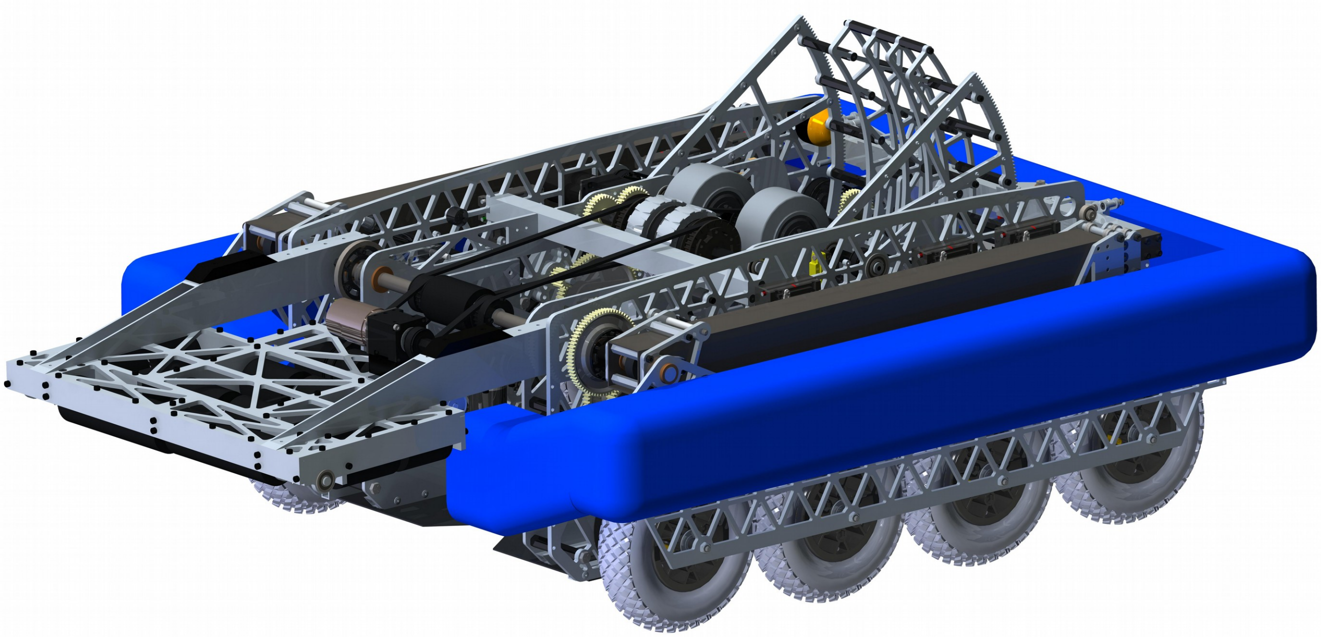
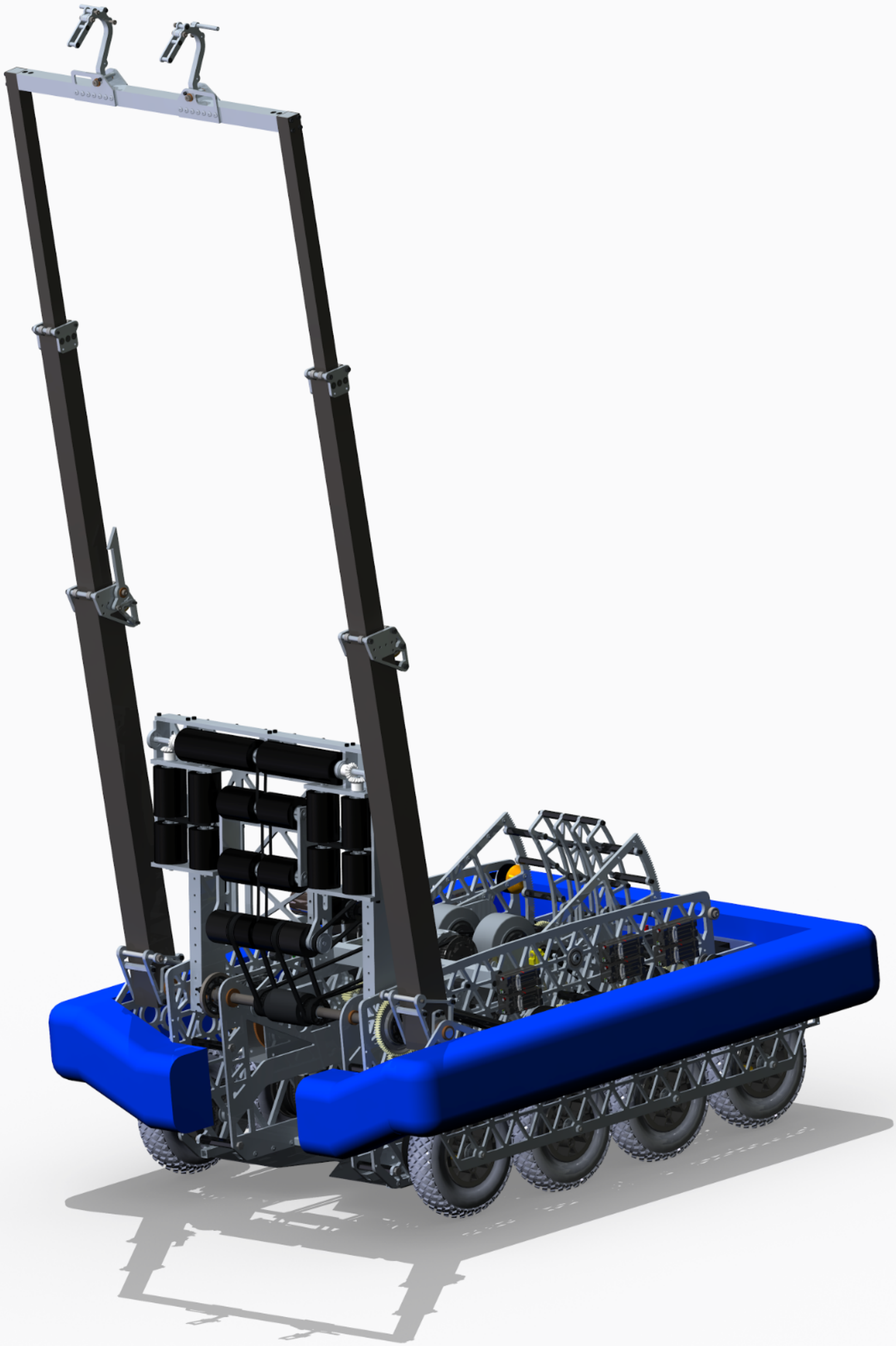
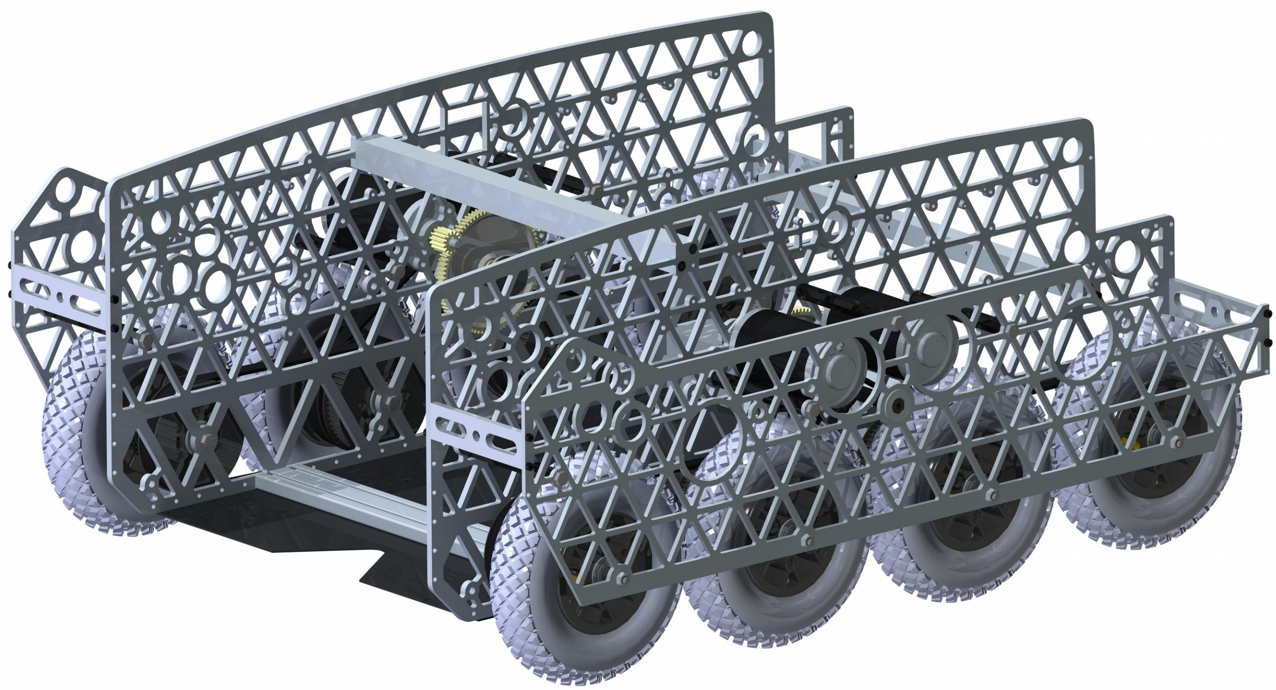
1768

2016 Technical Overview





Chassis and Drivetrain



From the beginning, it was clear that an effective frame and drivetrain for FIRST Stronghold would have to be extremely rugged in order endure the huge shock loads involved in the game. We used parallel plate construction to create an 8WD with 8" pneumatic wheels.

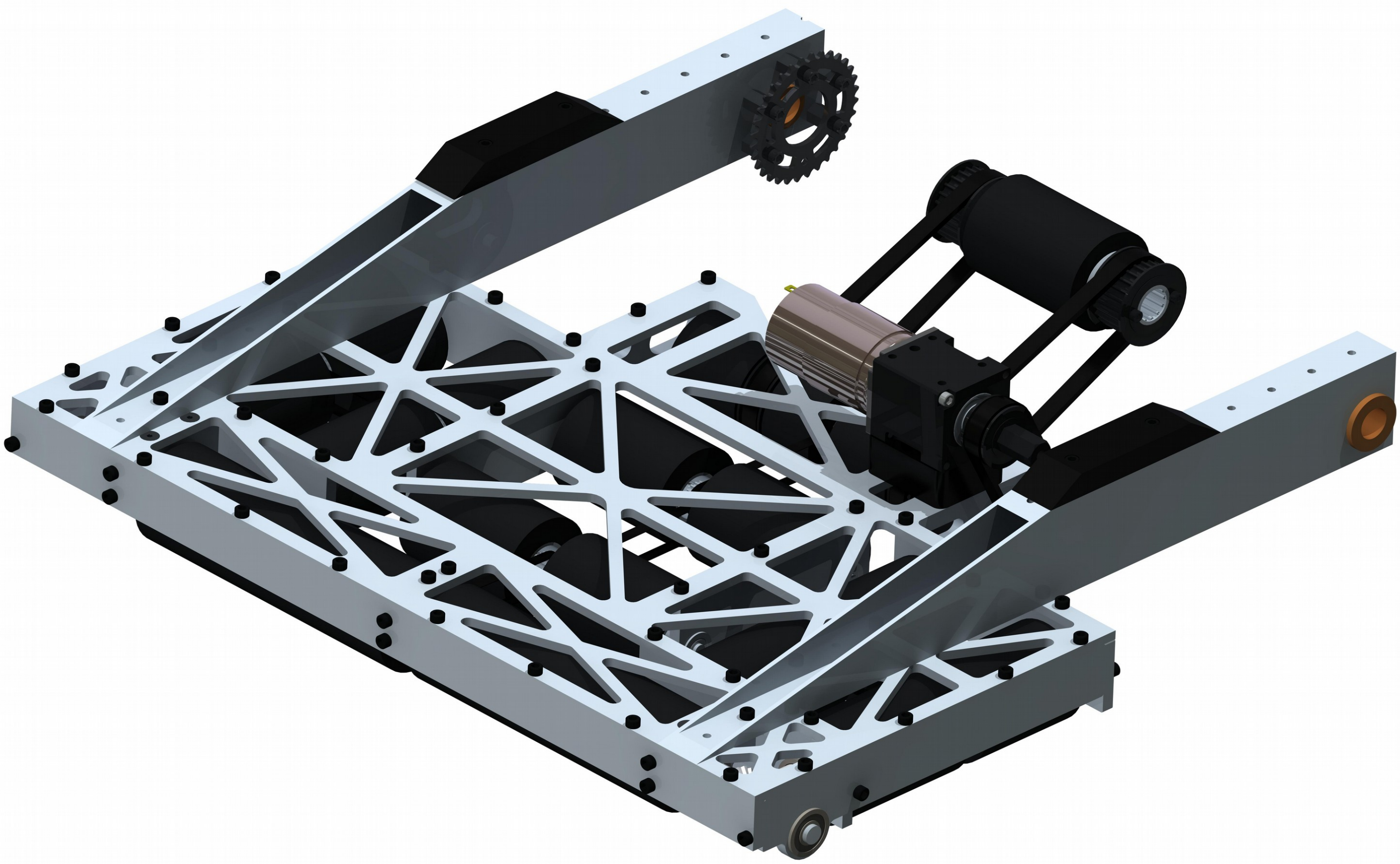
Parallel plate construction is convenient with waterjetting resources, allows drive axles to be double supported for robustness, reduces the total part count because so many components can be mounted directly to the plates, and makes spur gear and timing belt center-to-center distances throughout the robot easy because pairs of axles go through the same plates (no tolerance stackups). We used an isogrid lightening pattern for low weight, high strength, and ease of modeling in CAD.

We chose a tank drive with Andymark 8" pneumatic wheels for a simple, battle-tested solution for navigating the field and traversing terrain obstacles. Our 8WD left very little space between wheels, making it impossible to bottom out on the frame between them.

Specs:

- 8WD with 8” pneumatic wheels, ¼” Drop, 30 psi
- 4 CIM motors in a “flipped” configuration
- 15:1 gear ratio—transmission incorporated into frame plates
    - Final reduction done in a 60:24 timing belt stage—convenient for flipped motor configuration, plus we weren’t 100% confident that VexPro gears were strong enough to do the final reduction with an 8” wheel drive
- 15mm drive belts on 60T VexPro pulleys
- Double-supported dead axles to withstand the extreme forces involved in FIRST Stronghold and to act as structural standoffs
- Rubber-mounted Grayhill 63R256 encoders (AMT-103-V encoders added later for higher resolution for our auto-align velocity control loop)
- NavX MXP board, foam-mounted for vibration damping

Intake



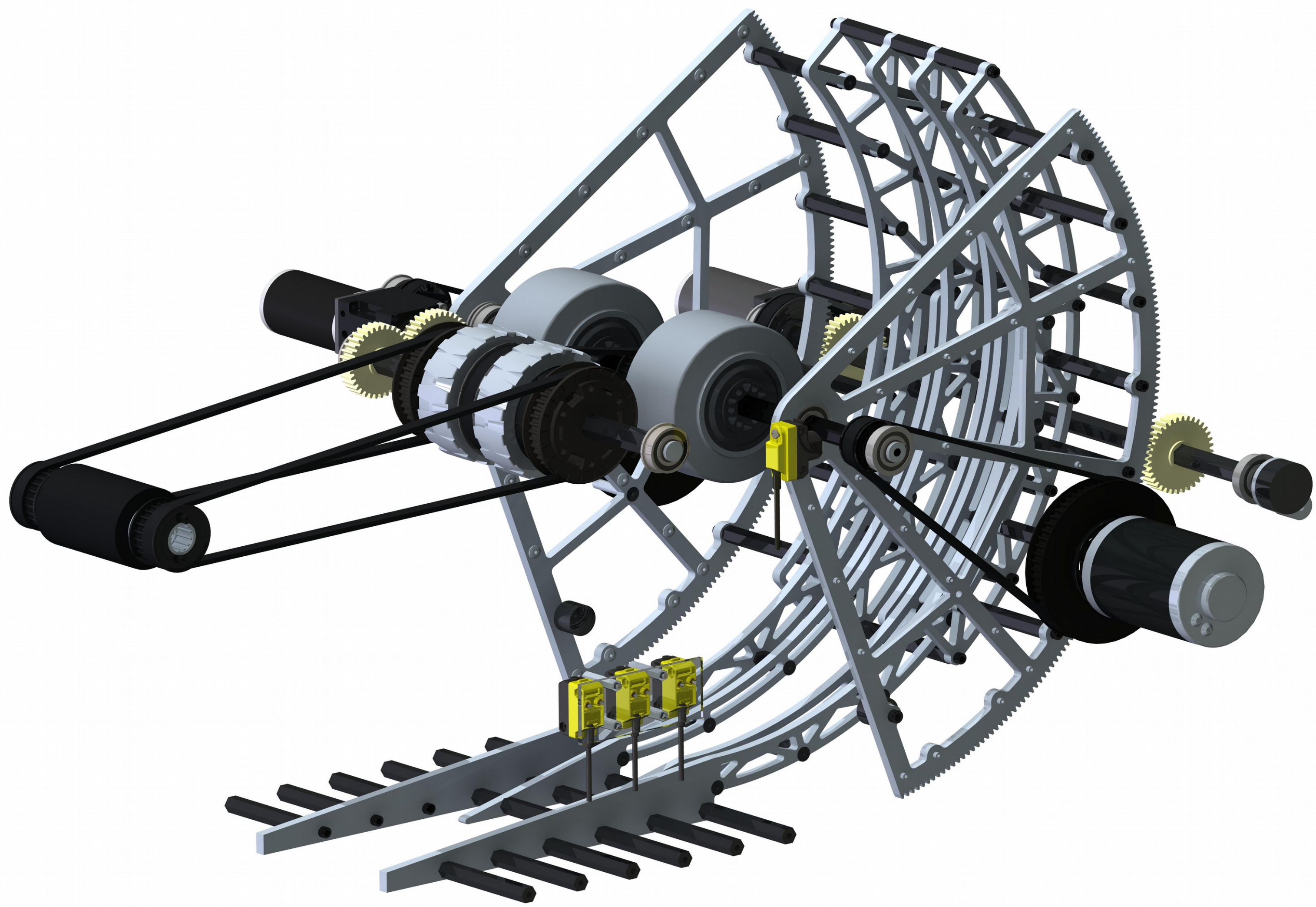
In past years, we have sometimes struggled with being able to gain control of a game piece quickly and effectively. In order to ensure efficient manipulation of boulders, we decided to create an omnidirectional intake with high-surface-speed rollers to pull the ball in from all directions. The versatility of this design was particularly well suited to the poor visibility this year. The arm was powered by a BAG motor using a custom worm gearbox and a final sprocket reduction stage. We also used the arm to manipulate the Cheval de Frise and Portcullis through pre-programmed sequences activated by the push of a button.

Specs:

- Nine foam rollers wrapped in friction tape
- Rollers powered by 775pro motor, 7:1 reduction
- ~20 ft/s roller surface speed

- SDP-SI nylon bevel gears used in front corners (A 1M 4-Y16016)
- Arm drive powered by one BAG motor, total reduction of 600:1
- Custom 180:1 worm gearbox built into 2” x 4” rectangular aluminum extrusion
- Adjustable friction clutch added to protect system in case of extreme overloading (McMaster Carr item 6524K33)
- 5-turn potentiometer downstream of the friction clutch
- Top and bottom limit switches and hard stops on arm

Feeder and Shooter



Once inside the robot, the boulder's position is detected by three photoelectric sensors and its movement is controlled by two timing belts. The ball's position is monitored and corrected on-the-fly as the robot barrels over defenses and collides with other robots in order to prevent accidental firing.

After extensive prototyping and after mocking up several options in CAD, we decided to use a hooded shooter with a continuously variable shot angle. We found that a powerful, straight-line shot minimized the effects of boulder variations. Both the high exit velocity of our shots and the large amount of backspin (Magnus effect) we put on the ball contribute to

our "laser beam" shot trajectory. The backspin on our shots also helps with batter shots and with preventing "bounce outs."

We used a ribbed hood design with interlocking static and moving ribs, leading to a seamless transition between the two different types of ribs, regardless of hood angle. The ribbed hood design also has the benefit of allowing to hood to be contoured to match the curvature of the ball, helping to minimize the side-to-side variation of our shots.

A 20 DP tooth profile (sector of a 400T gear) was waterjetted into the hood side plates to enable precise adjustability and to provide a stage of very high reduction. Because the hood is assembled with a series of standoffs, meaning that the individual parts aren't located with high precision, the hood must be twisted a fraction of a degree in order for both sides to engage with the gears on the drive shaft. This slight "springing" of the hood has the effect of creating an anti-backlash gear set, which contributes to the hood's being rigid with negligible play.

Specs:

- 2x 9mm belts to hold/manipulate the boulder while inside of the robot
- Feeder powered by 1 BAG motor, 30:1 reduction
- 3x Banner QS18VN6AF300 Photoelectric sensors to detect ball position for on-the-fly adjustment
- Hooded wheeled shooter
- 2x Mini-CIM, geared up 1:3.33
- Closed loop velocity control of main wheels at 11,400 rpm using Banner QS18VN6D photoelectric sensor
- 2x 4"x2" Colson Wheels for main shooter wheels, 2x 3.25" VersaWheel DT "kicker wheels" at ~⅓ the surface speed of the main wheels to provide the initial acceleration of the boulder
- 20 DP gear tooth profile waterjetted into hood side plates for accurate, low-backlash adjustment (0.125" plates stacked to minimize kerf taper). 2083:1 total reduction on BAG motor
- Grayhill 63R256 encoder used to measure angle—zeroed every time bottom limit switch is hit
- Hood ribs made of different radii to match the curvature of the ball
- Backspin curves the ball up for a straight line shot and makes close shots from against the tower wall easy and reliable
- "Laserbeam," straight line shot to minimize the effect of boulder variations
- Exit velocity of 44 ft/s with over 1000 rpm of backspin
- Peak performance of 93% accuracy at Mayhem in Merrimack

Climber

It was clear to us immediately following kickoff that consistent climbing would be very rare, particularly at the district and regional level, because of the poor effort to reward ratio, especially for teams short enough to fit under the low bar. The difficulty of this task made it appealing in that it would help to stand out from the crowd, but we were careful not to sacrifice too much in order to create a climber. Toward this end, we waited until after our first district event to finish and attach our climber, giving our programming team and our drivers more time with the robot instead of frantically scrambling all the way until bag-and-tag. After a few tweaks, we made our climber extremely consistent, so much so that we scored the most qualification scaling points in Archimedes. The location of our climber in the front of our robot also allowed for our "slam dunk" shot, which, in addition to pleasing the crowd, has also been 100% accurate after we set the proper angle before Finals 2 at Boston. We've used it dozens of times since then, including in the rare cases where we climb on only one hook or don't make it all the way up, and it has never missed.

Specs:

- Two-stage, telescoping elevator using a series of nested aluminum tubes (2×2, 1.5×1.5, and 1×1, all 1/16" wall 6061). Hard-anodized for superior performance as bearing surfaces.
- Powered by 1 CIM motor with 24:1 reduction (gear train incorporated into frame plates) for a 4-second climb
- Grayhill 63R256 encoder used for sensing
- Webbing spools on 5/8" drive shaft inside of the 2×2
- Unfolds with 2x 100 lbf gas springs (McMaster Carr item 9416K15)
- First stage extended with Vulcan constant force springs (SH16P40), second stage extends with a cascade-rigged strap
- Hooks spring out in two directions for increased reach
- Elevator released by extending out from under two "hold-down" bearings
- Passive latches to prevent backdrive
- All motion (Extension, retraction, pivoting up, latching, and hooks unfolding in two axes) is controlled/triggered by the one motor
- Dynaroll track rollers used at the bottoms of the stages to allow for low friction rolling in the very limited space (McMaster Carr items 3668K23 and 3668K24)
- Delrin pads used to take side loads
- Seatbelt straps unroll and act as tension members during the climb. Automatically spool back up with Vulcan constant torque springs (SV6G96).

Vision and Auto-Align

In the first couple days of the build season, we decided that our goal should be to automatically aim and shoot from anywhere in the courtyard, but that we should also have a solid batter-shot to fall back on. We used our batter-shot right out of the gate while we worked on completing our auto-align software, eventually implementing it later in the competition season. We used a Nexus 5X smartphone for a camera and as a coprocessor, running a Java app that we wrote to perform vision processing using OpenCV. The app sends distance and angle information to the RoboRio over USB using Team 3847's rioDroid.

In order to use the distance and angle information we receive from our vision code, we needed to create a few calibration curves. First of all, the information that our app uses to calculate distance is the height of the target in pixels from the bottom of the camera's field of view. Creating a curve to map height in pixels to distance simply required recording height in pixels at known distances and curve-fitting in Excel. We marked out distance from our mock goal in one foot increments and collected this data.

We also needed a curve to relate distance to optimal hood angle for our shooter. We could have combined these last two steps and just mapped height in pixels to hood angle, but having an actual distance number to display on the dashboard makes idiot-checking easier while developing our auto-align software. We repeated the above described process, taking shots from know distances from the goal to create a curve. Since the Nexus is mounted approximately 8 inches to the left of our robot's center-line, we also recorded what angle (according to our Java app) resulted in a dead-center shot into the goal, allowing us to also create a curve that relates distance to angular offset.

With calibration completed, the only part left is the control loop to use the vision data to aim. Initially we had trouble dealing with the "spring-back" that is apparent with flexible wheels like the 8" Andymark Pneumatic Tires and with tuning our control loop, but, with the help of a couple of tips from Austin Schuh and Jared Russell, we developed a control loop that aims within a fraction of a degree of the target in 1.5 seconds. We used two nested PID loops, the outer of which uses our gyro for feedback and feeds into the inner drivetrain velocity loop. We use our vision code in a "video-style" method, constantly updating the setpoint of our outer loop as new frames are processed to ensure accuracy. In order to prevent tire spring-back from causing an issue, we don't terminate our control-loop until after we've shot, allowing the integral term in our outer loop to hold the robot in place against the restoring forces of the wheels.