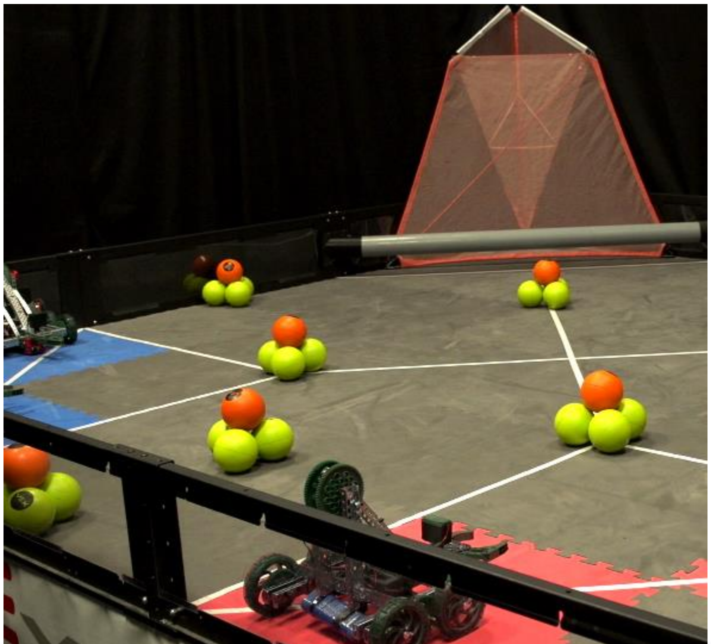
BRAMPTON ROBOTICS

EDUCATION

1104G

2015-2016 “Nothing But Net” Design Book

TABLE OF CONTENTS



VEX COMPETITION ENGINEERING NOTEBOOK

Team 1104G Vex Competition Robot

The Vex Robotics Competition has turned out to be a fun and excellent learning experience for students in team 1104G. This notebook was created to reveal our integrated and intricate project on the Vex Robot.

SKYRISE

INTRODUCTION OF SPICE DESIGN PROCESS

ROBOT STRUCTURE AND MOTION

RESEARCH AND ROBOT DESIGN PROBLEMS

MATERIALS USED

STRATEGIES AND TEAM BACKGROUND

PREPARED BY:
1104G, DISCOBOTS CANADA

10 Judson Gate

VEX GAME: NOTHING BUT NET

RULES AND REGULATIONS

- It is played on a 12′ x 12′ field
- Two alliances, one "red" and "blue" composed of two teams each face off against each other
- The objective of the game is to attain a higher score than the opposing alliance
- Each match starts with a 15-second period where robots autonomously operate, and then a 1 minute 45 seconds driver-controlled period begins
- At the end of the match, teams can low (4″ off the ground) or high (12″ off the ground) elevate their partners to earn bonus points
- The field consists of two goals in opposite corners of the 12′ x 12′ square, each one made up of one low-goal and one high-goal
- Ten pyramids of balls are spread across the board, each one consisting of three normal balls and one bonus ball
- Four balls can be preloaded on the robots when the match begins
- Points can be scored in the following ways:
    - Landing a normal ball in the low goal
    - Landing a normal ball in the high goal
    - Landing a bonus ball in the low goal
    - Landing a bonus ball in the high goal
    - Elevating one robot from an alliance 4″ off the ground for a low elevation
    - Elevating one robot from an alliance 12″ off the ground for a high elevation
- Each alliance receives 24 balls as driver control loads
- 30 balls and 10 bonus balls are placed on the field

SCORING

Scenario	Points Awarded
Ball in Low Goal	1
Ball in High Goal	5
Bonus Ball in Low Goal	2
Bonus Ball in High Goal	10
Low Elevation	25
High Elevation	50
Winner of the Autonomous Period	10

S.P.I.C.E

BRAMPTON ROBOTICS

EDUCATION

1104G

2015-2016 “Nothing But Net”
Design Book

S.P.I.C.E IN OUR DESIGN

WHAT IS SPICE?

SPICE is a design process model that separates different stages of a project (usually involves with engineering design). It is used to bold each individual stage of the project to easily distinguish the problems and flaws and relationship between each stage. SPICE makes it so that we are able to go into each stage of the process with ease. In this section we will talking about this design process and how it has been incorporated to help build our entire robot and so on.

S – Situation and Scenario	This is where the situation of the problem is stated. Most of the research is done about the situation and ideas are brainstormed.
P – Problem	This is where the problem is stated and the problem is discussed and ideas are beginning to be tested. The focus of the technological solution is also discussed.
I – Investigation	Brain storming and drafting usually begins this time. The performance specifications and constraints are determined.
C - Construction	Designing (CAD) and technical drawings are made. Sometimes a mock up is created, a prototype, and then the final product. In our case we created multiple designs before arriving at the final design we used.
E - Evaluation	This is where the other parts of SPICE are investigated and flaws are found. The product is to be improved to change any of these known flaws. It is a continuous process meaning it should be used in every stage of SPICE.

SPICE IN OUR DESIGN: SITUATION, SCENARIO, AND PROBLEM

Our situation was to create a fully functional robot that was able to participate in the Vex game "Nothing But Net" This robot will be used to perform in various Vex Competitions and be able to work well in different situations that features different forms of alliances and opponents. The robot's performance must also be flexible in terms of strategic execution which will be discussed in the strategy section.

The robot must be light weight, stable, quick, and very precise since it needs consistently score balls into the low and high goals. It has to be robust in order to last through multiple matches without requiring extensive repair. The robot also needs to be very quick in order to collect all the field balls as fast as possible. According to the game, the robot must be able to somehow place balls into the net. The robot must also be multifunctional and adaptable to any situation, giving it both defensive and offensive capabilities to overcome any obstacle standing in its way. Giving it such flexibility allows the driver to control the pace of the game, and react to both his/her ally's and opponent's moves to gain the edge in battle. Finally, the robot must follow the 18" x 18" x 18" size restriction, except this year, the 18" x 18" x 18" limit is placed on the whole match except for the last 30 seconds in the loading zone. This means that your robot cannot exceed size for normal match play. This is a huge obstacle that teams must be able to overcome.

We began to consider the types of robots other teams had created in past years so we could innovate and build/improve the design to suit the game. Being a rookie team, we strive for success and hope to obtain what we came for. We gained inspiration from each team's innovations and added our twist to the design to create an original and effective design for the robot.

SPICE IN OUR DESIGN: INVESTIGATION

Research was then done both on possible robot designs and the entire game so that we could find one of the best designs possible and adjust it to our driving, as well as building skill. We considered both a variety of different launchers including: a double flywheel, single flywheel with a hood over it, and a puncher. Some launchers have advantages to others but it all came down to the most efficient one. We decided to create a mind map to brain storm. Our ideas that had been considered are laid out here. Further discussion on the ideas will follow the Mind Map.

CHASSIS / DRIVETRAIN

For this year's game, since we are not allowed to expand out of the 18" cube size limit, we decided that we want to create a fast, and very light weight drive due to the lack of a lift, or expanding this year. The things we had to think about were what type of wheels, and what type of ratio we were going to use for the drive train (ie. Geared for speed/torque.) But since this year, the game is about who can get to the balls on the field first and score them, we decided that a drive train either geared externally or internally for speed would be our best bet.

We created a chart for the different wheel configurations, and a chart for the different ratios that we could possibly use to discuss the pros and cons of each.

DRIVE TRAIN GEAR RATIOS

	2:1 (INTERNAL TORQUE)	INTERNAL HIGH SPEED	INTERNAL TURBO
FREE SPEED	200 RPM	160 RPM	240 RPM
STALL TORQUE	7.3 LBS	9.2 LBS	6.2LBS

As you can see, there is a direct correlation between stall torque, and the speed of the motor. When the speed of the motor goes up, it loses torque. This could be very problematic if we use a very high ratio for our drive train, and means that we will have to create a light robot. The ratio we choose will also depend on what type of wheels that we use. This is due to the fact that many of the wheels are different sizes meaning ratios will be different as smaller wheels won't travel as fast as a bigger wheel on the same ratio.

WHEEL CONFIGURATION

After some brainstorming, we came up with 3 types of wheel configurations that would suit our strategy well (see pg.44) The 3 drive configurations we thought of were; six 3.25" wheels with a traction wheel in the middle, a standard 6 wheel Omni drive, and a four wheel Omni drive with two 2.75" Omnis at the front. We created a cad for all of these drives and explained pros and cons below.

BRAINSTORMING

BRAMPTON ROBOTICS

EDUCATION

1104G

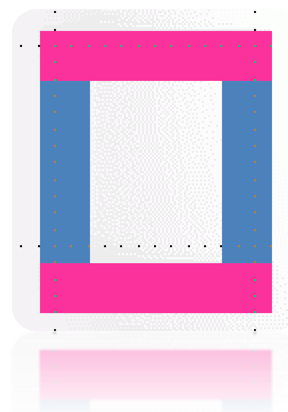
2015-2016 "Nothing But Net"
Design Book

TYPES OF CHASSIS DESIGNS

The shape and structure of a drivetrain is influenced by a number of key factors, including but not limited to:

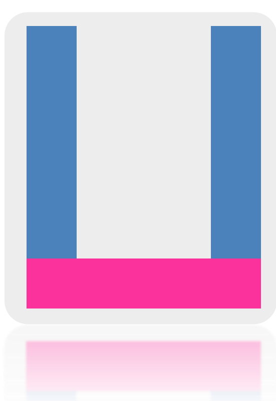
- The objective of the game
- Size constraints
- Shape of the field, in order to maneuver through it
- Possibility and strength of external collisions

Square or Rectangular Base



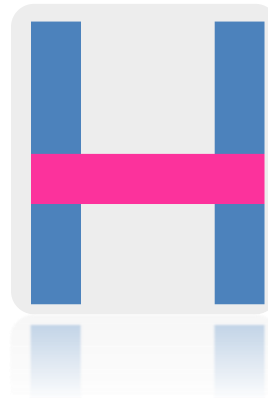
This base is structurally sound, and can help the robot become agile if it is built to be small. It can work with tank treads, and even up to a 6-wheel configuration. However, it allows no space for an intake system, or a grabber since the inner section of the robot is blocked off.

U-Shaped base



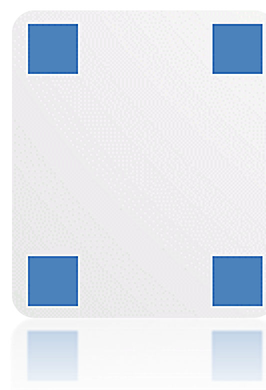
This base enables the robot to use a grabber, arm, or an intake system due to the wide space in the center of it. Issues with this chassis are that it is not very strong near the front, where it is open. It requires further bracing to prevent extensive damage caused by other robots. Also due to its shape, it makes it very difficult to retrieve objects that are stuck in corners. Looking at the design itself, placements of loads must be carefully considered. The chassis itself is asymmetrical, and so putting too much weight into one section could definitely cause a large amount of stability problems. It can fit up to a 6-wheel configuration, and overall is a strong design choice when used correctly.

H-Shaped Base



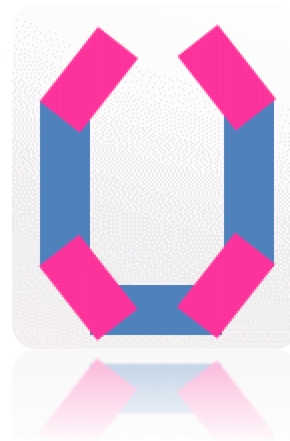
This base is structurally robust, offering a symmetrical design with space in the center for additional attachments such as an intake system or arm. This base also has trouble grabbing game pieces which are stuck in corners, and it offers less space in the center for a singular system to be placed. However, given the placement of the central bar, it offers a lot of room to install multiple functioning systems, making it more useful for certain game modes. Examples of this could be an intake and launcher system placed on both sides of the central bar, allowing for an efficient design which can truly put this base to its full use. Again, the base does not have many structural concerns, however it must be used in the correct situation to get something out of its design, or else it will have gone to waste.

Point or Pin Base



This base consists of four pillars moving onto the ground, down from a tray. This design is very useful for collecting game pieces that are put at a tall height, since it can quickly grab each one through some sort of a hook, or intake system. This design has a different purpose altogether, separating it from the earlier options shown above. Those bases are all used to launch, or grab the game pieces from a low height, and this one is made for competing on a very high playing field. For the "Nothing But Net" Competition, this design would not be very good at all, due to its structure. Having to lift a robot such as this is very difficult, and the system used to collect and launch the balls would have to be very intricate in order to pull it off. It just requires too much work to create, and it is a very risky design itself. However, if it were to be used in a different sort of game, then it could be very useful. The key to perfect this design is tweaking its mechanical components to structural perfection, since the center of gravity will be very high, and with most of the weight being installed near the top, it could make it very easy to tip over the robot and render it useless.

Holonomic U-Base



This is the same as the U-Base, but with all of the wheels turned to a 45-degree angle. This design makes it difficult place sensors, and build attachments around simply due to its shape. Also, the opening of the “U” at the front is much narrower, making it difficult to add many components to the robot’s attachments. The multiple changes in angle can cause many design issues since parts such as gears may not work as effectively when spinning at angles, and so forth. However, if done correctly, this design has a great amount of potential to dominate the autonomous period. Due to its angled wheels, the robot can move at exact X and Y coordinates throughout the playing, and this makes it much easier for designers to accurately program the robot, and maneuver the map with skill and precision. The downside to that is learning to operate the robot as a driver, since using the X and Y coordinate system can be tricky.

Holonomic X-Base

This is very similar to the Holonomic U-Base, however offers a bit of a more open design, enabling designers to customise it to even greater lengths. It shares many of the advantages and disadvantages with the previous Holonomic Base. However, considering that it moves along such a coordinate system, it becomes possible to install systems on both sides of the robot for double the efficiency (such as two arms to grab even more balls). This again is difficult to execute, but very effective when done properly.

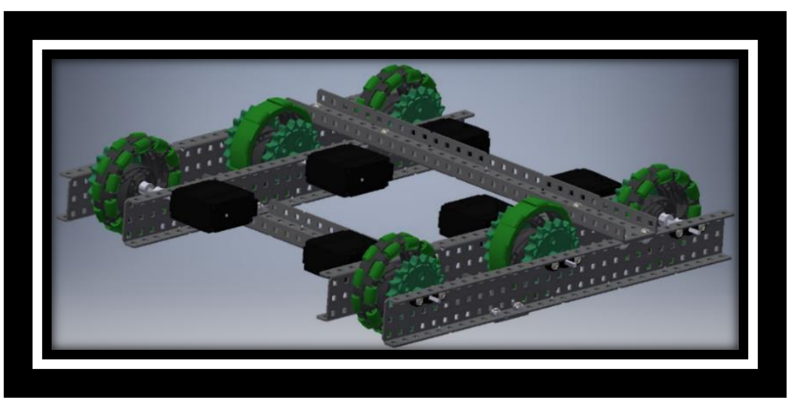
Modifications of Any Base-Type

Of course, all of the above are simply guidelines and starting points for designing the base of the robot. Any of these shapes can be combined, edited, and refined until the team is content with the design that they have. Each design is effective in its own way, and at the end of the day, it all comes down to the objective of the game which teams are competing in. There is no perfect design for any competition, and so finalising on a set design comes down to a mixture of extensive planning and experimenting.

Drive Base Conclusion

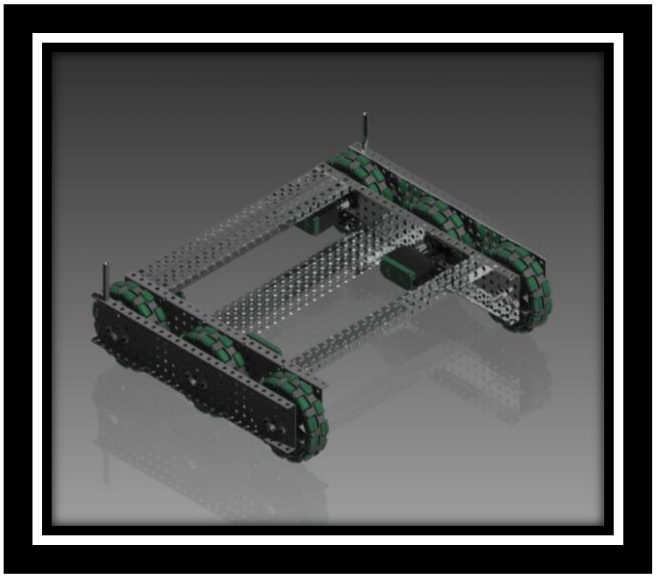
Out of all of these designs on the drive base, we decided to go with the U shaped normal base. This is because it allows the most room for the intake at the front of the robot, and has a high amount of space for us to create structural supports to make sure the launcher is kept rigid. The reason why we chose this over the others was due to the fact that it is extremely simple, and easy to pull off, and as the great Leonardo DaVinci said, simplicity is the ultimate sophistication.

3.25" WHEEL DRIVE



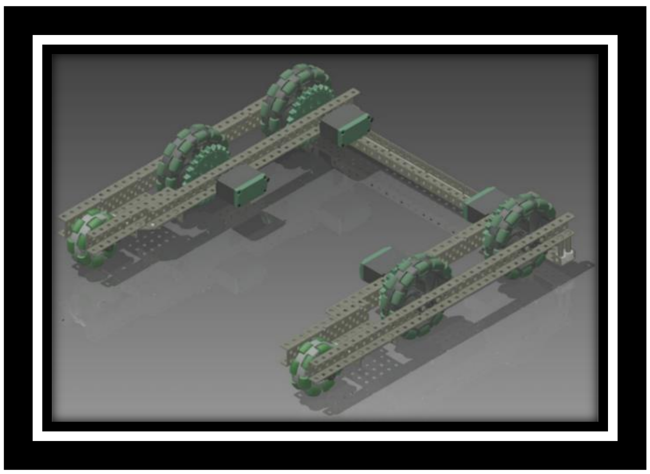
- Extremely lightweight.
- Harder to push sideways due to the traction wheel
- Not as fast as 4" wheels since there is a smaller circumference.
- Front of the drive is more narrow than others, meaning less room for intaking balls.

STANDARD 6 WHEEL OMNI DRIVE



- Very compact and strong
- Less stress on motors when turning as the rollers on the Omnis let the drive slide around. Instead of a traction wheel which has a lot of grip on the floor.
- Exponentially faster than most other drives, it has the highest circumference on wheels, meaning it will travel further per motor spin.
- Has a wider front since all the motors are direct, making each side narrower creating more room for the intake.
- Heaviest type of wheels for drive.

STANDARD OMNI WHEEL DRIVE WITH 2 SMALL OMNIS



- Light weight.
- Very fast.
- Wider front of robot as a result of the 2 unpowered small Omnis.
- Allows the robot to start off the tile as the 2.75" Omni hovers off the tile, giving us a head start in autonomous
- Low center of gravity

CHASSIS TESTING

After weighing out the pros and cons of the different wheel configurations and ratios, we decided to do the ultimate stress test. Using the 4" Omni wheels and 6 motor turbo, we put on a 10 lb weight, as well as a 5 lb weight to simulate an extremely heavy robot.

Before the test, we believed that it would run for less than a minute and easily stall out due to the insane speed. But after a while of driving, we were all in awe. The robot did not stall out at all, and was still going strong after 3.5 minutes at roughly the same speed as what it started at. We immediately started to lean towards turbo gears for the drive train, as we knew for a fact that our robot would not be more than around 12 lbs.

We then conducted the same test with 4 motors on turbo gearing. The robot ran for a total of 2 minutes before it stalled, which is the full length of a match, and is very risky. From this test, we concluded that 6 motor turbo drives are viable for drive trains, especially for a game with light robots that need to be fast, unlike a game such as Skyrise.

CHASSIS CONCLUSION

After some testing, and weighing out pros and cons we decided to go with a turbo geared drive with 2 small free wheels at the front. This was a very easy decision for us after testing the drive train with weights. Turbo gears are clearly capable enough to support an entire robot without stalling. Another reason we chose turbo gears was because of our overall strategy during the game (see section....)

The wheel configuration with 2 small Omnis at the front seemed the best due to the fact that there are almost no cons with it, and more pros than every other design we thought of. It is nimble, light, wide, and extremely maneuverable. This chassis seemed like a good start to us.

THE LAUNCHER

When the game was first released, we originally had planned for a design using a lift. Our initial design was an 8 bar as it is agile, and light enough to lift quickly. After some time, when the manual was released, we quickly discovered that this would not be possible. Simply due to the fact that we are not allowed to expand out of the 18”^3.

One of the hardest things in this game is finding a way to get the balls into the net. We created some ideas for a launcher. The first design was a dual flywheel, then was the single flywheel with the hood over top of it, and lastly, was the catapult.

We began some prototyping for both of the flywheel designs, and had another team in the DISCO VEX ALLIANCE (523T) test the catapult and compare.

DUAL FLYWHEEL

Upon the first week of coming back from worlds, we created a dual flywheel design to prototype. Since we had absolutely zero experience with flywheels, we geared it very high hoping for the best, and using 1 motor per wheel. Our final ratio was a 49:1. We used two 4” traction wheels on each side, and had 0.5” compression. The balls ended up shooting further than needed, so we tuned the flywheel to shoot perfectly into the net. The only problem with this is that even if one side is 15 RPM higher than the other, the shot will not be accurate, hence the reason we ended up switching designs.

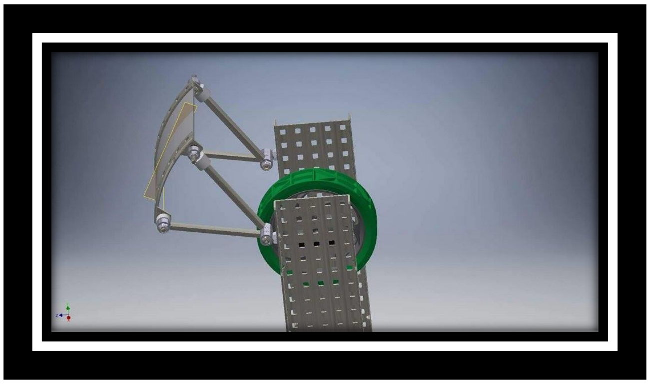
Here is a QR code of another dual flywheel in action!

SINGLE FLYWHEEL

After testing the dual flywheel and deciding not to use it, we went on to prototype a single flywheel with a hood overtop of it. Our initial gear ratio that we started off with is a 25:1, using 3 motors.

This is the first flywheel that we created. It was extremely accurate and had a long range distance, meaning that we could run motors at a low power and still scoring, which will keep the endurance of the motors high.

After testing, we tried out a different type of wheel on the flywheel; the standard 4" wheels. These wheels ended up shooting much further, despite the fact that they are the same size. We think that this is due to it having more grip on the balls.



This is a CAD of the Lexan hood we designed. This design is light weight, and has the same arc of the wheels throughout meaning that the compression of the ball will be constant.

The total amount of compression on the ball with this design would be 0.3". And after testing, this design seemed to be extremely accurate as it put a ton of backspin on the ball, allowing it to "float" in the air.

The only problem with this design is that 3 motors is not enough motors to keep the flywheel at a constant speed while shooting. So after the new rule, where you can either use pneumatics or 12 motors, we decided to dedicate 2 more motors to the launcher (total 5).This strategy worked great, we rarely ever missed a ball.

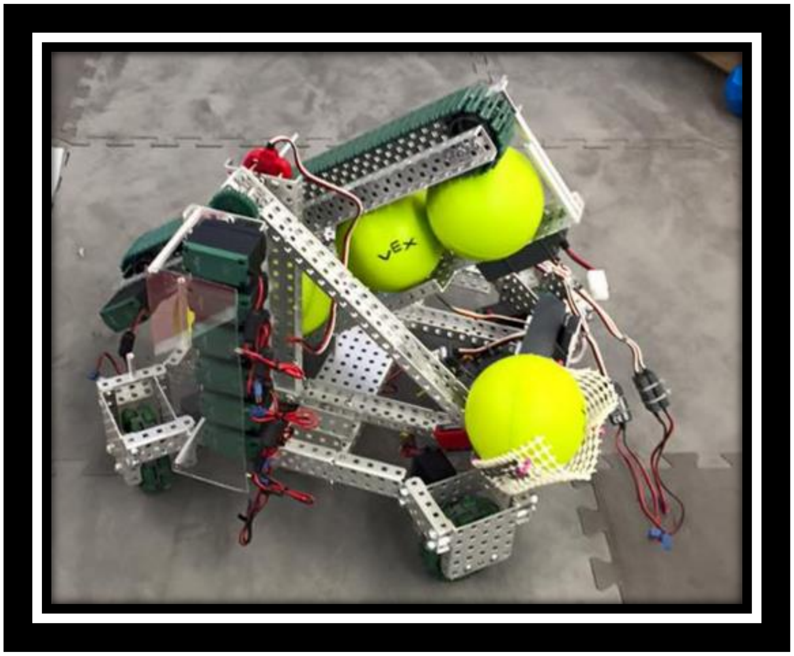
There was yet again, another problem that we faced with this design, which was how many wheels we should use for the launcher. Since a high momentum would cause the flywheel to lose less speed per shot, we tried to get the highest amount of momentum possible

Considering that momentum is the direct product of mass and velocity, if we increased the mass of the flywheel (amount of wheels) then momentum will have to go up, and we would not lose as much speed per shot and will be able to sustain a constant velocity. But if we decreased the mass, which would mean that momentum will go down, and to keep momentum high, we would have to increase the velocity which we do not want to do. Our best option seemed to be to increase mass. We ended up using 4 wheels on the flywheel as it seemed to be the sweet spot after testing.

Here is a video of our first ever test!

CATAPULT

One of the teams (523T) in the DISCO VEX ALLIANCE decided to prototype a catapult and send us the data. Harrison Freni, the team leader researched about catapults used in the medieval age, and conducted extensive research on how catapults work, and ended up creating a working vex prototype



The catapult has 6 motors on it, all geared at the turbo speed to make sure the arm has enough angular velocity to launch a ball full court

After some testing, he concluded that the catapult was extremely accurate, but it just simply is not feasible due to the 6 motor launcher, which means we will have to take motors off of another subsystem. The launcher was also not very fast, about 1 ball in 1.5 seconds due to the catapult having to wind up before every shot.

CONCLUSION

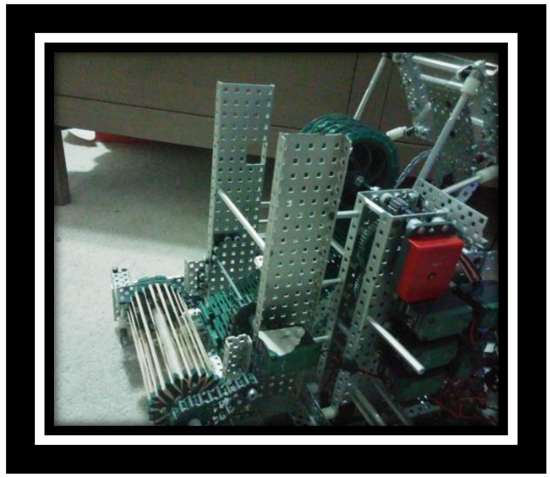
After collecting as much data as we could from ever type of launcher, there was one that just put us into awe, which was the single flywheel launcher. This design has a flexible amount of room for error and the ball tolerances (for example; Ball weight, density, squishiness). It was also the fastest at shooting, as well as the most precise/accurate.

THE INTAKE

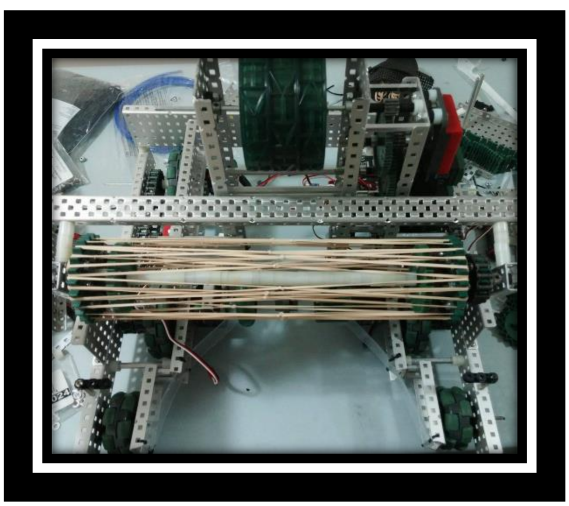
After deciding what type of launcher and drive to use, the only thing left to do was create/design an intake. Some different ideas we had for our intake were; a conveyor at an angle, a pneumatic flipper to pick up a whole stack at once, and a conveyor in the shape of an L.

We first started by prototyping a conveyor at an angle with 1 motor on turbo. It was extremely fast and grippy on the ball. We used tank tread links over normal chain as it is thicker and has rubber grips. The problem with the conveyor was that if we intake more than 2 balls side by side, it would get stuck and stall out the intake. Another problem was that since chain is meant to be kept loose, the compression would change if there is multiple balls inside, meaning some balls wouldn't even get touched by the chain!

Here is a picture of this kind of intake on our robot:



The second type of intake we created was a pneumatic flipper connected to a conveyor to pick up a full stack at once. This idea seemed very promising, but since stacks can be knocked down so easily, especially at the starting of a match, we decided to scrap it. But we know for a fact that we will use it for our first ever skills competition in Ontario since stacks won't get knocked down by us.



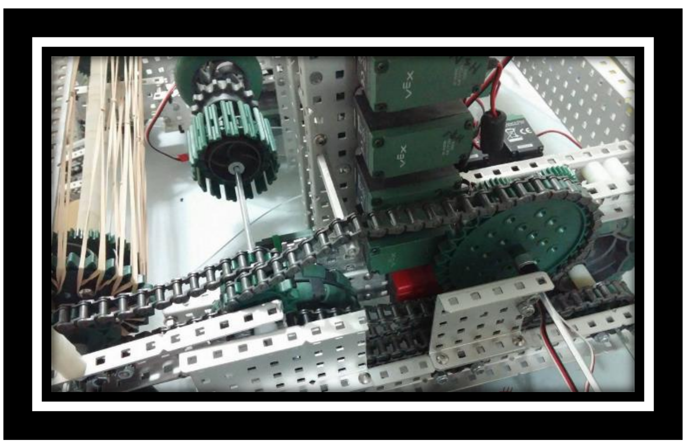
This intake was extremely good at picking up the balls, and stacks. But since most teams will be defensive and knock stacks down, we thought this idea would not be a good use of a piston, unless for skills.

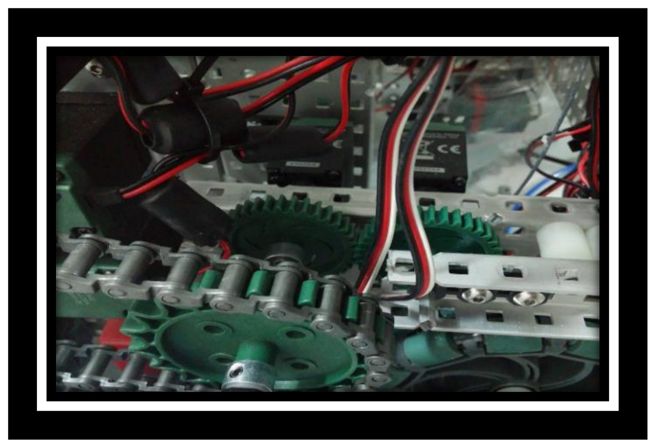
The Rubber banding between the 2 sprockets is extremely grippy on the balls, and is easy to use. It is also very wide, meaning we can pick up from multiple areas instead of just the middle.

Finally, we prototyped a conveyor in the shape of an L with a front roller. This intake was extremely fast, and did not have any compression issues such as the previous conveyor intake. We created a triangle of chain, where the ball would interact with every side of it except the hypotenuse. Instead of just picking up the balls using the ground, we used a Lexan sheet, and cut it at 6" x 20" and bent it into the shape of an L, providing low friction between the ball and the conveyor.

We decided to use 2 motors on this intake instead of 1 due to the fact that some of the previous intakes were stalling. We had to take a motor off of the launcher to accomplish this.

We created two gearboxes to test, one with a 7:2 on the conveyor (also 3.5:1) with turbo, and a 2:1 with turbo. The 7:2 was just too fast for our liking, so we decided to stick with the 2:1 on the conveyor.





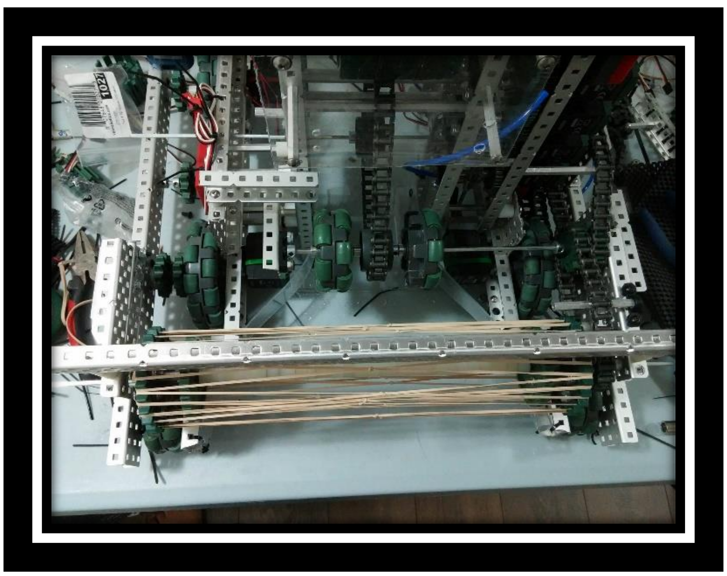
CONCLUSION

We ended up sticking with a 2:1 + turbo ratio with 2 motors on the L shaped conveyor. This is because the L shape had the least amount of friction, and the 2:1 turbo was a sweet spot for out intake, it was no too fast, it was just right.

INVESTIGATION: INTAKING MULTIPLE BALLS

In this year's game, there are stacks of balls scattered across the field. And in our opinion, we feel that to play this game effectively, we will need to either be able to pick up a stack in one go, or pick up multiple balls side by side. We figured that since we aren't using pneumatics for the early game, we will just create something to "filter" the balls into the conveyor. We tested multiple roller configurations. Some with a traction wheel, and a sprocket, and some with 2 Omni wheels with half of one Omni cut in half to prioritize. We tested all of these ideas, and ended up sticking with the small 2.75" Omnis as they have a lot of grip on the ball. Although it slows the intake down, the intake effectively takes 2 balls side by side

Here is a picture of the rollers:



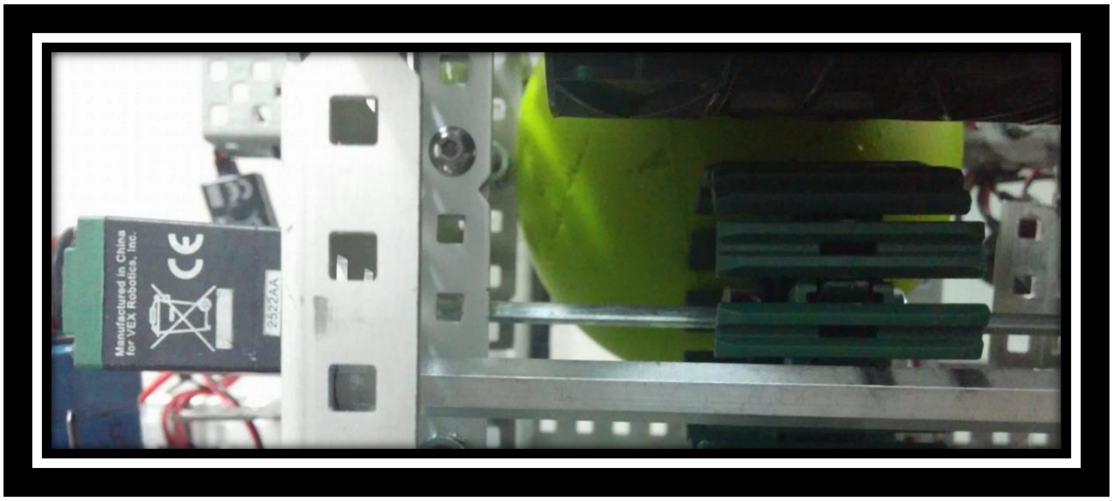
- As you can see, there is 2 Omnis right after the rubber band roller, and one of the Omnis only has rollers on half of the side. The reason for this is so that the right Omni with full rollers takes priority on balls
- Also, since the Omni rollers can move sideways, it allows the balls to move freely, whereas a traction wheel would not allow for this

INVESTIGATION: INDEXER

After testing with the robot, we reached a dilemma. Many of the balls we intake were getting shot right away since we had nothing to stop them so that we could pick up even more balls. So we decided to take a motor off of the intake and create a roller (indexer) at the top of intake so that we can keep flywheel running and not shoot by accident.

This addition proved to be extremely useful as we could keep the flywheel on for most of the match instead of turning it on only when we want to shoot balls.

Here is a picture of our indexer, it is geared turbo internally and 1:1 externally.

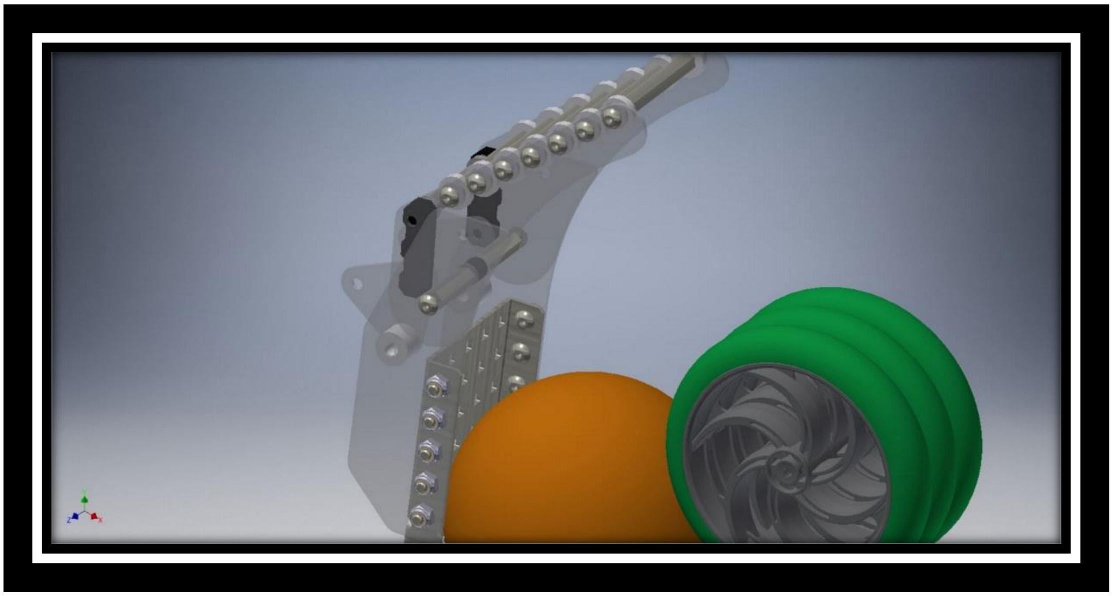


INVESTIGATION: THE HOOD

After creating the flywheel, there was yet another thing that we had to create to get the ball to launch. We could not simply just put the ball onto of the flywheel, we need a surface with a perfect amount of compression to get the ball to launch the optimal distance. The surface would have to take the ball, and carry it from the conveyor to the launching point, the hood is also supposed to have not too little, or not too much compression. If there was too much compression, the flywheel will stall out, but will shoot further. With too little compression, the ball would not get launched as it would not grip the flywheel hard enough.

We created a hood using CAD first, the hood has a short range angle, as well as a long range to provide optimal shooting. Since we did not have Lexan at first, we laser cut wood to simulate it, since the friction on the surface is almost identical to Lexan (this was only for practising purposes)

Here is the picture of the hood, it is powered by pneumatics to change the angle on the collar joint



We then moved on to a simpler design, as this was very hard to execute and needs constant maintenance due to the angle change. We looked back at some FRC rebound rumble robots and saw what they used, a particular design caught our eyes. It was a simple bent piece of metal over top of a single flywheel and was extremely efficient, as well as had a good back spin on the ball

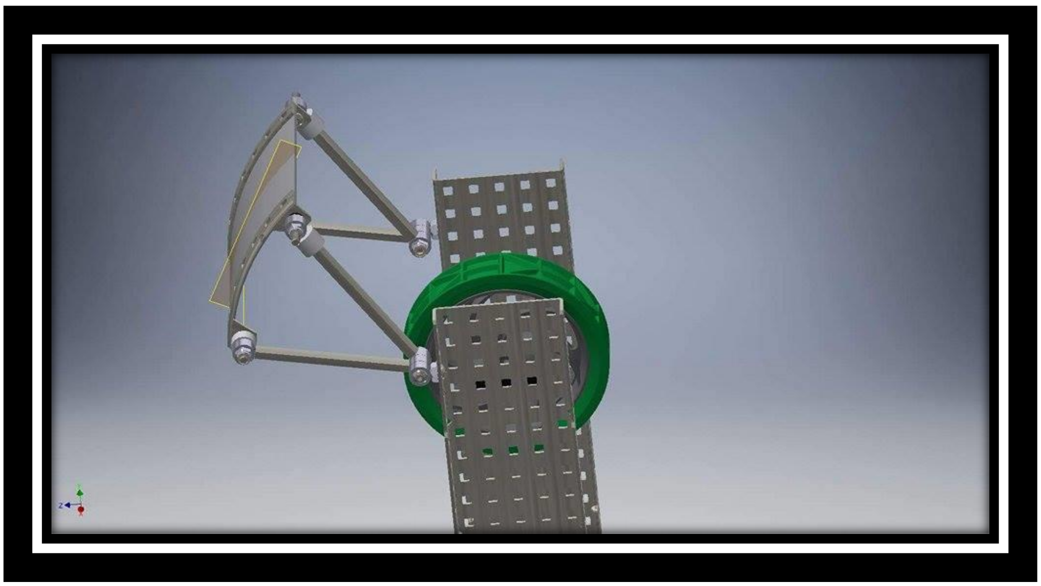
After testing for accuracy and range, we found the optimal compression to be around 0.4”. This hood allowed us to shoot 16 feet at the full speed. Which means that we can shoot at about 70-80% speed to score a goal with our ratio. This is very good because we will never need to run the motors at full power and they will last longer than motors running on full.

We then tested anti slip mat on the flywheel, the thin version. The slip mat allows for balls to have more of a grip on the Lexan, allowing it to shoot further. At full rpm in our first couple of tests, we reached around 18.4 ft total, 2.4ft further than without the slip mat.

We then tested angles on the new hood. We could shoot from directly at the bar at 2000 rpm, 2400 rpm from the middle of the field, and 2925 rpm for full courting. This is without angle change.

Since we changed to the new hood design,that meant that we do not need to use pneumatics for it, giving us yet another reason not to use pneumatics, and to use a 12 motor robot.

This is the picture of the new hood:



CONSTRUCTION

1104G

2015-2016 “Nothing But Net”
Design Book

PNEUMATICS OR 12 MOTOR

Pneumatics	2 extra motors
• 0.68 lbs per reservoir, plus pistons, solenoids, etc. • Only linear motion (no rotational) • On and off, can't create control for pistons. • Useful for brakes, indexer, and transmissions. • Can have theoretically 30 motors (10 motors on each subsystem with a transmission if created	• 0.192 *2 lbs = 0.384 lbs, about ¼ the weight of two reservoirs. • Can have power throughout match. • Don't have to pump, or make any preparations beforehand. • Add power to flywheel, or any other subsystem. • Have to use a Y cable, meaning you are drawing twice the current from a single port

We ended up sticking to 12 motors for the first competition for more power

ALUMINUM VS STEEL VEX PARTS

Aluminum	Steel
• Very light and excellent for versatility • Sturdy and very strong • Easy to cut, bend, and manipulate into different pieces. • Can reduce the weight of the robot making it faster • The reduce weight is ideal for lifting so that the robot can be lifted up more easily • Expensive • Is a weaker metal than steel • Aluminum is 1/3 of the weight of steel	• Good for a ramp, for late season lifting. • Can slow down the robot • Can potentially create stress within the tower of the flywheel causing it to fall apart or break • Harder to cut, bend, and manipulate into different pieces • Cheaper than aluminum • Is a stronger metal than aluminum

SPICE IN OUR DESIGN: CONSTRUCTION

After brainstorming and thinking of a good design we began to draw and CAD everything out. After getting out initial design we had been able to easily create any almost exact replica in real life. This goes to shows that pre designing saves a lot of time if though it consumes a lot of it. We decided that construction had been one of the toughest stages of the robot because this was where we had to actually think of how the robot would look like. This stage requires a lot of imagination and thinking just to be able to visualize the final product in our heads.

AUTODESK INVENTOR: CAD (COMPUTER AIDED DESIGN)

CADing software is practically 2d and 3d modelling on the computer. It helps us create a design with having to waste any money or time reproducing it over and over again. We were able to see a deeper understanding of how our robot works and what flaws that were happening in the robot. Using the method of constraining in AutoDesk we were able to see how individual parts moved and we also could tell how they would react with their environments.

CAD helps improves the productivity of the creation of the project and help people visualize individual parts in their design. CAD also reduces time on math calculations. Most CAD programs allow the user to measure any objects they have created in almost any way they want. Autodesk also includes features such as stress testing hich basically assesses the engineering of the design and identifies the points with the greatest and least tension using a heatmap.

We used CAD in our robot design by using the "assembly" feature of Vex. This allowed us to use Vex made parts (in the form of .STEP files) to create our robot without having the issue of re-creating the pieces individually in .ipt files. CAD had also helped communicate different ideas within our group. We were able to visually show different members of our group different parts of the robot in the CAD to allow them to fully understand before we can continue any further with robot. The CAD of the robot was done with extreme precision and because of this we were also able to extrapolate calculations from the CAD.

AUTODESK INVENTOR: ASSEMBLY AND SUBASSEMBLIES

For the CAD we used constraints to create real life interactions in the actual CAD. This helps us learn more about the part and how it functions on our robot. Constraints had also allowed us to check if there were any flaws wrong with actual robot and if we should change anything. Our CAD had used a variety of mate constraints and pre made iMate Insert Constraints. This made assembly much faster than having to constrain multiple sides to keep each individual piece in the correct spot.

We had decided to customize some parts just to add a more realistic feeling to the robot such as the Motor cover customization that spells Vex 393 Motor. We also decided to use the color functions of Autodesk to also give a realistic color aspect to the robot. We had to pre color each object because the .STEP files had been uncolored.

We use the advantage of different aspects of Inventor too such as the side menu box that gives you a set amount of details on the particular piece on the robot. We were also able to learn how to change the workflow of the program to fit our comfort zone so we can work well and finish the CAD with max quality.

The assembly files come in *.ipa* files. We decided to make multiple subassemblies that allow us to edit them without a whole lot of distractions of other pieces coming in the way or having to turn off the visibility of pieces. This option seemed better that assembling the whole robot in one file because the part not being used would just be considered distractions at the time. Subassemblies also solved any disagreements in the program because some pieces had been configured in certain ways and could actually over constraint in different areas.

We had decided to begin with making the drive train of the robot so we could get started right away. It was a good stage to start off because it is almost impossible to attach a premade launcher or lift to a drive train. It seemed logical so we could save time on the robot build.

After adding and constraining our parts we were able to finish this subassemblies and move to others such as the intake, and launcher.

Assemblies also provide a variety of stress tests and measurements that would take very long time to figure out by hand. A major test would be the center of gravity. The program locates it and we are able to get a report of how the robot would do during the competition. When we encounter a problem we switch some parts or we move them around. This saves a lot of time because if CAD wasn't available we would have to rebuild and take apart everything until they worked out perfectly

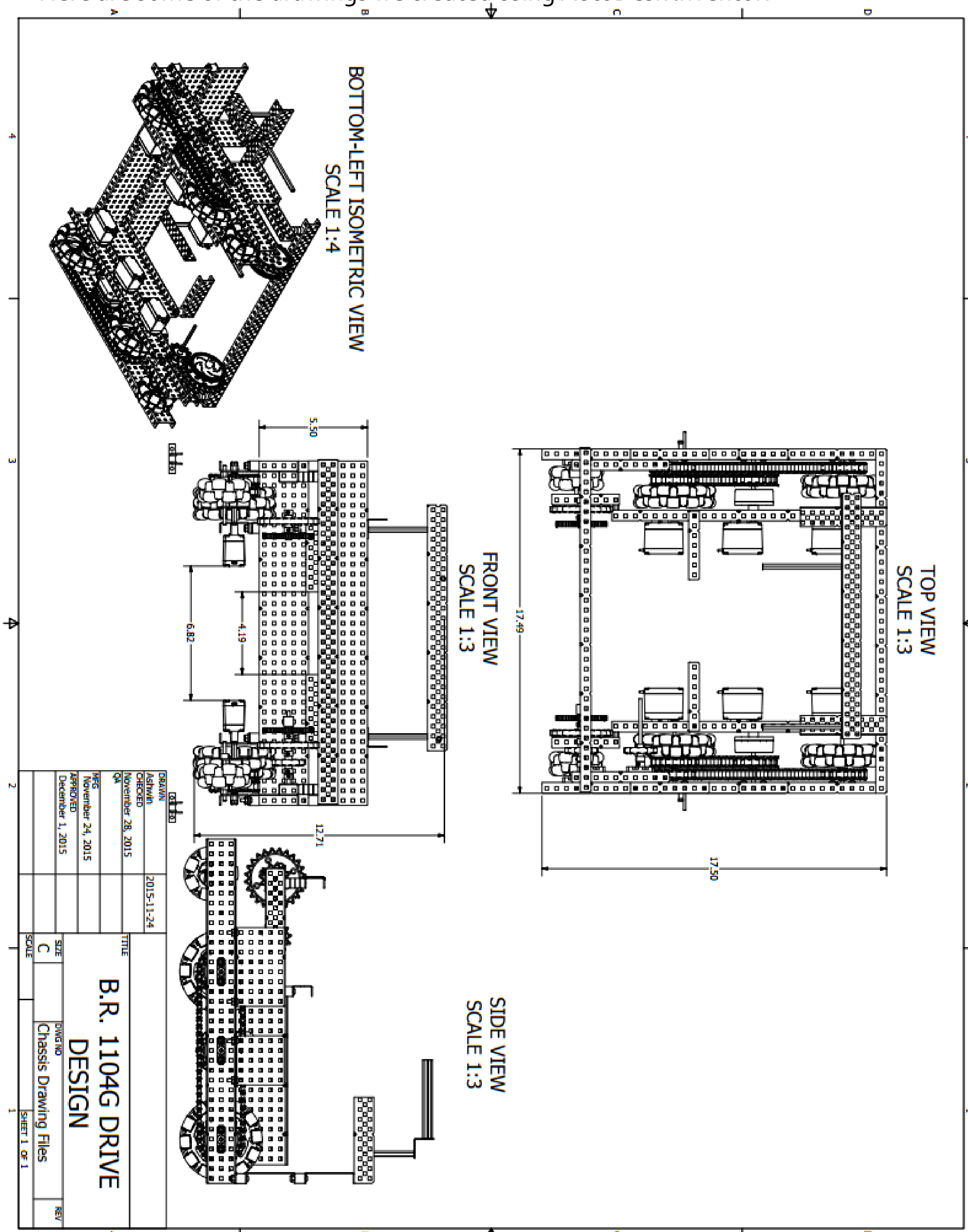
We had run different tests on the robot. Such as dynamic simulations. They give back information on various different aspects of the simulation. We also used stress and frame analysis. These features of Autodesk Inventor act as though it is putting the object out in a real life environment and how it would work out.

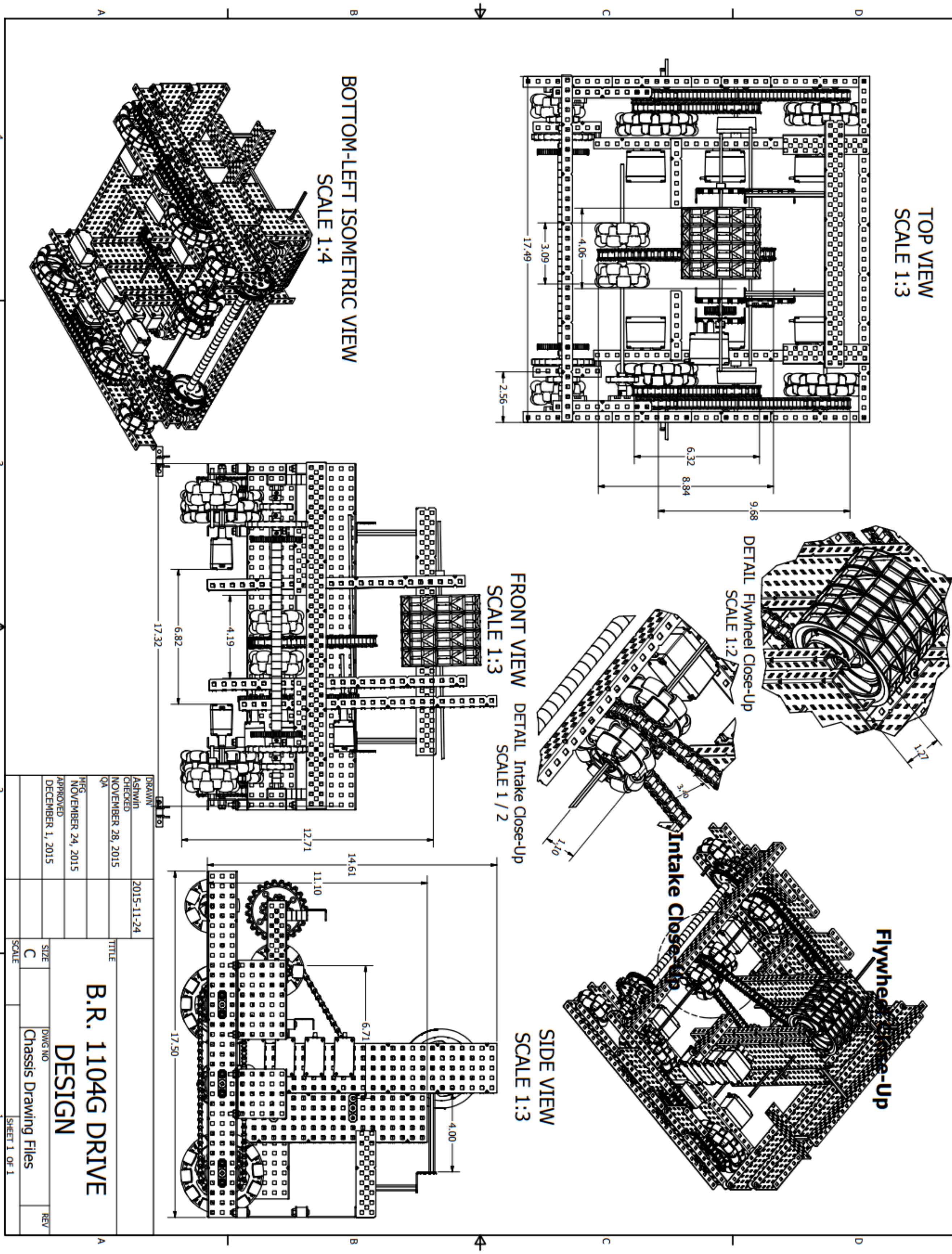
AUTODESK INVENTOR: DRAWING

AutoDesk Inventor has a drawing file function for looking at different views of a robot. The program professionally sets up different aspects of the robot onto the drawing sheet. The general views included Top, Side, Front, and Isometric Views. Isometric views are views that practically shows the robot drawn in 3D at an angle.

The drawing files also all us to put in a variety of annotations. These include text boxes and measurements. We could also add vital parts onto to robot design such as a center, hidden, and solid lines. Each of our subassemblies had a drawing that represented the parts and were all annotated to be used as reference when we were to make the robot.

Here are some of the drawings we created using AutoDesk Inventor:





TOP VIEW
SCALE 1:3
17.49
3.09
4.06
2.56
6.32
8.84
9.68
BOTTOM-LEFT ISOMETRIC VIEW
SCALE 1:4
DETAIL Flywheel Close-Up
SCALE 1:2
1.27
FRONT VIEW
SCALE 1:3
17.32
6.82
4.19
12.71
DETAIL Intake Close-Up
SCALE 1 / 2
3.40
1.10
Intake Close-Up
Flywheel Close-Up
SIDE VIEW
SCALE 1:3
14.61
11.10
6.71
17.50
4.00
DRAWN
Ashwin
CHECKED
NOVEMBER 28, 2015
QA
MFG
NOVEMBER 24, 2015
APPROVED
DECEMBER 1, 2015
2015-11-24
TITLE
B.R. 1104G DRIVE DESIGN
SIZE
C
DWG NO
Chassis Drawing Files
REV
SCALE
SHEET 1 OF 1

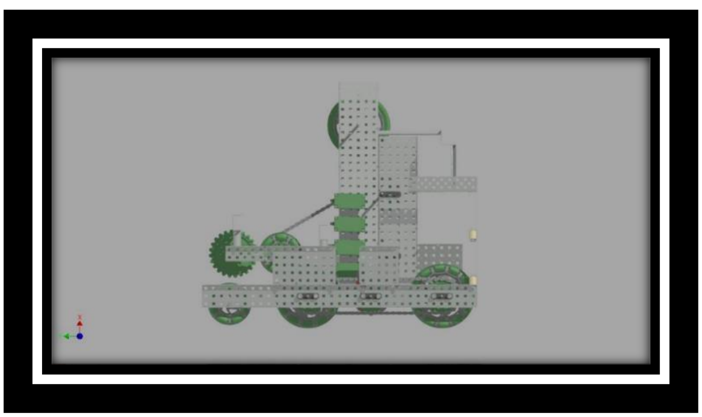
AUTODESK INVENTOR: RENDERING

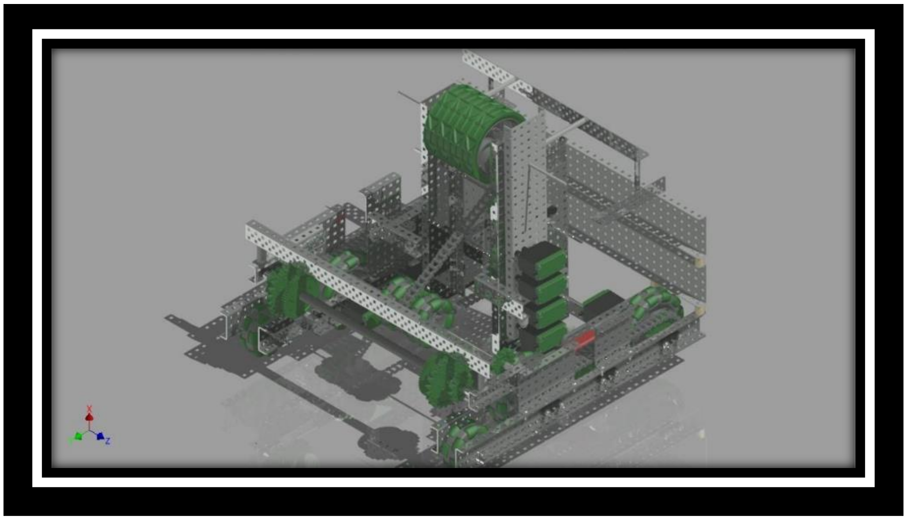
This part of Autodesk Inventor gives the CAD a more realistic look to it. The good thing about this we get to see an overall appearance of the robot. Once we see it we ask us the question: Does it look good? It doesn't just matter if the robot works well but it also must turn out to look good.

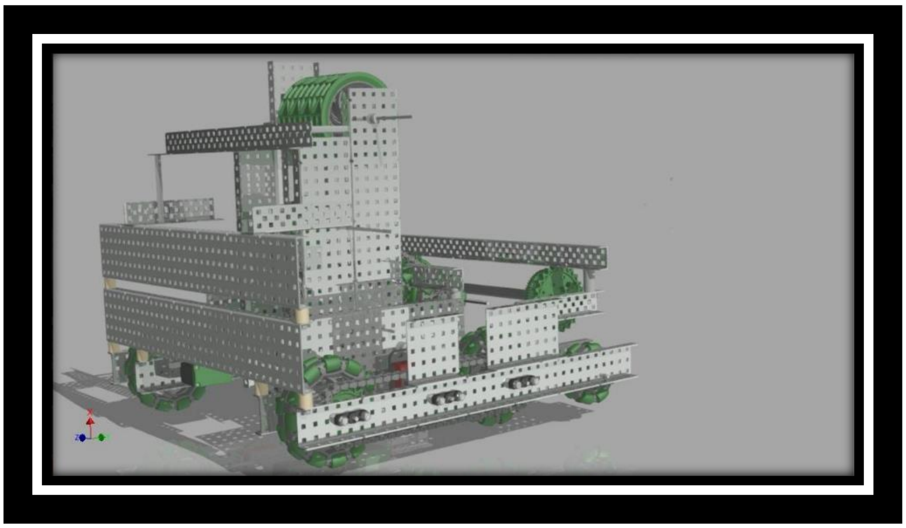
Autodesk provides different forms of rendering such as orthographic and perspective views. It also features different realism tools such as Ray Tracing. You can also add the reflections and see different shadows. Such as Ground, Object, and Ambient shadows.

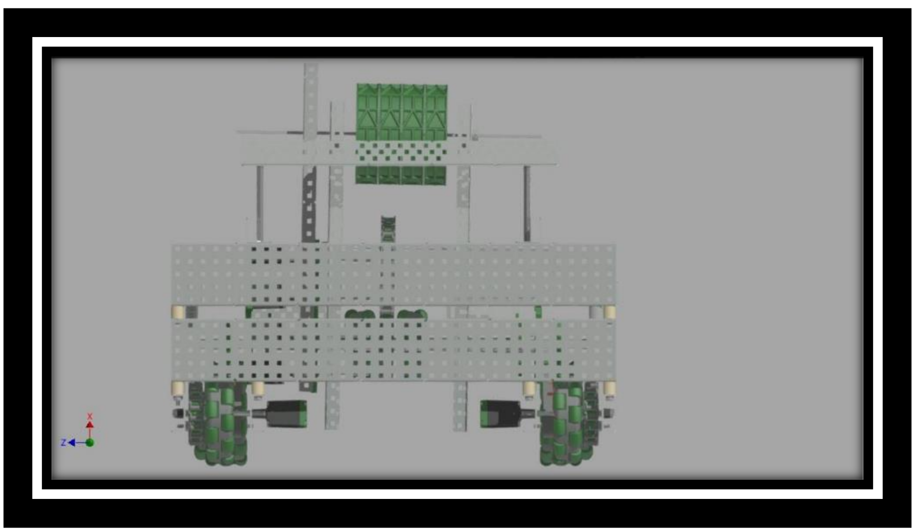
Each color and material in Autodesk Inventor contain a specific set of textures. This textures can be added and can be enhanced using Ray Tracing. You could also change the background to contrast with the CAD or you can change the background reflection.

But for our purposes, we used a program called "Keyshot" as it provides more realistic renders for inventor files, and lets you choose what resolution you want the picture at.









THE BUILDING PROCESS

The building was fairly easy because we were able to reference back the drawings we had made on Autodesk and that we knew a lot about it thanks to CAD even if we hadn't built it right. We had begun with making the drive train. It was the first part because it was practically the base of we had to build off of. We had run into a few problems during building. The financial problem of getting the money to buy the parts were one of the biggest problems.

Over the summer, we started the robot and finished a tangible version right before school started. We had at first tried Keps nuts when fastening our robot together. We noticed in practice sessions that Keps nuts occasionally fall off due to the extreme vibrations of the flywheel

By January end we had completely finished our 8 bar and started our intakes. To fix the Keps nut from coming off, we decided to replace them for vibration resistant nylocks, which are extremely dense, but worth it as they will not come off.

EVAULATIONS

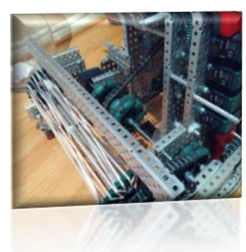
Goals

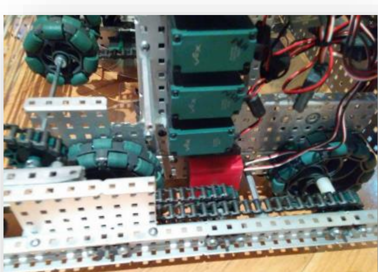
Prior to designing and building the robot, our team compiled a set of goals we hoped to achieve in time for the first regional competition. One of the numerous goals we agreed upon was the expectation to score a minimum of 300 points during the skills competition at a consistent rate. Not only is our robot able to achieve this, it is capable of achieving up to 346 points, depending on the consistency of those loading the robot with the driver loads. In regards to the tele-up period of the main match, we set a reasonable expectation of our robot scoring a minimum 250 points **on its own**. To achieve this, we would need to ensure that our robot would be capable of scoring as many points possible within 1 minute and 15 seconds, and then shoot and score at least 95% of the driver loads in under 30 seconds. In addition to completing the tele-up (1 minute 45 seconds) period of the match, we must also be able to program our robot to shoot 4 preload balls during the 15 second autonomous period of the match. The balls must all land in the upper net, scoring us a total of 20 points (5 points per ball). Our robot is able to successfully perform all these tasks, with additional time to spare. This additional time is extremely useful as it allows for some room for error as a precautionary measure. The robot is also able to shoot accurately and precisely from various ranges of the field, allowing for extreme versatility during the match.

All in all, our robot has exceeded most expectations originally set. We focused our robot on being well rounded in all elements of the competition and have successfully achieved as such plus some.

Pros

- Requires less than a total of 1 second to intake and shoot a ball
- Hood compression of 0.4 inches which only creates a loss of 200rpm per shot, though our coding takes this into account when choosing an accurate rpm
- capable of intaking and shooting 4 balls into the net in less than 1.5 seconds while taking a close range shot, 2 seconds at mid-range, and 3 seconds from a full court shot
- Suited to shoot 32 balls in less than 25 seconds in *skills competition*

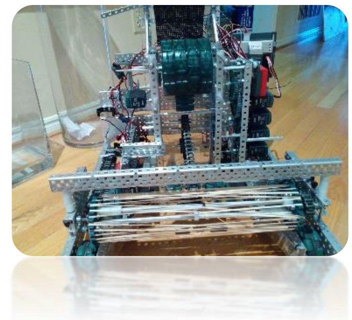




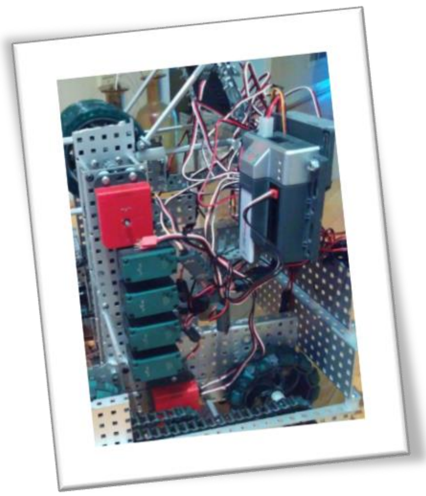
- Intake capable of holding over 4 balls at any given moment
- 6 turbo motor drive with 4, four inch rear Omni-wheels + 2, two inch front Omni-wheels
- Shooting mechanism consists of a 4 motor flywheel with a gear ratio of 25:1, controlled using a pid loop gain of 0.05

- Pid loop calculated by multiplying the set gain value with the approximate error value

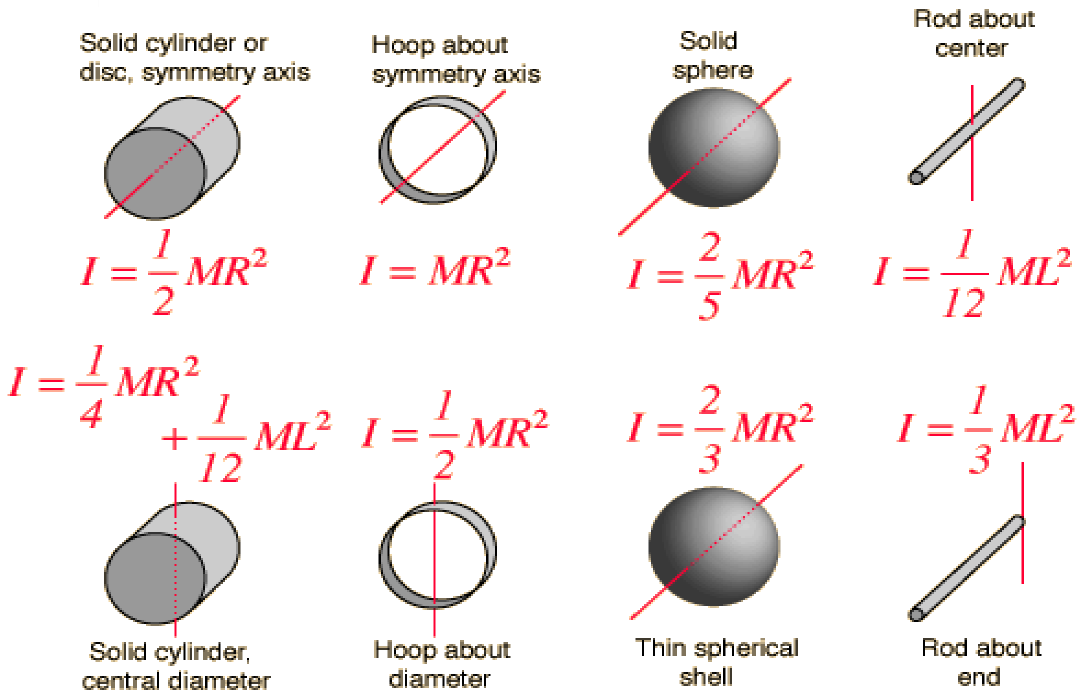
Improvements



- intake jams upon attaining 2 balls side by side as well as immediately after one another due to a large compression being formed
- The initial gain of the flywheel is slow as it requires up to 3 seconds to attain max rpm of 2925
- unfit to intake a complete ball stack at once (3 green, 1 bonus)
- Loss of rpm per shot should be decreased from the current 200 rp
- Currently lacks aesthetic appeal (ie. Wire management)



CALCULATIONS

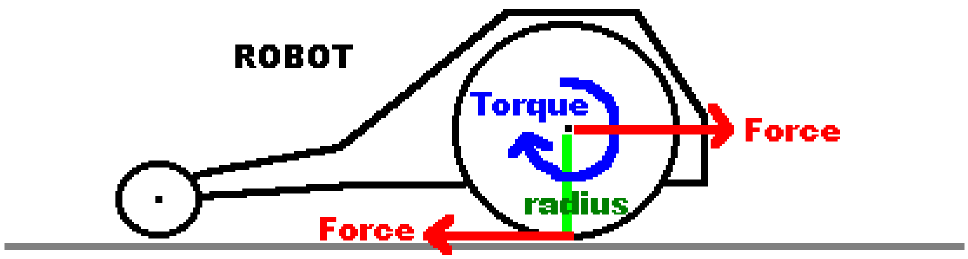


Our flywheel is a solid cylinder with a symmetry axis running directly through the wheel on the x axis. With this information, we can apply the first formula of inertia: $\boldsymbol{I} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{MR^2}$, where M represent the mass of our flywheel and R represents the radius. Upon measuring the specific dimension of our flywheel, we determined the total mass of each wheel to be 70 grams with a radius of 2 inches (5.08 cm). The mass of our entire flywheel is 70*4 (given that we have 4 wheels) for a total of 280 grams. Plugging the values into their respective variables in the formula, we can conclude that the total inertia of our flywheel is 3612.896 g. cm^2 (when calculated in centimetres), and 560 g. in^2.

- Momentum of flywheel

The momentum of the flywheel can be calculated using the formula Momentum = Mass*Velocity. Currently, the mass of our flywheel is 280 grams as previously calculated. The velocity at which the flywheel rotates is relative to the robot's location on the playing field. For long distance shooting, we position our robot for a full shot with an RPM (rotations per minute) of 2925, which converted to rotations per second (RPS) would be 48.75. Multiplying the Mass and velocity together we can see the total momentum of our flywheel to be 13 650 g r/s (gram rotations per second). Our mid-court, and close-up shot has the flywheel spinning at a total of 2400 RPM and 2200 RPM, which converted would be 40 RPS and 36.7 RPS respectively. The total momentum of the flywheel at these ranges are 11 200 g r/s for mid-range, and 10 276 g r/s for close-up shots.

- Chassis Speed (Drive train speed)



The chassis Speed is important as it allows us to calculate how far the robot will move based on the inputted code. This allows us to alter the speed of our flywheel according to the relative position of the robot in the field. The speed of the chassis also allows us to evaluate the effectiveness regarding the relationship between the mass and the functionality of the build. The chassis speed can be calculated using the basic formula of Circumference*RPM. To convert the speed to inches/hour, we would multiply the product by 60 (as there are 60 minutes in an hour). Circumference of each wheel can be calculated using the formula $2\pi r$, where 'r' represents the radius of the wheel. From previous calculations, we know that the radius of each wheel is 2 inches. The RPM of each drive wheel is 240 RPM. Knowing these values, we can plug in the respective values into the formula to determine the chassis speed to be $2\pi(2)*(240)*(60) = 180955$ in/h.

- Speed of flywheel

Using the same method we used to calculate the chassis speed, we can determine the various speeds of the flywheel for a long range shot, mid-range shot, and short range shot. The RPM's for long, mid, and short range shots are 2925 RPM, 2400 RPM, and 2200 RPM respectively. Plugging the values into the equations, we must convert each product form inches to kilometres by dividing the entire equation by 39370.1 (as there are 39370.1 inches in a km). Therefore, the formula to calculate the speed is [2πr*(RPM)*(60)]/39370.1. Upon plugging in the respective values into the equation (wheel radius remains 2 inches), the results returned are as follows:

Long range shot: [2π(2)*(2925)*(60)]/39370.1 = 56 km/h

Mid-range shot: [2π(2)*(2400)*(60)]/39370.1 = 46 km/h

Short range shot: [2π(2)*(2200)*(60)]/39370.1 = 42 km/h

- Why white lithium grease?

We periodically applied a substance called white lithium grease to the numerous gears on our robot. The grease acts as a lubricant for the gears, allowing them to spin faster and more efficiently by reducing the amount of friction the gears undergo upon rotating. Though VEX allows participants to use any grease (non-aerosol based), we decided to use white lithium grease for a variety of reasons, including its high melting point and its strong durability. When choosing the appropriate grease, we took into consideration the melting point of the lubricant. White lithium grease out-ranks competing lubes, melting at a temperature of 193°C. It is important for the lube to have a high melting point as melting will prove the lube to be useless. Furthermore, melting of the lube would leave residue on the gears, which when rotating, would cause the gears to slow down and due to the residue's adhesive nature. The high durability of the white lithium grease allows for a cushioning of the gear teeth upon contact, allowing for smoother rotations and increases its life span, allowing us to play multiple matches without the need for reapplication of the grease.

TEAM BACKGROUND

BRAMPTON ROBOTICS

EDUCATION

1104G

2015-2016 “Nothing But Net” Design Book

Name	Role
Prabhjot Lakhesar	**Team Captain:** In charge of organizing the team, planning for competition dates, gathering merchandise or resources, designing and building the robot, planning for practice sessions, networking with other teams/companies in conjunction with Manav, head financial manager, creating the engineering notebook, log sheets, and programming the robot.
Ashwin Singh	**Designer** A member of the scouting division for gaining live-time strategies and counter-tactics to face opposing teams. Main 3D modeller, responsible for assisting with sponsorships deals/offers, designing and building the robot, and creating the engineering notebook and log sheets.
Ryan Gharbaran	**Funding/accountant** In charge of the sponsorships division, assists the team with funding and public relations. Also, assists the team in the creation of the engineering notebook.
Manav Khakh	**Main scout** Important member of the scouting division, in charge of researching other teams before each competition and ensures that we have a game-plan set in stone. Head strategist and game analyst for the team, and a member of the building division.
Taijus Aggarwal	**Programmer/Fine tuning** Member of the programming division, and building division. Assists in the design of the robot by keeping track of and fixing design flaws found through practices, and the creation of the engineering notebook.
Dev Pancea	**Pit manager** Member of the building division, sponsorship division, and responsible for marketing and promotion of the team

STRATEGY

BRAMPTON ROBOTICS

EDUCATION

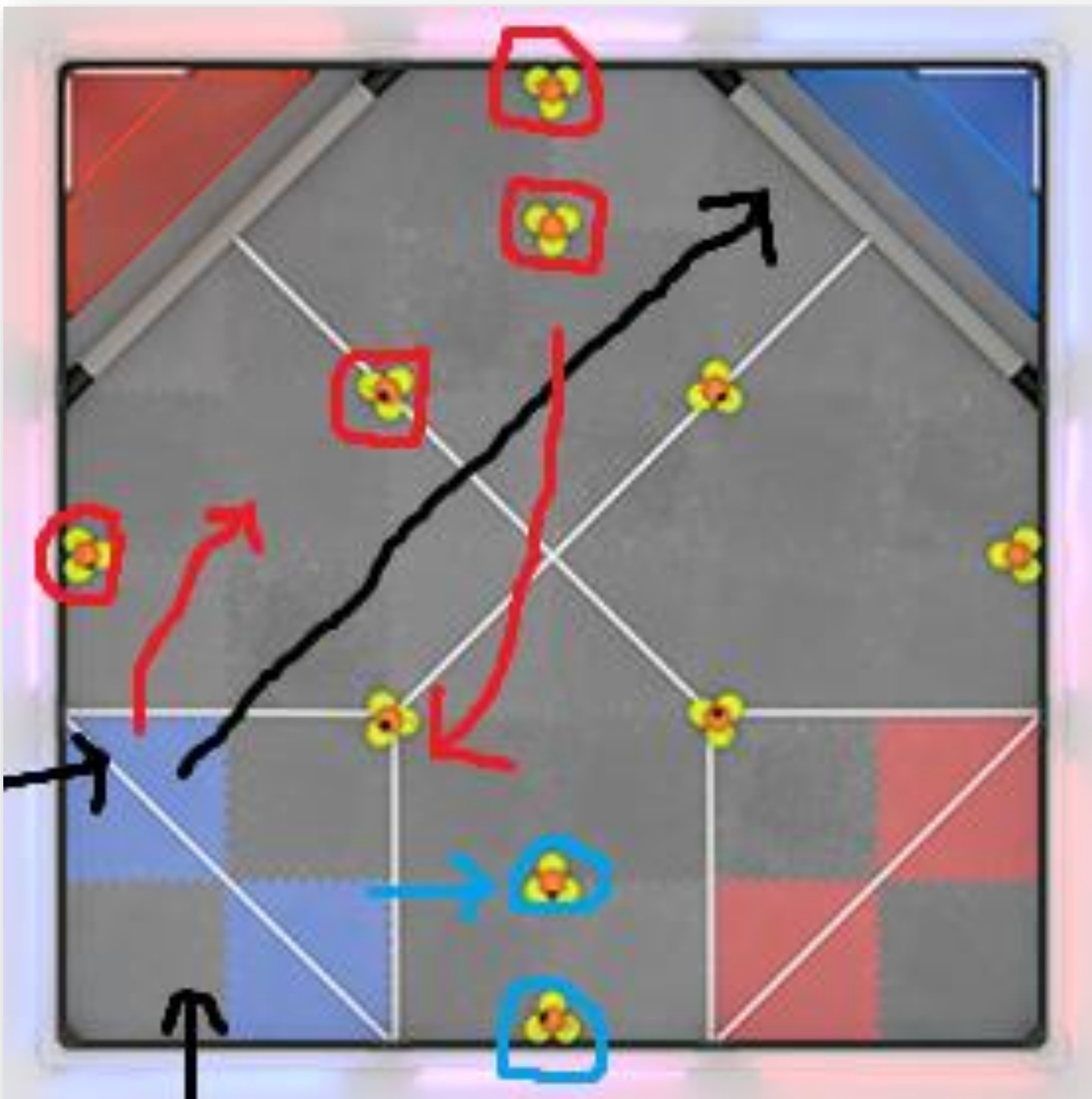
1104G

2015-2016 "Nothing But Net"
Design Book

STRATEGIES AND TACTICS

GAME TACTICS

Gain Objective Control (Diagram) **Black Arrows** represent the control load and trajectory of balls being shot into the net. The **blue** arrows represent the initial path of the robot which collects the balls. While the **red** arrows represent the second path that it takes in order to prioritise points which are easily collectible.

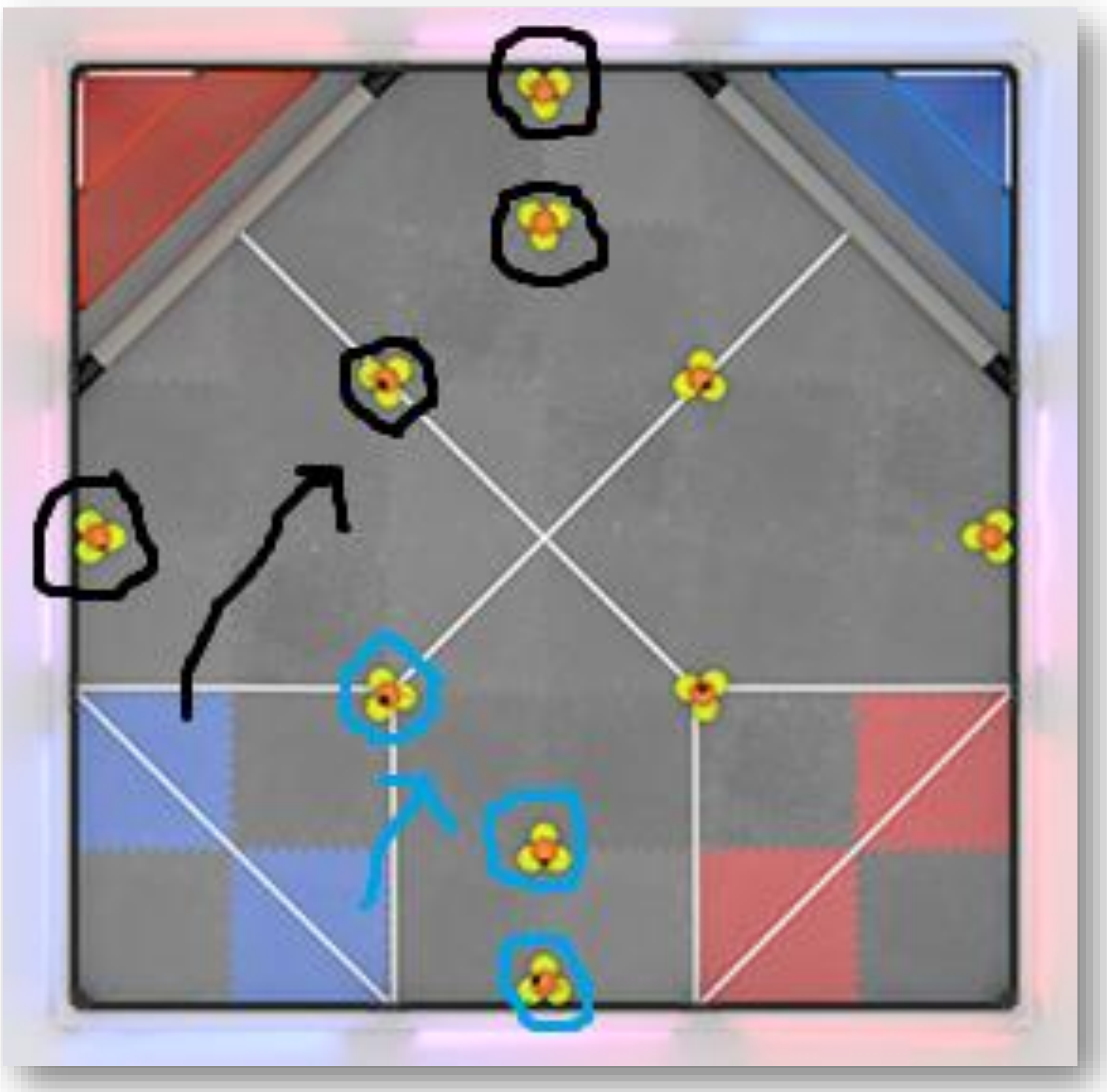


Gain Objective Control (Plan)

- One robot sweeps in as many balls as it can within a short time span
- Meanwhile the other robot is scoring balls into both the low and high goals by accessing the driver control load
- All of the collected balls are dumped into a site, where one robot can attempt to score more shots, while the other one is protecting the balls
- Denies the enemy of important points, looking for dominance throughout the driver-control stage
- The robot which zones the enemy out, can effectively control at least one quarter of the board
  - This could deny them of up to 75 points, giving our alliance an advantage
- Allows the Alliance to still be competitive even in case of a high or low elevation failure
- The robot whose job is to collect balls can have a secondary purpose of doing a high or low elevation
- This plan can only work if the robot collecting balls is faster than the robots on the enemy alliance, and if it is strong enough to withstand head-on collisions
- Very efficient in the sense that points are constantly being score, and the robot shooting balls is staying in one spot, meaning that it can consistently score goals from there
- It does not have to move around and shoot balls
- This can continue until the last 15 seconds, or until no balls are left on the field, where both robots can attempt to score points using the driver control loads

Full Offense (Diagram)

Black arrows represent the path of the first robot, while **blue arrows** represent the path of the second robot. Both are moving simultaneously, and will alternate between shots to ensure neither ball collides.

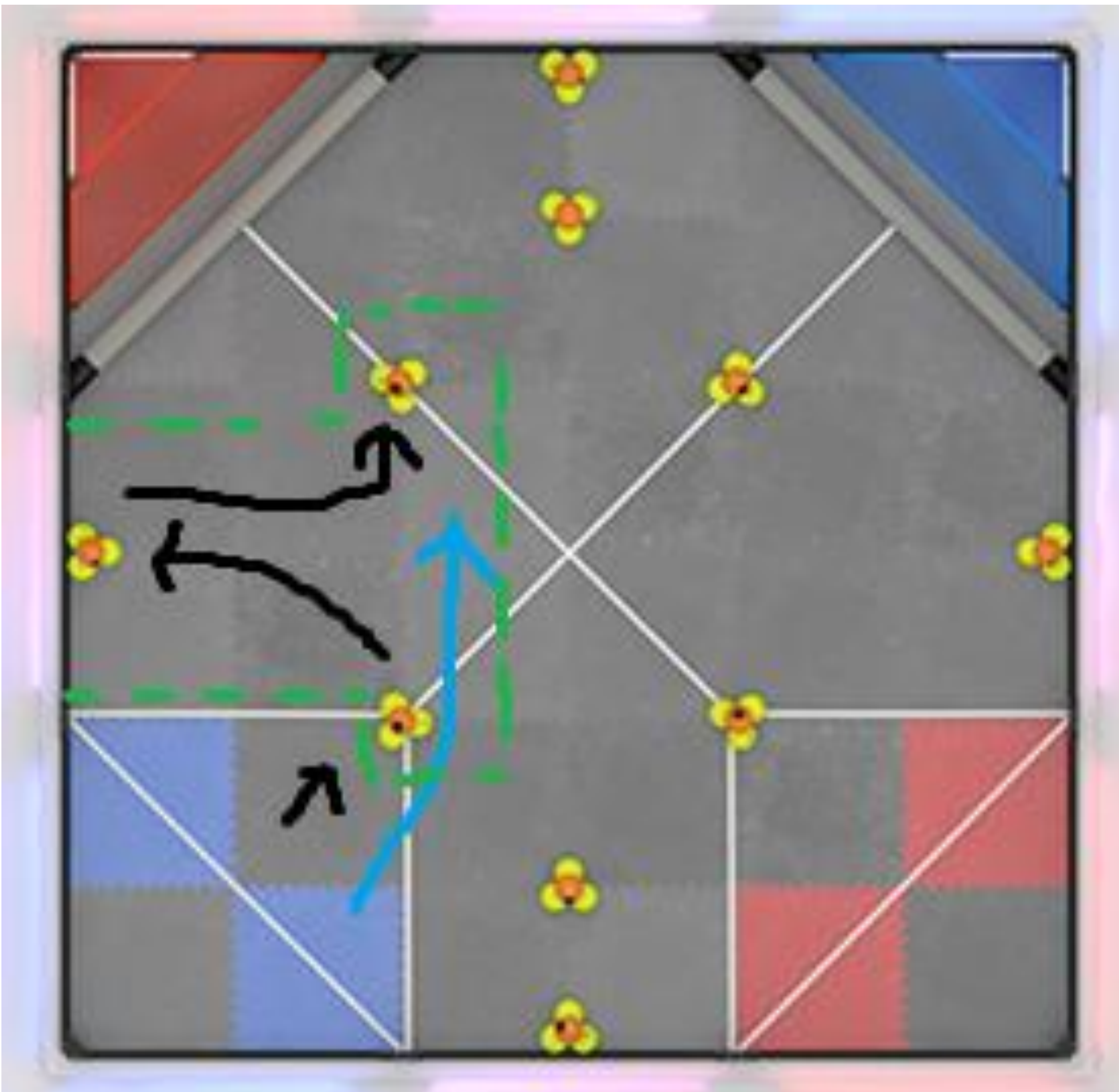


Full Offense (Plan)

- Both robots sweep balls off the game floor themselves, and shoot them into the high and low goals
- Each one aims for one goal and attempts to garner as many points as it can
- One robot can travel to its end of the board, while the other moves to the center
- This way, it maximises the number of balls that each one can collect, since they can quickly get to those points before the enemies can
- Once there aren't any more balls left on the field, the robots can use the driver control loads to maximise points gained
- This plan is good, but with the possibility of up to four robots travelling across the game board, it could result in some extensive damage being caused to either robot
- As well, with so many robot competing against each other, there is less room to maneuver, making it difficult for drivers to quickly complete the challenge
- This is a good counter-strategy for the "Gain Objective Control" one mentioned earlier, since two robots working simultaneously increases the chances of scoring more points, making it faster and possibly even easier to garner a higher score for the driver-control period
- This also requires each robot to collect their own balls and shoot from changing angles, creating possibly inconsistency per shot, and losing points overall
- Both robots must move back to a point on the map where the teams are certain it can score a goal from, or else this plan will fail
- Robots continue with this strategy till the last 15 seconds of the game
    - At this point, drivers can access the driver control load to score as many points as they can

Support and Siege (Diagram)

One robot highlighted by the **blue** arrow moves into the block zone, and covers the area which the other robot must move into. This covered area is highlighted by the **green** dotted line. The pathway of the ball which attempts to score goals is shown above by the **black arrows**.

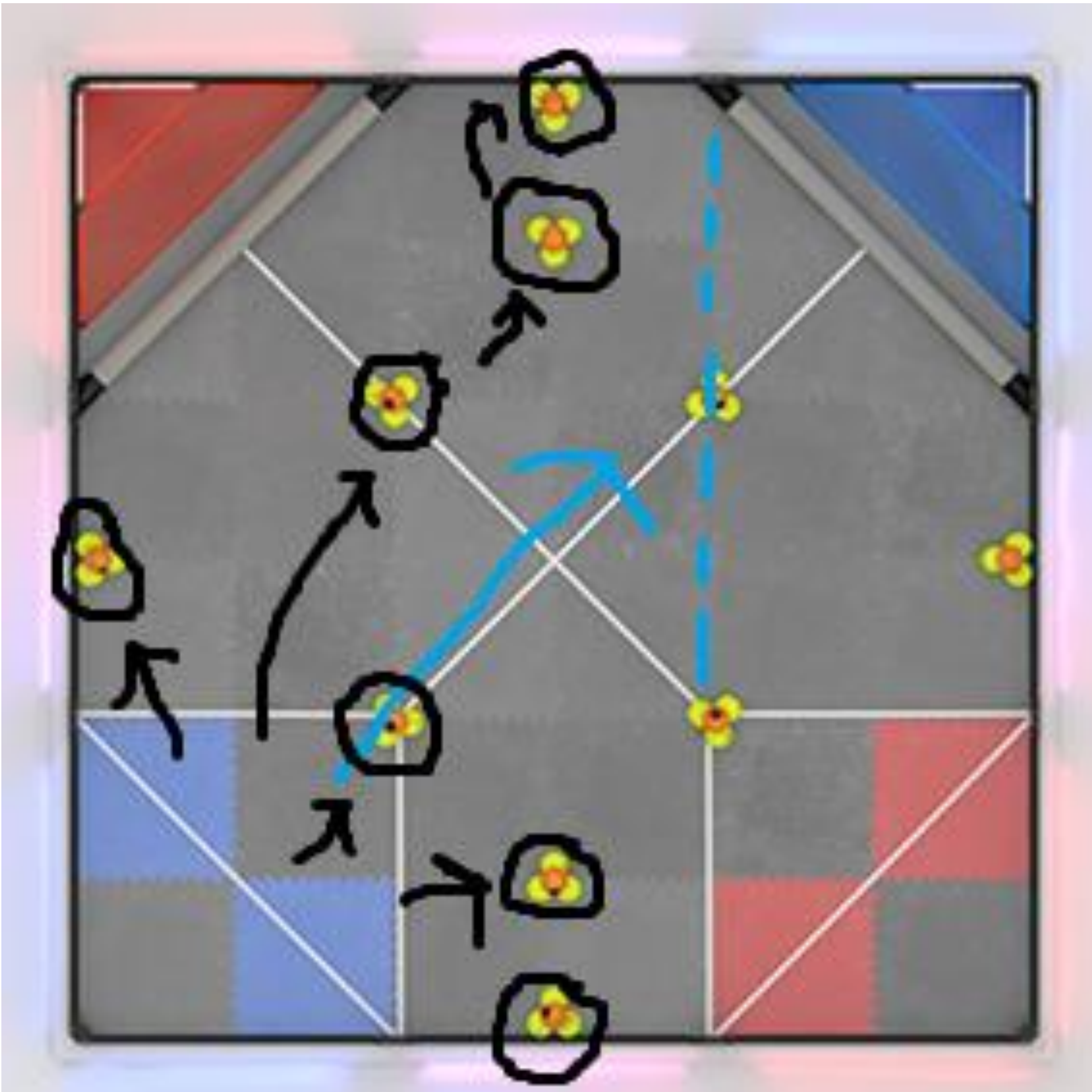


Support and Siege (Plan)

- One robot collects the balls and shoots them into the high and low goals
- Another robot stands by it, protecting its ally during the driver control period
- The main purpose of the second robot is to create space control, and deny the enemy from coming into its territory and taking balls away
- The second robot can limit the enemy alliance to only half the field, denying them of up to 100 points (assuming they can consistently score balls into high goals)
- This strategy puts a lot of pressure onto one team, and can become problematic since it is not very efficient given the short time-span of each competition
- It can consistently work well if the supportive robot does its job correctly, and minimizes the amount of points that the enemy team can score
- Again, the secondary robot can transition into becoming a powerhouse with a lift system to garner bonus points off of a high or low elevation
- This is a good counter-strategy for the “Offensive Block” strategy shown below, since it can forcefully push out the enemy alliance, and create enough space for the allied robot to score points
- The launching robot, or both robots can attempt to score the remaining balls from the driver control load in the last 15 seconds of the game

Offensive Block (Diagram)

The **black** arrows represent the path of the first launcher robot, and the balls that it will collect. The **blue** arrow represents the path of the second robot, which is to block off the enemy alliance's robots. The blue line represents the area that it blocks off.



Offensive Block (Plan)

- One robot quickly moves into enemy territory, and blocks off the path for the enemy alliance to move through
- The purpose of this robot is to buy time and simply force the enemy out, making it difficult for them to collect additional balls and score points
- The other robot will be collecting balls on its side of the map, and shooting them into the low and high goals
- Again the robot which blocks off the path for the enemy alliance can be installed with a lift system for completing a high or low elevation lift
- This is a good counter strategy for the "Full Offense" one mentioned above, as it slows down and even denies the enemy alliance of the chance to score more points
- In the last 15 seconds of the competition, both robots can shoot the remaining balls from the driver control load into the goal

Other minor strategies:

- In the last 15 seconds, shoot half of the driver loads while the alliance shoots the other half, this allows for more time for in-fielding
- In the last 30 seconds, if the opposition has a very good driver load robot, block off their alliance tile so that they can't shoot the driver loads.

Skills Strategy

Like every year in robot skills, alliance colour does not matter when scoring, so this year, we came up with a strategy to shoot at the opposite net. (ex. Blue tile shoots to red goal) This is to reduce the chance of error, and a less of a distance for the ball to travel. By shooting directly next to you, you will get a higher accuracy, leading to a higher score

After testing skills full court, and skills with the strategy we made, we compared the two in the chart below.

Runs	Full court score	Opposite net score
• **1**	210	274
• **2**	240	298
• **3**	230	305
• **4**	232	294
• **5**	226	310
• **6**	261	327
• **7**	197	302
• **8**	244	346
• **9**	203	331
• **10**	231	325

As you can see, the scores of the opposite net are exponentially higher, and the amount of balls out of the 32 driver loads are higher, as there is a bigger target due to the shorter distance. The average score was nearly 100 more than the full courting, and we beat the highest driver skills score at this point in time (298) by 48 points (unofficially), which is a great feat for us.

Skills is very important to our club, this is partially because we have 4 teams. The goal in Brampton robotics is to promote stem, and create competitive teams at the worlds level, and to get everyone in the club to worlds, our best bet is to use skills, as the top 30 teams in the world for skills will qualify.

Building Tactics

This year, in order to keep our structures completely rigid, tight, and easily fixable, we as a club came up with a few strategies in building.

Strategy 1, LocTite:

Over the past 2 years of our team doing vex, we found problems with standoffs and motors easily unscrewing themselves after vibrations, so we came up with a simple solution: LocTite

LocTite is a blue thread locker (non-permanent) which has a blue goo that you put onto the screw before putting into a motor or standoff. LocTite dries in the absence of oxygen, and expands the threading of the screw, which means that we cannot use it on things such as nuts. Here is a picture of a LocTite bottle:

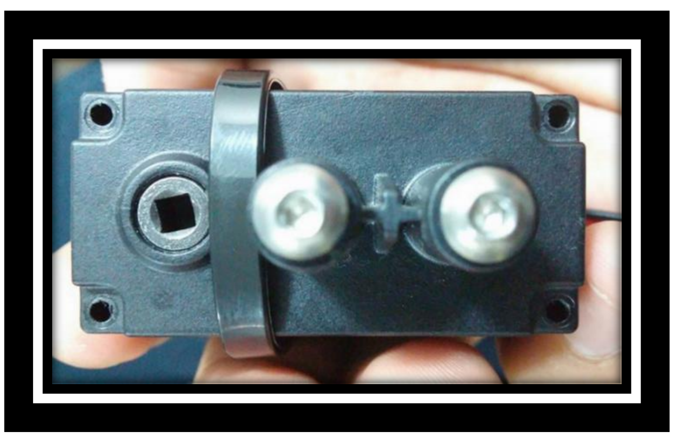
These are the LocTite bottles.

Strategy 2, Motor Zipties:

Over the course of our Vex years, we had motors constantly breaking, or needing a different internal ratio to power a mechanism. The strategy we used was to take all of the motor screws out, and replace it with 1 big zip tie. This way, if we ever needed to change ratio or had a dead motor, we would not need to take it all apart, we can just cut off the zip tie and complete what is needed.

Here is how we did it:

This proved to be very useful as the flywheel needs running motors for the full match, this lead to many dead motors which we could replace in less than 10 mins, whereas without this, it would be closer to 1 hour.



Strategy 3, Motor extension cable wiring:

In this year's game, vex came up with a new product called the extension cable retaining clips. These clips allowed us to create many different wire changes without them coming apart. These cables were great for making motor wires not come apart (such as motor controllers)

Here is a picture of them:

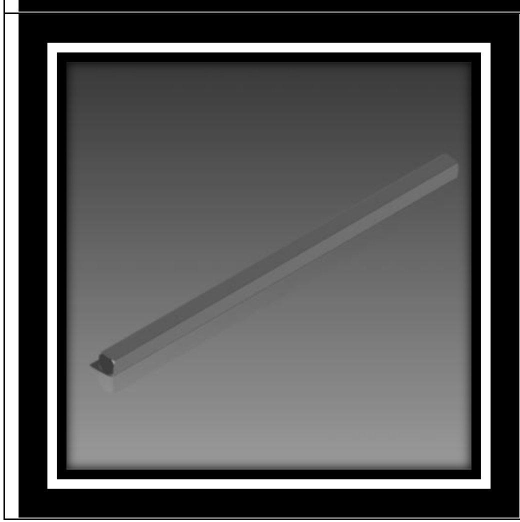
MATERIALS

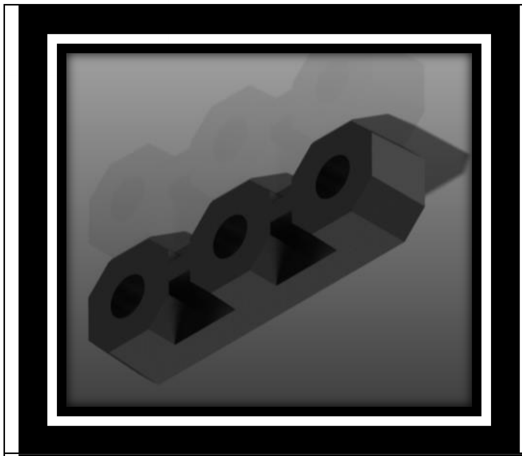
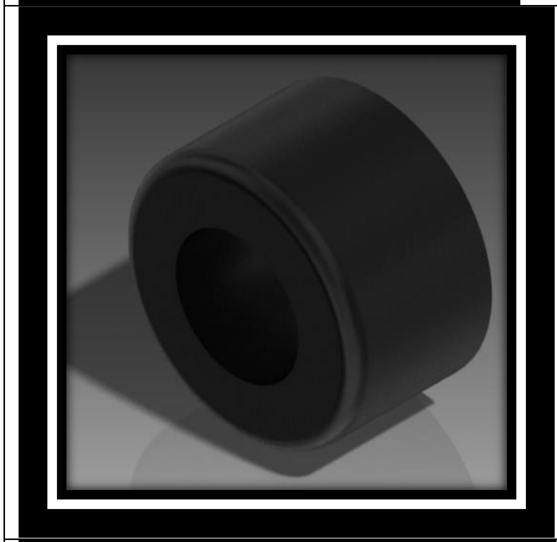
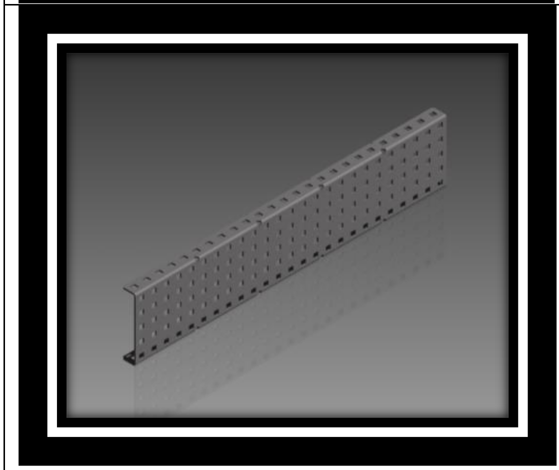
BRAMPTON ROBOTICS

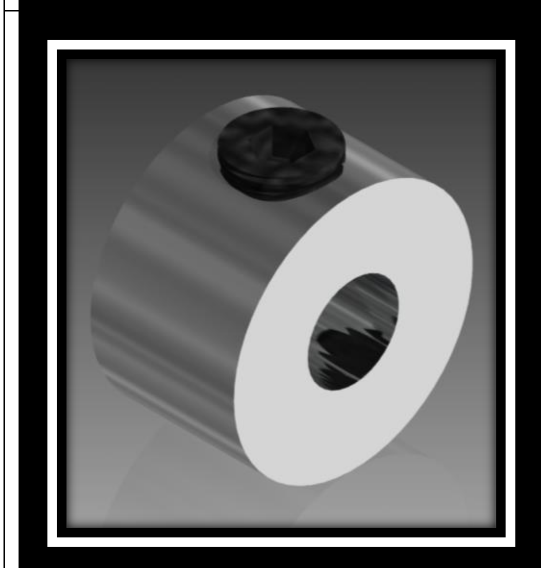
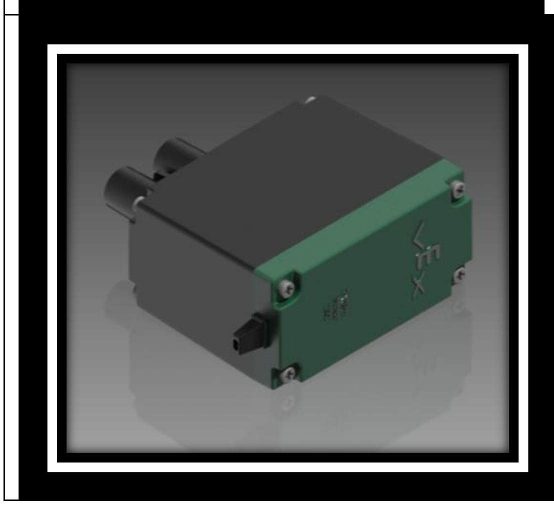
EDUCATION

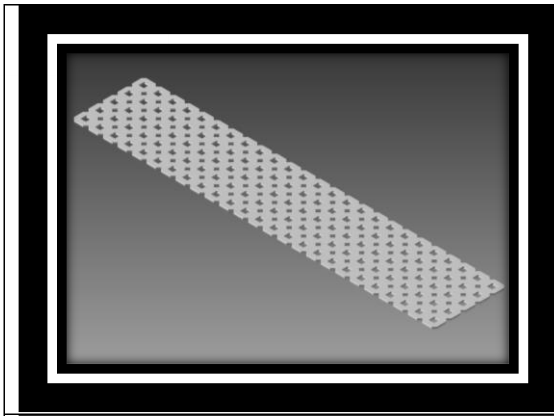
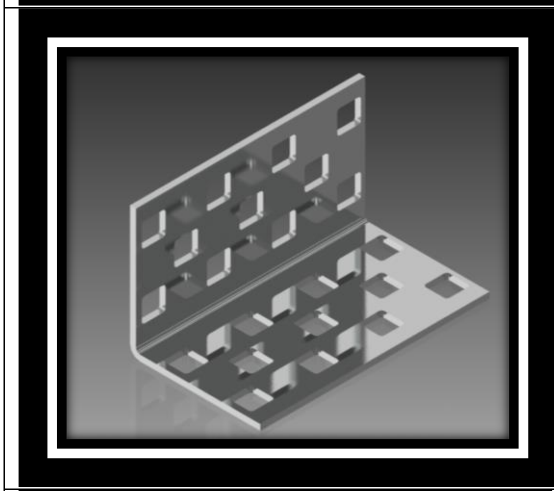
1104G

2015-2016 "Nothing But Net" Design Book

Component	Description And Calculations (If Applicable)
	**THE OMNI WHEEL** The Omni wheel was used in the drivetrain of our robot. If mechanized it allows the robot to move sideways. It also provides good traction for the robot almost better than other wheels. It helps move around the field because it get slide sideways. They consist of a wheel frame and separately made Omni wheel rollers
	**8-32 THREAD/ 6-32 THREAD SCREWS** These screws are used to attach individual pieces together and are fastened by Vex standard nylocks, Keps nuts, or regular nuts. They come in a variety of sizes ranging from .5 inches to 1 inches and above. The 6-32 Thread screws are thinner and are used to screw in motors.
	**SHAFTS** Shafts are used as pivot points for different parts of our robot especially on the drivetrain. It helps spin many parts of the robot. They are spun by Motors. Shafts range from .5 inches to 10 inches and above. Shafts can be cut using metal cutters to fit the size of a particular application

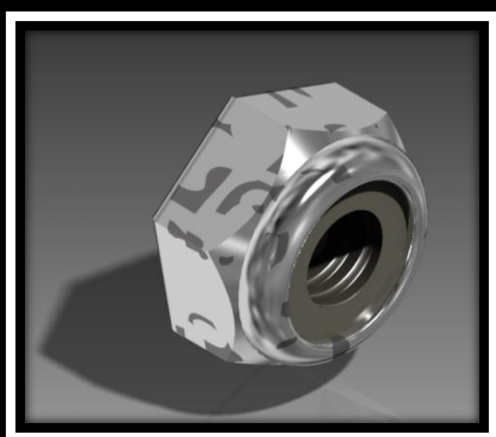
	**BEARING FLATS** These are used to guide and fasten different parts of the robot. For the drivetrain the bearings allows the shafts to turn smoothly and properly.
	**SPACERS (THICK AND THIN)** These were used to create space on different parts the robot. This was done to probably adapt to other parts that required to be placed in a certain position without moving the process. Depending on the amount of space needed spaces come in both thick and thin sizes.
	**C-CHANNEL** C-Channel is used for the main structure frame of the robot. It comes in many different and adaptable sizes such 151-25 (A seen on left) or 121-25. The 1 of 121 is the number of holes in the top extrude while the other 1 is for the bottom extrude. The middle number 2 represents what is the width of the amount of holes on the front surface on the C-Channel. The final number 25 tells what the length in number of holes is.

	**KEPS NUTS** This is one of the VEX fasteners we had used. It gets a good grip on the structure because of the grips on the circumference of it. This is good for fastening metal together. It uses an 8-32 Thread. These however begin to loosen after a while meaning they aren't resistant to vibrations.
	**SHAFT COLLARS** These Vex fasteners are used to fasten shafts to the structure and all them to pivot easily. They contain a setscrew in a hole on the side that is used to hold it in place one the setscrew is tightened
	**VEX 393 MOTOR** The motors are responsible for most of the motion the robot creates. It moves the shafts that is connected too. The motors were used are set at High Torque Settings. This means the motor rotates at 100rpm. The motors can be configured to meet different rotational speeds by replacing the gears within the motor.

	**VEX METAL PLATES** These are another structural component on our robot. Most popular ones are 5x25 holes. Some are made in other different sizes. They can however be cut and modified to fit the robots needs.
	**ANGLE PLATES** These structural components come in longer sizes. The one shown on the side was modified to attach both the tower and the drive train together. This mean they can be modified to fit the needs of the robot like the other structural components.
	**HIGH STRENGTH 60 TOOTH GEARS** These were used in the tower. The larger extrude allows it to be able to pull a larger amount weight. There are 2 different types of Inserts that are put into the gears. These allow them to work differently. One of the inserts we used was called a rubber shaft insert.

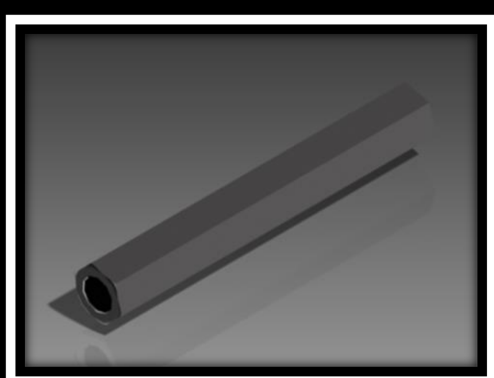
HIGH STRENGTH 12 TEETH GEARS

These were the pinion gears for the tower. They were added to add more torque to the rotation of the larger 60 teeth high strength gear. This hadn't required an insert because they were made to fit the shaft.



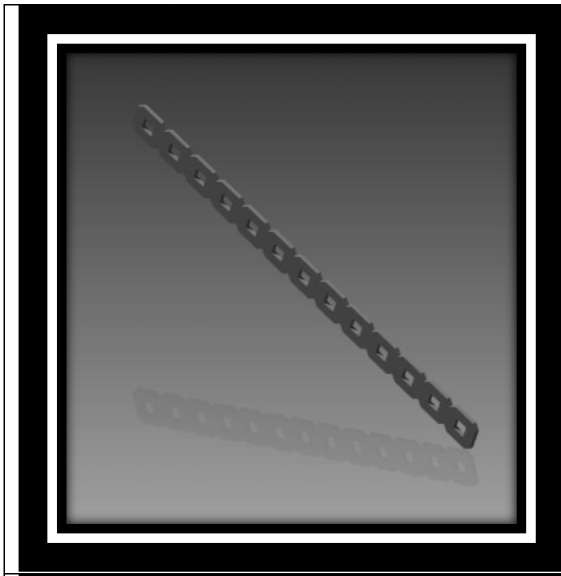
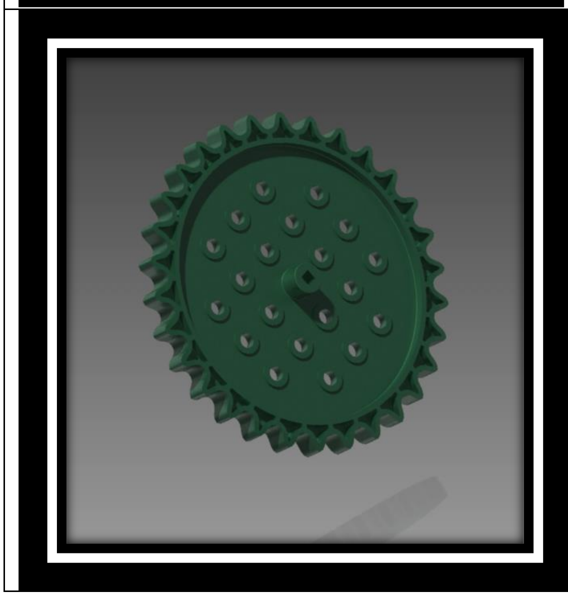
NYLOCK FASTNERS

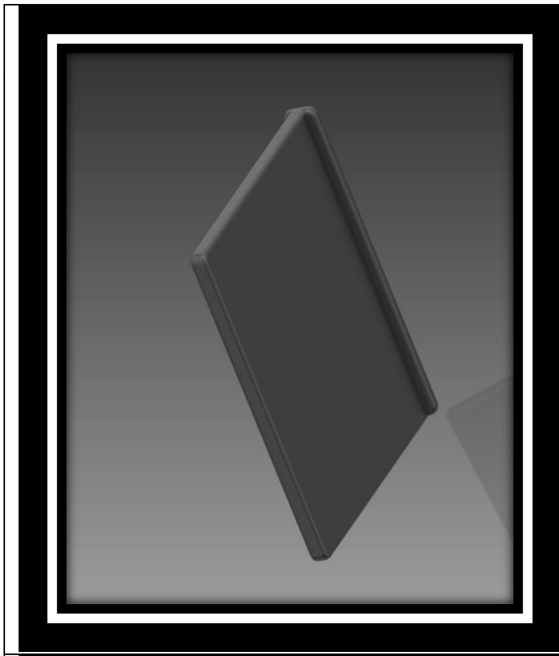
These are the other types of fasteners we used. They work just like a Keps nut but they contain a nylon lining in the inside of it. This makes it vibration resistant and makes it ideal for any lift or part that seems to move a lot.



STEEL BEAMS (STAND OFFS)

These are different form of spacers but they also act like a supportive binding for different parts of the structure. They come in different sizes like 3 inches and can be cut and modified to fit the right space. Beam couplers can also be used to connect 2 different beams.

	**METAL BARS** These are straight bars of metal. They are always 1 hole in width. We used these for added support on the robot, primarily our Lexan hood.
	**SPROCKETS** Sprockets come in different sizes ranges from different teeth counts. They are usually moved together using a chain. They are helpful for making intakes because they are compatible with the tank treads which intake flaps can be inserted into.



CONVEYOR FLAPS

The conveyor flaps were used for the intakes where they intake the balls in. They are slipped into the tank conveyor treads. They come in 3 different shapes (the one on the left is the largest) large, medium, and then small. Large seemed like a more strategic flap because it could pick up the balls more easily.

BASE AND CHAIN LINKS

This picture shows a previous intake we had done. It shows the base and chain links. The conveyor flaps slide into the base links to create the intake.