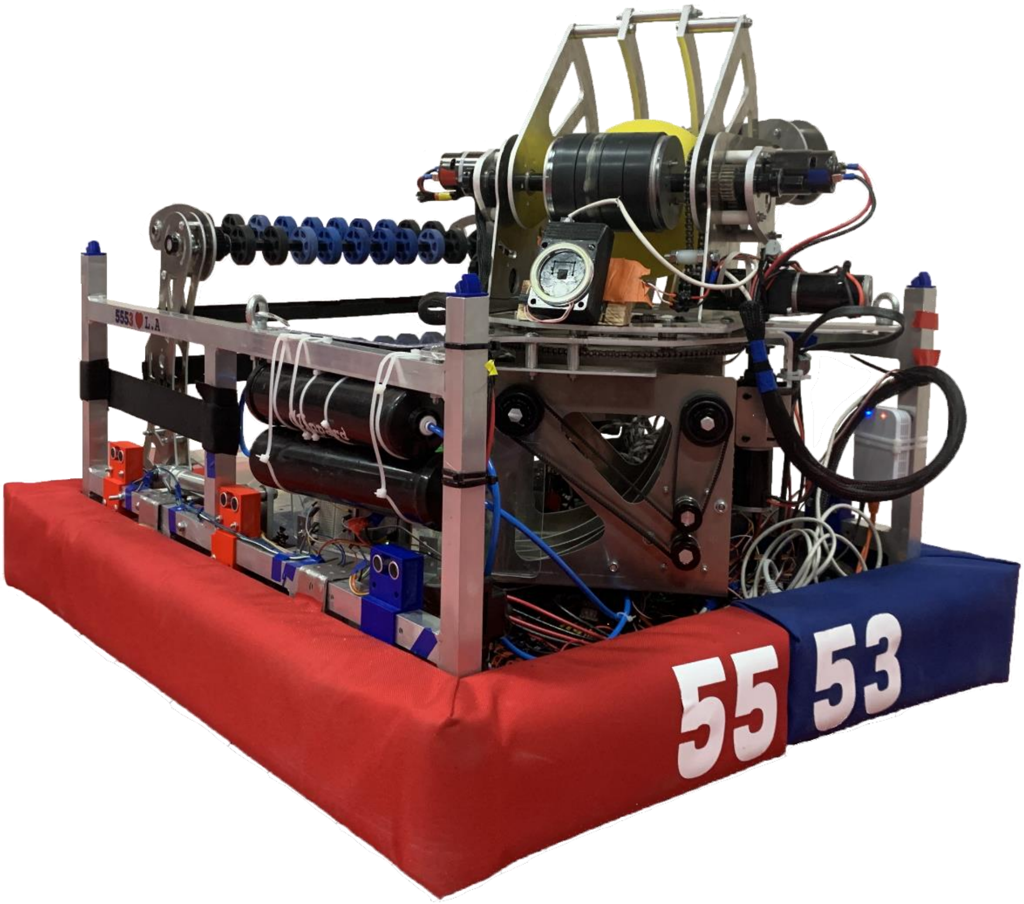


SCRUMTROOPER

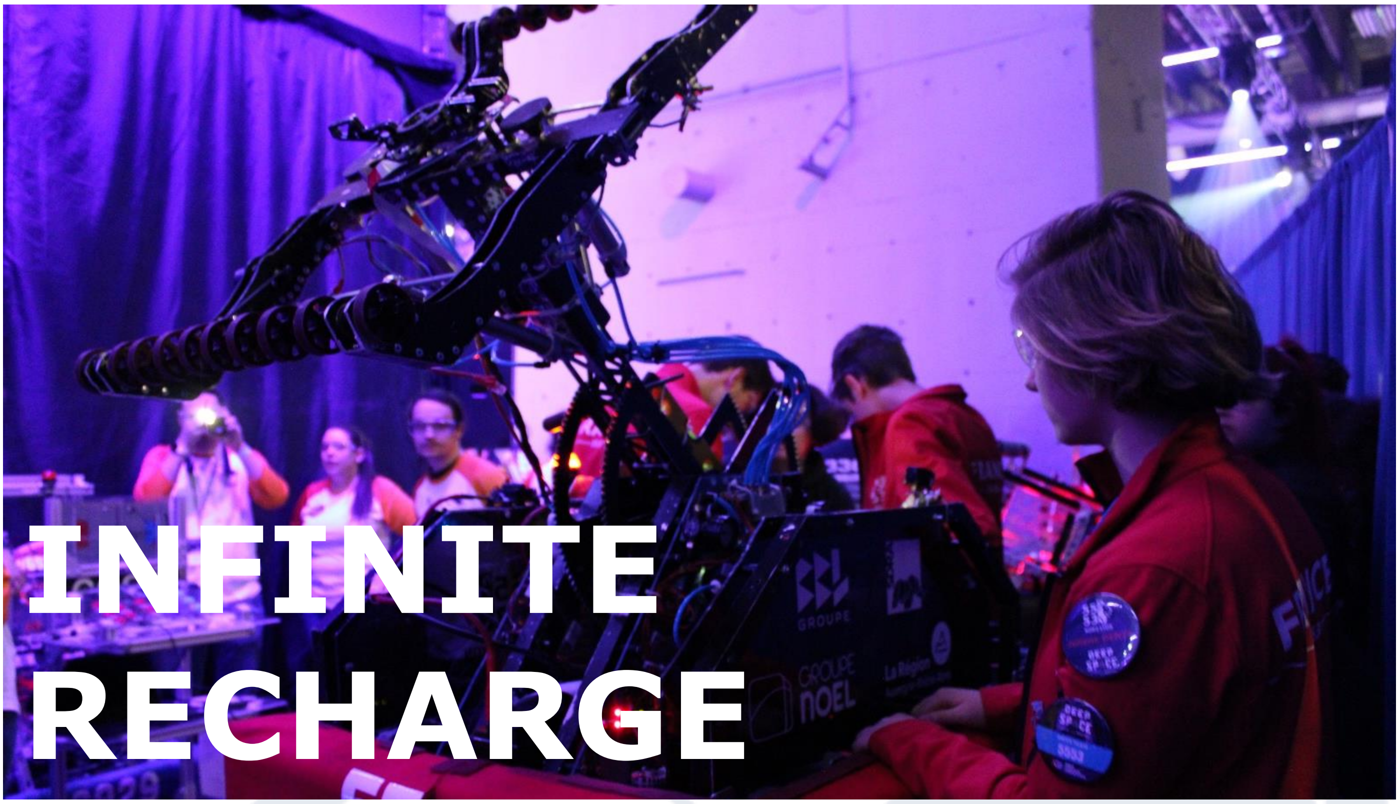
SUMMARY

I. Game Analysis

II. Design

III. Programming

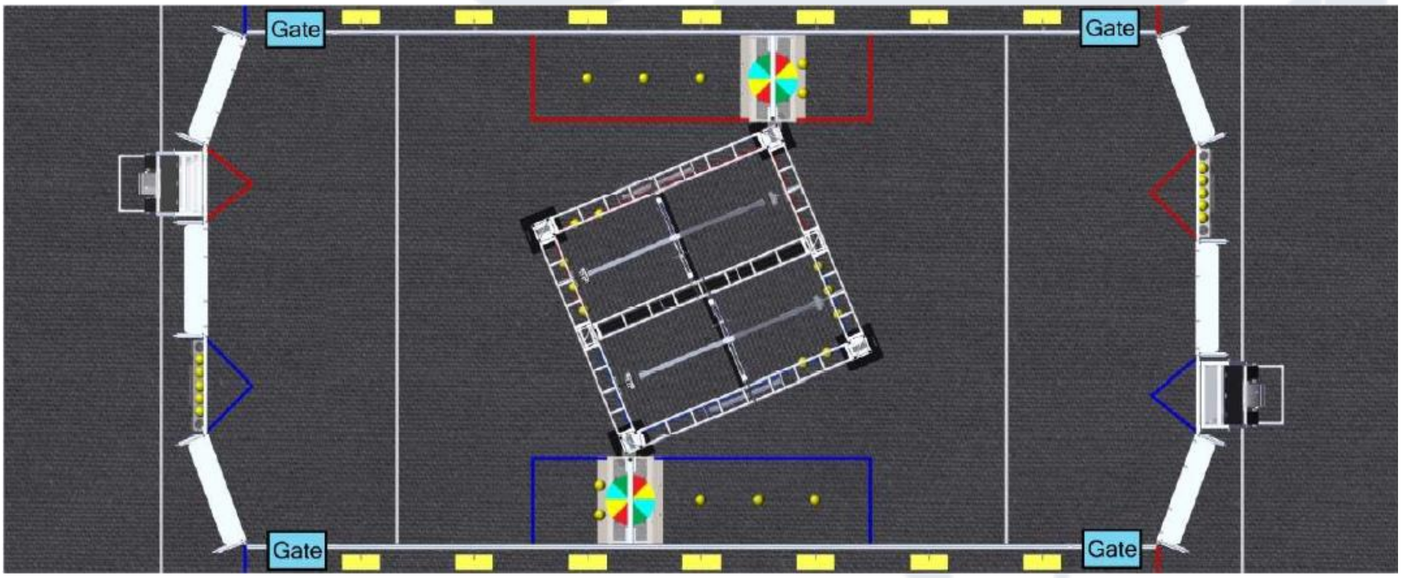
I. Game Analysis



INFINITE RECHARGE

This year’s game consists in taking power cells to throw them in the alliance’s power port in order to power shield generators. Moreover, teams need to spin the control panel to score points and unlock different stages, enabling them to continue the game. In the endgame, robots need to climb the generator’s switch in the center of the field.

The match lasts 2min30. The first 15 seconds correspond to the autonomous period. After the autonomous period, the robot is driven by the driver until the end of the match, where he has on the structure in middle of the field.



- <u>Priority list:</u>

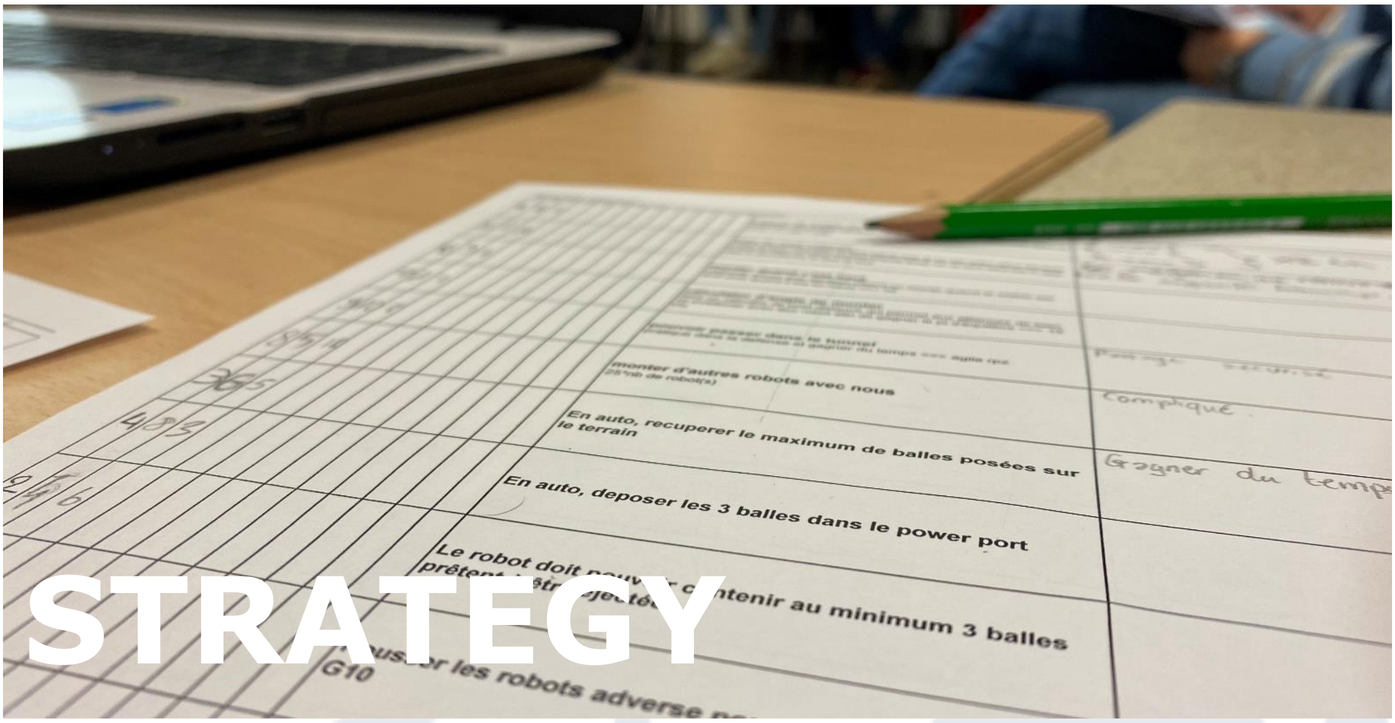
Action	Priority	Comment
Throw balls in the inner port	1	6 points in autonomous & 3 in tele-operated
Pick up power cells on the floor and store five of them	2	Points can be scored faster
Climb the generator's switch	3	Score a lot of points in the endgame
Have a climbing angle calculator to be in the right position to climb	3	Can give a ranking point
Be able to drive in the trench	4	Give a fast and protected way to cross the field, useful to score point even faster
Spin the control panel the right number of rotations	5	Give 10 points easily
Pick as much power cells as possible in autonomous mode	6	Gain of time, possibility to score immediately after the start of tele-operated period
Carry one or two other robots when climbing	7	Can give as much as 75 points or 90 points and a ranking point if switch is balanced
Force the opponent to commit a foul play	8	Can give additional points

- <u>Ranking points:</u>

Action	Ranking points
65 or more endgame points	1
At least 20 points and control panel stopped on the right color	1
Tie	1
Victory	2

- <u>Points</u>:

Action		Value	
		In autonomous period	In teleoperated period
	Initiation line crossed	5	Ø
Power cells	In bottom port	2	1
	In outer port	4	2
	In inner port	6	3
Control panel	Spun the right number of times	Ø	10
	Stopped on the right color	Ø	20
Endgame	Robot hung	Ø	25
	Robot parked	Ø	5
	Switch balanced	Ø	15



STRATEGY

When we received the game manual on Kick-off day, the team met in order to brainstorm about the game. This brainstorming aim was to rank all the game's action by priority order.

General Design

This year's main goal is being able to do all the action in order to diversify our robots' abilities for our future alliances. Scoring balls and climbing giving a lot of points, we decided to focus on those two mechanisms. Moreover, passing under the trench enables us to cross the field quickly, but puts a limit to the robot's height. Although the control panel manipulator is an important mechanism, its simple design allowed us to design it later on. Therefore, the robot is to be built around the throwing line and the climbing system while staying under 65cm.

Autonomous Period

We aim to score as many balls as we can in autonomous period. Indeed, if our minimum of 8 balls are put in the inner port, at least 48 points are scored to which are added the 5 points for crossing the initiation line, a total of 53 points. In addition, crossing the 9-balls threshold activates Stage 1 and brings us closer to the Ranking Point.

Teleoperated Period

Our goal in teleoperated period is to score once again the most balls in the inner port to win the match and quickly activate Stage 1 (if not already activated during the Autonomous Period), then Stage 2 and 3. We plan to avoid the enemies' defense by passing under the trench. Moreover, our gearbox combines speed and strength in order to defend on other robots.

Endgame Period

In Endgame, we determine before the game where to place climbing robots so that the generator switch is balanced and win as much points as possible.

GOALS

- **<u>Driving:</u>**
  - <u>Drivetrain</u>
    - Simple and resistant
    - Fast
  - <u>Gearbox</u>
    - A high speed for long trips
    - A low speed to defend, for short trips and for powerful accelerations

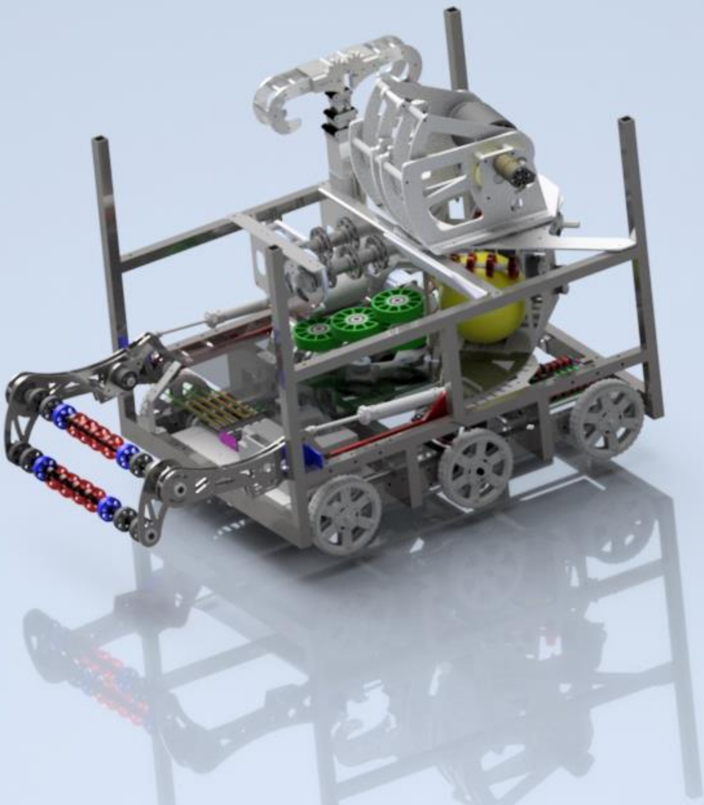
- **<u>Mechanism to manipulate the power cells:</u>**
  - <u>Intake</u>
    - Able to catch the balls on the ground and in the loading station
    - Reliable
    - Easy for the driver
  - <u>Conveyor</u>
    - Able to carry five power cells at once
    - Able to know precisely how many power cells are loaded
    - Fast and reliable
  - <u>Turret</u>
    - Able to rotate in every direction
    - Can position itself autonomously
  - <u>Shooter</u>
    - Able to shoot from far on the field with precision

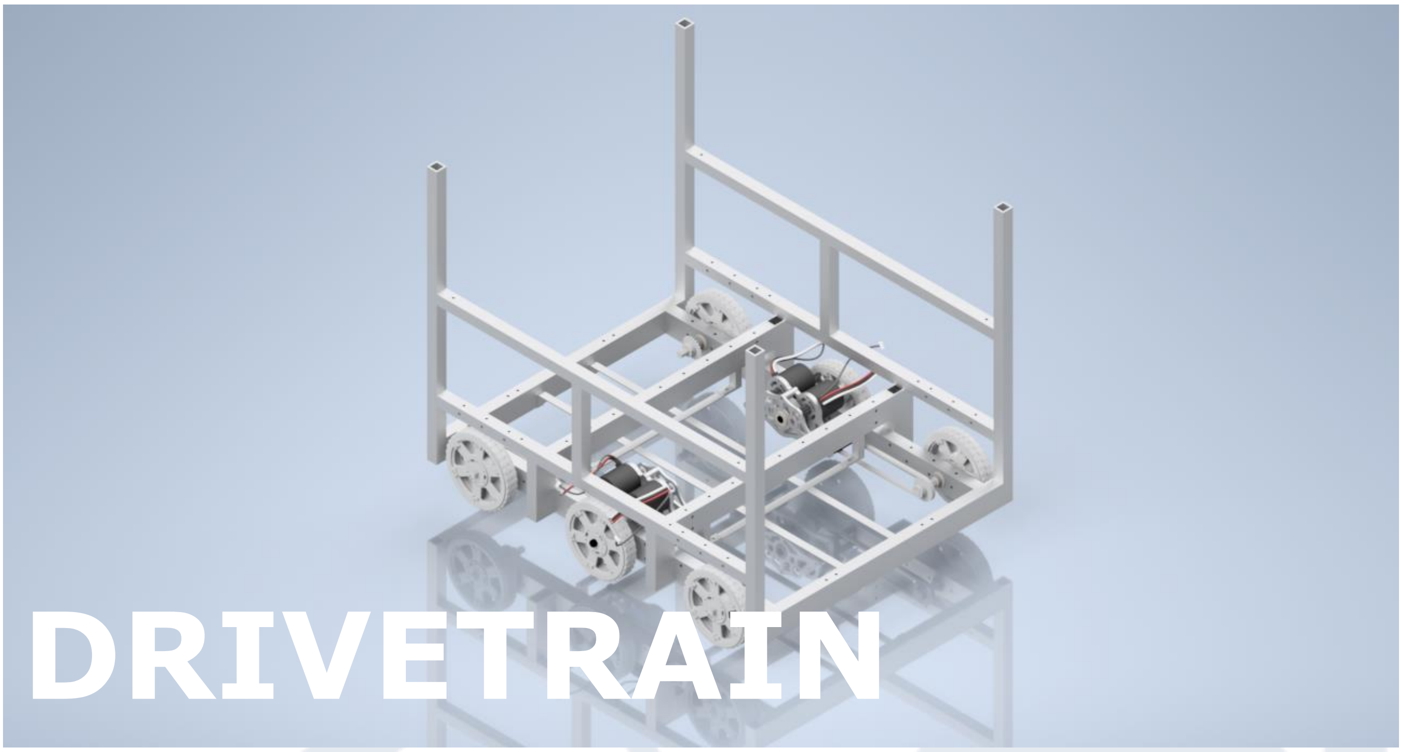
        - Can shoot five power cells in a short amount of time

- **<u>Control panel mechanism & Climbing:</u>**
    - <u>Control panel</u>
        - Can spin the wheel the right time
        - Can stop the panel on the right color
    - <u>Climbing</u>
        - Fast and reliable
        - Resistant

- **<u>Programming</u>**:
    - <u>Color recognition</u>
        - Useful to control the wheel and on which color it stops
    - <u>Visual processing</u>
        - Enable the turret to always face the power port
        - Help to shoot in the inner port

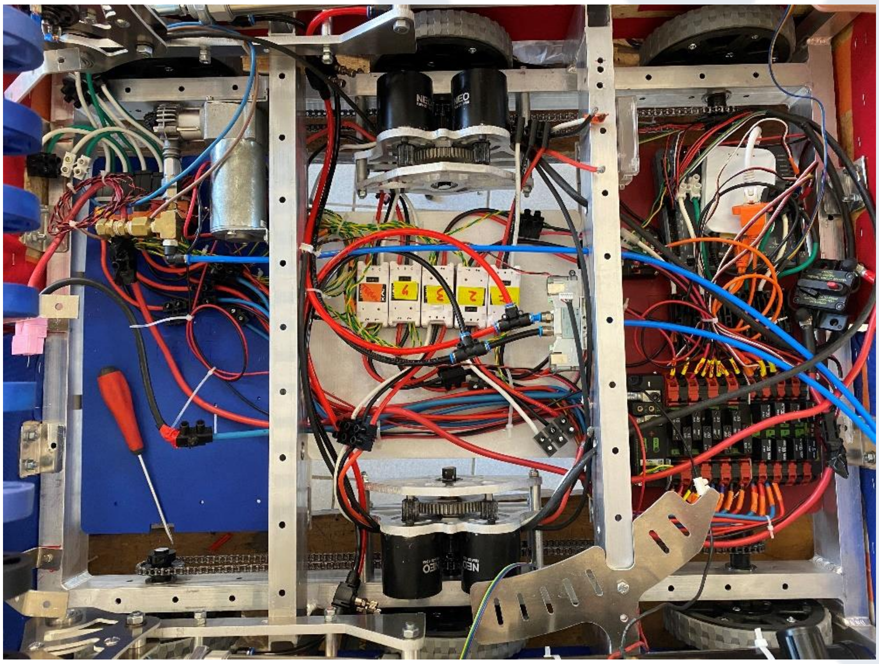
II. DESIGN





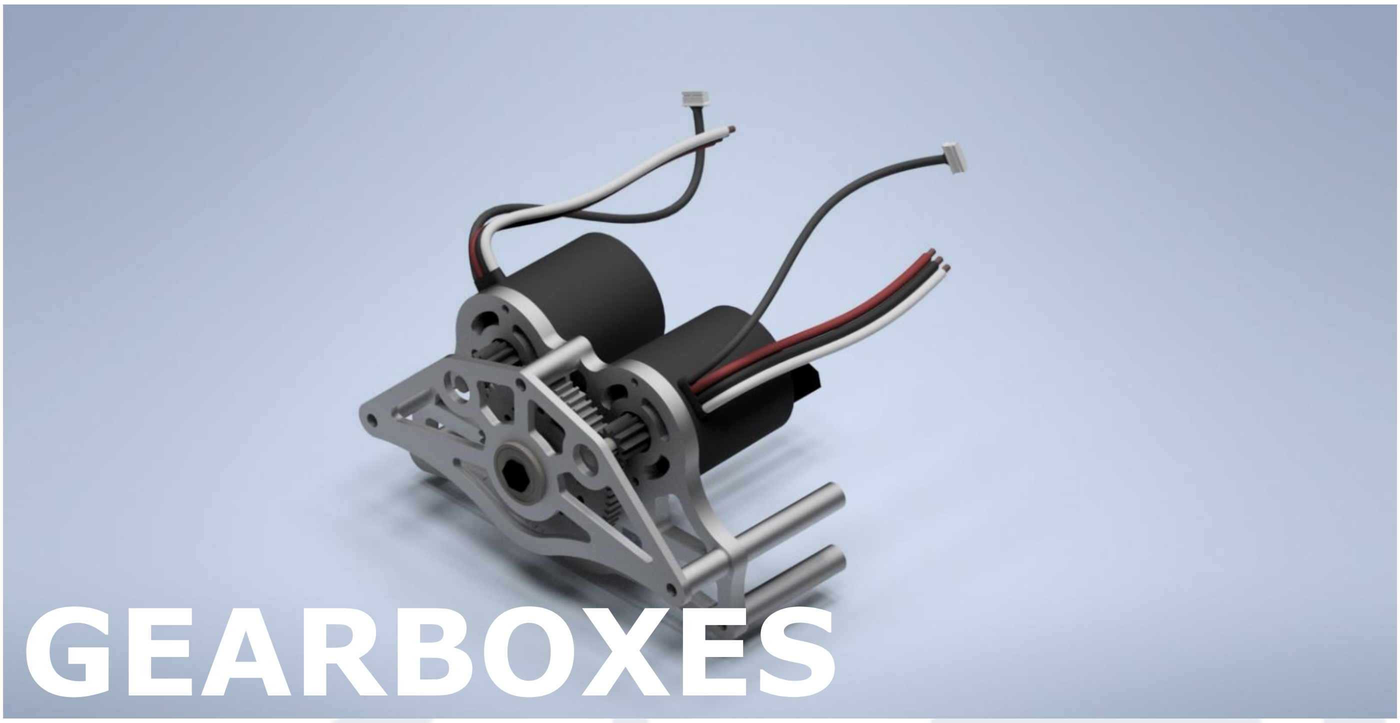
DRIVETRAIN

- ❖ The drivetrain is arguably the most important part of the robot, as it supports the other mechanisms. So, we built a very tough frame.
- ❖ 25x50x3 mm (1x2x0.1in) aluminum soldered profiles for maximum robustness
- ❖ Chain transmission
- ❖ 3 mm (0.1in) drop center to pass little obstacles
- ❖ Six 6in traction wheels



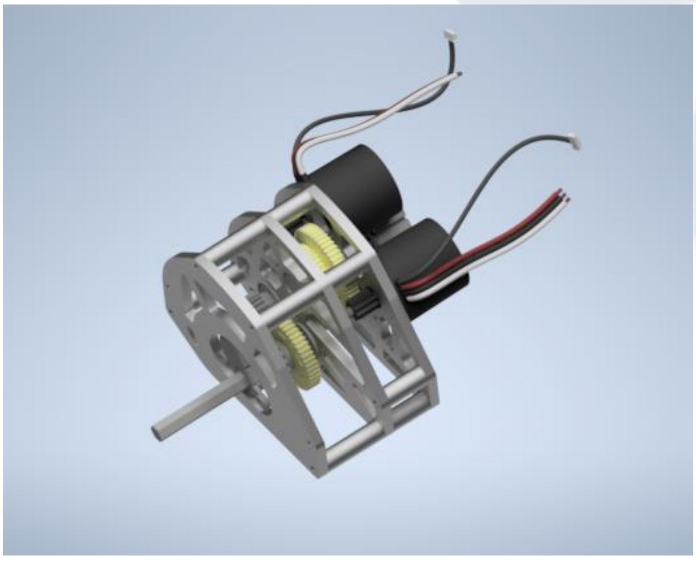
<u>For electronics parts:</u>

We designed a custom-made electronical plaque using our CNC, allowing us to manage our cable easily.



GEARBOXES

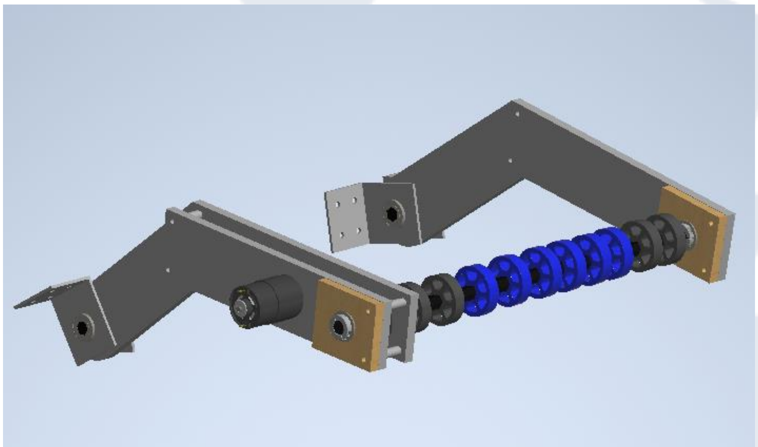
- ❖ We designed the lightest gearbox possible, as the first we mounted were too heavy and the robot wouldn't be able to pass the inspection.
- ❖ Reversed 1 speed gearbox to reduce obstruction
- ❖ Motorized by 2 NEO
- ❖ 10.8:1 reduction to have speed as well as strength
- ❖ Speed: ~36.8 ft/s (~11.3 m/s)



At first, we designed a two speed ballshifter gearbox in order to have one speed dedicated to torque and the other one to velocity. But as our robot was way too heavy for the game, we redesigned a much lighter version to fit the rules.

INTAKE

Prototype

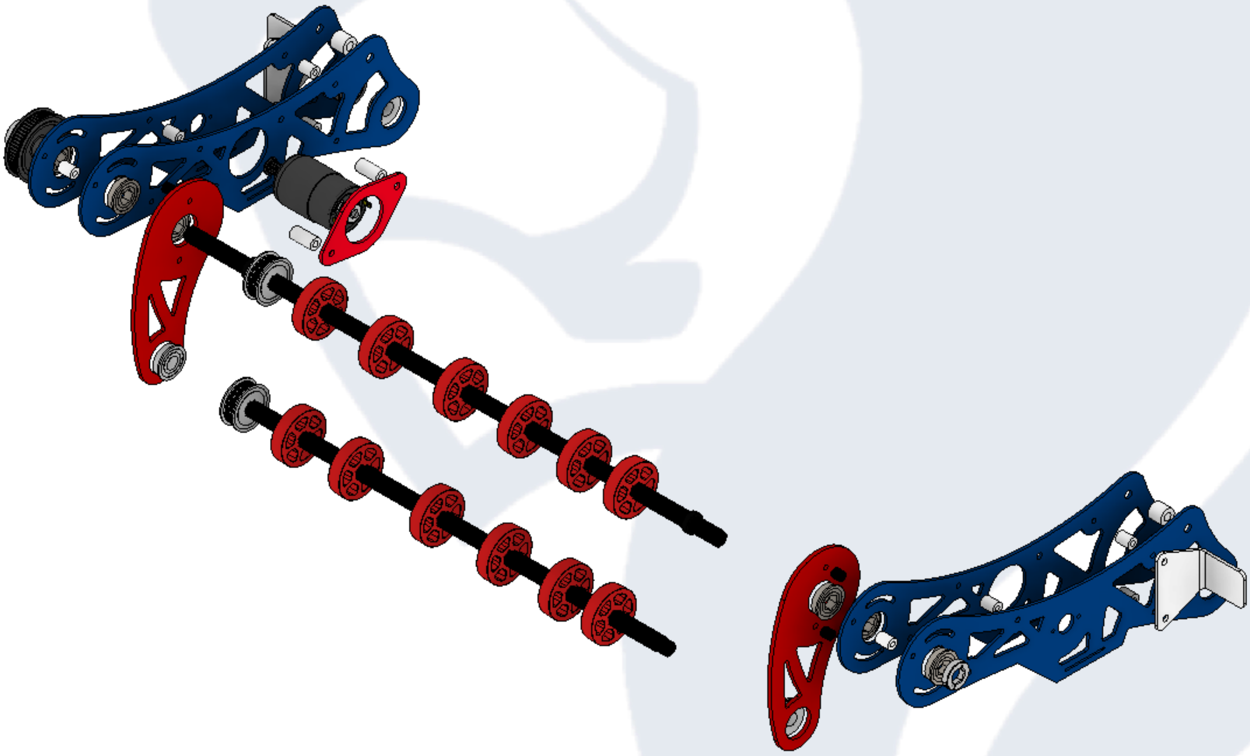


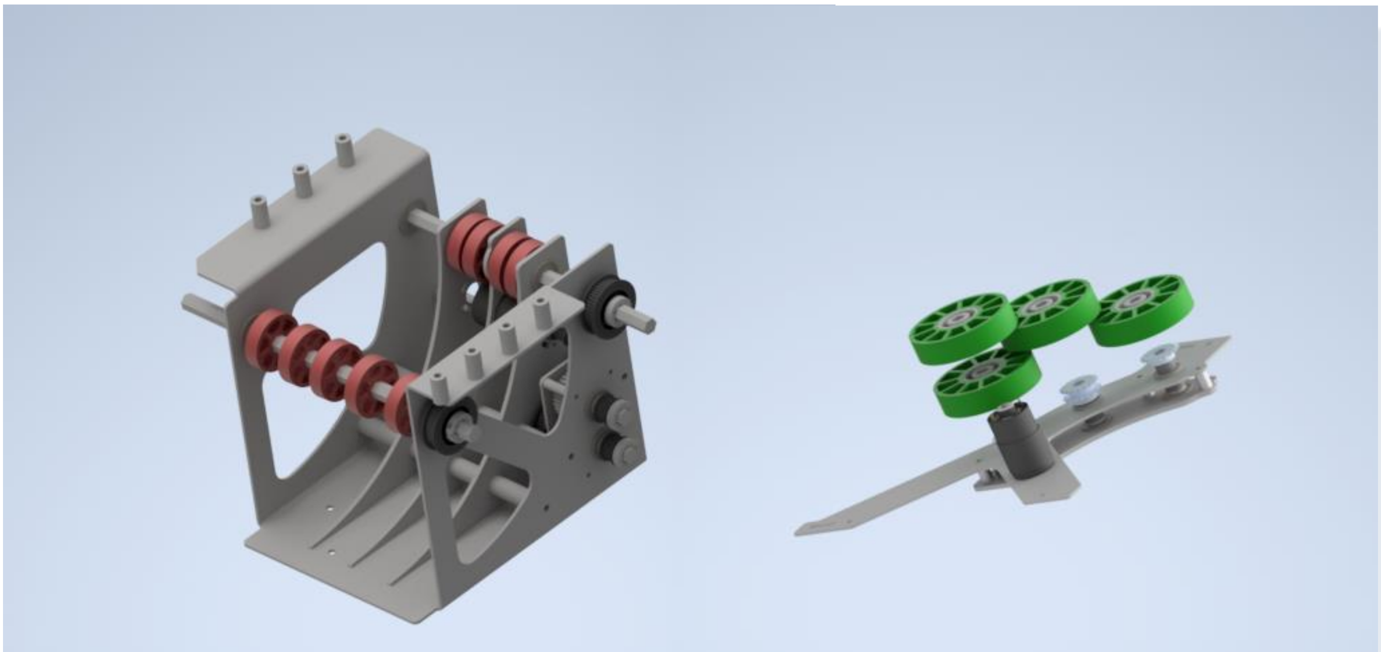
We’ve prototyped several designs in order to:

- Determine the number of shafts needed in order to grab efficiently the balls on the ground
- Determine the exact position of these shafts to regulate the ball’s compression against the ground and the bumpers
- Determine the way to move the intake

Final design

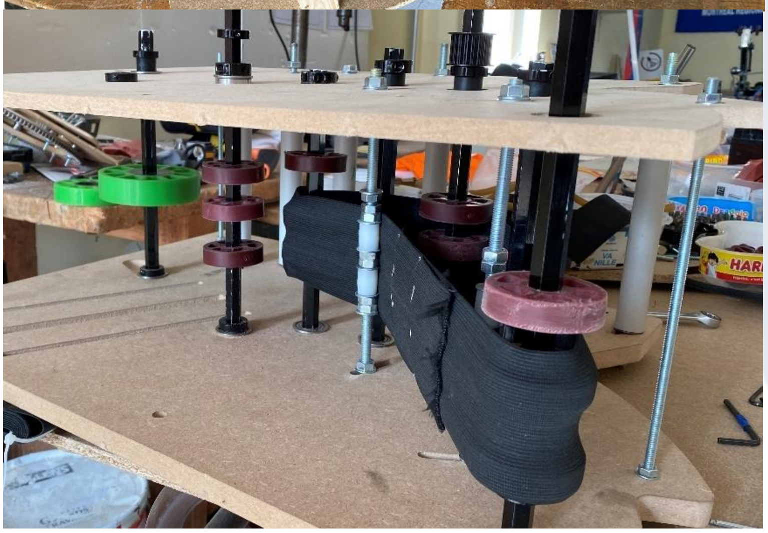
- ❖ The intake grabs the ball on the floor in order to push them into the conveyor. There are two sets of ANDYMARK 2in black and blue compliant wheels on a movable head to adjust the ball compression to perfection before each match
- ❖ Motorized by a 775 Pro
- ❖ Reduction of 5:1
- ❖ Transmission assured by pulleys and belts
- ❖ Actuators with a 100mm stroke and a 20mm diameter on each side of the intake to lower and raise it





CONVEYOR

Prototype



We've prototyped several designs in order to:

- Determine how to bring the balls to the feeder the fastest
- Determine how to stock the ball to drop them just when we want them
- Determine how to minimize the loss of the balls when we activate the mechanism
- Determine how to drive the balls without blocking them

To create the most optimal design, we came up with three different ideas:

❶ Conveyor with belt in order to create a ball's way

- ➔ Pros: regular trajectory so it is easy to manage for the shooter.
- ➔ Cons: Balls obstruction because of the compression, not enough space to stock 5 balls.

❷ Ramp with motorized wheels at the end

- ➔ Pros: few resources, large storage capacity, no need to motorize a lot of wheels.
- ➔ Cons: random trajectory, balls obstruction in the non-motorized zones, balls could fall.

❸ Rotating drum

- ➔ Pros: precise and accurate, regular trajectory, fast.
- ➔ Cons: Heavy, takes too much place on the robot.

❹ Ramp with 4 wheels in order to motorize the bottom half of the ramp and walls

- ➔ Pros: Exactly 5 balls storage capacity, few parts, few losses of balls.
- ➔ Cons: Quite random trajectory.

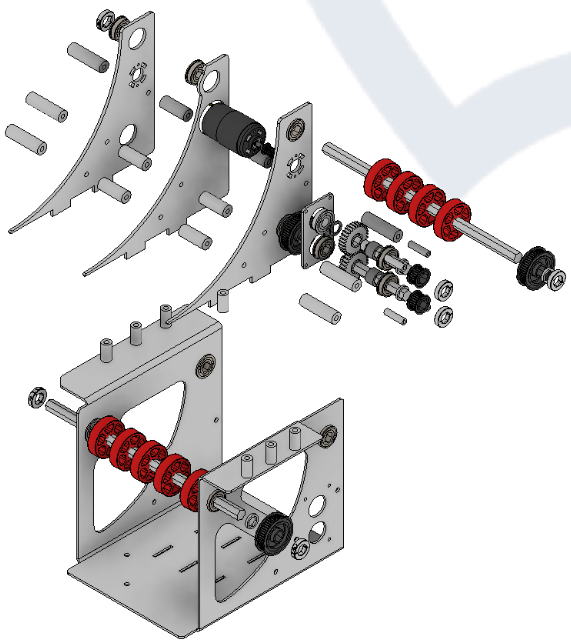
Final design

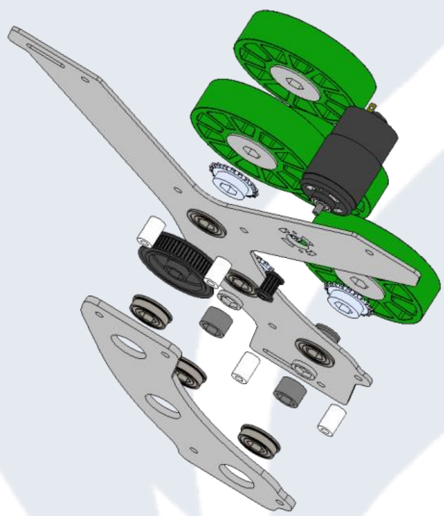
Conveyor:

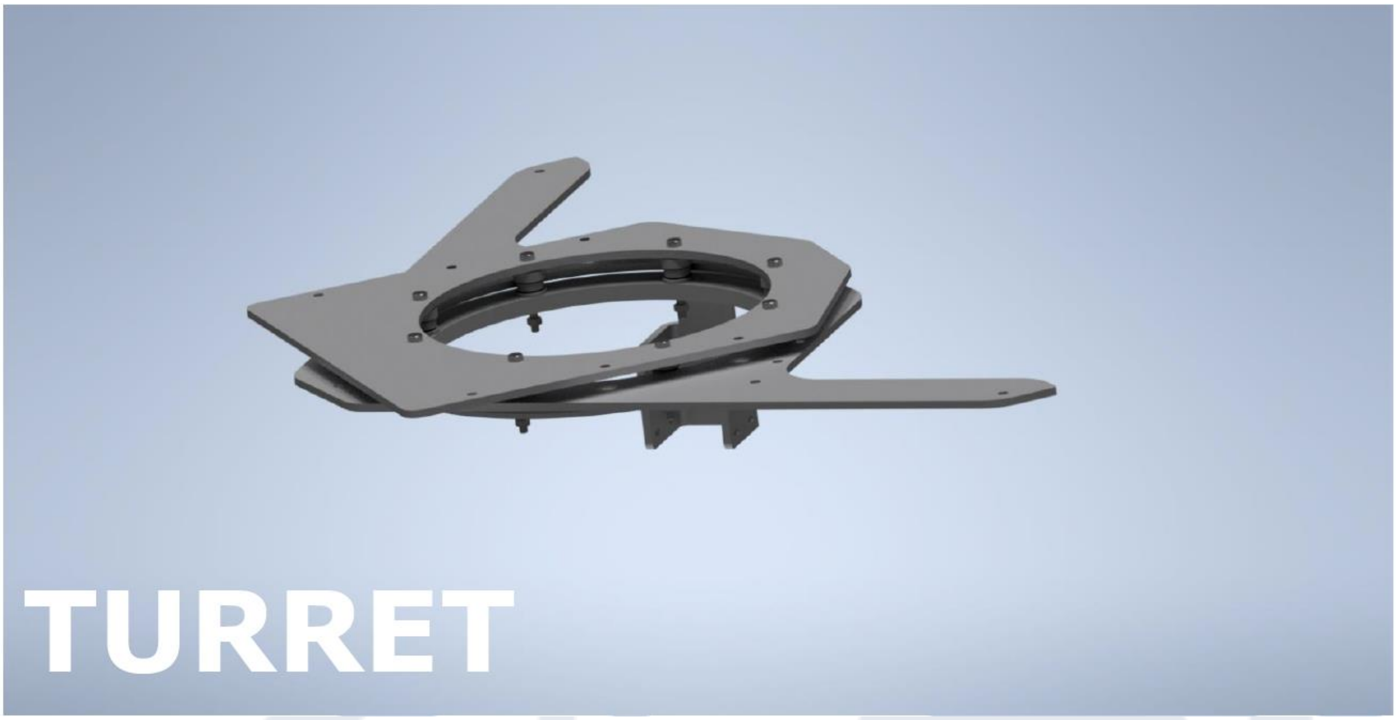
- The conveyor is the mechanism connecting the intake with the shooter and also stocking the balls before shooting them. We created a ramp with ANDYMARK 4in green compliant wheels to motorize the mechanism when we want to send the balls in the turret and the shooter
- Motorized by a 775pro
- Reduction of 5:1
- Transmission by pulley and belt, and by sprockets and chains.
- 3 sets of ANDYMARK 4in green compliant wheels
- Ramp and walls cut with our CNC

Feeder:

- The feeder is a way to present the ball to the shooter without the magnus effect. This is why we have two set of wheels to cancel the magnus effect. The first set of wheels is here to grab balls from the conveyor to the feeder, once they are inside the feeder, they are guiding by a ramp. To finish, the second set of wheels pushes balls up to the shooter
- Motorized by a 775 pro
- Reduction of 8:1
- Transmission ensured by pulleys and belts





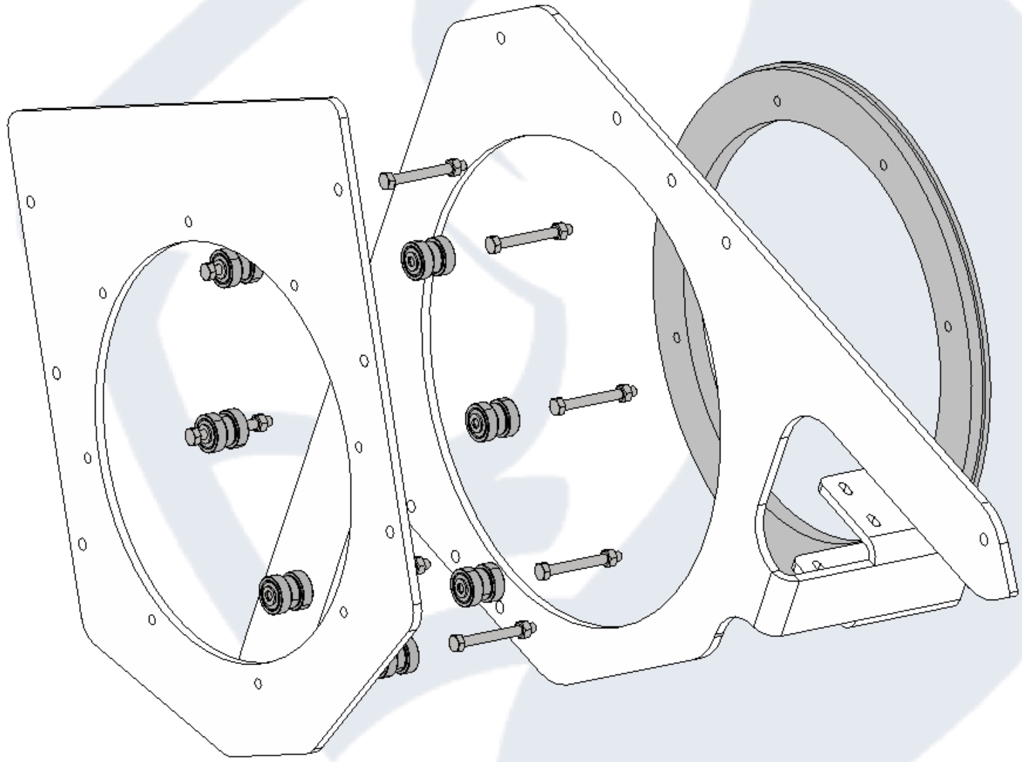


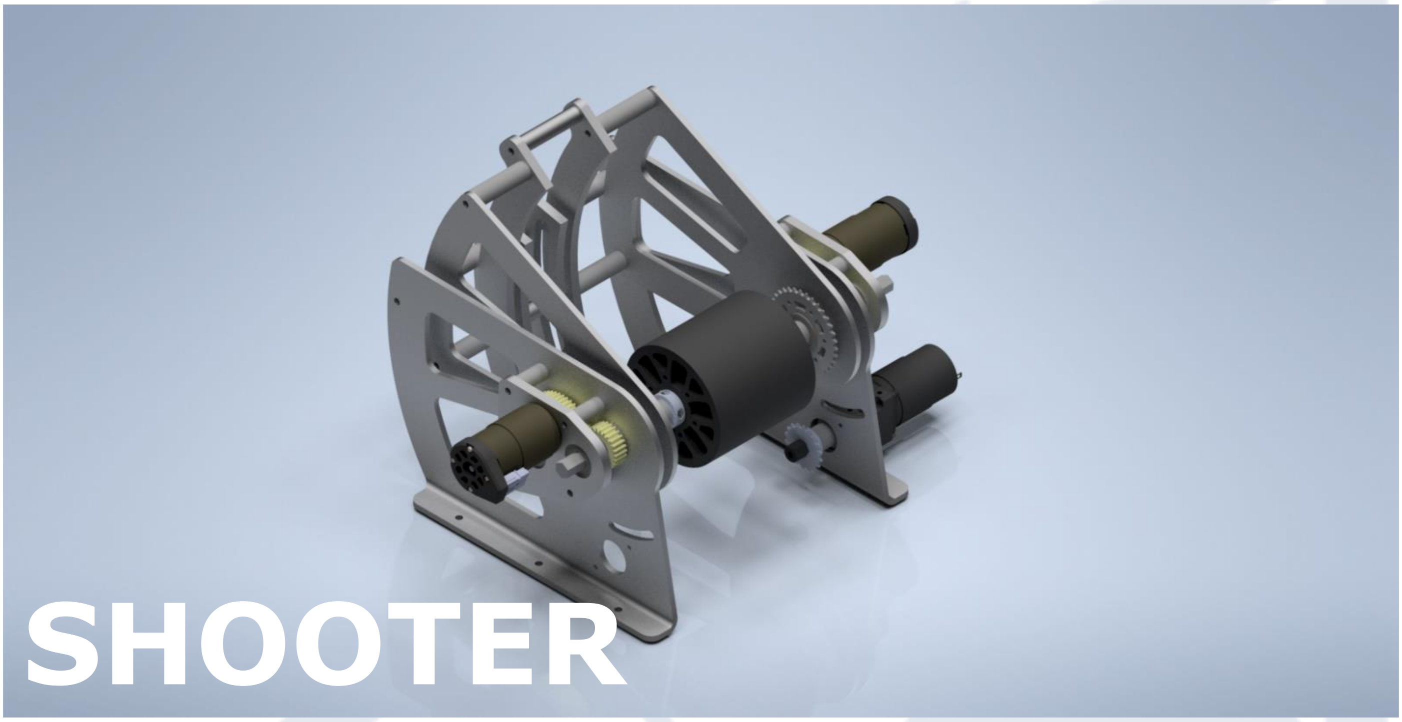
Prototype

For this mechanism, we were inspired by the design of a tank turret. So, we prototyped several iterations in order to have the smoothest rotation and the most accurate turret possible.

Final Design

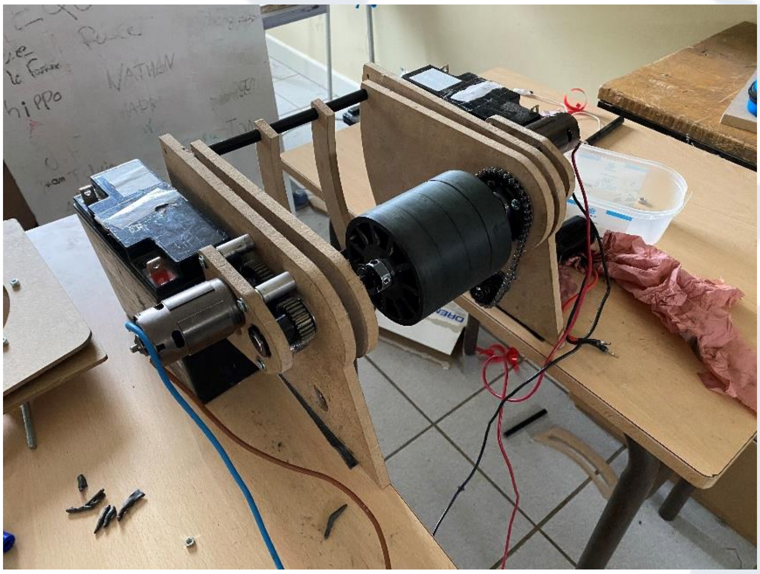
- One aluminum still disc attached to the drivetrain
- Two removable aluminum discs maintained by bearings
- Motorized by one BAG
- VersaPlanetary reduction of 177.5:1





SHOOTER

Prototype



To shoot power cells, we almost immediately came up with the idea of a shooter consisting of a single set of wheels in order to be more accurate by giving the ball a Top spin effect, causing a straighter trajectory and giving the shooter a better precision. With this kind of shooter, we can also choose the exit angle of the ball, enabling us to reach our power port from almost anywhere on the field.

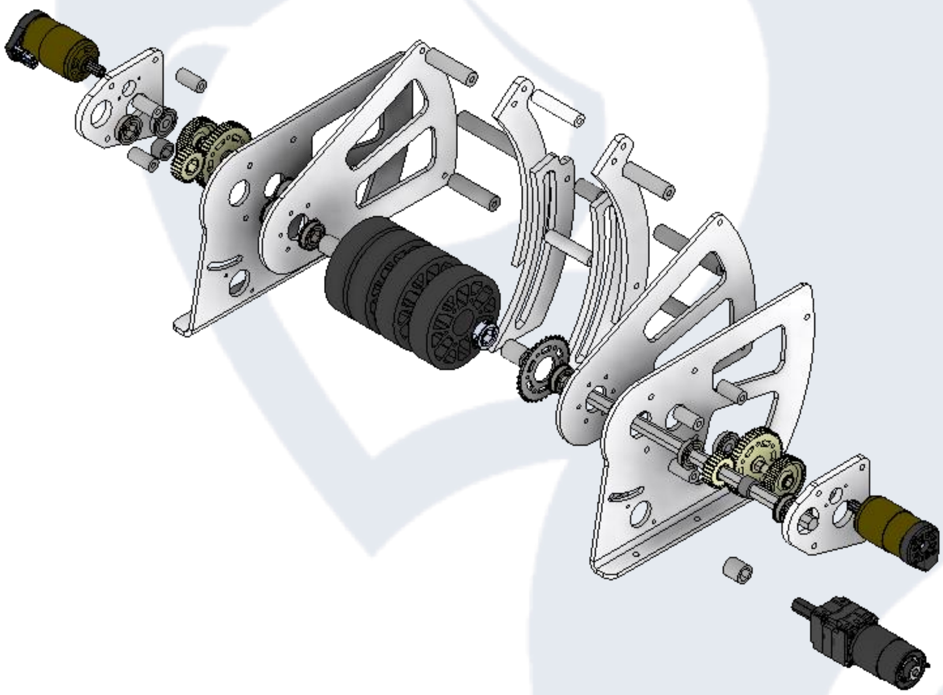
Final Design

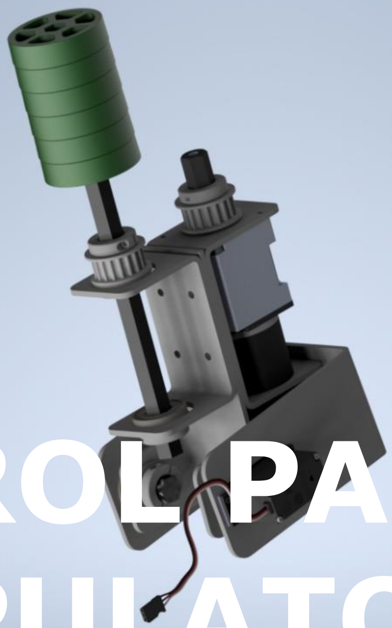
Shooter:

- ❖ Motorized by two 775 Pro
- ❖ Reduction of 2.5:1
- ❖ Ball propulsion ensured by four 4in Straight Flex wheels
- ❖ Flywheel discs for more efficiency

Adjustable Hood:

- ❖ An adjustable flap for control angle of the shot
- ❖ Motorized by one BAG
- ❖ VersaPlanetary reduction of 100:1





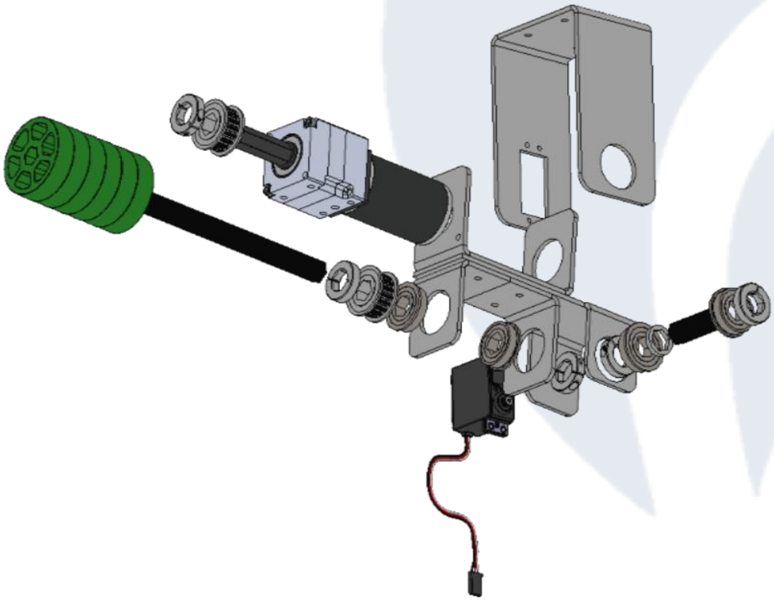
CONTROL PANEL MANIPULATOR

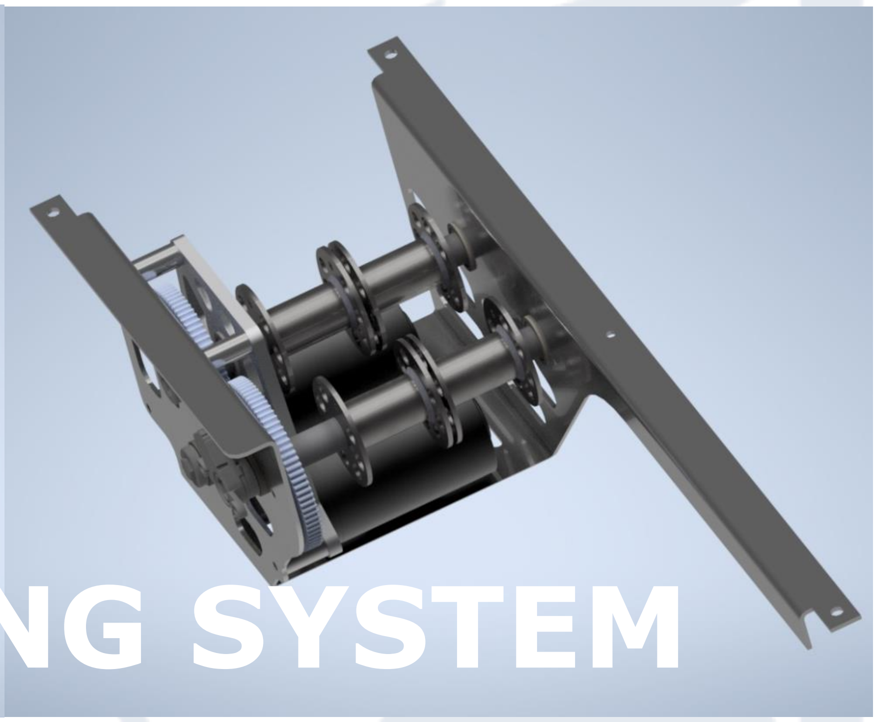
Prototype

As this mechanism design was pretty self-evident, we didn't really prototype this part of the robot and directly came with the final design in mind.

Final Design

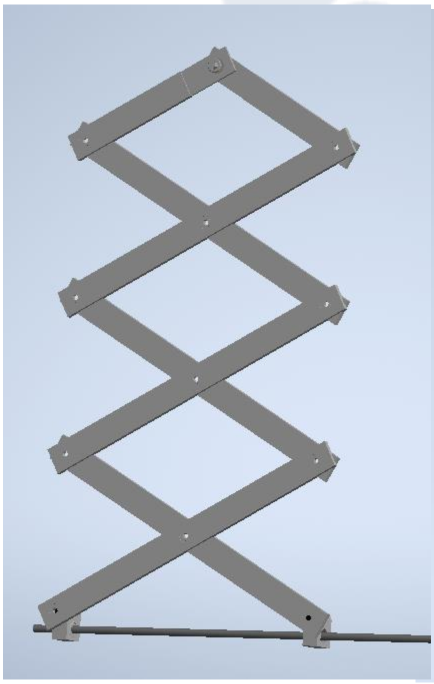
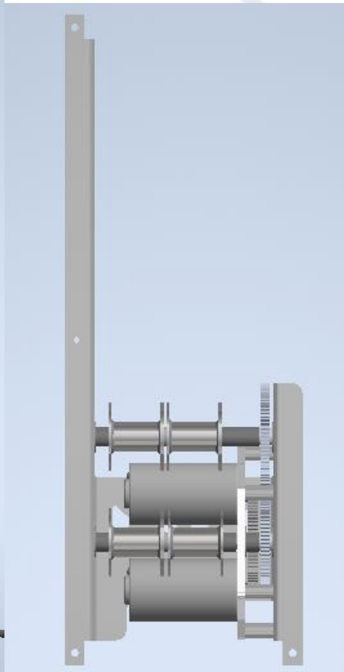
- ❖ Motorized by a bag motor
- ❖ VersaPlanetary reduction of 9:1
- ❖ Four 2in Straight Flex wheels
- ❖ Color sensor
- ❖ Motorized arm with a servo motor



CLIMBING SYSTEM

Prototype

To climb on the generator switch, we’ve envisaged two radically different mechanisms. The first in the form of a scissor lift. Although this design was the most practical, it was pretty big so we would have lost precious space for the other mechanisms. The other idea was a telescopic arm paired with a winch. This solution enables us to gain space as the winch isn’t mounted on the floor of the robot but rather on reinforcement higher on the drivetrain.

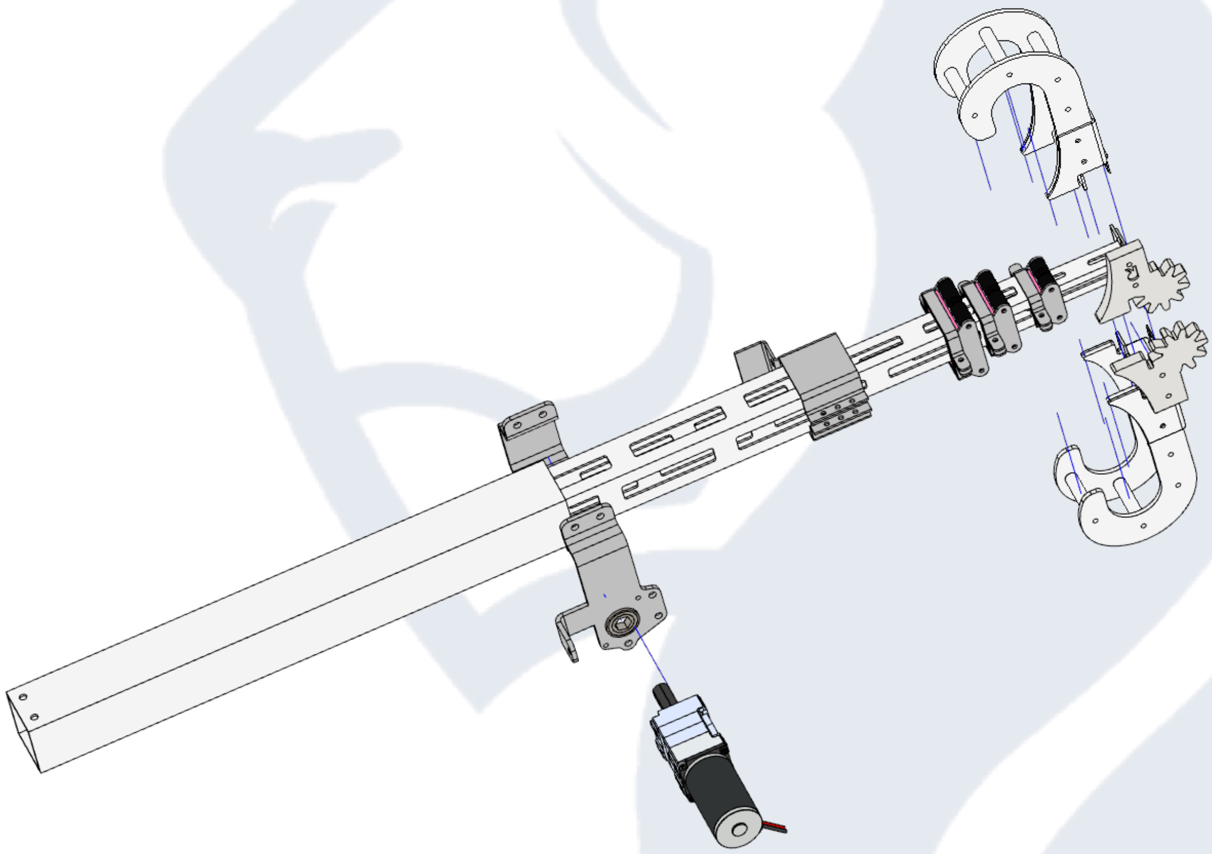
Final Design

Telescopic arm:

- ❖ Double crochet thereby grabbing the shield generator from both sides
- ❖ Crochet rise by 4 profiles and a cable system
- ❖ Motorized by a BAG
- ❖ Reduction of 10:1 using a VersaPlanetary gearbox

Winch:

- ❖ 3 cable drums.
- ❖ Intelligent cable management
- ❖ Motorized by a NEO Brushless motor
- ❖ Reduction of 37.3:1



III. PROGRAMMING

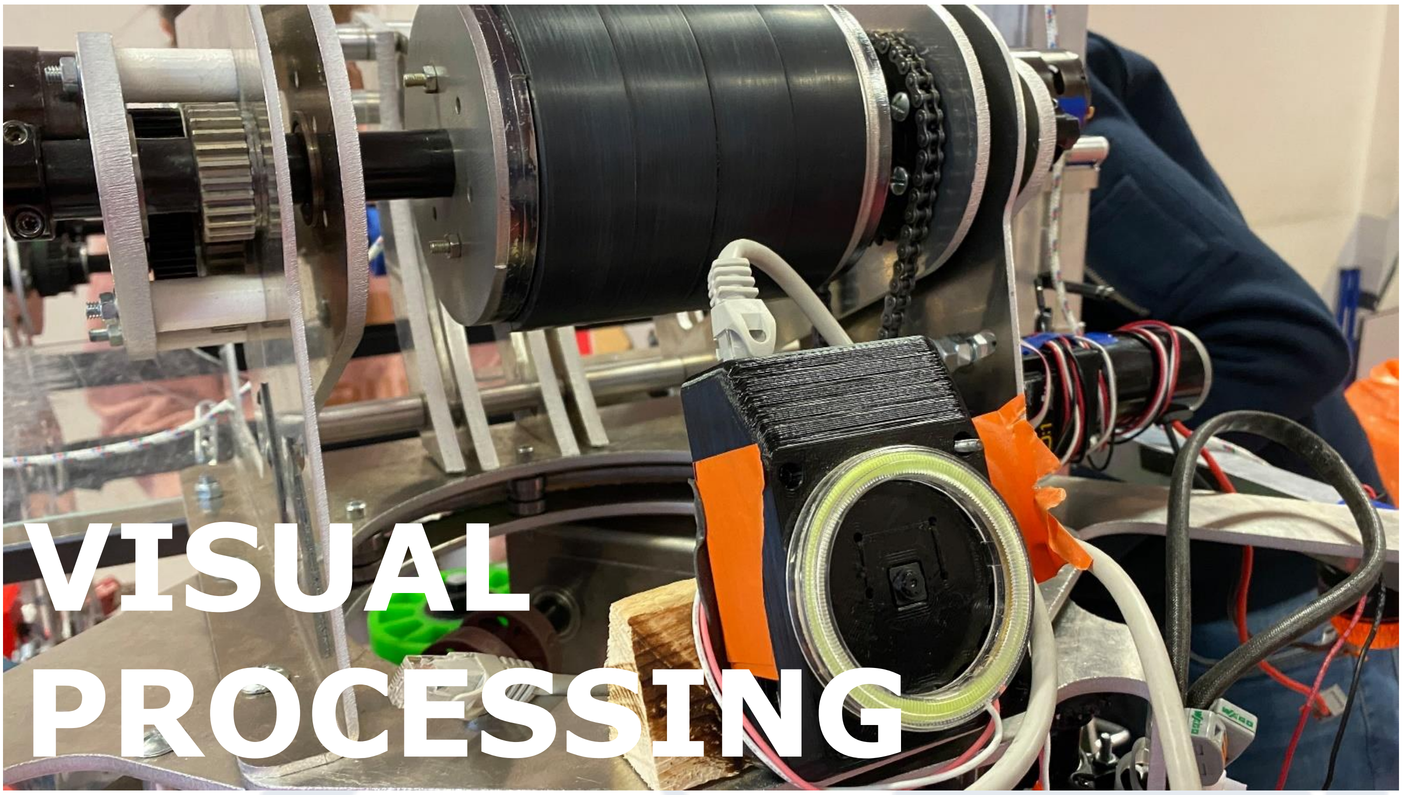
ROBOT'S PROGRAM

Like every year since 2017, we've coded our robot in C++.

This year, we also used PID controller to control both our turret and our adjustable hood. Indeed, the use of PID enables us to be more precise, in order to score as much in the inner port as possible. Moreover, our code includes ramps for every motor which make them accelerate gradually, therefore preventing them from being damaged.

We've also crafted a side controller consisting of five buttons, allowing to have more mechanisms controlled by the drive team. Thus, we can raise the intake or lower the adjustable hood in case of emergency. One other advantage is that the pilot has a lighter task during the match, and can focus on driving the robot and shooting the power cells.

Finally, we have ultrasonic sensors to guide our pilot in the tunnel, those are controlled by an Arduino which communicate with the Roborio over a I2C Bus.



VISUAL PROCESSING

This year, we focused our efforts on the visual processing, as it is particularly important in Infinite Recharge. We've decided not to use the Limelight camera, so we instead use a Raspberry Pi with the chameleon visual processing program which allows us to know our relative angle with the power port. The image is provided by a PiCamera surrounded by a green LED Circle.

We use the Open CV library for image processing and the Network Table library to communicate with the Roborio.

The process for image recognition is as follow:

1. Converting the image from RGB to HSV
2. Filtering each pixel according to its color
3. Image processing with canny filter
4. Contours detection
5. Contours filtering (according to their properties)
6. Contours coupling, creating a new target
7. Tracking the optimal trajectory to reach the detected target thanks to the use of odometry and motion control

Thanks to this process our turret and our shooter’s adjustable hood are always aligned to the target. We also have LED on the robot which indicate to the pilot, by changing colors, if the robot is aligned with the target or not.