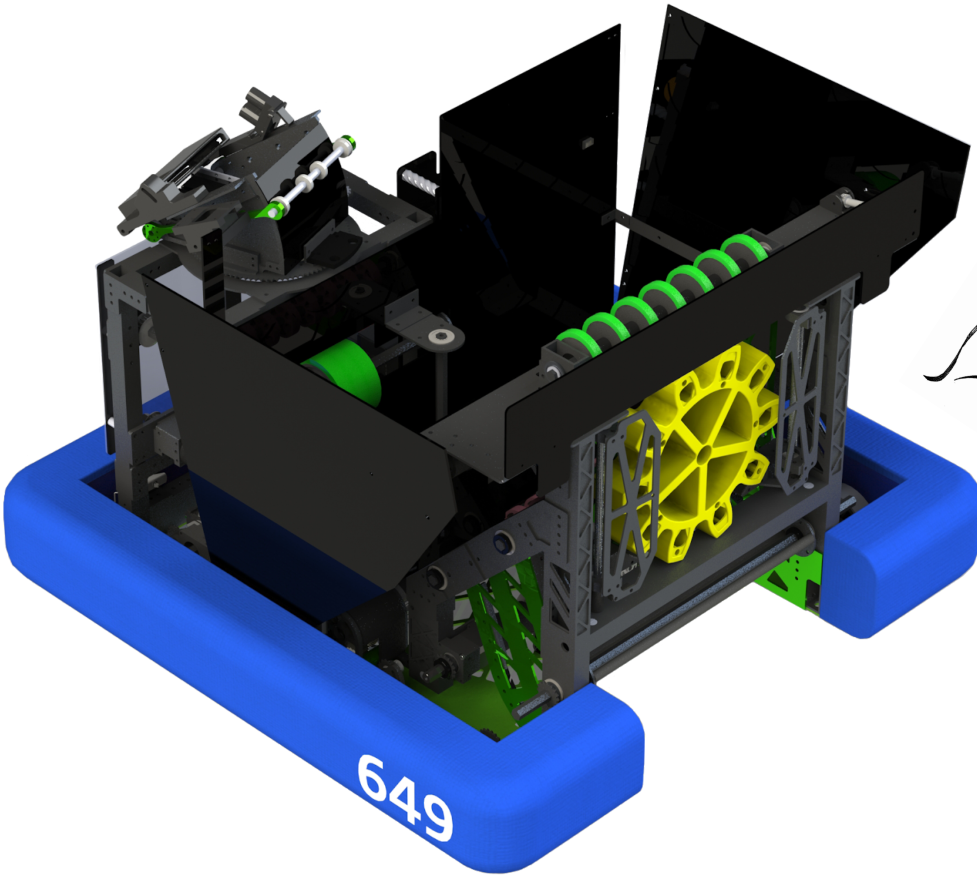


Nautilus

FRC Team 649, M-SET Fish
2017 Technical Binder

Saratoga High School
Saratoga, CA

Angle of attack

2016 was one of our most successful years so far, and we hoped to emulate that success by approaching this year's game with a similar mentality. Our goal for this game was to create an effective, reliable, and flexible robot to compete at the highest level of competition. We decided that the best way to accomplish this was to design a robot that was capable of accomplishing every scoring aspect of the game.

Game Analysis

M-SET spent the first weekend of the build season studying the game manual and developing a strategy going into build season so that we could prioritize our subsystems. We made sure we fully understood the rules of the game, and how to design a robot to play it best.

We identified the three scoring methods: gears, fuel, and hanging. As these are very different ways to score, we analyzed each independently.

One of the earliest design decisions we had to make was to decide which volume configuration we wanted to design for. We decided to go for the short configuration, as a lower center of gravity would allow us to move rapidly across the field, and the height of two feet was perfect for interacting with many of the field elements, like the hoppers and gear station.

Prototyping approach

This build season we emphasized the importance of prototyping by devoting a large fraction of our time to testing different ideas, more so than any of our previous seasons. First, each subsystem brainstormed different ways to solve the problems that their system posed. Then, rather than theoretically predict how our game pieces would behave and interact with our proposed mechanisms, we built proofs of concepts for each idea to observe and identify areas of improvement.

If at first you don’t succeed, try try again

We weighed the robot after bagging it and found it to be 117.4 pounds. Since the robot was bagged incomplete (missing the topper assembly and parts of the hang), we knew we needed to make significant changes in order to be competition-ready by the time our first regional came around.

We identified a few places where weight could be cut: the hopper, the agitators, and the hang.

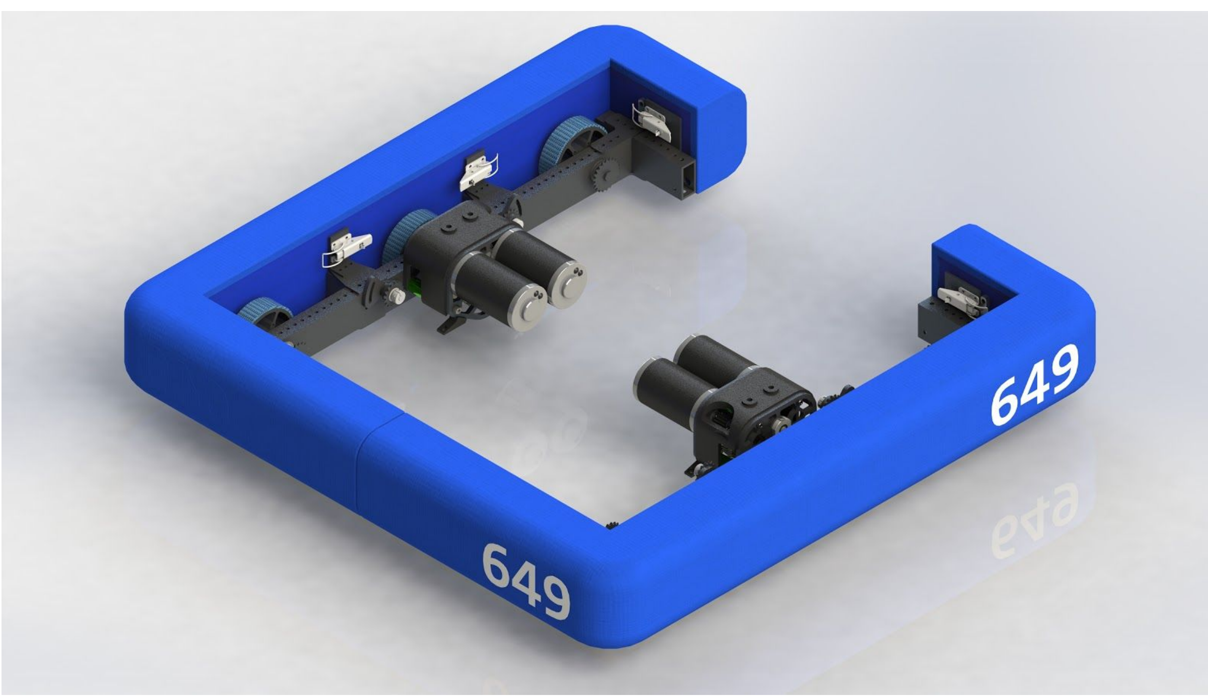
Drivetrain

Strategy

After this year's game reveal, there were several aspects that we had to keep in mind during drivetrain design. The flat playing field allows for low bumpers and smaller wheels, which give you a pushing advantage and lower reduction gearboxes. We also had to consider the two height and size configurations given to us, a tall skinny robot, or a short and fat one. We opted for the shorter size because of its better interactions with the gear peg and the field hoppers.

Final Design

We decided to create a broken frame west coast style drivetrain. This means we cantilever our wheel axles in order to avoid additional support on the outside of the robot. Doing so cut down on valuable weight which became much more scarce later in the build season.
We chose to use VexPro dual speed two cim gearboxes for their small form factor and ability to shift. It features drivetrain speeds of 7 feet per second in low gear, and 18 feet per second in high.



Shooter

Strategy

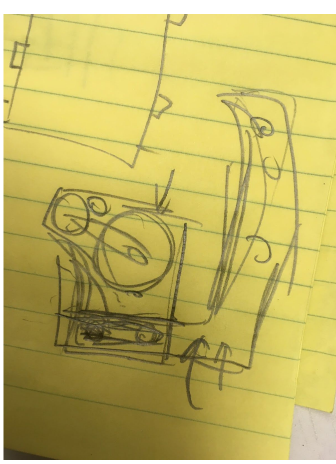
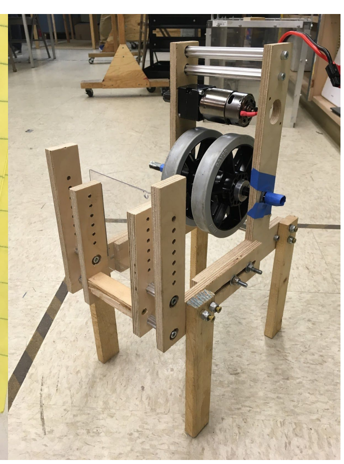
Each ball in this years game is worth a much smaller amount of points compared to game elements in past games. To equal the point value of 1 rotor in teleop, which required 1-6 gears, 120 fuel in the high efficiency boiler or 360 fuel in the low efficiency boiler were necessary, a number we deemed to be too high to replace gear scoring. However, we still saw the value of fuel, since scoring from gears had a limit, and fuel would act as a tiebreaker in cases where both teams had an equivalent number of rotors and hangs.

Prototyping

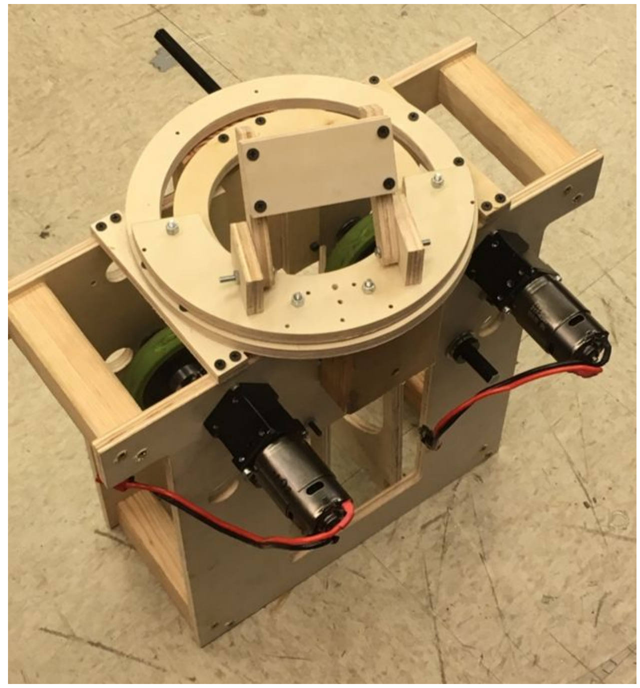
After analyzing different types of shooters (flicker, catapult, linear slide) we decided to pursue a flywheel because we needed to be able to shoot large volumes of balls quickly and consistently. The flywheel shooter would be easy to package in a robot with tight size constraints. The other shooting methods could not handle more than one ball at a time without sacrificing consistency and this design would allow for better integration and packaging relative to other subsystems.

Variables

- ❖ Wheel material, diameter, and weight
- ❖ Number of wheels and spacing
- ❖ Compression
- ❖ 360 degree turret
- ❖ Shooting angles


Findings



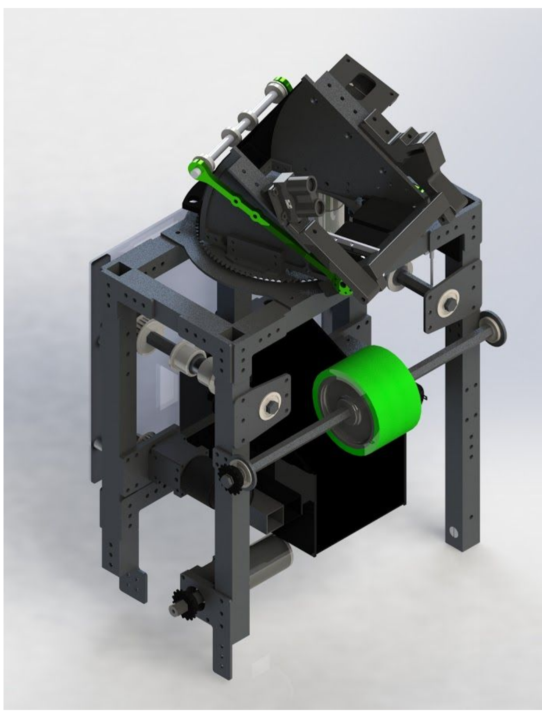
After multiple weeks of testing, modifying, and more testing, we discovered that 6” Colson wheels combined with 0.75”worth of ball compression worked best to shoot the balls both quickly, accurately, and with minimal spin without a significant loss in wheel velocity after each ball shot, especially when weights were added alongside the flywheels. We also found that minute adjustments to the left or right made a significant difference in our shot accuracy, so we decided to make a turret to cover all potential shot-angles. Having a variable shot angle would also allow us to shoot from multiple positions on the field, an important factor in both autonomous programming and overall robot capability. In order to more easily package our turret, and for significantly increased shot power, we decided to shoot with 2 wheels instead of one. A 2 wheel shooter would also eliminate unnecessary spin, assuming the flywheels are spinning at relatively the same speed.

Initial design

With regards to the flywheels, having established optimal distances and speeds, we directly translated the dimensions on the prototype to the assembly in Solidworks, integrating the same wheels, weights, and motor positions. The ramp leading the balls into the range of the flywheels had originally been made of two slabs of wood on the prototype, but we determined that a bent, polycarbonate sheet would work with greater consistency. This 1/16” polycarb curve was sandwiched between two other polycarb boards and mounted to the back railing of the shooter structure. But due to the nature of a front-feeding shooter, another feeding flywheel was essential to push the balls up the ramp and into the two shooter flywheels, a component we decided to add above the beginning of the ramp and just low enough as to not be contacting the ball at the same time as the shooting flywheels, which would induce extra, unnecessary spin to the projectiles, throwing the trajectories into disarray.

However, due to a more extensive and rigorous prototyping period with the shooter, we were left with less time than usual to make our “final bot” design of the subsystem, and elected not to finalize the shooter topper, which consisted of our shooter’s turret and aiming. Instead, we left the topper off of the final bot, and spent the following six weeks designing a high-functioning turret with appropriate sensor mounting.

Final design



To implement the turret-shooter, we decided to mount the turret — both the variable hood and the various sensors — on a large custom sprocket powered by its own separate motor. The sprocket, with a hole just large enough for fuel in the center, has the ability to spin 360 degrees and can be controlled via the encoder mounted besides the turret. As for the pieces on the turret itself, the hood consists of a ⅛” polycarbonate sheet that bends to predetermined angles with the help of two servo motors attached up front, and the sensors (phone, camera, and LIDAR) are mounted alongside rails bordering the turret hood. However, because the code necessary for the implementation of the LIDAR required far too much time that could be better spent on driver practice, the sensor was deemed impractical and was removed from the assembly. The ⅛” polycarb hood was also split into two 1/16” pieces in order to relieve stress in the servo motors; as it turns out, two thinner pieces have the ability to slide past one another, alleviating the buildup of tension of a thicker piece.

Discounting the topper section of the shooter, we adhered largely to the initial design of the subsystem, with some small exceptions.

Upon testing this finalized shooter system with our practice robot, we discovered that the fuel units would often roll in place within the shooter feed due to low compression of the ball and the slick, low-friction finish of the polycarbonate feed. Our solution? We added a layer of thick grip tape to the polycarbonate, which mostly solved our issue with friction, rendering our shooter’s fuel feed smooth and uninterrupted.

To solve our low fuel compression issue, we used a larger-diameter feeder wheel, switching out BaneBot feeder wheels for a single omni wheel. This also helped decongest the fuel in the hopper, as the omni wheels allowed for balls in the hopper to slide past the feeder wheels with greater ease, promoting fuel flow within the hopper and eliminating a major source of jams within the hopper.

Ball Intakes

Strategy

Our intake system had three objectives.

1. Integrate with the gear system
2. Don’t intake crushed balls or gears
3. Intake balls

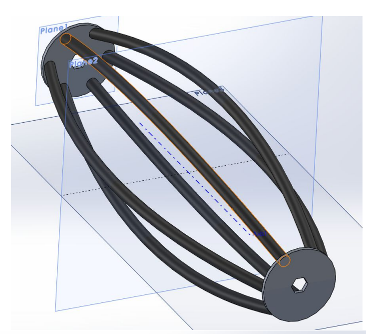
Prototyping

The intake system was designed to intake fuel quickly, while remaining compact and space efficient.

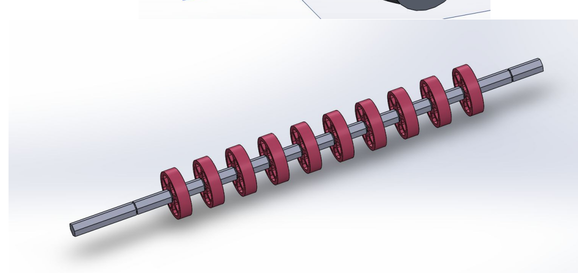
Design Considerations

Open vs. Closed Frame
Although an open frame cannot use the full length of the robot to intake balls, we found it was more beneficial to not pull the fuel over our bumpers.

Deposit balls over top of hopper vs. underneath
We chose to push balls through the bottom even though they need to be forced upward into a possibly full hopper because this method saves space. By using another set of rollers to compress the balls as they pass through, we can prevent balls from falling through.

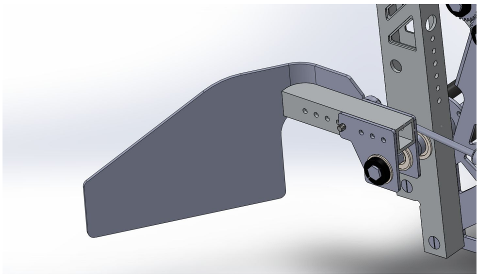


Front Roller Design: surgical tubing vs. wheels
After testing, we found that at high speeds a surgical tubing intake system whips out and grabs closeby fuel while the wheels, although easier to assemble, were not as effective at retrieving balls. The surgical tubing allows us to have a wider ball-intake radius.

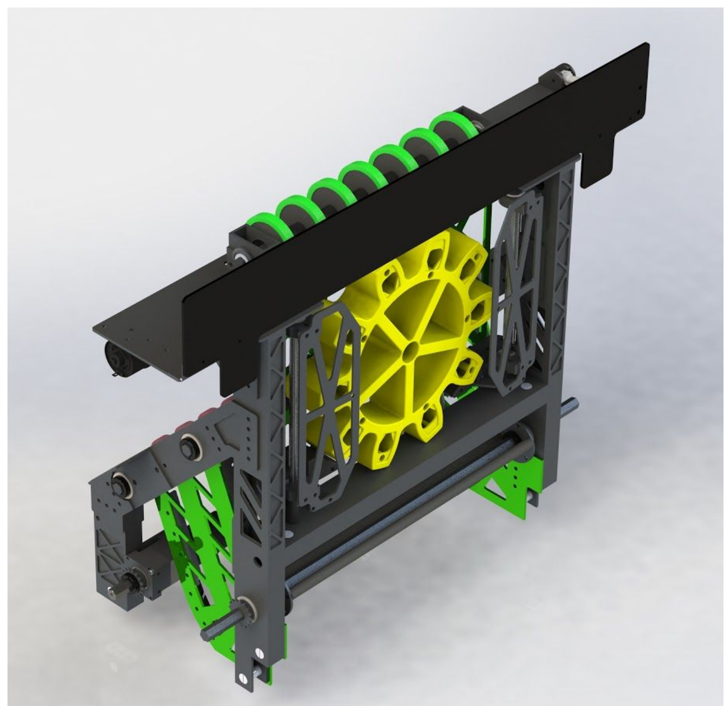


Ball Funnel

Adding a funnel allows us to “herd” balls in front of our intakes, helping our in-frame intake come in contact with balls.



Final design



Our final design matched our prototypes very closely. We created a direct ball path from the floor into the bottom of the hopper. Our front roller uses two bands of surgical tubing to grip the balls, which push the balls up a curved ramp. At the top of this ramp, a set of rollers using flexible wheels forces the balls further upwards and prevent any backflow of balls. We found throughout testing that the ball funnels made a negligible impact with intaking and therefore was cut due to weight restrictions.

Gear System

Strategy

We determined that scoring gears to turn rotors would be the main scoring method in most situations. It has the highest realistic scoring potential of 200 in qualification matches, and 300 in elimination matches, and a single robot could reasonably score 140 points with just gears. Gears do have a downside, however. The scoring of gears is nonlinear, and there is a hard cap on the number of points that can be scored with gears.

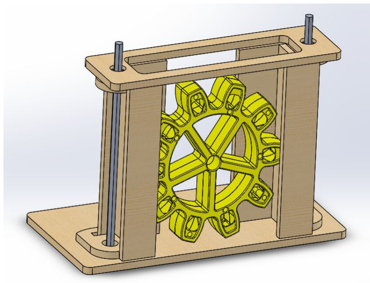
Prototyping

After identifying gears as our main focus for scoring points, we recognized the importance in having a reliable gearing system. But before prototyping our gear-scoring mechanism, we first had to make a couple decisions regarding our gear game plan.

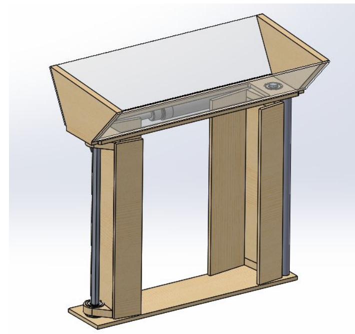
The first decision was to funnel gears from the Retrieval Station into our robot, instead of picking gears up off the ground, as this subsystem would require a large amount of floor space within the frame perimeter and resources which we felt could be better allocated elsewhere. We then identified two viable options for gear delivery—active and passive. We tested the feasibility of a passive gear system very early in the season, but were concerned that this system relied on the agility and skills of the human pilot. By going with an active deployment system, we eliminated the factor of uncertainty and human error. As the build season progressed, we realized that the weight of the gear would often overcome the tension of the spring, making it even more difficult for the pilot to lift the gear. This made the ability of the gear deployment system to “push” the gears to the base of the spring (where stability is greatly increased) a necessity.

On the Origin of Gear Systems: An Evolution

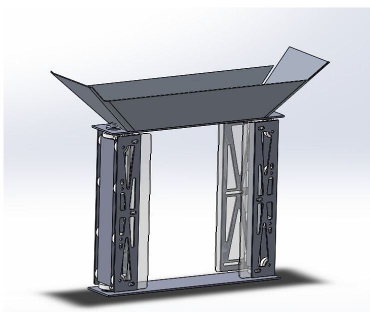
V1



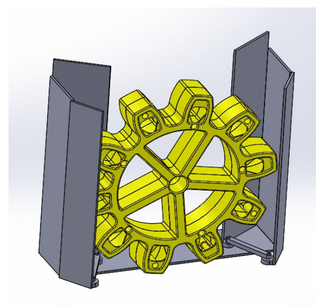
V2



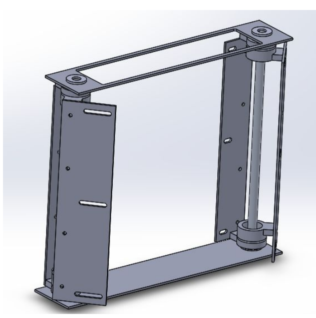
V3



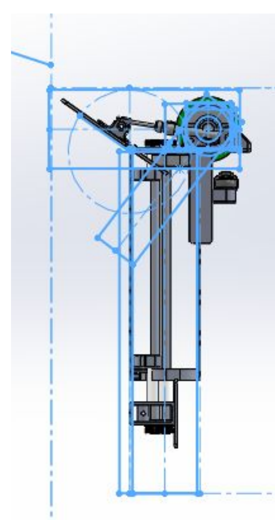
V4



V5



Gear Funnel Design and Rationale

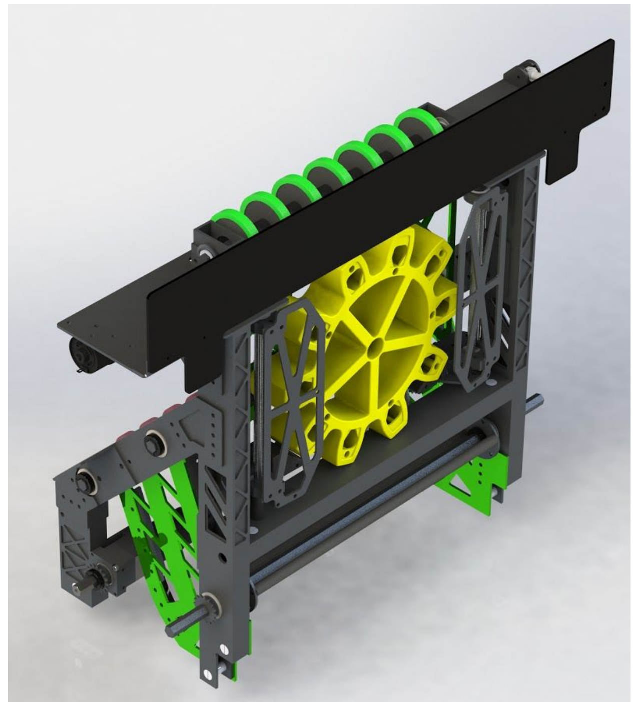


The gear funnel (the system that rotates the gear and transports it from the Retrieval station into the gear deployment system) also underwent many iterations, the first of which was a completely passive angled funnel system similar to that on top of gear deployment systems V.2 and V.3 (pictured above). This design, although effective, required a lot of width on the robot directly taking space away from the hopper, it was decided to opt for a roller design that used very grippy Banebot wheels to actively push the gear into the deployment system . This new design not only made packaging easier it also decreased intake time thus cutting the overall cycle time.

To the right is the final gear funnel and intake system used, along with it the geometry sketched overlaid which was essential for calculating the optimal distances and angles.

We also determined that the gear system needed to go on the **same side of the robot as the intakes**, so travel to the Retrieval Station and to the Gear Pegs would be cleared of balls.

Gear and ball systems



The design process for the subsystem integration began with the need for a structurally rigid frame as this the only piece closing our “Broken Frame” drivetrain (please refer to the Drivetrain section for more details). This is also the reason the weight reduction pockets do not extend past the cross member box tubing as it could jeopardize structural integrity. It was noticed that an intake and gear system could both fit if placed on top of each other, however the exact location of the cross member came to be by a series of compromises by both gear and intake subsystem teams in order to optimize the gear peg to gear contact area and the intake surgical tubing expansion (please refer to the Ball Intake section for information).

Hopper

Strategy

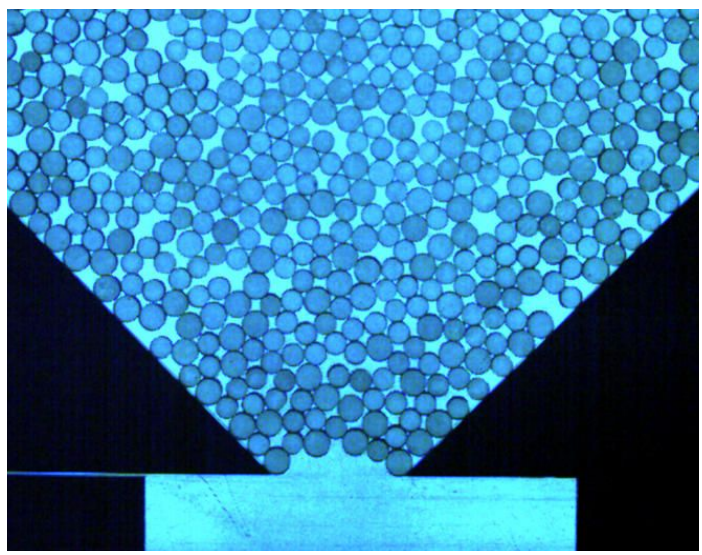
The hopper was one of the most difficult subsystems to create, mostly because its main constraint was to take up as much available space as possible in as little weight as possible.

Prototyping

The hopper was constrained to the remaining volume of the robot, after the intakes, gear system, hang, and shooter were mounted to the drivetrain. This made for an interesting geometry to work with, so when we first started testing, the balls jammed a lot. After identifying this problem, we did some research into the granular flow of materials and the physics behind force chains.

Force Chains

Occur when material supports its own weight against the sides of a funnel. It's basically the formation of an arch (a super strong structure) due to the nature of round objects flowing from a wide to narrow channel.

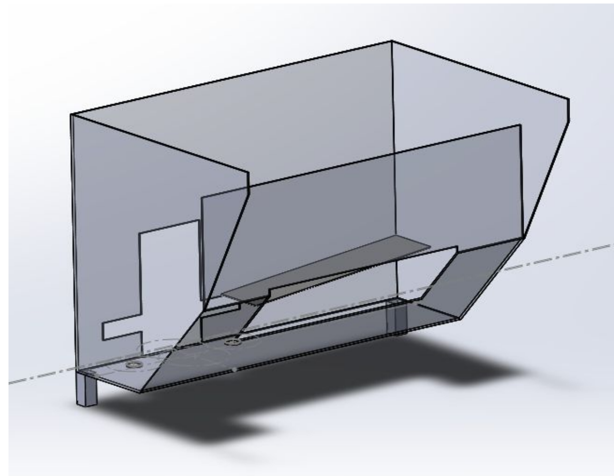


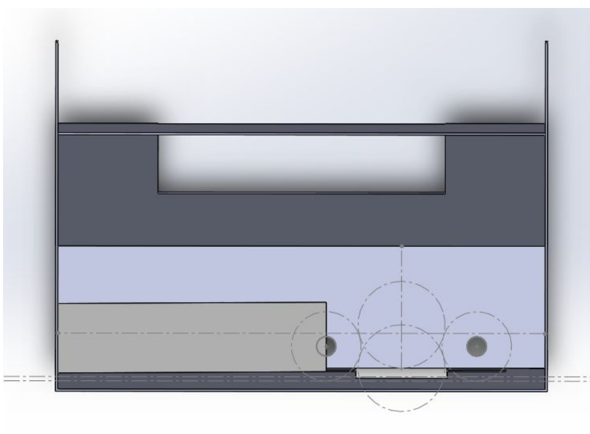
Hopper Requirements

- ❖ Removable
- ❖ No jams
- ❖ Hold at least 50 balls
- ❖ Feed consistently

Findings

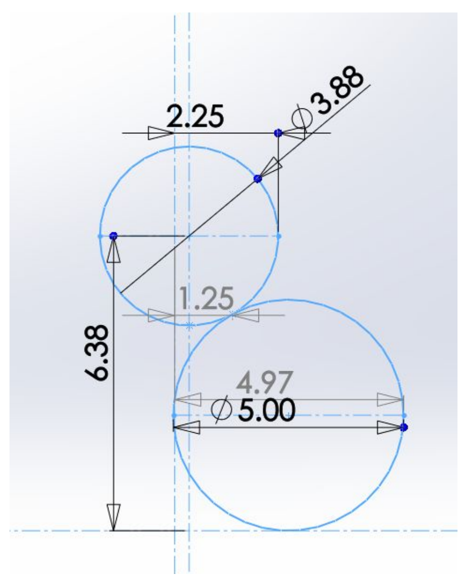
From our research, we found that separating the feeding layer from the ball-storage layers to be extremely important in helping prevent jams, so we accomplished this by using a downward slope to prevent balls from occupying space between the two layers.



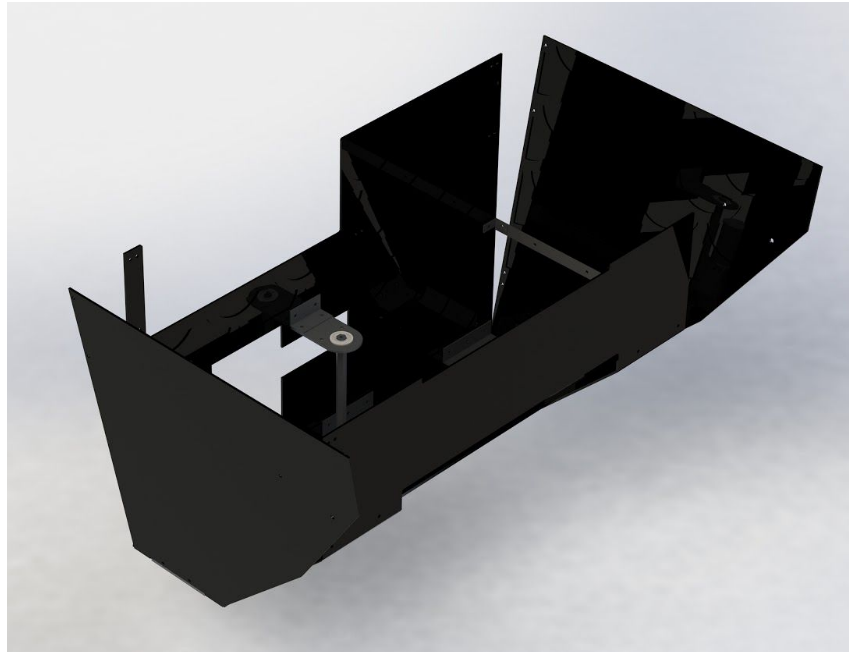


We also identified two main methods to move balls through the hopper: active and passive (gravity). We found that using gravity alone does not suffice in stopping force chains from forming. So, we developed agitators to actively move balls through the hopper.

We also use an active roller to feed balls from the hopper and into the shooter. To identify the correct geometry for components interacting with the balls in the hopper, we sketched it out (side view) in SolidWorks.



Final design

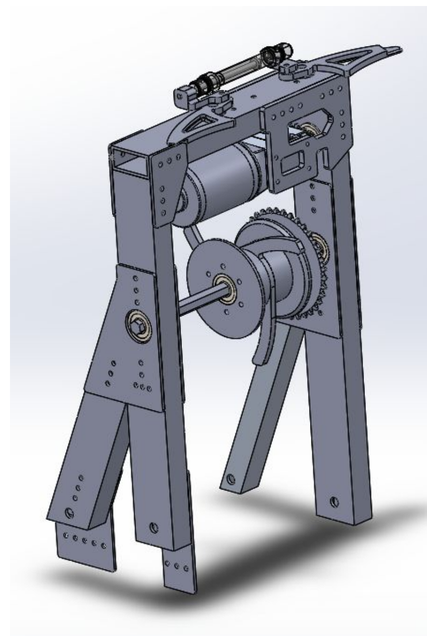


The hopper we bagged had an aluminum floor and un-pocketed side panels; we easily reduced weight by replacing the floor with polycarbonate, and pocketing the side panels. Additionally, we decided to hinge the side panels at the bottom to allow the hopper to expand, increasing ball capacity and improving ball flow.
Our original agitators comprised of a lot of heavy aluminum tubing, gussets, and required two motors. They also turned out to be severely underpowered. After a week of further agitator prototyping, we replaced the original agitators with a new pair of agitators which mounted with much lighter aluminum plates, and utilized a single, more powerful motor instead of the two weaker ones we originally had. Not only did it save a
significant amount of weight, it was also more effective.

Hang

Strategy

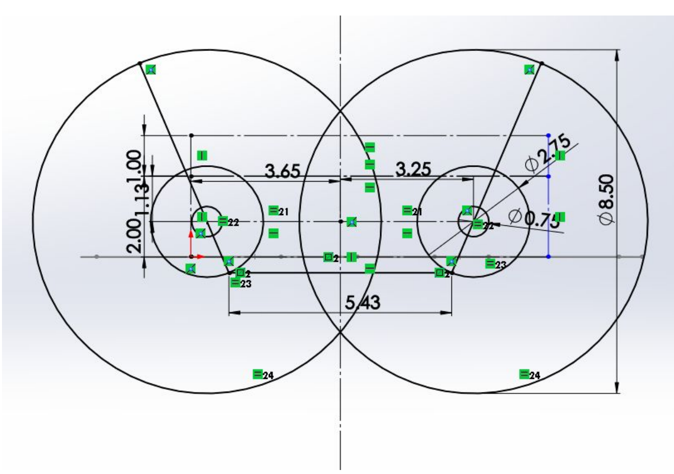
We decided that if we wanted to build a high scoring robot, hanging was essential. Not only is it a quick way to score 50 points, these points cannot be made up for by another team, unlike gears and fuel.



Because we established that hanging is an essential scoring element of this year's game, we knew we needed a robust and reliable hang mechanism. Going into the design process, we originally thought of having a completely independent hanging mechanism for reliability, robustness and speed.

Initial design

We first designed a drum with multiple hooks to grab the rope. This rope would be centered by two fingers that protrude from the top of the superstructure. A single piston actuates the fingers which "grabs" the rope, centering it right over our drum. This would allow the drivers to have greater room for error, since the fingers would account for some misalignment, consequentially increasing the speed and reliability of our hang.



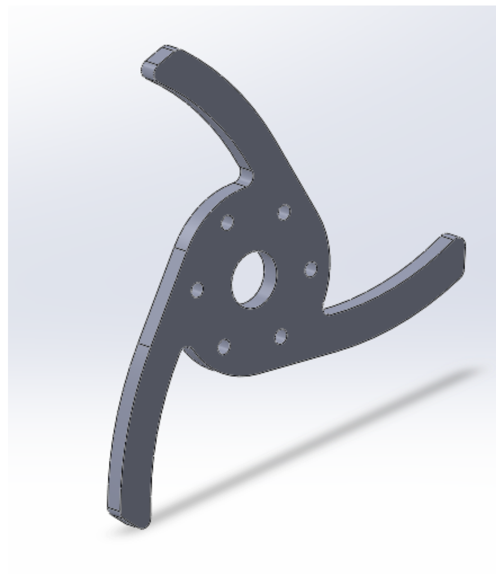
Fingers

The fingers center the rope and line it up for the drum to easily catch and wind on. The fingers were originally made to be single strips of aluminum that wrapped around the rope.

However, as we tested this method, we found out that having single strips does not accurately center the rope and sometimes even pinches the rope before center. Therefore, we ended up with this pincer-like design as pictured below. By making the fingers this shape, we still ended up with the same amount of reach to grab the rope, while also centering the rope.

Drum

We decided to have this drum powered by a CIM so that if the motor stalls, there won't be any serious repercussions, unlike a 775 pro, which would burn out immediately after stalling. Another design feature of the drum is that of the hooks. We wanted to have three hooks on a single axis of rotation. This would increase the number of chances that the hook has to grab onto the rope and therefore would increase the speed of our hang.



Final design

We realized that in order to save the most amount of weight, a complete redesign of the hanger subsystem was necessary. We discussed many possibilities, such as designing a lighter dedicated hang system, or adapting either the gear intake roller or the ball intake so that they would able to be taken off. Eventually, we settled on repurposing the ball intake to also be able to hang by adding velcro strips to both the surgical tubings and the axle of the intake. The velcro allowed for quick engagement with the rope, resulting in a less than 5 second hang. By rethinking the hang subsystem, we were able to both fulfill the task effectively and save around 10 pounds in tubing and gussets alone.

Software Overview

Philosophy

This season M-SET's software teams goal is to automate and improve all process on and off the robot. Our robot's control system aims to allow the robot's operators control the robot easily with various semi-autonomous functions while still allowing the operators to have full manual control of the robot when needed. We worked to enable front end simplicity and intuitive functionality through layers of abstraction and forethought integration both up and down the stack (up to drivers and down to hardware).

Overview

With 12 Talons, 2 Servos, 4 pistons, and 7 subsystems there was a lot of systems that needed control and automation. To accomplish this in addition to the functionality of our Talon SRX's we integrated 12 sensors, many of which we have have never implemented before (for example our lidar sensor). Though we have not just implemented new sensors, we have also used new algorithms to utilize them.

Autonomous Sequences

This season unlike many others offered a wide variety of autonomous sequences and many ways to score in auto. Our goal for autonomous is to be able to score the most points possible in any alliance. This is why we wrote 13 unique autonomous sequences so we can be prepared on any alliance. Our robot can successfully hang a gear on all three sides of the airship and shoot while hanging a gear from two of them. We also have the ability to shoot from the hopper in auto if fuel is a priority. We ensured the accuracy of our autonomous through well tuned PID loops. In addition as a fail safe we have a gyro based driving algorithm that we can run in the case that our PID based driving fails.

Benefits of Semi-Auto

With all that is going on in a match the easier it is for the operators to control the robot, the more successful you will be. This is where semi-autonomous control systems come into play. That's why this year we implemented numerous semi-autonomous functions into our code. These functions allow the operator to control the robot much faster and allows the robot to do its tasks with greater control. Though as safety of the robot and others is always a priority we have many safeties implemented in our control system to ensure nothing on the robot breaks. An example of this being the hall-effect sensors we use to ensure the turret doesn't turn too far.

Software Subsystems

Turret

This season was the first time any current members of the software team had to write a control system for a turret. One of the primary reasons our team decided to design a turret this season was for additional control.

Sensors

- ❖ Absolute Encoder (x1)
    - ➢ To accomplish this added control we used an absolute encoder to accurately track the position of our turret and allow us to PID to saved positions.
    - ➢ We also used the absolute encoder to ensure that our turret doesn't turn pasts its' wires.
- ❖ Hall-Effects (x3)
    - ➢ Hall-Effects are used to zero the turret at the start of the match because one absolute encoder rotation is not one rotation of the turret. So to ensure the turret always zeros at the right place we use a Hall-Effect and two magnets on the turret to ensure the turret centers in the right place.
    - ➢ Hall-Effects are also used to ensure the turret doesn't go past the turret turns past its' wires.
- ❖ Nexus 5 (x1)
    - ➢ The nexus 5 is used for vision aiming using an app loaded onto the phone.

Shooter

This season the goal with our shooter flywheels was to be able to recover quickly and maintain a consistent RPM.

Sensors

- ❖ Photoelectric Sensors (x2)
  - ➢ Used to *bang bang* the flywheels by going full power until the RPM of the flywheels nears the targeted speed then switching between more accurate motor speeds. This is done to recover the flywheels fast while still maintaining a consistent RPM.
- ❖ Lidar (x1)
  - ➢ Reads distance from boiler then refers to a table of positions and determines flywheel speed target.

Drivetrain

Sensors

- ❖ Encoders (x2)
  - ➢ Encoders are used for PID based driving.
- ❖ Potentiometers (x2)
  - ➢ Potentiometers are used to select which autonomous sequence is run. This is needed because code cannot be pushed in pits.
- ❖ Gyro (x1)
  - ➢ Used as a backup if PID based driving fails so the robot can drive straight.

Hood

It is hard for the operators to judge a correct hood angle therefore there is a need for semi-autonomous hood control.

Sensors

- ❖ Lidar
  - ➢ Measures distance then adjusts the hood angle based on a tuned table of values.

Scouting App

This year, the software team created a scouting app to aid us in scouting teams for alliance selection. The scouting app is extremely useful because it allows easy organization of scouting data and facilitates an objective data based scouting analysis. The scouting app also allows less experienced scouters to collect data while leaving the more experienced head scouters to do more in-depth and qualitative scouting on specific teams.

Front End

The front end of the scouting app is the android app that is run on tablets. Each tablet is assigned to a team on each alliance, and every match, scouters enter numerical data on the team they are scouting. At the end of the match, scouters also write qualitative observations on the team they were scouting. Once the form is submitted its contents are saved as a text file on the tablet.

Back End

When data needs to be displayed for a pre-match scout or for alliance selection, the tablets are hooked up a computer and the text files are transferred through ADB (Android Debug Bridge). Then, the python backend compile data by team, calculates averages, and ranks teams based on certain metrics. Once these calculations are run, the data is sent back to the tablets and the data is displayed in a searchable and easy-to-read way.

Competition Season

Sacramento (CADA)

Because of the extensive changes that were made between our initial design (the bagged robot) and our final design (the final robot), we needed a cohesive plan to make the most efficient and effective use of our unbag time at competition. Thus, the **eight-hour plan** was born! It's attached below.

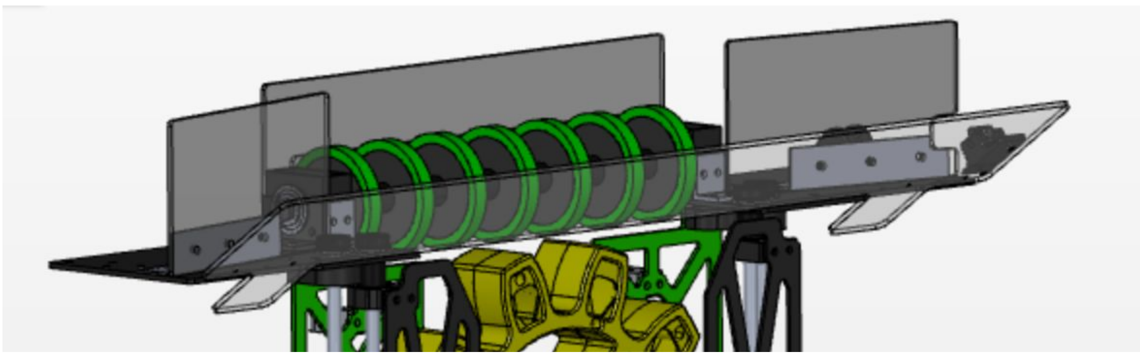
	person accountable	tools needed	parts involved	weight loss/addition	estimated amount of time (min)
remove hopper	nicole				
check solenoids	samay/basil/valen				
remove intake funnel pistons and solenoids	kai	hex key set, ziptie snips, wrench		-1	10
flash router	samay	router		0	2
remove climb	akhilesh	1/4-20 hex key		-7	6
remove agitator tubes	nicole	1/4-20 hex key, 10-32 hex key		-3	10
shooter wheel spacers	nicole	hex key that fits into collar	3d printed flywheel spacers, hex shaft collars		10
install polycarb gear flicker front panels	akhilesh	black buttonhead, 10-32 hex key	front panels	-0.1	
install gear centerer	akhilesh	10-32 hex key, vhb tape	centerer pieces, VHB, 10-32 bolts	0.05	
install radio	samay				5
mount autonomous selector	samay	vhb tape	autonomous selector, potentiometers		5
install polycarb gear funnel	akhilesh	drill, 10-32 hex key, rivet gun, wrench	polycarb gear funnel, rivets, 10-32 nuts and bolts		
replace compliant feeder with omni	nicole	hex key set	omni wheel		7
install battery holder	samay	rivet gun	rivets		
install topper	david	rivet gun	topper, rivets, zipties		
install axis camera	david	drill + stepper bit	3d printed axis mount		
other wiring in general and SENSORS	samay	electronics toolbox, vhb tape			
move wires under air tanks	samay				
install hopper gear box cover	nicole		gear box covers, 10-24 bolts	0.1	
remove old intake, install new intake	kai	collar hex keys, 10-32 hex key, ziptie snips, gentle pursuader	velcro, flanges, zipties, aluminum collars	0.5	15
driver's tape on gear flap	kyle	gaffer's tape			
install sponsor panel	akhilesh	rivets, rivet gun, vhb tape	sponsor panel, standoffs, polycarb strips		
install ratchet	kyle	drill, imp. hex key set, wrench	10-32 nuts and bolts		
install hopper retention brackets	nicole	drill, imp. hex key set, wrench	10-32 bolts and nuts		
put green acrylic in	kyle	adhesive	acrylic triangles		
install electronics cover	samay	10-32 hex key and wrench, rivet gun	10-32 nuts and bolts		
fit hopper	nicole	none			
fit bumpers	david	screw driver			
lights	kyle				
check solenoids	samay			0	15-45
possibly cut talon bolts	samay	dremmel,5			
white rows indicate that the task is independent					

Las Vegas (NVLV)

We will also be competing at the Las Vegas Regional during Week 6 of the 2017 Competition Season.

Changes made to the robot include an improved gear funnel to better center gears at the retrieval station to cut down on cycle times and padding the shooter guide to reduce our spread of shot balls.

In qualifying matches at the Sacramento Regional, we had trouble lining pup to the retrieval station from across the field when visibility was low. Furthermore, many gears went over our gear intake and into the hopper. We fixed these two problems by adding a gear funneling system behind the gear intakes which widens our target area for gear retrieval and prevents gears from going into the hopper. This has drastically reduced our cycle time from ~2 gears in tele-op to 5 gears in tele-op.



We also learned from other teams at the Sacramento Regional that compressing the material around the projectile rather than compressing the game piece directly improves the shot even with inconsistencies in the game piece. To replicate this in our already-designed shooter, we used foam (compressible material) to pad the shooting ramp (from hopper to shooter). We also added a ball guide to funnel balls through the shooter, centering them before contact with the flywheels, rather than deflecting them after they've been shot. In testing, we've shot faster, and more consistently: with proper alignment and tuning, our shot has an accuracy of about 50% as compared to approx 25% with the old system.