GAEL FORCE ROBOTICS 5327D

Ashir Bhalla-Levine, Christy Koh, Gustavo Silvera, Justin Lee, Kavin Kasi, Kunal Shah, Sarang Vadali

Table of Contents

Team Overview

Led by a captain with 3 years of experience, Gael Force Robotics Team 5327D is a diverse and motivated group of high school students eager to tackle the challenges of competing in VEX Robotics Competitions. Our team is composed of one junior, four sophomores, and two freshmen. Each of our members has a passion for learning engineering through robotics, both on the software side and on the mechanical side. We work together by dividing up tasks: conducting extensive research on problems we encounter, testing and evaluating solutions, and communicating the results. Each member brings their strengths in certain skills, using them to support other members who may perhaps be less strong in those areas. In this way, our team forms a cohesive, cooperative group, dedicated to helping its individual members learn and grow.

As we gain experience through scrimmages and competitions, we assess our performance and discuss how to improve for the next time. We constantly refine our design process, seeking out the best and most creative ways to achieve the task given. Making new friends and forging connections with other teams from different schools is also an important part of competitions.

Ultimately, through the experience of competing as a team, we aspire to gain valuable knowledge and experience applicable to future careers in engineering.

Team Member Biographies

Team Captain - Sarang Vadali

Hello future readers. My name is Sarang Vadali and I am currently a junior in Dublin High School and the captain of the 5327D Robotics Team Gael Force. It is very exhilarating to be a part of this team and I hope to achieve many awards and titles in our competitions ahead of us. As for my robotics experience, I have done FLL for two years in middle school and did VEX Robotics for two years as well. My FLL experience was not well as we did not achieve much. On the other hand, I feel happy about my first year on VEX because we were tournament champions last year for two competitions, one qualifying tournament competition and in the States Competition and qualified to Worlds. Additionally, we won three design awards and two robot skills awards. At worlds I was able to make new robotics friends, learned new techniques from other teams, and most importantly have fun. My sophomore year we won a tournament, a robot skills awards, and a think award. We were unfortunately not able to qualify from the States competition as we lost in a very close game in the Semi-Finals rounds. However, we were ranked 12th in the world for programming skills and was able to attend Worlds. Additionally, I am part of the Engineering Academy in our school. Last year, I have taken Principles of Engineering, Computer Science/Software Engineering, and am currently taking AP Computer Science A. In the Principles of Engineering class, we did use some VEX parts to build mechanisms which made a bit of practice for me with VEX while I wasn't at robotics. Some of programming languages I know are Python, HTML, CSS, JavaScript, Java, C, RobotC, C++, Scratch, Netlogo, XCode and I am currently learning Ruby on Rails. With my programming knowledge, and group of people and I created an iOS application for an Engineering Entrepreneur Competition. This used the Parse and Social Framework. What this application did was that it notified a designated person when someone is getting bullied in a certain place. With our design, we won third place in the competition. Apart from school and robotics, I am a huge basketball and football sports fanatic. My favorite basketball team right now is the Golden State Warriors and favorite football team is the Pittsburg Steelers. Watching these games are stress busters for me. I love robotics because when I grow up, I want to be a Software Engineer. With robotics, I can get exposed to both and give me some type of pre-knowledge before I go to college.

Ashir Bhalla-Levine

My name is Ashir and I'm a sophomore on the '16/'17 5327D VEX GaelForce Robotics team. I am interested in (generally) STEM topics as well as music and tend to spend my free time on either of the two. I was first introduced to larger projects and competitions such as VEX in my 8th grade year when my brother was on a GaelForce VEX team. I have some engineering experience in designing, building, programming, and testing for VEX Robotics competitions and am taking Principles of Engineering this year. I have also participated in a few projects at DHS such as the Pancake Robot, Hand Controlled Radio, and Special Olympics. Along with robotics and engineering at DHS, I am also a part of the Dublin High School Irish Guard Marching and Symphonic Band as a flutist and enjoy playing music and participating in different band events. This year, I hope to be able to get more experience with Vex Robotics, different projects, and different fields of engineering as a part of the Academy and as a part of this GaelForce Robotics team. I also hope that, this year as a team, we are able to work well together and work to accomplish the VEX competition challenge in an efficient and effective fashion. Apart from that, in spare time, I enjoy photography and video production. After graduating from Dublin High School I plan on attending a 4-year college where I will study a STEM topic as my major and a minor or extracurriculars in music and photography/film. I plan on working for some sort of technology company after college and pursuing photography or music as a side hobby that may grow to be my main field of work at some point. Overall, I my participation this year on the 5327D VEX team will help me improve my abilities in VEX and as a team member in general.

Gustavo Silvera

My name is Gustavo and I'm currently a sophomore attending Dublin High school in 2016-2017 and am in the VEX GaelForce Robotics team. I am interested in robotics and computer science, including cryptography, programming, compression, video editing, artificial intelligence, and computer networking. I am currently taking Principles of Engineering and completed computer science and software engineering. I have participated in the pankake robot last year and have had some experience with vex and robotc through camps and classes. Along with my normal studies, I played for the varsity tennis team in 2015-2016 and ranked 9 out of 30. In my spare time I sometimes work on self interested projects (usually computer programming problems) and sometimes I play computer video games and learn more about technology and computer hardware. I consider myself a PC enthusiast, as i have a custom build with some of the "top of the line" components which allow incredible amounts of floating point operations as well as rendering/ video game performance. After graduating from Dublin High, I plan on completing a bachelors and masters in a field of computer science, however i am not sure which one I like the most, and hope Vex robotics will be able to assist me in choosing as well as improving skills in teamwork and participation.

Justin Lee

My name is Justin and I am a sophomore at Dublin High School and am in the VEX GaelForce 5327D robotics team. I am currently in the Dublin High Engineering Academy taking the computer science pathway and I plan on using robotics to help me progress as an engineering student at Dublin High. I was a part of this same team last year, and I had the privilege to experience VEX Worlds 2016. As an engineering course, I am taking the Principles of Engineering class provided at Dublin High and took the Introduction to Engineering and Design class last year. As a student, my favorite subjects are math and science other than engineering. Aside from academics, I was also a part of the Dublin High junior varsity swim team last year, and I plan on joining the team again this year. My hobbies outside of school are reading, playing soccer and video games, and programming. I have a high interest in computer science, and how new technologies help shape the world we live in today. After high school, I plan to go to college and take computer science courses, ultimately achieving my dream job of becoming a software engineer. Through robotics, I hope that the knowledge and experience I acquire will help me reach my goal.

Kunal Shah

My name is Kunal, and I am a freshman at Dublin High school in the VEX Gael Force 5237D robotics team. I am currently taking Intro to Engineering and Design as an elective at Dublin High School. I have done First Lego League for two years, the first year we got a distinguished award for good sportsmanship, and the second year I got the championship award for the qualifying round. I made an extremely complicated program which kept the robot going straight by speeding one side of the tank drive depending if the gyroscope is off or not, and it would make sure that we would not mess up because of mechanical failure even though you are not supposed to be doing open loop. We also made a super complicated line following program which was way more efficient, and it predicted where the line was supposed to be so it could go faster and it worked much better then normal line following programs. On one of the runs, my robot clipped the edge of the table and went in all the wrong directions, but luckily it hit a bunch of things and it earned us an extra 45 points. Other than going to robotics competitions, I help my parent’s robotics business by teaching the instructors and making tutorial videos for everyone to see. My favorite class of the day so far is Intro to Engineer and Design. I go biking with my friends at least 3 times a week, and I am learning how to skateboard because I think It can be fun. I love interacting and improving technology on a day to day basis. I played a lot of Minecraft between 4'th and 8'th grade, and I knew all the Minecraft codes, physics, and strategies. I helped code a server for Survival Games and a Factions server. I even made a Minecraft Rainbow mod using Eclipse and Java. Recently I have been playing Call of Duty Black Ops 3, because it is the only game where you can do 4 player couch multiplayer. While trying to get the best gaming experience I have to learn how to get the most out of what I have. I previously did tricks on my 50$ scooter including a very wimpy grind, and I once was able to do a tailwhip. I stopped because some of my friends who lived in that area moved around the same time I did. I then biked more because I had to bike to school and back home, and I ended up liking biking so me and my friends went out to bike around the community biking. Once I managed to no hands on my bike from Positano Street all the way to Fallon. I learned a lot about gear ratios from my bike because of the different size gears on my bike. I wanted to learn more about how gear ratios worked, so I learned online that depending on how many teeth each gears have you get a gear ratio. I want to continue to pursue my goal of becoming someone who does what they love. I am on this robotics team to gain experience, to learn, and to pursue my passion.

Christy Koh

Hi! My name is Christy and I am a sophomore attending Dublin High School. This is my second year in Gael Force VEX Robotics. Besides being a part of robotics, I am the World/National Editor for my school newspaper *The Dublin Shield* and a member of the Chemistry Club. I am a part of the Dublin High Engineering and Design Academy, on the Computer Science/Software Engineering Pathway. Most of my serious endeavors in engineering only began in freshman year, when I first joined robotics. The competitive nature of the tournaments and the intellectual stimulation of designing an effective robot to complete a task appealed to me, instilling in me a love for engineering. I enjoy the design and construction aspects of robotics, since it gives me a perspective on engineering outside of the computer science pathway I am currently in. In addition, I love being a part of the team bond and working together as a group. One notable outside project I have been working on with a partner is an application that aims to help students at DHS graduate with their target diploma (our school offers different tiers of diplomas). Last year, we won 2nd place in the school's Entrepreneurship Competition despite only developing the app in 2 weeks. Over the summer, we improved the app further, implementing a server side hosted by Amazon Web Services and a more professionally designed client side interface. After high school, I hope to attend college and major either in biotechnology, chemical engineering, or nanotechnology. As a member of this robotics team, I hope to learn essential skills to develop myself as a female engineer in a highly competitive environment.

Kavin Kasi

Hi, my name is Kavin and I'm a freshmen at Dublin High School. This is my first year in Gael Force Vex Robotics. Aside from being on the Dublin High 5237D team, I am a member on the Dublin High Speech and Debate team. I am also a part of the Dublin High Engineering and Design Academy, on the Computer Science/Software Engineering Pathway. My interest for robotics started when I joined my middle school robotics team. Our team wasn't very successful, but I learned a lot about teamwork, and communication from the experience. The reason I love engineering is because there is no real limit to what you can make. I like competing in robotics tournaments because the competitive atmosphere makes me want to work harder. Outside of engineering, I love programming in Python. I made an game that is script based, and is an adventure. In robotics itself, I enjoy building and designing the robot itself, as it lets me use my creativity. Some of my hobbies are playing video games, and hanging out with my friends. My favorite video game is probably Halo for the Xbox. I also love eating food (Chipotle being my favorite). In the future I want to work as a civil engineer, or for a web-based startup. In this team, I hope to apply the skills I have learned my previous year in robotics while learning new skills. I also want to learn how the Vex system works, and be able to use it in the years to come.

Meeting 6/9/16 - First Meeting

Today was that first meeting for the 2016 - 2017 season for 5327D. Our primary goal was to get to know each other and set out a plan for meetings. We dedicated this day just to understand everyone on the team as we will be working with this team for about an entire year. To learn a little bit about each other, we began with an icebreaker: two truths and one lie. We learned a lot of interesting facts about each other. Kunal said his favorite company was apple, he built his own computer, and he has been to Quebec. Turns out that Kunal doesn't like the company apple that much. Christy said that she has a Lenovo thinkpad laptop, her last name is only three letters long, and she can do a triple backflip. That is when we found out that she could not do a triple backflip. After we gained some insight on everyone's personalities and interests, we all talked about our background experiences in robotics. The two experienced members, Justin and Christy, began to outline a timeline for the summer. In the meantime, Sarang taught Kunal and Gustavo the parts we were provided with. Lastly, we finalized the meeting dates based off availability. For the most part, the majority of the team was only available during the weekends as Kunal and Gustavo had DPIE summer school from 8 am to 1 pm while Sarang had Elite SAT Test Prep from 2pm to 7pm. Justin and Christy were the only ones free all time except for a few vacations here and there. Our tentative summer schedule is shown below:

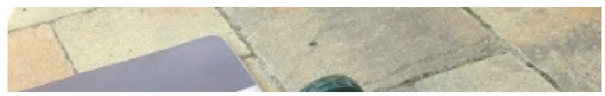
- 6/15/16
- 6/17/16
- 6/20/16
- 6/24/16
- 6/28/16
- 7/2/16
- 7/3/16
- 7/5/16
- 7/8/16
- 7/9/16
- 7/10/16
- 7/12/16
- 7/13/16
- 7/16/16

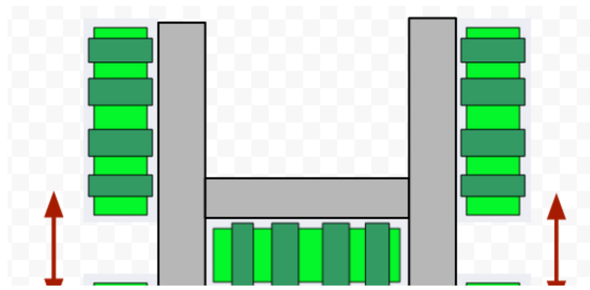
Meeting 6/15/16 - Game Manual And Strategies

Today everybody came to the meeting. The goal was to become familiar with this year's game and come up with strategies for the competition. We looked at the VEX competition rule book and watched the video for Starstruck. We also conducted research on the low and high hangs, which are rewarding but more challenging to accomplish with our motor constraints. Then we started discussing strategy and our Robot building game plan. After close consideration of many individual strategies such as high hanging, cube and intake launching, and pushing stars off the fence, we concluded that it would be best logically to have a mix of the three. However we don't have the extra motor capacity nor ability to complete a hang, thus we wanted to focus more on capturing additional intake and launching them as well as pushing stars off the fence in front of our robot. Although this is our current strategy for autonomous, we considered the other possibilities such as hoarding stars until the end and launching as many as possible at once, then also to launch the intakes from where we were at the far zone vs the near zone where we can easily knock stars down from the fence. The advantages and disadvantages of all the other options were that for instance with the star hoarding option, it is risky because if the opposing team ends with a block, they could possibly block most or some of the stars that we would have used as our final attempt to gain points, the advantage to this method is that if the opposing team was not blocking, and the launches all went as planned, then we could score a large number of points in a short period of time at the end, where it matters most. Onto the method of staying far back and launching stars from there, we considered this because we would be able to collect the stars that the other teams would have thrown if they were in the far zone, however we would be vulnerable to stars and cubes that landed in the near zone and would not be able to launch as far, as well as launching distance, it takes more force to launch from further and in the probable case of having the stars miss and not even go off the fence, it would be more difficult to collect the stars again, and much more avoidable if in the near zone already. Finally, seeing the pros and cons of staying in the near zone, the we would have a better chance to get the stars launched further because of our initial location, and we would be able to block dump bots and near lands, however, we would lose valuable time because we have to drive up to the front rather than beginning at the launching point, also we would be vulnerable to stars that go above us and reach the far zone, which are technically worth more. We found that the best position to be in is the combination of the far and close zone because a robot has the capability to launch far, block when near, and defend against farther stars.

Meeting 6/17/16 and 6/20/16 - Prototyping and Testing for Bases

Everyone on the team came to these two meeting. The primary goals for these meetings were to prototype bases for driving our robot. We figured out the advantages and disadvantages between using a tank drive, X-drive, and H-drive.

Drives	Picture
Tank-Drive: Advantages: - On the software side, easier to program and implement the true speed function - B Can turn - More feasible and less time-consuming to build and wire Disadvantages: - Unable to strafe sideways - No diagonal movement	
X-Drive: Advantages: - Able to strafe sideways, and go forwards and backwards - $\sqrt{2}$ times faster - Diagonal movement - Tighter turns - 360° strafing movement based on programming strategy Disadvantages: - Weight needs to be distributed carefully for maximum performance - More time-consuming to build and wire - Need to get correct and exact	

spacing and angles in order to function properly - Wheels can get in the way of the intake, overall less space on top of the base - Driver needs to get used to different controls	
H-Drive: Advantages: - Able to strafe sideways, and go forwards and backwards - More feasible to build than X-drive - More space for intake than X-drive - Easy to program Disadvantages: - Needs more motors designated to the drive than an X-drive would - Middle wheel can stall more quickly than the other wheels	

We also brainstormed the benefits and disadvantages of different wheel sizes and types for our robot. We tested the different wheels to help determine the best wheel and its size:

Wheel Size/Type	**Advantages**	**Disadvantages**
3.25'' Omni Directional Wheel	- Fast acceleration - Easy to replace (taking off and putting on) when making fixes to the base - Applicable for X-drive, can strafe sideways	- Results with a weaker robot, meaning easier to be pushed around, but this would not be significant in this year's game
4'' Omni Directional Wheel	- Also applicable for X-drive - Higher overall top speed than the	- Harder to replace/make changes to the base

	smaller 3.25'' omni wheels	
4'' Mecanum Wheels	- Saves a little bit more space than omni wheels - Able to strafe sideways	- Less traction than regular omni wheels
4'' Traction Wheels	- Grips the foam tiles on the field better than other wheels - Good pushing ability - Unable to be pushed sideways (but not really important to this year's game)	- Cannot produce lateral movement
5" Wheel	- Higher top speed than other wheels because of its bigger size - makes a good flywheel	- Cannot produce lateral movement

Meeting 6/24/16 and 6/28/16 - Prototyping and Testing for Launchers

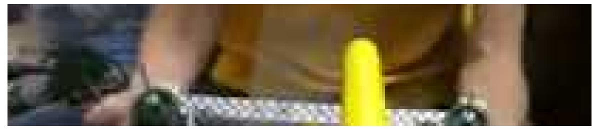
Everyone on the team came to these two meetings. Today, our primary goal was to prototype launching mechanisms for driving launching stars over the fence. We tested two mechanisms: the catapult and dump.

Launcher Mechanisms	**Picture**
Catapult: Advantages: - Can reach far zone, if used correctly - Can launch over blocking robots - Can use elastics for increased range and stability - Can have greater versatility for range and location of launch (allows for opponent to more more) Disadvantages: - Requires more force - Risky, not guaranteed over fence if made poorly - Difficult to use correctly - Distance of launch varies on weight of projectiles, more = heavier - Can be blocked if objects are launched at too low a trajectory	
Dump: Advantages: - Can block other dump bots - High chance of getting stars over fence - Easy to use correctly - Typically has larger carrying capacity thus can hold more - Requires less force Disadvantages: - Difficult to reach the far zone - Can be blocked easily - Slow and large	

Meeting 7/2/16 - Prototyping and Testing for Intakes

Everyone on the team came to these two meetings. Today, our primary goal was to prototype mechanisms that could intake the stars and hold them as well. We tested three mechanisms : forklift, claw, and side rollers.

Intake Mechanisms	**Picture**
Forklift: Advantages - Able to intake multiple game pieces at once - Easy to build - Very suitable for stars and cubes - No additional button needed to control and program Disadvantages - While lifting, game pieces can fall out of intake if not designed properly (no grip) - Big intake that can increase weight on the robot	
Claw: Advantages - Good grip per star - Difficult to drop or lose a star - Can be powerful with pneumatics - Easy to program - Small form factor Disadvantages - Difficult to control if new - Difficult to collect large quantities - Can be weak with motors	

Side Rollers: Advantages - Not very hard to build, if a working design is ready to use - Relatively easy concept - Easy to program - No need for use of elastics Disadvantages - Extremely slow intake - If not used as part of the launch mechanism, requires a transition to the launch mechanism - Needs extra button to control leads to a more complex driving experience - Takes up at least one motor to be functional - Can be heavy on the robot	

Meeting 7/3/16 - Further Prototyping and Testing for Intakes and Decision Matrix

Base

Type	Speed	Turning	Control	Least Complexity	Total
Tank	2	2	3	3	10
X-Drive	3	3	2	1	9
H-Drive	1	1	1	2	5

Key: Best = 3, Worst = 1

Wheel

We created this design matrix with a tank drive (chosen above) in mind.

Type	Speed	Lateral Movement?	Traction	Versatility (for diff. drives)	Total
3.25'' Omni Directional Wheel	1	2	3	4	10
4'' Omni Directional Wheel	2	2	4	5	13
4'' Mecanum Wheels	2	2	2	3	9
4'' Traction Wheels	2	1	5	2	10
5" Wheel	3	1	1	1	6

Key: Best = 5, Worst = 1 Yes = 2, No = 1

Intake

Type	Suitability for Stars	Suitability for Cubes	Speed	Least Complexity	Total
Forklift	2	3	3	3	11
Claw	3	1	2	1	7
Rollers	1	2	1	2	6

Key: Best = 3, Worst = 1

Launch

Type	Speed	Strength	Least Complexity	Can it be adapted to hang the robot?	Total
Dump	3	3	3	2	11
Catapult	2	2	2	2	9

Key: Best = 3, Worst = 1 Yes = 2, No = 1

Meeting 7/5/16 - Building the Tank Drive

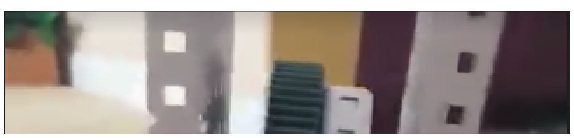
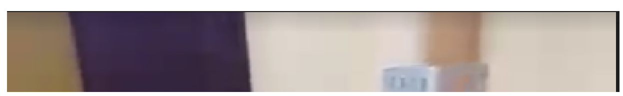
At today's meeting, we constructed our 6-motor tank drive we plan to use for our base. We chained the motors so that the front two wheels had four motors and the back wheels had two motors powering them. We changed all the internal gears to turbo gears. The reason for this was that we wanted to be as fast as we could. In previous years the opponents could push robot causing motors with less torque to stall such as speed or torque. With a boundary present this year, we decided that we could go turbo internal speed with 4" omni-wheels. The only potential drawbacks were weight of the robot and if our alliance was pushing us for some apparent reason. We built this base so that there would be little to no friction for the motors powering the wheels. This was done by adding spacers. This was our base we built today, and to the right is the simple code we used to control it:

The code above demonstrate the calling of each motor we used on our 6 motor base. As seen above, the function calling uses the VexRT[ChX] with X as the axis in which we want the joystick to control the wheel.

After everything was complete, we drove it around to determine any flaws. There were no major flaws except one side of the base moving slightly faster than the other side. This was because of a bent axle that we immediately fixed. We decided to test our base when we had the lift and forklift attached as they would add the weight we would need to carry.

Meeting 7/8/16 and 7/9/16 - Building the Lift

At today's meeting everyone showed up. Our primary goal was to make the gear ratio for our lift and attach these lift towers to the robot. We first built a gear ratio of 5 to 1 for our lift with four motors powering it. This means that for every five times a motor turns an axle (which is meshed in the gear), the final axle turns once. This is very useful for carrying stars as a lot of strength is required by the motors. However, too much torque also compromises speed since they are inversely proportional. This is also harmful to our robot as we want to be a fast and efficient robot. Thus, we had to try to find the perfect speed and torque needed to be the robot we wanted. At first we tested with internal motors being torque. This would mean an overall ratio of 12 to 1. We used four motors for our lift so that we could allocate two motors for our hang. To make the gear train we meshed three pinions with a 60 tooth gear. Two of the pinions had motors on them and a middle pinion was used as an idler gear as well as space so that the motors didn't hit each other. We used spacers so that the spacing would be proper. We also used a crossbar to connect both sides of the c-channels and used shoulder screws for further strength and alignment. All of this was done to reduce friction so that our gear train for our lift can function to its greatest potential. We additionally added these to the base we had built in the previous meeting. We again used should screws so that they would stay firm on the chassis and again reduce friction. This was how the gear train was configured:

Because there were two sides of the lift, they wouldn't necessarily go at the exact position at the same time. To combat this we added a crossbar from one side of the c-channel on the left tower to the other c-channel on the right tower. This way they would move in synch. To test with stars we temporarily added a platform. The finished product for today looked like this:

After we tested the lift we saw that it was able to efficiently carry many stars (six stars) but scored slowly. A dumper bot like this one needs a lot of momentum to score to the far zone. With the surplus of torque it was lacking this capability as stars did not go too far from the robot when we tested it. We decided that the overall ratio needed to be fixed.

Meeting 7/10/16 - Fixing Lift

As decided in the previous meeting, we decided to switch the overall ratio of the lift and decrease it so we could be faster yet not compromise the efficiency of the star capacity. We initially switched the 60 tooth with a 36 tooth. This made it a 3 to 1 torque ratio and a 7.2 to 1 overall. However, we were only able to score a maximum of just 2 stars efficiently (3 if we got lucky). Additionally, we tipped after every throw. On the bright side, the speed was higher and the stars went further because of the momentum generated. We realized that we went past the balance of speed and torque, favoring speed, and decided to test a configuration of 5 to 1 again yet this time with speed internal for the motors. This would mean a 8 to 1 overall ratio. We found this as the perfect ratio as we tested. This was able to throws four stars consistently, as well as a cube, in the far zone. We decided to stick with this configuration.

```
//lift
void up(int speed)
{
    motor[leftLiftMotor] = speed;
    motor[rightLiftMotor] = speed;
```

```
    //for lift
```

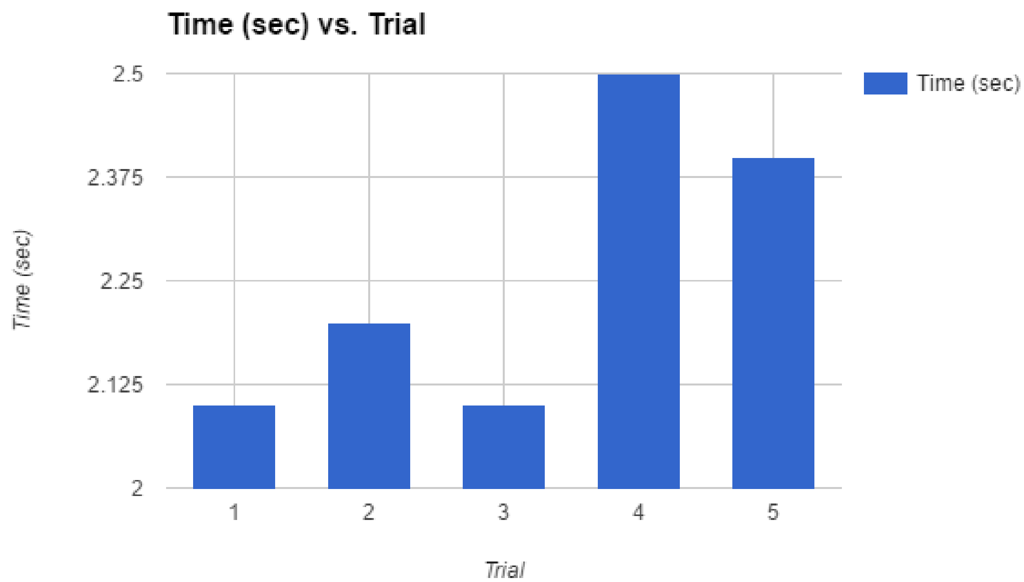
The above codes are our initial functions and callings for moving of our 6 motor lift. Which are modified versions of our single motor lifts, which call the up() and down() functions that have parameters that take in the speed value that control how much power the motors deliver.

Meeting 7/12/16 and 7/13/16 - Base Testing and Fixing

As decided in the previous meetings, we tested the base today. We conducted various experiments such as stall tests. The results are shown below:

This test measured time from wall to fence:

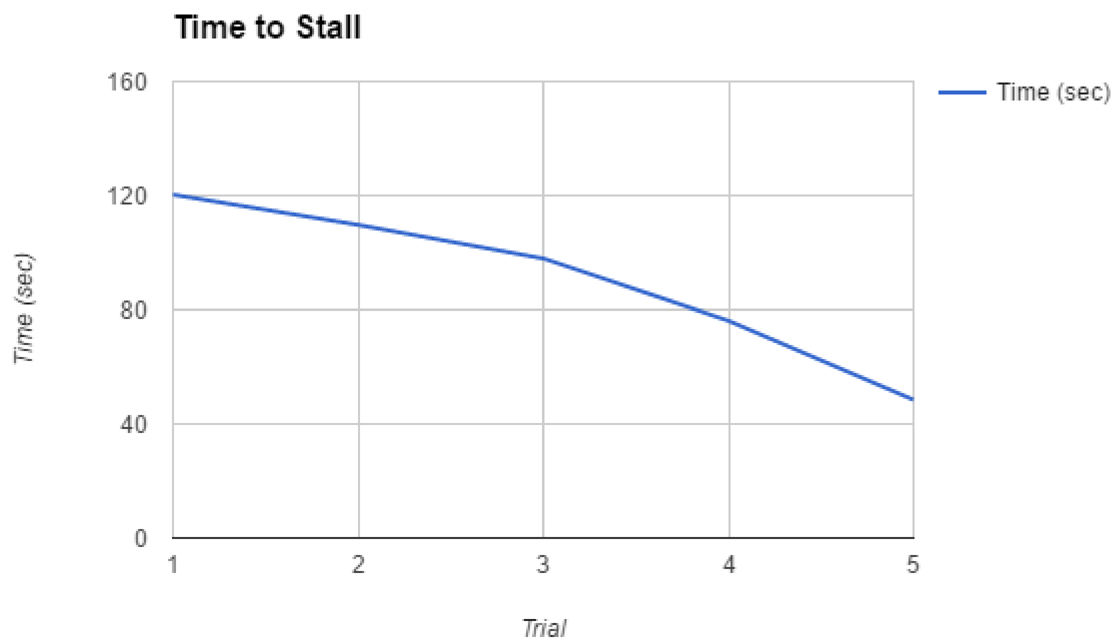
Trial	**Time(sec)**
1	2.1
2	2.2
3	2.1
4	2.5
5	2.4
Average	**2.3**



The time it took to drive from the wall to fence stayed constant the many trials. We could account the tiny variation we had with battery amount.

This test measured time until the base stalled:

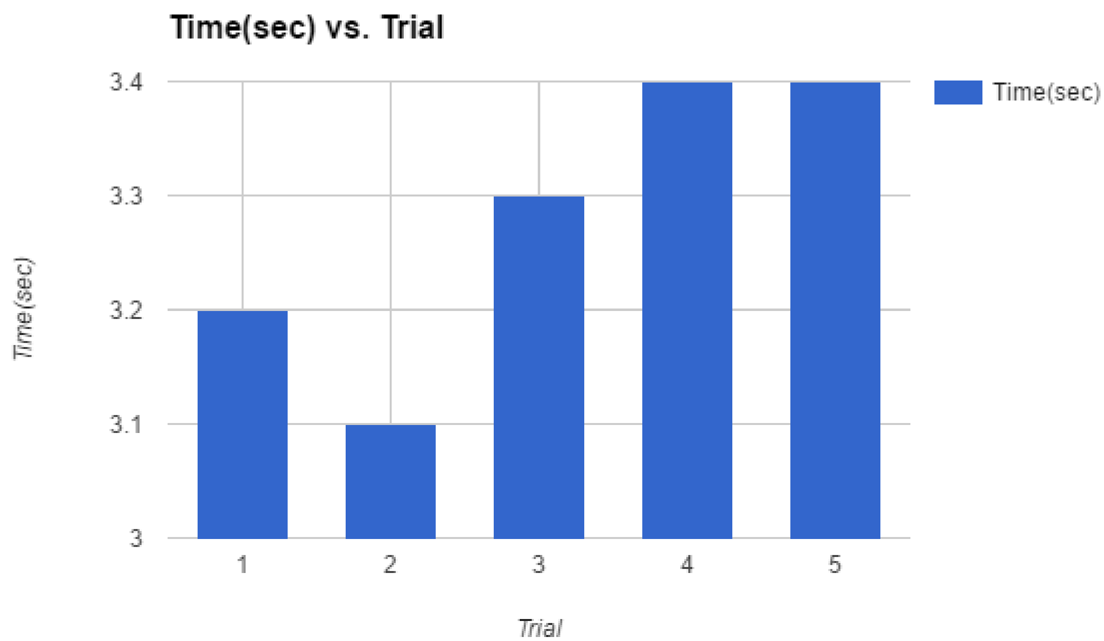
Trial	Time(sec)
1	120.3
2	109.7
3	97.9
4	76.2
5	48.5
Average	**72.52**



We found based on our data table that the time it took for the base to stall decreased exponentially. The equation that described change in number of seconds to stall over each trial was $y = -17.71x + 143.65$. This meant that for each trial, the time it took to stall decreased by 17.71 seconds.
After our initial tests, we proceeded to change our base motors from turbo internal gearing to speed. These are following tests we did to see if there was any change between having turbo and high speed base motors.

This test measured time from wall to fence:

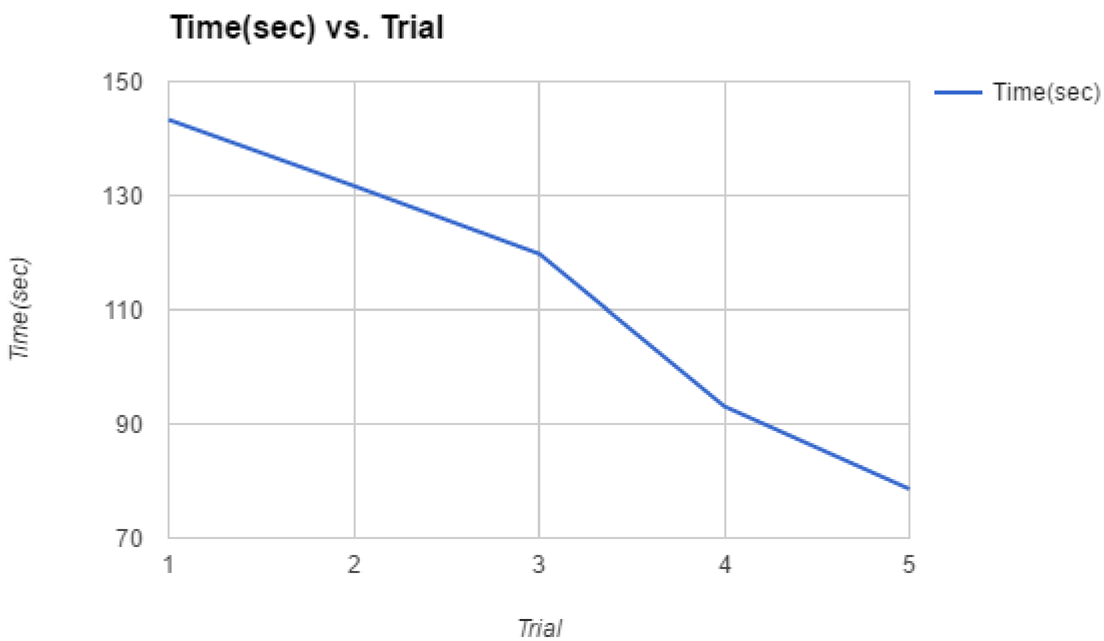
Trial	Time(sec)
1	3.2
2	3.1
3	3.3
4	3.4
5	3.4
Average	**3.28**



Although we had an almost constant speed throughout the five trials, it was obvious that the high speed base was significantly slower than the turbo base.

This test measured time until the base stalled:

Trial	Time(sec)
1	143.3
2	131.7
3	119.8
4	93.1
5	78.6
Average	**113.3**



Based on the data, we found that the time it took to stall exponentially decreased and the high speed motors took longer to stall than the turbo motors. The equation that described change in number of seconds to stall over each trial was $y = -16.8x + 163.7$. This meant that for each trial, the time it took to stall decreased by 16.8 seconds. The smaller slope than the turbo motor equation meant that the high speed motor had a less drastic change in stall time.

Meeting 7/16/16 - Building the Forklift

Our primary goal for today was to build an efficient forklift that could lift both stars and cubes. The reason we chose a forklift was that its passiveness (no motors required) allowed us to add motors to our robot. Additionally, it is properly able to grab stars and cubes by just picking them up. We started with 2 c-channel on the sides of the lifts and thread string through the holes on the sides of the c-channel so it created a platform made of string. We choose the do this method first because we thought that the string would be a good passive intake. Our first forklift we built consisted of string. How this worked was that there would be two parallel c-channels attached to the robot. Some sort of tight string would be connecting the two channels and in a zigzag formation. In a way, this would be similar to acting like a platform for the stars when we lift up. To ensure the string was tight we fitted them through bearings. String is very light which was one benefit of using it on a forklift. However, we saw that the string would become loose often and would blocks the cubes from entering as it was too thick. We decided to come up with a better design instead. We eventually came up with an intake that consisted of just c-channels and were used essentially as prongs. We attached 4 c-channels to our lift that could act as a fixed “shovel” platform. We also added standoffs connecting them to the side c-channels we had for our string intake which allowed a greater firmness. The configuration worked well for stars as they could fit between the spaces of the four c-channels we attached. However, for cubes, the c-channels were too tall and the tray could not slide underneath the cube. This was because the metal was bending upwards. With multiple attempts to face the metal downwards, we came up with the idea of adding bendable bars to the tip of the c-channels so that they could face downwards and cubes would be picked up. This modification called for a platform consisting of half c-channel and half bars that were bent so that cubes could be picked up. We saw that cubes would be picked up not all the time. The position of the cube heavily mattered for this configuration. We decided we would improve on this in the next meeting.

Meeting 7/18/16 - Finish Building Forklift and Brainstorming Lock Mechanism

Today, we finished our fork lift After seeing the variance in reliability for the previous design with the forklift (built with c-channels and bars) we decided to build a platform entirely with bars. We thought bars would be a good idea because they were flat and could theoretically slide under cubes easily. Additionally, they had all the features of a platform with c-channels including less weight as well. When we actually tested them, the bars could not support the weight of the cube and would bent. They were too flimsy on their own. We would have to unbend them after every time we picked up a cube proving its complete inefficiency. Our final and best design consisted of standoffs for our platform. These standoffs were attached how the c-channels and bars were attached. Standoffs were firm, but not as flat as the bars. They worked relatively well to pick up the stars and cubes. Very rarely they did push the cubes away, but for the most part they were successful in picking up both types of scoring objects. To counter this we saw that by pushing the cubes to the wall would assist in picking up the cubes if this had indeed happened. We decided to use standoffs since they were more stable and did not bend as easily. We added two 5-hole c-channels higher up on the lift arms in order to keep cubes on the forklift when launching. Since the tray was obviously out of size, we needed to find a way for it to fold out. We decided to do research and look at how other teams solved this size issue. We decided to continue our research until the next meeting.

Meeting 7/20/16 - Building Lock Mechanism

After extensive research from the last time we met, Christy came up with a lock mechanism that allowed the forklift to be folded inside 18x18x18 and was able to flip out to the ground. The lock mechanism consists of a standoff and a c-channel cut lengthwise attached by a collar going through a screw, allowing the joint to rotate. The rubber band attached from the c-channel to the lift provides tension so that once the tray falls out from the raised position, it will remain locked in a horizontal position. After testing different lengths of standoffs and c-channels, we managed to find the perfect balance so that the tray locks at the perfect height to pick up both stars and cubes. We constructed two of these mechanisms to attach to both sides of the tray. The image below shows one side of our tray with the lock mechanism attached:

Concept of TruSpeed

Initially, we could easily implement the trueSpeed function on our tank drive because of its programmatic simplicity. We could and did use an array function for a while, but then we developed a nonlinear function code that would be more reliable, realistic, and less choppy for our values:

Initial TruSpeed Ideas (for tank drive):

Old Array Function:	Newer Exponential Function for Tank Drive

```
int truSped[128] =
{
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0, 21, 21, 21, 22, 22, 22, 23, 24, 24,
  25, 25, 25, 25, 26, 27, 27, 28, 28, 28,
  28, 29, 30, 30, 30, 31, 31, 32, 32, 32,
  33, 33, 34, 34, 35, 35, 35, 36, 36, 37,
  37, 37, 37, 38, 38, 39, 39, 39, 40, 40,
  41, 41, 42, 42, 43, 44, 44, 45, 45, 46,
  46, 47, 47, 48, 48, 49, 50, 50, 51, 52,
  52, 53, 54, 55, 56, 57, 57, 58, 59, 60,
  61, 62, 63, 64, 65, 66, 67, 67, 68, 70,
  71, 72, 72, 73, 74, 76, 77, 78, 79, 79,
  80, 81, 83, 84, 84, 86, 86, 87, 87, 88,
  88, 89, 89, 90, 90, 127, 127, 127
};
```

The above code essentially took the X and Y values and changed them to reference a certain value in this array, with this we were able to manually select what values we wanted to have the array output. This could give us a fully controlled curve and dead zone, however, could laso have weird effects because the jumping of the values could be drastic which could affect our driver's performance.

```
float TruSpeed(float joystick)
{
    int value = abs(joystick*joystick);
```

https://www.desmos.com/calculator/uoysvxxi0d

As seen with the above code and Desmos example, the function TruSpeed() is a 5327D© exclusive that essentially takes the values of the joystick, and inserts them into an equation that will allow for similar results to our previous array example. However, this will have a more realistic driving experience as the curve is more outlines and not completely defined by the user.

Our TruSpeed function takes the "value" variable and sets it to the square of the joystick value (either x or y, depending on the call). We then divide the value by 127 to have a maximum value at 127 (which is the maximum value for our motors)

Meeting 7/22/16 - Adding Motors to the Lift

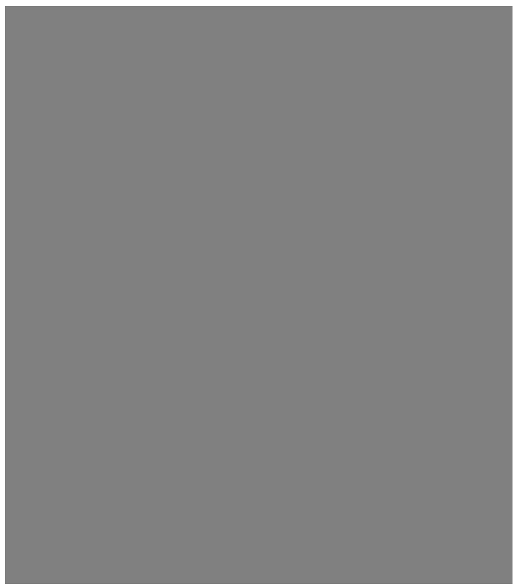
After the previous meetings, our product was a 6 motor tank drive, 4 motor lift, and a passive intake and a lock mechanism. We wanted to use the two motors left over for some sort of hang mechanism. We brainstormed three ideas. The first idea was a linear hang mechanism. It would be built with a rack and pinion system where the motors would power the pinion moving a rack which would be attached to a c-channel with a hook on the tip. This would take around twenty seconds to hang. The drawback with this would be that it may be flimsy and could bend the hang mechanism with the weight is has to pull. The next design we brainstormed was a four bar with two motors powering it. It would be attached behind our lift. The only problem we saw was that two motors could not probably carry both the four bar's weight along with our robot. The final idea was to add a hook to our lift and add two more motors to the lift. We decided to go with this because it was the most simplest yet effective way for us to hang. We would be also able to lift in under five seconds which is very quick. Additionally, by adding motors to the lift, we could pick up more stars because of the extra torque. By unanimous decision, we added two more motors to the lift. Our robot was now complete in terms of motor distribution : six motors each for the base and lift.

Meeting 7/30/16 - Anti-tipping Mechanism

This meeting, we made final adaptations to our robot before the competition. Since our robot tended to tip after dumping over the fence, we decided to build an anti-tipping mechanism to prevent this. These were the various mechanisms we tested on our robot:

1. Standoffs
   We had previously tried using standoffs near the wheels. Although this helped the robot to tip slightly less often, it was not reliable enough to prevent tipping 100% of the time.
2. 5-hole C-Channel
   We gained inspiration for this idea from a previous anti-tip design developed by the 5327B team of 2013-2014. Our adapted mechanism consisted of a 5-hole c-channel rubber banded to the base. It would be folded up before a match; however, once the lift was raised it would flip out. Short standoffs attached to the end of the c-channel dug into the foam tiles, allowing us to dump without tipping. It was mostly effective, however, sometimes our robot would still tip. The main problem was that it dug into the foam tiles, which could damage the field and cause disqualification. This also resulted in considerably slowing our robot down.

3. 2 2-hole C-Channels
   After conducting some online research, we discovered a mechanism involving two short c-channels folding out of the robot. Similar to our previous mechanism, the c-channels would be folded under the base and would flip out when the lift was raised. This solution worked much better than the other two mechanisms tested; it reliably stopped our robot from tipping 100% of the time. The picture to the right shows how the rubber bands were connected to a sprocket on the lift, and would release when the lift was raised, allowing the anti-tipping mechanisms to flip out. We decided to go with this.



The above picture shows the rubber bands from the anti-tip pegs on a small sprocket. As the lift moved up about halfway, the gear would turn the sprocket, causing the rubber bands to slip out and activate the anti-tip mechanism.

This shows the before and after action of the pegs when the gear turns the sprocket, causing the rubber band to slip.

Meeting 8/2/16 - Autonomous and Other Programming Testing

With the competition around the corner, we needed to program our autonomous and skills programs. Since at the competition teams would be placed by performance in skill, the success of our programmed routine would be very important. For driver control, we used a truespeed function:

```
#pragma config(Motor,  port2,           leftBaseMotor2,    tmotorVex393_MC29, openLoop)
```

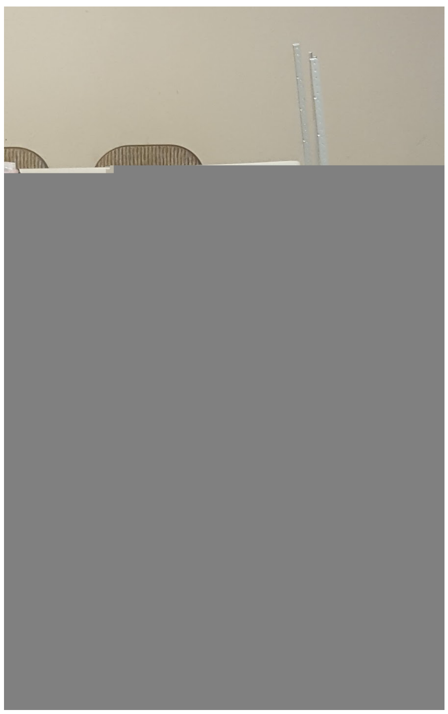
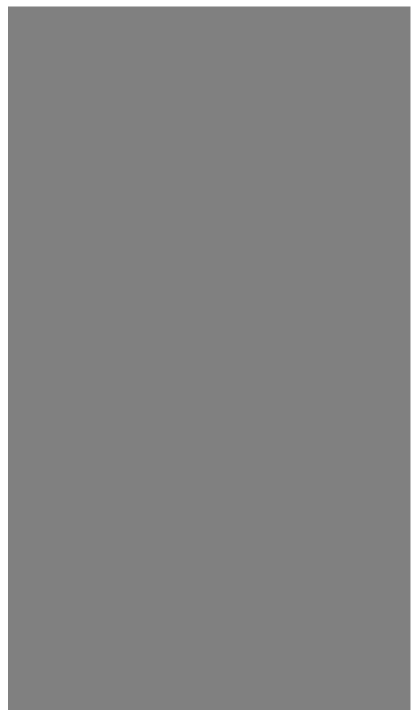
In programming skills, we programmed our robot to dump 3 stars and one cube to achieve a maximum of 10 points. As we were programming, we realized that our robot did not travel exactly in a straight line even though we modified the code. To make sure our robot was aligned, we programmed the autonomous to drive the robot up to the fence before making other carefully timed movements.

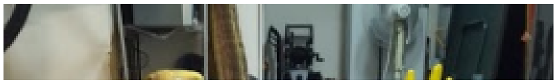
Lastly, for autonomous, we dumped one star and then blocked at the fence. Since our tray is quite large, it would be very effective in blocking other robots.

Meeting 8/3/16 and 8/4/16 - Driver Practice and Final Adjustments

These two meetings were dedicated solely to driver practice and placing the finishing touches on the robot. We made minor adjustments such as fine-tuning the lock mech, making small changes to the code, and attaching the nameplates. Thanks to our hard work the past few meetings, the robot works quite reliably. Sarang will be the driver for this competition, and Kunal and Christy will be coaches.

This is our finished robot before the competition:



Santa Clara Fair Competition 8/6/16 and 8/7/16

Sarang, Christy and Kunal attended the competition.
This event was essentially a skills competition; teams had 6 skills runs each day over a period of 3 days, and the alliances playing in the elimination matches would be assigned from the skills rankings. Our robot performed quite well compared to other teams; we placed 5th in programming skills and 8th in robot skills. Our relatively high ranking allowed us to be selected for 3rd seed. Our alliance was a clawbot, which worked well together with our dump bot. We focused on dumping the cubes, while the clawbot threw the stars. We won every quarter- and semifinals match, even achieving a ‘clean sweep’. We battled our way to the finals match, where we were defeated 2-0-1. Overall, this small competition was a good experience, allowing us to bond as a team and also figure out what we needed to improve to succeed in more serious competitions coming up.

WHAT WORKED	WHAT DIDN’T WORK
Blocking stars, especially against clawbots	Forgot joystick -- need to pack everything needed the night before
Autonomous program was able to block stars	Lift was slow
Scored high in programming skills	Cubes would slide off the side of the lift when turning

Anti-tipping mechanism allowed the robot to stay upright - never tipped	Not enough driver practice
Good alliance during elimination	Lift sometimes stalled
Getting to know other teams	Drive was slow
Lock mechanism allowed tray to fold out	Difficult to turn
Got the high hang mechanism working	Took a long time to align robot to pick up stars/cubes
Scored primarily far zones in matches	Once we took a dead battery to a match -- need to check voltage with a multimeter
Was able to sometimes hold as much as five stars	Hanging mechanism affected effective intake of stars, so we removed it.
	Programming skills program was inconsistent

Meeting 8/13/16 - Brainstorming after Santa Clara

After the Santa Clara competition, we held another brainstorming online meeting to figure out what we need to improve in order to perform well in the first major competition at DV. Based on what we observed from other robots at the competition, we reevaluated the effectiveness of different intakes and drives, creating new matrixes for each category.

Base

We observed that robots with X-drives were able to move more quickly and strafe quickly from side to side to effectively move around without needing to turn, which was one of the weaknesses of our robot. Other robots which used tank drives were relatively fast, but the ability to strafe greatly increased maneuverability and speed when navigating around the field, especially when it was crowded with stars.

Type	Speed	Turning	Strafing?	Driver Control	Least Complexity	Total
Tank	1	1	1	1	2	6
X-Drive	2	2	2	1	1	8

Key: Best = 5, Worst = 1 Yes = 2, No = 1

Intake

At the competition, most of the robots had one of two methods of intaking: forklift or claw. Teams which used claws had lighter lifts and as a result were faster. They could also lift up to 5 stars (max of 10 points). However, claws seemed more difficult to control and were liable to tip often. Forklifts were also quite effective, and were excellent for blocking clawbots.

Type	Suitability for Stars	Suitability for Cubes	Speed	Carrying Capacity	Driver Control	Blocking	Total
Forklift	1	2	1	1	2	1	8
Claw	2	1	2	2	1	2	10

Key: Best = 5, Worst = 1 Yes = 2, No = 1

Meeting 8/23/16 - Building and Programming the Lift Towers

Today we took apart the lift towers motors and changed them from torque to speed because we didn't have enough speed when lifting. The launched game pieces could not make it consistently in the far zone, and our external gearing already provided enough torque. Our resulting gear ratio was 7:1, giving us 7 times more torque on our driven gear; speed was accounted for in our motors. We also modified our previous design of our lift towers by adding idler gears between pinions in order to leave more space between the motors. This would lessen the likelihood of the motors overheating and stalling as a result.

The following is our code to raise and lower the lift:

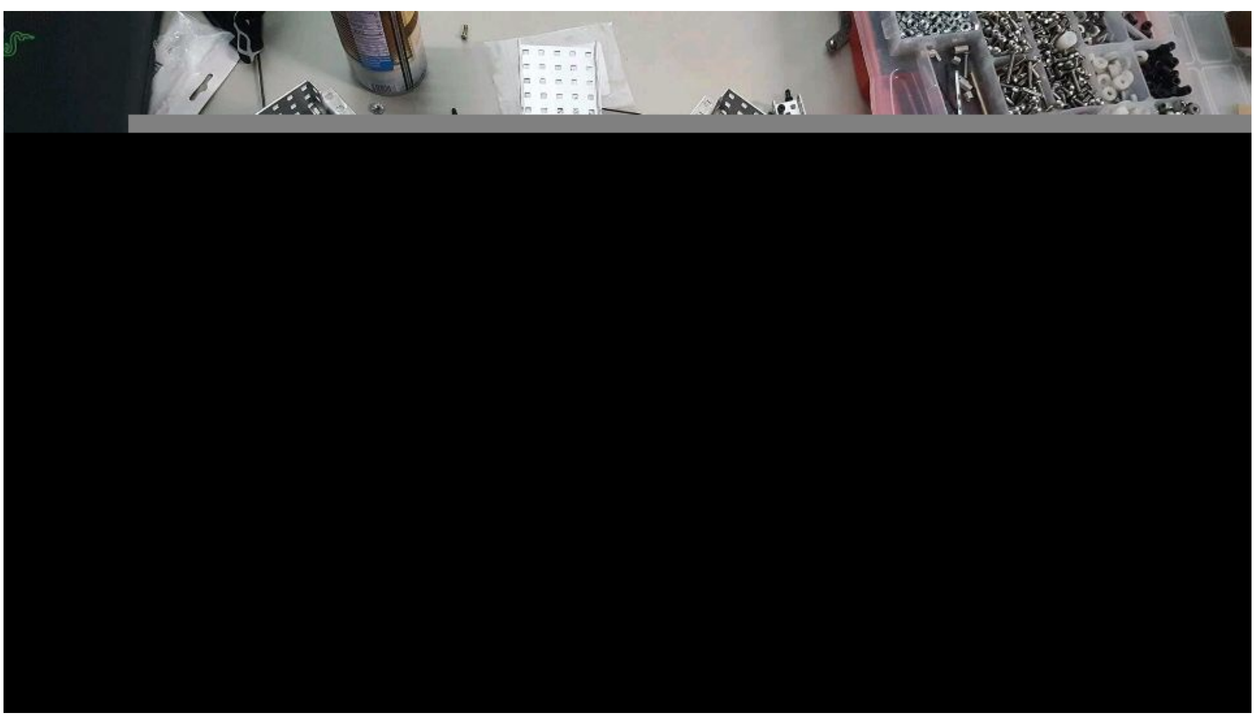
```
//lift
void up(int speed)
{
    motor[leftLiftMotor] = speed;
    motor[rightLiftMotor] = speed;
```

```
//for lift
```

The above codes are our functions and callings for moving of our 6 motor lift. They call the up() and down() functions that have parameters that take in the speed value that control how much power the motors deliver. These are also used for our autonomous function because they are far simpler and more efficient than calling each motor at a specific speed multiple times.

Meeting 8/25/16 - Building the X-Drive

In this meeting, we took apart the robot. After that, we started building our X-drive based on our brainstorms and after carrying out additional research. We wanted to try something new and came up with this configuration of an X-drive:



How this was made was that this only required one “shell” for an X-drive. Additionally, this was able to block off stars from entering the inside of our base, deterring potential drag from stars causing the base motors to stall. The only major drawback with this was that it was very flimsy. This would cause severe stalling with the base. We collectively decided that stalling the base was more weighty than the benefits and decided to build another version of the X-drive.

X-drive Code:

```
//drive
void drive()
{
    motor[frontLeft] = -1*Xcomponent - Ycomponent - Rot;
    motor[frontRight] = -1*Xcomponent + Ycomponent - Rot;
```

This drive() function takes the values given by the right joystick as values to control the strafing of the X-drive, while then adds the value of the x axis of the left joystick to control the rotation of the x-drive. The variable values in the above code are configured in a was that they would not be reversed with the motors of our X-drive, as the motors for our specific Xdrive were flipped, and rotated from each other, resulting in the need to reverse the motors programmatically, or through the physical wires and connections.

Meeting 8/27/16 and 8/28/16 - Fixing the X-Drive

Today we built a new and improved X-drive based on our previous meeting’s findings. Instead of having one “shell,” we built two shells to have more sturdiness in our base. With two shells we also wouldn’t need to completely enclose our wheels, leaving less weight on the base. Although stars would be able to enter the base, we decided that this would not happen often since our intake would leave little space between the front two wheels. After a long process of determining the correct angles and sizes, we finished our X-drive and attached two crossbars so it would be more sturdy and we would be able to attach our lift towers to the base. Overall, our new design proved less likely to stall than the previous X-drive.

In order to make our movements more accurate, we added a encoder to one wheel. We incorporated controls using the encoder in our program:

```
void rotate(bool direction, float rotations, int speed)/
{
  SensorValue(Encoder) = 0;
  //direction TRUE == Clock Wise
  //direction FALSE== Counter Clock Wise
```

As explained on page 35, certain motors have to be reversed to satisfy their physical placements.

Meeting 8/30/16 - Attaching the Lift Towers and True Speed

Today we attached the lift towers to the base. We attached them in the middle of the base so that the robot would be more stable. We moved the crossbar in order to move the lift towers to the perfect position. We also added the arm to the lift, using it to estimate the best position to place the lift. We still found that the arms were too slow, so we attached elastics to the lift by adding rubber bands that extended from the back of the lift tower to the middle of the arm. We also added mesh to a C-channel at the back of the lift, so that it would serve as a physical limit and also a cushion that would not damage the arms In the down position, the rubber bands were stretched, giving us potential energy stored in the rubber bands. When the arm was at a raised position, the rubber bands would retract, releasing the stored energy and giving us more speed. However, when we were lowering the arm, it was significantly slower, since the rubber bands were being stretched.

We also worked on the trueSpeed function. The following is our code for this function:

```
//drive
void  drive()
{
    int X = (Xcomponent*Xcomponent*Xcomponent)/(127*127);//allows for CUBED tru speed
```

What we now recently designed here is a model of our new TruSpeed functions. This time we omitted the function all together in search for a simpler alternative. By using the typical arcade driver controls for the X-Drive, we could control the strafing and rotation of the base in a linear fashion. Our goal however was to move away from the linear movements and focus on a greater joystick dead zone and a more comfortable driving experience. Rather than having a function being called every time we are calling each base motor, we are now determining the X and Y values directly from the joystick readings, then converting them, and finally exposing them to our base motors. Our decision to use a cubic function rather than quadratic, or a higher degree is because with the third degree, a variable will keep its sign after being multiplied out, rather than a even degree power as the variable will always become positive in that case (-x*-x*-x = -x^3, while -x*-x = +x^2). The overall benefits from this change is that the dead zone for our joystick now increased to ~20 ticks, and the acceleration rate allows for a more realistic

driving simulation. We also do not need as much precision in our handling of the joystick to minimize accidental strafing or speed changes during the drive.
Additional C++ based X-Drive Simulation:

```cpp

void VexSimulationApp::mouseMove(MouseEvent event)
{
    if (abs(event.getX() - getWindowWidth() / 2) <= 127)
    {
        j.x = event.getX() - getWindowWidth() / 2;
    }

    if (abs(event.getY() - getWindowHeight() / 2) <= 127)
    {
        j.y = -1 * (event.getY() - getWindowHeight() / 2);
    }

}

void VexSimulationApp::update()
{
    //TRUSPEEDDD
    int x = (j.x *j.x*j.x) / (127 * 127);
```

By using our spare time as well as Cinder C++ and Visual Studio, we were able to complete a fully functional simulation that could determine the speed and direction of each wheel on our X-drive based on the position of the mouse, which represents the right joystick. This simulation was the main assistance that allowed for the debugging and testing of the truspeed functions and base code. With this we discovered that unless the joystick X value was at 0, and the Y at 127, the other motors would not reach their full potentials of 127, instead they would linger and fluctuate under or around the ideal position. Now with the added truSpeed functions, we can (as seen above in the VexSimulationApp window) the joystick x position does not need to stay at 0 to have the other motors have perfect conjunction and unison.

Meeting 9/3/16 and 9/4/16 - Building and Programing the Claw

We began building the claw today. We constructed the claw using 4 gears, with 2 idler gears in the middle for spacing and so that the gears connected to each side would spin in opposite directions, resulting in the claw opening and closing. We decided to have a motorized claw; however, there was not enough grip on the game pieces, so we resorted to two options:

1. Programming a toggle button that would constantly close the claw, so the game pieces would not slip out when lifting
2. Attaching rubber bands to each side of the claw

After testing both options, we found that the toggle button complicated controlling the claw even more, and that it could break the claw motors, since it was always commanded to be closing. The rubber bands proved slower to open the claw, but it also provided the grip that we needed and it did not damage the claw motors like the toggle button did. We decided on using rubber bands, as a result. This is what our claw looks like:

The code to the bottom is an example of our recent conditionals for controlling the dual motors on the lift:

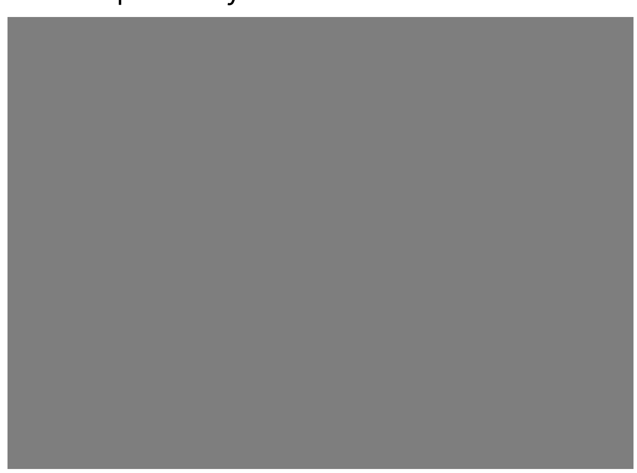
```
if(vexRT[Btn5U] == 1)
{
  motor[claw] = 127;
}
else if (vexRT[Btn5D] == 1)
```

Our method of controlling when the claw opens was by using the vexRT[BtnAB] with A being the button number and B being which specific button for that button number (ex. 5U and 6D). We chose to use 5U and 5D for the claw values because they are the most comfortable for the drivers. The claw values are also always going to be maxed out at 127 which allow the claw to be as fast as possible.

Meeting 9/6/16 - Attaching the Claw and Building the Lock Mechanism

We attached the claw to the lift arm using two hinges at two attachment points. We used 90-degree gussets to fix the claw at a good angle. However, we discovered that the claw was out of size. In order to fix this, we adjusted the c-channels and the gussets to fit in the smallest space possible. The picture to the left shows the connection.

We then attached the same lock mechanism we used for our previous robot. However, since we did not have space to attach two mechanisms on either side of the claw, we only used one in the middle. The half c-channel fit perfectly between the two motors powering the claw. We spent some time perfecting the angle in order to both allow the claw to fold up within 18X18X18 and also fold out at a good height to pick up stars and cubes. This is the final lock mechanism and claw respectively:

Meeting 10/4/16 - Designing a Claw With Pneumatics

As we were testing the claw with both stars and cubes, we discovered that using motors for our claw was not effective. The motors were too weak, and the idler gears in between tended to get loose and would disengage. The motors would produce an annoying clicking sound, and the claw would open much more than was necessary. Even with the rubber bands between the c-channels of the claw, it had difficulty exerting enough pressure on multiple stars. If the load was too heavy, it would slip out of the grip of theclaw. We decided that using pneumatics would allow us to hold stars and cubes more securely, while also giving us much-needed speed.
The code below is what was used to open and close the claw:

```
//claw
void open()
{
	SensorValue[leftSolenoid] = 1;
	SensorValue[rightSolenoid] = 1;
}

```

This program had the solenoids listed as sensors, not motors, and they could only be on at full power, or off. We made this code when we had multiple solenoid motors, one for each arm of the claw. Now however, we have only one solenoid controlling the entire claw, which seems efficient and simple.

Meeting 10/11/16 and 10/13/16 - Replacing Motors with Pneumatics

Over the course of two days, we worked on replacing the motors with pneumatics. First, we took apart the old claw. We then attached a 5-hole c-channel and a longer 2-hole c-channel to the lock mechanism and the hinges, connecting it to the arm. A longer 2-hole c-channel was required in order to have enough space for the 2 pistons we wanted to use. We fixed two pistons to the outside of each c-channel on the claw, and used 2 solenoids, one for each piston. We had some problems with one of the solenoids, which was leaking air. In the end, we connected both pistons to the one working solenoid, which worked well.

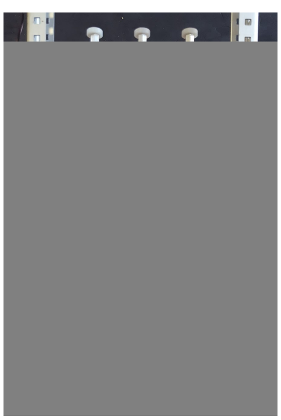
In addition, we added mesh on the c-channels to prevent stars from slipping down the claw. This helped the stars to stay at the tip of the claw as much as possible, allowing the robot to throw the stars farther. We also added short standoffs to the end of the claw in order to keep stars and cubes in the grip of the claw when turning.
This is our claw:

We modified our code to control the pistons:

```
//claw
void open()
{
	SensorValue[Solenoid] = 1;
}

void close()
{
	SensorValue[Solenoid] = 0;
}
```

Meeting 10/15/16 - Adding Elastics to Lift



Today we decided to add elastics to our lift so that it can lift more stars and cubes with more strength. We experimented with different places to put the rubber bands which would yield the most strength while not working too much against bringing the lift back down. The picture to the left shows our final configuration for the rubber bands. After adding the rubber bands, we could raise up to 1 cube and 2 stars. When carrying a lighter load, we could throw the scoring objects much farther and more consistently into the far zone.

Meeting 10/20/16 - Fixing the Base

This meeting, we fixed a problem with our base. It kept stalling when the robot grabbed multiple stars or a cube. In order to prevent this, we switched our 4-inch omni wheels to 3.5-inch omni wheels, since wheels with a smaller diameter have more torque and are less likely to stall. Although in our decision matrix we evaluated the 4-inch wheels to be more advantageous in speed, it turned out to be impractical and caused our base to stall. Changing the wheels solved the problem, and now our robot is able to pick up one cube and two stars without stalling the base. Although it is slightly slower, the base is now much more reliable.

Meeting 10/23/16 - Autonomous Program

Today, we worked on coding the autonomous program for our robot. We also mounted two speakers on the robot. Our objective for autonomous was to get 4 stars and 1 cube over the fence, which can score a minimum of 6 points and a maximum of 12 points. We used the encoder to accurately control the movement of the robot. For driver control, we adapted our truespeed function to work with an X-drive. During the previous meetings and on his own time, Gustavo completed the program:

```
task autonomous()
{
```

The controls were programmed for arcade mode, with one joystick controlling the rotation of the robot, and the other controlling forward, backward, and strafing movements. Through a series of testing the correct values of the encoder, we ended up having the base move back during its pre-autonomous period to activate the lock mechanism and open the claw. The robot would then proceed to intake the stars in between the two starting tiles, and move back behind the starting position, and strafe sideways toward the fence. After doing so, the robot would turn into a position to launch the stars, and lift. We tested the autonomous successfully many times, and the encoder proved very accurate and helpful for our purpose. As for the code, much of the autonomous task consists of calling many functions that direct the robot's movement. By doing this, we enabled the code to be easier to read, and the autonomous to be easily changed when needed. The majority of the base functions convert the wheel revolutions parameter into encoder ticks, and power the motors to a specified power parameter.

Meeting 10/25/16 - Adding a Potentiometer and Improving the Lift

At today's meeting, we added a potentiometer to the lift. We decided that we needed this sensor because the physical limit we constructed for the lift was causing our gears to click. This was harmful to both the motors and the gears. To solve this, we coded a limit for the lift into the program. Using the values given to us by the potentiometer, we determined the number of degrees that there would be at the height of the lift we wanted it to stop at. Once the potentiometer would reach this value during a match, the code would stop the lift motors without the need of a physical limit. The following is the code we used to program the potentiometer:

```
void up(int degrees, int speed)
{
    while(abs(SensorValue[liftPotent]) < degrees)
```

As a result, there was no use in having the physical limit, so we removed it from the robot. We also realized that having the rubber bands extend across the lift was not good for the motors, causing them to stall often. Instead, we decided to extend the lift arms so that the rubber bands could pull down in a linear motion, increasing the effectiveness in boosting the lift while not harming our motors. The physical limit would prove as an obstruction in this change, giving us another reason to remove it. In addition to this, we also added two C-channels as cross bars, one for each lift tower, to provide more stability and decrease the chance of the motors from stalling. The changes we made to the lift greatly improved our ability to score in the far zone and decreased the frequency of motors stalling. The rest of the meeting was driver practice. For the upcoming competition at Dougherty Valley High School, Sarang will be the driver and Christy and Kunal will be coaches. Kavin will scout out other teams and evaluate their robots to seek out a good alliance. Justin and Gustavo will be pit managers.

Dougherty Valley High School Competition 10/29/16

Today, we attended our first major competition of the season at Dougherty Valley! During qualification matches, we were given a difficult schedule against many strong teams. Because we were also paired with weak alliances, we ended up winning only 3 matches.

Despite this, we managed to demonstrate our robot's strengths and were selected as the third pick of the third seed to join teams 5776T and 5327C. In the elimination rounds, we fought our way to the top. After an intense finals match with the first seed alliance, we clinched the title of tournament champions! We are now qualified for States. We were very happy about the outcome of the tournament and felt that it reflected how hard we worked up until this point. Overall, this tournament was a great learning experience and a phenomenal start to the competition season.

WHAT WORKED	WHAT DIDN'T WORK
Brought all needed supplies to competition	Base was wired incorrectly, causing us to underperform during our first match. We spent a lot of time fixing the drive.
Scouting allowed us to get a good estimate of our opponents' strengths	Blocking with a claw was ineffective
Revised anti-tips, consisting of a standoff and a c-channel, worked slightly better compared to only standoffs	Tipped during a match and were unable to right the robot again
Revised autonomous program allowed us to collect 3 stars	Base was acting weirdly, causing autonomous to not work as expected
A second revised autonomous program based on time allowed us to score our preload and knock 2 stars off the fence; worth 3 points	Base encoder stopped working; our revised autonomous could no longer work
Scored 25 points in driver skills; a great improvement over last competition	Base was faster than last competition, but still slow when picking up stars/cubes
Communication and connections with other teams as well as performance got us into a good alliance	Robot turns too slowly when holding 3 stars/ 1 cube

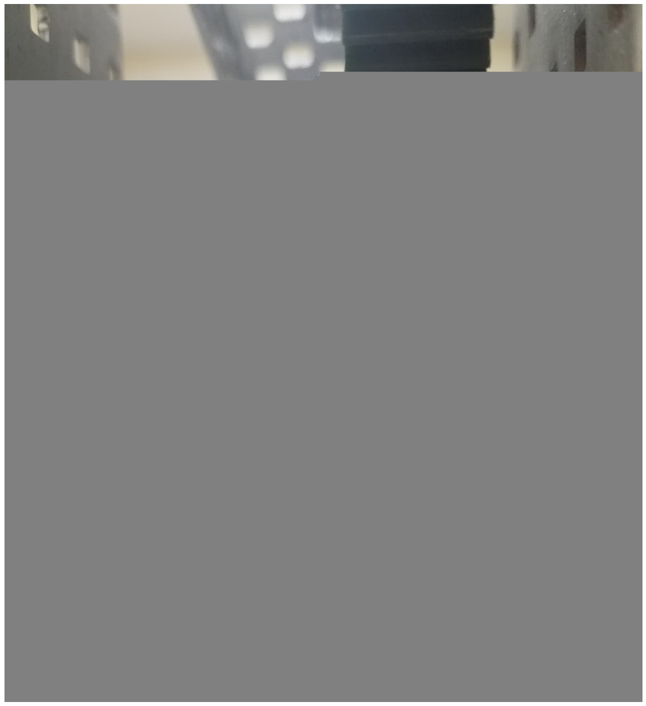
Lots of driver practice in between matches	During finals, a loose connection caused our pneumatics to stop functioning
Planning out presentation to judges	Need to be more fluent and coordinated when speaking to judges
Talking about the Truespeed function and simulation	Loose battery connection to voltage expander rendered our lift useless in a match
Pit managers ensured charged batteries were always on hand, multimeter allowed us to ensure batteries had an adequate voltage	
Got to know other teams, formed relationships	
Cheering on our teams	

Based on our performance at the competition, this is what we have to do to prepare for the next competition at Bellarmine:

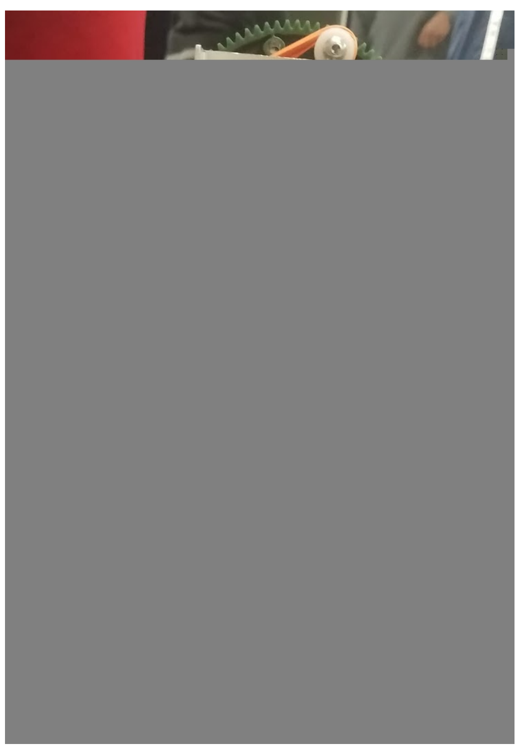
- switch encoder
- fix anti tips to be in size
- switch solenoid wire
- move speaker and c-channel so that it is easier to change rubber bands
- improve wire-management
- improve autonomous
- prepare for programming and driver skills
- (maybe) implement a high hang mechanism on the claw

Meeting 11/1/16, 11/3/16, 11/5/16, and 11/6/16 - Improving the Lift and Claw

Our primary goal for these meetings was to reduce the friction from our lift. In the last competition, we stalled a few times during both the qualification and elimination round. In addition, we discovered an alignment issue which had been causing damage to the teeth of the gear attached to the lift arm. We started by detaching the lift towers from the base and redoing the spacing. This was our final spacing:



While working on this, we simultaneously shifted the speakers and c-channels to the sides in order to provide easier access to the rubber bands attached to our gear. This is one side of our lift:



Meeting 11/11/16 - Fixing Claw and Lock Mechanism

Today, we focused on fixing the structure and pneumatics of the claw. Our pneumatics were slightly off-center and misaligned, so we fixed them. We also replaced the 5-hole c-channel originally used to connect the claw to the hinges with a 2-hole c-channel. We were able to move the two c-channels on our claw up and also move them farther apart. These changes increased our carrying capacity to 6 stars. This is our modified claw:

We also changed our lock mechanism to keep the claw in position to something more simple. Our new locking mechanism consists of a c-channel which is rubber banded to our lift crossbar. When in the closed position, it sits horizontally to the lift arm. When the claw falls out, the short c-channel is pulled by the rubber band into a position parallel to the lift arm. A standoff on the c-channel pushes against the hinge, preventing the claw from closing again. We also added rubber bands to the claw so that it would flip out more easily.

This is a picture of our new locking mechanism:

The rest of the meeting was dedicated to driver practice.

Meeting 11/8/16 and 11/10/16 - Encoder and Potentiometer

Since we discovered that our single quad encoder had a somewhat high possibility for failure or inaccuracy, we decided that it would be of great use to add a second quad encoder and re-calibrate or test the values of our potentiometer. With two quad encoders we will be able to check for values from both the encoder thus if one fails, then we still would have a backup and our auton would not be in jeopardy. Our potentiometer was acting differently than most we have heard of online, as it ranges from 0-400, rather than 0-4000 in ticks/values. After taking this into account, we were still able to program the right values and measurements for the degrees we need during our auton. The picture below shows the two encoders on our base:

Meeting 11/12/16 and 11/13/16 - Anti-tipping Mechanism

One of our goals after the DV tournament was to fix the anti-tipping mechanisms. At the competition, we were only using standoffs which were not enough to prevent tipping. Even our makeshift solution at the competition, which consisted of standoffs attached to short c-channels, was not extremely effective and would sometimes come loose and drag on the ground. We decided that we needed a new anti-tipping mechanism. Kunal came up with the idea of using linear slides. Two c-channels attached in a T-shape would be attached to a linear slide; when the claw was activated, the rubber bands would cause the anti-tip to slide out. The image below shows the anti-tip when it is deployed:

However, the anti-tips would often fall off the linear slide after stopping our root from tipping. We secured them by adding a rack gearbox bracket to the c-channel, and adding a nylock to stop the c-channel from sliding out. This is the finished anti-tipping mechanism:

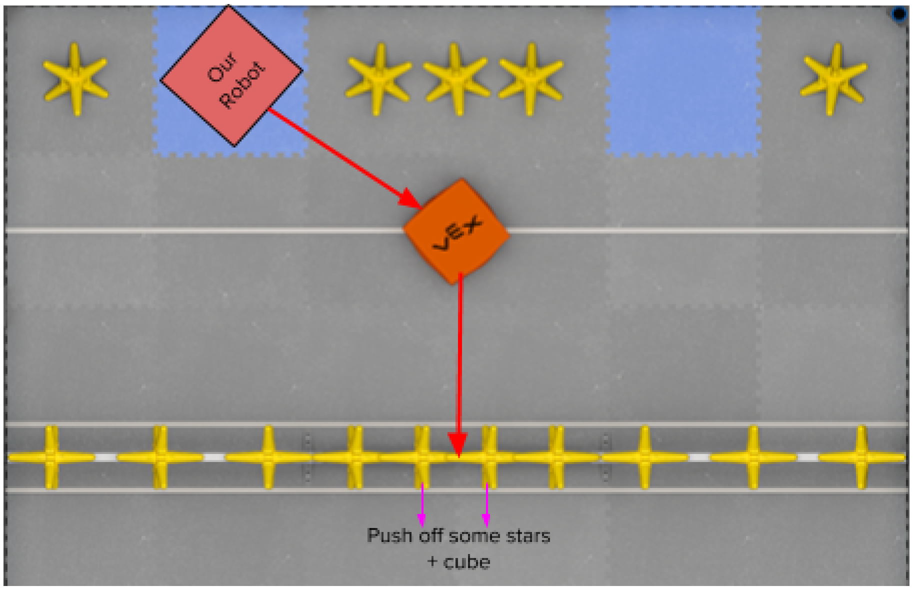
<IMAGE OF ANTI TIP>

The mechanism worked well, and prevented us from tipping 100% of the time. We were satisfied with the result. One thing to consider is the weight of the steel parts. When we develop a hanging mechanism, we may need to lighten the weight in order to lessen the strain on the motors when hanging.

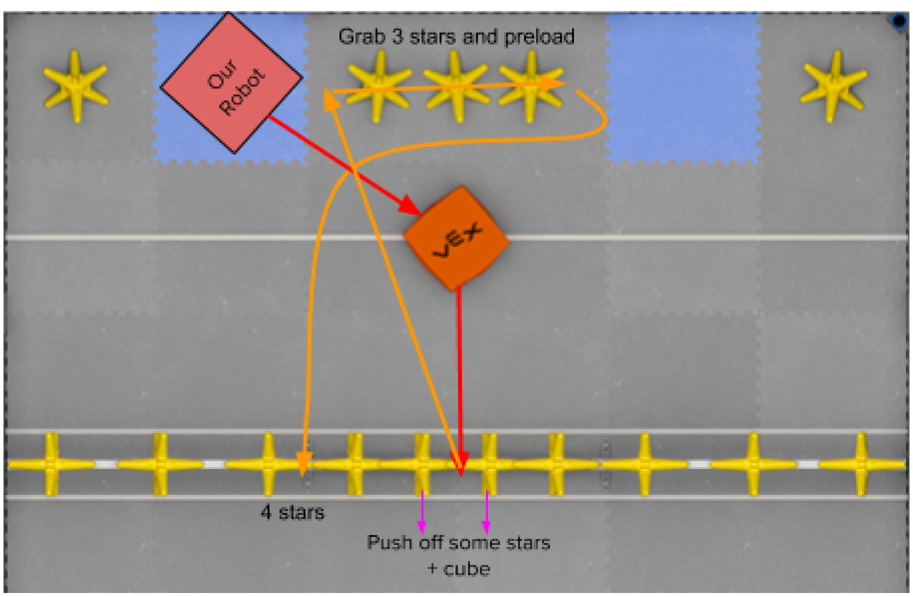
Meeting 11/17/16 and 11/18/16 - Driver Skills and Autonomous

With these meets, we were able to refine our auton and complete more testing because we were having difficulties during our previous meetings. Although the process was long and tedious, we tried many scenarios and tests and were able to gather the correct values for a very effective autonomous. We have a field mockup and used an actual field in our school room for precise measurements, but we spent most of our time having the backbone of the autonomous finished and thinking about more effective ways to gather as many points as possible. Because of our past difficulties with our encoders we added a second encoder as a failsafe and are planning to double the potentiometer count, so that if one fails, then we have a plan B.

For the 15-second autonomous period before matches, we will grab the cube in the middle on our side and throw it into the far zone, knocking over several stars in the process. This would yield about an 8 point auton. This is a diagram of our match auton:

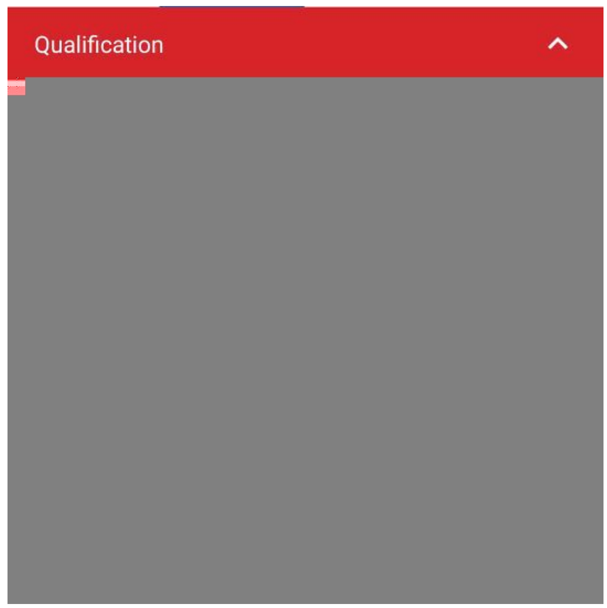


For programming skills, we are planning to have a 12+ point auton composed of the back middle stars, our preload, and the cube in the center of the field. We are planning on having the main recursion of hitting off fence stars and launching off back stars after the main autonomous and have a good autonomous so that we can build onto it for programming skills. Currently we have it so that the robot completes the original auton (red arrows), then goes to gather the other back stars and hit off the other stars that are on the fence (orange arrows).



This is the last meeting before the competition. At Bellarmine, Sarang will be the driver, Christy and Kunal will be coaches, Justin and Gustavo will be pit managers, and Kavin will scout other teams.

Bellarmine Competition 11/19/16 - “It Doesn’t Work”

WHAT WORKED	WHAT DIDN’T WORK
Anti-tips worked successfully in preventing the robot from tipping	Anti-tips during autonomous dragged on the ground, causing the base to stall for the rest of the match 
Lift didn’t stall - was faster due to capability of adding more elastics	Star got stuck inside robot and on the anti-tip, causing the lift to stall once
Improvised autonomous program worked a few times	Autonomous program was not very consistent nor accurate throughout competition
Wires stayed intact	Wiring still needed improvement in organization
Better pit management	Sensors were not adequate and unreliable

Speakers worked	Elastics/tubings got caught on anti-tip, preventing our lift from functioning
Higher star capacity than previous claw	Autonomous program was programmed last minute and didn't allow for testing
Lock mechanism worked properly	Low driver practice time
	Claw had poor grip (for large quantities)
	Far zone frequency was low

Meeting 11/27/16 - Brainstorming and Stuff

This was our first meeting after the competition. Given our performance, we needed to seriously rethink our strategy and our robot.

As a team, we decided to rebuild our robot to have a 4-motor tank drive and a 6-motor 4-bar lift with a passive flicker, while keeping our pneumatic claw. We also decided to redesign our claw into a more bent shape in the middle, giving us a bigger star capacity. The lift and flick design would help us to throw scoring objects with more force and more consistently into the far zone. In addition, the flicker would allow us to reach the far zone without as much momentum and speed as our previous design, resulting in optional elastics, making high hang a more viable option for us.