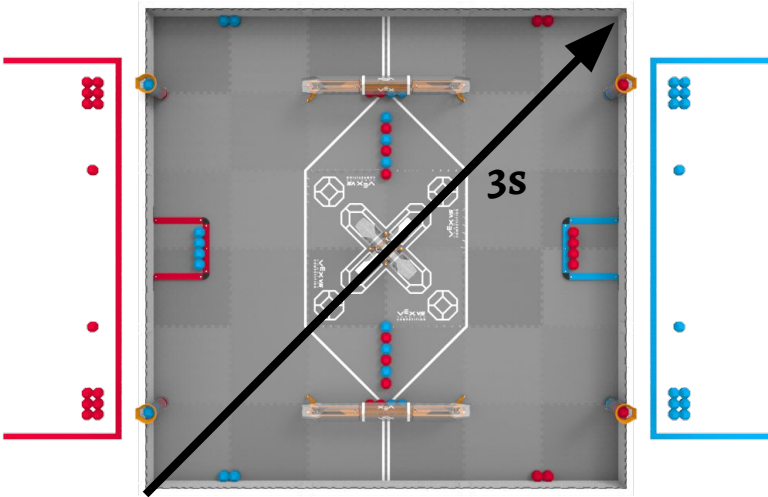
Problem Statement

We need a drivebase that can support the rest of the robot and quickly traverse the field to interact with game objects.

Solution Goals

The robot must be able to...

- 1. Traverse the field diagonally in 3 seconds
- 2. Drive with 20 sof additional weight (Holding balls)
- 3. Complete a lap around the field in 9 seconds



A diagram of Solution Goal 1

Solution Constraints

- 1. Must use a maximum of 66 watts as we want a 2 motor intake
- 2. Must fit within an 18” by 18” square
- 3. Must use only legal VEX parts

Goal

We will outline the different subsystems of our robot before we start designing each one individually. This way we have a clear idea of what we need to build and can refer back to this page throughout the season.

Initial Impressions

After watching the Push Back game reveal video, we sat down as a team and discussed what our robot would need to do. The main actions are: collecting blocks from the field and from match loaders, scoring them into the Long Goals and Center Goals, and parking at the end of the match. Based on this, we came up with the following subsystems.

Robot Subsystems

After watching the Push Back game reveal video, we sat down as a team and discussed what our robot would need to do. The main actions are: collecting blocks from the field and from match loaders, scoring them into the Long Goals and Center Goals, and parking at the end of the match. Based on this, we came up with the following subsystems.

1. Drivetrain

Responsible for all movement. Carries the rest of the robot and needs to be fast enough to traverse the field quickly while being sturdy enough to take contact from other robots.

2. Intake

This is the biggest system on the robot. It handles everything from picking up blocks to actually scoring them. Since there's a lot going on, we decided to break it down further into subsections:

- Ball Movement - Whatever mechanism moves blocks up through the intake
- Match Loader - How we will take blocks from the match loaders on the field wall.
- Long Goal Scoring mechanism- How we get blocks into the Long Goals
- Mid Goal Scoring mechanism- How we score on the Center Goals (upper and lower)

3. Tracking System

This includes aspects such as an odometry tracking wheel setup as well as distance sensors which we will use to aid our robot during the autonomous period. Having more sensors/tracking will allow us to have a more consistent robot, which makes performance better and coding easier.

Why We Split It Into Subsystems

We decided to split the robot into subsystems because:

- It lets team members work on different parts at the same time, which speeds up the build process.
- If something breaks at a competition, we only need to take apart that one subsystem instead of the whole robot.
- It makes documentation easier. Each subsystem gets its own design cycle (identify problem, brainstorm, select, build, test).

Obstacles We Think We'll Face

1. Parking barrier - We're not sure yet how our drivetrain will handle crossing the barrier. Wheel placement, our odometry and ground clearance could be an issue.
2. Two different goal heights - Long Goals and Center Goals are at different heights, so our one intake/scoring system needs to be able to handle both.
3. Control zones - Control zones are worth 10 points each and are determined by which colour has more balls in that section of the goal. This means our intake needs to be able to score into the control zones or
4. Match loader, our intake needs to be able to consistently take balls from the match loader
5. Consistency - We need our robot to perform the same way every match. If our intake jams sometimes or our autonomous works only half the time, it doesn't matter how good the design is on paper. Staying consistent across multiple matches in a row is one of the hardest things to do

Goal

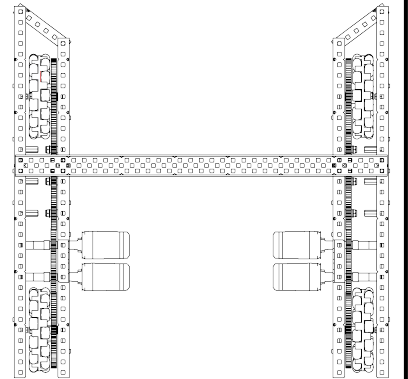
We will compare and contrast different types of drivebases to select the optimal one and plan our drivebase design implementation.

Possible Solutions

Omnidirectional Tank Drivetrain

- Four or more omnidirectional wheels set up in a tank-drive formation
- Can use 2, 4, 6, or 8 motors

*Omnidirectional wheels



Pros

- 100% efficiency
- Simple to program
- Takes up minimal space
- Minimal turning scrub

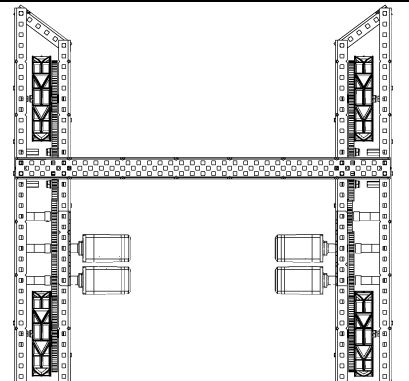
Cons

- Bilateral movement

Traction Tank Drivetrain

- Four or more traction wheels set up in a tank-drive formation
- Can use 2, 4, 6, or 8 motors

*Traction wheels



Pros

- 100% efficiency
- Simple to program
- Takes up minimal space
- Cannot be pushed

Cons

- Severe turning scrub
- Bilateral movement

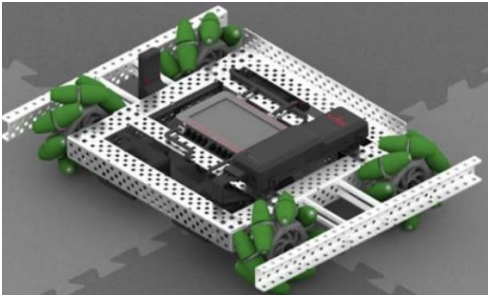
We will compare and contrast different types of drivebases to select the optimal one and plan our drivebase design implementation.

Possible Solutions pt 2

Mecanum Drive Base

- Uses 4 mecanum wheels to drive forward/back + strafe (side-to-side)
- Typically uses 4 motors (one per wheel)
- Can also use 6 or 8 motors for extra power

*Mecanum wheels



Pros

- Strafing helps alignment for pushing Blocks into Goals and positioning for zone control
- Easier to slide around traffic and adjust angle without turning

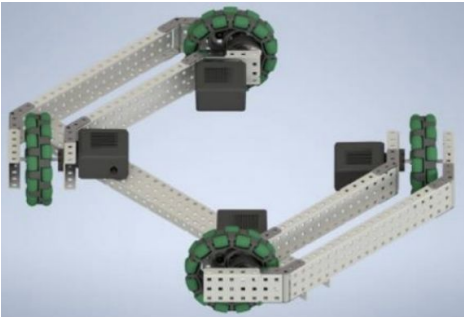
Cons

- Traction can be bad compared to conventional drivebases
- Difficult to build and tune

Holonomic or X-drive

- Uses 4 omni wheels mounted in an “X” pattern (each wheel angled ~45°)
- Can also use 6 or 8 motors for more power, but 4 is most common

*Omni Wheels



Pros

- Great for dodging traffic and slipping past defenders
- Can strafe and rotate smoothly to align with goals

Cons

- Difficult to Building and program
- Lower pushing power than traction/tank drives (easy to get bullied in contact)
-

Goal

We will compare and contrast different types of wheels and their sizes to select the optimal one and plan our drivebase design implementation.

Possible Solutions

4" Wheel - Wheel size with the largest diameter	*4" Traction Wheel 
Pros	Cons
- Fastest wheel size with the same relative rpm - Can traverse over objects up to ~3" large	- Very imprecise in programming because one rotation equates to a longer distance

3.25" Wheel - Wheel size with the second largest diameter	*3.25" Omnidirectional Wheel 
Pros	Cons
- 2nd fastest wheel size with the same rpm - Can traverse over objects up to ~2.4" tall - 2nd most precise wheel in programming	- 2nd slowest wheel with the same relative rpm - 2nd most imprecise wheel in programming

2.75” Wheel

- Wheel size with the smallest diameter

*2.75”
Omnidirectional
Wheel



Pros

- Most accurate wheel in programming

Cons

- Slowest wheel with the same relative rpm
- Cannot traverse large objects

Goal

We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the drivebase design.

Decision Matrix - Drivebase Type

Rubric	Omnidirectional Tank-Drive	Traction Tank-Drive	Holonomic Drive	H-Drive
Efficiency (10)	10	10	5	8
Simplicity (10)	10	10	5	8
Compactness (15)	15	15	5	10
Mobility (15)	10	5	15	13
Total (50)	45	40	30	39

Decision Matrix - Wheel Size

Rubric	4" Wheel	3.25" Wheel	2.75" Wheel
Autonomous Precision (15)	5	10	15
Speed (5)	5	4	1
Mobility (20)	20	15	10
Compactness (10)	3	10	10
Total (45)	33	39	36

Conclusion

Using our decision matrices, we have decided to continue with an omnidirectional tank-drive with 3.25" wheels.

Note

Although we normally continue with “Brainstorm Solutions” following the engineering design process order, gear ratios specifically have only one solution as we aim for being as close to a certain speed as possible (defined below). Therefore, we are proceeding to the “Select and Plan” phase, detailing what tool and what criteria we base our gear ratio on.

Goal

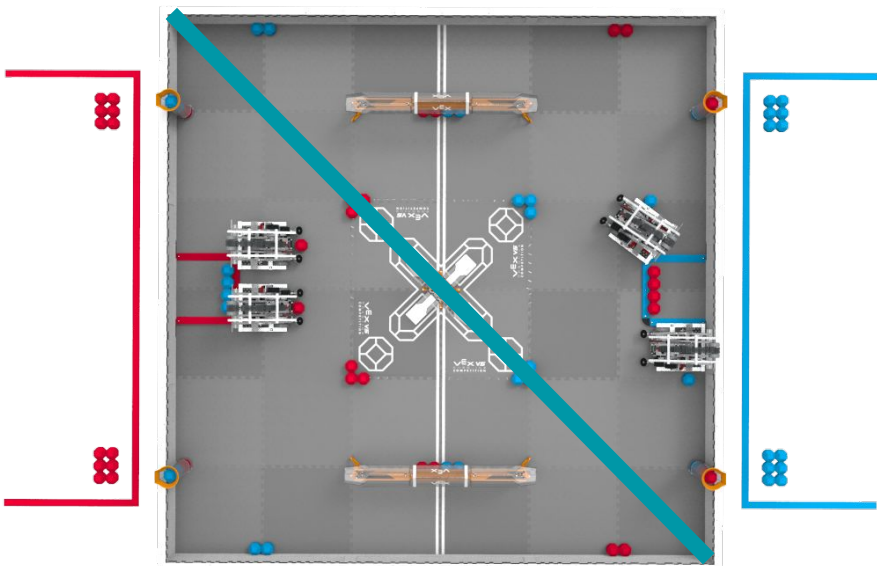
Before implementing the omnidirectional tank drive with 3.25” wheels, we must decide on an RPM that fulfills our solution goals for the drivebase.

Solution Goals (from pg. 22)

The robot must be able to...

- 1. Traverse the field diagonally in 3 seconds
- 2. Drive with 20 lbs of additional weight
- 3. Complete a lap around the field in 9 seconds

Referencing our 3 solution goals, we can find the linear speed and the corresponding gear ratio that aligns with our requirements



16.97ft

Using the pythagorean theorem, we have calculated the diagonal of the field to be ~16.97 ft. Based on this and Solution Goal #1, our approximate linear

speed is 6.38 ft/s. The closest RPM to a linear speed of 6.38 ft/s is 450 RPM with a gear ratio of 3:4 on a 600 RPM motor cartridge using a 36 tooth driving gear and a 48 tooth driven gear (*reference VEX Gear Ratios spreadsheet). This ratio also balances speed and torque, hopefully fulfilling both solution goals 2 and 3 as well.

Field diagonal = $\sqrt{(12^2 \text{ ft} + 12^2 \text{ ft})} = 16.97 \text{ ft}$
Linear speed = $16.97 \text{ ft} / 3 \text{ s} = 5.65 \text{ ft / s}$

*https://docs.google.com/spreadsheets/d/1RSOLv3tnpiCgFyHboQayxK-42r9MgVRD_4QQmeFM618/edit?usp=sharing

VEX Gear Ratios

Output RPM	Linear Speed (ft / s)	Gear Ratio
450	6.38	3:4

Goal

This section was done for brainstorming and gathering data. Since it's the beginning of the season we have no idea what's good and bad, hence these concepts were created to visualise our ideas.

Selected Solution

- 6 11W motors
- An omnidirectional tank drive with 2 traction wheels centered on each side
- 450 RPM output
- 3:4 gear ratio on a 600 RPM cartridge
- 36 tooth driving gears and 48 tooth driven gears
- 3.25" wheel size for all wheels

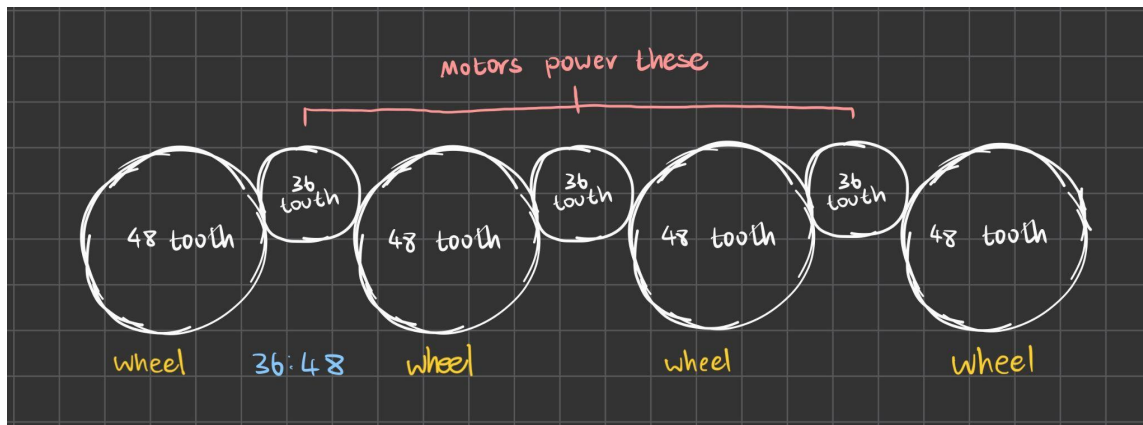
2 Goal

After analyzing the wheels sizes and drivetrain type, we have decided to finalise it in CAD, specifically on fusion 360.

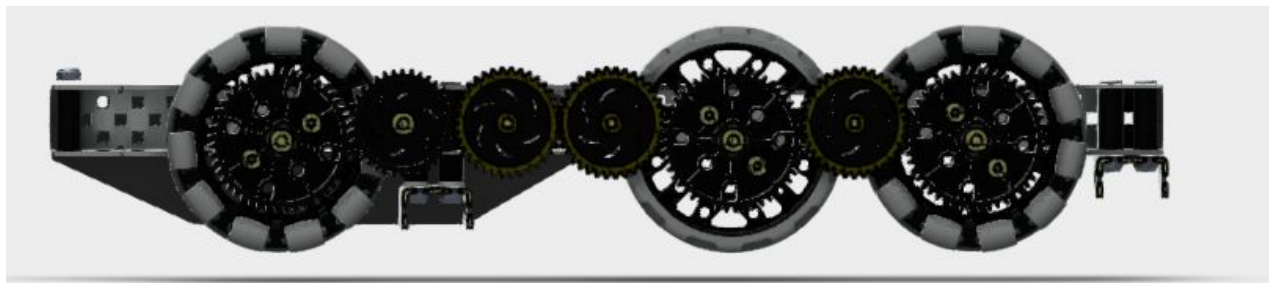
Selected Solution

- 6 11W motors
- An omnidirectional tank drive with 2 traction wheels centered on each side
- 450 RPM output
- 3:4 gear ratio on a 600 RPM cartridge
- 36 tooth driving gears and 48 tooth driven gears
- 3.25" wheel size for all wheels

CAD Plan



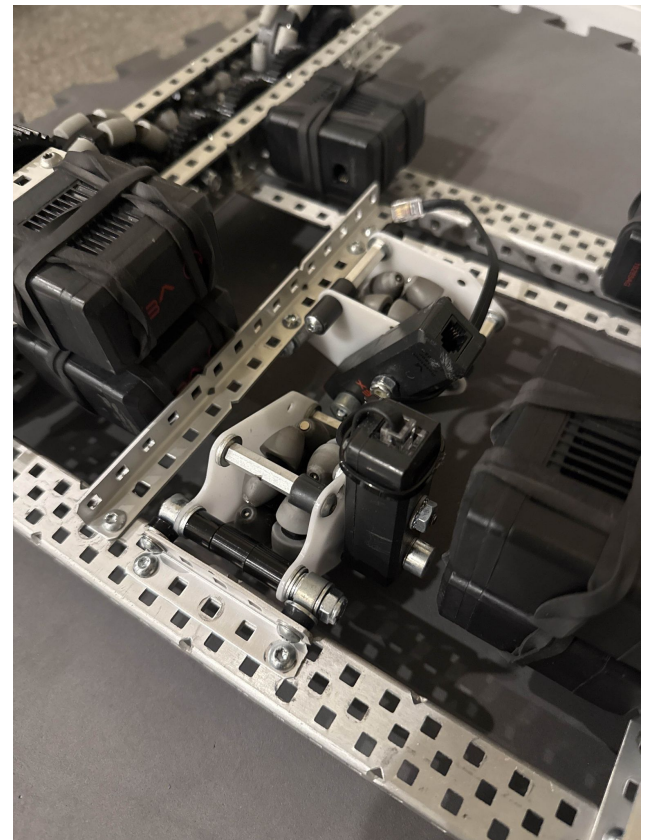
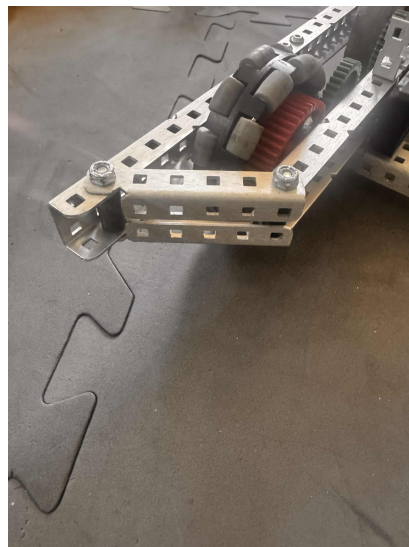
Final Cad Design for 1 Side



CAD Notes: We determined that the presence of the parking barrier required a drivetrain adjustment. To improve our ability to cross the barrier, we removed one wheel and repositioned it farther back on the chassis, reducing interference and making barrier traversal easier.

Assembly Process

1. Place bearings in proper positions for all 4 main drive c-channels, leaving them disconnected
2. Screw motors into the correct holes
3. Place 4 inch axles in the proper positions for powered axles
4. Complete powered with appropriate spacing, gears, and collars
5. Place long screws and keps nuts in the proper positions for screw joints
6. Complete screw joints with appropriate spacing, gears, wheels, nuts, and standoffs
7. Connect the two inner drive channels with the back crossbar
8. Attach the two outer drive c channels with the back crossbar
9. Add the Odometry pods on



Goal

We will identify design objectives for our intake to ensure we spot and prevent deficits in efficiency across a variety of parameters we will outline.

Problem Statement

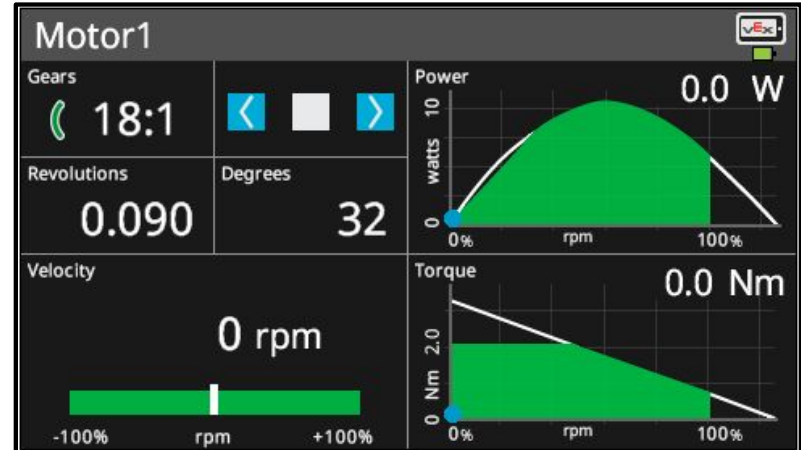
We need an intake that can collect blocks from the field and match loaders, transport them through the robot, and score them into both the Long Goals and Center Goals without interfering with other robot functions.

Solution Goals

1. Score on both the Long Goals and Center Goals (two different heights)
2. The power draw (friction) in watts should be under 3W when running with no load
3. Stay consistent, and be able to store at least 5 balls

Solution Constraints

1. Must use a maximum of 22 watts
2. Must use only legal VEX parts



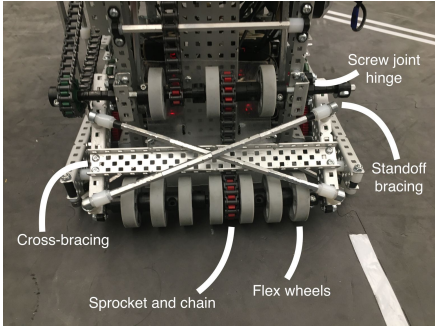
The VEX V5 robot brain can be used to check the wattage of running motors

Goal

We will compare and contrast different types of intakes to select the optimal one and plan our intake design implementation.

Possible Solutions

Claw Shaped Flex Wheel Intake - Two sets of flex wheels aligned horizontally with the ring forming a claw shape	Citation: Team 1690 (FRC) 
Pros	Cons
- Rapid object collection - Able to transport and remove rings easily	- Harder to drive - Greater building complexity - Can have trouble intaking a round object (ring) from the side

Vertical Flex Wheel Intake, similar to full volume - Multiple stages of flex wheels aligned vertically along a tray - Motorized and run on sprockets and chain	Citation: Team 2570R 
Pros	Cons
- Can collect balls from any angle - Easy to drive	- Takes up a lot of robot space - Have to manage tension - Difficult to tune for the most efficiency

Claw

- A motorized or pneumatic activated set of two levers which open and close on a shared hinge



Pros

Cons

- | | |
|---|--|
| <ul style="list-style-type: none">- Simple build implementation | <ul style="list-style-type: none">- Hard to drive- Easy to lose hold of rings- Difficult to move rings to other parts of the robot- Inefficient and slow- |
|---|--|

Goal

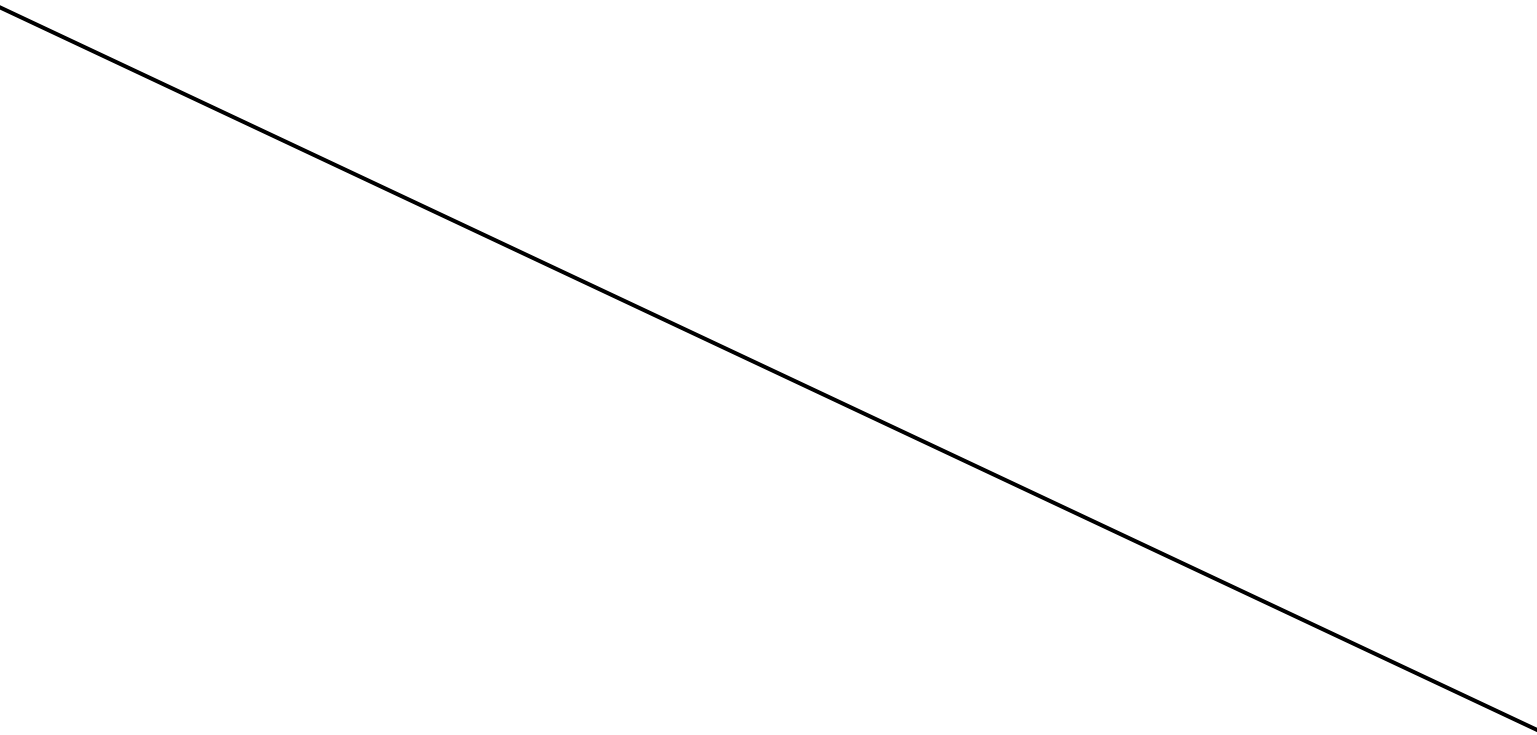
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Rubric	Claw Shaped Flex Wheel Intake	Vertical Flex Wheel Intake	Claw
Ease of Driving (10)	6	9	4
Speed of Intaking (10)	7	7	3
Build Complexity (10)	6	7	8
Total (30)	19	23	15

Conclusion

Using our decision matrix, we have decided to continue with a vertical flex wheel intake. This design can also hold a lot of balls which is important when scoring on either the mid goals or the long goals.



Pseudocode

1. INTAKE CONTROL:

If button L2 pressed $\Rightarrow$ Spit all balls out

If button R1 pressed $\Rightarrow$ Run storage stage.

If Button R2 is pressed $\Rightarrow$ Release all balls. Run both motors

Code Blocks

```
// ===== REFINED INTAKE CONTROL ===== //
```

```
// Simplified intake logic - same functionality, cleaner code
```

```
void runIntake(int speed, IntakeMode mode) {
```

```
    // Middle piston override - takes priority when active
```

```
    if (middleState && speed != 0) {
```

```
        intakeMotor1.move(voltage: -50);
```

```
        intakeMotor2.move(voltage: 100);
```

```
        return;
```

```
    }
```

```
    if (middleState) { // Middle active but intake off
```

```
        intakeMotor1.move(voltage: 0);
```

```
        intakeMotor2.move(voltage: 0);
```

```
        return;
```

```
    }
```

```
    // BOTH mode - simple passthrough
```

```
    if (mode == IntakeMode::BOTH) {
```

```
        intakeMotor1.move(voltage: speed);
```

```
        intakeMotor2.move(voltage: speed);
```

```
        return;
```

```
    }
```

```
    // PORT3_ONLY mode
```

```
    if (speed > 0) { // Forward
```

```
        intakeMotor2.move(voltage: speed);
```

```
        intakeMotor1.move(voltage: shouldStopIntake() ? -10 : 50);
```

```
    } else if (speed < 0) { // Reverse
```

```
        intakeMotor2.move(voltage: speed);
```

```
        intakeMotor1.move(voltage: 0);
```

```
    } else { // Stop
```

```
        intakeMotor1.move(voltage: 0);
```

```
        intakeMotor2.move(voltage: 0);
```

```
    }
```

```
}
```

Explanation

1. When either r1 or r2 is pressed, the motors run differently. To allow the robot to hold balls, when r1 is pressed only the storage stage motor is run while the scoring motor is disabled
2. Once r2 is pressed, both motors run simultaneously to release and score all the balls
3. To outtake balls to the bottom goal, the driver needs to press L2

Goal

We will identify various ways to score balls onto all three types of goals on the field. Being able to score on every goal gives us more flexibility in matches and makes it harder for opponents to shut us down.

Problem Statement

We need a robot intake system which can efficiently score on the Long Goals, the Upper Center Goal, and the Lower Center Goal. Each goal sits at a different height, so we need a way to adjust where the balls are output. Scoring on all three is important because:

Long Goal control zones = 10 pts each
Upper Center Goal control zone = 8 points
Lower Center Goal control zone is = 6 points

Being limited to just one or two goal types means we leave points on the table and become predictable for opponents.

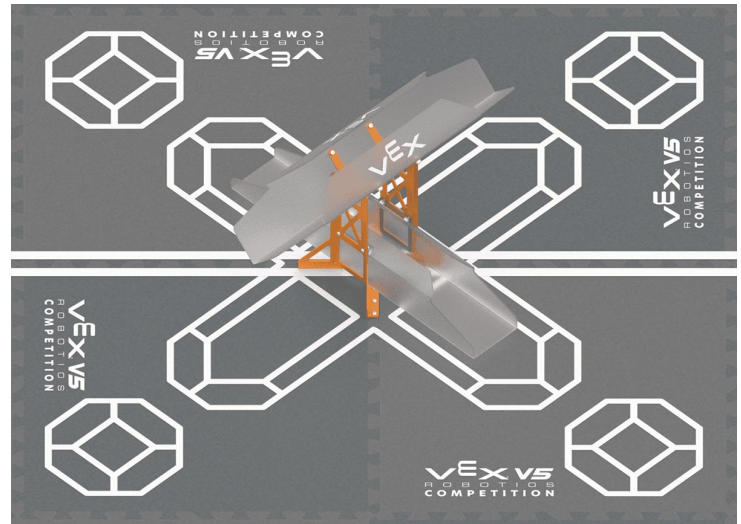


Image of a the center goals. The middle center goal is approximately 118.41mm lower than the long goal

Solution Goals

Solution Goals

1. The robot must be able to...
2. Score on the Long Goals efficiently
3. Score on the Upper Center Goal
4. Score on the Lower Center Goal
5. Switch between goal targets without needing to rebuild or reconfigure mid-match

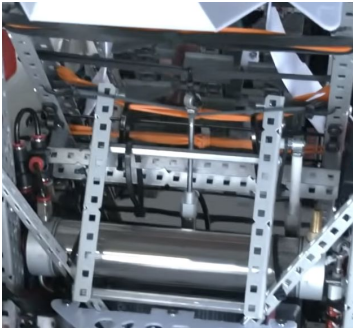
Solution Constraints

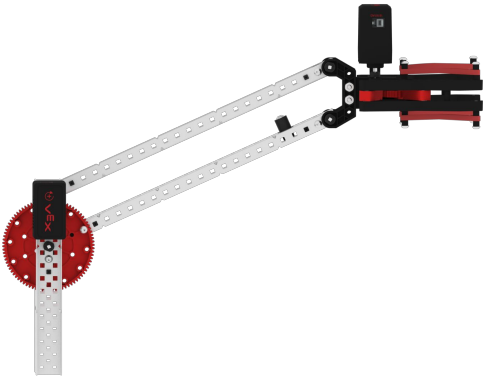
1. Must use a maximum of 22 watts
2. Must use only legal VEX parts

Goal

We will compare and contrast different mechanisms which can help us to score onto the middle center goal.

Possible Solutions

Ruiguan Piston Design - A lane for balls to travel up and score on the long goals. A piston can on the middle can pull down to allow balls to score on the middle goals.	 Credits: 9123x
Pros	Cons
- Simple to build and design - Lightweight - Efficient. No complex mechanisms	- Scoring may be slower - Reduced scoring precision

Lift scoring mechanism - A lift which we can use to score at all 3 levels (long goal, high middle goal, low middle goal) - We would have intake on this lift and lift to score.	
Pros	Cons
- Efficient - Reuses same mechanism/intake for all goals (consistency)	- Could be slow, not strong due to weight - Expansion limits

Goal

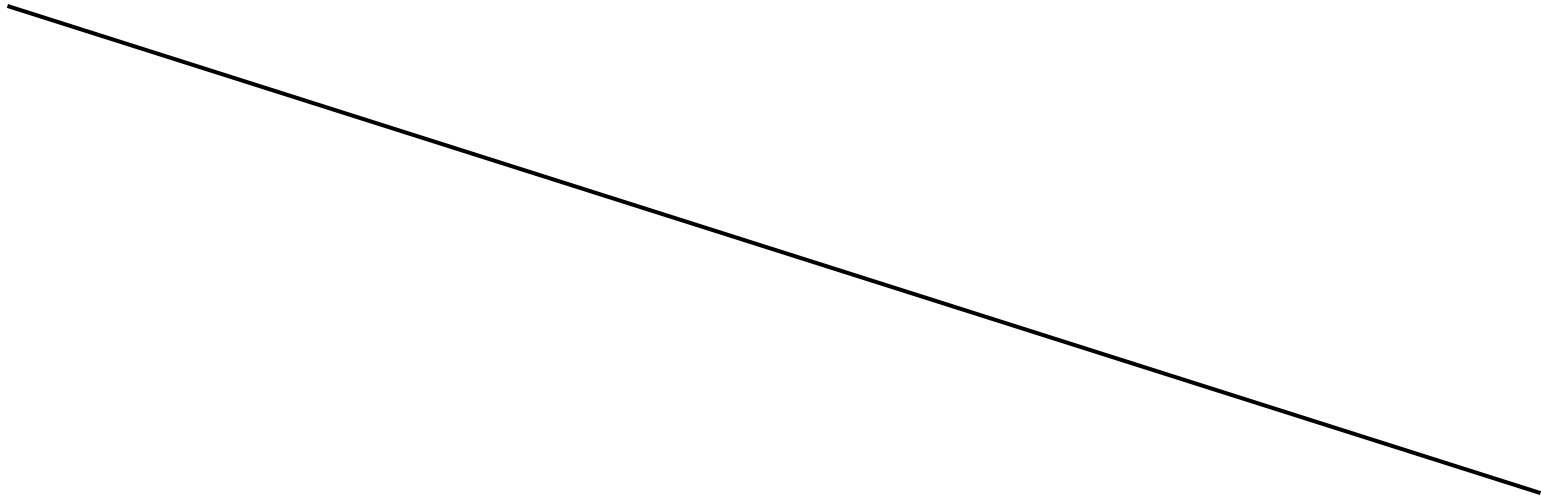
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the lift design.

Decision Matrix

Rubric	Lift Design	Ruiguan Piston
Height Reach (10)	5	7
Speed (10)	9	6
Stability (10)	8	8
Power Efficiency (10)	9	6
Weight Capacity (10)	4	9
Total (50)	33	36

Conclusion

Using our decision matrix, we have decided to go with the lift design. The main reason is scoring flexibility. The Ruiguan piston design is faster and more power efficient, but it can really only switch between two heights. Since we want to be able to score on all three goals (Long Goal, Upper Center, Lower Center), the lift gives us that ability. Being able to hit any goal in a match means we can adapt our strategy depending on what the opponents are doing, which is worth the trade-off in speed and power.



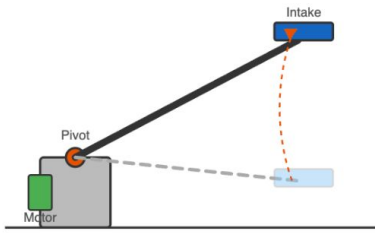
Goal

We will compare and contrast different types of bar lifts to score on the three goals.

Possible Solutions

1-Bar Lift (Single Arm)

A single arm that pivots from one point, like a seesaw. The motor sits at the base and rotates the arm up and down. The intake sits at the end of the arm. Simple as it gets.



PROS

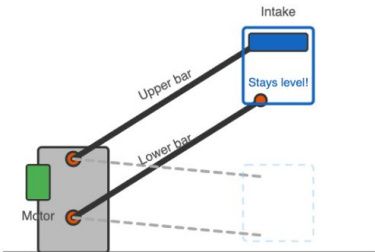
- Very simple to build
- Lightweight, minimal parts
- Fast movement between positions
- Power efficient, only 1 motor needed
- Easy to program (just set motor angle)

CONS

- The platform tilts as the arm raises, so the intake angle changes at different heights
- Limited reach compared to multi-bar lifts
- Less stable at full extension
- Heavier loads at the end cause more strain on the motor

4-Bar Lift

Two parallel bars of equal length connected at both ends. As the bars rotate, the platform at the end stays perfectly level throughout the entire range of motion. This is the key advantage over a 1-bar.



PROS

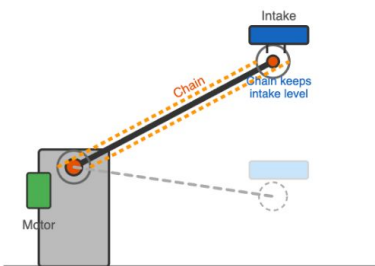
- Platform stays level at all heights (great for keeping intake angle consistent)
- Stable and predictable motion
- Good weight capacity since load is shared across two bars
- Smooth, controlled movement

CONS

- Heavier than a 1-bar due to extra bars and pivots
- Slower because it's moving more mass
- More complex to build and align properly
- Takes up more space on the robot
- Uses more power than a simple 1-bar

Chain Bar Lift

A single arm with a chain and sprocket system running along it. The chain connects a sprocket at the base to one at the end of the arm. As the arm rotates up, the chain keeps the platform level, similar to a 4-bar but using chain instead of a second bar.



PROS

- Platform stays level (like a 4-bar) thanks to the chain
- Lighter than a 4-bar since chain weighs less than a second bar
- Can reach higher than a 4-bar for the same footprint
- Compact design

CONS

- Chain can skip or break under heavy load
- Needs proper chain tension to work reliably
- Less rigid than a 4-bar, can wobble
- If the chain breaks mid-match, the whole lift stops working
- Harder to tune and maintain

All three are arm-style lifts that pivot from a single point. The difference is how they keep the platform stable at the end.

Goal

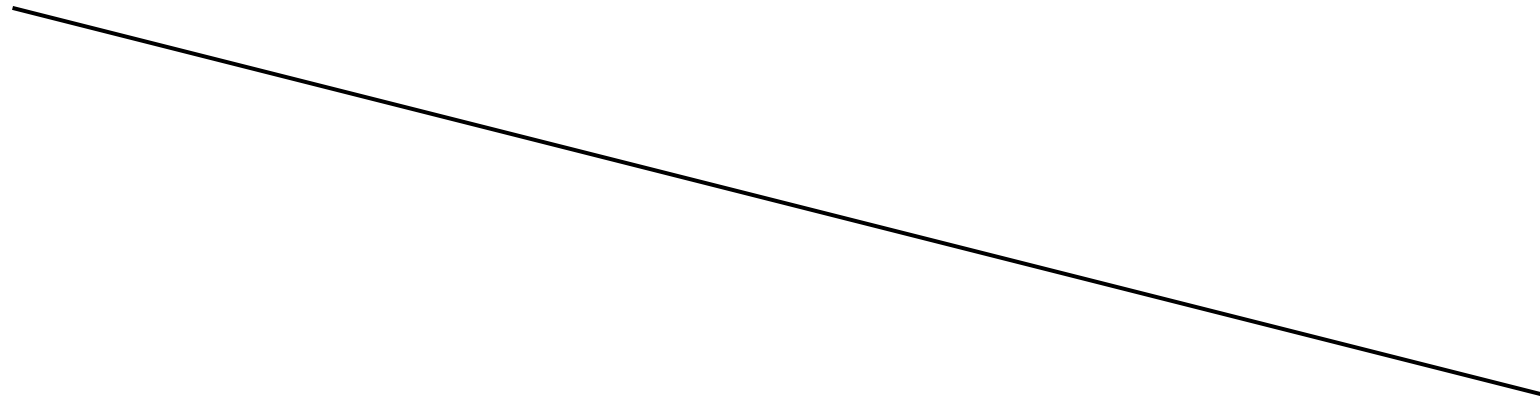
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the lift design.

Decision Matrix

Rubric	1-Bar Lift	4-Bar Lift	Chain Bar Lift
Height Reach (10)	5	7	9
Speed (10)	9	6	7
Stability (10)	5	10	5
Power Efficiency (10)	9	6	7
Reliability (10)	8	8	4
Total (50)	36	37	32

Conclusion

Using our decision matrix, we have decided to go with the lift design. The main reason is scoring flexibility. The Ruiguan piston design is faster and more power efficient, but it can really only switch between two heights. Since we want to be able to score on all three goals (Long Goal, Upper Center, Lower Center), the lift gives us that ability. Being able to hit any goal in a match means we can adapt our strategy depending on what the opponents are doing, which is worth the trade-off in speed and power.



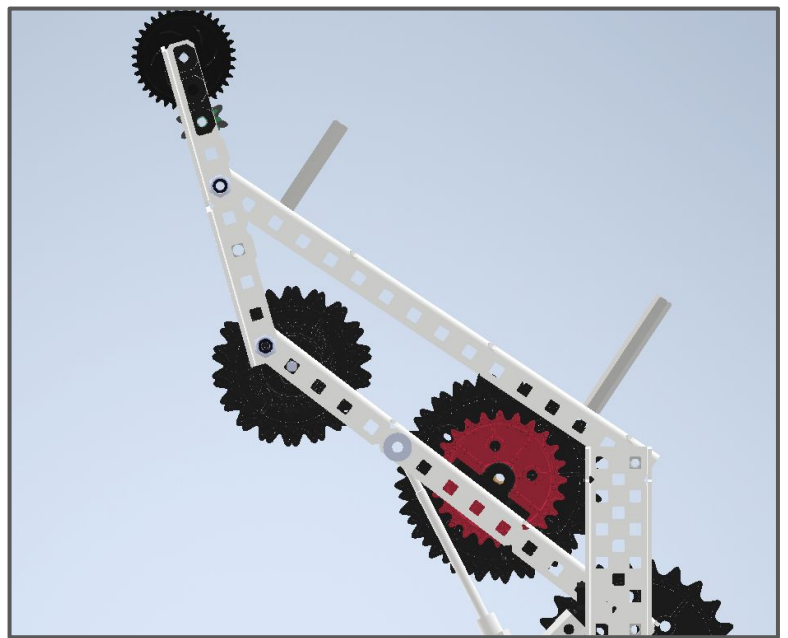
Goal

We will select our final drivebase design and utilize Autodesk Inventor Professional 2024 to construct a CAD model before building.

Selected Solution

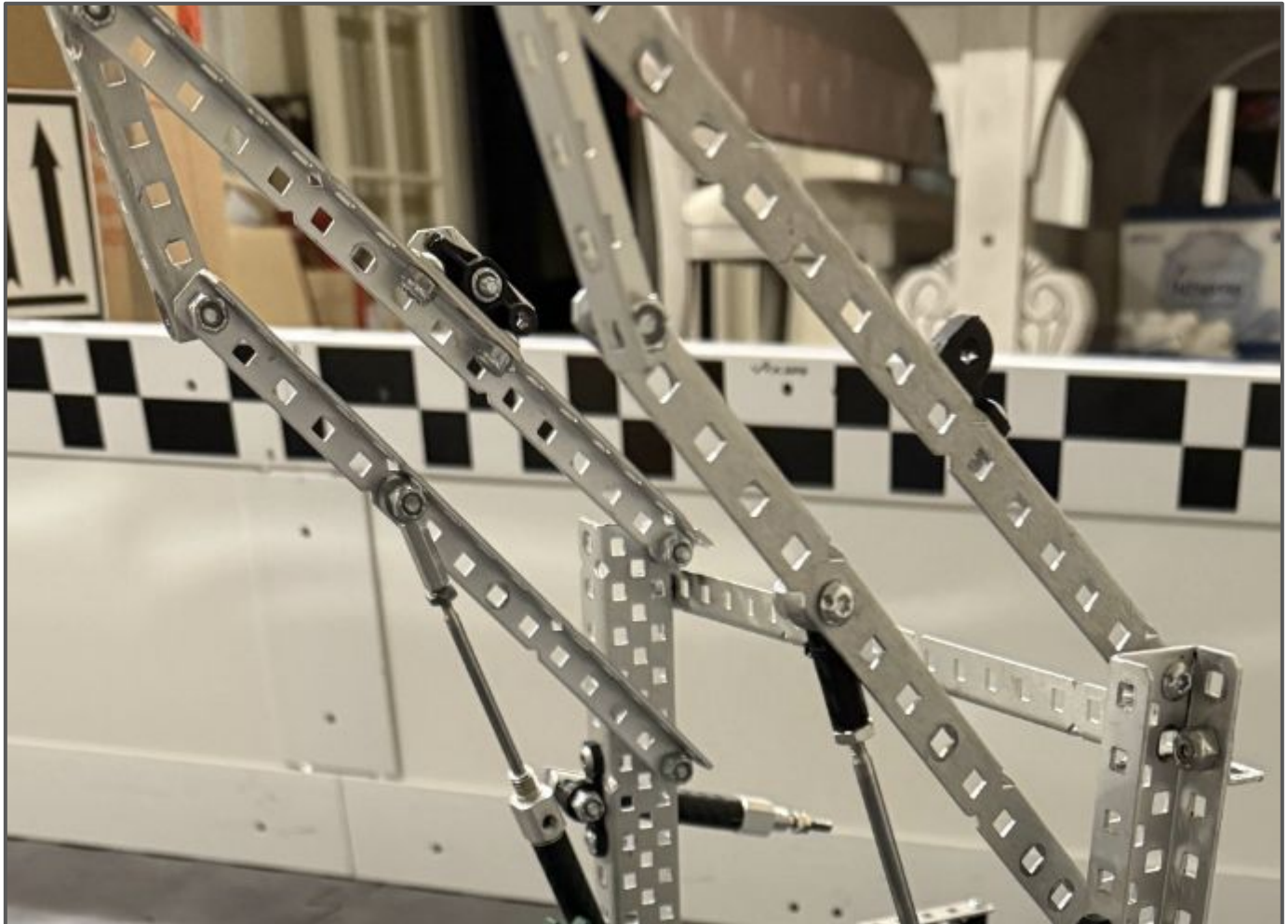
- 1 11W motor
- 1 bar lift
- 20 RPM output
- 1:5 gear ratio on a 100 RPM cartridge
- 12 tooth driving gear and 60 tooth driven gears
- 4 bar power transfer to ensure hs axle does not interfere with intake

CAD Plan



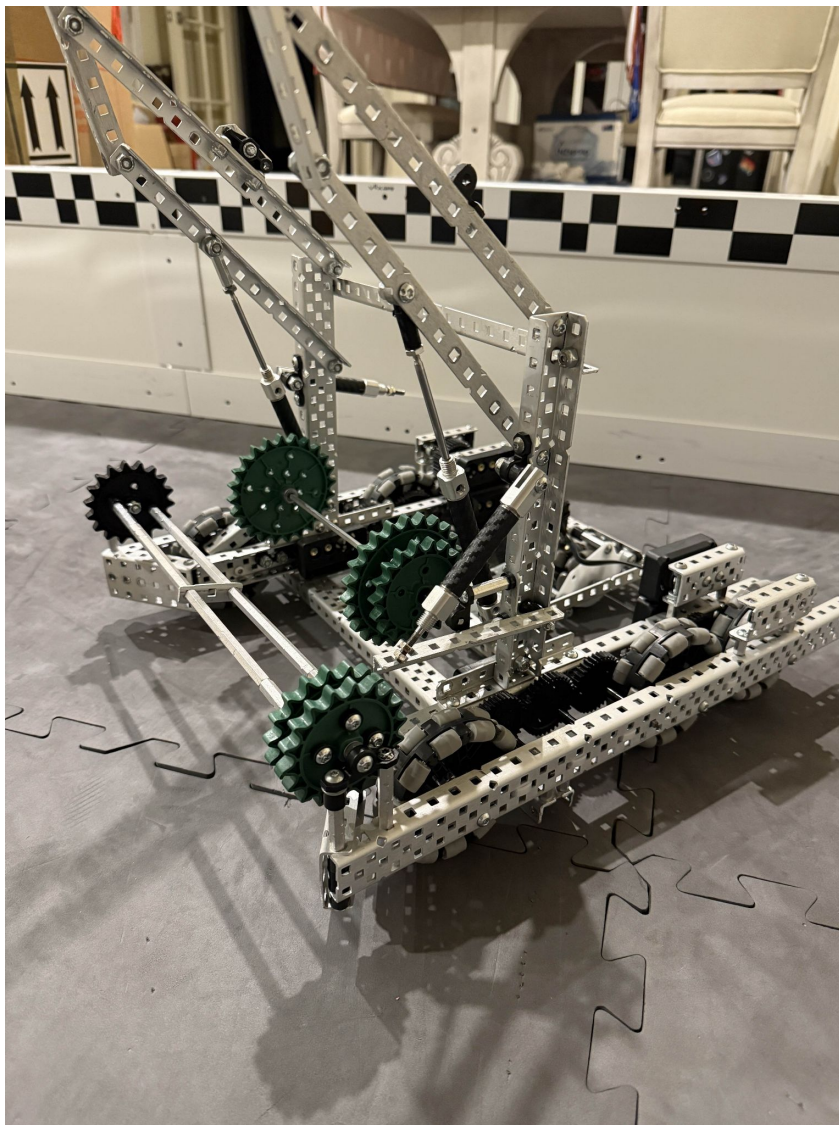
Steps

1. Mount lift towers
2. Attach lift motor to lift towers
3. Assemble the lift gearbox
4. Insert high strength axle
5. Assemble two lift linkages
6. Attach lift linkages to intake l-channels



(CONT.) Steps

1. Attach intake axle poly to l-channels
2. Attach intake backing to l-channels
3. Attach center crossbar to intake backing
4. Attach motor mount to intake backing
5. Mount middle axle
6. Build the hood
7. Attach the hood
8. Attach lower intake structure to intake poly
9. Attach lower intake poly to middle intake axle poly
10. Insert axles with wheels, sprockets, and spacing
11. Attach chain



Pseudocode

WHEN R2 is Pressed,
Erect both pistons to score on the top goal

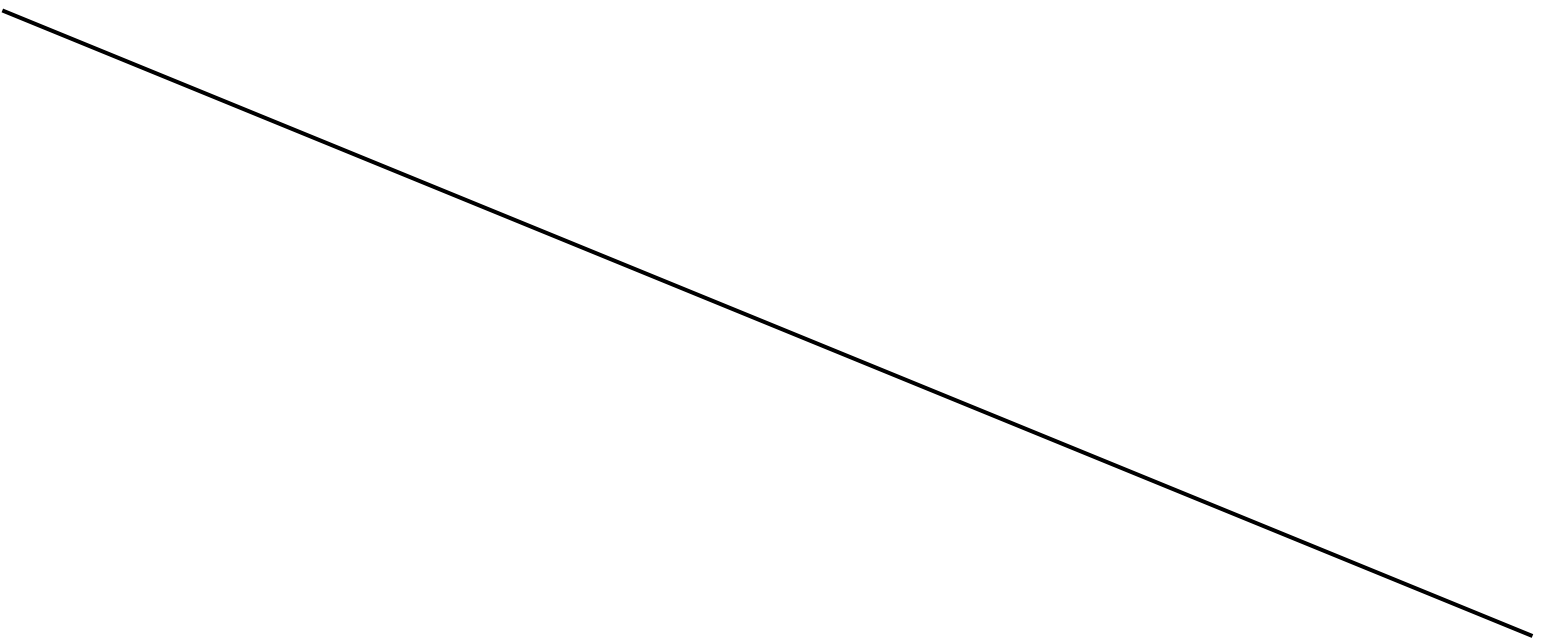
Code Block

```
// R2 - Hold to score on long goal
bool pressedR2 = controller.get_digital(pros::E_CONTROLLER_DIGITAL_R2);

if (pressedR2) {
    // HOLDING R2 - Erect pistons for long goal
    scoringPiston.set_value(true);
    scoringPistonState = true;
} else {
    // RELEASED R2 - Return to neutral (storage position)
    scoringPiston.set_value(false);
    scoringPistonState = false;
}
```

Explanation

To ensure that scoring on long goals is extremely straightforward, only when R2 is pressed the pistons erect to score on the long goal. Once R2 is released, the pistons return to the neutral state which allows balls to be stored and scored easily. This essentially makes it a toggle switch.



This is the feeding mechanism. This is what we want it to achieve.

- Be able to hold multiple at the same time, but here's where the problem arises. We have 2 main strategies.
 - Take a few balls. This strategy is based on mobility. Because if we take too many balls at the same time, the bot must be bulky, heavier, and slower. Being fast allows fast scoring and descoring, combined with more speed. However, it can't score many balls at one, so if there's a golden opportunity to score, we would only be able to put in only 2~3 balls, and we'll have to keep on picking up balls.
 - Take many balls. This strategy is based on stability. If we take many balls at once, the bot will naturally be bigger and stronger. The increased size will help the bot better defend against "attacks" from our opponents, and if we hoard all the balls, the opponent will have no balls to use to score. However, the size will result in a slower and less mobile bot and the added size will make it hard to move under the long goals.
- Be reversible so we can score into the lower goals using the bottom
- Be able to transfer the balls quickly enough to store them instantly once we reach the goals.

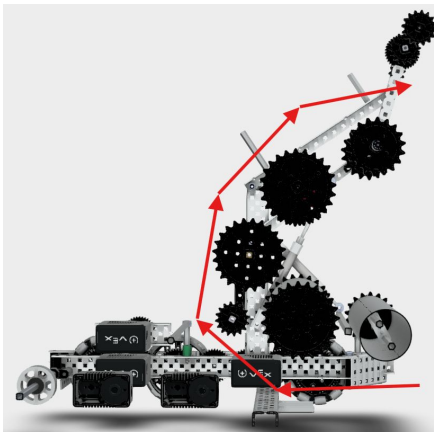
Now that we have the aim in mind, we made our first design.

We made a prototype via fusion 360, which plan to use on this first bot.

Our thought process was:

- 1) We first ran through some intake designs and decided to go with the C-shaped intake. This might seem counterintuitive now, but at the time of its development, we didn't know this was such a bad idea as we only wanted a design that could take many balls
- 2) We used coges wrapped with rubber bands to move the balls and create this C-shaped feeding mechanism

Citation: Team 44252A



Pros

- It's simple
- It holds multiple balls balls
- It allowed us a space in the back of the bot for a ball storage, which was in the works. (The ball storage would later be cancelled)
- It allowed us to concentrate the weight to one side, which we believed would make the bot harder to knock over. (It did do that, but once it fell, it really fell)

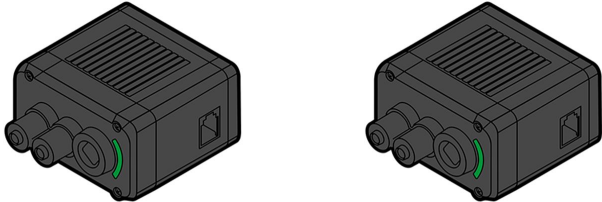
Cons

- We had to turn around to score
- By placing the bulk of the weight on one side, it becomes unbalanced. We originally thought the weight of the drivetrain would be enough to balance it, but no, no siree. This also made it kind of wobbly when it moved, which limited its maximum speed.
- The balls sometimes became stuck around the curve as we realised the design didn't take gravity into account
- Once the bot was tilted beyond its tipping point (which was very easy to reach) it would fall uncontrollably and had not way to stop itself.

Goal

We will compare and contrast different types of ball storage methods to select the optimal one and plan our intake design implementation.

Possible Solutions

Split motor setup - 1 11W motor to intake balls while another 11W motor to actually store balls	 Storage Motor Outtake motor
Pros	Cons
- Storage is extremely easy as the intake power is split across 2 motors	- Intake is less powerful because only 1 motor is used for storage intaking - Less efficient and jamming may be more frequent

String Piston Locking mechanism - A piston pulls a piece of string to lock balls. This forces balls to stop and does not allow them to move past a certain point.	Citation: Team 471B 
Pros	Cons
- Simple to implement - Effective design	- Uses another piston - Having to tune the tension of the string can be difficult

This is the scoring mechanism. This is what we want it to achieve.

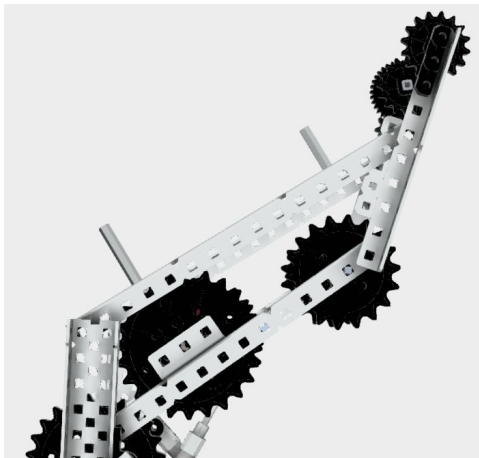
- Score the balls in the two higher goals. Now, there are two ways to achieve this
 - Adjustable scoring. This means our mechanism will be able to move upland and down. If we do this, it'll be easier since there's only one place the balls can go out, so the feeding mechanism won't be interrupted. The cons, on the other hand, mean that the mechanism will be less stable since we're making it go up and down. Also, it might get jammed while transitioning, and the transition from the long goal scoring mechanism to the central goal scoring mechanism will take time, and might require more pneumatics.
 - Two separate outputs. This means that each output will be more stable since they'll be integrated into the main haul of the bot, and we won't need an extra permutronic to move the scoring mechanism. But this will mean our feeding mechanism needs to be split into two(?) or at least have two output pathways, which may or may not be impossible to do.

Now that we have the aim in mind, we made our first design.

(cont.)

we made a prototype, which we used on the first bot. Our thought process was:

- 1) Make an adjustable score,
- 2) Realise that's too hard
- 3) Make an unadjustable Scoring mechanism to score quickly after we slam the bot into the goal.



Pros

- It can't move, so we've calibrated the scoring mechanism to make it able to score in the high goals without adjusting)
- I's able to score relatively efficiently in the high golds despite its flaws.

Cons

- It can't move so it can only score in the high goal and not the middles goal.
- The chains sometimes dislodged for the gears which might be a fundamental issue with the structure or how they're connected

This is the aligning mechanism. This is what we want it to achieve.

- Align the bot to the long goals in the back to make sure we can score accurately. Now we have two options:
 - An “open” aliner that is bigger than the goalpost of the long goals, which acts like a claw to grab onto the goal. This version will allow more stability, and if another team tries to ram into us, it probably will have no effect. This will also allow us to feed the balls in faster without fear of misalignment due to the stability of the aliner. However, it might make us a bit slower, clip onto other things more quickly, and there's a chance that we could accidentally jam it into one of the holes in the goal post and break it.
 - A “triangle” is a smaller design that kind of looks like a triangle. This design is smaller and sleeker. It will allow us to slot into the goalpost quickly and begin to score. It will also increase speed as it's going to be lighter than the claw. However, it is also less stable than the claw design and might make us vulnerable to getting pushed around by the other robots when we're scoring.

NOTE: We didn't think an aliner was necessary for our first bot as we thought we could just manually align the bot. It was only after we went to a competition and saw the other bots' designs that we realised that it would be better to align it because.

1. When we manually aligned it sometimes we made a mistake and the balls hit the side of the goals and bounced out instead of in.
2. In some parts of the field, our vision is impaired, which hinders our ability to align the scoring mechanism to the goals.
3. We realised that without an aliner, our bot could be easily moved via another bot ramming into it while it's scoring, and we would lose the balls. But with an aliner, we hoped that the bot could endure some level of ramming into and keep scoring.

Design 2: Under goal robot

As we kept making our first bot, we begun to see videos online of other vex bots that were far better than ours. Before we even finished building our first bot, it was already outdated in many many ways. This combined with the underperformance of our own bot lead to us destroying our bot before it was even halfway done and then starting again, making a far better design. Below is all the reasons why we felt like we needed to change our bot.

Changes in the META

With our current robot design, we noticed that the META was changing extremely fast away from storage bots so we started to consider new designs

Importance of Fast Match loading

- Instead of focusing on bots that have high amounts of storage, teams started focusing on fast match loading instead.
- **Problems:** Our robot design can fit a maximum of 4-5 balls which is not even enough to obtain the control bonus. In skills, we are also unable to clear the match loaders due to the low amount of storage

More competitive Designs

- We noticed that there were fast and more competitive robots now that we're able to score middle goal and long goal alot faster.
- **Problem with current bot** : Based on some of the bots we saw on youtube, our robot was not able to score as fast and efficiently. With the focus being on fast match loading and storage, our robot falls behind

Moving beneath the long goals

- We noticed that more and more teams were beginning to build robots that could move under the long goals which was much better for maneuverability. We wanted to follow this trend as we believe it would make us much more competitive
- **Problem with current bot** : We are unable to move under the long goal due to the height of our robot.

Goals for Robot Iteration 2

With our final decision to build a new robot for push back, we decided to set a few goals for the development of this new bot

Moving underneath the goals

- In order to improve our maneuverability around the field, we wanted to build a robot that can move under the long goals. This essentially allows us to play better defense and also we can move faster to the long longs

Faster Scoring on long goals and Bottom Goals

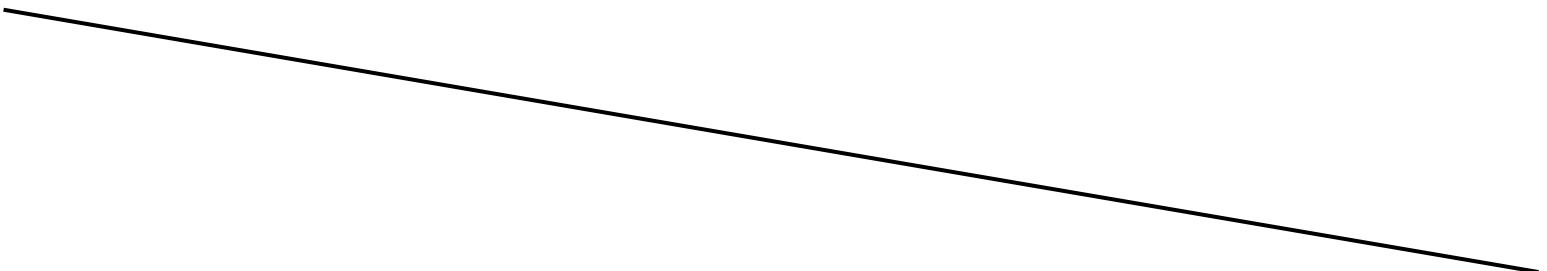
- We want our robot to score on the long goals fast and also able to score on the bottom middle goals. This is because many teams, are running the ruiguan (2 ball wide intake design by team 9123X)) which are unable to score on the bottom. If we can score on the bottom, we would be in an advantageous position.

The ability to Descore Long goals

- With our old robot, we noticed that since there wasn't a plastic descore flap, we were essentially unable to descore balls on the long goal by simply hitting them out with our drivetrain. For our new robot we want some sort of descore hood.

Mechanism to store balls

- We want to have some sort of mechanism to store balls. With teams scoring around 6-7 balls at once, we want to be able to hold a similar amount of balls to clear matchloader tubes.



Goal

We want to effectively design a robot that can score on the long goals but still able to move underneath the long goals to play better defense/offense

Problem Statement

Compared to our prior design, we need a robot that can move around the field more efficiently and score faster

Solution Goals

The robot must be able to...

1. Consistently move underneath the long goal while also being able to score on the long goal

Solution Constraints

1. Must use a maximum of 22 watts
2. Must use only legal VEX parts

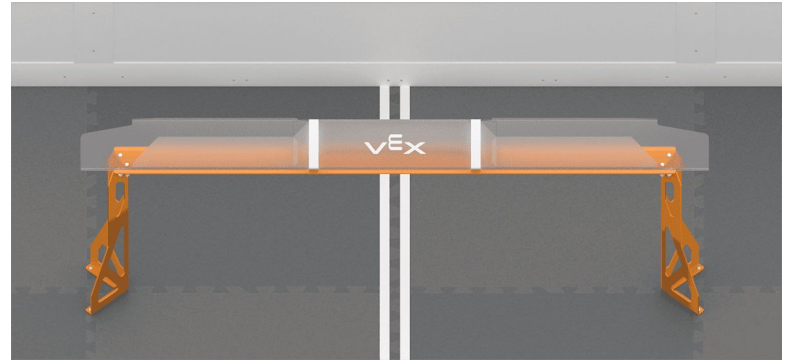


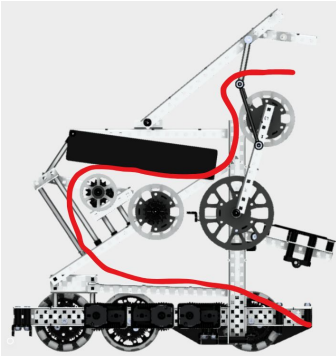
Photo showing vex Long goal

Goal

We will compare and contrast different types of long goal scoring designs to select the optimal one and plan our design implementation.

Possible Solutions

Claw - 2 sets of claws that pinch together, securing rings between them	
Pros	Cons
- Uses motor/piston power to grip rings - Versatile (can also be used to hold mogos, sweep corners, touch climb bar in auton) - Can likely fit under the long goal if tuned right	- Requires using an extra motor/piston - Claw must deploy onto the ground, making it prone to impending defense - Extremely hard alignment during pressure

Pistonised Long goal scoring (C Shape) - We made a design in CAD that stores balls in a C shape and can be pistonised to move up and down to fit under the long goals	 Inspiration from team 334U
Pros	Cons
- Efficient path - Can efficiently fit under the long goal - Storage amount is high as path is longer	- May have scoring delays if not tuned correctly - Jamming might be a concern

Goal

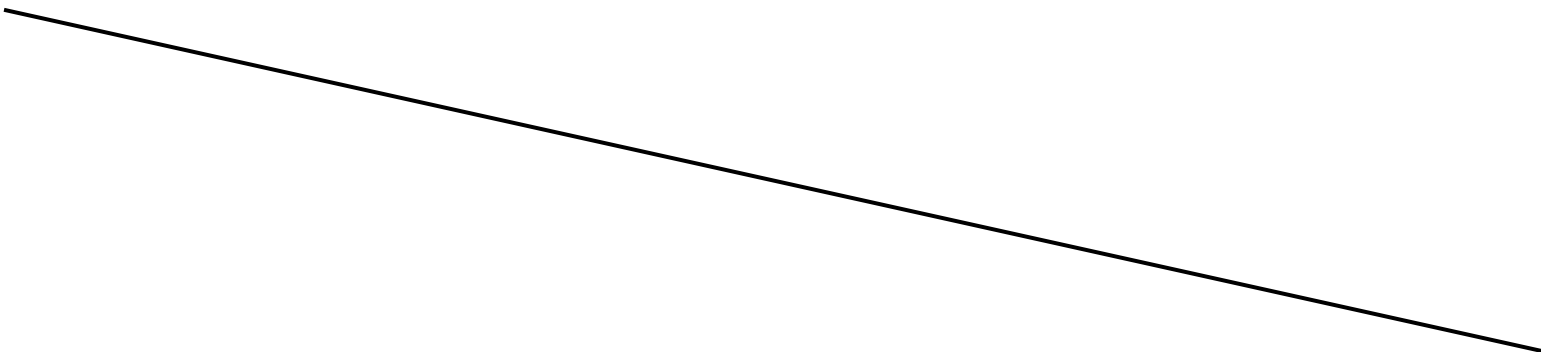
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Rubric	Claw Shaped Intake for balls	Snail Bot	C Shaped intake
Ease of Driving (10)	1	9	9
Speed of Intaking / Outtaking (10)	1	7	9
Can it fit under the long goal?	Yes	Maybe	Yes
Build Complexity (10)	6	7	8
Total (30)	8	23	26

Conclusion

Using our decision matrix, we have decided to develop a C shaped robot. This is quite a big change from our previous design which is sort of like a snail bot. We made this change because the C shaped intake path is able to consistently store many balls (teams like 334u) around 6-7 balls. Additionally, this design can be easily tuned/developed to fit under the long goal compared to a snail bot which is vertically a lot higher. The C Shaped design also has better center of gravity as the robot is lower overall. In conclusion, we have decided to build this design



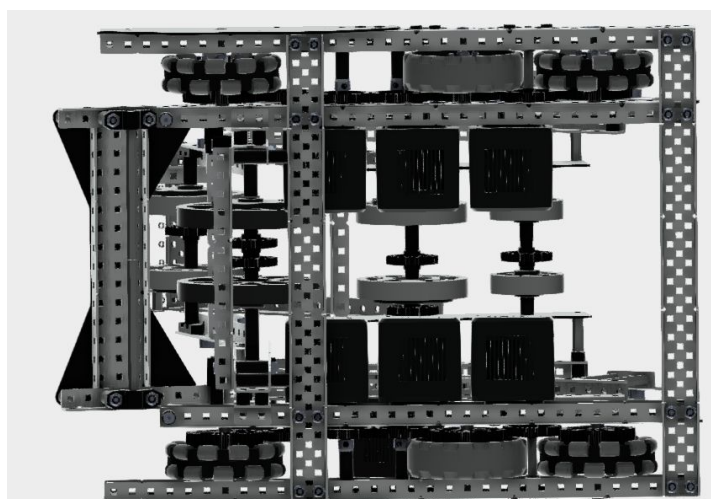
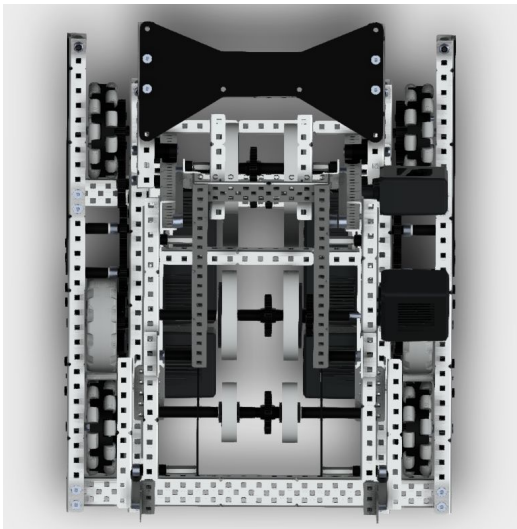
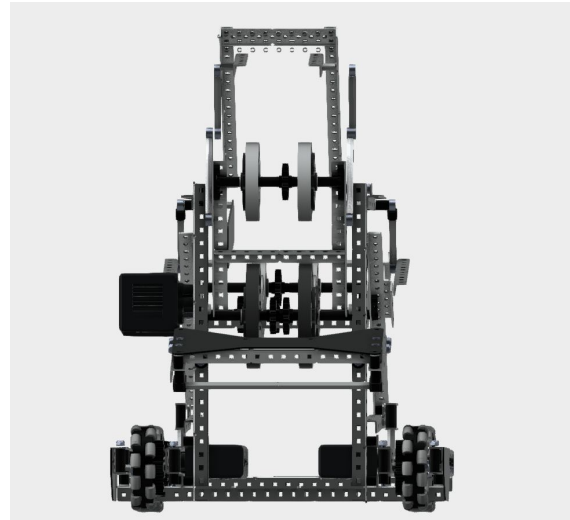
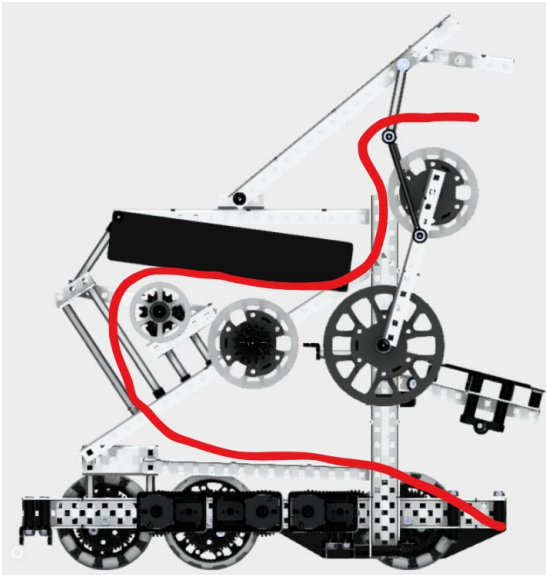
Goal

We will select our final Long goal scoring design and utilize Autodesk Fusion 360 to construct a CAD model before building.

Selected Solution

- 2 11W motors
- Plastic funnels to guide balls
- C shaped path
- 1:1 Gear ratio (600rpm)

CAD Plan



Goal

We will identify a potential design to implement on our C Shaped robot to effectively store 6-7 balls which is our objective

Problem Statement

Compared to our prior design, we need an intake that can collect more balls, around 6-7 balls. Since we want the intake to be powered by 22W, we need a way to hold balls.

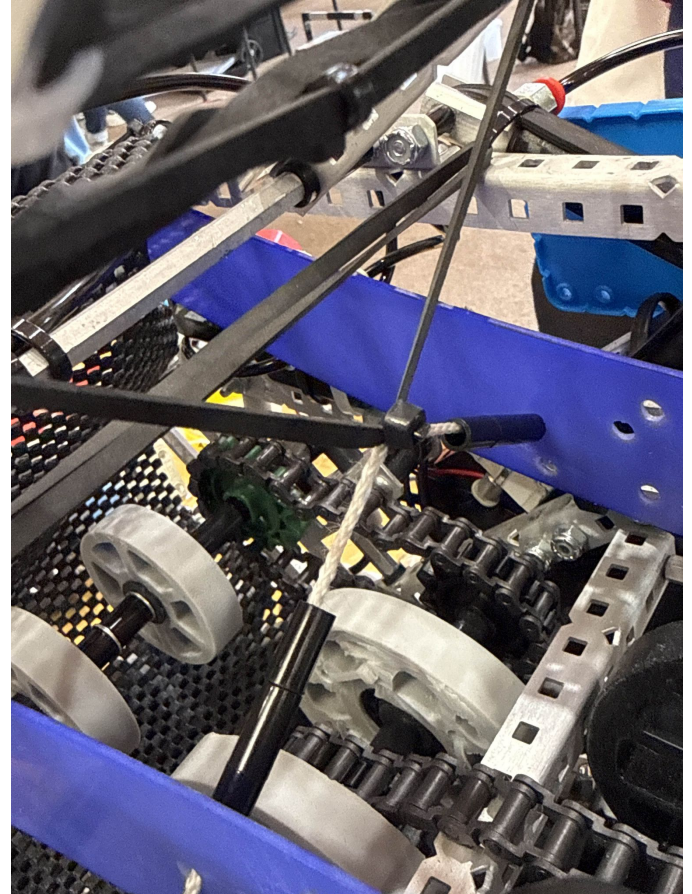
Solution Goals

The robot must be able to...

1. Hold 6 balls without jamming
2. Hold balls without them falling out or slipping out

Solution Constraints

1. Must use only legal VEX parts
2. Must not use anymore motors



Testing of our locking mechanism to trap balls

Goal

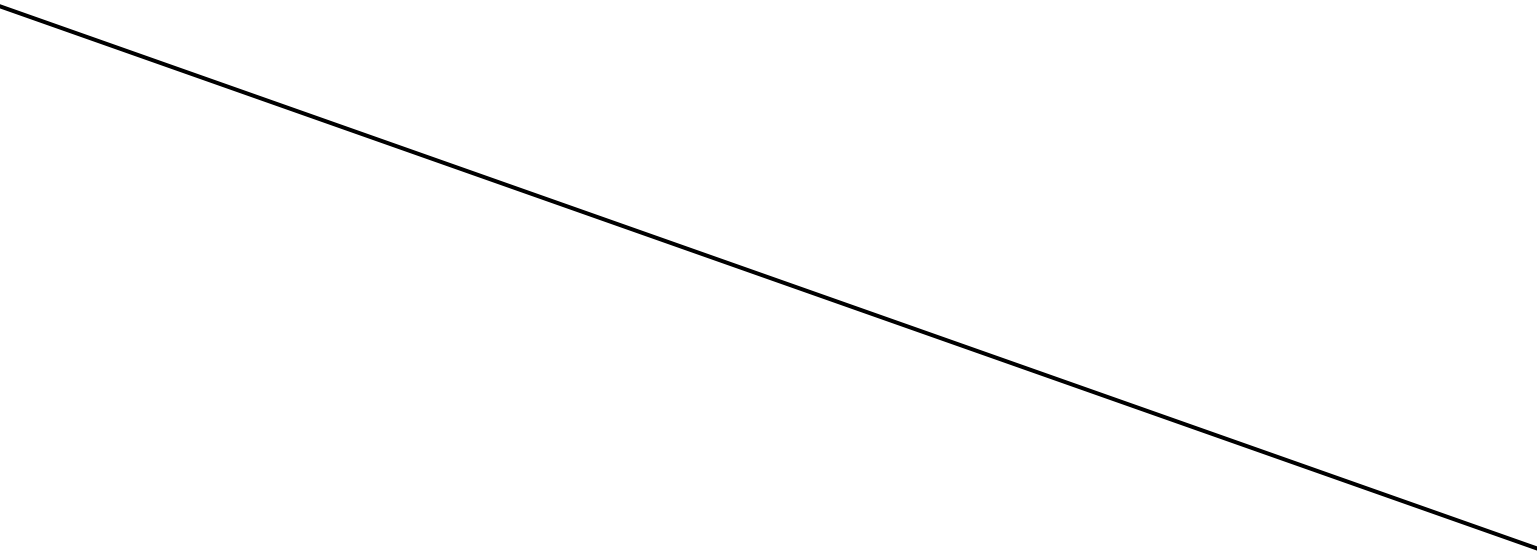
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Rubric	Split 11W motor setup	Piston locking string
Efficiency	3	8
Speed of Intaking / Outtaking (10)	7	10
Build Complexity (10)	6	5
Total (30)	16	23

Conclusion

Using our decision matrix, we have decided to build a storage system using a piston string setup. This setup uses a piston to release a piece of string to allow balls to pass through. This allows us to achieve our goal of using both 11W motors for the whole intake and is a much more efficient and less complex design process compared to a split 11W motor setup



Goal

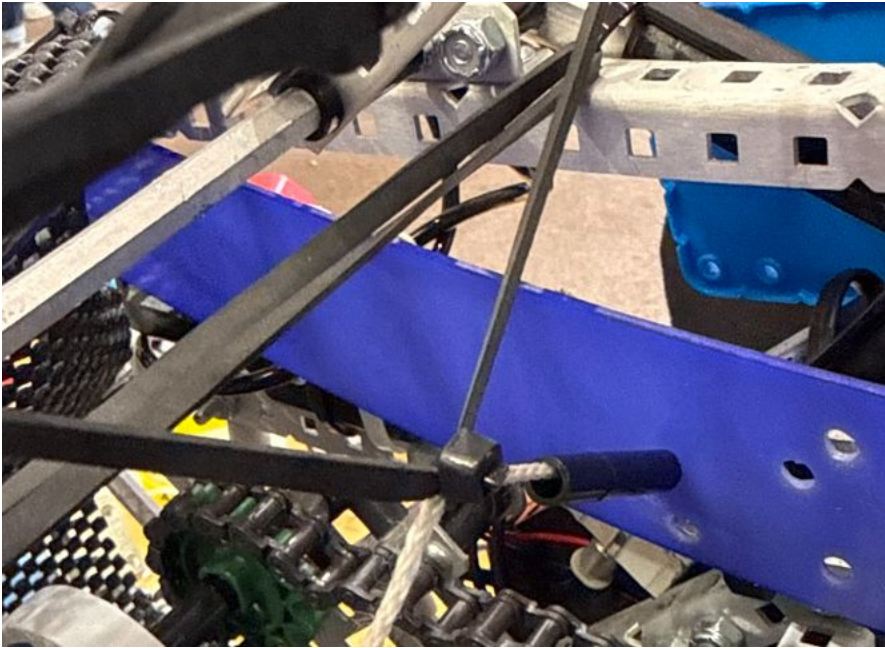
We identified a design flaw within the old robot. The balls keep getting jammed and they're hard to unclog quickly. We need to find a way to fix this on our new bot.

Solution ideas

To make the balls not clog, we have two options. Either make the passway so thin it can only allow one bal to pass through at once, or make the walls slanted so the path becomes like a funnel and the ball unclog by itself. We went with the second option.

Solution

1. First we found a suitable material. It needs to be flexible but hard inforder to be able to be bent like a funnel but also not entrap the balls with its softness. We choose plastic.
2. We then designed a blueprint and used our school laser cutter to cut out 2 pieces.
3. Then we tried to attach the pieces but found slight misalignment with the screw holes, so we manual drilled into the plastic and attached the pieces



Close up of the funnel plastic piece on one side

Solution evaluation

Pros	Cons
- It solves the issue of clogging so our balls no longer clog. - It doesn't decrease the amount of balls we can store. - It's flexibility allows us to drill more holes inside and use it as a support if we ever need to attach anything else there	- It takes up 2 pieces of plastic 12, which could be used elsewhere - It's a bit too soft. This is due to the type of plastic we used, not due to a design flaw. In future uses we'll find a harder plastic. - The extra holes we drilled makes it look bad.

Goal

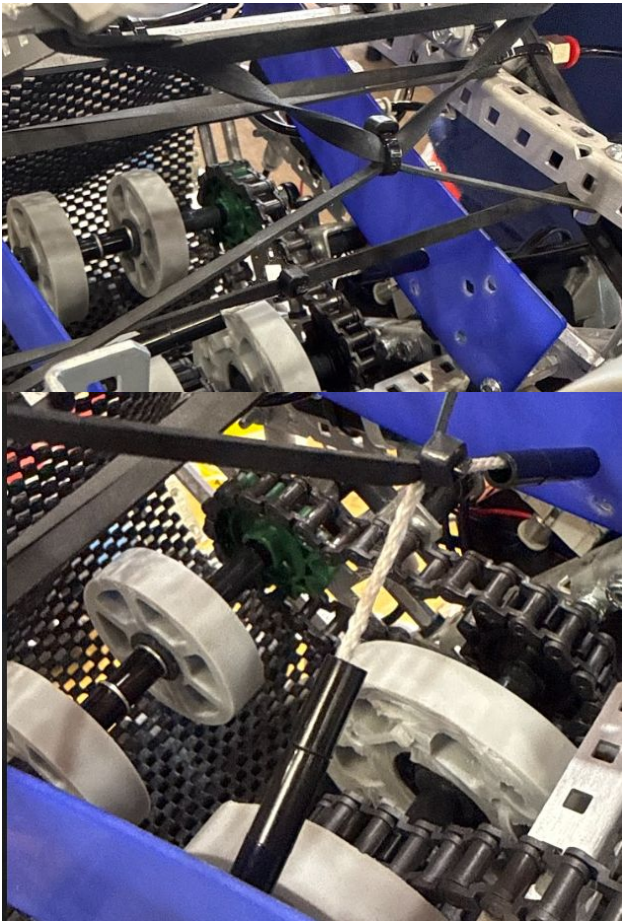
We identified another design flaw within the old robot. The balls just roll out as soon as they pass through the ball storage. We need to make something that can stop the balls temporarily before we reach the goals.

Solution ideas

To stop the balls, we needed a stopper from a flexible material so they balls will be stopped but be able to pass through.

Solution

- 1. First designed a stopper in the shape of an X that would attach to the two plastic funnel pieces.
- 2. We made entirely with rubber bands to test and found that this x shaped design worked.
- 3. Then we worked on a system to open and close the stopper. We settles on a design made with string and uses a piston (not seen) to pull the stopper up and down letting the balls stop and go.



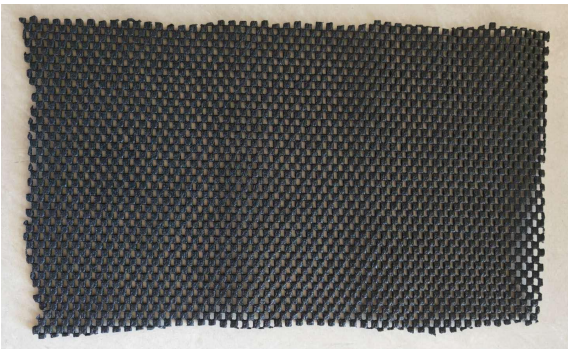
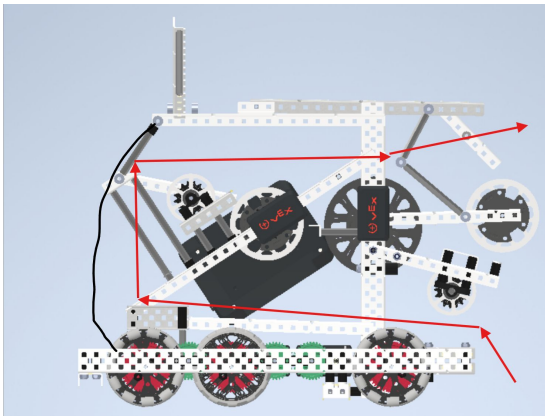
Ball stopper

Solution evaluation

Pros	Cons
- It allows the balls to stop and go at will. - It opens and closes quickly. - This design only uses one piston and doesn't;t use a lot of pressure to move, making it very energy efficient.	- It's a bit fragile and breaks sometimes. - This design is highly depended on the tow plastic funnel pieces so if we ever have to change them this design becomes no longer viable

Goal

As part of our design for the ball storage we need a material to stop the balls from falling out at the back of the ball storage. We had a few materials in mind.



Mesh

The material is used as indicated by the black line

Decision Matrix

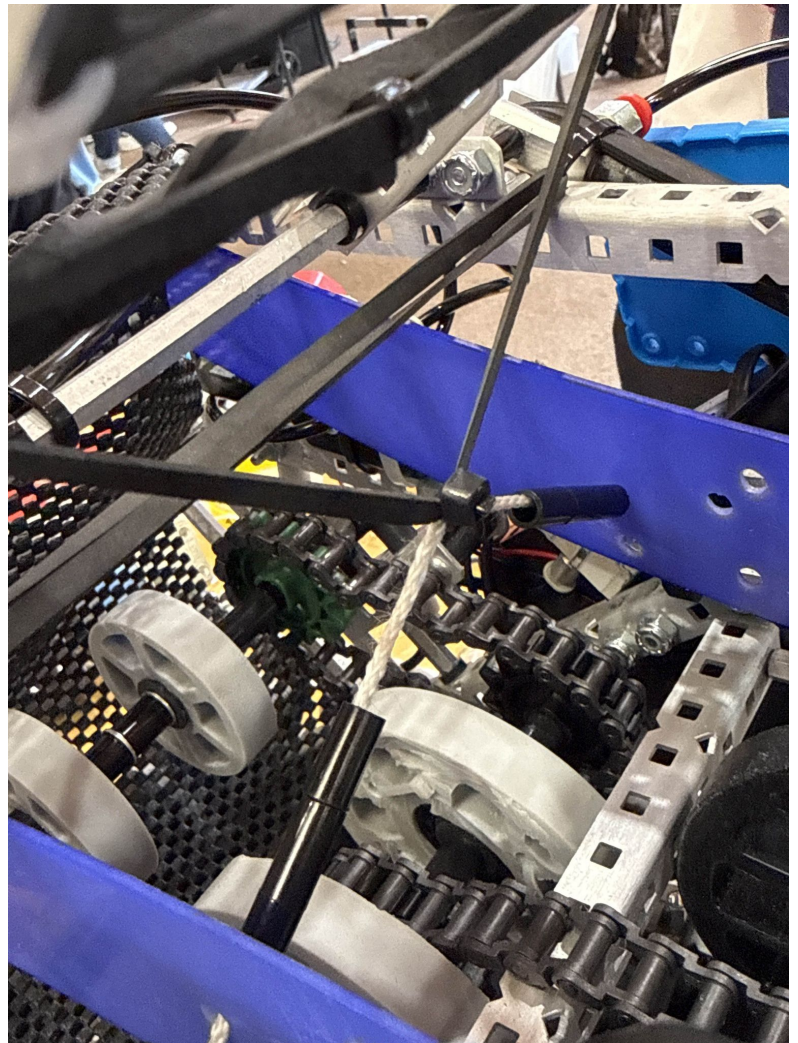
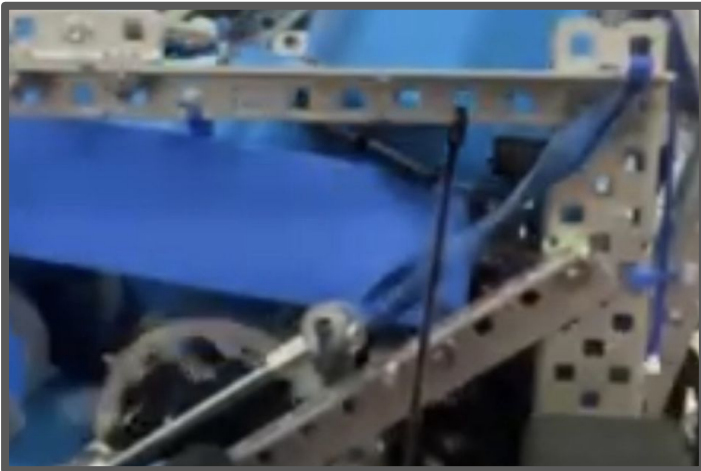
Rubric	Plastic	Zip ties	Rubber bands	Mesh piece
flexibility	3	2	4	8
Fluidity of balls passing through	5	3	7	8
Durability	10	5	5	9
Ease of repair	3	10	10	8
Price	2	8	8	6
Build Complexity	2	8	8	6
Total (60)	35	36	42	45

Conclusion

Using our decision matrix, we have decided to use the mesh piece for our chosen material as it received the highest score

Steps

1. Cad 2 pieces of plastic to mount the string and rubber bands on. This will also act as the funnel for the balls
2. Drill a hole on either side of the funnel to allow string to pass through
3. Mount a piston on a L channel which is connected to the string
4. Create an X shape using rubber bands to make sure that the balls get stopped properly
5. Add spacers on the string to allow balls to efficiently slide through the gap to score
6. Tune the length of the string to open and close to allow balls to pass through
7. Test the design

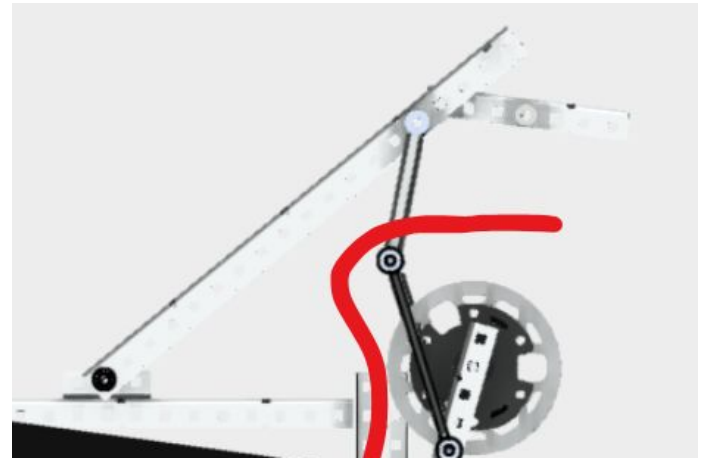


Goal

We want an adjustable scoring mechanism able to reach both the long goals and the high centre goal.

Problem Statement

In our last bot we couldn't score in a long goals, so this time we want to make a retractable scoring mechanism that's able to help us score in both the long goals and the high centre goal. We also want a descoring mechanism as we noted that in America, the teams are starting to descoring and it has become the new meta, so if our bot didn't have that capability we'd be at a huge disadvantage



Original CAD of the scoring mechanism

Solution Goals

The robot must be able to...

1. Score in the long goals
2. Score in the high centre goal
3. Go under the long goal.

Solution Constraints

1. Must use only legal VEX parts
2. Must use plastic (Acetal 1.5mm)

Material

Goal

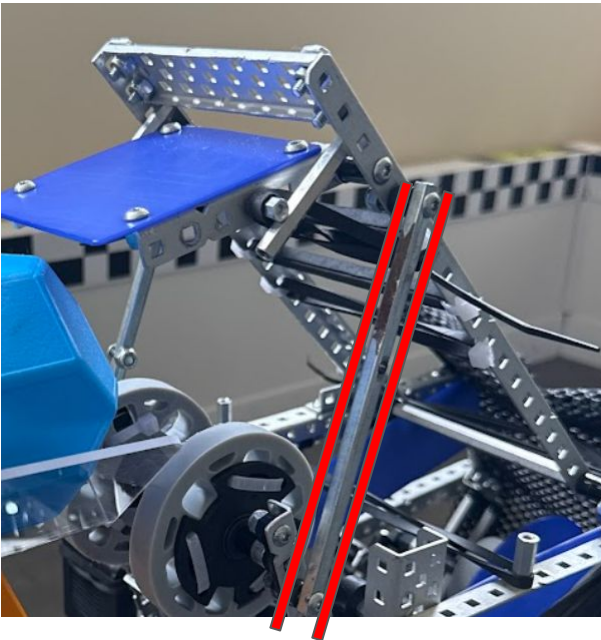
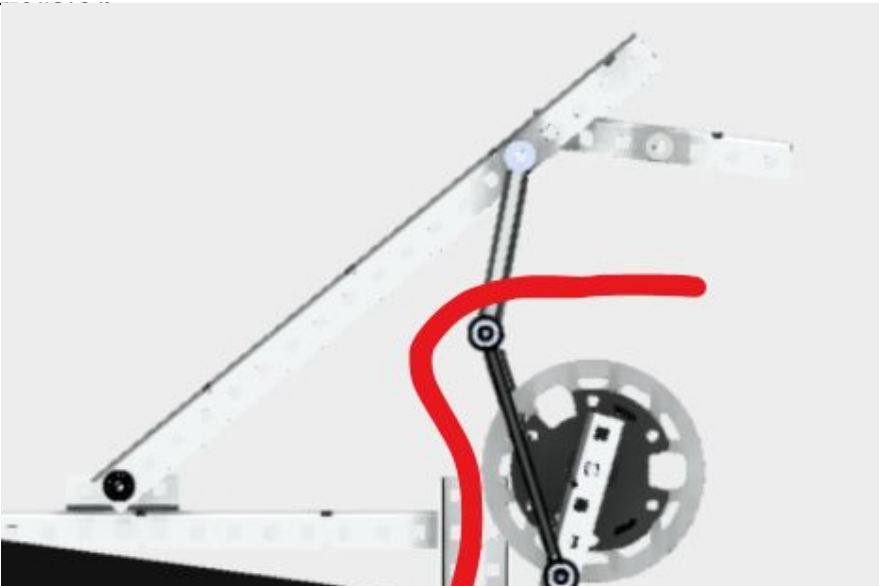
We identified a design flaw within our original design. We originally used standoffs for support as we thought using two standoffs connected together using a shaft collar would allow the robot to pivot to reach both middle and long goal scoring, but it's too unstable. To ensure stability we need to find a different material.

Decision Matrix

Rubric	Standoff (original)	Half C channel	High Strength Axle
Speed	8	5	8
Stability	2	7	10
Durability	3	8	10
Sleekness	9	6	9
Total (30)	22	26	37

Conclusion

Using our decision matrix, we have decided to modify the design for the scoring mechanism using the high strength axle, Below is a comparison between the original CAD and the high pin axle



Goal

We wanted to make a descoring mechanism on top of the scoring mechanism so we can dislodge the balls in eh long goals by ramming into it

Solution ideas

We could a C channel on another C channel to form a T shaped tool and knock out the balls, or we could mount a plastic piece atop the scoring mechanism, or we could use 2 stand offs to form an triangle shaped edge atop the scoring mechanism. We went with the second option

Solution

- 1. First we designed a blueprint and used our school laser cutter to cut out 2 pieces.
- 2. We mounted it on top of the scoring mechanism and did some testing to see it's effectiveness.



Close up of the descoring mechanism

Solution evaluation

Pros	Cons
- It it able to dislodge 7-8 balls when rammed at full speed. - It's stable and durable	- N/A

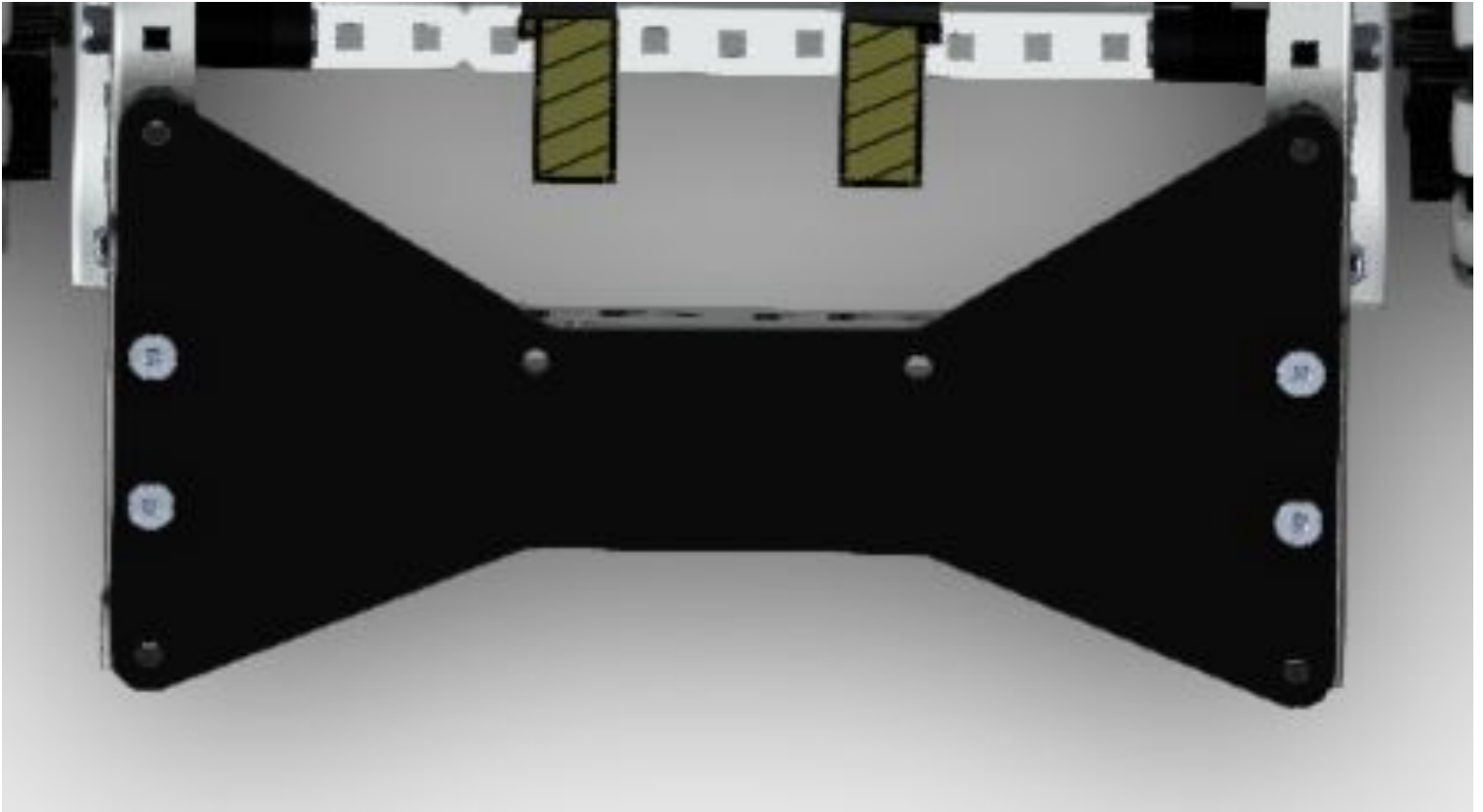
Goal

We will develop a similar style aligner to the 334U robot but compatible with our existing robot. We will take inspiration and utilise fusion 360 to develop the cad model

Selected Solution

- 1.5 MM thick Acetal (delrin)
- Aligner that braces the C channel of the intake
- Aligns to the long goals

CAD Plan



Goal

We want to implement some sort of aligner to allow us to align to the long goals and mid goals efficiently

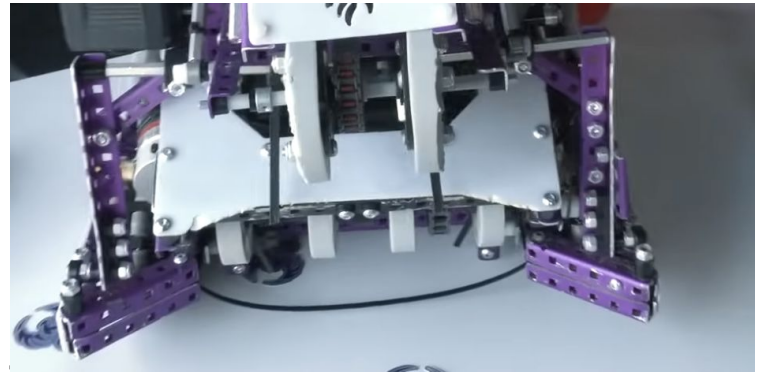
Problem Statement

Compared with our previous bot that didn't have an aligner, we noticed that aligning and scoring on the long goals fast was extremely difficult and slow. We wanted to align faster

Solution Goals

The robot must be able to...

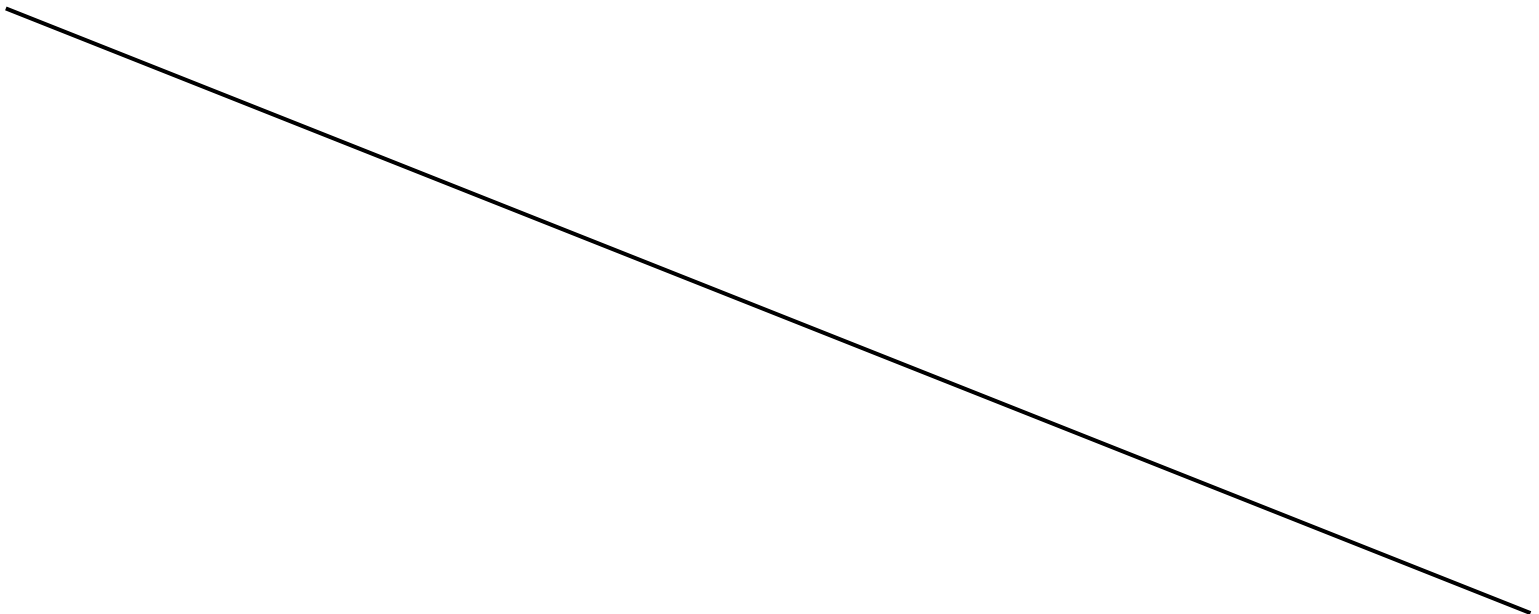
1. Align to both the long goal and mid goal
2. Aligner must be at the correct distance to score balls properly



Aligner used by team 334U

Solution Constraints

1. Must use only legal VEX parts
2. Must use plastic (Acetal 1.5mm)



Goal

We will create a new intake that pushes the balls into the ball storage.

Problem Statement

Compared to our prior design, we need an intake that can actually work. Ideally it will push the balls in once it touches them to speed up the process of collecting balls

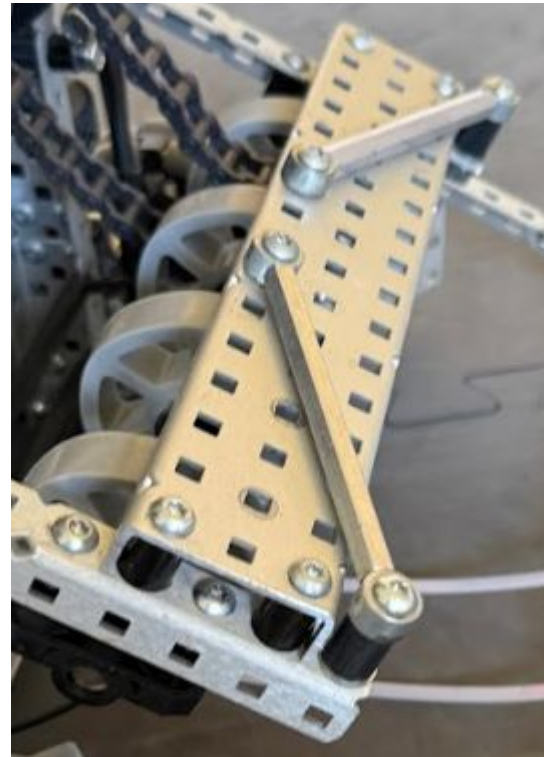
Solution Goals

The robot must be able to...

1. Push the balls in upon contact
2. Be flexible so it can “eat” the balls and not let them back out.

Solution Constraints

1. Must use only legal VEX parts
2. Must not use any more of our plastic pieces.



Close up of the intake mechanism

Goal

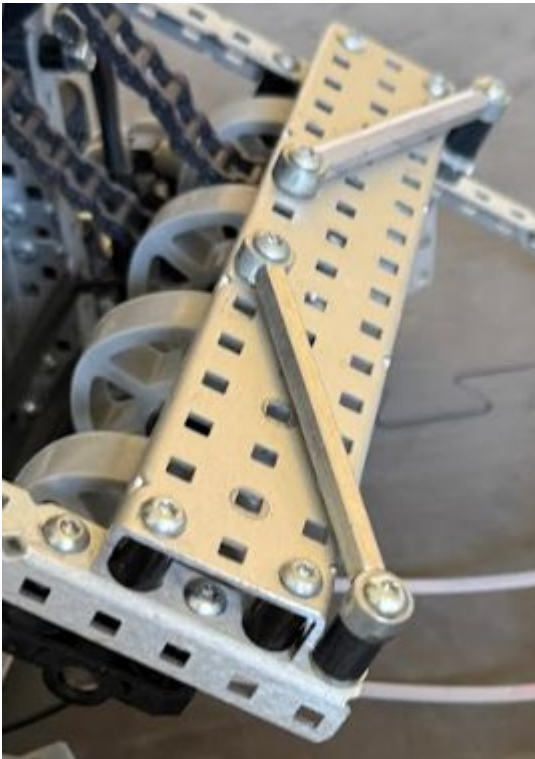
We wanted to make mechanism that can push the balls in as soon as it touches the intake.

Solution ideas

We could use either gears or flex wheels on top and make them spin to push the balls in. We could have them on the sides to push the ball in. We could have a layer of track that feeds the balls in once the balls end up on top of it, We chose the first option

Solution

- 1. First we put some flex wheel on a steel rod, secured in place by plastic spacers.
- 2. Then we mounted the rod onto the bot via 2 half C channels on either side.
- 3. We added a chain hooked up to a motor to make it run
- 4. After some testing we added a roof of a three wide C channel to protect the flex wheels if another bot were to ram into it during the competition.



Close up of the intake

Solution evaluation

Pros	Cons
- It it able to push the balls in quickly - It can “eat” two balls at once - With the roof it’s highly stable and can withstand being smashed by another not	- It's a bit bulky and when sometimes it can't life fast enough for all the balls to go in before it rams into them.

Goal

We will create a new match loader better than our old bot which can intake balls quickly.

Problem Statement

Compared to our prior design, we need a match loader that actually works, and works quickly.

Solution Goals

The robot must be able to...

1. Quickly feed the balls into the intake
2. Be able to dislodge the balls from the match loaders in the corners of the field
3. Move up and down as to not get in the way of normally intaking the balls from the ground

Solution Constraints

1. Must use only legal VEX parts
2. Must not use more than 1 piston to conserve gas pressure



Close up of match loader

Goal

We wanted to make mechanism that is able to dislodge the balls from the match loader, while forming a makeshift funnel that feeds the balls to the intake. It also needs to be raised up and down.

Solution ideas

We could use a plastic piece on a rod, have “teeth” sticking out from a rod, or use the same flex wheel rod design from our intake. We chose the second option as it was the lightest weighing out of the three and we believed it needed to be as light as possible to lower and retract up quickly.

Solution

1. First we needed to find out what kind of ‘teeth’ worked. Then mount it to a rod. Here are some trials:

Method	Works?
Nails on rod	N
C channel on rod	N
Thick flat rod	N
Smallest stand off on rod	Y



Half of the match loader

Solution evaluation

Pros	Cons
- It is able to push the balls towards the intake. - It's able to lift up and down quickly	- It sometimes failed and is kind of fragile. - It needs to be rammed hard into the match loaders for it to work. - It probably won't survive getting rammed hard by another bot

Goal

We have created a mechanism to locate the position of the bot to help with the autonomous phase.

Parts

The odometry is comprised of two pieces. Each piece has small omni wheels sandwiched by 2 diamond chapped plastic pieces to keep them in place, and a rotation sensor.

How it works;

The odometer works by tracking the amount and direction of spins by the wheels. By doing that It can calculate how far the bot has deviated from its original position and where the bot currently is. Our automitus program will use the position of the bot as trigger for its' programs/instructions. Hence the accuracy and position of the odometry is vital

Constraints

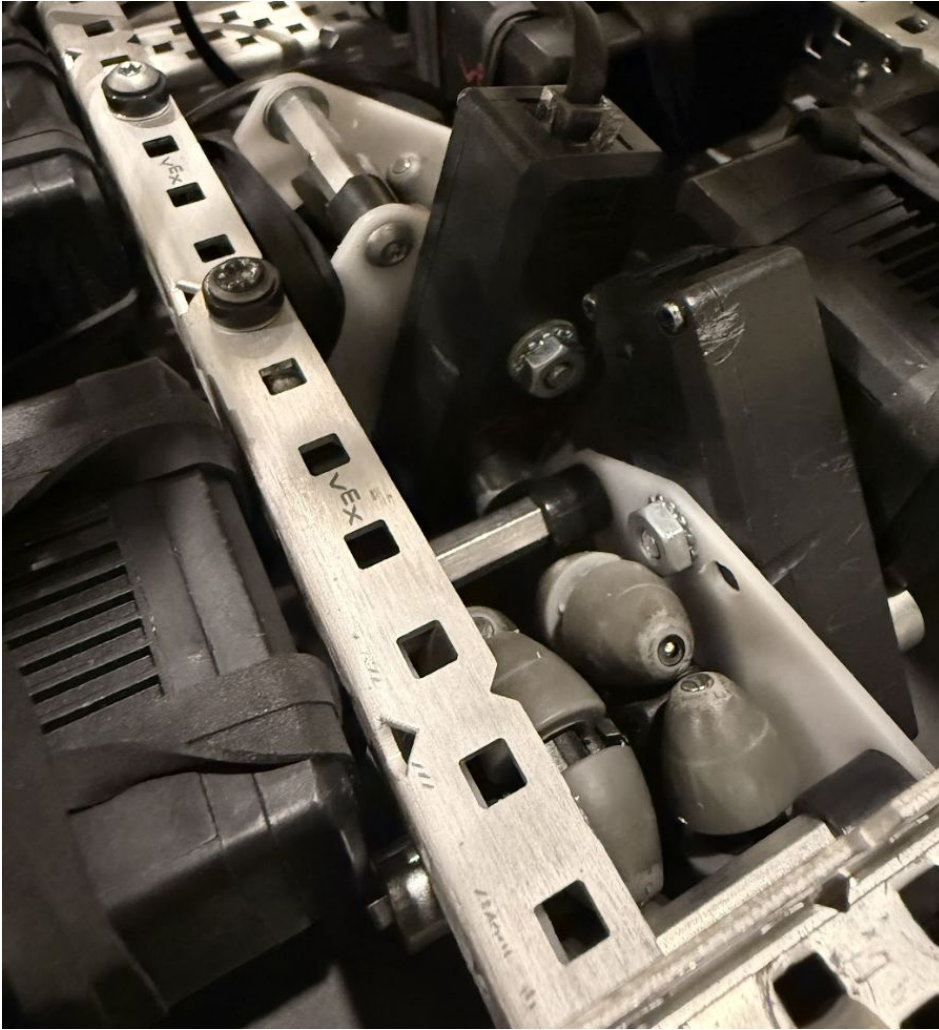
1. The odometry much be placed as shown below.
2. Both wheels must touch the ground at all times.
3. The wheels must not be too tight or too loose, and the way it touches the ground must be just right or it will skew the location.



Close up of odometry

Odometry Pod Setup

To allow the robot to move and comprehend its location on the field, we will need some way for the robot to tracking the direction it moves in. To do this, we have implemented Odometry pods + an inertia sensor to track Heading changes



```
lemlib::TrackingWheel vertical(encoder: &verticalEnc, wheelDiameter: lemlib::Omniwheel::NEW_2, distance: -0.2);  
lemlib::TrackingWheel horizontal(encoder: &horizontalEnc, wheelDiameter: lemlib::Omniwheel::NEW_2, distance: -0.1);
```

```
class CustomIMU : public pros::IMU {  
    public:  
        CustomIMU(int port, double scalar) : pros::IMU(port), m_port(port), m_scalar(scalar) {}  
        double get_rotation() const override { return pros::c::imu_get_rotation(port: m_port) * m_scalar; }  
    private:  
        const int m_port;  
        const double m_scalar;  
};
```

IMU Code Setup

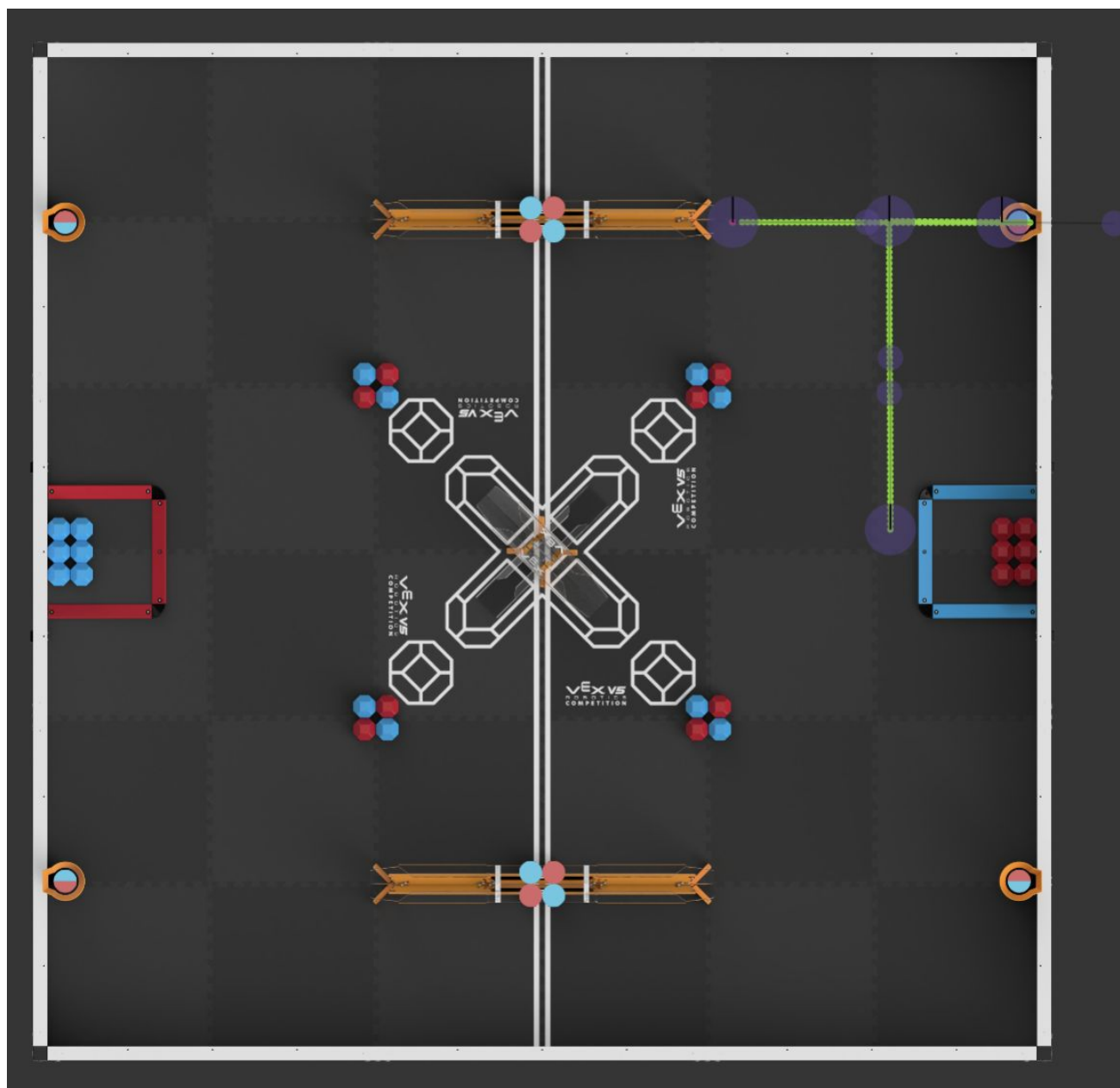
We noticed that VEX imu's often have about a 1-5 degree change for every 360 degree rotation. For example, if the robot does a rotation of 360 degrees, some IMU's can be off by about 1-5 degrees. We noticed during testing that our IMU had this behaviour. To Fix this, we implemented IMU scaling code:



```
class CustomIMU : public pros::IMU {
public:
    CustomIMU(int port, double scalar) : pros::IMU(port), m_port(port), m_scalar(scalar) {}
    double get_rotation() const override { return pros::c::imu_get_rotation(port:m_port) * m_scalar; }
private:
    const int m_port;
    const double m_scalar;
};
```

1. We created a **Custom IMU** class that inherits from the standard **pros::IMU** class to add scaling functionality
2. The class accepts a scalar parameter (typically between 0.95-1.05) that adjusts the IMU readings to compensate for drift
3. The custom IMU is instantiated with the correction factor: `Custom IMU my_imu(port, scalar)` and integrated into LemLib's odometry system

4 Ball Long Goal Autonomous



1. Start with preload in the robot
2. Drive forward until parallel with match loader
3. Turn To Heading 90 degrees and match load for 3 seconds
4. Drive Backwards and score onto the long goal

This will guarantee control zone for one of the long goals

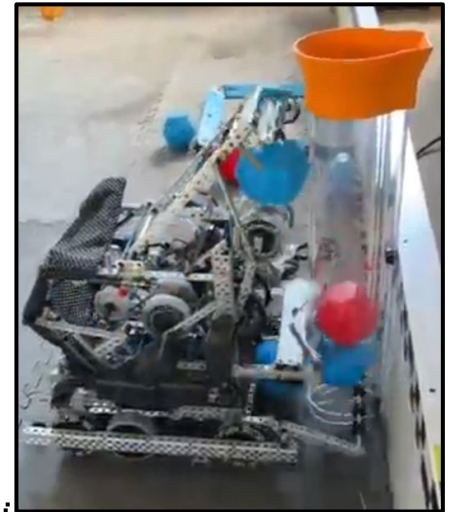
Driver Practice

Score on Long goals

- Practiced scoring on the LOng goals quickly
- Result: We are consistently able to relatively quickly and constantly,

Speed

- Practiced driving around quickly, getting into various scoring positions then move on to the next
- Result: We were able to get to a fast speed and familiarize ourselves with the field

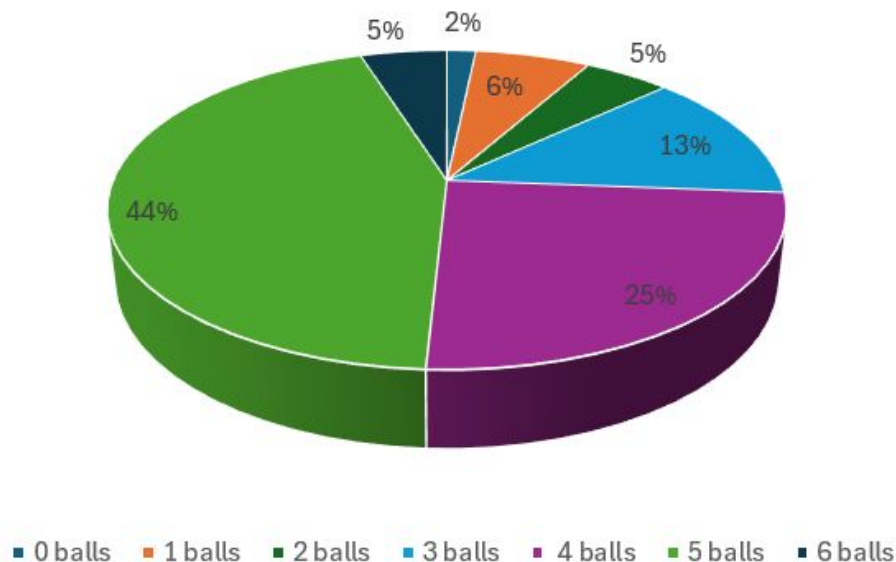


Speedy match loading

Match loading

- Practiced going to the match loaders and collecting as many balls as possible before quickly scoring into the long goals
- Result: We were able to take ~5 balls on average successfully and was able to score relatively quickly. Here are our results:

Match loading practice

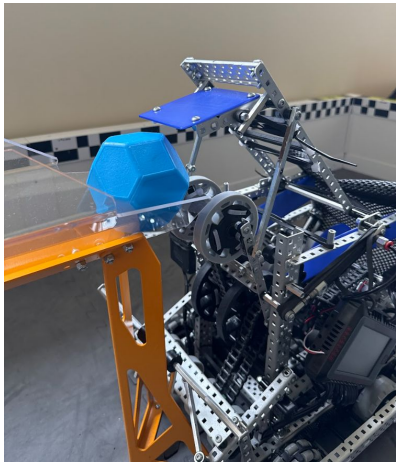
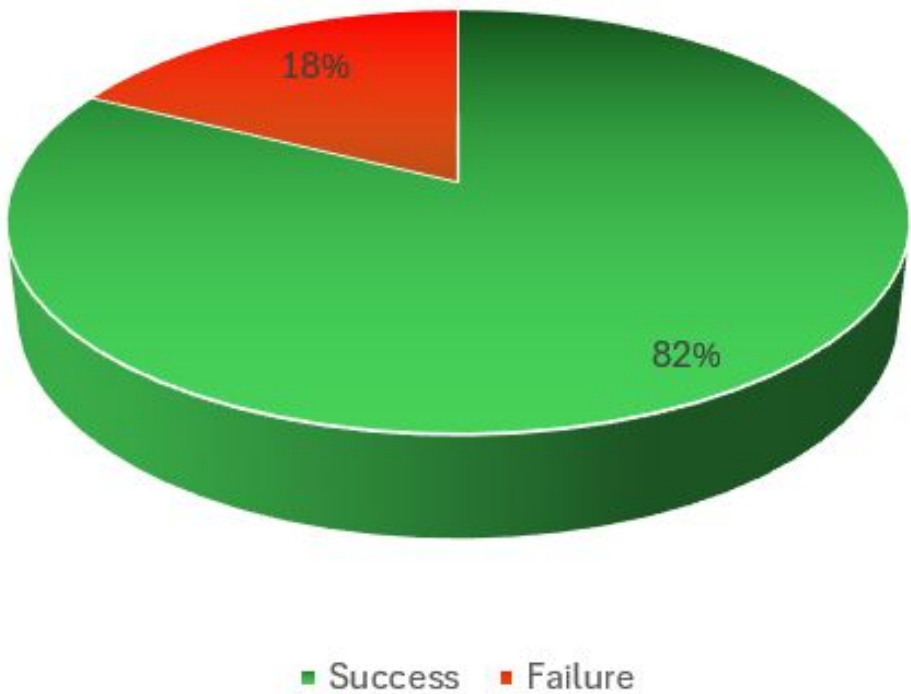


Retracting and erecting the scoring mechanism

- Practiced putting the score up, scoring, putting it down, going under the long goals and repeating this process over and over
- Result: We saw some inefficiencies with intake as the rings wouldn't enter in properly when they are stacked on top of each other.

Autonomous Testing

Testing Autonomous



Success

After testing out our qualification autonomous routines, we heavily tested our elimination autonomous for over 10 hours. Here are the overall statistics for each portion of the routine.

Section	Works:
Match loading	95%
Scoring in the long goals	82%

We only had one elimination autonomous routine as this was a 4 ball Long Goal that was extremely fast. This auton allows us to get the long goal faster than most teams currently. This autonomous routine also increases our likelihood of obtaining the control zone during the autonomous

Goal

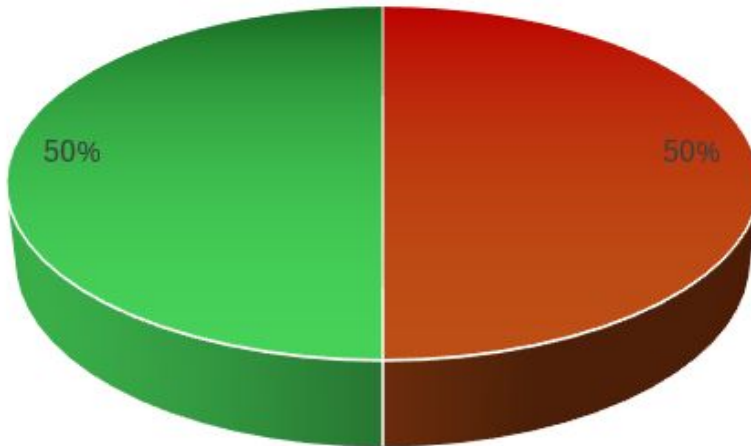
After competing in our first ever vex v5 tournament for the push back season using this bot, we decided to evaluate our first competition.

Results

Match	Red Alliance		Blue Alliance		Auton	AWP	Result
Q2	8115X 3168H	15	2915V 2923A	61			
Q9	2915F 11044C	55	8115X 2915K	39			
Q14	2921R 52392B	9	2900A 8115X	91			
Q22	22020C 8115X	75	52392A 22020Z	17			
Q28	70591A 22020X	28	2915N 8115X	5			
Q38	8115X 2908X	48	2915G 11044B	27			
QF #3-1	2908E 2908X	16	8115X 22020Z	53			
R16 #6-1	2915G 70591A	30	8115X 22020Z	0			

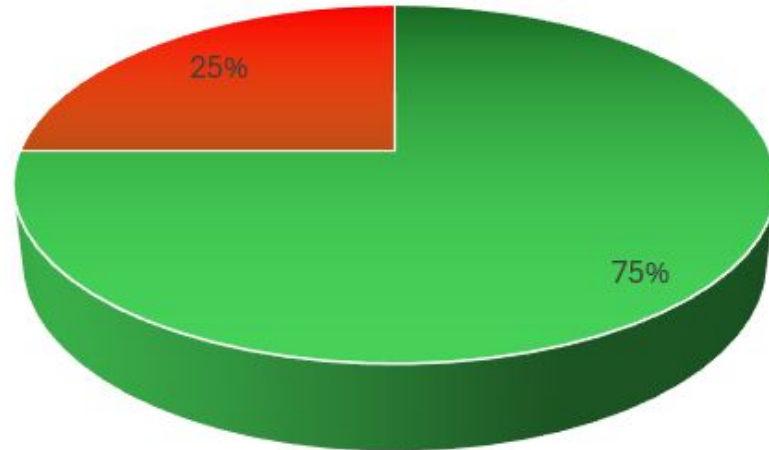
This section will discuss the list of issues that we faced with our robot at the competition.

Wins vs Losses



■ Wins ■ Losses

Autonomus Wins vs Losses



■ Wins ■ Losses

Consistent Auto

- In order to improve our maneuverability around the field, we wanted to build a robot that can move under the long goals. This essentially allows us to play better defense and also we can move faster to the long longs

Storage Issues on Robot

- In order to improve our maneuverability around the field, we wanted to build a robot that can move under the long goals. This essentially allows us to play better defense and also we can move faster to the long longs

Ball Release Mechanism

- We want our robot to score on the long goals fast and also able to score on the bottom middle goals. This is because many teams are running ruiguan (2 ball wide intake) which are unable to score on the bottom. If we can score on the bottom, we would be in an advantageous position.

Goals for Robot Iteration 2

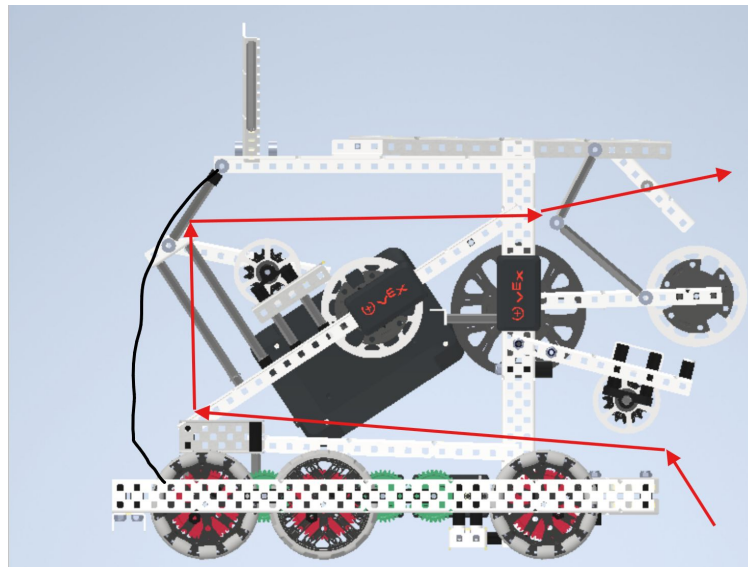
This section will discuss the list of issues that we faced with our robot at the competition.

Moving underneath the goals

- We also noticed that during competition, there wasn't a significant advantage of being able to move under the goals. Since there are also extra buttons for moving under the goal, this added to the complexity of our bot

Autonomous Issues

- One of the biggest issues we faced with our robot utilised front to front scoring. This meant that the robot intake from the front but also scored on the front. This was a very big issue for autonomous because when we moved from the match loaders to the Long goals, the robot had to do a 180 degree turn which used a lot of time.



Goal

After competing in our first ever vex v5 tournament for the push back season using this bot, we decided to evaluate our first competition.

Results

Match	Red Alliance		Blue Alliance		Auton	AWP	Result
Q7	2941H 2941D	39	22020Z 8115X	13			
Q12	8115X 2941A	82	11044C 2921F	14			
Q21	8115X 2996C	96	8757B 59833A	32			
Q25	2915M 3168H	36	52392A 8115X	37			
Q33	11044A 8115X	90	52392D 2921R	22			
Q47	2915Y 11044B	18	8115X 2900A	67			
R16 #5-1	8115X 2941H	94	2921F 2941G	18			
QF#1-1	8115X 2941H	114	52392D 8757C	11			
SF#1-1	8115X 2941H	70	22020C 2996B	3			
Final#1-1	22020X 2915V	74	8115X 2941H	25			

Design 3:
Front to back scoring
robot

Goals for Robot Iteration 3

With our final decision to build a new robot for push back, we decided to set a few goals for the development of this new bot

7-8 Balls of Storage for intake

- In order to improve our maneuverability around the field, we wanted to build a robot that can move under the long goals. This essentially allows us to play better defense and also we can move faster to the long goals

Front to Back scoring robot

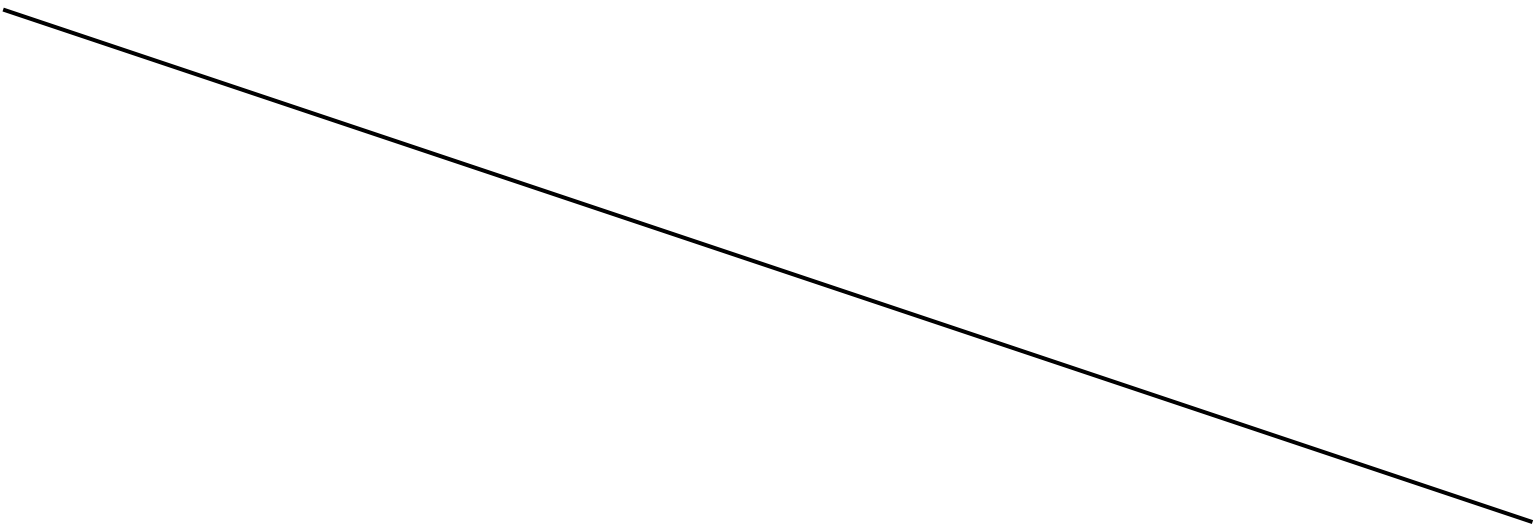
- We want our robot to score on the long goals fast and also able to score on the bottom middle goals. This is because many teams are running ruiguan (2 ball wide intake) which are unable to score on the bottom. If we can score on the bottom, we would be in an advantageous position.

Descore Hook

- With our old robot, we noticed that since there wasn't a plastic descoring flap, we were essentially unable to descoring balls on the long goal by simply hitting them out with our drivetrain. For our new robot we want some sort of descoring hood.

More Efficient Middle scoring mechanism

- We want to have some sort of mechanism to store balls. With teams scoring around 6-7 balls at once, we want to be able to hold a similar amount of balls to clear matchloader tubes.



Goal

After the competition we found that the bot couldn't park quickly enough in the end game so we need to develop a piece to help it with parking.

Solution ideas

We could make a plastic piece that mounts to the side of the drivetrain and slants the edges to make parking simpler. Or we could raise the drive train a bit so the wheels stick out and allow us to drive over the parking space barriers. We went with the first option due to ease of implementation.

Solution

1.

First we made a blueprint of the plastic piece, then we used out school laser cutter to cut it out.
2.

The we drilled some holes into it and attached it onto our bot via various nails and nuts on the side of the drivetrain.
3.

We had to cut a bit off the bottom so it didn't touch the ground. This was to prevent small particles on the ground from getting caught in between the wheels and the plastic and jamming the wheels.
4.

Then we did some testing and practiced parking.

Close up of the parking assist



Solution evaluation

Pros	Cons
- It allows us to park faster now. - We don't need maximum speed to get into the park. - It doesn't hinder the bot in any way.	- Since it's plastic and on the drive train, we think it could get damaged if we get rammed too many times in the competition. - It's far from perfect and parking is still not easy simply due to the sheer weight of our bot.

Goal

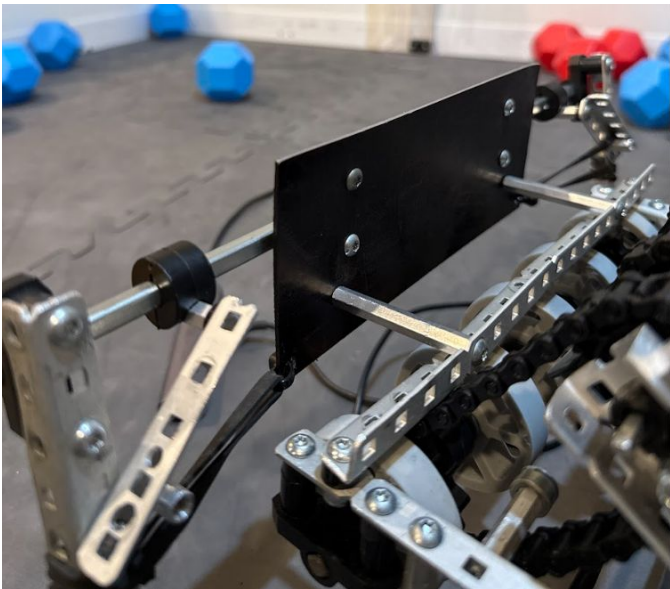
After the competition we found the match loader was to ineffective and we need to make a new one, that was faster, and could create a smoother makeshift funnel to allow the balls into the intake.

Solution ideas

We analysed what didn't work on our last design and realised it was too rigid. We need something that had flexibility. We brainstormed many designs and eventually settled on a plastic flap.

Solution

- 1. First we made a blueprint of the plastic piece, then we used out school laser cutter to cut it out.
- 2. The we drilled some holes into it and attached it onto our bot, replacing the old teeth.
- 3. To make it more stable we added some rubber bands on the tow corners closest to the bot which also allowed it to be flexible enough to form a slope while inside the match loaders and allow all the balls easy entry into the intake.



Close up of the improved match loader

Solution evaluation

Pros	Cons
- Upon testing it was very fast and allowed a smooth transfer of balls into the intake. - It's flexibility allows a very fast transfer of balls. - It's very lightweight and the whole thing can be lifted up and down in less than a second.	- It's a bit wobbly and it sometimes gets stuck on the bottom of the match loaders. However the time sit does are few and far between. - It's not as durable as the other parts of the bot and I think it might break if another bot were to meet us head on.

Goal

After we changed the design we realized we couldn't use the mesh piece and gears to move the balls along the intake anymore, so we needed to find a new way. We will identify design objectives for our intake. We have already chosen to use a mix of rubber band rollers and flaps.

Problem Statement

Compared to our prior design, we need a robot that intakes from the front and scores on the back fast. We also need to store 7-8 balls. We will keep the same drivetrain but change Intake

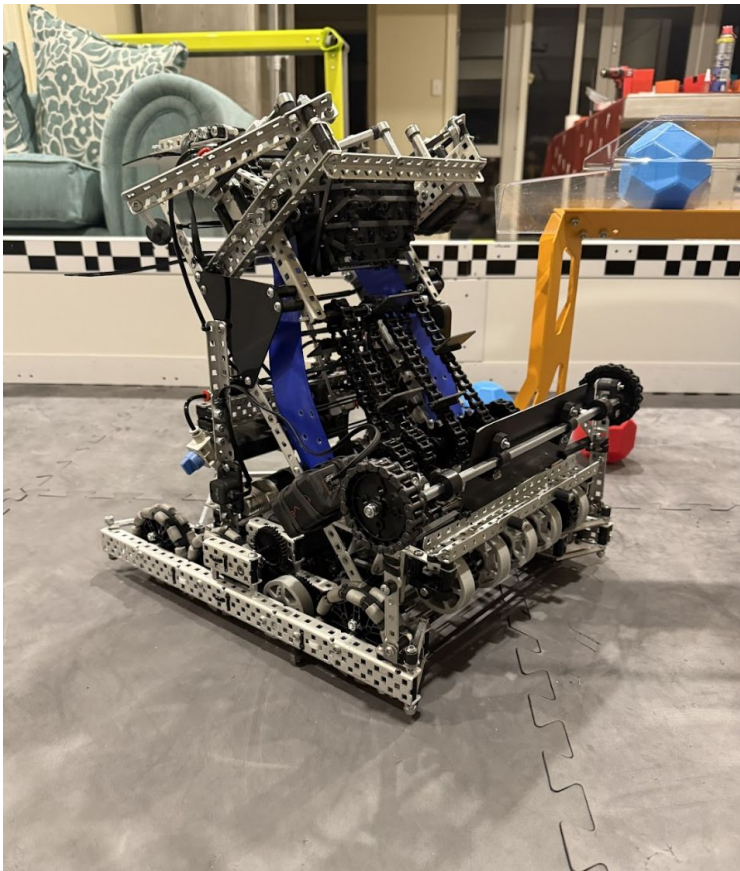
Solution Goals

The robot must be able to...

1. Store 7 balls and not jam on the band rollers

Solution Constraints

1. Must use a maximum of 22 watts
2. Must use only legal VEX parts



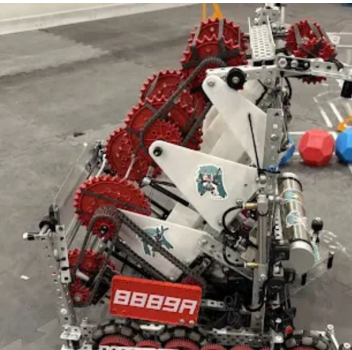
Bot 3 Testing

Goal

We will compare and contrast different types of long goal scoring designs to select the optimal one and plan our design implementation.

Possible Solutions

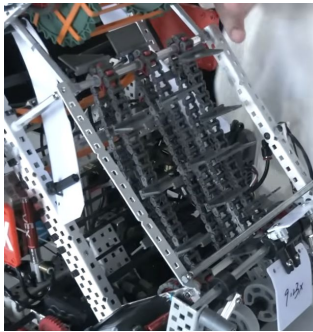
Claw - 2 sets of claws that pinch together, securing rings between them	
Pros	Cons
- Uses motor/piston power to grip rings - Versatile (can also be used to hold mogos, sweep corners, touch climb bar in auton) - Can likely fit under the long goal if tuned right	- Requires using an extra motor/piston - Claw must deploy onto the ground, making it prone to impending defense - Extremely hard alignment during pressure

Full Front to back Rubber bands - Rubber band rollers from the bottom all the way to the top for scoring.	 Citation from 8889A
Pros	Cons
- Very fast and efficient - Simple implementation	- Prone to jamming as rubber bands are high friction - May overheat easily - Spacing needs to be tuned a lot

Chain with flaps

- Chain from Point A to point B with several flaps attached on. This ensures that the flaps only touch the balls occasionally

Citation: 9123X



Pros

- Jamming is less
- Flaps can be run at a very fast speed

Cons

- Scoring may not be as efficient as the flaps may occasionally slip on the balls

Goal

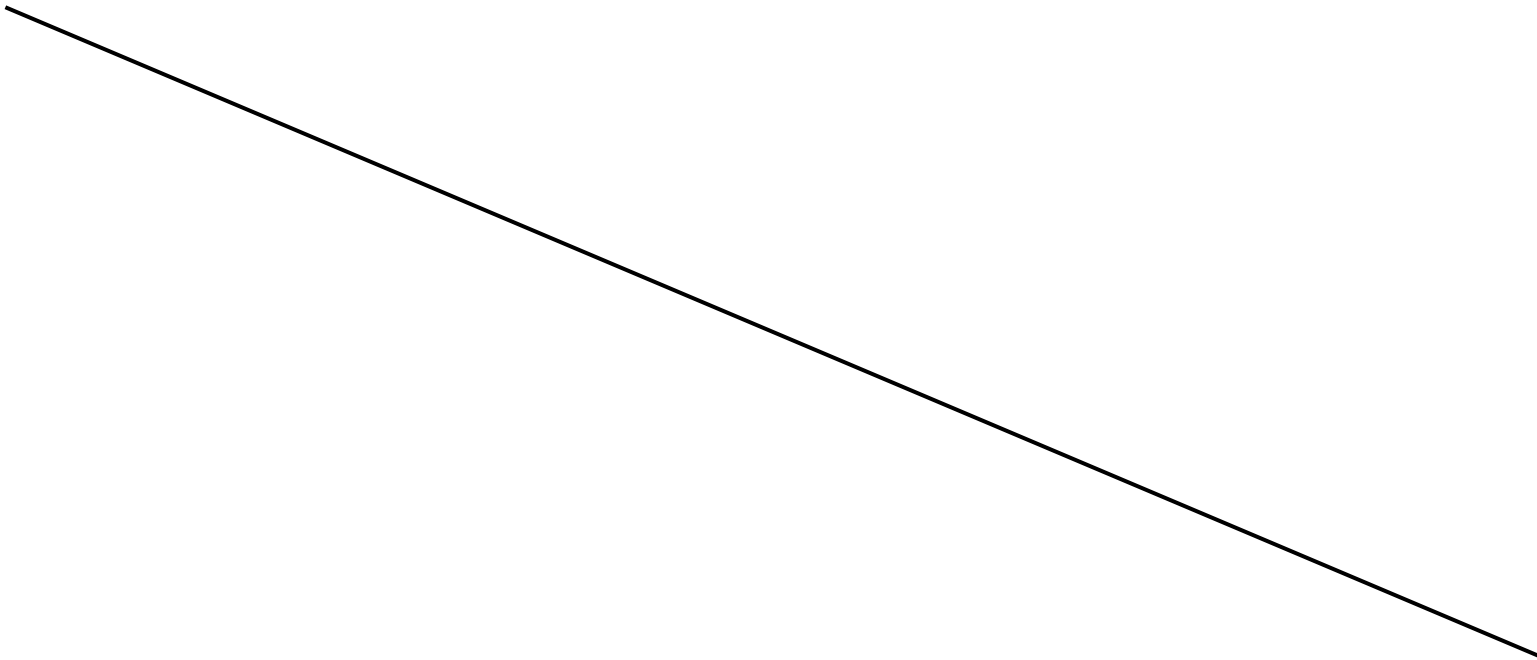
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Rubric	Claw	Rubber band Rollers	Chain with Flaps
Efficiency	1	9	7
Speed of Intaking / Outtaking (10)	1	7	9
Build Complexity (10)	6	7	10
Total (30)	8	23	26

Conclusion

Using our decision matrix, we have decided to continue with the chain with flaps style intake. As seen with team 9123X and the mall of america signature event, these flaps can efficiency score and store balls while intaking and out talking fast. The build complexity is also easier compared to methods like rubber band rollers and claw designs.



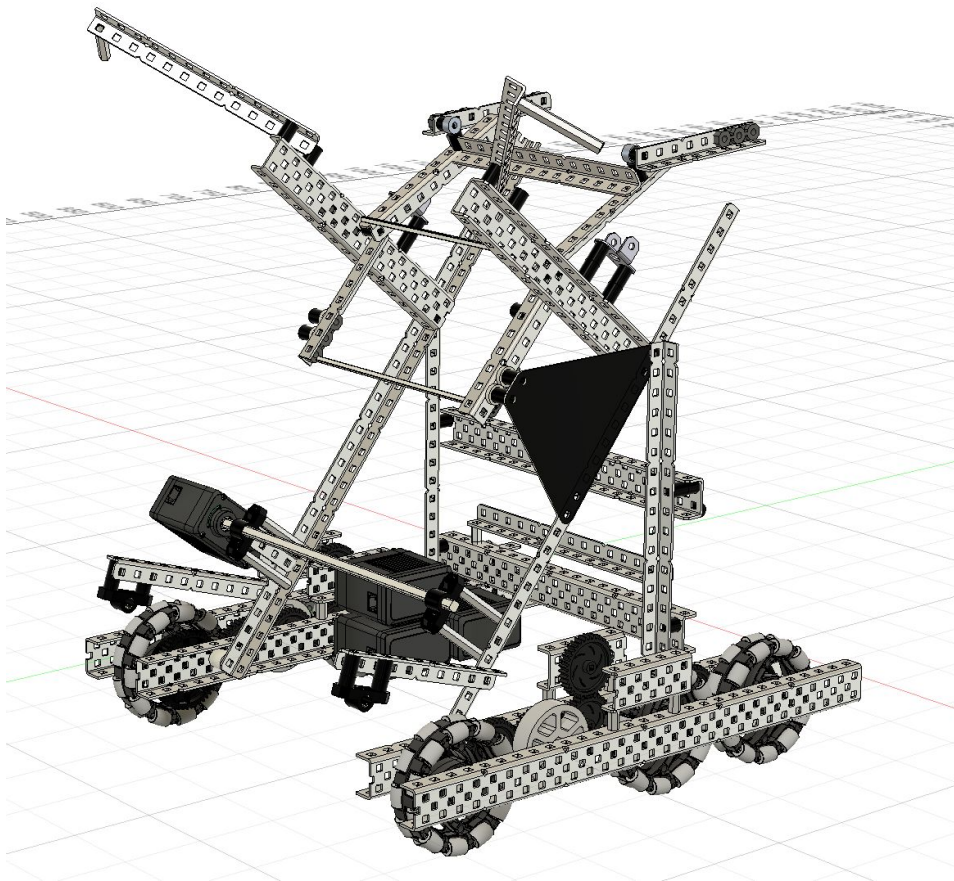
Goal

We will select our final wall stake mech design and utilize Autodesk Fusion 360 to construct a CAD model before building.

Selected Solution

- 2 11W Motors
- Flaps throughout
- Rollers for the top stage

CAD Plan



Goal

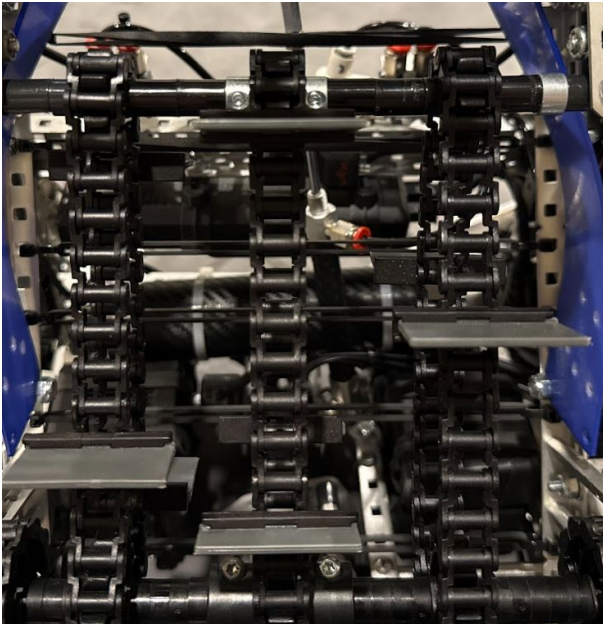
Now that we've decided on using chain and flaps, we need to implement it.

Solution ideas

We has three length of flaps, small, medium, and long. After some testing we realised that long flaps worked best but was to big for the chains on the side and got stuck. So we'll have the big flaps on the central chain and medium flaps on the side chins.

Solution

1. We begun by stringing gears on rods, secured by plastic spacers and collars.
2. We then mounted the rods on the C channels on the sides of the ball storage and mounted chains onto the gears.
3. Then we added the flaps and made them perfectly spaced so that the space between two flaps could fit only one ball.
4. Then we decided to vary the position of the flaps on the three chins to prevent clogging.



Close up of the chains on the new ball storage

Solution evaluation

Pros	Cons
- It can feed the balls along quickly and can move 4 balls at the same time, better than our old design. - The flaps on different levels help stop clogging completely. - It's more stable and durable compared to our old design and the plastic funnel pieces can double as protection.	- The chains system in connected to the new intake which is in turn connected to a single motor. If even one chain breaks the whole system goes down.

Goal

We will test our solution goals for the intake to discern if we must repeat the design process.

Solution Goals (from pg. 96)

The robot must be able to...

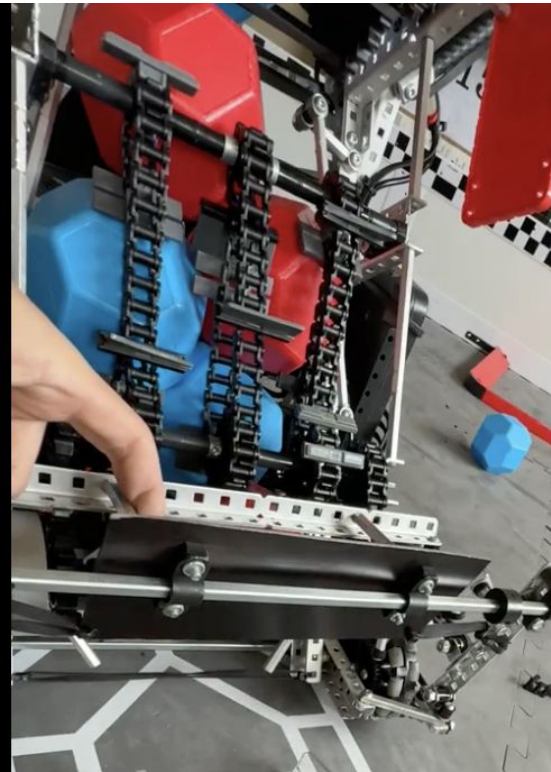
- 1. Store 7 balls and not jam on the band rollers

Test #1 - Store 7 balls and not jam on the band rollers

Test procedure:

- 1. Set up the robot stationary and flat on the field, with 7 balls in one line
- 2. Drive forward to pick up 7 balls while running the intake at maximum speed
- 3. The robot should hold 7 balls with rollers still spinning to show that it is not jamming

Trial	1	2	3	4	5
Outcome	Pass	Fail	Pass	Pass	Pass



Conclusion

Our flap intake design was able to pass our testing phase with a score of 4/5. Overall, it was very reliable with minimal jamming. We noticed that even with 7 balls in the robot, the flaps could still spin and pickup potentially an 8th ball. However, on trial 2, the robot failed to store 7 balls. This was because our battery was low and therefore power going to the motors was less

Goal

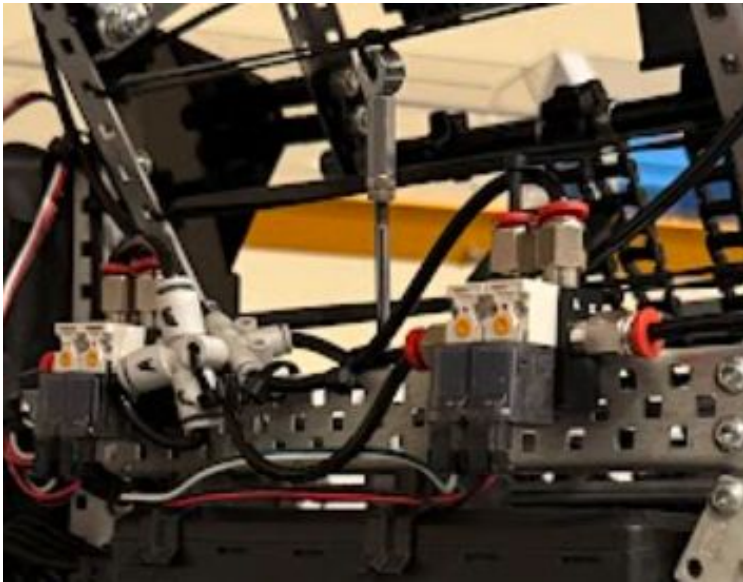
After the competition we found that the ball stopper in our old robot had some structural flaws and we wanted to improve the design.

Solution ideas

Our old design was a bit fragile and it sometimes didn't work. We think this it because it uses a rope that was pulled by a piston, if we could remove the rope and hook it up[to the piston directly, we believe all the problems would be gone.

Solution

1. We removed the rope and mounted a piston directly to the bottom of the bot.
2. To attach the rubber and to the position we used a collar and threaded a zip tie through it to attach it to the rubber band.
3. We found the no matter where we attached the piston, It couldn't pull the rubber band down enough to let the balls through.
4. We realised this was because the [iston was too short so attached a spacer in between the collar and the piston to extend it and that fixed the problem.



Close up of the improved stopper

Solution evaluation

Pros	Cons
- It improve upon the fragility of the old stopper - It now uses less air per opening closing compared to design 2 due to the direct attachment of the piston to the rubber band allowing it to only pull on the stopper while the old design had to pull on the entire structure - It's able to open a bit wider due to directly pulling on the rubber band.	- It's a bit wobbly and can be manually pushed open with enough force. We don't know if this is going to be a big problem since the balls are probably not going to have enough force to push it open, but if a bot rams into ours, it might open. -

Goal

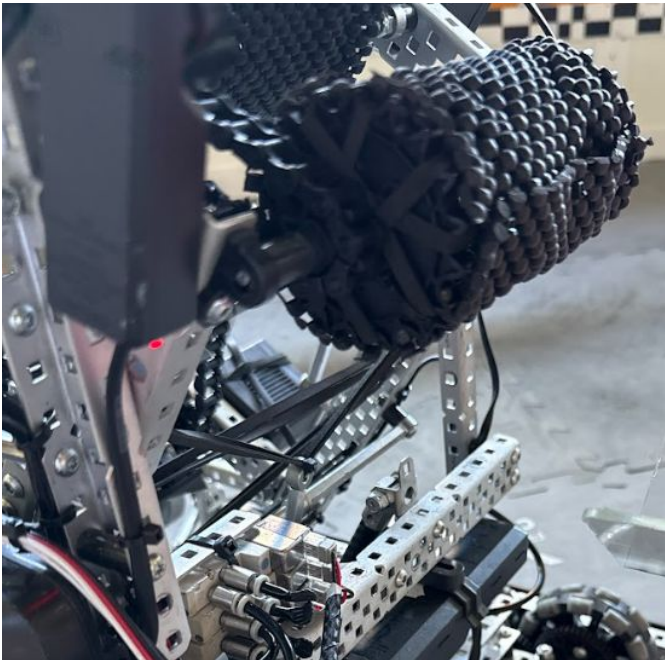
After the we finished with the new ball stopper we realised it could also double as the exit for a new middle scoring mechanism. But to do that we needed to add a roller to help guide the balls out

Solution ideas

We want a roller positioned in such a way that when the ball stopper is opened, it could guide the balls out and into the high centre goal.

Solution

1. First we mounted two big gears and a small gear onto a steel rod and secured them with various plastic spacers and collars.
2. We then put mounted the rod onto the bot amd attached the small gear to the central system via a chain to let the roller spin.
3. Then we bound numerous rubber bands in between the chain to allows it to actually move the balls.
4. After some testing we realised it didn't have enough friction so we added a mesh piece around the roller to give ti more friction



Close up of the new roller to help push the balls out of the middle scoring mechanism.

Solution evaluation

Pros	Cons
- It serves its purpose and now we have a middes scoring mechanism. - It works most of the time and we can transition between the two scoring mechanisms relatively quickly.	- It is't very accurate and we sometimes miss the goal if we don't align it well. We need to make an aliner in the future -

Goal

After the we made a scoring mechanism for the central goals, we decided to make an aligner to help us score faster and more accurately.

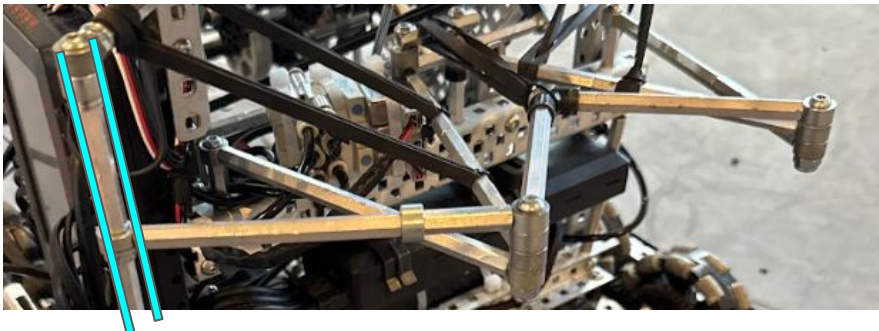
Solution ideas

We could make a plastic piece that sticks out from below the lower scoring mechanism, or we could make a structure out of metal. We chose to make a metal structure as that worked very well for the long goal alner.

Solution

- 1. To begin we didn't know what shape would work since the central goal is... kind of flat. SO we cut out various shaped in card board and tested them against the central goal to see which ones worked well. Eventually we settled on a M shaped sliner.
- 2. We made it out of metal spacers and collars according to the cardboard cut out and ,onted it onto the bot.
- 3. To ensure stability we made many supports attaching to various parts of bot including one that goes all the way down to the drive train (shown in blue) to make it very stable.

Close up of the middle aligner



Solution evaluation

Pros	Cons
- It works very well and allows us to score quickie simply by ramming the bot into the central goal and the but aligns itself. - It's very stable so there;s no fear of damage.	- It's supports are heavily linked to the design of this bot. If we ever switch to another robot we'll need to rework all the supports completely.

Goal

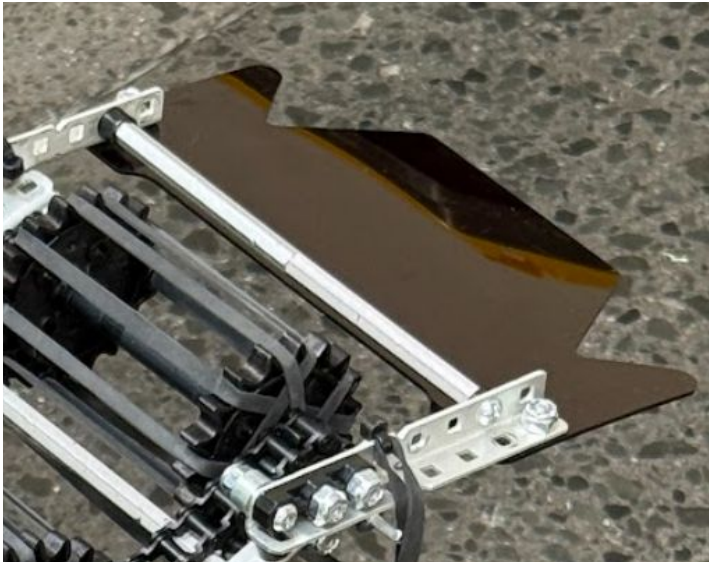
While we were working on our new bot we realised if we could make our top scoring mechanism double as a top aligner it would be of great help in the autonomous phase and when we need to score quickly.

Solution ideas

We want to make the descorering mechanism a bit wider so it can “clamp” onto the long goals and ensure the bot doesn’t move while we score. To make sure the bot still descore while also being able to score, we also need to indents on both sides so there's a length of plastic inside the goals

Solution

- 1. First we made a blueprint of the plastic piece, then we used out school laser cutter to cut it out.
- 2. The we drilled some holes into it and attached it onto our bot via two half C channels on each side.
- 3. After some testing we realised it lacked stability so we added a bar made from twlo metal standoffs to make it stable.
- 4. We also added a small roller before the descorering mechanism to help feed the balls along.



Close up of the new roller to help push the balls out of the middle scoring mechanism.

Solution evaluation

Pros	Cons
- It works very well and now we can descore 10 balls at once, basically the whole goal if we go at full speed. - It’s very stable and can withstand numerous blows. - If we have many balls it can even score from a distance by launching the balls at the goals.	- Since the alinger doesn’t move by itself and we can only score by pushing it up with balls, sometimes it doesn’t lift easily if there’s only one ball as their isn’t enough force to push it up, creating a slight clog for a few seconds.

Goal

After our descoring mechanism was completed we realised it couldn't align the bottom of the bot quickly enough since it was mounted at the very top of the bot. To mitigate this we decided to create a bottom alinger to increase stability.

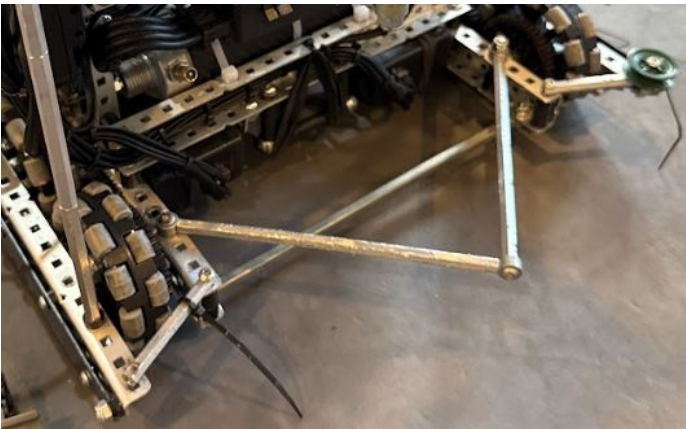
Solution ideas

Since we'd already built a good aligner for the central goal, we decided to use a similar design of metal standoffs and collars

Solution

- 1. First we made a triangle shaped aligner with stand offs and a collar then we mounted it to the bot.
- 2. It was perfect.

Solution evaluation



Close up of the middle aligner

Pros	Cons
- It works perfectly and now the bottom of the bot is secure when we score in the long goals. - It's durable and lightweight.	- The aligning is a bit inconsistent and bumpy, but this should not matter too much since were driving manually.

Goal

We will identify design objectives for our descore hook. This enables us to descore balls that are already deep inside the long goals

Problem Statement

Compared with our old C shaped robot design, we noticed that teams had much taller robots which could descore the long goals through a long hook. We realised that this would soon become META.

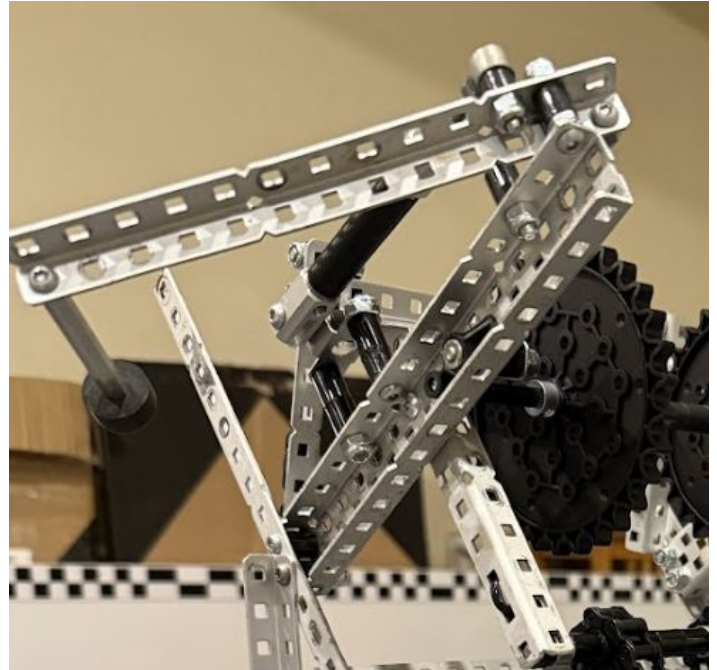
Solution Goals

The robot must be able to...

1. Enter the Long goal and de-score at least 6 balls

Solution Constraints

1. Must NOT use any more motors
2. Must use only legal VEX parts



Testing Descore Hook Solution

(cont.)

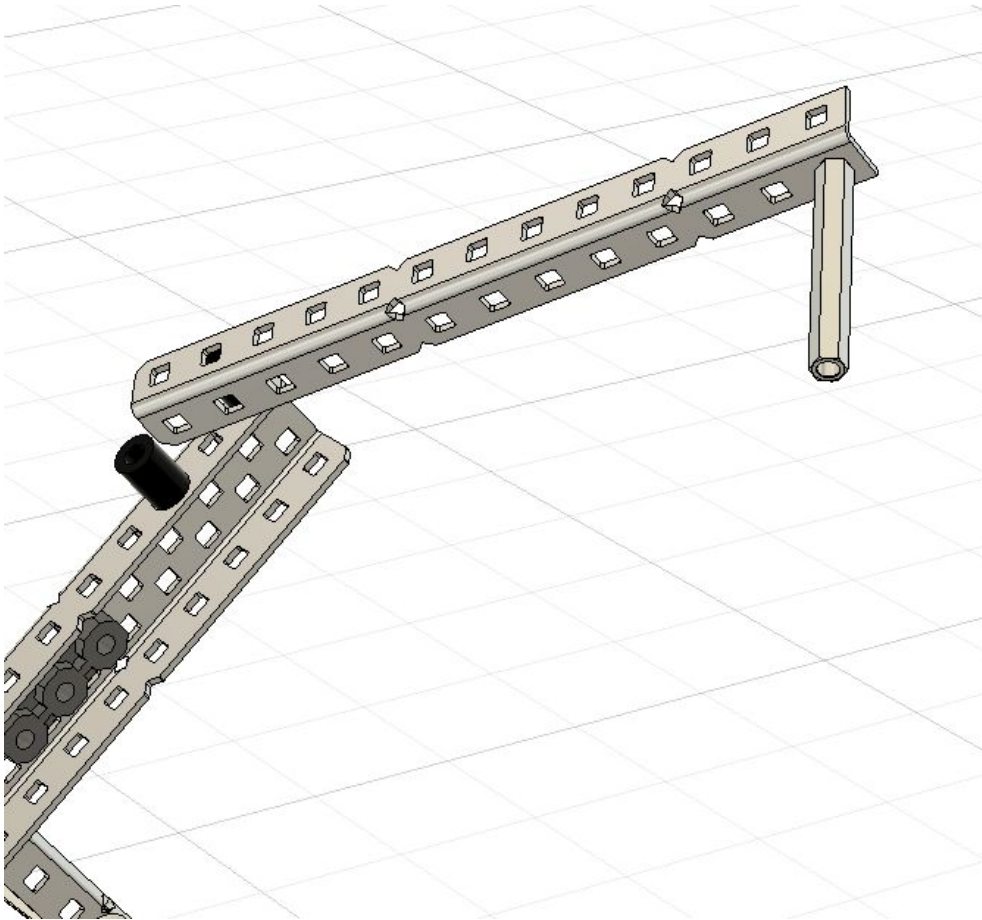
Goal

We will select our design and develop our descore hook and utilize Autodesk Fusion 360 to construct a CAD model before building.

Selected Solution

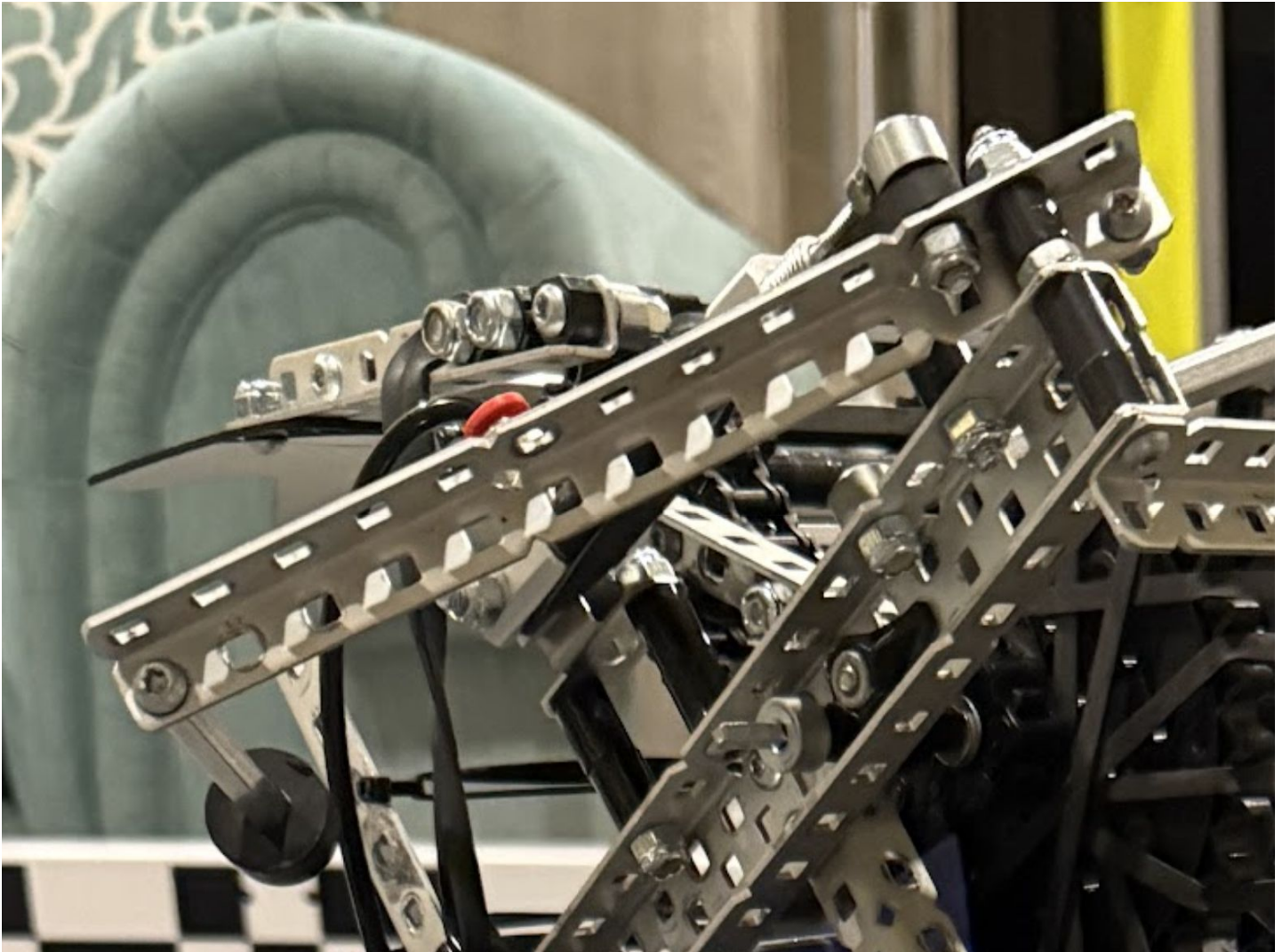
- 2x Pistons to deploy and close the descore hook
- L Channel metal
- Standoff which acts as a hook

CAD Plan



Steps

1. Cut out a 13 Hole long L-Channel (1x1)
2. Attach a 3 inch long standoff at the end of the L- Channel
3. Attach the L-Channel onto the pivot point on the robot
4. Secure it using a screw joint (Kep nut and then Locking Nut)
5. Attach a 50 mm piston onto the L-Channel so that it can open and close
6. Attach the Piston onto the frame of the robot



Goal

We will test our solution goals for the descore hook to discern if we must repeat the design process.

Solution Goals (from pg. 101)

The robot must be able to...

- 1. Descore at least 6 balls out of the long goal

Test #1 - Descore at least 6 balls out of the long goal

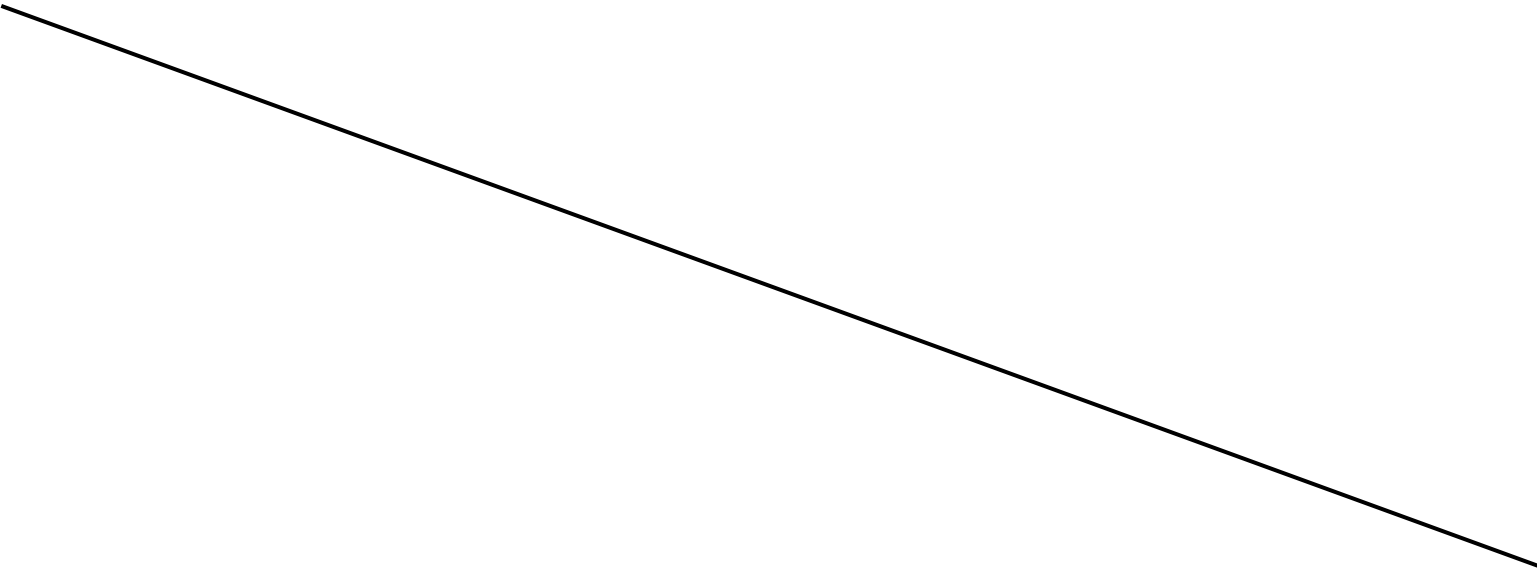
Test procedure:

- 1. Set up the robot stationary and flat on the field, placing it parallel with the long goal with 5 inches of space
- 2. Drive the robot into the long goal at full speed.
- 3. The descore hook should be able to descore more than 6 balls

Trial	1	2	3	4	5
Outcome	Pass	Pass	Pass	Pass	Pass

Conclusion

We noticed that during testing, the descore hook was extremely powerful. Even when driving in slowly, the hook was able to descore a lot of balls. Overall, our descore hook design was successful.



Pseudocode

Note: the Descore Hook has 2 states but since we have the descore hook on both sides, there is a combination of 4 different states.

IF Button 1 is pressed

 Extend Piston

IF Button 1 is pressed again

 Retract Piston

IF Button 2 is pressed

 Extend Piston 2

IF Button 2 is pressed again

 Retract Piston 2

Code Blocks

```
35
36 // L1 → descore
37 bool pressedL1 = controller.get_digital(button: pros::E_CONTROLLER_DIGITAL_L1);
38 if (pressedL1 && !lastL1) {
39     descoreState = !descoreState;
40     descore.set_value(descoreState);
41 }
42 lastL1 = pressedL1;
43
44 // X → doinker
45 bool pressedX = controller.get_digital(button: pros::E_CONTROLLER_DIGITAL_X);
46 if (pressedX && !lastX) {
47     doinkerState = !doinkerState;
48     doinker.set_value(doinkerState);
49 }
50 lastX = pressedX;
51
52 // A → middle
53 bool pressedA = controller.get_digital(button: pros::E_CONTROLLER_DIGITAL_A);
54 if (pressedA && !lastA) {
55     middleState = !middleState;
56     middle.set_value(middleState);
57 }
58 lastA = pressedA;
59
60 // Y → middle descore
61 bool pressedY = controller.get_digital(button: pros::E_CONTROLLER_DIGITAL_Y);
62 if (pressedY && !lastY) {
63     middleDescoreState = !middleDescoreState;
64     middleDescore.set_value(middleDescoreState);
65 }
66 lastY = pressedY;
67
```

Explanation

In the code, when L1 is pressed, the left descore state is set to true and the descoreState is set to NotDescoreState. This essentially makes it a toggle switch. So if L1 is pressed again, it will retract. This goes on forever until the code is closed

Pseudocode

IF R1 is pressed

Spin Intake Forward

If L1 is pressed

Spin Intake Backwards

Code Block

```
void runIntake(int speed, IntakeMode mode) {
    // --- MIDDLE OVERRIDE ---
    // When middle piston is on AND intake is actually running
    if (middleState && speed != 0) {
        intakeMotor1.move(voltage: -50); // Port 2 reverse full
        intakeMotor2.move(voltage: 100); // Port 3 forward 80
        return;
    }

    // If middle is on but intake is OFF (speed == 0), stop both motors
    if (middleState && speed == 0) {
        intakeMotor1.move(voltage: 0);
        intakeMotor2.move(voltage: 0);
        return;
    }

    // Base speeds from caller
    int topSpeed = speed; // port 2
    int bottomSpeed = speed; // port 3

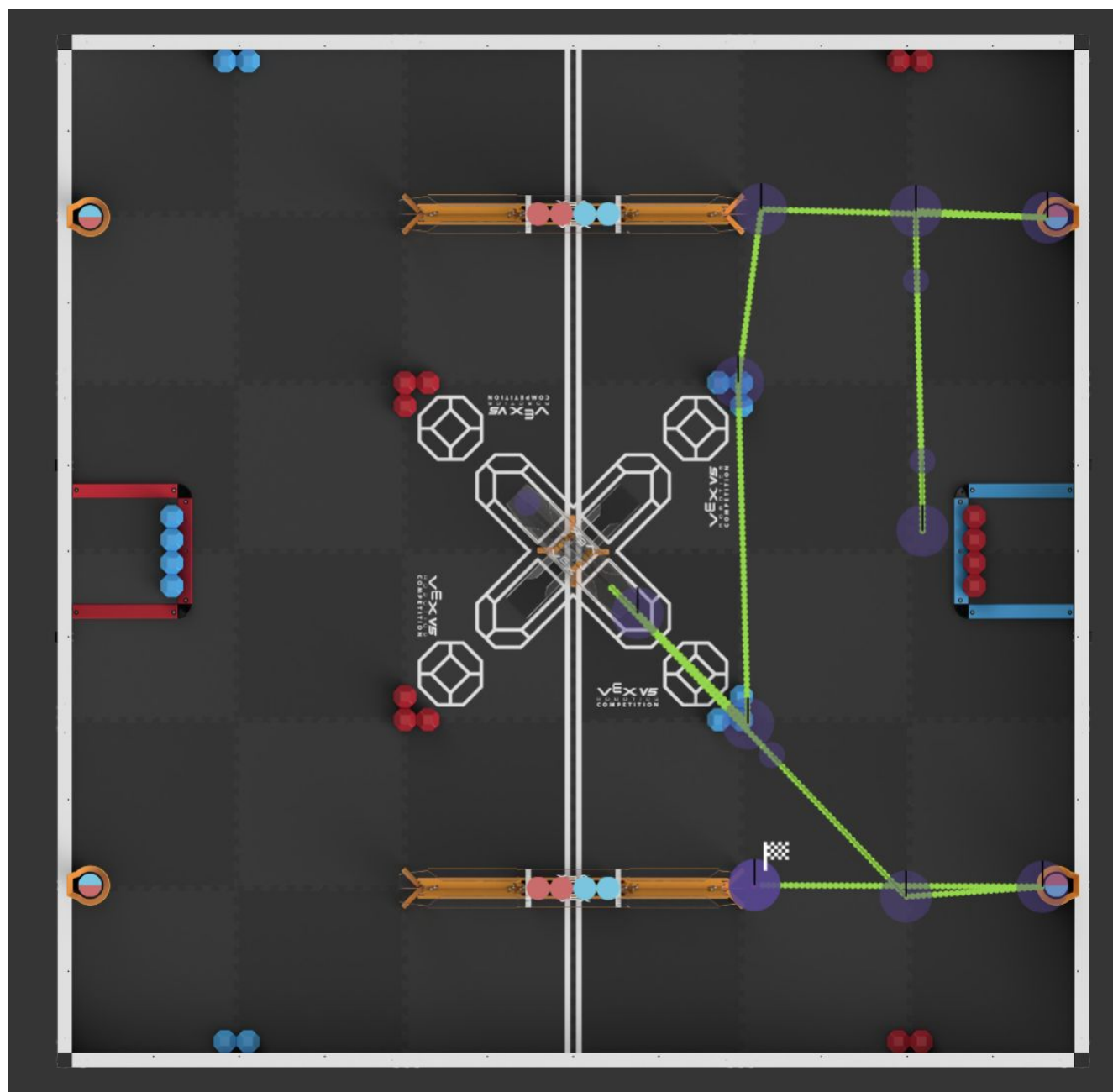
    // ---- BOTH motors mode ----
    if (mode == IntakeMode::BOTH) {
        intakeMotor1.move(voltage: topSpeed);
        intakeMotor2.move(voltage: bottomSpeed);
        return;
    }

    // ---- PORT3_ONLY mode ----
    if (speed > 0) {
        // Forward intake:
        intakeMotor2.move(voltage: bottomSpeed); // bottom full speed

        // top slow + sorting
        if (shouldStopIntake()) {
            intakeMotor1.move(voltage: 0);
        } else {
            intakeMotor1.move(voltage: 30);
        }
    } else if (speed < 0) {
        // Reverse/outtake: only bottom motor runs
        intakeMotor1.move(voltage: 0);
        intakeMotor2.move(voltage: bottomSpeed);
    } else {
        // speed == 0, stop both
        intakeMotor1.move(voltage: 0);
        intakeMotor2.move(voltage: 0);
    }
}

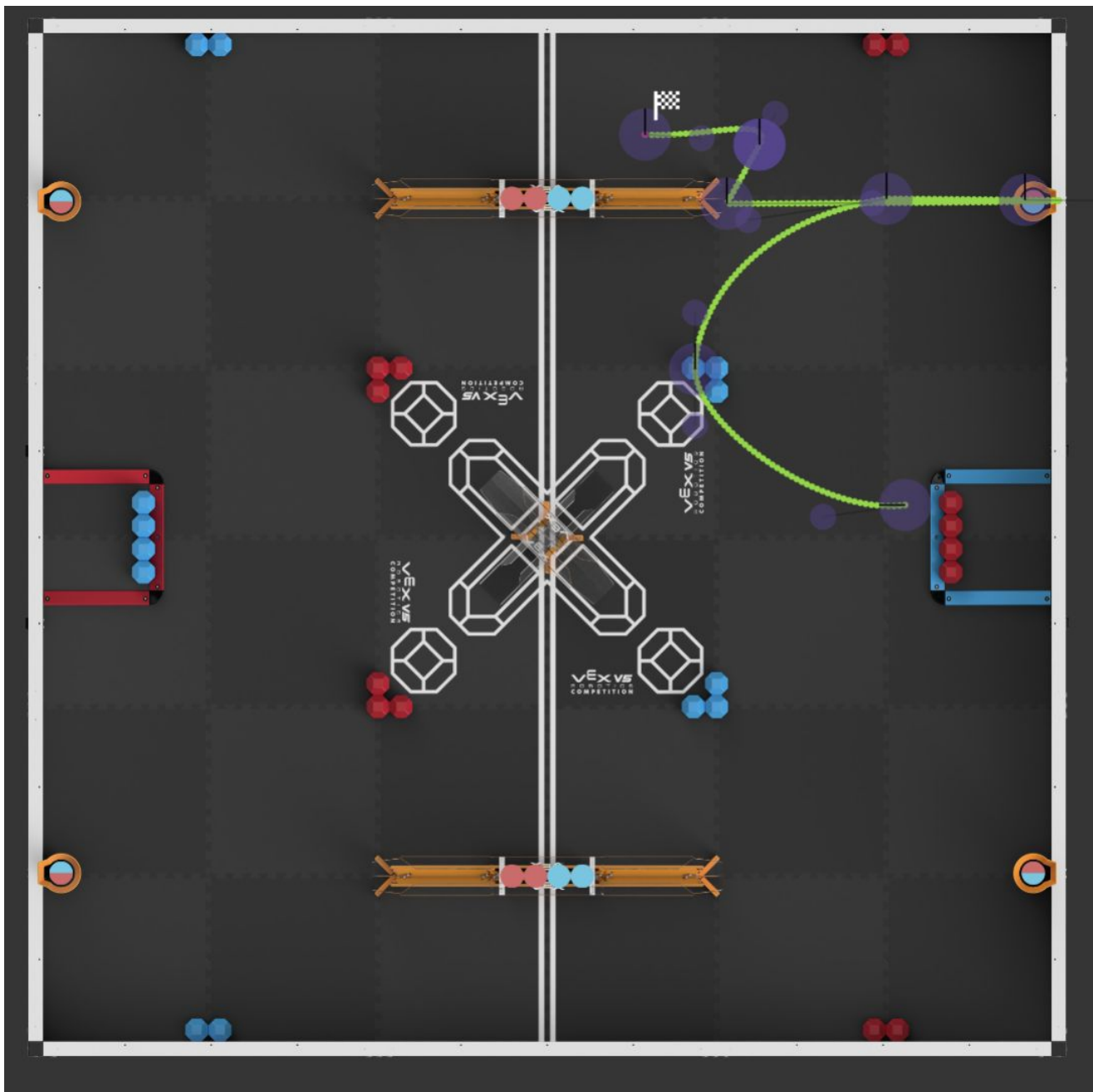
constexpr int INTAKE_REVERSE_SLOW = -100; // tweak this if you want

// Background intake task
pros::Task intakeTask(function: []-> void {
    while (true) {
        runIntake(speed: intakeSpeed, mode: currentIntakeMode);
        pros::delay(millisecods: 10);
    }
});
```



1. Go to the match loader and get 3 balls
2. Score them into the First Long goal
3. Move to middle and pickup 6 balls
4. Score 3 into Middle goal
5. Matchload 3 balls
6. Score 6 into the Long goal

If this autonomous is successful, this will guarantee the AWP point which would boost our ranking in tournaments



1. Get 3 balls from the mid
2. Go to match loader and pickup 3 balls
3. Move to long goal
4. Score 7 balls
5. Move to the side of the long goal
6. Hook the balls into the control zone

This is an extremely competitive autonomous routine that guarantees one control zone after the auto with 7 balls. This is because the wing pushes the balls deep into the tubes.

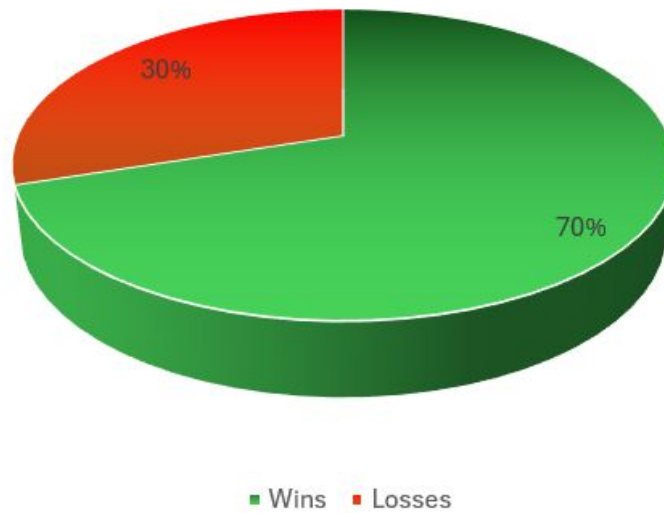
Goal

After competing in our second vex v5 tournament for the push back season using this bot, we decided to evaluate our first competition.

Results

Match	Red Alliance		Blue Alliance		Auton	AWP	Result
Q6	11044C 2915F	37	8115X 2941D	89			
Q10	8115X 2915D	0	22020Z 2941A	63			
Q17	22020C 2915D	48	8115X 2996C	32			
Q724	3168H 2941F	24	8115X 2908X	67			
Q27	8115X 2932A	86	2918X 2915G	46			
Q33	2921F 2996C	40	8115X 2915K	84			
Q40	8115X 2908X	79	2908E 68505A	26			
Q51	8115X 2996B	48	2921R 2996A	11			
QF#1-1	8115X 22020C	86	2915G 2918X	0			
SF#1-1	8115X 22020C	83	2921F 3168H	34			
Final#	8115X 22020C	0	22020X 2923A	71			DQ

Wins vs Losses



Autonomous Code (7/10)

- Pros: We ranked higher than our last competition because our autos worked better
- Cons: There are still some inconsistencies with our odometry setup and alignment with goals

Unique Ideas

Team 22020C

Vertical Descore Hook

- Advantages: Able to go in and out of the long goals without breaking any rules
- Design: Instead of Descore Hook on angle, it is mounted vertically in line with the goal

Planned Improvements

Descore Hook

- We were called out numerous times for breaking rules because our descors hook was mounted at an angle. This meant that occasionally the hook would enter the white taped area of the control zone

Improved skills auto/Barrier cross

- Since we want to score a high skills score during nationals, we realised that our barrier cross was not very smooth and we were not able to pickup all 6 balls.

Conclusion

Given that our next competition is at the end of November, with over 28 days we have a lot of time to address the defects in our robot and tournament approach. We will spend significant time improving our programming and look for more ways to improve our existing robot design.

Context

Between developing our second robot design and competing at Auckland Competition #7, our primary strategy focused on filling up the long goal with as many balls as possible. However, after analyzing our losing match, and looking into signature events, we realised there was a meta change to securing the control zone rather than filling up the long goal with as many balls as possible. This means our descoring hook must be fully refined in order to remain competitive

Changes

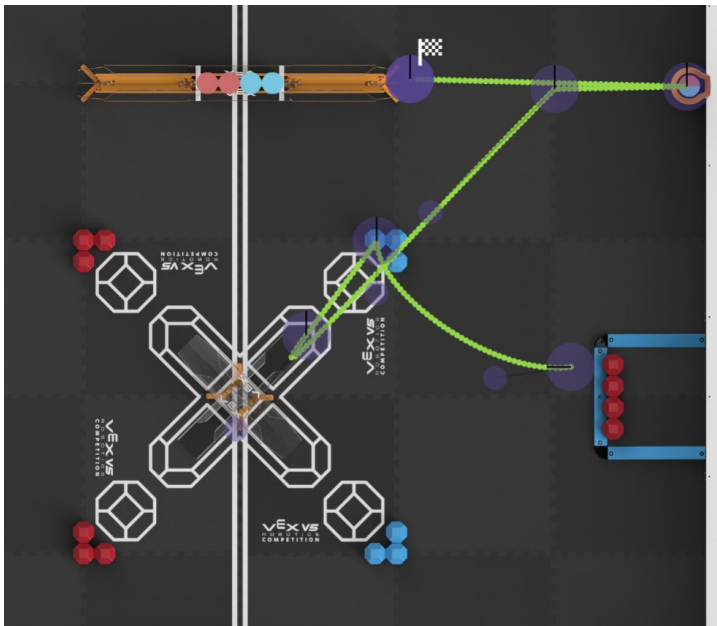
Much of our original strategy has stayed the same (anything not listed here is consistent with the original).

Our primary goal:

Win auton, secure the match: Start driver control with a strategic advantage, making it difficult for opponents to recover.

However, we will focus on **securing control zones** over **filling the long goal** as the control zones are worth more and by filling up the goals completely, teams can easily descoring using their hood or hooks.

AUTONOMOUS - Long goal + Mid



Process:

1. After picking up 3 balls from middle
2. Score 3 onto the Middle
3. Go to Match Loader and then score all onto the Long goal and use hook

To win the autonomous, the alliance who has the most control zones and most balls scored wins. If we are able to put 3 balls in the middle and Score on the long goal simultaneously in the auto, we will essentially control 2 goals at once. This will help us rank high and secure the autonomous bonus

Goal

We noticed at our competition that aligning to the long goals was extremely difficult. We want to implement a long goal aligner

Problem Statement

Without a long goal aligner, aligning to the long goals to score is extremely difficult. This can create inconsistencies.

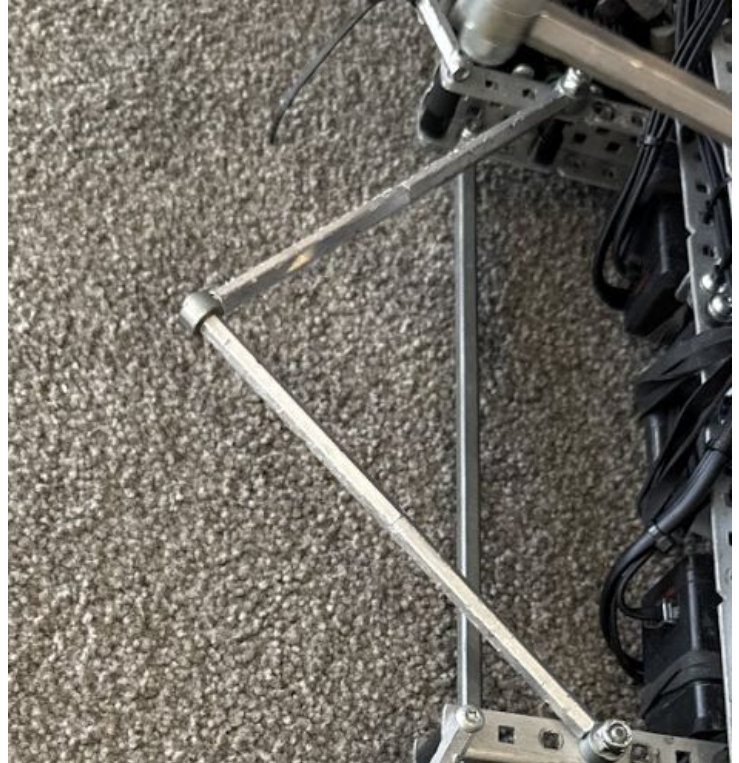
Solution Goals

The robot must be able to...

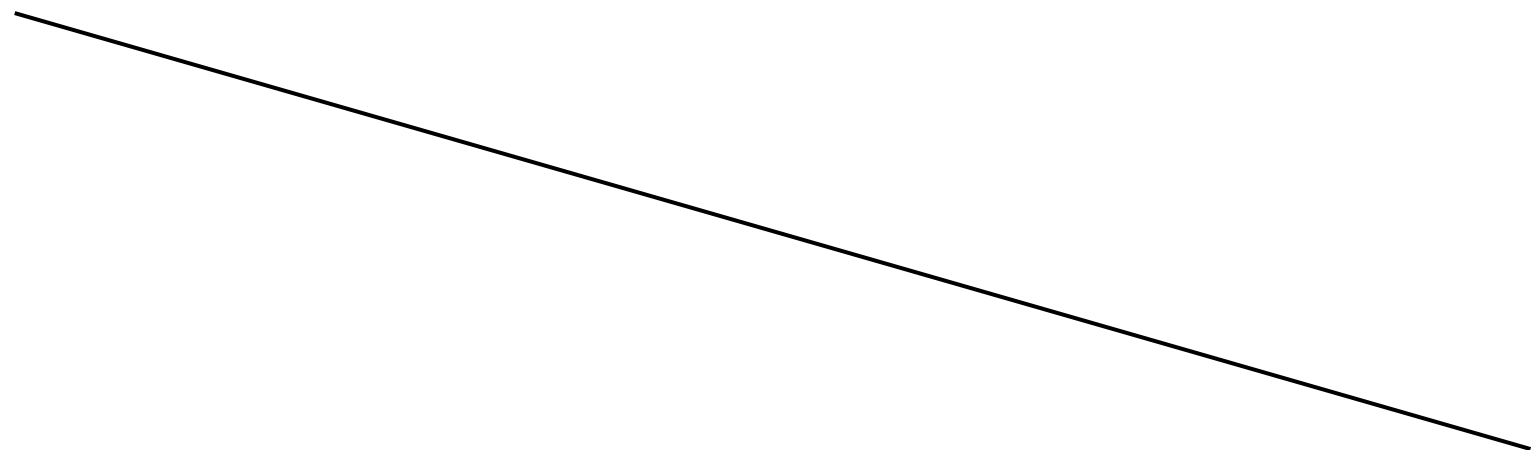
1. Auto Align to the long goal if the robot is off by 3 inches

Solution Constraints

1. Must use only legal VEX parts



Our chosen Aligner Design



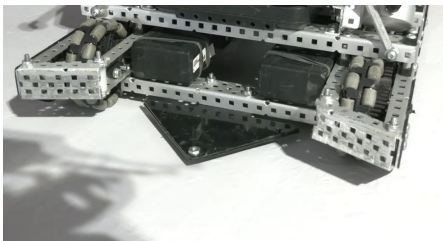
Goal

We will compare and contrast different types of designs for the long goal aligner. This aligner will help us align faster and more efficiently to the long goals

Possible Solutions

Plastic Aligner

- A Plastic Triangle Aligner which can auto align to the long goal



Citation
from 6842K

Pros

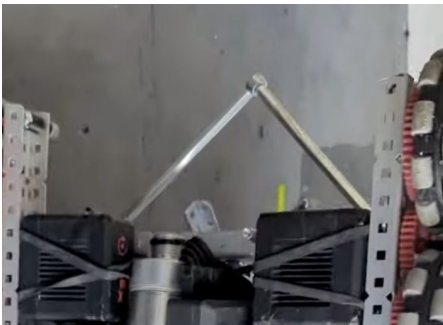
- Easy to Build (just need to CAD)
- Highly efficient as the plastic is smooth so it can align easily

Cons

- If rammed fast enough, the plastic will bend or break
- It uses another piece of plastic +1/12 limit

Metal/Standoffs

- Metal/Standoffs which are connected together in a triangle to make an aligner



Citation
from
Joseph973
on youtube

Pros

- Very strong and durable
- Not likely to bend or break

Cons

- Can be hard to build compared to a regular plastic triangle piece
- May not align fully as metal is not as smooth as plastic

Goal

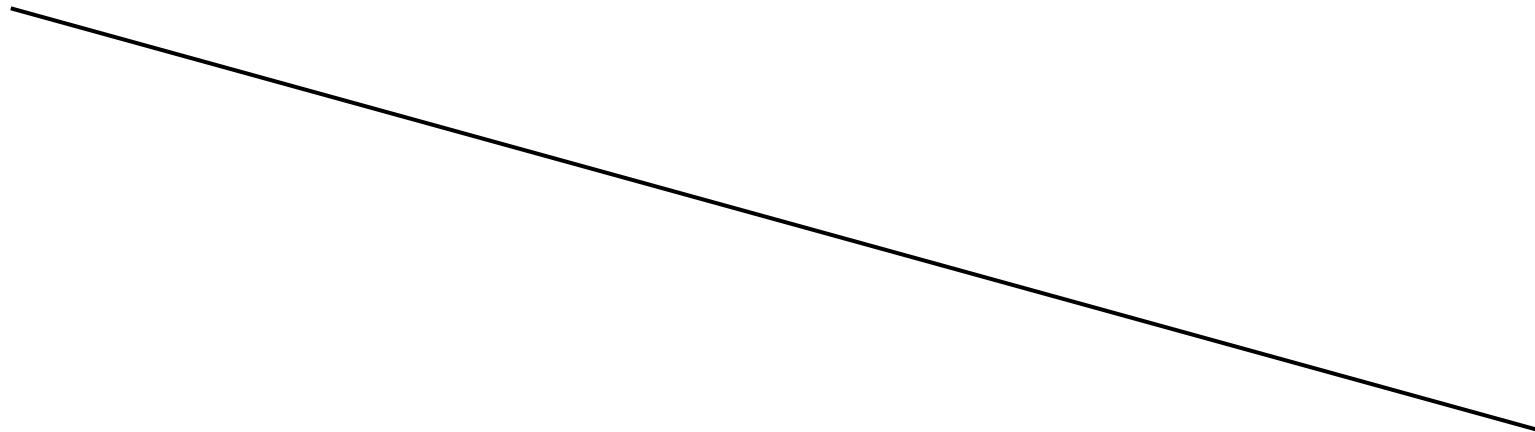
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the Aligner design.

Decision Matrix

Rubric	Plastic Aligner	Metal Aligner
Use of Material	7	9
Aligner Quality	8	7
Durability	5	10
Build Complexity (10)	6	7
Total (30)	26	33

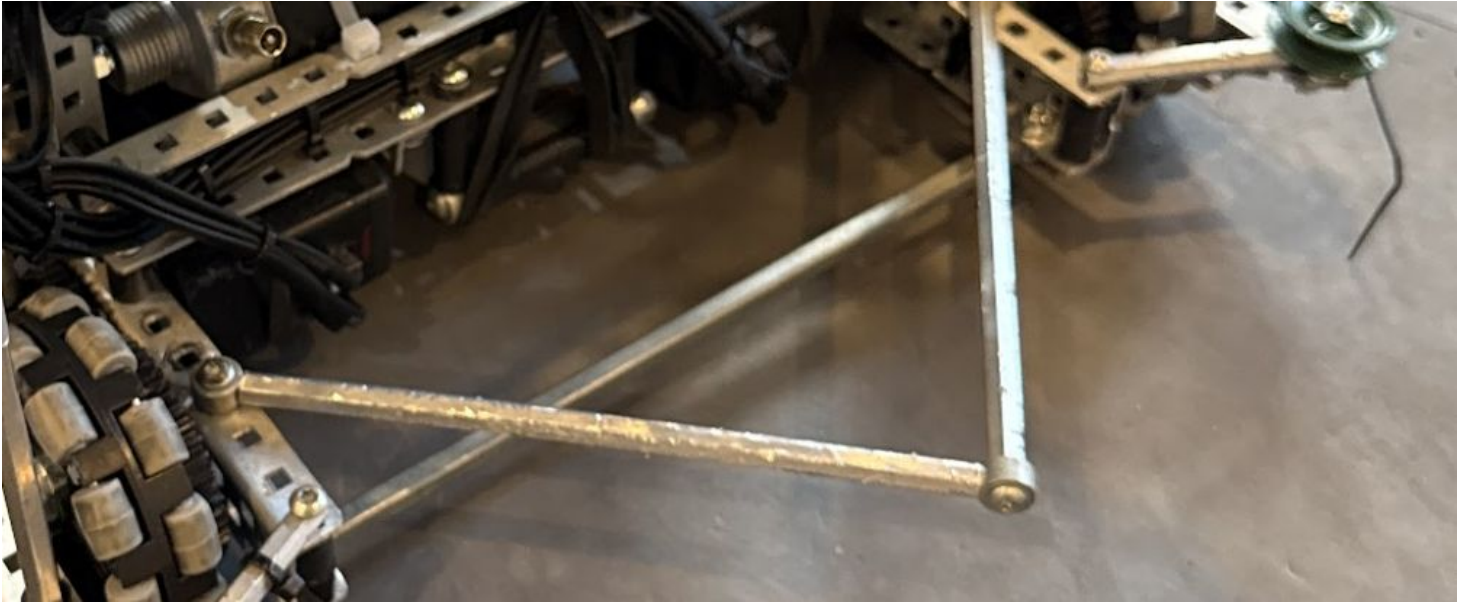
Conclusion

Using our decision matrix, we have decided to continue with the metal aligner design. This is because the durability of a metal aligner is much better than a plastic aligner. While the build complexity may be higher, we will save on 1 piece of plastic which can be used for something else.



Steps

1. Obtain 2, 6 inch long metal standoffs and a few empty shaft collars
2. Use Couplers and put them into the shaft collar
3. Screw one of the long metal standoffs into the coupler
4. Use a screw into the shaft collar to connect the second 6 inch metal standoff
5. Connect Both sides to the drivetrain metal and secure it tightly.



Context

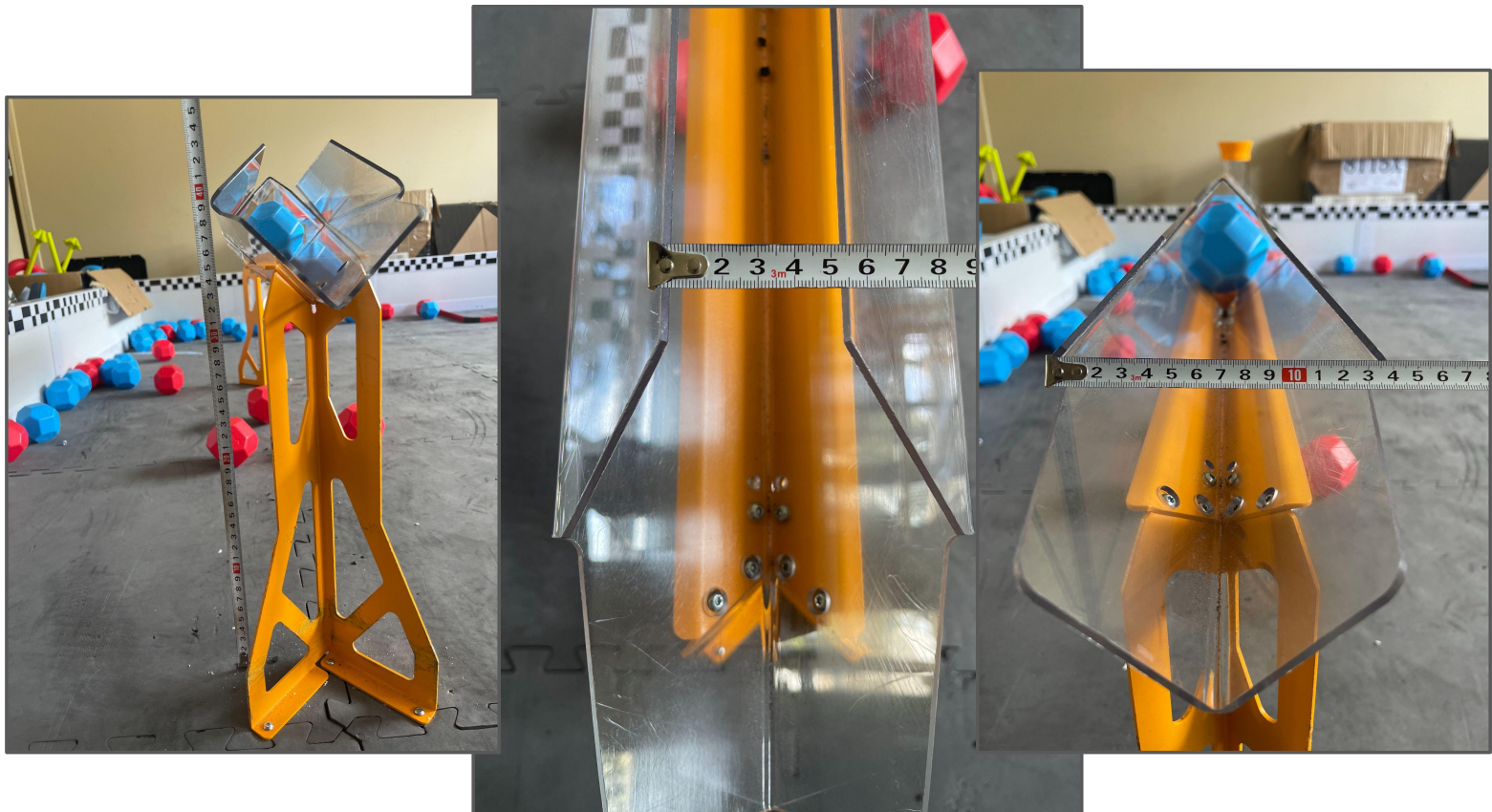
A descore wing and hook mechanism provides a strategic advantage by preventing the opposing alliance from: scoring additional balls into the long tubes, pushing other balls into the control zone of the long tubes, and descoring our balls out of the goals. Our current game strategy prioritizes securing control zones over filling the long goals, by having a efficient descore mechanism we will be able to fulfill this priority much easier during matches.

Problem Statement

Our current descore hook is mounted on a angle resulting in it occasionally reaching inside the control zone within the long goals which results in minor violations (SG10). After observing other teams, we have noticed that by simply mounting the descore wing vertically, it would eliminate this issue as well as minimising the chances that it gets stuck within the long goals. However,

Measurements

To ensure that our vertically mounted wing and hook placement is high enough to reach into the long goals while still maintaining in size, we took some measurements of the robot and the long goals.



Steps

1. Cut out a 11 Hole long L-Channel (1x1)
2. Attach a 3 inch long standoff at the end of the L- Channel
3. Attach the L-Channel onto the pivot point on the robot
4. Secure it using a screw joint (Kep nut and then Locking Nut)
5. Attach a 50 mm piston onto the L-Channel so that it can open
6. Attach the Piston onto the frame of the robot
7. Rubber band the piston to the main structure of the robot so that the wing can close



Goal

After competing in our first ever vex v5 tournament for the push back season using this bot, we decided to evaluate our first competition.

Results

Match	Red Alliance		Blue Alliance		Auton	AWP	Result
Q5	8115X 22020Z	89	8757C 22020C	26			
Q10	291F 11044C	20	8115X 2915K	88			
Q16	8115X N/A	81	2996A 7031A	40			
Q24	8115X 8757A	84	2915G 11044A	12			
Q27	8757B 22020X	27	8115X 2923A	73			
Q37	2921R 2915M	32	38115X 2915D	68			
QF#1-1	8115X 2915V	91	11044A 11044C	15			
SF#1-1	8115X 2915V	113	22020Z 2915M	14			
Final#1-1	8115X 2915V	85	8757B 22020C	17			



Goals for Robot Iteration 2

This section will discuss the list of issues that we faced with our robot at the competition.

Slow Scoring Because of Flaps

- We noticed during the competition that when we intake 7-8 balls sometimes there could be upwards of a 3 second delay before all the balls were scored. This is because flaps do not fully contact the balls at all time

Structural concerns

- After the competition, we noticed that our whole robot structure was bending and one of the outer C-channels was actually fully bent. This caused a lot o drivetrain friction and inconsistencies.

Robot Geometry Issues

- One of the biggest issues we faced with our robot utilised front to front scoring. This meant that the robot intake from the front but also scored on the front. This was a very big issue for autonomous because when we moved from the match loaders to the Long goals, the robot had to do a 180 degree turn which used a lot of time.

Design 4
Nationals Robot (Front
to Back Roller)

Goals for Robot Iteration 3

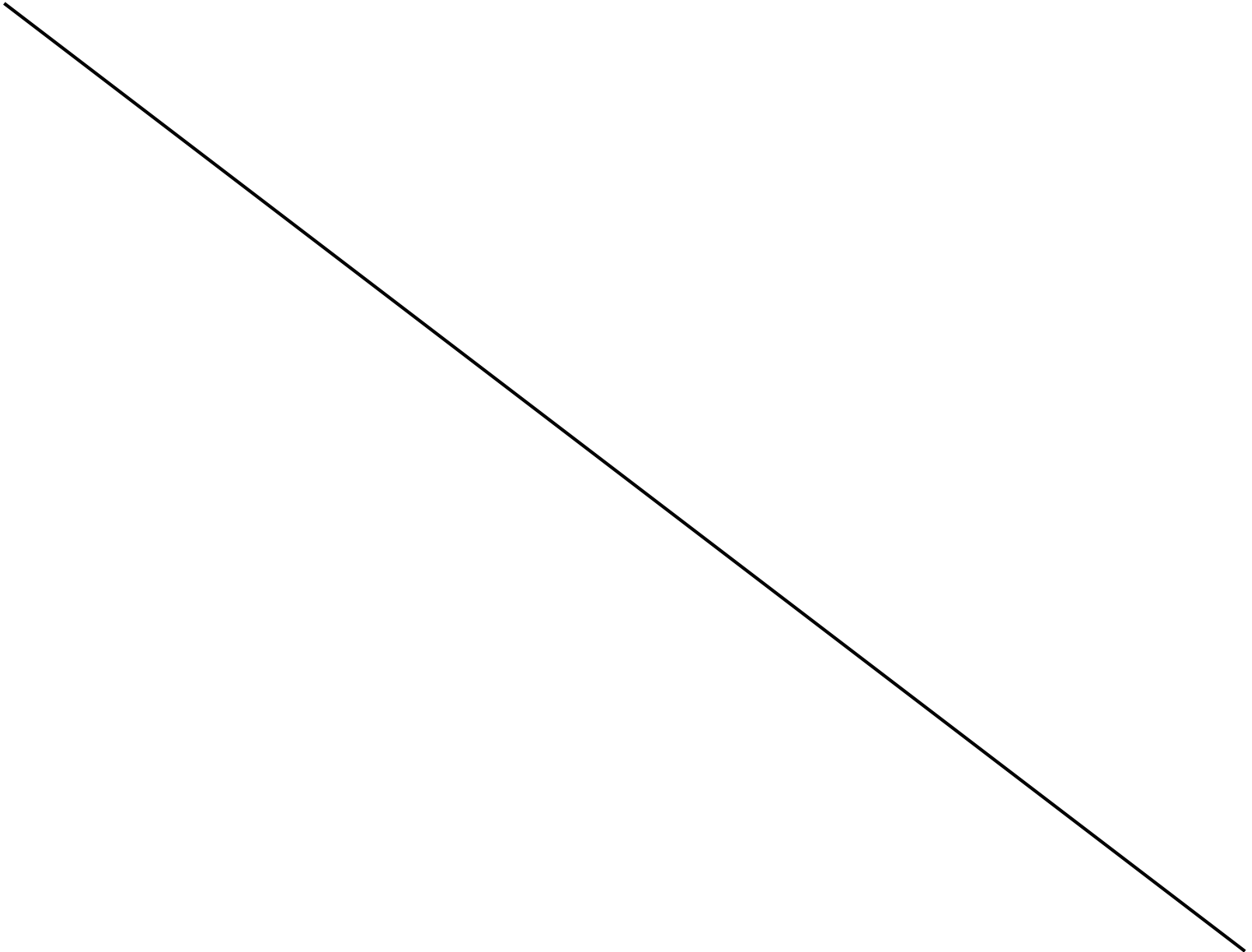
In preparation for the nationals competition, we have decided to do a full robot rebuild to fix all the issues that we faced.

7-8 Balls of Storage for intake

- In order to improve our maneuverability around the field, we wanted to build a robot that can move under the long goals. This essentially allows us to play better defense and also we can move faster to the long longs

Score Middle goal with no jamming

- We want to be able to score middle goal with o jamming. This will help ensure we get the control bonus



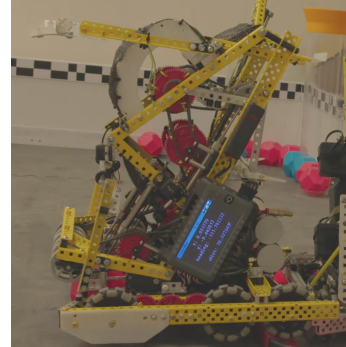
Goal

We will compare and contrast different types bot designs to select the optimal one and plan our design implementation.

Possible Solutions

S Bot

- Balls travel up in an S- shape. Balls are in 1 line



Citation
from team
16610A

Pros

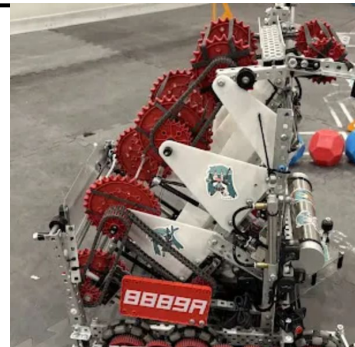
- Ball Storage is around 9 balls
- Can score on all goals efficiently

Cons

- Building is quite complex
- May be very hard to tune
- Time consuming

Ruiguan Design (Rollers or flaps)

- Rubber band rollers from the bottom all the way to the top for scoring.



Citation
from 8889A

Pros

- Very fast and efficient
- Simple implementation
- Can score Mid very fast

Cons

- Prone to jamming as rubber bands are high friction
- Bottom goal scoring sucks!

Goal

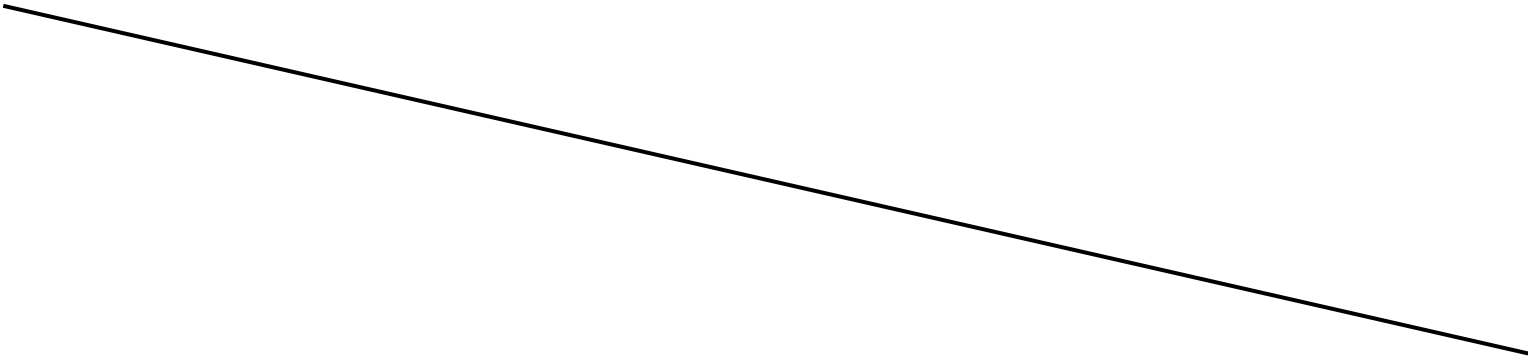
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Rubric	S Bot	Ruiguan Style Design
Scoring Speed	9	9
Ball Storage	8	7
Tunability	5	8
Build Complexity (10)	5	9
Total (30)	27	33

Conclusion

Using our decision matrix we decided to implement the front to back roller design. Since the flaps on our old robot were too slow, we decided to try using flaps again but tune them better. If it doesn't work we will switch and try mesh rollers. This design is also much simpler to tune and the scoring speed is on par with a well tuned S-Bot. The only downside is the bottom goal scoring capabilities since the robot fits 2 balls side by side, it cannot efficiently score on the bottom



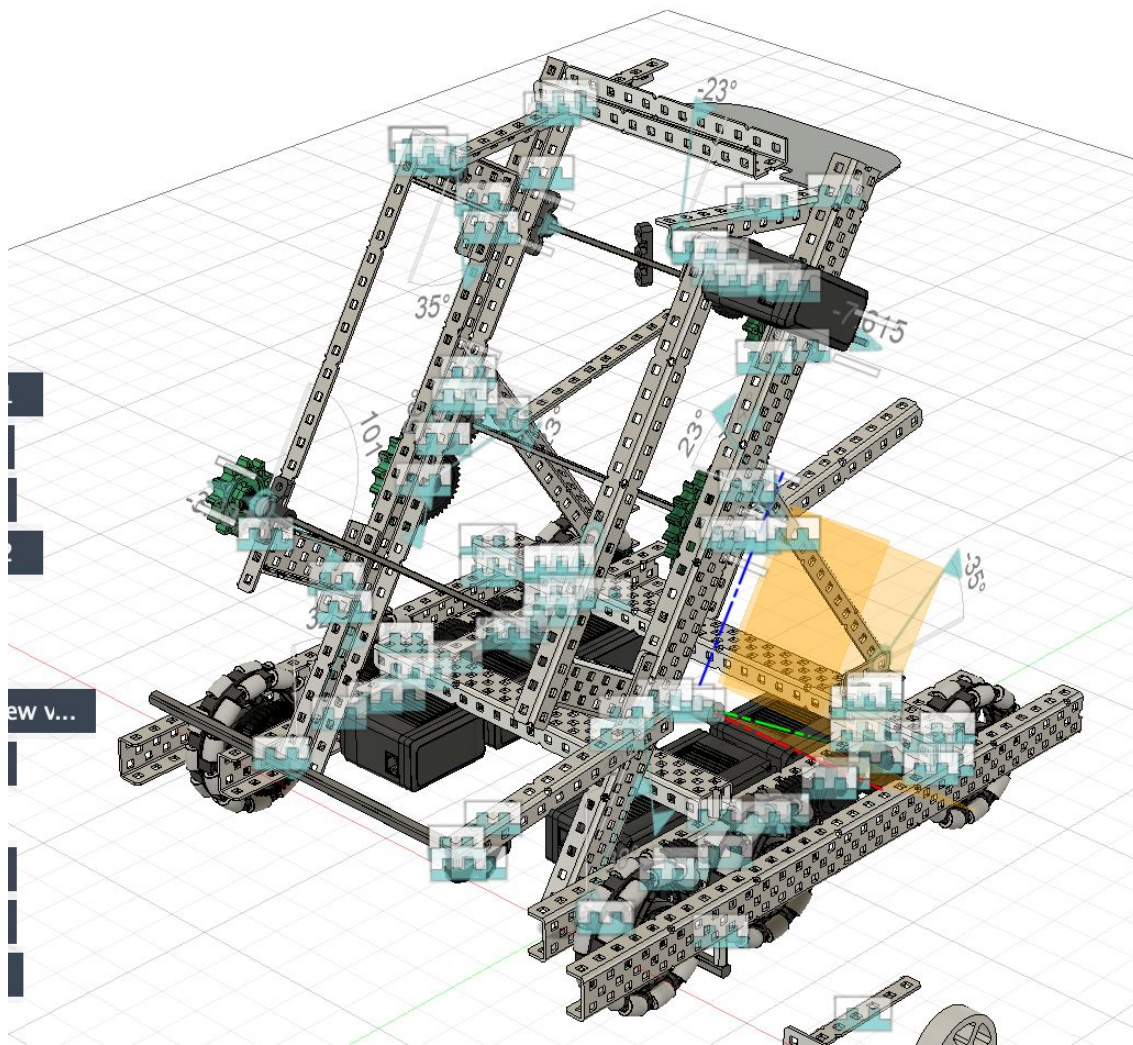
Goal

We will select our final wall stake mech design and utilize Autodesk Fusion 360 to construct a CAD model before building.

Selected Solution

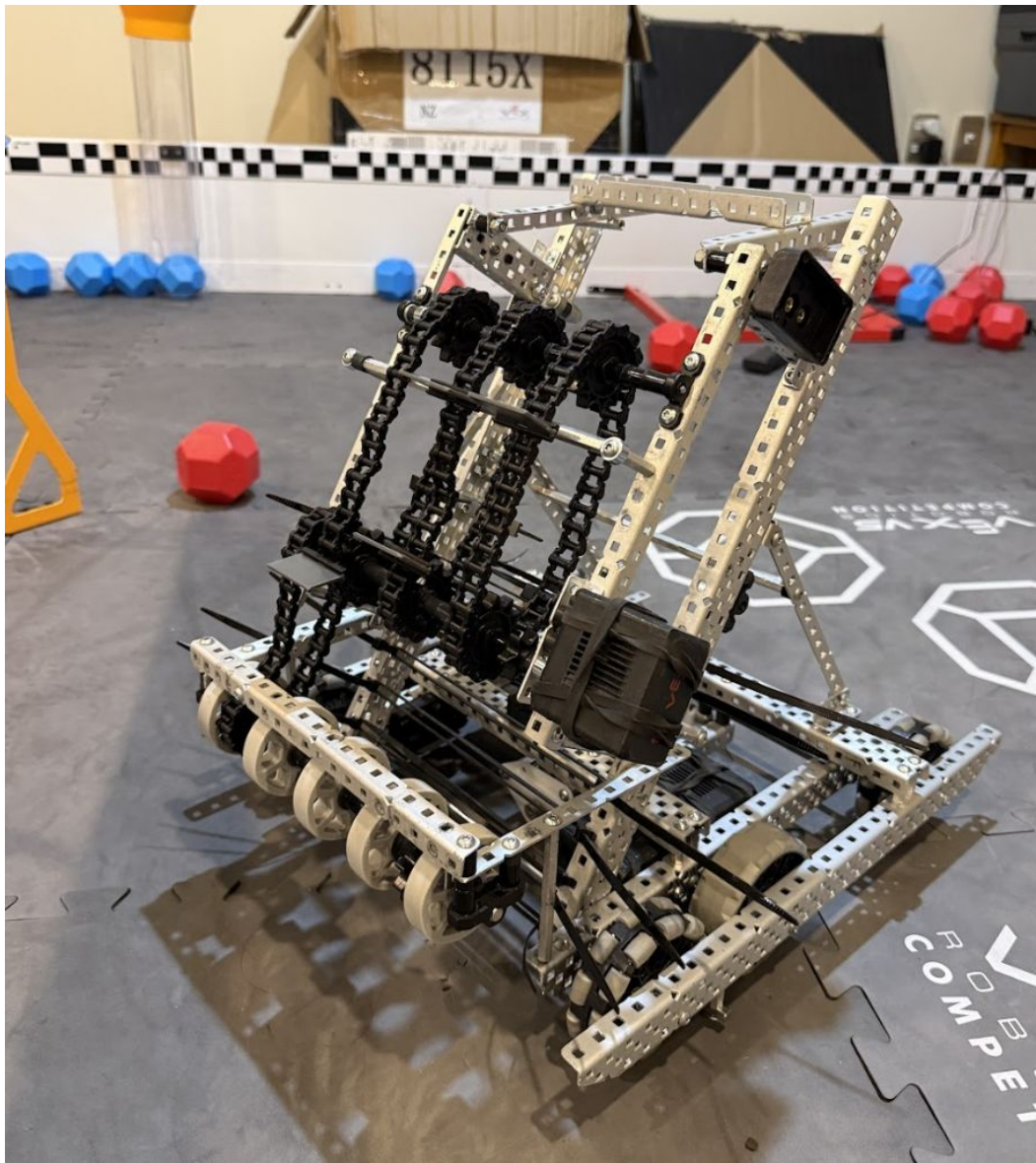
- 2 11W Motors
- Rollers for the top stage
- Rollers Throughout the robot

CAD Plan



Steps

1. After building the general structure of the robot according to the CAD
2. Attach the chain onto the robot to allow balls to travel up
3. Connect intake motor 1 and chain it to the flex wheel stage
4. Put chain from the bottom sprockets to the top sprockets
5. Add 2-3 Flaps onto each chain to prevent too much jamming from too many flaps
6. Tune the robot



Goal

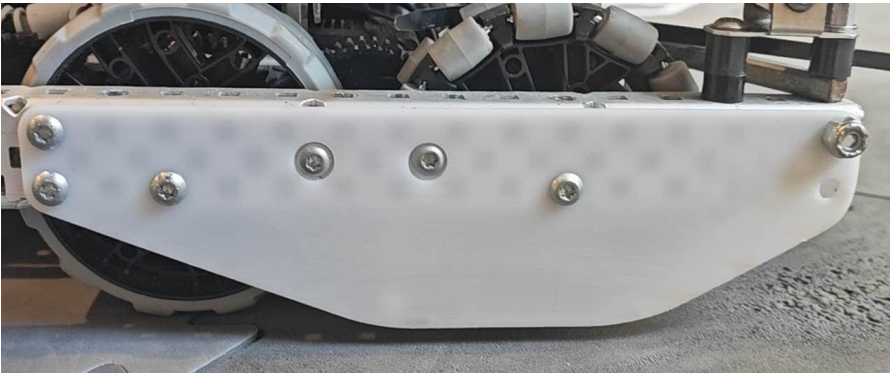
Sometime around our last competition the rules were changed so we can't have a plastic piece longer than 20 cm, so we need to change the design of the plastic piece to fit the rules.

Solution ideas

Since we can't have one too long, we'll need to make a new one that's much smaller. We decided to make some modifications while we're at it. We'll make the piece more slanted and make it taller so it's easier to enter the parking space

Solution

- 1. First we made a blueprint of the plastic pieces, then we used out school laser cutter to cut them out.
- 2. The we drilled some holes into them and attached them onto the front of the drivetrain.



Close up of the new

Solution evaluation

Pros	Cons
- It makes parking much easier since the new slant allows a much smoother assent.	- It only allows us to enter the parking space from one side of the bot.

Goal

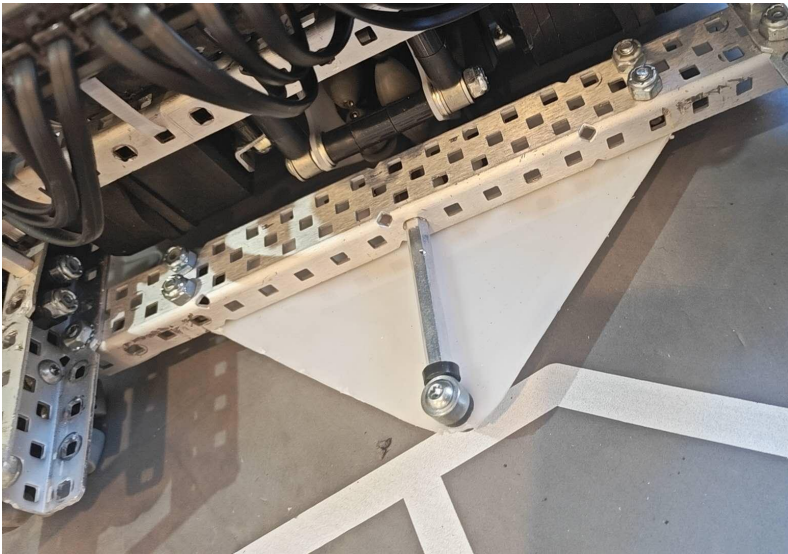
After some competitions with our last bot we realised that the aligning with the long goals was a bit bumpy and inconsistent, so we decided to make the aligner smoother.

Solution ideas

The main thing with the metal aligner was it was too rigid and it has a big bump on the end, this made aligning inconsistent and sometimes lead to slight misalignments in the autonomous phase. TO mitigate this we need a smoother material: plastic. But plastic is quite fragile so we need something to re-enforce it.

Solution

- 1. First we made a blueprint of the plastic piece, then we used out school laser cutter to cut it out.
- 2. The we drilled some holes into it and attached it onto the bottom of our bot via some screws and bolts.
- 3. We added a metal stand off to take the brunt of the force when it was aligning to protect it.



Close up of the new long goal aligner

Solution evaluation

Pros	Cons
- It's a much smoother alignment process and I believe it will solve out misalignment problem in the autonomous phase. - The stand off is able to take most of the force when aligning so the plastic piece isn't damaged.	- Even though the stand off protects the plastic piece from damage when aligning, we're not sure if it will survive getting repeadedly rammed by other bots kin the competition.

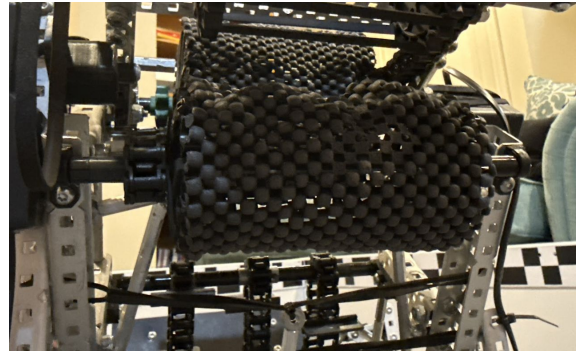
Goal

We will compare and contrast different types bot designs to select the optimal one and plan our design implementation.

Possible Solutions

Pistonised middle goal

- Piston pulls down a rubber band with roller to allow balls to score on middle. This was used in our robot design #3



Pros

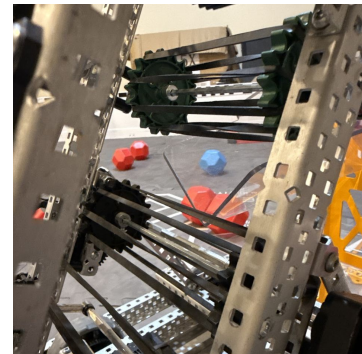
- Ball Storage is around 9 balls
- Can score on all goals efficiently

Cons

- Building is quite complex
- May be very hard to tune
- Time consuming

Double Roller Middle Scoring

- This is a new design that we came up with that allows us to score on the middle goal. Essentially instead of a piston to pull down a rubber band, there are 2 rollers no and both spin in opposite directions to allow balls to score on the middle goal



Pros

- Saves 1 piston
- Should be theoretically faster

Cons

- Could jam needs further tuning.

Goal

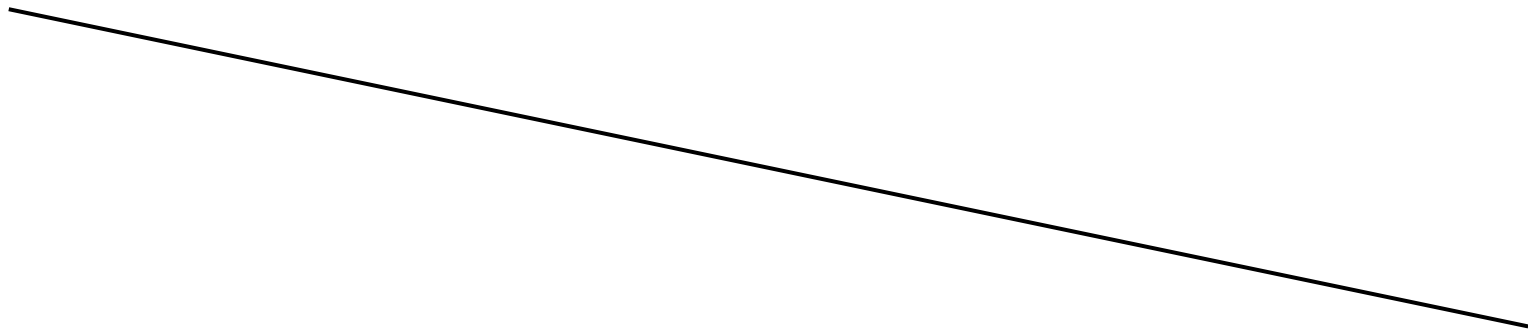
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Rubric	Pistonised Middle Scoring	Double roller
Scoring Speed	9	9
Efficiency	5	10
Tunability	5	8
Build Complexity (10)	7	5
Total (30)	26	32

Conclusion

Using our decision matrix we decided to try and implement the double roller. If it does not work we will switch back the pistonised middle scoring with the roller. This is because based on our predictions, the double roller should score faster and also save us air which be used for the match loader.



Goal

During our building process we noticed that some other teams online had a descoring mechanism for the central goals, so we realised we also needed one.

Solution ideas

Would could make a plastic piece like the long goal descoring mechanism and mount it on our bot, or we could make a metal one. We originally wanted a plastic one as it's a tried and true design, however we realized that a plastic descoring mechanism would be too fragile since we're mounting it on the front of our bot where another bot could easily ram into it and break it.

Solution

1. First we made C shaped bar uses a C channel and two half C channels.
2. Then we mounted it below our lower scoring mechanism and added pistons so it would go up and down, similar to the original long goal descoring mechanism.
3. Then we made a bump that protrudes from the front of the bar to allow it to extend into the central goals and descore the balls.



Close up of the new roller to help push the balls out of the middle scoring mechanism.

Solution evaluation

Pros	Cons
- It works very well and now we can descore ~3 balls at once. - With the positions it can go up and down without hindering the scoring.	- It's implementation is highly dependent on the structure of the bot so if we were ever to make any structural changes to the bot we'd have to change the design of this descoring mechanism to match.

Goal

We will test our solution goals for the intake to discern if we must repeat the design process.

Solution Goals (from pg. 124)

The robot must be able to...

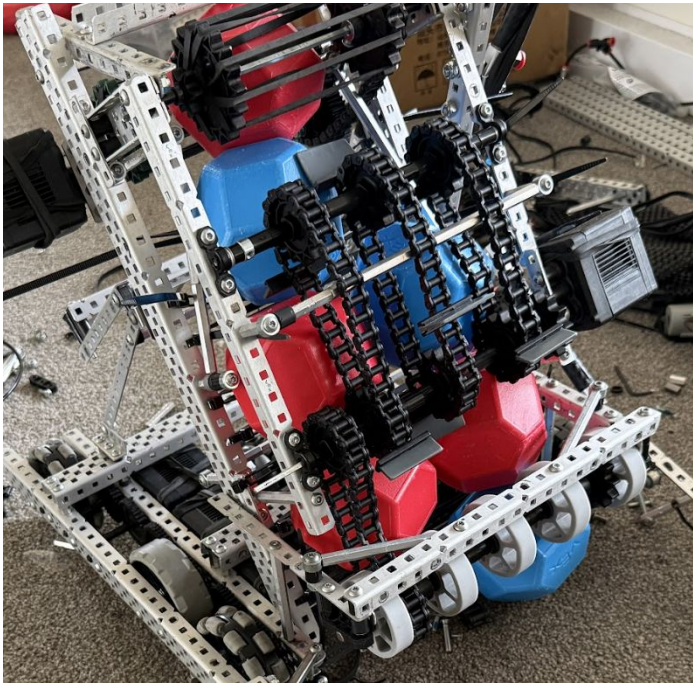
- 1. Store 7-8 balls and not jam

Test #1 - Store 8 balls and not jam on the band rollers

Test procedure:

- 1. Set up the robot stationary and flat on the field, with 7 balls in one line
- 2. Drive forward to pick up 7 balls while running the intake at maximum speed
- 3. The robot should hold 7 balls with rollers still spinning to show that it is not jamming

Trial	1	2	3	4	5
Outcome	Fail	Fail	Pass	Pass	Fail



Conclusion

Using out flap design from pg 129, we found that the robot was not very successful in holding 7-8 balls. We frequently jammed when just 7 balls were in the robot. This was a significant problem because if we weren't able to hold 7 balls, we cannot get the middle control bonus in driver skills. This means we cannot use the flaps to store balls, we must find another more efficient design.

Goals

Goal

We will test our solution goals for the intake to discern if we must repeat the design process.

Solution Goals (from pg. 123)

The robot must be able to...

1. Score balls into the middle goal with no jamming

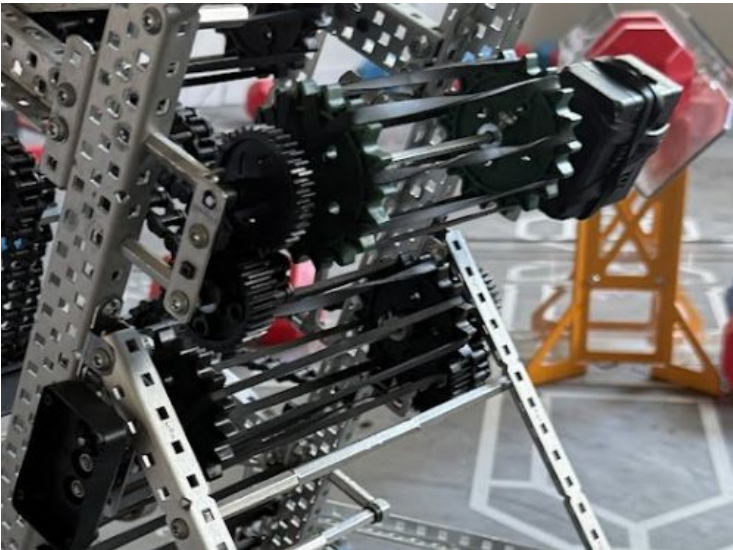
Test #1 - Score 7 balls into middle with no jamming

Test procedure:

1. Set up the robot stationary and flat on the field, with 7 balls in one line
2. Drive forward to pick up 7 balls while running the intake at maximum speed
3. The robot should hold 7 balls with rollers still spinning to show that it is not jamming.
4. Score them into the middle goal

Trial	1	2	3	4	5
Outcome	Fail	Fail	Fail	Fail	Fail

Conclusion

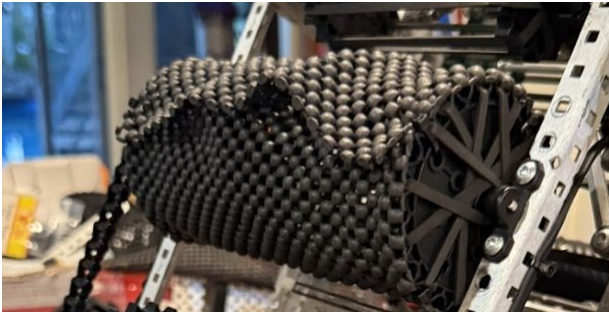


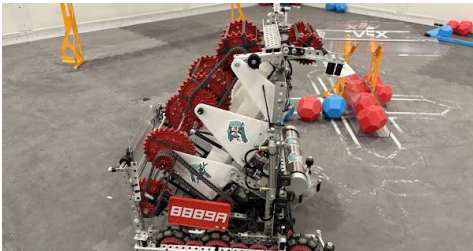
In conclusion, we realised that the double roller method for the middle goal was not great. It jammed A LOT and it caused the storage to be very bad. Also, while trying to store balls some of the balls would slip out through the gap which caused huge amounts of inconsistencies. Since we want our robot to be as consistent as possible, we must change our design to the previous Pistonised design.

Goal

We will compare and contrast different types of intake design options to prevent jamming and fulfil our goal of holding 7-8 balls.

Possible Solutions

Mesh Roller - Instead of using flaps, sprockets are used which are wrapped in mesh	
Pros	Cons
- Makes the balls travel up smooth without tensioning the bands too much - Efficient and fast	- Can be slower - Harder to work with mesh

Rubber band rollers - Rollers which have rubber bands attached to intake balls	 Citation: 8889a
Pros	Cons
- Fast and efficient scoring - Secure grip - Easy to build	- Can jam easily since the rubber bands have a lot of force when spinning

Goal

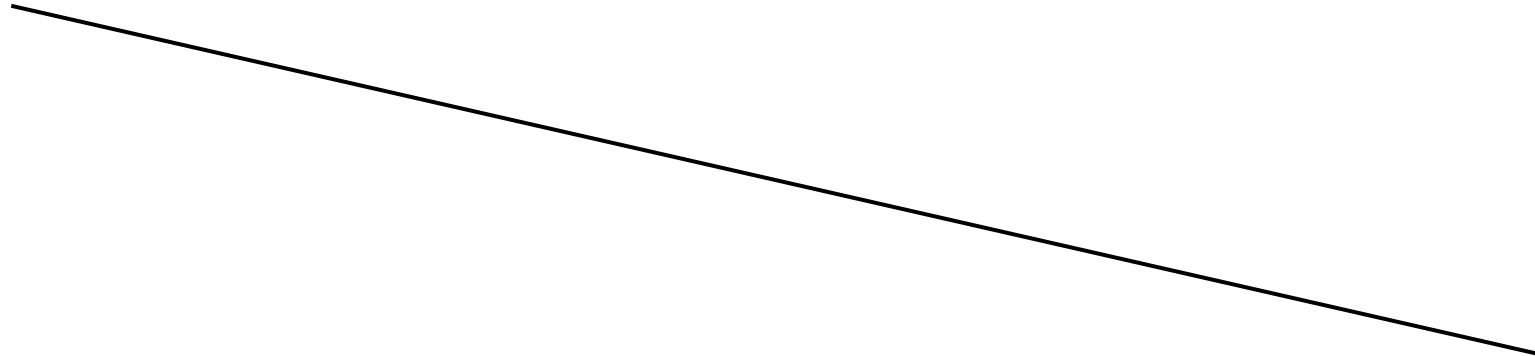
We will use a decision matrix to select the best solution from our brainstorming and move on to implementing the intake design.

Decision Matrix

Criteria List	Rubber band rollers	Mesh Roller
Scoring Speed	9	9
Ball Storage	8	7
Tunability	5	8
Build Complexity (10)	5	9
Total (30)	27	33

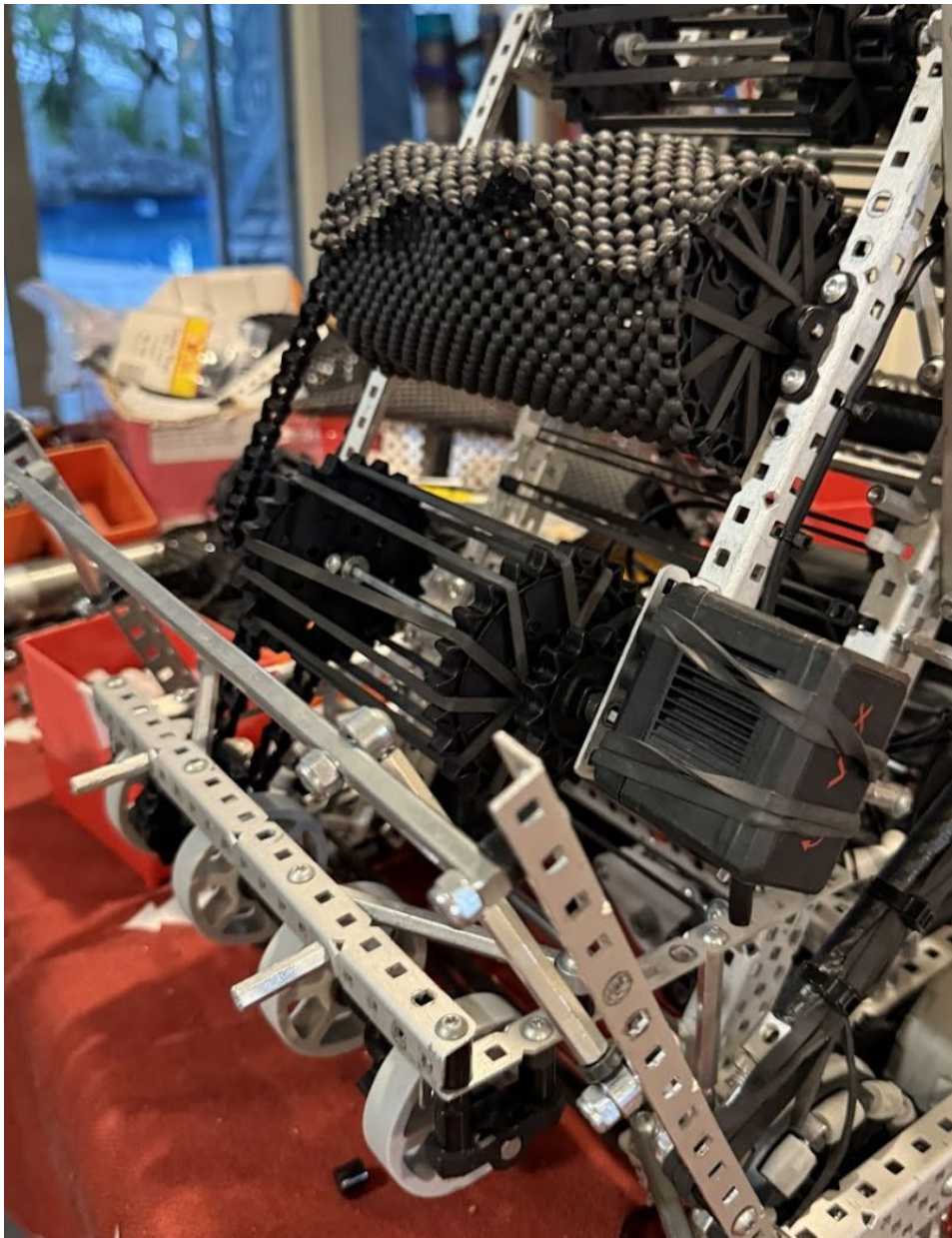
Conclusion

Using our decision matrix we decided to implement the front to back roller design. Since the flaps on our old robot were too slow, we decided to try using flaps again but tune them better. If it doesn't work we will switch and try mesh rollers. This design is also much simpler to tune and the scoring speed is on par with a well tuned S-Bot. The only downside is the bottom goal scoring capabilities since the robot fits 2 balls side by side, it cannot efficiently score on the bottom



Steps

1. Attach the big sprockets onto the robot instead of the flaps
2. Ensure that the big sprockets have enough contact with the balls
3. Start by banding the rollers. Put a band on 1 side and stretch it to the other side
4. Repeat this until every single tooth is banded
5. Then cut the mesh to fit the size of the roller
6. Attach the mesh by using 6 zip ties on either side of the rollers
7. Connect the mesh together using 3 zip ties in the middle
8. Repeat this for the bottom roller.
9. Replace all the flaps with rollers



Goals

Goal

We will now test the new mesh roller intake to see if it can fulfil the goals we set before.

Solution Goals (from pg. 123)

The robot must be able to...

1. Store 7-8 balls and not jam

Test #1 - Score 7 balls without jamming

Test procedure:

1. Set up the robot stationary and flat on the field, with 7 balls in one line
2. Drive forward to pick up 7 balls while running the intake at maximum speed
3. The robot should hold 7 balls with rollers still spinning to show that it is not jamming.

Trial	1	2	3	4	5
Outcome	Pass	Pass	Pass	Pass	Fail

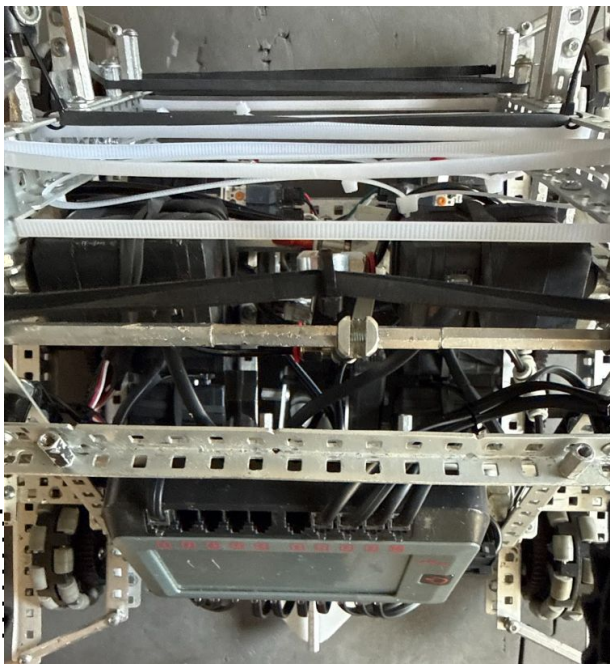
Conclusion

In conclusion, our mesh roller method was able to store 7-8 balls easily. The balls almost never got jammed in the rollers because the mesh provided a constant force upwards but not enough to jam the motors. However, we noticed on the final test that the mesh rollers sometimes provided too much force because we used rubber bands on the base of our robot. We have decided to switch to a more smooth base instead of using mesh.



Goal

Now we need to find a new material to replace the rubber bands. The main reason the rubber bands didn't work was because of its elasticity bouncing the balls around and clogging it up, so we need to prioritise rigidity, the opposite of elasticity when finding a replacement material. The new material must also prioritising smoothness and speed of travel. We had a few materials in mind.



Our new base for the bot made out of rubber bands.

Decision Matrix

Rubric	Rubber bands	Zip ties	Plastic piece	Mesh piece
rigidity	1	7	10	5
Fluidity of balls passing through	5	7	4	6
Durability	7	9	10	7
Ease of repair	10	10	3	6
Price	7	8	2	4
Build Complexity	8	8	2	5
Total (60)	38	49	31	33

Conclusion

Using our decision matrix, we have decided to use the zip tie for our chosen material as it received the highest score

Goal

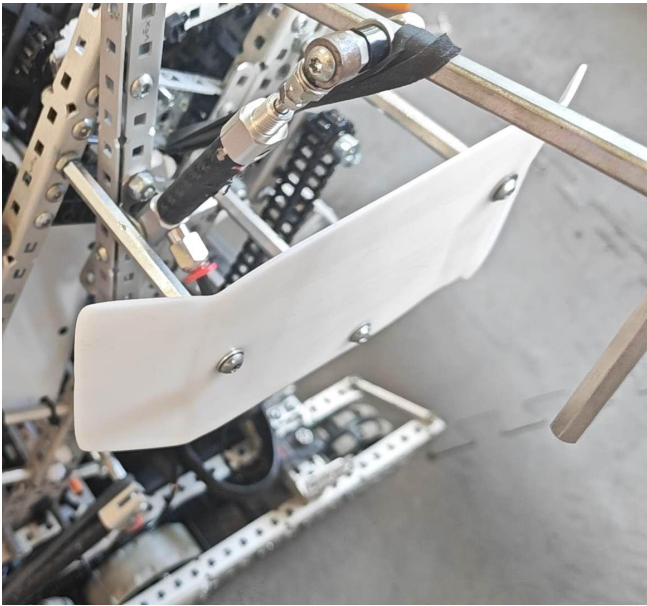
After some competitions with our last bot we realised that the aligning the descore hook with the long goals extremely difficult and we got disqualified once due to it getting struck, so we decided to make the a wing aligner to help with aligning the descore hook. We need a aligner that can align the descore hook from all angles.

Solution ideas

Our original idea was to use two pulley wheels on top of each other however that idea quickly fell apart after we realised it did nothing during initial testing. We quickie pivoted to using a big plastic piece. We'll sant the sides of the plastic piece to allow it to align the descore hook at any angle.

Solution

- 1. First we made a blueprint of the plastic piece, then we used out school laser cutter to cut it out.
- 2. The we used a heat gun to heat up the plastic and slowly bent it to createthe slant.
- 3. The we drilled some holes into it and attached it onto the bottom of the descore hook via three stand offs.



Close up of the new long goal aligner

Solution evaluation

Pros	Cons
- It fixes the alignment problem for out descore hook - It's durable and stable	- Sometimes when we ram it into the long goals too fast it gets slightly stuck for a seconds or bounces in a weird angle destabilizing the bot slightly before regaining balance. These shouldn't matter too much but could still be improved.

Goal:

We implemented LemLib, a motion control library, to enable our robot to autonomously navigate to precise coordinates on the field using odometry and PID control. LemLib handles complex path planning, allowing smooth arc motions to target points while maintaining accuracy.

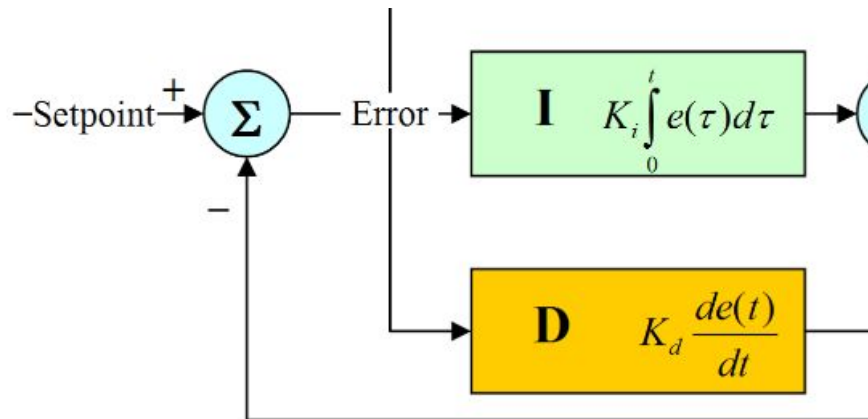
PID Control Fundamentals

PID (Proportional-Integral-Derivative) control is a feedback mechanism that calculates motor power based on error (distance from target). It consists of three components:

- **Proportional (P):** Increases power proportionally to error. Larger error = more power.
- **Integral (I):** Accounts for accumulated error over time. Prevents getting "stuck" by gradually building force.
- **Derivative (D):** Responds to rate of change in error. Slows down if approaching target quickly to prevent overshooting.

Formula: $\text{Output} = (k_P \times \text{error}) + (k_I \times \text{integral}) + (k_D \times \text{derivative})$

Example Implementation in LEMLIB:



```
lemlib::ControllerSettings lateral_controller(kP: 4, kI: 0, kD: 1, windupRange: 3, smallError: 1, smallErrorT... 100, 3, 500, 50);
lemlib::ControllerSettings angular_controller(kP: 5, kI: 0, kD: 40, windupRange: 3, smallError: 1, smallErrorT... 100, 3, 500, 0);
```

Goal:

Our autonomous routines are primarily based on Lemlib's MoveToPoint Algorithm. My goal is to fully understand how it works.

LemLib's moveToPoint Algorithm

How It Works:

LemLib's moveToPoint combines two PID controllers simultaneously to create smooth, curved paths:

1. Lateral PID: Controls forward/backward motion to minimize distance to target
2. Angular PID: Controls turning to point toward target

Key Innovation: Cosine Scaling

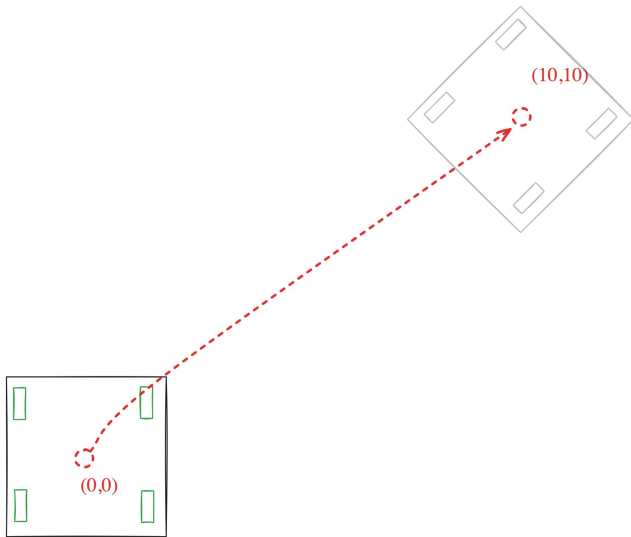
To prevent inefficient wide arcs, LemLib multiplies lateral output by $\cos(\text{angular error})$:

When facing target (0° error): $\cos(0^\circ) = 1.0 \rightarrow$ full forward speed

When 45° off: $\cos(45^\circ) = 0.71 \rightarrow 71\%$ forward speed

When 90° off: $\cos(90^\circ) = 0 \rightarrow$ no forward motion, only turning

This ensures the robot turns toward the target before driving forward.



```
float angularError = angleFrom(robot, targetPoint);
float lateralError = distance(robot, targetPoint) * cos(angularError);

// If facing >90° away, don't drive forward at all
if (abs(angularError) > 90°) lateralError = 0;

float lateralOutput = lateralPID.update(lateralError);
float angularOutput = angularPID.update(angularError);

// Combine outputs using arcade drive
driveArcade(lateralOutput, angularOutput);
```

Challenge:

When is the robot "close enough" to stop?

Our Implementation:

1. Small Exit Condition: Robot must be within 2 inches for 100ms
2. Large Exit Condition: Robot must be within 7 inches for 500ms (fallback if stuck)
3. Timeout: Maximum 3000ms safety limit
4. Early Exit Range: For motion chaining, can exit once within specified distance

Turn Disabling:

Within 7.5 inches of target, angular PID is disabled to prevent random spinning:

Code implementation:

```
if (distance(robot, targetPoint) < 7.5) {  
    angularError = 0; // Stop turning  
}
```

```
lemlib::Controller Settings angular_controller(2, // proportional gain (kP)  
    0, // integral gain (kI)  
    10, // derivative gain (kD)  
    0, // anti windup  
    0, // small error range, in inches  
    0, // small error range timeout, in milliseconds  
    0, // large error range, in inches  
    0, // large error range timeout, in milliseconds  
    0 // maximum acceleration (slew)  
);
```

In this code, the user can set the timeout for both the angular and lateral controller in ms.

Goal:

At our previous competition, we noticed that over time in our skills route and in matches, our autonomous would drift. This meant that the robot would slowly lose track of where it was located on the field. To mitigate this, we decided to integrate distance sensor resets.

How it works:

Essential, distance sensors correct odometry drift by resetting position using wall measurements. This is because the vex distance sensors essentially point a laser and can detect the distance to specific objects. The distance measurement range is 20mm to 2,000mm. Below 200mm the accuracy is approximately +/-15mm, above 200mm the accuracy is approximately 5%.

Our Sensor Setup:

Left Sensor (Port 1): Measures distance to left wall

Right Sensor (Port 10): Measures distance to right wall

Front Sensor (Port 7): Measures distance to front wall



```
void smartOdometryReset(ResetDir dir, ResetSensor sensor) {
    lemlib::Pose pose = chassis.getPose();

    const double mmToInch = 1.0 / 25.4;
    const double fieldHalf = 72.0; // 144in field / 2
    const double sensorOffsetLeft = 173.0; // mm
    const double sensorOffsetRight = 173.0; // mm
    const double sensorOffsetFront = 65.1; // example, measure yours
    // const double minValidDist = 30.0;
    // const double maxValidDist = 1500.0;

    double rawDist = 0;
    double offset = 0;

    switch (sensor) {
        case ResetSensor::LEFT:
            rawDist = dist_left.get();
            offset = sensorOffsetLeft;
            break;

        case ResetSensor::RIGHT:
            rawDist = dist_right.get();
            offset = sensorOffsetRight;
            break;

        case ResetSensor::FRONT:
            rawDist = dist_front.get();
            offset = sensorOffsetFront;
            break;
    }

    // // validity check
    // if (rawDist < minValidDist || rawDist > maxValidDist) return;

    double d = (rawDist + offset) * mmToInch; // inches from robot center to wall
```

Our Implementation in games:

During the autonomous phase, we often turn 90 degrees and then do a full front and side distance sensor reset before going into a match loader. This ensures that our autonomous routines are always reliable

```
pros::delay(milliseconds: 50);
smartOdometryReset(dir: ResetDir::POS_X, sensor: ResetSensor::FRONT);
smartOdometryReset(dir: ResetDir::NEG_Y, sensor: ResetSensor::RIGHT);
chassis.moveToPoint(x: 65, y: -47, timeout: 1200, params: {.maxSpeed=100});
```

Goal

We will test our solution goals for the intake to discern if we must repeat the design process.

Solution Goals (from pg. 142)

The robot must be able to complete the full autonomous routine without losing track of the position on the field

Test #1 - Autonomous test using distance sensors

Test procedure:

1. Run the 7 ball Wing code but before match loading delete the code
2. Do a full x and y distance sensor
3. Check the vex brain. The position should be correct and within a 1 inch range from the actual position

Trial	1	2	3	4	5
Outcome	Pass	Pass	Pass	Pass	Pass



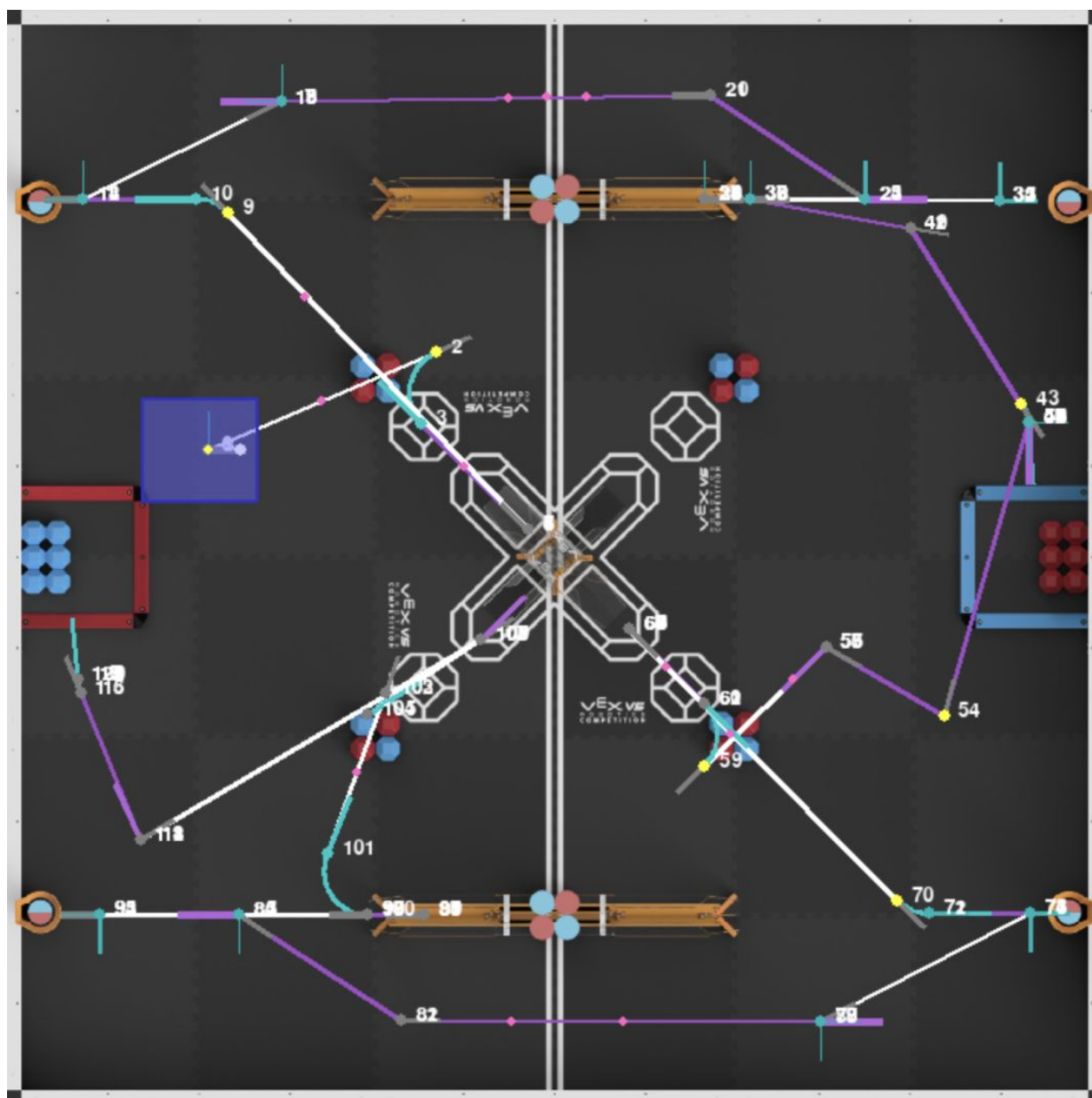
Conclusion

In conclusion, the distance sensors improved our autonomous routines drastically. We were able to complete the testing with an accuracy of 100%. However, since we did only 5 tests, there is still possibility that the autonomous routines could fail. This is due to the nature of the vex hardware and the software limitations in vex v5.

Goal

Our goal is to design a skills auto that can score 96 points. This is important because our team can potentially worlds qualify through skills dropdown. This is always important for getting other awards which require a high skills score

Below is our full autonomous route which can score 96 points.



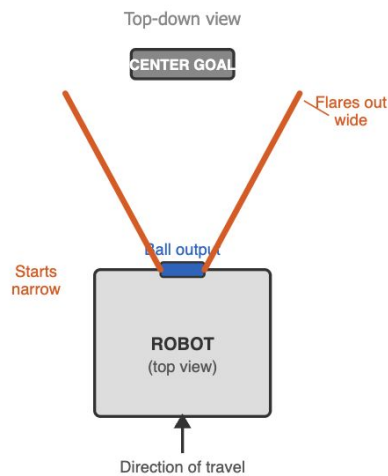
Goal

We will compare and contrast different ways to align our robot to the Center Goals so we can score accurately and consistently without relying on the driver to line up perfectly every time.

Goal

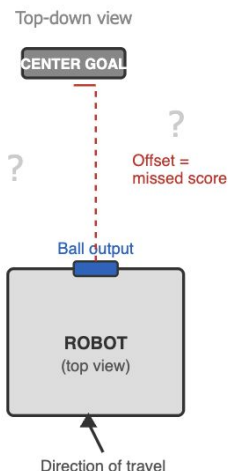
When we drive up to the Center Goals to score, even small misalignments can cause balls to miss or bounce off the side. During a match there's a lot going on and the driver doesn't always have time to carefully position the robot. We need some kind of physical guide on the robot that helps us line up with the goal as we approach it.

Option 1: Flared Guide / Reverse V Shape



How it works: Two angled pieces stick out from the front of the robot in an outward V shape. This is a top-down view of what the aligner needs to do. The wide end catches the goal post even if we're slightly off center, and guides us into the correct position as we drive forward. Alignment happens passively just by driving in.

Option 2: No Aligner (Driver Skill)



How it works: Nothing on the robot to help align. The driver has to line up manually every time. Saves weight and space but any small offset means balls miss or bounce off the goal. Under match pressure this gets inconsistent fast.

Goal

We will decide on a material for our aligner and plan our approach to building it.

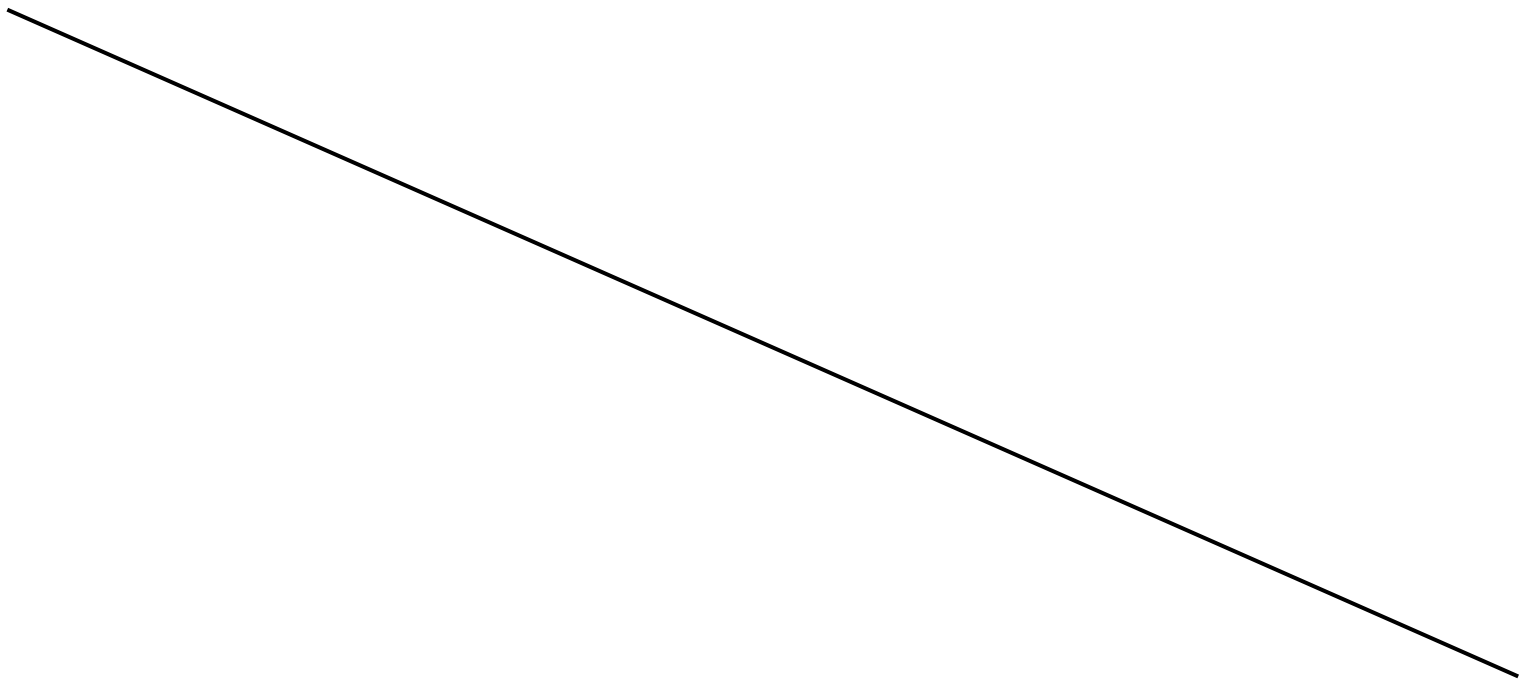
Material Choice

Since we knew we needed an aligner anyway, there wasn't really a need for a decision matrix. The main decision was what material to make it out of. We decided to use plastic (polycarbonate) because:

- We still had capacity since we hadn't reached the 12 piece limit yet
- Plastic is smoother than metal, which helps the robot slide into the goal without catching
- It's easier to work with because we can heat and bend it into the shape we need

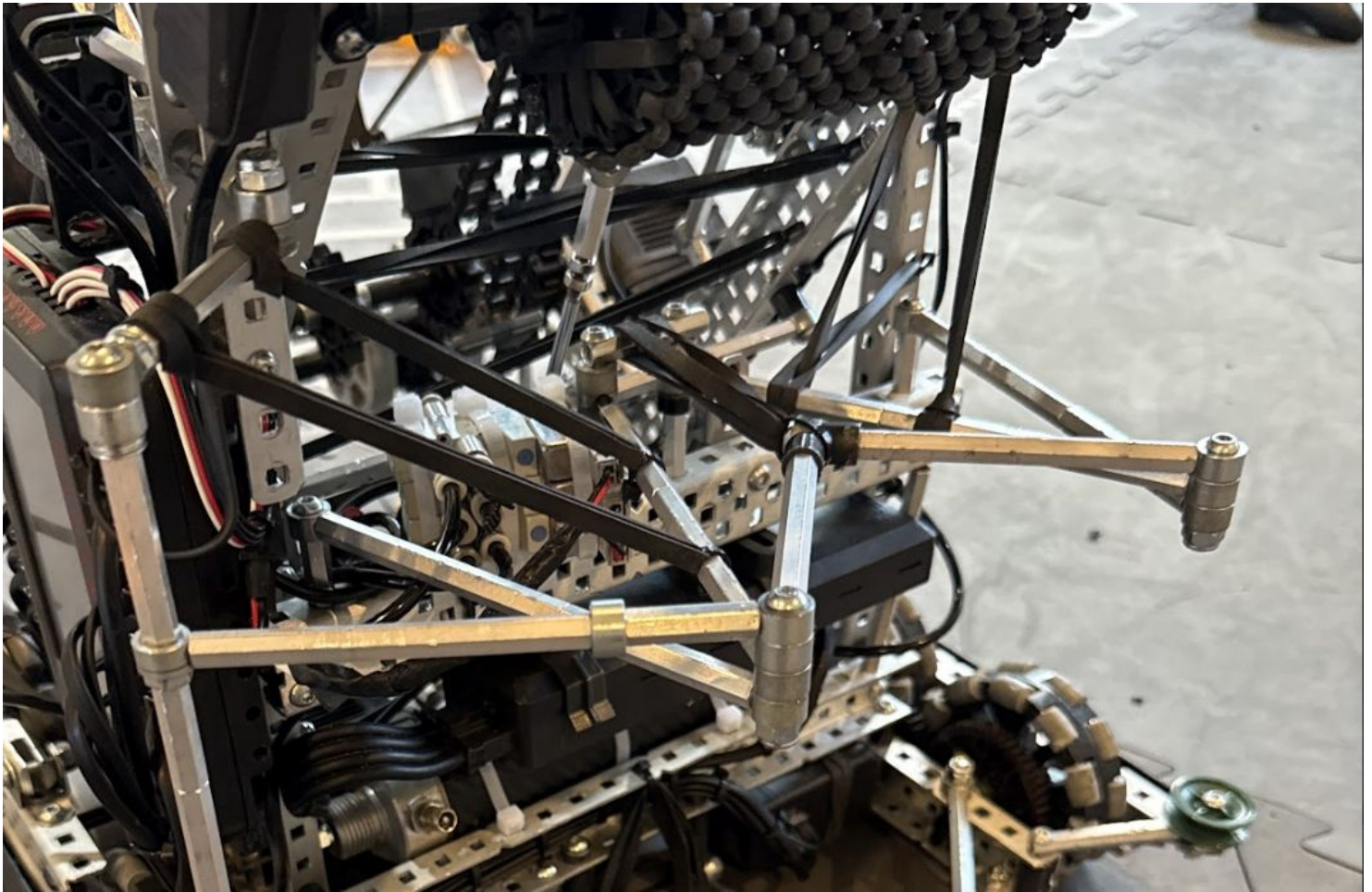
Build Approach

For this system, we didn't do a full design beforehand. We just planned the general shape (outward V / flared guide) and then iterated as we built. We freebuilt it, testing the fit against the center goal and adjusting the angle and length until it lined up properly.



Steps

1. Found standoffs to use as the arms of the flared guide
2. Connected the standoffs together with shaft collars, angling them to form the outward V shape
3. Mounted the assembly onto the main body of the robot, underneath the Ruiguan piston area
4. Tested the fit against the center goal and adjusted the angle until alignment was consistent



Goal

We will evaluate our first aligner design and iterate on it if needed.

Issues with Metal Standoff Aligner

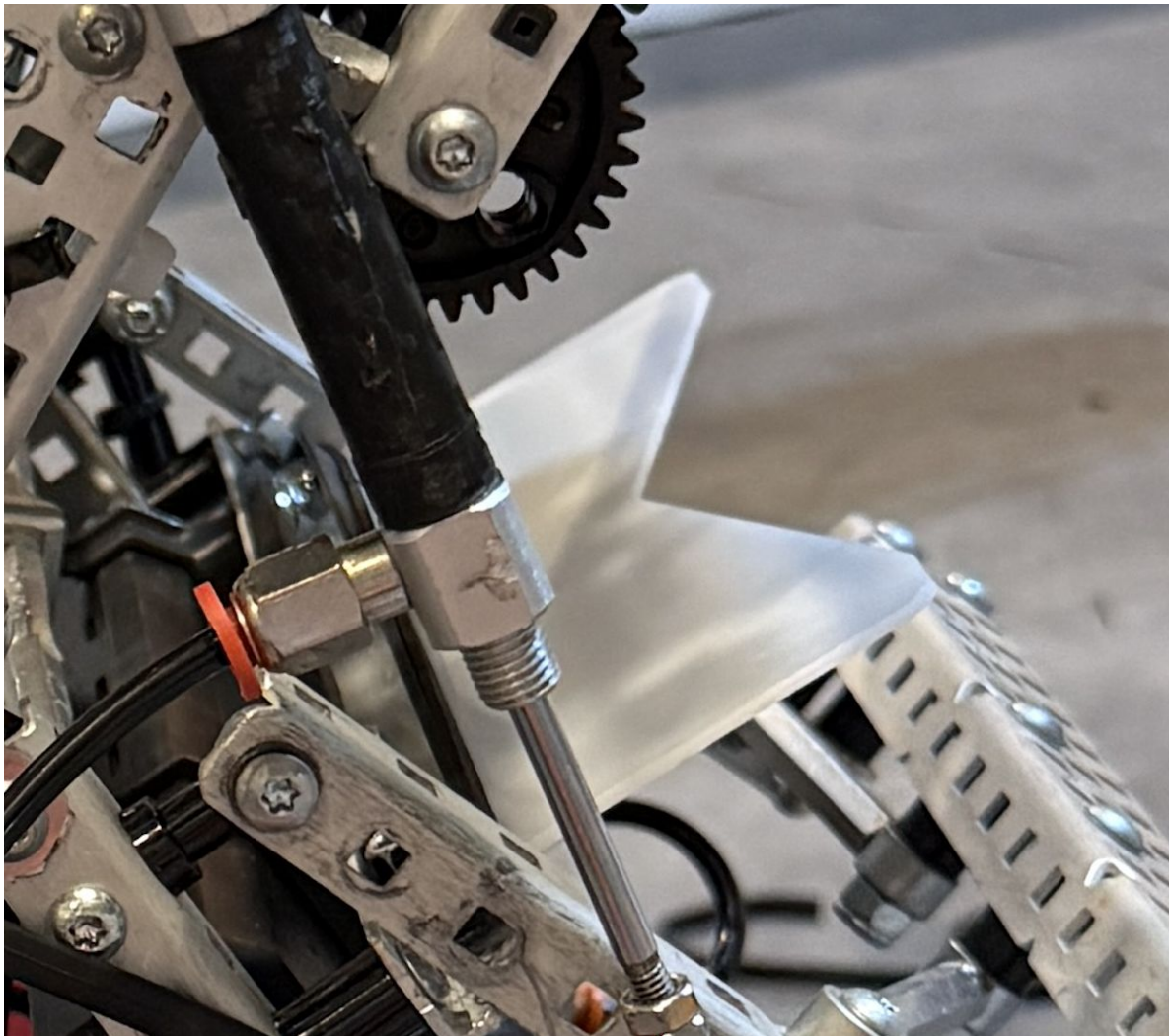
1. After building and testing the metal standoff aligner, we found a couple of problems:
- 2.
3. Used too many standoffs - The aligner took up a lot of standoffs which we could have used for other parts of the robot. Standoffs are a limited resource and using this many on just the aligner wasn't ideal.
4. Unstable - The standoffs could shift too easily during use. Since there weren't that many points of contact holding everything together, the V shape would move out of position when we drove into the goal. This meant the alignment wasn't consistent, which defeats the whole purpose of having an aligner in the first place.
- 5.
6. The concept of the flared guide still works, but the way we built it with metal standoffs wasn't reliable enough. So we decided to try the same thing but with polycarbonate instead.

Why Polycarbonate

- We still hadn't reached the 12 piece plastic limit so we had capacity to use it
- Polycarbonate is smoother than metal, which helps the robot slide into the goal without catching
- We can heat and bend it into the exact shape we need, so the whole V can be one continuous piece instead of multiple standoffs connected together
- Fewer parts means fewer points of failure and less chance of things shifting out of position

Build Approach

- Same as before, we freebuilt it. Cut the polycarbonate to size, heated and bent it into the outward V shape, and mounted it underneath the Ruiguan piston area. Tested it against the center goal and adjusted until the alignment felt right.



Goal

We will evaluate our flat polycarbonate aligner and iterate to a better design.

Issues with Metal Standoff Aligner

1. Slid under the goal - Since the polycarbonate was flat, sometimes it would push underneath the middle goal instead of catching against it. This is really bad for alignment because instead of guiding us in, it just goes under and we lose contact with the goal entirely.
2. Bending - The flat polycarbonate would bend and flex on contact. It didn't hold its shape well enough to consistently funnel the robot into position.
3. Poor funneling - Because of the bendy nature of the flat plastic, it would flex out of the way instead of redirecting the robot towards the center.
4. Inconsistent depth - Sometimes we would drive too far into the goal and other times not far enough. There was nothing stopping the robot at the right distance.

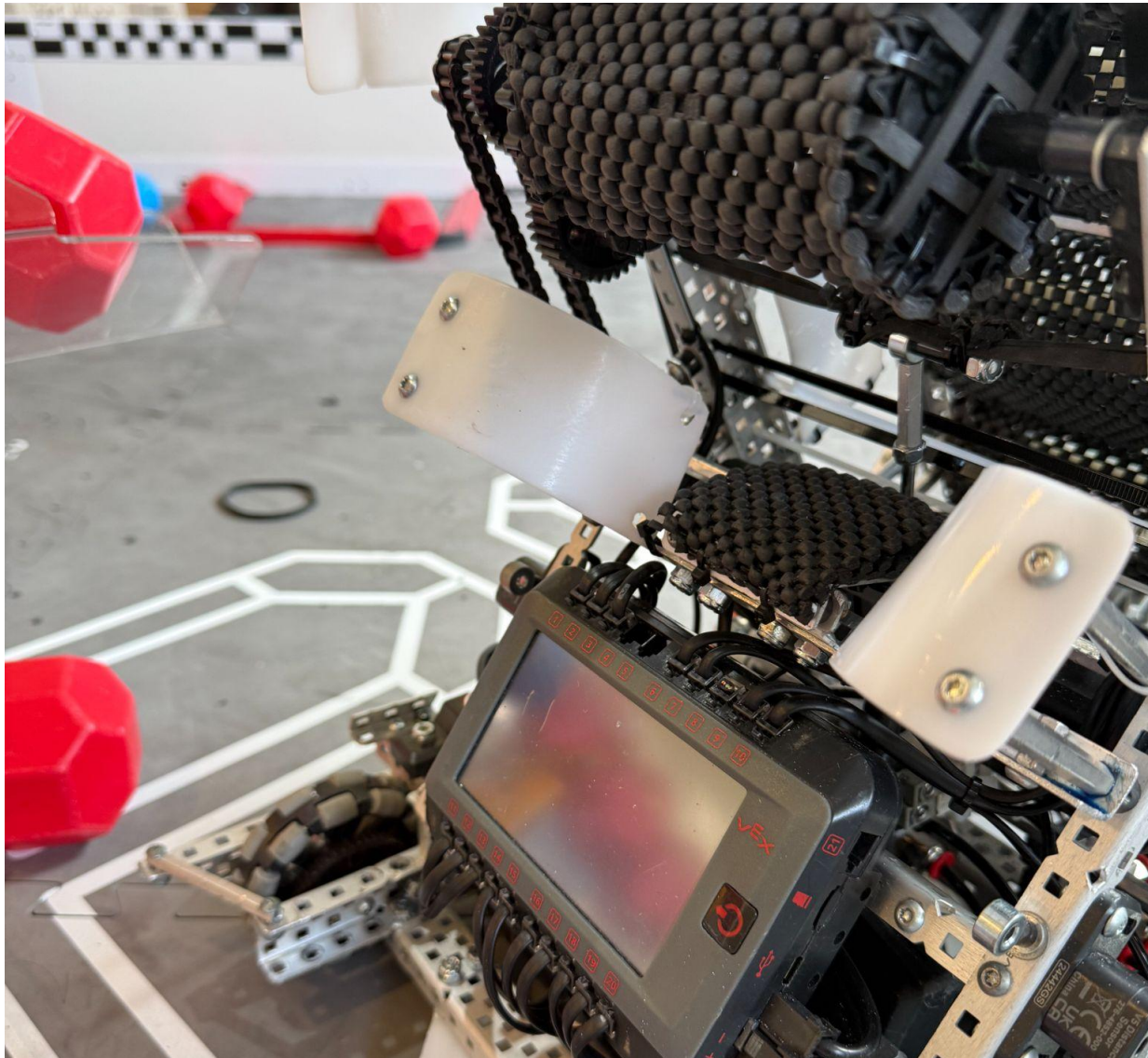
Iteration 3 Proposal: V Shape with Sides and Stopper

To fix these issues, we redesigned the aligner with the following changes:

- Polycarbonate sides - Instead of a flat piece, we added polycarbonate on the sides of the V shape. This means the aligner funnels and aligns the robot regardless of height, so it can't slide under the goal anymore. The sides catch the goal post no matter where we make contact.
- Stopper - We added a physical stopper so that when we drive into the goal, the robot stops at the right distance every time. This fixed the issue of going too far in or not far enough. Now the depth is consistent.

Build Approach

- Freebuilt again. Cut and shaped the polycarbonate for the side pieces, mounted them in the outward V formation, and added the stopper at the correct depth. Tested against the center goal repeatedly until we were happy with the consistency.

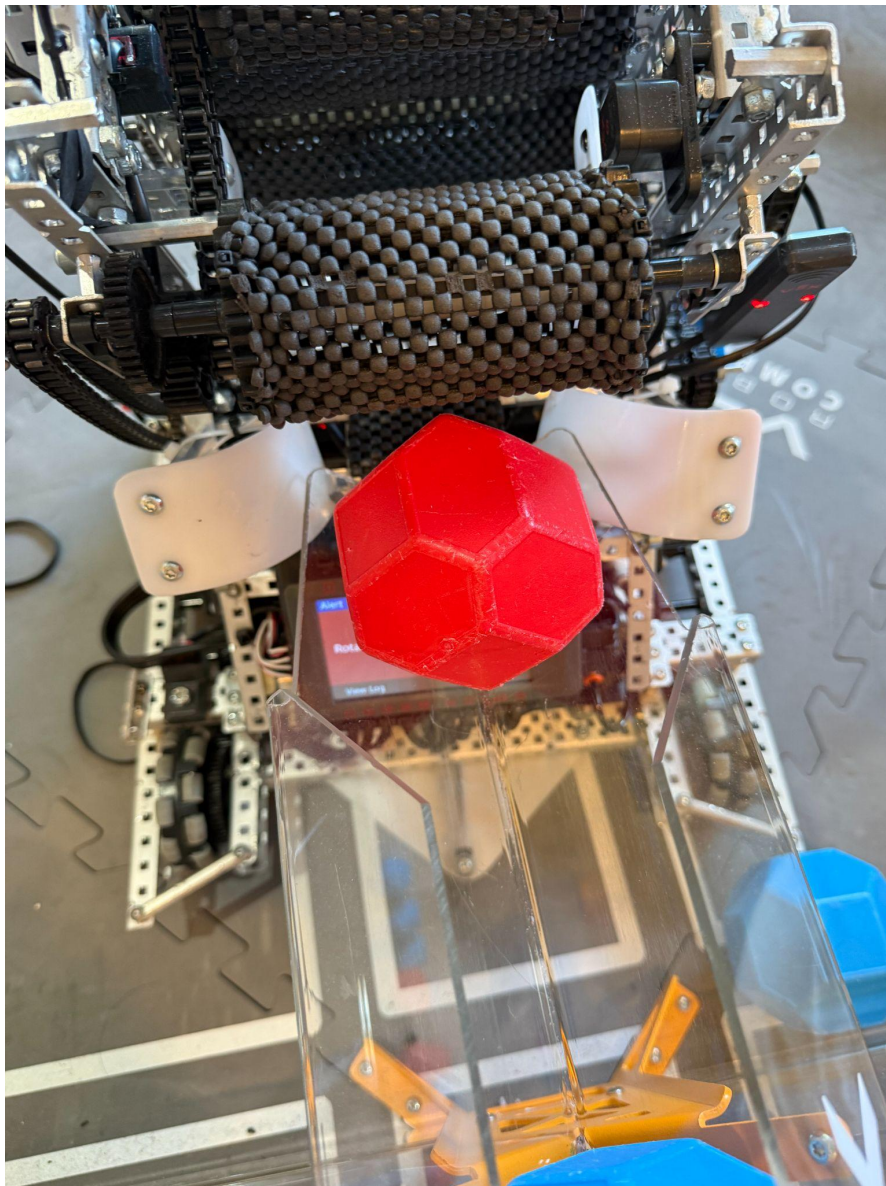


Goal

We will evaluate our aligner design.

Results

1. The V shape with polycarbonate sides and the stopper works well. It aligns us to the center goal consistently, doesn't slide under, and the stopper means we end up at the right distance every time. No further changes needed. A little bit of fine tuning was needed (for things such as spacing, distance, etc.) but in the end it works well.



Goal

We will evaluate the weight distribution of our robot and address any stability issues.

Problem

We noticed that the robot was tilting a lot during matches, and sometimes it would even tip over completely. After looking at the build, we realised the issue was that we had too much weight sitting high up on the robot. The two intake motors were mounted up high, and combined with the brain, battery, and pneumatic tank also being positioned higher up, the center of gravity was way too high.

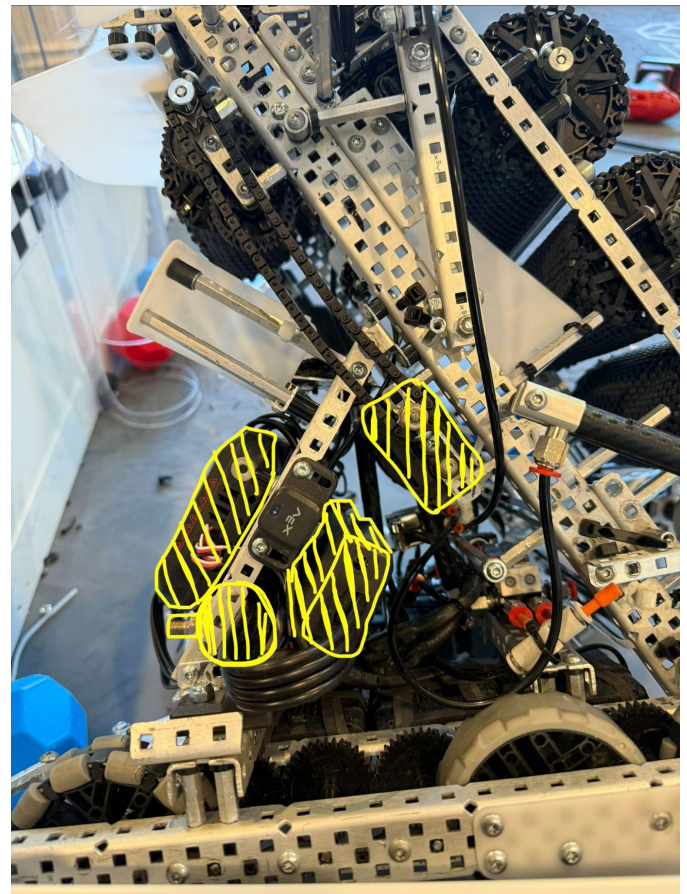
Fix

We moved the following components lower on the robot:

- Brain - Repositioned lower on the chassis
- Battery - Moved to a lower mounting point
- Pneumatic tank - Brought down closer to the base
- Intake motors - Moved from their high position down to a lower mount

Result

After making these changes the robot was much less tippy. The center of gravity dropped significantly which made the whole robot more stable. It no longer tilts as much during normal driving and we haven't had any tipping issues since.



Goal

We noticed while preparing for nationals that our intake would occasionally Jam. We wanted to implement a fix so that we can ensure a higher degree of consistency during matches and skills runs

Solution ideas

Our initial idea was to fix our funneling system to jam less. However, even after changing it to plastic funnels, it did not fix the problem. We then decided to implement fix through code, specifically through our AntiJam function

Solution

STEP 1: Monitor intake motor speed - Continuously check if motor is running slower than expected - If motor velocity drops below 30% of commanded speed, possible jam detected

STEP 2: Confirm jam and reverse - Wait 300ms to confirm jam is real (not a false alarm) - Once confirmed, spin motors backward at full speed (-127) for 100ms to free the jam

STEP 3: Clear and resume - Spin motors forward at full speed (127) for 200ms to push ball through - Return to normal intake operation automatically

```
void updateAntiJamStateMachine() {
    uint32_t now = pros::millis(); // Current time
    uint32_t timeInState = now - enhancedAntiJam.stateStartTime; // How long in curr

    switch (enhancedAntiJam.state) { // Enumeration value 'COOLDOWN' not handled in
        // ===== NORMAL STATE =====
        // Just monitoring for jams during normal operation
        case JamState::NORMAL: {
            // Check if jam is detected
            if (detectJamVelocity()) { // Or use detectJamCurrent()
                // Possible jam! Move to DETECTING state to confirm
                enhancedAntiJam.state = JamState::DETECTING;
                enhancedAntiJam.stateStartTime = now;
            }
            break;
        }

        // ===== DETECTING STATE =====
        // Jam detected, but waiting to confirm it's not a false alarm
        case JamState::DETECTING: {
            // Check if jam has persisted long enough to be real
            if (timeInState >= antiJamConfig.detectionTime) {
                // Confirmed jam! Start clearing sequence
                enhancedAntiJam.state = JamState::REVERSING;
                enhancedAntiJam.stateStartTime = now;
                enhancedAntiJam.jamCount++;

                // Vibrate controller to notify driver
                controller.rumble(rumble_pattern: ".");
            }

            // Check if jam went away (false alarm)
            else if (!detectJamVelocity()) {
                // False alarm, go back to normal
                enhancedAntiJam.state = JamState::NORMAL;
            }
            break;
        }
    }
}
```

```
// Configuration for anti-jam behavior
struct AntiJamConfig {
    int detectionThreshold = 2400;
    int detectionTime = 300;
    int reverseSpeed = -127;
    int reverseDuration = 100;
    int clearSpeed = 127;
    int clearDuration = 200;
    int cooldownTime = 1000;
};
```

Our AntiJam Configuration

Goal

We will test our solution goals for the AntiJam code to ensure that our new code is compatible with our robot

Solution Goals (from pg. 155)

The robot must be able to...

- 1. Score 8 balls without jamming

Test #1 - Score 8 balls without jamming

Test procedure:

- 1. Set up the robot stationary and flat on the field, placing it aligned with the long goal
- 2. Put 8 Balls inside the robot
- 3. Run the intake with the AntiJam code.
- 4. The intake should no longer Jam

Trial	1	2	3	4	5
Outcome	Pass	Pass	Pass	Pass	Pass

Conclusion

We noticed during the testing that with the AntiJam code, our robot would never ever have blocks jammed. Because the antiJam function automatically reverses the intake a bit, it ensures that blocks can get stuck. Overall, this implementation has been very effective.



Goal

While preparing for nationals, we wanted to tune our match loader to be able to match load even when the balls were orientated in a non-ideal position.

Problem Statement

Our existing match loader solution cannot match load when the balls are orientated in a certain way. This is because it does not have a cut out/groove that is required to wedge the balls

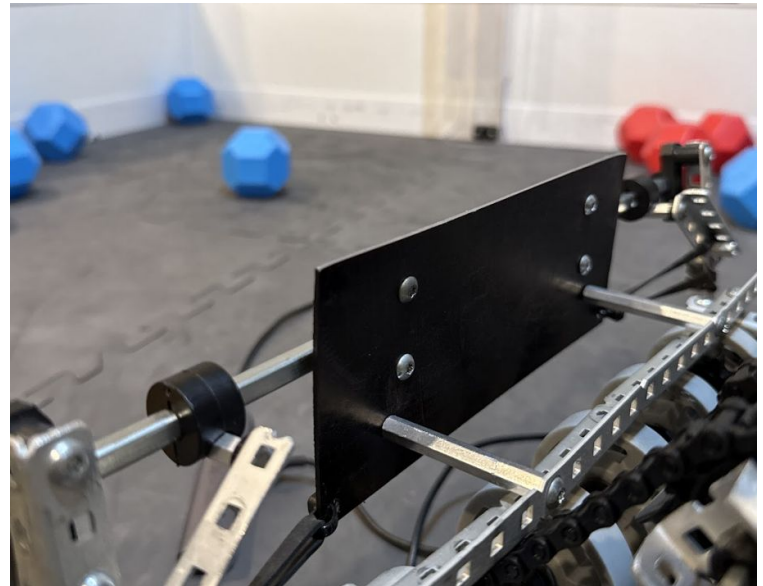
Solution Goals

The robot must be able to...

1. Effectively matchload balls from the tubes without issues no matter the orientation

Solution Constraints

1. Must use Acetal Plastic
2. Must use only legal VEX parts



Our existing match loader solution without any cut outs or grooves. Cannot match load properly

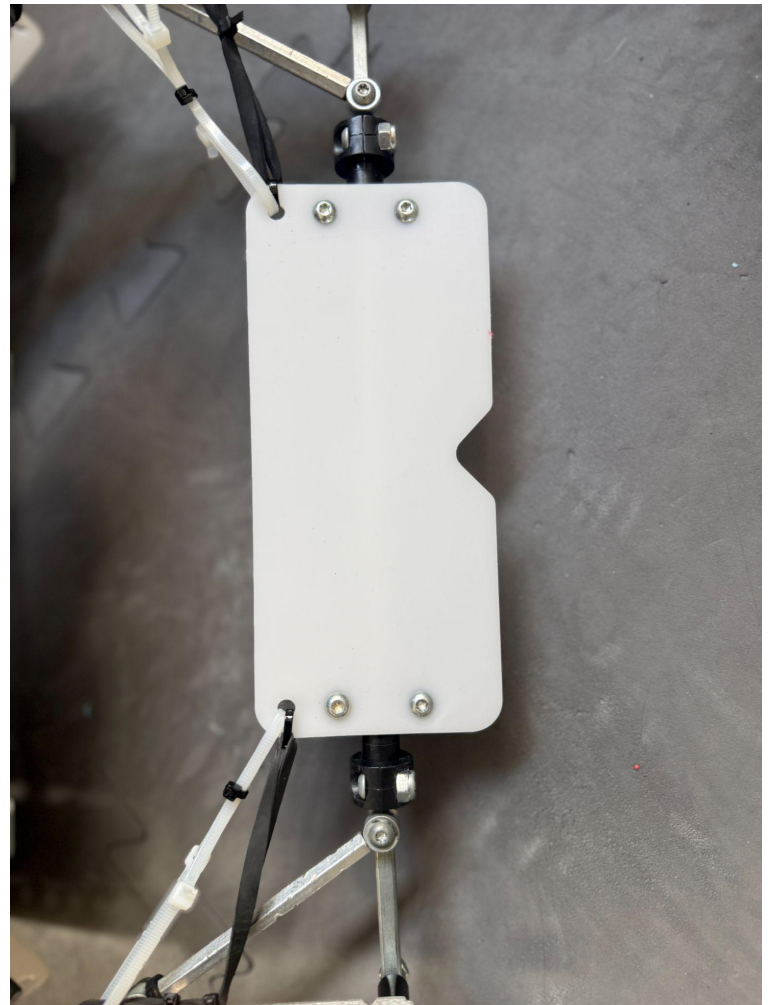
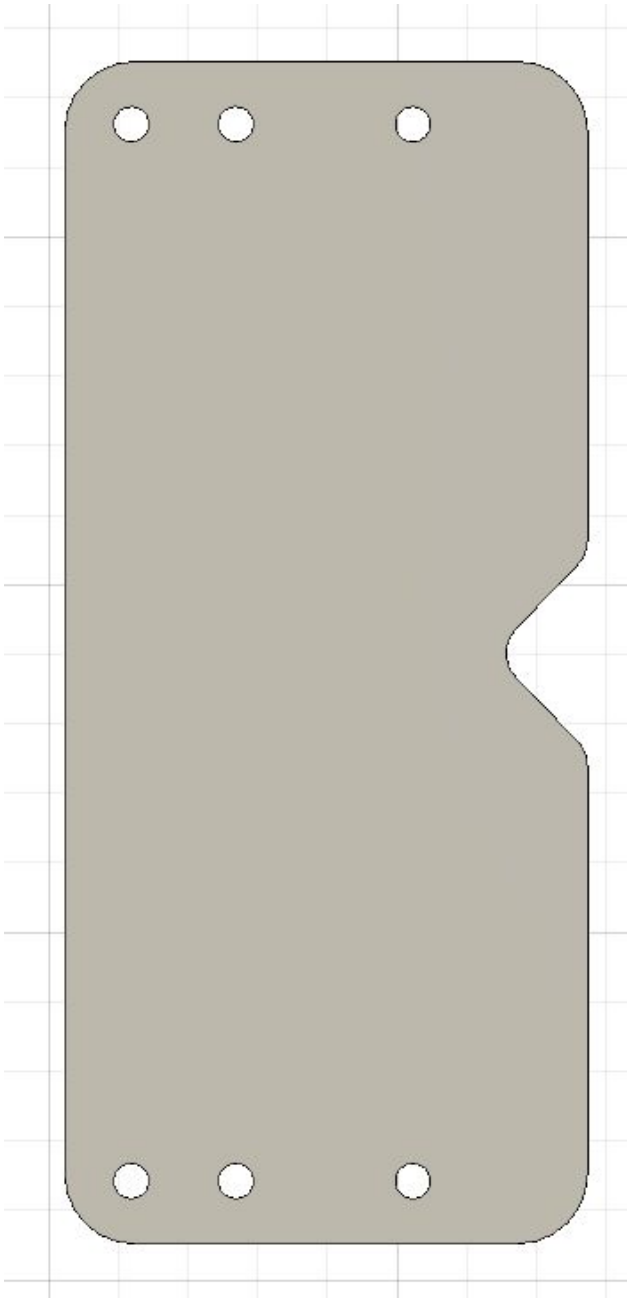
Goal

We will design a match loader that has a groove/wedge on it to matchload balls no matter the orientation

Selected Solution

- Acetal plastic 1.5mm

CAD Plan:



Goal

We will test our solution goals for the matchloader to ensure that it works with our robot

Solution Goals (from pg. 153)

The robot must be able to...

- 1. Matchload balls no matter the orientation

Test #1 - Match Load balls no matter the orientation

Test procedure:

- 1. Setup robot with Match Loader deployed
- 2. Put 6 balls into the match loader and drive robot forward while intake
- 3. The robot should pickup all 6 balls without any jamming or inconsistencies

Trial	1	2	3	4	5
Outcome	Pass	Pass	Pass	Pass	Pass

Conclusion

We noticed during testing that our match loader worked flawlessly. No matter the ball orientation, it was able to match load because the cutout allowed the balls to be wedged upwards and into the intake. Our testing also shows that the matchload is extremely consistent and overall the test was successful.

