Table of Contents

1. Introduction

Team 5401, the Bensalem High School Fightin' Robotic Owls, is thrilled to be participating in 2019 Destination: Deep Space. Last year's season, Power Up, was not as successful as the previous, Steamworks season. Fortunately, that past year had been an incredible growth opportunity for us as a team, and we will be coming back happily. We have welcomed a huge group of new members onto our team, including students from Neshaminy High School in the hopes for them to start their own FRC team next year. Our more hands-on student involvement, led by our expanded Student and Mentor Leadership positions, demonstrates our goal of teaching STEM and business values, inspiring younger generations, and preparing students for life after high school. This year, Team 5401 presents, Mr. Krinkle. Mr. Krinkle is named after a song from the band Primus, because this year's game is set on Planet Primus.

2. Kickoff Weekend

Our kickoff weekend began by sending a small group of about 10 students to Lenape High School. The rest of the team reported to Bensalem High School where they watched the kickoff video and immediately began reading the rules and game manual. By the end of Day One, we had determined our overall strategy for playing the game, and students were sent home with homework to read the game manual and ensure they have a full understanding of the year's game. Day Two was spent entirely on robot design, where we discussed potential robot systems. At the end of Day Two, we had concluded with a tight collection of robot designs we wanted to approach and came up with the finalized one through prototyping in the week one of build season.

3. Robot Design and Control

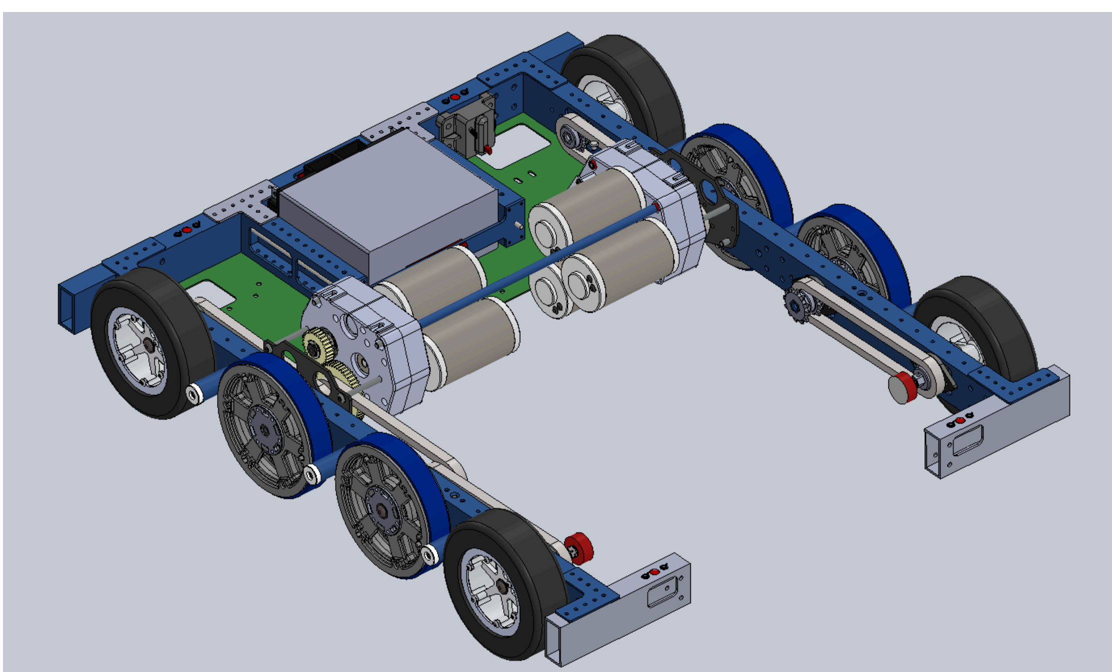
3.1 Drive Train

3.1.1 Initial Design and Prototyping

We started off Build Season already knowing that we wanted to have a West Coast Drive (WCD) system with either 6 or 8 wheels. After looking at the field model and specifically at the Habitat 1st tier angle, we decided to have an 8 wheel drive base. 8 Wheels will work perfectly for the angle of the Habitat and we wouldn't get stuck on it when driving on or off.

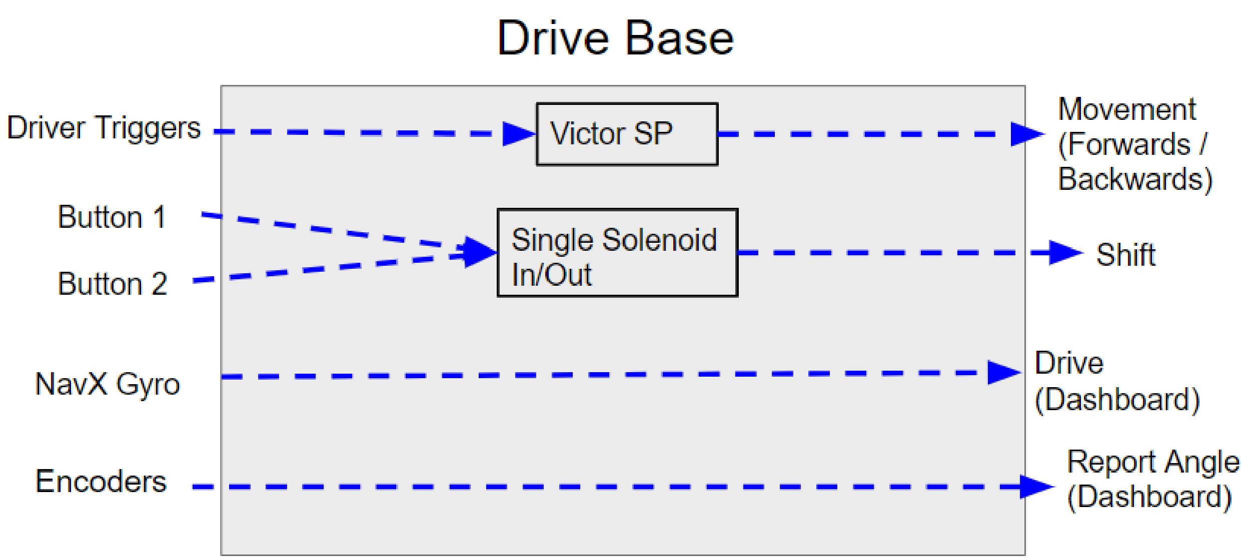
3.1.2 Final Design

Our robot is a West Coast Drive with 8 wheels. The wheels are driven by 2 3 CIM motors. The CIMs are Ball Shifting Gearboxes (217-3195). The low gear ratio is 34:50. The high gear ratio is 54:34. The wheels are 6 inch diameter pneumatic wheels and they help our Sandstorm jump down from tier 2 and they also help us bounce up for tier 2 in Endgame. Our robot's frame perimeter is 32 x 27.5 inches. The drivetrain uses #35 chain and 12 tooth sprockets and with six 19.5 inch chains. The frame rails are .1 thickness 1 x 2 aluminum box tube, riveted in the corners by VexPro L-Gussets and a 1/16" lexan plate on the back end of the drivebase.



3.1.3 Control System

This uses two Victor SP's as well as a Gyro and Encoders. One notable thing is that we configured the Speed Controllers as PID controllers through the use of FRC libraries allowing us create an Auto-Drive and Auto-Turn function using PID. In teleop, the driver uses Grand Theft Auto style tank controls. The gear transmission is manual.



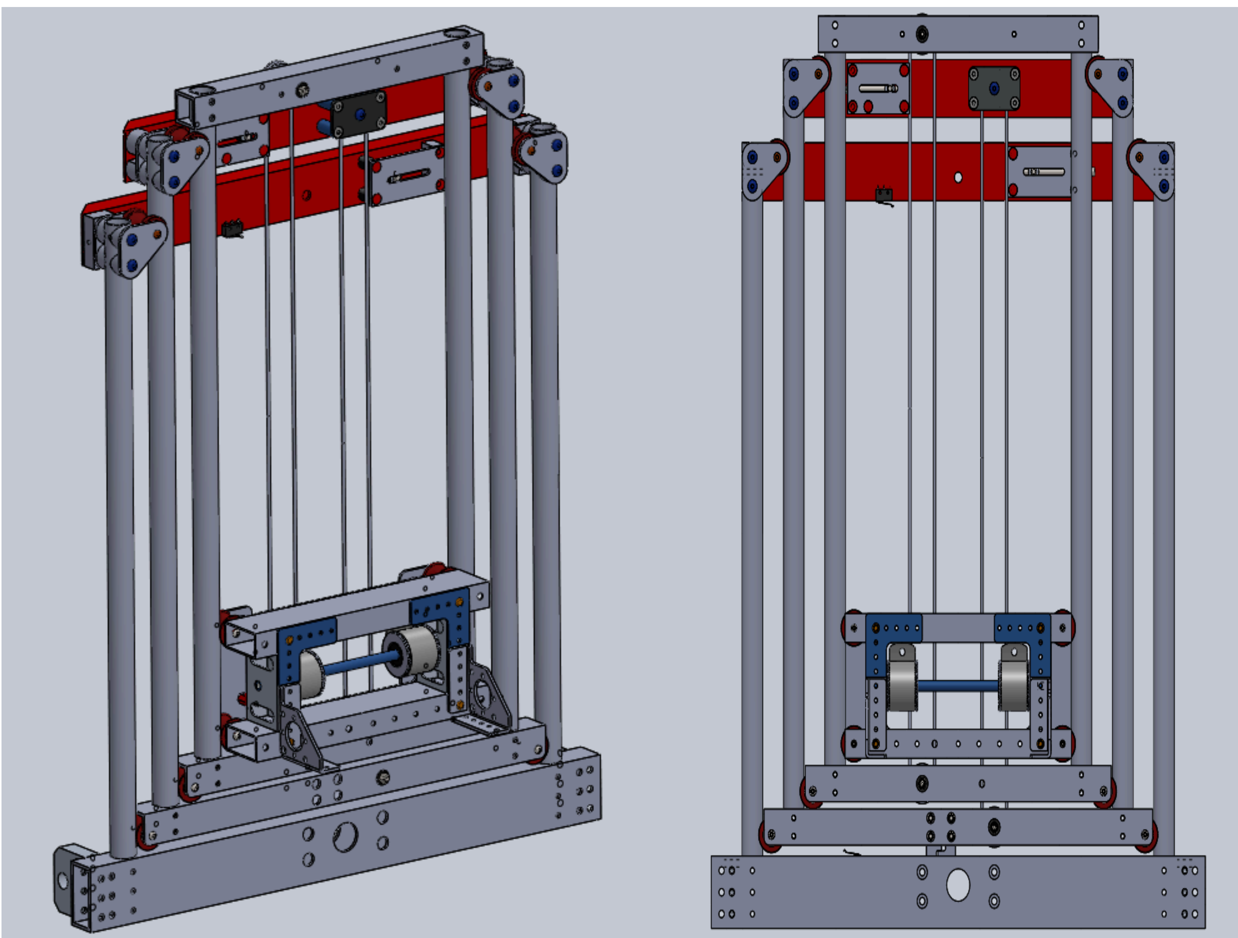
3.2 Elevator

3.2.1 Initial Design and Prototyping

We started Build Season off not knowing if we wanted to go for high game or low game. After multiple designs and changes we finally decided to go for High Game with an elevator. We prototyped a few of the 3D printed parts to see if they would be strong enough to work on the robot and it was concluded that they would work. We printed out the elevator rollers and the rope pulleys and they worked.

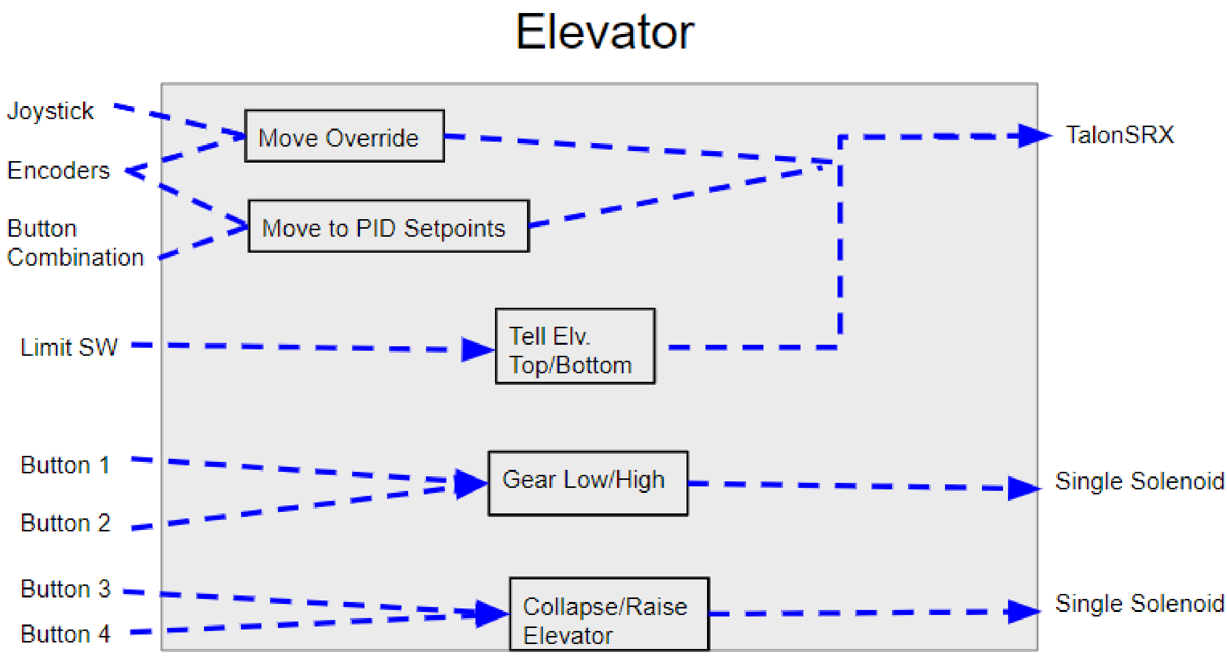
3.2.2 Final Design

Our final design is a cascading elevator with 3 stages and a carriage in the middle. The first stage stays still and is mounted to the drivebase by a pivot system. The elevator is driven by a 2 CIM gearbox. The stages of the elevator are connected together by pulley systems. We used 3D printed pulleys that cords wrap around. We use #35 chain that is cut short to have a chain tensioner to drive the first stage and the other stages with it. The cords are wrapped around systems and connecting by a screw on each stage to make the stages go up at the same time. The Elevator is able to pivot back on a 90 degree angle. Actuators are attached to the pivot system of the elevator so when they actuate, they bring the elevator up and down. When collapsed, the elevator should be a little more than a foot off the ground. This will allow us to start flat up against the alliance wall to maximize Sandstorm points and fit another robot on the second tier with us.



3.2.3 Control System

This system takes advantage of two pneumatic actuators to raise itself off of the robot's frame. Two TalonSRXs within a master and slave configuration are also used, which can use either move the elevator to PID setpoints. In case of a failure a manual override function to move the three stage elevator was also implemented.



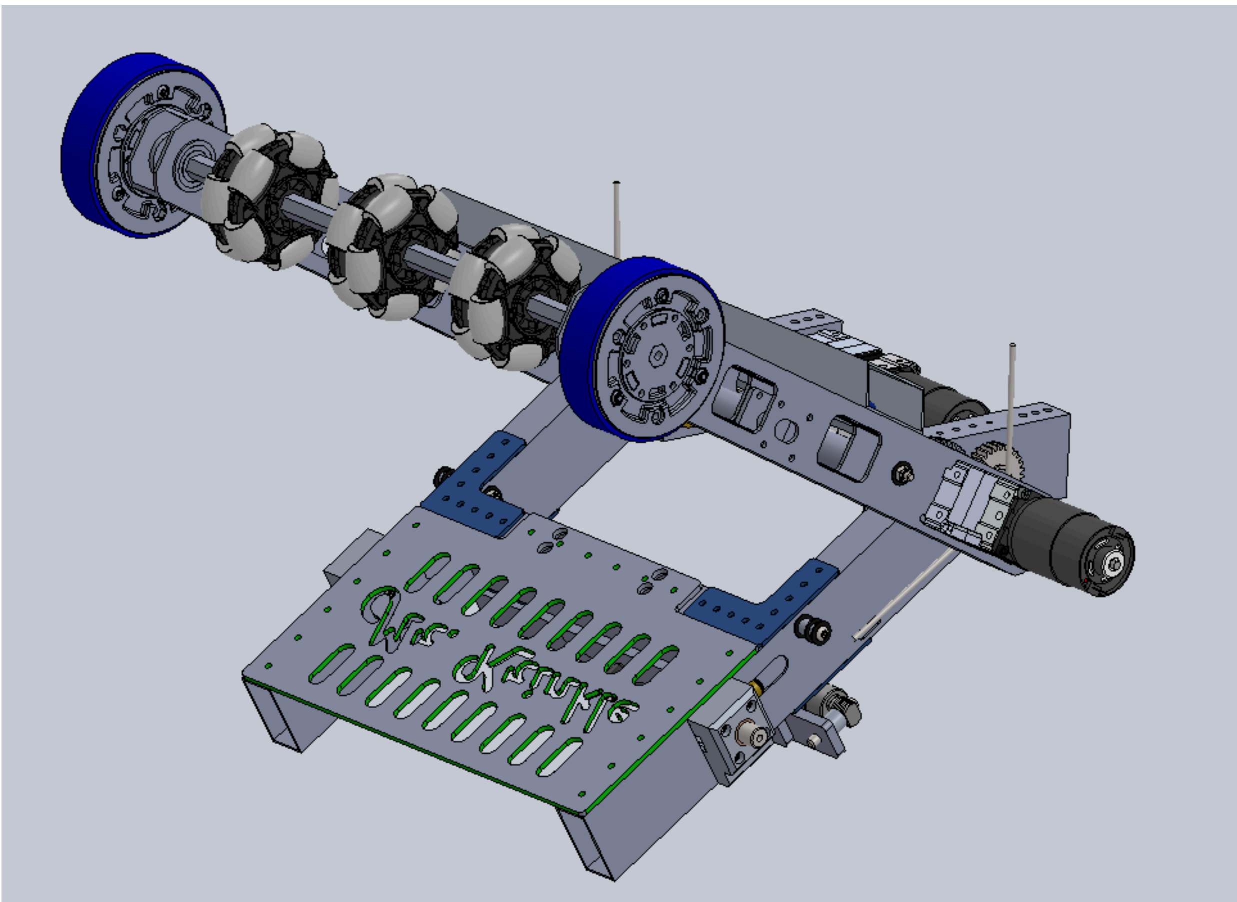
3.3 End Effector and Carriage

3.3.1 Initial Design and Prototyping

On Kickoff Weekend, we got into small groups and brainstormed multiple different end effector designs. We originally thought we needed two completely different systems for the hatches and the cargo, so we made designs for both of them and tried to find a way to mesh them together into one. We tested a hatch ground pickup by driving a piece of lexan under a hatch and rolling in the hatch by wheels. We tested it and in the end didn't use it.

3.3.2 Final Design

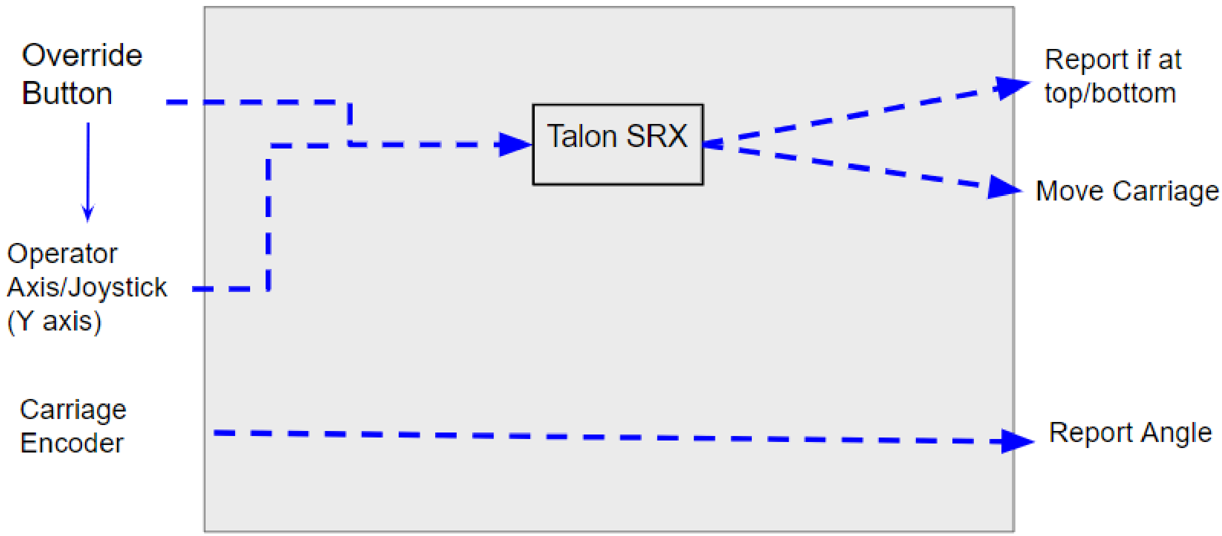
The final design of the end effector is a combination of two systems, the ball infeed and the hatch mechanism. The ball infeed's movements depend on the movement of the hatch mechanism. The two systems are connected by rope. The hatch mechanism is driven by a 490:1 gearbox and a series of two 30 tooth and a 48 tooth gear to give it the torque it needs for Endgame. When the gearbox turns and moves the hatch mechanism forward, it will move and once it hits a certain angle, the rope will be fully tethered and the ball infeed will move with it. The ball infeed includes three 3.25 inch omni wheels on the inside and two 4 inch traction wheels on the outside. The wheels are driven by a 35:1 gearbox. The hatch mechanism has two pneumatics that are attached to the “fingers”. Our fingers are what the hatch goes on. This will secure it in place and will not move.



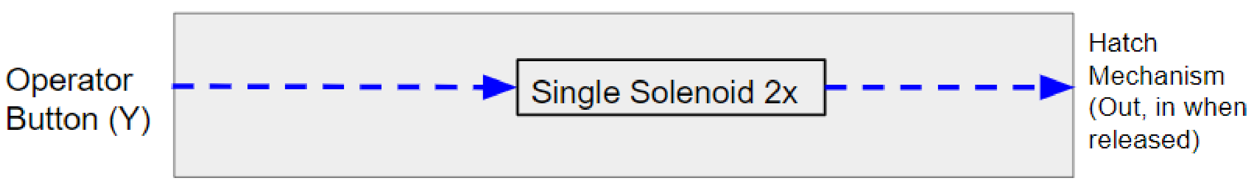
3.3.3 Control System

This end effector was split up into 3 different subsystems as to make it easier to comprehend. The hatch mechanism within this end effector takes advantage of two pneumatic actuators which allow for the grabbing and releasing of hatches during gameplay. Next, the cargo infeed on this end effector implements one VictorSP to rotate a bar allowing for the infeed and export of cargo. Finally, the last subsystem was the carriage subsystem, which is controlled by a single TalonSRX, which allows the entire effector to move out of it’s collapsed position and collect more game pieces.

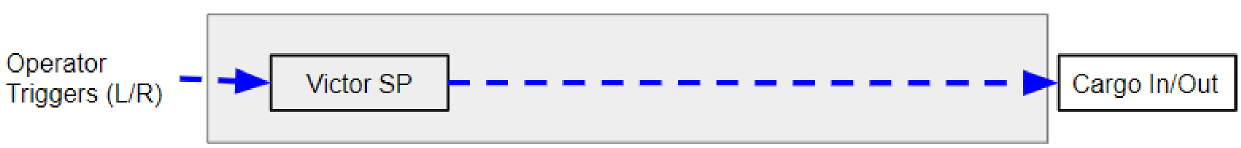
Carriage Mechanism



Hatch Infeed/Outfeed



Cargo Infeed/Outfeed



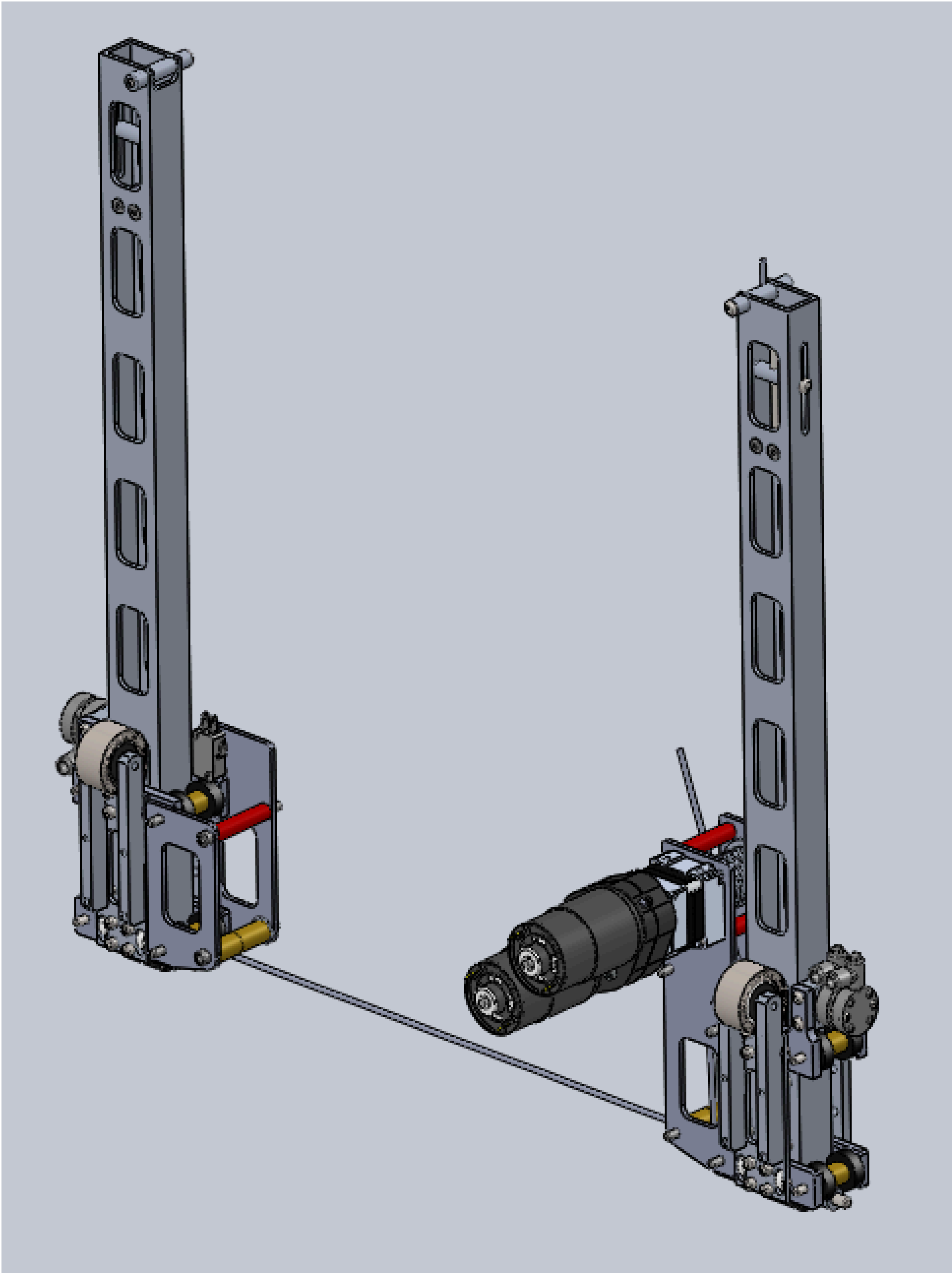
3.4 The Feet

3.4.1 Initial Design and Prototyping

Since day one, our team has seen endgame as a daunting obstacle and until recently we have put it on the back burner. In the past few weeks, we have come up with a way of getting onto the level 3 habitat by using a pneumatic actuating foot to go onto level 2, turn, then go onto level 3.

3.4.2 Final Design

The feet are attached to the back of the drivebase, out of the way of the elevator when it pivots to a 90 degree. There are two long box tubes that are swiss cheesed to reduce the weight of the feet system. There is a gearbox that is attached to the boxtube so that when the gearbox turns, the box tubes go down and push up the back of the robot. We then go to the habitat and use our end effector to push down, and with the help of the feet in the back pushing up, the robot goes up to tier three. There are force springs to help with the force of pushing down and pulling the box tubes back up.



3.4.3 Control System

This system takes advantage of two single pneumatic actuators, which allows the robot to elevate at an angle off of the ground. This then allows for the driver to use the drivebase and elevator control system to move and get onto the next level of the HAB.

4. Software and Controls

4.1 FRC Architecture and Object Oriented Programming

Like many other teams, team 5401 utilizes Java as the programming language for the robot. Java is most known for its object oriented nature, which allows creations of software objects which mimic physical objects in real life.

Team 5401 programs their robot through the FRC Architecture. Code is split into Subsystems and Commands. Subsystems mimic the corresponding physical parts on the robot. The Commands execute actions that the physical robot can perform. The “main” block of code runs in a loop and is able to call the Commands through input from a controller. Autonomous code are made in a CommandGroup which combine multiple Commands and require no human input.

4.2 Collaborative Work Environment

The software team for Team 5401 have a weekly meeting to discuss goals for that week. In addition, meetings are organized to discuss how the general logic implementation of each individual Subsystem. Outside of meetings, programmers may work individually or in a team depending on the code or robot part that is being tested.

The following apps are used to communicate and organize the software team:

- Slack: This application is mainly used for communication, such as writing down goals and reminders. Slack also notifies us when changes are made to the code.
- GitHub: This application is used to store the team’s code online as a code repository. As a result, any computer can pull the code from GitHub and edit it. In addition, GitHub offers the ability to “branch” code which creates a copy of the current working code. The existence of the copy allows experimental code to be created and tested without overwriting the current code that works. If the experimental code fails, the branch can be abandoned without modifying the working code. It the experimental code works, the code can then be merged into working code.
- Google Drive is utilized to share other files that are not code. Code documentation and block diagrams are created and kept in Google Slides. Mechanical and software teams share a Google sheet detailing all the PWM, sensor, and pneumatic solenoid channels for the robot. Controller mappings are documented and stored on Google Drive as well.

4.3 Implementing the Programming Process

Identifying the Problem

- In the beginning of the build season, the software team has a meeting with the design team. The design team gives the software team a brief explanation of each subsystem, or group of mechanical parts that work together on the robot and its corresponding movements.

Designing/Drafting the Solution

- The software team begins to draw Block Diagrams of each subsystem. The Block Diagrams are very similar to a typical IPO chart: the Block Diagram displays the possible control inputs, the necessary processing, and the actions of the robot as a result.
- The software team then converts the Block Diagrams into pseudocode. Pseudocode is not "real" code and consists of comments strung together in order to provide an outline for the real robot code. Pseudocode also helps work out logic and structure for the future code.

Writing the Program

- Branches for each Subsystem are made off the main code/branch. Each programmer is assigned a Subsystem to write by converting pseudocode to actual code.
- During this step, programmers will often talk with the members of the design team. The goal is to discuss the specific components and movements of each Subsystem.

Testing and Revising

- After the practice robot is built, the software team tests each Subsystem on the practice, one Subsystem at a time. If the Subsystem works, the branch for that Subsystem is then merged into the main code/branch. Discrepancies between the practice robot and competition robot are noted and updates for the competition robot are made.

4.4 Teaching New Programmers

Every year, several rookie programmers join the team with little to no experience in writing code for the robot. The senior members of the software team are responsible for bringing the rookie programmers up to speed. They begin with teaching Java basics such as variables, loops, logic statements, arrays and objects. After the Java basics are mastered, the rookie programmers are taught the FRC Architecture. They can then begin to experiment with the FRC Architecture on the previous years robot.

4.5 Human Machine Interface

The team utilizes two Xbox controllers to run our robot this year. This is due to many being familiar with its overall layout and design. This also allows for customization of the buttons and joysticks.

4.6 Scouting App

- The “FROScoutingApp” was developed using Android Studio. The app is optimized to run on an Amazon Fire 7 Tablet running Android 5.1 (API 22).
- This application can be used to scout any years game, as it is entirely customizable, allowing for users to setup and create the layout and scoring guidelines they find to be the most crucial that year. A file containing the layout of inputs can be exported from the app and imported on other device.
- The data from scouting is outputted into a csv file which can be easily imported into sheets to allow for fast and easy data collection.

4.7 Robot Camera Streaming

- The “OwlVision” project is a continuation of the “FRCVision” project created by WPI, the creators of many software libraries for the robot. It is a ready to use image for the Raspberry Pi, a low cost Single Board Computer, allowing teams to stream video from a camera on their robot to the Driver Station.
- We modified the image to incorporate GSteramer, an open source “pipeline-based multimedia framework that links together a wide variety of media processing systems to complete complex workflows”.
- GStreamer allows us to utilize the native H.264 encoding of the Logitech C920 camera, which utilizes much less bandwidth than traditional video compression methods.
- After implementing these changes in our own robot, we were able to achieve a 1080p, 30fps, 300ms latency video stream from our robot to the Driver Station, while still remaining under the 4mbps bandwidth limit imposed by FIRST. This is a huge improvement to the 480p, 15fps, 500ms latency video stream provided out of the box by the FRCVision image, which is the highest we could achieve while adhering to the strict bandwidth limit.
- The OwlVision project is fully open source, allowing for other teams to view and modify the image as they see fit. We also provide a ready to use image on our team website.

5. Manufacturing

The STEM program at Bensalem High School received a new technical education facility as part of the renovations that took place in 2017. Thanks to our new space, all manufacturing of Mr. Krinkle was done in-house in the Technical Education Shop of Bensalem High School, adjacent to our robotics lab, by our student-led manufacturing team consisting of four students. The frame, elevator cross tubing, carriage, and end effector are constructed of box tube aluminum, while the elevator rails, custom gussets, blocks, spacers, and shafts are made from a wide range of different sizes of aluminum, polycarb, and Delrin. All shafts and other turned parts that were not purchased complete were turned on one of the two shop lathes in the Technical Education area. All drawings given to manufacturing for production can be seen in the Drawings Section at the appendix of this technical summary. Due to our intricate design, new techniques were learned in order to drill small holes in odd shapes and round corners for clearance on the robot. To maximize production time on the 234 parts on Mr. Krinkle, two manual mills and two lathes were run at the same time. Mr. Krinkle was assembled in the robotics lab, in the team's fully functional pit, which is discussed further below.

In order to support the ambitious strategy the team decided on, and to make use of the internal machining capability, a prototype practice robot was built to learn lessons, provide a tool for drive team to practice on, and a test bed for software while the competition robot was built.

6. Practice Field

Using the Technical Education Shop of Bensalem High School, wooden field elements were built to assist prototyping of robot assemblies, as well as give the drive team a practice field on which to hone their skills. The assembly of the practice field this year was an incredible opportunity to get as many of our rookie students involved as possible. Since we have such a large team this year, we struggled to find things for them to do, part of their build season experience was assembling the field. This helped introduce them to basic engineering concepts to prepare them going into assembly, CAD, or manufacturing. They were the ones that found, printed, and read the measurements on the FIRST released field elements, an experience that helped them become familiar with the field on a much deeper level.

7. Conclusion

With our team of 48 members, a broader student leadership team, Team 5401 has created a robot that brings the team a sense of pride and achievement. From day one of kickoff weekend, our goal was simple: build a machine that achieves our goals, is expertly engineered to exceed our quality standards, and is a robot we are proud to show off at competitions. Although it was tough build season with many setbacks and challenges, we have achieved all of our goals and we look forward to powering up at competitions. We hope our performance this year excites you as much as it does us.