Issaquah Robotics Society

2020 Engineering Notebook

Our Robot

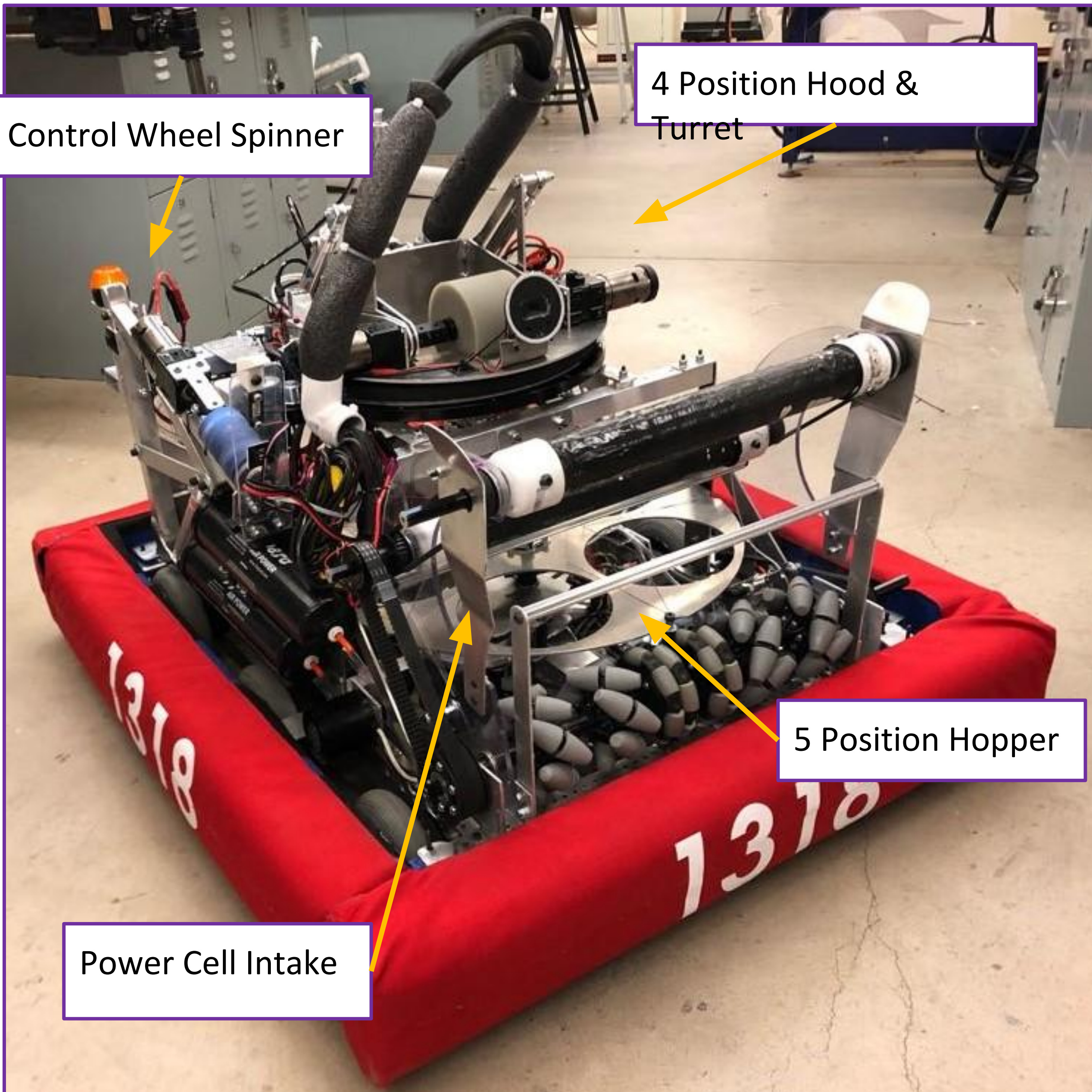

Control Wheel Spinner
4 Position Hood & Turret
5 Position Hopper
Power Cell Intake
1318
1318

Table of Contents

Engineering Story

One of the goals of *FIRST* is to encourage students to apply the engineering process to various problems. IRS Team 1318 believes that engineering is about developing and implementing a design both effectively and efficiently. We incorporate the engineering process into the design process for the entire robot, integrating engineering into entire meetings instead of just using this process for specific tasks.

We used the same adaptability and change required to be a good engineer to organize and run our team this year and accommodate the new FRC game challenge. To keep the design of the robot progressing, we organized into various sub-teams, each responsible for their own tasks, which were then integrated into the whole robot. The effective communication between our sub-teams allowed our robot to be designed and built with such precision that we were able to swiftly create a nearly identical second robot for competition use.

Our Product Cycle

To ensure efficient and effective engineering we follow a product cycle. This allows the IRS to continuously improve our robot while following a reliable process.

<u>Brainstorming</u>

1. Establish a strategy for the game.
2. Determine the most important tasks.
3. Conceive mechanisms that fulfill the tasks.

<u>Design</u>

1. Build CAD and physical mechanism prototypes.
2. Arrive at consensus on preferred designs.
3. Fine-tune and iterate prototype designs.

<u>Build</u>

1. Assemble e robot based off of prototype designs.
2. Fabricate & assemble mechanisms.
3. Build competition robot.

<u>Robot Evaluation</u>

1. Test each mechanism separately.
2. Run robot through integrated tasks.
3. Practice with competition robot at field.

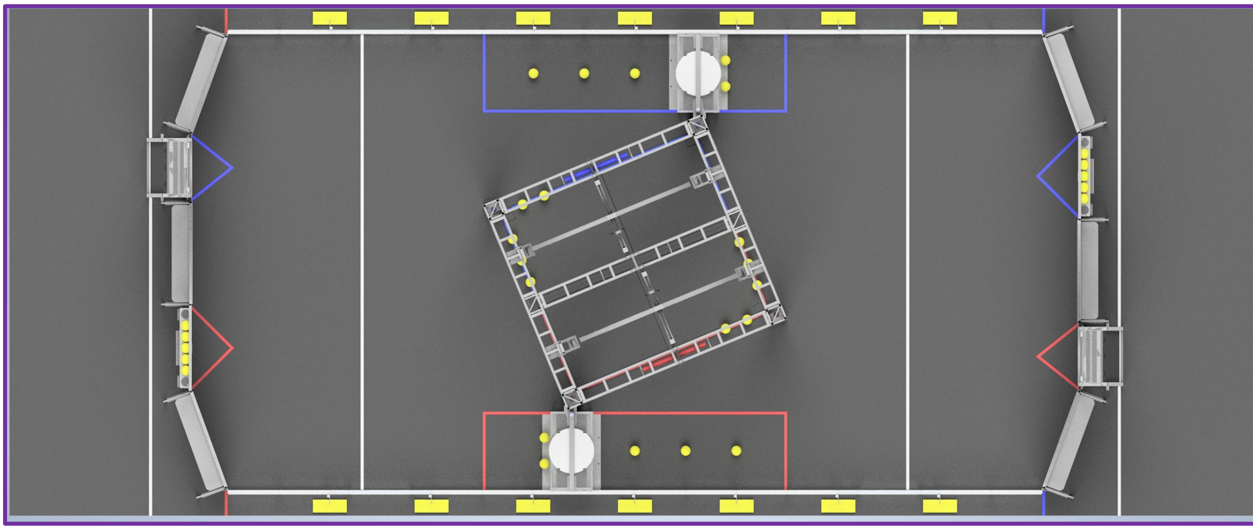
<u>Post-Competition Analysis</u>

1. Analyze recruitment, training, and build season.
2. Evaluate student leadership models.
3. Review our design and engineering processes.

The Game

Summary

INFINITE RECHARGE

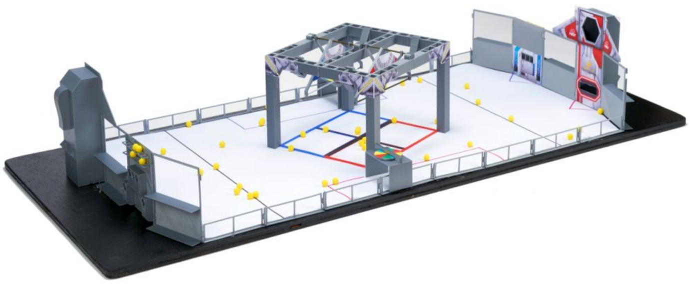


In INFINITE RECHARGE, two alliances work to protect FIRST City from approaching asteroids caused by a distant space skirmish. Each Alliance, along with their trusty droids, race to collect and score Power Cells in order to energize their Shield Generator for maximum protection. To activate stages of the Shield Generator, droids manipulate their Control Panels after scoring a specific number of Power Cells. Near the end of the match, droids race to their Rendezvous Point to get their Shield Generator operational in order to protect the city!

During the 15 second Autonomous Period, droids follow pre-programmed instructions. Alliances score points by: 1. Scoring Power Cells in the Power Port 2. Moving from the Initiation Line In the final 2 minutes and 15 seconds of the match, drivers take control of the droids. Alliances score points by: 1. Continue to score Power Cells in the Power Port 2. Completing Rotation Control 3. Completing Position Control 4. Hanging from the Generator Switch 5. Getting the Generator Switch to the level position The Alliance with the highest score at the end of the Match wins.

The summary above was extracted from the 2020 FRC Game Manual

The Game
Tasks & Points



Task	Autonomous	Teleop	Qual	Playoff
Initiation Line	5	----------	------------	---------
Lower Port	2	1	------------	---------
Outer Port	4	2	------------	---------
Inner Port	6	3	------------	---------
Control Panel Rotation Control	0	10	------------	---------
Control Panel Position Control	0	20	------------	---------
Park	0	5	------------	---------
Climb Shield Generator	0	25	------------	---------
Balance Shield Generator	0	15	------------	---------
Complete Shield Generator	------------------	----------	1 RP*	15
Complete Level 3	------------------	----------	1 RP*	30

*RP stands for Ranking Point

Strategy Development

Game Analysis - Points

Using a scoring matrix, we could test out different robot configurations to see which one would optimize our number of points.

Point-Analysis	
Accuracy(overall)	0.95
Accuracy into inner goal	0
Shots taken in autonomous	8
Inner points (auto)	0
Outer points (auto)	32
Reasonable number of cycles in teleop	5
Power cells / cycle	5
Power cells scored	23.75
Inner points	0
Outer points	47.5
Total shooting points	79.5

Strategy Development

Game Analysis - Timing

Using previous games, we could roughly identify how long given tasks would take, enabling us to make our scoring analysis more realistic.

Timing-Analysis

Realistic	
Action	Time (seconds)
Traversal to loading station	8
Traversal to goal	8
Pickup power cells	8
Shoot power cells	5
Time / cycle	29
Number of cycles	4
Climb	15
Rotation control (command center)	10
Position control (command center)	5
Shooting time	116
Other time	30
Total time	146
Time in match(tele op)	135

Strategy Development

Game Analysis - Design

Critical Robot Functions

- Autonomous
  - Reliably score the starting 3 power cells
  - Score at least 3 more power cells than we started with
- Power cells - shooting
  - Shoot reliably into outer goal with incidental scoring into the inner goal
  - Shoot all the power cells gone in 2 seconds or less
- Power cells - intaking
  - Intake power cells from the ground while driving at top speed
  - Intake power cells quickly and efficiently - “touch it, have it”
- Power cells - indexing
  - Index and serialize power cells without jamming
  - Index and serialize power cells in under 3.5 seconds
  - Indicate to drive team that the hopper is full

Strategy Development

Robots 1 & 2

Robot 1

Robot 1 is the first robot we build over the course of build season. It allows us to test designs that we think will help us succeed in competition. We bring Robot 1 to our first competition and it allows us to see if the designs we made function.

Robot 2

Robot 2 serves as a complimentary robot to robot 1. Robot 2 allows us to iterate and improve the design of robot 1, based on the information we learn at our first competition, thus allowing us to compete at a higher level.

Strategy Development

Approach To Design

Team Prototyping Decisions

Climber CAD Model

Intake Prototype

Drivetrain Fabrication

Full Robot Design

Our robot began in prototyping. To achieve our final robot, sub-teams formed, each testing out prototypes or systems which eventually developed into our final mechanisms and robot.

Task Identification

Team Prototyping

Prototype Evaluation

Focused Prototyping

Robot Mechanism

A summary of our prototyping process

Hardware

Drive Train

Needs	Wants
Drive through the trench.	Maximize maneuverability
	Lightweight
Drive over rendezvous point bumps.	Balance maneuverability and stability

Possible Ideas

8 Wheel Kitbot	6 Wheel Kitbot	West Coast Drive
More maneuverable than 6 wheel	Fast	Fast
Complex	Simple	Simple
Easiest navigation over static defenses	Wheels not exposed from sides	Lightweight with exposed wheels
More points for failure	Large	Custom components and fabrication required
High traction	Reliable movement	Reliable movement

Notables

- The drive train is powered by four Falcon 500 motors
- The drivetrain has PID control for position and velocity control, as well as a PID-based break mode

Hardware

Drive Train

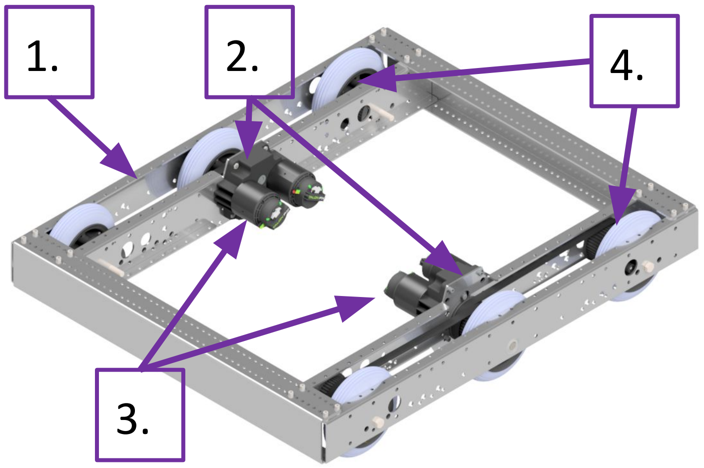
Falcon 500 Drive Train Design

The drive train this year is a 6-wheel differential drive that utilizes six 6” pneumatic wheels. It also has 4 falcon 500 motors coupled with a 8.46:1 gear ratio(about 13.56 feet/second max) within a toughbox mini gearbox.

We started with a KitBot frame (AM14U4), but have made a couple of modifications, including:

- Decrease the width of the frame by 0.5 inches.
- Additional structural cross beams.

The Final Product



Iur modified AM14U4 Chassis

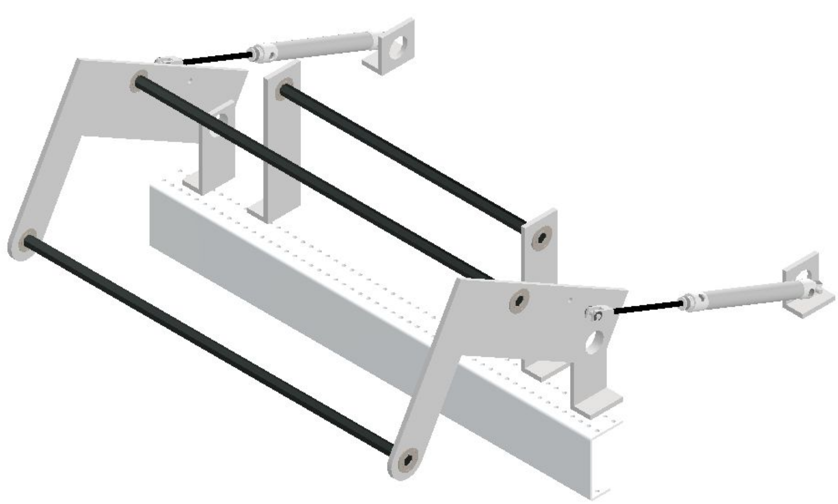
Parts:

1. 1x AM14U4 Kitbot frame.
2. 2x Toughbox Mini Gearboxes.
3. 4x Falcon 500 Motors.
4. 6x Andymark 6 inch Pneumatic wheels.

Power Cell System

Intake

Needs	Wants
Smoothly grab and transfer to Hopper	Intaking with little driver accuracy
No possibility of jamming or easy way to remove jams	Intaking 5 power cells at once
Easy way to accomplish ground and station pick-ups	Intake easily while traveling at max speed



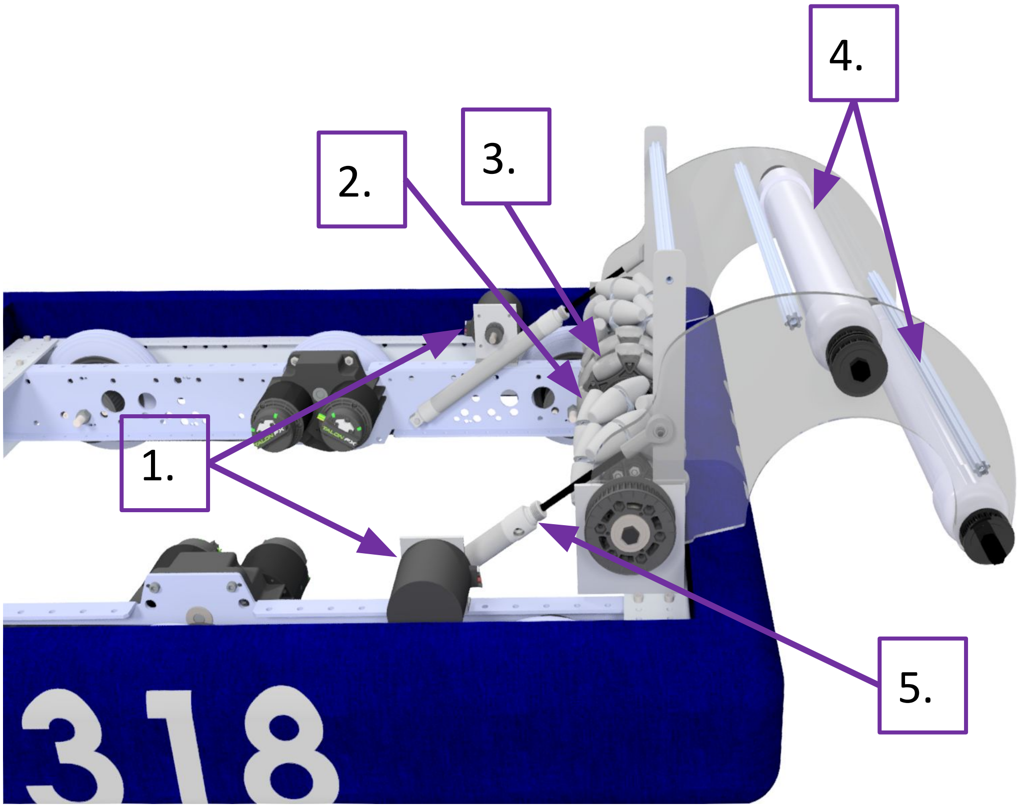
An early CAD model of our intake linkage

Notables

- PVC rollers for greater surface area

Power Cell System

Intake



A side view of our intake assembly

Features:

- Over-the-bumper power-cell intake
- Retractable
- 17-inch width to intake multiple power-cells in one instance

Parts

1. 2- REV NEO motors
2. 4- 4” Mecanum wheels
3. 2- 4” Omni-wheels
4. 2- PVC rollers
5. 2- 5” stroke pistons

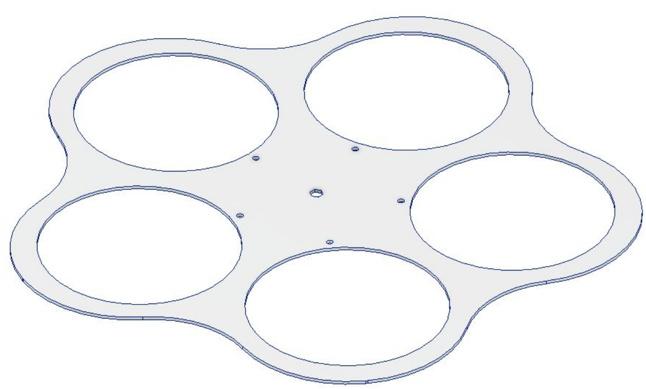
Power Cell System

Hopper

Needs	Wants
Index/serialize power-cells.	Have the ability to outtake 5 power-cells in 1 second.
Pass through power-cells from the intake to the shooter.	Manufacture the hopper from aluminum.

Possible Ideas

Serializer	5-slot Rotary Indexer
• Simple • May exhibit a problem of jamming power-cells	• Slightly more complex than a shooter • Indexes power-cells efficiently without the issue of jamming



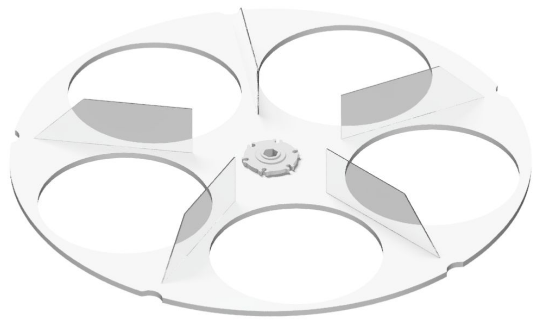
Initial CAD model for our 5-slot indexer

Notables

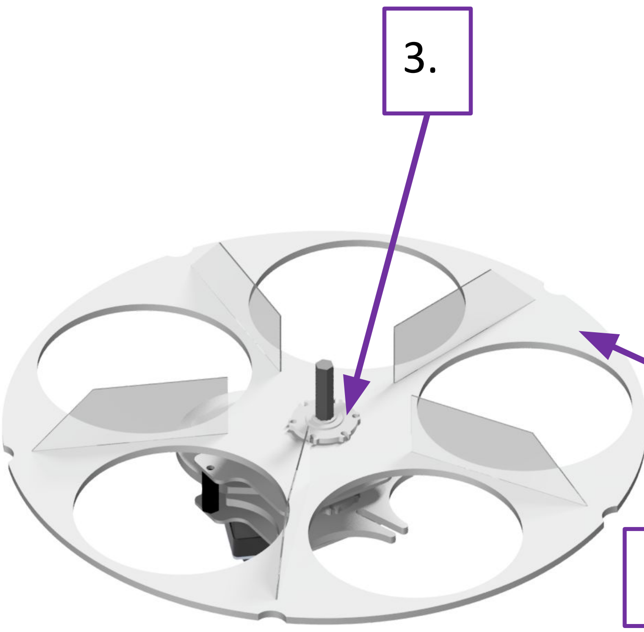
- Our hopper is driven by a unique Geneva mechanism that allows us to rotate the hopper in exact increments, giving us both accuracy and precision without the need for a programmatic approach such as PID.

Power Cell System

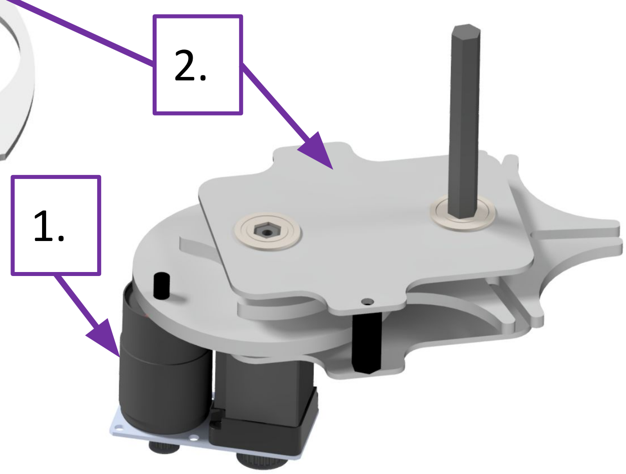
Hopper



Our hopper assembly



The hopper assembly attached to the geneva mechanism



The geneva mechanism used to spin the hopper

Features:

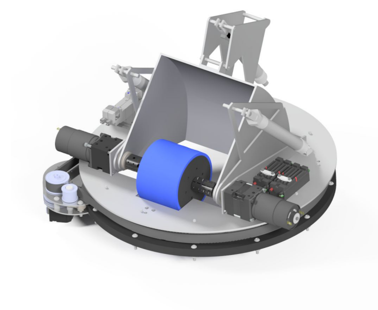
- A 5-slot power-cell indexer
- A 775 Pro motor with a versaplanetary 180 drive gearbox
- A Geneva Mechanism for incremental rotary motion

Parts:

1. 1- VersaPlanetary 180° drive kit with a single 775 pro motor
2. Fabricated aluminum plates
3. 2- hex hubs

Power Cell System

Shooter



Needs	Wants
Ability to shoot the ball into the outer port	Ability to shoot the ball into the inner port
Limited 90° degree turret motion	Limited 380° turret motion

Possible Ideas

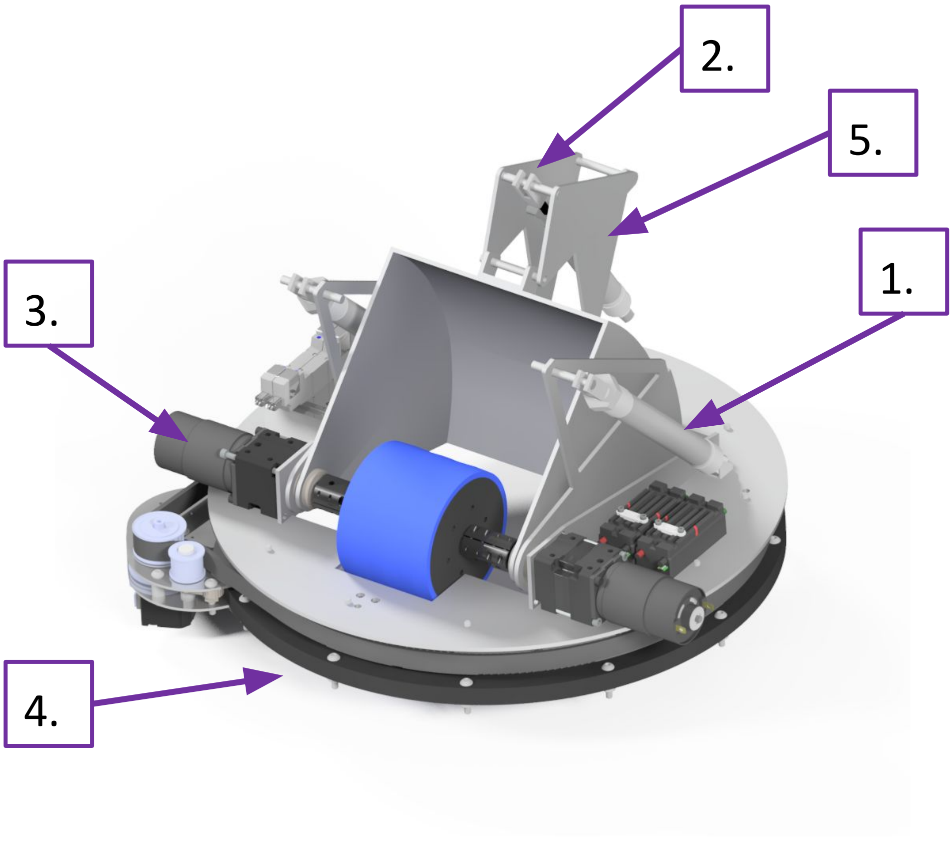
2 Piece Hood	3 Piece Hood
Pros	**Pros**
• Simple • Multiple positions to shoot from	• Can shoot from almost anywhere on the field • Smaller and collapsible

Notables

- Continuous, real-time tracking of retro-reflective targets
- Retractable hood for a low profile
- Turret for versatility in shooting positions

Hardware

Shooter



Features:

- A variable hood that allows for multiple shooting angles.
- An Armabot Turret240 for increased maneuverability.
- Two 775 pro motors for fast shooting.

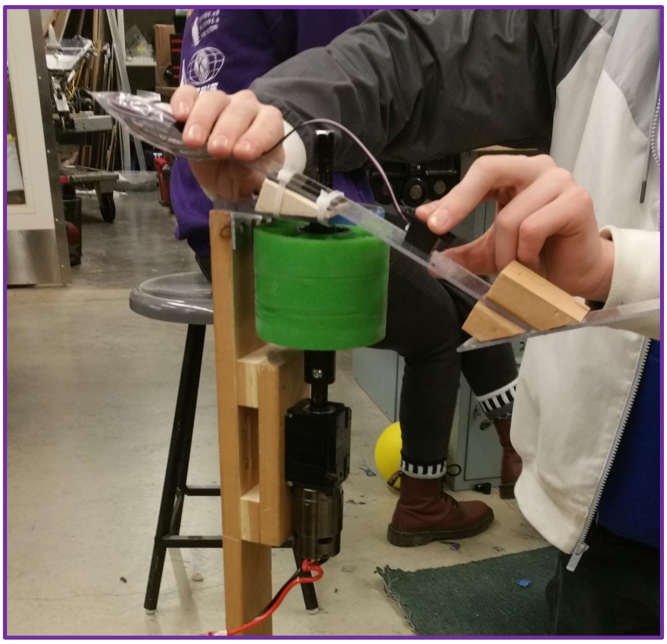
Parts

1. 2- 2.5” stroke pistons
2. 1- 2” stroke piston
3. 2 direct drive versaplanetary gearboxes with 775pros
4. Armabot Turret240
5. 4-bar kite linkage

Hardware

Control Panel Spinner

Needs	Wants
Reliably spin the control panel wheel.	Retract below the plane of the shooter.
Spin control panel wheel at 30 rpm.	Autonomously identify color sectors.



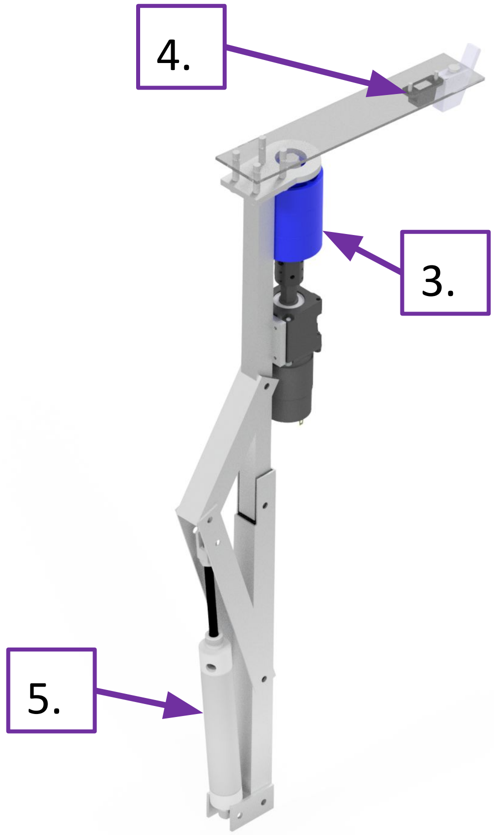
An early prototype of the control panel spinner

Notables

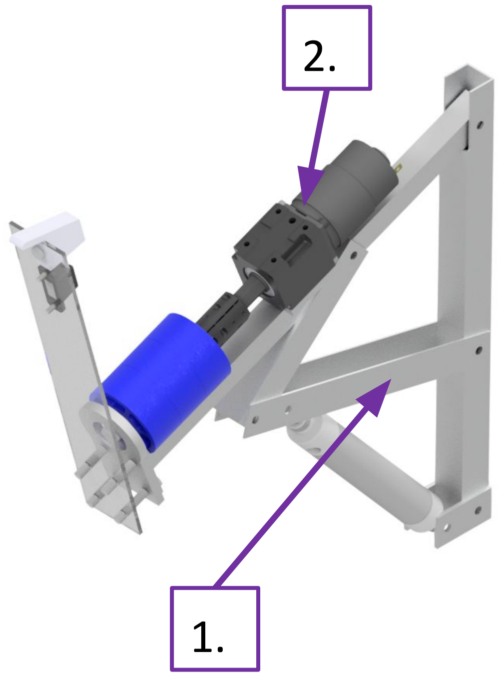
- Manipulator deploys using a unique 4-bar linkage designed for optimal range, strong reliability & robustness without the use of a motor.

Hardware

Control Panel Spinner



The control panel spinner extended



The control panel spinner retracted

<u>Features:</u>

- A retractable design to keep the robot's profile low
- Fast and effective interaction with the control panel
- Piston for quick actuations
- Autonomous capable

<u>Parts:</u>

1. 4- Aluminum beams
2. 1- 10:1 Versaplanetary drive with 775 Pro
3. 3- 2" Stealth wheels
4. 1- REV color sensor
5. 1- 4" Stroke piston

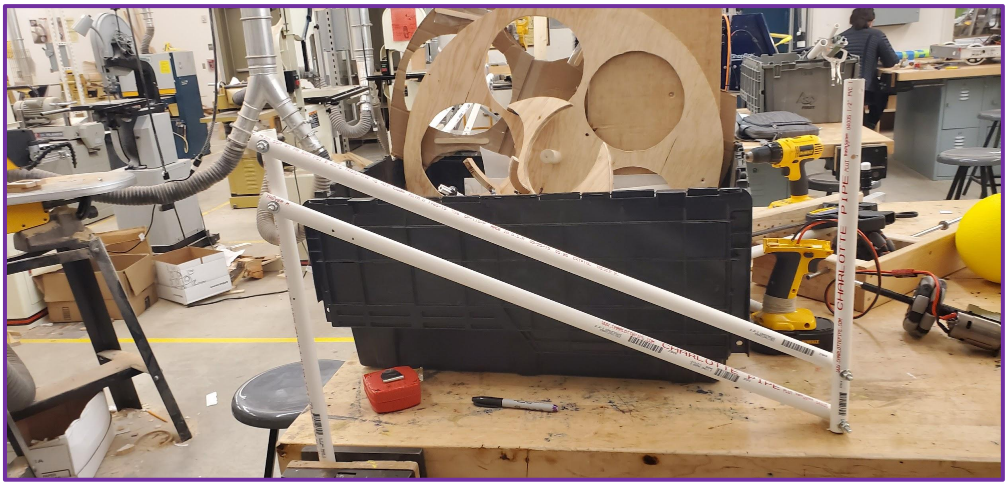
Hardware

Climber

Needs	Wants
horizontal climb	side climb
middle climb	lightweight, space efficient

Possible Ideas

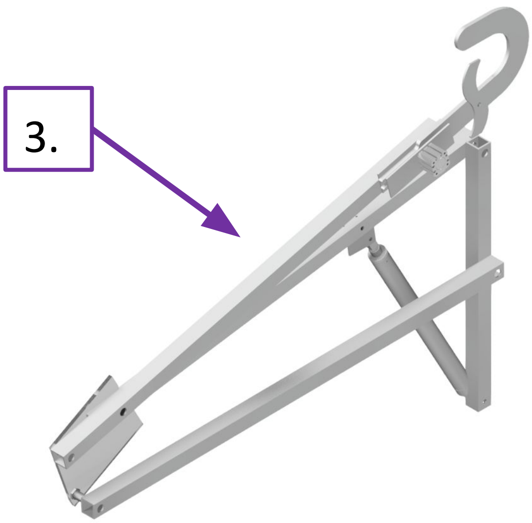
Elevator	4 bar linkage	Rack and Pinion
Cascading rigid climb	Hook deployment with winch and pulley system to keep level	Hook deployment with winch



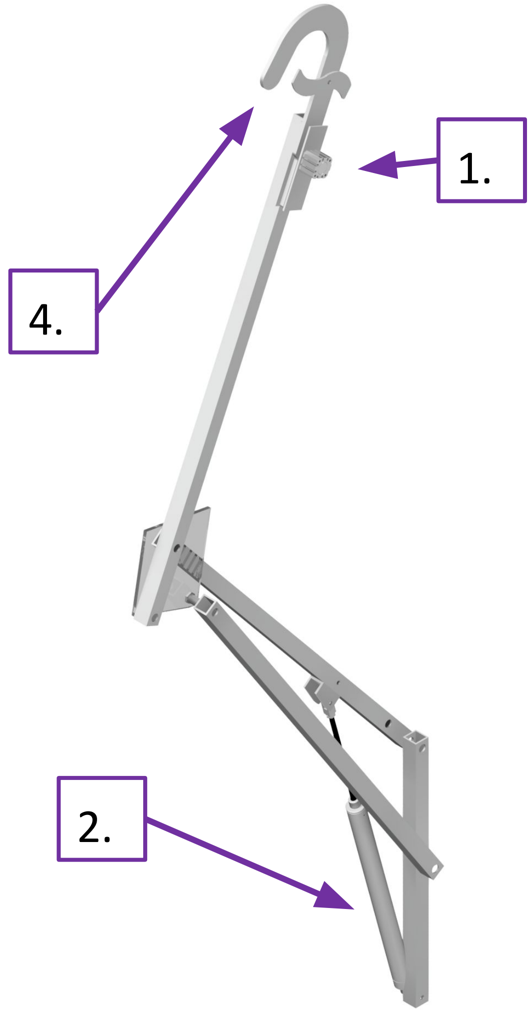
An early prototype of the climber's 4 bar linkage.

Hardware

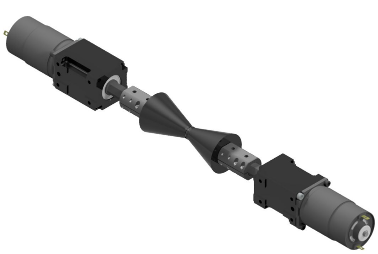
Climber



The climber linkage retracted



The climber linkage extended



The winch to rope up the robot

Features:

- An extendable 4-bar linkage
- A deployable aluminum hook
- A pancake piston to hold the hook in place.
- A winch to rope up the robot

Parts:

1. Pancake piston
2. 8” stroke piston
3. 1”x1” Aluminum box beam
4. Aluminum hook and grip

Electronics & Pneumatics

A section of the left-side electrical board.

<u>Electronics Notables:</u>

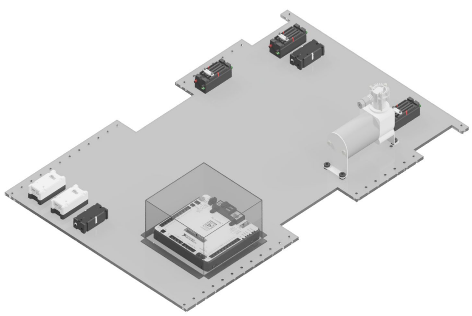
- 3 electrical boards
- Polycarb RoboRIO box for protection
- 2 CAN terminals, prevents RoboRIO from losing connection
- Some electrical components mounted on relevant mechanisms to decrease complexity

Electronics & Pneumatics

Left board	**Right Board**
Our left board contains electronics such as 2 PCM’s, a relay, and one of the manifolds and solenoids.	The right board contains the Power Distribution Panel, and the Radio in mounted above the right board.

BellyPan board

The BellyPan board is a polycarbonate board containing 7 motor controllers, CAN terminals, a RoboRIO, and a Raspberry Pi and a VRM.



<u>Pneumatic Notables</u>

- 2 PCM’s
- 2 Manifolds
- 2 double solenoids mounted on the turret
- Three air tanks
  - 2 on one side
  - 1 on the other
- The ball manipulation system contains 6 pistons
- Air compressor and solenoids for 9 pneumatic cylinders

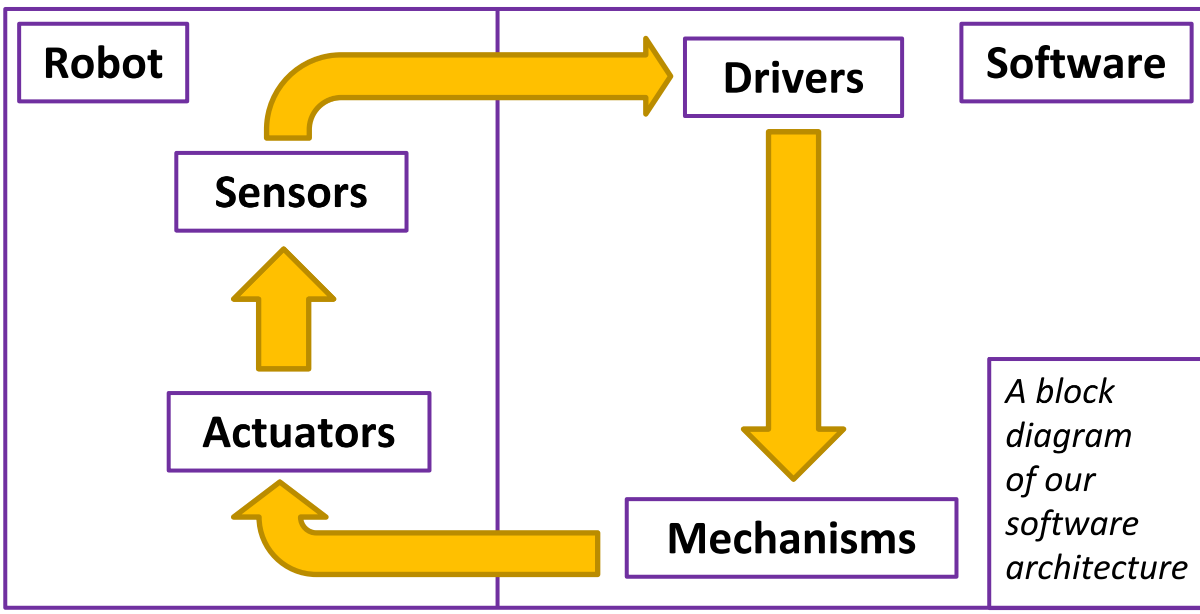
Software Architecture

Notables

- ✓ Macros
- ✓ PID feedback control
- ✓ Autonomous
- ✓ State machines
- ✓ Vision System

Dual input controllers with motion control

Our team uses Java for robot programming and utilizes Github as our source code repository.



A block diagram of our software architecture

Vision

Vision allows us to detect the retro reflective tape on the outer port of the scoring walls to align ourselves to be able to shoot power cells accurately. 2020 is the first year we have vision on a Raspberry Pi instead of on the RoboRIO, which will increase the framerate.

Software Architecture

Proportional Integral Derivative (PID)

- Velocity PID control allows us to regulate the velocity of the drivetrain to deal with inaccurate movement.
- Positional PID control gives us the ability to move our drivetrain to a specific position and maintain that position.

Smart Dash

Smart Dash receives information from the robot and displays selected data. This allows us to identify and/or solve problems more quickly.

Software Brakes

Virtual brakes to keep the robot still and utilizing PID

Scouting Network

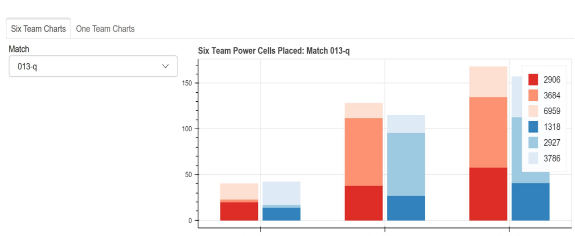
About

Our scouting network allows us to collect large amounts of data on the abilities of other robots. We then use this data in making better game strategies as well as for choosing teams during alliance selection.

Programming

The server app of our scouting program was coded in Java JavaScript, and Typescript. The data is stored in a database using Python in tandem with SQL. The graphs are then created using a Python graphing package called Bokeh and displayed on a dynamic server.

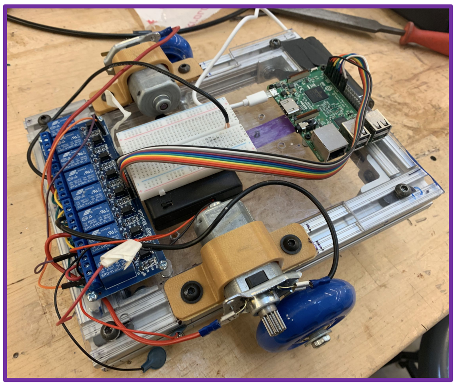
Data Analysis



The Bokeh server for generating dynamic graphs

2020 UI Design for the Teleop page of the Scouting System

Extras



A Raspberry Pi training robot

Fall Training

Before kickoff day students and mentors prepared for the build season ahead. Classes included fabrication, electronics, programming, and CAD.

Fall Training Robot

Our new members are split into different teams that are tasked with creating a small robot to compete in a small scale FIRST style game to prepare them for building a robot.

Field Pieces

To organize for the game without a field, we built the various field pieces out of wood to practice with a more accurate representation of the field.

CAD

We use CAD to design parts and assemblies to make a completed robot model before its fabrication.

Thank you to all of our sponsors!