
VEX WORLDS 2019
Team #: 2616Y
Division: Arts
High School
Phone #
ENGINEERING NOTEBOOK
2019 HACKATHON SEASON
MLH
MAJOR LEAGUE HACKING
I Robots
VEXROBOTICS.COM
MLH
MAJOR LEAGUE HACKING
East
Hacks
VEX ROBOTICS
Mango

Team 2616Y: Y-Naught

Caylin Payne
"Yurrrrr"

Vidhya Shunmugasundaram
"I forgot my calculator at home"

Nafessa Jaigirdar
"Please don't make me cry"

Caroline Cheung
"Sub-tle"

Iris Kim
"There is no 'I' in team"

Kaileigh Scott
"Happiness is having a cat as a friend"

Shir Goldfinger
"Only YOU can prevent wildfires"

INDEX

- Problem + Solution
- Brainstorming
- Calculations
- Programming
- Strategy
- Other / Misc.
- Building

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

Continued from page

THE GAME

The 2018-2019 VEX Robotics Competition Game, Turning Point, consists of many parts. The game is played on a 12x12 ft foam-mat, and it is surrounded by a sheet-metal and polycarbonate perimeter. There are 8 caps that can be low scored on the playing field tiles or high scored on 6 posts around the field. Additionally, there are 9 flags: 3 of which are low flags and can be toggled by robots and 6 of which can only be toggled by being hit with balls. Toggled simply refers to the flag's status of which color of alliance it shows. A flag is toggled when the flag's pointer is not nested in the detent and the flag is not touching a robot of the color alliance for which the flag would award points. When toggled, points are awarded to the red alliance if the pointer is to the left of the detent, and awarded to the blue alliance if the pointer is to the right of the detent. Teams also score points for alliance parking at the end of the match on their own alliance platform, or by center parking on the center platform, which can be used by either alliance. During the autonomous period, robots must operate and react only to sensor inputs and to pre-programmed commands in order to score points. This period lasts for 15 seconds, and robots may only move on their side on the field. On the other hand, the driver controlled period lasts for 1 minute and 45 seconds where the drive team operates their robots. Each robot must be pre-loaded with one ball before the match begins. During the match, robots may be in contact with up to 2 balls. Robots also must start the match on their designated starting tile. Robots cannot access more than one cap. Caps have one red and one blue side as well as a yellow core in the middle. These caps can be flipped to the correct alliance color according to the robot's alliance color.

Continued to page

SIGNATURE

DATE 5/7/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 5/7/18

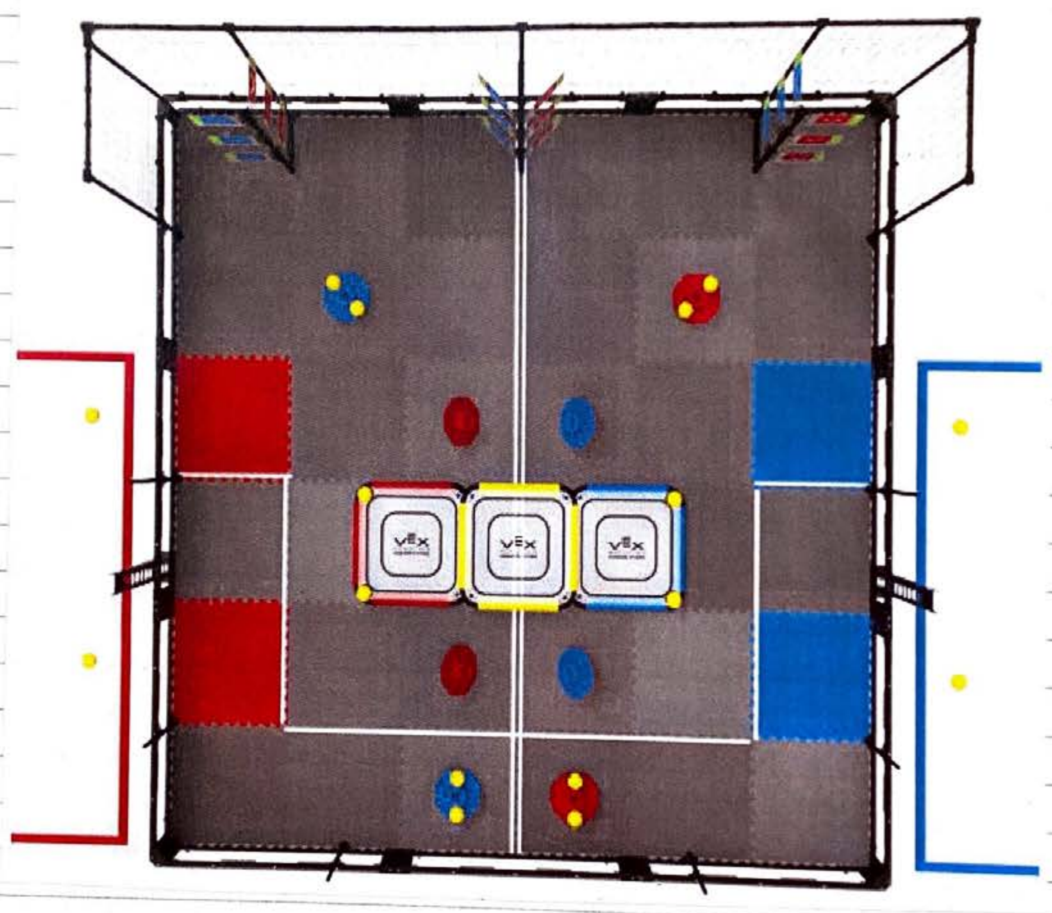
Continued from page

THE GAME (cntd.)

Scoring:

Toggled High Flag : +2
Toggled Low Flag : +1
High Scored Cap : +2
Low Scored Cap : +1
Alliance Parking : +3

* Autonomous Bonus : +4
* Center Parking : +6



SIGNATURE [signature]

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 5/7/18

Continued to page

DATE 5/7/18

Continued from page

CHALLENGES OF GAMES

The VEX Robotics games tend to have simple, yet intricate tasks to accomplish. Turning Point, however, presents us with much more challenges at first glance. The High Flags compel teams to come up with a structure design to be able to shoot balls to toggle these flags, in order to score extra points. On top of that, robots should have considerable height to them in order to score the High Score Caps. On the topic of caps, robots' intakes should be constructed in a way that allows them to flip caps so that the correct alliance color is facing up. Also, intakes should be able to easily, and securely, place caps onto posts so that they do not fall off due to an unstable intake structure. The matter of weight is a topic of concern as well. Robots with lighter weights possess faster speed, which can be an advantage in autonomous periods, while heavier robots are less likely to tip over and act as a great defense mechanism. In general, both weights have their own pros and cons, but in the case of Turning Point, the challenge associated with the center parking comes to mind. The center parking platform will nonetheless be fought and sought after and will involve pushing of other robots. In this circumstance, a heavier robot would be more beneficial as it would stand firm in stance despite pushing from opposing robots. However, other factors must be ruled out before deciding upon a heavy robot. Height comes into play once again when robots toggle low flags. Additionally, the autonomous line that prohibits opposing robots to cross over to the other alliance's side may serve as a setback, or another challenge to think about. The autonomous line would have to be considered when programming the autonomous code, so that the robot does not cross the line.

NJ

Continued to page

SIGNATURE

DATE 8/8/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 8/8/18

PROPRIETARY INFORMATION

Continued from page

INITIAL THOUGHTS ON GAME

Upon seeing the game reveal on Turning Point, we could not help but compare it to the previous game, In the Zone. We immediately saw how much more complex Turning Point is as opposed to stacking cones from In the Zone. Because Turning Point is so much more complex, as it contains more components, we came to the conclusion that the new game is a bit harder than last year's. Turning Point presents us with more challenges to face. The most apparent challenges we saw were the ball launching as well as the intake. This is because ball launching is a foreign field for us, as we are unfamiliar with them since In the Zone did not include that game component. We would have to incorporate the ball launcher design and structure into our robot design. Also, we would have to conduct research on where the optimal location would be for a ball launcher structure. On the other hand, the intake design was brought to our attention as we watched the game reveal. In last year's game, the intake had mostly one job: to grab onto cones. However, in Turning Point, we believe the intake would have to perform multiple tasks: flipping caps as well as grabbing onto them. Thus, finalizing on an intake would be a challenge as we would have to rule out different types of intakes in order to select the best one. Another thing we realized was that defense autonomous would not be allowed since there is an autonomous line that deters robots from crossing over to the other alliance's side.

SIGNATURE

Continued to page

DATE 5/8/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 5/8/18

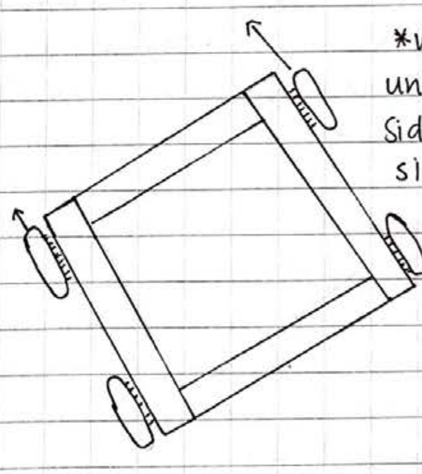
BRAINSTORMING: DRIVE

The drive is the most important part of any robot. The drive is what connects the robot to the ground, allowing the robot to move around. It serves as the robot's base, and without a stable base, we will not have a stable robot. Stability is crucial because, as seen from last year's game, robots frequently tipped over, disabling them to participate in the game rounds. We also expect the overall robot to be lighter in weight because there does not seem to be a need for the robot to reach great heights (which means less metal) or a need for additional lifts, like a mobile goal lift. With a smaller weight, our robot this year will probably be geared toward speed. Thus, this year's drive will most likely require less torque in order to move the robot as a result of its weight.

In order to evaluate the different variety of drive types available to us, we conducted an analysis to compare and contrast each drive type to each other that will be most optimal for Turning Point:

1) Tank Drive

The tank drive is built in the shape of a rectangle or square. In this design, there are wheels on both the right and left sides of the drive. The wheels on both sides of the drive are powered by separate motors. With pieces on all four sides, the base is fairly stable and unlikely to tip over. This is because there are no empty spots in the drive, but rather a closed off 4-sided drive. The wheels on a tank drive do not steer as they are locked pointing forward/backward. Steering is accomplished by varying the speed of the different sides. The tank drive is capable of zero radius turns as well, but it would be great at holding the parking platform.



* with the rubber grip tank treads, the tank drive may help the robot against pushing

Continued to page

SIGNATURE [signature] DATE 5/10/18

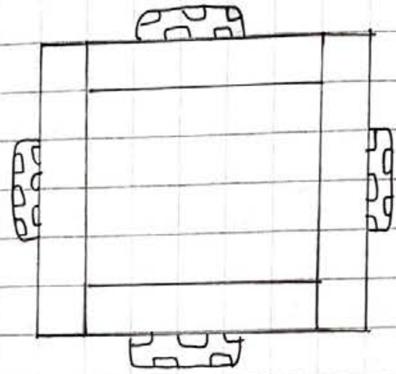
DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 5/10/18

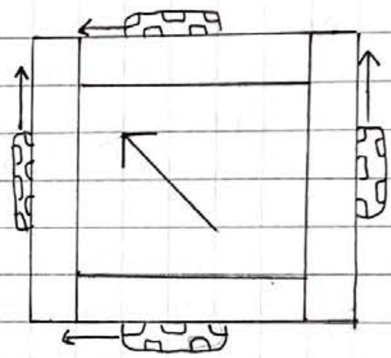
Continued from page

BRAINSTORMING: DRIVE (CNTD.)

2) Omni-Directional Drive

A drivetrain that can move in any direction at a given moment, without waiting for the wheels to ~~str~~ steer is called an Omni-Directional drive. These drives utilize special wheels, called "omni-wheels." Omni-wheels are wheels with small rollers around the perimeter that freely spin perpendicular to the wheels' rolling direction. With the small rollers, the wheels can slide sideways with very low friction. This is beneficial because the wheels allow the robot to glide smoothly sideways instead of not being able to move to the sides. The two configurations in which the omni-wheels can be arranged are shown to the right.



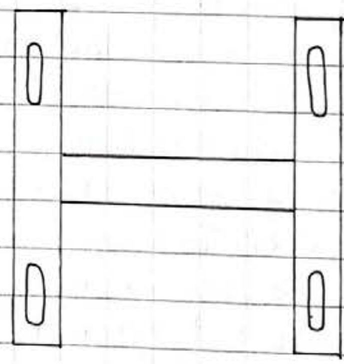


Since omni-wheels do not have any sideways friction, the wheels facing forward/backward can drive without the wheels facing right/left dragging.

The problem with the omni-directional drive is the need for multiple motors, with only some of the motors contributing to the robot's forward motion most of the time. The above example sketches of omni-directional drives require 4 motors.

3) H-Drive

The H-shaped drive is another sturdy and stable base, It utilizes less metal than other bases. The central c-channel bar adds to its structural integrity while allowing for a front and back area to perhaps load game pieces or additional parts for the robot's design. However, this drive can make wide turns, which can be problematic during the game. The H-Drive also may require multiple motors as well, which is not very desirable since we can only use 8 motors total for our robot.



SIGNATURE

DATE 5/10/18

Continued to page

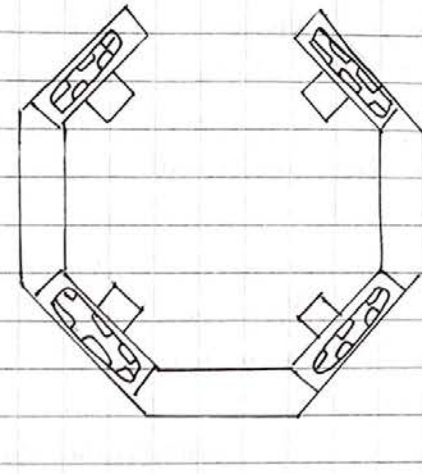
DISCLOSED TO AND UNDERSTOOD BY

DATE 5/10/18

Continued from page

BRAINSTORMING: DRIVE (CNTD.)

4) X-Drive



Like the H-Drive, the X-drive can strafe left, right, up, dawn with its 4 omni-directional wheels arranged in an "X" formation. This drivetrain can go in all x and y directions and is not constrained by any direction. However, the X-drive lacks torque and can be pushed around easily. That being said, these driver are great for speed and agility. X-drives will usually have greater frictional losses since there is friction between the omniwheels and omniwheel rollers. Strafing may be advantageous in Turning Point due to the flags and poles all being lined up on the perimeter walls.

OUR DECISION: H-DRIVE

The drive is one of the most important parts of the robot, so after analyzing different drive designs, we realized that the H-drive was the most logical choice for us in Turning Point. Though a tank drive is more stable and can ensure the alliance ~~period~~ parked or center parked bonus, the negatives of the tank drive outweigh the potential benefits. In general, tank drives are significantly slower than H-drives and tend to be larger. Tank drives will take up more space, meaning there is less space left for building a stable lift, intake, and flywheel for our 18"x18"x18" design. Similarly, X-drives, though quick at strafing and turning, have multiple cons. For instance, they will usually have greater frictional loss due to the friction between the omni-wheels and omni-rollers, will not drive straight unless the robot's center of mass cannot go over bumps well (like the center platform), and are somewhat difficult to build on. Since Turning Point is a rather complex game that involves multiple objectives, our goal this year is to heavily on ~~a~~ creating a versatile robot that can accomplish all aspects of the game relatively well. H-drives are fast, stable, maneuverable, and capable of strafing, if we choose that route.

Continued to page

SIGNATURE

DATE 5/10/18

DISCLOSED TO AND UNDERSTOOD BY Mufassa

DATE 5/10/18

Continued from page

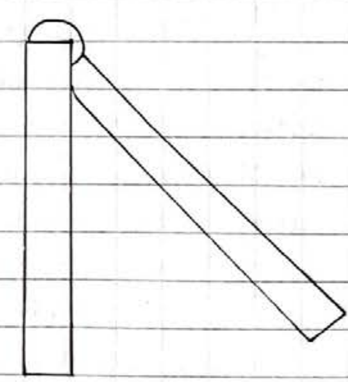
BRAINSTORMING: LIFT

Another essential component of the robot is the lift. The lift is a mechanical system in the robot that allows the change of positioning, typically height, of another mechanism or object. Lifts are normally used in conjunction with a claw, intake, or launcher to move, place, reposition, or retrieve objects to places the robot would otherwise be unable to reach. The lift serves an important role in the game Turning Point. This is because the lift would not only be able to reach the posts, but also reach the ground to flip over the caps with ease. A concept to understand regarding lifts is degrees of freedom. A degree of freedom refers to an object's ability to move in a single independent direction of motion. To be able to move in many directions means something has many degrees of freedom. For example, moving up and down is one degree of freedom while something that can move up/down and right/left has 2 degrees of freedom. We believe that we will only need to move in the up and down directions, thus we will build a lift with one degree of freedom. Unlike last year's game, In The Zone, we will not need another life like a mobile goal lift, and there is no need for our robot to reach such high heights since we are not stacking any objects. That being said, our lift for our robot should be able to reach the height of the posts.

There are a great number of different types of lifts, each with their own advantages and disadvantages. Therefore, we conducted an in-depth analysis to compare and contrast lift types to rule out which lift is best for Turning Point.

1) 2 Bar Lift

A 2 bar lift offers great simplicity, little weight, but little to no control of the caps. It uses rotational motion to rotate an arm typically made of c-channels to reach high places. However, since the 2 bar lift only contains 2 bars or pieces of metal to make up the lift, we believe that the 2 bar lift will not be able to reach such heights like the height of the posts.



Continued to page

SIGNATURE

DATE 5/15/18

DISCLOSED TO AND UNDERSTOOD BY Mussa

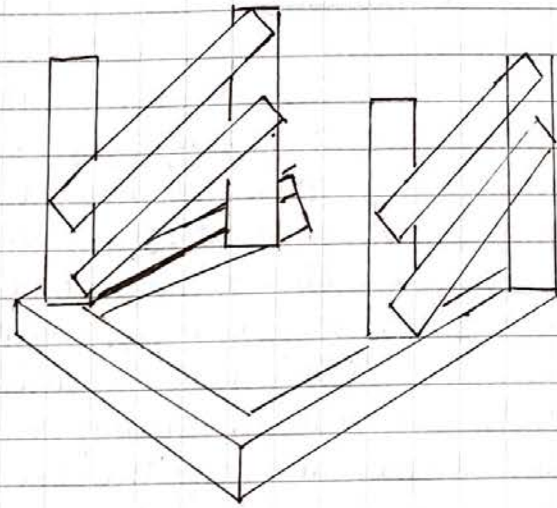
DATE 5/15/18

PROPRIETARY INFORMATION

Continued from page

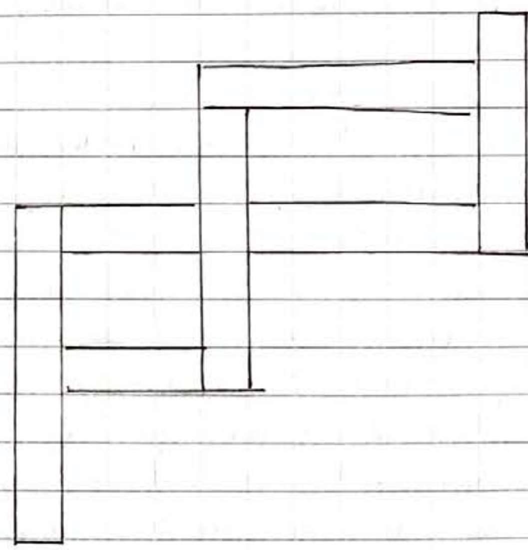
BRAINSTORMING: LIFT (CNTD.)

2) 4 Bar Lift



Like a two bar lift, a four bar lift is relatively easy to construct, and gives the intake a simple way to always remain vertical. It is composed of c-channels with movement in rotational motion, which allows objects attached to the lift to rotate at an arc, usually resulting in the attached mechanism to reach out as it gets higher. However, any bar lift is limited by their comparitively low reach, but they can reach higher by stacking more bars above the existing bar. Compared to the 2 bar lift, the 4 bar lift is able to reach higher heights.

3) 6 Bar Lift



Essentially, a 6 Bar Lift is a taller version of a 4 bar lift. That is, it is basically two 4 bar linkages superimposed on top of each other. What it does is effectively mak a 4 bar where the top bar can go all the way from the top corner of the robot to the ground at the bottom of the robot. This allows the arm to be several inches longer than a four bar. Another difference is that, with its higher height, the 6 bar can reach greater heights as well as has a greater torque effect. That is, the intake is being lifted at a greater distance and thus has a greater torque effect on the arm. One thing to consider is that there will be a greater force ~~component~~ on all the components in the lift, and could burn out motors + break gears.

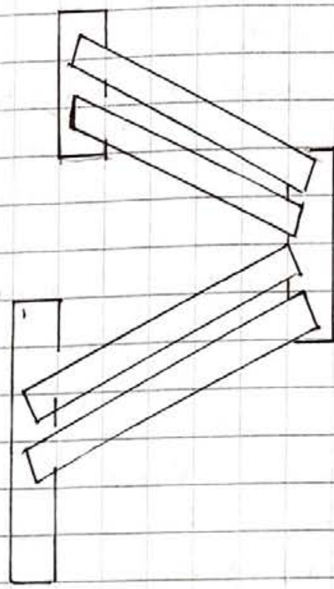
Continued to page

SIGNATURE [signature] DATE 5/15/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 5/15/18

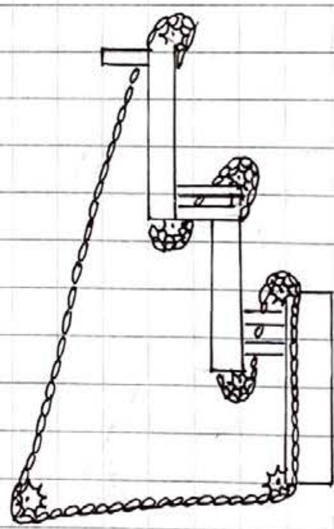
BRAINSTORMING: LIFT (CNTD.)

4) Double Reverse 4 Bar



Reverse bar lifts are a variation of the bar lift that utilizes a bar lift attached to another bar lift facing opposite of the first lift. This allows the robot to reach somewhat higher than a typical bar lift and lift objects attached to the lift in a near vertical motion. Despite being a variation of the bar lift, reverse bars are slightly more difficult and time consuming to build as the mechanism connecting the two bar lifts on each side is usually a sort of gearbox that moves the second bar lift as the first one is moving. This lift is overall stable and reliable, but requires more metal, and would add weight to the robot, thus slowing it down. It is also noted that our robot may not need to reach such great heights in Turning Point.

5) Linear Lift



Linear lifts use linear motion to raise a set of linear slides vertically. The mechanism that achieves this motion varies from a rack and pinion gear to a pulley system. The linear lifts are generally compact and stable, however there can be an issue with the speed of the lift. In other words, the friction from the linear slides can slow down the entire lift if not properly mitigated. Additionally, this lift is difficult to tune and optimize. Single stage elevators are able to lift up only one extension of their length, but multiple stage mechanisms are also possible. That is, by stacking multiple linear elevators together, a mechanism that can reach up much higher than their own height can be created.

Continued to page

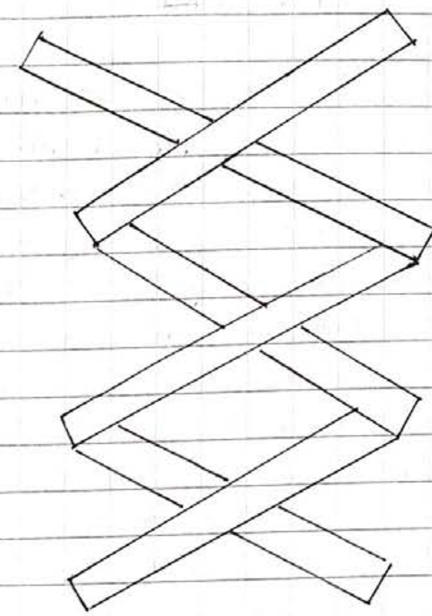
SIGNATURE [signature] DATE 5/15/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 5/15/18

Continued from page

BRAINSTORMING: LIFT (CNTD.)

6) Scissor Lift



The scissor lift allows for a wide platform to attach mechanisms to, has low friction, less jamming, and is easier to power with elastics. Scissor lifts allow the robot to reach high places by a series of two c-channels attached together at the center called stages. These stages are attached at each end of each c-channel and stacked upon each other. However, the scissor lift is harder to maintain and can be slower or less efficient. That is, not only is this type of lift difficult to build, but require a large amount of material. They have horizontal instability and become unstable at higher levels. As a result, robots with scissor lifts run the risk of tipping over because the lift sways sideways.

OUR DECISION: 6 BAR LIFT

After analyzing all the potential lift designs— from a 2 bar lift to scissor lift— we ultimately decided that a six bar lift best suited our needs for this year's game. The six bar lift is relatively compact and effective, especially in comparison to ~~a~~ ~~elev~~ linear lift, which has the tendency to be slow. The six bar will allow us to score consistently on the middle and top sized posts when compared to the four bar. We are aware that a six bar adds the horizontal component when extended, potentially making it difficult to score caps onto posts. That being said, it is something our driver(s) are getting used to accounting for during the game. On the other side, many drivers with linear lifts will struggle to line up their robots at the edge of the field quickly. While two-bar lifts are too simple of a design, and scissor and double reverse four bar are too complex for Turning Point's game objectives, the six bar equips us with just what we need. The six bar's height is appropriate in lining up with the posts, and it does not contain too much metal, which adds to its simplicity.

Continued to page

SIGNATURE

DATE 5/15/18

DISCLOSED TO AND UNDERSTOOD BY

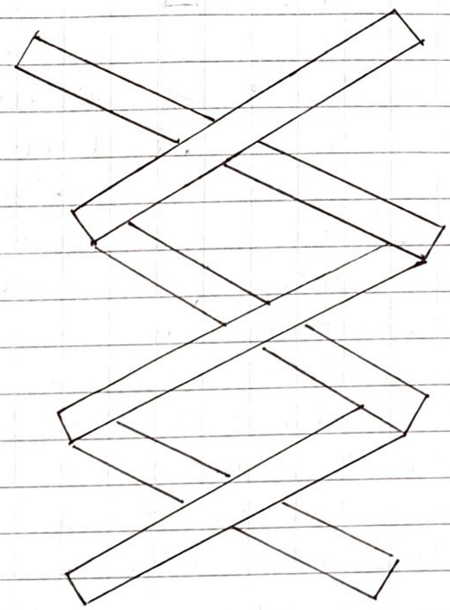
Melissa

DATE 5/15/18

Continued from page

BRAINSTORMING: LIFT (CNTD.)

6) Scissor Lift



The scissor lift allows for a wide platform to attach mechanisms to, has low friction, less jamming, and is easier to power with elastics. Scissor lifts allow the robot to reach high places by a series of two c-channels attached together at the center called stages. These stages are attached at each end of each c-channel and stacked upon each other. However, the scissor lift is harder to maintain and can be slower or less efficient. That is, not only is this type of lift difficult to build, but require a large amount of material. They have horizontal instability and become unstable at higher levels. As a result, robots with scissor lifts run the risk of tipping over because the lift sways sideways.

OUR DECISION: 6 BAR LIFT

After analyzing all the potential lift designs— from a 2 bar lift to scissor lift— we ultimately decided that a six bar lift best suited our needs for this year's game. The six bar lift is relatively compact and effective, especially in comparison to a ~~elev~~ linear lift, which has the tendency to be slow. The six bar will allow us to score consistently on the middle and top sized posts when compared to the four bar. We are aware that a six bar adds the horizontal component when extended, potentially making it difficult to score caps onto posts. That being said, it is something our driver(s) are getting used to accounting for during the game. On the other side, many drivers with linear lifts will struggle to line up their robots at the edge of the field quickly. While two-bar lifts are too simple of a design, and scissor and double reverse four bar are too complex for Turning Point's game objectives, the six bar equips us with just what we need. The six bar's height is appropriate in lining up with the posts, and it does not contain too much metal, which adds to its simplicity.

Continued to page

SIGNATURE [signature] DATE 5/15/18

DISCLOSED TO AND UNDERSTOOD BY Nafessa DATE 5/15/18

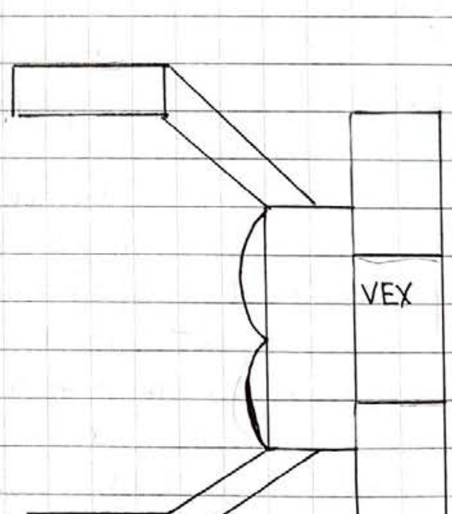
Continued from page

BRAINSTORMING: CAP INTAKE

An intake is a mechanism that allows the robot to acquire and store game objects. The term is also used to loosely refer to any mechanism that can grab objects. While an intake is technically not needed in order to have a running robot, it is an essential mechanism for teams to be competitive. This year's game calls for an intake that is able to pick up the caps, rotate and flip the cap, and possibly take off the opponent's caps from the posts, as a defense mechanism. At the same time, the cap intake should be able to perform these tasks in a secure and fast way. Since the caps are octagonal prisms with cylindrical protrusions in the center, it will be relatively difficult to construct an intake that can grip and flip the caps. Thus, we conducted another analysis to weigh in on the pros and cons of each cap intake type.

1) Claw Intake

A claw intake can allow for a secure grip on the game objects. Instead of expanding horizontally, the claw for Turning point would expand vertically to have a firm hold on the caps. Or, the claw would grab onto the cylindrical bottom protrusion. Although claws are heavy and a bit harder to operate, they leave a lot of room for correction. We would also have to add friction padding or some sort of grip to help the claw achieve a strong and secure hold on the caps. We can program the claw to rotate as well in 360° and flip the caps to the correct color. Regarding the posts to score points, our driver would have to be very precise in picking up the caps at the right spot as well as precisely placing them on top of the posts.



Continued to page

SIGNATURE

DATE 5/17/18

DISCLOSED TO AND UNDERSTOOD BY Nafessa

DATE 5/17/18

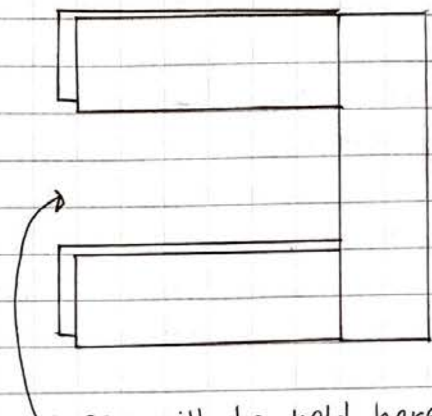
PROPRIETARY INFORMATION

Continued from page

BRAINSTORMING: CAP INTAKE (CNTD.)

2) "Waffle" Intake

The "waffle" intake is essentially a passive intake modelled after an actual waffle flipper. It is composed of 2 c-channels on each side of the intake, with space in between. This space allows the robot to slide the cap into the intake and secure its grip to flip the cap to the right color and set it on the posts to score. The waffle flipper will slide the caps between the cylindrical protrusions. This intake probably adds about the same weight to the robot as the claw intake since they use relatively similar amounts of metal.

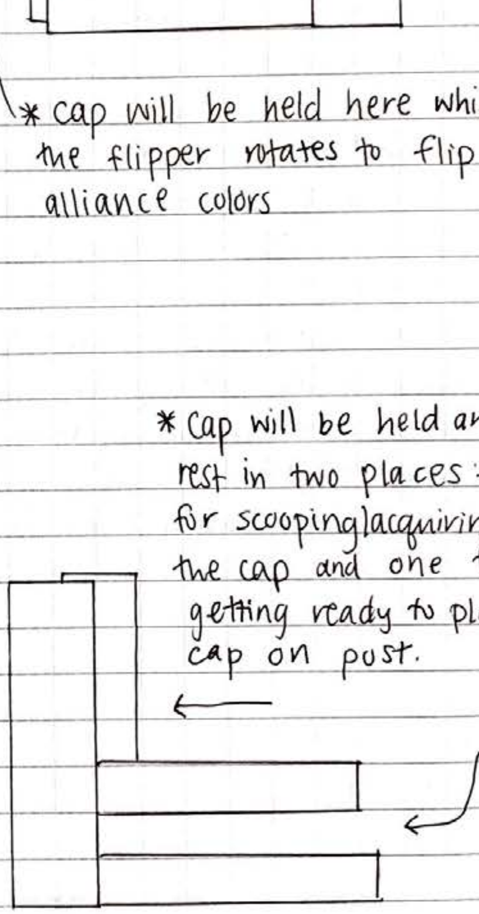


* cap will be held here while the flipper rotates to flip alliance colors

3) ~~Dis~~ Cap Intake

Like the waffle intake, the cap intake is passive and slides the cap into the intake. The way it can slide the intake is that the intake is built in the form of an "L" shape, kind of like arms. Thus, the cap placed on the intake can be placed onto the posts. On the other hand, the cap intake can flip the cap's color in a different way. That is, the tip of the intake can flip the cap over by moving in a quick, upwards motion. That being said, we can utilize this intake effectively for defense because we can simply knock off the opponent's caps. ~~to d~~ We can also secure our caps in the intake by flipping them onto the other side of the "L" shape.

* Cap will be held and at rest in two places: one for scooping/acquiring the cap and one for getting ready to place cap on post.



Continued to page

SIGNATURE [signature]

DATE 5/17/18

DISCLOSED TO AND UNDERSTOOD BY Nefissa

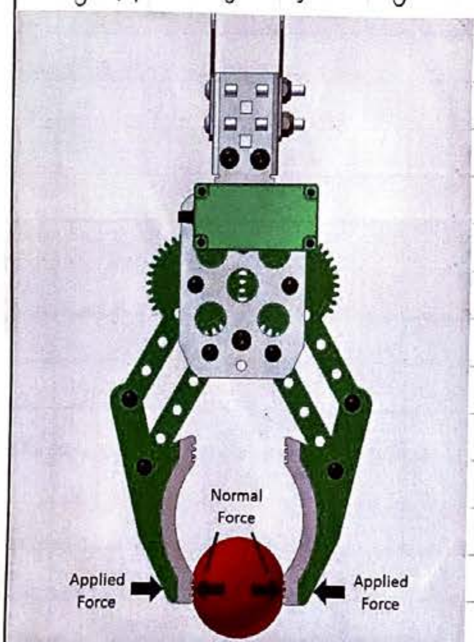
DATE 5/17/18

PROPRIETARY INFORMATION

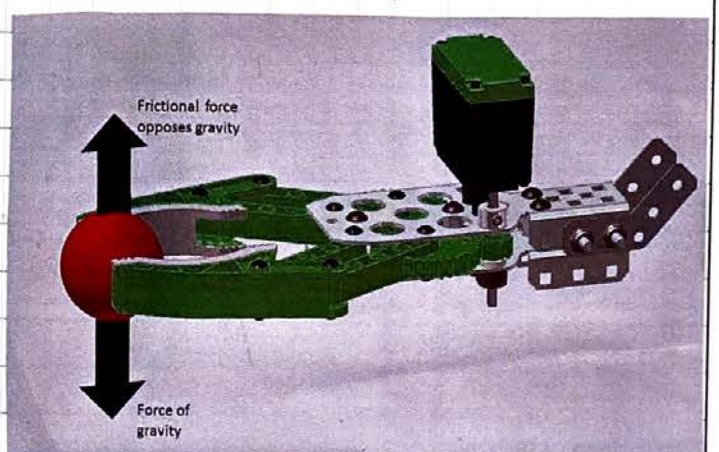
OUR DECISION: CLAW

The claw offers us with a simple intake solution to grasp the caps. The claw is relatively easy to construct and we believe it can perform the job of flipping, grasping, and setting the caps onto the post. We also have experience with building and programming claw intakes and we think the claw offers a good starting point to begin the season and to see even more aspects that need to be fulfilled ~~that~~ by an intake in Turning Point. Since we started out last season with claw intakes, we also are comfortable driving with claws. We believe that with friction padding lining the inside of the claw, the claw will be able to securely grab onto the cylindrical protrusions of the caps. We can also program the claw to rotate by first building a rotating joint so that the claw can grab the caps and turn them over to the correct alliance color. Like last year's claw, we can construct our claw for Turning Point to grasp and release objects with a wide distance. That is, unlike the basic claw bot claw, we can create our claw to open up wider in order to have greater accuracy in grabbing the caps.

In theory, the two sides of the claw provides applied forces to the cap while the normal force of the cap is exerted back on the sides of the claw. At the same time, the frictional force between the cap and claw is directed upwards since it opposes its weight, which is defined by mass x gravity. With such balanced forces, we can be sure that the cap is gripped tightly enough to be held securely by the claw.



Though a ball is depicted in the claw to the right + below, we believe the same forces can be applied to the cap.



SIGNATURE [signature]

DATE 5/17/18

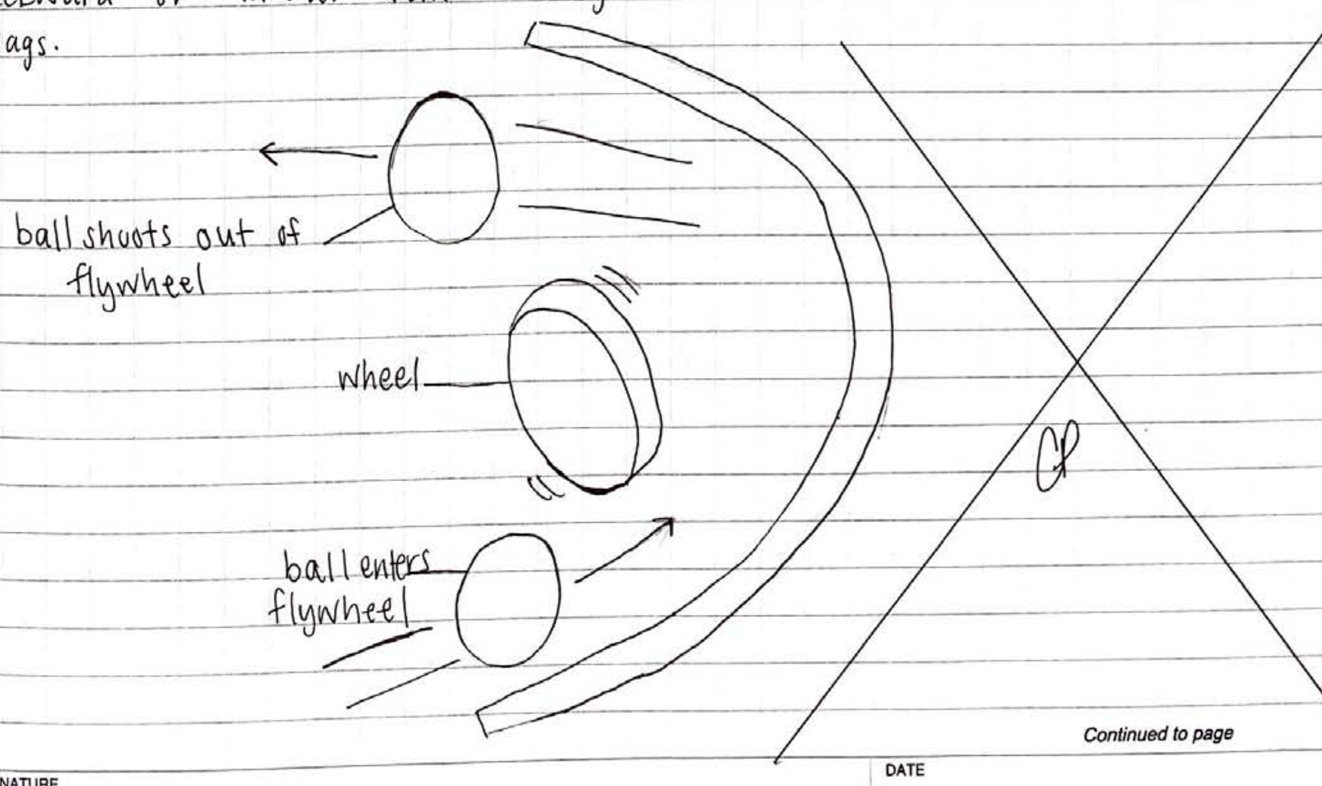
DISCLOSED TO AND UNDERSTOOD BY Mylissa

DATE 5/17/18

Continued from page

CHOOSING THE BALL LAUNCHER

From the moment we saw the game reveal for the 2018-2019 season, we knew and understood that our robot would have to have some sort of contraption/design to shoot balls to toggle the flags. We have seen from previous years' games that called for shooting balls of robots that were constructed around a type of ball launcher. That being said, there really is one design that was and is popular and efficient in shooting balls: the flywheel. We need a ball launcher that can shoot balls at varying heights and projections in order to toggle the top, middle, and bottom flags. Thus, the flywheel design allows our robot to accomplish just that. The flywheel design consists of two wheels spinning in opposite directions close together, and, when the ball goes in between them, it launches the ball. We can build a flywheel that uses a single wheel spinning outwards as well. Attached to the flywheel mechanism would be a sort of ramp to direct the ~~wall~~ ball towards the flags to be toggled. With this design, it is easy for us to launch the balls at different heights. That is, we can move forward or closer to the flags to shoot higher while moving backward or further from the flags will allow us to hit the lower flags.



Continued to page

SIGNATURE

DATE 6/7/18

DISCLOSED TO AND UNDERSTOOD BY Mufassa

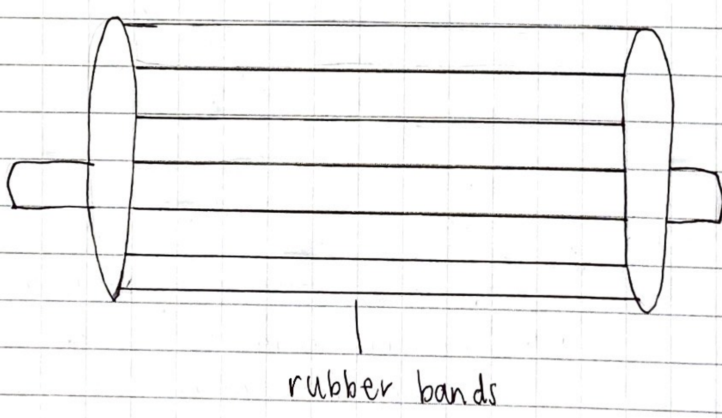
DATE 6/7/18

PROPRIETARY INFORMATION

Continued from page

CHOOSING THE BALL INTAKE

Similar to last year's Goliath intake for the cones, the intake for this year's competition will be one rubber band roller. This rubber band roller will guide the balls into the robot and eventually lead the ball into the flywheel. The balls will be directed into the robot through a ramp to help push the ball to the loading zone and prepare for it to shoot. This intake is the most efficient way to pick up the balls off of the field because the roller will be continuously running, which allows the driver to drive over the ball and prepare it for launch. That is, all the robot/driver has to do is drive over the ball to acquire the ball in an efficient and labor-saving way. Unlike our rubber band roller intake from last season, we will only have one roller since we do not need to roll in an object in an object in order to hold it. The roller intake is significant in Turning Point as well as our plans for our robot design because we are launching the balls and need to devise a solution to guide the balls into the robot where the launcher resides. We also believe the roller intake for taking balls in will be very popular and competitive in competitions. This being said, we believe the goliath will be especially competitive as the season progresses for the reason that robots can consistently load balls into their launchers to project balls to the flags at a seemingly constant rate.



To the left is a rough sketch of what the roller intake will look like. We will have rubber bands going from one side to the other to grip the ball into the robot for launching.

SIGNATURE

Continued to page

DATE 6/7/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 6/7/18

PROPRIETARY INFORMATION

Continued from page

TORQUE COMPARISON

VEX 393 Motors vs. V5 Motors

	V5 Smart Motor Specifications	2-Wire Motor 393 + Motor Controller 29
Speed	Approximately 100, 200 or 600 RPM	120, 160, or 240 RPM
Peak Power	11 W	3.93 W
Continuous Power	11 W	2.70 W
Stall Torque	2.1 Nm	1.67 Nm
Low Battery Performance	100% Power Output	51% Power Output
Feedback	Position Velocity (calculated) Current Voltage Power Torque (calculated) Efficiency (calculated) Temperature	Position[1]
Encoder	1800 ticks/rev with 36:1 gears 900 ticks/rev with 18:1 gears 300 ticks/rev with 6:1 gears	627 ticks/rev with High Torque gears[1] 392 ticks/rev with High Speed gears[1] 261 ticks/rev with Turbo gears[1]
Dimensions	2.26" W x 2.82" L x 1.30" H (57.3 mm W x 71.6 mm L x 33.0 mm H)	1.97" W x 2.16" L x 0.98" H (50 mm W x 55 mm L x 25 mm H)
Weight	0.342 lbs (155 grams)	0.209 lbs (95 grams)

2 Wire Motor 393

Free Speed = 100 rpm / 160 rpm (High Speed Option)

Stall Torque = 1.67 N-m / 1.04 N-m (High Speed)

Stall Current = 4.8 A

Free Current = 0.37A

Stall Torque (N-m): amount of load placed on motor that will cause it to stop moving

Free Speed (RPM): maximum rotational speed a motor will run at when it is under no load

Stall Current (Amp): the amount of current a motor will draw when stalled

Free Current (Amp): amount of current a motor will draw when under no load

Motors have limits. At some point, the force of external objects will be too much for the motor's electrical windings to handle, and it will fail. Specifically, the maximum weight the robot can hold stationary occurs at the stall torque of the motor. In the calculations that follow, we estimated the maximum force one 2 wire 393 motor and one V5 motor can handle respectively.

How much torque is needed to lift caps?

Cap weight = 0.335 kg

$(0.335\ kg)(9.81\ m/s^2) = 3.286\ N$

$(0.59055\ m)(3.286N)$ = Torque load = 1.941N

$$\text{Torque load} = \frac{(\text{current} - \text{free current})(\text{stall torque})}{(\text{Stall current} - \text{free current})}$$

$$= \frac{(1 - 0.37)\cancel{(0.37)}(1.67)}{(4.8 - 0.37)}$$

$$= 0.237\ N \cdot m$$

23.25 in. = 0.591 m = length of arm

Stall Torque = Max. Force x Arm Distance

Max. Force = Stall Torque / Arm Distance

Force = 1.67 / 0.591 = 2.828 N

Number of motors required: $\frac{3.286N}{2.828N} = 1.162$ motors

* One 2 wire ~~M~~ 393 motor can sustain a maximum of 2.828 N of force. One cap alone has a force of 3.287 N, meaning more than one 393 motor will be needed.

Continued to page

DATE 8/29/18

SIGNATURE

DATE 8/29/18

DISCLOSED TO AND UNDERSTOOD BY

TORQUE COMPARISON (CNTD.)

V5 Motor

Free speed: 200 rpm

Stall Torque: 2.1 N'm

Stall Current: 2.5 Amps

Free current: 0.8594 Amps

CP

Ohm's Law states that the potential difference (voltage) across an ideal conductor is proportional to the current through it. Resistance is the constant of proportionality, (R). Ohm's law is given by $V = I \times R$, and we are using it to calculate the V5 motor current.

Power = Voltage x Current

11 watts = (12.8 volt) (current)

Current = 0.8594 Amps

$$\text{Torque load} = \frac{(2\text{Amps} - 0.8594\text{Amps})(2.1\text{ N'm})}{(2.5\text{ Amps} - 0.8594\text{ Amps})}$$

= 1.459 N'm

Max. Force = Stall Torque / Arm Distance

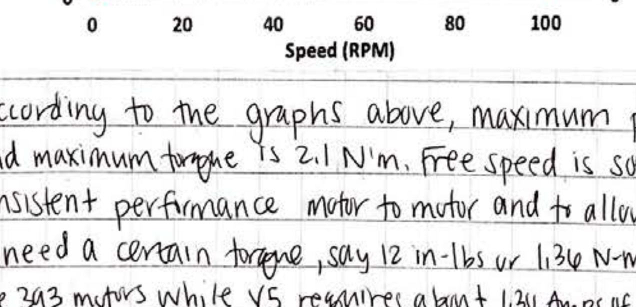
= 2.1 N'm / 0.59055m = 3.556 N

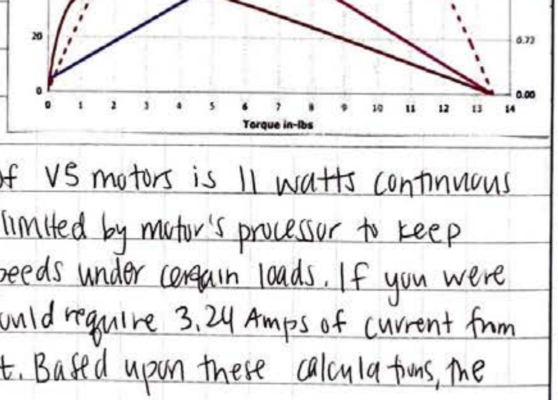
Number of motors required: 3.286N / 3.556N = 0.9242 motors

*One V5 motor can sustain a maximum force of 3.556 N. One cap alone has a force of 3.286 N, so one V5 motor is sufficient to withstand the force of one cap.

In comparison to the 2 wire 393 motors from last year, the V5 motors can handle significantly more torque load.

Torque speed graph of v5 motors





According to the graphs above, maximum power of V5 motors is 11 watts continuous and maximum torque is 2.1 N'm. Free speed is software limited by motor's processor to keep consistent performance motor to motor and to allow top speeds under certain loads. If you were to need a certain torque, say 12 in-lbs or 1.36 N-m, it would require 3.24 Amps of current from the 393 motors while V5 requires about 1.30 Amps of current. Based upon these calculations, the V5 motors are significantly more powerful.

Continued to page

SIGNATURE

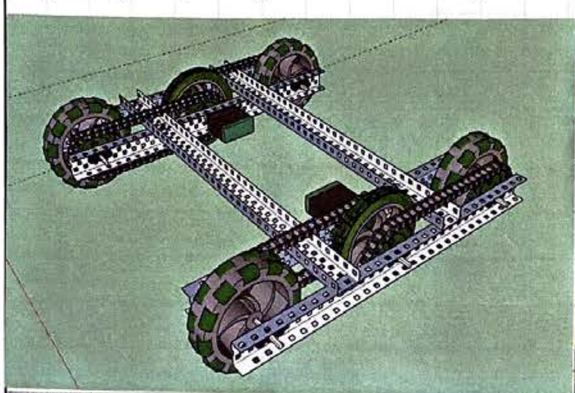
DATE 8/29/18

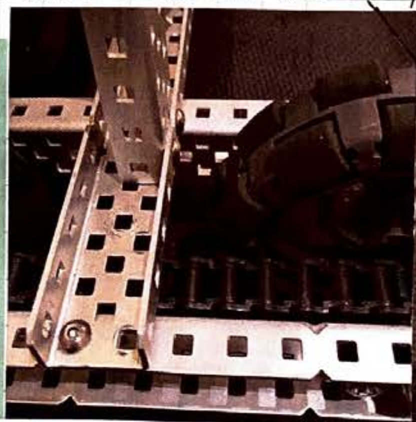
DISCLOSED TO AND UNDERSTOOD BY

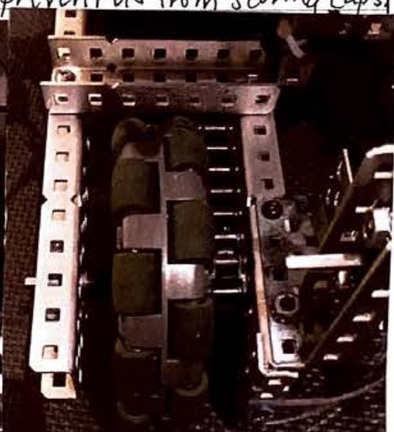
DATE 8-29-18

BUILDING THE DRIVE

We are constructing the drive with 1x2x1x30 c-channels and running a 1x2x1x35 c-channel across the center for stability. This structure will ensure that the robot meets the 18" requirements. We are using omni-wheels to make sure the robot can make quick turns for speed. These wheels allow for smooth rotations since they have very little friction, providing greater efficiency for the robot. On the other hand, the c-channels have strong properties, which prevents any dents or bending. Because the c-channels can be cut into smaller pieces, we cut the c-channels and attached the smaller pieces to the drive where we could not fit a full length c-channel across the drive for stabilization. We put these pieces between the omni-wheels and friction wheels since these locations lacked stability. As a result, our robot can stand sturdy without tipping or falling over. This is a paramount aspect to cover since Turning Point will involve lots of pushing, especially for parking. Since we are only allowed to use a maximum of 8 motors, the two motor drive will allow us to complete the rest of the tasks while still being quick and powerful. The drive is chained so that the power is transferred into all 3 wheels ~~w~~ on each side. The 3 wheels are the 2 omni-wheels and the 1 friction wheel. The drawing below shows a friction wheel in the middle that will prevent other teams from pushing us around. This game can be very defensive, and since we hope to build a scoring robot, we must protect ourselves from others trying to push us off the platforms (so we can achieve the six point advantage by high paring) or change the direction of our drive to prevent us from scoring caps.



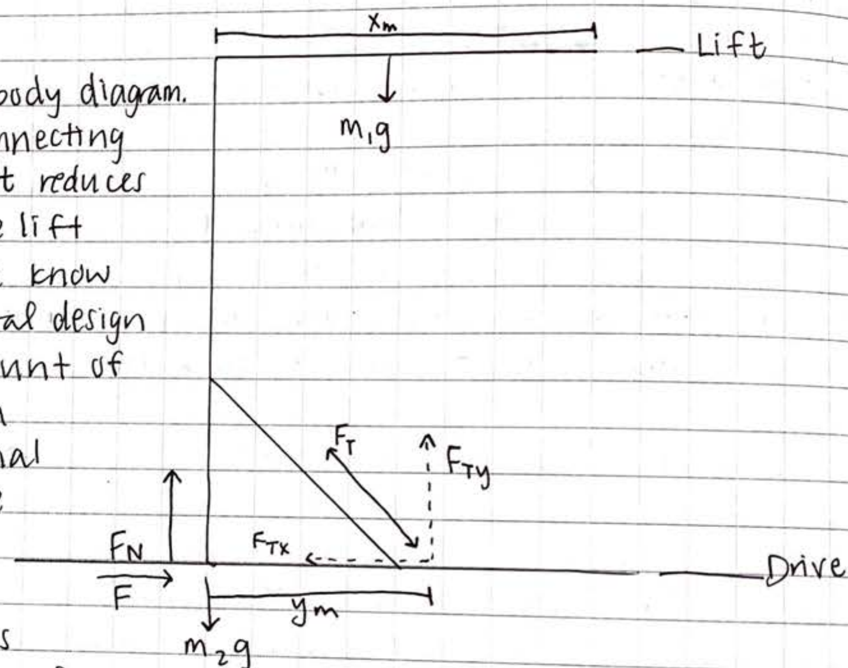




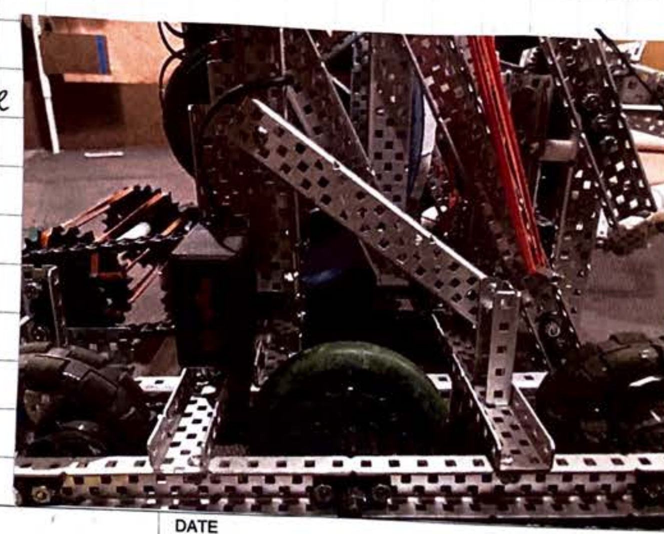
BUILDING THE DRIVE (CNTD.)

In addition to the structure of the drive, we have a triangle structure for added stability. The hypotenuse of the triangle connects the lift to the base of the robot, which helps to stabilize the overall structure of the robot.

To the right shows the free body diagram. Since the triangular structure connecting the drive to the base of the lift reduces the force of the weight of the lift exerted in the y-direction, we know that the current architectural design will provide the greatest amount of stability. If there was not a triangular structure, the normal force at the point where the base of the lift connects to the drive would have to be much greater because it has to equal the total weight force of the lift. By adding this shortened c-channel diagnally to form a triangle, a tension force with a component force in the y-axis direction helps to bear some of the weight force from the lift, thus evenly distributing the force better throughout the drive, creating structural stability. This triangle structure also provides a tension force with an x-component force acting against the forward force, which adds more security to the design.



$$\sum F_x: F_{Tx} = F$$
$$\sum F_y: m_2g + m_1g = F_{Ty} + F_N$$



SIGNATURE

DATE 9/11/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 9-11-18

PROPRIETARY INFORMATION

Continued from page

GEAR REDUCTION SPECS

Gear reduction occurs when the drive gear is smaller or has fewer teeth than the driven gear. We want to calculate the gear reduction ratio to get an understanding of what gear ratio will allow us to achieve a certain speed.

$\text{Circumference}_{wheel} = \pi(\text{diameter})$ diameter = 4 inches

4 inches

$= \pi(25.4 mm)$

$= 79.796 mm(4)$

$= 319,184 mm$

The robot moves 319,184 mm per 1 wheel revolution.

wheel is moving at 200 revolutions per 60 seconds

Linear ground speed $= \frac{319,184 mm}{60 sec} = 5.3197 mm/s$

CP

Gear reduction required:

$900 \frac{mm}{s} \times \frac{1 \text{ revolution}}{319,184 mm} = 2.8197 \text{ rev/sec}$

$2.8197 \frac{rev}{sec} \times \frac{60 sec}{min} = 169.181 \text{ rev/min}$

Gear reduction required $= \frac{200 RPM}{169,181 RPM} = 1.182$

Based on the above calculations, our team needs to utilize a gear reduction of 1.182 or less in order to achieve a top speed greater than 900 mm/s.

CP

Continued to page

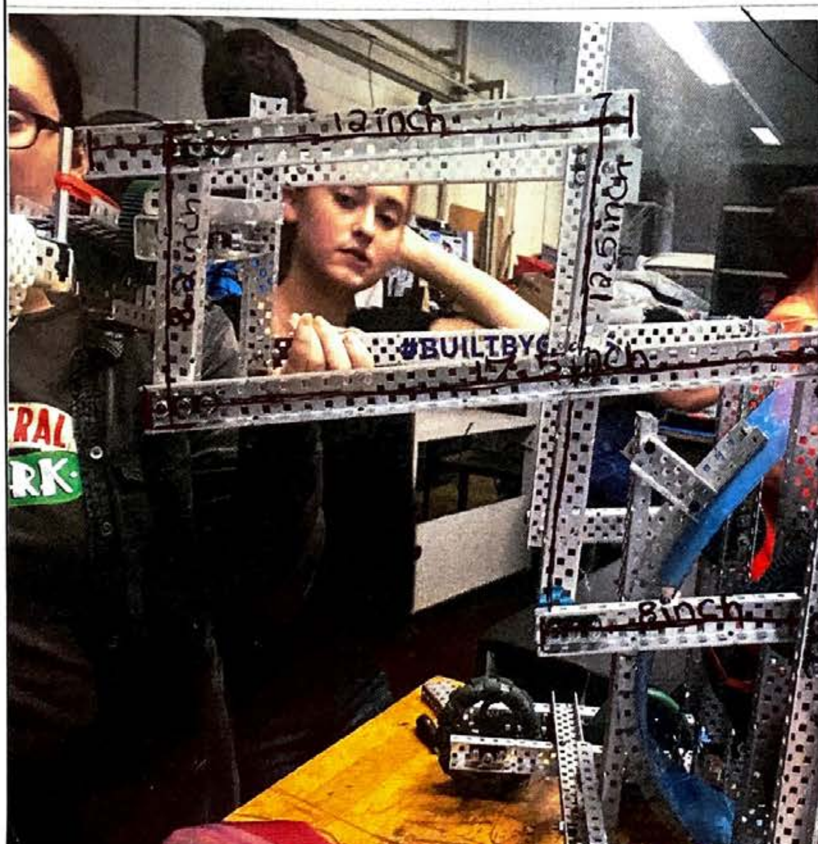
SIGNATURE [signature]   DATE 9/13/18

DISCLOSED TO AND UNDERSTOOD BY Caylin Dorgue   DATE 9-13-18

PROPRIETARY INFORMATION

BUILDING THE LIFT

A six-bar linkage is essentially two four bar linkages super-imposed on top of each other. The six-bar lift is ideal for meeting our goals for Turning Point. That is, while other lifts extend only in one direction, the six bar lift extends both vertically and horizontally. We attached the lift to the drive by using L channels. After learning that any vibrations cause the hex nuts to turn loose, we chose to use lock nuts for these structural purposes. The gears were screwed to the metal, and the circular inserts were inserted into the gears. This permits the arm of lift to raise and move downwards while the gear turns. This motion and movement is critical since it will help us score the caps on the posts. We used 12 aluminum c-channels to construct the six-bar lift, in which we had three equivalent links. Linkages consist of a series of rigid bodies called links, connected together by freely rotating joints and placed parallel to the other.



As we finalize our design, we plan on cutting/trimming the lift to meet the 18X18X18" size requirements.

Continued to page

SIGNATURE [signature] DATE 9/20/18

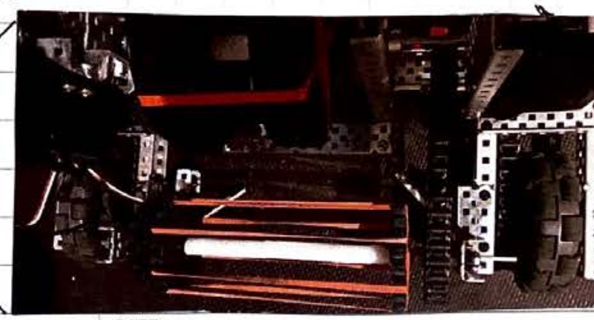
DISCLOSED TO AND UNDERSTOOD BY Caylin Payne DATE 9-20-18

BUILDING THE BALL INTAKE

Similar to last year's Goliath intake for cones, the intake for this year's competition will be one rubber band roller to pick up the balls into the robot. In order to ~~push~~ direct the balls into the robot, there will be a ramp to help push the ball to the loading zone for shooting preparation. This intake is the most efficient way to pick up the balls off of the field because it will be continuously running, which allows the driver to simply drive over the ball and prepare it to launch. Additionally, we do not need to be as accurate when trying to acquire the ball because of the nature of the roller. In other words, the length of the roller is generally long enough to the extent that we do not need to worry about being precise in setting the robot to the right location to get the ball in the robot. All we have to do is drive the robot near the ball, or at least slightly touch the ball to get the ball into the ball launcher.

To construct the ball intake, we took two 24 tooth sprockets and inserted spacers in between the two sprockets to set it to the correct length. We then put rubber bands across the two sprockets horizontally to grip the ball when it enters the robot. The rubber bands were assembled in this horizontal manner and spaced out at every third tooth, to turn the ball at a consistent rate on the plexiglass ramp. This prevents the ball from launching into the flywheel without the command from the controller. We decided on a constant roller speed of 45 rpm. In order to get the ball into the flywheel, we added a button that prompts the roller intake to speed up to a higher rpm to force the ball up from the ramp. The flexibility of the plexiglass aids the ball by bouncing it up into the flywheel when the rpm increases to a higher rate.

CP



SIGNATURE

DATE 9/25/18

DISCLOSED TO AND UNDERSTOOD BY

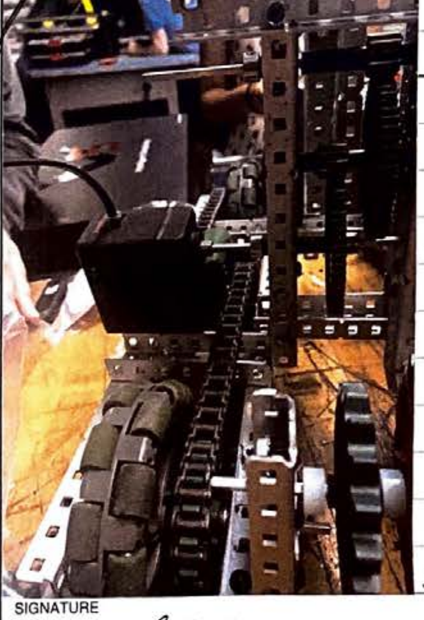
DATE 9-25-18

PROPRIETARY INFORMATION

BUILDING THE BALL LAUNCHER

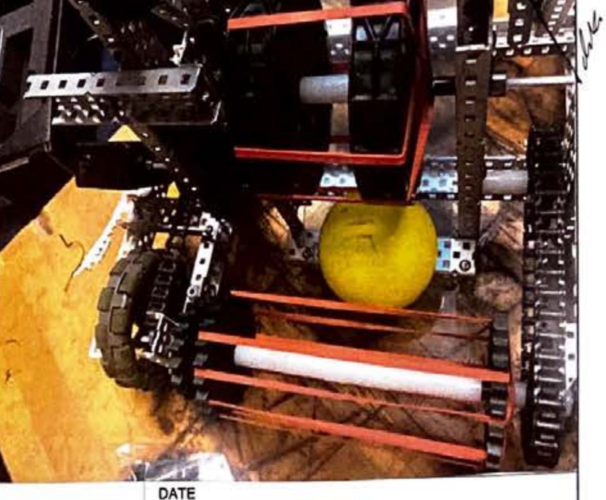
To have a smooth transition from the ~~inka~~ intake to the launch, we are going to use the flywheel design. The flywheel allows the ball to shoot at the same angle each time, and we can easily change the height of the shot by changing the robot's position on the field.

We constructed our flywheel with two wheels with spacers in between so that the ball can be guided and forced up the launcher. Next, to outline the base of the base of the flywheel, we used a 1x5x1x25 aluminum c-channel. We chose aluminum because it is lighter than steel, and there is no need for our robot's flywheel to possess so much weight. For every two holes in the c-channel, we cut slits in the metal with a Dremel which weakens the metal so that we can bend the metal in order to add the needed curvature to our ball launcher. We believe this is one of the most unique features of our robot because in our club for robotics, most teams utilized the plexiglass cut-out to construct the ball launcher. However, we wanted to find a way that is more efficient in directing the ball at an angle geared most towards the setup of our robot. By being able to bend our plexiglass and c-channel to our desired angle, we are able to optimize launching the balls directly to the flags. We also have rubber bands on the two wheels of the flywheel to help shoot the ball up the launcher because of their elasticity.



Gear ratio is 1:25

CP



SIGNATURE

DATE 9/27/18

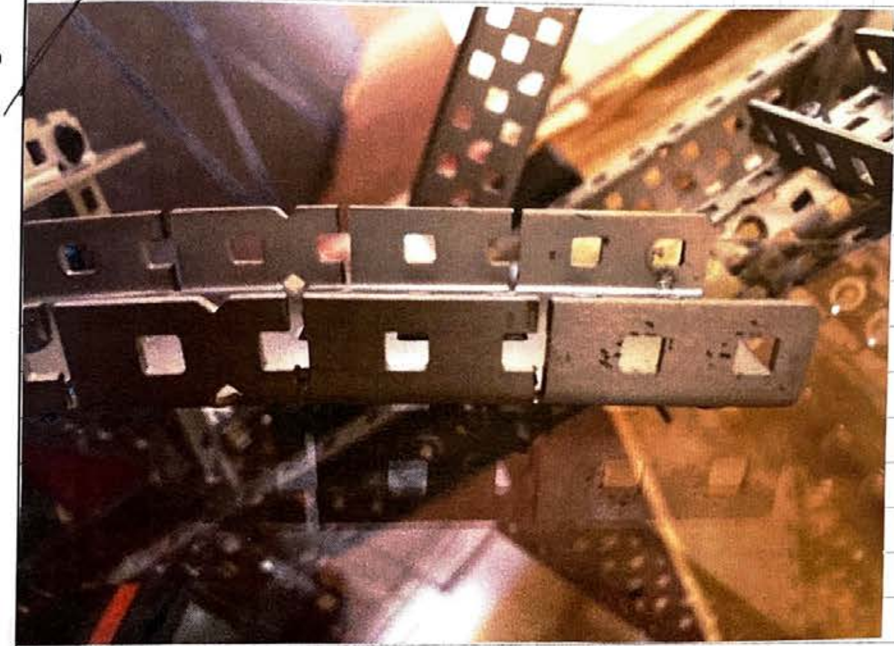
DISCLOSED TO AND UNDERSTOOD BY Caylin Payne

DATE 9-27-18

PROPRIETARY INFORMATION

Continued from page

BUILDING THE BALL LAUNCHER (CNTD.)



As seen in the photo to the left, we made incisions to the c-channel that serves as the spine of the flywheel. As a result, we can bend the c-channel and get it to curve to the angle needed to direct the ball out towards the flag for shooting. We chose to cut after 2 holes because one hole would make it too flexible while 3 is slightly rigid for us.

With the flywheel completed to the right, one can get an idea of how we assembled the flywheel with the curved c-channel with the plexiglass. The plexiglass allows the ball to travel through the ball launcher on a smooth surface, which allows for a seamless transition path for launching. We decided upon a 55° angle for our flywheel, because after testing running multiple trials at a distance of around 2.5 feet away from the post and a wheel speed of 200 rpm, we recognized this as the most optimal angle to aim the balls. Our flywheel is powered by 2 motors and some rubber bands. We decided upon a double wheel launcher with dual flywheels set beside eachother, spinning at a target of 200 rpms. The housing of the flywheels is stood horizontally.

SIGNATURE

DATE 9/27/18

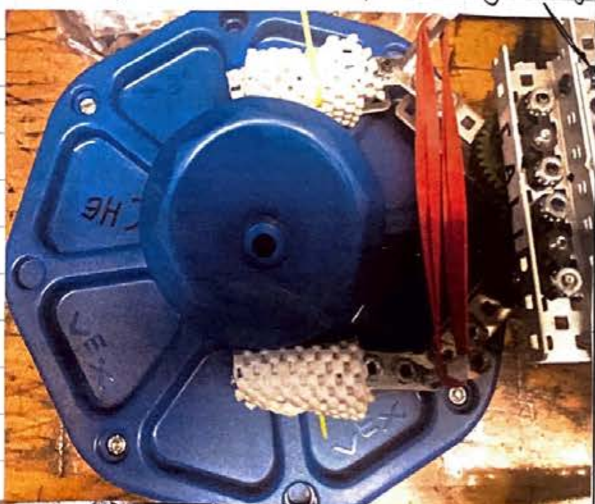
DISCLOSED TO AND UNDERSTOOD BY Cayetano Raya

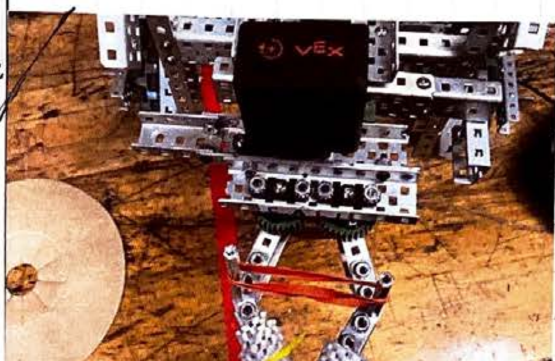
DATE 9-27-18

Continued from page

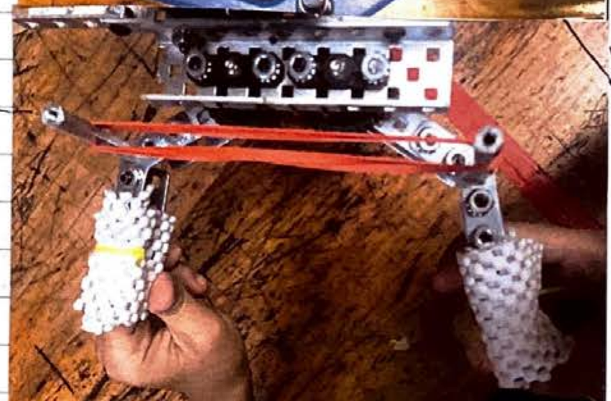
BUILDING THE CAP INTAKE

Upon brainstorming intake types, we deduced that the claw offers a simple solution and tactic to acquire the caps. We envision that the claw will grab onto the cylindrical protrusions of the caps in order to grab onto them. To construct the claw, we cut c-channels into 4 small segments – 2 for each side of the claw. We connected the 2 segments using a gusset for each side. To avoid the cap from slipping from the grip of the claw, we must add a frictional factor. Thus, we added friction padding to both ends of the claw and secured them with zip ties. We also added rubber bands to the claw to strengthen the grip of the claw on the cap, so we can be sure that the cap will not slip out of the claw's grasp. Since the claw will be grabbing onto the ~~cly~~ cylindrical protrusions on top of the caps, we cannot have the rubber bands directly on the claw from the right to the left sides of the claw. If this was the case, then the protrusions we plan on grabbing would be obstructed by the rubber bands. As a result, we placed stand offs on each side of the claw in order to still place the rubber bands on the claw for added strength, but in a more favorable location, without blocking the cylindrical protrusions. The stand offs offer the space needed between the cap and the claw when we grab onto the caps, and the rubber bands are wrapped around the two standoffs.





Standoffs



SIGNATURE

DATE 9/29/18

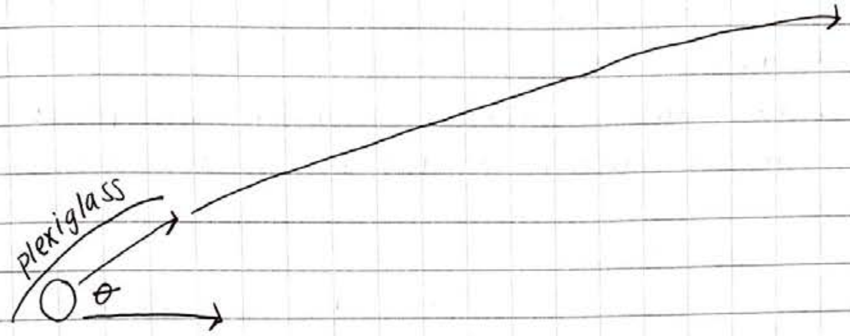
DISCLOSED TO AND UNDERSTOOD BY

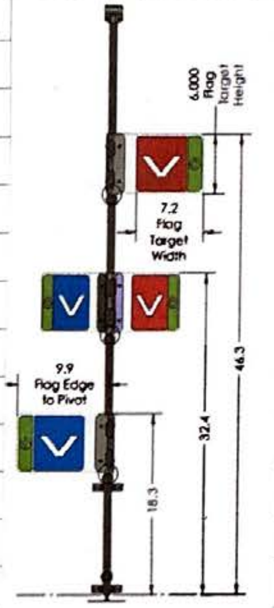
DATE 9-29-18

PROPRIETARY INFORMATION

Continued from page

FLIGHT EQUATIONS





θ estimate ≈ 55° to 60°

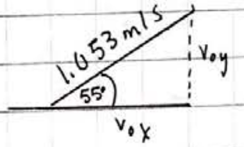
Flywheel motors travel at 198 rpm.

Radius of 4" wheel - (276-1497) = 2 inches

Circumference of wheel = $2\pi r = 4\pi = 12.566$ in.

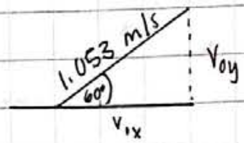
$$198 \text{ rpm} \times \frac{12.566 \text{ in.}}{1 \text{ rotation}} \times \frac{0.0254 \text{ m}}{\text{in.}} \times \frac{\text{min.}}{60 \text{ sec}} = 1.053 \text{ m/s}$$

The wheels have a linear speed of 1.053 m/s. This translates to an initial velocity of the ball of 1.053 m/s.



$V_{ox} = 1.053 \cos 55°$
$= 0.604$ m/s

$V_{oy} = 1.053 \sin 55°$
$= 0.863$ m/s



$V_{ox} = 1.053 \cos 60°$
$= 0.527$ m/s

$V_{oy} = 1.053 \sin 60°$
$= 0.912$ m/s

Goal = Find range to hit center of top flag

$\Delta xy = 31.5$ in. $= 0.8001$ m

$31.5 \text{ in.} \times \frac{0.0254 \text{ m}}{\text{in.}} = 0.8001 \text{ m}$

$V_{Fy}^2 = V_{oy}^2 + 2a\Delta xy$

$V_{Fy}^2 = (0.863)^2 + 2(-9.81)(0.8001)$

$V_{Fy}^2 = 14.953$

$V_{Fy} = 3.867$ m/s

Continued to page

SIGNATURE [signature] DATE 10/1/18

DISCLOSED TO AND UNDERSTOOD BY Caylee Payne DATE 10-1-18

FLIGHT EQUATIONS (CNTD)

$$0.8001\ m = \left(\frac{0.863\ m/s + 3.867\ m/s}{2}\right) t$$

$t = 0.338$ seconds

In order to reach the vertical height of 31.5 inches above the point of release of the flywheel, the ball must be in the air for 0.338 seconds.

Solving for horizontal range:

$\Delta x_x = (0.604\ m/s)(0.338s)$

$= 0.204\ m$

CP

One of the objectives of the game is to position our robot to hit one of the 3 flags at the edge of the field. Due to the angle of our flywheel shooter and the fact that we are using a six-bar lift, as opposed to something shorter, it is unlikely to aim for the bottom-most flag. So, we will aim at the middle and top flags. In order to ensure that we consistently hit the center of the flags, we strive to position our robot in the same general spot on the field. Based on our measurements, the distance from the bottom of the field to the center of the middle flag is 29.5 in. and the distance from the bottom of the field to the center of the top flag is 43 in. The distance from the point of release of the flywheel to the center of the middle flag is 18 in. and to the third flag is 31.5 in.

According to our calculations, the ideal horizontal position for the point of release is 0.609 meters, which is 1.998 feet, away from the base of the pole with the flags. When taking into account the fact that the position of the flywheel is not on the edge of the robot, we can add an additional 10 inches to that measurement. When we tested out this value, our robot was only slightly off. This calculation is a very rough estimate of where our robot should be prior to launching. That being said, there are still other factors such as topspin/backspin as well as air drag that we need to account for.

CP

Continued to page

SIGNATURE [signature] DATE 10/1/18

DISCLOSED TO AND UNDERSTOOD BY Caylis Payne DATE 10-1-18

PROPRIETARY INFORMATION

Continued from page

PROGRAMMING USER CONTROL

We began programming the robot using Vex Coding Studio (VCS) since it was designed to be easy to use for V5 robots. VCS allowed us to quickly and more efficiently program the robot, as we currently don't have complex sensory functions to code.

```
while (1){

    // DRIVE CONTROL
    LeftDrive.spin(vex::directionType::fwd, Controller1.Axis2.value()/1.3, vex::velocityUnits::pct);
    RightDrive.spin(vex::directionType::fwd, Controller1.Axis3.value()/1.3, vex::velocityUnits::pct);

    // LIFT CONTROL
    // up
    if(Controller1.ButtonR1.pressing()){
        RightLift.spin(directionType::fwd,200,velocityUnits::rpm);
        LeftLift.spin(directionType::fwd,200,velocityUnits::rpm);
    }
    // down
    else if(Controller1.ButtonR2.pressing()){
        RightLift.spin(directionType::rev,200,velocityUnits::rpm);  //down
        LeftLift.spin(directionType::rev,200,velocityUnits::rpm);
    }
    // brake when not pressed
    else{
       RightLift.stop(vex::brakeType::brake);
       LeftLift.stop(vex::brakeType::brake);

    }
```

→ programmed through if-else statements

R1 button pressed → left + right lift motors spin forward → lift moves up.

R2 button pressed → left + right motors spin in reverse → lift moves down

neither button pressed → lift motors do not move

The drive is programmed so the forward motions of the left + right drives are controlled by the right + left joysticks, respectively. When the joysticks are moved along their axes, a value is transmitted to the robot telling it how to move. This value is divided by 1.3 (determined experimentally) in order to make the movement smoother + allow for easier driving.

```
//FLYWHEEL
  // get to the max speed
  if (Controller1.ButtonRight.pressing()){
      Shooter.spin(directionType::fwd,200,velocityUnits::rpm);

  }
  // operate at 25 rpm
  else {
      Shooter.spin(directionType::fwd,25,velocityUnits::rpm);

  }

 // BALL INTAKE
 // turn at max speed
 if(Controller1.ButtonLeft.pressing()){
     BallIntake.spin(directionType::rev,200,velocityUnits::rpm);

 }
 // operate at 40 rpm
 else{
     BallIntake.spin(directionType::rev,40,velocityUnits::rpm);
 }
```

In contrast to the lift code, the flywheel & ball intake code program those motors to always operate. If the right button is pressed, the flywheel motor (named "Shooter" in our code) operates at maximum speed. If it is not being pressed, it operates slower at only 12.5% of the maximum speed. The ball intake is programmed similarly, its motor named "Ball Intake" operating at maximum speed when the left button is being pressed and only 20% of max speed if it isn't.

Continued to page

SIGNATURE [signature]

DATE 10/1/18

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 10-1-18

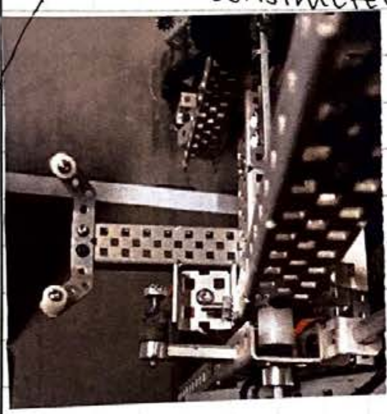
PROBLEM + SOLUTION

Problem: Robot has trouble moving onto parking platforms.

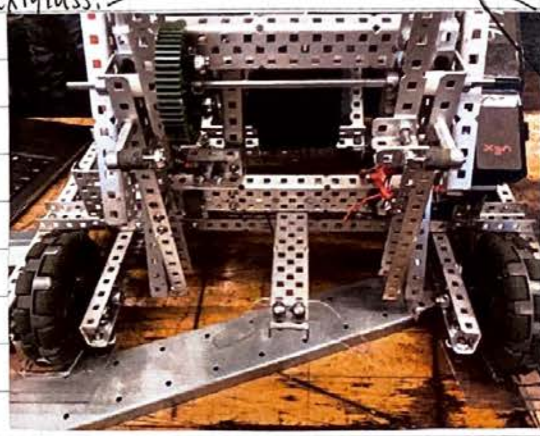
Solution: We added plexiglass on the drive near the wheels. This helps the robot to push up onto the platforms.

Problem: It is difficult for the robot to precisely line up with the posts to stack the caps.

Solution: We created a guide for the robot to help it line up with the posts. We created a trial design made of metal, and since it worked well, we constructed the final design, made of plexiglass.

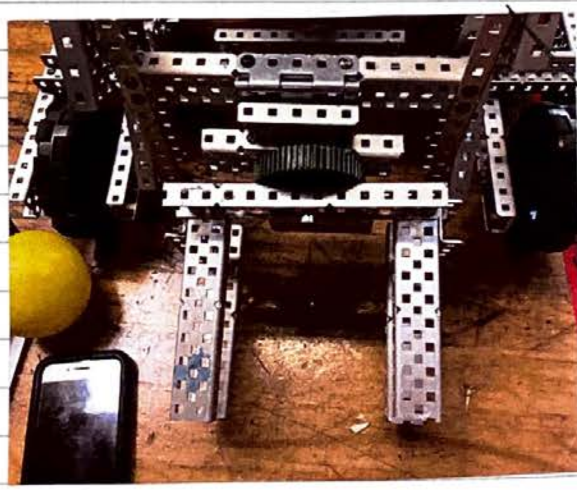


initial → final



Problem: The claw intake is not efficient in picking up the caps. Our driver takes a long time trying to precisely get hold of the top cylindrical protrusions.

Solution: We built a new cap intake from our brainstorming page: the "waffle" intake. With this intake, we can scoop up the caps from its sides, rather than from the tops. The waffle ~~flip~~ intake is shown to the right.



Continued to page

SIGNATURE

DATE 10/2/18

DISCLOSED TO AND UNDERSTOOD BY

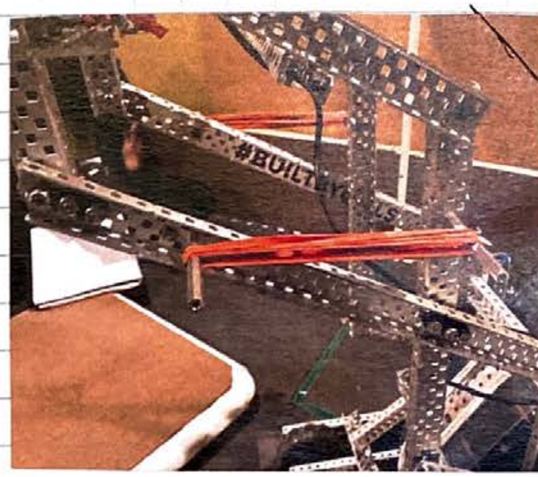
DATE 10-2-18

PROPRIETARY INFORMATION

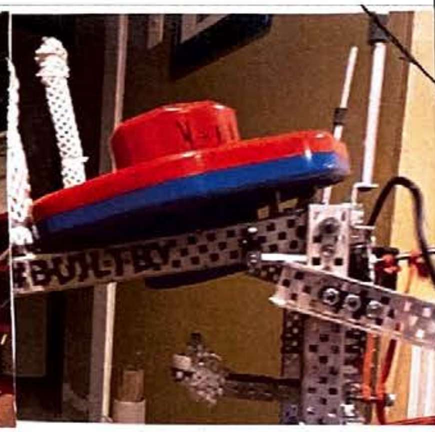
PROBLEM + SOLUTION (CNTD.)

Problem: There is not enough room around the omni-wheels to add a motor for the ball intake.
Solution: We used chain to connect the roller intake with the flywheel.

Problem: The lift takes a long time to move up—the speed is not as fast as we like.
Solution: We added rubber bands to the lift to help the lift go up faster. The rubber bands help due to their elasticity.



Problem: Our new waffle intake is still not the intake best suited for our purposes. After using this intake, we saw that it was difficult to stack on higher poles and hard to create a guide for cap placement.
Solution: As seen on the right, the L-shaped intake allows the drive to go straight into the cap and secure it between the 2 bottom parts of the L. Then, when the cap is flipped up, the color changes to the correct team color, which can stack on the higher poles.



Problem: Ball would get stuck in the sprocket, which causes the robot to be immoble.
Solution: We added plexiglass and curved it to the drive to steer the ball away from the sprockets.

SIGNATURE

DATE 10/5/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 10-5-18

PROPRIETARY INFORMATION

Continued from page

UPDATED USER CONTROL CODE

With the addition of the L-intake to flip the caps, the user control code was updated to give the L1 and L2 buttons of the controller control over the L-intake.

```
//CAP INTAKE CONTROL
// foward/up
if(Controller1.ButtonL2.pressing()){
   Cap.spin(directionType::fwd,300,velocityUnits::rpm); // CAP FLIPPY DO forward


}
 // backward/down
 else  if(Controller1.ButtonL1.pressing()){
    Cap.spin(directionType::rev,300,velocityUnits::rpm); // CAP FLIPPY DO backward
}

// brake when not being pressed, this will help stand the weight of the cap
else{

    Cap.stop(brakeType::coast);

}

  vex::task::sleep(20); //Sleep the task for a short amount of time to prevent wasted resources.
 }
}
```

The ~~Claw~~ L-intake is programmed similarly to the lift - The first line tells the controller that when the button L2 is pressed, the motor controlling the L-intake should spin forward at maximum speed. The next else-if statement indicates that when button L1 is pressed, the L-intake motor should spin in reverse at maximum speed. If neither is pressed, the L-intake motor will not operate. These lines of code allow the cap intake to flip caps forward or backwards based on the button that is being pressed.

STRATEGY TRIALS

An important part of our game strategy includes getting off the platform, picking up a cap, and scoring it on the post, making it a High Scored Cap. At this point in the season, we aim to be able to do this in under 25 seconds, as this allows us to earn 2 points in less than 1/4 of the game. We conducted timed trials for this strategy to examine its effectiveness + our driving skill.

Trial	Time (seconds)	Success?	Stats
1	25.77	✓	$\bar{X} = 23.99$
2	26.95	X	$\Sigma X = 119.96$
3	21.46	✓	$S_x = 3.55$
4	19.03	✓	$IQR = 6.61$
5	26.75	✓	

SIGNATURE [signature]

Continued to page

DATE 10/5/18 - 10/17/18

DISCLOSED TO AND UNDERSTOOD BY Caylen Payne

DATE 10-5-18

Continued from page 32

STRATEGY TRIALS (CNTD)

Our data indicates that this strategy has an 80% success rate in an average of approximately 24 seconds with our current driving skill. This meets our goal of under 25 seconds, though the standard deviation of 3.55 seconds may indicate that we require more practice to be confident about the effectiveness of this strategy.

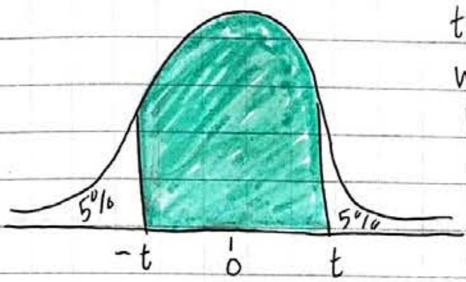
We used our trial data to construct a 90% confidence interval. We constructed a t-interval using a t-distribution, since our sample size is too small to use a z-distribution ($n<30$)

Formula for t-interval of the mean ($\bar{x}$):

$$\bar{x} \pm t\frac{s}{\sqrt{n}}$$

Where $\bar{x}$ is the sample mean, s is the standard deviation, and t is the t-distribution value for a 90% confidence interval

t-distribution ($n=5$)



t values were chosen for the t-distribution where $n=5$, so that 5% of the curve's area lies to each side of $-t$ and t, creating an area of 90% in the middle for the confidence interval.

After evaluating the confidence interval, we can be 90% confident that it will take us between 20.60 sec to 27.38 sec to complete this strategy, indicating that we are well on track to meeting our goal of under 25 seconds, but we required some more practice to confidently meet it.

CP

SIGNATURE lK

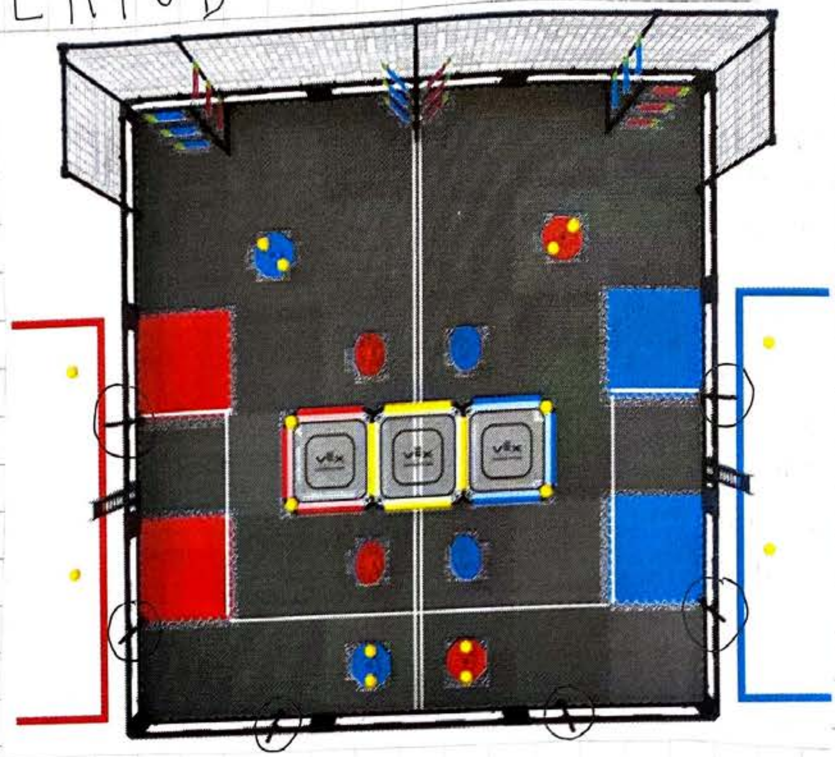
DATE 10/17/18

DISCLOSED TO AND UNDERSTOOD BY Carlos Payne

DATE 10-17-18

PROPRIETARY INFORMATION

DRIVE PERIOD STRATEGY



After watching many videos of competitions, we have realized that there are many robots that are able to quickly descore the low caps from the poles. For this reason, we are aiming for the high poles at the beginning of the competition, so that it takes the defensive robots a longer time to focus on the higher poles. During that time period, we will go to the lower poles and place the caps on those 4 posts. If the higher posts with caps are knocked off by the defense robots, we will quickly pick up the cap and place it on the top post so we can distract the opposing team. We hope to have an alliance partner that can focus on flags or caps, and if not, we will first stack on the high post and then focus on the flags. We will quickly hit the lowest flags with the "L" flipper out horizontally. Then, we will line up on the way back and hit the second flag. If there is time left in the match, we will either hit the top flags or play defense, depending on how the match is looking.

SIGNATURE

DATE 11/1/18

DISCLOSED TO AND UNDERSTOOD BY Caylee Dayne

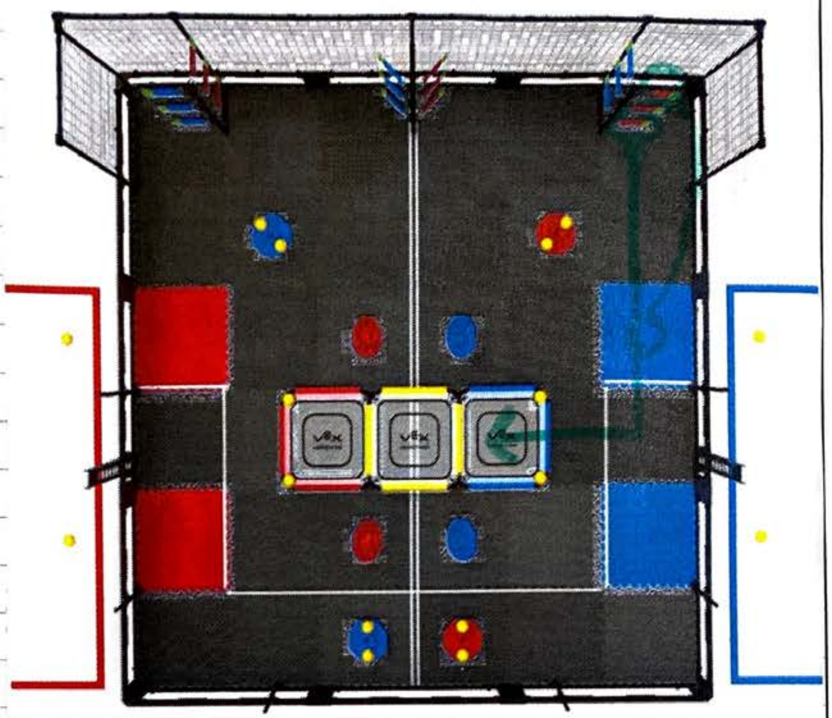
DATE 11-1-18

PROPRIETARY INFORMATION

Continued from page

AUTONOMOUS STRATEGY + CODE

We have programmed our robot for ~~a~~ the autonomous period to shoot a ball at the highest flag, move forward to hit the bottom flag, and then move backward to park on to the parking platform. For this auton, we used the "rotateFor" function on VEX Coding Studio, allowing us to get more precise and consistent movements as opposed to the fickle and battery-dependent movements characteristic of timed autons. The velocity of the robot stays at 100 rpm for most of the auton, but switches to close to 200 rpm near the end, since the extra speed is needed to get the robot onto the parking platform.



```
while(1) {
LeftDrive.setVelocity(100, velocityUnits::rpm);
RightDrive.setVelocity(100, velocityUnits::rpm);

 //shoots a ball
Shooter.spin(directionType::fwd,200,velocityUnits::rpm);
task::sleep(3000);
BallIntake.spin(directionType::rev,200,velocityUnits::rpm);
task::sleep(1000);
Shooter.stop(brakeType::coast);
BallIntake.stop(brakeType::coast);
task::sleep(250);


//moves forward and hits the flag (44 inches)
float revsToFlag = -3.351;
LeftDrive.startRotateFor(revsToFlag, revolutions);
RightDrive.rotateFor(revsToFlag, revolutions);
vex::task::sleep(250);

//moves backward (68 inches)
float revsFromFlag = 5.411;
LeftDrive.startRotateFor(revsFromFlag, revolutions);
RightDrive.rotateFor(revsFromFlag, revolutions);
vex::task::sleep(500);
```

The autonomous code shown to the left carries out the autonomous strategy on the blue side of the field. When on the red alliance, the same code can be used with one minor modification: the float variable revsTurn (which can be found highlighted on the next page) is made negative. This negative sign keeps the magnitude consistent, so the robot turns in the opposite direction to climb onto the red alliance's parking platform.

Continued to page

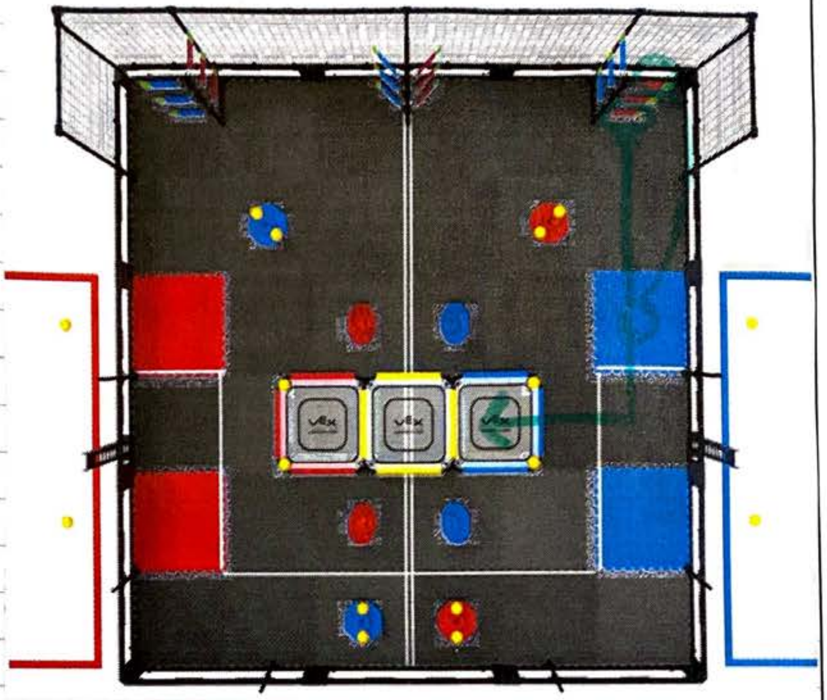
SIGNATURE [signature] DATE 11/15/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 11-15-18

PROPRIETARY INFORMATION

Continued from page

AUTONOMOUS STRATEGY + CODE



We have programmed our robot for ~~a~~ the autonomous period to shoot a ball at the highest flag, move forward to hit the bottom flag, and then move backward to park on to the parking platform. For this auton, we used the "rotateFor" function on VEX Coding Studio, allowing us to get more precise and consistent movements as opposed to the fickle and battery-dependent movements characteristic of timed autons. The velocity of the robot stays at 100 rpm for most of the auton, but switches to close to 200 rpm near the end, since the extra speed is needed to get the robot onto the parking platform.

```
while(1) {
LeftDrive.setVelocity(100, velocityUnits::rpm);
RightDrive.setVelocity(100, velocityUnits::rpm);

 //shoots a ball
Shooter.spin(directionType::fwd,200,velocityUnits::rpm);
task::sleep(3000);
BallIntake.spin(directionType::rev,200,velocityUnits::rpm);
task::sleep(1000);
Shooter.stop(brakeType::coast);
BallIntake.stop(brakeType::coast);
task::sleep(250);


//moves forward and hits the flag (44 inches)
float revsToFlag = -3.351;
LeftDrive.startRotateFor(revsToFlag, revolutions);
RightDrive.rotateFor(revsToFlag, revolutions);
vex::task::sleep(250);

//moves backward (68 inches)
float revsFromFlag = 5.411;
LeftDrive.startRotateFor(revsFromFlag, revolutions);
RightDrive.rotateFor(revsFromFlag, revolutions);
vex::task::sleep(500);
```

The autonomous code shown to the left carries out the autonomous strategy on the blue side of the field. When on the red alliance, the same code can be used with one minor modification: the float variable revsTurn (which can be found highlighted on the next page) is made negative. This negative sign keeps the magnitude consistent, so the robot turns in the opposite direction to climb onto the red alliance's parking platform.

Continued to page

Continued from page

AUTONOMOUS STRATEGY + CODE (CNTD.)

```
//right turn
//left forward right back
float revsTurn = 0.69;
LeftDrive.startRotateFor(-1*revsTurn, revolutions);
RightDrive.rotateFor(revsTurn, revolutions);
vex::task::sleep(1000);

LeftDrive.setVelocity(190, velocityUnits::pct);
RightDrive.setVelocity(200, velocityUnits::pct);


//drives onto platform (36 inches)
float revsToPlatform = 5.0;
LeftDrive.startRotateFor(revsToPlatform, revolutions);
RightDrive.rotateFor(revsToPlatform, revolutions);
vex::task::sleep(250);
```

The code shown is continued from the previous page. First, the drive velocity is set to 100 rpm at the beginning, and the flywheel and ball intake are activated to hit the top flag before being braked. Since the distances the ~~robots~~ robot needs to move in order to hit the flag and move backward is constant, we measured the two in inches and also measured the circumference of the wheel to find the number of revolutions needed to reach the desired positions.

Calculating Distances for Autonomous Period

44 inches forward:

Radius of wheel = 2 inches

Circumference of wheel: $2\pi r = 12,566$ inches

$\frac{44 \text{ inches}}{12,566 \text{ in.}} = 3,5$ revolutions forward

68 inches back:

Radius of wheel = 2 inches

$\frac{68 \text{ in.}}{12,566 \text{ in}} = 5,411$ revolutions in opposite direction

The calculated values for the revolutions are used in the float variables "revsToFlag" and "revsFromFlag" to improve the precision of the autons. We tested the reliability of the auton by conducting 13 timed trials and recording if the auton successfully hits the high flag and scored points by shooting the ball and if it successfully earned points for parking on the alliance platform.

From the trials, the auton hit the flag and got points 8 out of the 13 tries while the auton got points for parking 9 out of the 13 attempts.

Flag?	Platform?
✓	✓
✓	✓
X	X
X	✓
✓	X
✓	X
✓	✓
✓	X
✓	✓
X	✓
X	✓
✓	✓
X	✓

Continued to page

SIGNATURE [signature] DATE 11/15/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 11-18-18

Continued from page

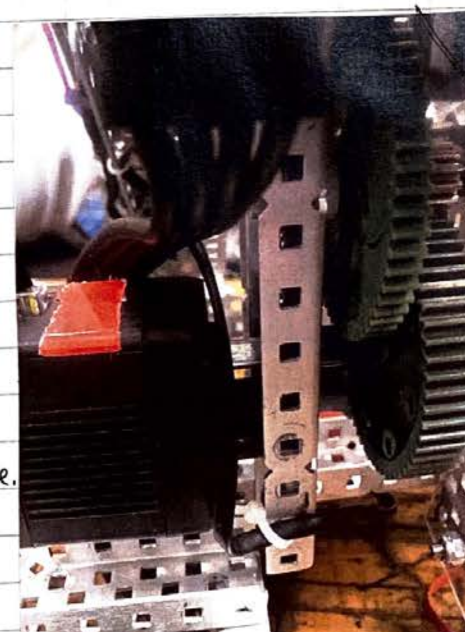
PROBLEM + SOLUTIONS

Problem: When acquiring caps, caps would keep on falling and flipping onto the robot, thus getting stuck there.

Solution: We attached standoffs to the L-intake to prevent the caps from flipping onto the robot. (shown to the right)

Problem: The metal that is attached to the standoffs would break off because there was nothing between them, and thus would bend due to its weakness.

Solution: Add spacers in between the two structures to offer more strength and support.



Problem: The low strength axle for the flywheel was not strong enough and as a result, was getting twisted whenever the flywheel would run.

Solution: Replace the low strength axle with a high strength axle because they are able to withstand twisting and bending fatigue, efficient for when there is high torque and/or high impact drives. (see photo) →

Problem: The caps are too heavy to flip on the L-intake.

Solution: We initially added a brake to the motors through our code where when the buttons controlling the cap were not pressed, motors would be braking, which helped, but the caps were still too heavy to lift. Therefore, we replaced the cap motor with a torque motor using a torque insert for the regular motor.

Continued to page

SIGNATURE [signature]

DATE 12/4/18 - 12/6/18

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 12-5-18

Continued from page

JOYSTICK CONTROL

L2 : L-intake goes forward

L1 : L-intake goes backward

movement

runs ball intake at high speed

R2 : Lift goes down

R1 : Lift goes up

movement

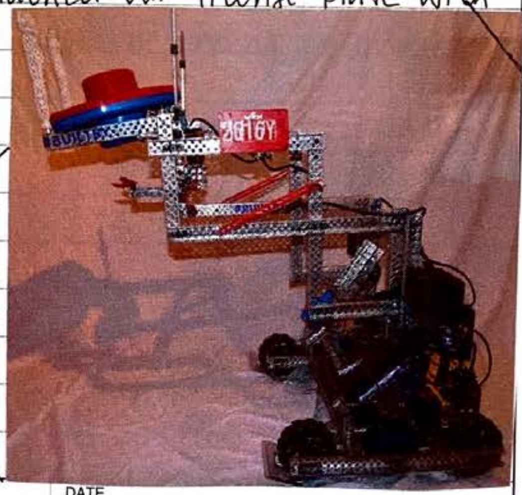
runs flywheel at maximum speed when pressed

REVIEWING OUR WORK

Upon finalizing our autonomous code, practicing driving, making last minute fixes to our robot and running test matches with our robot, we reviewed all of our work so far. First, we made sure our robot fit within the 18x18x18 inch confinement. When we found our robot was slightly larger, we trimmed down. We also found that our axle was hitting part of the flywheel interfering with its functionality. We practiced driving to get used to the joystick controls and scrimmaged with team 2616J. Lastly, we attached our license plate with one screw so that we can quickly switch team alliance colors.

CP

CP



SIGNATURE

DATE 12/7/18 - 12/8/18

DISCLOSED TO AND UNDERSTOOD BY

DATE 12-8-18

PROPRIETARY INFORMATION

Continued from page

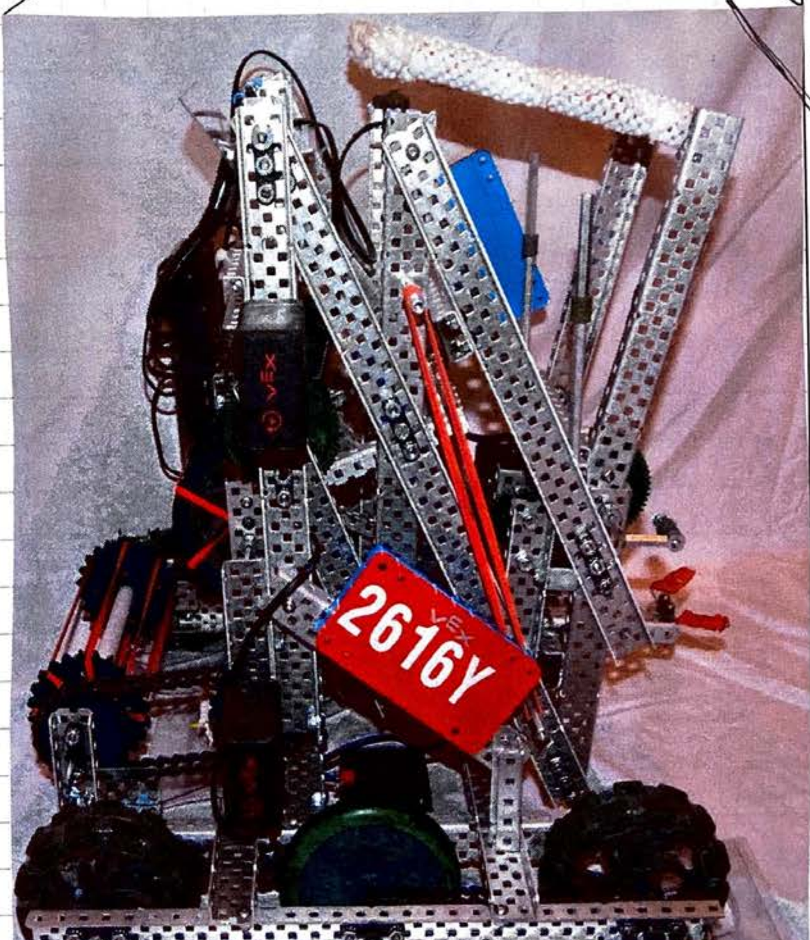
NIGHT BEFORE COMPETITION

Competition Checklist

- ☑ Battery
- ☑ Toolbox / replacement wires
- ☑ Gearbox
- ☑ Glasses
- ☑ Joystick
- ☑ Metal Shaver
- ☑ Computers
- ☑ Robot
- ☑ USB Cable
- ☑ License Plates
- ☑ Friction Padding

LK

Since the competition tomorrow is our first competition of the season, it is of utmost importance that we do not forget anything and that we double and triple check everything so as to prevent easily fixable mistakes. We also ~~added~~ checked our robot for any loose nuts or screws. Our goal for the TCNJ competition is to do our best and to improve our strategy by collaborating with teams in the region. We still have much to learn and tomorrow is the perfect opportunity to better understand how the game works in practice.



SIGNATURE LK

DATE 12/8/18

DISCLOSED TO AND UNDERSTOOD BY Caylin Payne

DATE 12-08-18

PROPRIETARY INFORMATION

DAY OF COMPETITION

Upon arriving at the College of New Jersey Competition, we registered and got our robot inspected, and later went through an interview about our team and our engineering process with the robot. Soon enough, we did the skills portion of the competition: first starting with driving and then to autonomous. For driving, we were able to get 11 points by flipping cap colors and parking onto the center platform. On the other hand, our auton did not work correctly for it was supposed to launch a ball to the top flag, move forward to hit the bottom flag, and move backwards onto the parking platform, but our robot missed the flag and started to turn towards the right, hit the flag pole, and missed toggling the flag. After that, we had 6 matches to compete in. Aside from our auton problem, one problem we had was the fact that we only had one battery. Because the V5 batteries are different from the legacy batteries, our school has not received all of the parts, so we had to compete with only one battery. This is problematic because our robot functions best with a fully charged battery, but since the battery would run out throughout the competition, we had to charge the battery in between each match. That being said, the time between each match was definitely not sufficient to get adequate charging, and we competed in each match with around one bar of battery. Another problem was that our launcher for balls was faulty as it hit too far to the left of the flag and was unable to aim correctly at the flag when positioned toward the flag during the autonomous period. Our decision to have an all-around robot also had some pros and cons. That is, while we were able to shoot balls, stack caps on the high and low posts, flip caps, and get onto the parking platforms, we were not especially skillful in one specific task. For instance, stacking caps were difficult as they require precision. Thus, we took a long time trying to get a cap on the post, despite our guide. Our flywheel takes a couple seconds to get running and to get the ball launched into the air, only for it to miss the flag since our aiming was off. Parking was particularly challenging as we only made it up on the platform once out of our six matches due to our robot's heavy weight of about 14 lbs. We observed that the roller ball intake was popular among teams, but we were one of the few teams with a flywheel as teams had ball shooters that "punch" the ball to the flags or

Continued to page

SIGNATURE [signature] DATE 12/9/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 12/9/18

Continued from page

DAY OF COMPETITION (CNTD.)

utilized a catapult launcher. We also observed how, like in our prediction, the majority of robots focused on shooting and/or flipping caps, thus weighing very little and they had facility getting up on the parking platforms. Because our cap intake and ball launcher were not optimal, we played some of our games defensively. However, because many robots we played alliance with were not as advanced in a specific task, both of us played defense, which was not as effective for gaining points. As we played defense, we were able to disqualify a team, but unfortunately we got entangled with another team through our roller ball intakes and were unable to move/play for the rest of that specific match. Ultimately, while we tried our best, we ended the competition with a ranking of 22 out of 23 teams.

Overall, our performance statistics from The College of New Jersey are as follows:

Skills:
Rank - 9 Driver Attempts - 1 Driver Highscore - 11 Programming Highscore - 0
Total Highscore: 11

Matches:
1) 14 - 15
2) 6 - 19
3) 15 - 5
4) 29 - 7
5) 8 - 20
6) 16 - 13

Ranking:
Rank - 22 W-L-T: 1-5-0 WPs/APs/SPs: 2/4/53

Continued to page

SIGNATURE

DATE 12/9/18

DISCLOSED TO AND UNDERSTOOD BY Nafessa

DATE 12/9/18

PROPRIETARY INFORMATION

REFLECTION OF COMPETITION

After the competition, we realized many things about Turning Point as well as changes we need to make to our robot and performance. First, we saw that the first 45 seconds into the minute and 45 second drive period does not matter. That is, any flag or cap scored does not matter. That is, any flag or cap scored does not matter since the opposing robots can simply reverse the flag or cap to get the point. We realized this after we spent around 10-20 seconds trying to put a cap on a post only to get it knocked off by the opposing team. The same applies to the flags as we observed how most teams go for the flags as they take a shorter time to toggle compared to attempting to get the same point values from the caps that generally take longer to score. Thus, we must fix our design.

BRAINSTORMING / OUR PLANS

Drive:

We plan on building a 4 motor drive, which will equip us for the highly aggressive fighting/defense that goes on in Turning Point, so that we can stand our ground. A 4 motor drive will provide us with greater speed, which is important because getting onto the parking platforms are worth a lot of points and we want to be sure that we do not have a difficult time getting up on the platforms. Turning Point revolves around speed and power, and successful robots are all about build quality rather than complexity. If we utilize a 4 motor drive, we can have one motor for our lift, one motor for the ball intake, one motor for the ball launcher, one motor for the ball intake, and one motor for the cap intake.

We also plan on finding a way to figure out why our robot "creeps" when we do not press any buttons on the joystick. When we say "creeping", we mean that the robot slowly moves and turns on its own when it is not supposed to be moving.

Continued to page

SIGNATURE [signature] DATE 12/10/18

DISCLOSED TO AND UNDERSTOOD BY Nafessa J DATE 12/10/18

REFLECTION OF COMPETITION

After the competition, we realized many things about Turning Point as well as changes we need to make to our robot and performance. First, we saw that the first 45 seconds into the minute and 45 second drive period does not matter. That is, any flag or cap scored does not matter. That is, any flag or cap scored does not matter since the opposing robots can simply reverse the flag or cap to get the point. We realized this after we spent around 10-20 seconds trying to put a cap on a post only to get it knocked off by the opposing team. The same applies to the flags as we observed how most teams go for the flags as they take a shorter time to toggle compared to attempting to get the same point values from the caps that generally take longer to score. Thus, we must fix our design.

BRAINSTORMING / OUR PLANS

Drive:

We plan on building a 4 motor drive, which will equip us for the highly aggressive fighting/defense that goes on in Turning Point, so that we can stand our ground. A 4 motor drive will provide us with greater speed, which is important because getting onto the parking platforms are worth a lot of points and we want to be sure that we do not have a difficult time getting up on the platforms. Turning Point revolves around speed and power, and successful robots are all about build quality rather than complexity. If we utilize a 4 motor drive, we can have one motor for our lift, one motor for the ball intake, one motor for the ball launcher, one motor for the ball intake, and one motor for the cap intake.

We also plan on finding a way to figure out why our robot "creeps" when we do not press any buttons on the joystick. When we say "creeping", we mean that the robot slowly moves and turns on its own when it is not supposed to be moving.

Continued to page

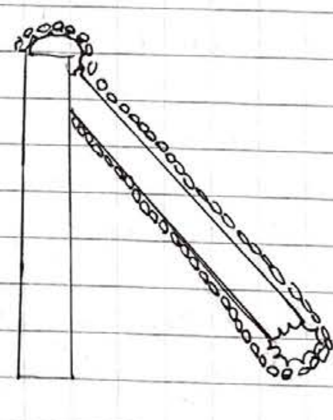
Continued from page

REFLECTION OF COMPETITION (CNTD.)

BRAINSTORMING / OUR PLANS (CNTD.)

Lift:

Additionally, it was blatantly apparent that we were one of the teams with a more developed lift, so there must be a reason why teams do not focus on a "complicated" lift. After the competition, we realized that anything more complex than a 2 bar lift yields unnecessary weight and thus is inefficient to the robot. As a result, based on how the game is scored and the point values for each of the tasks, we reasoned that our lift should be made simpler. With a simpler lift, we can dedicate more motors to our drive since we are limited to a maximum amount of 8 motors to our robot. As of now, we see the 2 bar lift as a viable solution for achieving the tasks needed to flip caps, etc. The 2 bar is simple yet capable of performing what we need it to do in the game, thus we believe it will serve us well in terms of attaining a lower robot weight but still provide a successful performance. Any lift more complex than a 2 bar would be unnecessary since simple lifts can accomplish enough to get points to win. It is better to have a 2 bar than our current 6 bar lift because we can accomplish more with our time through flipping other caps off, or changing the color of the caps. So, we will not be stacking caps as of now in the season. When building the 2 bar, we plan on chaining it to be able to add extensions so we can flip off caps on the tall posts as observed in last competition, where teams struggled to reach the tall posts.



Above is what we plan on constructing for our lift: the 2 bar with chain so that we can add additional pieces in order to reach the tall posts.

Continued to page

SIGNATURE

DATE 12/10/18

DISCLOSED TO AND UNDERSTOOD BY

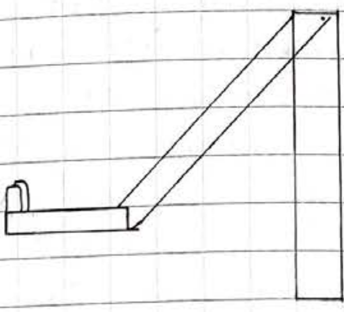
DATE 12/10/18

PROPRIETARY INFORMATION

REFLECTION OF COMPETITION (CNTD)

BRAINSTORMING/ OUR ~~A~~ PLANS (CNTD)

Ball Launcher:



Our ball launcher also needs improvement. As much as we had hoped that our flywheel would be able to seamlessly shoot balls powerfully to the flags, this was not the case based on our performance at the competition. The flywheel did not offer us as much power we had hoped for when shooting balls once we saw the powerful and precise ball shooters that other teams had. Our flywheel also had to run for a few seconds before finally launching the ball out. Every second counts in Turning Point, so we knew that there must be a better ball launcher we could use to shoot balls with more power and accuracy. At the competition, we saw punchers and catapults besides other flywheels.

- A puncher is a mechanism that winds up to store potential energy, and then using a slip gear or a nautilaus gear, releases it to punch the ball out with enough force.

- A catapult is a device in which accumulated tension is suddenly released to hurl the ball out.

Both the puncher and catapult are more powerful than the flywheel, so we plan on changing our design to either one. However, after seeing how the catapult can load two balls into the launcher, we see ourselves shifting to a catapult shooter, to maximize the amount of balls we can shoot at once, and ultimately the points we can score. Such a shooter is called a double catapult, and it is a beneficial launcher because we can hit the top two flags at once.

Continued to page

SIGNATURE [signature] DATE 12/10/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 12/10/18

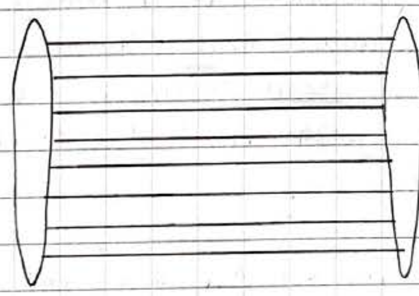
Continued from page

REFLECTION OF COMPETITION (CNTD.)

BRAINSTORMING / OUR PLANS (CNTD.)

Ball Intake:

The rubber band roller intake for taking balls into the robot seems to be the most used and effective ball intake we have seen thus far in the season. However, the way in which the roller intakes are arranged is different among teams. In other words, we have seen single roller intakes as well as multiple rollers assembled together to take in balls and guide them toward the ball launcher. Because we are changing our ball launcher, we will have to modify or add onto our current single roller intake. We will have to do so in a way that we can guide the balls to the catapult launcher.



Cap Intake:

With flags and the ability to park easily as some of our priorities and goals for our robot, we are not sure yet of what we plan to do for caps. We will focus more on building a quality drive and ball launcher and intake before moving on to the cap intake.

"L" intake is not the most efficient and effective cap intake design/structure. We must change our design and re-evaluate our goals for the robot.

Continued to page

SIGNATURE

DATE 12/10/18

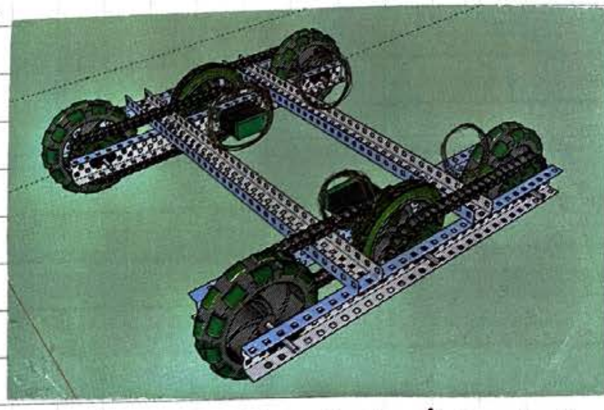
DISCLOSED TO AND UNDERSTOOD BY Nafissa

DATE 12/10/18

Continued from page

BUILDING THE 4 MOTOR DRIVE

We do not have to make major modifications to change our current 2 motor drive into a 4 motor drive. All we have to do is add 2 more motors on the middle friction wheels of the drive, because there is enough space for these motors. We now have one motor on each of the back omni-wheels and one on each of the ~~two~~ middle friction wheels. Having a 4 motor drive will help the robot to move faster, which not only helps us maneuver faster around the field, but also provides us with more time to score points. Additionally, a faster drive aids us in getting up onto the parking platforms successfully, which was a problem we had last competition.



* The circled areas on the drive to the right locate where we plan to place our 4 motors to construct the 4 motor drive.

Problem: When the drive was not being controlled on the joystick, the robot would move slightly, or "creep." This is extremely inconvenient because when we would pick up caps or shoot a ball, from last competition, it would change the position of the robot and prevent the robot from picking up the cap or shoot at a different angle.

Solution: We solved this by implementing a threshold/deadband into our code:

```
while (1) {
if  (Controller1.Axis3.value() > 10 || Controller1.Axis2.value() > 10
      Controller1.Axis3.      () < -10 || Controller1.Axis2.value() < -10 ){
  LFront.spin(vex::directionType::fwd, Controller1.Axis3.value()/1.3, vex::velocityUnits::pct);
  RFront.spin(vex::directionType::fwd, Controller1.Axis2.value()/1.3, vex::velocityUnits::pct);
  LBack.spin(vex::directionType::fwd, Controller1.Axis3.value()/1.3, vex::velocityUnits::pct);
  RBack.spin(vex::directionType::fwd, Controller1.Axis2.value()/1.3, vex::velocityUnits::pct);
}
  else{
  LFront.stop(brakeType::brake);
  RFront.stop(brakeType::brake);
  LBack.stop(brakeType::brake);
  RBack.stop(brakeType::brake);
}
```

When the controller's 3 and 4 axis values were between -10 and 10 (very small values, but enough that when buttons are not being controlled, it would still be read and cause movement), the motors would stop and prevent drifting. This is beneficial for us when we are on the top platform because when motors are stopped, great amount of force must be applied to push the robot.

Continued to page

SIGNATURE [signature] DATE 12/10/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 12/10/18

Continued from page

PLANNING OUT CATAPULT DESIGN

First, we are going to mockup a design of the double ball catapult. We are starting to brainstorm different ideas to prevent future problems with this mechanism. To be successful, this design must consist of

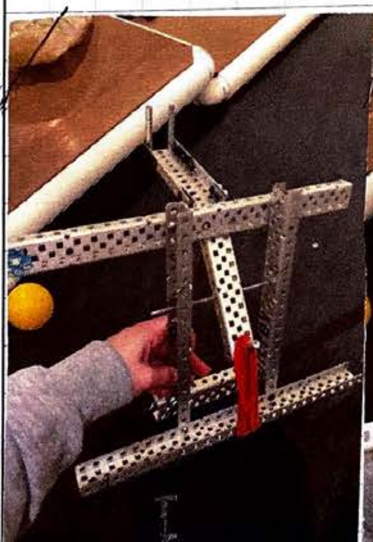
1) A place to hold both balls in place

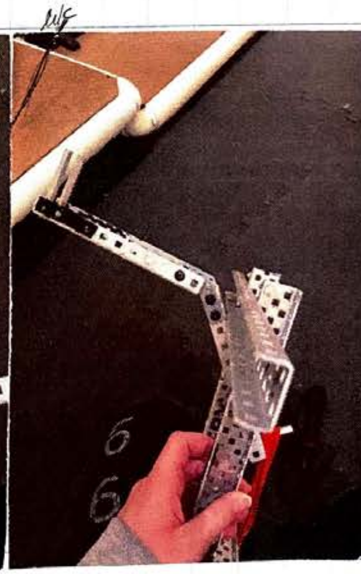
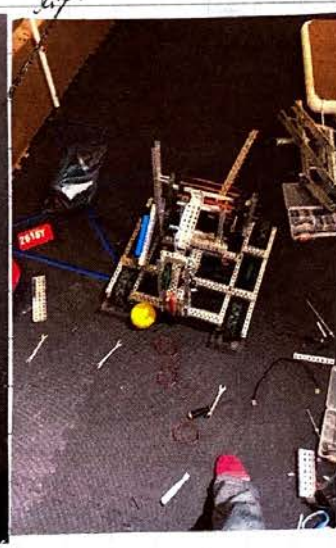
2) A bar connected to the holder, in which an axle will run through to eventually connect to the motor (depending on gear ratio)

3) A stopping bar to stop the catapult itself, but fling the balls forward due to inertia.

4) Rubber bands to create tension

The rubber bands have to extend from the base, which will be connected to the drive, and to the bottom of the catapult. This will allow the bands to stretch, which will build/store potential energy. As the catapult is released, the rubber bands return to equilibrium position, converting their potential energy to kinetic energy.

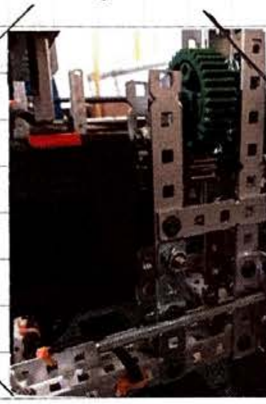


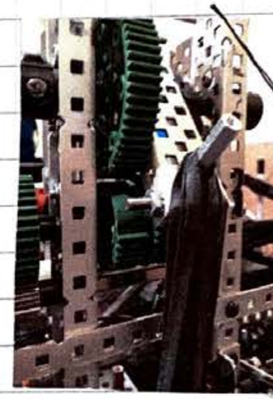



Continued to page

BUILDING THE CATAPULT GEAR BOX

To construct the slip gear, a 60 tooth gear (connected to the catapult) is matched with a 36 tooth gear that contains the slip. In order to slip the gear, we used the dremel to cut off the proper amount of teeth for the launch. Next to the 36 tooth gear, there is another gear that will later have a pawl attached to it, so that we are able to stop the catapult at a position to load the balls. Outside of this setup, there is another setup to make the torque ratio of 1:5. We accomplished this by having the drive gear as a 12 tooth gear and a 36 tooth gear as an idler gear, which then is connected to the inside, which makes the 36 tooth gear also an idler gear. Therefore, the 12 tooth and 60 tooth create the 1:5 gear ratio.





Problem: When the catapult is shot, the shock from the stopping bar causes the whole structure to shake back and forth.

Solution: To stabilize this structure, we will be able to add triangles to this design. To connect the catapult structure to the base, we will use standoffs with a screw that is threaded through the collar to allow the standoff and screw to create a 90° angle. A triangle structure was added to reduce the force created by the catapult as it shoots forward. The triangle reduces both the horizontal and vertical force. Without the triangle structure, the catapult would instead cause the base of the robot to undergo mechanical shock, a sudden acceleration caused by a transient physical excitation. The equilibrium of the system is disrupted. By adding the triangle, the catapult is significantly more stable.

Continued to page

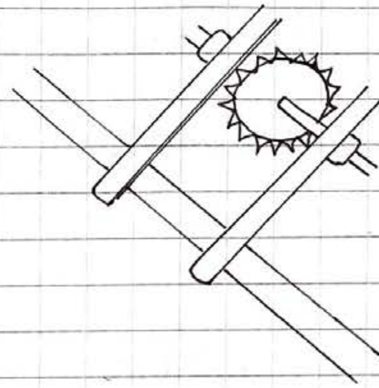
SIGNATURE lk DATE 12/11/18

DISCLOSED TO AND UNDERSTOOD BY Nafeesa DATE 12/11/18

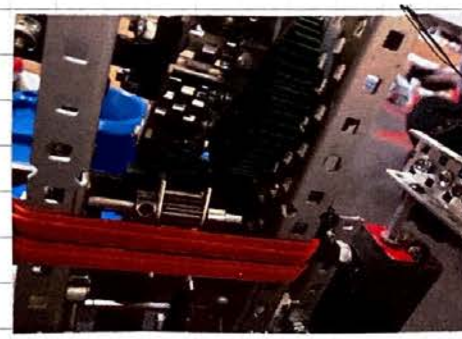
Continued from page

DESIGNING THE PAWL

The pawl is attached to the second 36 tooth gear that is on the inside gear box (this is not the slip gear). The pawl is placed so that as the motor pulls the catapult back, it locks onto the gear to prevent the catapult from shooting, or going back up without the prompting of the driver. The pawl will allow us to load two balls into the catapult in one position. When the motor is turned, the pawl bounces back to allow the motor to turn backwards, but when the motor is not being pressed, it locks onto the gear to prevent the tension of the rubber bands from pulling it back to its original position. This simple design consists of an axle running through the 12 tooth gear with a square insert, and an axle running through the last hole, which is held with colors outside of the gearbox.



Problem: The pawl sometimes falls back, taking it away from the gear.



Solution: In order to keep the pawl from losing position when the catapult is being pulled back, we stretched rubber bands across the structure, which forces the pawl to bounce back into position.

Continued to page

SIGNATURE O.K.

DATE 12/13/18

DISCLOSED TO AND UNDERSTOOD BY Nafissa

DATE 12/13/18

PROPRIETARY INFORMATION

Continued from page

HOW MANY RUBBER BANDS?

When designing the catapult, it was determined that rubber bands would need to be added in order to create tension. We used a tension equation to calculate the correct number of rubber bands needed, taking into account the force from the motors needed to bring the catapult back.

Force created by the stretch of one rubber band:

$F = kx$

$x = 6\text{inches} - 4\text{inches} = 2\text{inches}$

$2\text{inches} = 0.0508\text{meters}$

$k = 40\text{Nm}$

$F = (40\text{Nm})(0.0508\text{m}) = 2.032\text{Nm}^2$

Force required to launch ball:

$\frac{200\text{rpm}(4\pi)}{60} = 1.06395\text{m/s}$

Total velocity: $\sqrt{(1.06395)^2 + (0.11806)^2} = 1.06396\text{ m/s}$

Horizontal velocity: $1.06396\cos 20 = 0.434183\text{m/s}$

Vertical velocity: $1.06396\sin 20 = 0.971337\text{ m/s}$

Acceleration: $1.06395/0.011 = 96.7236\text{m/s}^2$

Linear Force $= (96.7236)(0.054431\text{kg}) = 8.7655$

Torque $= (8.7655)(0.1524\text{m})\cos 20 = 9.51064$

Torque + Linear Force = Total Force

$11.51433/2.032 \approx 6$ rubber bands

It was mathematically determined that we should use 6 rubber bands. After using this as a starting point, we adjusted to use 7 rubber bands, that is, newly, unstretched rubber bands. We added the linear force to the angular force (torque) and then dividing the total force by elastic force created by stretching one rubber band a distance of 2 inches, or 0.0508 meters. The estimated value came to be 5.8 bands, or 6 bands, but we adjusted to 7 rubber bands later on.

Continued to page

SIGNATURE [signature] DATE 12/13/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 12/13/18

Continued from page

BUILDING THE LIFT

To create the 2 bar lift, one bar will be extending vertically from the drive to act as the structure to the arm. The arm is a 26x2x1 length c-channel. A sprocket is attached to the vertical c-channel at the top with a circular insert. This sprocket will have an axle running through it that will be connected to the motor and to the arm. A bearing with a square insert will be added to the outside of the arm so that the arm will rotate when the motor is turning. At the bottom of the arm, another sprocket is attached with a circle insert and a screw running through to a piece of metal on the other side of the sprocket, which it is secured to. The piece on the outside of the sprocket has an extension added so that we will be able to hit the caps off of the high poles. The chain is then run between the two sprockets so that no matter where the chain bar runs, the extension piece will always be at the same angle.

Torque required to Flip cap

How much torque is needed to flip caps?

cap weight = 0.335 kg

$(0.335\,kg)(9.8\,m/s^2) = 3.286\,N$

$(3.286\,N)(0.4572\,m) = 1.50236\,N$

Torque Load

Length of new lift = 18 in.

Max. Force = Stall Torque/5 / Arm Distance

= 2.1 Nm / 0.4572 m = 4.593 N

3.286 N / 4.593 N = 0.715 motors

The torque required to flip one cap with our new lift is 3.286 N and torque load is 1.50236 N, with only 0.715 motors required.

Continued to page

SIGNATURE

DATE 12/13/18

DISCLOSED TO AND UNDERSTOOD BY Nafissa

DATE 12/13/18

PROPRIETARY INFORMATION

Continued from page

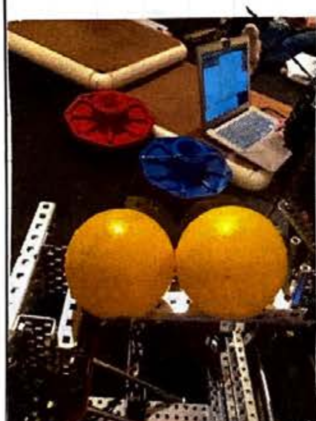
BUILDING THE BALL INTAKE

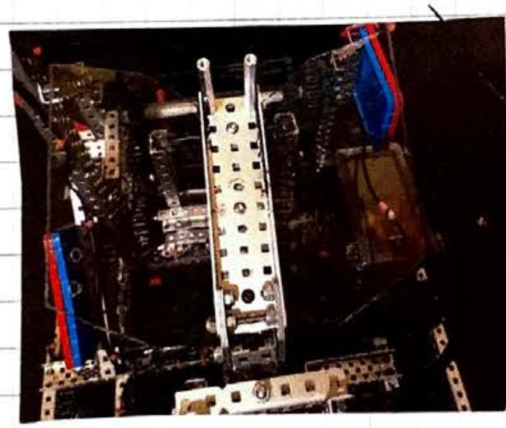
Since we are keeping the same roller intake concept, we are following the same build process of using sprockets and rubber bands. However, we are building 2 rollers on top of each other to lead the ball up to the catapult and into the plexiglass guide. Like in our first ball intake design, we are having a plexiglas ramp that feeds the ball into the rollers.

Problem: When the balls are loaded into the catapult, they begin to fall out as the catapult is brought back.

Solution: In order to line the balls up correctly, we will construct a plexiglas guide that will surround the catapult to keep the balls from falling out of the sides.

Building: Using a heat gun, we are slowly bending the plexiglas to the shape of the catapult.





Problem: The catapult shifted, and its axle moved out of the motor.

Solution: We inserted a bar across, so that the motor stays put and keeps the catapult from shifting positions.

Continued to page

SIGNATURE [signature] DATE 12/14/18

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 12/14/18

PROPRIETARY INFORMATION

Continued from page

BUILDING THE CAP INTAKE

Because we do not want to focus on stacking caps, we will make sure that our robot is able to flip caps to the right color to add to the tasks that our robot can accomplish to ultimately score more points. We must think simply as there are many ways to carry out a cap flipping action, and a complex design does not necessarily ~~gaur~~ guarantee the most efficient design. Instead of utilizing the L intake as the cap intake, we decided to build another mechanism to flip caps. We used a standoff screw and attached it to half of a metal c-channel to create an L shape. When the motor runs, the L channel can flip up and down, so when we approach the cap, we can flip it up and it will successfully flip.

Problem: The bearing with square inserts broke, so it essentially became circle inserts. As a result, the axle spinning through it did not let the flipper, or the standoff, to go up and down.

Solution: We doubled up on square inserts so that the inserts did not break, and the flipper is secured.

Continued to page

SIGNATURE [signature]

DATE 12/14/18

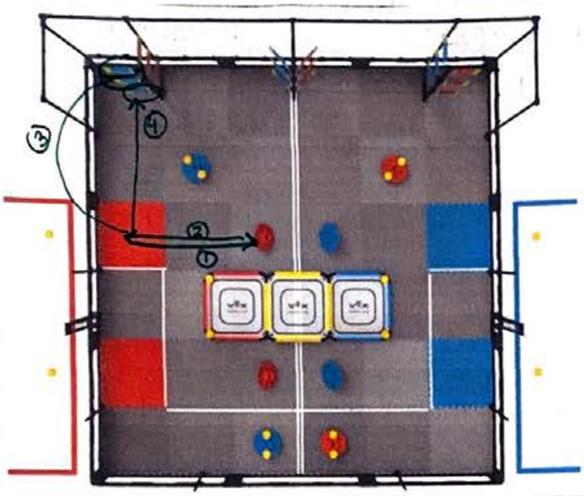
DISCLOSED TO AND UNDERSTOOD BY Nafissa [signature]

DATE 12/14/18

Continued from page

AUTONOMOUS CODE: FRONT

We decided to update our autonomous code to score more points and work more effectively.



Strategy: (red side example)

our strategy is to begin on the front platform with the rollers facing towards the slanted disk with the ball under it. The robot then moves forward to take the ball in, scoring ~~two~~ one point by moving the disk to red. The robot will then move back to the platform before turning to face the flags and shooting the balls (one preload, one obtained) at the top two flags, scoring two points each → 4 total. Finally, the robot moves forward to hit the lowest flag, scoring one point.

This program scores a total of 6 points in 15 seconds.

```
//rollers spin some more
float roll = 3.4;
Rollers.rotateFor(roll, revolutions);
vex::task::sleep(250);

//moves backward back to start
float revsFromBall = 3.1;
LFront.startRotateFor(revsFromBall, revolutions);
RFront.startRotateFor(revsFromBall, revolutions);
LBack.startRotateFor(revsFromBall, revolutions);
RBack.rotateFor(revsFromBall, revolutions);
vex::task::sleep(250);

//left turn
//right forward left back
float revsTurn = 0.9;
LFront.startRotateFor(revsTurn, revolutions);
LBack.startRotateFor(revsTurn, revolutions);
RFront.startRotateFor(-1*revsTurn, revolutions);
RBack.rotateFor(-1*revsTurn, revolutions);
vex::task::sleep(250);

//shoot balls
float catapultFling = 0.5;
```

The code to the left controls the intake of the ball under the disk, movement back to the platform, and turn towards the flag before the balls are launched. The highlighted lines show the physical manifestation of the code. The left turn makes use of −1 as a multiplying factor for the right side of the drive in order for the robot to turn in the correct direction. When this auton is executed on the blue side, the left side of the drive is multiplied by −1 for the number of wheel rotations it has to make instead → left wheels spin backwards → successful right turn to face flags.

Continued to page

SIGNATURE [signature] DATE 12/15/18

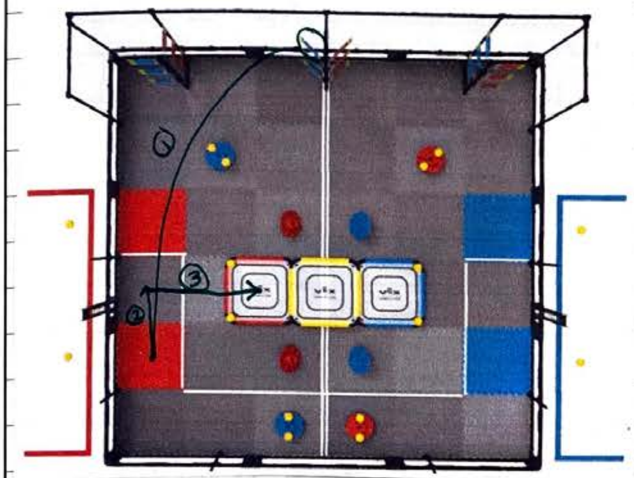
DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 12/15/18

PROPRIETARY INFORMATION

Continued from page

AUTONOMOUS CODE: BACK

We decided to take advantage of the ability to upload multiple programs onto the V5 cortex and strategized an autonomous program for the back platform.



Strategy: (red side example)
When our alliance partners have effective autons for the front platform, it is efficient to start from the back platform and park in order to score 3 points and add to the total alliance autonomous score. However, we observed this strategy at our previous competition and devised a plan to score even more points from the back platform without interfering with the front auton. We begin with our robot at an angle facing the middle column of flags. We then shoot the preloaded ball at the top middle flag before straightening our position, moving forward, turning towards the platform and parking, scoring 5 points from the back platform without interfering with most front autons.

```
Flipper.spin(directionType::fwd,200,velocityUnits::rpm);
   task::sleep(100);

//shoot
float revsShoot = 0.50;
Catapult.rotateFor(revsShoot,revolutions);
vex::task::sleep(1000);
Catapult.stop();

//turn
float revsTurnn = 0.25;
LFront.startRotateFor(revsTurnn*-1, revolutions);
LBack.startRotateFor(revsTurnn*-1, revolutions);
RBack.startRotateFor(revsTurnn, revolutions);
RFront.rotateFor(revsTurnn, revolutions);
vex::task::sleep(500);
```

Near the beginning of the autonomous code, the flipper motor is spun in the forward direction for 100msecs in order to put the flipper up and keep it there. This allows the robot to alliance park successfully at the end of the autonomous period. The left drive revolutions are multiplied by -1 so the left drive spins in reverse, allowing the robot to turn right and climb the red platform. When on the blue alliance, the right drive is programmed to spin in reverse instead, turning the robot left + onto the blue platform.

Continued to page

SIGNATURE [signature] DATE 12/15/18

DISCLOSED TO AND UNDERSTOOD BY Nafissa J. DATE 12/15/18

Continued from page

AUTONOMOUS TRIALS

To determine any issues with the autonomous strategy or code, we conducted trials for both the front and back autons.

Data Table 1: Front Autons

Attempt	# of Points	Notes
1)(Blue)	6	
2)(Blue)	6	
3)(Blue)	0	Lined up incorrectly
4)(Red)	6	
5)(Red)	4	Missed top flag
6)(Red)	6	

$\bar{x} = 4.\bar{6}$

$S_x = 2.42$

$n = 6$

$\Sigma x = 28$

With a small # of trials, we had a relatively large standard deviation, yet the average indicates that we can expect to consistently score at least ~4 points, given the robot is lined up correctly.

Data Table 2: Back Autons

Attempt	# of Points	Notes
1 (B)	0	Lined up incorrectly
2 (B)	5	
3 (B)	5	
4 (B)	3	Lineup incorrectly - missed flag
5 (R)	5	
6 (R)	~~8~~ 3	Missed flag
7 (R)	3	Missed flag
8 (R)	5	

$\bar{x} = 3.625$

$S_x = 1.77$

$n = 8$

$\Sigma x = 29$

The trials indicated that we needed to practice successfully aligning the robot in order to score the intended 5 points, but that we could consistently expect the 3 points from alliance parking.

Continued to page

SIGNATURE [signature] DATE 1/3/19

DISCLOSED TO AND UNDERSTOOD BY Nafessa J. DATE 1/3/19

DRIVING SKILLS STRATEGY LAB

Objective: To find the most efficient path for the driving period of the skills competition.

Hypothesis: If we are able to hit both the top and middle flags from the auton line (4 points), hit the bottom flag (1 point), turn the other bottom two flags (2 points), flip 2 discs (2 points), and then park (6 points) to obtain a total of 15 points, then the following path will be the most efficient path for the skills competition.

Materials:
1) Timer
2) Robot and robot materials (joystick, fully-charged battery)
3) Field
4) Data Sheet

Procedure:
1) Set up the field with all colors facing blue, and place robot in the red box closest to the flags. The skills competition always has the robot start in the red auton box.

2) Start with the first routine, go through each step, and then record the time it takes to complete that round.

3) After each round, explain the issues and successes during the trials.

4) Complete this step three times, and average out the times of ~~the~~ runs in seconds (s).

5) Complete steps 1-4 for the following two paths.

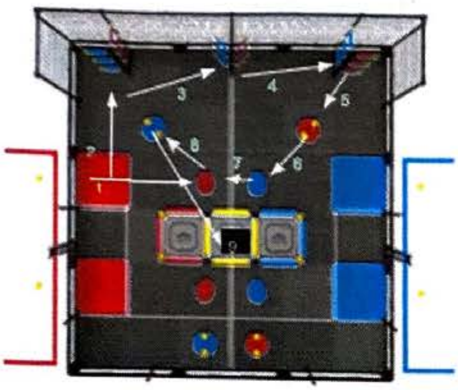
SIGNATURE [signature] DATE 1/3/19

DISCLOSED TO AND UNDERSTOOD BY Nufessu [signature] DATE 1/3/19

Continued from page

DRIVING SKILLS STRATEGY LAB

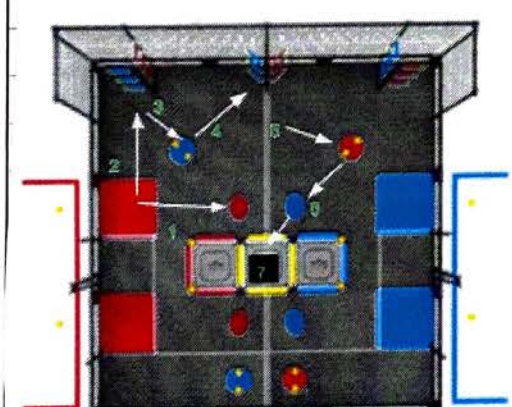
Data Table 1: Grab ball from cap, back up, shoot top 2 flags, hit bottom flag, hit last 2 bottom flags, flip 2 caps, park on top platform. (15 pts.)



Trial #	Time (s)	Success/Issues with Explanation
1	63.00	Issue: It took longer than expected to set up the robot to aim and hit the top 2 flags.
2	56.00	Success: The robot quickly took up the first ball with no faults.
3	59.00	Success: The robot was perfectly in line to hit all bottom flags, which reduced our overall time.

Average Time: 59.33 s
S_x = 3.51 s

Data Table 2: Grab ball from disk/cap, back up, shoot 2 top flags, hit bottom, flip cap, grab 2 balls, hit second top 2 flags, hit bottom, flip ~~another~~ 2 caps, park on top platform. (18 pnts.)



Trial #	Time (s)	Success/Issues with Explanation
1	75.00	Issue: Trouble getting balls; they roll away to fast.
2	57.00	Success: Robot was perfectly allgned and hit top 2 flags easily.
3	59.00	Success: The robot easily ascended the platform.

Average Time: 63.66 s
S_x = 9.87 s

Continued to page

SIGNATURE [signature] DATE 1/8/19

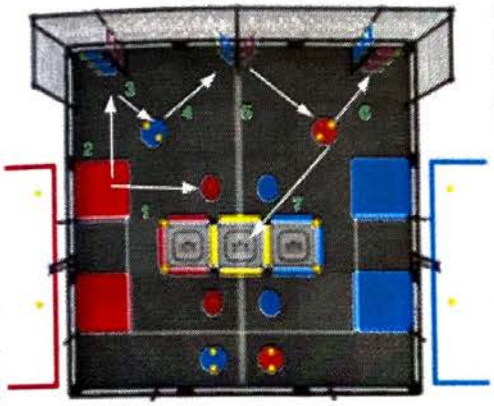
DISCLOSED TO AND UNDERSTOOD BY Nafessa [signature] DATE 1/8/19

PROPRIETARY INFORMATION

Continued from page

DRIVING SKILLS STRATEGY LAB

Data Table 3: Grab ball from cap, back up, shoot top 2 flags then bottom, flip cap, grab 2 balls, hit second top 2 flags then bottom, flip another cap, grab 2 balls, hit last top 2 flags then bottom, park on top platform (21 pts.)



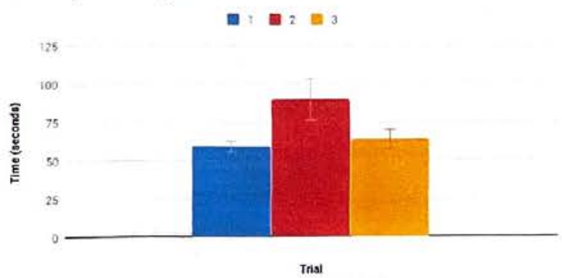
Trial #	Time (s)	Success/Issues with Explanation
1	90.00	Issue: It took extra time to flip the cap because the flipper was not working at full power.
2	105.00	Issue: Difficulty parking on top platform. Robot kept rolling off first platform.
3	75.00	Issue: The robot had trouble grabbing the 2 balls that rolled off cap when flipped.

Average Time : 90.00 s
S_x = 15.00 s

Analysis: The third trial takes the largest average time to complete, at 90 seconds, though it scores the most points. However, it also encountered an issue in each trial with a high standard deviation of 15 sec, leading us to doubt its reliability and feasibility within the time limit. The first trial can be completed significantly faster at about 60 sec with its low standard deviation, indicating its reliability. However, it only scores 15 points. The second trial can be completed in an average of about 64 sec, close to the time of the first trial, yet scores 3 more points. Though it has a higher standard deviation than the first trial, it has the same reliability record.

Conclusion: From the results and analysis of this lab, we decided to use the second strategy to score 18 points for skills. The 21 point strategy was not feasible in the given time period. Though the standard deviation is larger and takes longer than first strategy, the 18 point strategy is most feasible to score the most points during skills.

Graph 1: How does the mean average of points scored in each trial vary according to the standard deviation?



SIGNATURE

CK

DATE

1/8/19

DISCLOSED TO AND UNDERSTOOD BY

Nafisser

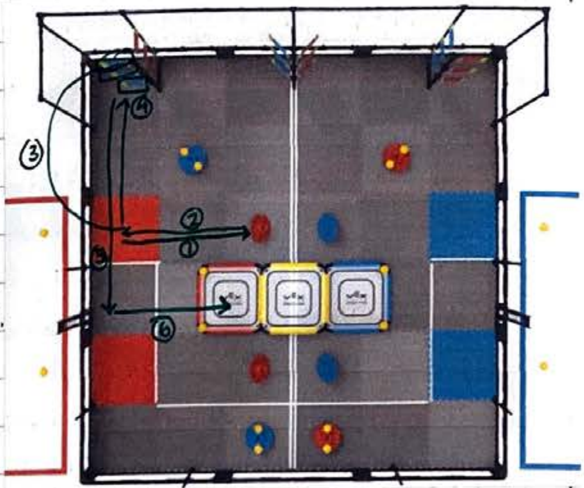
DATE

1/8/19

Continued from page

SKILLS CHALLENGE

Strategy: During the autonomous period during competitions, our front auton scores 6 points, but is unable to earn parking points due to the time constraint and presence of the auton line. During the auton skills challenge, however, the robot has 60 seconds. This gives us time to program the robot to back up after hitting the bottom flag, turn right, and climb onto center platform, scoring a total of 12 points.



```
//moves back
float revsBack = -3.0;
LFront.startRotateFor(revsBack, revolutions);
RFront.startRotateFor(revsBack, revolutions);
LBack.startRotateFor(revsBack, revolutions);
RBack.rotateFor(revsBack, revolutions);
vex::task::sleep(5000);

//turn
float revsTurnPlat = 0.8;
LFront.startRotateFor(revsTurnPlat, revolutions);
LBack.startRotateFor(revsTurnPlat, revolutions);
RBack.startRotateFor(revsTurnPlat*-1, revolutions);
RFront.rotateFor(revsTurnPlat*-1, revolutions);
vex::task::sleep(500);

//smash into back wall??
float revsSmash = -0.15;
LFront.startRotateFor(revsSmash, revolutions);
LBack.startRotateFor(revsSmash, revolutions);
RBack.startRotateFor(revsSmash, revolutions);
RFront.rotateFor(revsSmash, revolutions);
vex::task::sleep(1000);

LFront.setVelocity(200, velocityUnits::rpm);
RFront.setVelocity(200, velocityUnits::rpm);
LBack.setVelocity(200, velocityUnits::rpm);
RBack.setVelocity(200, velocityUnits::rpm);

//climbs platform
float revsToPlatform = 6.7;
LFront.startRotateFor(revsToPlatform, revolutions);
LBack.startRotateFor(revsToPlatform, revolutions);
RBack.startRotateFor(revsToPlatform, revolutions);
RFront.rotateFor(revsToPlatform, revolutions);
vex::task::sleep(1000);

Rollers.stop(brakeType::brake);
LFront.stop(brakeType::brake);
RFront.stop(brakeType::brake);
LBack.stop(brakeType::brake);
RBack.stop(brakeType::brake);
Catapult.stop(brakeType::brake);
Flipper.stop(brakeType::brake);
```

The code from the Competition autonomous period is shown to the left, which performs the actions highlighted: moving back, turning, moving back, and climbing the parking platform. Notably, the robot gently backs into the back wall in order to line up correctly with the center platform to score the 6 parked points.

The shown code is appended to the end of the competition code's autonomous section and is uploaded into a different slot on the cortex, allowing us to quickly and efficiently create the program.

Continued to page

SIGNATURE lik

DATE 1/10/19

DISCLOSED TO AND UNDERSTOOD BY Nafissa J.

DATE 1/10/19

PROPRIETARY INFORMATION

Continued from page

NIGHT BEFORE COMPETITION

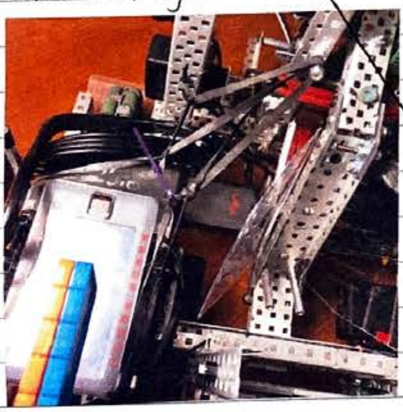
With tomorrow's competition as our second competition of the season, we want to make the ~~p~~ most of it by practicing on the field to get a better and more realistic feel for the competition. We scrimmaged with team 2616E on our homemade field to get comfortable with other robots on the field and to improve our driving skills. That is, our driver appeared to be more apt and fluid with handling the joystick control compared to the last competition. Also, because we added extra motors to create a 4 motor drive, the robot is easier to drive with. We practiced all of our robot's tasks – shooting two balls at once, hitting the bottom flag, flipping caps, and parking onto the center platform. We saw that the robot must align precisely perpendicularly with the parking platforms to get on successfully, which is something the drive team will remember for the competition.

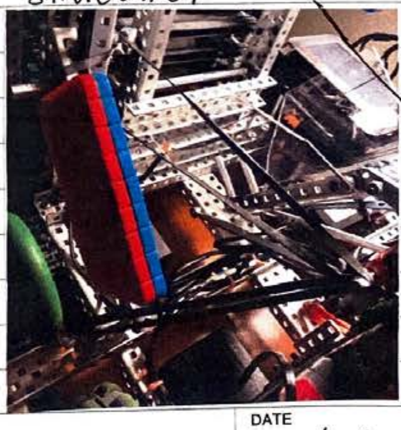
Problem: The left side of the drive is moving slower than the right side, which is a problem because when the robot aims to go straight, the robot turns

Solution: We remounted the front left motor, and the drive started working normally again.

Problem: When the robot took in balls from the ball roller intake into the catapult for shooting, balls would sometimes fall into the robot.

Solution: We prevented balls from falling into the robot by designing and constructing a rubber band structure.





Continued to page

SIGNATURE

DATE 1/11/19

DISCLOSED TO AND UNDERSTOOD BY Nafissa

DATE 1/11/19

DAY OF COMPETITION

Upon arriving at Sparta High School for our second competition, we immediately went onto the field to test out autons and drive our robot. Then, we went to get our robot inspected before turning in our notebook for judging. Compared to our first competition in December, there were significantly more teams here: 54 teams! We observed many robot designs, from catapults to roller ball intakes. Only a good handful of teams had cap intakes and lifts to accomplish stacking caps. In the skills challenge, we scored a column of flags, all of the bottom flags, flipped 2 caps to our color, and parked onto the center platform for a total of 14 driver points. For skills autonomous, we only scored 3 points by hitting the top and bottom flag. We had 8 matches total, and we changed our main auton to use for most of the matches. That is, we programmed the robot to autonomously get the ball from under the cap next to the parking platforms, to score the cap with the correct color, and then park onto the alliance parking platform. But, for our second and third matches, we had disconnecting issues with our brain. Thus, we could not drive our robot for the majority of the match, and ultimately lost those 2 matches.

Problem: Robot keeps disconnecting during driving.

Solution: Update the firmware to the most updated version.

After talking to the officials of the competition, we took their advice of updating the firmware to the 1.0.5 version. This was an imperative action that we took because it improved handling of temporary field disconnects, fixed controller lockups, improved battery stability and other bugs. Thus, for the rest of the competition, we did not have disconnecting issues and we were able to finish the competition 5-2-1. However, in our 6th and 7th matches, we experienced some misunderstanding marked our 6th match as our loss, when it was actually a tie, and our 7th match as a tie, when we won. In the end, we regained our true scores after discussions and a rematch. Overall, many robots played defense, by parking on the opposing platform, but we were able to end the competition with a rank of 12, which we are very satisfied with. On top of that, we became the 8th alliance, and picked team 5249S as our partner! Although we lost the quarterfinals, 10-24, against the first alliance team, we are very pleased with today's performance.

Continued to page

SIGNATURE eik

DATE 1/12/19

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 1-13-19

Continued from page

REFLECTION OF COMPETITION

Ending the competition with a skills ranking of 9 and qualification ranking of 12, we are very satisfied with our performance at Sparta High School. As a result, we believe our robot design was effective since it proved generally successful through our performance and end rankings. While most teams' robots had catapults that were off-centered and placed towards either side of the robot, our robot was the only one at the competition with a centered catapult. Most robots that had their catapults on the side most likely had sensors that helped them line up with the flags, but since we do not have sensors, our centered catapult is important because it is easier for us to align the robot to shoot the balls accurately towards the flags without the use of sensors. Also, the centered catapult allows us to be spatially aware of the space available on our robot. For instance, with the catapult placed in the center of the robot, we have space to add other specs, like a cap descorer.

We do, however, want to improve on our autons for the skills challenge. We think we can definitely score more autonomous points than the 3 points we scored at the competition. So, we will strategize a new plan for the upcoming competition this saturday, to increase our overall skills score and ranking. We also want to improve our efficiency in flipping the caps. During the competition, we found that though our cap flipper worked, it was a bit of a hassle to flip the caps. That is, to move the flipper up and down, we had two buttons to control those actions. Thus, it was a bit confusing and not the most efficient way to flip caps for our driver. It was also not guaranteed that the cap was flipped on the first try, so we want to fix our cap intake to improve efficiency and accuracy before our next competition.

L.K.

Continued to page

SIGNATURE

DATE 1/13/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1-13-19

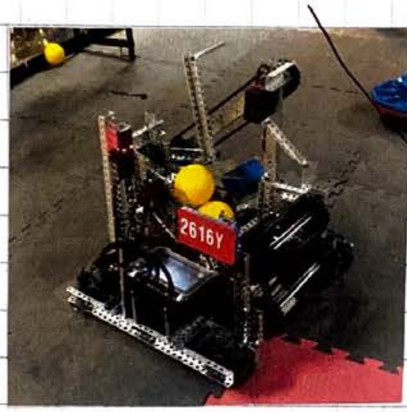
PROPRIETARY INFORMATION

Continued from page

MODIFYING ROLLER & CAP INTAKES

To improve our efficiency in flipping caps, we decided to remove our cap flipper completely and modify our roller intake so that it can not only take in balls, but also roll to the extent that it can flip caps. We see this modification helping us drive the robot better and easier because our driver does not need to use more buttons on the joystick to control the robot's functions. Instead, she can just use one button to flip caps and take in balls by using the roller to serve as a mechanism for caps and balls. This will make it much easier and accessible during competitions when the driver is under pressure and thus will help us not waste time and score more points. In theory, the rollers will run while the driver runs the robot into the caps to flip them.

In order to modify the roller intake to be able to flip the caps, we need to lower the rollers down on the robot so that it can reach the caps. To do this, we dropped the bottom roller one hole on the c-channel. Then, we drilled holes into the metal and dropped the bearings one hole as well. Next, we made the roller longer by adding spacers in between the two sprockets. Now, the roller can flip the caps because it has been dropped. Testing out our new and refined roller/cap intake, we find that our driver has an easier time driving the robot and flipping caps. She has to drive the robot to run fast into the caps, however, to make sure they are flipped. Still, we find that our decision to remove the cap intake from our previous design was beneficial to maintain our performance.



Problem: The robot is moving and spinning in a circle by itself (we had this problem earlier in the season).

Solution: We increased the threshold in the code.

Continued to page

SIGNATURE [signature] DATE 1/15/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 1-15-19

PROPRIETARY INFORMATION

Continued from page

UPDATED SKILLS AUTON

We want to change our skills autonomous in order to score more points for the skills challenge, and we feel that our robot is capable of accomplishing many tasks during the 60 second long skills challenge. So, after strategizing the best route for the robot, we came up with the following autonomous skills plan:

1) Move straight to get the ball from the cap located towards the middle of the field (2 total balls in catapult) and flip cap - 1 point

2) Move back to starting position and shoot at top 2 high flags - 4 points

3) Move straight into bottom low flag - 1 point

4) Back up and turn right to park onto center platform - 6 points

```
//moves forward picks up ball
float revsToBall = -3.200;
LFront.startRotateFor(revsToBall, revolutions);
RFront.startRotateFor(revsToBall, revolutions);
LBack.startRotateFor(revsToBall, revolutions);
Rollers.startRotateFor(-1*revsToBall, revolutions);
RBack.rotateFor(revsToBall, revolutions);

LFront.setVelocity(180, velocityUnits::rpm);
LBack.setVelocity(180, velocityUnits::rpm);
RFront.setVelocity(180, velocityUnits::rpm);
RBack.setVelocity(180, velocityUnits::rpm);

//rollers spin some more
float roll = 3.4;
Rollers.rotateFor(roll, revolutions);
vex::task::sleep(250);

//moves up
float revsUp = -0.90;
float rollFlip = -2.80;
Rollers.startRotateFor(rollFlip, revolutions);
LFront.startRotateFor(revsUp, revolutions);
RFront.startRotateFor(revsUp, revolutions);
LBack.startRotateFor(revsUp, revolutions);
RBack.rotateFor(revsUp, revolutions);
vex::task::sleep(50);

LFront.setVelocity(130, velocityUnits::rpm);
LBack.setVelocity(130, velocityUnits::rpm);
RFront.setVelocity(130, velocityUnits::rpm);
RBack.setVelocity(130, velocityUnits::rpm);
```

With this autonomous skills plan, we expect to score 12 points, which is significantly an improvement from 3 points in our last competition. This will help us rise in rankings seeing how the highest autonomous skills was 11 points last competition. While our previous skills auton aimed to follow this path, we ran into multiple problems that we fixed in our code. We made additions to ensure that the angled cap with the ball under would be flipped to score an additional point, since all game elements are set to blue during skills. We realized that a lack of speed prevented us from flipping the cap, so robot velocity is increased after picking up the ball under the cap using the setVelocity function, as shown in the highlighted lines. The robot then drives forward with rollers spinning in the reverse direction, successfully flipping the cap. The speed is then moved back to normal ~~force~~ before running through the rest of the auton.

Continued to page

SIGNATURE

DATE 1/17/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1-17-19

Continued from page

NIGHT BEFORE COMPETITION

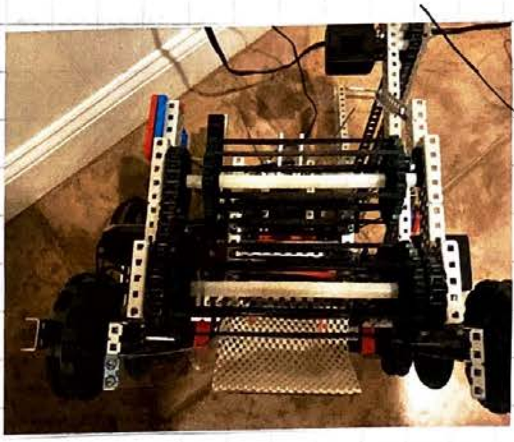
With tomorrow as our third competition, of the season at Camden County Technical School, we strive to be more comfortable with driving and we want to minimize any problems we could encounter tomorrow. So, we scrimmaged the entire afternoon and evening to get the practice we need on the field, and also to see if there are any problems we can fix before the competition. In the end, the best teams are those that are very comfortable with driving their robot and can smoothly maneuver it throughout the field, which comes from driving practice and scrimmages with other teams. We also tested our autons and refined them along the way. While practicing shooting balls, we had to change out some rubber bands to increase the tension since some of the balls were not reaching the flags. One thing that we need to do the morning of the competition before qualification matches is to tighten the motors since loose motors affect our robot's performance, and we want to do anything in our control to ensure a successful day.

Problem: The plexiglas on the drive, to guide balls into roller intake, cracked.

Solution: We replaced the cracked piece by creating a new one.

Problem: One side of the drive is going slower than the other side, which hinders the robot from moving in a straight line.

Solution: We reset the robot by resetting the brain.



Continued to page

SIGNATURE

DATE 1/18/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1-18-19

Continued from page

DAY OF COMPETITION

As soon as we got to the competition at camden county Technical Schools, we went to the practice fields to test out our skills auton. We programmed our skills auton to grab the ball from the cap, shoot at the top two flags, move forward to toggle the low flag, then back up to park onto the center platform. However, we found that our skills auton was not very consistent and reliable as it would sometimes miss the top two flags, or the second ball would not be placed in the preferred position in the catapult to shoot successfully when the robot grabbed it from the cap. Because of these inconsistencies, we chose to go with our center parking auton to play it safe and be guaranteed 6 AP points in skills. Additionally, we figured we would play it safe with our 6 point auton since we planned on running driver skills twice, thus we wanted to use the code that would guarantee us points rather than one that could not. With 6 AP skills points, we were able to get a total of 17 driver points by toggling a column of flags, all the bottom flags, flipping 4 caps, and parking in the center platform. However, during the second skills run, our controller disconnected briefly, and we were unable to toggle the middle flags. We believe our score would have been easily higher if not for the disconnection, and ended up 4th in skills, with a score of 23 points.

Throughout the competition, we observed robots of varying development, which is a tell-tale of how teams in NJ are doing thus far in the season. For our first and second matches, we encountered something we never expected: our catapult shot out towards us in the drive team box during the autonomous period, hitting our driver in the process! While we did not find a definite solution to explain this phenomena, we tested it out multiple times with a practice match wire to ensure it did not shoot during auton period.

Problem: Balls keep getting stuck in robot when they are being fed into catapult.

Solution: We did not find a permanent solution, but had to be cautious or did not load a preloader

Overall, we were successful pushing other robots off the center platform within the last seconds of the match. We ended qualifications 5-3-0, and were picked by the 5th alliance, but lost in quarterfinals against the 4th alliance.

Continued to page

SIGNATURE

DATE 1/19/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1-19-19

Continued from page

REFLECTION OF COMPETITION

Even though we were not able to carry out our planned skills auton, we are satisfied with our performance in skills regardless. Our goal for every competition is to improve in skills, and thus increased by 6 points from our last competition. Yet, we still aim to code a skills autonomous program that will guarantee more AP points for skills. We choose to do this because many teams do not have high AP points, so we can be ranked high for skills even if we do not have extremely high driver points. Because our previous skills plan could not secure us the 12 AP points due to many factors that could hinder it from accurately hitting the flags, such as positioning, we want to create a new auton that is more certain in scoring points. Also, since shooting flags seemed to be a factor that causes our auton to fluctuate in toggling the flags, we may reconsider our autonomous skills goal/objective. Not only do we want to improve our skills auton, but we also want to improve our driving for skills. Besides the disconnecting issue we had, our robot driving should be improving everytime we are on the field, so we know we can go for more points.

This competition also showed us that our robot design is generally effective and that stacking caps is still not a top priority for Turning Point. We still saw very few robots go for cap stacking on the posts, which is currently a task we do not plan on accomplishing during matches. However, we want to rebuild some aspects of our robot, so we can score points more easily. For example, we plan on designing a way for the robot to toggle the bottom flag better by adding more structures to the sides of the robot. We also noticed that some parts were run down due to constant use and overuse, so we must refine those for the next competition.

Continued to page

SIGNATURE

DATE 1/20/19

DISCLOSED TO AND UNDERSTOOD BY

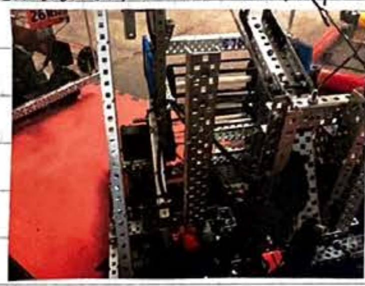
DATE 1-20-19

PROPRIETARY INFORMATION

Continued from page

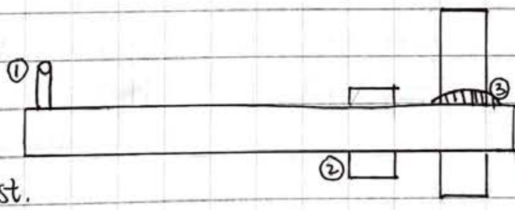
CHANGES TO ROBOT DESIGN

After the competition, we noticed some improvements that must be made to the robot in order to make it more efficient. First off, the robot took a long time to hit the flags. No part of the robot was tall enough to hit the bottom flag except for the catapult, so we had to hit the flag at an angle, which sometimes did not end up turning the flag and also wasted time. In order to fix this, we are adding 2 c-channels vertically at the front of the robot on the catapult side so that we are easily able to toggle the flag. On the rollers side of the robot, we will add a c-channel across the top for stability and also so that we can add standoffs to reach the low flag.



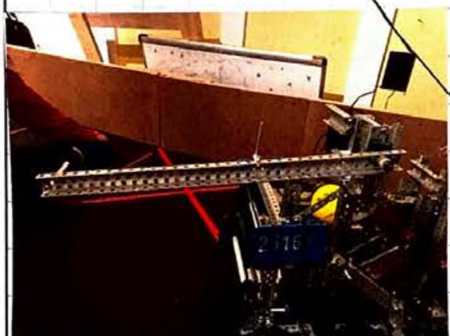
Another issue we had at the competition was that balls would get stuck in between our descorer and catapult. This would inevitably lead to a ball stuck in the robot base, which did not allow the catapult to shoot. To solve this problem, we are redesigning our descorer so that there is more room for the balls to slide out of the robot. On top of this, we will reconfigure the rubber band web so that balls will not get stuck in the robot.

① The standoff at the end of the descorer allows us to physically flip the cap off the post.

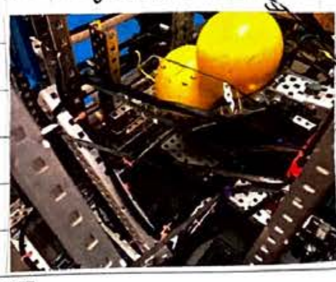


② This c-channel stops the descorer c-channel so that it is easy to drive up the post and flick off the cap.

③ We will use a 1:1 gear ratio with a torque motor since it is fast, but still has enough torque from the motor.



To the right depicts our updated rubber band web to prevent balls from getting stuck →



SIGNATURE

DATE 1/22/19

DISCLOSED TO AND UNDERSTOOD BY

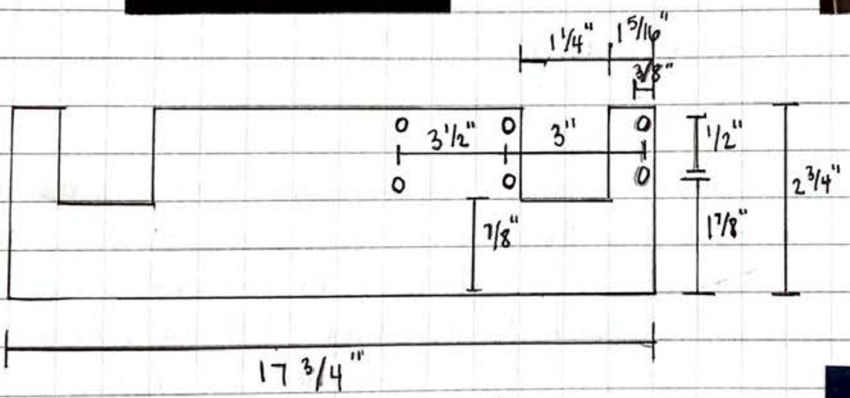
DATE 1-22-19

PROPRIETARY INFORMATION

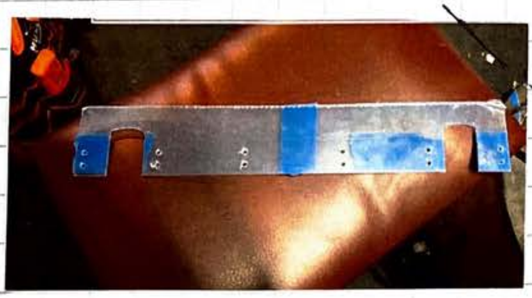
Continued from page

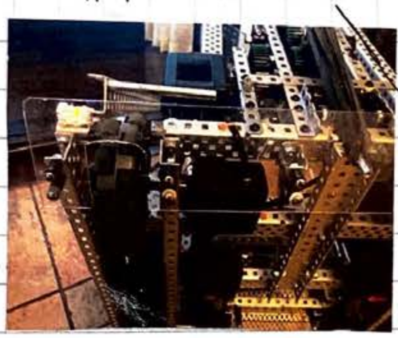
FIXES FROM OVERUSE

The plexiglass ramps, which allow the robot to easily slide onto the parking platform, cracked during the competition. As a result, we are going to redesign the design and find better ways to attach the plexiglass to prevent cracking. Since the robot had to be square with the platform in order to climb, we will now extend the design all the way across the front of the robot so that at any angle, the robot will be able to climb the platform. In order to prevent cracking in the future, we will have additional connection points from the robot to the plexiglass, which strengthens the design and makes the plexiglass less flexible. We do not want the plexiglass to be very flexible and flimsy because it will not allow the robot to securely maneuver up the platform. This updated design will allow for more time to score points in matches because it will take less time to get up the platform, and when there is little time left in the match, we can get onto the platforms quickly.



To the left, we have sketched out our design plan for our redesigned plexiglass ramp. We measured each measurement of the piece to help us make the appropriate cuts and drills for holes.

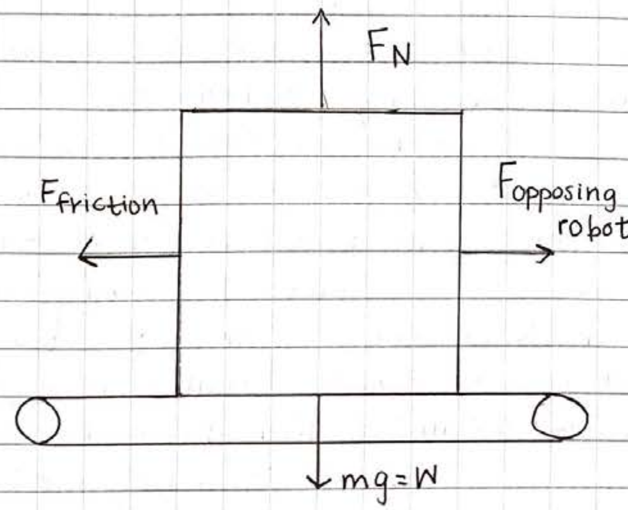




Continued to page

Continued from page

FORCE TO PUSH ROBOT OFF PLATFORMS



In order to calculate the approximate force required to push our robot off of either the center or the two side alliance platforms, we needed to solve for our force due to static friction. The opposing robot will need to overcome the force due to static friction so as to successfully push our robot off of a platform. The coefficient of friction between the polycarbonate center platform and the TPR rubber of our omni-directional wheels was found to be 0.31. Using that value, we plugged it in accordingly:

12.5 lbs = Weight of Robot

$$F_{opposing\ robot} = F_{friction}$$

$$F_{opposing\ robot} = (\mu)(F_n) = (\mu)(m)(g)$$

$$F_{opposing\ robot} = (0.31)(5.6699\ kg)(9.81\ m/s^2)$$

$$F_{opposing\ robot} = 17.2401\ N$$

$$12.5\ lbs \times \frac{0.453592\ kg}{1\ lb} = 5.6699\ kg$$

The force required for opposing robots to push our robot off of the center or the two alliance platforms was calculated to be 17.2401 N, as shown above.

Continued to page

SIGNATURE [signature]

DATE 1/23/19

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 1/23/19

PROPRIETARY INFORMATION

THE ROBOT'S CENTER OF GRAVITY

Unlike last year's game in which tipping was a very common problem with multiple cones stacked upon the mobile goal, this year's game does not require an anti-tip structure as the robot is not required to carry multiple game objects at a time. Previous robots also needed anti-tip structures in order to counteract the effects of tall lifts. Also, a lift like a four bar or a six bar or a chain bar, which move in a circular path are not necessary for this game, which ensures no excess angular and circular momentum that would otherwise cause the robot to tip. The current robot's mass is equally distributed throughout the robot. When the weight of the one or two balls is applied to the robot, the ball's mass is not enough to greatly affect the overall speed and effectiveness of the robot. The weight of each ball is approximately 0.12 lbs, or 55 g.

Robot = ~~#~~12.5 lbs.
One ball = 0.12 lbs.
One cap = 335 g = 0.74 lbs.
Two balls = 0.24 lbs.

$$\% \text{ of Change} = \frac{\text{weight of ball(s)}}{\text{weight of robot}} \times 100$$

$$\% \text{ of Change for One Ball} = \frac{0.12 \text{ lbs}}{12.5 \text{ lbs}} \times 100 = 0.96\%$$

One ball's weight has a 0.96% effect of weight on robot's speed.

$$\% \text{ of Change} = \frac{\text{Weight of one cap}}{\text{weight of robot}} \times 100$$

$$\% \text{ of Change} = \frac{0.74 \text{ lbs.}}{12.5 \text{ lbs.}} \times 100 = 5.92\%$$

One cap's weight has 5.92% effect of weight on robot's speed, which is not significant to create an anti-tip structure.

$$\% \text{ of Change for Two Balls} = \frac{0.24 \text{ lbs.}}{12.5 \text{ lbs.}} \times 100 = 1.92\%$$

Two balls have a 1.92% effect of weight on robot's speed.

Continued to page

SIGNATURE [signature] DATE 1/23/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 1/23/19

Continued from page

AUTONS

It is advantageous to possess multiple autons for competitions because we need to be able to maximize our scored points by having back and front autons for both alliance colors. Having a variety of autons that can cover all aspects of the field as well as coordinate with our alliance partners' autonomous paths is a critical component to our game strategy since it can help us obtain the autonomous bonus. Therefore, we created 4 autons for the 15 second autonomous period for each side of the field. With many options for the autonomous period, we can take a run at our auton simultaneously as our alliance partner, and we can both take a shot at getting more points than our opponents. Below, we have outlined all of our autons that we can use in matches.

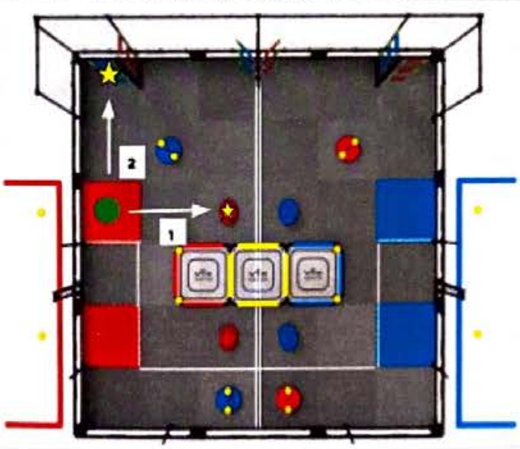
KEY:

Starting Position

Scored

Direction of robot's pathway

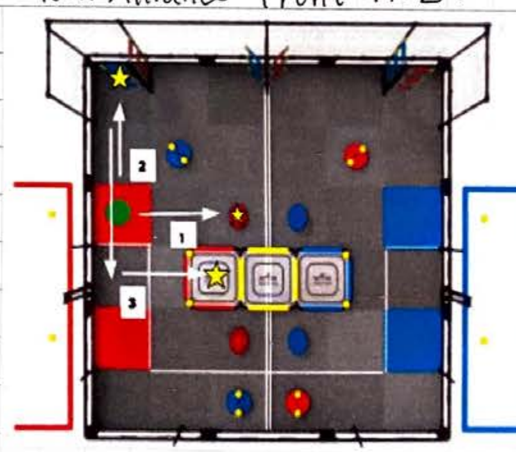
Red Alliance Front #1:



1. The robot will first get a ball and score a cap
2. The robot will shoot and score two high flags.
3. The robot will drive and toggle the low flag.

TOTAL: 6 points

Red Alliance Front #2:



1. Robot will get a ball and score cap.
2. Robot will score two high flags and low.
3. Robot will park on alliance platform.

TOTAL: 9 points.

Continued to page

SIGNATURE

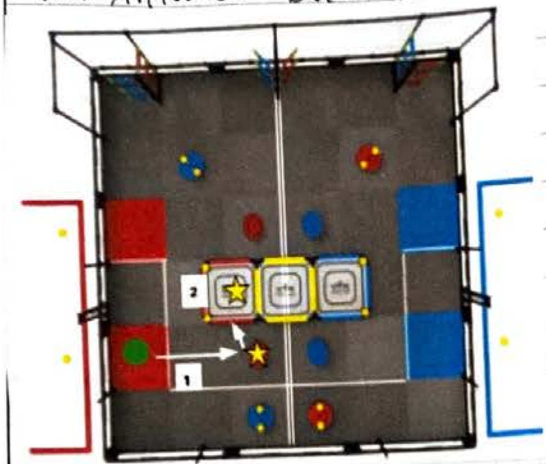
DATE 1/24/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1/24/19

AUTONS (CNTD.)

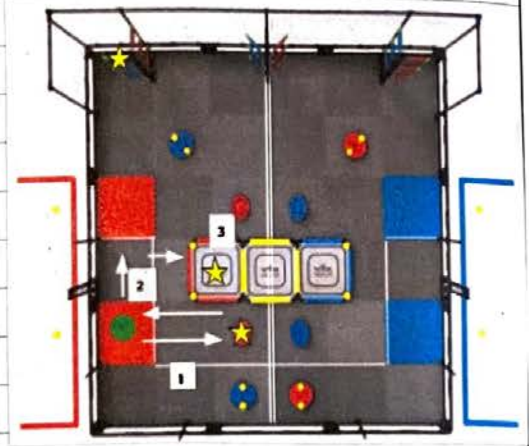
Red Alliance Back #1:



1. Robot will get a ball and score a cap.
2. Robot will park on platform.

TOTAL: 4 points

Red Alliance Back #2:



1. Robot scores cap and gets ball.
2. Robot returns to starting position and shoots at top or middle flag or both.
3. Robot parks on alliance platform.

TOTAL: Min: 6 ; Max: 8

The four autonomous strategies that we have depicted here and in the previous page are the same as the four autons for the blue alliance side of the field, which we have not illustrated. In this way, we can be prepared for any autonomous path.

```
//turn
float turnShoot = 0.27;
LFront.startRotateFor(turnShoot, revolutions);
LBack.startRotateFor(turnShoot, revolutions);
RBack.startRotateFor(turnShoot*-1, revolutions);
RFront.rotateFor(turnShoot*-1, revolutions);
vex::task::sleep(700);

float catapultFling = 0.7;
Catapult.rotateFor(catapultFling, revolutions);
vex::task::sleep(250);

float turnPlat = (0.8 - turnShoot);
LFront.startRotateFor(turnPlat, revolutions);
LBack.startRotateFor(turnPlat, revolutions);
RBack.startRotateFor(turnPlat*-1, revolutions);
```

Since this auton shoots the middle column of flags, the robot needs to turn to a certain position in order to accurately shoot the middle and top flags. In order to determine this, we drove to the position that allowed us to accurately shoot at the middle column and measured the angle we were turned at: 30 degrees.

0.8 revolutions turns the robot 90°, so we divided 30/90 and multiplied by 0.8 to calculate the number of revolutions to turn the robot 30°: about 0.27 revolutions. 0.8 revolutions gets the robot onto the platform, so we subtracted 0.27 from 0.8, so our total turn would be 90°, guaranteeing the park. This was executed in our code by subtracting the value in turnShoot from 0.8, in case we need to change degrees rotated to shoot the middle flags.

Continued to page

SIGNATURE [signature] DATE 1/24/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 1-24-19

Continued from page

UPDATED SKILLS AUTON

Our last skills auton was inconsistent in guaranteeing our expected scored points, so we speculate that the reason for that inconsistency was due to our focus on scoring flags. To be able to toggle flags successfully, the robot must be positioned in a specific position. Thus, we could not rely on our previous skills auton for consistent results. To get more consistent results, we will turn our focus from flags to flipping caps. We have more confidence in accurately scoring by flipping caps as opposed to toggling the high flags. Thus, we plan on flipping all of the caps on the left side of the field, and we can ensure these points by utilizing the wall as a guide to maintain the correct angling and positioning of the robot as it moves across the field to reach the remaining caps. Our plan for autonomous skills is as follows:

1) Flip first cap.

2) Move back to starting position and shoot top two flags.

3) Move forward and right to flip second cap.

4) Move back towards wall.

5) Continue to move back towards other end of field.

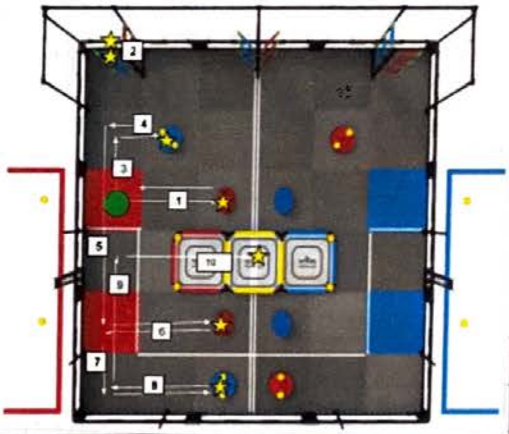
6) Move to flip third cap and move back.

7) Move back to position for fourth cap.

8) Move right to flip fourth cap and move back to wall.

9) Move forward towards parking platforms.

10) Move right to park on center platform.



TOTAL: 14 POINTS

Continued to page

SIGNATURE

DATE 1/26/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1-26-19

PROPRIETARY INFORMATION

Continued from page

AIR DRAG CALCULATIONS

The aerodynamic drag on an object depends on several factors, including the shape, size, inclination, and flow conditions. All of these factors are related to the value of the drag through the drag equation:

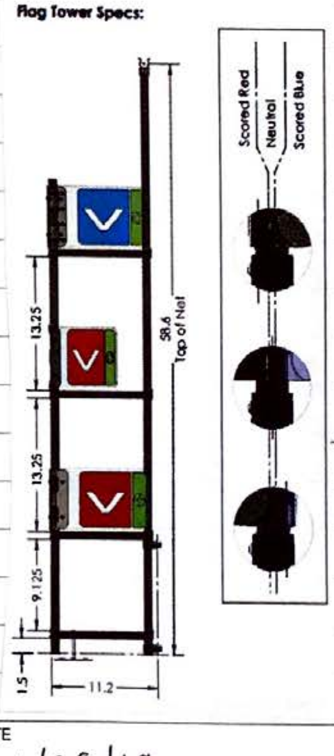
$$D = (Cd)(0.5)(\rho)(V^2)(A)$$

Where D is equal to the drag, ρ is the air density, V is the velocity, A is the reference area, and Cd is the drag coefficient.

In the following graphs, we showcased the trajectories for the ball's motion with and without air drag affecting the ball's motion. In order to create the graphs, we needed to take into account various factors:

- The angle at which balls were launched ($45°$)
- The initial velocity of the launch (calculated to be 1.069 m/s, a value we converted from 200 revolutions per minute value inputted into VEX Coding Studio)
- Mass of the projectile (one ball = 0.055 kg)
- Acceleration due to gravity (on Earth $-9.81\ m/s^2$)
- Density of the fluid through which the projectile travels (density of air = $1.225\ kg/m^3$)
- Projected surface area of the projectile in y-direction
  - Surface area of sphere = $4\pi r^2$
    $= 4(3.14)(0.0381)^2 = 0.0182\ m^2$
  - Diameter of one yellow ball = 3 in. = 0.0762 m
  - Radius of one yellow ball = 0.0762 m / 2 = 0.0381 m
- Projected surface area of the projectile in x-direction
  - Surface area of a sphere = $4\pi r^2$
    $= 4(3.14)(0.0381)^2 = 0.0182\ m^2$
  - Diameter of one yellow ball = 3 in. = 0.0762 m
  - Radius of one yellow ball = 0.0762 m / 2 = 0.0381 m
- Drag coefficient of the projectile in y-direction
- Drag coefficient of the projectile in x-direction

Flag Tower Specs:



SIGNATURE

DATE 1/29/19

DISCLOSED TO AND UNDERSTOOD BY

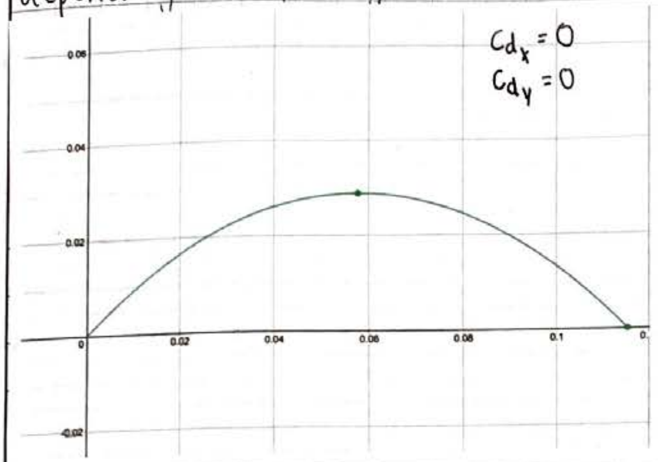
DATE 1-29-19

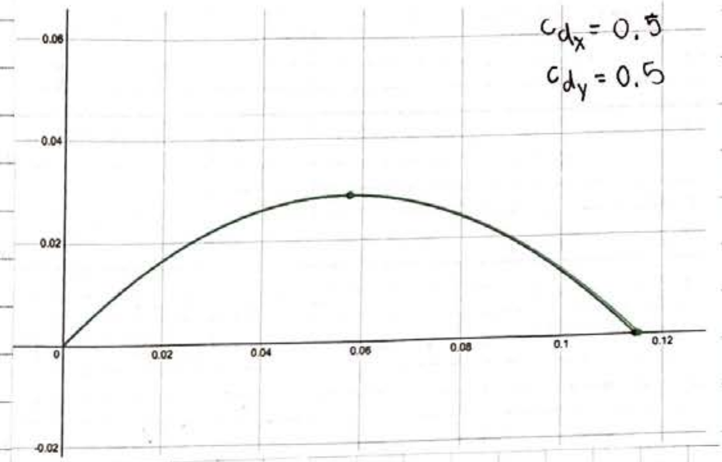
PROPRIETARY INFORMATION

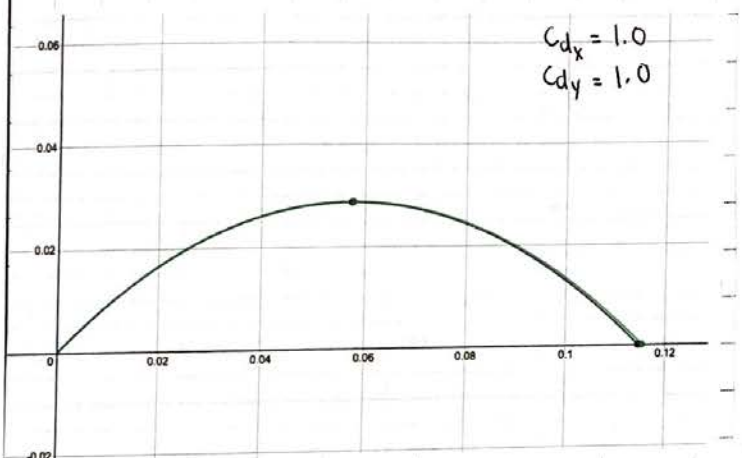
Continued from page

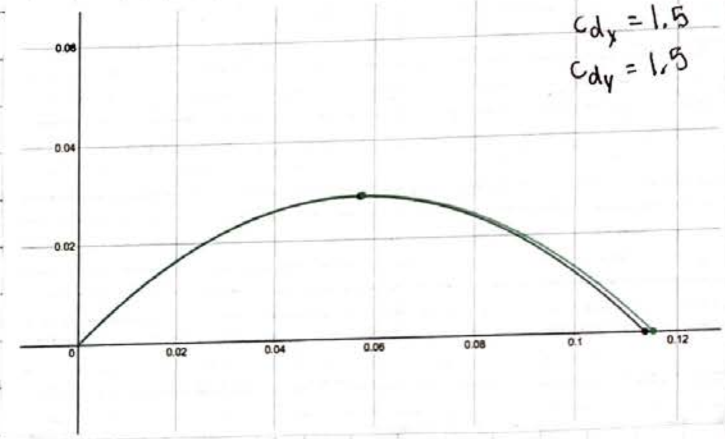
AIR DRAG CALCULATIONS (CNTD.)

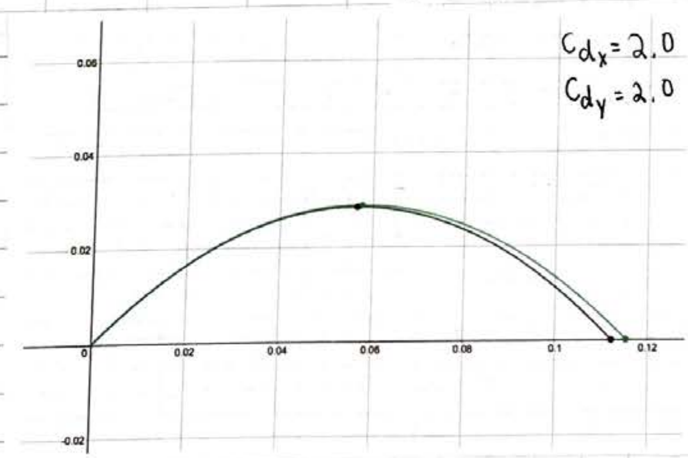
Since the drag coefficient cannot be algebraically determined, we opted to try out various experimental values of the coefficient to see how the final trajectory of the motion of the ball would be impacted depending on the given coefficient of drag value.











Continued to page

SIGNATURE

DATE 1/29/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 1-29-19

PROPRIETARY INFORMATION

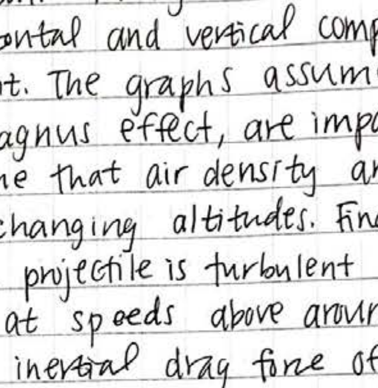
Continued from page

AIR DRAG CALCULATIONS (CNTD.)

Even when the coefficient of drag is set to 2, a significant increase from reality, what we notice is that air drag does not have a significant impact on the ball's trajectory regardless of what the actual coefficient of air drag is. We can assume that this is due in part by the smaller surface area of the balls and the relatively higher mass of the balls.

Of course, these graphs are still not perfectly exact as certain assumptions were made in their creation. For example, the graph assumes that the projectile's orientation is static throughout its flight (that the balls do not rotate in mid-air). The graph also assumes that the horizontal and vertical components of the drag force of the balls throughout the flight will appropriately match with the corresponding horizontal and vertical components of the velocity of the balls throughout the flight. The graphs assume that no other physical phenomena, such as the Magnus effect, are impacting the trajectory of the balls. The graphs also assume that air density and the gravitational force remain constant, even at changing altitudes. Finally, the graph assumes that the flow of air behind the projectile is turbulent and not laminar. For objects in the air, this holds true at speeds above around 4 m/s. By extension, the graph assumes that the inertial drag force of the fluid medium is far more significant than the viscous drag force, and the graph assumes that the drag coefficient of an object does not vary with velocity.

Even so, these graphs offer a far better approximation than a simple parabola.



Continued to page

Continued from page

SIGNIFICANCE OF DRAG

In order to determine the significance of air drag to decide if we should devote a lot of time to account for it, we used a paired t-test. A paired t-test compares data on the same subject by calculating a t-statistic using the formula:

$$t = \frac{(\bar{x} - \mu_0)}{(s/\sqrt{n})}$$

It then locates the t-statistic on a t-distribution at the specified degrees of freedom (df = n-1), finding a p-value, or probability value. We compare this p-value to a significance level of $\alpha = 0.10$, meaning that if the p-value is greater than this, we will not have convincing evidence to reject H_0, since that would indicate that the observed differences in height were likely due to chance.

In this case, we want to test if air drag has a statistically significant effect on the trajectory of the ball. To do this, we selected nine horizontal distances on the trajectory graph, and listed the height of the ball at that distance both with and without air drag taken into consideration. Since the ball is relatively small and compact, we hypothesize that air drag will not have a statistically significant effect on the ball's trajectory. Thus, we can say that our null hypothesis, H_0, is the mean difference in the ball's height at a given distance with and without air drag would be 0. The alternative hypothesis would have a mean difference greater than 0. This can be written as:

$$H_0 : \mu_d = 0$$
$$H_a : \mu_d = 0$$

With 9 data values and using the t-statistic formula, t = 1.3661. Since there are 9 data values, we use the t-distribution at 8 degrees of freedom, obtaining a two-tailed p-value of 0.2091, which is greater than the selected significance level $\alpha = 0.10$, indicating that the differences in height and trajectory were not significantly affected by air drag — the effect of air drag is not statistically significant. Thus, we cannot conclude that air drag significantly affects the ball's trajectory, allowing us not to have to account for it in further calculations and game strategy.

Continued to page

SIGNATURE [signature] DATE 1/31/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 1/31/19

Continued from page

DAY BEFORE COMPETITION

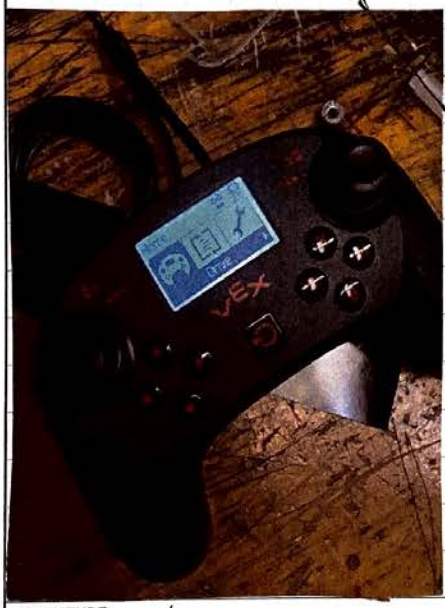
We decided to switch to dual drivers for the rest of the season. This way, our main driver does not have to focus on retracting the catapult down while driving. With single drive, our driver had to stop driving to pull the catapult down, wasting time and allowing our opponents to score more points. Dual driving allows the second driver to pull the catapult down, allocating more time to drive and increasing efficiency, ultimately making it easy for us to gain more points.

```
//if else for catapult launcher
if (Controller1.ButtonL2.pressing() || Controller2.ButtonL2.pressing()){
    Catapult.spin(directionType::fwd,200,velocityUnits::rpm);
 }

else { Catapult.spin(directionType::rev,0,velocityUnits::rpm);

    }
```

We had to account for the new dual drive strategy in our usercontrol code. We added the second controller object into the program, named Controller2, and used this addition in the code for the catapult launcher, since that was the purpose of the second controller. We used the "||", the "or" statement, to indicate that when the first or second controller has button L2 pressed, the catapult would retract. This allows the second controller to retract the catapult, but does not remove the retraction ability from the first controller, in case our main driver needs to use it unexpectedly.



For tomorrow's competition at Millville, one of our goals is to execute a successful skills run, in both autonomous and driver runs, especially since we changed to dual driving. Because we switched to dual driving today, we made sure to practice all we could so that both of our drivers were comfortable with their controls. We also received another battery, which is extremely beneficial since at competitions we will not have to worry about running out of battery seconds before a match.

Continued to page

SIGNATURE

DATE 2/1/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2/1/19

Continued from page

DAY OF COMPETITION

Upon entering the competition in Millville, we got our robot inspected and turned our notebook in. We then went onto the practice fields for our two drivers to practice their driving collaboration since we just switched over to dual controls. After a period of practice, we went over to do our driverskills run, but was only able to score 8 points. Because we scored so low, we reasoned that we must go back to the practice field and work on better communication and cooperation with one driver putting down the catapult in sync with the other driver feeding and launching balls. Then, we went back to driver skills, but still were unable to score more than 8 points. Because we were given 3 tries for both driver and autonomous, we stategized on a plan to score points for our last driver skills run. We planned on toggling the first column of flags, all bottom flags, flipping all 4 caps towards the front of the field, and finally parking on the center platform. We executed this strategy successfully, and got a driver skills score of 17 points!

For our autonomous skills, our original plan was to score a total of 14 points (refer to page 75). However, our alignment of the robot for our first run was off, so we were only able to score 1 point with a cap. Since we had two skills runs left, we decided that scoring the two caps in the back of the field would not be our priority since we wanted to guarantee at least a park and venturing to the back may mess with the alignment even more and flipping both caps may not be guaranteed. So, we decided that we would stay in front of the field, and thus slightly adjusted our code, Our robot was programmed to get the ball from a cap, flip the cap, shoot at the two high flags, flip the other cap near the flags, and finally park on the center platform to get a total of 12 points.

SIGNATURE

DATE 2/2/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-2-19

PROPRIETARY INFORMATION

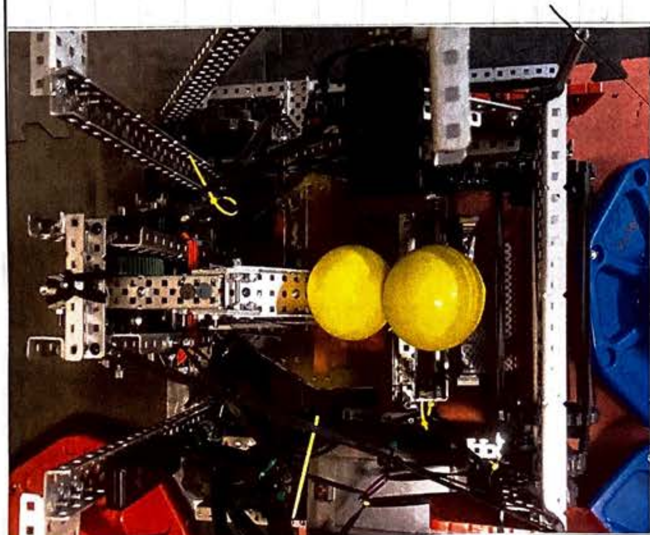
DAY OF COMPETITION (CNTD.)

After running our auton on the practice field with the code modifications, we went to the skills field and ran our auton. We scored 11 points; our robot missed flipping the second cap, thus resulting in only 11 points rather than our expected 12 points. However, for our third skills run, we still could not get the second cap to flip, and ended with an auton score of 11 points. Ultimately, we finished skills with a total of 28 points! For the rest of the competition, we had 8 qualification matches, in which we did not encounter any robot issues, except for one where we disconnected for the entire driver period until the last 18 seconds. Luckily, we were able to park on the center platform and won the match.

Problem: The robot unexpectedly disconnects during matches.

Solution: We kept a screwdriver with the drive team so that we could reset our controllers when needed in order to get the robot connected again.

We ended our qualifications undefeated with a ranking of 3rd place, which has been our best run yet this season! We were selected by the first alliance seed, 2616 J, and were able to get all the way to the finals, but lost 15-16 by the second alliance seed. Even though we were unable to be the competition's tournament champions, we are completely pleased and proud of our performance and teamwork.



To the left, we have a picture that further explains our problem. That is, the second ball does not roll back in place behind the first.

Continued to page

SIGNATURE

DATE 2/2/19

DISCLOSED TO AND UNDERSTOOD BY Cayen Payne

DATE 2-2-19

Continued from page

REFLECTION OF COMPETITION

After the Millville competition, one of the aspects of our robot design we need to adjust is our catapult. That is, we still have the problem where the second ball that is feeded into the catapult lands on top of the first ball instead of behind it, so we cannot effectively shoot the 2 balls at the two high flags accurately. Thus, we must devise and brainstorm a design or modification to fix this problem. Additionally, we came across some instances during matches where a ball got stuck in the robot, which hindered us from utilizing our catapult since the ball got stuck right underneath the catapult. Our rubber band web thus did not prove to be completely effective to prevent the balls from falling into the robot. Another aspect we want to improve upon is while our skills run was our highest run yet, we know we must add to our skills auton and also improve driving in order to have a competitive skills score for the state competition. We are considering mapping out a whole new skills auton that will ensure us the points we expect it to score. For instance, our intended skills run had to be shortened to just score the front caps because once the robot got to the back caps, the alignment was off and we were not getting consistent results. On top of that, the last front cap that the robot was coded to flip was inconsistent as well, and during our actual skills runs, our robot was unable to flip it. Such instances from yesterday encourage us to devise a new skills plan.

We also had some of the highest scores at the end of the competition. Our Offensive Power Ranking (OPR) was 15.5, showing that we have become a stronger team, our Defensive Power Ranking (DPR) was 0.3, which was the lowest score at the competition and shows that we have become a stronger team with a lower value, our Calculated Contribution to Winning Margin (CCWM) was 15.3, which was also the highest at the competition, while our Autonomous Points (AP) was 28, which was tied for second highest and tells us we should improve on our autons since teams are gradually having more advanced autons.

Continued to page

SIGNATURE [signature] DATE 2/3/19

DISCLOSED TO AND UNDERSTOOD BY Cayero Payne DATE 2-3-19

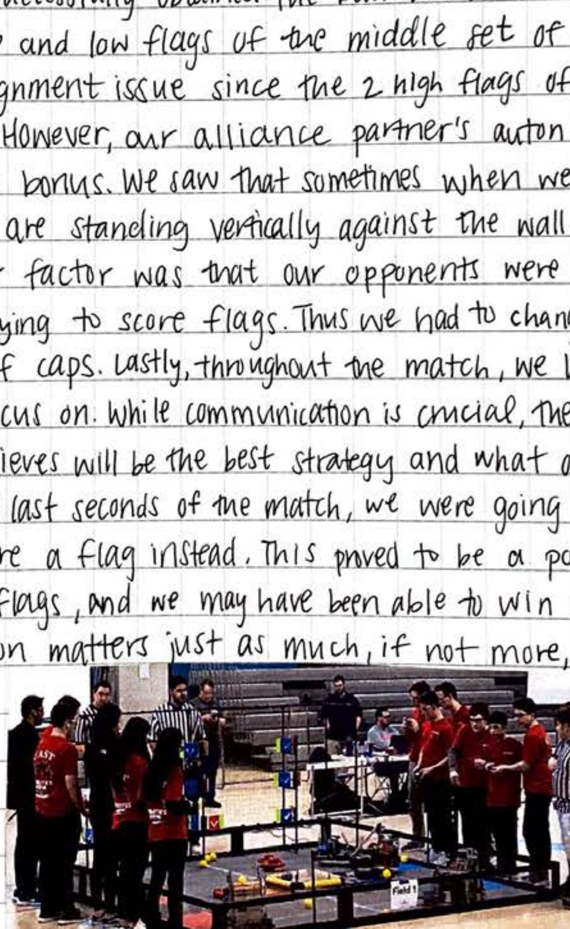
PROPRIETARY INFORMATION

REFLECTION OF COMPETITION (CNTD.)

Every competition we go to, we make sure to observe the types of robots and robot designs competing with us. At Millville, we saw more less developed robots that made it easy for competitors to accidently destroy during a highly defensive match. For example, during many of our matches when all robots go for the center parking, we may have dismantled some parts of robots due to our defensive brakes that allow us to push others out of our way and off the platform. Throughout the season, this braking programming aspect of our robot has been advantageous.

Analysis of Finals Match

We also want to conduct an analysis on our finals match, and factors that may have caused us to lose the match. In order to do this, we have a video of the match than one of our team members recorded, and we will watch it to find things that went wrong and tactics that helped us score more points or defend the other alliance. In the video, our auton successfully obtained the ball while scoring a cap on the right color side, toggled the top and low flags of the middle set of flags, and finally parked. This is most likely an alignment issue since the 2 high flags of the middle set of flags are supposed to be toggled. However, our alliance partner's auton did not work, so we did not get the autonomous bonus. We saw that sometimes when we attempt to flip caps, we cannot flip them if they are standing vertically against the wall of the field, and thus took up valuable time. Another factor was that our opponents were defending our alliance partner while they were trying to score flags. Thus we had to change our goal of the match by scoring flags instead of caps. Lastly, throughout the match, we listened to whatever our partner told us to score/focus on. While communication is crucial, there must be a balance between what our team believes will be the best strategy and what our partner believes to focus on. In other words, during the last seconds of the match, we were going up the center platform, but our partner told us to score a flag instead. This proved to be a poor strategy since parking offers more points than flags, and we may have been able to win with the extra points. We learned that our intuition matters just as much, if not more, than the commands/advice of our alliance partner.



Continued to page

SIGNATURE [signature] DATE 2/3/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 2-3-19

PROPRIETARY INFORMATION

FIXES TO ROBOT DESIGN

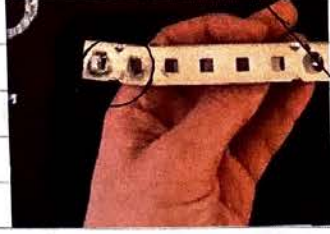
We have noticed that before state competitions, a lot of teams rebuild their entire robots. However, we are mostly happy with our design since it has performed great for the past couple of competitions. Our goal before states is to fine tune some of the minor issues to allow the robot to reach its maximum effeciency.

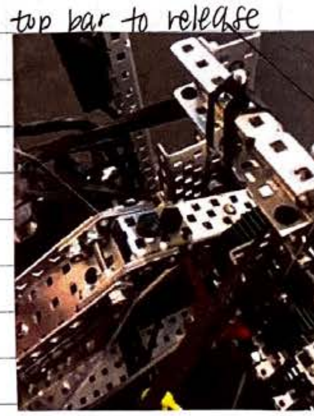
Checklist of Fixes Before States

- ☐ Change out catapult bar
- ☐ Strengthen the front of the ramp to prevent cracking in plexiglas
- ☐ Reclip wire for descorer (cap) to port 18
- ☐ Strengthen the drive
- ☐ Find a solution to keep balls from falling out of catapult
- ☐ Add extension to the low flag toggle
- ☐ Practice!

Changing Out the Catapult Bar

Due to overuse, the bar connected to the gear started to concave. This changed the shooting angle, therefore it is imperative that we change out this bar and enforce the structure so that it will not bend during the next few competitions. One of the reasons that the bar bent was because the place where the catapult hit the top bar to release the balls was placed on a divot in the metal. This weakened the metal, causing the force of the launch to bend the metal.





SIGNATURE

DATE 2/4/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-4-19

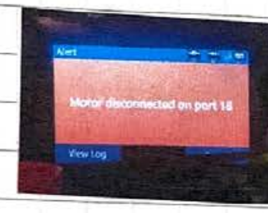
PROPRIETARY INFORMATION

FIXES TO ROBOT DESIGN (CNTD.)

Strengthening the Plexiglass Ramp

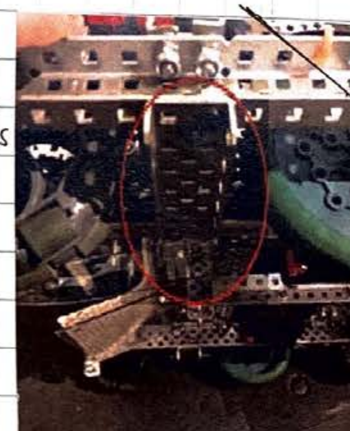
To prevent future cracking, we realized that a lot of the force was exerted onto the outer plexiglass screws. This was cracking the plexiglass around the screws as a result. To fix this issue, we added half of a c-channel to the outer connection points over the cracked plexiglass.

Reclip Wire



The wire that is used for the descorer bar has been losing connection throughout competitions because it is a bad wire and port. To fix this, we will strip the wire lower and reclip the inserts onto the wire. This will allow for better connection from the wire to the cortex.

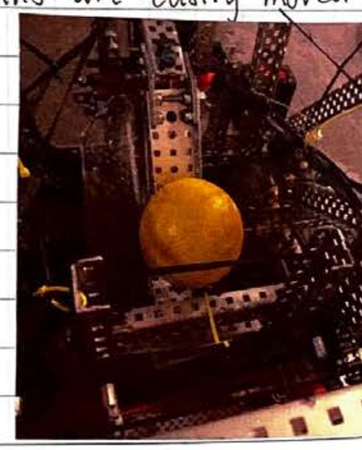
Strengthening the Drive



To strengthen both sides of the drive to prevent a wobbly structure, we added enforcements on the bottom of the drive. The bar across the drive connects the two c-channels on each side of the drive in order to ensure the stability of the overall robot by steadying the base of the robot all around.

Balls Into the catapult

After fixing the angle of the catapult, we were forced to grind off another gear on the slip gear, which allows the catapult to go further back before it shoots. By pulling the catapult further back, the balls are easily moved into place to shoot the double shot. If the catapult can be pushed down all the way, then the second ball will have no problem getting ~~dressed~~ behind the first ball, for the correct alignment.



SIGNATURE

DATE 2/5/19

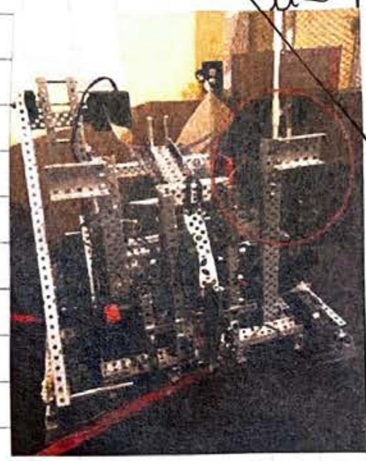
DISCLOSED TO AND UNDERSTOOD BY

DATE 2-5-19

PROPRIETARY INFORMATION

FIXES TO ROBOT DESIGN (CNTD.)

Extension on Lower Flag Toggle



To help during autonomous periods, the extensions on the lower flag toggle will make it easier for the robot to stay in alignment by preventing the base of the robot from hitting or getting caught on the flag pole. These will also help during competitions by helping the drive up to the flags easier as well as quickly driving back out. The extensions can be seen to the right in the red circle.

Practice with Fixed Robot

As we practiced driving with the robot with its fixed and updated design, we definitely saw some improvements. The robot appears stronger and more stable as it is hard for other robots to push us around and we can go up the platforms seamlessly. The extension on the lower flag toggle aids us in getting the low flag toggled, but one thing we noticed was that the robot is a little wide in its dimensions and drive, so when we try to toggle the low flags next to the corners of the field, the robot can instead hit the flag pole. Thus, we have to make sure that the robot is an inch closer to the wall of the field so that we can swiftly move in, toggle the flag, and move out with ease.

Problem: The motors were becoming hot after little use of the drive. One side of the drive seems much faster than the other, so we cannot drive straight.

Solution: Spacers were taken out of the drive because there was too much friction. The motors were working too hard and becoming overheated. The friction wheels as they were positioned was not optimal for ease and fluid movement of the wheels. This was because the spacers placed on the sides of the wheels were creating extra and unnecessary friction. The motors, as a result, needed to exert more force to get the drive to move. To solve this problem, we removed the spacers from their original position so that the motors were not working as hard and thus overheating.

SIGNATURE

DATE 2/5/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-5-19

Continued from page

INERTIA OF WHEELS

For this year's game of "Turning Point," we wanted to calculate a VEX wheel's moment of inertia. With this value, we will know our robot's tendency to resist angular acceleration, which is the rate at which the wheels turn). The mass of a wheel is on its rim, so its moment of inertia about its rotating axis would be $I_{rim} = r^2 m$, where m is mass and r is radius. To determine I_{wheel} experimentally, we used the Parallel Axis Theorem and the dynamics of a pendulum. The Parallel Axis Theorem states that any object rotated about an axis parallel to and a distance, d, from an axis going through the centroid of the object will add an amount equal to md^2 to the moment of inertia about the centroid:

$$I_{parallel} = I_{centroid} + md^2$$

If we theoretically swing the mass (m) about the parallel axis like a pendulum, using the torque from gravity, pulling on the mass, then we must show that the period (T) is related to the distance (d) and moment of inertia.

$$I_{wheel} = I_{parallel} - d^2 m = d^2 m \left(T^2 g / d / (2\pi)^2 - 1\right)$$

Given: $r_{wheel} = 2$ in.
Mass of wheel = 105 g → 0.105 kg
d = 2 in. → 0.0508 m
Avg. period (T) = 0.607 seconds

$$\begin{aligned} I_{wheel} &= d^2 m \left(T^2/d \cdot 9.8 \,/\, (6.28)^2 - 1\right) \\ &= d^2 m \left(0.248 T^2 \,/\, d - 1\right) \\ &= (0.0508)^2 (0.105)(0.248 \cdot 0.668^2 / 0.07 - 1) \\ &= 0.00051 \ \text{kg} \cdot \text{m}^2 \end{aligned}$$

$$\sqrt{I_{wheel}/m} = 0.0533 \text{ m} = 2.1 \text{ m}$$

This value means that the wheel behaves as if the mass is at 84% of the radius of the wheel, and our robot is likely to move with ease and exhibit less resistance to change.

Continued to page

SIGNATURE

DATE 2/6/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-6-19

PROPRIETARY INFORMATION

Continued from page

DOES SPROCKET SIZE IMPACT SPEED?

To evaluate the question, "Does sprocket size impact speed?", we must use calculations to delineate whether there is a relationship between sprocket size and speed. We wanted to calculate this in order to justify our sprocket size. The equations that we must manipulate are:

Circumference = (π)(diameter)

Speed = (circumference)(Theoretical Rotations per minute)

10T Sprockets (0.526" diameter)

C = π(0.526 in.) = 1.65 inches

Speed = (1.65 in.) (200 rpm) = 330.5 in./min. travelled by roller

15T Sprockets (0.75" diameter)

C = π(0.75 in.) = 2.35 inches

Speed = (2.35 in.)(200 rpm) = 471.24 in./min travelled by roller

24T Sprockets (1.18" diameter)

C = π(1.18 in.) = 3.71 inches

Speed = (3.71 in.) (200 rpm) = 741.42 in./min. travelled by roller

36T Sprockets (1.42" diameter)

C = π(1.42 inches) = 4.46 inches

Speed = (4.46 in.) (200 rpm) = 892.21 in./min. travelled by roller

40T Sprockets (1.93" diameter)

C = π(1.93 in.) = 6.06 in.

Speed = (6.06 in.)(200 rpm) = 1212.65 in/min. travelled by roller

48T Sprockets (2.3" diameter)

C = π(2.3 in.) = 7.23 in.

Speed = (7.23")(200 rpm) = 1445.13 in./min. travelled by roller

What does this really mean?

From this, we can conclude that the size of the sprocket used on the rollers does, in fact, matter. Between these equations, the only thing that changes is the coefficient of π, which is represented as the diameter of the sprocket in inches. This may seem confusing since the rpm and gear ratios remain constant. However, as wheel size increases, the effective distance travelled by the rollers also increases. In other words, when analyzing 2 different points on the perimeter of the sprocket, the larger the diameter, the shorter the amount of time needed for the roller to intake balls successfully.

Continued to page

SIGNATURE

DATE 2/6/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-6-19

PROPRIETARY INFORMATION

Continued from page

PROGRAM DESIGN REVISIONS

Problem: Our programming skills code is lengthy and complex, making it difficult for us to even add minor adjustments and also increases the likelihood of careless errors as a result.

Solution: We can create functions for the robot's movements and then implement them into our skills code to simplify making changes and thus improving accuracy.

Creation of Functions

To run pre-autonomous setup functions in VEX Coding Studio, they must be stored under the pre_auton function. By storing the functions we created in the pre_auton, we can use them successfully in the autonomous function as they have already been initialized.

All the functions we created begin with void, as this sets up the function's return type. When an object calls a void function, it carries out the body of that function without returning a value. The functions created for the autonomous skills period are all concerned with the movement of the robot, so they do not ~~to~~ need to return a value.

```
void leftTurn(float revsTurn){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
LFront.startRotateFor(-1*revsTurn, revolutions);
   LBack.startRotateFor(-1*revsTurn, revolutions);
   RFront.startRotateFor(revsTurn, revolutions);
   RBack.rotateFor(revsTurn, revolutions);
   vex::task::sleep(250);
}

void flipCap(float revsFlip){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    Rollers.startRotateFor(revsFlip, revolutions);
    LFront.startRotateFor(revsFlip, revolutions);
    RFront.startRotateFor(revsFlip, revolutions);
    LBack.startRotateFor(revsFlip, revolutions);
    RBack.rotateFor(revsFlip, revolutions);
    vex::task::sleep(50);

}

void pre_auton(void){

}

void robotMove(float revsMove){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    LFront.startRotateFor(revsMove, revolutions);
    RFront.startRotateFor(revsMove, revolutions);
    LBack.startRotateFor(revsMove, revolutions);
    RBack.rotateFor(revsMove, revolutions);
    vex::task::sleep(250);
}

void rightTurn(float revsTurn){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    LFront.startRotateFor(revsTurn, revolutions);
    LBack.startRotateFor(revsTurn, revolutions);
    RFront.startRotateFor(-1*revsTurn, revolutions);
    RBack.rotateFor(-1*revsTurn, revolutions);
    vex::task::sleep(250);
}
```

The robotMove function takes an input of a number of revolutions and rotates all the drive motors for the input number of revolutions. This function allows us to move the robot a certain distance without having to set the number of revolutions for each individual motor.

The rightTurn and leftTurn functions work similarly to robotMove, rotating the drive motors for the input number of revolutions before pausing for 0.25 seconds. However, the rightTurn function reverses the rotation direction of the right drive motors, allowing it to turn right, given that the robot faces

Continued to page

SIGNATURE

DATE 2/6/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-6-19

PROPRIETARY INFORMATION

Continued from page

PROGRAM DESIGN REVISIONS (CNTD.)

forward. The leftTurn function similarly reverses the rotation direction of the left drive motors, turning the robot left.

The ~~shootBall~~ flipCap method flips caps by moving the robot while the roller motor spins for the input number of revolutions before pausing. This function allows us to program the robot to flip a cap with 4 less lines of code.

```
void setDriveVelocity (float velocity){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    LFront.setVelocity(velocity, velocityUnits::rpm);
    LBack.setVelocity(velocity, velocityUnits::rpm);
    RFront.setVelocity(velocity, velocityUnits::rpm);
    RBack.setVelocity(velocity, velocityUnits::rpm);
    Rollers.setVelocity(velocity, velocityUnits::rpm);
    Catapult.setVelocity(velocity, velocityUnits::rpm);
}

void rollers(float roll){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    Rollers.rotateFor(roll, revolutions);
    vex::task::sleep(250);

}

void setRC (float velocity){
    Rollers.setVelocity(velocity, velocityUnits::rpm);
    Catapult.setVelocity(velocity, velocityUnits::rpm);

}


void shootBall(float shoot){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    Catapult.rotateFor(shoot, revolutions);
    vex::task::sleep(250);

}
void catapultBack(float retract){
    vex::rotationUnits revolutions = vex::rotationUnits::rev;
    Catapult.rotateFor(retract, revolutions);
    vex::task::sleep(250);

}
```

The shootBall and catapultBack functions both work very similarly, both rotating the catapult motor for the input number of revolutions to accomplish their goal, both essential moves in our autonomous skills program.

The rollers function rotates the roller motor for the input number of revolutions.

Both the setDriveVelocity and setRC functions set the velocity of the specified motors as the input number. setDriveVelocity allows us to change the velocity of all the robot motors with a single function, which is incredibly useful as this would otherwise take 4+ lines and is used often, particularly to gain the needed speed before flipping a cap. setRC sets the velocity of the rollers and the catapult motors as the input number.

Implementation of Functions

The images of code provided on these pages show the autonomous skills program. They portray how the functions we created are easily implemented in our code, allowing us to make changes to the movement of multiple motors by simply altering one line.

Continued to page

SIGNATURE [signature]

DATE 2/6/19

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 2-6-19

PROPRIETARY INFORMATION

TYPES OF BRAKES

Problem: The catapult keeps going up / shooting on its own when we practice driving.

Solution: To prevent the catapult from shifting up when pulling it back, we decided to implement a brake into the code. Currently, when the catapult is not being pulled back, the button is not being pressed, and the velocity is set to 0. We realized that this was not efficient because it could cause balls to be stuck. The new V5 motors allow for 3 types of brakes to be implemented: brake, hold, and coast.

Coast - allows motors to move freely (equivalent to robot turned off)

Brake - increases the resistance of the motor by shortening the windings together

Hold - keeps motor gears in its exact position by adding electricity from the battery

We decided to test brake and hold as our brakes as they seemed to have the ability to hold the catapult in the same position. We first chose the brake type of 'brake', which was able to hold the catapult down and prevent it from shifting up. However, we ultimately decided to use the brake type of 'hold' because it provides an even greater resistance and allows the driver to put the catapult down smoothly and with ease.

Before:

```
else {
    Catapult.spin(directionType::rev,0,velocityUnits::rpm);
 }
```

After:

```
else {
    Catapult.stop(brakeType::hold);
 }
```

Continued to page

Continued from page

HOW EFFECTIVE ARE OUR BRAKES?

As seen in all of our previous competitions, one of the unique and effective aspects of our design is our coded brake that helps us resist getting pushed off the platform from other robots. Thus, we want to prove that our brake is actually helping us rather than relying on pure observation. To accomplish this, we can use numbers to clearly show the effect of having the brake versus not having the brake. We can conduct an experiment to find the force/weight required to push us off the platform to prove that it is hard to push us off the center platform.

In our code, we implemented the brake type of 'brake' as it "increases the resistance of the motors by shortening the windings together" (according to page 92). It allows for the motors to hold its position, but still allows for it to move when driving, as we implemented the brake within sticks where if we were not moving it, the motor would brake. We chose not to implement the 'hold' brake because when it would slow down the driving as the 4 motors would have a tendency to want to stop due to how strong it was. In addition, during competitions, the brake type of 'brake' was able to withstand our opponents.

```
else {
    LFront.stop(brakeType::brake);
    RFront.stop(brakeType::brake);
    LBack.stop(brakeType::brake);
    RBack.stop(brakeType::brake);
}
```

Continued to page

SIGNATURE

DATE 2|8|19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-8-19

Continued from page

HOW EFFECTIVE ARE OUR BRAKES?

Methods: To prove that our brakes are effective and work to help stay put on the center platform, we carried out the following procedure:

1) Place the robot on the center platform and upload the code without the brake to the robot.
2) Using a spring scale, hook the end with the hook onto the front of the robot and pull the other end just until the robot begins to move from the exerted force.
3) Record the force in Newtons displayed on the scale.
4) Repeat steps 1-3 three times.
5) Repeat steps 1-4 three times, but hook the spring scale onto the side of the robot.
6) Repeat steps 1-5, but upload the code with the brake to the robot, and collect 3 trials of data from the front of the robot and 3 trials from the side of the robot.

Data:

Force with Coded Brake

Trial	Front	Side
1	19 N	9 N
2	20 N	10 N
3	21 N	9 N
AVERAGE	20 N	9.33 N

Force without Coded Brake

Trial	Front	Side
1	12 N	8 N
2	13 N	8 N
3	14 N	8 N
AVERAGE	13 N	8 N

Analysis / Conclusion

Clearly, the coded Brakes provides us with an advantage, seeing how the force required to push us off was 20 N compared to 13 N, for a percent difference of $\frac{|20-13|}{\frac{20+13}{2}} \times 100 = 42.42\%$. On the other hand, our results from the side force yields a percent difference of $\frac{|9.33-8|}{\frac{9.33+8}{2}} \times 100 = 15.35\%$ between the values of 9.33N for the coded brake an 8N without the brake. There is a more impactful effect from the force exerted onto the robot from the front, which is useful when battling for the center.

SIGNATURE

DATE 2/8/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-8-19

PROPRIETARY INFORMATION

Continued from page

FACTORS THAT AFFECT SHOOTING BALLS

Effect of changing Angle of catapult shot on the Trajectory

In the week leading up to the state championship, the drive team was having difficulty with regards to shooting accurately and consistently. Unsure of whether the issue was in the individual driving or in the way our catapult was designed, we decided to conduct an in-depth analysis of the way in which the launch angle and launch velocity will impact the trajectory of the balls. These conclusions apply both when the catapult shoots 2 balls (as a double catapult) and in the situation that we shoot only 1 ball. In our analysis of the effect of changing the angle, we split the 90° angle (drawn below) into 2 separate sections: θ_1 and θ_2. Based upon these 2 angles, we can draw out 8 relationships when changing the angle of the catapult. These relationships would by extension affect the launching angle and trajectory.

Angle	Increase/Decrease Angle	Direction in X/Y	Increase/Decrease in Distance/Height
θ_1	Increase	X-horizontal	Decrease
θ_1	Increase	Y-vertical	Increase
θ_1	Decrease	X-horizontal	Increase
θ_1	Decrease	Y-vertical	Decrease

θ_1 and θ_2 are complementary angles, which essentially means that the 2 angles add up to 90°. As one angle increases, the other decreases, and vice versa. In the tables, we listed how changing each angle would impact the range in both the horizontal and vertical directions.

Angle	Increase/Decrease Angle	Direction in X/Y	Increase/Decrease in Distance/Height
θ_2	Increase	X-horizontal	Increase
θ_2	Increase	Y-vertical	Decrease
θ_2	Decrease	X-horizontal	Decrease
θ_2	Decrease	Y-vertical	Increase

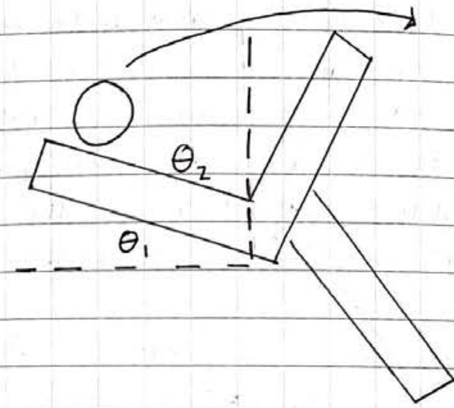
Continued to page

SIGNATURE [signature] DATE 2/9/19

DISCLOSED TO AND UNDERSTOOD BY Cayman Rayne DATE 2-9-19

FACTORS THAT AFFECT SHOOTING BALLS (CNTD.)

The tables on page 95 depict the relationship between the 2 complementary angles used in constructing the catapult. What this essentially proves is that a 45° launch angle provides the maximum range for our motion both in the horizontal and vertical directions, so this is what we pursued. This is because of its relationship with θ_1 and θ_2. When using a 45° angle, both θ_1 and θ_2 are 45°. We attached a c-channel to our bended plexiglass in order to create a very strong launcher that is further secured by approximately 7 rubber bands.

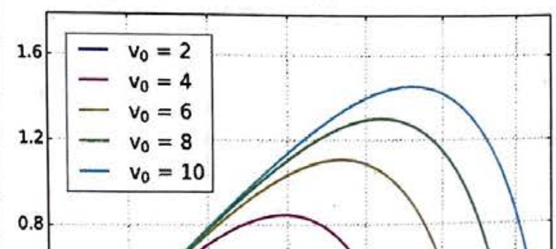


Impact of Varying Initial Velocity of Catapult

Figure 1 below shows the distance that the balls will travel when launched with different initial velocities at a 45° angle. Given the robot's distance from the goal, the speed of the motors attached to the catapult can be changed to control the ball's initial velocity prior to shooting. The main trend we can see from this graph is how as the starting velocity increases, the maximum range in both dimensions increases exponentially. Currently, our launch velocity is set to 200 rpm. Another way to test for the optimal angle is that the distance travelled by the ball will be greatest when $\sin(2\theta)$ is greatest. The sine function reaches its largest output value, 1, with an input angle of 90°, so we can see that for the longest-range trajectories, $2\theta = 90°$ and $\theta = 45°$, as a result.

One important aspect is to take into account the centrifugal force. Centrifugal force is an apparent force that acts outward on a body moving around a center. This force comes from the body's inertia. The centrifugal ~~(directed~~ force (directed towards the center) can cause the projectile to slip out early and thus send it up at a steeper angle. We counteracted this force by putting multiple rubber bands holding the catapult and also implemented the specific stopping location of the catapult after it is fired in code.

Figure 1:



SIGNATURE [signature]

DATE 2/9/19

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 2-9-19

PROPRIETARY INFORMATION

Continued from page

BRAINSTORMING PROGRAMMING SKILLS

Our goal for the state competition is to score at least 16 points in programming to attain a competitive AP and skills score. To accomplish this, we must account for some factors that may affect our program and aspects of the field we can take advantage of to help the auton yield consistent results. One factor is that since balls are used to score the flags, we will use the balls placed underneath the caps since they will be less likely to move, unlike the balls on top of the caps. The balls underneath the caps have better predictability. Like in previous autons, we believe caps may be more ideal to score since they also have better predictability and consistency than gathering balls to shoot at flags. In other words, balls can roll away or they may not be aligned correctly when they are fed through the rollers. Thus, we will shift our auton skills goal to caps because if we focus on collecting balls to shoot at flags, the balls may get in the way and ultimately mess up our alignment throughout the autonomous pathway. In the following pages, we have created programming skills diagrams that are labeled with the various autonomous pathways that we have brainstormed. Each diagram has a ranking out of 5 to indicate how efficient it performs and thus how likely it is that we select the respective autonomous plan to code for the competition. A rank of 1/5 represents unlikely and inefficient while a 5/5 indicates a likely and effective auton that we may choose.

Continued to page

SIGNATURE

DATE 2/9/19

DISCLOSED TO AND UNDERSTOOD BY Cayeo Raupl

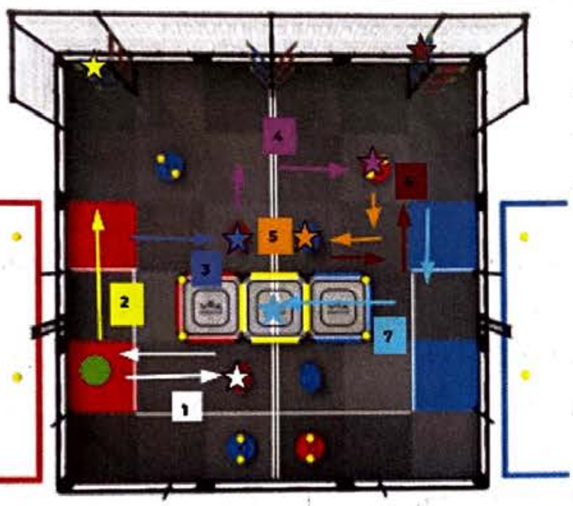
DATE 2-9-19

Continued from page

BRAINSTORMING PROGRAMMING SKILLS (CNTD.)

KEY:
- Starting Position
- Scored
- Direction robot travels in

Each color corresponds to the step, which is also colored, to prevent confusion and ensure the steps are clear.



1. Starting from the back tile, with the rollers facing out, the robot will drive straight and retrieve a ball into the catapult and score the cap to the opposite color and return to starting position.
2. Robot will turn so that the catapult is facing flags and drive forward and shoot, scoring middle and bottom flags.
3. Robot will retrieve a ball and score a cap.
4. Robot will score cap on other end of field.
5. Robot retrieves another ball under a cap, and scores the cap, and then flags
6. Robot parks on center platform.

Rank: 1/5

This strategy would not be efficient due to factors such as time constraints and alignment issues. Our current skills run has our drive set to about 130 rpm, and for our robot to be able to score an entire side of a field, plus half of the other side, we would have to set it at its maximum speed (200 rpm) to complete all tasks, which may cause the robot to go off course and miss scoring certain elements. If our robot also goes off alignment, we would not be able to park, causing us to miss scoring 6 points. Ideally, if our skills does go off alignment, we still would want to guarantee a park at the end for at least 6 points. Thus, we ranked this skills plan a 1 out of 5 since it is not effielent and it is not likely that we choose this strategy for skills.

Continued to page

SIGNATURE

DATE 2/9/19

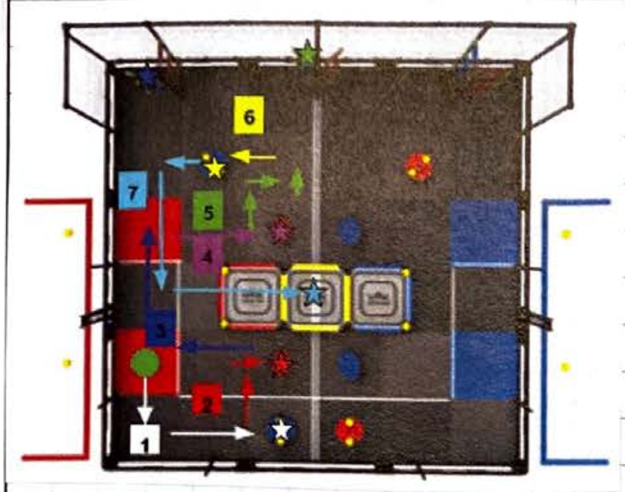
DISCLOSED TO AND UNDERSTOOD BY

DATE 2-9-19

Continued from page

BRAINSTORMING PROGRAMMING

SKILLS (CNTD.)



1. Move forward towards back of field, turn and score cap.
2. Score another cap while getting the ball from underneath.
3. Get to ball shooting position, and shoot at high flag.
4. Turn and score a cap while getting ball from underneath.
5. Turn so that catapult faces flags and shoot at high flag.
6. Turn and score cap by flipping.
7. Park on center platform.

Rank: 2/5

This skills run would be able to work because it only scores on the half of the field. So, we would be able to set the drive velocity to a slower rate to allow the robot to score the elements accurately. However, in step 6, after we flip the caps with the balls on top, the balls may fall into a position where it would set us off course and/or get in our way. As a result, step 7 may be inconsistent with what was diagrammed above, and we may not be able to park on the center platform. We gave this strategy a ranking of 2 out of 5 since it is slightly more efficient and consistent with its results compared to that of the previous strategy on page 98. However, it will be hard to accomplish all the tasks seamlessly to guarantee our goal of scoring 16 autonomous points, at least.

Continued to page

SIGNATURE

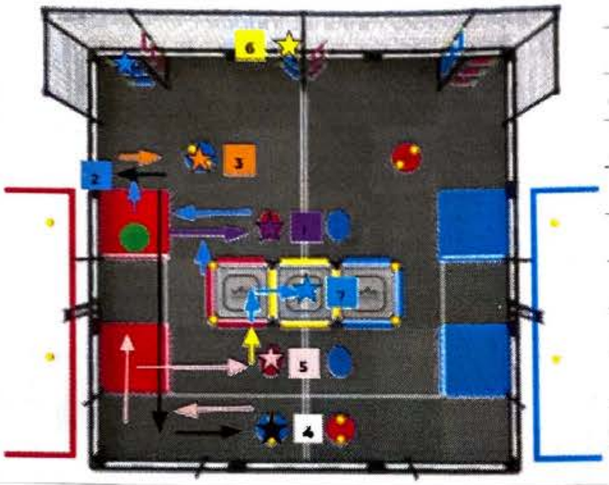
DATE 2/9/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-9-19

Continued from page

BRAINSTORMING PROGRAMMING SKILLS (CNTD.)



1. Move forward towards cap and get the ball underneath to score the cap.
2. Move back and shoot the 2 high flags.
3. Turn and flip the cap.
4. Move back and go to back of the field and flip the cap.
5. Move towards field wall and turn to flip cap.
6. Shoot top middle flag, using red platform
7. Climb onto center platform using alliance platform

Rank: 3/5

This programming skills run is similar to the previous one, scoring the same elements, but in a different path. In step 6, the robot shoots the high flag from the lower red platform. However, after testing this step with our robot by placing it on the platform with a ball, we realized it would not work because with one ball in the catapult, it would miss the flag. If we had two balls in our catapult, the robot would be able to hit the high flag. But, due to time constraints, we would not be able to obtain another ball, losing 2 points. Even so, this plan is still more consistent than the previous diagram, seeing that we would be able to park as no balls would get in our way. We ranked this plan a 3 out of 5 because while all of the steps are feasible to carry out, steps like step 6 where we expect the robot to toggle a flag by shooting from the red alliance parking platform seems to be challenging and inconsistent. The rank of 3 also tells us that there is a fair chance we may select this plan for programming skills.

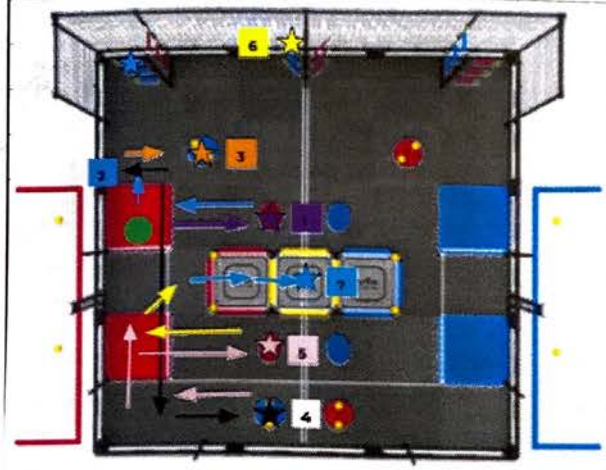
Continued to page

SIGNATURE
DATE 2/9/19

DISCLOSED TO AND UNDERSTOOD BY Cay Ragan
DATE 2-9-19

Continued from page

BRAINSTORMING PROGRAMMING SKILLS (CNTD.)



1. Robot starts with rollers facing the caps, and gets a ball from under the cap and flips it.
2. Move back to starting position and toggle the 2 high flags on our side (left).
3. Flip the cap by turning.
4. Move to back of the field and flip the cap.
5. Move back towards wall and turn to flip second back cap.
6. Shoot at high flags in the middle.
7. Move onto platforms and park on center.

Rank: 4/5

This skills run implements our front and back autons, with a few minor adjustments. This autonomous strategy is ideal for us since we can reuse code from our front red and middle back shoot autons. Also, this outline of programs is similar to the previous skills run drawn, except that the robot shoots the top middle flag from the field rather than on a platform. We tested shooting from the field, and we were able to successfully hit the top middle flag from the back red alliance tile. A score of 4 reflects the efficiency of this strategy.

OUR PROGRAMMING SKILLS DECISION

Seeing how the highest ranked programming skills plan was the plan above (page 101), with a rank of 4 out of 5, we concluded that we will carry out this plan for the state competition. Using the functions we created on page 90, we will code the programming skills plan accordingly, as well as utilizing wheel rotations. We believe it is feasible to attempt to score 16 points at least with this plan, since we can perfect our alignment easily by pressing against the wall to straighten ourselves out before turning and driving up to elements like caps. The other programming run options because they consisted of many instances that would cause inconsistencies and problems aside from the code.

SIGNATURE

Continued to page

DATE 2/9/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-9-19

Continued from page

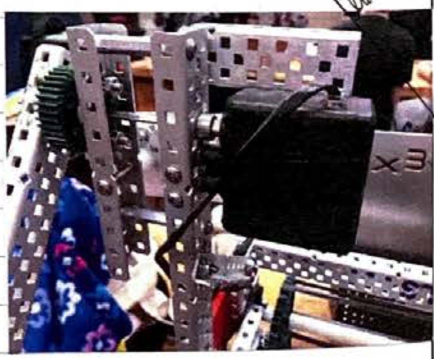
MOTOR POWER REQUIRED TO LAUNCH

Throughout the season, we have always been faced with the instance in which our motors overheat when we are practicing driving. So, we wanted to see how much power the motors go through to cause the overheating, and possibly the maximum power of the motors. After a launch of the ball, the energy lost from the spinning wheels is transformed into ball kinetic energy and heat due to friction and ball compression. After each shot, the wheels are brought back to their initial spin speed by the power of the motors. The maximum time allowed for respinning is the cycle time of the firing sequence. In order to get a better idea of the motor required, we will take a look at the VEX pieces for this year's game.

Mass of one ball = 0.12 lbs / 55g
Ball launch speed = 200 rpm, 6 m/s
One cap = 0.74 lbs / 335g
Ball kinetic energy = $\frac{1}{2}mv^2 = \frac{1}{2}(0.055g)(6)^2 = 0.99$ J
Energy loss due to compression = E_c
Energy loss due to friction = E_f
Time between shots = 1 second
Average power required = $P_{avg} = (K + E_c + E_f)/t$

Force on ball during acceleration : $F = d(mv)/dt$
Change in momentum of the ball over the time of acceleration, dt can be approximated as the contact distance / tangential speed of the wheel, v.

The contact distance is approximately 3 cm., therefore,
$dt = 0.03/6 = 0.05$ s
$d(mv) = (0.055)(6) = 0.36$ kg/m
$F = (0.36)/(0.005) = 72$ N
$F_n = F /$ coefficient of friction.



Continued to page

SIGNATURE
DATE 2/10/19
DISCLOSED TO AND UNDERSTOOD BY
DATE 2-10-19

Continued from page

MOTOR POWER REQUIRED TO LAUNCH (CNTD.)

The normal compression force on the ball is therefore $F_n = 72$ N (assuming the coefficient of friction = 1).

The assumed compression distance is approximately 1 inch or 2.54 cm. So,
$E_c = F_n d / 2 = 72(0.0254/2) = 0.91$ J

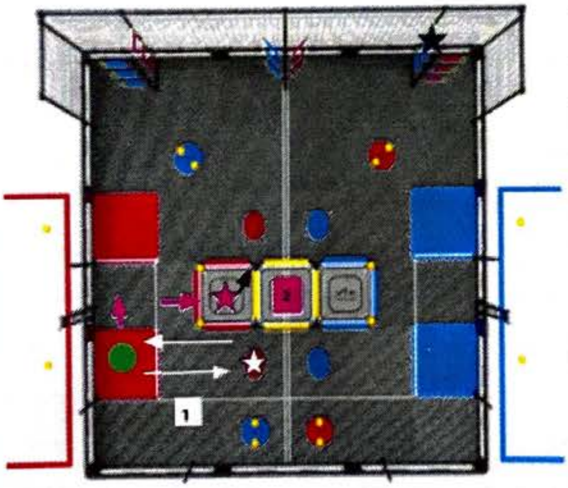
With a good design, the friction loss in the drive tain can be small. For the purposes of this design, we will assume that $E_c + E_f$ is approximately equal to K, which is equal to 1.06 J. From this we can find the average power as follows:
$P_{avg.} = 2K/t = (1.06)(2) = 2.12$ watts or 1.06 watts per motor

Since it is known that the V5 motors have maximum power at about 11 watts, the motors will stop if they run continuously with currents equivalent to more than 25% maximum torque. At this point, the motors only deliver 3/4 of the maximum power, which is about 8.25 watts. There are also frictional losses from the sprockets attached to the catapult. The net power to the catapult motor will be $(0.9)(11) = 9.9$ watts, which is more than the required amount to launch a ball. With this taken into account, we have confirmation that it is not the torque load that is the issue. Rather, the issue may come from overuse. For instance, at time we will run the roller even when the robot is not intaking balls, which is a waste of motor power.

Continued to page

AUTONS

For state competition, we decided to add an additional auton to our library of autons (pages 73-74). This auton functions for our robot on the back alliance tiles: it grabs the ball from underneath the cap to set it on our alliance color and then promptly parks onto the alliance platform to descore the opponent's scored flags. This is assuming that the other alliance has scored their set of flags, which is mostly always the case since robots competing at the state level are advanced and will surely have autons. On top of that, the most common autons we have observed throughout the competition season are shooting and toggling one's alliance color's flags since they are the most accessible and, arguably, easy to score. Knowing this, we created a back auton since many of our alliance partners have front autons, and thus we can maximize our autonomous points. If our auton plan is carried out successfully, we should be able to score 7 autonomous points. We have created this auton plan for both alliance colors. We chose to park first on the platform in order to make sure that we got the parking points since parking offers more points than toggling flags. In other words, if we had shot the flags prior to parking on the platform, then either time may run out or the alignment to get onto the platform might get messed up. If we park first, we will guarantee the 3 points and with the time remaining, attempt to score more points by shooting the flags.



1. Move towards cap and grab ball to set the cap on alliance color.
2. Park on alliance platform.
3. Position robot to descore flags that opponent scored. Robot can descore top and low flag.

Continued to page

SIGNATURE [signature]

DATE 2/10/19

DISCLOSED TO AND UNDERSTOOD BY [signature]

DATE 2-10-19

PROPRIETARY INFORMATION

Continued from page

AUTONOMOUS STATISTICS

Below are the statistics for the 3 main types of autonomous programs we plan to run during the competition. Each autonomous was run 5 times and analyzed to determine if the program needed to be revised in order to improve accuracy.

RED BACK

Trial	Number of Points Scored
1	7
2	6
3	4
4	7
5	7

Mean: 6.2
Sample size: 5
Minimum: 4
Maximum: 7
Sample Standard Deviation: 1.304
90% Confidence Interval: (4.957, 7.443)

The red back auton, our most used one, averages 6.2 points, and our data indicates that we can be 90% confident that we will score between 4.957 to 7.443 points (approximately 5-7 points). This, and the relatively low standard deviation indicates that we can expect to earn our target number of points.

RED DESCORER *assuming opposing alliance scores their set of flags*

Trial	Number of Points Scored
1	4
2	6
3	7
4	5
5	6

Mean: 5.6
Sample size: 5
Minimum: 4
Maximum: 7
Sample Standard Deviation: 1.140
90% Confidence Interval: (4.513, 6.687)

The red descorer auton averages 5.6 points, and our data indicates that we can be 90% confident that we will score between 4.51 to 6.69 points. Our low standard deviation indicates that it may be difficult to score the full 7 points, so our red back auton may be more effective depending on our opponent, but we can still expect approximately 5 to 6 points.

Continued to page

SIGNATURE [signature] DATE 2/12/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 2-10-19

Continued from page

AUTONOMOUS STATISTICS

Below are the statistics for the 3 main types of autonomous programs we plan to run during the competition. Each autonomous was run 5 times and analyzed to determine if the program needed to be revised in order to improve accuracy.

RED BACK

Trial	Number of Points Scored
1	7
2	6
3	4
4	7
5	7

Mean: 6.2
Sample size: 5
Minimum: 4
Maximum: 7
Sample Standard Deviation: 1.304
90% Confidence Interval: (4.957, 7.443)

The red back auton, our most used one, averages 6.2 points, and our data indicates that we can be 90% confident that we will score between 4.957 to 7.443 points (approximately 5-7 points). This, and the relatively low standard deviation indicates that we can expect to earn our target number of points.

RED DESCORER *assuming opposing alliance scores their set of flags*

Trial	Number of Points Scored
1	4
2	6
3	7
4	5
5	6

Mean: 5.6
Sample size: 5
Minimum: 4
Maximum: 7
Sample Standard Deviation: 1.140
90% Confidence Interval: (4.513, 6.687)

The red descorer auton averages 5.6 points, and our data indicates that we can be 90% confident that we will score between 4.51 to 6.69 points. Our low standard deviation indicates that it may be difficult to score the full 7 points, so our red back auton may be more effective depending on our opponent, but we can still expect approximately 5 to 6 points.

Continued to page

SIGNATURE [signature] DATE 2/12/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 2-10-19

AUTONOMOUS STATISTICS (CNTD.)

RED FRONT

Trial	Number of Points Scored
1	6
2	4
3	6
4	2
5	6

Mean: 4.8
Sample Size: 5
Minimum: 2
Maximum: 6
Sample Standard Deviation: 1.789
90% Confidence Interval: (3.095, 6.505)

The red front auton averages 4.8 points, and our data indicates that we can be 90% confident that we will score between 3.095 to 6.505 points. Our standard deviation is relatively higher for this auton, indicating its unreliableness, as does the range of the confidence interval. This compels us to use our red back auton when possible.

PROGRAMMING SKILLS STATISTICS

Trial	Number of Points Scored
1	16
2	12
3	10
4	16
5	14

Mean: 13.6
Sample Size: 5
Minimum: 10
Maximum: 16
Sample Standard Deviation: 2.608
90% Confidence Interval: (11.114, 16.086)

The standard deviation of 2.608 indicates that we must perfect our code to score unreliable points. Our data indicates that we can be 90% confident that we will score between 11.114 to 16.086 points. Our mean score shows improvement from the 11 points scored last competition.

DRIVER SKILLS STATISTICS

Trial	Number of Points Scored
1	20
2	18
3	21
4	20
5	20

Mean: 19.8
Sample Size: 5
Minimum: 18
Maximum: 21
Sample Standard Deviation: 1.095
90% Confidence Interval: (18.756, 20.844)

After practicing driver skills, we saw that we were able to reliably score within our target score of 20 points, with the low standard deviation allowing us to feel confident in our driving. We aim to continue practicing to increase this accuracy.

Continued to page

Continued from page

JOYSTICK CONTROL

Because our robot is controlled by dual driving, it is useful to keep reference of each button on the joystick, especially keeping reference of which functions of the robot each driver controls. Below, we have outlined what each button does in our dual drive controllers.

Driver 1:

L2: Catapult Control

L1: Rollers Forward

Rollers Reverse: R1

Drive Control

Drive control

X: Descorer Forward

B: Descorer Reverse

Driver 2:

L2: Catapult Control

L1: Rollers Forward

R1: Rollers Reverse

X: Descorer Forward

B: Descorer Reverse

Continued to page

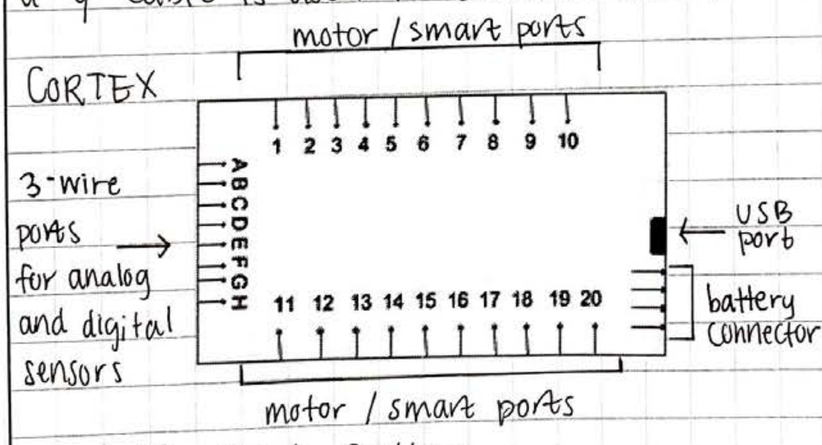
WIRING DIAGRAM

In order to make the wiring more comprehensible, a wiring diagram for the robot was developed. The diagram is laid out as the robot is, with all of the robot's components placed in the same location as they would appear on the robot. The diagram allows for others to see how the robot is wired, how its motors and other components are used, and can offer others guidance when wiring ~~theire~~ robots.

Rather than using a wire schematic, a wiring diagram was used due to its simplicity and clarity. A wiring schematic would be a more standardized method of displaying the wiring of the robot, however, the reasoning for the diagram was to provide an accurate pictoral representation of the wires of the robot. A wiring diagram also allows the use of a physical representation of the components rather than a standardized symbol, providing a more comprehensible representation for all to understand. Below is a legend explaining how each different component is represented, each custom made by our team.

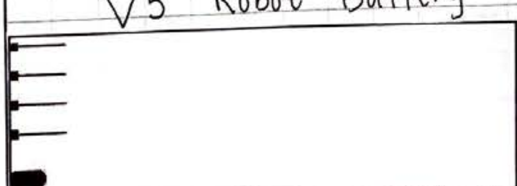
Similar to that of a wiring schematic, contact points are shown using a shaved circle and no contact points as jumps, or semicircles. This allows for a better understanding of where wires are connected and where a y cable is used to connect two different cables.

CORTEX

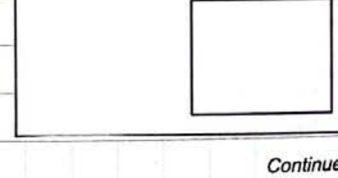


V5 Smart Motor

V5 Robot Battery

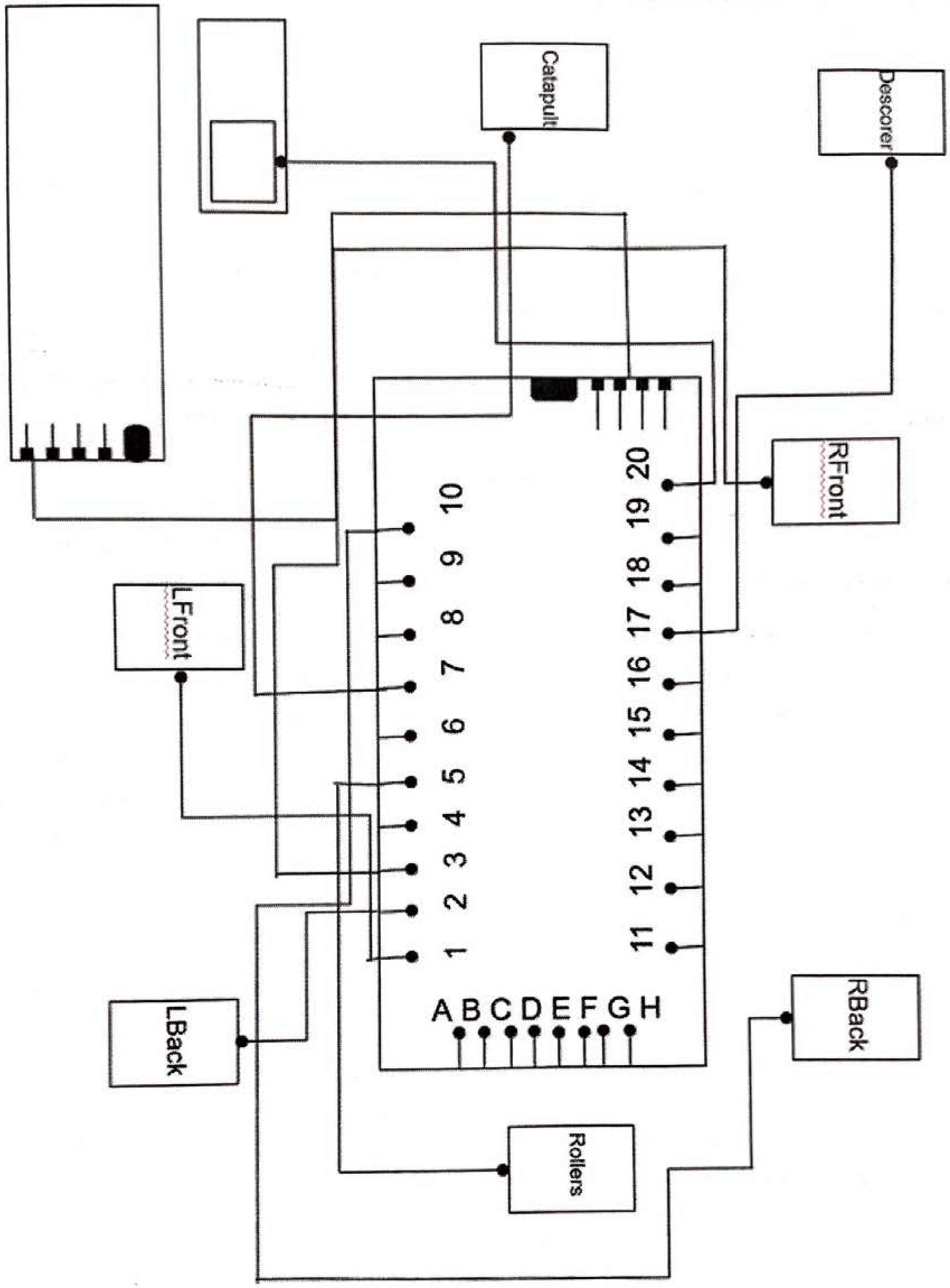


V5 Robot Radio



Continued from page

WIRING DIAGRAM (CNTD.)



Continued to page

SIGNATURE [signature] DATE 2/14/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 2-14-19

PROPRIETARY INFORMATION

Continued from page

GOAL PRIORITIZATION

We plan on meeting everyday to work on autonomous programming and driving. Most of our mechanical problems stated earlier have been resolved and we will only plan on fine tuning any parts of our robot if need be.

Friday 2/15

We began programming our new 16 point skills run, using the one we sketched in page 101, where we brainstormed programming skills runs. Our most important goal was not necessarily to get all of the rotation distances exactly correct, but to come up with a basic skills code that followed the brainstorm, so we code improvements and change the distances throughout the week. Next, we worked with both drivers, especially the driver with secondary controller, as they were given new controls. These controls were changed in the code, but we really wanted the drivers to feel comfortable with the new distribution of tasks. Lastly, we tested all of our autons on both sides of the field to make sure that they all worked properly.

Saturday 2/16

We began practicing driver skills with both of the drivers. This not only helped to make the drivers more comfortable with the partner control, but the repeated practice also made sure that both drivers memorized the driving skills strategy from page 58 and were comfortable with it. Since we did not completely finish fine tuning our autons and making our programming skills' basic code, we made sure to finish that today.

Problem: We are having catapult issues. First, when the second ball is going into the catapult, it does not land behind the first ball as it is supposed to for a successful launch. Instead, the ball either rolls out of the catapult and onto the floor, or it gets stuck in the robot, or it lands between the gap between the catapult and rollers. Or, it lands on top of the first ball. This is the same problem we had at the Millville competition, but it seems to be happening more frequently and more problematic. As a result, our drivers have to move the robot back and forth to try and move the ball back behind the first ball.

Continued to page

SIGNATURE

DATE 2/15 - 2/16/19

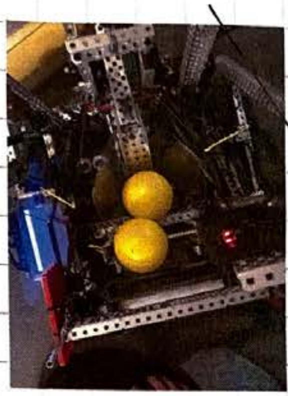
DISCLOSED TO AND UNDERSTOOD BY

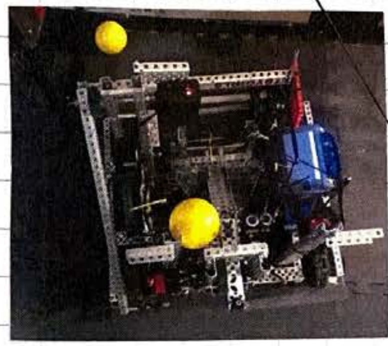
DATE 2-15-16

Continued from page

GOAL PRIORITIZATION

Here, the balls get stuck between the gap between the rollers and catapult.



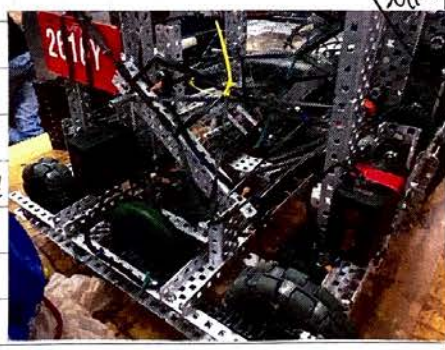


This picture shows how the ball can miss the catapult and land some place else on the robot, when taken in.

Solution: We put a rubber band across the two c-channels behind the rollers to increase the bounce, so that if a ball gets stuck on its way to the catapult, it will rebound back towards the rollers. Then, the rollers will give the ball a slight push along with the push of the rubber band on the ball so that it can land back into the catapult. Even so, this solution is not the perfect solution, since there are still instances where we must jolt the robot back and forth to get the ball back in position.

Week of States 2/17-2/22

Throughout these days, we continued practicing driver skills since our goal is to score at least 20 points in driving. The drivers were now evidently more comfortable with the strategy and seemed more confident overall with the driver skills. Nevertheless, we must fine tune programming skills by running it a few times to see if there is anything else we need to complete. We also scrimmaged with the rest of the 2616 teams. We practiced shooting flags and flipping caps since the 2 drivers cannot get through much if they do not collaborate and practice communication. We also practiced shooting from the back of the field so we can shoot without robots around us.



Problem: The right side of the drive is slower than the other.

Solution: We replaced the motors with new ones and saw that the axle motor has too much pressure and will not spin. So, we added more spacers to create room for the robot to run and took away the friction.

SIGNATURE

DATE 2/17/19 - 2/22/19

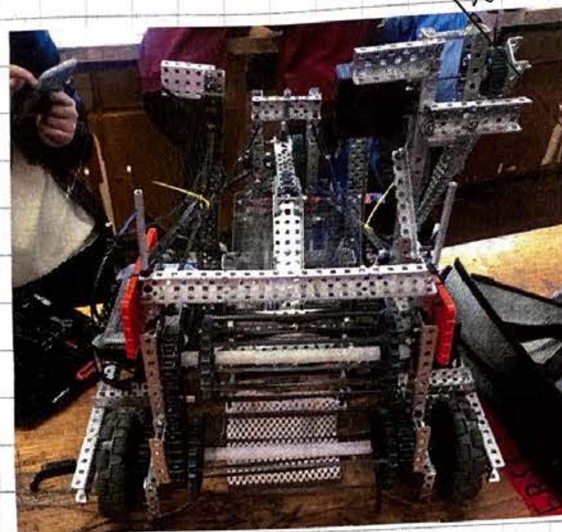
DISCLOSED TO AND UNDERSTOOD BY

DATE 2-17-22

PROPRIETARY INFORMATION

Continued from page

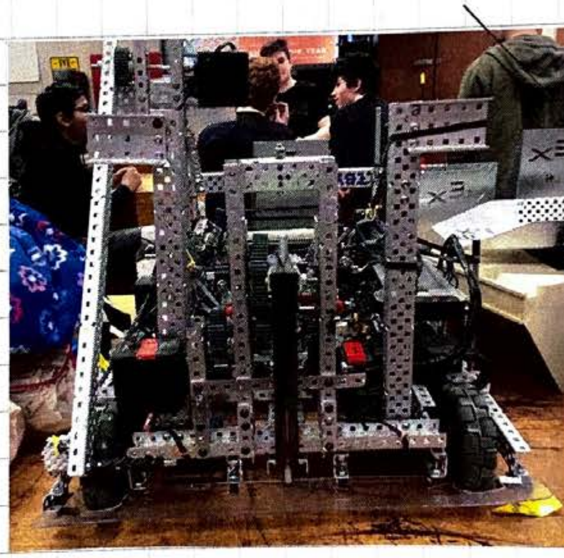
ROBOT INDEX



Catapult Ball Launcher - pages 24-25, 48

Drive - pages 19, 20, 46

Rollers - pages 16, 23



Lift - pages 22 and 51

Rubber Bands - page 50

Plexiglass Ramp - pages 30, 70

The above labelled pictures of our robot serves as an index of its essential parts, allowing us to easily access information about our experience with ball intakes, ball launchers, cap intakes, etc. This also provides a quick way for us to strategize and plan designs and ideas for future competitions as we can readily access our past analyses of different structures. We hope that our notebook will serve as a reference for future teams and games if the goal of the game is similar.

Continued to page

SIGNATURE

DATE 2/22/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-22-19

Continued from page

TEAM REFLECTION

This past season has been truly remarkable in so many ways. With charged laptops and boxes of tools, we came back to the robotics room ready to feel the thrill of another robotics season. We came into this year fully equipped with one year of experience and knowledge on our belts. Our passion and excitement drove us to constantly build, code, and improve, and we loved every second of it. We love the small moments before the competition when a team asks us to help get their competition code template, when a freshmen team turns to us to ask about how they should improve their build, and when other teams look at us for our sportsmanship and positivity.

We have come a long way from last year, as each of us gained more experience and knowledge in STEM fields from both our classes and from last year's competition. But more importantly, we became better teammates and closer friend along the way. We now not only have the same drive to succeed and desire to prove ourselves, but we also have more experience to go along with it. We are more confident, trust one another more, and are excited to see what our future has in store. As sad as it is that it is our last year in robotics, we are graduating having created an impact not just in our school, but in the community at large. Today, our robotics club boasts 3 all-girls teams, whereas one year ago, it was just us. Many of this girls look to us as their role models, coming to us for design and programming help. Being able to spread our knowledge and experience with underclassmen girls is incredibly rewarding, and we can graduate~~d~~, rest assured, that these girls will continue our legacy.

Continued to page

SIGNATURE [signature] DATE 2/22/19

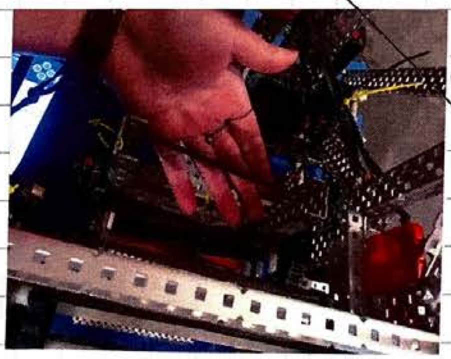
DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 2-22-'19

Continued from page

NIGHT BEFORE STATES

Immediately after school, we went to the robotics room and helped set up for the state competition held at our high school, Cherry Hill High School East. When we finished setting up, we went on the field to fix and run our skills auton as well as regular auton programs. On top of that, we practiced driving and driver skills to go over the strategy and gameplay. We luckily did not have many problems today - all aspects of the ~~party~~ robot appeared to be functioning well. We scrimmaged with other teams to get a feel for tomorrow.

Our goals for the state competition is to always inspect the robot for problems before ~~inviting her~~ getting on the field for a qualification match. For instance, we do not want to perform badly if we do not enough rubber bands for the catapult. For the issues that we do have control over, we want to ensure that we take care of those problems to prevent any careless mistakes. We also want to carry out the strategies we discussed such as shooting flags from the end of the field as opposed to the front of the field. We aim to have a successful run, especially with our added rubber band to help prevent balls from not getting into the catapult, as seen below.



Regardless, we aim to perform the best we can as we meet new teams and that we compete with complete focus. Another one of our goals is to communicate effectively within the drive team and with our alliance partner.

Continued to page

SIGNATURE

DATE 2/22/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-22-19

PROPRIETARY INFORMATION

Continued from page

DAY OF STATES

Upon arriving at the state competition, we went to get inspected and turned in the notebook. Then, we went to the skills fields to begin driver skills. We were able to get 3 tries and were able to score 22 points as our highest run. We scored all the high flags, 3 low flags, and parked on the center platform. Unfortunately our score got marked as 19 and we were unable to correct the score. Since skills ran throughout the day, we ran our programming skills later, and ultimately scored 8 points. We believe that our auton failed to score the 16 points because of the entanglement issue. That is, there was an announcement stating that teams with entanglement caused by their ball intake rollers would be at fault, so it ~~it~~ was suggested that teams cover their rollers to prevent entanglement. Thus, we covered our rollers with friction padding, which may have caused the inconsistency of our skills auton and its failure to score the 16 points. For our qualification matches, we ran into some problems that came to our disadvantage on top of the fact that we were competing against highly competitive robots. For 2 of our matches, a ball got stuck under our catapult, immobilizing the catapult for the entire match. This problem occurred because of our catapult / ball intake system problem where the second ball that is fed into the robot does not land in its optimal position for shooting. As a result, the first ball shoots but the second ball simply falls underneath the catapult.

Continued to page

SIGNATURE

DATE 2/23/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-23-19

PROPRIETARY INFORMATION

DAY OF STATES (CNTD.)

Another issue we had was that our auton did not shoot at the flags towards the middle of qualification matches. Our catapult was unable to shoot or move, but we were able to win the match 23 - 4 since we played heavy defense on the other team and scored some caps while our alliance partner focused on the flags. After the match, however, we figured that the motor of the catapult must be broken since the code for the catapult was never changed and was correct. So to fix this problem, we took off the descorer and replaced the catapult motor with the descorer motor. We chose to do this since we rarely make use of the descorer in matches as we mainly focus on scoring points during the entirety of the match.

Problem: The catapult is not moving when we use the joystick control to call upon it. This is most likely a motor issue since the code was unchanged as well as the physical structure of the catapult. As a result, we cannot shoot at flags, which is our main game goal.

Solution: We replaced the catapult motor with the descorer motor since we rarely make use of the descorer, thus our performance in matches will not be affected significantly.

We finished the qualification matches 4 - 4 with a rank of 22 and skills with 30 points with a rank of 13. We were not selected for an alliance, but we were nevertheless pleased with our performance given the many problems we encountered and how we solved them. Fortunately, we received the Design Award, which qualified our team to Worlds!

lk

lk lk

Continued to page

SIGNATURE

DATE 2/23/19

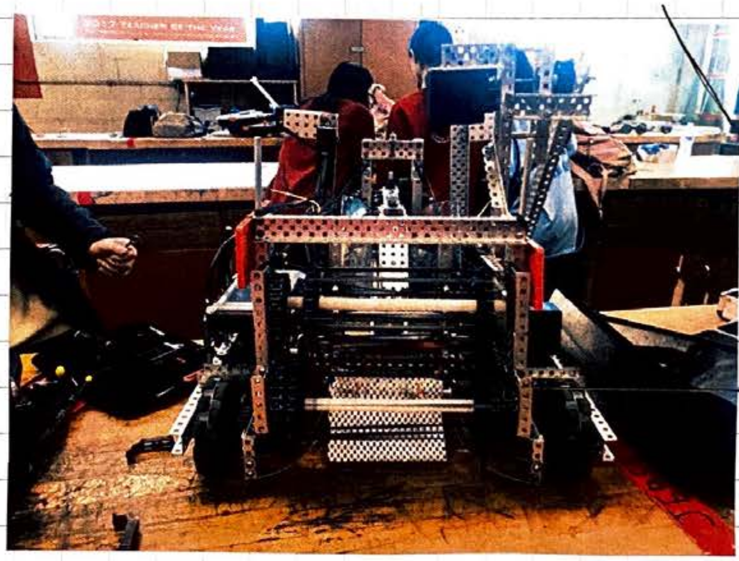
DISCLOSED TO AND UNDERSTOOD BY

DATE 2-23-19

Continued from page

REFLECTION OF STATES

After the state competition, we knew we had to fix our robot design to improve our overall performance for the world competition. One fix would be extending the ball intake to the entire front side of the robot. This change to our design would help us achieve our goal of being able to get a ball into our catapult in the most efficient way possible. With our current ball intake, the rollers are pretty narrow and thus only allows for an exact position for the balls to be in for them to get fed into the robot. By extending the rollers to the entire front side of the robot, the ball just has to be within the front side of the robot, and from there, the ball can be guided into the catapult with the plexiglass. With a longer roller, we can have more accuracy in flipping the caps. Our current narrow rollers must be in an exact position to touch and flip the caps. By extending the rollers, we do not have to be in a specific position and can instead drive forward while having the rollers run.



l.K.

Narrow ball intake rollers

As shown above, our current rollers are pretty narrow compared to the longer, full length rollers we have seen in other robots' designs. We can improve our design by extending the roller to start and finish from each wheel.

Continued to page

SIGNATURE

DATE 2/24/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-24-19

Continued from page

REFLECTION OF STATES (CNTD.)

Another change that we should make to the robot is the way in which the balls land into the catapult. Our catapult ball intake has always been giving us problems, mostly the fact that the balls would not land right behind or next to each other to be able to double shoot. We must think of a way to redesign the way in which the balls get to the catapult so that the balls can land into the catapult in the most optimal way and position.

We also want to improve upon our autonomous performance. Unlike our last regional competition, our autons did not have a high success rate at states, and instead failed to score points that we were certain the robot could score. Autonomous points are important to us because they give us an advantage for the autonomous bonus as well as gaining AP points that may place us higher in rankings. To make sure that our autons can guarantee us points at any time, we must carry out more trials and tests before the world championship.

Lastly, rather than completely rebuilding our robot design, we will make fixes and adjustments to the aspects of our design that need improvements. We will take this approach because we believe it is far more efficient to use our time mostly to practice driving rather than spending it to rebuild. Our team will be more successful with a robot that can be driven well than a robot with a good design and poor driving.

lik

lik

lik

Continued to page

SIGNATURE lik

DATE 2/24/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 2-24-19

PROPRIETARY INFORMATION

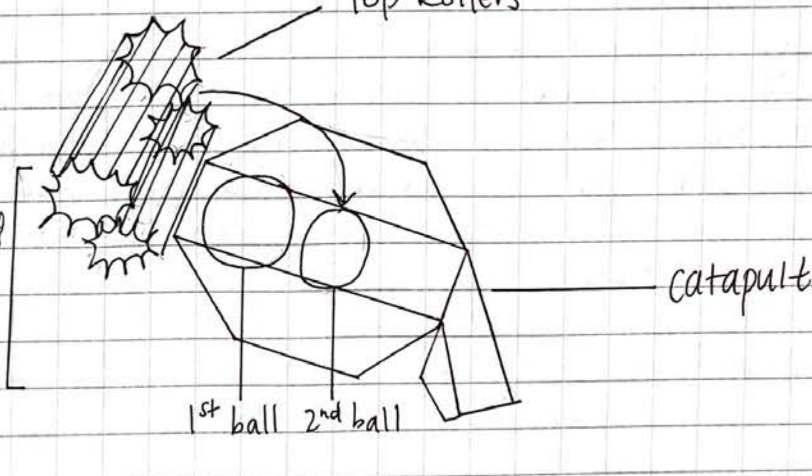
Continued from page

BRAINSTORMING FIXES FOR ROBOT

Ball Roller Intake Redesign

We can redesign the ball intake rollers by removing our current rollers and extending our sprockets by the wheels of the robot. By doing this, we can have a wider range of intake, we do not have to spend as much time chasing after balls because we will increase our intake by almost 2 times the length. More time can now be spent on other things, as it will be easier to pick up balls. The more space we have for rollers, the more opportunity we will have to pick up balls. That being said, we must move the wheels of our robot back in order to fit the new and improved rollers in the front since they extend across the entire front. Also, to solve our problem with the balls landing incorrectly into the catapult, we can make the catapult rollers higher. This may fix our problem because the balls will fall from a higher height, which will allow the second ball to land behind the first. As a result, the balls will be aligned correctly in the catapult in order to ensure toggling 2 flags at once.

We can use an increased roller height to ensure that the second ball lands behind the first, which would be a greater horizontal distance. An increased starting height means a greater horizontal distance travelled because it will take longer to reach the ground (since gravity stays constant). Also, during that time the horizontal acceleration stays 0, so the horizontal distance travelled will increase linearly with time.



Continued to page

SIGNATURE

DATE 3/1/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 3-1-19

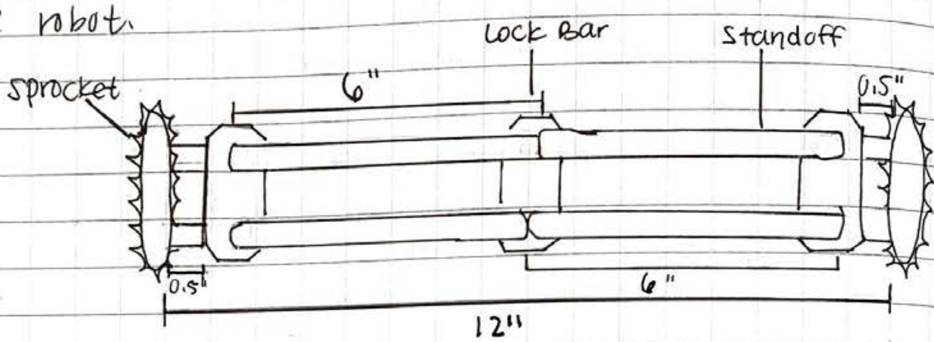
PROPRIETARY INFORMATION

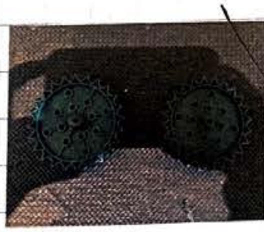
Continued from page

BUILDING THE NEW BALL INTAKE

To build the new ball intake system, we will first construct the bottom roller where the robot will be able to take in the balls from the ground. We will make it so that the roller is longer in length, approximately 12" to cover more space so that the balls will easily be guided into the robot.

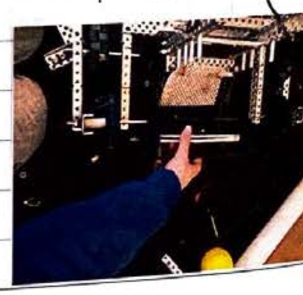
Parts List:
- (4) 6" standoffs
- (4) 0.50" standoffs
- (6) couplers
- (2) 24T sprockets
- (16) half spacers
- (16) lock nuts
- (2) axles
- (3) drive shaft bar lock
- Collars
- Rubber bands





Steps to Build:

To make the bottom roller, first we will connect the 6" standoffs to the lock bar with couplers, 2 standoffs will be on one side of the lock bar and the other 2 symmetrically on the other side. Then, on the ends of these 6" standoffs, we will add a 0.5" standoff through another lock bar to complete the length at 12". After creating the structure, the sprockets will be attached to the 0.5" standoffs with screws. 2 axles must run through each side of the sprockets so that the roller can be attached to the drive base.



SIGNATURE

DATE 3/5/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 3-5-19

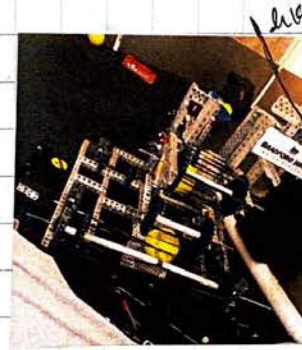
PROPRIETARY INFORMATION

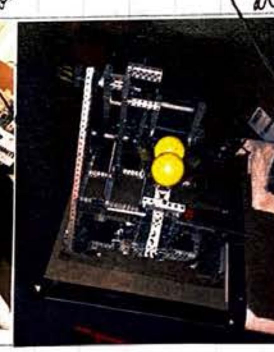
Continued from page

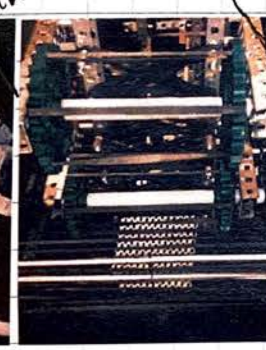
BUILDING THE TOP ROLLERS AND RUBBER BAND WALL

To finish building the roller system that guides balls into the catapult, we must recreate rollers that will drop the ball into the catapult at a height greater than before, so that we can get rid of the previous rubber band web. If the balls are not dropped at a higher height, then we would still have the recurring problem of misaligned balls in our catapult, which hinders the robot's ability to accurately double shoot.

Parts List:
- (2) 18" sprockets
- (2) 24" sprockets
- (2) axles (to be cut)
- Spacers
- C-channel (to be cut)
- Standoffs
- Rubber Bands
- Collars
- Bearings







Steps to Build:

First, to make the top 2 rollers, run the axle through the first sprocket. Then, put the spacers onto the axles so that it is wide enough to allow the rubber bands to roll the ball through. Then, place the second sprocket on the end to close off the roller, and attach both ends of the roller into the c-channel structure. Repeat this process for the 24" sprockets and place the second roller on the top c-channel. The rubber band wall requires 2 vertical C-channels attached to the drive base. Every 3 squares of the c-channel we will put a 0.5" standoff on both sides. After this, we will diagnally connect the rubber bands to the standoffs, creating a wall to allow the ball to have traction to be able to climb vertically up the bot.

Continued to page

SIGNATURE

DATE 3/5/19

DISCLOSED TO AND UNDERSTOOD BY

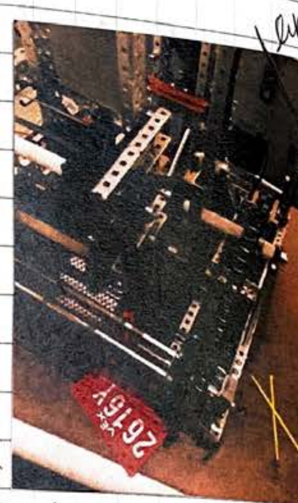
DATE

PROPRIETARY INFORMATION

Continued from page

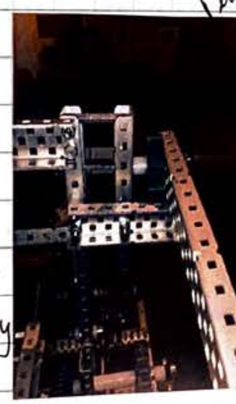
ADDITIONS TO ROBOT DESIGN

Chain Intake System



This system allows all 3 rollers to use the power of only one motor. The motor will be attached to the second roller up out of the 3 rollers. To connect the different ~~motors~~ rollers to one motor, we will start by adding a 6 tooth sprocket to the right side of the bottom roller, which will then be chained to a 12 tooth sprocket helped up by 2 standoffs with an axle running through. This sprocket ratio will also allow for the bottom roller to spin faster than the top 2 rollers, so that picking up the balls from the ground will be quick. This same axle will have another 12 tooth sprocket on it that will connect the bottom roller to the top 2 rollers. In a chained triangle, these 3 rollers will connect, finishing the system and only requiring one motor, which will save weight.

Descorer



As noted in our "Day of Competition" log (see page 116), we had to remove our descorer from our robot in order to use its motor for the catapult. We figured we could enhance our descorer design by adding a structure that will stop the descorer from falling all the way. That is, we hypothesize that instead of having the descorer rotate fully, we could stop it about halfway. This will help stabilize the descorer since previously it was very wobbly.

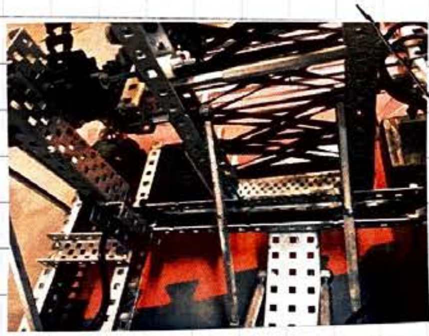
Continued to page

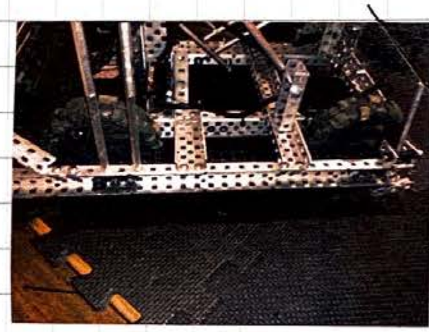
Continued from page

ADDITIONS TO ROBOT DESIGN (CNTD.)

Drive:

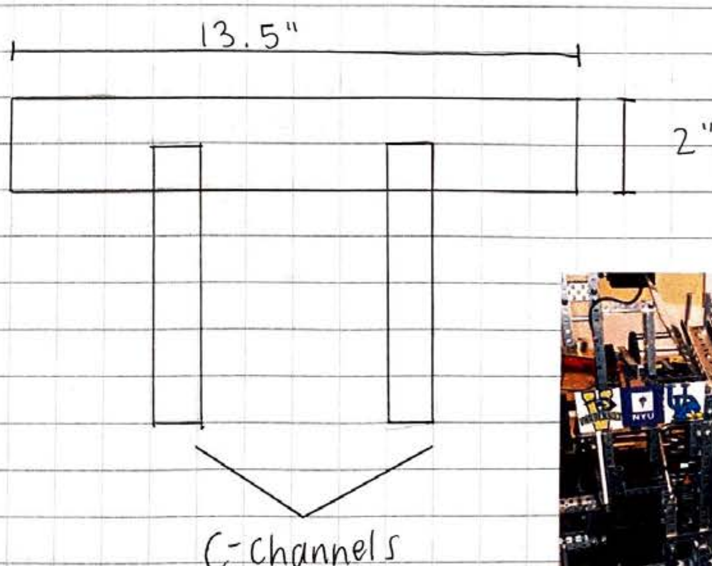
To clean up our base design, we removed any excess metal to also make our robot lighter. For example, some of the metal that we removed contributed to the triangle design that helped keep the base stable. To ensure that we still have enough stability in our drive and overall robot, we will add long standoffs diagonally to form the triangle shape. Adding standoffs instead of metal allows us to maintain a low robot mass to still have speed when we drive. We also had to move the wheels back since the roller now has extended to 12". We had to take out the friction wheels because there was less space with the new structure.

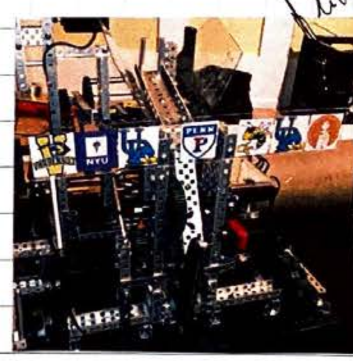




Lexan Bottom Flag Toggler:

In order to easily toggle the bottom flags, we created a lexan toggler that is a 13.5" x 2" rectangle. This will allow the driver to easily drive the robot into the bottom flag due to the protrusion of the rectangle, and will especially help during autons.





SIGNATURE

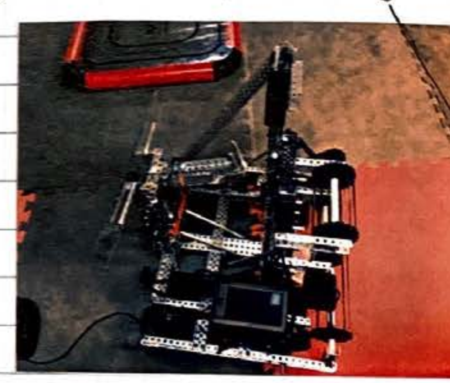
DATE 3/10/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 3/10/19

THE ROBOT'S CENTER OF GRAVITY (UPDATED)

As previously mentioned, this year's game of Turning Point does not require anti-tip structures like last year's game. There are rarely robots with tall lifts or lifts that move in a circular path, which ensures no excess angular and circular momentum, that would otherwise cause the robot to tip. Regardless, we would like to solve for our updated robot design's center of gravity, since we made changes to it that would make the mass differ. Our robot's mass is equally distributed throughout the robot. Thus, when the mass of the one or two balls is applied to the robot, the ball's mass is not enough to greatly affect the overall speed and effectiveness of the robot. The weight of each ball is noted as approximately 0.12 lbs, or 55g.



Robot = 13.0 lbs
One ball = 0.12 lbs
Two balls = 0.24 lbs
One cap = 335 g = 0.74 lbs

$$\% \text{ of Change} = \frac{\text{weight of ball(s)}}{\text{weight of robot}} \times 100$$

$$\% \text{ of change for One Ball} = \frac{0.12 \text{ lbs}}{13.0 \text{ lbs}} \times 100 = 0.92\%$$

One ball's weight has a 0.92% effect of weight on robot's speed.

$$\% \text{ of Change for Two Balls} = \frac{0.24 \text{ lbs}}{13.0 \text{ lbs}} \times 100 = 1.85\%$$

Two Balls have a 1.85% effect of weight on robot's speed.

$$\% \text{ of change} = \frac{\text{weight of one cap}}{\text{weight of robot}} \times 100$$

$$\% \text{ of change} = \frac{0.74 \text{ lbs}}{13.0 \text{ lbs}} \times 100 = 5.69\%$$

One cap's weight has 5.69% effect of weight on robot's speed, which is not significant to create an anti-tip structure.

Continued to page

SIGNATURE

DATE 3/15/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 3/15/19

Continued from page

BRAINSTORMING AUTONS

For the world competition, we wanted to create autonomous programs that would score a high number of points. From our observance at states, we saw that our back middle flag auton (see pages 73-74) scored well and was consistent. We decided that we will keep that auton for worlds and brainstorm new ones that we could possibly use. It is extremely crucial for our team to have a variety of autons ready since it offers us an advantage in the match and rankings. Our goal for autonomous programs and periods is to be consistent in our scored points.

KEY:

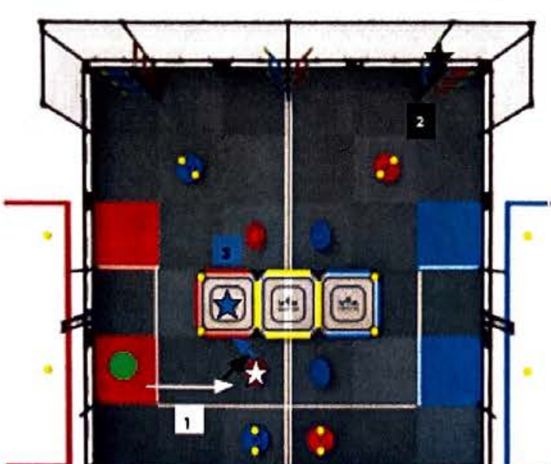
Starting Position

Scored

Direction of robot's pathway

* Each auton is diagramed on the RED side, but it applies to the BLUE side as well. *

Descoring Back



1) Robot will move to get a ball under the cap, and roll it in halfway, which scores the cap.
2) Robot turns to shoot top flag on the opponent's side, descoring by 2 points
3) Climb onto platform to park.

TOTAL: 6 points

Continued to page

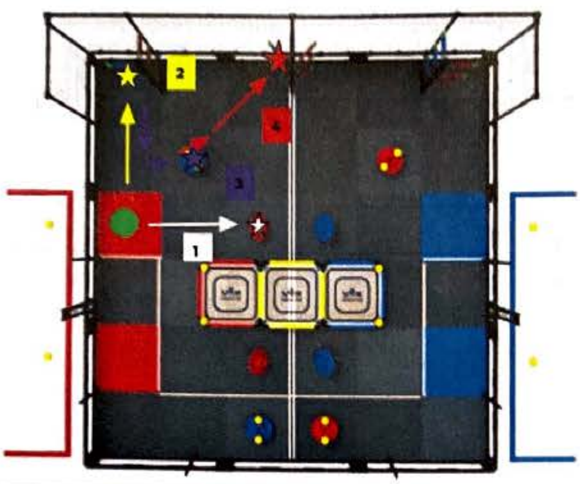
SIGNATURE

DATE 3/28/19

DISCLOSED TO AND UNDERSTOOD BY

DATE

BRAINSTORMING AUTONS (CNTD.)



Front:

1) Move to score the cap by getting the ball.

2) Shoot top and bottom flag, turn so rollers are facing flags, roll inward while hitting bottom flag (chance the shot balls can roll in)

3) Move to score another cap.

4) Shoot middle flags and toggle bottom flag.

TOTAL: 12 points

Front Revised

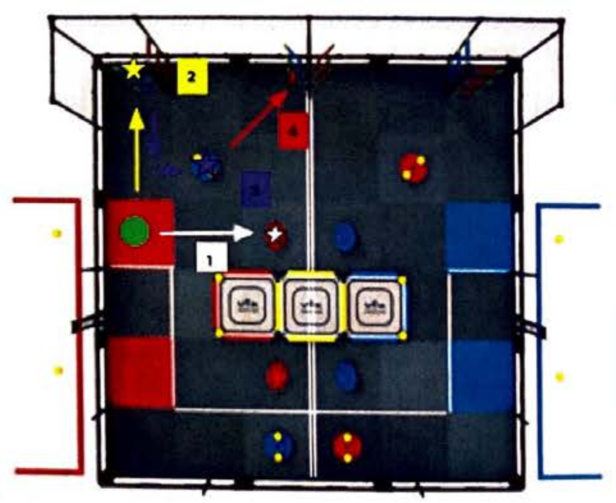
1) Move to score cap and acquire ball.

2) Shoot top and middle flags

3) Score another cap by flipping it.

4) Move to toggle flag in the middle column.

TOTAL: 7 points



Conclusion

After an analysis of the 3 autons brainstormed above, we will be adding on the first (Descoring Back) and third (Front Revised) to our library of autons as they appear the most reliable and feasible in scoring points. The second auton (Front) is too unrealistic as there are many factors we cannot control. For instance, the alignment always gets messed up when scoring the bottom flag (which is 1 point and thus not worth it) and time may run out with this auton's many tasks.

Continued to page

SIGNATURE

DATE 3/29/19

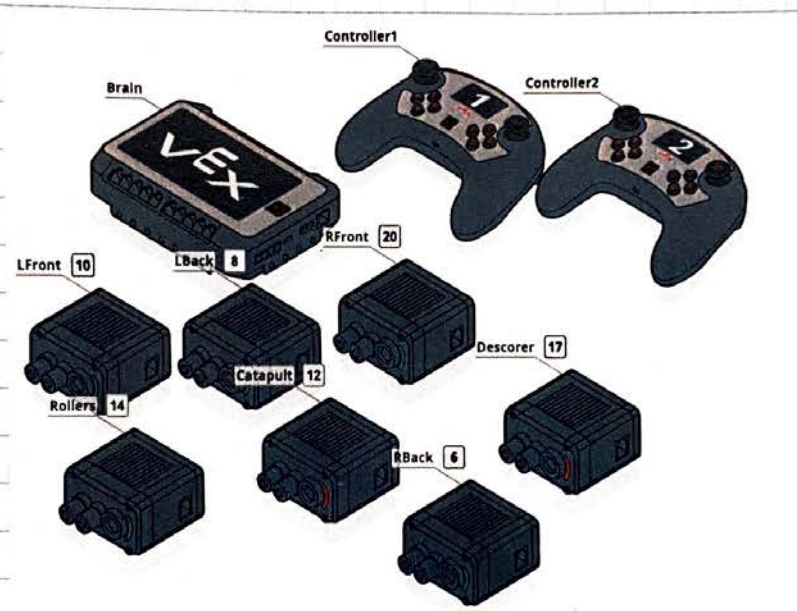
DISCLOSED TO AND UNDERSTOOD BY

DATE

Continued from page

CODE REVISION

Because we rebuilt parts of the robot, we must also revise our code. That is, we will need change the ports for each motor. The structures of each component of the robot need to be updated after our robot redesign, and thus we can accomplish this by also updating the code. It is important to update and revise the robot's code while the robot is being rebuilt, or undergoing a fix or change. Below depicts the new port assignment for each motor, as seen with the number listed next to the name of the motor and what it controls:



During the rebuild, the friction wheels of the robot needed to be taken off of the drive in order for the longer intake to fit into the robot. Because of this, the force required to push us will decrease. We decided to implement the hold type brake (see page 92) in addition to the brake type brake that triggers when the robot is not moving. However, the hold brake will trigger when the driver presses a button because the drive will malfunction if it is implemented automatically due to the strength it delivers.

Continued to page

SIGNATURE

DATE 4/2/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 4/2/19

Continued from page

WORLDS AUTONS STATS

Below are the statistics for the 3 primary autons we plan to run at worlds. Each autonomous program was run 5 times and analyzed to determine if the program needed to be revised in order to improve accuracy and consistency.

RED MIDDLE FLAG BACK (diagramed on pg.74)

Trial	Number of Points Scored
1	4
2	7
3	7
4	6
5	7

Mean: 6.2
Minimum: 4
Maximum: 7
IQR: 2
Sample Standard Deviation: 1.304

BLUE MIDDLE FLAG BACK

Trial	Number of Points Scored
1	7
2	5
3	7
4	7
5	~~4~~ 5

Mean: 6.2
Minimum: 5
Maximum: 7
IQR: 2
Sample Standard Deviation: 1.095

The middle flag back autonomous has the highest average score out of the 3 autonomous programs we plan to use at worlds, and has the same mean for both the red and blue sides, indicating its consistency. Its relatively low standard deviation illustrates that it is more realiable and will likely consistently score more points, than our other back autonomous program. Although it has the same amount of maximum points as our front auton, it is more reliable and will therefore be our most used go-to program.

Continued to page

SIGNATURE

DATE 4/9/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 4-9-19

PROPRIETARY INFORMATION

Continued from page

WORLDS AUTONS STATS

RED FRONT (diagramed on pg. 126)

Trial	Number of Points Scored
1	6
2	7
3	5
4	6
5	5

Mean: 5.8
Minimum: 5
Maximum: 7
IQR : 1
Sample standard Deviation: 0.8367

BLUE FRONT

Trial	Number of Points Scored
1	5
2	6
3	7
4	6
5	7

Mean : 6.2
Minimum : 5
Maximum : 7
IQR : 1.5
Sample Standard Deviation : 0.8367

Though the higher consistent averages of the middle back auton compels us to use that as our first choice auton, our front auton is also fairly reliable. It's lower standard deviation can indicate higher reliability.

RED BACK DESCORING (diagramed on pg. 125)

Trial	Number of Points Scored
1	6
2	4
3	4
4	3
5	6

Mean : 4.6
Minimum : 3
Maximum : 6
IQR : 2.5
Sample Standard Deviation: 1.342

BLUE BACK DESCORING

Trial	Number of Points Scored
1	4
2	6
3	6
4	4
5	6

Mean : 5.2
Minimum : 4
Maximum : 6
IQR : 2
Sample Standard Deviation : 1.095

This relatively large standard deviation of the descoring autons points to their unreliability, particularly in descoring the opposite side's flag. The success of this auton depends on too many external factors, so it will likely not be used unless deemed necessary.

Continued to page

DATE 4/11/19

SIGNATURE

DATE 4-11-19

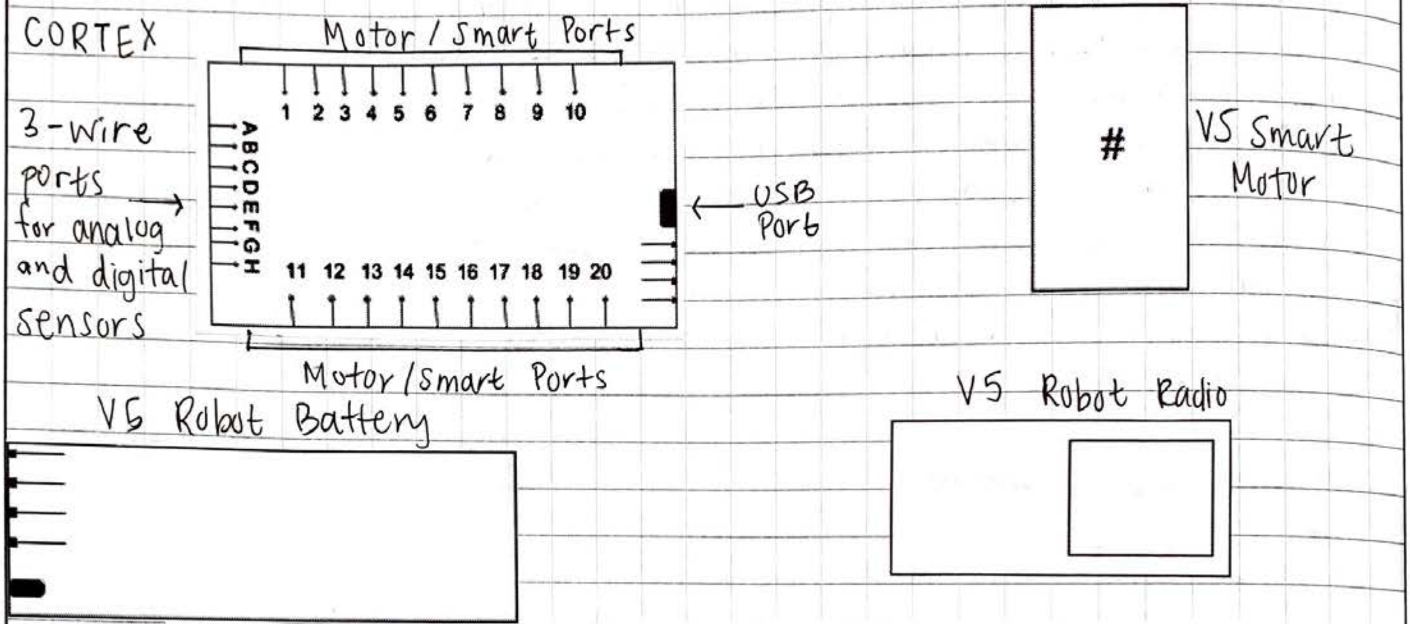
DISCLOSED TO AND UNDERSTOOD BY

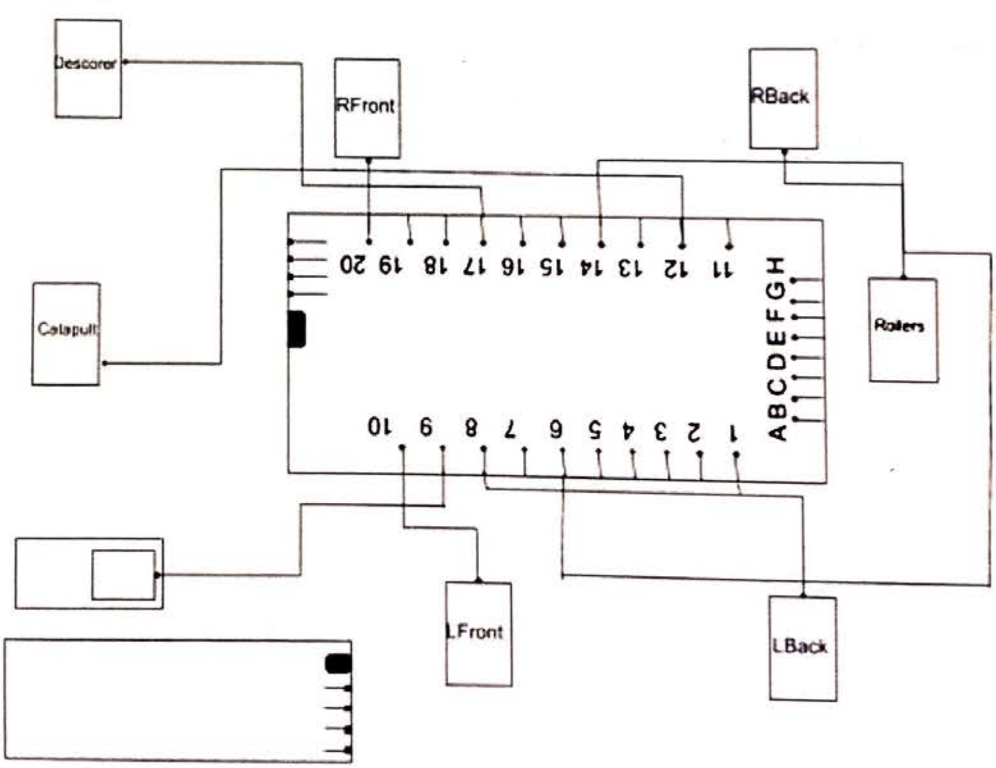
PROPRIETARY INFORMATION

Continued from page

UPDATED WIRING DIAGRAM

As we update our robot's design, our wiring must update as well. Refer to page 108 for more information. Below is the legend and our updated diagram





to page

SIGNATURE

DATE 4/18/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 4/18/19

PROPRIETARY INFORMATION

Continued from page

WORLDS CHECKLIST

As we prepare for the World Championship, we practiced driving as well as tested out our autons. It is important to focus on driving at this point after states because we will be more successful with excellent driving as opposed to making continued modifications and adjustments to our robot design. As part of our preparations for worlds, we want to engage and establish relationships with the teams we meet. We can do this by having free goodies to offer to teams in our division and establish friendships. Like last year, we plan on having business cards with information about our robot such as its design and autonomous programs.

World Competition Checklist

- Robot
- Laptop
- Batteries
- Toolbox / replacement wires
- Gearbox
- Drive Team glasses
- Joystick partner controls
- Metal shaver
- USB Cable
- License Plates
- Friction Padding
- Business cards
- Pens (S.WAG)
- Buttons (SWAG)
- Webcam covers (SWAG)
- Basketball hoop
- Banners
- Candy
- Speaker
- Booth supplies
- Robot case
- VEX Awards

C.C

Continued to page

SIGNATURE [signature] DATE 4/20/19

DISCLOSED TO AND UNDERSTOOD BY [signature] DATE 4-20-19

Continued from page

TEAM REFLECTION

Iris's Last Entry

Notebooker

Robotics has taught me not to fear the unknown. To be unafraid in trying new things, and to go out of my comfort zone, because I grow the most when I challenge myself. I came to love my team as I truly value teamwork and collaborating with my peers, and I look forward to continuing working with others beyond college to make a difference and improve lives.

— Georgia Institute of Technology '23, Biomedical Engineering

Vidhya's Last Entry

Programmer

I never expected to grow so passionate about robotics - to love the thrill of competition, the challenges of programming and repeated analysis and modification, or the VEX community. Aside from showing me a future with the engineering process and allowing me to learn so much from so many different people, robotics allowed me to meet and grow close to some of my favorite people. Love you guys.

— Vanderbilt University '23 Biomedical Engineering

Continued to page

SIGNATURE

DATE 4/21/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 4/21/19

Continued from page

TEAM REFLECTION

Caylin's Last entry

Builder

Robotics has taught me to never be afraid to try new things. Just because something is new does not mean that you won't succeed. It has given me the confidence to not take the easy way out and follow what you love. At the beginning of freshman year I thought I wanted to go into business, but after competing with VEX for two years, I realized that I truly enjoyed the engineering process.

- University of Delaware '23
Mechanical Engineering

Caroline's Last Entry

Driver, Programmer

These past two years have been a heck of a journey. Looking back to the end of my sophomore year, I would have never seen myself in the position I ~~was~~ am in now. Robotics has taught me so much. It has taught me how to think like an engineer, how to work in a team, and most importantly sisterhood. Without these girls, I wouldve never found myself and stepped out of my comfort zone.

- New York University '23
Integrated Digital Media

Continued to page

SIGNATURE

DATE 4/21/19

DISCLOSED TO AND UNDERSTOOD BY

DATE 4-21-19

TEAM REFLECTION

Kaileigh's Last Entry

Robotics has taught me to try any and all new experiences. Being the newest member of Y-naught, I was definitely intimidated at first. But, the warm environment set forth by the team led me to enjoy my time here. The countless nights spent together and the endless dance parties all contributed to me finding new friends, and a newfound passion. Going forward I know I want to incorporate some form of engineering in my life, whether it be for fun in college, or through research.

Drive Team

-University of Delaware '23
Biological Sciences

Shir's Last Entry

Robotics has taught me the importance of collaboration. While many times in school it is easy to get overwhelmed in your own work, working on Y Team has helped me understand the importance of dependence. Whenever I feel confused or stuck on a problem, I know that I have a team of six other talented girls to depend on. I now believe that dependence is just as important as independence, and I can't wait to bring that collaboration to my work in the future.

Programmer

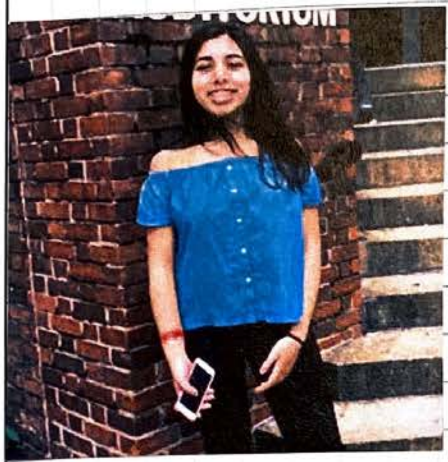
- California Institute of Technology '23
Chemical Engineering

Continued to page

Continued from page

TEAM REFLECTION

<u>Nafessa's Last Entry</u>



Robotics showed me what it means to be a part of a team. Working with a group of 7 girls, I learned how to collaborate on projects and how to come to decisions that we can all agree on. Even though robotics can be stressful at times, we need to stay calm and work through our problems logically. The ability to think under pressure and to work as a unit with my teammates are skills that I will carry with me for the rest of my life.

– University of Pennsylvania '23
Finance

Continued to page

SIGNATURE

DATE 4/21/19

DISCLOSED TO AND UNDERSTOOD BY Nafessa J

DATE 4/21/19