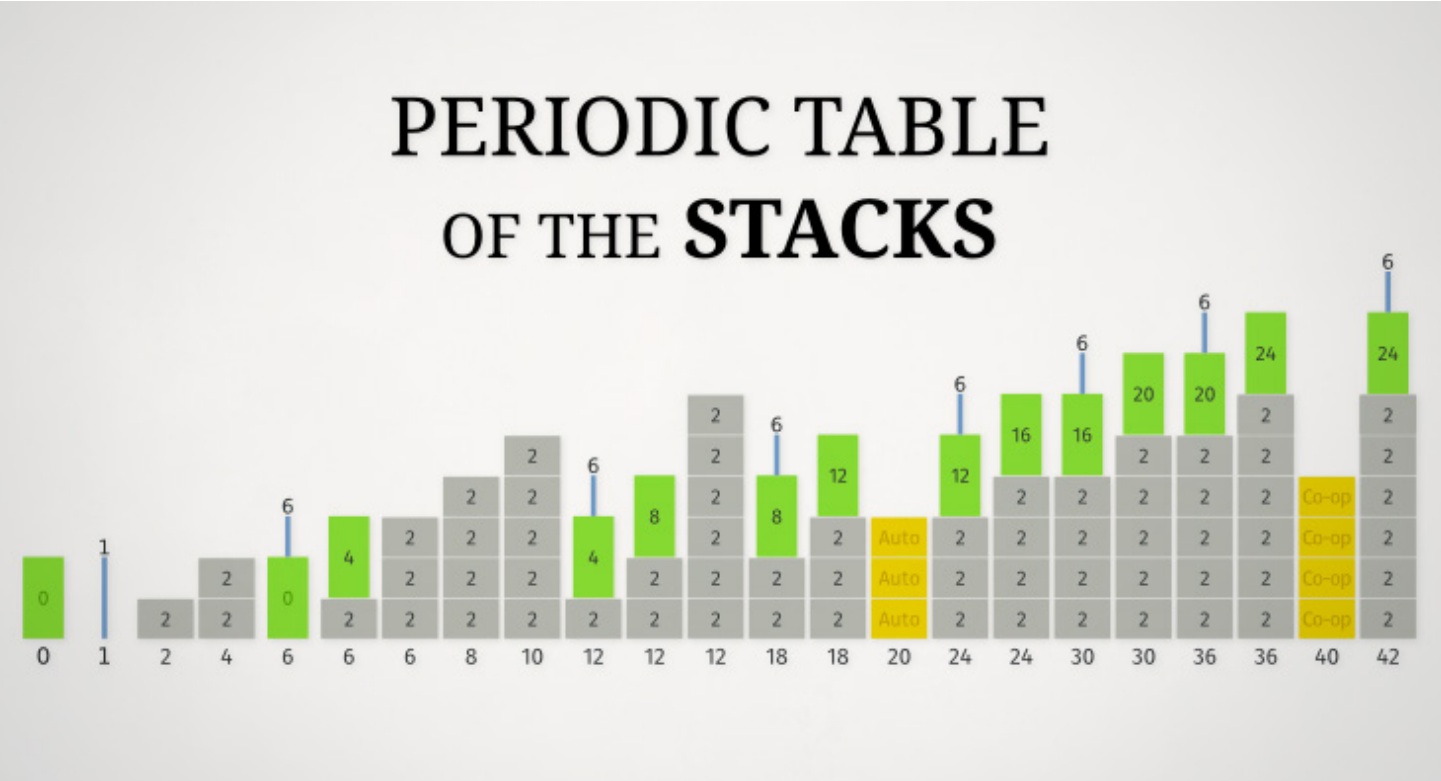
254

2015 Technical Binder

Table Of Contents

Kickoff & Game Analysis



Chief Delphi user ThomasClark FRC #0237 04 Jan 15

Strategies and Goals

- Due to the lack of defense, we determined that an offensive robot that could perform all of the tasks would be the most versatile and have the highest chance of seeding first or of being picked first.
- We debated between scoring many short stacks or a few 6 stacks, but decided on scoring tall stacks in order to maximize the number of points per recycling container.
- For autonomous, stacking 3 totes in the auto zone is worth more points than anything else, so it became a top priority to score those points, regardless of our alliance partners.
- To maximize efficiency, we planned to score the co-op stack right at the beginning of teleop .

Autonomous Scoring

- Although the best a single robot can score in autonomous is 28 points, we placed it as a lower priority than a consistent 20 point autonomous due to the extreme difficulty and small point reward.
- To increase our versatility and ability to partner with any other robot we also wanted to be able to grab the recycling containers off the step.
- At the end of autonomous it would be ideal to have possession of at least one recycling container so that we could immediately start our first stack.

Teleop Scoring

- The team debated between two routes of teleop play. We realized we had to choose between focusing on the landfill or the human player station because it was not feasible to drive back and forth frequently. Initial estimates on how long it would take to acquire totes from either area led us to believe that human player stacking would be the best route.
- Thus, at our first regional we focused only on scoring from the human player station and were able to score up to 3 stacks or 2 stacks and a co-op in a match. However, after seeing how easily teams were able to grab totes from the landfill, we realized that we would need to improve.
- From there, our team completely redesigned strategy and for our second regional we switched to landfilling with redesigned intakes and using a tethered ramp to allow us to score from both areas. With this strategy we were able to score up to 4 stacks and a co-op.

Requirements and Priorities

1. Stacks of 6 with a recycling container on top
    a. Stack and carry 6 totes
    b. Pick up horizontal recycling containers
    c. Place recycling container on existing 6 stack
    d. Release stack onto scoring platform quickly

2. Traverse Field
    a. Drivebase with high acceleration, but not necessarily high top speed
    b. Drive over scoring platforms
    c. Make sharp turns quickly
    d. Control and maneuverability was especially important on a shorter field
        1. We could not risk knocking over created stacks and losing points

3. Pick up from Landfill Zone
    a. Have very effective intakes that require little driving to acquire totes
    b. Align right-side up totes from virtually any orientation

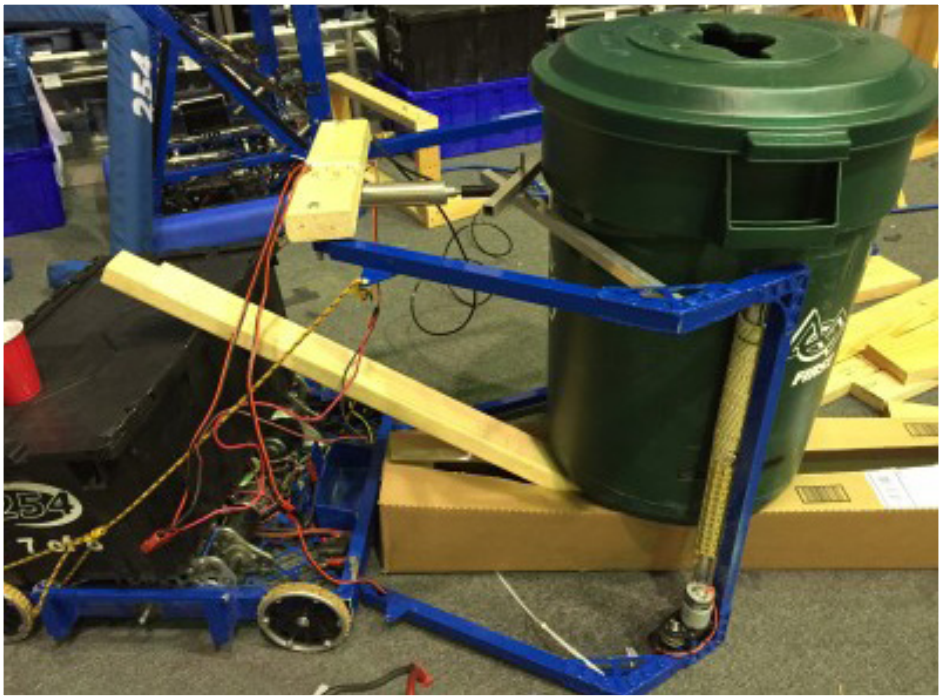
4. 3 Tote and 3 Recycling Container Auto
    a. Stack all 3 yellow totes while leaving recycling containers upright
    b. Possess at least one recycling container to begin first cycle

5. Human Load
    a. Minimize time of transition for Human Loading Chute to robot
    b. Align totes as they enter the robot
    c. Make sure totes land upright

6. Co-op Stack
    a. Place stack of 3 yellow totes on step as soon as possible
        1. Avoiding driving over thrown noodles later in matches

7. Other Considerations
    a. In order to be one of the most competitive teams, litter in the recycling container was a must.
    b. We decided that the upside-down totes placed in front of the step were too hard to manipulate to be worth our time, provided that there were 18 totes in the landfill, and 30 more in the human player station

Brainstorming Prototyping & Design Selection



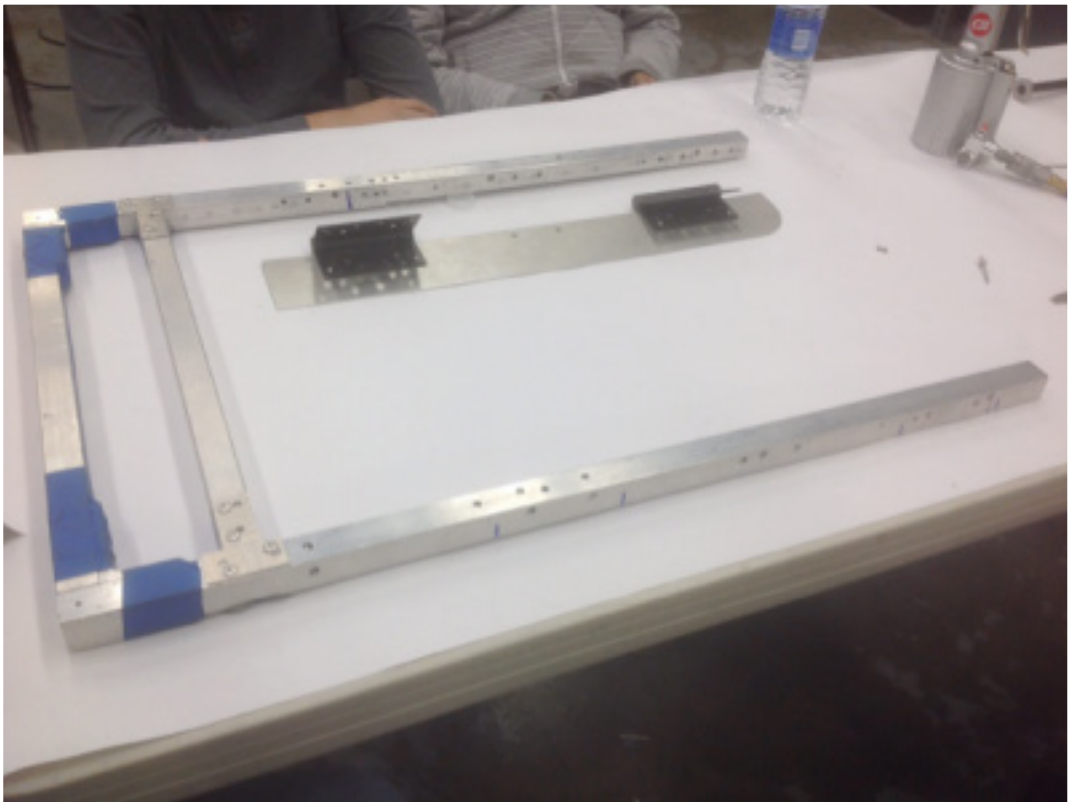
Intake

We prototyped an intake system using two of Barrage's (last year's robot) intakes. We decided that grabbing the totes horizontally (roller) would help us intake them more effectively than using a vertical intake (conveyor). We did not want to apply downward force on the tote, as that would increase the friction between the tote and the ground. We used polyurethane rollers to grip the totes. Despite being heavy, the polyurethane was the perfect choice, as it has an extremely high coefficient of friction, in addition to minimal compression.

Our second intake prototype used 2 pairs of rollers on pivoting arms to improve alignment of the totes from the landfill. We decided between Banebots' wheels or a short section of a polyurethane tube on a custom hub.

Claw

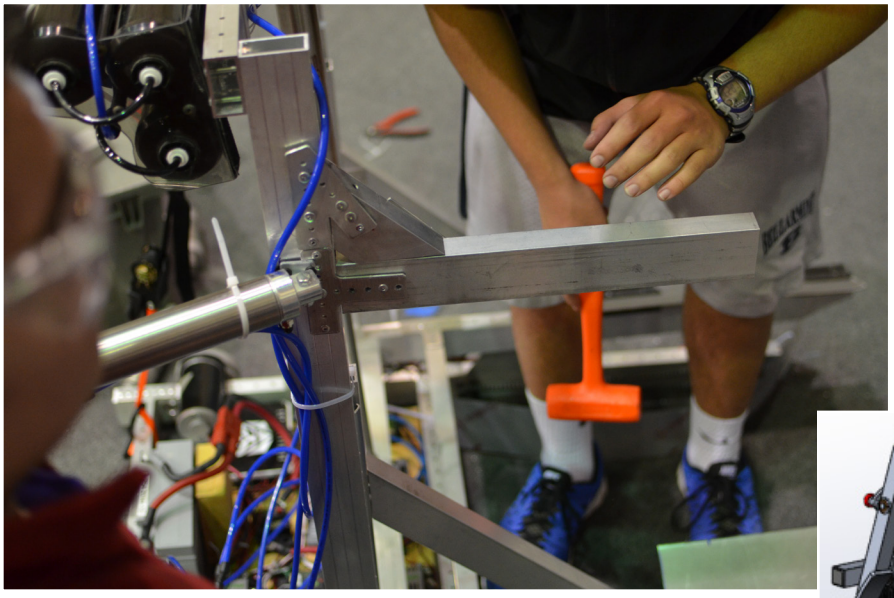
We iterated the recycling container design multiple times. The driving factor was that we aimed at having the ability to put a recycling container on an existing six stack. In order to place a recycling container on top of an existing 6 stack we would need to grip the recycling container upright at the very bottom. Another factor that was considered is that recycling containers will often end up tipped over, so we would need to be able to manipulate them when they were on their sides. The autonomous bin grabbing considerations (starting with a recycling container in the claw) also drove some of the design.

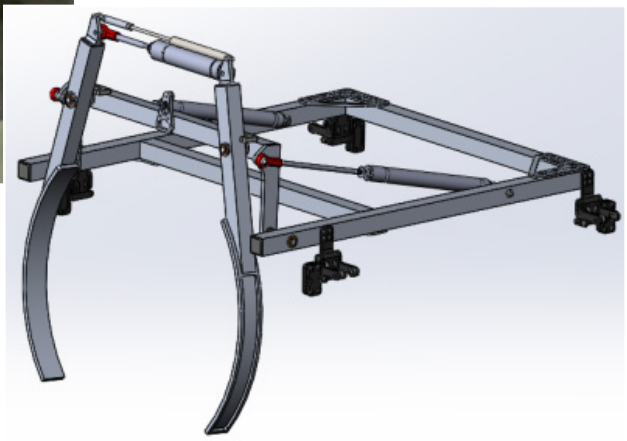


Lower Carriage

We experimented with the angle and actuation of the lower carriage. We wanted a design that would allow us to bottom stack instead of placing totes on top of one another. We ended up going with a spring return pneumatic cylinder connected to metal flaps so that we could open them manually if needed.

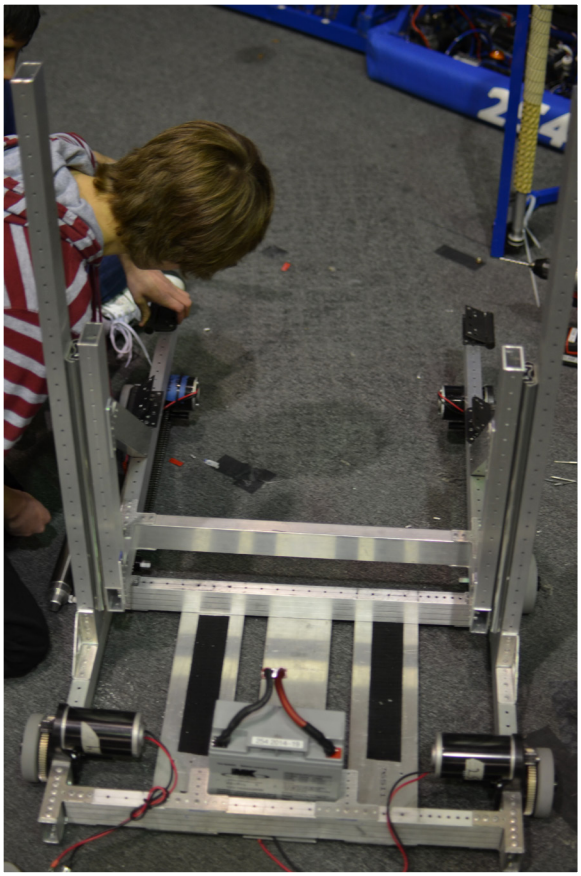
Initial tests featured only a lower carriage, and thus we were able to use one large pneumatic cylinder instead of belts to lift the carriage. However, after the need for an upper carriage developed, we were forced to use timing belts (a lighter alternative to #25 chain) for the elevator.

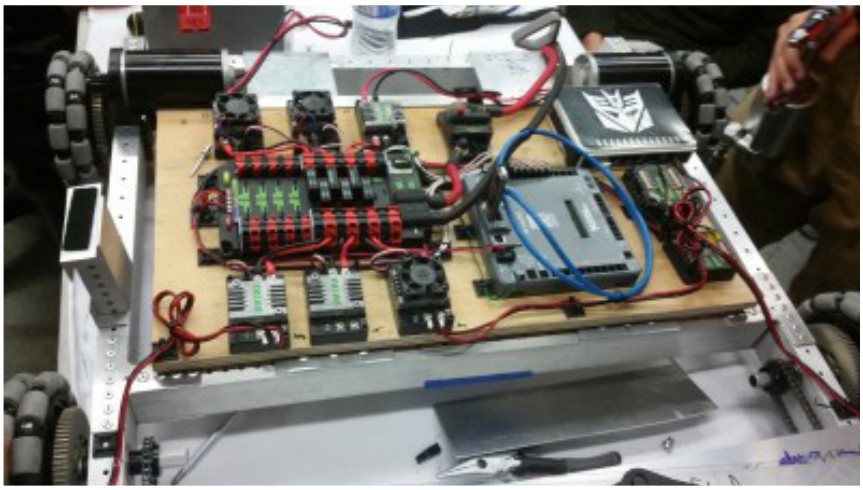




Upper Carriage

The upper carriage allowed for the claw's vertical range. We designed it similarly to the bottom carriage for manufacturing simplicity, with minimal changes made to accommodate the mounting of the claw. We wanted the upper carriage to function the same way so it was driven by belts, and had a large range of motion. Later, we incorporated software to securely hold the totes inside the robot by driving the top carriage into the bottome carriage.





Drivebase

We prototyped a slightly new style of drive base this year. Since the field is very different from previous years, we realized that we would need a higher acceleration and a lower top speed. The main driving component of the design was maneuverability. We realized that with a six stack and a recycling container, our center of gravity would be thrown way off. We aimed at a design that could overcome this and to drive smoothly with the added weight.

We tested multiple drivebase dimensions and settled on a long drivetrain that would allow most of the tote to be held behind the front wheels, greatly reducing the chance of tipping.

Skystalker

We combined many of our prototypes into a basic robot called Skystalker 0.2, named after last year's Skystalker 0.1. This helped us test not only our individual designs, but how we could integrate them with each other, and how they would function as a whole. We spent a lot of time playing around with possibilities of how to improve Skystalker, so we could incorporate those improvements in our final design. This also helped us test this years brand new control system on working hardware, giving both our drivers and our programmers a robot to practice with.

Human Loading

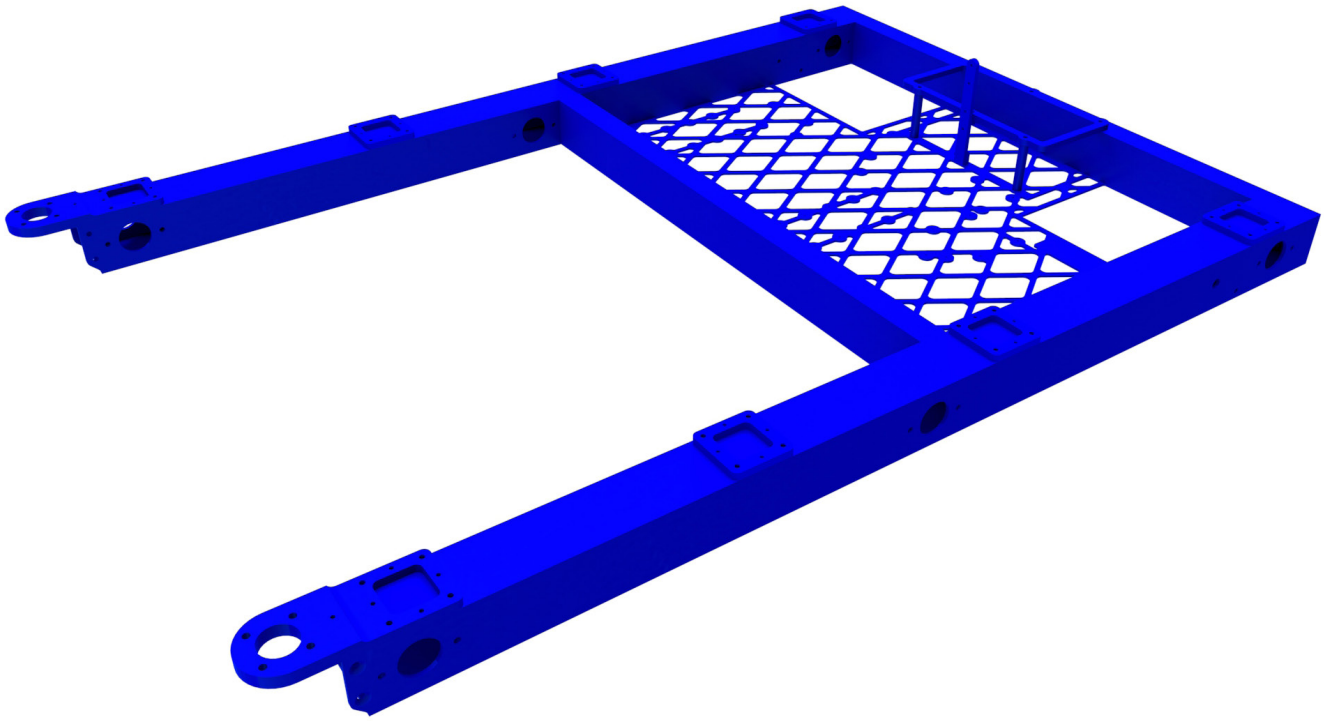
We briefly played around with the idea of adding a human loading ramp to help us quickly stack totes from the human player station. We eventually added this after our first regional and used it in place of the extra tall intakes pictured above.

The original intakes were designed to be extra tall so that the rollers could contact a tote as soon as it came out of the chute and suck it into the robot. This greatly sped up the process, but the extra height on the intakes meant they were too heavy to use after our first regional demanded weight for changes.

Due to the ramp's simplicity, it was quickly designed in CAD and needed little prototyping. See RAMPage for more information on the ramp.

Final Mechanical Design

Subsystems Overview



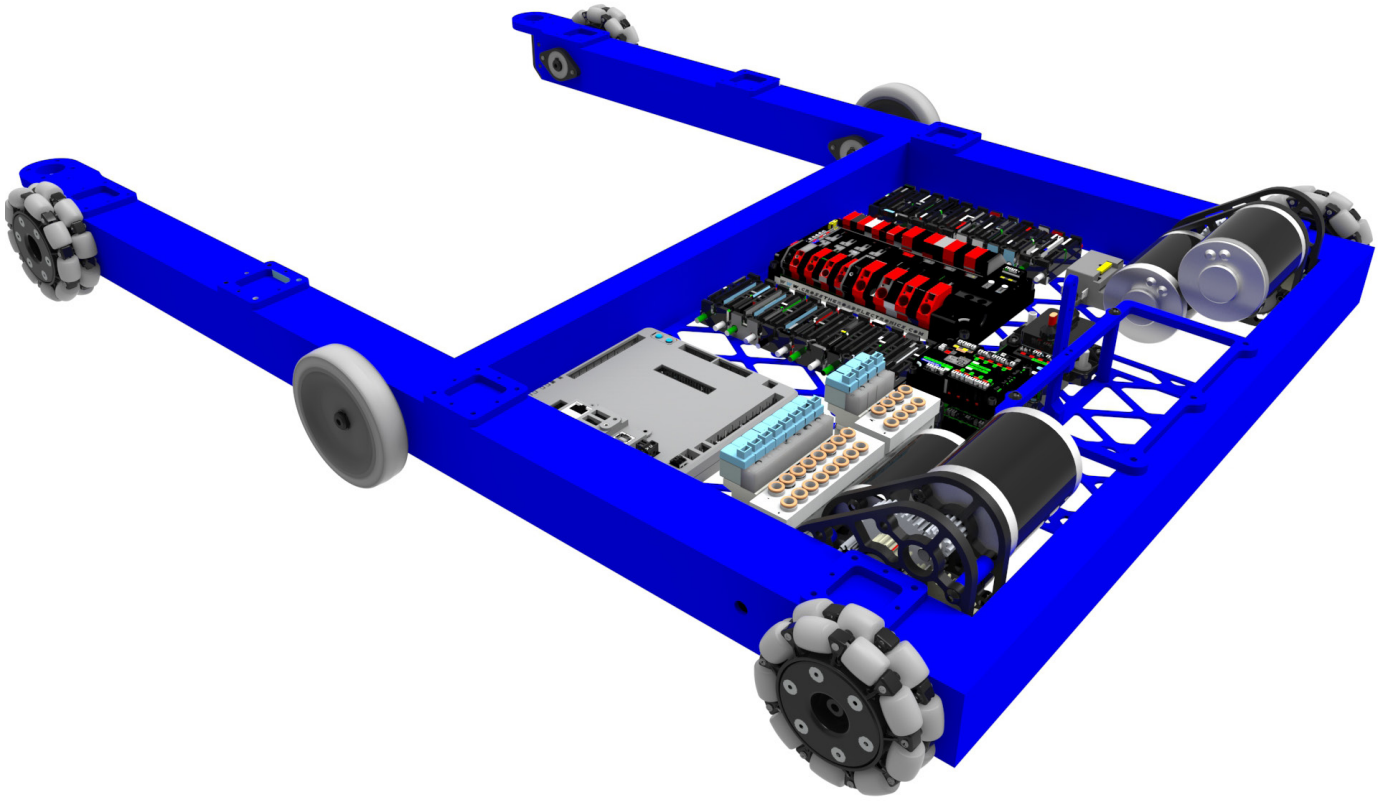
Drive Chassis

The shape of the chassis was designed to keep the center of gravity inside of the robot.

- 25.25" Wide
- 38.00" Long
- Siderails are 2"x2"x1/8" and cross bars are 1"x2"x1/16"
- 3/32" thick water-jetted bellypan features tapped holes to screw electronics in

Plates with a standard hole pattern were welded onto the top of the rails early on in the design process and later used to attach elevator uprights, A-frame, and intakes.

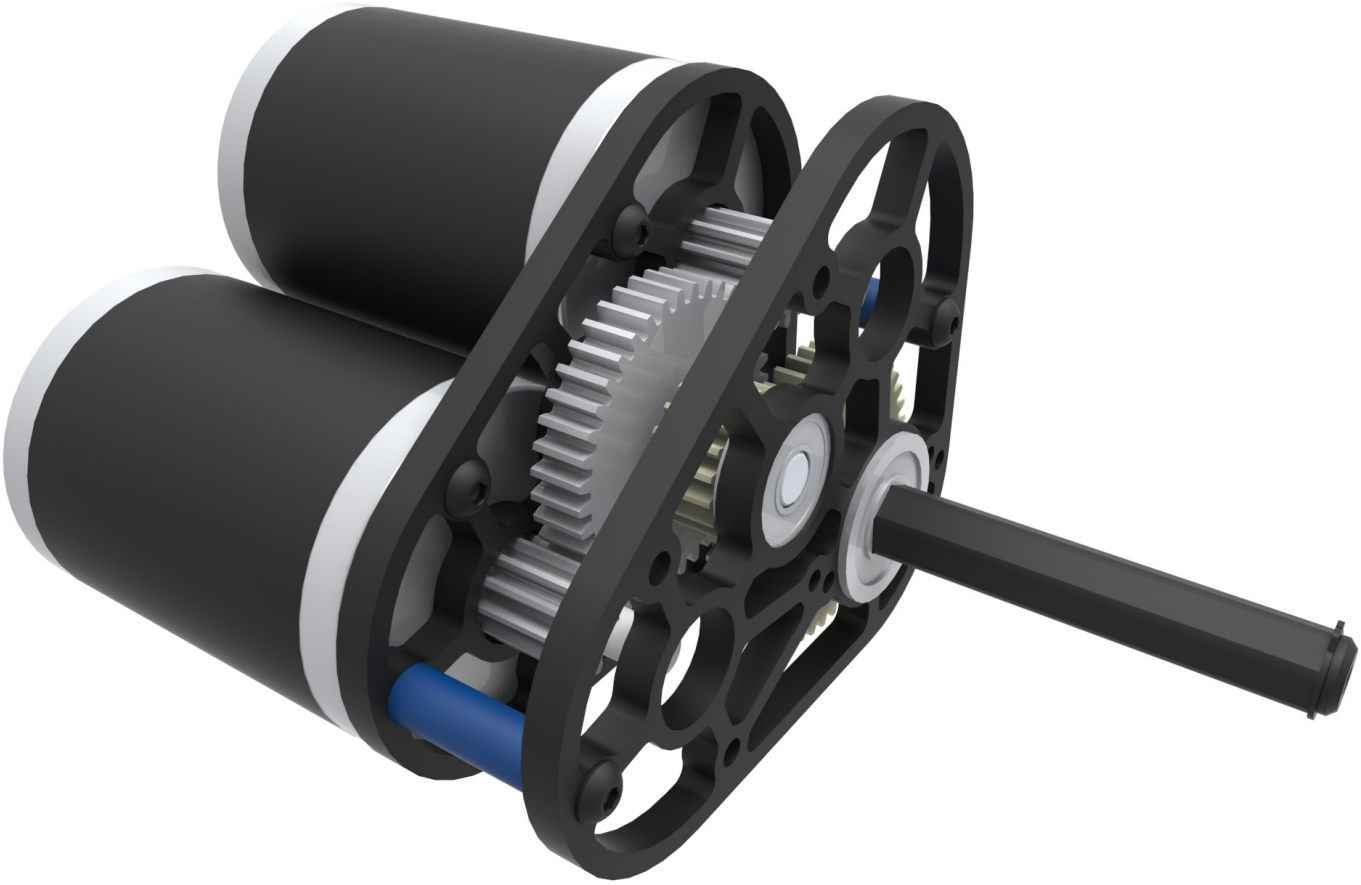
Battery box is welded onto bellypan and back rail.



Drivetrain

We used a similar drivebase to what we have been using for the past several years

- 6 Wheel, "West Coast Drive," with a center drop of .100"
- 4 Omni wheels in the corners ease turning and 2 Colson wheels in the center grip the scoring platform
- The gearbox is placed at the back of the robot to help offset the totes' affect on the center of gravity
- #25 Chain runs in the 2"x2"x1/8" siderails, keeping them protected and maintenance free. No tensioning cams were required



Drive Gearbox

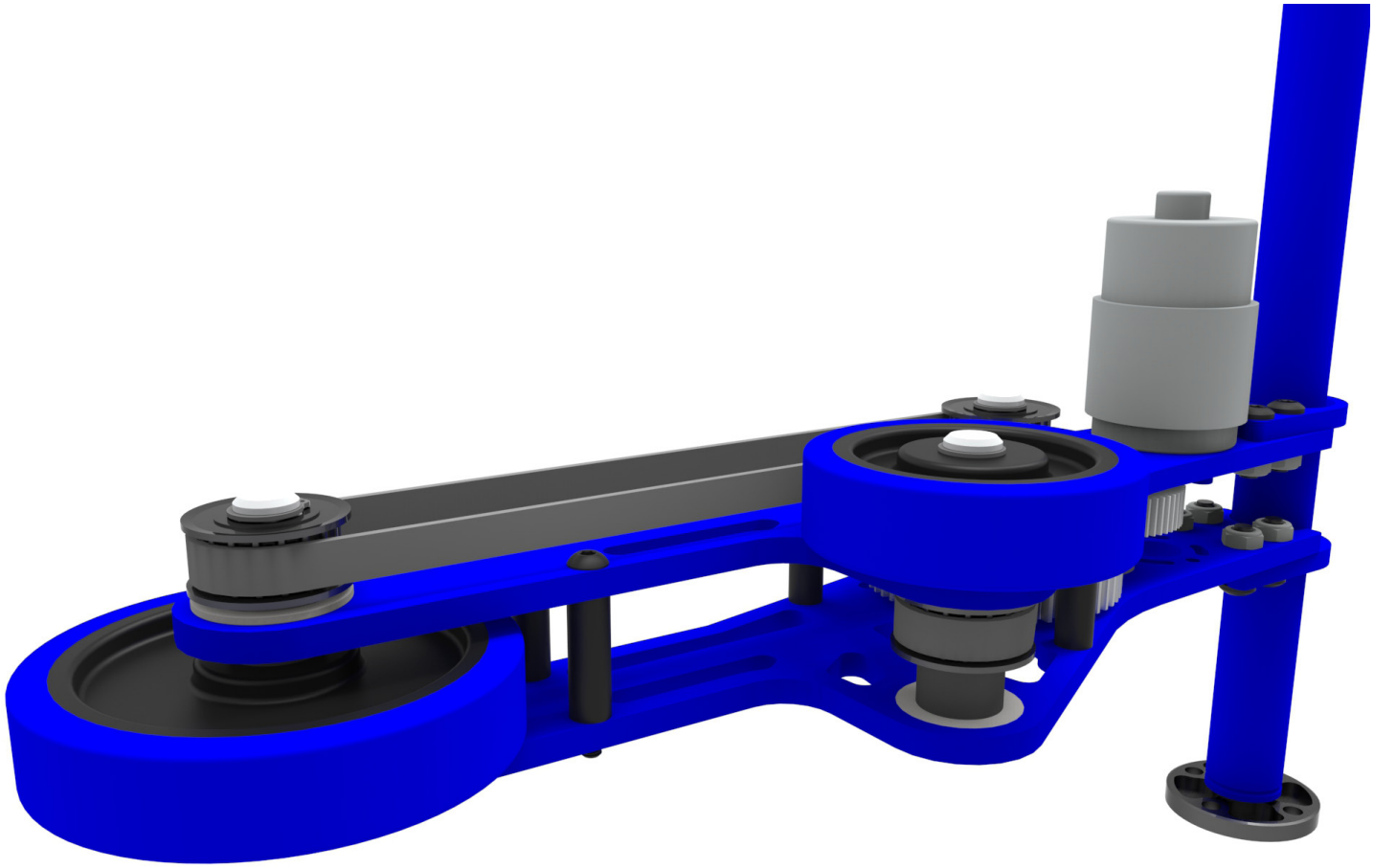
2 Mini-CIM motors per gearbox with one speed

- Reduction: 7.77:1
- Theoretical Top Speed: 12.00 feet per second

Designed such that multiple ratios could be achieved with one gearbox

- Can fit gears needed to create a 6.66:1 reduction or 5.00:1 reduction.

All gears, spacers, and standoffs are COTS. Utilizes new Thunderhex output shaft and Thunderhex bearings.



Intakes

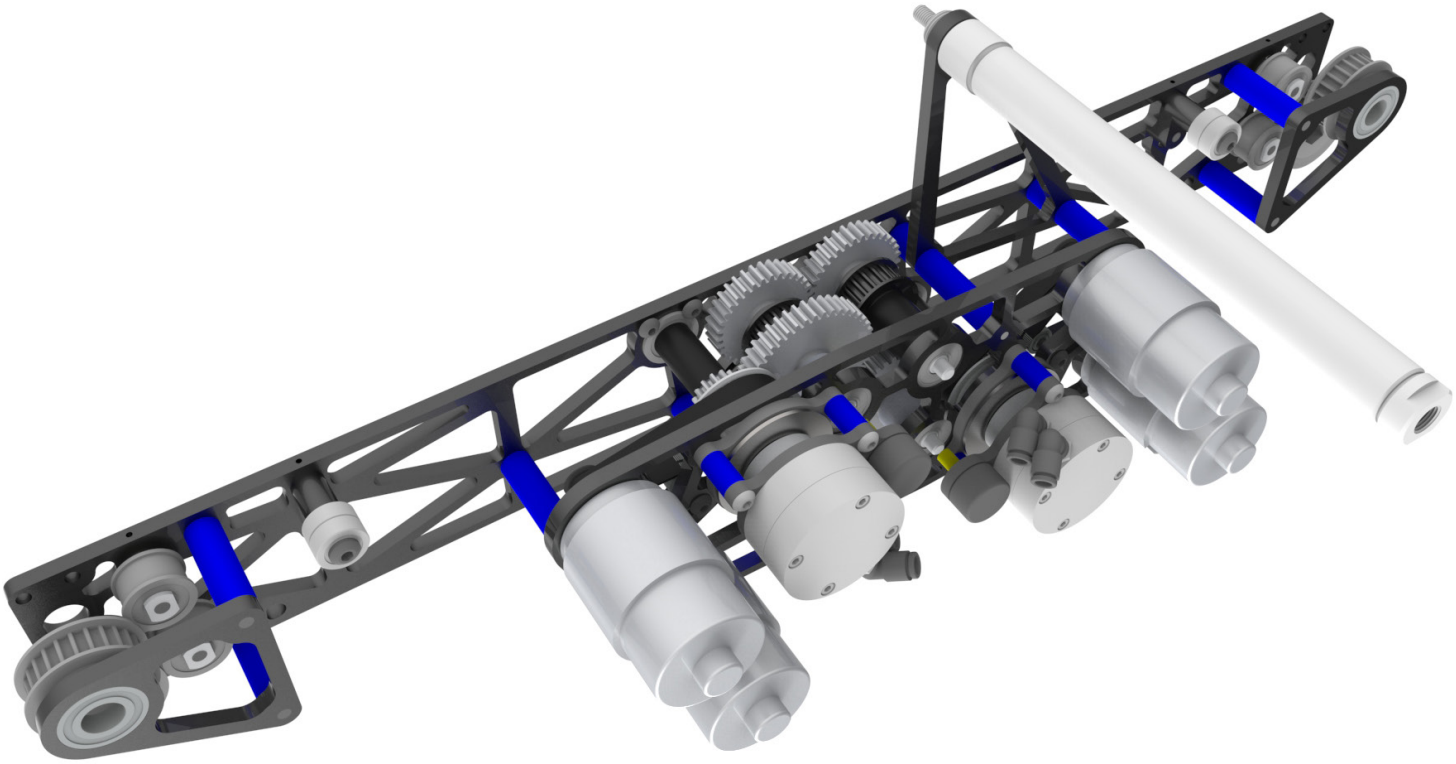
Brings totes and recycling containers into robot to be picked up

- Banebot wheels have optimal friction on the recycling containers and totes
- Combination of 4 wheels aligns totes and greatly improves landfill capabilities
- Can intake totes from virtually any angle

Pneumatic cylinders mounted to A-Frame to actuate intakes open and closed

- Small bore allows for compliance in the arm; the arms stay actuated closed while intaking
- Open intakes allow for quick release of stack

1 Banebot RS-775 motor per intake on a 19:1 2-stage custom gear-box reduction prevents jamming/stalling, but still intakes the totes and recycling containers quickly.



Elevator Gearbox

4 Banebots RS-775 motors

- 2 motors drive the lower carriage and 2 drive the upper carriage
- Motor-driven pinions connect to a gear reduction which drive the open-ended, serpentine belts attached to the carriages
- Reduction: 13.97:1
- Theoretical Top Speed of carriages: 5.85 feet per second

2 friction brakes actuated by 2 pancake pneumatic cylinders allow for independent braking of each carriage

- Allows carriages to hold a full stack of six totes without the motors running continuously, drawing power, and potentially burning up

Bottom Carriage

Supports and stacks totes within robot

- ¼" horizontal flaps slip over tote when going down and catch when traveling up
- Single action pneumatic cylinders allow flaps to be manually actuated up, allowing carriage to travel up the stack or let go to score it.

Attached to 4 bearing blocks that are clamped to elevator belts to move carriage up and down.

Elevator brake is actuated every time carriage stops moving.

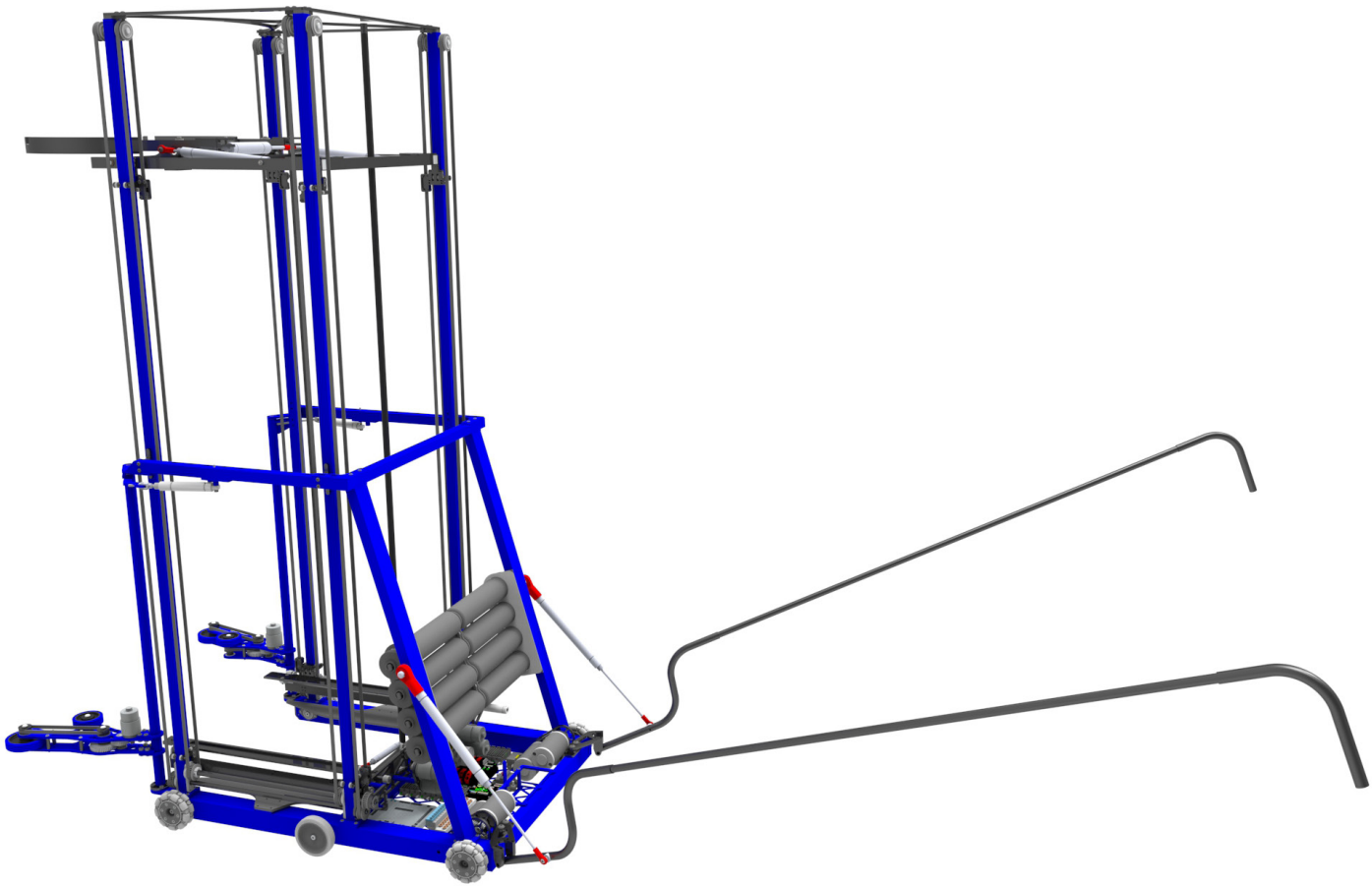
Top Carriage

Picks up and carries recycling container outside of robot

- The Polyurethane on claw increases friction on recycling container, helping secure it
- Perfect amount of compression to allow recycling container to slide up but not fall out of claw
- Diameter of the claw is optimized for bottom 3" of the recycling container, thus compressing the recycling container greatly when it grasps in the middle
- Claw opens up extra wide to allow robot to back off quickly

Claw can pick up both upright and tipped over recycling containers by pivoting down 90 degrees.

Carriage can be driven down into stack to provide a compressive force in order to securely hold it.



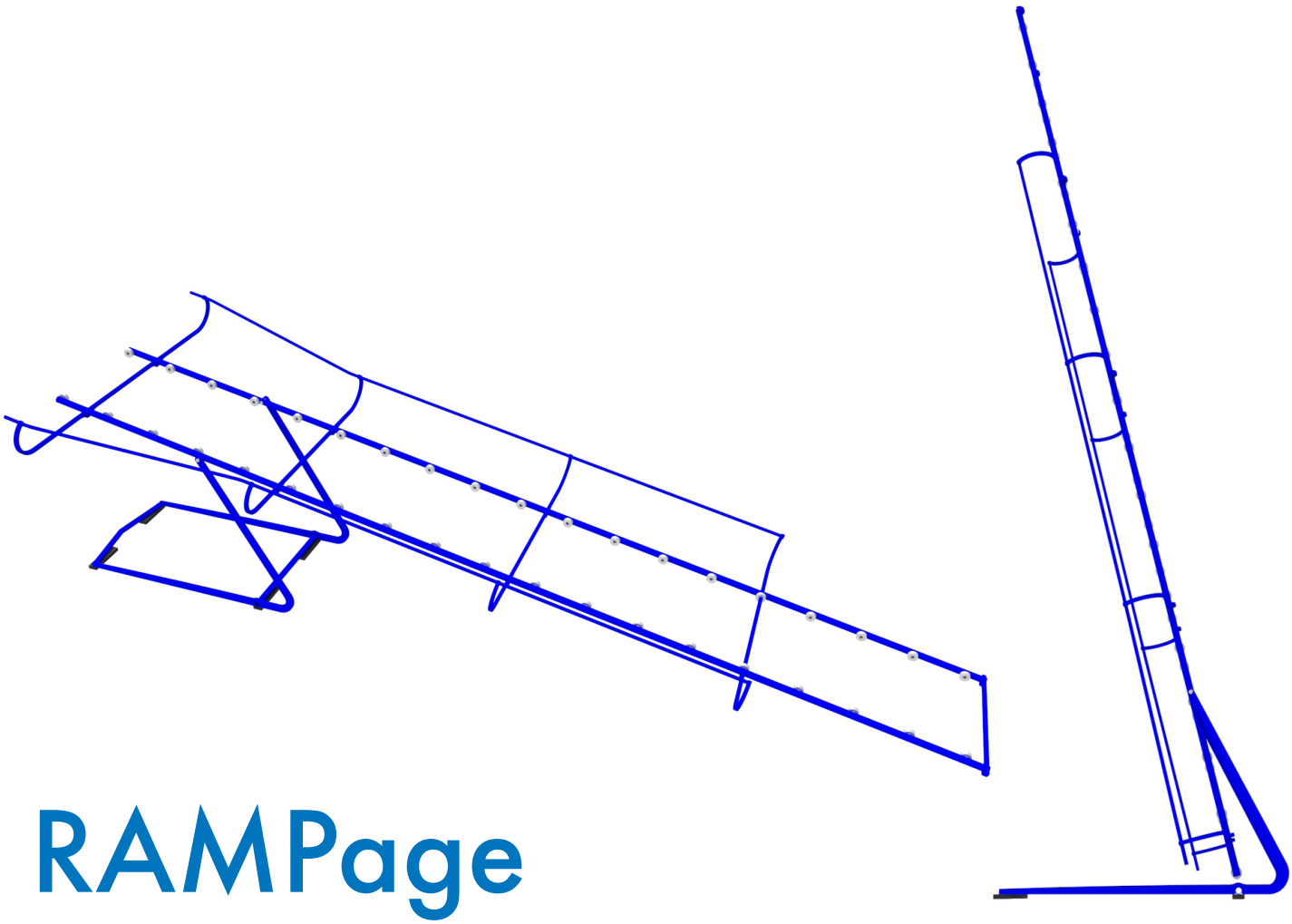
Recycling Container Grabber

The full assembly weighs less than 6lbs. The components include bent thin-wall aluminum tubing, pivoting carbon fiber rods, and pneumatic cylinders.

Initial bend near base allows arms to avoid landfill totes.

Uses drivetrain to pull recycling containers off step.

Arms fold up after auto to stay out of the way during teleop.



RAMPage

77.5" long. Can preload 4 totes, allowing robot to load 6 from ramp in about 8 seconds.

Weight: ~4lbs including tether

- Welded and powder-coated assembly is made of water-jetted plates and thin-wall tubes

Starts folded up against human player station to stay clear of autonomous routine.

Back plate of base is angled to offset the ramp from the other wall, making it easier for the robot to get to the totes.

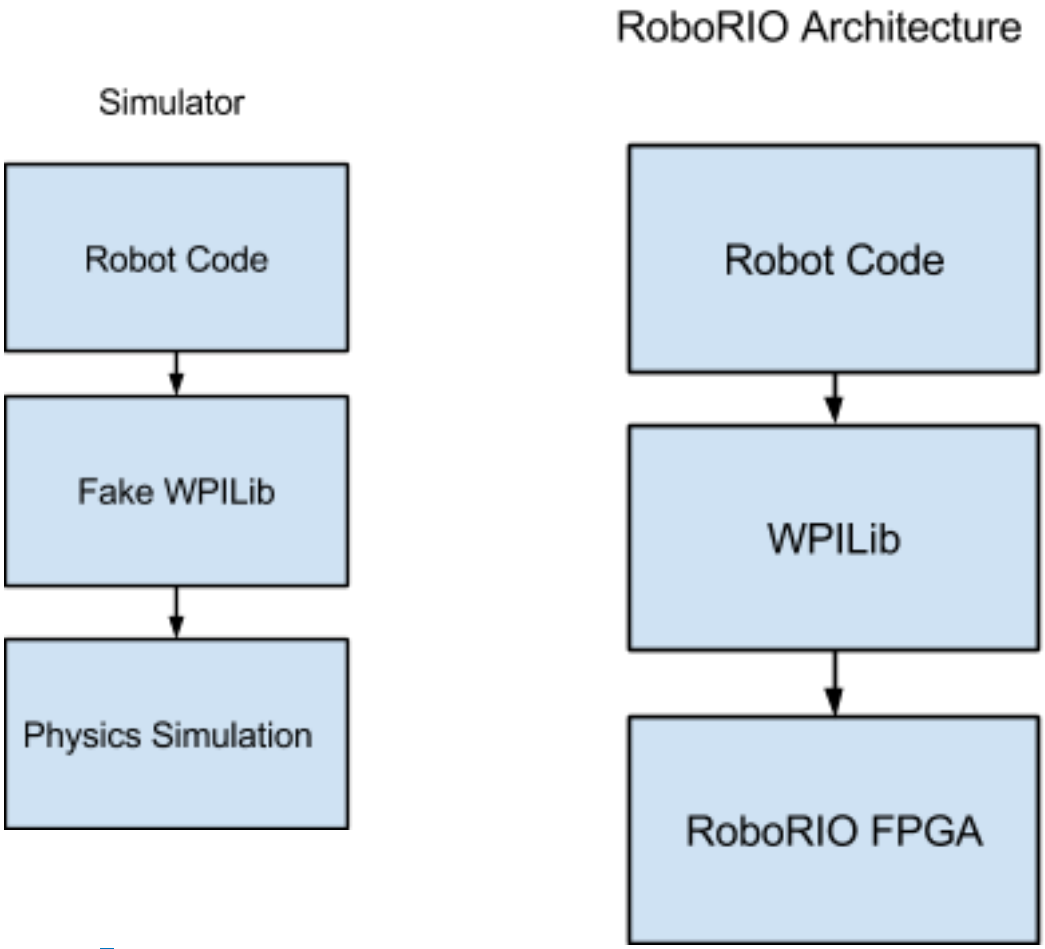
Software and Controls

```
private void enableInterrupts() {
    if (!m_interrupts_on) {
        m_home.requestInterrupts(isr);
        m_home.setUpSourceEdge(false, true); // Pulled high be default, need falling edge
        m_interrupts_on = true;
        m_home.enableInterrupts();
    } else {
        System.err.println("You done goofed. Don't request interrupts twice!");
    }
}

public void disableInterrupts() {
    m_home.disableInterrupts();
    m_interrupts_on = false;
}

public void findHome() {
    m_initialized = false;
    enableInterrupts();
}

protected Limits m_limits = new Limits();
boolean m_brake_on_target = false;

public class Limits {
    protected double m_min_position;
    protected double m_max_position;
    protected double m_rezero_position;
    protected double m_home_position;
}

public ElevatorCarriage(String name, CheesySpeedController motor,
                        Solenoid brake, Encoder encoder, DigitalInput home, boolean needs_init
    super(name);
    m_motor = motor;
    m_brake = brake;
    m_encoder = encoder;
    m_home = home;
    m_initialized = !needs_init;
    reloadConstants();
}

public Controller getCurrentController() {
    return m_controller;
```

```
protected void routine() throws AutoModeEndedException {
    autoModeTimer.reset();
    autoModeTimer.start();
    waitTime(.225); // Weird gyro init bug
    waitForGyroData(.25); // Weird gyro init bug

    bottom_carriage.setFlapperOpen(true);

    waitTime(.1);

    // Move can
    double start_height = top_carriage.getHeight();
    top_carriage.setFastPositionSetpoint(start_height + 16.0);
    waitForCarriageHeight(top_carriage, start_height + 15.0, true, 1.0);

    // Grab first tote
    intake.setSpeed(Constants.kAutoIntakeSpeed);
    drive.setDistanceSetpoint(24);
    waitForDriveDistance(10, true, 1.0);
    intake.close();
    waitForTote(1.0);

    // Turn on squeeze
    top_carriage.squeeze();
    waitForDrive(2);

    // Lift tote a bit off ground
    bottom_carriage.setFlapperOpen(false);
    waitTime(.1);
```

Software Overview

The 2015 Codebase implements many new algorithms previously un-used by our team. Most notably, we used both Feed-back PID and also Feed-Forward PID to predict and react to sen-sor values. We created a new web interface using WebSockets and HTTP, allowing us to debug it and control various aspects, such as autonomous mode, on and off the field. We also introduced a test harness, or simulator, that allows a team to write a hardware simulator in Java that interacts with Java robot code. This past season, the simulator boosted our productivity and allowed more collaborative work.

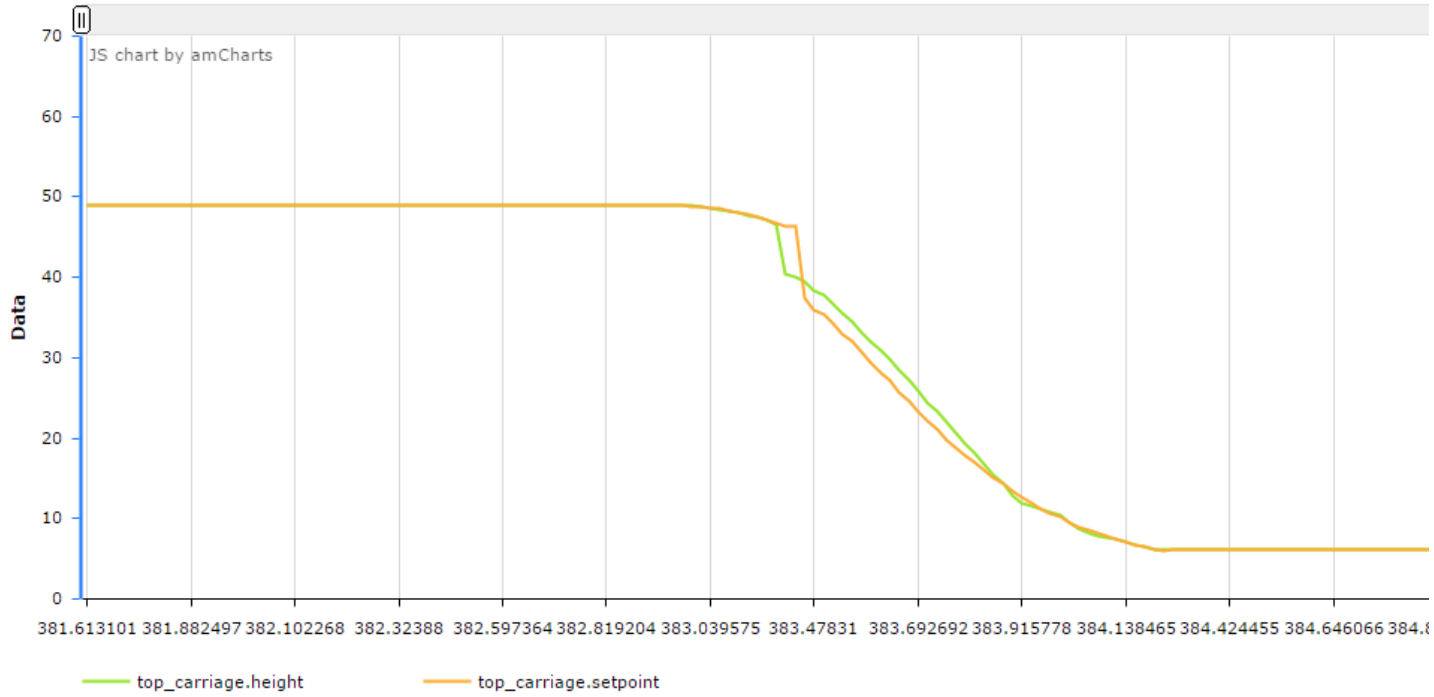


Simulation

As the design of the robot progressed, it became apparent that software would make or break its success. The programming team determined it needed to build a robot simulator to test control loops, create an operator interface logic, and build up tools that could be used to tune the real robot.

Requirements:

- Run on x86
- Run unmodified robot code
- Allow attachment of robot physics simulations to motor outputs/sensor inputs



Visualization Tools

In addition to other HTTP server features on the robot, the visualizer uses web sockets to provide real-time graphing of virtually any data in the robot code. This can be used to test the physics simulation and tune the real robot.

Features of the visualization tool include:

- JSON over WebSocket protocol
- Adjustable axes to scale data
- Subscription-based data stream. Only relevant data is sent to the browser from the robot
- Runs on FRC-allowed network ports
- Server is built on Java Jetty


254
254


254
254


254
254


254


Bellarmine College Preparatory