



"Game Manual 0"
a guide for FTC teams
enjoy!

Published on : September, 2019

Last updated : February, 2020

Game Manual 0 Table of Contents

1	Introduction	1
1.1	Meta	2
1.2	Getting Started in FTC	5
1.3	Rookie Mistakes	10
1.4	Kit and Hardware Guide	15
1.5	Tools List	41
1.6	Materials Guide	42
1.7	Drivetrains	45
1.8	Power Transmission	81
1.9	Linear Motion Guide	102
1.10	Arms	136
1.11	Passive Intake/Claw	142
1.12	Active Intake	145
1.13	Motor Guide	159
1.14	Servo Guide	165
1.15	Electronics	166
1.16	Software	182
1.17	Appendix	202

Introduction

Hello! Welcome, and thank you for taking the time to read Game Manual 0 (GM0), the premier resource for new and upcoming FIRST Tech Challenge teams! If you are a rookie or a new team, we would especially like to welcome you to the FTC community. We want to make your beginning steps and transition into FTC as seamless as possible, and offer some of our own experience and advice compiled over the years.

The goal of GM0 is to create a comprehensive guide for FTC teams. Physical resources for FTC are few and far between, and online resources are scattered across the interwebs. Typically, the largest inhibitor for newer teams is the lack of knowledge base, as robotics is a relatively new STEM field. New teams also have not made connections to experienced teams who might mentor or offer advice throughout the season. Thus, they are left in the dark to figure things out, seemingly with no way out. While the journey of learning cannot be shortcut, GM0 seeks to address these shortcomings by providing a starter's guide to the hardware and software in FIRST Tech Challenge.

When perusing this guide, it is important to keep in mind the authors' perspective. Many of the teams who contributed in the writing of this guide are veteran championship level teams in the upper echelon of FTC. **This means that most of our recommendations are almost solely from the competitive advantage standpoint.** For example, teams can learn just as much from using a Tetrix kit compared to an Actobotics kit. Just because there is a competitive advantage to one does not translate into a learning disadvantage for the other. The guide has a plethora of useful information and knowledge (such as FTC and basic engineering principles) for all teams, including teams which are primarily trying to learn rather than compete. However, do take our recommendations with a grain of salt. We want to help all teams in FTC, but as our experience has been on the competitive side of the fence, certain parts of the guide may not be very applicable to some teams.

Furthermore, the guide is written with a specific audience in mind - first year rookie teams who have just begun their journey in FTC. Undoubtedly there is information in the guide that could benefit nearly every team (it's impossible to know everything about FTC) - but our recommendations are geared for the more inexperienced teams. For example, we generally discourage teams from using arms and instead use linear extensions. However, there are many experienced teams with the capability to build such arms, and we do not discourage such an endeavor. Keep on the lookout, though - a future version of the guide for veterans may appear.

Before diving in, we would like to add a short disclaimer: this guide is not about how to build a specific drivetrain, linear slide, intake, etc. The purpose of GM0 is simply to provide knowledge and advice to teams about the possible options, as well as including some tips on how to get started. GM0 has knowledge and advice, but not instructions. **It is of the utmost importance that all teams learn the proper way (by following the engineering design process), not by reading a step-by-step guide or instruction manual. Thus, while GM0 has plenty of advice, we do not have specific steps included. Good luck, and have fun in FTC!**

1.1 Meta

1.1.1 Contributors

Game Manual 0 would not have been possible without the many contributors who have sacrificed their precious summer freetime (or merely found another excuse to procrastinate their work). GM0 had been my dream for quite some time, and as an alum tremendously grateful for all that FIRST has taught me, I wanted to create GM0 to preserve and spread the knowledge base that might have been lost with the FTC class of 2019 and beyond. For this reason, I wanted to recognize all the contributors who have helped to make GM0 a great resource for FIRST Tech Challenge teams. Feel free to contact any of the names listed below via the email (gamemanual0@gmail.com) or through the other contact emails - I'm sure that they would be more than willing to help you out!

Kleptomaniac

Managing Editors

- Tom - FTC Alum - 3736 Serious Business- seminole3736@gmail.com
- Frank - FTC 8581 Aedificatores - fgportman00@gmail.com
- Justin - FTC 9656 Omega - evhsomega@gmail.com
- Nathanael/Kleptomaniac - FTC Alum - 13075 Coram Deo Robotics - nathanchu@utexas.edu

Hardware & Electrical

- Kelvin - FTC Alum - 731 Wannabee Strange
- David - FTC 2753 Team Overdrive
- Sam - FTC 2753 Team Overdrive
- Tyler - FRC 3184 Blaze Robotics
- Tom - FTC Alum - 3736 Serious Business- seminole3736@gmail.com
- Derek - FTC Alum - 5484 EnderBots
- Karter - FTC Alum - 5975 Cybots
- Neo - FTC 6710 Sigmas - ftcsigmas@gmail.com
- Ethan - FTC Alum - 7236 Recharged Green - goBILDA
- Cole - FTC 7548 SPAREPARTS
- Ian - FTC 7842 Browncoats
- Andrew - FTC 8417 'Lectric Legends - team@lectriclegends.org
- Eric - FTC Alum - 8417 'Lectric Legends - team@lectriclegends.org
- Frank - FTC 8581 Aedificatores - fgportman00@gmail.com
- Kevin - FTC 9048 Philobots
- Justin - FTC 9656 Omega - evhsomega@gmail.com
- Baylor - FTC 10641 Atomic Gears

- Peter - FTC 12533 Inception - team@ftc12533.org
- Nate - FTC 12897 Newton's Law of Mass
- Adham - FTC 14875 LightSpeed
- Dom - FTC 15692 Rust in Pieces huppdo@rustinpieces.tech
- John V-Neun - FRC 148 Mentor - JVN Design Calculator

Software

- Wes - FTC 3658 Bosons - wgfransen@gmail.com
- Daniel - FTC 7026 JDroids - dansman805@gmail.com
- David - FTC 7236 Recharged Green - pellarobotics@gmail.com
- Frank - FTC 8581 Aedificatores - fgportman00@gmail.com
- James - FTC 14298 Lincoln Robotics - ftc14298@gmail.com

Hosting

- Benjamin Ward - FRC/FTC Alum - Copperforge - blward@copperforge.cc



1.1.2 Credits

We would also like to recognize the teams who have provided pictures of their robot(s) for this guide. We hope you find the pictures to be informative as real examples of what teams have achieved in the past seasons.

FTC Teams

- 724 RedNek Robotics Wun
- 731 Wannabee Strange
- 2818 G-Force
- 2856 Tesseract
- 3736 Serious Business
- 3846 Maelstrom
- 4042 Nonstandard Deviation
- 4545 ViperBots Ouroboros
- 5040 Nuts and Bolts

- 5064 Aperture Science
- 5975 Cybots
- 6022 To Be Determined
- 6299 ViperBots QuadX
- 6929 Data Force
- 7161 ViperBots Hydra
- 7172 Technical Difficulties
- 7182 Mechanical Paradox Cubed
- 7203 KNO3
- 7209 Tech Hogs Robotics
- 7236 Recharged Green
- 7244 OUT of the BOX Robotics
- 8103 Null Robotics
- 8148 Aleph Bots
- 8221 Cubix
- 8375 Vulcan Robotics
- 8393 BrainSTEM Robotics
- 8417 'Lectric Legends
- 8640 Trojan Robotics
- 8644 Brainstormers
- 8680 Kraken-Pinion
- 8686 Height Differential
- 9794 Wizards.exe
- 9804 Bomb Squad
- 9829 MakBots
- 9889 Cruise Control
- 9971 LANBros
- 10030 7Sigma
- 10641 Atomic Gears
- 11115 Gluten Free
- 11190 Mechadojos
- 11260 Up-A-Creek Robotics
- 11316 Weapons of Mass Construction
- 12599 Overcharged
- 12670 Eclipse
- 12897 Newton's Law of Mass

- 13075 Coram Deo Robotics
- 15692 Rust In Pieces

FRC Teams

- 694 StuyPulse

1.1.3 Team Interest Form

The editors of Game Manual 0 would like to be able to gauge the different kinds of teams who have read the guide. If you could be so kind, please fill out the [Team Interest Form](#)¹. It asks for some information about your team demographics and whether your team would be interested in receiving further information and updates regarding Game Manual 0. We would appreciate your feedback!

1.2 Getting Started in FTC

Welcome to FIRST Tech Challenge! As a new team, you may be slightly overwhelmed and disoriented, wondering where to start. No fear! This guide is meant to help navigate you through your beginning year in FTC.

After registration and going through the STIMS process in the FIRST website, you'll likely have to purchase a starter kit (**which we highly recommend for new teams**) and electronics kit in order to build your robot. The electronics kit is standard for all teams, but please read our kit guide to find out which kit is right for your team.

The physical components of a robot are split into the hardware and electronics section. On the hardware side, you have your [drivetrain](#) (page 45), [linear extension mechanisms](#) (page 102), a [claw](#) (page 142) or [intake](#) (page 145) to pick up game elements, as well as any other mechanisms specific to the game. On the electronics side, there are the two REV expansion hubs and the Android phone, along with extras such as the Servo Power Module, if necessary.

The kit guide provides information on [selecting the right kit for your team](#) (page 15). Along with this, we have included a [list of tools](#) (page 41) that may be necessary or beneficial for your team to purchase. The building guides for drivetrain, linear extension, and intake go over the different routes teams can take, as well as providing advice on which to choose. However, we want your team to have the full learning experience, so we don't provide specific instructions for building mechanisms.

Here's an overview of how FTC works. Each season, Game Manual 1 (GM1) will be updated sometime in the summer. GM1 contains the general rules of FTC, competition and advancement, as well as hardware and software restrictions. Game Manual 2 (GM2) is released on kickoff day. It provides details about the specific game that your team will be participating in this year. Every year, teams are required to build a robot according to rules in GM1 and document what they have done in an Engineering Notebook. Each team will have at least one chance to compete, either at Qualifier Events or League Meets. At Qualifier tournaments, every team will play 5 to 6 matches. The top 4 teams are captains and form four alliances of three teams each. More information on the selection process are in GM1. The top teams will advance to the next level, generally the State or Regional Championship. Leagues are different, having a few meets where teams participate in matches. At the end, the top teams from the League Championship will also advance to the State or Regional Competition.

¹ https://docs.google.com/forms/d/e/1FAIpQLSeRHE0rTH2KcW1URXdBlq9mqAJaUC36MDjEHv7sTCEF3SHHZw/viewform?usp=send_form

Game Manual 2 is published when the season officially starts, and provides game specific rules that change from year to year. While we cannot provide any help for GM2, here are some rules in GM1 for teams to be aware of, as many new teams run into problems during their tournament because of it.

- All teams **MUST** bring the Consent and Release forms to every competition. Teams will not be allowed to compete if the forms are not signed.
- The maximum **starting** size up the robot is an 18 inch cube. The robot may expand after the match starts. This sizing limit has been in place since the start of FTC.
- Starting in the 2018-2019 and continuing in 2019-2020 season, the maximum weight of the robot is 42 lbs. This may change in later games, but as of the time of writing, this remains the restriction.
- Sharp corners, liquids, and certain wheels are not permitted. Be sure to file down all sharp edges for your safety (sticking your hand into the robot can result in a gash otherwise) and so that your robot won't damage the field or other robots. We advise you to use the wheels that are listed in this guide only, since you risk unknowingly using illegal wheels otherwise.
- Any wheels that the inspectors think will damage the field tiles are not legal. This includes any wheels with ridges or bumps, such as RC wheels, as they can dig into the floor tiles and rip them up.
- If the inspectors are uncertain about the legality of your wheels, they will have you do the Wheel/Tread Playing Field Damage Test, where the robot is run into an immovable surface (wall) for up to 60 seconds to see if there is any damage to the field tiles.
- Hydraulic and pneumatic devices are prohibited. Sorry, only the FRC kids get nice toys.
- Commercial Off the Shelf (COTS) parts are allowed **as long as they only have one degree of freedom**.
- One degree of freedom generally means that the mechanism can only move along one axis. Typically, one DoF = one joint. Multiple joints allow movement in 2 or more planes.
- **Each team must have a robot power button next to the robot power switch.**
- Each team must have a secure battery and phone mount.
- The recommended batteries are the REV slim battery and Modern Robotics Battery (sold on goB-ILDA.com). The Tetrax battery is fine, **however, Tamiya connectors are fragile and unreliable.**
- Only 8 DC motors and 12 servos are allowed. The VEX 393 EDR counts as a servo.
- Modifying servos and replacing fuses are not allowed.

1.2.1 Know Your Lingo

Competitions

Affiliate Partner Regional director for teams and runs official competitions.

Qualifier Tournament with 5-6 qualifying matches and best-of-three alliance elimination matches. Judging included. Will advance to States/Regional level.

League Meet Low-stress tournament with matches counting towards league rank. No judging.

League Championship Tournament with ranked matches and alliance elimination matches. Judging included. Will advance to States/Regional level.

Alliance Group of two or, in eliminations, two or three teams that compete. In elimination matches each team must play at least once.

Ranking Points (RP) Primary basis in rankings. Win = 2, tie = 1, loss = 0.

TieBreaker Points (TBP) Secondary determinant in team rankings. Number of points the lower-scoring alliance scores.

Qualification Match 5 qualifying matches with random alliance partners and opponents will be played to determine team rankings.

Elimination Match 4 alliances of 2 or 3 teams each compete in best of 3 matches to determine winning alliance. Also known as elim matches, or just elims.

Robot Components

DC Motor Open-loop RS-555 series 12 volt motors used to power mechanisms.

Servo Closed-loop servo-motors used for precise movement. Controlled by PWM signals.

Drivetrain (DT) Mechanism responsible for the movement of the robot.

Intake Mechanism responsible for picking up game elements from the field by using rotational motion.

Claw Mechanism responsible for picking up game elements by grasping.

Linear Extension/Slide Mechanism responsible for extending parts beyond the starting configuration of the robot.

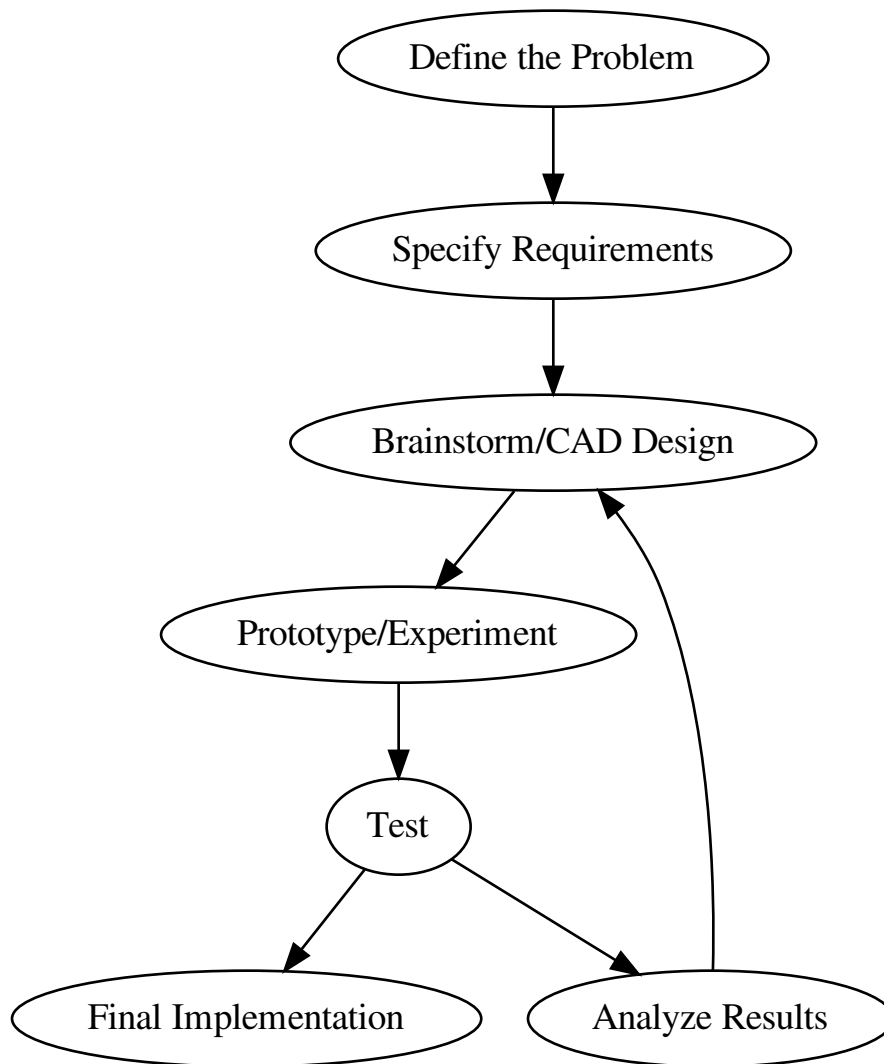
REV Expansion Hub Hardware controller for the robot, connected to Robot Controller phone. Controls motors, servos, sensors. 2 hubs maximum.

Driver Station Android phone connected to the gamepad controller. Drivers interact with the DS phone to start/stop robot.

Robot Controller Android phone connected to the Expansion Hub. Connects to DS phone via WiFi-Direct.

1.2.2 Engineering Design Process

Note: The exact process will vary from team to team, so the graphic is not meant to be a hard and fast rule.



Components of the Engineering Design Process

In general, the design process contains these steps; however, every team will modify the process to their needs and limitations.

- Define the problem - what are you trying to solve? what is the timeline?
- Specify requirements - what are necessities for your proposed solution?
- Brainstorm/CAD design - draw or sketch a prototype on paper or in CAD
- Prototype/Experiment - put a first design together using physical materials
- Test - ensure you thoroughly test every part of your prototype to detect flaws

- Analyze results - examine what you learned from your tests to iterate successively
- Final implementation - polish and solidify a final design that won't change

As an example, let's look at an imaginary team building a drivetrain for Rover Ruckus (RR2), which was 2018-2019's game. In RR2, there was a crater, which was about 3" tall, and could be traversed. In that year's game, there were two main options: traverse said crater, or not traverse it, and instead reach over with an arm.

The team must first specify requirements for the drivetrain. One of the most important aspects of a drivetrain is maneuverability. Another requirement could be speed, traction, reliability, etc. From the team's specified requirements, they would now look at drivetrains that fit their requirements. If the team wanted to traverse the crater, then a 4WD or 6WD would be an optimal design. If they did not need to traverse the crater, many options are still on the table such as the holonomic drivetrains.

The next step would be to brainstorm actual designs. It would be prudent to have more than one student to design one drivetrain prototype, so that the team could test more than one drivetrain.

From there, a team could begin the prototyping and testing process. This could involve tests of the time taken to cross the crater, time from crater to lander, top speed, maneuverability, etc. Simple tests may be conducted instead of complicated ones. For example, if the robot is projected to weigh 30 pounds, a dumbbell placed on top of the drivetrain simulates the extra weight well to see if the drivetrain can get over the crater easily and reliably, so the robot will not get stuck on the crater edge.

Next, analyze results and iterate. If, for example, there was too little clearance, and the drivetrain got stuck on the crater part of the time, then it would be necessary to raise the drivebase in order to remediate this problem. However, don't change too many things at once - **you want to change ONLY ONE variable at a time**, or else you won't know what may cause a further problem should it arise. Always try to change one variable at a time. **It often takes multiple iterations to get things right, so don't be discouraged if your second or even third attempt doesn't perform as well as you think it will.** Many teams have 10+ iterations of intake designs in order to refine and hone down their design to be optimally efficient. While that many tries isn't recommended for new teams, don't be afraid to modify *one factor* at a time to isolate and solve problems.

Your final implementation could be just an upgraded prototype, or if you had used subpar/scrap materials for a prototype, you could change them out for durable ones.

1.3 Rookie Mistakes

1.3.1 Rookie Mistakes: Building

Problem	Solution
2 motor drivetrain - Less power - Less acceleration	4 motor drivetrain + More power, higher efficiency + Improved acceleration
Pushbot - Poor agility - Poor top speed	Mecanum, 6WD, etc. + More agile/higher top speed + Customizable gear ratios
Claw - Control 1 element at a time - Easy to break	Intake + Control multiple elements at a time + Much more efficient
Spur gear gearboxes - Not for high load use cases - Will break under shock load	Planetary gearboxes + For drivetrain and high load + Resistant to shock loads
Single/multi-axis arm - Requires high gear ratio - More complex	Linear extension + Generally faster than arm + Much more precise

2 Motor Drivetrain → 4 Motor Drivetrain

In general, it is not recommended for teams to use 2 motors on the drivetrain, but instead use 4. This is mainly due to the added power and increased acceleration 4 motors provide. Typically, top speed is determined by the gear ratio and the motor specifications, not the number of motors. **However**, acceleration is affected by the number of motors, and as FTC robots need to change direction and accelerate numerous times per match, slow acceleration has a significant adverse effect on the competitiveness of the robot. In addition, 2 motor robots may struggle to get over obstacles or climb up ramps, due to less power. Typically, it is always possible to build a competitive robot with 4 motors allocated to drivetrain, and 4 motors to other mechanisms, so there should be no reason to skimp on drivetrain motors.

Pushbot → Mecanum, 6WD, Other Recommended Drivetrains

The pushbot drivetrain, commonly built by first-year teams using the FIRST-provided guides, should not be used as a competitive drivetrain. We do recommend teams who have purchased the Tetrix kit to build it

for educational purposes only - that is, to get familiarized with the parts and basic building principles using a channel-based kit. However, we do not advise that teams use that pushbot at a competition due to its many flaws. The first major flaw is that the pushbot is powered by 2 motors, and as stated above, there isn't a reason to stay with 2 motors on drivetrain. Secondly, the pushbot has poor top speed and turning ability, given that the gear ratio (40:1 on 4 inch wheels) is half the speed that many teams use. Thirdly, it is not advisable to use direct drive. However, most if not all of these problems will be solved by using a four-motor drivetrain such as the ones recommended in the Drivetrain guide (Mecanum, 6WD). Therefore, it is recommended for teams to refer to the [Drivetrain](#) (page 45) section and see which drivetrain would fit best for their overall game strategy.

Passive Intake/Claw → Active Intake

Active intakes (ones with continuous rotational motion) should always be prioritized over passive intakes and grippers because active intakes are much more efficient and effective in picking up most game elements. The exception is that a claw should be used for irregularly shaped objects that would be impossible to control via intake; for example, the relic in Relic Recovery. Intakes have two major advantages over claws - the first being that intakes can control multiple game elements at a time, and the second being that intakes are indiscriminate at picking up objects, making them much more efficient. Claws can only pick up one object at a time, and the driver needs to aim the claw at that specific object to grab it. With an intake, the driver does not need to focus on one game element - instead; intakes will just pick up anything in its path, if designed properly. Claws are also prone to breakage, and thus suffer to defensive robots. They are also generally more fragile than intakes. Therefore, active intakes are as a result much more efficient than claws. Nearly every competitive robot from past years have used active intakes to great effect, so there is plenty of precedent to follow.

Spur Gear Gearboxes → Planetary Gearboxes

Spur gear gearboxes have inherent disadvantages to planetary gearboxes. Spur gear gearboxes should not be used in high-load situations, primarily because the gears can strip and destroy the gearbox. Planetary gearboxes are much better suited for drivetrain and arms, due to the configuration of the sun and planet gears. In addition, spur gear gearboxes are prone to shock loads; therefore, direct drive is definitely not advisable on drivetrains. Refer to the Motor guide for more complete information on gearboxes. **This refers to using spur gear gearboxes which are attached directly to the pinion gear of the motor. It does not mean external ratios outside of the motor gearbox, which will always be in a spur gear configuration.**

Single/multi axis arm → Linear extension

Teams are generally advised to stay away from arms and move in the direction of linear slides, primarily due to the issue of complexity. This is because arms typically are less effective than linear extensions and are harder to implement properly. Arms require a high gear ratio, which usually in turn will require an expensive gearbox (e.g. [VersaPlanetary](#)²) that teams may not want to invest in. Furthermore, arms must be supported extremely well to bear the torque that the motor provides. In contrast, linear extensions do not need to worry about gear ratios and gearboxes. They can be optimized to be more efficient than arms, and typically are more precise, as linear motion is easier to control than angular motion. Furthermore, linear slides can have more extension than arms, with some reaching over 3-4 feet in length.

² <https://www.vexrobotics.com/versaplanetary.html>

1.3.2 Rookie Mistakes: Competition

Problem	Solution
Neglect drive practice - Drivers unfamiliar with robot - Robot reliability untested	Constant driver training + Drivers comfortable with controls + Proven and tested robot
No game strategy - Lack of contingency plans - Weakens alliance strength	Strategical driving + Only purposeful actions taken + Efficient and effective driving
Fully driver-controlled - Less efficient - Requires more practice	Partially automated tasks + Relieves stress on driver + Removes human error

Neglect driver practice → Constant driver training

A persistent problem with teams is the lack of driver practice. No matter how good your robot is, the robot is only 50% of the equation. The driver(s) are the other 50% which determine the success of the team as a whole. Just like a sports team, a robotics team requires practice. Even if individual players have great talent, only practice and communication can guarantee success as a team. Even if your robot is the best in the world, and inferior robot with a competent drive team more than likely will beat your robot. Drive teams need to communicate with one another and practice for the unexpected. A high school sports team practices about 15-20 hours weekly. To be successful, that is the same amount of time that you need to devote to your robotics team. By April's world championships, many top teams had finished hundreds of practice matches. This means that 1) the teams were dedicated to consistent drive practices and 2) their robot was reliable enough to survive hours upon hours of operation. Driver practice not only familiarizes the driver(s) with the robot and serves as a test of robot reliability, it also simulates in-game conditions. Learning to push the limits of your robot should be done in practice, not at competition. This way, drivers will become more comfortable driving under stress and pressure. Some teams even go so far as to play loud music or say distracting things during practices to add in a little extra.

No game strategy → Strategic driving

Similar to drive practice, this is something that many inexperienced teams ignore. Another sports example is handy - even with the most talented team, they won't go far without good game strategy. An inferior team with better strategy execution could even pull off an upset. Planning a strategy ensures that every second in the 2:30 game time is used to the maximum efficiency, which yields more points. For example, drivers should know exactly where the robot needs to be positioned after the autonomous → tele-op switch. Practicing this switch will save a couple of seconds when drivers have to think "what do I do now?" Knowing when to transition from a tele-op to endgame objective is equally important (hint: perfect one first) and will save valuable time. Strategy should **always be used to maximize points** - whether this is a positioning strategy to access the game elements, or a defensive strategy to hinder the other alliance from scoring. **Remember, denying the other alliance 10 points is the same value as scoring 10 points.** However, it is not

advised for new teams to play defense due to the specific rules surrounding this strategy. If a team wishes to execute a defensive strategy, be sure to read all the rules as defense can incur penalties/cards if done improperly.

Fully driver-controlled → Partially automated tasks

Autonomous should not be limited to only the autonomous mode. Automating simple tasks can be a real time-saver and efficiency boost to teams. First of all, automating tasks can save time and reduce the need for driver multi-tasking. Drivers should always be controlling the robot with as few button presses as possible. For example, automatically lowering the lift with a delay after the game elements have been deposited would save a button press. This also has the advantage of eliminating driver error if the driver has too many operations to carry out and relieving stress on the driver.

1.3.3 Rookie Mistakes: Principles

Problem	Solution
Do everything at once - Robot becomes half-baked - Cannot excel in one area	Perfect one objective first + Robot is highly optimized + Consistently excels in one area
Overcomplicate - More time needed to iterate - Less reliable	Simplify + Best designs are usually simplest + Less moving parts
Score-first designs - Neglect proper principles - Often wildly inconsistent	Design for consistency + Usually reliability > scoring ability + Great plus for alliance selection
Build haphazardly - Build with subpar materials - Inadequate support structure	Build for reliability + Remove unneeded moving parts + Eliminate single points of failure
Fret about design - Wastes testing time - Design alone isn't enough	Focus on execution + Make a decision, then stick to it + Execution often beats design

Do Everything At Once → Perfect One Objective First

A common pitfall for first year teams is trying to accomplish all the game objectives at once, especially in tele-op and endgame. **This is highly discouraged** because oftentimes new teams may not have the experience to do so. Too often, we see teams bring half-baked robots that will attempt to do everything in

a match, but excel at nothing. This robot could be much more successful if the team spent their time to perfect one mechanism first. Teams should always remember the principle that a robot that can complete one thing consistently will likely be more competitive than the robot that does everything inconsistently. We recommend teams focus on one objective during tele-op/endgame and perfect it.

Typically, teams which have a solid autonomous and consistent endgame can be competitive at the Qualifier level, and this is a recommended goal for new teams.

Overcomplicate → Simplify

Another common trap that teams fall into is to overcomplicate needlessly. Simplicity and compactness are two hallmarks of engineering. While some robots are very complicated, keep in mind that those teams are generally experienced. However, many world-class teams often build designs that are ingenious yet ridiculously simple. Some advantages to simplicity are that the robot usually has less points of failure, given that the robot has less moving parts. Additionally, it takes much less time to iterate through and perfect a simple mechanism as opposed to a complicated one. Keeping things simple can be achieved through a couple of ways. First, try to limit the degrees of motion that the mechanism operates in. For example, a linear slide goes in and out in a straight line, as opposed to an arm, which rotates along an axis. Doing so will serve to eliminate forces that otherwise could adversely affect the mechanism. Another way to simplify is to build for the shortest travel distance. Obviously, the shortest distance from A to B is in a straight line, so teams should strive to keep the game elements approximately within a reasonably straight line. This can help in solving possible problems if the game elements need to change direction too many times.

Score-first designs → Design for Consistency

Teams should prioritize consistency over scoring ability. The tortoise beats the rabbit. An overused parable, but it still holds a kernel of truth. Why? Because the tortoise, which plodded along consistently, beat the rabbit, which had hot and cold streaks. A hallmark of any successful team is consistency and reliability throughout the competition season and even across seasons. Without the power of consistency, it will be nearly impossible to win games, let alone a tournament. Too many teams fall into the pit of prioritizing scoring ability more than anything else, which is a grave error. In keeping with the first tip, to perfect one objective first, this practice will serve to increase consistency. **While scoring ability should be a priority and objective when designing mechanisms, it is not everything in this game. We advise being consistent at low and medium scoring levels than inconsistently scoring at a high level.** Focus on being able to do that one thing every single time throughout your matches, and you will begin to see how important consistency is. **This tip is equally as important during alliance selections. Top teams will prioritize teams that are consistent far more than scoring ability.** They are not afraid to look at teams who can't score much, but can contribute every time to the alliance score, rather than selecting a boom-or-bust pick.

Build haphazardly → Build for reliability

When building, teams often overlook a key principle: build for reliability. All too often, teams skimp on the quality of construction as well as materials, which leads to one of the most common reasons for broken robots: part failure. Teams also do not take into account the rigors of competition and build as if the robot will not encounter opposing robots. Sufficient driver practice will be able to better simulate in-game conditions and test the reliability of the robot. To remedy this problem, refer to the [Materials Guide](#) (page 42) to gain a better understanding of what materials are recommended for use. In addition, teams often forget to account for twisting or compression forces that may occur upon the mechanism. While we cannot give any specific recommendations, do keep in mind what forces the support structure of your mechanism must bear along the full range of motion, and account for what occurs when it might hit another robot/field wall/field. Building more robustly is always worth the time spent. Furthermore, a common cause of robot disconnect

is wiring issues. Refer to the [Wiring section](#) (page 167) for more information; in short, make sure to plan ahead and leave space for wires, and use strain relief whenever possible. All these tips combined will help your robot become more reliable, a key characteristic of all world-level robots.

Fret about Design → Focus on Execution

A good execution of a bad design will beat a bad execution of good design. FTC is all about how well you execute in both the mechanical aspect and the driver aspect. It is very possible to take a bad design, execute it well, and still be competitive at a high level. Even though not many teams are able to do so, it still goes to show that the method of implementation is very important. When brainstorming designs, try not to get hung up on small details. It is important to discuss different designs and debate the pros & cons, but after a design has been picked, stay with it unless there are major flaws that were originally overlooked. Changing designs will throw away the time spent on the original design, when teams could have kept improving it or practiced more. It is possible to rebuild your robot mid-season, and many top teams have done so to great success. However, this is not recommended for rookie and new teams due to the general lack of experience. Focus on how you can implement your design to be as effective, efficient, and refined as possible, instead of “what if I did this?”

1.4 Kit and Hardware Guide

Before beginning your season, your team is presented with a crucial decision - one that will undoubtedly heavily influence the team’s direction of hardware design in at least the first year. The FIRST storefront currently offers two kits for new teams: the TETRIX kit from Pitsco, and the REV Starter kit from REV Robotics. However, teams should carefully consider other options, such as Actobotics from Servocity and goBILDA, before selecting a kit. Every build system has advantages and disadvantages, which we have highlighted in our detailed build system guides below. While the guide may be slightly biased, we purposefully chose to do so, stemming from our own experience with the different kits, so the slant is there with reason. As with pretty much anything in FTC, there is no one right answer - but of course there are better answers than others. We hope this guide gives some solid advice on which kit might be the best for you.

1.4.1 Why use a kit?

After all, plenty of successful robots have not followed this framework. However, we still recommend new teams purchase a starter kit for one big reason. Established teams are all but guaranteed to have spare parts lying around to use to build their next bot. However, rookie teams, as is obvious, don’t have the plethora of parts from previous seasons to use. Thus, new teams should purchase these parts themselves in order to have something to build from once the season starts, and starter kits offer these parts for less money than they would be if bought individually. As will be discussed, options that don’t involve kits exist and are certainly very useful, but sticking to parts designed for FTC is recommended for new teams as a starting point.

1.4.2 Kit Options

Tetrix

The TETRIX build system is a fixed pitch, channel-based building system that has existed for many years. The TETRIX build system revolves around 32 mm aluminum c-channels and a 16mm bolt circle, on 16mm spacing. This means that the centers of the circles are 16mm apart from each other. TETRIX is patterned

along these channels to create plenty of mounting options for building structures. These channels have a tendency to flex and bend under load, often requiring reinforcement using multiple channels to maintain their shape. While TETRIX channel is measured in metric units, the system employs Imperial (SAE) bolts and Imperial chain. This mix of units means that things often don't quite line up how they're supposed to. This means that additional tensioners are required when using chain on TETRIX. Beyond this, TETRIX gears and sprockets offer very limited options for creating ratios, and the aluminum gears tend to grind away very quickly, especially if they are not supported correctly on both sides. As a young team, you may be considering TETRIX as a starter kit, but we encourage you to explore your options and consider your club limitations before selecting a kit. While Tetrix does pick up quite a lot of flak from the community, it isn't a bad choice in terms of physical qualities such as strength, but the mounting options and hole patterns leave more to be desired and restrict teams in terms of flexibility.

Kit Parts

https://asset.pitsco.com/sharedimages/resources/ftcset_productlist-0719.pdf

(Note: this is for the 2018-2019 season) * 32x32mm Aluminum C-Channel, various lengths * Assorted plates and brackets, flat and 90 degree * Flat plates * Axles (6mm d-shaft & 4.7mm d-shaft), hubs, spacers * Aluminum gears * Standoffs * HS-485 HB servo (4x), assorted accessories * TETRIX Torquenado motor (4x) * Bars and angles * omni and traction wheels * Nuts, screws, fasteners * Battery * Tools (screwdrivers, etc.)

Advantages

- The TETRIX kit, being the most basic of all kit options, is easy to learn and provides a variety of options in building. The kit itself comes with c-channel, which is aluminum shaped in a C. It has pre-cut holes so motors, gears, or drive shafts can be seamlessly integrated into the channel. For a beginner team with little to no experience, a TETRIX kit allows you to assemble a working drivetrain in a couple of hours (Note: Most other kits allow you to do the same thing, but with more customization options).
- Furthermore, TETRIX is decent in terms of structural integrity, as long as the channels are loaded in the correct orientation. Typically, the orientation should be like an "n", with the top face upward. Connecting the channels with locknut instead of regular nuts aids in longevity. However, TETRIX is the weakest of the build systems and is very prone to bending, especially with long pieces of channel that have multiple axes of load.
- It is simple to build basic mechanisms such as an arm using the gears and d-shaft. However, as will be explained later, there are many potential drawbacks to doing so.

Disadvantages

- The TorqueNADO motors are comparable with NeveRest 60:1. They are usable, however have more limited uses due to their slow gear ratio, which means high torque but relatively slow output speed. The TorqueNADO uses spur gearboxes, and will not handle as much shock load as other planetary options, namely AndyMark 20 Orbitals, REV HD Planetary, or the goBILDA Yellowjacket Planetaries.
- The aluminum gears seem to grind against each other often, even with correct spacing, and do not last very long, especially under high torque situations.
- TETRIX hubs are 6mm or 4.7mm round and are based on set screws which are torque transferring. These set screws are notorious for coming loose under load, so special care must be taken to continuously tighten these screws. Additionally, it is highly recommended that teams use some sort of threadlocker on set screws, whether it be Loctite Blue (removable) or Loctite Red (nonremovable). A

possible workaround is purchasing 6mm D clamping hubs from goBILDA to use on TETRIX shafts, however, this also necessitates the use of a pattern adapter from the goBILDA pattern to TETRIX.

- Due to using a metric based pattern but using imperial based holes and chain standards, chain will not have perfect tension when running from one hole in channel to another. An external tensioner will be required to implement chain systems.
- TETRIX is also the most expensive kit on average while providing the most limited build options. Finally, the tight spacing and mix of units limits teams if they would want to integrate custom parts with TETRIX.
- Set screws on TETRIX hubs can mar motor shafts, and the bore of the hub itself can enlarge, leading to a wobbly or misaligned hub.

Tips for use

In any build system, it is important to properly support the structure of your robot. This includes supporting axles at two or more points of contact, or having multiple points of support for a piece of channel. Due to TETRIX's relatively fragile nature, this advice goes double with TETRIX. Instead of just supporting a shaft at two points, it is recommended to support the shaft at 4 or more (if you are using a 4.7mm Standard). When mounting motors, it is recommended that two clamping motor mounts are used to mount one motor. Using standoffs inside channel to prevent the channel from bending inward or outward is also highly recommended.

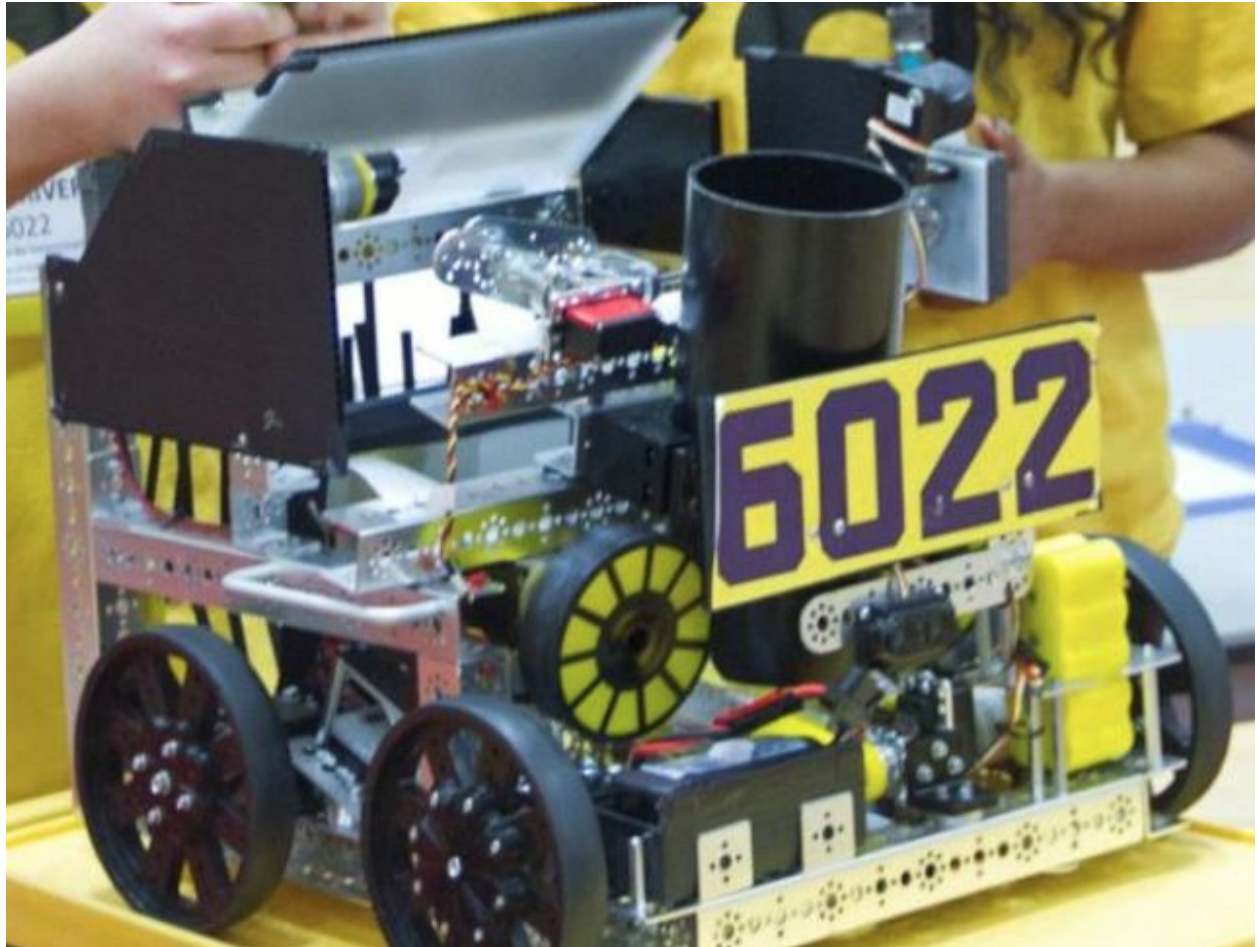


Fig. 1: TETRIX based robots have succeeded in competition, albeit it has become rarer in recent years.
6022 To Be Determined: Worlds Semifinalist (St. Louis), Velocity Vortex

Actobotics

The Actobotics kit from ServoCity is another fixed-pitch, channel-based building system. Its 1.5" c-channel and ball-bearing based motion system allows teams to iterate and build mechanisms entirely from kit parts. One advantage of Actobotics is that their channel hole pattern has many more mounting holes than a Tetrax channel, so it is less restrictive in terms of mounting. Actobotics uses imperial units across the board, allowing for clean spacing and fitting. A variety of mounts and brackets make strong, compact joints between channels. Actobotics also sells mini-channel in addition to the standard size channel. In addition to the channel, the main structural component in Actobotics kits, the X-rail extrusion system offers immense adjustability. To complement the structural offerings, Actobotics uses a series of clamping hubs and a robust 1/4" steel D-shaft. Finally, Actobotics has a Linear Motion Kit which uses X-rail extrusion to build extending mechanisms. The ServoBlock, which prolongs the life of a servo, is recommended for all teams, regardless of kit; however, ServoBlocks seamlessly interface with the Actobotics ecosystem. Actobotics furthermore offers options such as bevel gears and Linear Actuator kits (using lead screws) for specific use cases. Actobotics can interface with other kits such as REV through a variety of Pattern Adapters. **ServoCity offers a 25% off discount for all FIRST teams, making pricing competitive. Actobotics is a solid choice for new teams, offering a reliable base kit with many options to expand upon.**

Kit Parts

<https://www.servocity.com/ftc-competition-kit>

- Aluminum c-channel and mini channel, various lengths
- X-rail (a special type of extruded rail similar to 8020)
- Assorted plates and brackets, flat and 90 degree
- Flat plates
- Clamping hubs, spacers
- Gears and sprockets
- 1/4" d-shaft
- Standoffs
- HS-488 HB servo (4x), assorted accessories
- 5201 Series, 53:1 Ratio, 105 RPM Spur Gear Motor
- 5201 Series, 26.9:1 Ratio, 210 RPM Spur Gear Motor
- Bars and angles
- Omni and traction wheels
- Nuts, screws, fasteners
- Battery
- Tools (screwdrivers, etc.)

Advantages

- Actobotics provides a great value fixed pitch build system that is generally easier to assemble than extrusion-based systems which require cutting extrusion to length.

- Actobotics has the patented ServoBlock, which helps drastically increase servo life by protecting it from shock loads. More information may be found in the glossary.
- Actobotics is easily compatible with other build systems such as REV's kit using adapters which can be found on the Servocity website.
- Actobotics is more sturdy than TETRIX in terms of drivetrain flex and has more support options to prevent structural bending.
- The clamping hubs offered are more reliable than set screws, as clamping engages the shaft in more places than a set screw (one face). As discussed in the [TETRIX section](#) (page 15), set screws in particular are vulnerable to loosening, especially without application of Loctite.
- Actobotics' motion system is **very** robust and relies on ball bearings, which have a lower coefficient of friction than bushings. Actobotics is compatible with the 5mm hex used by REV with their adaptable hubs.
- Actobotics also allows for face mounting of motors as opposed to clamp mounting. This is explained in the glossary section.

Disadvantages

- Actobotics is not very cheap, so its cost may be prohibitive for some teams with a low budget. Note that with the 25% off FTC team discount, Actobotics can be cheaper than TETRIX. The only cheaper build system is REV.
- The belts from ServoCity are not great. It is recommended that belts are purchased from vendors other than ServoCity. This may necessitate 3D printed belt pulleys and other parts, which is more difficult for newer teams. There also isn't a large amount of space inside the channel for a belt or chain run, vastly limiting the size of the pulleys.
- Channel takes up more space than extrusion, so mechanisms can be a bit larger with the Actobotics kit. To remediate this issue, ServoCity sells mini-channel which is a similar size as extrusion.

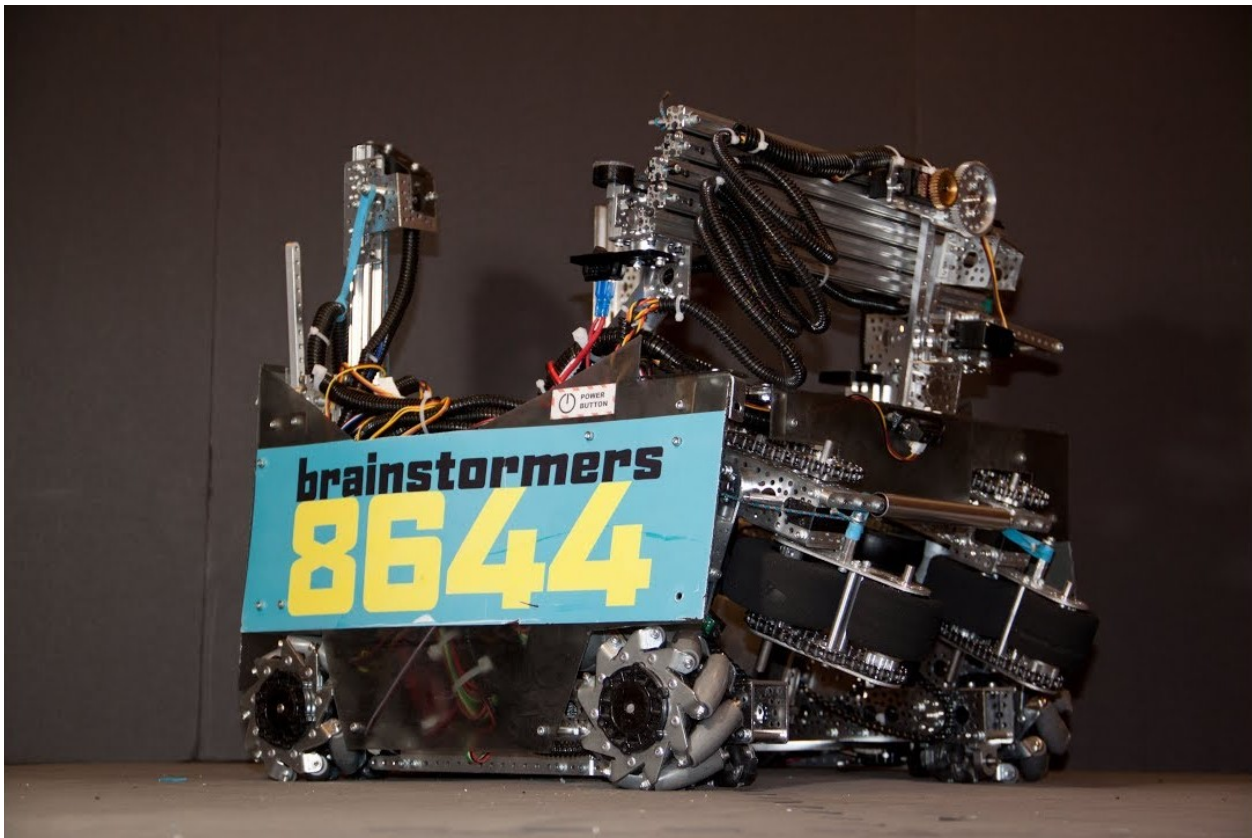


Fig. 2: An example of a successful Actobotics based robot, 8644 Brainstormers: 2018 Winning Alliance Captain (Detroit), Relic Recovery

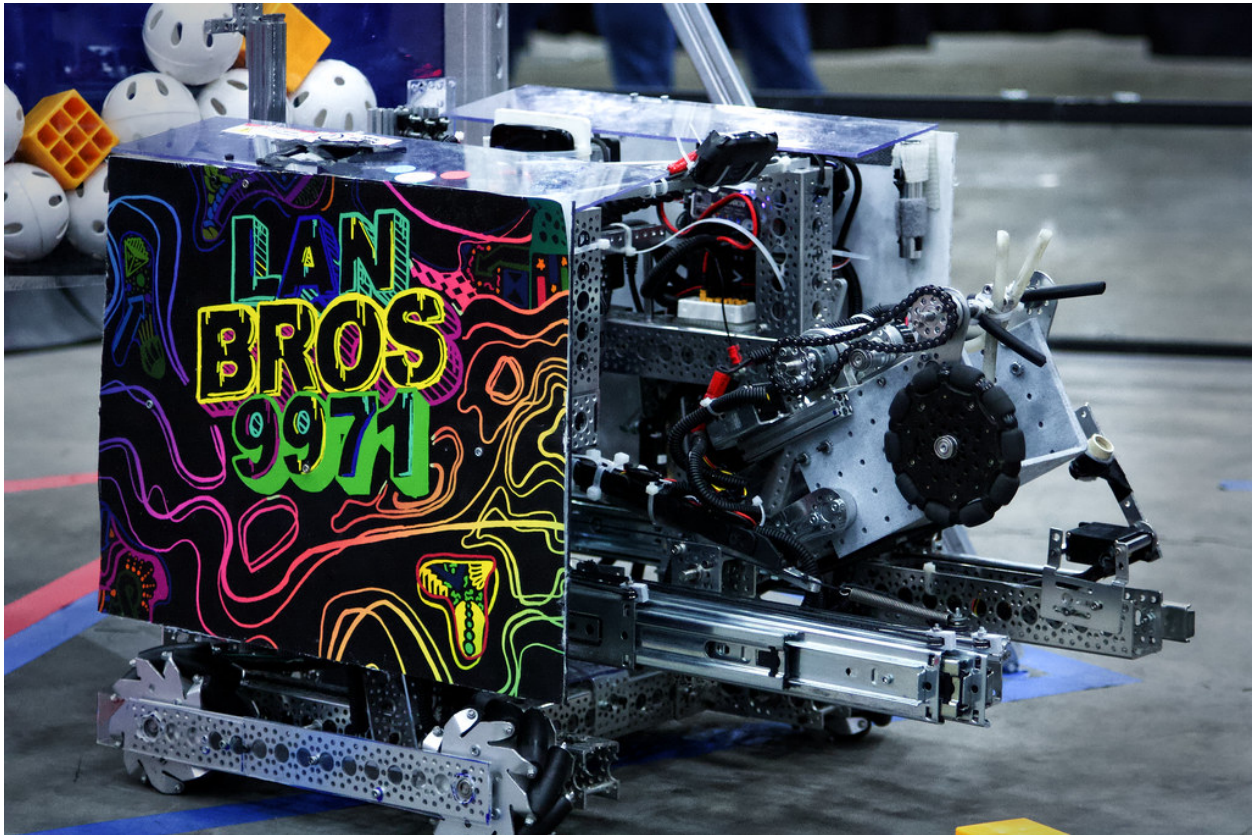


Fig. 3: Another successful Actobotics based robot, 9971 LanBros: 2019 Winning Alliance Captain (Detroit), Rover Ruckus

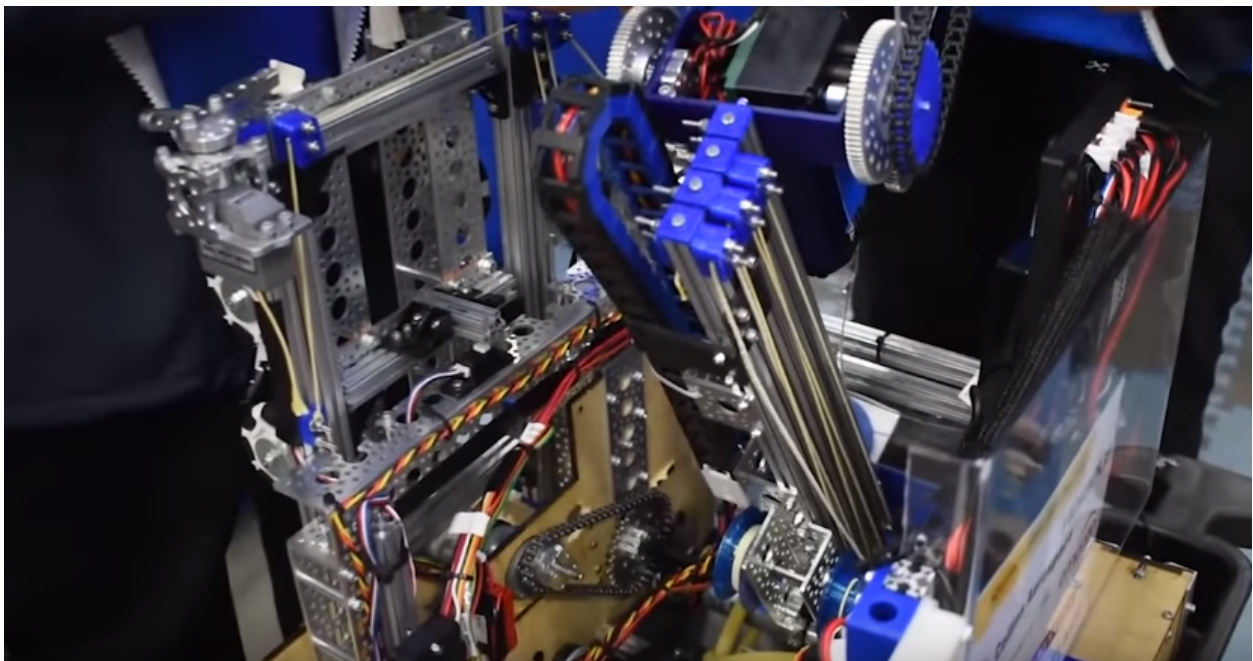


Fig. 4: 9794 Wizards.exe used both Actobotics and REV extrusion to build their Rover Ruckus robot

goBILDA

goBILDA is a fixed-pitch, channel based building system that was fully released in the 2018-2019 season. The goBILDA build system revolves around 48mm aluminum U-channels and the pattern features a unique 8mm grid with 16mm square clocked slots. It uses a metric hole pattern, M4 screws, 8mm pitch chain, HTD 3 belt, and metric shafts (6mm D). goBILDA's robust motion system, which is all metric, allows teams to easily build drivetrains and mechanisms without worrying about placement and tensioning. Much like Actobotics, it is channel based with an additional extrusion system for flexibility and adjustability. Unlike Actobotics, however, low-side channel (smaller profile channel) is the foundational building material to supplement full size U-channel. Even though this may initially seem like a stepback, the use of low side channel opens up many new possibilities such as a kit-based parallel plate drivetrain. Additionally, goBILDA is still being developed at a quick pace and is responsive to the needs of the FTC community. **goBILDA offers a 25% discount for FIRST teams.**

Kit of Parts

<https://www.gobilda.com/master-ftc-kit-2019-2020-season/>

- 5202 Series, 26.9:1 Ratio, 223 RPM Planetary Gear Motor w/Encoder **x2**
- 5202 Series, 13.7:1 Ratio, 435 RPM Planetary Gear Motor w/Encoder **x2**
- 2000 Series Dual Mode Servo (25-1) **x4**
- Servo mount plates, attachments, accessories
- Matrix 12V 3000mAh battery with XT30 to Tamiya Adaptor
- U-channel, assorted lengths
- goRAIL extrusion, assorted lengths
- Pattern and grid plates
- Square beams, u-beams, l-beams
- Assorted brackets, standoffs, spacers
- Gears, sprockets, chain (8mm plastic chain)
- Bearings, hubs, collars
- Shafting (6mm D-shaft)

Advantages

- goBILDA's low sized U-channel opens up many new possibilities due to its flexibility and compactness. For example, one can now make a parallel plate drivetrain without custom machining, or make custom width U-channel.
- goBILDA also has ServoBlocks (identical to Actobotics other than hole pattern) which drastically increase servo life.
- goBILDA has native large bore hex shaft support (12mm REX is comparable to $\frac{3}{8}$ Thunderhex) which is one of the main advantages of custom fabrication brought to a kit based system.
- goBILDA also is able to interface with TETRIX channel because they share some holes. For more information about TETRIX-goBILDA compatibility, check out this [website](https://gobildatetrix.blogspot.com/)³.

³ <https://gobildatetrix.blogspot.com/>

- goBILDA has a well thought-out ball bearing based motion system with smart motion transfer. It is easy to do chain or belt in channel. 8mm chain and HTD 3 belt can do perfect C2C (center to center) on the goBILDA pattern.
- goBILDA has plenty of shaft, bearing and pillow block options.
- goBILDA motors can face mount natively into channel, eliminating the need for motor mounts and providing a robust, reliable way to mount motors.
- goBILDA has some special parts are unavailable in other systems, like square beam shafts.

Disadvantages

- goBILDA is not the cheapest build system; it is relatively equivalent in pricing to Actobotics, but REV is cheaper.
- To achieve perfect center-to-center (C2C), goBILDA chose to adopt the HTD-3 belt standard. While this is a good choice, note that under extremely high torque loads (like an arm) the belt may skip. An easy way to remedy this would be to 3D print pulleys and use the HTD-5 belt standard, which will have perfect C2C on goBILDA channel every 5 bearing holes.
- Because of metric pattern spacing, goBILDA utilizes 8mm pitch chain, as opposed to the FTC standard #25 Imperial chain. This means that other kits' chain and sprockets won't work with goBILDA chain and sprockets.



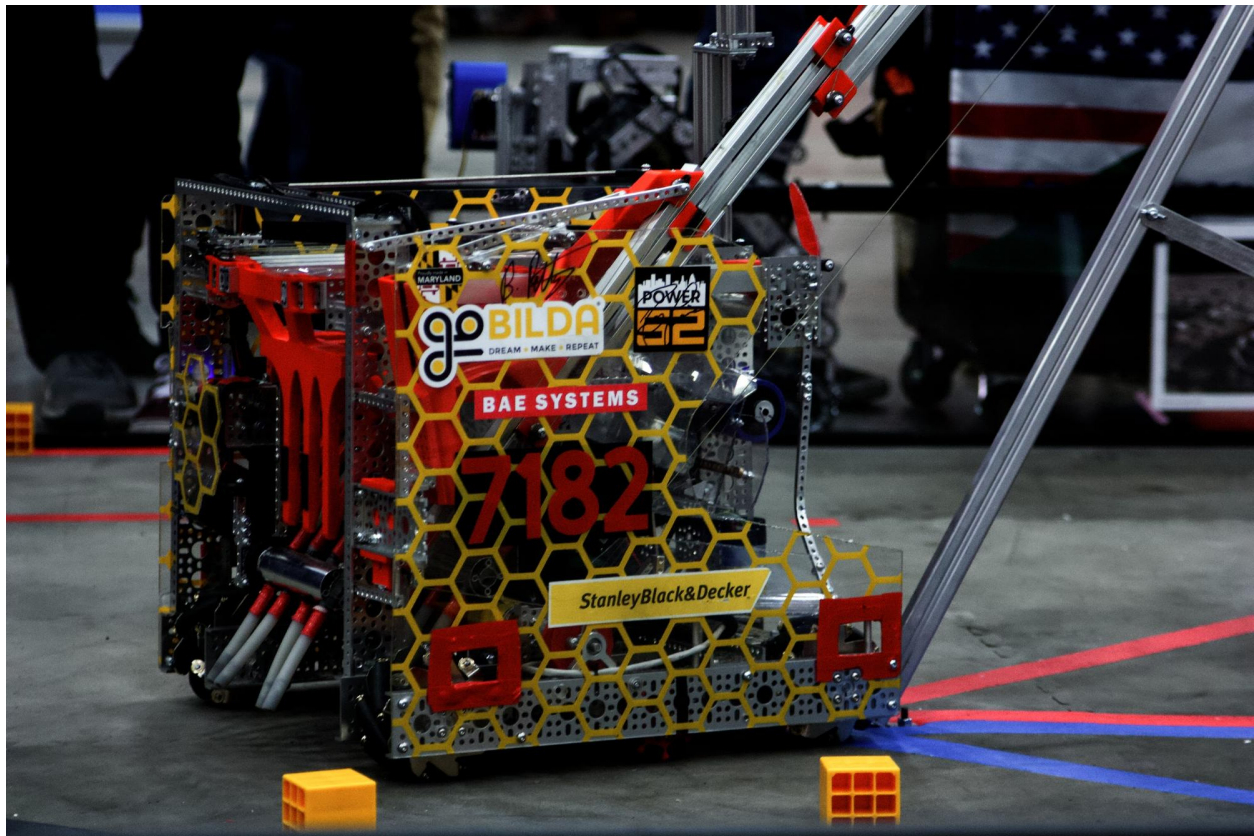


Fig. 5: An example of a successful goBILDA based robot, 7182 Mechanical Paradox Cubed: Finalist Alliance 1st Pick (Detroit), Rover Ruckus

Rev Robotics

The REV kit is an infinite-pitch, extrusion based building system which uses 15x15mm aluminum extrusion and is complemented by channels. With extrusion, there are no fixed mounting holes, increasing adjustability and flexibility. For example, tensioning chain is simple when sliding the mount or bracket increases tension. Extrusion systems are one step towards a custom bot, as it needs to be cut to length using a saw or bandsaw. In addition, REV also offers both 45x45mm U-channel and 15x45mm C-channel, which is stronger than extrusion and provides stiffness and support when needed. The REV system uses metric dimensions (15 mm extrusion, M3 hardware), with the exception of #25 roller chain. REV also uses 5 mm hex steel shafts for motion, which some vendors have added compatibility for in recent months. REV also has pattern adapters for other systems built into many parts such as the aluminum brackets. Another advantage to REV is the delrin (a kind of low-friction plastic) components reduce the kit cost, but many products have stronger aluminum options as well. While plastic typically is suitable for a new team, the aluminum parts are recommended if your budget allows for it. Furthermore, REV has released new products such as channel mentioned above, the UltraPlanetary gearbox which has customizable planetary ring gear ratios, as well as UltraSlide, a low-profile aluminum slide. **In general, REV has a steeper learning curve than other build systems**, owing to the fact that mounting points in mechanisms need to be accounted for. However, this is not necessarily a disadvantage. **The REV kit is good for teams willing to invest the time into an extrusion building system. It has the ability to integrate channel along with extrusion, as well as the option to upgrade parts** for those wanting a further investment into the REV building system.

Another advantage to the REV kit is the compatibility of 15x15mm Misumi extrusion. REV extrusion isn't as structurally strong as Misumi for two reasons. REV extrusion is 6063 aluminum, while MiSUMI extrusion is made out of A6N01SS-T5 aluminum, a stronger and stiffer alloy. Additionally, the MiSUMI extrusion has a larger surface area, so there is more area of contact. Misumi offers greater strength at a lower bulk cost. Additionally, Misumi will cut to the half millimeter for free, making it a great option for teams needing an exact cut. The drawback to Misumi is that it is quite a bit heavier than the REV extrusion. It is encouraged that teams use Misumi for drivetrain and structural support, and REV for mechanisms that will be under low to medium load. REV also sells punch tubing, which is 15 mm aluminum square tubing that is compatible with $\frac{1}{8}$ " rivets. This will prevent bolts from loosening or sliding around, as punch tubing requires teams to drill through the tubing in order to fasten parts to it.



Fig. 6: REV Extrusion

Kit of Parts

<http://www.revrobotics.com/rev-45-1270/>

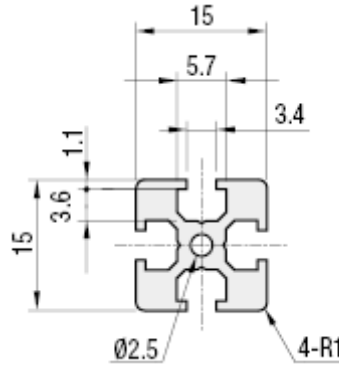


Fig. 7: MiSUMI Extrusion Profile

- 2 40:1 HD Hex Motors
- 2 Core Hex Motors
- 4 Smart Robot Servos with other attachments
- Smart Robot Servo Programmer
- 15 mm aluminum extrusion, various lengths
- Omni and traction wheels
- 32 gears in 7 different sizes
- 10 sprockets in 3 different sizes
- Delrin brackets
- Delrin bearings and pillow blocks
- Motor and servo power/data cables
- Slim Robot Battery and charger
- 5.5 mm nut driver and 5.5 mm hex wrench

Advantages

- Extrusion systems don't need to worry about tensioning as mounts are adjustable to your needs.
- Extrusion allows teams to save space as opposed to channel, and is lighter than aluminum channel.
- Extrusion allows infinite positioning options instead of being locked in to a specific distance - useful for fine-tuning a mechanism.
- Delrin products are inexpensive yet durable for most use cases.
- REV has the option to upgrade to aluminum parts if need be - something that no other build system offers. (not for all parts)
- 5 mm hex is a robust shaft and motion system and is easily adaptable to Ultrahex ½" hex shaft. Other companies have adapting options with 5 mm hex.
- Punch tubing is a great final iteration option if you are sure about placement.
- Channel complements extrusion extremely well - having the adjustability of extrusion and the rigidity of channel makes it a very solid build system

Disadvantages

- Steeper learning curve, more time consuming to prepare extrusion.
- Requires tools such as a saw and bandsaw
- Requires forethought and planning of extrusion length and placement
- Parts loosen over time (to remediate: use punch tubing)
- M3 bolts, especially those sold directly by REV, are prone to bending under higher load (such as when used as an axle for a pulley)
- 5 mm hex shaft is also prone to bending, especially if the shaft is long.

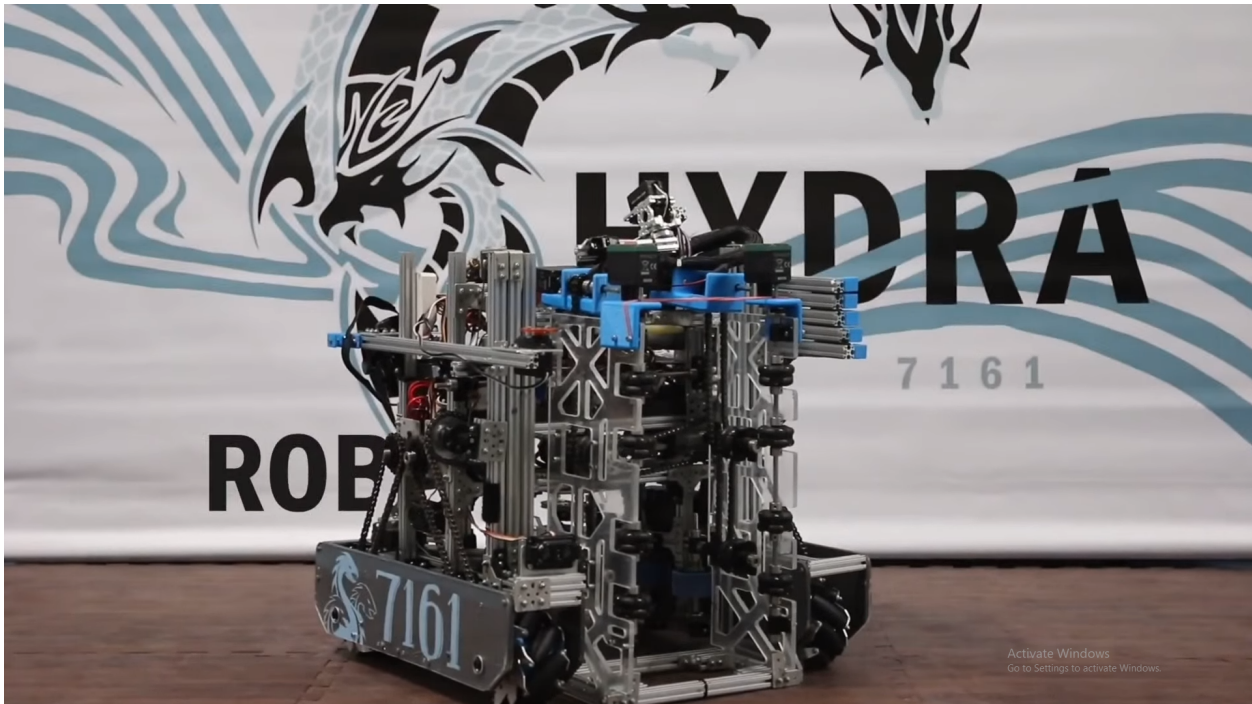


Fig. 8: 7161 ViperBots Hydra, Finalist Alliance 1st Pick (Houston), Relic Recovery

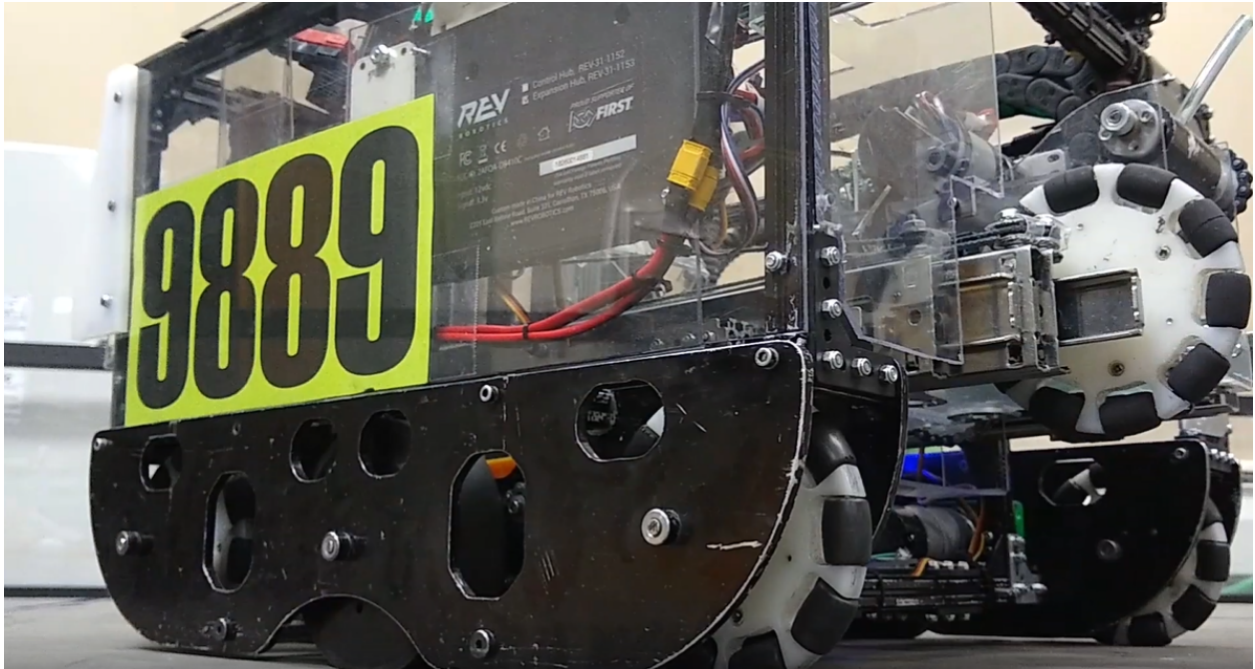


Fig. 9: 9889 Cruise Control, Rover Ruckus



Fig. 10: 6299 ViperBots QuadX, Velocity Vortex



Fig. 11: 11115 Gluten Free used both REV and Actobotics for the 2017-2018 season. Finalist Alliance Captain (Detroit), Relic Recovery

Custom

Many teams elect to ignore kits entirely and instead make their own mechanisms completely from scratch. While this **allows for much more design freedom**, keep in mind it **has many caveats and drawbacks**, especially for a new team. Additionally, a lot of teams choose to create a robot that is a mix of kit parts and custom parts.

In discussing custom options for teams, the two primary categories are as follows:

3D printing

Advantages of 3D Printing

- **3D printing allows teams to have customizable strain relief on wires and connections.**
- 3D printing allows for customizable sizing and perfect optimization; for example, teams can print a spool of the exact diameter needed for optimal speed.
- 3D printing allows teams to adapt between kits and individual parts easily, as not all kits have adaptable mounts or brackets. A good example of this are the Nexus mecanum bore adapters that teams 3D print.
- 3D printing allows teams to fabricate parts that would otherwise be impossible with materials such as aluminum due to machining restrictions.

Disadvantages of 3D Printing

- **If you are out of 3D printed spares at a competition, you're probably out of luck. Teams are advised to print at least one set of every single 3D printed part as spares for competition.**
- 3D printed components are generally weaker than other materials such as aluminum. However, printing in the proper orientation can be very strong - teams have 3D printed hooks and other parts to support their FRC robot (120 pounds) and FTC robots (40 pounds).
- 3D printed parts should only be loaded in one orientation. That is, if the robot is hanging from a hook, the only load should be on the bottom face of the curved part of the hook. Try to eliminate side loads as much as possible to avoid part failure.
- The size of 3D printed parts are limited by the size of your print bed.
- Large and thick prints can take a long time (overnight) to print and can run the risk of failure.
- 3D printing can end up quite expensive, though filament can be found for a reasonable price on online vendors such as Amazon.

Common Filament Types

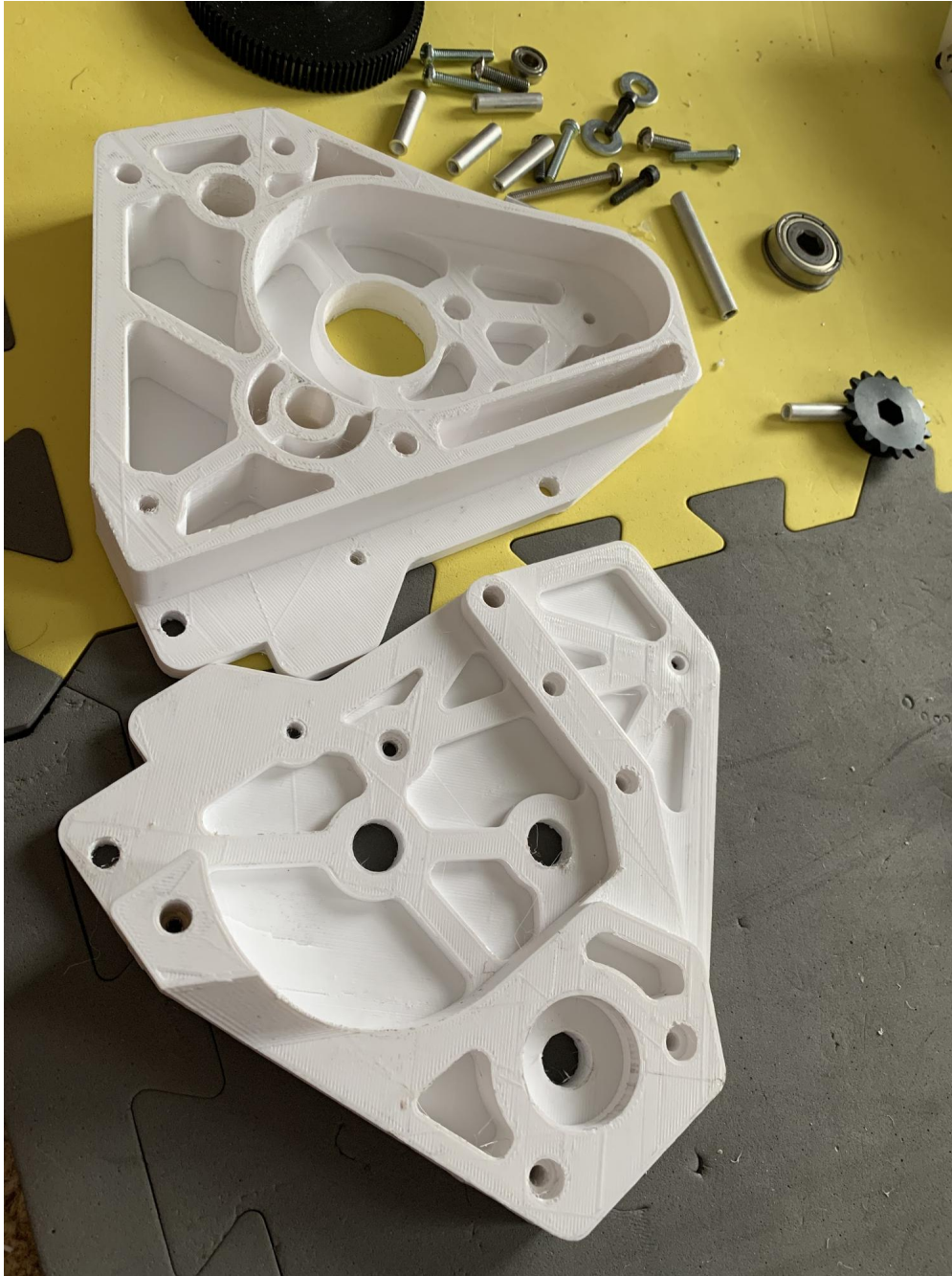
- PLA (polylactic acid) is a common filament that does not require a heated bed. PLA is stiff but more brittle than other filament options, and tends to have little to no warp when printing. PLA should be used for low or no load applications such as electronics retention mounting or cable management.
- ABS (acrylonitrile butadiene styrene) is another very common filament that is able to withstand higher temperatures than PLA. It is less brittle and more ductile than PLA, making it better for 3D printed parts that may be under load. Typically, ABS is used in higher-load applications such as 3D printed bores for mecanum wheels.

- PET/PETG (polyethylene terephthalate) can be described as a mix of PLA and ABS. It has little warp and is as durable as ABS. PETG is another great option for higher-load applications such as wheel bores.
- TPU (thermoplastic polyurethane) is a filament different from other options. Specifically, parts made with TPU can be flexible if the infill is set low, or by using flexible TPU filament. Teams use TPU to print customized intake flaps optimized for the right stiffness in order to make the intake more efficient. TPU also has very high layer bonding strength compared to other filament types, making it suitable for high load applications that do not need to be stiff. Must print TPU on a printer with a direct extruder, will not work on bowden extruders. A bowden extruder is mounted on the frame of the 3D printer and is connected to the nozzle with a flexible tube that guides the filament. A direct extruder will be mounted directly on the nozzle assembly, with filament feeding directly into the nozzle.

3D Printing Design Guide

The first consideration when designing 3D printed parts is print orientation. This refers to the side that contacts the print bed. Preferably, the part should have a flat bottom to maximize contact with the print bed. Maximizing contact with the print bed will make sure the part doesn't delaminate or warp from the bed and increase print quality.

If it is impossible for the part to have a flat side to print on, a simple solution is to split the part into multiple parts along a plane. For instance, the gearbox plate below didn't have a flat side to print on, so it was split in half. The part was later sandwiched with numerous screws and plastic glue. If this plate was printed as one part instead of split in half, support would have to be used to create all of the necessary holes. Parts that use no support material make sure that the least amount of plastic is wasted.



Another tip with the bottom side of a printed part is to not chamfer or round anything on the perimeter of the first layer. Chamfering or rounding will increase the chances of the part warping especially on unheated print plates.

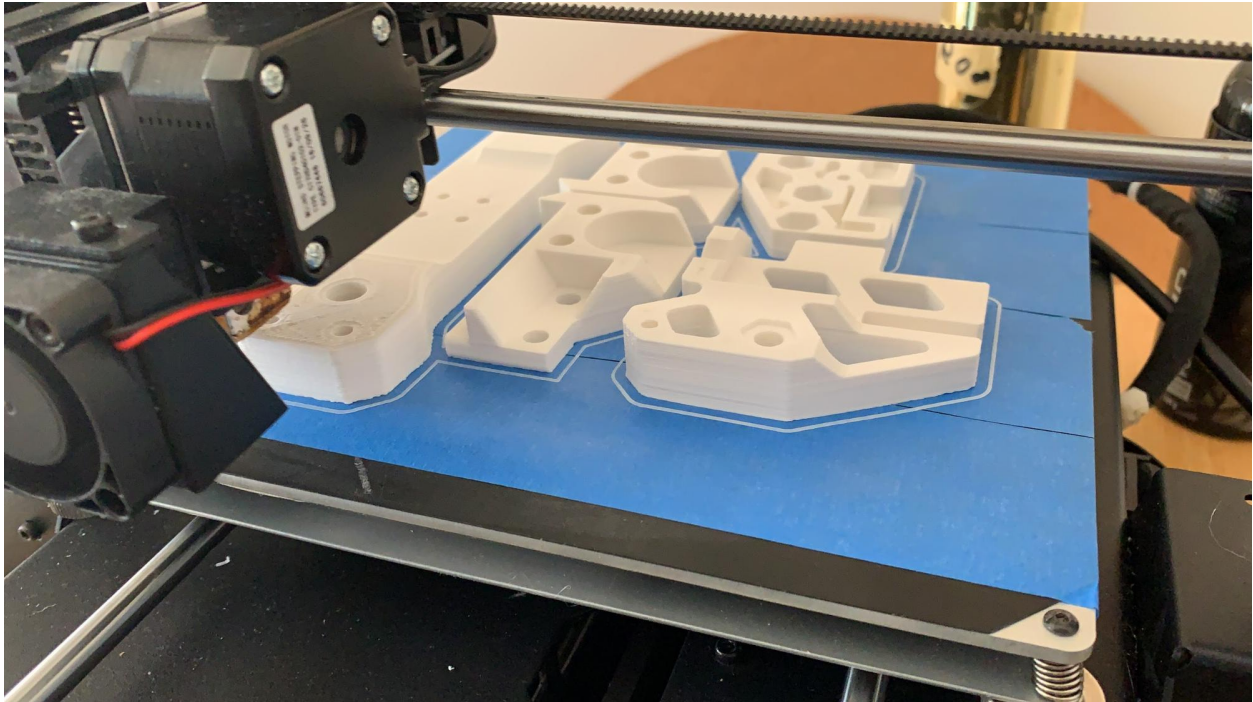
Draft Angle

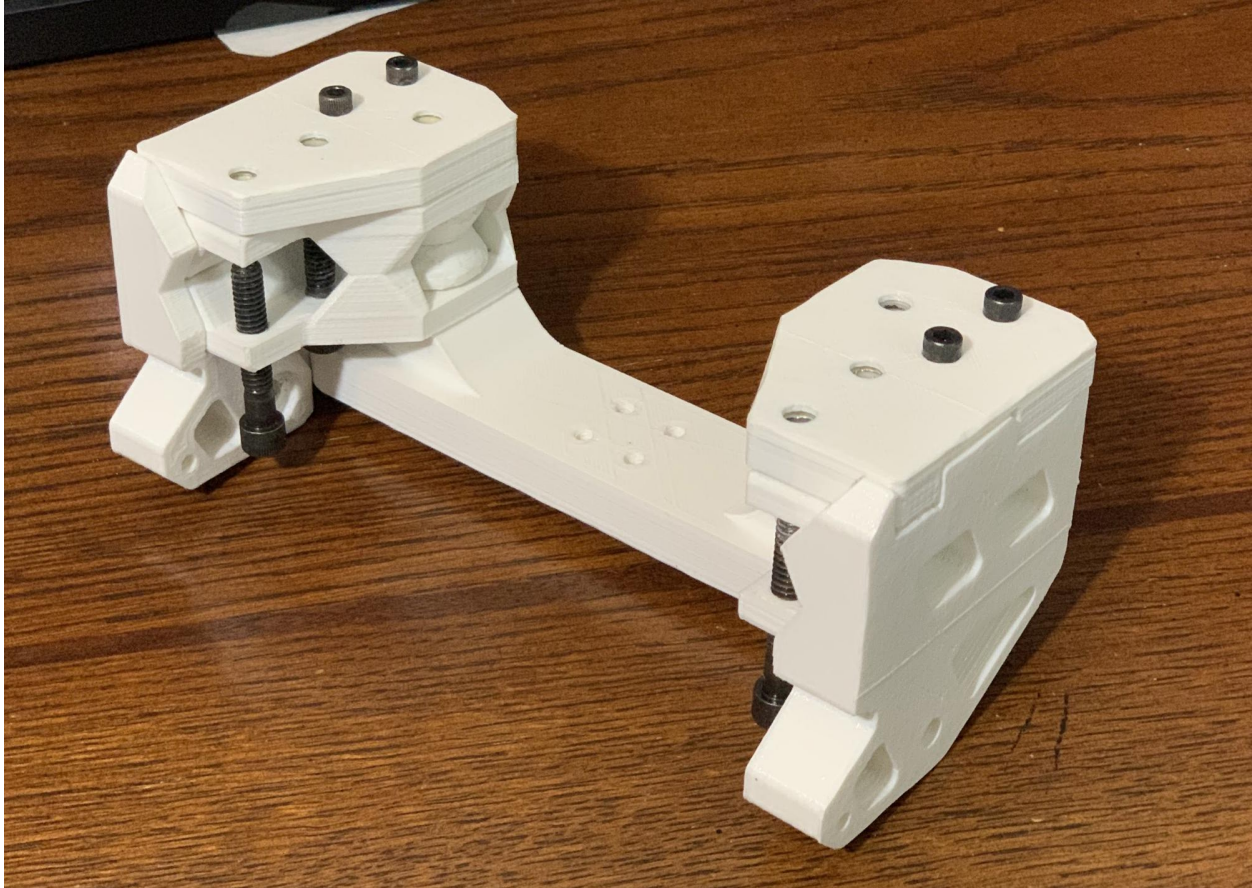
Draft angle refers to the overhang angle between the part side and normal vector from the print plate. The maximum draft angle refers to the maximum angle the printer can print without support material and is based on the printer, print settings (notably speed, temperature and cooling) and filament type. When trying to decrease support material, consider every overhang angle and make sure it is within the maximum draft

angle. Staying within the maximum draft angle will also decrease the chances of part warping.

Stress Vectors

Perhaps the most important consideration are stress vectors. 3D printed parts are inherently stronger on two axes and weaker on one axis. The weaker axis comes from the layering action that defines FDM 3D printing. A common fix to this is to simply increase the print temperature up to a certain limit until it starts decreasing print quality, but the weaker axis can be resolved by again splitting up into multiple parts. The point to get across is to try to increase strength by optimizing sections of the part on the plane they are being printed on. For instance, this assembly below was responsible for hanging the entire robot, so it had to be the maximum strength possible for a 3D printed part. It might seem counterintuitive to split up a part into multiple parts to increase strength, but there is a logic behind it. The part could have been easily printed as a single part, but it would be fairly weak when stresses are exerted in the upward direction. Splitting the part and creating new flat surfaces to print on will strengthen each axis. If one small part failed, the robot might still be able to somewhat function. This would be preferable to the entire piece failing at once. In this example the side pieces were printed as separate parts on a complementary axis to strengthen the assembly.





This assembly is a good example of considering part orientation, draft angles, and stress vectors in each part of the design. Complex parts can be made strong and without any support by simply splitting it up in the right way.

Machining

Advantages of Machining

- Machining allows teams to create practically any part without limitations.
- Machining allows teams to create custom drivetrain plates, arms, linear slides, etc to fit their design needs more effectively than kits.

Disadvantages of Machining

- Perhaps the biggest barrier when making a fully custom robot is that it requires very expensive manufacturing equipment. While 3D printers have become more and more common in FTC and allow teams to create or customize small parts, many teams do not have access to equipment such as lathes, routers, waterjet cutters, or CNC machines.
- Fully custom robots need to be fully designed in CAD. Sketching will not cut it when building custom robots.

- Another barrier is that prototyping with fully custom systems is almost always much slower than using kit parts. This is because the full CAD process takes time, and fabrication/getting parts fabricated by a shop or sponsor is another lengthy process. Notwithstanding those two factors, if something goes drastically wrong then it will require a repeat process of CAD and fabrication.

It must be noted that teams do not need to pick between a fully kit-based robot or a fully custom robot. In fact, many successful teams are actually a hybrid of both kit and custom parts, as can be seen in many of example robots in this section.

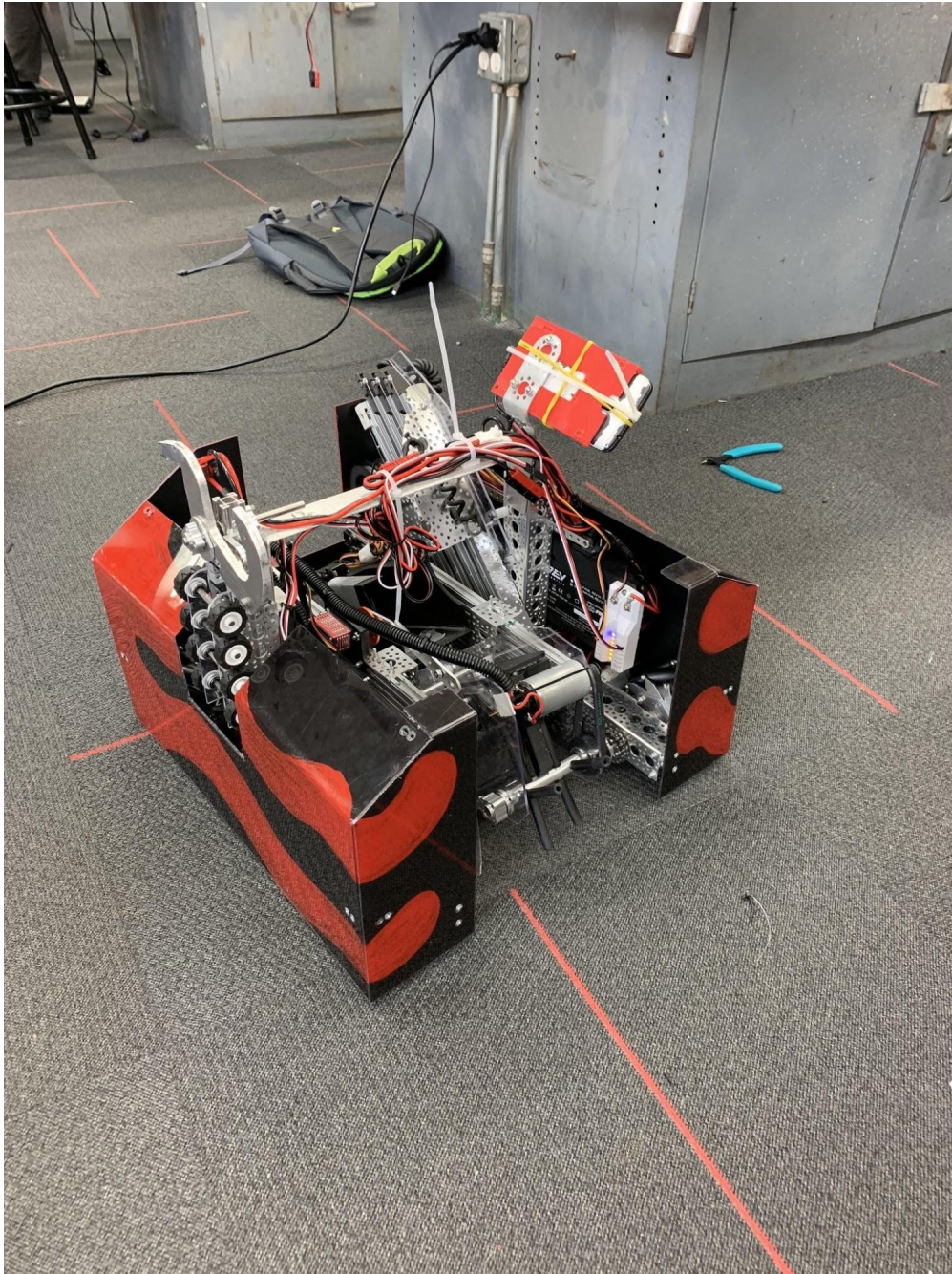


Fig. 12: 8680 Kraken Pinion, Division Semifinalist (Detroit Worlds), Rover Ruckus Kit-based drivetrain (Actobotics) and horizontal linear slides (REV) with a custom intake, housing, and mineral container. Kit linear actuator (Actobotics) with custom hook.



Fig. 13: 8393 Giant Diencephalic BrainSTEM Robotics Team
Winning Alliance Captain (Detroit), Relic Recovery, fully custom



Fig. 14: 731 Wannabee Strange, Rover Ruckus, Design Award Finalist (Houston)
3D printed (PETG and TPU) and machined (milled) arm/intake
Custom HDPE drive train (CNC routed)

1.5 Tools List

Here are a selection of necessary and helpful tools for your team. You'll obviously need to purchase the basics such as the right screwdriver, hex keys, and drill bits. For a bit more precision and the ability to dabble in custom, opt for a drill press and bandsaw. A router might be useful for teams interested in working with wood, a great material for prototyping (and also for final iterations!)

1.5.1 Necessary

- **Safety glasses. Wear them when you're using power tools. Seriously.**
- **Screwdrivers**
 - Philips, assorted sizes
 - 3/32", ball head for set screws is discouraged (TETRIX)
 - 5.5 mm hex nut driver (REV)
 - 2.5/3mm hex (goBILDA)
 - 7/64" hex (Actobotics)
- Drill and drill bits
- **Hex keys (metric and US standard)**
 - 1.5mm hex key (REV)
 - 2.5/3mm hex key (goBILDA)
 - 7/64" hex key (Actobotics)
- Pliers, needle-nose and locking
- Metal file (sandpaper not recommended)
- Quick lock clamps (2+) or vise
- Hammer and mallet
- Centerpunch
- Wrench/assorted spanners
- Hacksaw (cuts through steel shafts)
- Wire stripper/wire cutter
- Zip ties/Velcro ties
- Electrical tape

1.5.2 Helpful

- Bandsaw (**cannot cut through steel shafts**)
- Impact driver
- Drill press
- Chop saw

- Dremel (use sparingly)
- Grip tape
- Caliper
- Soldering iron
- Heat gun
- Router or table saw
- Jigsaw
- Metal brake

1.6 Materials Guide

In FTC, teams aren't limited in terms of what raw materials to use. However, there are definitely some important materials which you should be aware of. Here are the materials to use, listed in order of importance.

1.6.1 Recommended Materials

- Aluminum channel & angle
- Aluminum extrusion (15mm)
- Polycarbonate (Lexan™)
- Baltic birch
- Delrin
- HDPE

1.6.2 Conditionally Recommended

- Aluminum sheet
- 80/20 extrusion
- Plywood
- Polyvinyl chloride (PVC)

1.6.3 Not Recommended

- Medium-density fiberboard (MDF)
- Steel
- Acrylic

Aluminum

- A high strength, medium-high density material. Suitable for use in nearly every application; recommended in load-bearing applications.
- Aluminum channel is used in many build system kits and is very popular among teams of various skill levels. Channel is strong yet relatively lightweight, and offers many mounting options for teams.
- 15 mm extrusion is compatible with M3 hardware, allowing teams to slide in bolts to their desired location. REV extrusion is not as great structurally, but is lighter than Misumi extrusion, which is more resistant to flexing and/or twisting under load. Both REV and Misumi extrusion are sold in bulk quantities, and Misumi has the option to cut to the half millimeter. Keep in mind that a lot of extrusion can add up quickly in terms of weight.
- 8020 extrusion (1") is regularly used as the primary building system in FRC. It is definitely overkill for FTC purposes and should not be considered as the primary building component by teams. It may be useful in certain applications (for example, linear motion using v-groove bearings).
- Aluminum flat and aluminum angle is widely available at hardware stores. In certain applications, such as adapting from different build systems, it is possible to drill custom adapter plates to mount mechanisms to the drivetrain. Aluminum angle is also a very sturdy structural support piece that takes up relatively little space, and can adapt to any build system. We suggest using $\frac{1}{8}$ " aluminum with drivetrain or mounting applications, and $\frac{1}{16}$ " aluminum for low-load situations.
- Aluminum drawer slides, often with ball bearing sliders, are recommended over steel drawer slides primarily for weight purposes. Refer to the Linear Motion section for more information.
- Sheet aluminum is generally used for drivetrain plates on custom drivetrains. The recommended thickness is $\frac{1}{8}$ " or $\frac{3}{16}$ ". Because it is a plate, sheet aluminum will bend if not supported correctly with standoffs or channel. Only load the sheet in the plane that it is in (if the sheet is vertical, then only put vertical load on it; do not load it horizontally). However, there may be some applications that would benefit from a slight bit of flex for adjustability - in those cases, use your judgment and test it out for yourself!

Wood

- A medium to high strength material suitable for use in a wide variety of applications. Keep in mind that wood cannot be exposed to water or excessive humidity, as lumber for use in FTC isn't treated and may warp or expand. Do not try to use laser cut wood for a drivetrain. This is especially important if you live in a humid region such as Florida - wood expanding can completely ruin a custom drivetrain.
- Baltic Birch is the highest grade plywood, used in commercial applications and furniture. It contains 8+ layers (usually), is extremely dense, and is recommended for high-load applications or structure. It is quite an expensive material, so prototype and plan carefully before cutting.
- Plywood is not recommended for final iterations, but can be a cheap prototyping material. It doesn't bear load especially well and can flex quite a bit.
- MDF is generally discouraged as there are better options and absorbs water easily.

Steel

- Steel is unnecessarily heavy for FTC structure. Aluminum provides plenty of strength at a fraction of the weight, and doesn't require welding.
- The proper uses of steel in FTC are in shafts (most are made out of precision-ground stainless steel) and gearboxes.

- Steel drawer slides can be used, but aluminum slides are highly recommended.

Lexan

Polycarbonate, commonly known by its brand name Lexan, is a material that is great for applications such as drivetrain plates or intake collector boxes. It can bear load and is very impact-resistant. Many teams use it in the intake and deposit mechanisms as it is clear, allowing the drive team to see into the intake itself, an advantage over wood. Thick lexan can be used for drivetrain plates, though this is not recommended for inexperienced teams. Thin lexan can be bent with a metal brake or sheet bender. It is recommended for teams to use bends instead of connecting with bolts - bending tends to be much stronger than bolting as it means the part stays in one continuous piece. If a sheet bender is out of the question, it is possible to use a heat gun or camping burners to heat up the lexan in order to bend it. This is not recommended as it can cause injury as well as bubbling (which results from overheating the lexan). Alternatively, “cold bending”, bending along a straight edge without applying heat can work for thinner sheets. Lexan is one of the most expensive materials per square foot, so make sure you have carefully planned out what you are cutting before doing so.

Polyvinyl chloride (PVC)

- PVC is most commonly sold as pipe, though PVC sheet does exist. FTC teams generally use PVC in order to customize intake rollers, especially for surgical tubing intakes. Due to the soft plastic, it is very easy to drill through, so teams often attach the surgical tubing to a PVC pipe. The PVC roller will then be attached to the intake motor. Since PVC is cheap, teams can easily experiment with the number of tubes, angle, etc. to build an optimal intake for their use case.
- PVC is sometimes used in cable management as a channel to run wires through, as it comes in different diameters and can easily be mounted to nearly any surface.
- PVC sheet can be bought or made by cutting vertically along the pipe so that the profile looks like a C. Bake in oven. PVC sheet is pliable yet sturdy, making it a good option for backstops or customizing connectors that require some sort of flex.

Acrylic

Acrylic is not a load-bearing material. It will crack and possibly shatter under impact. Do not use it on drivetrains by any means! It should only be used in mechanisms where there is no chance of shock impact. It can be used for aesthetic purposes or as shielding (to protect game elements from falling into the robot or from other robots tangling with wires, etc.) Treat acrylic as fancy clear cardboard - it cannot be loaded and should only be used in very specific circumstances.

Delrin

- Delrin is a low-friction plastic used in many REV Robotics parts. It is a durable and strong plastic that is not easily cracked. Delrin can be used in drivetrain plates, but delrin sheet is also quite expensive.
- Delrin can be machined in a variety of ways including laser cutting, cnc routing and hand sculpting.

HDPE

- HDPE stands for High Density Polyethylene and usually comes in opaque white or black sheets and plates
- HDPE is a cheaper alternative with low-friction characteristics similar to delrin and flexibility/structure near polycarbonate.
- HDPE is easily machined by both hand tools and machine tools. However, it will burn in a laser cutter, so do not laser cut HDPE.

Carbon fiber

- Carbon fiber is one of the strongest materials that any FTC team could use. For most teams it is totally overkill, but it can be used in some specific applications. Carbon fiber rods are used in custom linear slide extensions or multi-axis arms. It is probably the most expensive material to purchase.
- **Machining carbon fiber, like any fibrous substance, is a significant SAFETY HAZARD! Carbon fiber dust especially can cause cancer and is incredibly dangerous. DO NOT MACHINE/CUT carbon fiber unless you know what you are doing. When you do, make sure to either use machinery that is designed to cut carbon fiber, or cut in a well ventilated area with sufficient respiratory protection and running water over the carbon fiber.**

Cardboard

Please do not use cardboard as a load-bearing material. We have seen too many teams use cardboard in ways that it shouldn't be used. Treat cardboard as a sheet of paper: it has no structural rigidity and only should be used as guides to channel pieces from A to B.

1.7 Drivetrains

This section will cover the heart of any robot, the drivetrain. The purpose of the drivetrain is to facilitate the movement of the robot, and thus is a mechanism crucial to the overall function of the robot. If the drivetrain doesn't work, the rest of the robot won't work either. There are many possible types of drivetrains in FTC, which we have covered in the guide. Drivetrains are split into two main types: tank (skid-steer) and holonomic.

1.7.1 What Are the Kinds of Drivetrains?

A tank drivetrain primarily utilizes traction wheels and cannot strafe (move sideways). To change directions, a tank drivetrain relies on either turning the wheels on the left and right side in the opposite direction (thus spinning the robot) or running one side faster than the other side (thus making the robot take the path of an arc). Tank drivetrains prioritize traction and acceleration over pure maneuverability, giving these drivetrains the potential to traverse obstacles and play defense. Tank drivetrains are relatively simple to build, yet are still competitive at the highest levels.

A holonomic drivetrain, in contrast to a tank drivetrain, can move sideways, due to using either mecanum or omni wheels. These kinds of wheels have special rollers that allow strafing movements. Thus, holonomic drivetrains prioritize movement over traction. Holonomic drivetrains eliminate the time it takes to turn for a tank drivetrain. However, holonomic is susceptible to defense and can suffer with a heavy robot. Holonomic

has been proven to be competitive at the highest level for many years, and is common among world-class robots.

1.7.2 How to Choose A Drivetrain

When building any mechanism, teams must list out some necessities or desired features. Here are some priorities for that we think are important for each drivetrain:

1. **Reliability:** The drivetrain must work every time, not break, and if there are flaws, these flaws must be reliably present so they can be corrected. One aspect of reliability to consider is the type of motor and gearbox that is used in the drivetrain. For example, spur gearboxes are much more likely to break under load than a planetary gearboxes. (See [Motor Guide](#) (page 159) for details). Thus, spur gear motors are not the optimal choice for drivetrain, especially if the robot is projected to be on the heavy side (30+ pounds). Reliability, the key to success in FIRST Tech Challenge, starts with the drivetrain, the foundation to any robot. Our advice for new teams is to stick to the simpler drivetrains such as 4 or 6 wheel drive and mecanum drive.
2. **Agility:** There are many factors to agility: top speed, acceleration, turning radius, turn speed, and ability to strafe. Note that turning radius is an often overlooked feature that is critical to the overall agility of the drivetrain. Generally, a solid drivetrain should have a maximum speed of around 2-3.5 feet/second, and be able to turn fully in around two seconds. A good resource to determine the agility of a drivetrain is the [JVN Calculator](#) (page 164) - refer to that section for more information.
3. **Number of motors and gear ratio:** Generally, new teams may try to use only two motors on the drivetrain. While this is possible, it is not recommended, as all competitive teams use four motors on the drivetrain. Another issue stemming from experience is that most teams' drivetrains are too slow. More advanced teams may focus on the ability to play defense, but in general, maneuverability and speed are the main factors to a successful drivetrain. 60:1 and 40:1 drivetrains are **almost always too slow for FTC use cases**. A popular setup in FTC is **4 motor, 20:1 using 4 inch wheels**. This ratio gives a great balance, having near instant acceleration and a high top speed. Teams can slow the drivetrain down in code by providing less power to the motors if needed. **It is not recommended for teams to use spur gearboxes on their drivetrain. Instead, use planetary gearboxes, as they are less prone to shock loads and breakage.**
4. **Traction/Pushing Power:** While this feature is often overemphasized, it is still very important. Pushing power describes drivetrain's ability to endure defense/engage in defense. Many factors affect the pushing power of a drivetrain, including wheel type, motor gearing, and overall weight of the robot. Note that if you already have a very agile drivetrain with experienced drivers, a team can avoid defense instead of having to fend it off or engage in it.
5. **Powering the drivetrain:** Generally, there are four options for power transmission: direct drive, chain, gears, and belt. More detail about each option can be found in the power transmission section. Teams should stay away from direct drive, as gearboxes are prone to breaking, especially under shock loads (for example, if the wheel is hit by another robot, or the wheel slams into the wall). We recommend belted drivetrains, but realize that belt is a difficult option for new teams. Chain and gears are also great options - chain requires a bit more forethought, as 1+ tensioner per side is required to maintain correct tension in the chain. CAD or a detailed sketch is generally recommended with chain in order to visualize the chain run (where the chain will be placed). Gears are slightly simpler, and can be a fantastic and easy option, especially when using extrusion. We would advise to stay away from TETRIX gears, and use the gears from other kits such as the REV delrin gears (with hex hub strengthener) or REV aluminum gears. See [Power Transmission](#) (page 81) for more details.

An important step is determining what you want out of your drivetrain.

- Do you want speed?
- Pushing power?

- Ability to get over terrain?
- Do you need to strafe?
- All of these questions need to be answered before choosing a drivetrain.

1.7.3 Drivetrain Options

Tank (Skid-Steer) Drivetrains

2 Wheel Drive (Pushbot Drive)

Recommended only for first drivetrain, not for competitions

This rookie drivetrain is considered one of the inferior drivetrains, though it is usable at low competition levels. This is the introductory drivetrain type for many rookies, as it is promoted in official guides published by FIRST (giving it the name pushbot). It often has direct driven traction wheels with unpowered omni wheels. This type of drivetrain has poor turning as the center of turning is at the back of the robot between the two powered wheels. In comparison to other drivetrains, it has poor acceleration due to only using two motors. Even though it may not be an optimal drivetrain, it is still possible to be competitive as long as the drivetrain is reliable. As a consensus, we would advise every new team to build the pushbot primarily to learn how to build with a kit. The pushbot is a good starting point and helps the team get familiarized with using kit parts, attaching wheels, mounting motors, etc.; however, it is subpar to every other drivetrain in a competitive context. **While the pushbot is a good first drivetrain for new teams to get acquainted with the kit, it is recommended that teams move away from this drivetrain when building their competition robot.**

Advantages

- Most simple drivetrain to build
- No need to worry about powering all four wheels

Disadvantages

- Slower than other options
- Underpowered (all other drivetrain options typically use 4 motors)
- Uses the Tetrix MAX motor, which is prone to burning out easily, is underpowered, and has a fragile gearbox.
- Lacks agility and maneuverability due to 2 motor turning
- Poor acceleration due to 2 motors
- Often direct driven, which is highly discouraged for drivetrains

4 Wheel Drive

In its most common form, the drivetrain uses the same wheel layout as the two wheel drivetrain (2 traction wheels in the back, 2 omnis in the front), but with one notable difference: **all four wheels are powered**. Some teams will put all four traction wheels or use all four omni wheels. It is not suggested to use all 4 traction wheels due to poor turning that results in this setup. This is caused by wheel scrub (friction between the

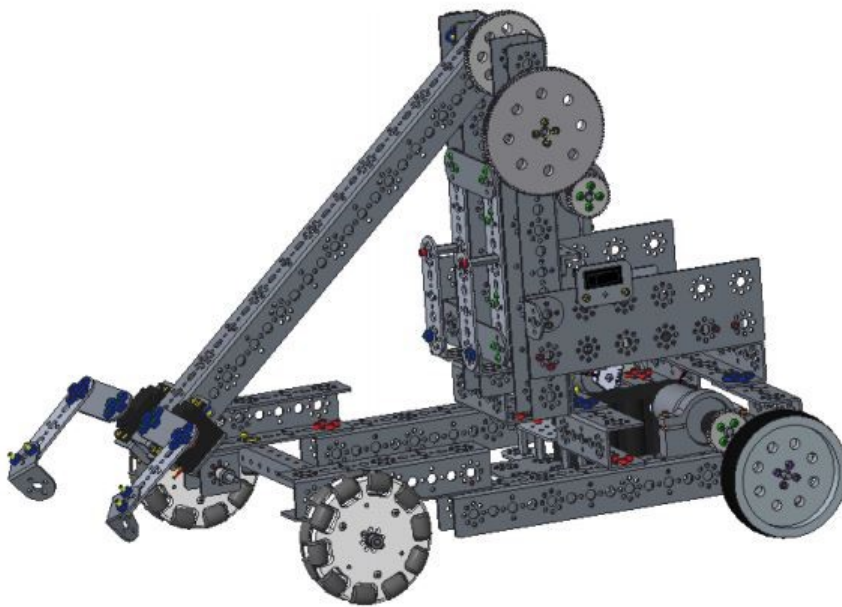


Fig. 15: Example pushbot drivetrain included in the starter FTC guide

side of the wheel and the floor tile). Using omni wheels for all four wheels will result in incredible turning, with the robot rotating around its center. However, this advantage comes with a major loss of traction. For these reasons, many teams choose to use two traction wheels and two omni wheels for a balance between quick turning and traction. The primary advantage of this drivetrain over other tank drivetrains is its ability to easily move across raised terrain when the bot's center is raised above the terrain.

It is suggested that a four wheel drivetrain be close to, or exactly, a square. Otherwise, one may encounter problems turning. **Weight distribution is furthermore a large factor that should be considered: the more weight in the back, the better.** Off-center turning, which may or may not be a drawback, is nearly ubiquitous among 4 wheel drives. This may not be a problem for teams, but it is good to be aware of it. Off-center turning can be an advantage, but be warned that turning will be slightly slower on 4 wheel drives than their six or eight wheel counterparts.

Advantages

- More maneuverable than 2WD
- Solid acceleration and traction
- Can traverse terrain if the chassis is raised high enough
- Good pushing power for defense, yet maneuverable enough to avoid it

Disadvantages

- Can tip more easily than 6WD/8WD with a high center of mass
- All traction wheel 4WD can have decreased maneuverability
- Weight distribution factors into the turning point and turning radius of the robot

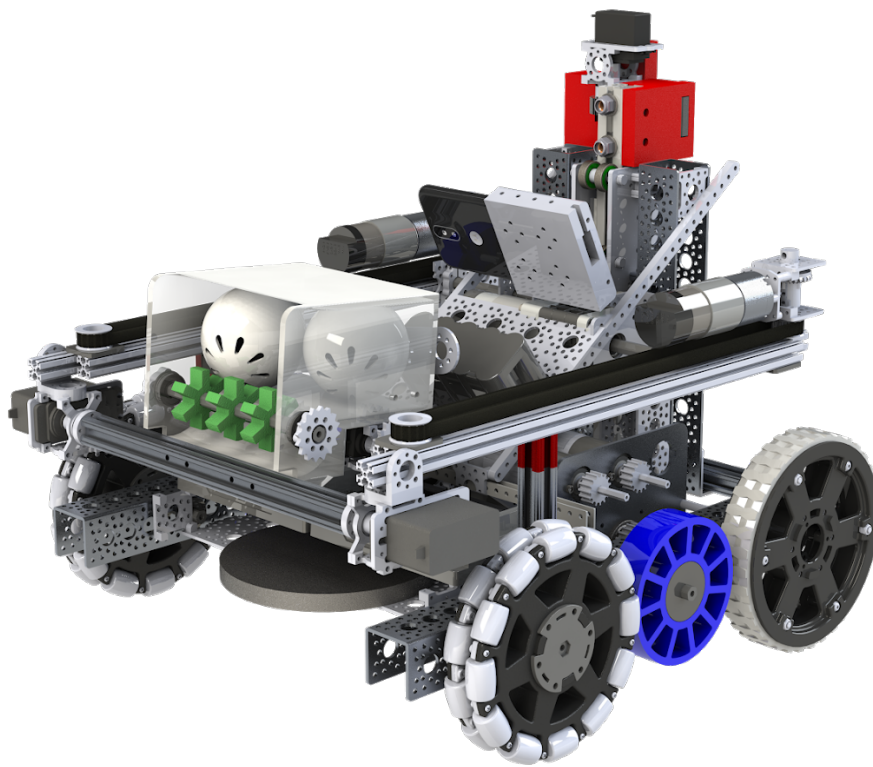


Fig. 16: 3736 Serious Business, Rover Ruckus; modified 4WD

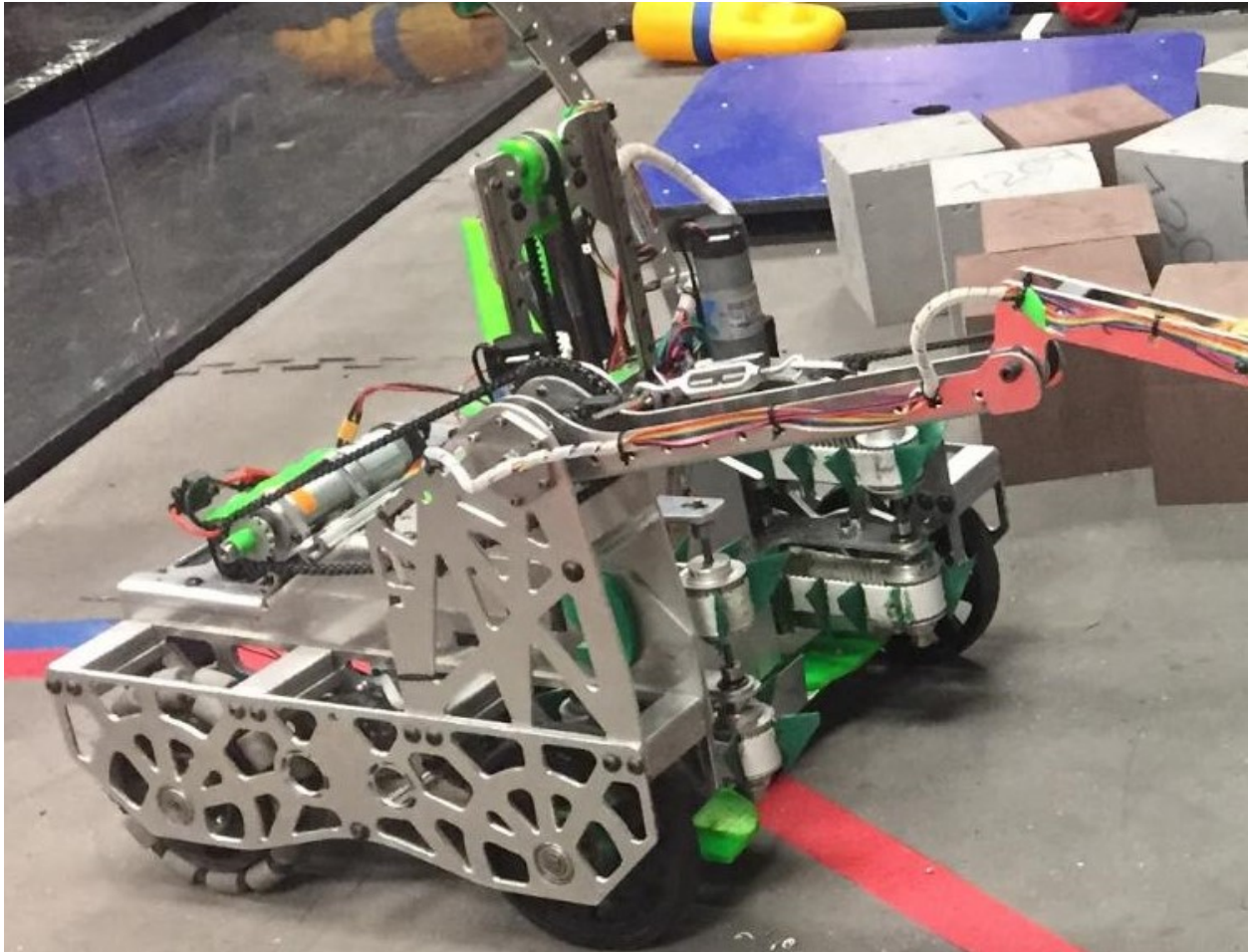
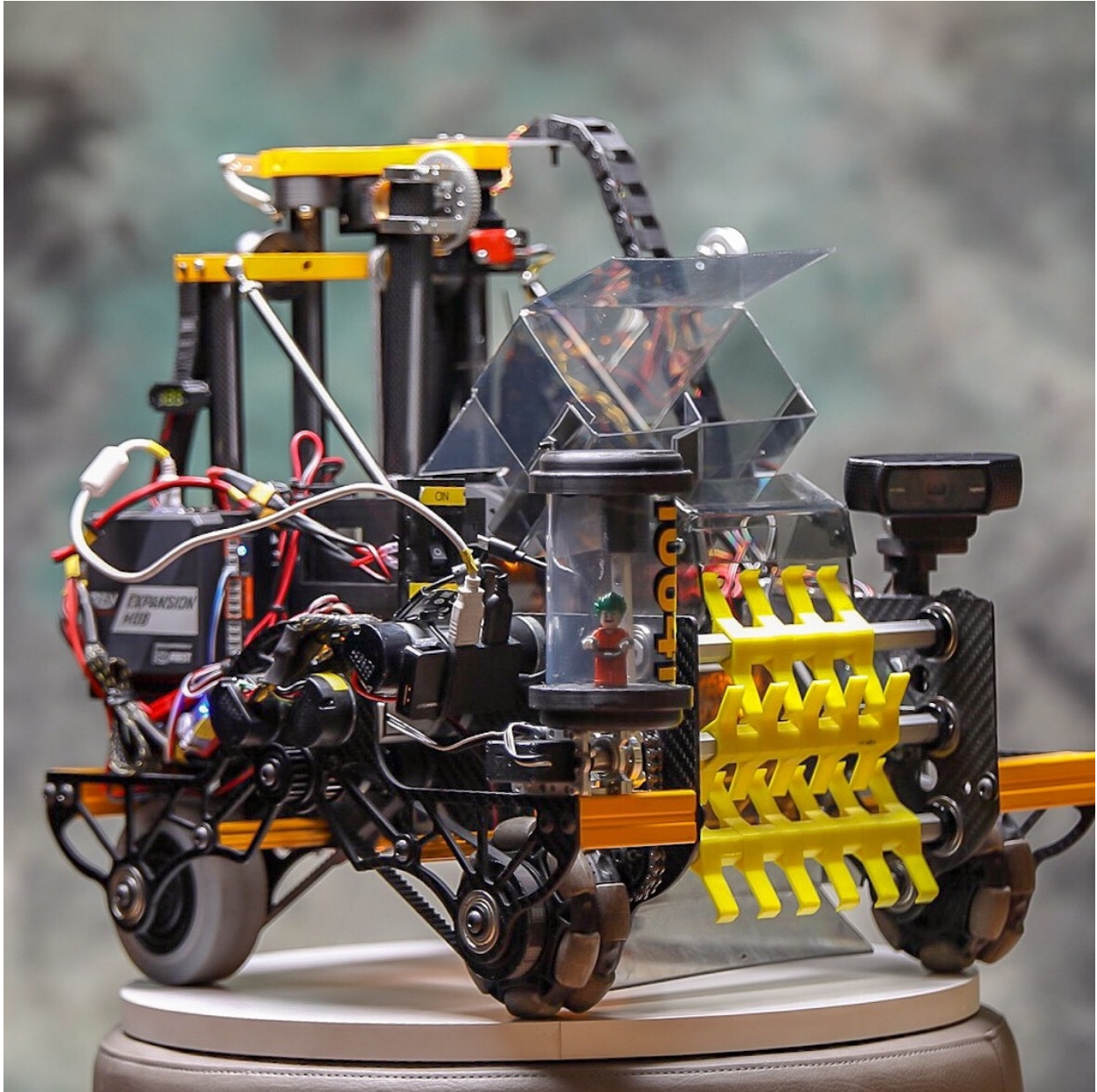


Fig. 17: 7209 Tech Hogs, Relic Recovery



6 Wheel Drive (6WD)

A 6 wheel drivetrain is a common competitive drivetrain in FTC for multiple reasons: it has fantastic traction, great turning, and by having 6 wheels, the drivetrain has more contact with the ground, helping with stability and traction. There are two main types of 6 wheel drivetrains: ones with **corner omnis** and ones with a **drop center**.

A drop center 6WD is a 6 wheel drive with the center wheel mounted slightly below the other two wheels. The drop should be anywhere from more than $1/16$ " up to $1/4$ ". However, the recommended drop is around $1/8$ ". Typically, lighter robots (<25 lbs.) can have less drop, while heavier robots (>25 lbs.) perform slightly better with more than $1/8$ " drop. The purpose of dropping the middle wheel is to ensure that only 4 wheels are in contact with the ground at all times. This is because turning with 6 wheels on the ground introduces lots of friction, making it very difficult to turn quickly. Note that the **required drop may vary depending on both field**

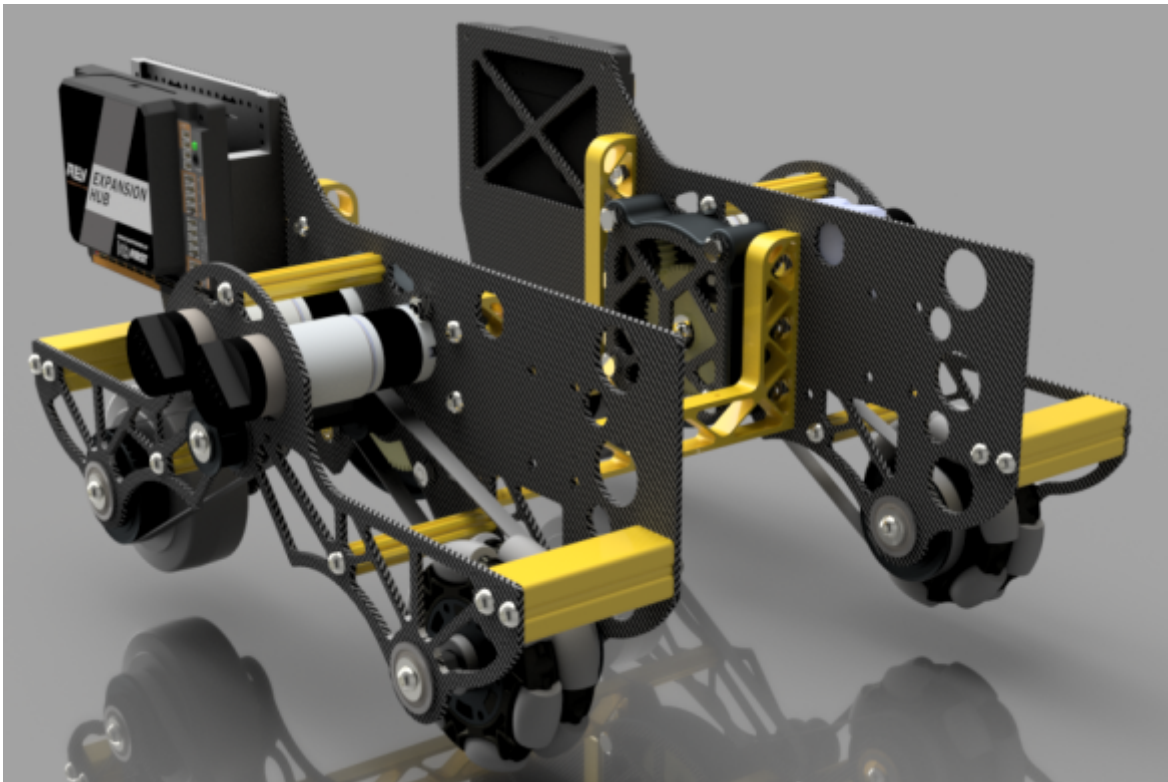


Fig. 18: 10641 Atomic Gears, Rover Ruckus

condition and weight of the overall robot. Turning can drastically degrade due to a difference in material underneath the field, leading to the robot sinking down further than usual. Moreover, **most drop center 6 wheel drives are made using custom drivetrains** because it is difficult to get the center wheel drop using a kit based build system (a notable exception being REV-based kit drivetrains). It is possible to execute a drop center using goBILDA and Actobotics using pillow blocks, but it is a little bit more awkward.

6 wheel drive with corner omni wheels does not have a center drop. It attempts to solve the issue of turning by replacing the corner traction wheels with omni wheels, allowing the drivetrain to achieve better turning, albeit with slightly less traction than a center-drop. This is very easily buildable in kits, and is a great all-around drivetrain. Drop center and corner omnis can be combined for maximum turning reliability, although this comes with side effects like rocking and reduced traction.

(Side note: tolerances in the size of the wheel may affect the traction of this drivetrain - many VEX Robotics Competition teams have built 6WDs like this only to find that their center traction wheel wasn't touching the ground due to variance in size. We don't have data on this phenomenon with FTC-class wheels, but it's something to be aware of.)

Advantages

- Great traction and maneuverability
- Good acceleration, can have a high top speed
- Great stability under all robot weights
- Able to play defense

Disadvantages

- Drop-center 6WD is tricky to build with channel-based systems
- Drop-center 6WD as slightly worse turning, but more traction
- Drop-center 6WD is dependent on field conditions
- Corner omni 6WD has less traction
- Slow gear ratios will make a 6WD feel sluggish

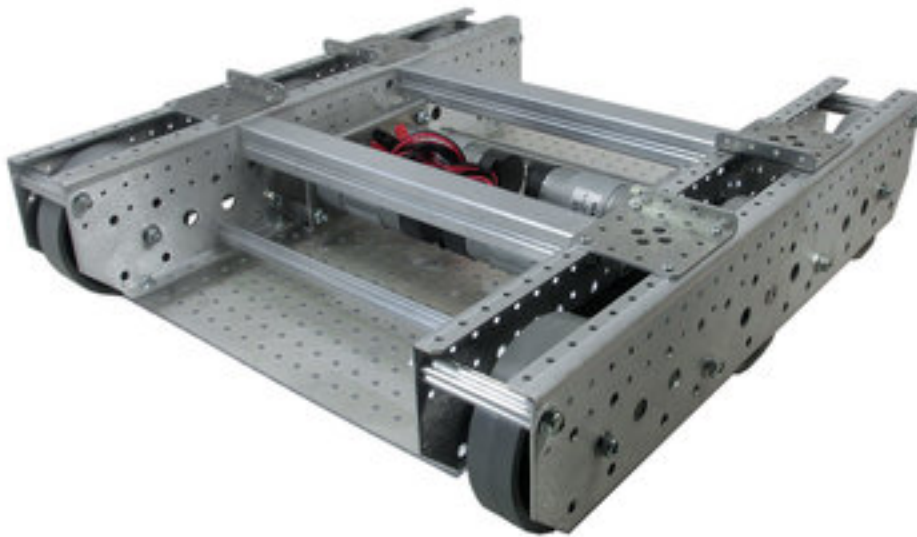


Fig. 19: COTS Andymark Tilerunner Drivetrain with 0.05" center drop. **Not recommended due to price.**

8 Wheel Drive

An 8 wheel drivetrain is less common than its 6WD counterpart, combining elements found in both 4 wheel and 6 wheel drivetrains. For example, the 6 wheel drivetrain generally will have a dropped center wheel so that the robot turns on four wheels instead of six, reducing friction and increasing turning mobility. On an 8 wheel drive, the center four wheels are dropped. This means that when turning, only these middle four wheels are touching the ground. Thus, the 8 wheel drivetrain has slightly more stability while turning than a 6 wheel drive, whereas 6 wheel drives can turn more quickly. Furthermore, since the 8 wheel drive has wheels in the same place as a 4 wheel drive, it has the stability of a 4 wheel drive. It is suggested that all



Fig. 20: 7236 Recharged Green, Rover Ruckus; drop center 6WD

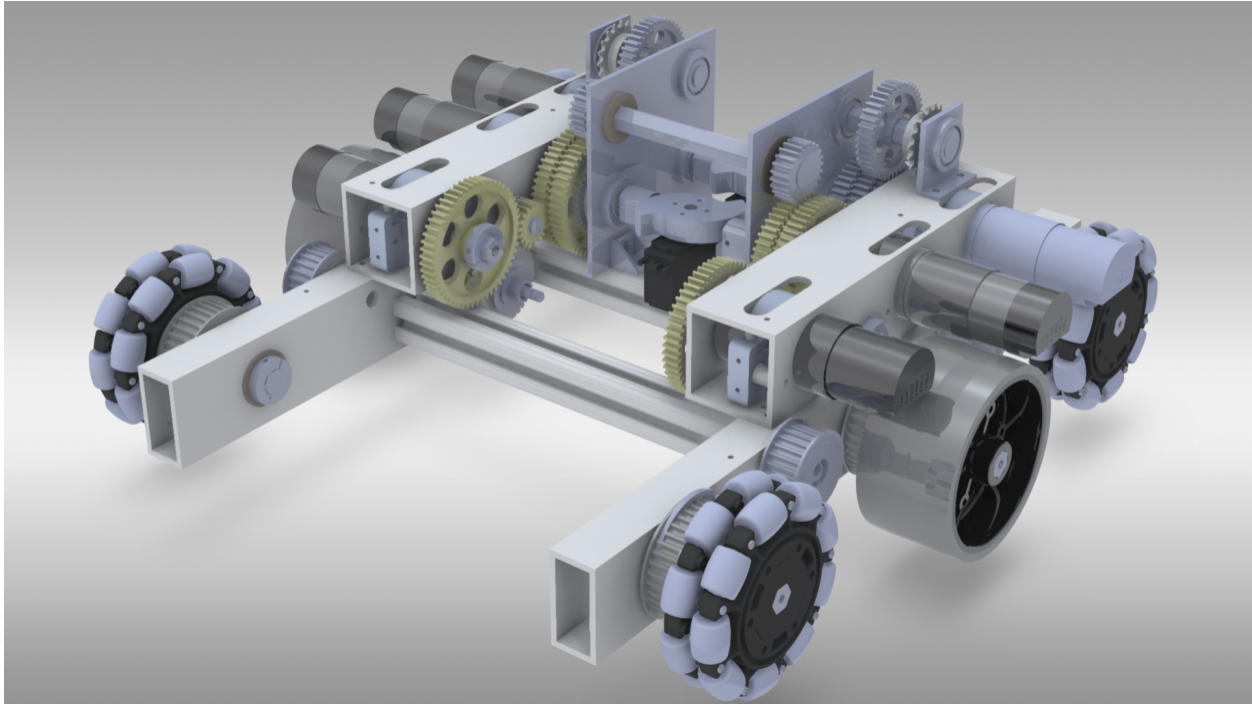


Fig. 21: 5975 Cybots, Velocity Vortex, drop center and corner omni 6WD

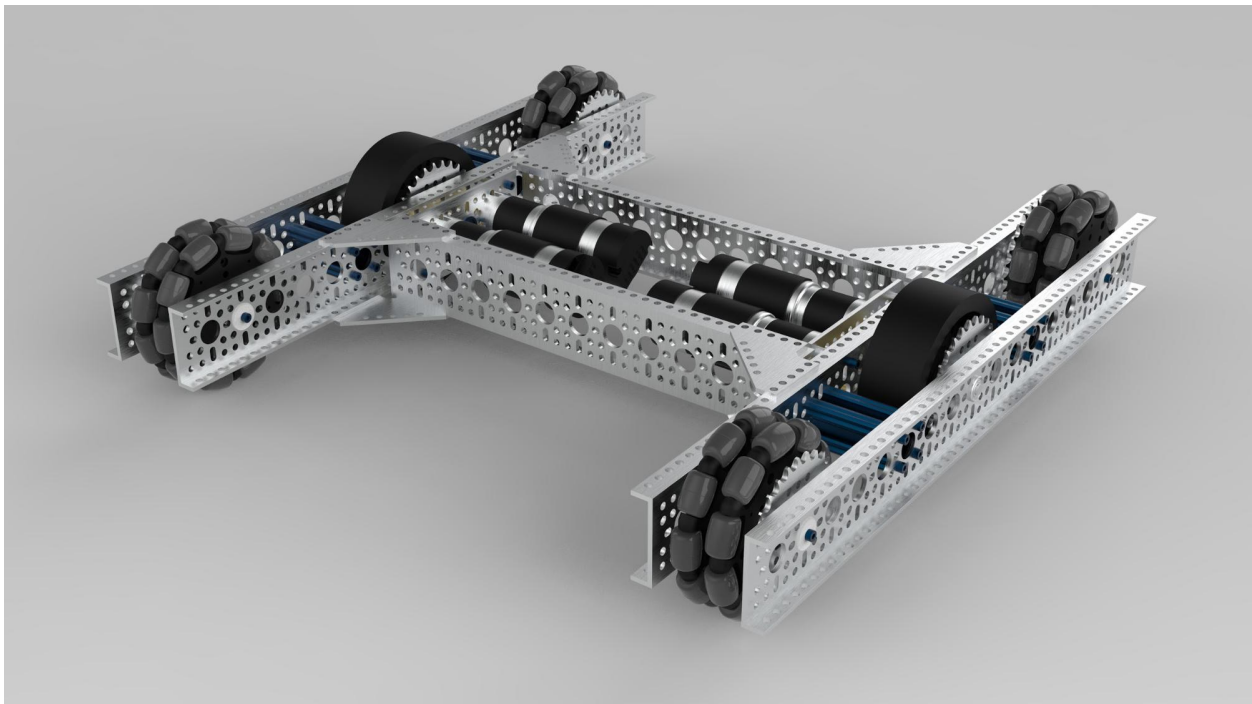


Fig. 22: Ethan Doak, goBILDA 6 wheel drive with corner omni wheels

eight wheels should be powered, and planetary motors should be used over a spur gear motor. Teams have the option of using doubled omni wheels on the outer four wheels. Doing so will reduce traction/pushing power and increase mobility.

Advantages

- Combines the stability of 6WD with the agility of 4WD
- Even more stable than 6WD
- Fantastic traction and acceleration
- Great for defense

Disadvantages

- Takes up more space than 6WD
- Powering all 8 wheels can be tricky
- Without adequate center drop, turning can be drastically reduced

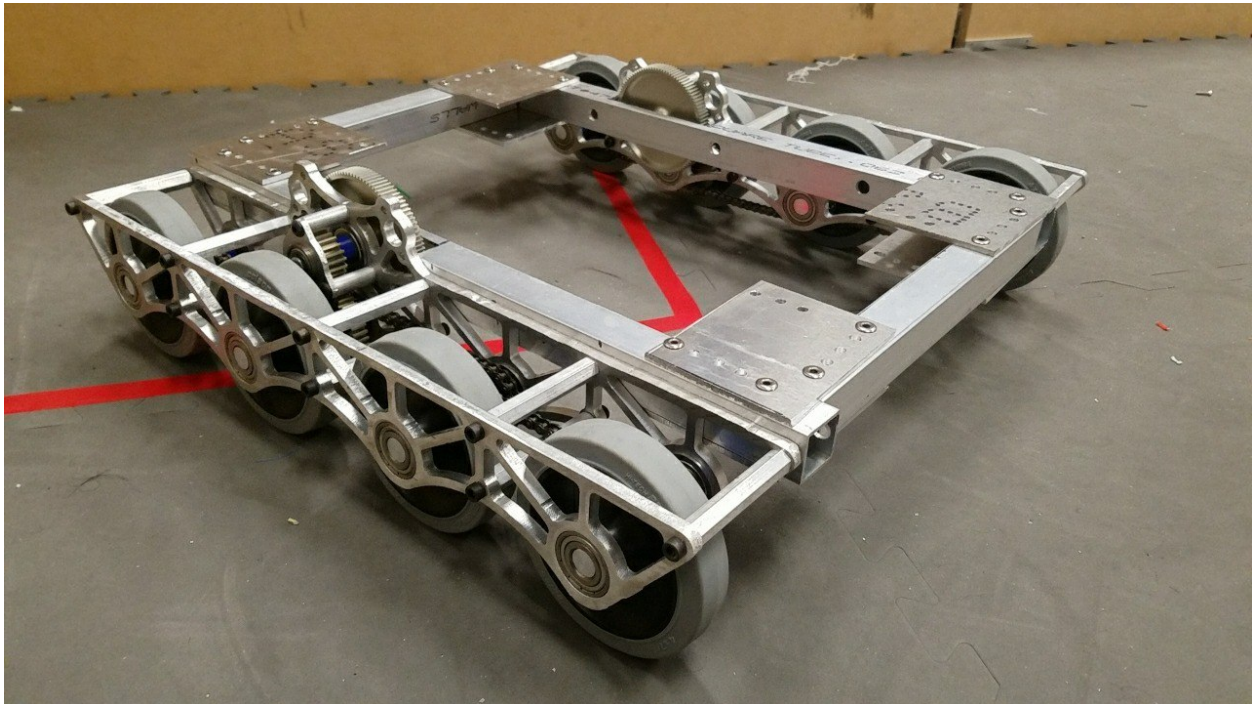


Fig. 23: 3846 Maelstrom, Rover Ruckus



Tread Drive

Tread drive is the use of tank treads or wide belts to power movement, much like a real life tank. Unfortunately, in FTC, it is not a competitive drivetrain for a number of reasons. Tread is complex, and has many points of failure. Treads are also very prone to defuse, and a slight hit from another robot is enough to misalign the treads. Commercial Off-The-Shelf (COTS) tread options aren't great either - TETRIX tracks have a tendency to snap and derail when used on robot drivetrains, making them not suitable for competition use. While it is possible to implement tread drive successfully, such as in the example below, most inexperienced teams do not have the capability and know-how to do so. Tread drive has negligible traction improvements at the cost of maneuverability. There are better options to traverse terrain, such as a 4WD.

Advantages

- Very good at traversing terrain
- Fantastic traction and pushing power

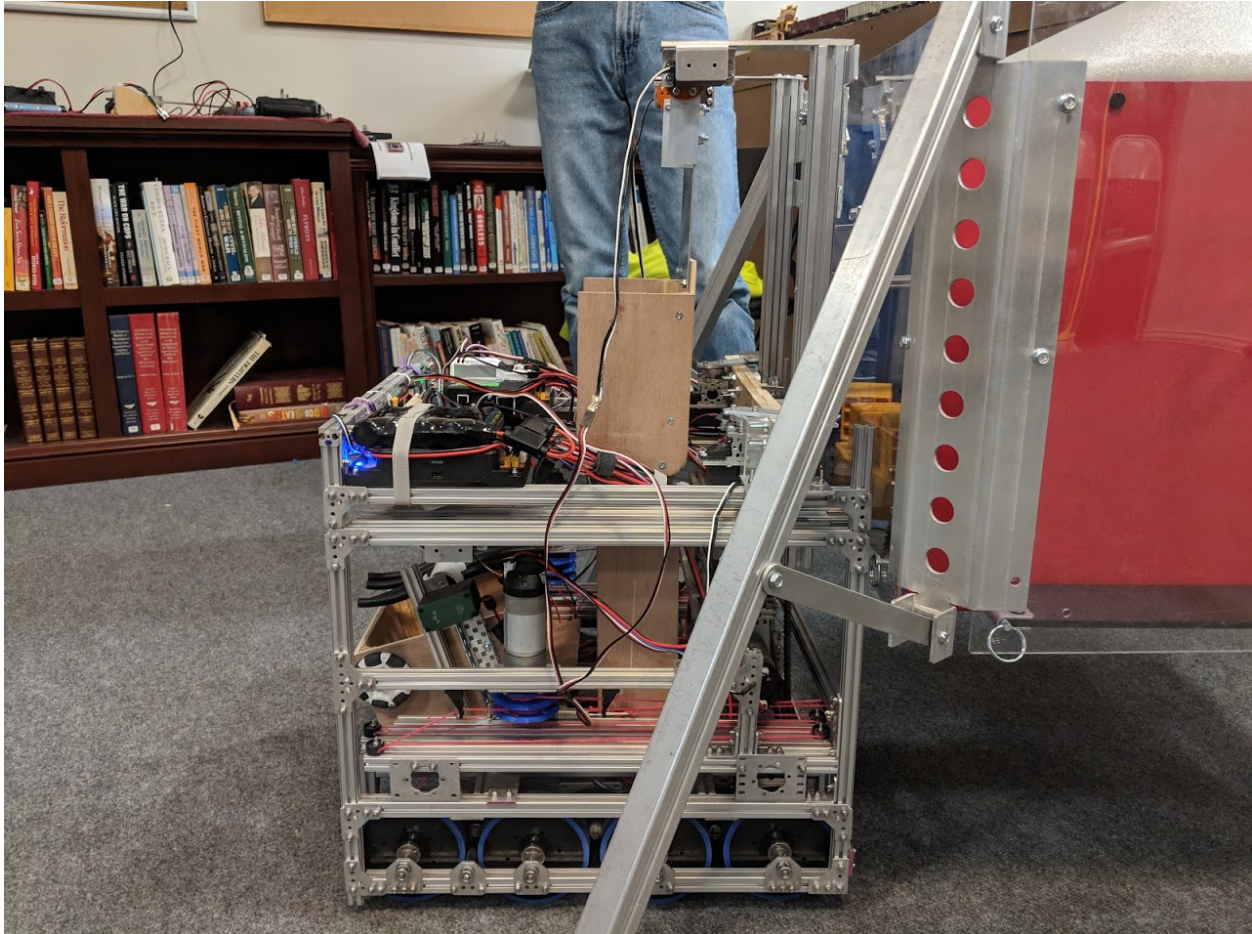


Fig. 24: 13075 Coram Deo Academy Robotics, Rover Ruckus

Disadvantages

- Suffers in maneuverability and top speed
- Very complex to implement
- Treads are prone to breakage and can fall off easily

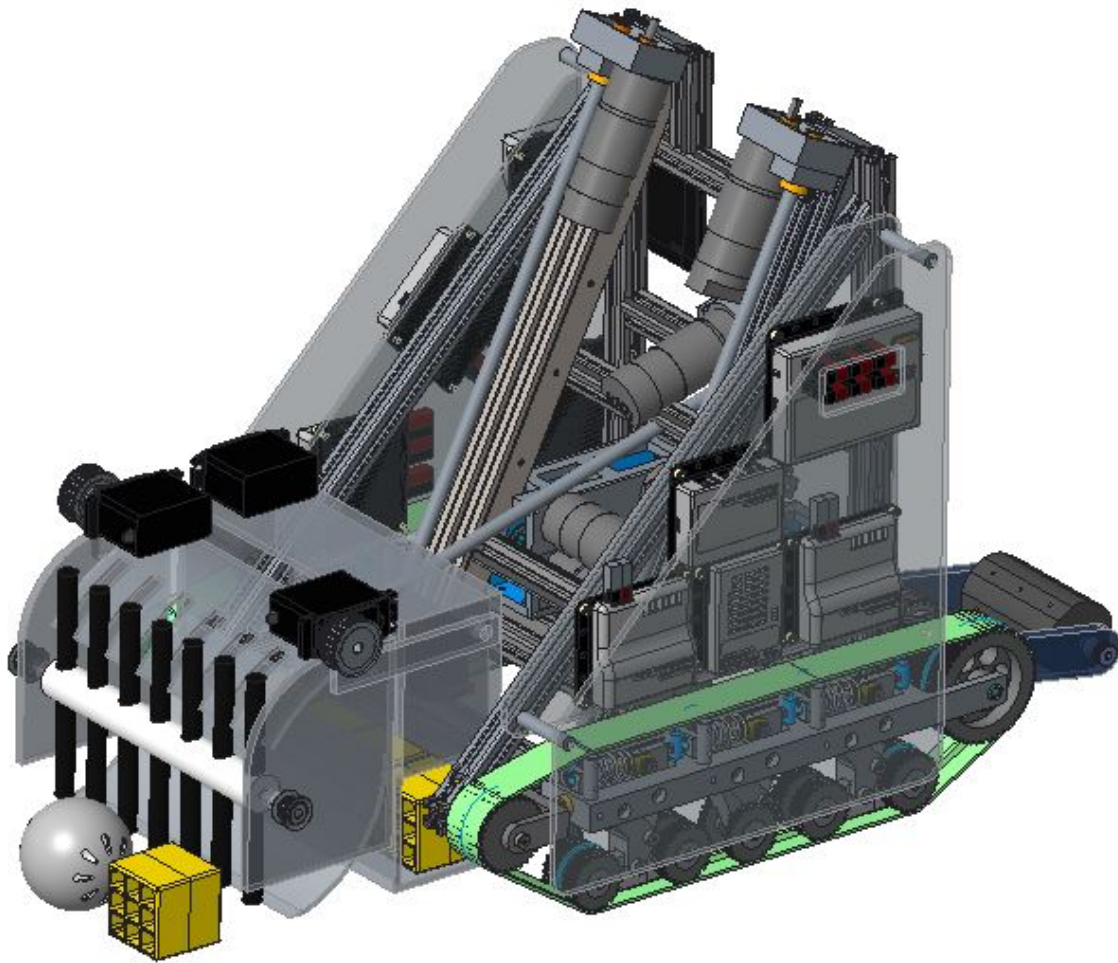


Fig. 25: 5975 Cybots, Res-Q

Holonomic Drivetrains

Mecanum Drive

Mecanum drivetrains consist of four mecanum wheels which are powered independently by one motor. This configuration angles the velocity of each wheel, allowing the robot to strafe (move side-to-side). The primary advantage to mecanum drive is the maneuverability it affords, especially because the robot can strafe instead of turn and drive. The rollers on mecanum wheels form a 45 degree angle with the wheel's axis of rotation, which means that mecanum drivetrains can't strafe as fast as they can drive forward. This

can be explained by discussing the forces involved. When each wheel rotates, it applies a friction force to the ground, which moves the robot. When moving forward, both sets of left wheels rotate in the same direction at the same speed, and both sets of right wheels rotate in the same direction at the same speed, meaning that the forces do not oppose each other. However, when strafing, neither the two left wheels nor the two right wheels are rotating at the same speed. In many cases, they even rotate in opposite directions. These two opposing forces cause the rollers to slip more and more, which angles the robot's velocity at the expense of traction (more slipping results in a loss of speed). However, the wheels do still slip when moving forward but not as drastically as they do when strafing. This is the primary disadvantage to mecanum drivetrains: they tend not to have much pushing power and thus, are vulnerable to defense by a sturdy tank drive. Due to the fact that mecanum wheels are more likely to slip because of the diagonal rollers, an optional addition to mecanum drives is a separate odometry mechanism in order to track the robot's location during autonomous.

It is important to note that in order to maximize the efficiency and stability of mecanum drives, when viewed from above, the rollers of each wheel should point towards the center of the robot, forming an X shape, rather than a rhombus. The primary reason for this is that it allows the drivetrain to turn significantly faster than it would otherwise be able to. When using the suggested setup, when viewed from the robot's underside, the rollers form a rhombus. This allows the force applied by the wheels on the ground to act tangent to the turn radius, leading to faster turning.

Advantages

- Fantastic maneuverability and agility due to strafing, can avoid defense very well
- Good acceleration, can have high top speed
- Very versatile drivetrain for nearly any game

Disadvantages

- Suffers in traction, as mecanum rollers have a lower coefficient of friction than traction wheels; cannot traverse terrain
- Able to be pushed around on defense
- Wheels must be powered independently, so there is no redundancy

Mecanum Wheels Miniguide

There are plenty of mecanum wheels on the market, and it can be very daunting to choose between the many vendors. An important feature is the type of mechanism that facilitates the motion, either **bushing** or **bearing**. Bearing based mecanum wheels often have superior strafing because there is less resistance for the rollers to overcome. Another important note is that some FTC teams invest in 6 inch mecanum wheels instead of 4 inch mecanum wheels, often at a much higher price. **It is highly recommended that teams stick with 4 inch mecanum wheels.** Here is a general list of the mecanum options ranked in order of recommendation.

1. [goBILDA Mecanum Wheels](https://www.gobilda.com/3606-series-mecanum-wheel-set-bearing-supported-rollers-100mm-diameter/)⁴ (\$105 with team discount): This recent addition to the lineup has become one of the strongest options for its variety of positive attributes. goBILDA Mecanum wheels are based on the tried and tested Nexus bearing mecanum wheels, which means it has fantastic strafing. It is also very robustly built, and is significantly more convenient to mount to FTC standard build systems. It has a built in goBILDA 16mm and 32mm hole pattern, and has easy support for dead axle. You can mount hubs in

⁴ <https://www.gobilda.com/3606-series-mecanum-wheel-set-bearing-supported-rollers-100mm-diameter/>

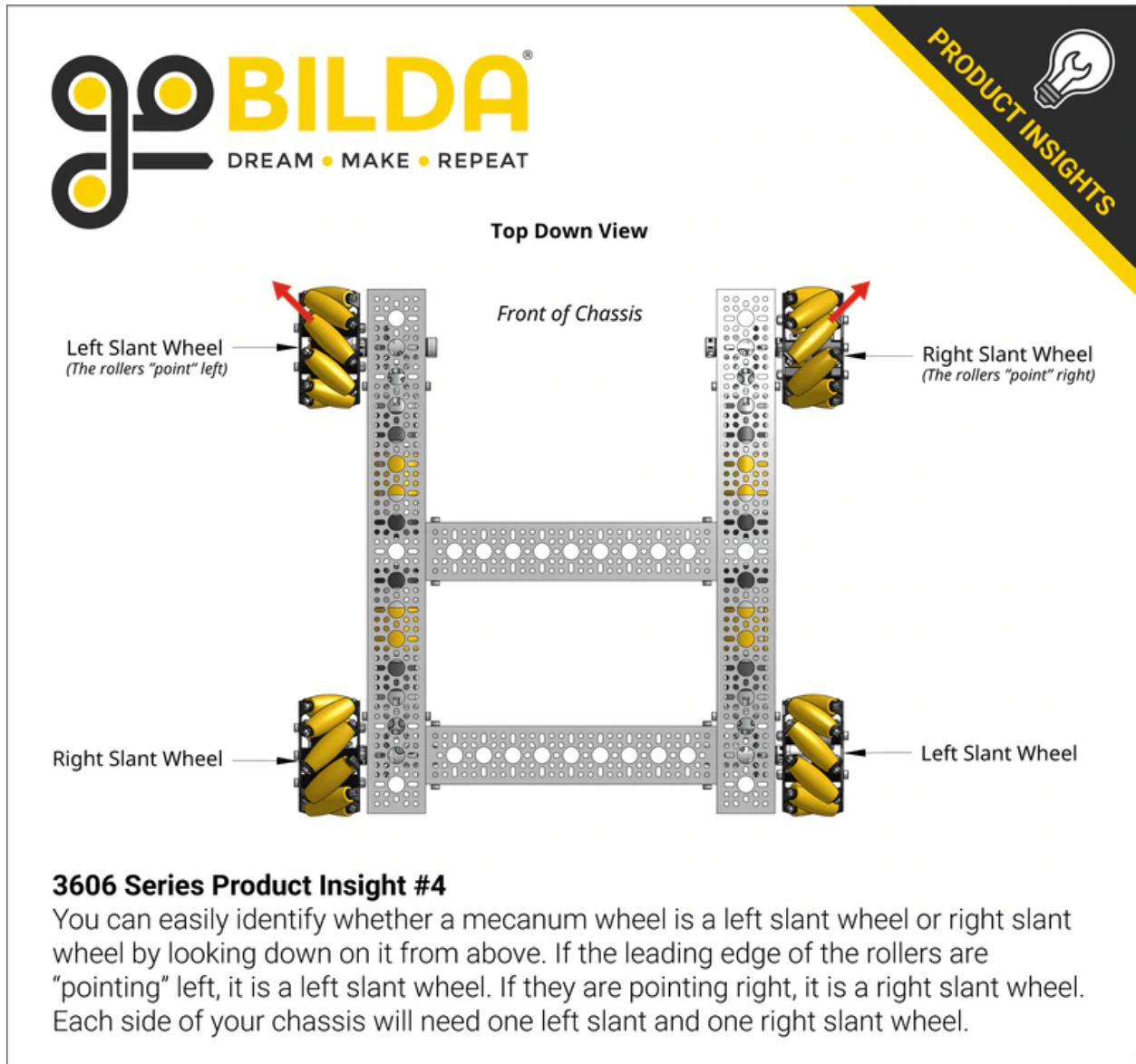


Fig. 26: Configuration for mecanum wheels, courtesy goBILDA

wheels and goBILDA mecanum wheels can easily mount to 1/2" Hex, 3/8" Hex, 12 mm REX, 6 mm D, 1/4" D, and many other shafts. Hubs can also be mounted inside the wheel for very low profile mounting. It is also the cheapest bearing mecanum on the market. Due to its convenient mounting and fantastic strafing performance, we recommend all teams consider goBILDA mecanum wheels.

2. **Nexus Bearing Mecanum Wheel**⁵ (\$134): This was the old gold standard, and still has fantastic performance for the price. This has identical performance with goBILDA mecanum wheels, however is slightly less convenient to mount to. However, these wheels feature the 1.875" bolt pattern commonly used in FRC motion products. It is also slightly heavier than goBILDA Mecanum wheels. Many teams will 3D print adapters or build new cores for Nexus Mecanum wheels. Even though the goBILDA mecanum offer advantages and very few disadvantages over Nexus bearing wheels, these wheels remain a solid option.

3. **AndyMark Heavy Duty 4" Mecanum Wheel**⁶ (\$225): These are easily the most expensive mecanum wheels on the list. These are bushing based mecanum, so they have decent strafing, albeit not as good as the goBILDA and the Nexus bearing mecanum wheels. What sets these mecanum wheels apart is the 80A roller material. AndyMark HD mecanum wheels have higher traction than all other mecanum wheels, which make them desirable for climbing terrain. For example, during the Relic Recovery season, teams had to climb a "balancing stone", and many teams chose to use the AndyMark HD mecanum wheels to be easily able to climb the balancing stone. However, in most cases, being able to more effectively strafe is more important than having good traction. For this reason, **teams are recommended to buy bearing based mecanum wheels like the Nexus or the goBILDA mecanum wheels instead of the AndyMark HD wheels due to the major price difference.**

4. **Nexus Bushing Mecanum Wheel**⁷ (\$84): This is the Nexus Bearing Mecanum wheel with bushings instead of bearings. Before the introduction of the goBILDA mecanum wheels, these were the best budget option, however, it is now more sensible to spend the \$21 premium to get the bearing goBILDA mecanum wheels.

5. **VexPro Mecanum Wheels**⁸ (\$119.96): These mecanum wheels are most suited for vectored intakes on FRC robots. They are relatively tough, but have somewhat poor strafing and are not quite as durable as the other wheels higher on the list. They are a decent choice if you already have them, but otherwise, there is no reason to consider them.

6. **TETRIX Mecanum Wheels**⁹ (\$113): At the time of writing this guide, these haven't been released so no verdict can be reached. However, they have a built in hub so they can be easily mounted on 6mm D shaft. They are bushing based, and due to no testing and evaluation outside of TETRIX, and its higher price than the goBILDA mecanum wheels, **we cannot recommend the TETRIX Mecanum Wheels.**

7. **VEX EDR Mecanum Wheels**¹⁰ (\$59.99): These are the cheapest mecanum wheels, but have a strange shaft standard (1/8 square) which require the use of 3D printed adapters. **There are not many reasons to purchase these wheels.**

8. **AndyMark Standard Duty Mecanum Wheels**¹¹ (\$92): **DO NOT PURCHASE THESE WHEELS.** These are terrible mecanum wheels. They barely strafe and are super fragile. Just buy goBILDA mecanum wheels for \$13 more.

Unrated: REV 75mm Mecanum Wheels¹² (\$76.50 with team discount): At the time of writing, these mecanum wheels were just released and teams have not had a chance to test them. They are bearing based, similar to the top-ranked wheels and have an included adapter which allows them to attach to 5mm hex shaft. Additionally, these mecanum wheels are 75mm (~3in) in diameter, instead of the typical 4in.

⁵ <https://www.superdroidrobots.com/shop/item.aspx/4-inch-nexus-mecanum-wheels-ball-bearing-set-of-4/1352/>

⁶ <https://www.andymark.com/products/4-in-hd-mecanum-wheel-set-options>

⁷ <https://www.amazon.com/100Mm-Aluminum-Mecanum-Wheel-Right/dp/B01CTUT4GY>

⁸ <https://www.vexrobotics.com/mecanum-wheels.html>

⁹ <https://www.pitsco.com/TETRIX-MAX-Mecanum-Wheels>

¹⁰ <https://www.robotshop.com/en/mecanum-wheel-set.html>

¹¹ <https://www.andymark.com/products/4-in-standard-mecanum-single-wheel?via=Z2lkOi8vYW5keW1hcmsvV29ya2FyZWE6OkNhdGFsb2>

¹² <http://www.revrobotics.com/rev-45-1655/>

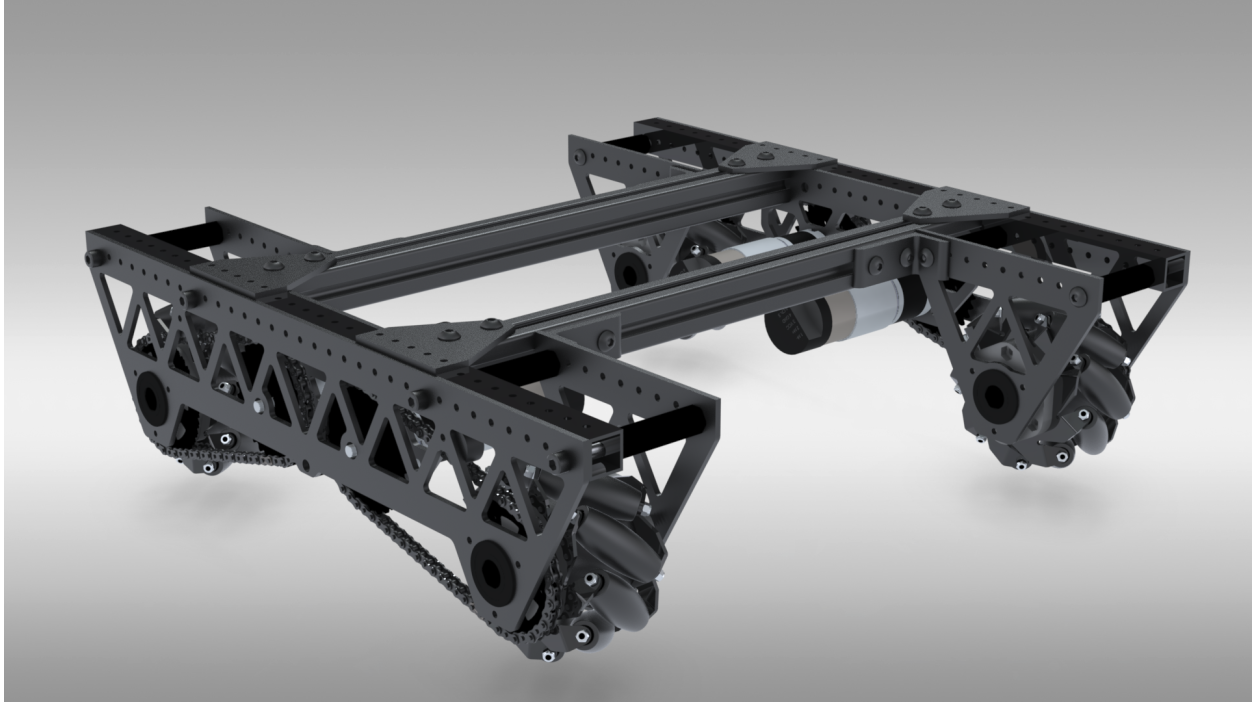


Fig. 27: 8103 Null Robotics, Rover Ruckus, using Nexus bearing mecanum

X-Drive

X-Drive is a holonomic omni-wheel based drivetrain. This type of drive involves mounting 4 omni wheels at the corner of the robot at a 45 degree angle. One notable difference between X-Drive and mecanum is strafe speed. While, as mentioned in the mecanum section, the ratio of strafe speed to forward speed is noticeably less than 1, the ratio on an X-Drive is exactly 1 due to the rotational symmetry of the wheel placement. This means that an X-Drive bot's strafe speed and forward speed are equivalent. The drivetrains are slower, however, when strafing at 45° (approximately $\frac{\sqrt{2}}{2}$ of its forward speed). Even though X-drive has good turning and acceleration, the main downside to the drive is packaging/form factor. Packaging refers to how easy/convenient the drivetrain fits into the overall design of the robot. Ideally, the drivetrain should take up as little space as possible to make it easier to design mechanisms around. Because the omni wheels are offset, packaging a X-Drive is more difficult than other types of holonomic drive like mecanum or H-Drive. Also because of the strange packaging, it is relatively difficult to cleanly transfer power from the motors to wheels, meaning that most X-Drives end up being direct-driven, which is bad on the lifespan of the motor gearbox.

Advantages

- Good maneuverability and agility
- Good acceleration

Disadvantages

- Prone to defense, pushed around easily

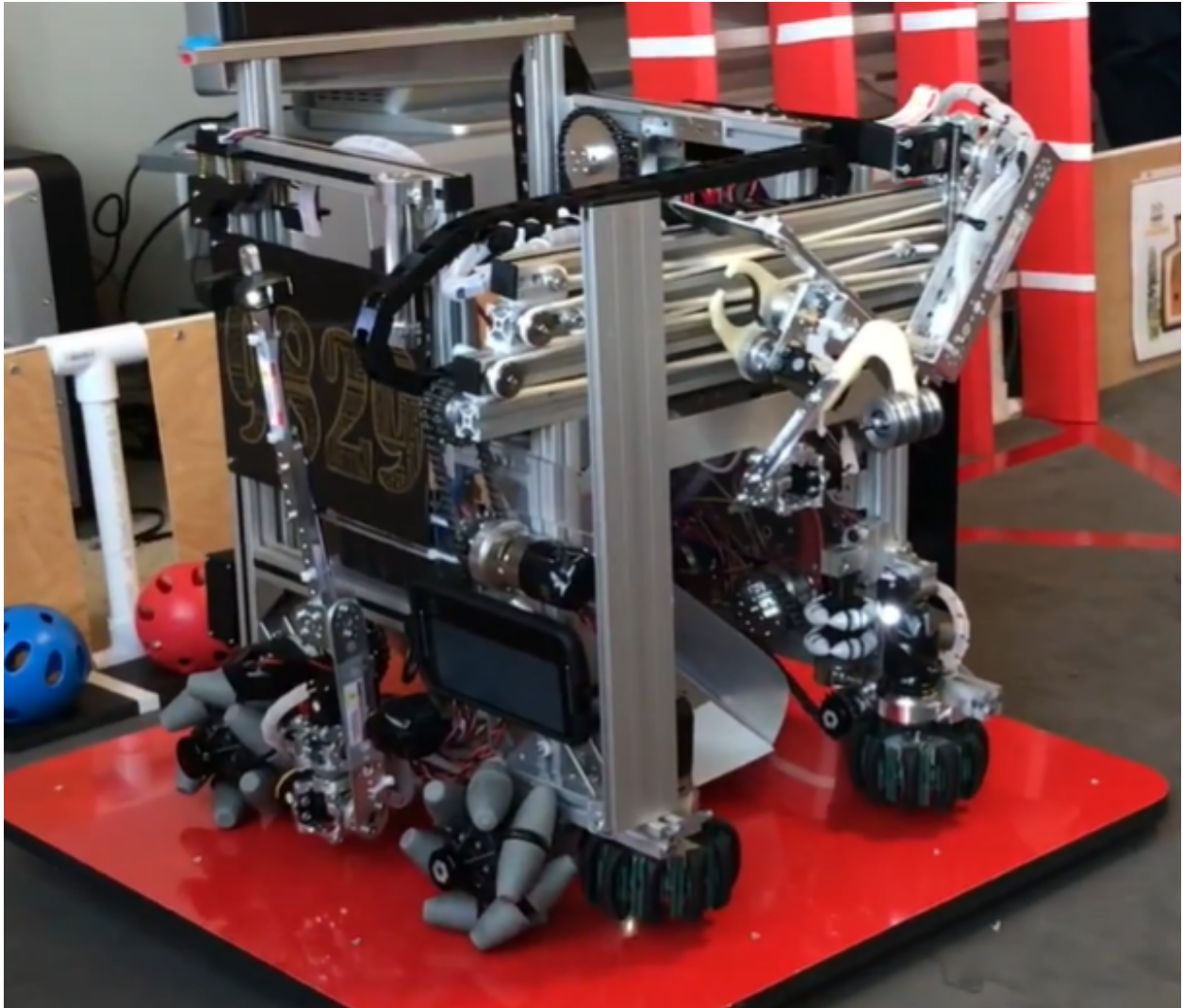


Fig. 28: 9829 MakBots, Relic Recovery, using **VexPro mecanum**

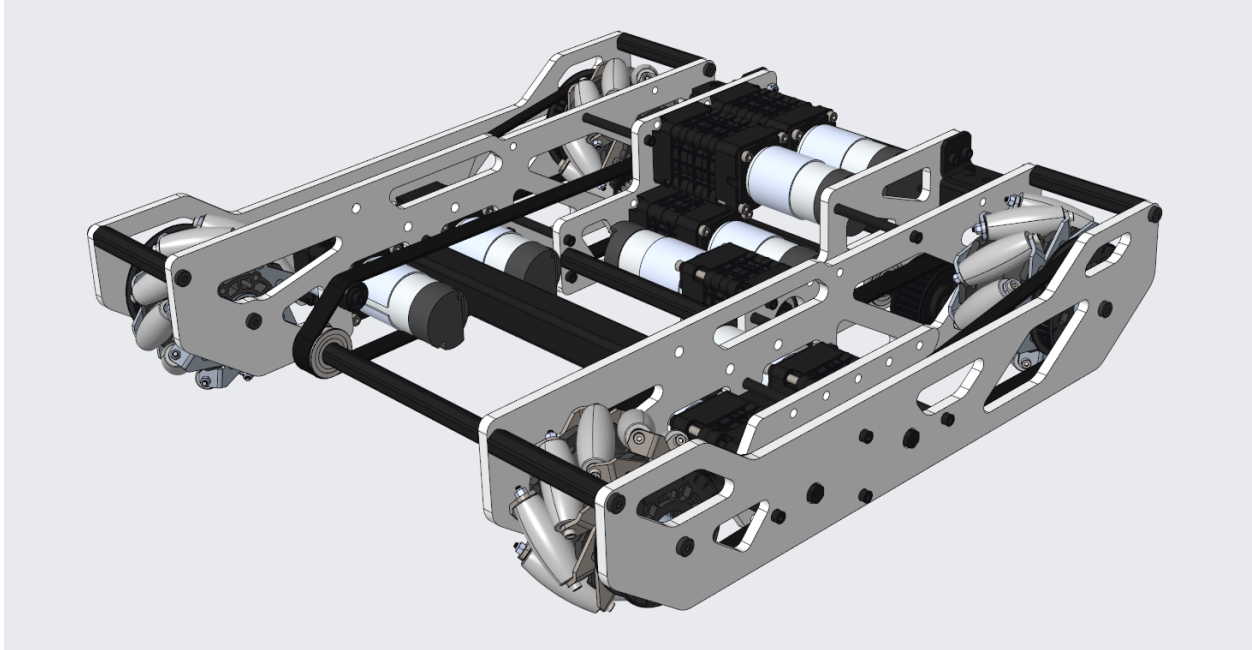


Fig. 29: 731 Wannabee Strange, Rover Ruckus, using **AndyMark HD mecanum wheels**

- Slower top speed than forward-facing wheels
- Often uses direct drive due to awkward form factor

H-Drive

H-Drive (also known as U-drive, depending on the configuration) is a holonomic type drive that uses all omni wheels. H-Drive relies on a set of “strafe wheels” that are perpendicular to the forward/backward wheels to achieve strafing. H-Drive is similar to a fusion of a tank drivetrain while retaining the maneuverability and strafing of holonomic drivetrains. H-Drive is theoretically very easy to code, but most teams employ some sort of gyro correction to strafe straight, although it is not necessary with proper weight distribution. H-Drive has a number of possible motor configurations - 1 or 2 motors can be put on each forward drive pod, and one or two motors can be put on the strafe wheels. In this configuration, H-Drive has slightly reduced acceleration compared to mecanum drive. For the highest possible reliability, many FRC teams will suspend their strafe wheels on a rocker system to ensure that all wheels are in contact with the ground while the robot is not strafing. By far the biggest advantage of H-drive is its ability to accommodate multiple motor distributions. For instance, if you want to dedicate only 3 motors to your mechanisms and you have a motor left over, using a 1 strafe motor, 4 drive motor configuration is absolutely viable. Or if you dedicate 5 motors for your mechanisms, H-drive with 2 drive motors and 1 strafe motor is definitely optimal.

Advantages

- Combines tank and holonomic drivetrain advantages
- Can be used with 3 or 5 motors
- Good traction and top speed
- Great maneuverability and agility



Fig. 30: 731 Wannabee Strange, Velocity Vortex



Fig. 31: 5040 Nuts and Bolts, Relic Recovery

Disadvantages

- Strafing is slightly less effective than mecanum
- Complex suspension occasionally needed, depending on design

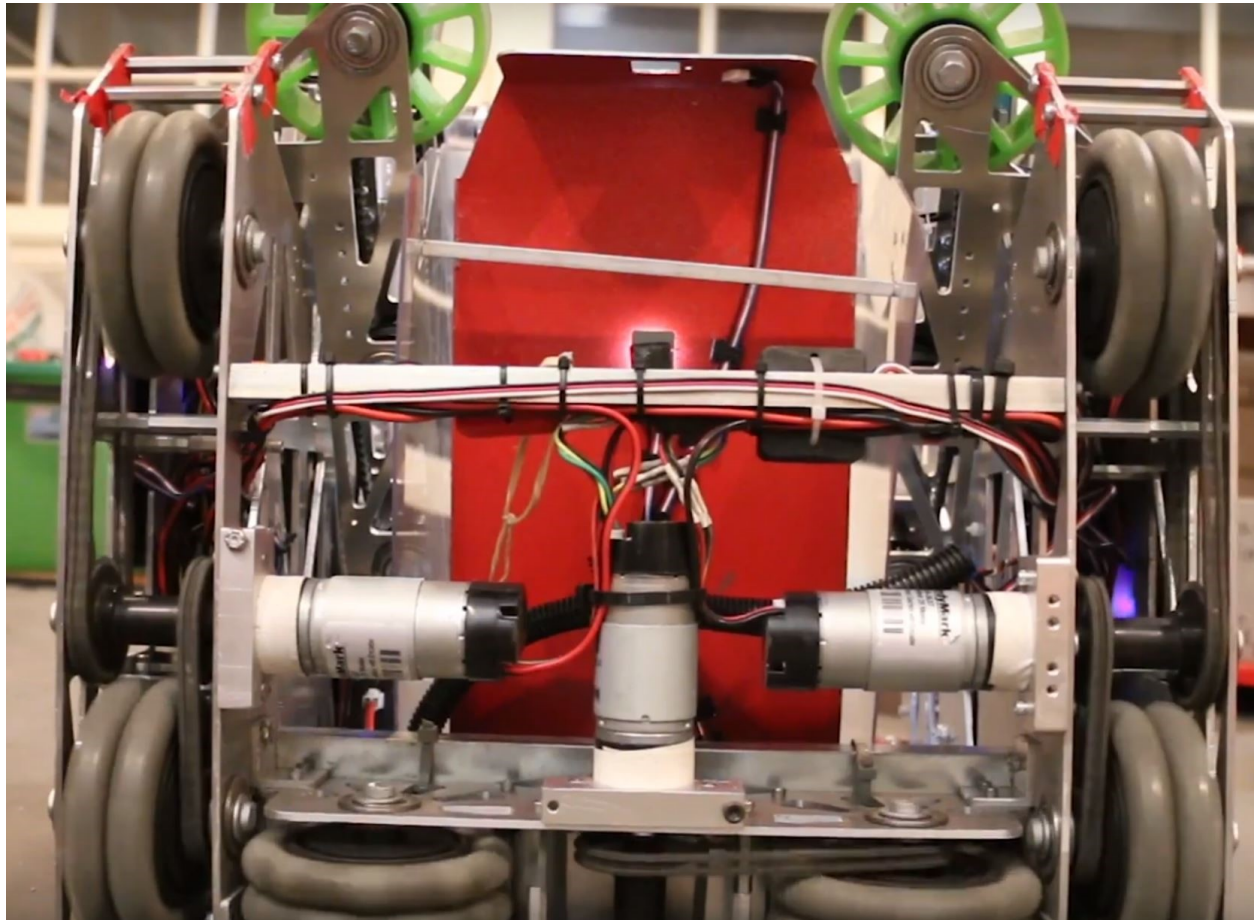


Fig. 32: 9804 Bomb Squad, Relic Recovery

CAD Tutorials

CAD Tutorial Part 1 - Drivetrain in an Hour

Introduction

Disclaimer: CAD is not necessary in FTC to build a successful robot. Many successful teams didn't CAD their robot at all. Other successful teams only made parts of their robot in CAD. Still other teams fully designed their robot in CAD. It is encouraged that new teams try their hand at CAD, especially if you have a mentor or parent who is experienced in such an area. CAD is beneficial for two reasons. First, CAD solves a lot of preventable headaches, such as spacing issues. Thus, it will save time when you discover problems in CAD that you can remediate before you build your robot. Second, CAD is a professional tool that is used in many STEM fields, and having knowledge of CAD will be beneficial in your future career, should you study and work in those fields.

However, CAD is not the magic genie that will guarantee you success in FTC. When used properly, it is a great tool to aid teams in building their robot. Keep in mind, though, that many teams have had success without CAD.

Starting off, choose a CAD program and learn it as well as possible. It may be a good idea to spend a few weeks just finding objects lying around, sit down with a ruler or calipers, and make a quick CAD model. A good way to test accuracy is to choose a solid object made up of one known material. Weigh the object and make the model in CAD, apply that material to it, and see what the weight difference is. This is a good way to test how accurate the CAD model is to the real part. It really doesn't matter what object it is - just find something and make it to the best of your abilities. There are also tons of videos on YouTube; a good one is TFI who makes detailed tutorials for Autodesk Inventor.

Overview of CAD Programs

There are many possible CAD programs that teams can learn. All of these programs can be downloaded for free under a student or FTC team license. You'll have to do a bit of research here, as the requirements for free copies vary based on the program. Here are a couple suggestions to consider:

SolidWorks An industry standard CAD package made by Dassault Systemes. It's as fully featured as CAD software gets, including great simulation features and a very robust assembly environment. It's used widely in industry and is also the program of choice for most college-level engineering classes. However, **it isn't available for Mac users**, and you'll need a pretty beefy computer to run it (16GB RAM is standard). If you have mentors or team members with previous experience in SolidWorks or an engineering class at your school that teaches SolidWorks, it will be your best choice.

Inventor Autodesk's industrial CAD offering. It offers many similar features to SolidWorks, but has a different UI and two distinct assembly modes. While it's used by many companies in the industry, it doesn't appear in very many college curriculums. Inventor is generally the second choice for companies who don't use SolidWorks but instead are based around the Autodesk universe. **Inventor is also not available for Mac**, but it may run better on lower-spec PCs. If you have mentors or team members with previous experience in Inventor or an engineering class at your school that teaches Inventor, it will be your best choice.

Onshape A fully-featured CAD package, but it runs entirely in the cloud. It can run on any computer (even Chromebooks!) and has iOS, iPadOS, and Android apps as well. It has all of the same core features as SolidWorks and Inventor, as well as the best collaboration workflow in the industry - many people call it "the Google Docs of CAD." Onshape allows multiple people to work on the same document at the same time, and allows users to "follow" each other and see what's on another user's screen. OnShape also has FeatureScript, a programming language where you can write custom features. The community has created a lot of very useful FeatureScripts already which you can use completely for free. Onshape also has a comprehensive tutorial system (<https://learn.onshape.com>) that will not only teach you how to use their software, but how to approach design problems. If you're just starting out with CAD, or you don't have access to powerful computers, or you want to have many people working on CAD; Onshape is the software for you.

Fusion 360 A cloud-based all-in-one CAD/CAM package, also made by Autodesk. Fusion 360 is cross platform compatible, although it doesn't run very well on low-spec computers. It has a powerful CAM environment for machining your parts, and it has intuitive and easy cloud rendering that gives the heavy lifting to Autodesk's servers. To maintain a simpler UI, Fusion skips out on a lot of the more advanced features found in SolidWorks and Inventor, although this isn't really much of a problem. However, a more noteworthy difference is that Fusion ignores every single industry standard, creating its own structure and organizational system. Beyond simple sketches and extrude features, Fusion's modeling and assembly system is unique and not compatible with any other CAD software, making it hard to switch away from Fusion. Because of this, if you aren't careful, Fusion's file hierarchy can actively encourage bad design habits and discourages reusability by allowing users to create new parts

without designing them individually first. If one is careful to follow good design practices, Fusion is a solid option.

Creo (formerly known as Pro/E) A family of CAD/CAM applications developed by PTC (Parametric Technology Corporation). Creo parametric is the main CAD package that includes robust assembly and part modeling similar to Solidworks. The main advantage of Creo compared to other CAD software is the complex part relations and constraints, however, most new users find this aspect difficult to grasp completely. The Creo package includes an integrated local rendering engine and thorough simulation system. The rendering engine can be used to create photo-realistic renders of anything between single-part simple to multi-component complex designs. Design enhancements can be directly integrated into parts from materials/geometry simulations in Creo Simulate. Creo has many features, but the general 80-20 rule applies - 20% of the features will create 80% of the designs. The Creo package also includes a version based file sharing system called Windchill which most professional companies use, but for FTC purposes Grabcad will suffice. Learning Creo can be more of a challenge than other CAD software, because of the limited available tutorials online in addition to the complex relations and constraints structure. Our recommendation is to learn Creo from someone who already has experience with the software. College-level engineering classes as well as numerous companies in the automotive, aerospace and consumer industry use Creo. Since Creo makes use of Windows OS file system it will not run on Mac. However, a Windows emulator can be installed to run Creo on Mac.

After learning your CAD program of choice, determine the necessary requirements for the drivetrain based on the current game. Teams should shoot for the wheelbase that works the best in that specific field's layout. For instance, in Relic Recovery (2017-2018) a drivetrain required precision to not only grab glyphs from the center pit, but also to line up against the cryptobox. Thus, mecanum wheels and a wide center section of the robot proved an advantage over a 6 wheel tank drive. (However, it should be noted that with sufficient practice and competent drivers, any drive base can be competitive up to a certain extent).

After selecting a drivebase, determine the number of motors. Keep in mind the eight motor limit is a pain that shouldn't be ignored. A good rule of thumb is four motors for driving and four motors for the other mechanisms (e.g. intakes, linear slides, arm, etc.) For most modern FTC games, you need minimum 7 motors to be highly competitive, although 8 is a good rule of thumb.

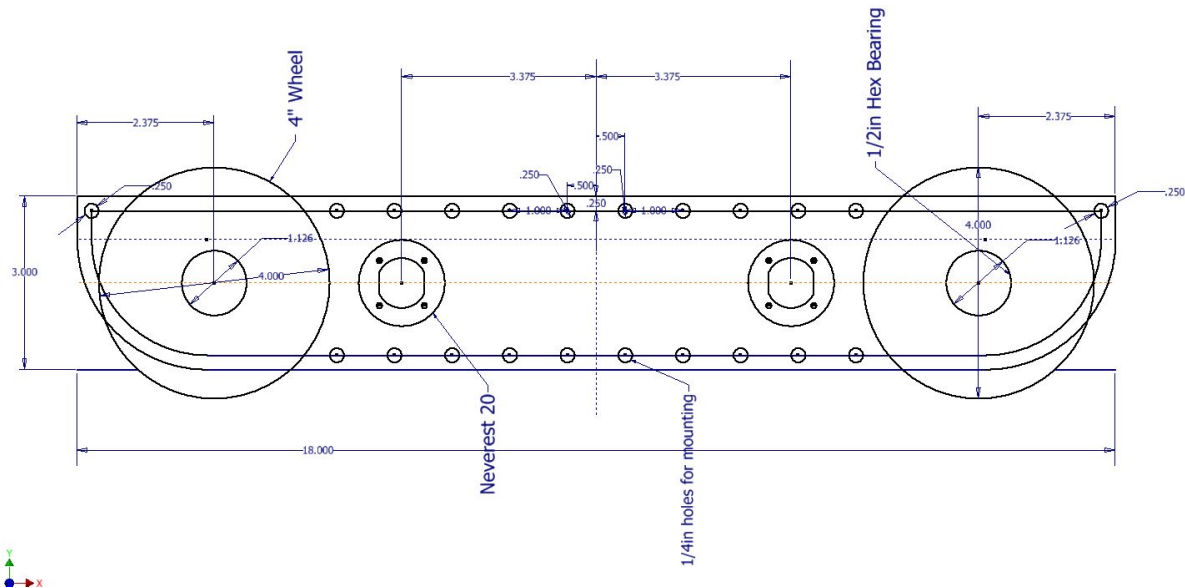
Designing the Drivetrain Plates

After learning the CAD software, it's time to start the actual design. Here are some things to figure out before starting:

- Drive Type (mecanum, 6wd, 8wd, etc.)
- Number of Motors (four motors recommended in most cases)
- Type of wheels (Colson, omni, etc.)
- Drive Power (belt, chain, gear)

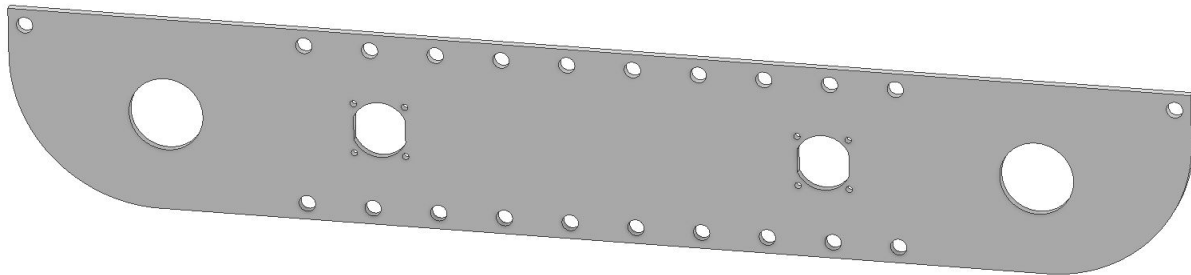
To keep it simple, this example uses a 4 wheel tank drive using four motors. The wheels selected are 2 Colson wheels for traction, and 2 omni wheels to aid in turning.

First, make the left side of the drivebase. After completing it, all you have to do is mirror the left side to the right, so you don't have to do each side individually. Start with a 2D sketch of everything before trying to extrude and make actual 3D objects.

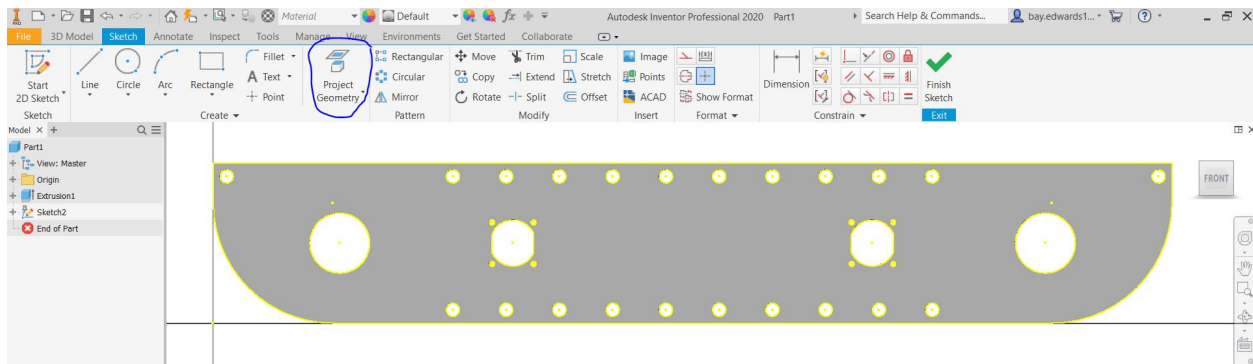


This is a sketch of the inner plate of the drive base. Everything should be laid out in a 2D sketch to determine the mounting holes, bores, center-to-center distance, etc. 2D sketches are extremely helpful and are highly recommended in any project. After the sketch is completed, everything else falls into place and becomes pretty simple.

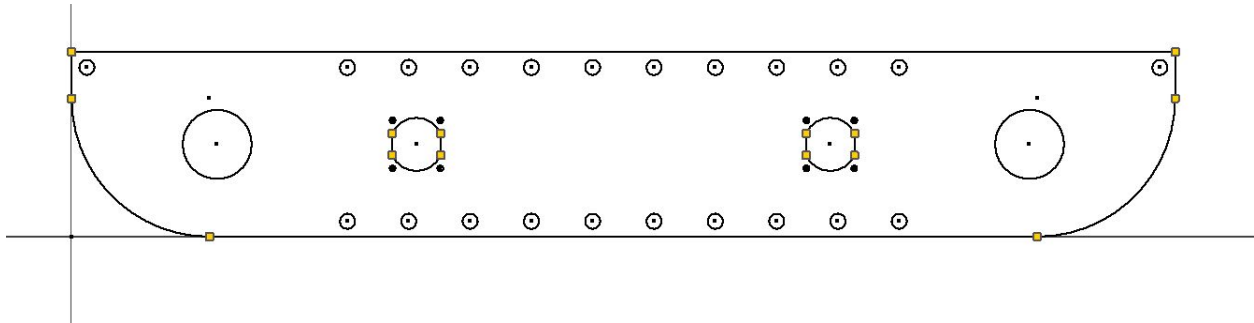
After this, extrude that sketch into the first plate of the drivetrain. Typically, a standard thickness of aluminum plate is $\frac{1}{8}$ ". Thinner plate ($\frac{3}{32}$ ") can be used as well, but generally most teams stick to $\frac{1}{8}$ ". Extrude the plate to that thickness. Below is the sketch after extruding.



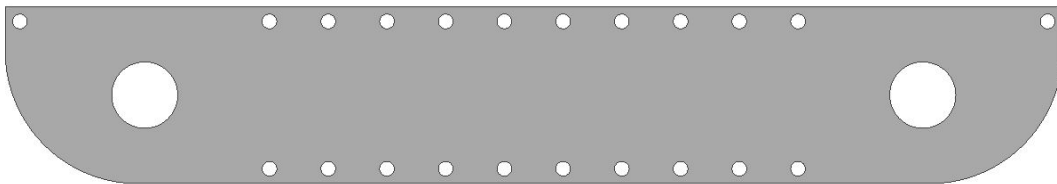
The next step will be making the outer plate for the drivebase. It is even faster to do than the inner. To do this, simply create a new part. Go back to your inner plate and start a 2D sketch.



After starting the new sketch on the inner plate, hit “Project Geometry” and just click anywhere on the part. It should highlight every outline of the part. (Shown here is a yellow line; yours might be red, blue or some other color.) Now click and drag across the part selecting every line on the screen. Now go hit CTRL + C, then go to the new part and hit create 2D Sketch. Next hit CTRL + V.



It should look like an exact copy of the inner plate but now as a sketch. Delete your motor mounts out of the middle, then extrude the outer plate.



This is what the outer plate looks like, an almost exact copy of the inner one without the holes for the motors. Now with those two plates made, it's really just time to assemble the rest of the drivetrain, which is by far the most time consuming. Now, for some info on what to use to attach the two plates together, generally standoffs or churro is highly recommended. To attach the two halves of the drivetrain, use either channel, extrusion, or a custom u-brace. Some teams prefer a custom brace as it is a good way to stiffen up the drivetrain while requiring very little maintenance over the season. It is possible to use peanut extrusion or kit channel, which alternatively works just as well.

Note that when using a custom drivetrain, you can cut out material from your drivetrain plates. This process is called **pocketing**. While not a vital step, pocketing helps you save weight. However, be careful not to remove too much material; if done, the plates become less sturdy. More about pocketing is in the next section.

Additional Considerations

Powering wheels can be done in a couple different ways through either belts and pulleys, chains and sprockets, gears, or even powered directly from the motor. Direct drive and chains are the simpler of the options, with direct drive not needing a calculated distance at all just have to set the motor exactly where the center of the wheel is. Chains allow for a little bit of slack not needing an exact center to center distance in the wrap like belts and pulleys do. Finally gears which need to be a certain distance apart from each other to mesh properly and not skip or bind.

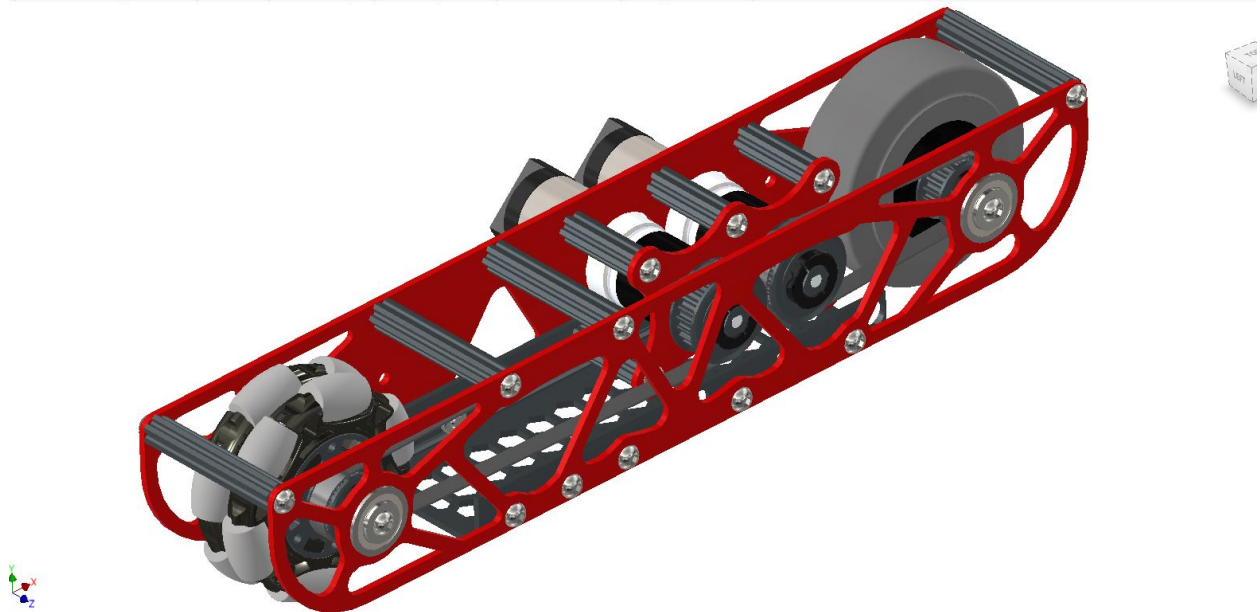
Mounting motors is done in a plate style by face mounting the motor into the innermost drivetrain plate. It can also be done by mounting the motors to a 3rd plate, located in between the outside and inside. This

allows for the motor to take up less space in the middle of the robot, but adds complexity. Motors should always be as low as possible and depending where you want the center of mass, either the middle or towards the back of the robot. It is also worth keeping in mind the type of power transmission and the expediency of doing so in light of the motor placement.

Ground clearance is all dependent on if there are any obstacles on the field, as well as what your team wants to do in that game in regards to said obstacles. For example, in Rover Ruckus some teams with tank drivetrains decided to enter the crater. Therefore, they left enough space to not beach themselves on top of the crater, a common mistake that inexperienced teams often make. Other teams decided to ignore driving over the crater and decided to reach over with an arm or slide system, which meant they didn't need a lot of ground clearance for their drivebase. Typically, anywhere from .25 inches of clearance to .5 inches (if you want to be safe) on a completely flat field will allow for the weight of the robot to push into the foam tiles. Nothing else from the robot should touch the ground. Something you can do is set the robot in CAD onto a field. Set up obstacles such as the crater and simulate driving over the crater by moving it across like you think it would in the real world. If either of the plates intersect with the obstacle, add some more clearance so you don't get beached like a sad whale. A general rule of thumb for most teams is the wider the intake, the better the chance of picking up the game piece. However, this is super game dependent. If you need to pick up a 6" cube like in Relic Recovery then you would not need 14" of space for your intake. However, if you need to pick up a ball like in Velocity Vortex, the bigger the intake gives you better chances of grabbing the balls. Keep this in mind when designing drive pods - try to keep them as thin as possible without sacrificing rigidity and strength to maximize space for other mechanisms and wiring.

Connecting your two plates together is really simple. Some standoffs or churro extrusion from AndyMark is a relatively easy way to connect them together with a few bolts. Just make a few 1/4 in. holes in your sketch where you want the churro tube to be. Decide how long the churro needs to be. Remember to leave enough space between the plates for your wheels, pulleys, sprockets, and spacers. You don't need to go overkill on how many standoffs you need in between your plates; however, put them in strategic places where support is needed.

Shown below is a drive pod, which is one half of the drivetrain, including the shafts, bearings, wheels, motors, belts, etc. In short, the drive pod has everything that will be built in real life. This particular one is the left side, but to make the right side create an offset plane, select the mirror tool, then hit mirror.



After mirroring the drive pod to make your opposite side, connect those two halves together and you're done with the drivetrain. Below is a rendering of the complete drivetrain in CAD.



CAD Tutorial Part 2 - Pocketing Guide

“Pocketing” is a common term in FTC and FRC lingo, and refers to material removal from plates or tubes for weight reduction and/or aesthetic purposes. For the former purpose, the idea is to strike a balance between sustaining structural integrity and reduce as much weight/material as possible. There are several ways to machine pockets into material including milling, routing, water jet cutting, laser cutting and even hand drilling. Depending on your access to tooling, pocketing can be more or less difficult for you. CNC milling and routing excel at pocketing aluminum box tubing, whereas water jet and laser cutting excel at pocketing plates. Whether pocketing on box tubing or plates, the design is fairly similar.

When designing pockets, **it's important to consider the type of material, thickness, and how much stress will be on the part.** Materials that are weaker, thinner or under significant stress should have less “aggressive” pocketing and materials that are stronger, thicker or under less stress can have more “aggressive” pocketing. Aggressive pocketing refers to the amount of material removal from the blank part (more aggressive = more material removal). Although a bit complex to understand, FEA (finite element analysis) can be used to determine appropriate strut thickness when pocketing. FEA can be used to generate pocketing geometry, but that is an entirely different rabbit hole.

Designing concise and advantageous pocketing is as simple as drawing circles and tangent lines. Parametric pockets can be defined by one or two offset values. The offset values determine the thickness of the remaining material. Parametric means that the entire sketch is defined by a parameter, in this case is the offset value which when adjusted will automatically adjust the entire sketch (in terms of material thickness). There are several references that can be drawn on every plate/tube which are screw holes, bearing holes, and corners. Each reference will get its own construction/sketch circle or two. Ideally all of the construction circles are one of less than 4 sizes to keep the pocketing consistent and simple. First are the screw hole construction circles with radius of the screw hole radius plus the offset value. Next are bearing holes with radius of bearing hole radius plus offset value. Then are edges with construction circles with the radius of an offset value. Then the most important circles are at each of the screw and bearing holes, which will define the strut thickness. The circles at the center of each screw and bearing hole will have the diameter of an offset value. After all of the construction circles are drawn, tangent lines can be drawn to create the

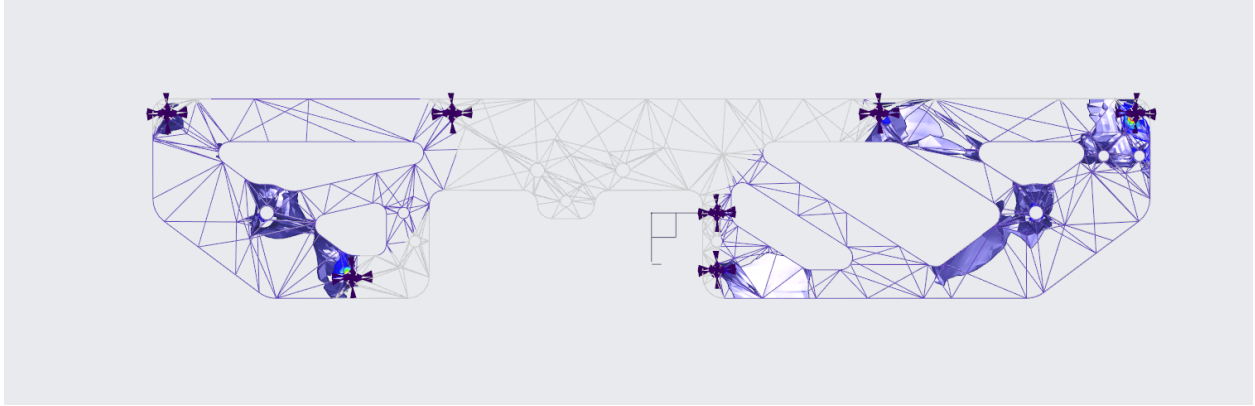
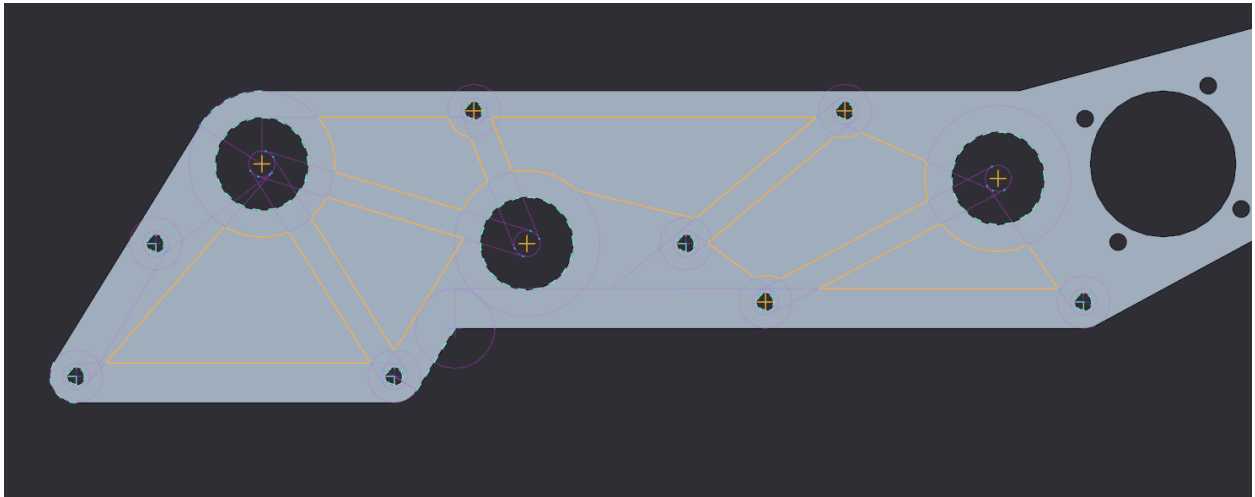


Fig. 33: 731 Wannabee Strange, Rover Ruckus, FEA of inner drivetrain plate

pocketing geometry. Using the parametric offset value will make it easy to adjust strut thickness by just changing one or two values.

Tangent lines are drawn between the circles on the edges with other circles on edges and between the circles at the center of each bearing and screw hole. The circles with radius of bearing hole and screw hole plus offset value make sure that there is enough material around the bearing and screw holes. An example is below.



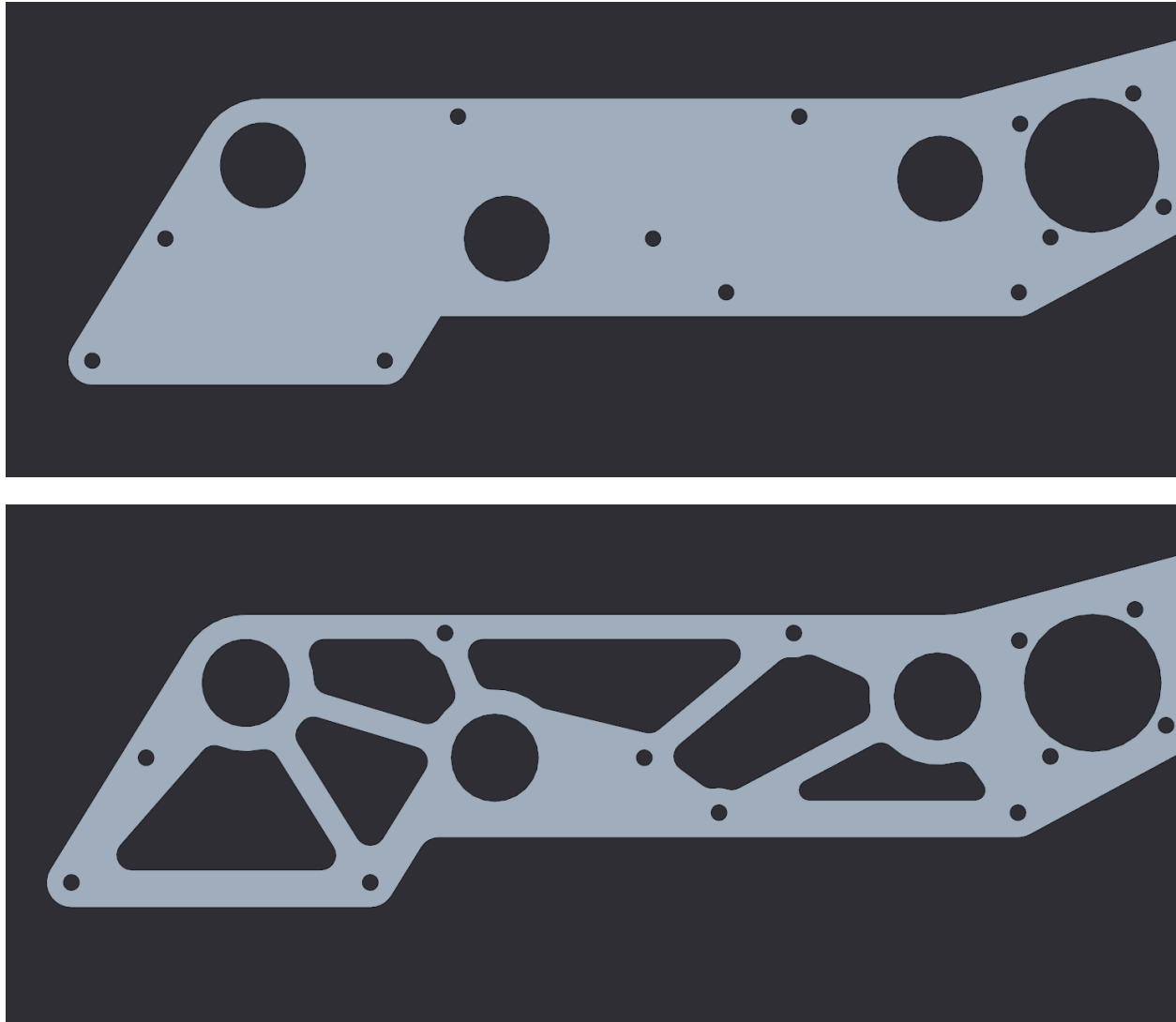


Fig. 34: 731 Wannabee Strange, 2019 Summer VCC Cadathon, Outer Mechanism Plate

The last step in pocketing is adding rounds to each and every corner, especially inner corners. Rounds relieve stress buildup at corners and make it easier to machine. Some machines, such as mills and routers, are also unable to machine tight internal corners. For those parts that need minimum rigidity loss and a lot more machine time on their hands, pockets don't need to be cut all the way.

Waterjet cutters and laser cutters are only able to cut material all the way through, but routers and mills are able to make surface pockets. These pockets don't go all the way through the material and are multitudes more rigid than thru pocketing. The downside is increased machining time. The increased time is from the "lawn mowing" tool cutting path versus simply cutting the edges of the geometry. It is also more difficult to machine, because more material is milled out and chip ejection becomes more important.

If you don't have access to any precision tools, a hand drill/drill press and large drill bit/flat bottom boring bits can create pockets in material. Although this is the simplest form of pocketing, there is a straightforward way to optimize the circular drill method. Since the main goal of pocketing is to remove as much material as possible without significantly sacrificing the structural stability, the holes need to be drilled in specific positions with the right size bit. The most effective way to find the specific positions and drill bit sizes, is to first create a pocketing design as you would do with circles and tangent lines. Then draw holes tangent to the struts created by the circles and tangent lines. An example is below with the orange as the holes to drill

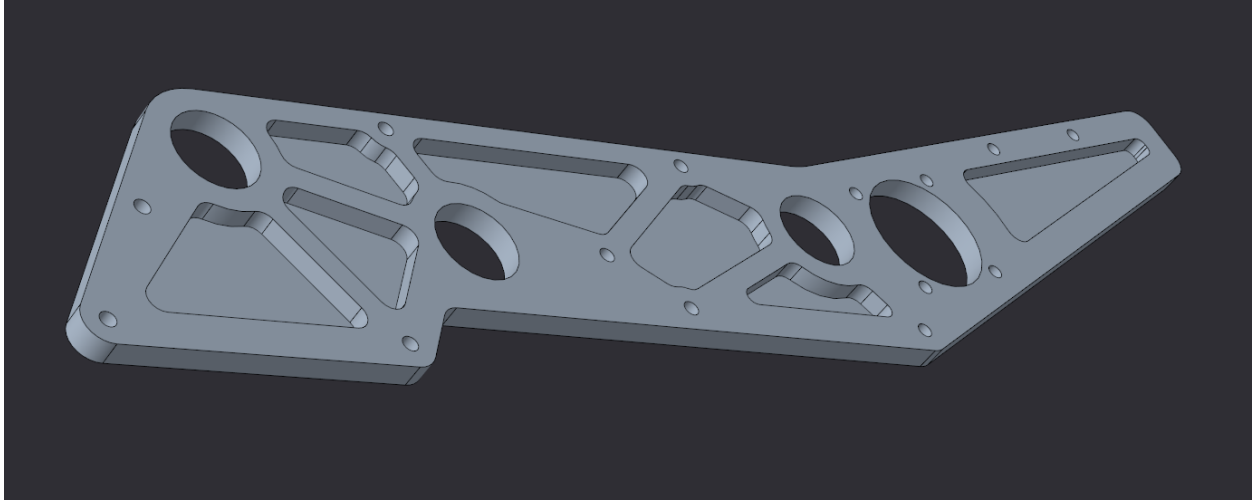


Fig. 35: Surface Pocketing Example

positioned tangentially to the regular pocketed edges.

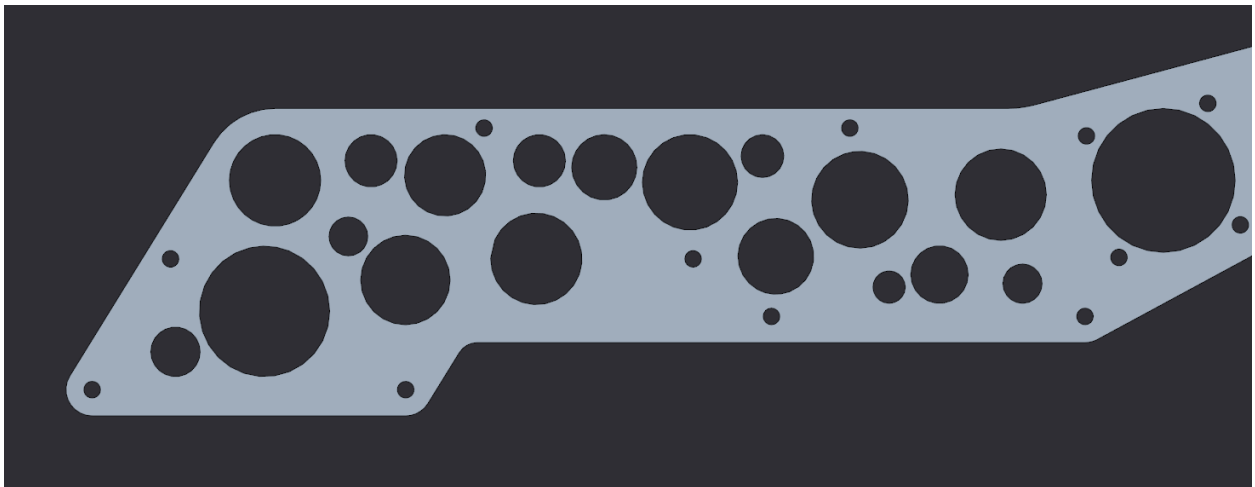
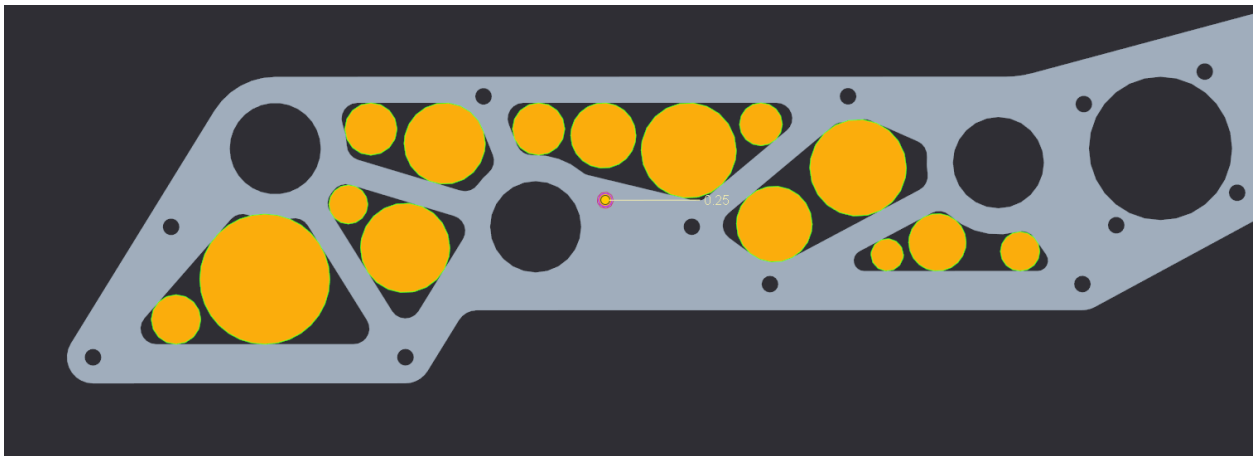


Fig. 36: Optimal Drill Pocketing Method Example

Although it may look like a random mess and it may take a while longer than just randomly “cheese holing”, this method will yield the greatest weight reduction to structural rigidity loss ratio using the drill pocketing method.

A very important tip to pocketing is to do it last when designing a part. Parts should not be designed around the pocketing pattern, rather the pocketing should be designed around the part. If there are too many holes in a part, or the part is too small to be pocketed with an offset value, then it’s probably not worth it to pocket.

Pocketing can reduce part weight, but when using traditional machining methods can take a significant amount of extra time. Although, when adding pockets to parts that are going to be 3D printed, it can in some cases decrease print time as well as material used. The pocketing method above is the simplest parametric method to pocketing, but more complex methods exist. For instance, the image below is an example of a complex double iso-grid pocketing pattern optimized for metal 3D printing.

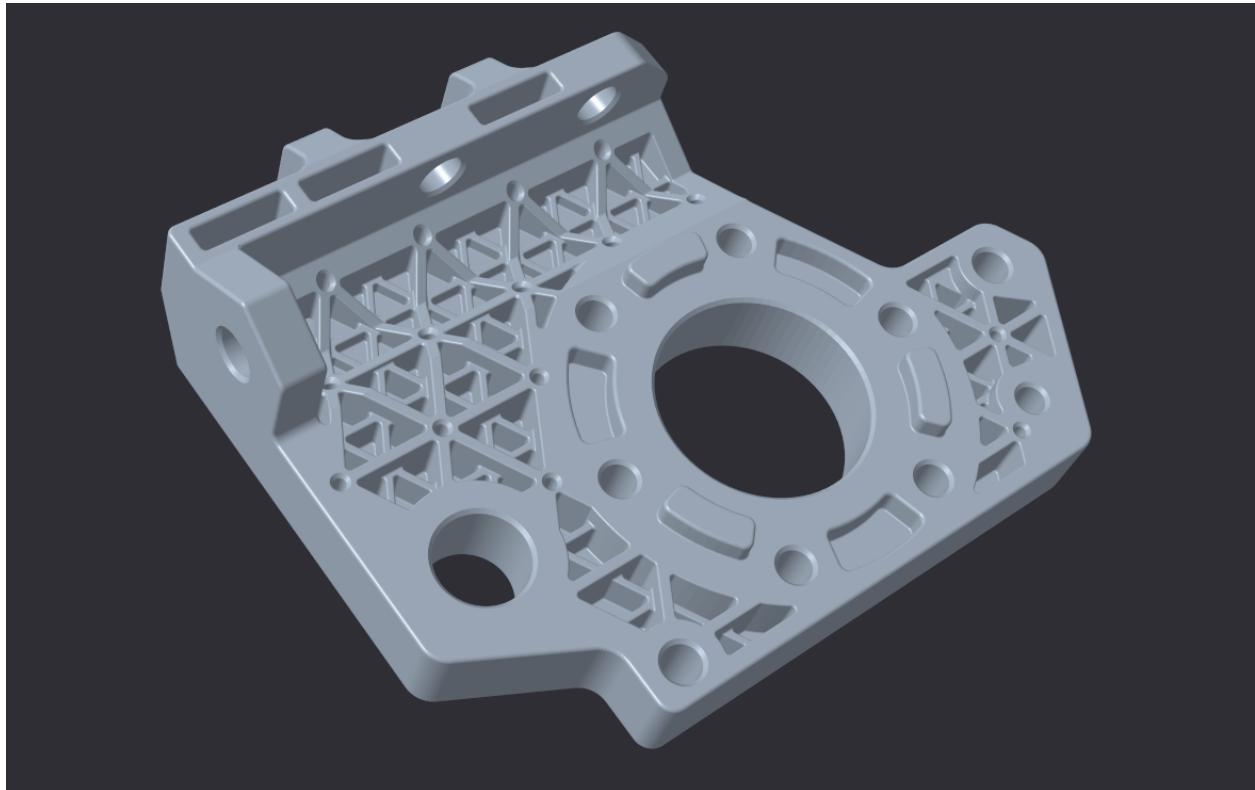


Fig. 37: 731 Wannabee Strange, Rover Ruckus, Arm Pivot Mount

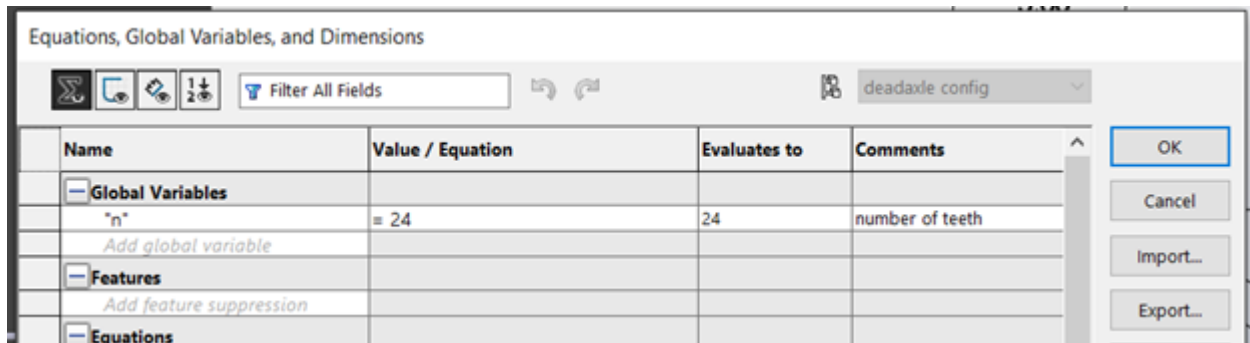
When the pockets are designed around a 3D printed part, many new possibilities open up in terms of minimum inner corner radius, resolution and dimensions. Now of course, 3D printed parts can be pocketed in the same way as traditional parts with similar results.

CAD Tutorial Part 3 - Custom Pulley Template

When designing methods of power transmission, it’s useful to have an adjustable pulley generator to rapidly rearrange C-C (center to center) distance for design changes. Typically, FTC teams use HTD5 5mm belt profile due to its deep tooth profile, which adds resistance to slipping and increases load capacity. This tutorial will focus on the HTD5 5mm profile, but it is relatively easy to adapt for different profiles.

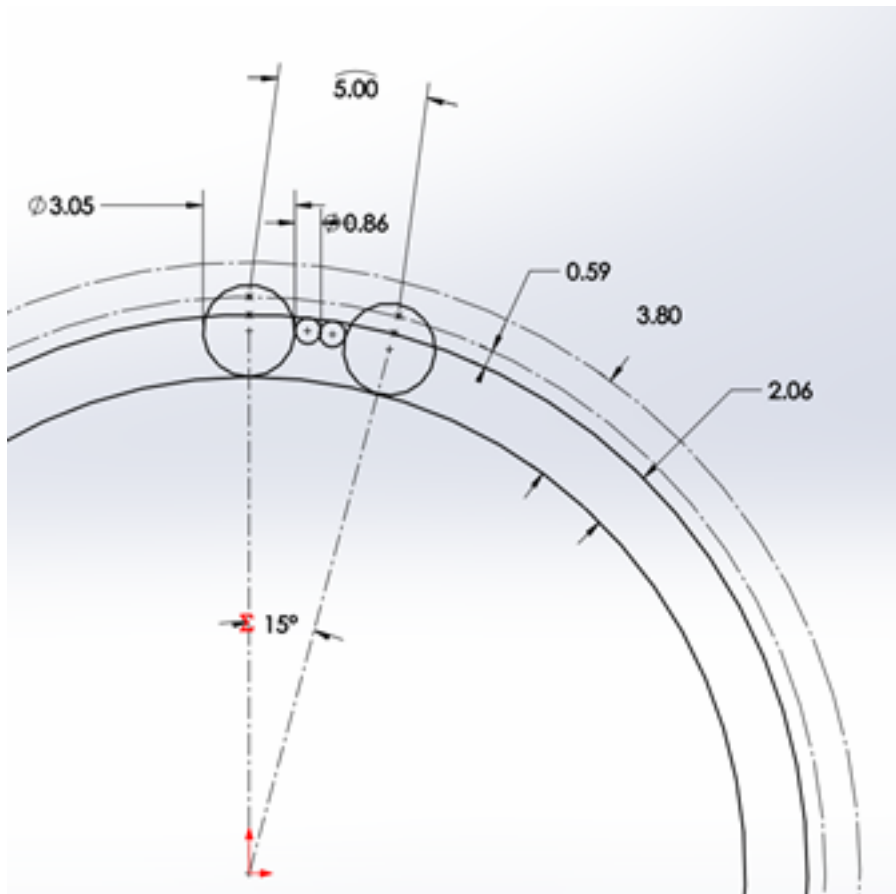
To make the pulley fully parametric (adjustable without redoing the base sketch), we will use Equations (in Solidworks and Creo) or Parameters (Fusion 360 and Inventor). Equations allow a user to quickly adjust

values and change multiple dimensions in a sketch or feature.



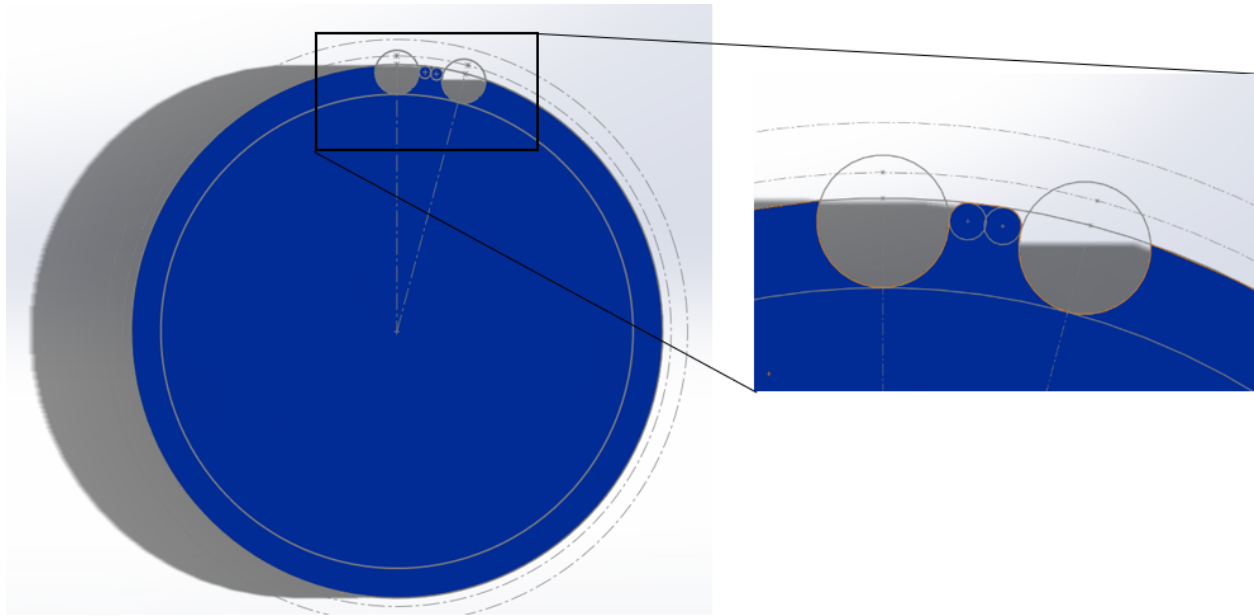
First, define a new variable "n" and set a default value of 24. This is crucial since "n" will affect the number of teeth – which will define the angle between teeth and the circular pattern.

Copy the sketch below.

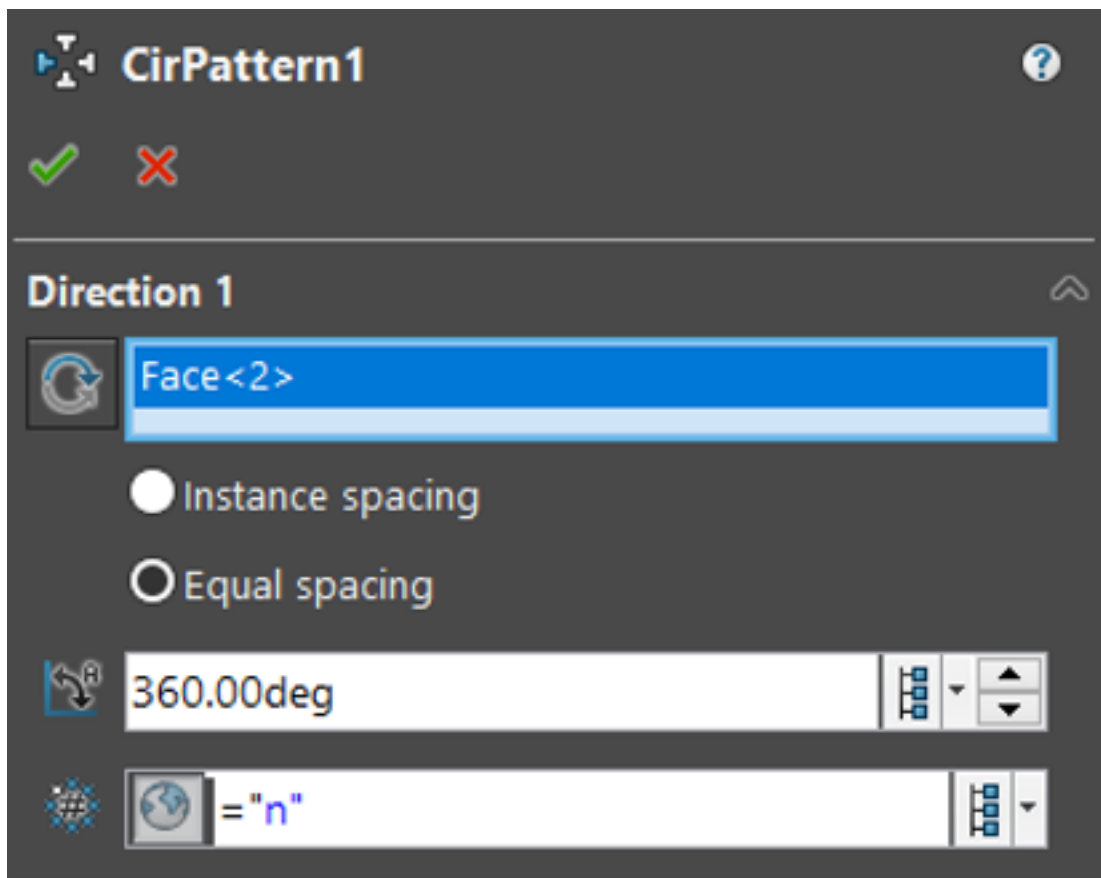


- The 15° equation is done by typing $=360/"n"$ into the text box.
- Note that 5mm dimension at the top describes arc length, which is done in Solidworks by first selecting the two points and the connecting arc.
- The two big circles are tangent to the two smaller circles, but the two smaller circles are not tangent to each other.

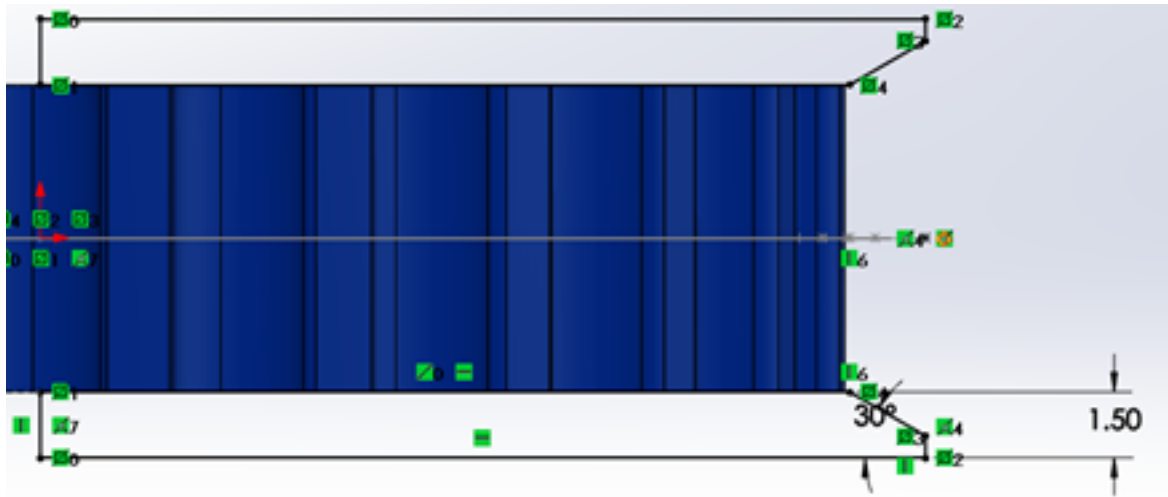
Leave this sketch as a reference and use "Convert Entities" to create sketches for additional features.



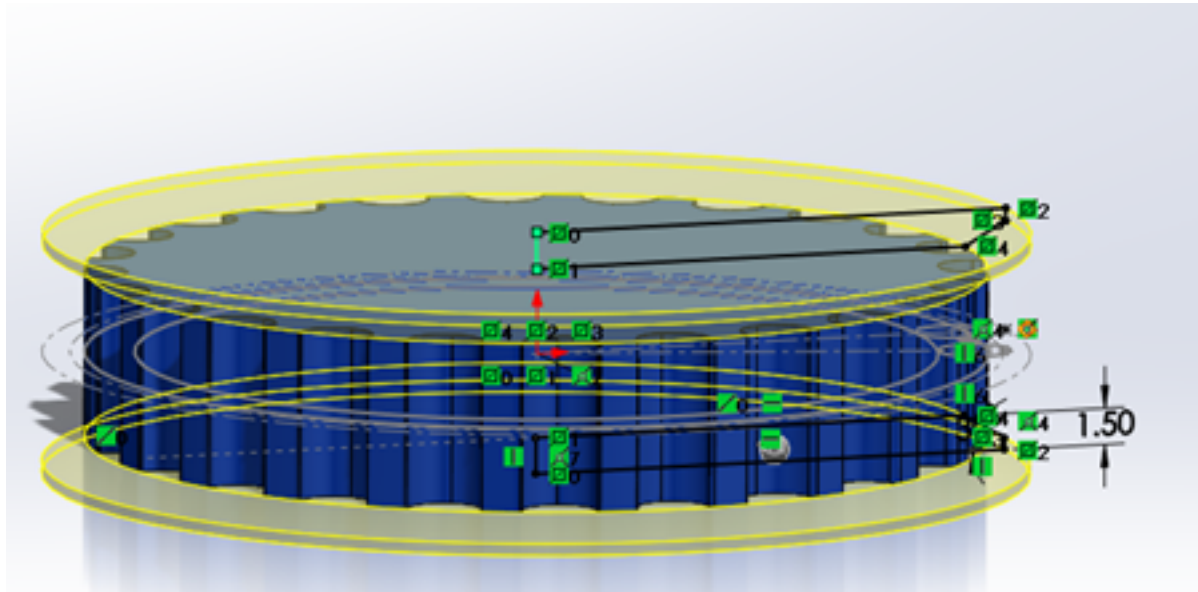
Next, extrude the outer bold circle. Cut-extrude the profile in the reference sketch. Do these features separately.



Now, just create a Circular Pattern. Define Direction 1 as the top face and create "n" instances of the cut-extrude feature.



Now just sketch on the side plane and sketch the flanges. This is up to you, but I prefer to keep the outer point vertical to a point pierced through the outer circle. That way, the flange changes with respect to “n”.



1.8 Power Transmission

When building any mechanism, it is important to consider how the mechanism will be powered, as gear ratio and form factor are two factors that affect both spacing and efficiency. There are four main forms of power transmission: direct drive, gears, chain, and timing belt.

1.8.1 Direct Drive

Direct drive refers to powering a mechanical part (most commonly a drive wheel) directly from the motor axis. Many new and inexperienced teams will use this method to power their drivetrain as it is the simplest way to do so. However, there are significant drawbacks to this method.

First, direct drive puts unnecessary load on the drive motor. This is because shock loads can destroy the gearbox, even planetary gearboxes. Gearboxes are able to withstand load along the axis of rotation, such as what occurs when the wheel changes direction. This is a normal situation of load. However, in direct drive, the gearbox shaft can be exposed to other shock loads outside of the normal axis. This happens when the wheel comes into contact with another robot or the field wall, which honestly happens more than you'd think. This can bend the motor shaft or permanently damage the gearbox.

Second, direct drive limits the gear ratio to whatever ratio the motor gearbox is at. One advantage of all the other three transmission systems is the ability to gear up or down, based on a team's needs. Direct drive cannot do so, and if your drivetrain uses 40:1 gearboxes, there is no way to reduce this ratio to a faster 20:1 gearbox, for example.

Third, motor shafts are not built to carry large amounts of downward load. This can lead to bending of the motor shaft, since in direct drive the shaft is only supported by one side, the gearbox. Generally, a principle among all engineers is to support the shaft on both sides, which isn't possible in this case. This leads to cantilevering the shaft, something that should be avoided in general. Thus, with a heavy robot the motor shafts can easily be bent due to a lack of support. (While typically it is impossible to support a motor shaft on both ends, it is a common rule of thumb to have dual support on wheels or other non-motor shafts.)

Advantages

- Saves space
- Easy to build; most simple form of transmission

Disadvantages

- Prone to shock loads which destroy the gearbox
- Limited gear ratio to the motor itself
- Wears the gearbox of the motor faster
- Can bend the motor shaft

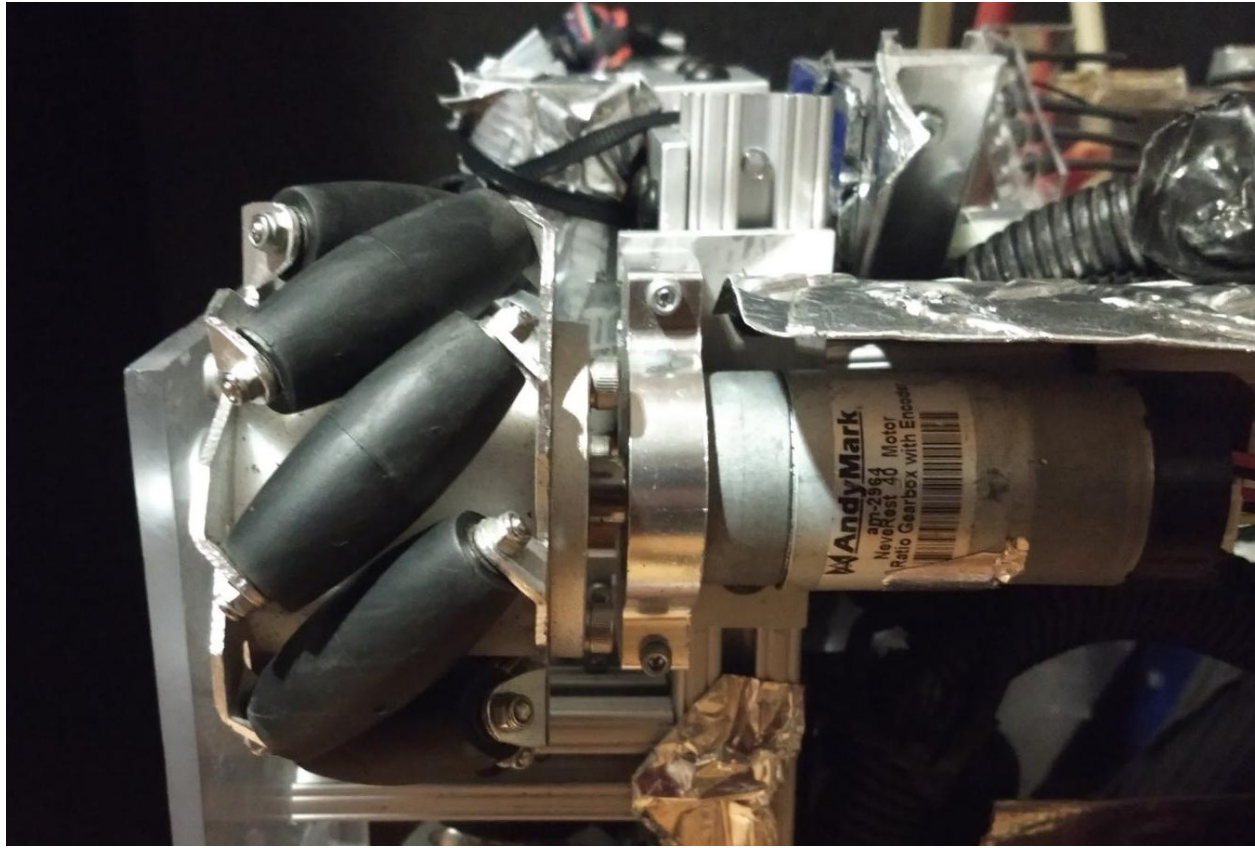


Fig. 38: *Outside Supported Direct Drive (Preferred Version of Direct Drive)*
11316, Weapons of Mass Construction

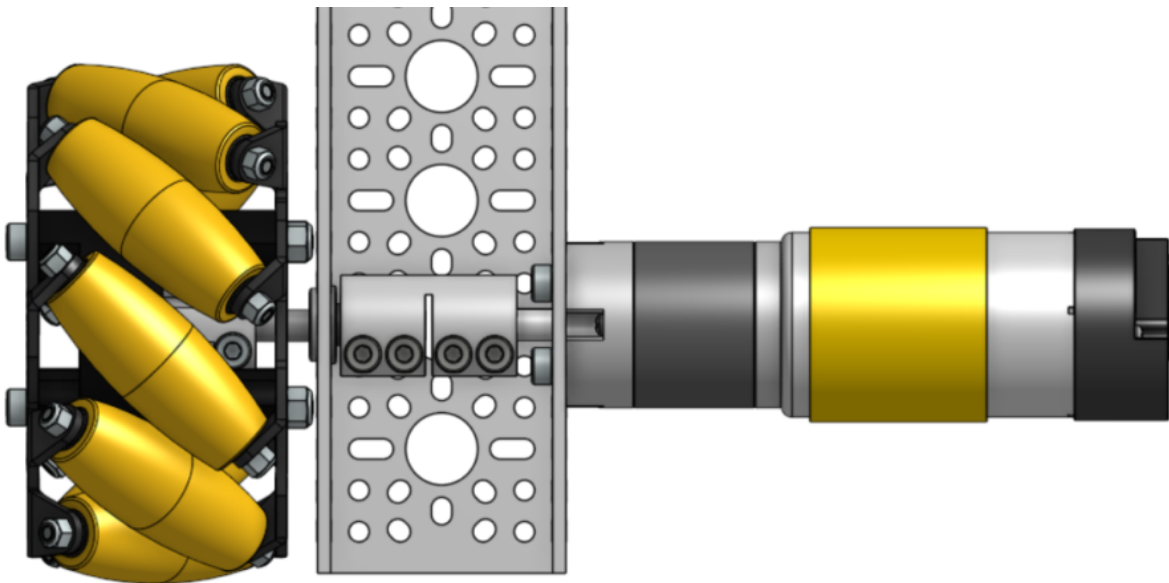


Fig. 39: *Inside Supported Direct Drive (Preferred Version of Direct Drive)*
Ethan Doak

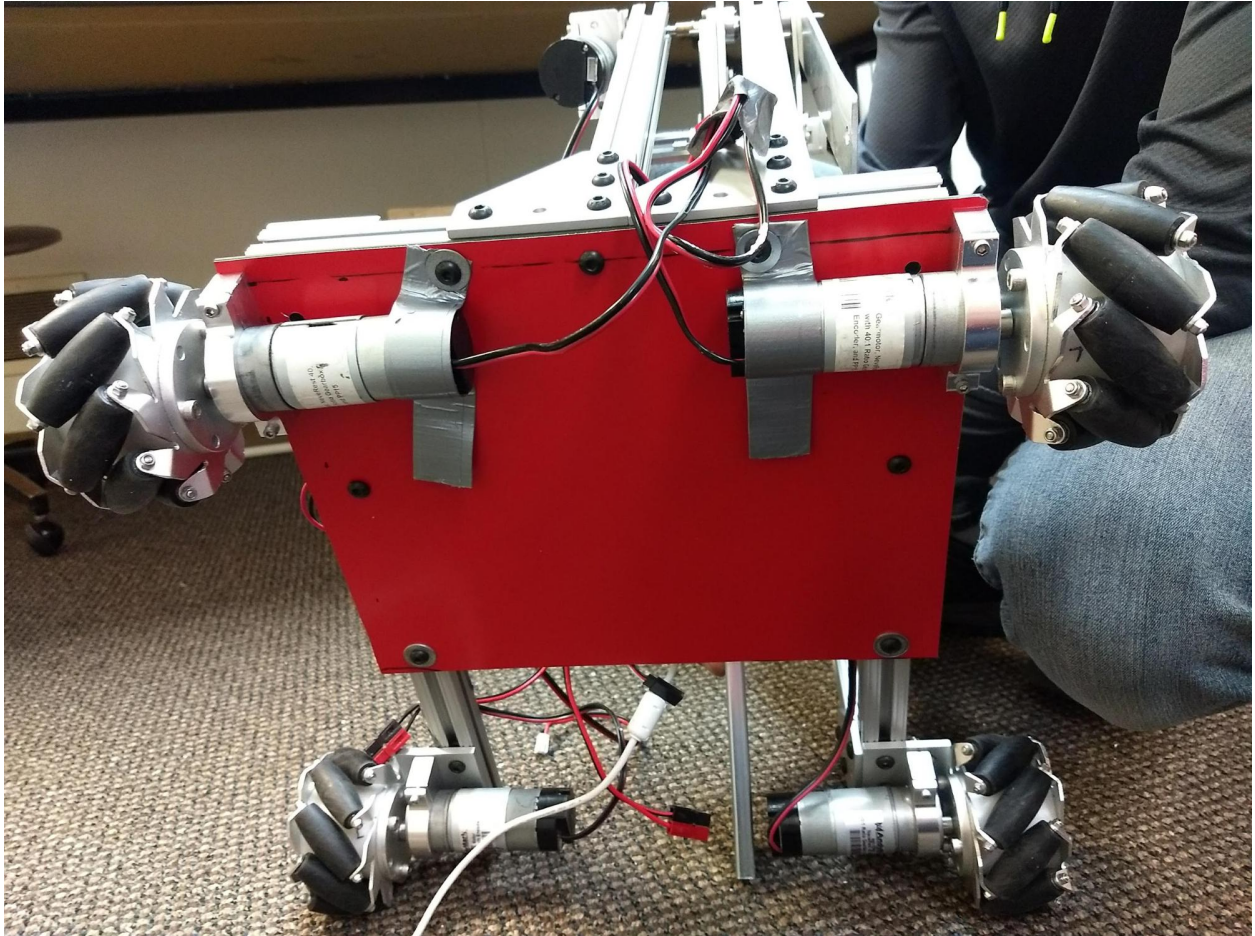


Fig. 40: Unsupported Direct Drive (**not recommended**)
11316, Weapons of Mass Construction

1.8.2 Gears

Gears, like sprockets and pulleys, are used in power transmission for three common applications: changing the direction of power, changing the amount of torque, and changing RPM. Gears are a less common transmission option than chain, but are still very viable for most use cases. Gears are equally as reliable as chain, but can't be used for transferring power over long distances. Many teams dislike gears when using kit-based channels because the hole pattern limits which combinations of gears can be used. Consequently, it may be difficult to achieve a desired specific gear ratio. However, with extrusion systems, it's very easy to use different ratios, since the gears can be moved around in extrusion.

Gears are made in different materials, with the most common being 7075 aluminum. **Never mesh plastic and metal gears together.** It is acceptable to mesh different types of metal gears together, as long as they have the same diametral pitch. It is advised to stay away from TETRIX aluminum gears as they wear down very easily. Some REV gears are made out of Delrin, a self-lubricating plastic. It is a durable material, but keep in mind that it is very possible to strip the bore using a plastic gear. Thus, we advise using the REV Hex Hub Strengthener to avoid stripping the bore on Delrin gears.

Bevel gears are a special type of gear that allows power transmission in two different planes. It is especially useful in tight spaces where a regular motor mounting position would not work.

Terms

- Module refers to the amount of diametral pitch in millimeters per tooth. Simply, it refers to the tooth size. Typically, gears of the same material from the same manufacturer should have the same module.
- Diametral pitch (DP) refers to the gear tooth size. It is expressed as the number of teeth per inch of pitch diameter. Teeth size varies inversely with DP.
- Pitch diameter (PD) refers to the diameter of the imaginary pitch circle that is drawn when the gear is meshed with the pitch circle of another gear. **For gears, PD < OD.** * PD = Module * number of teeth * PD = Number of teeth / DP * PD = OD - 2 * Module
- Outer diameter (OD) refers to the actual outside diameter of the gear. Simply, it is the circle that the outer face of the tooth draws when it rotates around.
- Pitch circle refers to an imaginary circle that contacts the pitch circle of any other gear with which it is in mesh. Basically, each gear has a pitch circle. When gear 1 is meshed with gear 2, the pitch circles of both gears should touch exactly in the middle of where the teeth interlock with each other.

Meshing Gears

Be sure to never mesh gears that are not of the same diametral pitch (A notable exception is 32 diametral pitch and 0.8 MOD gears. These are close enough to be perfectly fine.)

When meshing gears, it is important that the gears are not too loose nor too tight. If the gears are too loose, the teeth will easily wear out, decreasing its longevity. If the gears are too tight, however, they will have too much friction and possibly grind or bind up. The ideal way to mesh gears are to make sure the teeth interlock and just touch the base of the gear. **If possible, it's best to avoid meshing gears with a clamping motor mount – due to the sensitivity of the mesh, even the slightest movement of the motor inside the clamping mount can cause the gears to slip or damage each other.**

Calculating center-to-center distances using gears is quite simple. In order to calculate the desired center distance between two given gears, you must know the number of teeth for each gear and the **diametral pitch** of your gears (the number of teeth per inch of the gear's diameter). With these two pieces of information, you can use the equation $D = \frac{T_1 + T_2}{P}$. In this equation, D is the distance between the center of both gears, T_1 and T_2 are the number of teeth of each gear in question, and P is the diametral pitch of the gears.

The **module** (abbreviated **MOD**) of a gear is used similarly to diametral pitch. It is the number of millimeters of the gear's diameter per tooth of the gear. The equation to find center distance D is $D = \frac{(T_1 + T_2) * M}{2}$, where T_1 and T_2 are the number of teeth of each gear in question, and M is the module of the gear.

Additionally, it is possible to average the pitch diameters of the two gears to find the correct center-to-center distance.

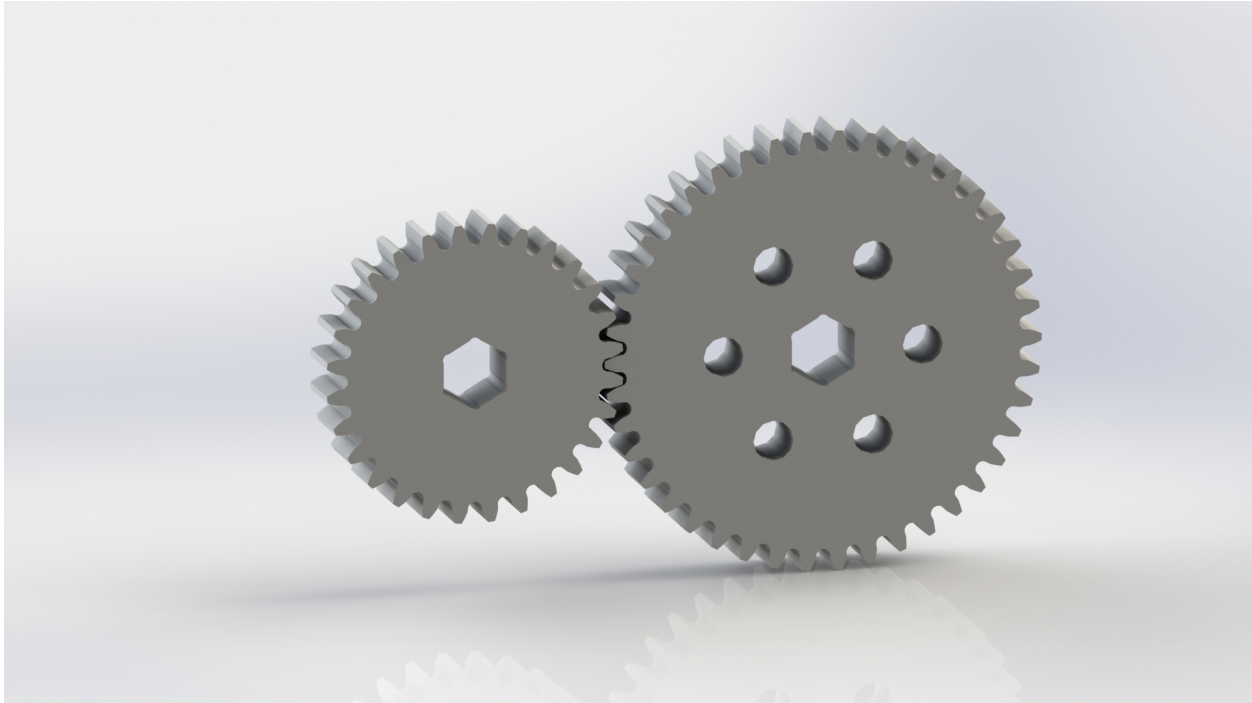


Fig. 41: Correct gear mesh

As with sprockets, it is important to line up the gears so that they do not accidentally slip. Especially when using extrusion, it is possible that the gear may not be parallel to the extrusion, as the two supporting ends may not be perfectly in line with one another. It is imperative that the gear be lined up as straight as possible to prevent damage or gear binding.

It is highly recommended to use white lithium grease or a similar lubricant between the gears to reduce friction and possible binding.

Advantages

- **Gears are a solid and proven power transmission method.** Early examples of gears date back to the 4th century BC, so you're using technology with millennia of development behind it. When it comes to gears, there's not much that we haven't figured out.
- **Gears are simple to use with both channel and extrusion.** On channel, your gears are already spaced correctly - you just need to choose the right pair of gears. Extrusion gives you even more flexibility - just slide your gears into mesh, and you can have whatever ratio you want.
- **Gears can give you big reductions in small areas.** Depending on the gear combination, one can achieve big ratios in reduction in very small spaces (for example, a 10 tooth gear and a 100 tooth gear will take much less space than a 10 tooth sprocket and a 100 tooth sprocket).

- **Gears require no tensioning. Once the spacing is correct, the gears will operate quickly.** Unlike chain or belt, there is nothing further transferring the power, which cuts out the need to properly tension chain or belt. This of course has the drawback of not being able to transfer power far distances.

Disadvantages

- **Sometimes, the ratio you want might not be easy to build.** Channel spacing limits gear ratios, but this can be circumvented with compound ratios and a bit of creativity.
- **Long distance power transfer is impractical with gears.** If you need to transfer power long distances, gear combinations can become complicated very quickly, so belt/chain is preferable.
- **Meshing gears can be tricky.** It's only made worse by the sensitivity of a gear mesh. However, channels do solve this problem, providing pre-spaced holes to easily mesh your gears. Do keep in mind that gear mesh may not be perfect, even with channel.
- **Gears usually wear faster than sprockets** if there is too much friction between the gears. Teams can use white lithium grease or similar lubricant to help remedy this problem.

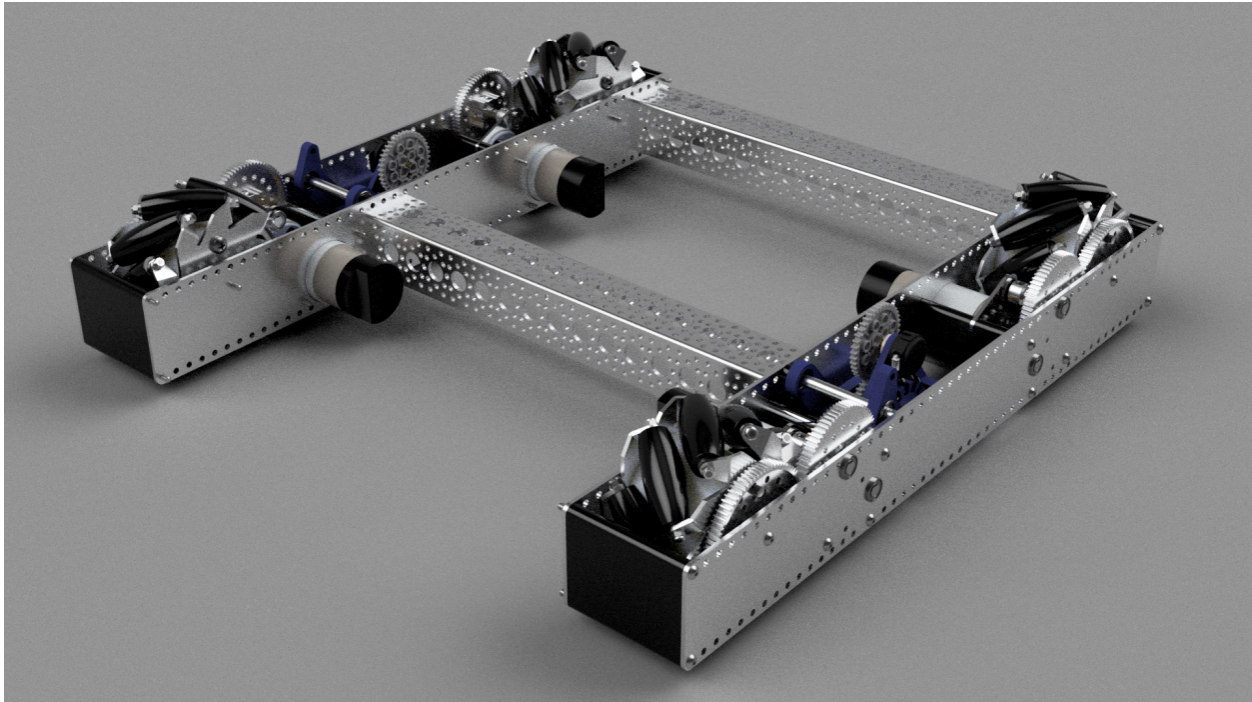


Fig. 42: 11115 Gluten Free gear-based drivetrain



Fig. 43: 13075 Coram Deo Robotics, Rover Ruckus gear-based drivetrain

1.8.3 Roller Chain

When your shafts aren't right next to each other, roller chain and sprockets will allow you to transmit power securely between your shafts. Roller chain is made up of a series of links joined by pins. Each link can rotate around its pins, creating a dynamic loop that can conform to any shape. The pins in the chain engage the gaps between teeth on each sprocket. If you've ridden a bike, chances are that you've already seen roller chain - the chain on your FTC robot is similar, but it's probably a different **pitch** (different size). Chains most commonly used in FTC are #25 or 8mm pitch.

When using chain, often there is a master link. This is a special type of link that has a removable end capsule in order to shorten the chain. However, as it is removable, it is not a very reliable chain link and can loosen and fall off under prolonged usage. There have been teams who have had master chain links fail during competition, costing them a match in the elimination rounds. A chain breaker eliminates the need for master links because it can break and join chain at any point. Chain breakers do so by pushing out the pins in the link and re-inserting them back.

It is highly recommended that teams purchase a chain breaker (we recommend the DarkSoul chain breaker) instead of using master links, which are prone to failure.



Fig. 44: The removable master link is shown on the right.

Terms

Sprockets

- Chain number refers to the type and size of chain that is compatible with the sprocket. Typically, #25 chain is used in FTC, though there is also 8mm chain.
- Chain pitch refers to the distance between the pin centerlines in a chain link.
- Pitch diameter (PD) refers to the diameter of the imaginary circle that passes through the center of the pins when the sprocket rotates. **For chain, $PD < OD < CCD$.**
- Outside diameter (OD) refers to the diameter of the circle that is generated by the top faces of the teeth when the sprocket rotates.
- Chain clearance diameter refers to the diameter of the circle generated when the chain is fitted over the sprocket. It is larger than both PD and OD.

Roller Chain

- Outside link refers to the two outside plates connected by two pins. The pins are inside the bushings and can rotate freely.
- Inside link refers to the two inside plates connected by two bushings. The bushings hold the pins in the outside link.
- Pitch refers to the distance between the center of the pins. The most common pitch is #25, or 1/4" inch.

Center-to-Center calculations

The equation to calculate center-to-center for chain is quite complicated. Many [online calculators](http://www.botlanta.org/converters/daile-calc/sprocket.html)¹³ can calculate C-C distances without going through the tedious calculations. However, the complete formula is

¹³ <http://www.botlanta.org/converters/daile-calc/sprocket.html>

below.

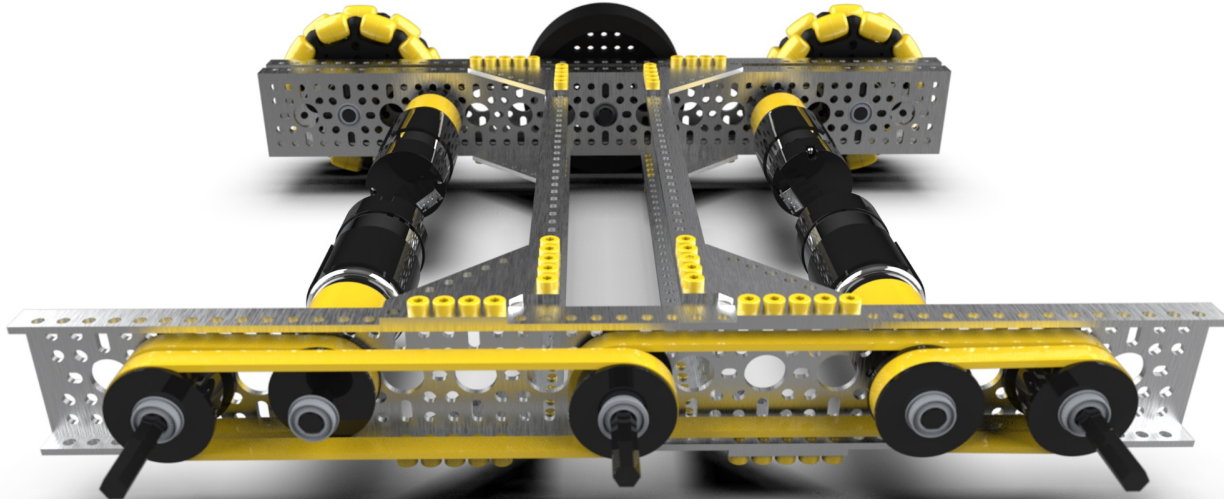
$$C = \frac{P}{8} * '(2L - (N + n) + \sqrt{(2L - (N + n))^2 - \frac{8}{\pi^2} * (N - n)^2})$$
$$L = \frac{2C}{P} + \frac{N + n}{2} + \frac{P(\frac{N-n}{2\pi})^2}{C}$$

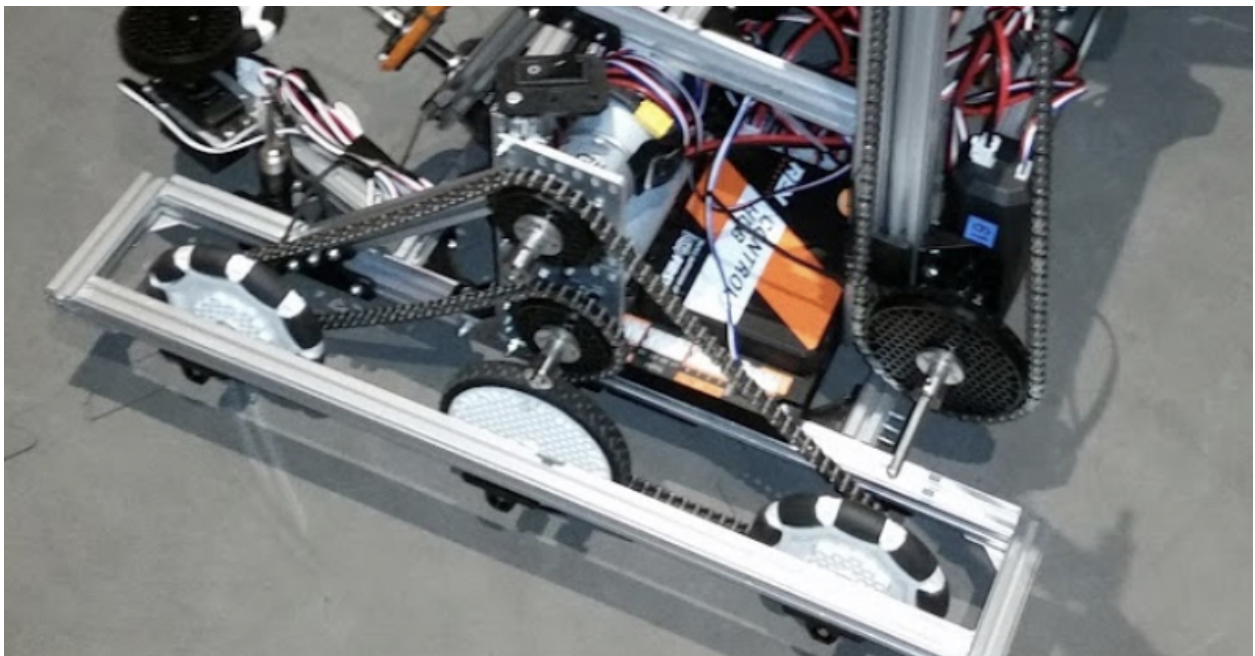
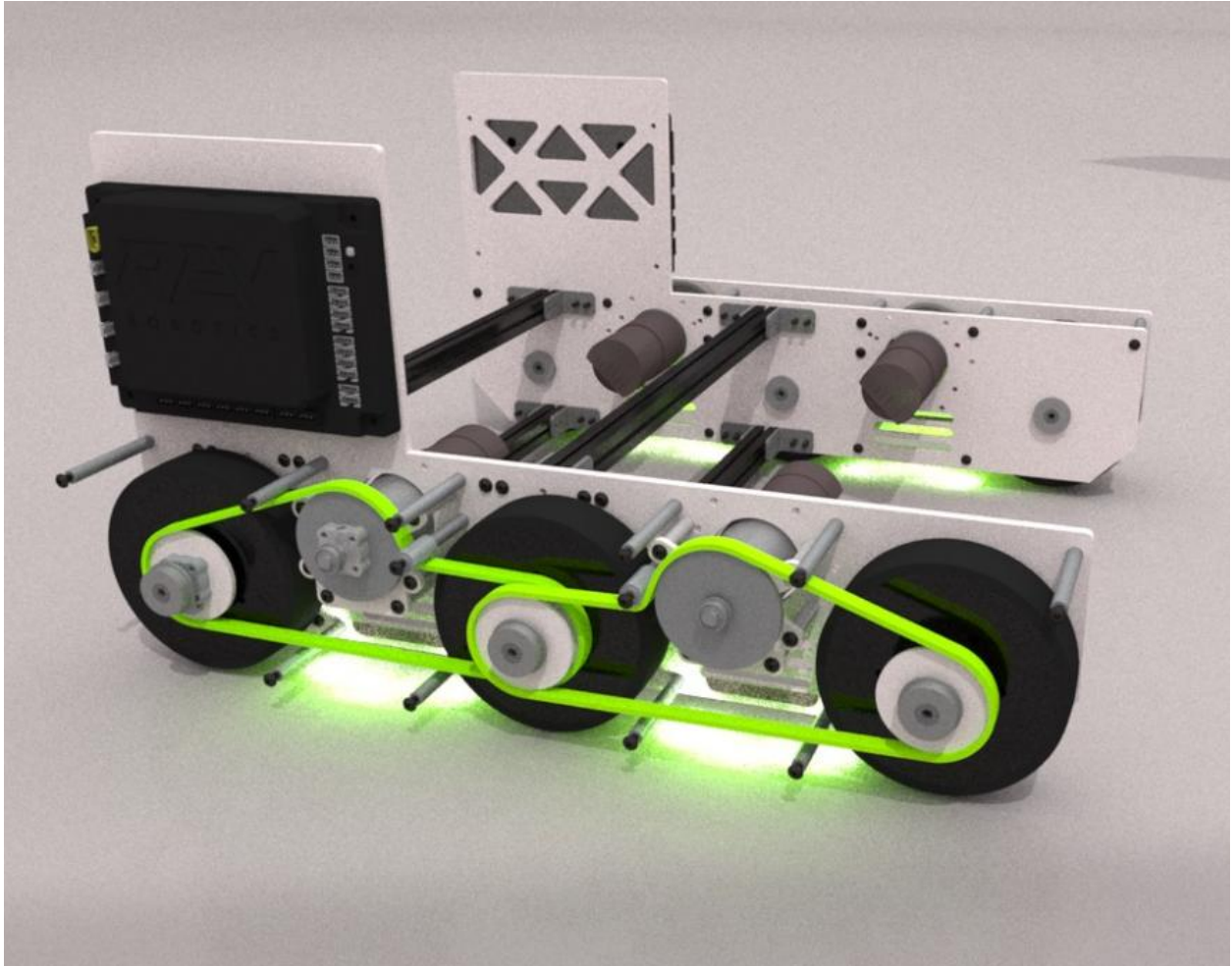
- C = center-to-center distance, inches
- L = chain length in pitches
- P = pitch of chain
- N = number of teeth in large sprocket
- n = number of teeth in small sprocket

Chain Wrap

Chain should, at the very least, have 90° of contact with the sprocket. The best practice is to have 180° or more of contact, as it is very unlikely to fall off with proper tensioning. Chain skipping, especially on drivetrains or arms, is very possible without proper chain wrap or tensioning. When tensioning chain, be sure to not undertension or overtension chain. Undertensioning chain can result in the chain falling off the sprocket or chain skipping, where the chain can skip along the sprocket. Overtensioning the chain often results in the motor burning out, or less seriously, a loss of efficiency. Push along the chain run, and if the chain moves slightly without significant resistance, chances are you've done it correctly. If it's too tight, then the chain will barely move under a gentle press.

Best practices for wrap





Advantages

- **Chain can take a beating.** No matter what your application is, metal chain is usually up for the challenge. #25 chain can hold up to 930lbs before breaking, and there's nothing you'll do in FTC that will exert that force. (If your chain does break, it's most likely due to a faulty Master Link or sprockets that are not correctly aligned.)
- **Chain can be however long or short as you wish.** If your ratio changes or your shafts move, it's easy to adapt your chain run - just break the chain and put it back together at its new length. You can often do this without even removing the chain from your robot.
- **Chain can be pretty precise.** When properly tensioned, roller chain doesn't have very much slop. However, you really need to get your chain tension right to reduce slop, and you'll probably want an adjustable tensioner for when the chain stretches. This can be done easily if using extrusion systems, as the sprocket can be adjusted for tension.

Disadvantages

- **Chain stretches over time.** As it's used, the connections between the links and rollers can stretch a bit. While it doesn't look like much, this stretching can introduce lots of slop into your chain run and even derail it in some cases. You'll most likely need an adjustable tensioner to keep your tension over time - some teams have used spring-loaded dynamic tensioners to automatically compensate for any changes. **The smaller the sprocket, the faster the chain stretch.** This is because when the chain is run on a smaller sprocket, more force is applied due to a smaller radius.
- **Chain wrap, especially in one chain run, can be problematic.** Typically, teams use either one or two chain runs (pieces of chain) per drivetrain side. However, one chain run can require more than one idler sprocket and get very complex in order to maintain proper chain wrap.
- **Sprockets are really big.** If you want a really high reduction using chain, you'll pay for it in the space that it takes up. Sprocket teeth are much larger than gear or pulley teeth, so your reductions are going to be much larger.

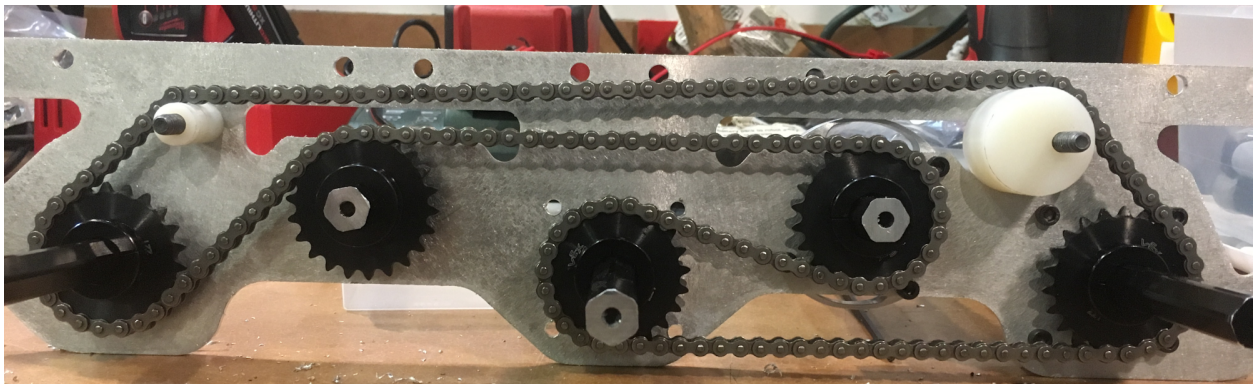


Fig. 45: 7244 OUT of the BOX Robotics, Relic Recovery

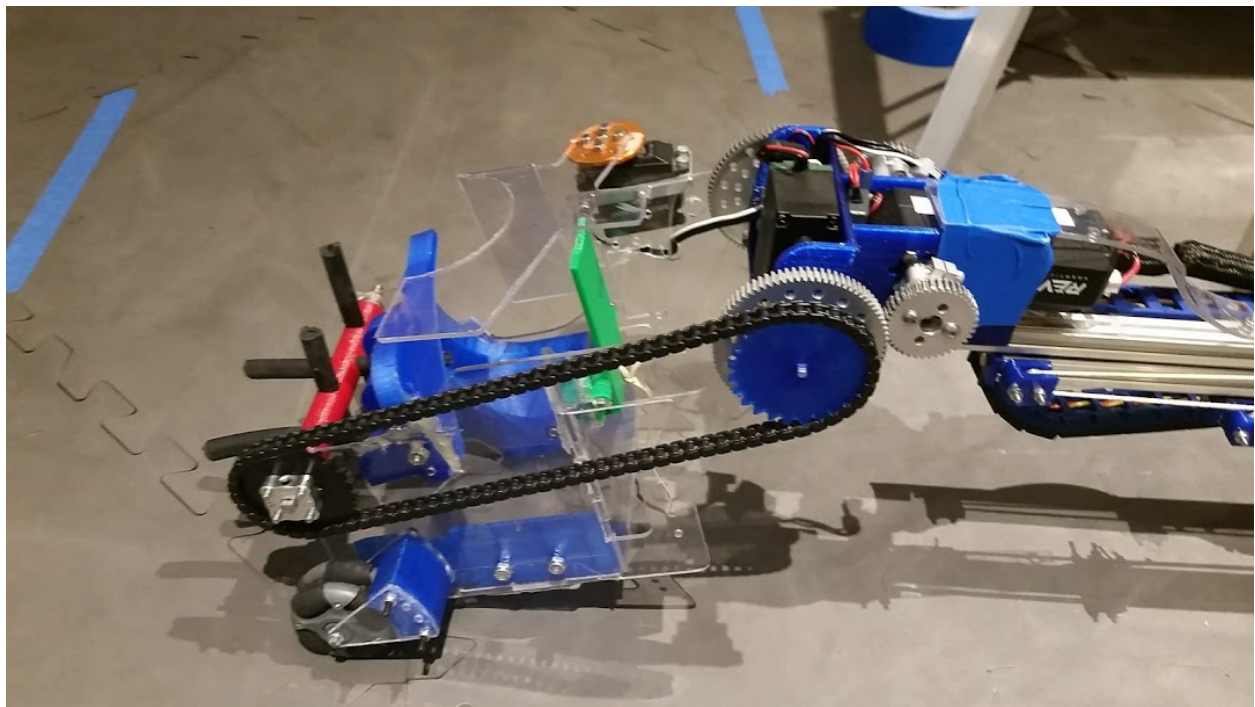


Fig. 46: 9794 Wizards.exe, Rover Ruckus, plastic chain on intake



Fig. 47: 724 Rednek Robotics Wun, Winning Alliance 1st Pick (Houston), Relic Recovery

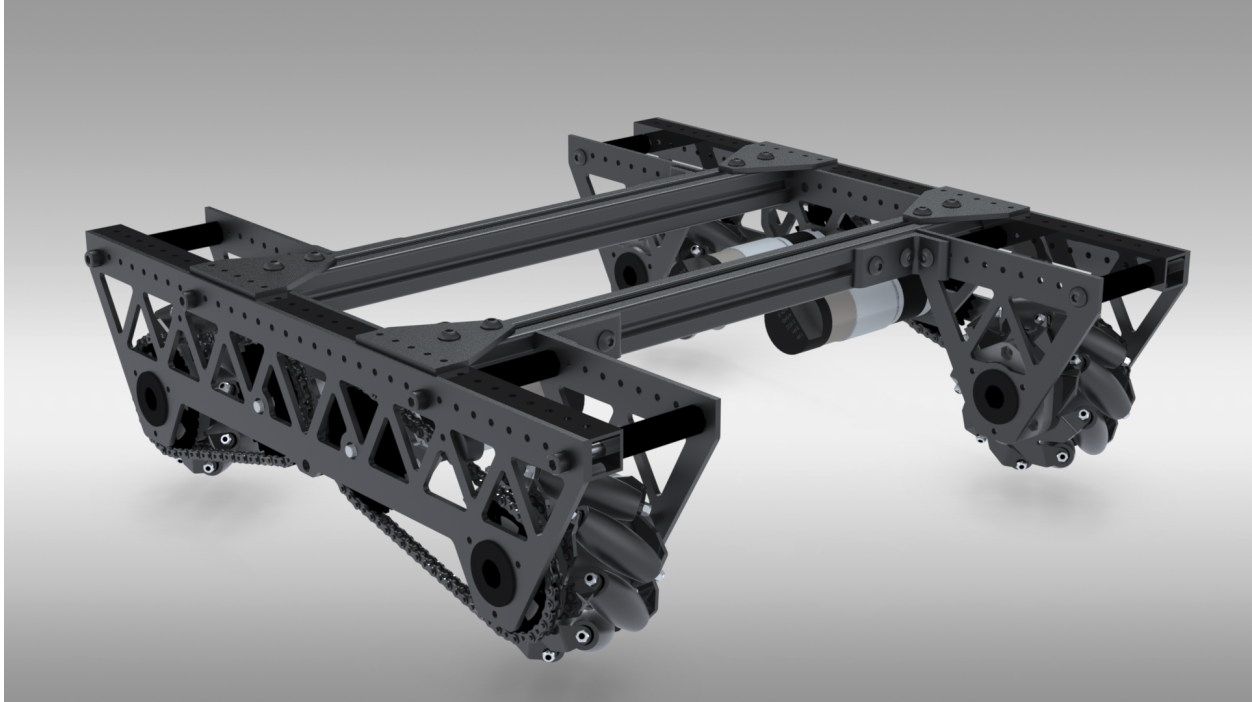


Fig. 48: 8103 Null Robotics, Rover Ruckus

1.8.4 Timing Belt

When you think of a belt, you're probably thinking of a very important men's fashion accessory. However, there's another type of belt, and it's way more relevant to robotics - the Timing Belt. If you've ever tinkered with the insides of a car before, you probably recognize timing belts as an important component designed to keep everything under the hood in sync. While a timing belt may complete a similar objective to chain, its characteristics and strengths are very different. Timing belts use a series of small, wide teeth to engage a pulley with a number of matching grooves. They earn their name because they can be very precise, transmitting power with absolutely no slop and ensuring a snug connection between shafts. They're lighter and more compact than chains, but they lack the customizability of their bulkier brother - belts come in a closed loop of predetermined length, and there's no changing that length on the fly. Like chain, belt is identified by its **pitch** - common pitches found on FTC robots include HTD 5mm, HTD 3mm, GT2 3mm, and XL 0.2".

When using timing belts, correct tension is very important. There are two main ways to get your tension right. The first is easy - goBILDA and Actobotics already have belts integrated into their hole patterns. You can buy correctly sized belt directly from each vendor, and your tension will be perfect as soon as the belt is installed. As your designs gain complexity, so will your belt runs - maybe there are more than 2 pulleys, and maybe your pulleys are all different sizes. To compensate for this, the second way to ensure tension is to use a dynamic tensioner, similar to those found in complex chain runs. To design for these tensioners, we recommend planning more complex belt runs in CAD before building them in real life.

Terms

- Pitch diameter (PD) refers to the imaginary circle which is traced by the middle of the belt as the pulley rotates. The outer edge of the circle is halfway between the outer face of the belt and the top face of the pulley groove. $PD = (tooth * pitch) / \pi$

- Pitch refers to the arc length between the centerlines of two adjacent pulley grooves. Pitch is simply the amount of pitch diameter in inches per tooth.
- Outside diameter (OD) is an imaginary circle which is traced by the top face of the grooves as the pulley rotates. **For pulleys, OD < PD.**
- Pulley clearance diameter refers to the outer diameter of the pulley + belt when the belt is meshed properly to the pulley. The pulley clearance diameter is larger than both PD and OD and should be used to check for interference with other mechanisms.

Center-to-Center calculations

Just like chain, the actual calculations for precise c-c distances for belts are complicated. Here is a [calculator](#)¹⁴ or [two](#)¹⁵ that simplifies the work.

$$C = \frac{P}{8} * (2L - (N + n) + \sqrt{(2L - (N + n))^2 - \frac{8}{\pi^2} * (N - n)^2})$$
$$L = \frac{2C}{P} + \frac{N + n}{2} + \frac{P(\frac{N-n}{2\pi})^2}{C}$$

- C = center-to-center distance, inches
- L = belt length in pitches
- P = pitch of belt
- N = number of teeth in large pulley
- n = number of teeth in small pulley

$$C = \frac{L - 1.57(D + d)}{4} + \sqrt{[(\frac{L - 1.57(D + d)}{4})^2 - \frac{(D - d)^2}{8}}$$

- D = chosen diameter of large pulley
- d = chosen diameter of small pulley
- L = length of belt
- C = center distance
- (all units must be the same)

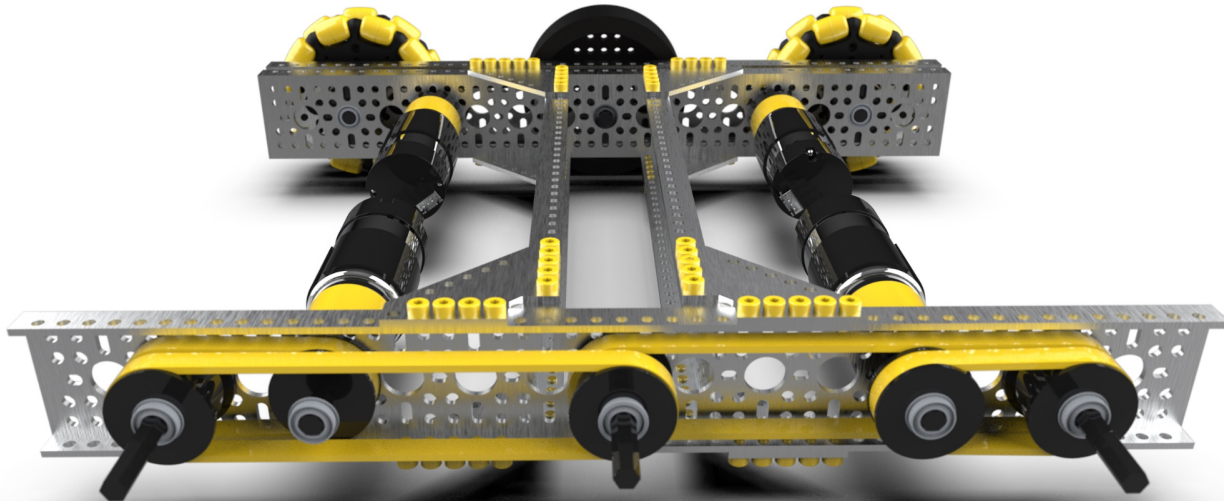
Belt Wrap

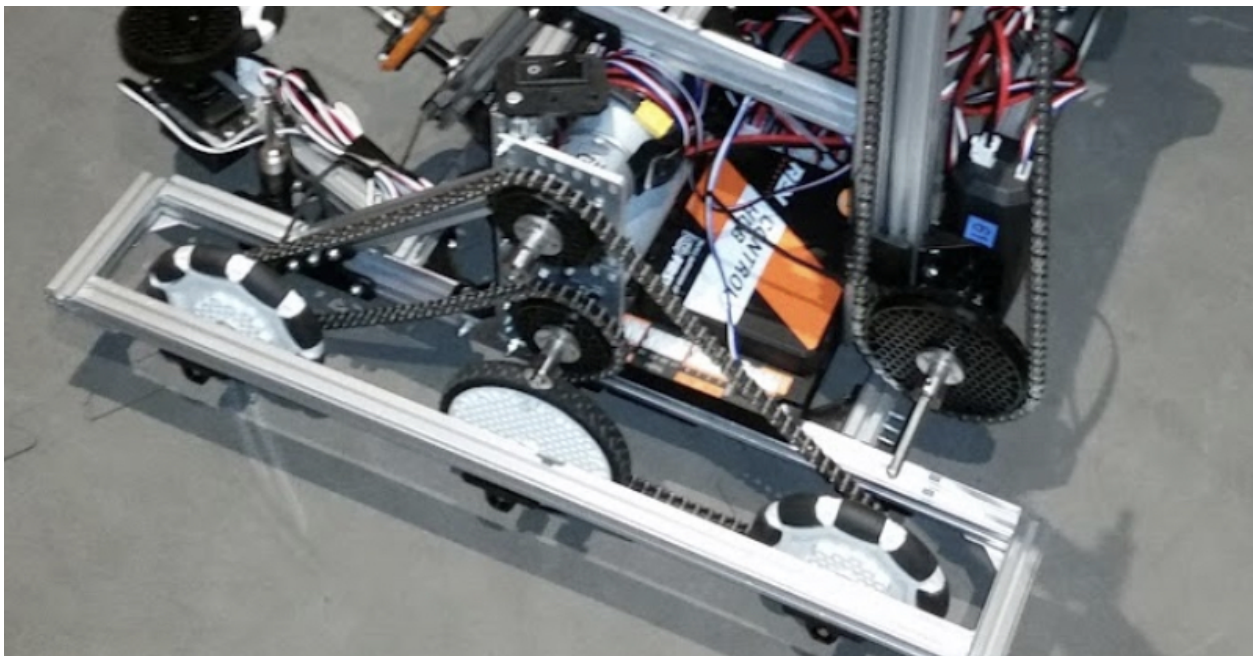
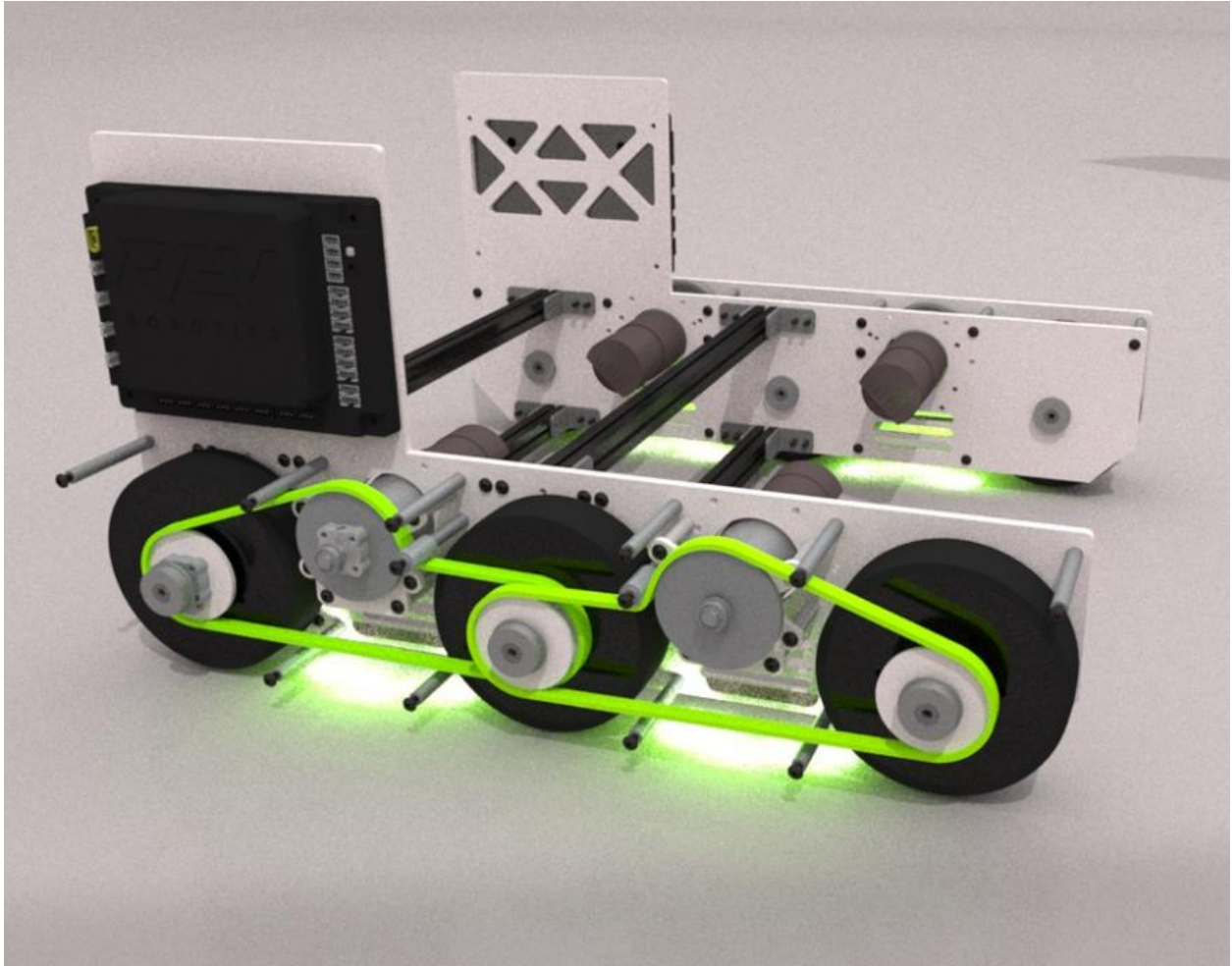
Belt should, at the very least, have 90° of contact with the pulley. The best practice is to have 180° or more of contact, as it is very unlikely to fall off with proper tensioning. Belt skipping, especially on drivetrains or arms, is very possible without proper belt wrap or tensioning. When tensioning belt, be sure to not undertension or overtension it. Undertensioning belt can result in the belt falling off the pulley or belt skipping, where the belt can skip along the pulley. Overtensioning belt often results in the motor burning out, or less seriously, a loss of efficiency. Push along the belt, and if it moves slightly without significant resistance, chances are you've done it correctly. If it's too tight, then the belt will barely move under a gentle press.

¹⁴ https://www.engineersedge.com/calculators/Pulley_Center_Distance/toothed_pulley_center_distance_calculator_12900.htm

¹⁵ <https://www.sudenga.com/practical-applications/figuring-belt-lengths-and-distance-between-pulleys>

Best practices for wrap





Advantages:

- **Pulleys can be made at home.** Pulleys can be 3D printed for most situations, allowing you to cut costs and create unique tooth counts easily.
- **Belts are very strong.** They're reinforced with fiberglass cords that are incredibly hard to break, giving belts immense strength. *(If you break a belt, it's most likely because it was out of alignment or tensioned far too tightly.)*
- **When tensioned correctly, there is absolutely no slop.** Engines use timing belt for a reason - because it's the best possible solution for them to perfectly synchronize their shafts. There's nothing that matches the rotational accuracy of a properly tensioned belt.
- **Belts are efficient and quiet.** Compared to the loud shredding sound of a chain run, belt runs are dead silent, and they're more efficient than chains (although this makes zero practical impact in the robotics use case).

Disadvantages:

- **Belts aren't customizable.** You buy a belt of a specific length and you're stuck with that length until you buy another one. This isn't too bad if you're planning out your robot properly, but chain will work better for prototypes where the chain length will be changing often.
- **Belts can be wider than alternatives (especially chain).** This probably won't have much of an impact, but belt can often be wider than other power transmission methods, so it may not always fit.
- **Belts can be expensive (but you'll save money with pulleys).** While you can buy chain 10 feet at a time, you'll most likely be buying each belt brand new. While this can get expensive, you'll be saving money on pulleys.

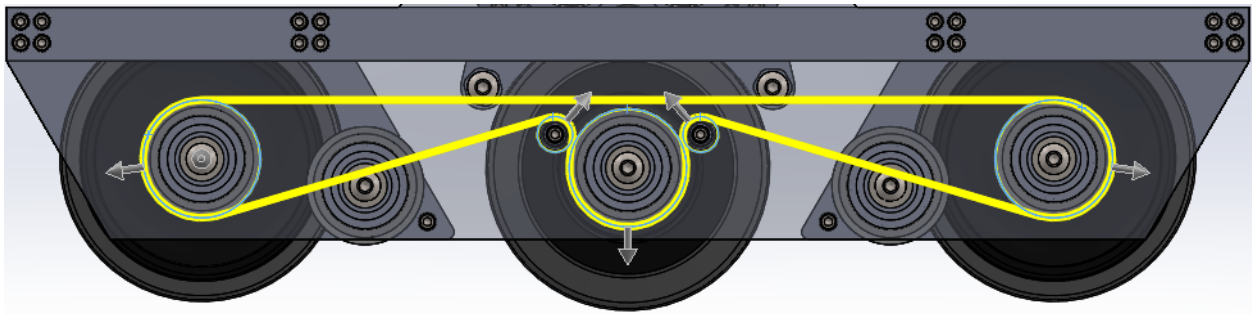


Fig. 49: 8103 Null Robotics, offseason prototype

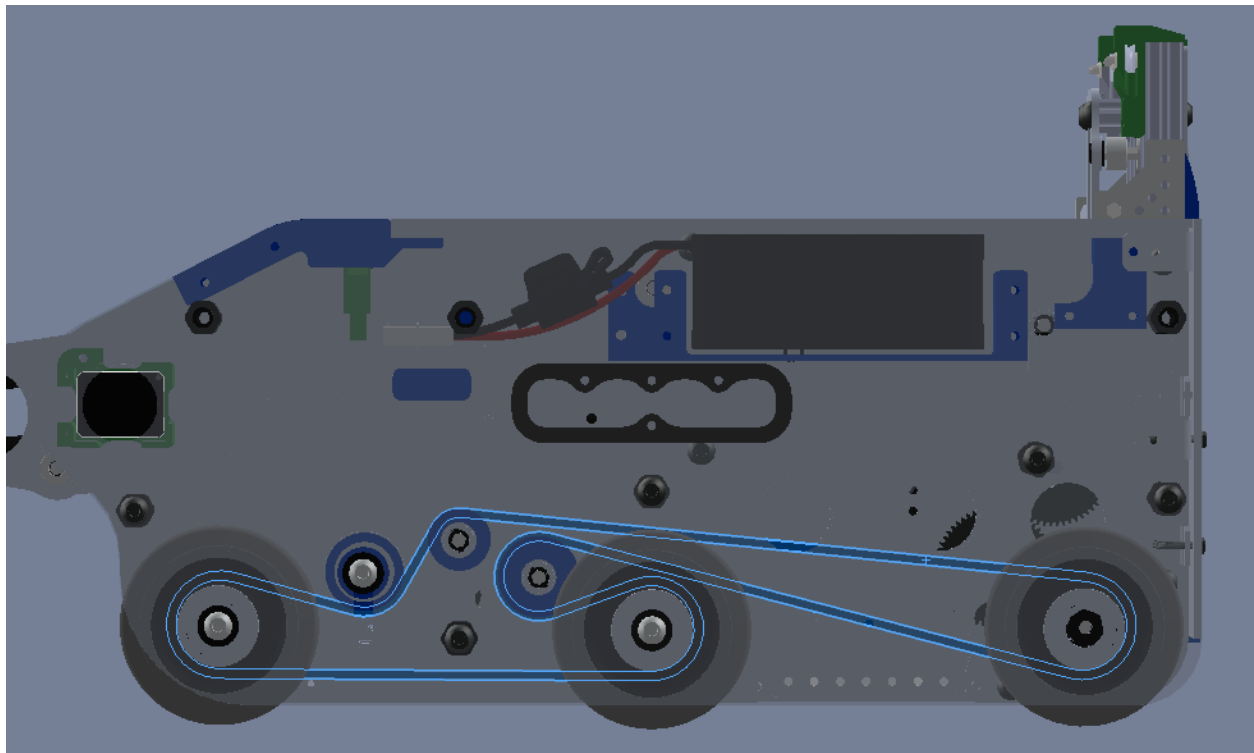


Fig. 50: 8417 Lectric Legends, Rover Ruckus

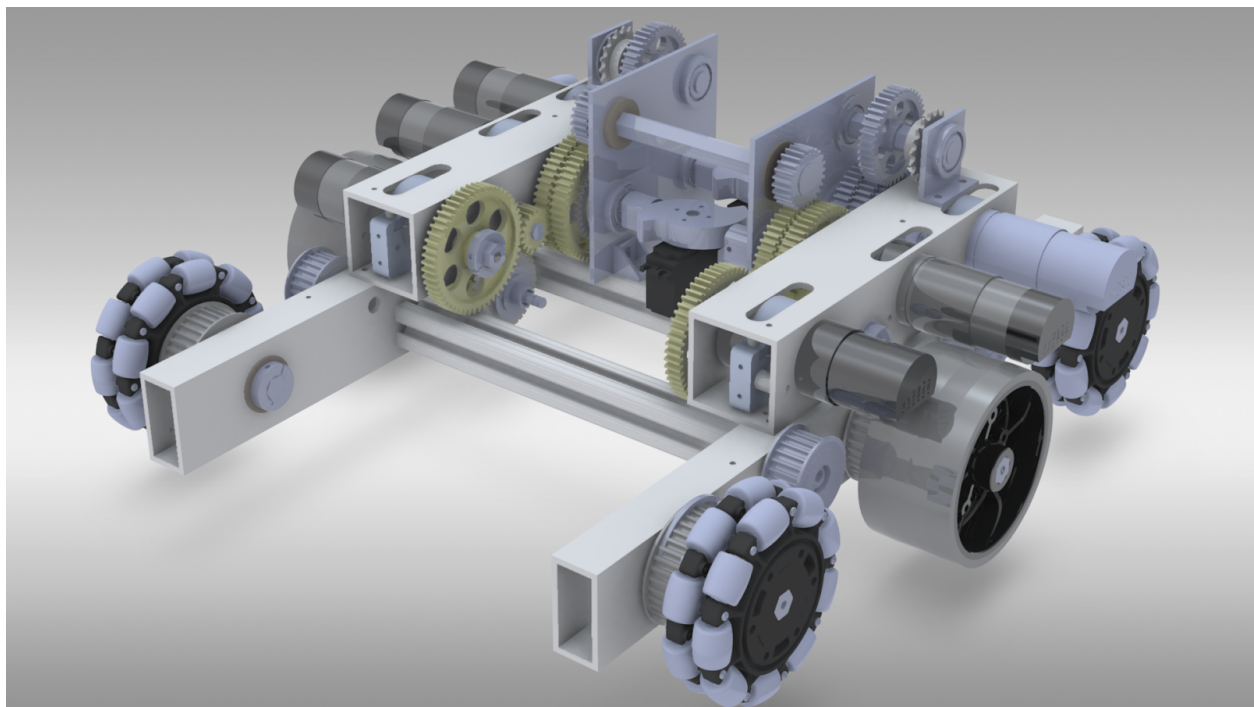


Fig. 51: 5975 Cybots, Velocity Vortex

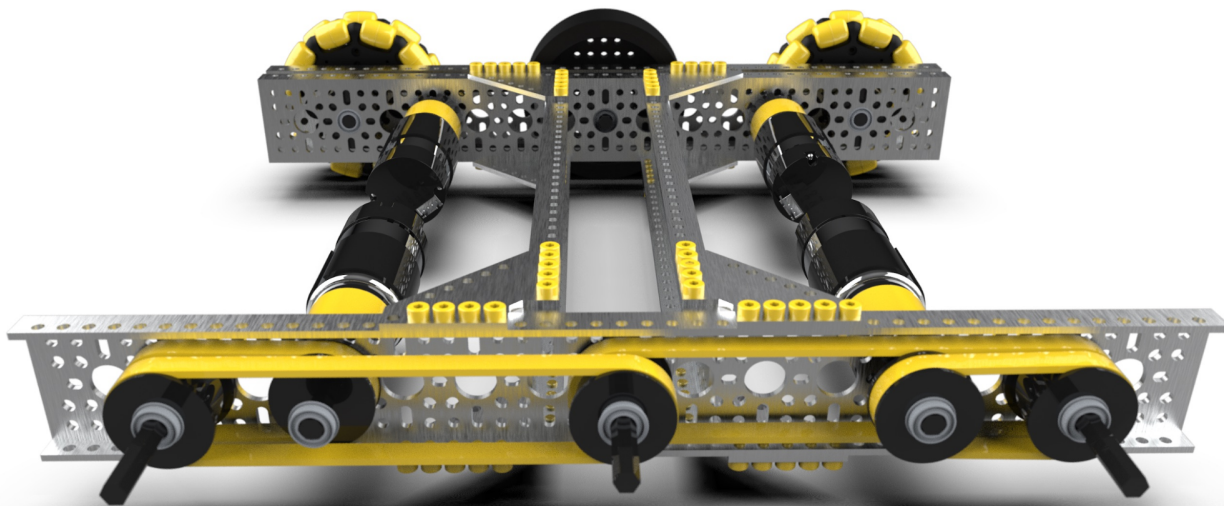


Fig. 52: Ethan Doak, goBILDA

1.9 Linear Motion Guide

Linear motion is one of the most important components of a successful robot. In most games, teams are required to reach into an area that the drivetrain cannot access in order to pick up or deposit game elements. For example, in 2017-2018, Relic Recovery, teams needed to extend an arm of some sort to grab the Relic that was placed in the corner of the playing field. This area of the game field was nearly impossible for a robot to drive to, so a linear extension was necessary.

Access of > 18" is necessary for nearly all games; >24" is preferred. In some games, an extension of 36" or more may be needed.

Note - it is possible to achieve extension with an arm, but since this guide is geared toward newer teams, linear extensions should be prioritized over arms. Refer to the [Arm Guide](#) (page 136) for more information.

1.9.1 Extrusion Slides

Extrusion slides are made up of a stack of extrusions that extend by sliding along each other. There are two ways to make this happen: bushings or V-wheels. Bushing slides connect the two slides with two self-lubricating plastic pieces that slide smoothly along the slots in the extrusion. V-wheel slides have V-shaped groove bearings on both sides of the extrusion that bite into grooves on the extrusion, allowing the stages to slide smoothly. REV, Actobotics, and goBILDA all sell extrusion slide kits that integrate nicely with existing FTC kit parts. Beyond this, MakerBeam sells components for a V-wheel extrusion slide, and Misumi offers a few different sizes of bushing-based extrusion slides. REV furthermore sells a 8020 v-groove bearing slide kit for FRC, that is not recommended for FTC use cases. **We recommend that newer teams stick to slide kits designed for FTC instead of DIY alternatives.**

REV Robotics 15mm Linear Motion Kit

The [REV 15mm Linear Motion Kit](#)¹⁶ is based off of the 15mm extrusion system. **This extrusion kit does not perform spectacularly without modification.** This has been partially remedied by REV as they have developed a second iteration of their slide kit, which has much better tolerances on the Delrin sliders. Still, you'll see a lot of competitive teams use this kit with multiple modifications, such as adding lots of lubricant and mounting the sliders differently. Teams have also 3D printed their own sliders, though this is not a great idea for teams without much 3D printing experience. One of the biggest issues with the stock REV kit is the tendency of the slides to bind. Additionally, since the only thing attaching one extrusion to another is the plastic slider, the REV slides are not particularly sturdy, and require crossbeams to keep alignment. Overall, this kit is lightweight, simple, and cheap. It can be a good start for teams using REV and needing a linear extension, and is generally usable out of the box. However, it is not very smooth and only achieves its maximum potential when modified and tweaked.

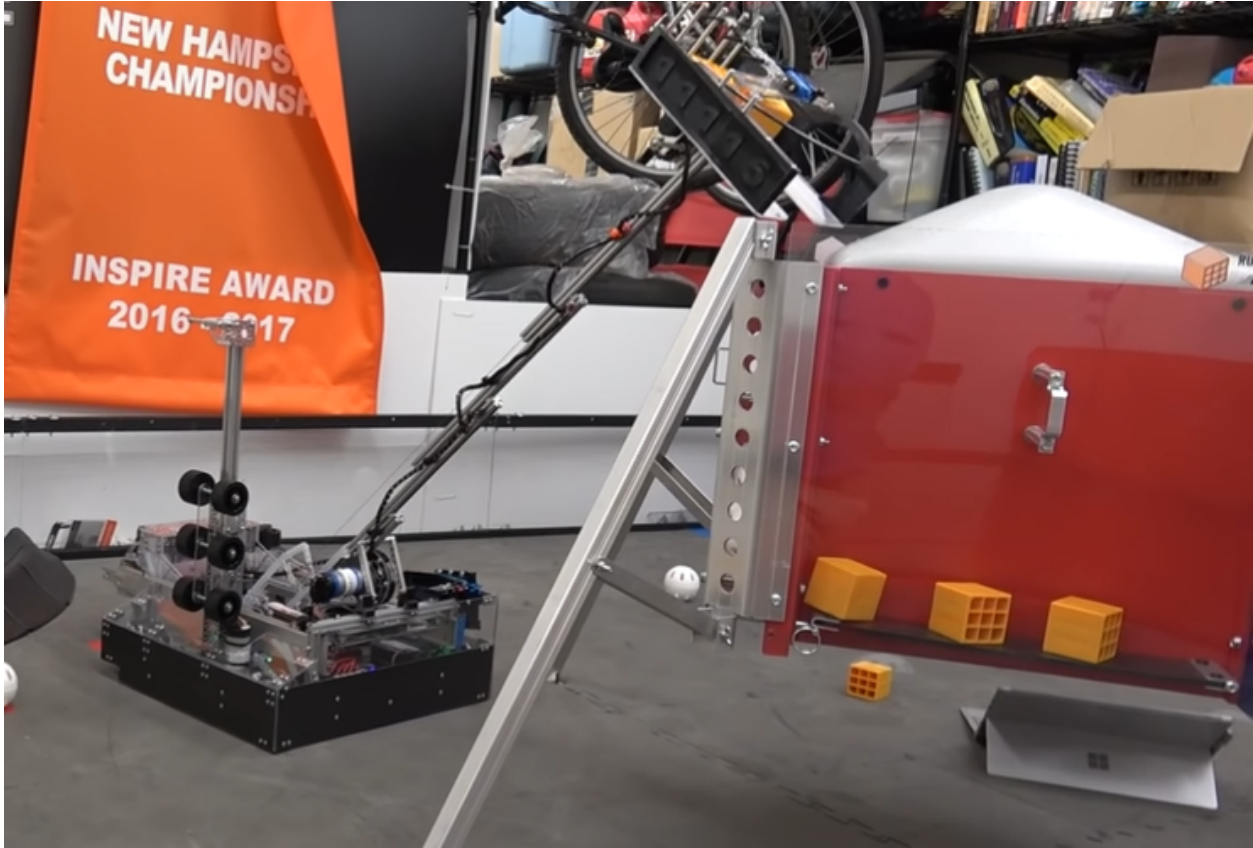
Advantages

- REV has complete guide on how to rig the linear slides
- Easily interfaces with REV building system
- Lightweight, should be used for light/medium loads only
- If tweaked, can be a very efficient linear slide

¹⁶ <http://www.revrobotics.com/rev-45-1507/>

Disadvantages

- Does not perform perfectly out of the box
- Can flex under load, needs additional support
- May need some modifications such as custom sliders



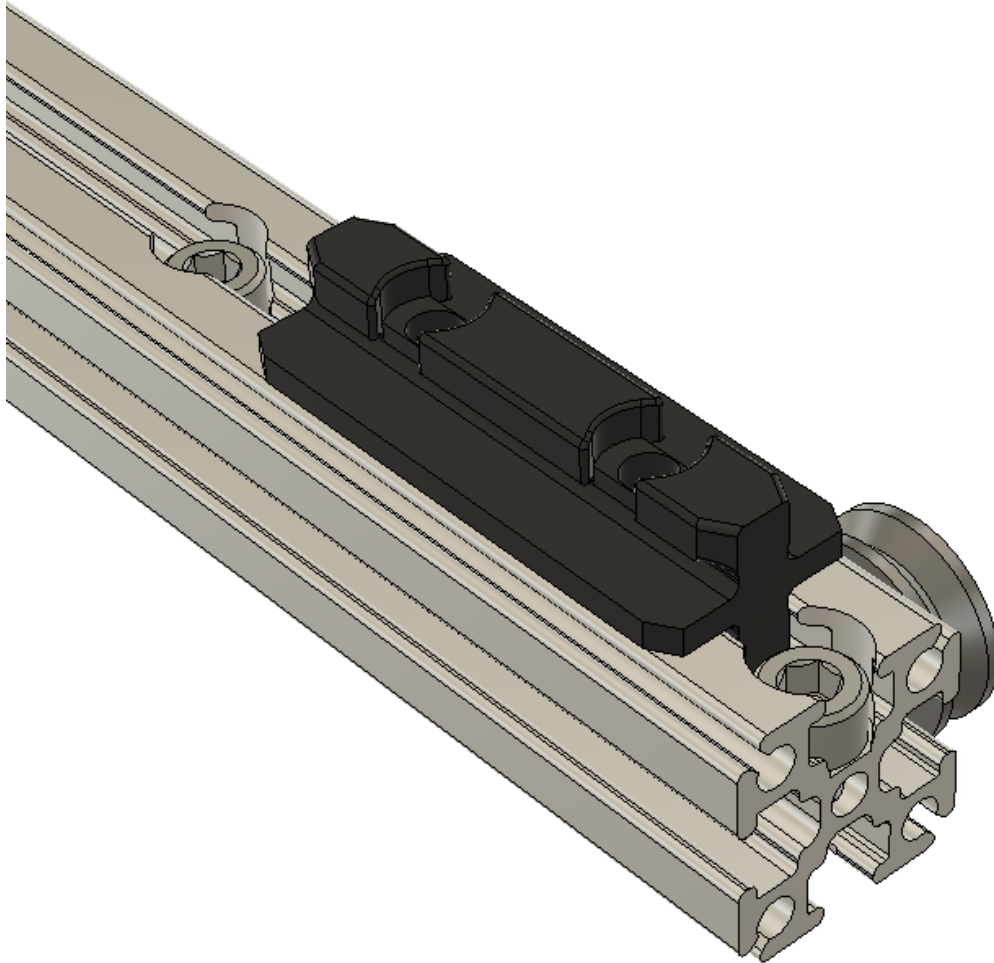


Fig. 53: 11115 Gluten Free, Rover Ruckus, custom REV slides

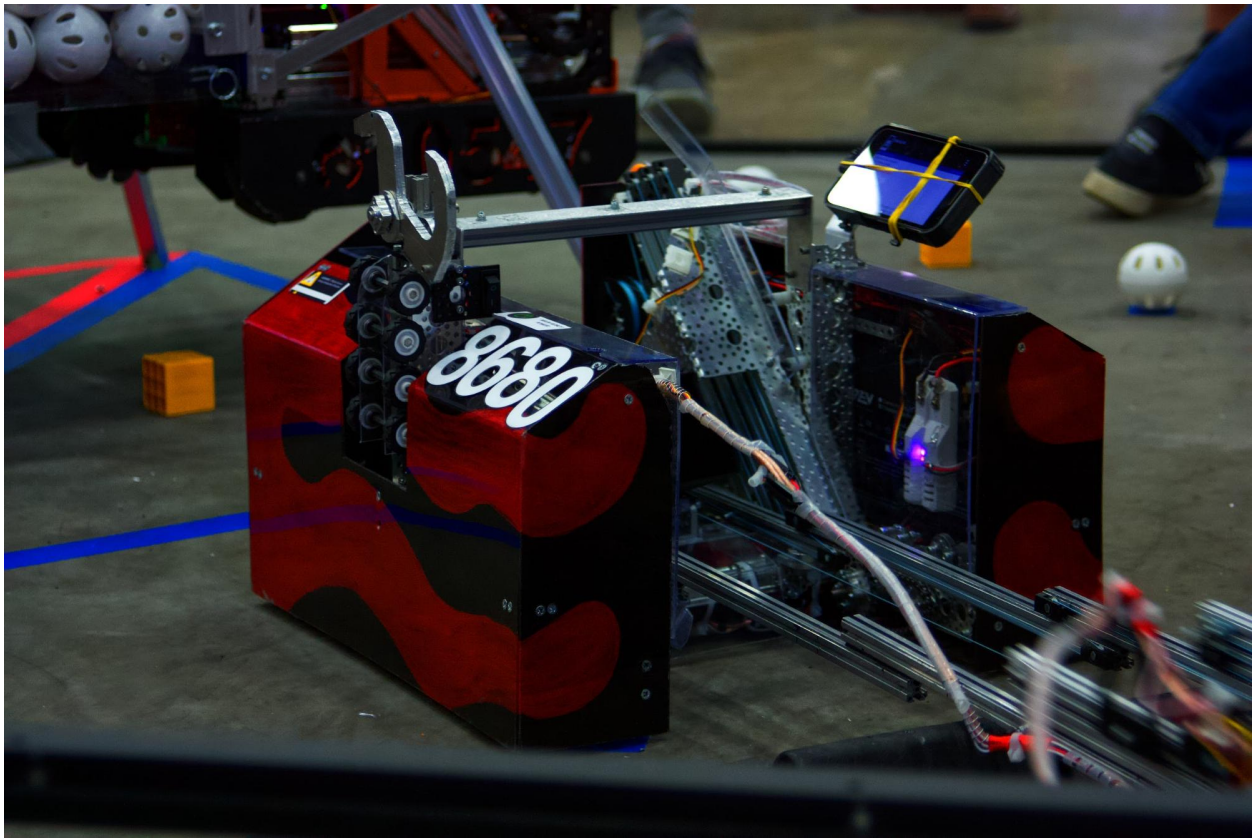


Fig. 54: 8680 Kraken-Pinion, Rover Ruckus, REV

Actobotics X-rail Slide Kit

The [X-rail Slide Kit](#)¹⁷ works well out of the box. However, the main caveat is that the kit has a **very low maximum load (2lb. at maximum extension)**. Teams will have to keep their designs on this kit particularly lightweight. This slide uses elastic retraction through the use of surgical tubing, which means that instead of having both an extend and return string, there is a retraction force applied at all times. This helps simplify tensioning and spooling, however, limits how fast the slide can be run. Additionally, the plastic end caps have a reputation of breaking regularly because they endure shock load every time the slide extends to maximum. 3D printed alternatives may be more sturdy than the stock end caps. **It is highly recommended that teams add an additional set of v-groove bearings at the end of each piece of extrusion to give each stage an additional point of support.** This will increase load capacity and possibly smoothness.

Advantages

- Easily interfaces with Actobotics building system
- Elastic retraction is a simple way to retract
- **Should be used for light/medium loads only**

Disadvantages

- Will flex under load, needs additional support
- Elastic retraction slows down extension speed and retraction will be slower than string retraction

¹⁷ <https://www.servocity.com/cascading-x-rail-slide-kit>

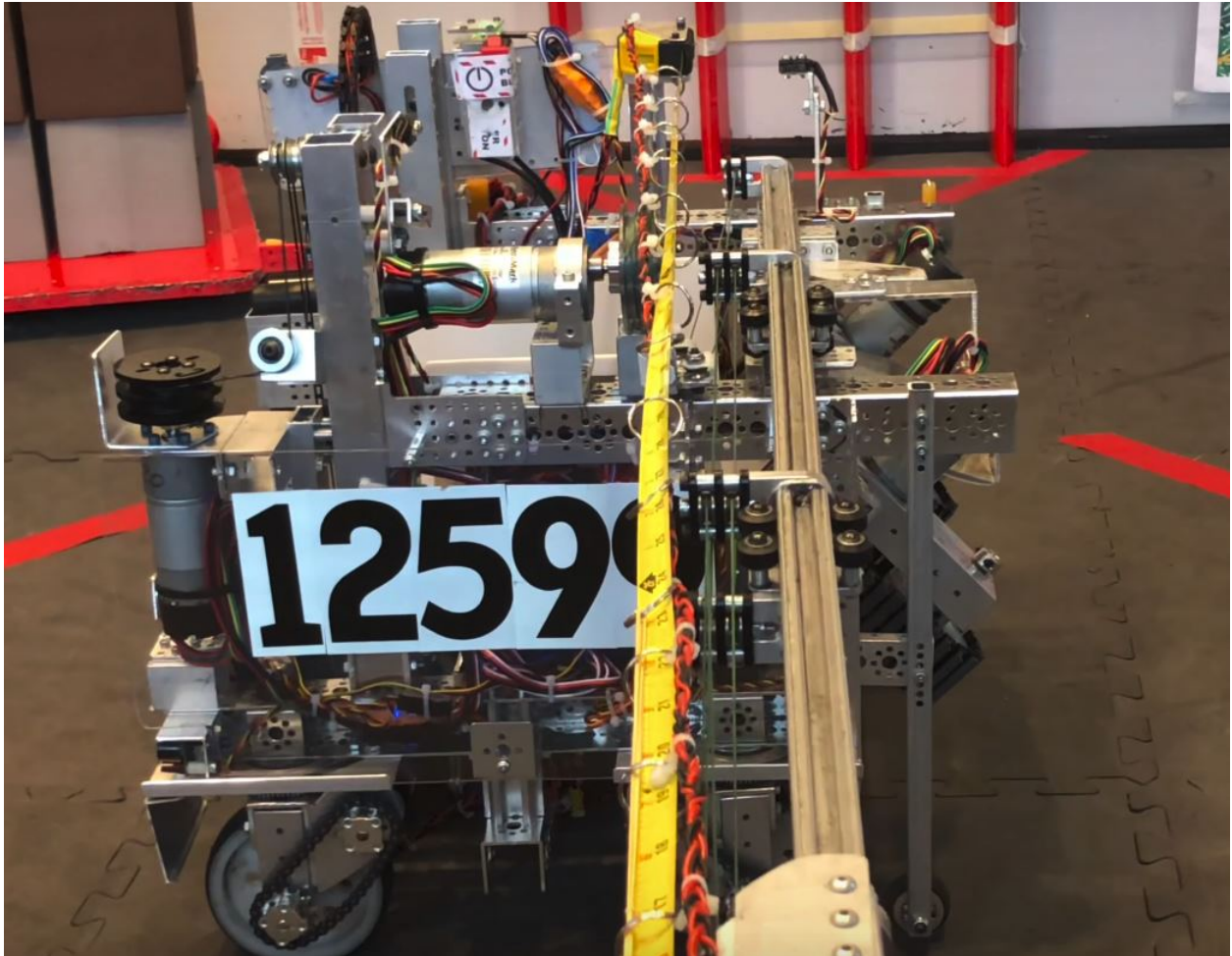


Fig. 55: 12599 Overcharged, Relic Recovery, Actobotics X-Rail

goBILDA goRAIL

Linear motion guides¹⁸ are goBILDA's take on linear motion using goRAIL, which is a type of extrusion compatible with v-groove bearings. Similar to a standard 8020 v-groove carriage, used in FRC, goRAIL is a lighter option that is better suited for FTC.

1.9.2 Drawer Slides

You've *definitely* used a drawer slide before - at least two of them are mounted to almost any drawer that you've opened. Teams use these drawer slides for linear motion, often stacking them using 3D printed spacers to achieve plenty of extension. These slides are available from a number of different vendors, and come in many varieties, so choosing the right slide can seem overwhelming. Steel drawer slides are common, but can be hard to mount, as they aren't made to be stacked. Aluminum drawer slides, such as the MiSUMI slides or the Hafele slides listed below, are generally the best option for teams. The Hafele slides are less than \$3.50 for a pair, but the MiSUMI rails offer much smoother motion and have a better ball bearing system. Igus is a brand known and trusted by robotics teams and manufacturers, but their parts are certainly not cheap, and can reach into the \$100+ range. **If linear slide kits aren't for you, we recommend the MiSUMI slides for newer teams**, especially if you are using REV, as MiSUMIi uses the M3 standard. **Drawer slides should be mounted oriented vertically, like in an actual drawer. They can be mounted horizontally, but this is not recommended as they will sag much more.**

Listed below are the recommended drawer slides.

Steel-rolled cabinet drawer slides

Available from your local hardware store, steel slides aren't a bad option for FTC teams, as they are heavy-duty enough for most use cases. However, these kinds of slides are much heavier than other aluminum slide options. Furthermore, these slides are not designed to have bearings or a second slide attached to them, because they only contain mounting for a standard drawer. Thus, these slides require drilling holes in order to mount the necessary parts for linear extension.

Advantages

- Commonplace at any hardware store
- Not very expensive

Disadvantages

- Heavier than other slide options (steel as opposed to aluminum)
- Sliding is usually good but not great
- Hard to adapt to building systems
- 3D printed spacers may be required

¹⁸ <https://www.gobilda.com/gorail-based/>

MiSUMI Telescopic Slide Rails

The **MiSUMI slide rails**¹⁹ are preferred by many top-tier teams because they are sturdy, very reliable, and ridiculously smooth due to the ball bearing system. **MiSUMI slides are able to withstand a significant amount of load with little flex.** They are also low-profile, and have a M3 mounting pattern, making it easy to attach to REV extrusion. However, MiSUMI slides have a higher price point, and it is often difficult to attach one slide to the next. An easy solution is to attach the end of one slide piece to REV extrusion, and do the same with the next slide. Then attach the REV pulley bearing on the top of the extrusion piece for the string to run through. Some teams have 3D printed an insert that goes between each slide instead of using the 15mm extrusion piece, as it takes up quite a bit of space. In order to attach the slides to anything, teams will need to purchase **countersunk M3 screws** from McMaster-Carr. For attaching to REV extrusion, buy 6mm M3 screws with the M3 nut (**not locknut**) to insert inside the extrusion. As a tip, try to protect chips or sawdust from falling into the slides, as the sliding will have a noticeable difference.

MiSUMI sells two different types of slides: SAR2 and SAR3. The SAR is a two-piece slide, while the SAR3 is a three-stage slide (has intermediate slider to increase the extension of the slide. Teams have used both successfully, and there isn't neither option is clearly superior.

Rail length: 400 mm, part number SAR240 (SAR2) or SAR340 (SAR3)

Advantages

- Best slide smoothness due to ball bearing cartridge
- Very little slide flex, robust build quality
- Can handle relatively heavy load use cases (within reason)
- Compatible with REV 15mm extrusion system

Disadvantages

- Not cheap
- Tricky to adapt if not using REV
- 3D printed spacers may be required

Hafele drawer slides

Hafele drawer slides²⁰ are a low-cost option for teams not wanting to invest in MiSUMI slides. Hafele slides are less smooth than MiSUMI, which is to be expected given their low price point. They become unusable when handling large amount of load (greater than 57 pounds). However, they are great for low/medium-load use cases. Hafele slides are a great option for teams looking for a cheap alternative to the other more expensive options at the cost of robustness.

Advantages

- Cheapest drawer slide option
- Great value for low/medium load applications

¹⁹ <https://us.misumi-ec.com/vona2/detail/110300072130/?HissuCode=SAR240>

²⁰ <https://www.cabinetparts.com/p/hafele-drawer-slides-side-mount-drawer-slides-HAF42058373>

Disadvantages

- Not the best slide smoothness
- Suffers in build quality (after all, it is very cheap)

REV UltraSlide

The REV UltraSlide is an aluminum ball bearing drawer slide that uses M3 mounting. It is of similar form factor to the MiSUMI SAR drawer slide. The key feature of the UltraSlide is the side extrusion profile, which is compatible with REV's 15mm extrusion. It allows the REV pulleys used in the Linear Slide Kit to be mounted directly to the slide, thus making each stage compact and lightweight.

Since this is a new product, currently a list of advantages/disadvantages is unable to be provided. More details will come in a future revision.

Igus

Igus slides used to be a popular option among top teams for linear extension back in the earlier days of FTC. However, with the arrival of MiSUMI slides, they have decreased in popularity due to their tendency to bind. However, if your team is looking for a high-load or special use case, by all means explore this option.

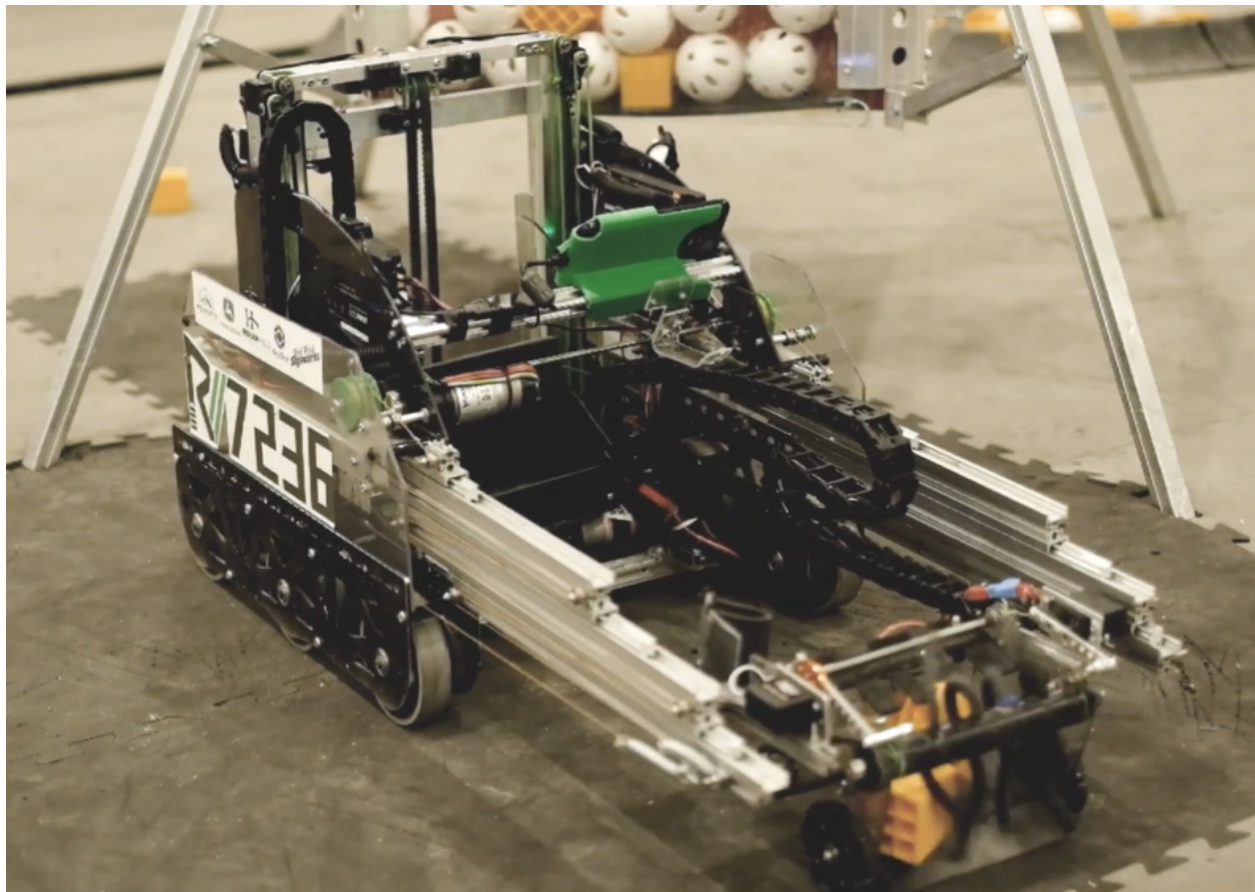


Fig. 56: 7236 Recharged Green, Rover Ruckus, Misumi SAR3



Fig. 57: 6929 Data Force, Rover Ruckus, Misumi SAR3



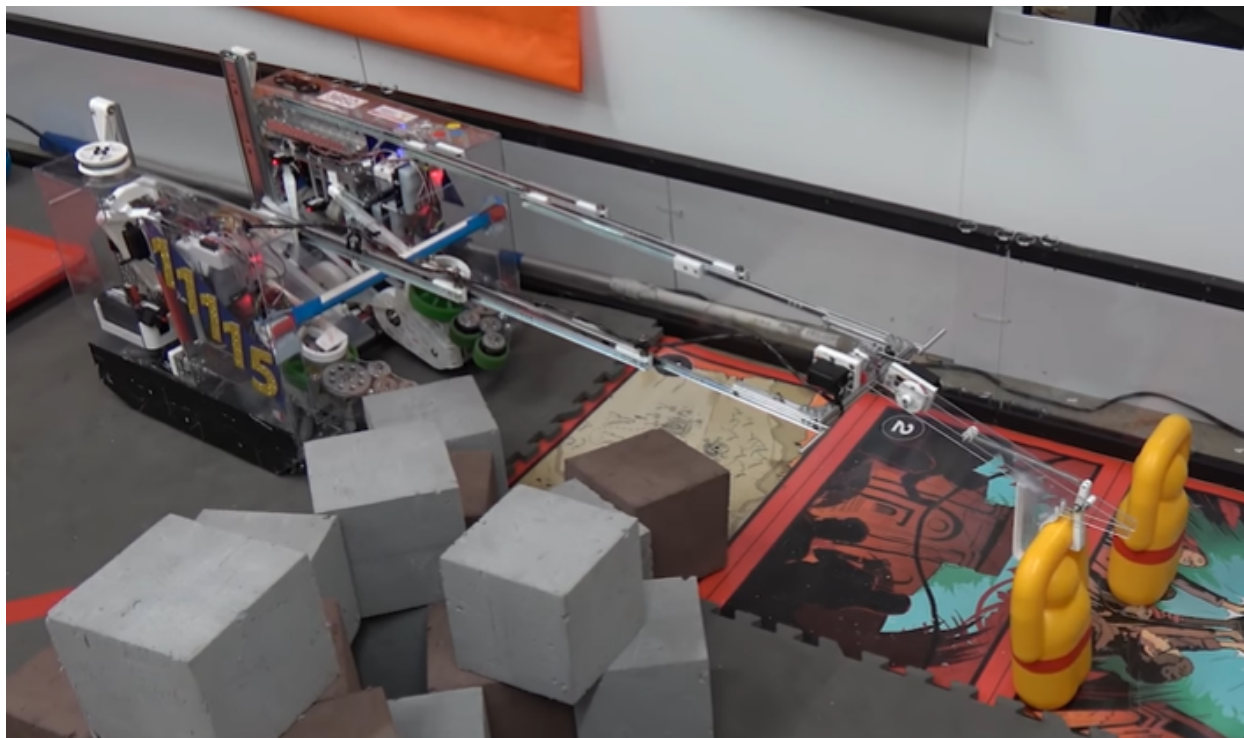


Fig. 58: 11115 Gluten Free, Relic Recovery, Hafele slides

1.9.3 Custom Options

Some teams have opted to build custom solutions for linear motion. Many of these teams borrow concepts from FRC teams, modeling their elevators off of the tall systems found on the larger robots. There's a reason that many competitive FRC teams build the same type of elevator - at their scale, the box tube elevator has proven to be the most efficient way to get game pieces off the ground at blisteringly fast speeds. When built correctly, an elevator of this type can withstand hundreds of pounds of load on any axis while barely weighing anything. However, existing off-the-shelf options already fill the linear motion needs of most FTC teams. Custom extension systems also require tons of work in CAD, hours upon hours of manufacturing time, and may need multiple iterations before they work correctly. Due to their complexity and how challenging they are to design, less experienced teams may encounter significant challenges. Additionally some teams will opt to use custom carbon fiber rod based slides. These slides are primarily designed to withstand loads perpendicular to the direction of the slide, so they are a light strong option for slides that need to extend far and quickly. However, note that lightweight drawer slides (MiSUMI aluminum) can offer similar performance at a fraction of the complexity.

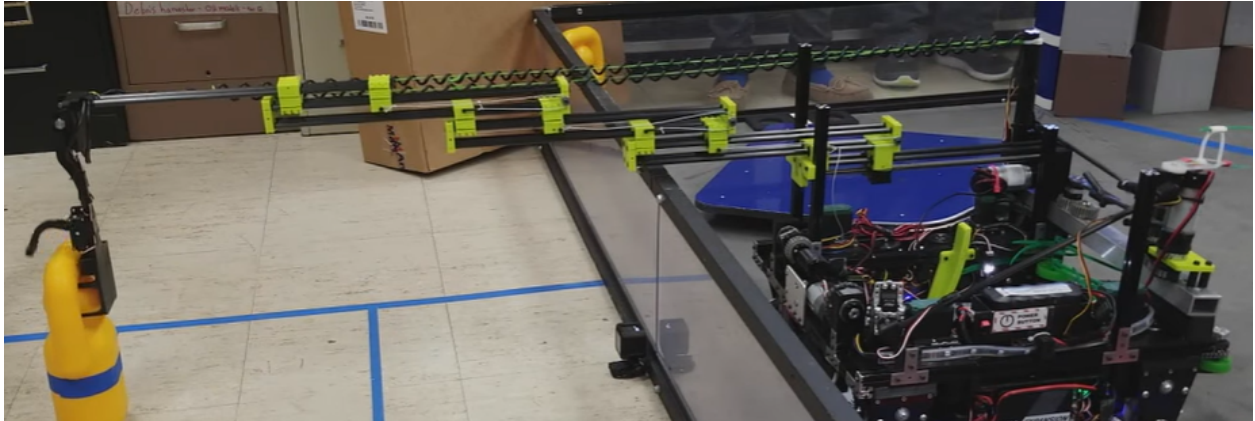


Fig. 59: 5975 Cybots, Relic Recovery, custom carbon fiber and linear bearing slide

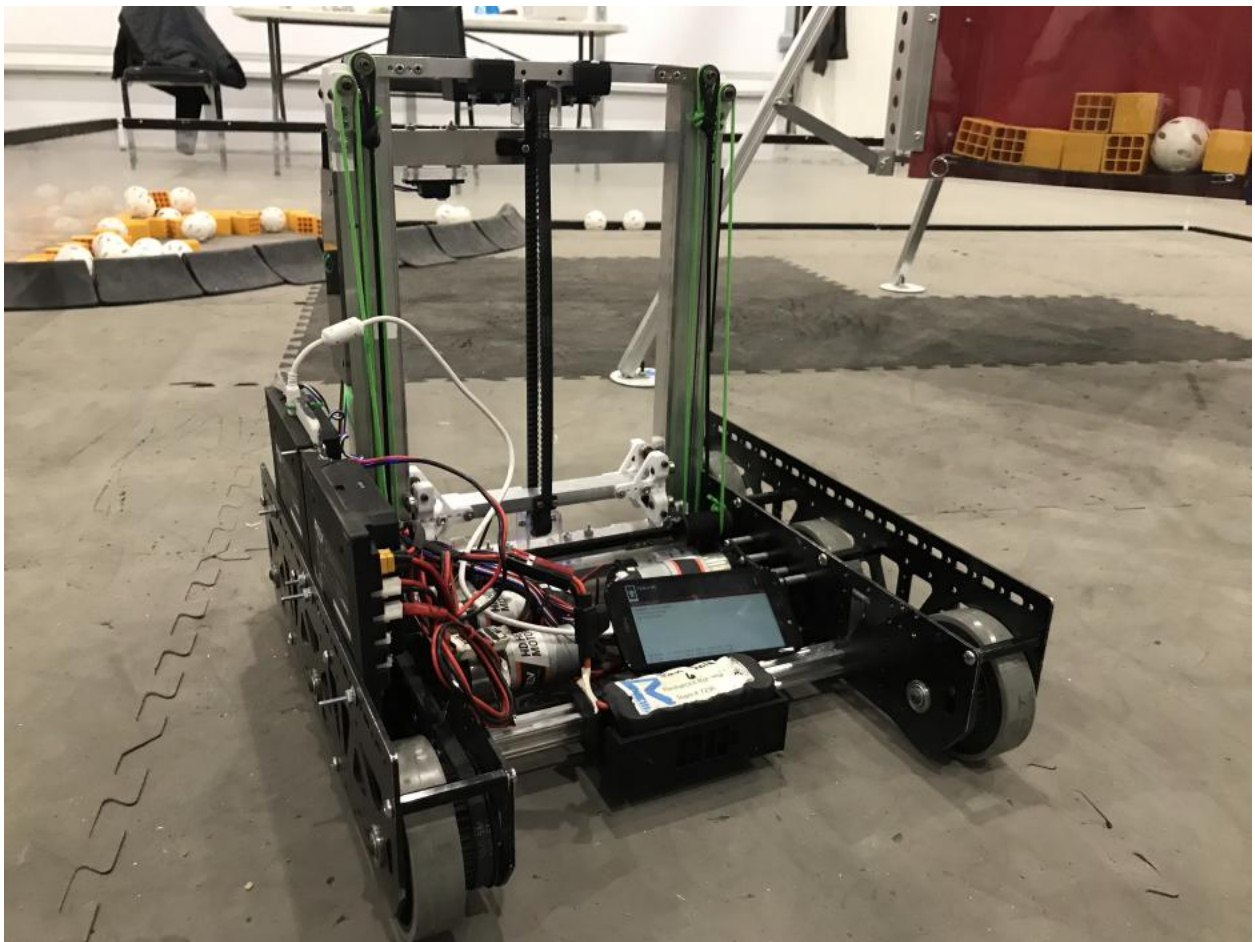


Fig. 60: 7236 Recharged Green, Rover Ruckus, custom box tube linear elevator

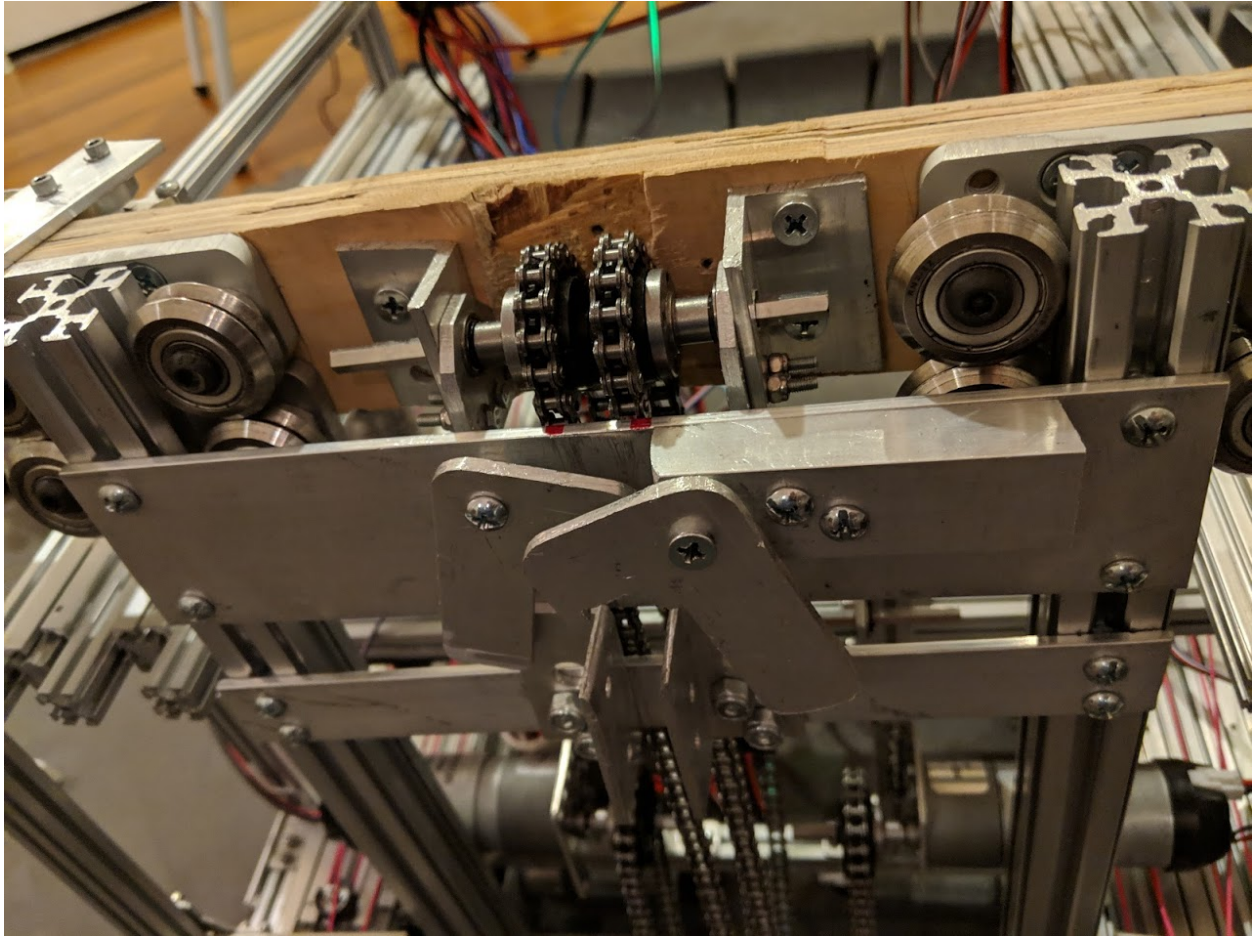


Fig. 61: 13075 Coram Deo Academy Robotics, Rover Ruckus, custom 8020 chain driven hang

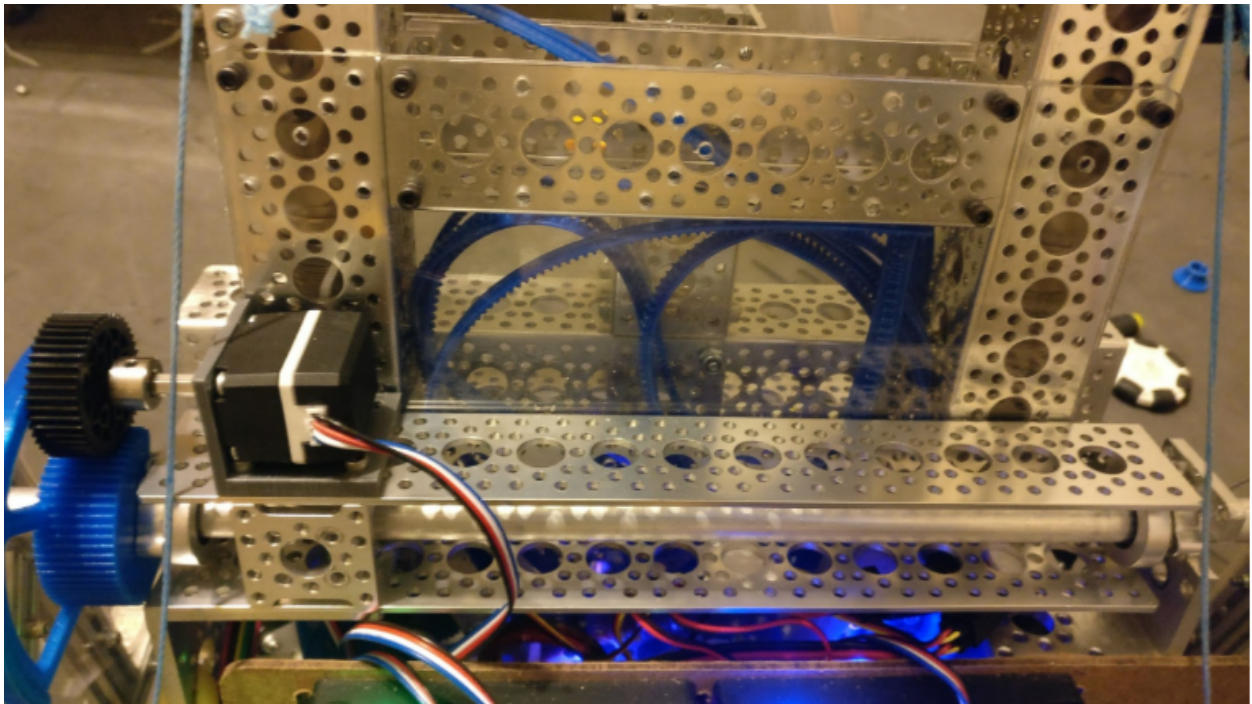
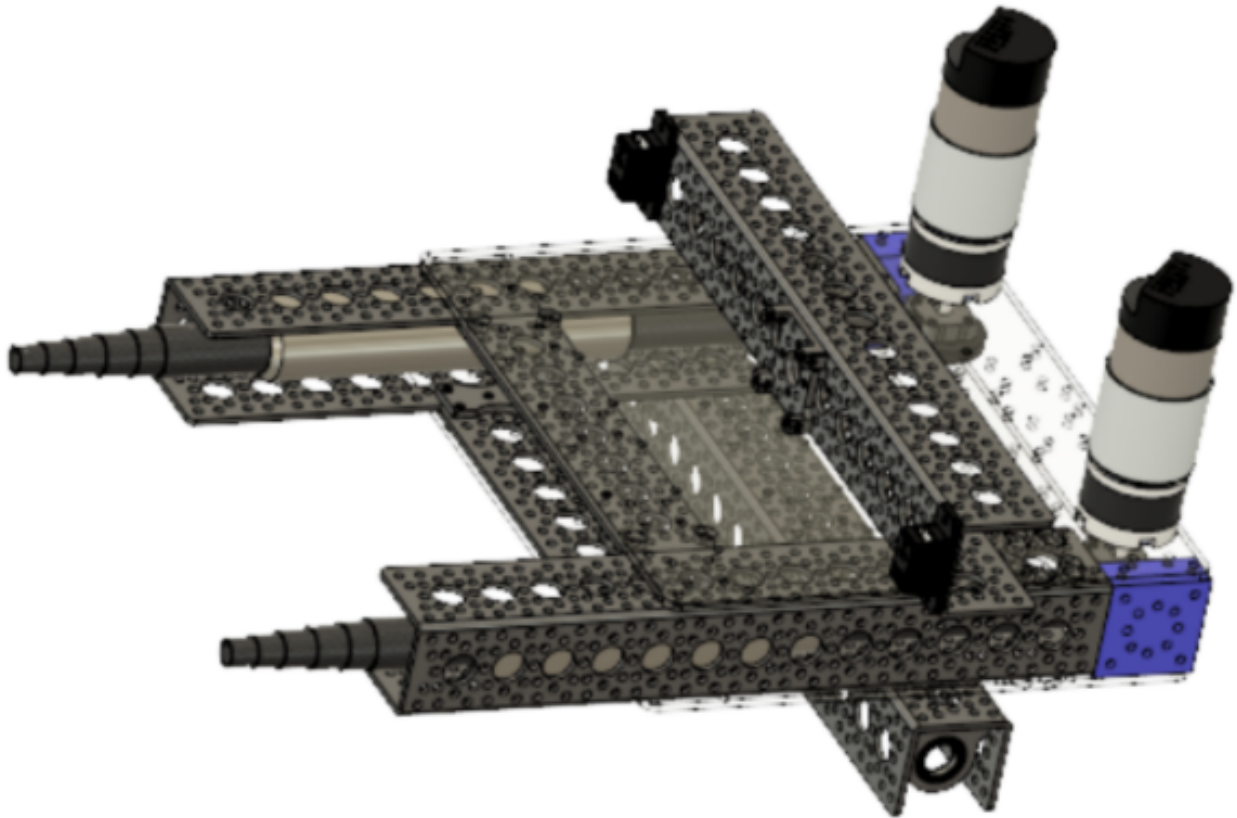


Fig. 62: 7172 Technical Difficulties, Rover Ruckus, fishing pole extension using gear rack

1.9.4 Lead Screws

Lead screws use a threaded rod to create high-torque linear motion. Their primary advantage is being able to handle much higher loads than other unmodified kit options. While this option was popularized in Rover Ruckus for hanging robots, variations have been in use since as far back as Res-Q. Another use case for lead screws is to change the angle of an arm platform, though this is more common in FRC. However, this ability to handle high loads comes at a cost: lead screws are extremely slow.

By far the most popular lead screw option is the ServoCity Linear Actuator kit. Overall, this kit is lightweight, simple, and cheap. It can be a good start for teams using REV and needing a linear extension, and is generally usable out of the box.

Advantages

- Easy way for high-load, high-torque applications
- Compact form factor
- Typically only requires one motor

Disadvantages

- Not for quick linear extensions
- High torque translates to slow extension speed



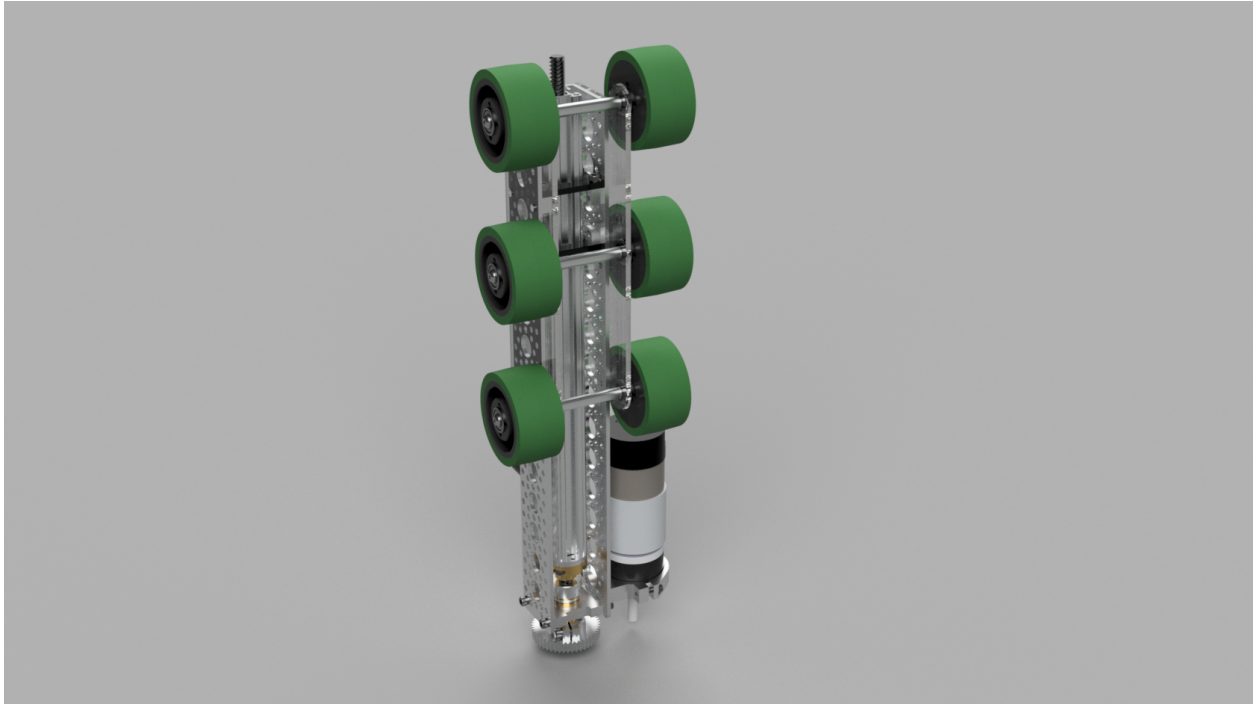


Fig. 63: 11115 Gluten Free, Rover Ruckus, Actobotics lead screw used for hang

1.9.5 Rack and Pinion

Rack and pinion refers to a circular gear (the pinion gear) attached to a shaft and meshed to a rack, which is a toothed linear gear. When the pinion gear is driven, it will drive the pinion gear upwards or downwards, depending on how the rack and pinion is mounted. Generally, rack and pinion is a good light-use option for FTC teams in terms of linear extension. However, there are some disadvantages to rack and pinion compared to the other options of linear extension. Therefore, rack and pinion is generally not recommended for teams.

Advantages

- Easy way to extend upwards.
- Power and linear motion in one package
- Can with proper support, sustain heavy load (e.g. hang robot)

Disadvantages

- Rack and pinion generally is only used in one stage, because multiple mechanisms require other forms of powering it (belt, string, chain, etc.)
- Rack and pinion needs to be supported very well to sustain heavy load, or else the mesh will fail. **It is not advisable to use the Tetrix rack and pinion in high stress conditions.**

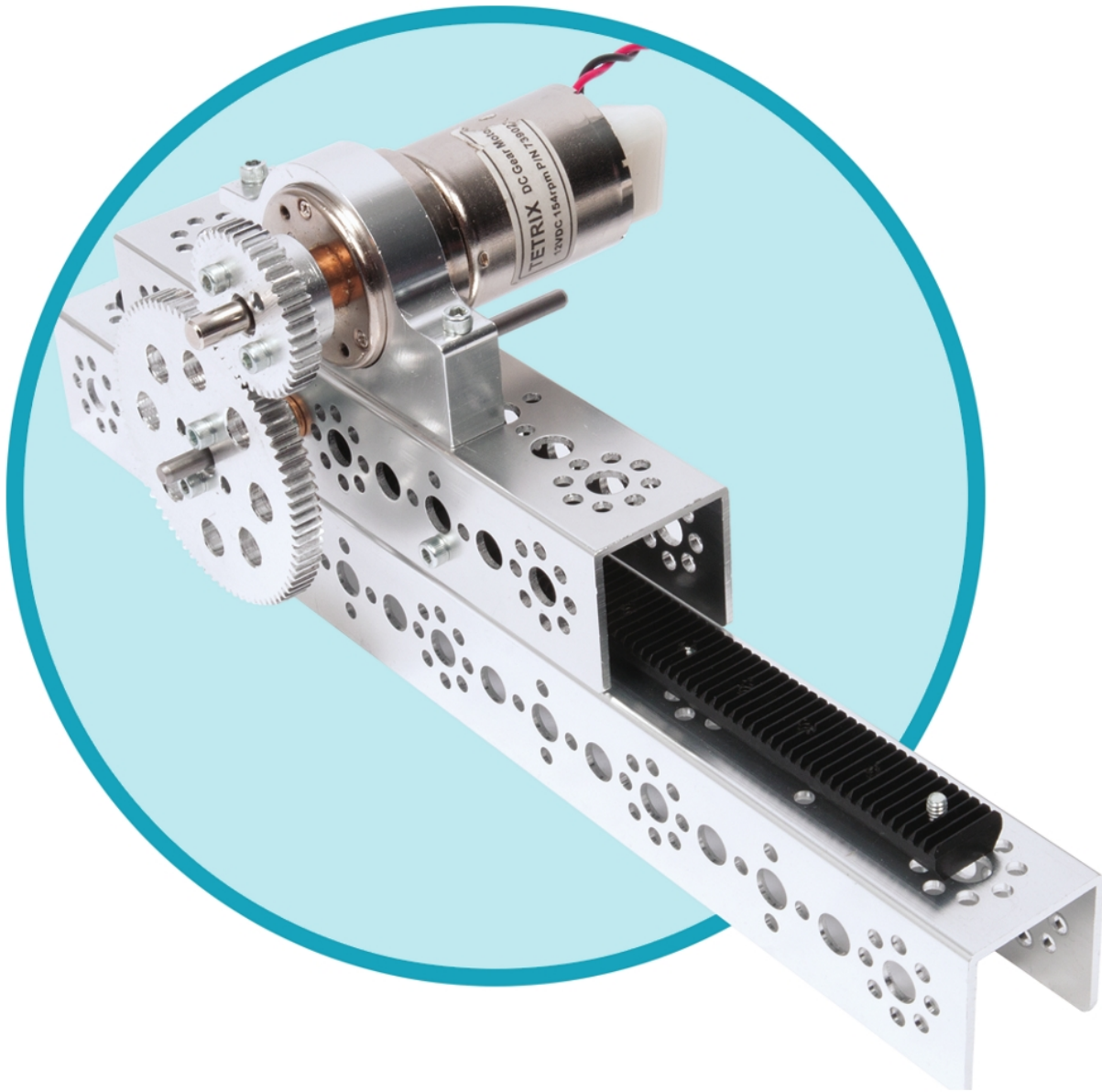


Fig. 64: The Tetrix rack and pinion has a tendency to fail under medium to high load.

1.9.6 Rigging

Rigging refers to the way that string, belt, or chain is set up to extend and retract a linear extension. This is an important and time-consuming requirement for any team that uses linear extension, so be sure to spend adequate time on it.



Fig. 65: Example rigging setup from team 7236 Recharged Green. This setup utilizes a continuous stringing setup for the first stage and a cascaded belt-driven second stage.

Continuous Rigging

Generally recommended

Continuous rigging entails rigging one long **extension string**, originating from a motor-powered spool, to the top of the base stage, then to the bottom of the first stage, then to the top of the first stage, then to the bottom of the second stage, etc. A **retraction string**, originating from a second spool on the same axis as the extension spool, is then anchored to the top stage. When the motor rotates one direction, the extension spool reels in the extension string, so it becomes shorter. In doing so, the distance between the top of one stage and the bottom of the next stage decreases, causing the system to extend. Note that **the last stage always extends and retracts before the other stages (this can be either an advantage or a disadvantage, depending on the application)**. Once the last stage hits its limit, the next to last stage extends outwards, and so on; the pattern repeats until every stage is fully extended. When the motor spins **in the**

opposite direction, the retraction string is reeled in, pulling the top stage closer to its starting position until the system is back where it started. For the retraction string, it is often necessary to add an extra pulley near the back of the extension. This is because the retraction will only retract to the farthest point, which generally is the spool. However, the spool may not be mounted at the very back of the robot - thus an extra pulley is needed. Note that for this to work, the extension string should be wrapped around the spool in the opposite direction of the retraction string. Thus, if the extension is wrapped clockwise, the retraction must be wrapped counterclockwise.

Here are some additional considerations when rigging a continuous system.

- As a general rule, continuous spools can be powered by a system with a relatively low gear ratio.
- The extension string and retraction string **do not need to be separate strings, but it is much easier to tension the system if they are separate.**
- The extension spool and retraction spool should be the same diameter.
- You need to make sure, as with any time you work with strings and pulleys, that you are pulling the string straight. Any amount of misalignment can lead to the string coming off of your pulley.
- The width and diameter of your spool should be enough such that once fully wound, your string never overlaps. The reason for this is because when the string starts to overlap, this can change the diameter of your spool, causing the tension in the string to change.

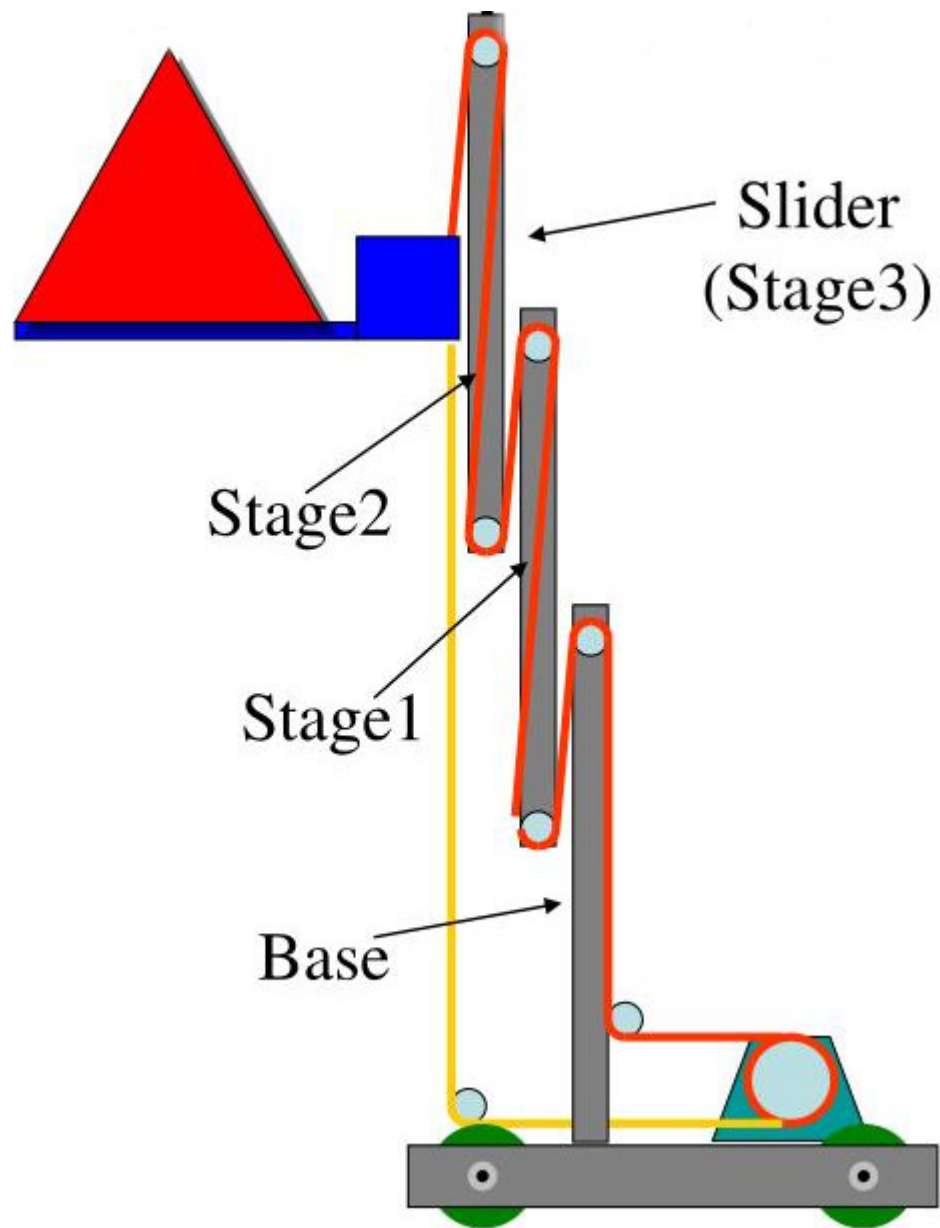


Fig. 66: Continuous rigging diagram

Cascade Rigging

Generally recommended

Cascade rigging is a bit more complicated than continuous rigging. Much like continuous rigging, an extension string originating from a spool is rigged to the top of the base, running down to the bottom of the first stage. However, instead of being rigged to the top of the stage, the extension string is anchored to the bottom of the first stage. A second extension string, anchored to the top of the base, is rigged to the top of the first stage and anchored at the bottom of the second stage. The pattern continues until all stages have been rigged. Note that **the number of strings required to extend is equal to the number of stages in the system**. When the motor rotates one direction, the extension spool reels in the first string, decreasing the distance between the base and the bottom of the first stage. This pushes the second string forward, decreasing the distance between the top of the first stage and the bottom of the second stage, and so on. Note that unlike continuous rigging, **every stage moves at the same time**. The second stage moves 2 times as fast as the first stage relative to the base, the third 3 times as fast, and so on.

A cascaded system can be retracted in three ways: using continuous retraction, elastic retraction, or reverse-cascade retraction.

Retraction Options

Cascade retraction

As the name suggests, continuous retraction utilizes the same retraction method as continuous rigging with one notable difference: if the variable N is the number of stages in the system, the diameter of the extension spool must be **N times smaller** than the retraction spool.

Advantages

- Requires less string (less strings to tension, less strings that can become loose)

Disadvantages

- Requires more space, as it demands a larger spool retraction spool

Elastic retraction

Instead of retracting using a retraction spool, one common way to retract is to attach a piece of elastic (commonly surgical tubing) to the last stage. The elastic applies a force on the last stage that is counteracted by the motor when extending. However, when retracting, the motor reels the last slide back in. While this is the retraction method recommended by many kit slide manuals, we do not recommend this method.

Advantages

- There is only one string to tension, instead of multiple, so tensioning is simpler.
- The elastic automatically tensions the extension string.

Disadvantages

- Since the elastic applies a force to the slide at all times, this force opposes the force applied by the motor when extending the slides. **Thus, elastic retraction decreases extension speed considerably.**
- The elastic does not apply a constant force at all times. It applies force proportional to the amount that the slide is extended, so retraction may not be smooth and controlled, like other rigging methods.
- It is very easy to unwind your extension spool when using elastic retraction.

Cascade retraction

Cascade retraction entails simply rigging another set of cascade string that can retract the system when engaged (see the image below).

Advantages

- Very space-efficient

Disadvantages

- Requires more string (less strings to tension, less strings that can become loose)

Additional Considerations

- If the system has only one stage, cascade rigging is **identical to continuous rigging**
- Unlike continuous rigging, each time a new stage is added to a cascaded system, the gear ratio required to maintain the same extension speed increases. For instance, if 2 stages are added to a 1 stage cascaded lift that is geared at a ratio of 3:1, the ratio must increase by a factor of $\frac{2+1}{1}$ to maintain the same speed, changing the ratio to 9:1.
- One disadvantage of cascade rigging is that each string must be kept tensioned. This is still the case with continuous rigging, but you have many more strings to keep track of, as tension must be maintained on all of them.
- You need to make sure, as with any time you work with strings and pulleys, that you are pulling the string straight. Any amount of misalignment can lead to the string coming off of your pulley.

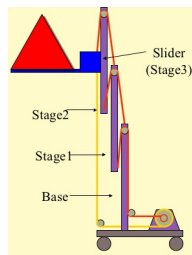


Fig. 67: Continuous retraction

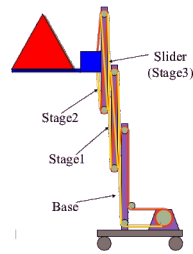


Fig. 68: Cascade retraction

Belt-driven slides

One increasingly popular alternative to traditional string-based rigging is belt-driven slides. This can be done continuously or using cascade rigging.

Advantages over string

Unlike string, belts used on slides never need to be tensioned. As discussed in the linear motion section, in order for string-driven slides to remain efficient, string tension must be maintained. Naturally, string loosens over time, so you either need a mechanism that can provide extra tension (a spring) or manually tighten string, which can get a bit tedious (especially for cascade rigging). However, belts do not have this issue. They tend not to stretch over time, meaning complex external tensioners are rarely needed. Belts are also on an automatic one to one loop, meaning that for every inch you pull in on the extension side, you always feed back out that inch on the retraction side, and vice versa. While this might not seem like a big deal, if a string winds itself up on a spool and overlaps at any point, the diameter of the spool changes, making the two spools out of sync.

Disadvantages over string

The main disadvantage of belt-driven slides are the size restriction. Simply put, belt pulleys take up much more room than the 4mm thick, 12mm diameter pulley bearings REV sells. When using belt-driven slides, pulleys are at least double that thickness and have a considerably larger diameter, meaning each stage must be thicker.



Fig. 69: 7236 Recharged Green, Rover Ruckus, **continuous** rig

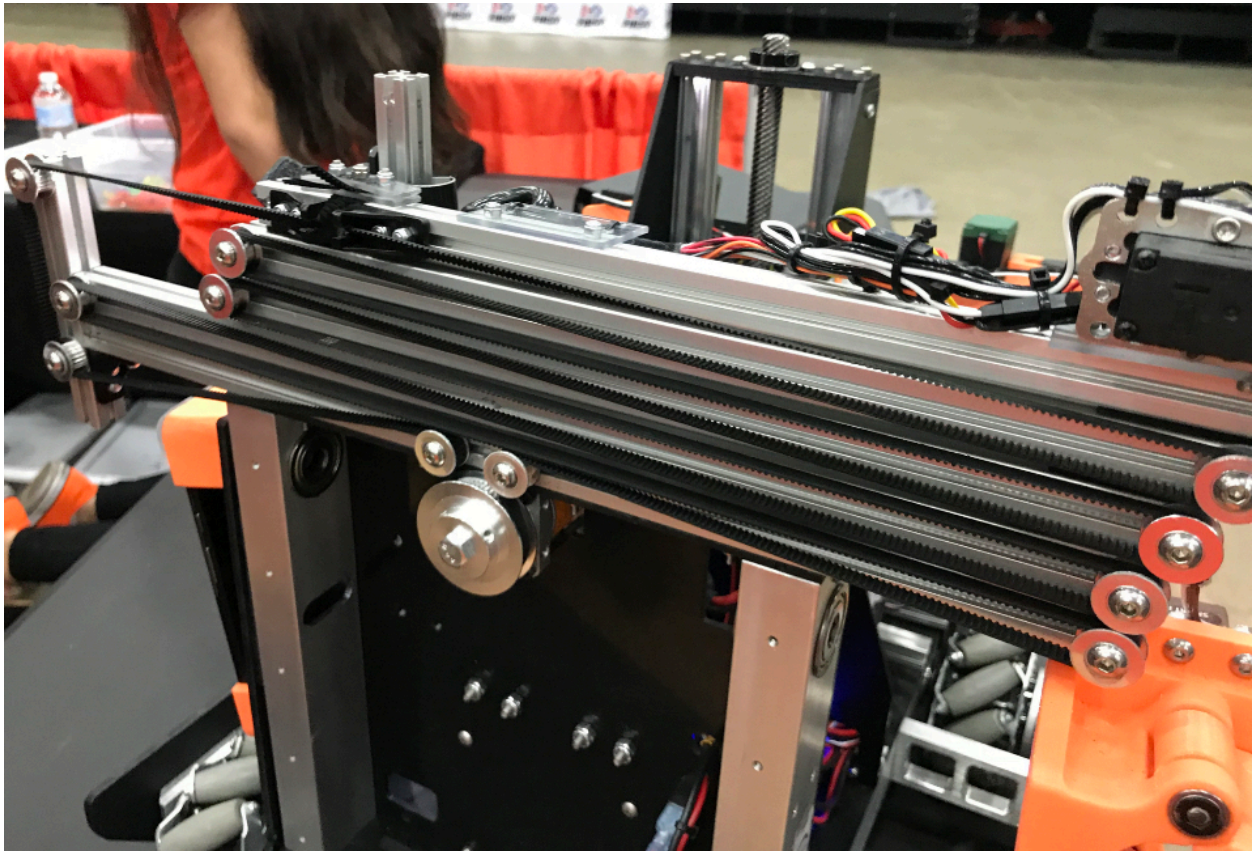


Fig. 70: 11190 Mechadojos, Relic Recovery, continuous belt rig

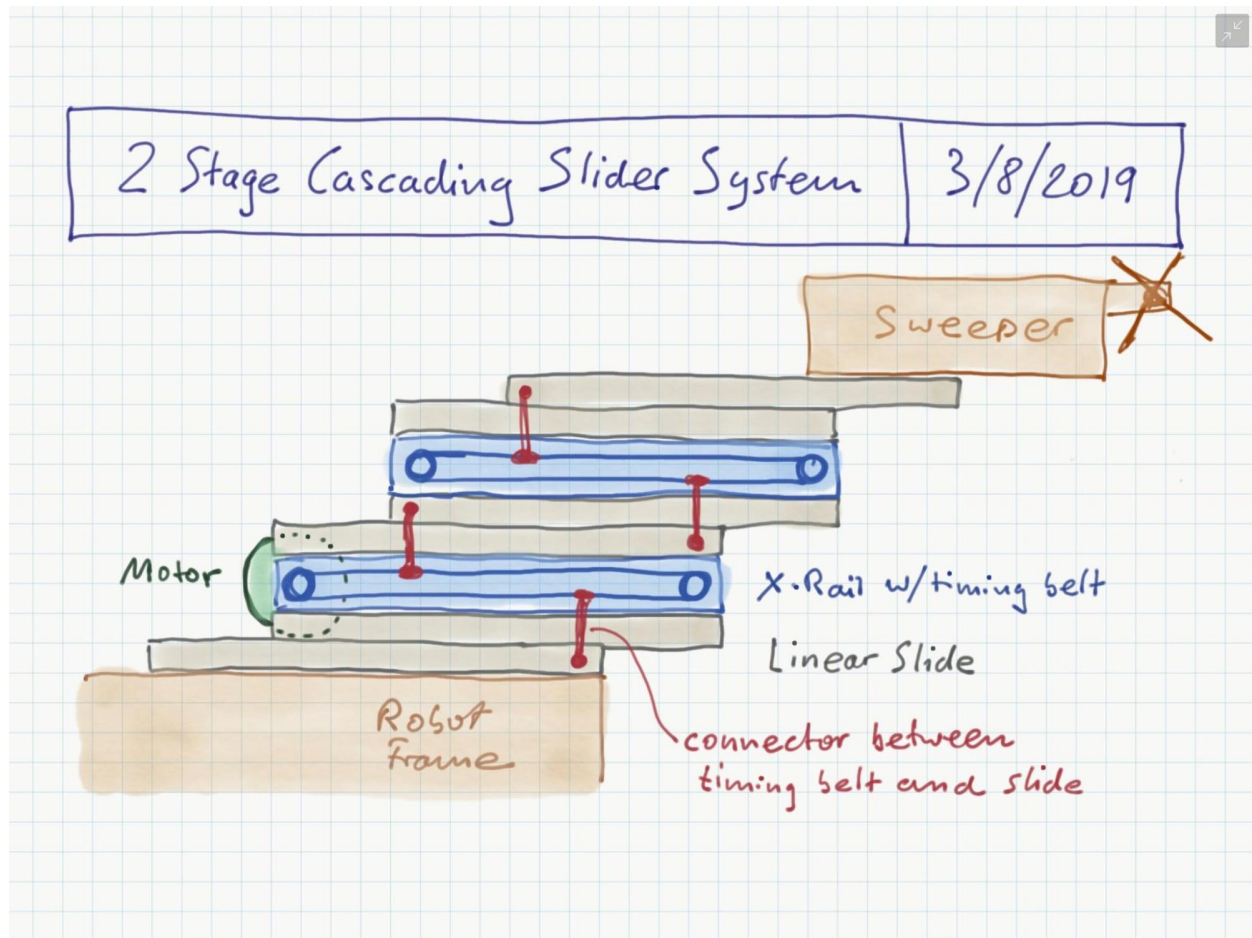


Fig. 71: 5064 Aperture Science Rover Ruckus **cascade** belt-driven proof of concept drawing



Fig. 72: 5064 Aperture Science Rover Ruckus **cascade** belt-driven final version

Belt or chain driven elevators

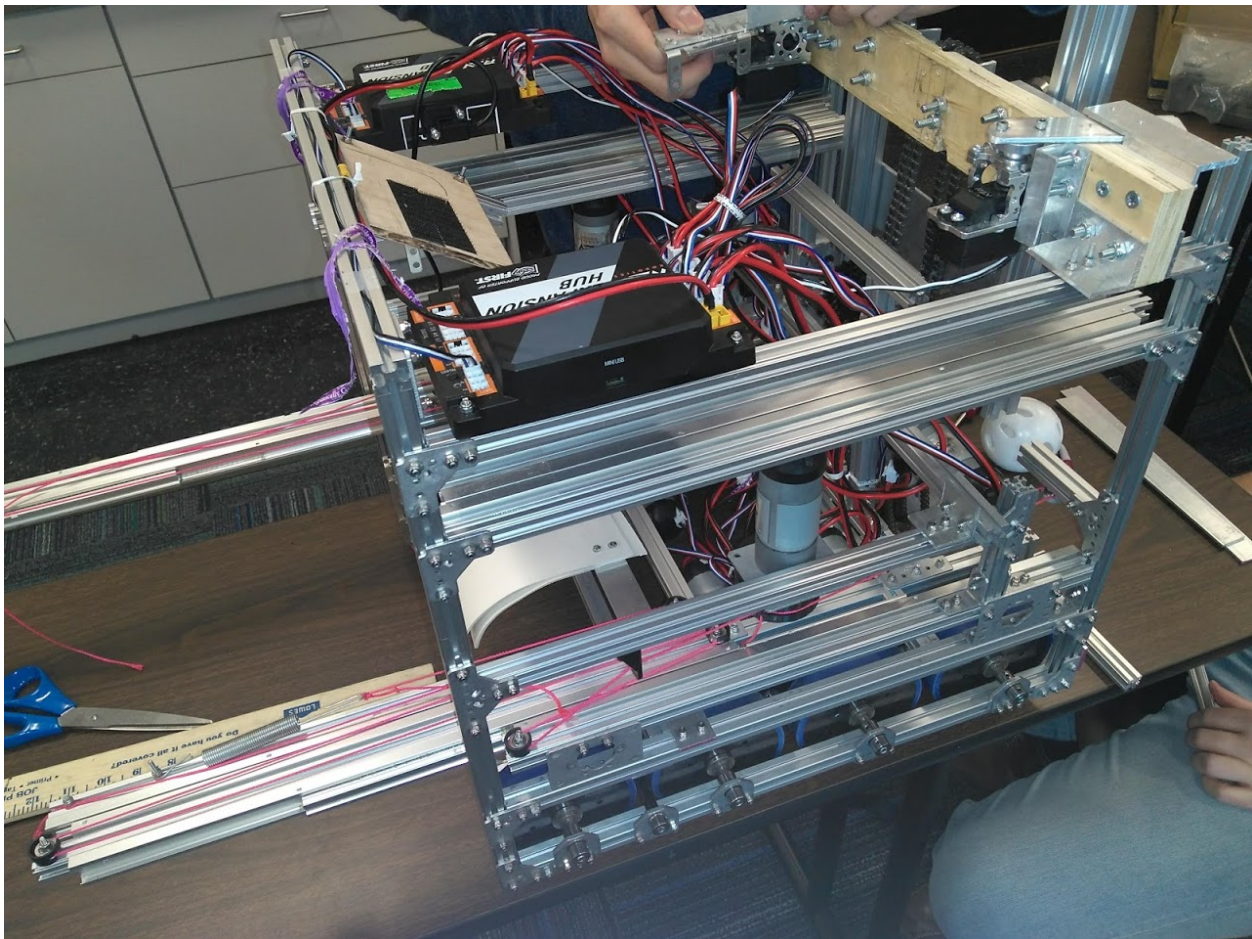
Almost ubiquitous in FRC, belt and chain elevators have existed seemingly forever. The basic premise is to have sprockets or pulleys mounted at the top and bottom of the elevator to the robot superstructure. Then, the belt or chain is hard bolted to the elevator near the bottom sprocket or pulley. When the chain or belt is driven, the elevator will move up and down. It is possible for elevators to have multi stage designs, but powering them will be more complex.



Fig. 73: 7236 Recharged Green, Rover Ruckus

Tensioners

Tensioning string is one of the most painstaking tasks for a builder in FTC. Ensuring that both sets of slides are tensioned evenly can be an arduous and annoying job. However, adding tensioners to your strings can help solve the uneven tension and ensure that both sides of slides run together. The most common type of tensioner is a spring that can be purchased at a hardware store. It generally will be placed at the end of the string run, near the part that extends farthest out from the robot's center. By doing so, the string will stretch out when the spool extends the arm, keeping tension so that the string does not detach from the pulleys in the linear slide extension. Another form of tensioner can be a spring-loaded pulley. Since the pulley is spring-loaded, it will take up the slack in the string. Alternatively, it is possible to mount a pulley on a piece of extrusion, and slide it so the string is taut. There are two main reasons that tensioners are highly recommended in string based linear slides, the first and oftentimes more important one, is that at its core, a run of string is a series of polygons, and as the slide extends the effective size of that polygon changes. Meaning that when fully extended, your slide may require more or less string compared to when halfway extended. Or partially extended. The closer each set of bearings are, the less this impacts string tension. Another (but less impactful) thing to keep in mind that spool size changes as string is added/removed from the spool. If the spool radius increases, the speed of the extension will also increase, and torque, which opposes speed, will decrease. Consequently, tension will change as well. Therefore, a bit of slack is inherent in all linear extension designs using string. It is highly encouraged that teams have at least one spring tensioner per linear slide set.



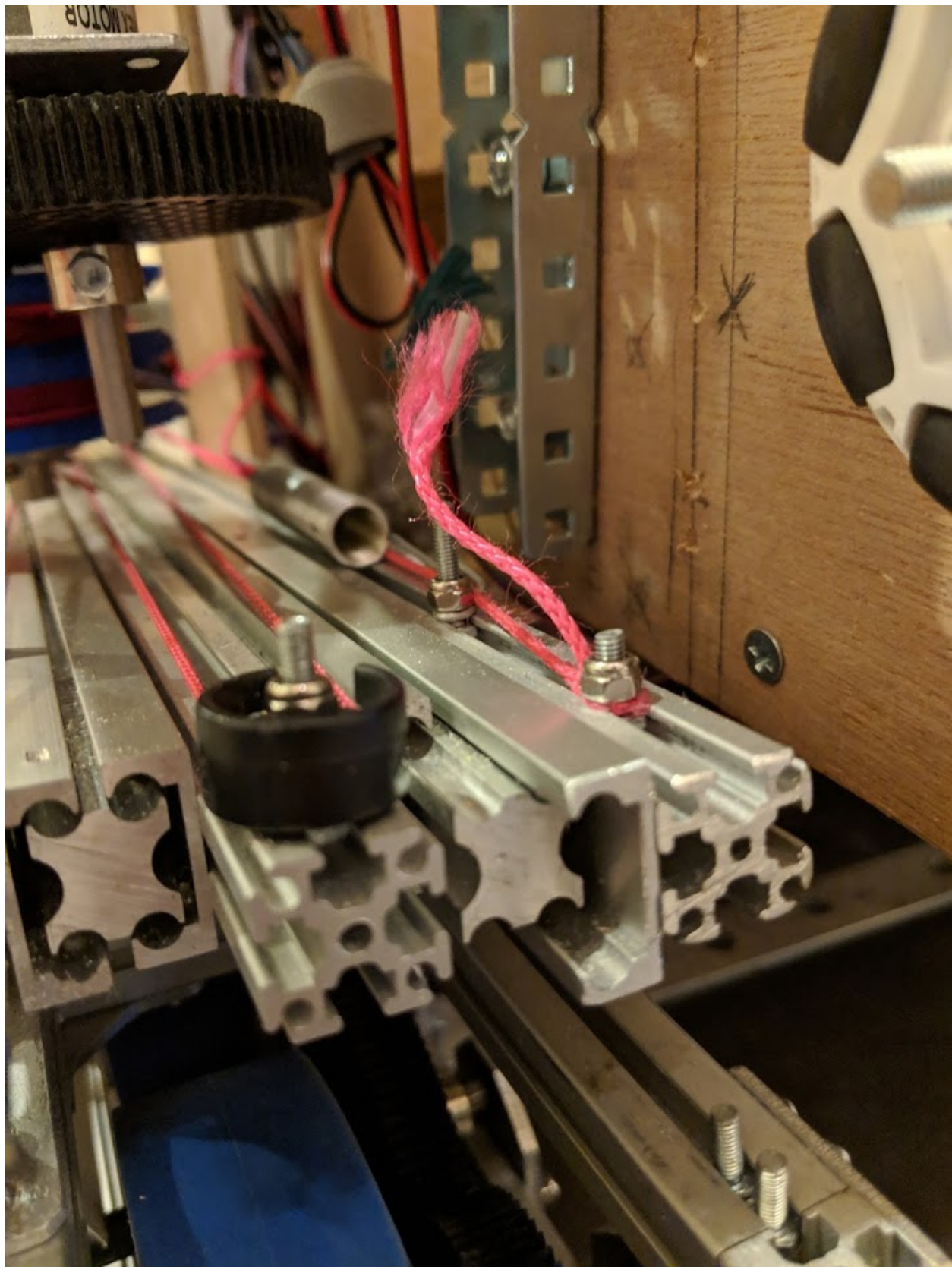


Fig. 74: 13075 Coram Deo Robotics, Rover Ruckus, string tensioner

Picking the right spool size

Spools have a special property that isn't often discussed, but is extremely useful when creating linear slide systems. Just as the system's speed and torque can be changed by changing its gear ratio, speed and torque can also be changed by changing the spool size. The motor rotates the spool at a constant angular speed. Thus, the translational speed (the speed of the slide) is proportional to the radius of the spool, and since torque is inversely proportional to speed, changing the spool size changes torque as well. This is important to recognize, as changing spool size is often more convenient than changing gear ratio to get the desired combination of speed and torque. To illustrate this, say you have a linear extension system with a 3.7:1 gear ratio. You then decide that a 5:1 gear ratio would provide a more desirable combination of speed and torque than your current 3.7:1 ratio. In many cases, instead of swapping gearboxes, it makes more sense to swap out spools to a smaller one. If your spool is currently 2 inches, your new size should be $\frac{2 \times 3.7}{5}$ inches to achieve the same result. You also need to make sure that when fully wrapped on the spool, your cable or string doesn't overlap. Overlapping can result in a change in spool diameter, which will change the tension in your string.

Cable management

When extending outwards, wire management becomes increasingly important. Obviously, it is a necessity to use wires slightly longer than the extension length. However, it is not recommended that these wires are left unprotected, as they can get tangled or caught in the slides much more easily than with protection. In general, teams should ensure that wires never protrude outside the structural parts of the robot, because they can get caught on other robots or game pieces. This can be accomplished by cable ties or Velcro ties, or by using acrylic plate to keep wires inside. However, for linear extensions, other forms of cable management are needed. The two types of cable management recommended are cable carrier and retractable coil cord. Refer to the *Electronics and Wiring* (page 166) section for more information.

Cable Carrier/Drag Chain

Cable carrier, the standard wire management method within industry, is plastic chain links with a hollow center. Cables are placed inside the chain, allowing the system to extend indefinitely. The links are somewhat stiff yet flexible, allowing cable chain to bend when the extension is retracted and straighten when extended. They typically are stiff enough not to sag excessively when retracted.

Here are some links to various drag chain products:

- [igus cable carrier](https://www.igus.com/info/energy-chains-e2-micro-small-cable-carrier)²¹
- [uxcell 10x10mm drag chain, from Amazon](https://www.amazon.com/uxcell-InnerH-InnerW-Plastic-Carrier/dp/B01LX02PSW/ref=sr_1_1?keywords=drag%2Bchain&qid=1566188144&s=gateway&sr=8-1&th=1)²²

Advantages

- Difficult to get tangled/hooked onto other objects or robots
- Sturdy and durable
- Protects wires very well

²¹ <https://www.igus.com/info/energy-chains-e2-micro-small-cable-carrier>

²² https://www.amazon.com/uxcell-InnerH-InnerW-Plastic-Carrier/dp/B01LX02PSW/ref=sr_1_1?keywords=drag%2Bchain&qid=1566188144&s=gateway&sr=8-1&th=1

Disadvantages

- Large form factor, takes up a lot of space
- Links need to be added if additional extension is needed
- Can be on the heavy side, especially with a long length of drag chain



Fig. 75: 7236 Recharged Green, Rover Ruckus Cable carrier on the left side of their horizontal slides and the right side of vertical slides

Retractable Coil Cord

While not common within industry, coil cords are still very common (coil cord is a nearly ubiquitous staple of older telephones). Retractable coil cord is more flexible than cable carriers, stretching when extended.

Here are some links to coil cord products:

- [Cable Science NEC \(non-electric cord\)²³](https://www.cablesience.com/coils/nec/nec-series.html)

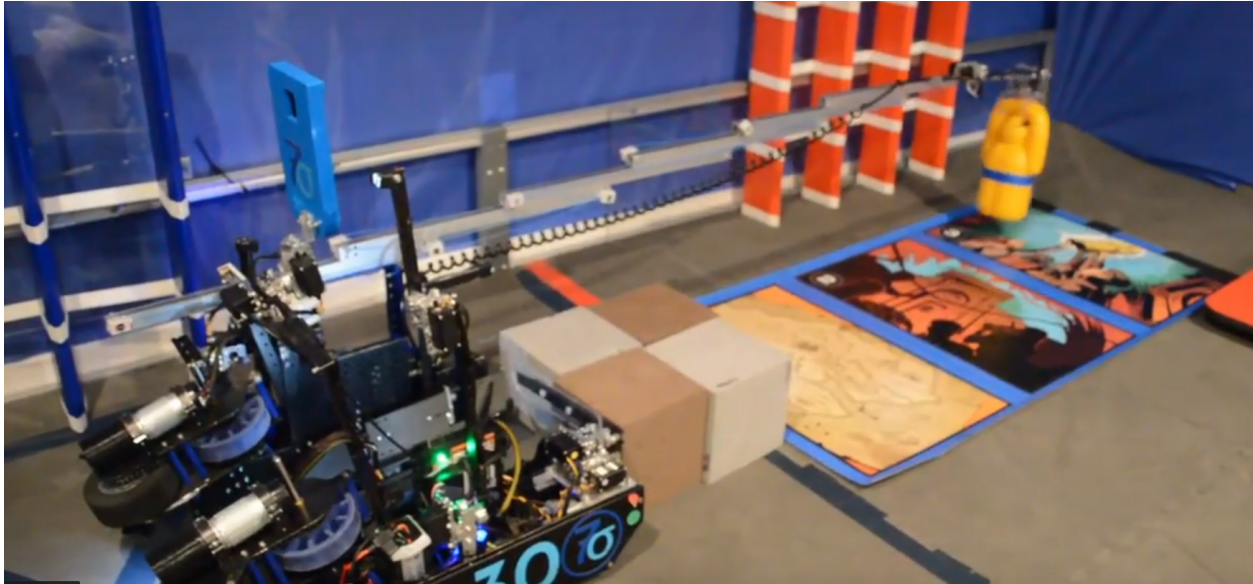
²³ <https://www.cablesience.com/coils/nec/nec-series.html>

Advantages

- Very space-efficient
- Flexible and can usually extend to any length needed (unlike cable carriers, no new links ever need to be added)

Disadvantages

- Can get tangled more easily, as it is less stiff than drag chain



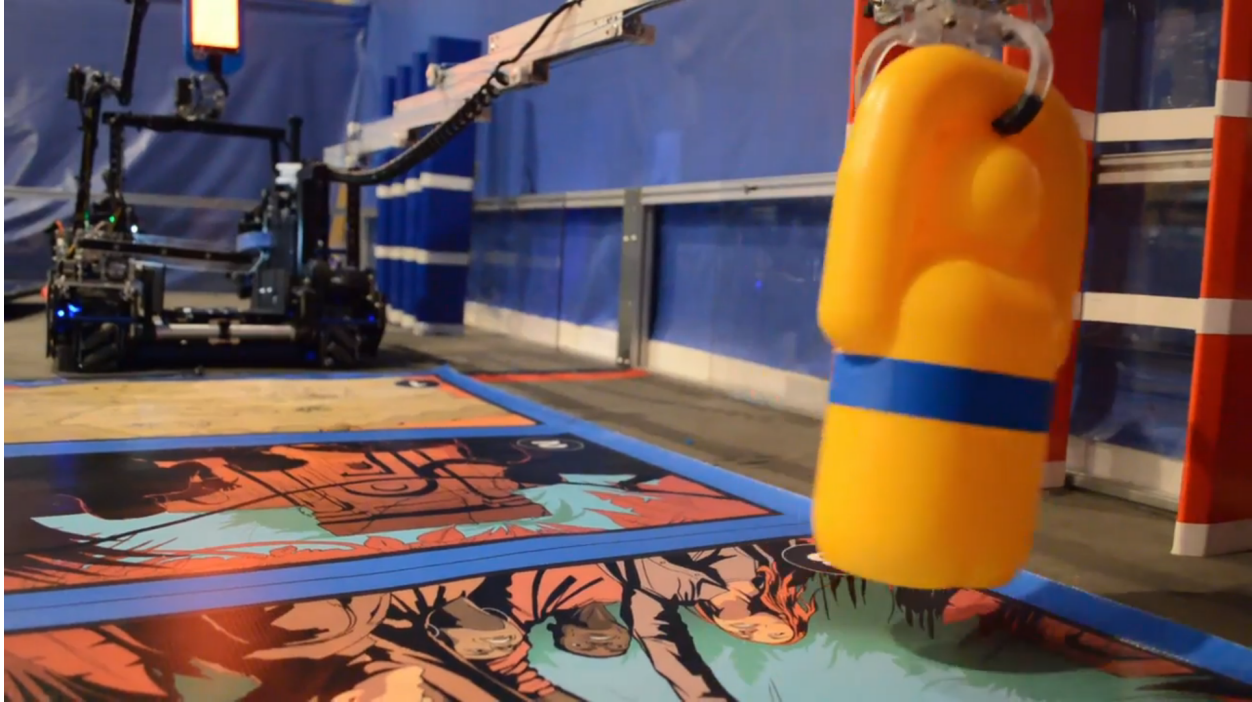


Fig. 76: 10030 7 Sigma Robotics, Relic Recovery Coil-cord on horizontal extension mechanism

1.10 Arms

Arms are another way to achieve extension past the 18" x 18" dimension of the robot. Unlike linear extensions, arms require lots of torque - a standard 40:1 or even 60:1 gearbox will not be suitable in most applications. For example, many teams will run a 256:1 gearbox for their rotation motor. A budget option is the goBILDA planetary 188:1 which gives a high gear ratio at a relatively low price of \$30. Note that these will have to be very well supported, more so than the VersaPlanetary or Sport options (more expensive and more robust alternatives). Rarely should an arm be directly mounted to the driving motor. Instead, torque should be transferred via gear, chain, or belt. Large arms can also be hard to control (with the addition of momentum adding load to the gearbox, it is hard to stop a three-foot arm that weighs five pounds quickly without breaking a gearbox). In many cases, this issue can be mitigated with software (see the [Control Loops](#) (page 196) section).

Depending on the application and implementation, arms can be either faster or slower than extension options.

The different types of arms in FTC include single arm, chain bar and multi-axis arms.

1.10.1 Advantages

- Single bar arms can be relatively simple to build.
- Arms can be useful in low-load applications; however, most mechanisms in FTC are not very light.

1.10.2 Disadvantages

- Arms require an insane amount of torque, and in order to do so, teams must purchase more expensive gearboxes not found in kits, such as the Andymark NeveRest Sport or VersaPlanetary.
- While single arm arms may be more simple, they cannot provide enough extension for most games.

1.10.3 Types of Arms

Single Arm

The most simple type of arm in FTC, a single arm refers to an arm on one axis of rotation. While it is possible to successfully build this kind of arm, generally a single axis arm will only afford around 15-16" of extension, which is inadequate for nearly every game. The reason for this is that the longest a channel can be is 18" (technically you could have a longer channel by placing it diagonally, but this complicates matters). Thus, with a maximum of 18" of extension, a couple inches must be subtracted, since the point of rotation is inside the 18" sizing cube; therefore the extension is around 15-16". Therefore,**a single arm with further linear extension is optimal**. For example, some teams built a single arm with an added linear extension mechanism to reach the desired extension length, which is generally >24".

The advantages of a single arm are that it is relatively easy to build, and can be a quick way to gain some form of extension outside of the robot cube.

However, there are many disadvantages such as having a high gear ratio, requiring much more support than a linear slide, and being hard to control without the proper software.

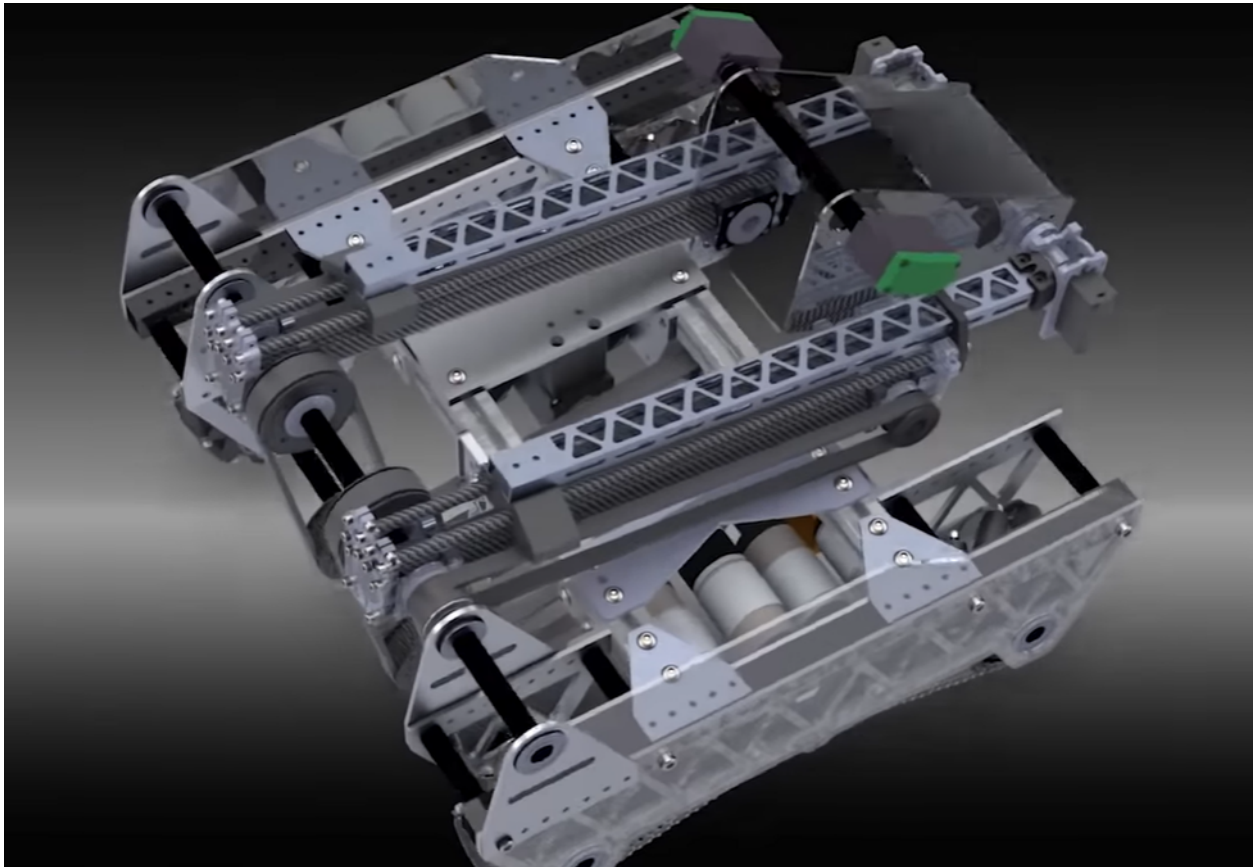


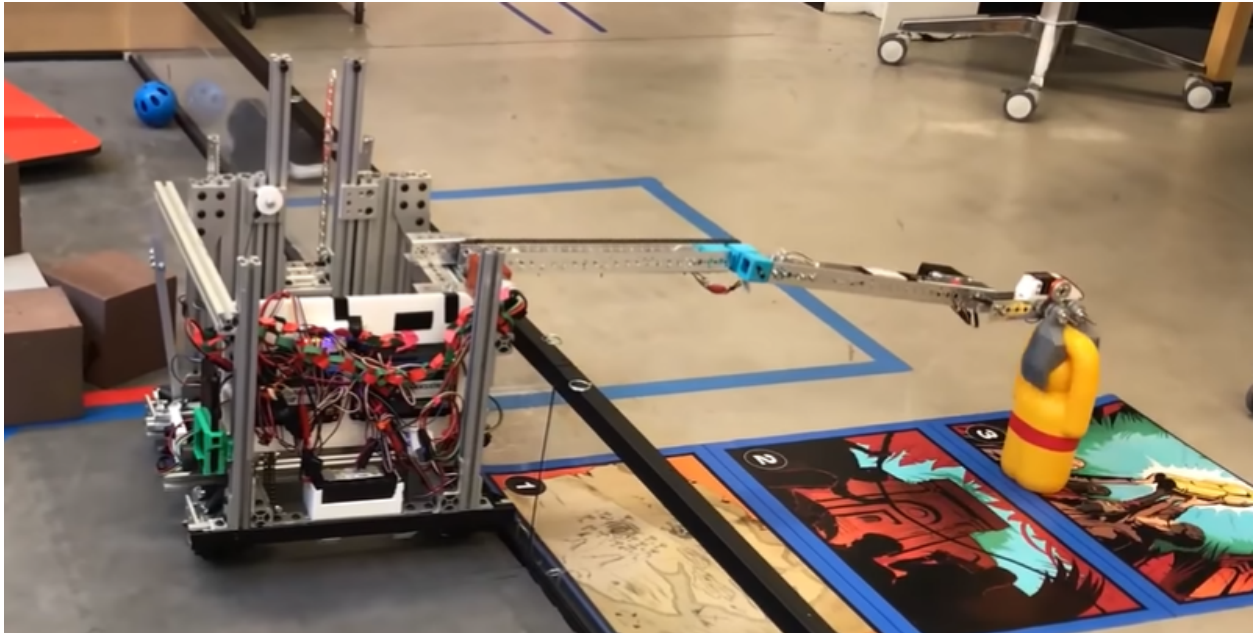
Fig. 77: 8103 Null Robotics, Rover Ruckus, single arm + custom belt driven linear extension

Chain Bar Arm/4-Bar Linkage

A chain bar arm is a special type of multi-axis arm that involves chain hard-bolted to a bar. The purpose of a chain bar arm is to make an extension such that the outer bar always keeps the same angle in reference to the ground. This means that if the outer bar is parallel to the ground in the stow position, it will remain parallel to the ground at all points in the extension of the arm. This is especially important if, for example, you have a claw to pick up an object, and you want the claw to remain parallel to the ground.

Note that if your sprockets are of different lengths, you can mechanically rotate a between two desired angles without needing to expend a servo slot.

Chain bar arms can provide the >24 optimal extension needed to be competitive in games, but it also has disadvantages. It is much harder to control than a linear slide, since a chain bar arm has momentum. Thus, it is very difficult to be precise with a chain bar.



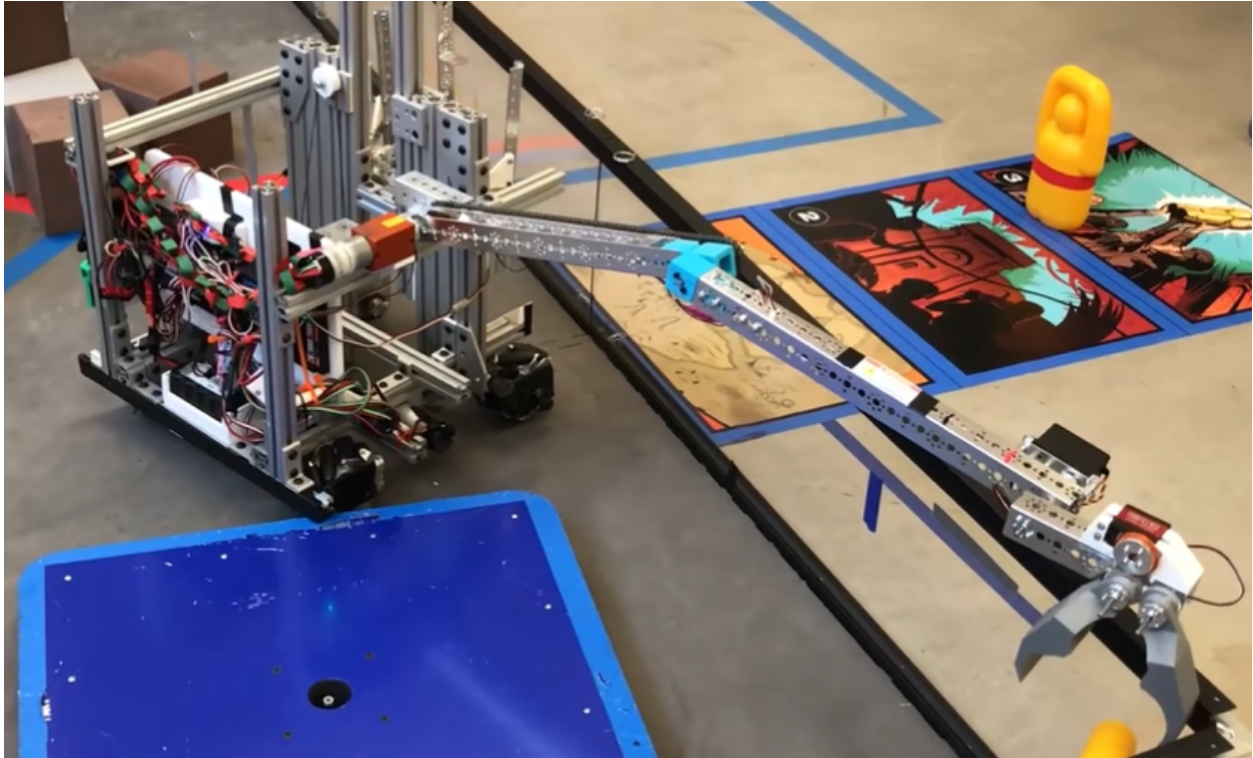


Fig. 78: 8103 Null Robotics, Relic Recovery

Multi-Axis Arm

A multi-axis arm is an arm which has multiple points of rotation. It is highly discouraged for inexperienced FTC teams due to the difficulty of building as well as the need for machine tools.

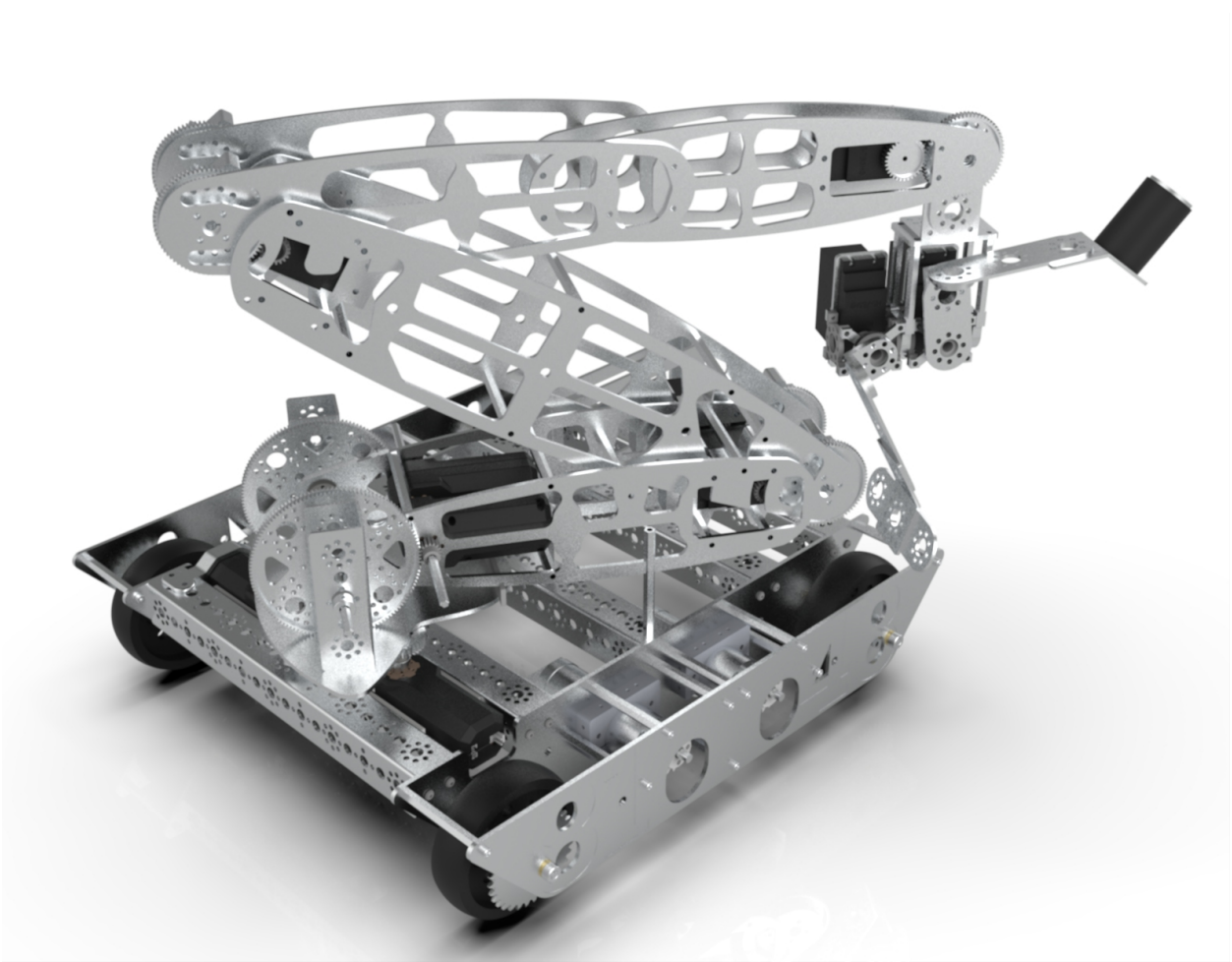


Fig. 79: 8148 Aleph Bots, Relic Recovery

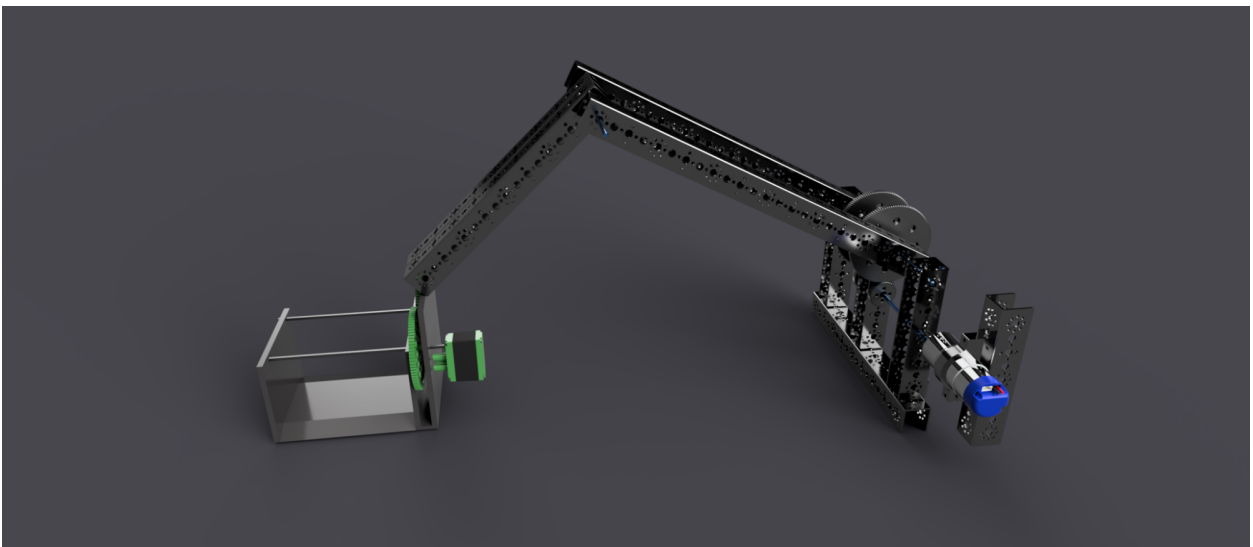


Fig. 80: 15692 Rust In Pieces, Rover Ruckus

1.11 Passive Intake/Claw

Passive intakes are mechanisms in which the driver must pick up individual items, usually by actuating some sort of gripper, which holds the game element. Claws are the most common sort of passive intake and probably one of the first mechanisms that jump to mind during brainstorming. Unlike most intakes, claws can be made with only servos, and thus don't require valuable motor slots. This is not to say there are no disadvantages; notably, unlike intakes, when you pick up something with a claw, it isn't easy to transfer the collected element to a separate scoring mechanism. For this reason, in most cases, claws double as both a collection method and a scoring method. **Claws are extremely useful for picking up large, unwieldy elements that cannot be controlled by active intakes.** For example, the relic in Relic Recovery would be much harder to control using a more active intake; so teams used a claw. **However, claws are outclassed by active intakes for picking up small game pieces in nearly every game.**

1.11.1 Advantages

- Much simpler than an intake
- Takes a lot less time to build than a reliable intake + transfer
- Less moving parts = less things that can go wrong
- Usually does not take up a motor slot
- Can pick up odd shaped game elements that active intakes can't

1.11.2 Disadvantages

- **Depending on their use case, claws are slower than intakes.**
 - The driver must angle the robot, position the claw, and grab the game element.
 - By contrast, with an intake, the driver needs only to position the intake and grab the game elements.
- Claws can be more fragile and break easily.
- In cases in which claws are slower than intakes, they can be much more prone to defensive maneuvers. The more time spent collecting, the more vulnerable you are.



Fig. 81: 11115 Gluten Free, Finalist Alliance Captain (Detroit), Relic Recovery



Fig. 82: 8680 Kraken-Pinion, Relic Recovery

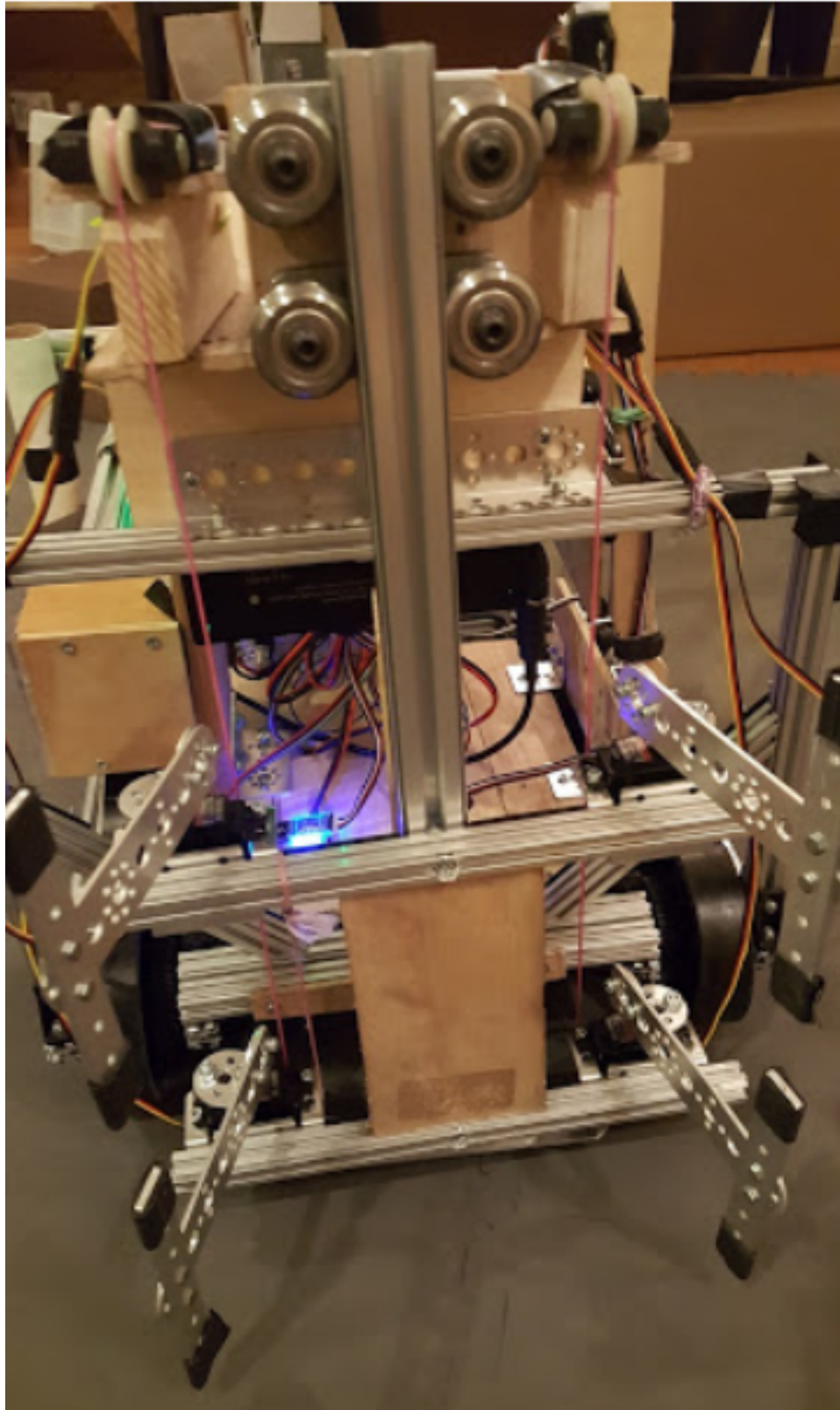


Fig. 83: 13075 Coram Deo Academy Robotics, Relic Recovery (**suboptimal use case**)

1.12 Active Intake

There's one main rule when it comes to intakes in FTC: you'll always need one. There are multiple parts of an effective intake: intake geometry, intake type and material, and speed. However, a common rule of them is the faster the intake, often the more effective it is. For example, near the end of Rover Ruckus, many of the top performing teams used a 3.7:1 (1600 RPM) motor or a 5.2:1 (1000 RPM) motor on their intake. **While not a necessity, it is highly recommended that teams dedicate at least one motor to their intake if possible.**

“Touch it, own it”: the JVN school of picking stuff up

When your intake comes in contact with game elements, you want it to instantly be controlling them. You should design with a margin for error, both in the orientation of the game piece and in the alignment of the robot. For example, look at FRC team 694's ball intake on their 2019 robot:



Fig. 84: FRC 694 StuyPulse, Deep Space

When this robot drives up to a ball, it barely has to point itself at the ball and it's instantly locked in between the two rows of wheels. What makes it so effective is the immense amount of testing that went into its design. The team tried a number of intake shapes, initially building with wood and rubber bands, and recorded which shapes were most effective. By testing their intake design out before competition, they didn't have to guess whether it would work as intended: they put it on the field with confidence.

1.12.1 Principles of an Intake

Reliability

- An intake must be reliable, as picking up and scoring game elements is the primary method to gain points in all FTC games
- Intakes have many moving parts and are susceptible to breaking, especially at high RPM. Thus, the intake material must be durable to withstand long periods of operation.
- Intakes often stick outside of the robot frame perimeter. In this case, durability becomes extremely important; the intake must be built so that they can withstand impacts/collisions with other robots or parts of the field. There are two ways to accomplish this - either by building the intake very robustly (lots of support so it doesn't break), or making the intake flexible (for example out of lexan or spring loading it) so that even though it may bend due to impacts, it will always spring back into place. Another way to prevent intake breaking is to make a fully retractable intake that won't protrude outside the 18" cube, though this is seldom needed.

Consistency

- The intake **must** be able to pick up game elements consistently. For example, in Rover Ruckus each robot could only possess 2 minerals at a time. Therefore, the intake should only pick up two pieces at a time, not zero, one, or three. This is a common mistake that many inexperienced teams fail to take into account.
- Another component is the varying angles that the game objects can be located in. This was especially apparent in the Relic Recovery season, where the "glyphs" (foam cubes) could be oriented in many directions. Even though it was relatively easy to make a compliant wheel intake that could effectively intake glyphs in one direction, it was hard to make an intake that could deal with angled glyphs. Being able to intake glyphs in all orientations was especially important for multi-glyph autonomous modes.

Controllability

- The intake must be able to consistently control the game elements. For example, if the intake is too fast and the collection box is not well designed, then game pieces might fly out. It is possible for pieces to get jammed at an unreachable angle, especially when using wheeled intakes. Optimally, the minerals should follow a certain path each time if funneling is done correctly.

Efficiency

- The key to any successful robot is cycle time. Reducing cycle time by having an efficient intake will lead to major improvements in score. A good intake should take no more than a few seconds to successfully collect the needed elements. For example, in Relic Recovery, the best intakes often had a <3 second collection time for two game elements, and in Rover Ruckus a 1 second collection time was desired at the highest level.
- A key rule to remember in FTC is the shortest distance rule: how can you get minerals from A to B in the shortest distance? The answer is usually a couple straight lines. The closer the minerals follow this path, the faster they will go from collection to deposit. Don't make overly long routes unless needed.

1.12.2 Types of Intake

Horizontal/top Intake A horizontal intake generally is a wider style of intake that has the intake rotating on a horizontal axis or plane. Horizontal intakes are generally used for intaking smaller game elements,

as these types of intakes can control more than one piece at a time. Horizontal intakes have been used successfully in games such as Res-Q and Rover Ruckus, where teams needed to pick up small cubes and balls from the floor.

Vertical Intake A vertical intake typically has wheels or other intake components rotating on a vertical axis (the z-axis). Vertical intakes are more controllable, as they can only pick up one element at a time. Vertical intakes excel at picking up large game elements which would be impossible to control more than one piece at once, such as the foam glyphs in Relic Recovery.

Below are different implementations of horizontal and vertical intakes.

Roller and Wheeled Intakes

Roller and wheeled intakes refer to types of intakes which work by having some sort of hard or pliable object rotate along an axis. Wheeled intakes use different kinds of wheels (solid traction, compliant, or foam wheels) that propel the game element to the collection bin. Roller intakes are wider, and can sometimes intake more than one game element.

Solid Wheel Intake

Solid wheel intakes use wheels such as grip and traction wheels to pick up game elements. They can be effective in picking up large game elements which do not have much tolerance difference. However, compliant wheel intakes generally will be more effective because compliant wheels offer more adjustability and forgiveness for the game elements. Compliant wheels also accounting for different game element orientations (the game pieces can get rotated when being controlled) and intake angles (the orientation of the game piece before it reaches the intake).

Compliant Wheel Intake

The compliant wheel intake is most commonly used with large game elements such as the glyphs in the 2017-2018 season, Relic Recovery. In this game, robots had to pick up glyphs, which were 6 inch foam cubes, from the center pit and place them in the cryptobox. This game had many wheeled intakes primarily because the wheels had consistent and controllable contact with the glyphs. Wheeled intakes were able to propel the glyphs in a consistent fashion from the point of contact to the deposit plate, which would flip up to deposit the glyphs.

Wheeled intakes can be spring-loaded or locked into place. A spring-loaded intake is able to pivot in order to accommodate a game element when it is passing through the intake, but will snap back when the element has gone through. A spring-loaded intake requires more thought, but guarantees that the intake wheel will always be in contact with the desired element. A locked intake simply means that the wheels are locked into place and cannot pivot. Teams could choose one or mix; in Relic Recovery, some teams spring-loaded the set of wheels that made contact first, and then had a fixed set in the back. This is up to the design team's choosing.

Wheeled intakes operate at much slower RPM than surgical tubing intakes, as wheeled intakes are meant to pick up **one** element at a time. They generally require more torque than a surgical tubing intake, which is geared for speed.

Advantages

- Very controllable

- Propels elements to desired location
- Great at picking up large elements

Disadvantages

- Picks up only one element at a time
- Elements can get stuck in a bad position
- Not generally used for picking up small elements
- Can generally only pick up one type of element

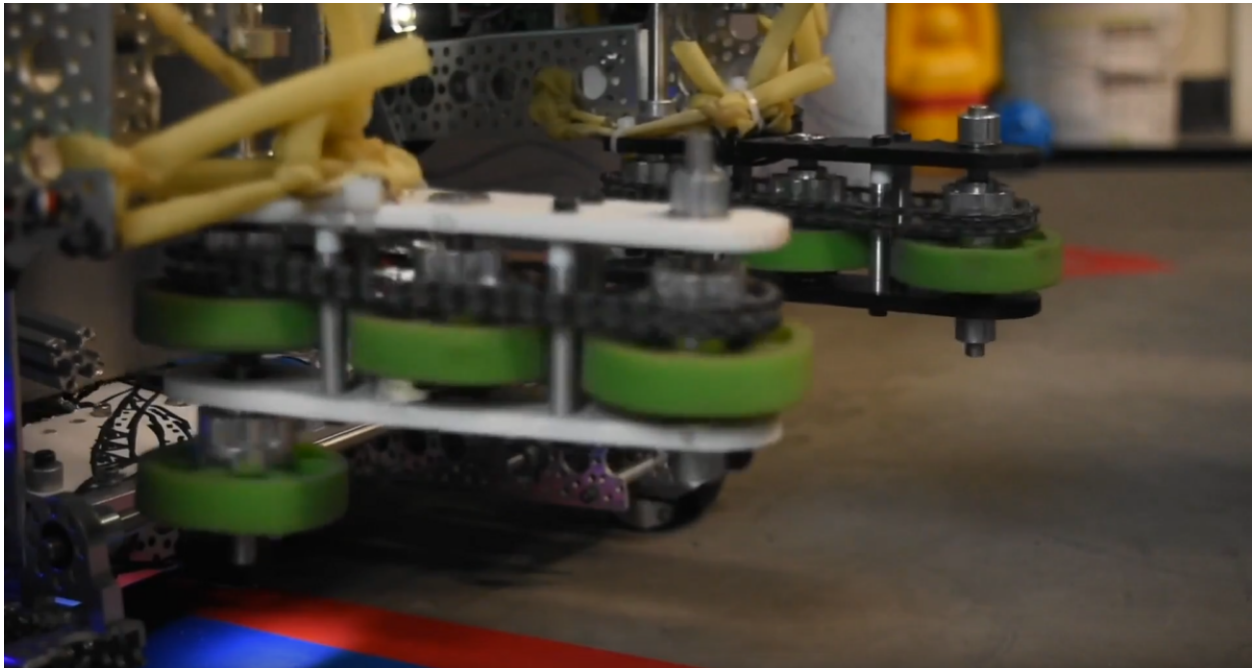


Fig. 85: 9971 LanBros, Finalist Alliance 1st Pick (Detroit), Relic Recovery, springloaded

Foam Wheel Intake

A foam wheel intake has the same principles as a compliant wheel intake, except that it uses foam wheels. It is generally recommended that teams stick to compliant wheels as they are grippier and easier to control.

Rubber Band Intake

Rubber band intakes, commonly used by in the VEX Robotics Competition, generally feature sprockets, wheels, or gears at two ends, with rubber bands interlaced in between to form a pliable and bendable roller. Generally, it can be actuated or adjustable with a servo, although this is not necessary. Rubber band intakes are great with intaking balls, but not so great with other types of game elements such as cubes. It generally is slower than a surgical tubing intake, and requires multiple stages to transfer elements from collection to deposit. Zip ties can be added to increase the intake's range to accommodate for smaller balls.

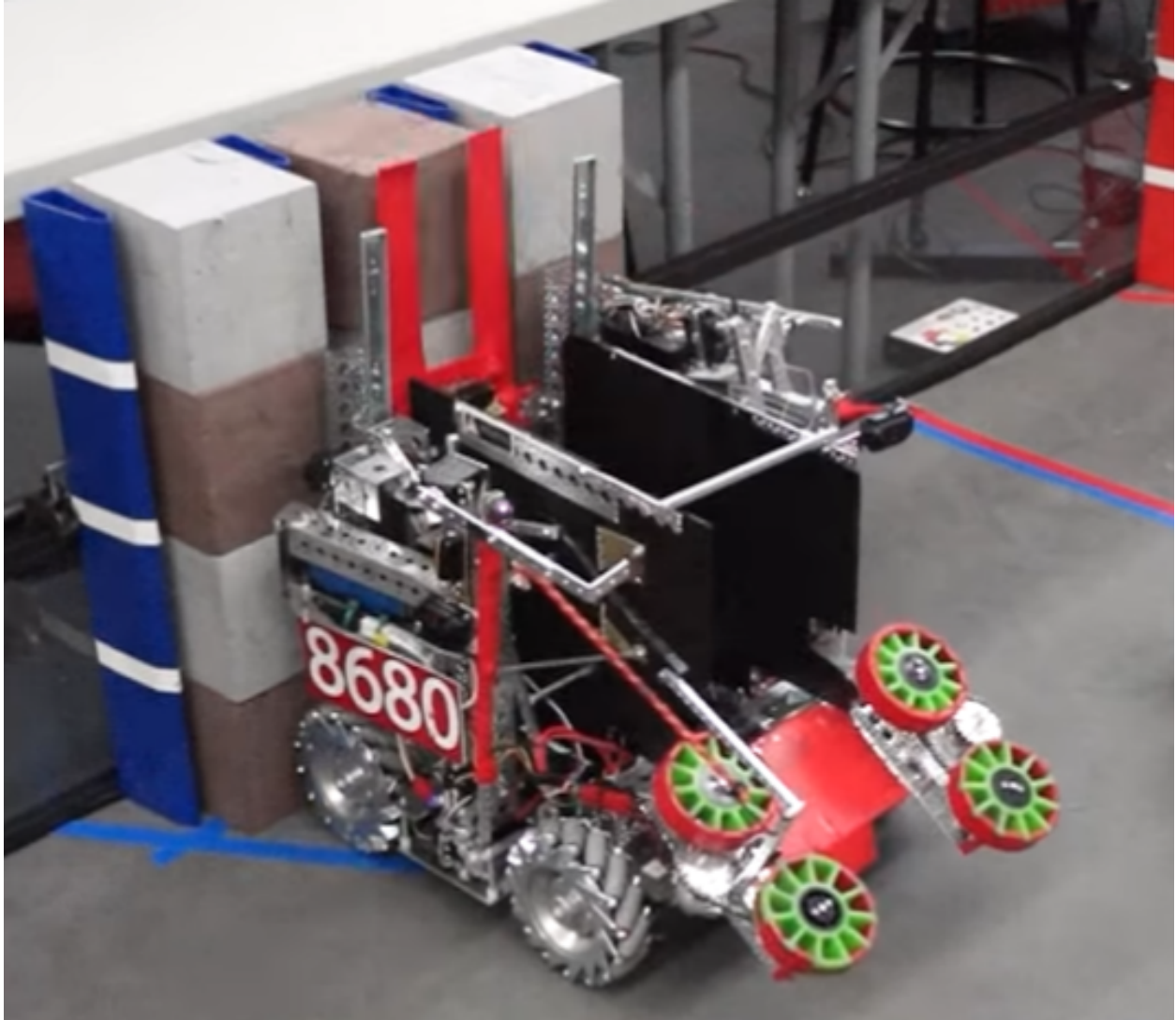


Fig. 86: 8680 Kraken-Pinion, Relic Recovery

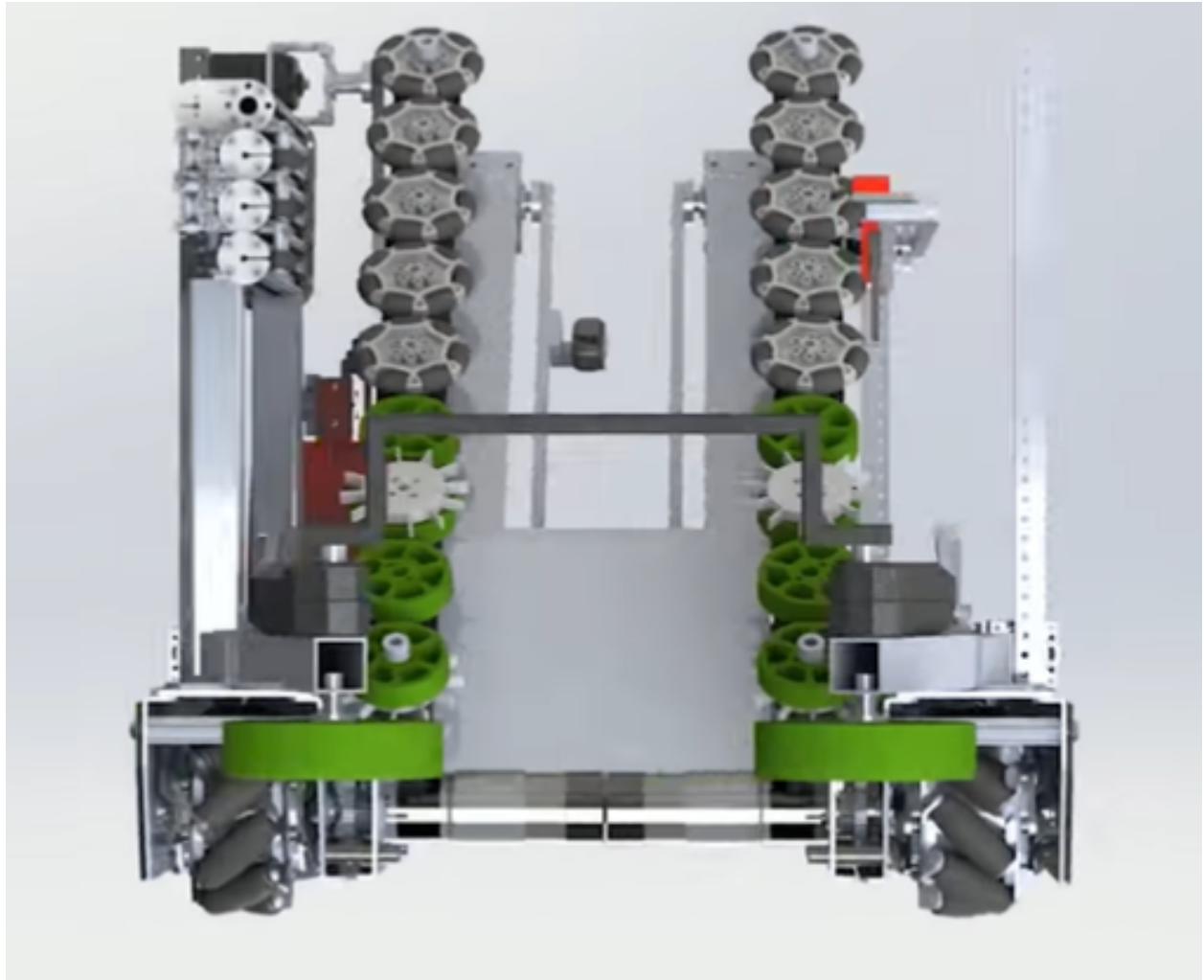


Fig. 87: 2856 Tesseract, Relic Recovery, 2 & 4 in. compliant wheels, 2 in. omni wheels



Fig. 88: 11115 Gluten Free, Finalist Alliance Captain (Detroit), Relic Recovery, springloaded

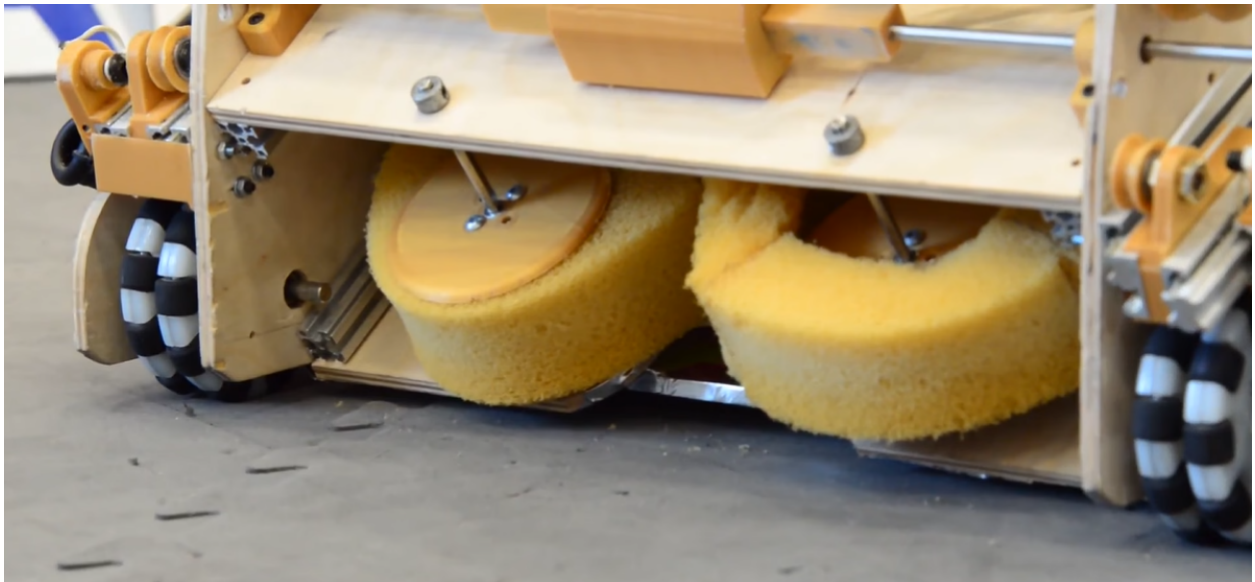


Fig. 89: 6299 ViperBots QuadX, Res-Q



Fig. 90: Ball intake for VRC Game Turning Point

Tubing Intakes

Tubing or noodle intakes typically use some sort of pliable tubing, which is rotated at high RPM to intake game pieces. Tubing intakes are particularly efficient at picking up small objects such as the balls and cubes from Res-Q, Velocity Vortex, and Rover Ruckus.

Surgical Tubing Intake

Surgical tubing or spearfishing rubber tubing, sold by many different manufacturers, is a great option for picking up small game elements such as the minerals from Res-Q or Rover Ruckus. Surgical tubing intakes can, and often, have multiple sets of tubing in order to move minerals from the collection point to the holding box. This was most often seen in games where robots had to transfer minerals from the field to an elevated location. Unlike wheeled and rubber band intakes, surgical tubing intakes are practically always fixed at a certain height and angle.

Surgical tubing by itself is soft and pliable. Teams have two options: 1) increase the RPM to 800-1000+ RPM, or use polyurethane tubing at a lower RPM (100-250 RPM). Polyurethane tubing can be purchased at a local hardware store and is a clear tubing that is quite stiff. Using some lubricant, insert the clear tubing into the surgical tubing for added stiffness. It is encouraged that teams test different RPMs and stiffness to develop the optimal intake.

Surgical tubing intakes are especially good at picking up multiple elements at a time, due to the high RPM (sometimes >1000 RPM) of the rollers. However, it suffers from a lack of controllability, as sometimes the driver may accidentally pick up more than needed, and have to spit it out.

Advantages

- Able to collect multiple elements at a time
- Generally more efficient than wheeled intakes
- Specializes in small elements

Disadvantages

- Requires high RPM
- Less controllable
- Not able to pick up large elements easily

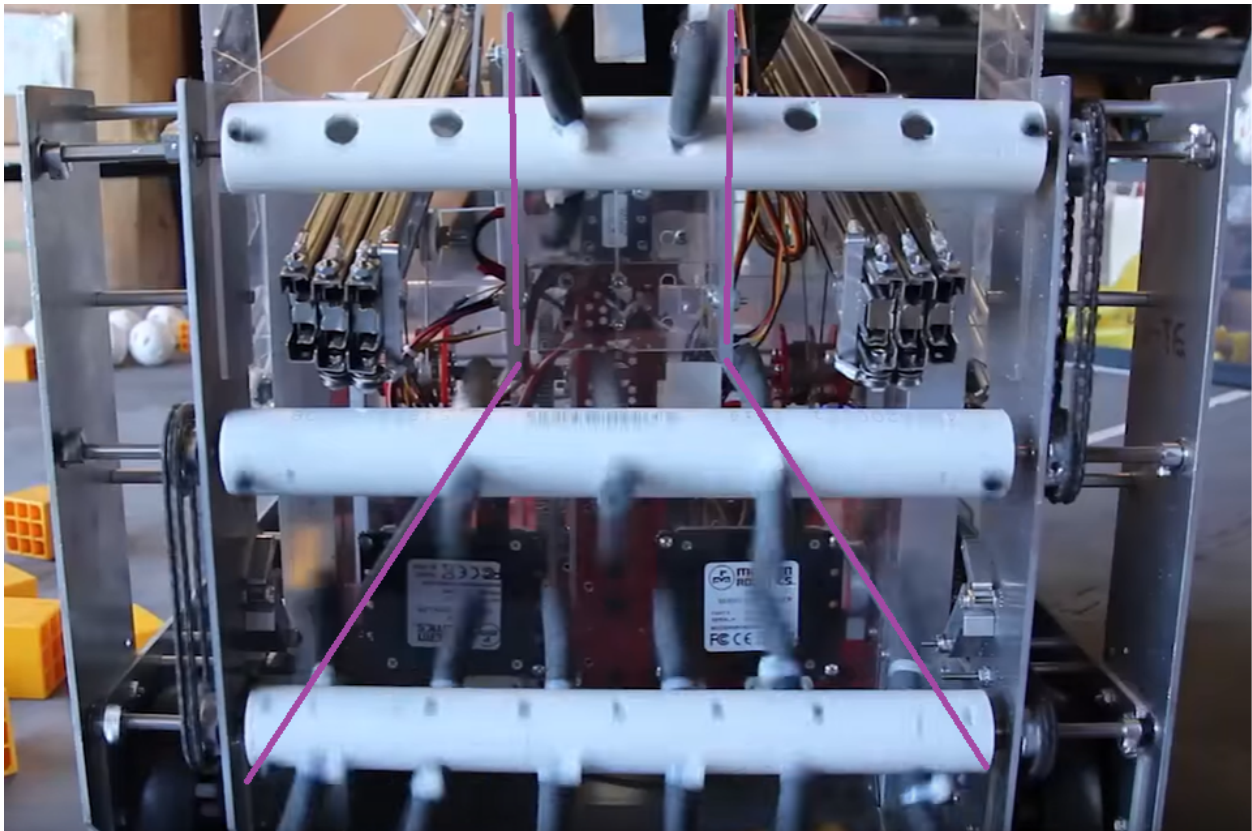


Fig. 91: 8375 Vulcan Robotics, Res-Q - great example of funneling

Zip Tie Intake

Instead of using surgical tubing, some teams opt for heavy zip ties instead. This can work, but we recommend surgical tubing as it is one of the most tried and tested methods for picking up nearly any game element.



Fig. 92: 8393 Giant Diencephalic BrainSTEM Robotics Team, Winning Alliance 1st Pick (St. Louis), Relic Recovery

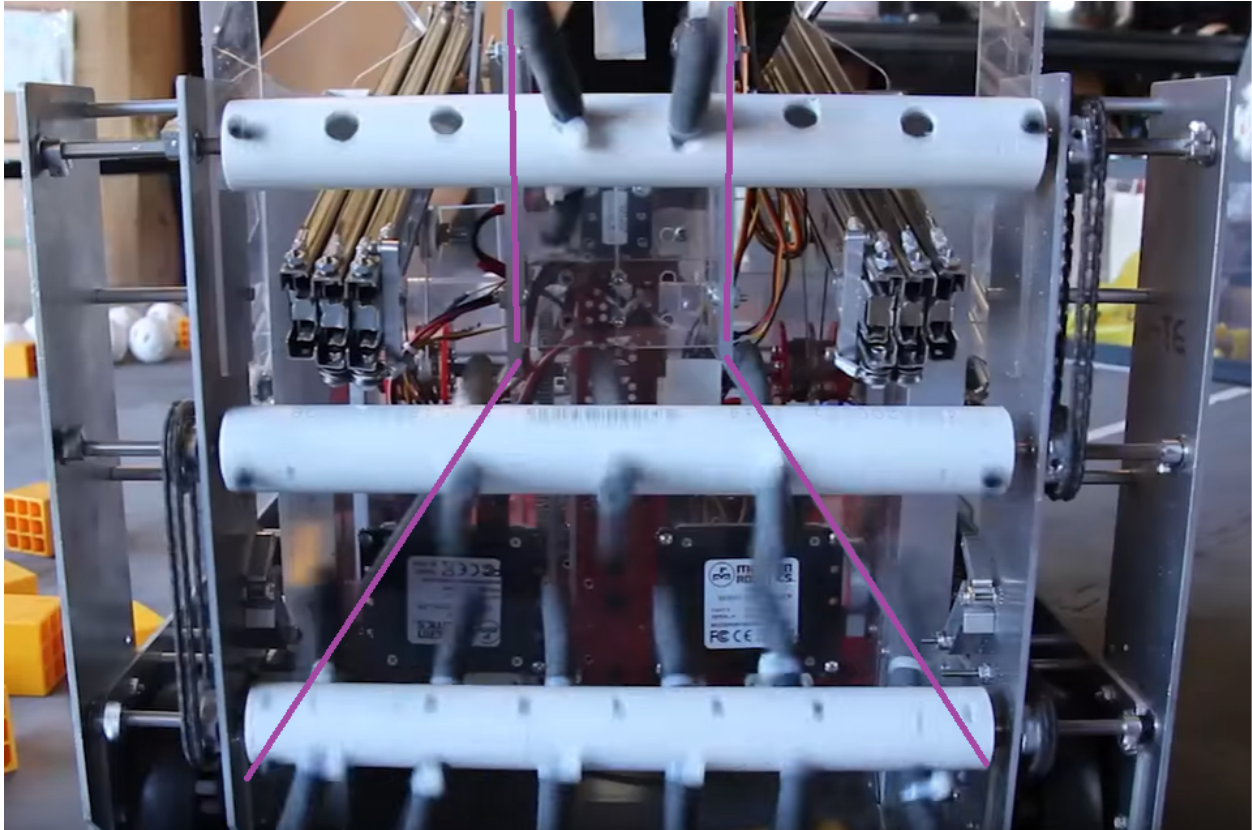


Fig. 93: 8375 Vulcan Robotics, Res-Q - great example of funneling

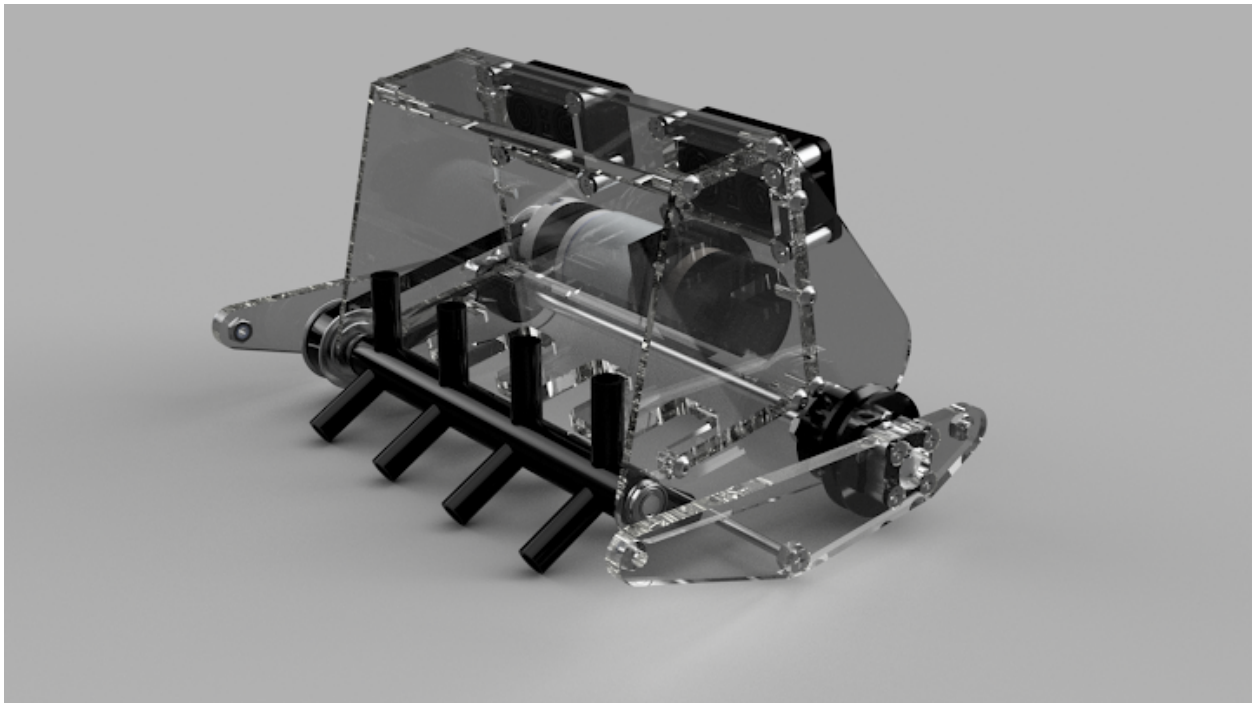


Fig. 94: 11115 Gluten Free, Winning Alliance 1st Pick (Detroit), Rover Ruckus

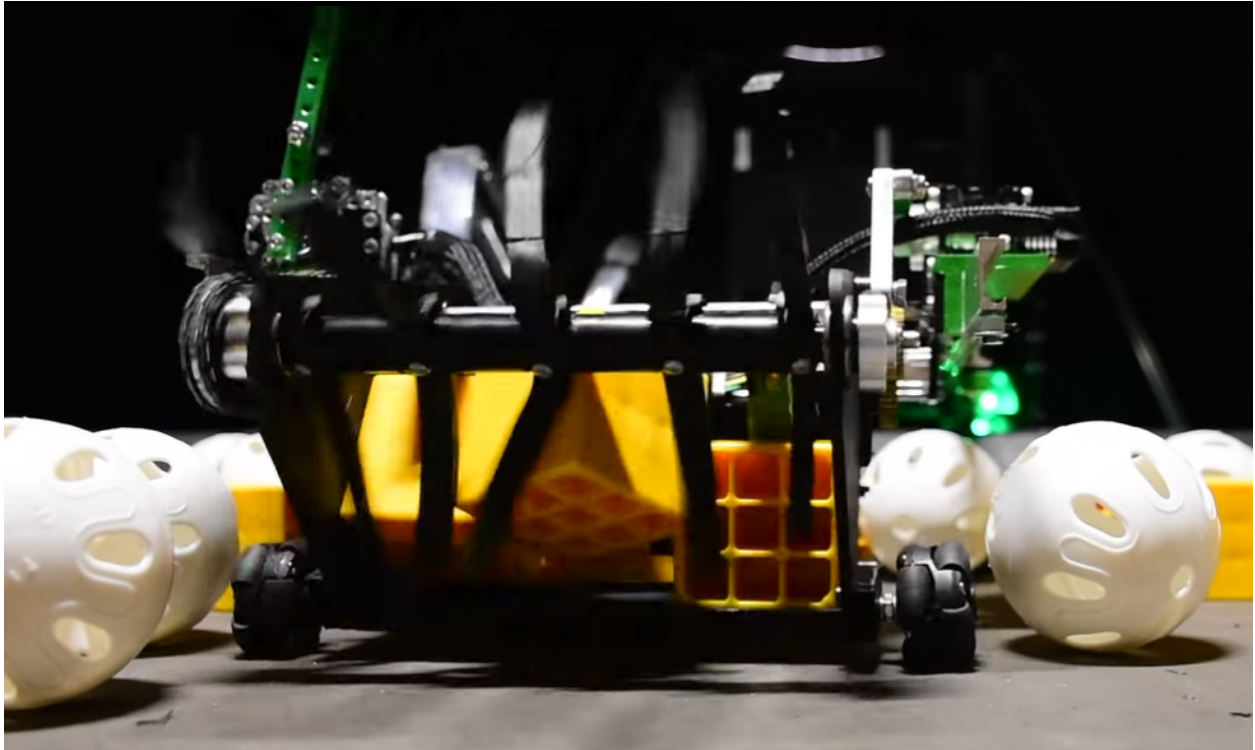


Fig. 95: 7203 KNO3, Rover Ruckus

3D printed intake (NinjaFlex/TPU Filament)

TPU/NinjaFlex 3D printer filament is a great low-RPM intake flap option, if designed right they work very well with VEX 393 motors (considered servos in FTC), and several teams have successfully used this configuration in competition.

We recommend 3D printed intakes only if your team has had experience in 3D printing parts.

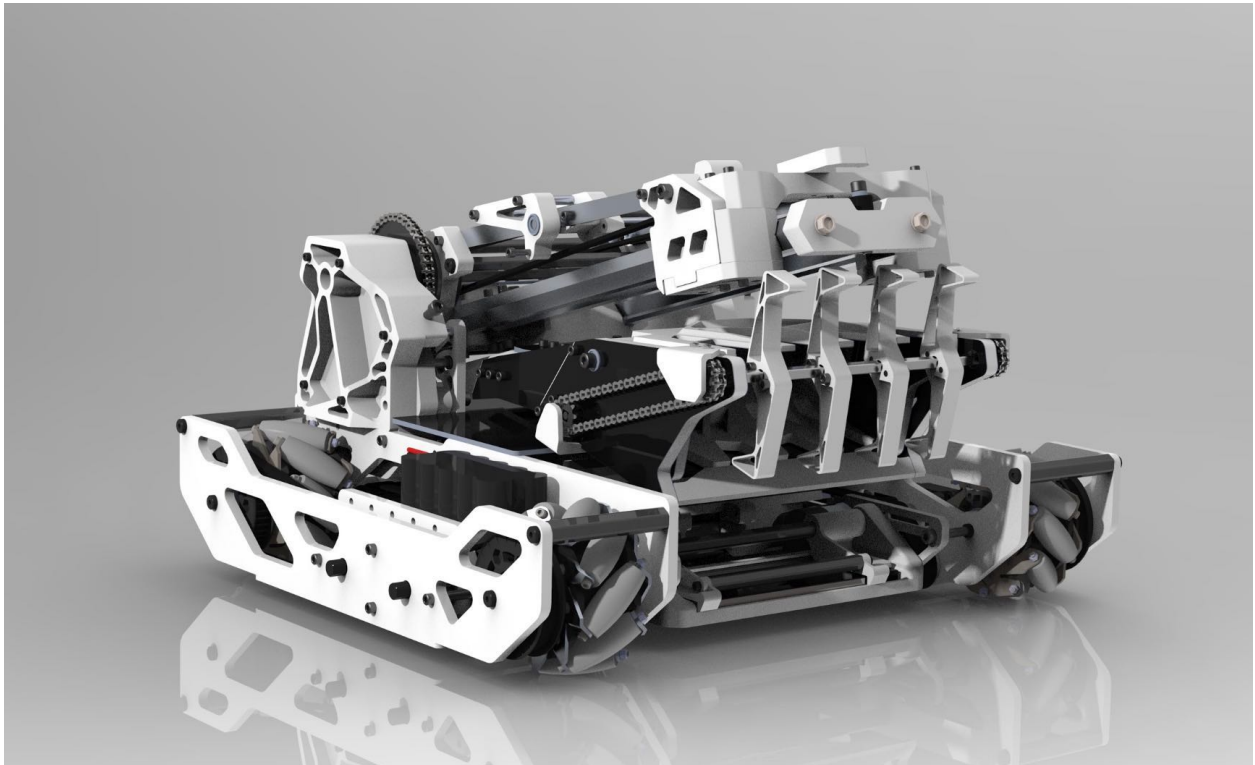


Fig. 96: 731 Wannabee Strange, Rover Ruckus

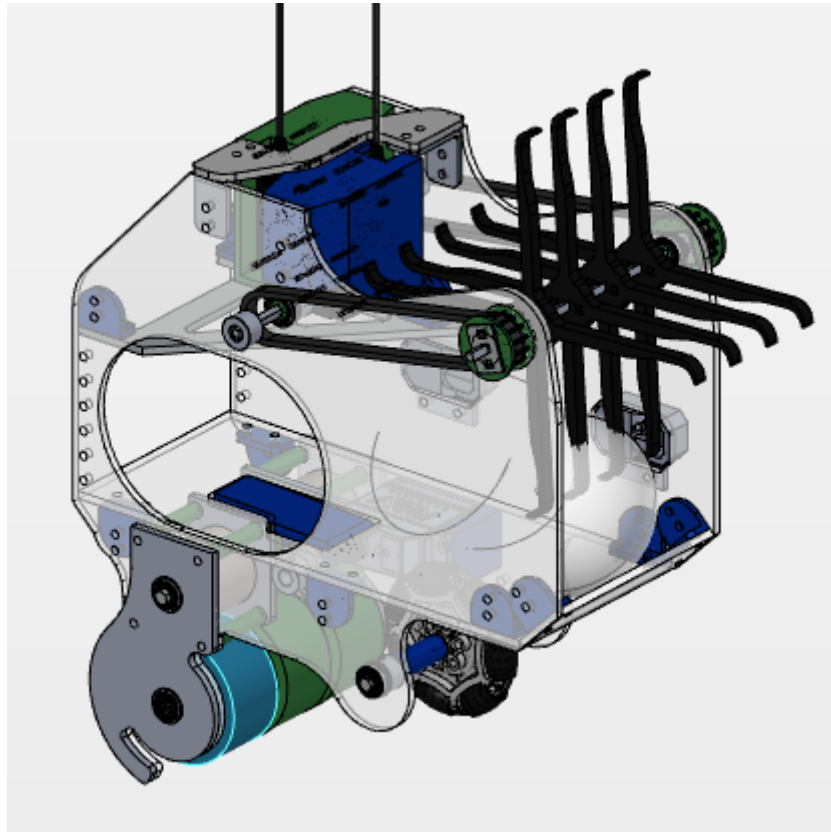


Fig. 97: 8417 Lectric Legends, Rover Ruckus - TPU intake flaps, powered by two VEX 393 motors

1.13 Motor Guide

1.13.1 Motor Hardware

Shafts

In FTC, shafts are the way to transfer power from the motor to the mechanism it is powering. Keep in mind that some shafts such as 8 mm round shaft cannot use set screws (although set screws are generally discouraged). Adapting from different shaft types can be very inelegant and inefficient, with the exception of 5 mm to/from ½" hex. The different (common) types of shafts in FTC are as follows.

- **Hex**
 - 5 mm hex
 - 3/8" hex (Thunderhex)
 - 1/2" hex (Ultrahex, Thunderhex)
- **D-shaft**
 - 4.7 mm D-shaft
 - 6 mm D-shaft
 - 1/4" D-shaft
- **Round**
 - 6 mm round shaft
 - 8 mm round shaft
- **Square**
 - 1/8" square shaft
 - 1/4" square shaft (Thundersquare)
- **REX (round and hex profile shaft)**
 - 8 mm diameter
 - 12 mm diameter

Connectors

There are different types of connections from various manufacturers, which is another aspect of motors to consider. Teams need to adapt to the JST system used by REV if they are using the REV Expansion Hub. Here are the different types of connectors for FTC motors.

- JST connector (standard REV)
- Anderson PowerPole
- Bullet connector

Gearboxes

Fundamentally, a gearbox is just a collection of gears and an enclosure that causes those gears to connect. Gearboxes have an output ratio, the final gear ratio between the motor input and the final output shaft. Gear ratios are further explained in the glossary.

In FTC, gearboxes may be more common than you think - every motor has a gearbox attached to it. These gearboxes are one of the following two types: spur or planetary. **Just for clarification, the gearboxes below are separate from the base motor.**

Spur Gearboxes

Spur gearboxes are an arrangement of gear ratios, often stacked to achieve a large compound ratio (e.g. 40:1). Each individual ratio only has two gears- one may be 8:1, another may be 5:1, but the final ratio will be 40:1. These gearboxes are used in the Andymark NeveRest Classic series and goBILDA's 5201 series motors, as well as some REV HD Hex Motors (gearbox types are denoted on REV's website very clearly). Due to the nature of how these gearboxes are built, each reduction only has a few teeth from each gear engaged, and those teeth carry the entire load of the gearbox. It's easy to damage a spur gearbox from shock load, and if anything happens to one of the gears, the gearbox will stop functioning. **We do not recommend using spur gearboxes on high-load applications such as drivetrains or arms. Instead, use planetary gearboxes.**

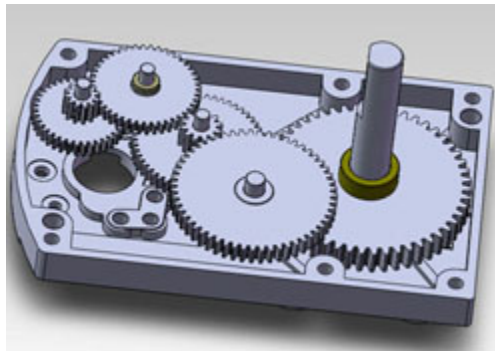


Fig. 98: Example of a spur gearbox. Note how all gears mesh with only one other gear.

Advantages of Spur Gearboxes

Cost. Generally, spur gearboxes are cheaper than planetary gearboxes. However, in FTC that price change is often minimal. A 20:1 planetary gearbox from Andymark is only \$2 more than a spur 40:1 (used 40:1 for cost comparison because there is a lack of overlap in gear ratios between planetary and spur gearboxes from vendors).

Spur gearboxes from Andymark NeveRest Classic motors, the REV HD Hex 40:1 Spur motor, and goBilda 5201 Series Yellow Jacket Spur Gear Motors. The gears in these gearboxes, while not interchangeable, are very comparable and practically indistinguishable in performance (when they are the same reduction). The main thing to consider here is your **desired reduction, your desired motor connections, and your desired output shaft type.**

- goBILDA's 5201 series spur gearboxes are much cheaper than either REV's or Andymark's; whether that's a good or bad thing is up to you. They utilize the rather uncommon (in the FTC world) bullet connection for power. The output shaft is a 6mm D-shaft.

- REV HD Hex Planetary 40:1 motor - This motor comes only in a 40:1 ratio, but does use the same connections (JST VH) as the REV Expansion and Control hub for power which means no adapter cables. The output shaft is a 5mm hex shaft. REV UltraHex has a 5mm hex bore running through the middle of a ½" hex shaft, which makes adapting this motor to any length of Ultrahex, and by extension, ½" hex shaft, very easy.
- Andymark NeveRest Classic motors come in a few different ratios, which are 40:1 and 60:1. The output shaft is a 6mm D-shaft, and like all NeveRest motors use the Anderson Power Pole to connect to power. This connector is perhaps the most robust of the ones listed here.

Planetary Gearboxes

Planetary gearboxes use a more complex system of gears to achieve a robust reduction in a compact space. In automotive engineering, planetary gear sets can achieve a few different ratios without changing gear size, but all planetary gearboxes that you will see in FTC only achieve one gear ratio. Planetary gearboxes are used in the Andymark Orbital series, some REV HD Hex Planetary and UltraPlanetary Motors (gearbox types are denoted on REV's website very clearly), and goBILDA's wide selection of planetary gear motors. Additionally, AndyMark sells a few aftermarket planetary gearboxes called NeveRest Sport and 57 Sport, and VEXpro sells the customizable VersaPlanetary gearbox. As you can see from the graphic below, there are more teeth meshing per stage than in the spur gearbox.

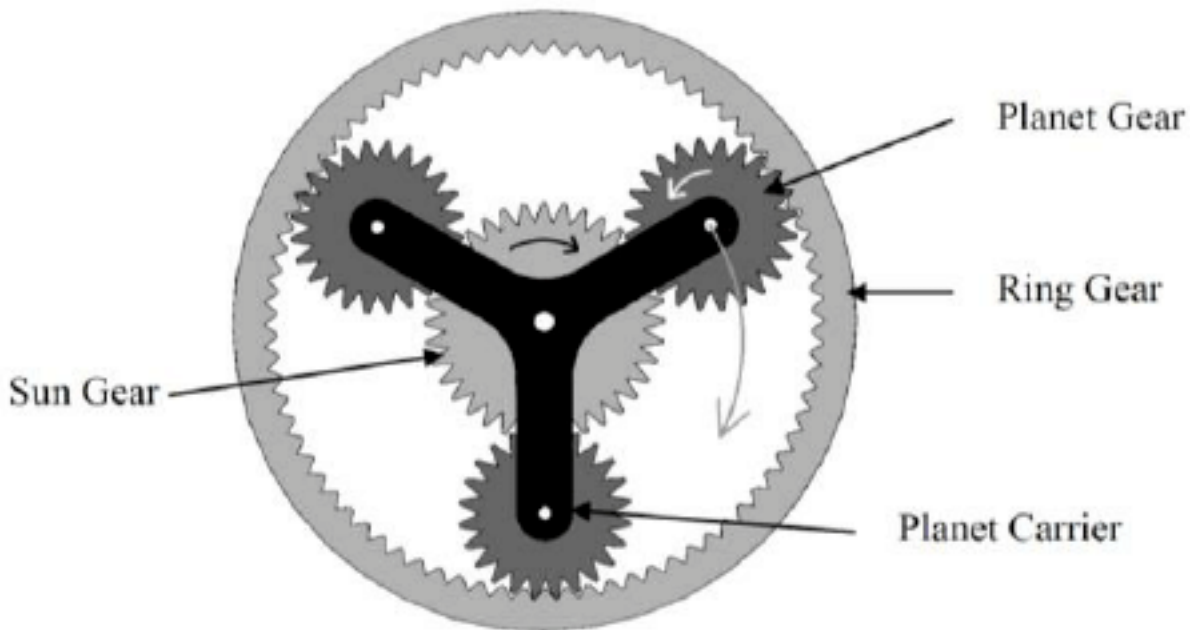


Fig. 99: Example of a planetary gearbox stage. Note how the sun gear meshes with >1 gear.

Advantages of Planetary Gearboxes

- Backlash is lower than spur gearbox equivalents. Backlash is defined as the clearance or lost motion caused by gaps between parts. This can easily be explained through putting a wheel or gear on a motor shaft and lightly rotating it. The part should be able to wiggle around a little without having considerable force imparted on it. This is caused because it is impossible for the gear teeth inside the gearbox to mesh perfectly, and is the same for chain and sprockets, or any other form of power transmission. However, planetary gearboxes have less backlash as they have less stages of gears.

- Efficiency is better than spur gearboxes. A typical two-stage spur gearbox is about 85% efficient, whereas most two stage planetary gearboxes are 94% efficient.
- Load capacity is higher for planetary gearboxes. This is due to having multiple teeth engaged per stage, which spreads the load. **This means planetary gearboxes will not break as easily when used in high-load applications such as drivetrains.**

Available Planetary Gearboxes and Motors

Standard planetary gearboxes include Andymark NeveRest Orbital motors, REV 20:1 Planetary motor, and goBILDA's 5202 Series Yellow Jacket Planetary Gear Motors.

Any of these "standard" gearboxes are more robust than the "regular" spur gearboxes. Like the spur gearboxes, the gears in these gearboxes, while not interchangeable, are very comparable in terms of robustness. Once again, the main thing to consider here is your **desired reduction, your desired motor connections, and your desired output shaft type.**

- goBILDA has, by far, the most varied selection of gearbox ratios with too many to list here, but utilize the rather uncommon (in the FTC world) bullet connection for power. The output shaft is a 6mm D-shaft.
- REV HD Hex Planetary motor - This motor comes only in a 20:1 ratio, but uses the same connections (JST-VH) as the REV Expansion and Control hub for power which means no adapter cables. The output shaft is a 5mm hex shaft. REV Ultrahex has a 5mm hex bore running through the middle of a 1/2" hex shaft, which makes adapting this motor to any length of Ultrahex, and by extension, 1/2" hex shaft very easy. The ratio of HD Hex is 20:1.
- Andymark NeveRest Orbital motors come in two ratios, which are 3.7:1 and 20:1. The output shaft is a 6mm D-shaft, and like all NeveRest motors use the Anderson Power Pole to connect to power. This connector is perhaps the most robust of the ones listed here.

High-end planetary gearboxes include Andymark NeveRest Sport and VersaPlanetary gearboxes, which can be quite expensive. However, the REV UltraPlanetary gearbox is a very affordable kit for new teams.

- REV UltraPlanetary gearbox - The UltraPlanetary is a customizable planetary gearbox that is designed for FTC. The three gearbox options are 3:1, 4:1, and 5:1, and can be mix & matched to create a custom ratio. It is possible to use from one to three gearboxes for a minimum ratio of 3:1 and maximum. The UltraPlanetary was intended to give teams the maximum customization without the typical limiting factor - high cost. The total cost for the three stage gearbox and motor is \$36 - a very affordable price, given that most spur gearbox motors are in the ~\$30 range. In addition, the UltraPlanetary has a female 5mm hex output shaft which allows teams to customize the shaft length. The UltraPlanetary is a great option for teams as it allows for customizability and very good value.
- Andymark NeveRest Sport gearboxes - These gearboxes were designed to be more robust than the gearboxes found on Andymark NeveRest motors. The NeveRest sport gearboxes can handle larger shock loads, and can provide a larger reduction with options up to 256:1. These gearboxes are relatively expensive, but still not the most expensive gearbox you can use in FTC, with prices ranging from \$46.00 for the 4:1 reduction to \$77.50 for the 256:1 reduction. There are no motors included with the gearbox, you will need a bare NeveRest motor, REV HD Hex motor, or goBilda motor to attach to the gearbox. The selection should be based only on which connection type you like the most (powerpole vs. JST VH vs. 3.5mm bullet) for power.
- VersaPlanetary gearboxes - These planetary gearboxes were designed for FRC use and can be very expensive (upwards of \$80 for a 40:1 ratio) but because they were designed for such heavy use the VersaPlanetary is arguably the most durable gearbox you can use in FTC. If you somehow manage you break one of the components of the gearbox, they are very easy to repair. Unlike other gearboxes, almost every part of the gearbox is replaceable and able to be bought by itself. Every other gearbox on this list would require you to buy a whole new gearbox if one part of it breaks. In addition to that, you

can disassemble the gearbox from the motor-side of the gearbox, meaning in the event that something breaks, you don't have to remove the gearbox from the plate it's attached to to be able to switch that part out. Another nice thing is that attaching the pinion to a bare motor is super easy since it clamps onto the motor shaft instead of being a press fit like the NeveRest sport pinion gear is. This means you don't need any special tools like a press and can switch motors out very quickly, even during a competition. VersaPlanetarys also allow you to choose your own ratio with swappable 3:1, 4:1, 5:1, 7:1, 9:1, and 10:1 stages. When using VersaPlanetary gearboxes there are a few important factors for you to consider.

- The higher the reduction, the weaker the stage is. So for instance, a gearbox with 4:1, 4:1, 5:1 stages would be stronger than a 10:1, 10:1 gearbox though they end up with the same output ratio of 100:1.
- Always place the highest reduction as close as you can to the motor. There is less load placed on it here, and is less likely to break than if you placed it higher in the reduction. Then after your weakest stage, you should use your next weakest stage, until you reach the output stage.
- Vex Pro also sells a VersaPlanetary Lite option, which replaces the outer aluminum ring gear and housing with a 30% glass filled nylon. VersaPlanetary Lite gearboxes use the same gear sets as the regular VersaPlanetary, so if you ever feel the need for a stronger gearbox you can mix and match parts to fill your requirement. While still expensive (compared to other planetary gearboxes from Andymark, REV, and goBilda) at about \$50 for one gearbox, it's still one of the strongest gearboxes available for FTC use. It should be noted that the lite version comes with a 3/8" hex output shaft only, whereas the more expensive VersaPlanetary comes with various options including 1/2" hex, 3/8" hex, 1/2" round with 1/8" keyway, a CIM Output option, and a Universal Female Output Shaft. However, the shaft can be swapped out for a different one. The VersaPlanetary lite also loses side mounting holes on the gearbox.
- Like the NeveRest Sport gearboxes, you will need to supply your own bare motor to attach to the gearbox.

Choosing the right gearbox

For regular use, any of the "standard" (see above) type of planetary gearboxes will fit your needs. Planetary gearboxes are just a tiny bit more expensive, but boast better backlash and efficiency, higher load capacity, and better capacity for shock loads than spur gearboxes. The tradeoffs, cost and mechanical noise, are almost never a factor. **Because both gearbox types are so similar in price for similar ratios, it's impossible to recommend the use of a spur gearbox over a planetary gearbox for most use cases in FTC.** If you already own spur gearboxes, try to use them in lower-load situations and try to put planetary motors at least on your drivetrain.

For larger reductions, the 57 Sport and VersaPlanetary gearboxes are a viable but expensive option. In some cases, it may be more economical to choose a motor you already own and build an external reduction using gears, chain, or belts. It should again come down to your desired output shaft, desired gear ratio, and for the VersaPlanetary, whether you want the ability to swap parts out on the fly. You also may be able to buy 57 Sport gearboxes off of local FRC teams at a solid discount - some may be willing to sell their older gearboxes.

The motor itself, meaning the part that actually has power running to it and makes the shaft spin, not the gearbox that is often mounted on the motor, doesn't matter all that much in FTC. The motors from Andymark, REV, and goBILDA might have different specifications on their websites, but they're all very similar. The VEX motor testing (link in the *jvn-calculator* section) shows the accurate specifications of a bare RS-555 series motor. When choosing between companies, always go with what works best with your motion system, and which is easiest to work with your existing electronics system.

1.13.2 JVN Calculator

With the JVN calculator for FTC, it is possible to calculate the gear ratio that is necessary for an arm. The JVN Design Calculator is a very valuable resource for picking appropriate gear ratios for almost application you need. It was developed by [John V-Neun](https://johnvneun.com)²⁴, coach of FRC 148, and was initially designed for FRC calculations.

It is fairly intuitive, but can save a lot of time doing calculations on your own, especially when developing designs like arms, linear slides, and drivetrains. This is as the balance between power/torque and speed is critical to the end design. For example, arms require much more torque than speed, while linear slides need very little torque to maximize speed.

To use the calculator for rotary mechanisms (arms), select the Rotary Arm sheet. A good practice is to “pencil in” VEX’s dynamometer testing for the bare NeveRest motor.

The VEX dyno test is a trusted and accurate resource for teams, even though its results are significantly higher than the specifications posted on vendor pages.

The test results may be found by searching VEX Dyno Testing, or by clicking [here](https://motors.vex.com/other-motors/am-NeveRest)²⁵. (Note: this testing is done to the bare motor, without the end gearbox.) Then, change the gear ratio in the sheet itself. The bottom left of the sheet has a gear reduction page. Simply put the overall gear ratio of your gearbox in the right side, keeping the one on the left. For example for a NeveRest 20:1, write in “1” in the driving gear column and “20” in the “driven gear” column. Then, add the additional gearing. For example, if the total ratio is 40:1, there should be a 1:2 ratio in the driving:driven gear ratio below the 1:20. It doesn’t matter whether it is 10:20, or 42:84 - only the proportion matters. This gives you a lot of flexibility when looking for the right ratio. Additional gear reductions can be added if needed.

The same goes for linear mechanisms, with one additional note. The calculator will ask for a pulley diameter, so provide the diameter of the spool that the string is wound on. For a horizontal linear mechanism, neglect stall load. For vertical linear mechanisms, only pay attention to stall load if the mechanism will hang the robot. It is recommended that the load is 60+ pounds for a 40 pound robot for safety. A good starting point is around 6 in/sec, but most teams will increase the speed over time.

For 1-speed drivetrains, just enter in the correct motor specifications and add in the total weight and wheel diameter. Don’t worry about the efficiency and speed loss, just keep it where it is. The adjusted speed will account for sag into field tiles, loss of efficiency through belts, gears, etc., and loss of grip as the wheels eventually wear out. A good starting point should be around 1.5-2 ft/s, though many teams have faster drivetrains.

General Tips

- Always use the VEX dyno test results for free speed, stall torque, stall current, and free current.
- Ensure that the total current draw does not exceed the stall current! Generally, this should not be a problem, but it is something to be aware of.
- Stall load refers to how heavy a weight the mechanism can lift at the end before it stalls. **It is a good principle for the stall load to be 1.5x-2x, or even more, of the actual load. This is called a Factor of Safety.**
- For linear slides, maximize linear speed by using a low gear ratio. For arms, maximize torque by using a high gear ratio.
- You need to keep in mind the loaded current draw on your motor. Your motor has a stall current of ~10 amps, so you don’t want to exceed that. You also have a 20amp fuse on your battery, so make sure that you don’t exceed that for too long, or else you’ll pop the fuse.

²⁴ <https://johnvneun.com>

²⁵ <https://motors.vex.com/other-motors/am-NeveRest>

- With each stage of gears, chains or belts, you lose efficiency. Depending on how well you build your system and what you are using, the percent loss changes. However, generally it is around 5% per stage. Meaning that if you had three stages of gears, your efficiency should be 0.95^3 , or approximately 0.857 or 85.7%.

[Download Link²⁶](#) (If the link is broken, search JVN for FTC)

1.14 Servo Guide

First, a disclaimer. **Servos are NOT replacements for DC motors, and should not be used as such. Servos are made for fine-tuned and accurate movement, not high-load or quick rotation applications.**

Servos are an interesting beast in FTC. They are their own animal and vary widely in price, performance and value. Picking the right servo for your application is a question that's almost impossible to give a blanket answer for. There are two subcategories of servo specifications that you should use to decide what servo you want to use: **output power** (measured in speed and torque) and **durability** (normally best represented by the gear material inside the servo).

Let's start with durability, as this is probably the easiest to summarize. The two things that threaten a servo's longevity are the motor inside burning out and more commonly, the gears stripping inside the servo. A motor burning out is pretty uncommon, but it can happen under large loads for a prolonged amount of time. This means **you should never stall a servo against an immovable object**. When selecting a servo, pay attention to the material the gears are made from. This ranges from plastic to titanium, so let's go down the list, starting from the weakest.

- **Plastic:** with low power servos, these are normally okay. Generally used for applications in model airplanes such as ailerons.
- **Karbonite:** Hitec's gear plastic is a very durable and long lasting plastic and is very good under long use and low load. Be aware that it can strip easily under the shock loads found in FTC.
- **Brass:** Brass gears are stronger than plastics but also suffer greatly when faced with shock loads in FTC, like intake wrists and deposit buckets. It's found on slightly higher end servos in the \$25-30 range, such as the REV smart servo.
- **Steel:** This is where we start getting big. Steel gears are very durable and you'll have a tough time stripping these. Expect to pay \$40+.
- **Titanium:** Titanium is where you get into really high end, virtually unbreakable servos. Starting from \$75, they can reach over \$150.

The next variable is output power, measured in both **speed** and **torque**. Speed (normally in seconds per 60°), refers to how fast the servo turns 60 degrees in Standard Rotation mode. Torque (usually measured in oz-in), refers to the amount of force the servo can apply to a lever. For reference, if you put a 1" bar on a servo, then put a force gauge on the end, the torque rating of the servo (in oz-in) will be measured. As you may know, speed and torque have an inverse relationship. Generally you can find some insanely powerful servos that are pretty slow (slower than 0.20 s/60°) or some less powerful ones with faster ratios (anything faster than 0.12 s/60° is considered very fast). Finding the right servo for your application can be tough, but a good way is trying to decide if you need more speed or torque, and if your servo will experience shock loads or not.

The last proposition is cost. Servos range from cheap \$7 servos for light applications, all the way up to some Hitec or Savox servos for close to \$200. By far the best bang for your buck servos out there are the Feetech dual mode servos, which is a programmable type of servo. This includes both the **REV SRS** (Smart Robot Servo) and **both goBILDA Dual Mode servos**. The biggest downside to the REV SRS and the old goBILDA

²⁶ <https://www.chiefdelphi.com/uploads/default/original/3X/1/6/16e019399060799a45f54f4d75a8aa5fee1f394f.xlsx>

servos are their brass gears. Coupled with high output power, this meant that stripping gears with any shock load was commonplace. The new goBILDA Dual Mode servos have steel gears now, but are new and aren't as competition tested as other servos. The next big name in FTC Servos is Hitec, who are a huge name in hobby servos for decades and are very well trusted. Their low end servos are inexpensive but easily broken.

A mid-priced Hitec servo is the HS 485-HB servo, with Karbonite gears. While it shouldn't be used in high load applications, it is fine for general use such as claws or trapdoors. Where Hitec really shines is the high end market. If your budget is over \$100, you can get into some very powerful Hitec servos. Most have titanium gears and are programmable, so you can dial in the performance and range to exactly what you need. The last big player in the servo market in FTC is Savox, which produces great mid-high range servos (think \$60-\$100+). They are made with titanium gears (close to bulletproof) and are **fast**. Savox servos are mostly brushless and coreless, so they do tend to scream a little under load, but they're definitely worth it if your budget allows for it.

Another question that often comes up is "what is the extended and continuous rotation options on the Servocity website/other sites?" You don't have to worry about those - as FTC ruled those modifications were illegal, just save a buck and buy stock servos. The only legal continuous rotation servos are servos that are continuous rotation from the factory. This includes the REV SRS servo and both goBILDA Dual Mode servos, programmed to continuous rotation mode. In other words, if you have a seller like ServoCity modify a servo for continuous rotation, that servo is not legal.

1.14.1 Best Value

- **Low Priced (~\$18)**
 - HiTec 485HB
 - Futaba servos
- **Medium Priced (~\$30)**
 - goBILDA Dual Mode Servo
 - REV Smart Servo
- **Best Performance (\$75+)**
 - Savox titanium servos
 - Hitec titanium servos

1.15 Electronics

1.15.1 Introduction

The FTC control system is based on using Android phones as a "Robot Controller" and a "Driver Station." The two major control systems of FTC are Modern Robotics and REV; however, nearly all teams are shifting over to REV because it is cheaper, simpler, and more reliable. More information about the FTC Control system can be found below.

Official control system Wiki on GitHub²⁷

REV Expansion Hub Documentation²⁸

²⁷ <https://github.com/first-tech-challenge/skystone/wiki>

²⁸ <http://www.revrobotics.com/content/docs/REV-31-1153-GS.pdf>

FIRST troubleshooting guide²⁹

Tips and Tricks

In addition to what is written in the official resources, there are a couple of additional tips.

The traditional XT30 connector that is used to REV is prone to breaking. It is highly recommended that teams replace XT30 connectors with Anderson Powerpole, or put adapters on their current wires. An example of an adapter is [this](#)³⁰. In lieu of this, teams can also 3D print strain relief connectors on the Expansion Hub to prevent XT30 disconnects. The file can be found on Thingiverse or through this [link](#)³¹.



Fig. 100: An example XT30 to Anderson Powerpole adapter

The Tamiya connectors found on many of the FTC legal batteries are very weak and prone to becoming unreliable after 10-20 repeated plug/unplug cycles. It is recommended that teams crimp new Anderson Powerpole connectors onto the battery.

It is highly recommended for teams to use the [REV grounding strap](#)³² and the [REV USB strain relief](#)³³ to help prevent disconnections.

To protect wires, teams often use [wire loom](#)³⁴, and to help wires extend far, teams often use cable carrier (also known as cable chain).

Wiring

Wiring is extremely important in FTC, as wires are crucial to the operation of the robot. It is highly discouraged for teams to overlook wiring, but many new teams seems to disregard it or throw it to the very last moment. While tedious and sometimes no fun, wiring can mean the difference between a win and loss.

²⁹ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/control-system-troubleshooting-guide.pdf

³⁰ <https://www.servocity.com/anderson-powerpole-to-female-xt30-adaptor>

³¹ <https://www.thingiverse.com/thing:2887045>

³² <http://www.revrobotics.com/rev-31-1269/>

³³ <http://www.revrobotics.com/rev-41-1214/>

³⁴ https://www.amazon.com/Black-Split-Tubing-Cover-Marine/dp/B00J7RD6ZI/ref=sr_1_13?keywords=wire+loom&qid=1562452458&s=gateway&sr=8-13

Connectors and Wires

There are many types of connectors for use in FTC. Here are the most common connectors you will find on an FTC robot.

For module power:



Fig. 101: Powerpoles



Fig. 102: XT30 connectors

For motor power:



Fig. 103: Powerpoles

For sensor wires:

For servo wires:

The connectors you **SHOULD NOT** use:

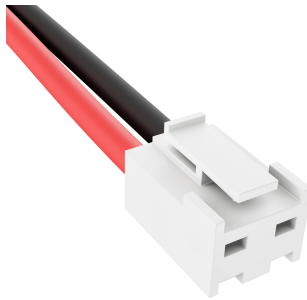


Fig. 104: JST-VH connectors

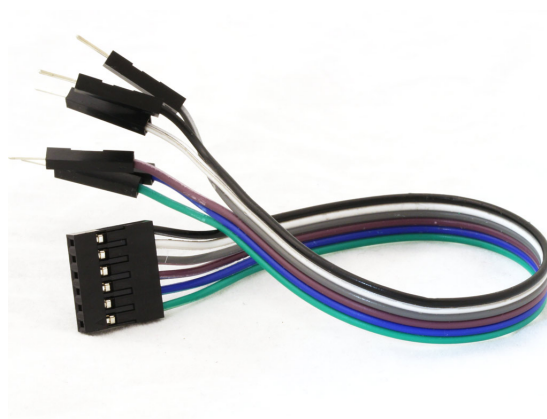
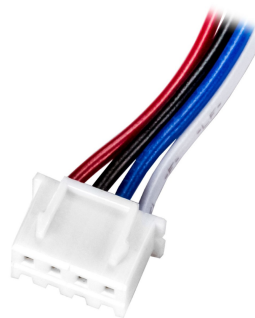


Fig. 105: Dupont 0.1"



JST PH Series



JST XH Series

Fig. 106: JST-PH or JST-XH connectors

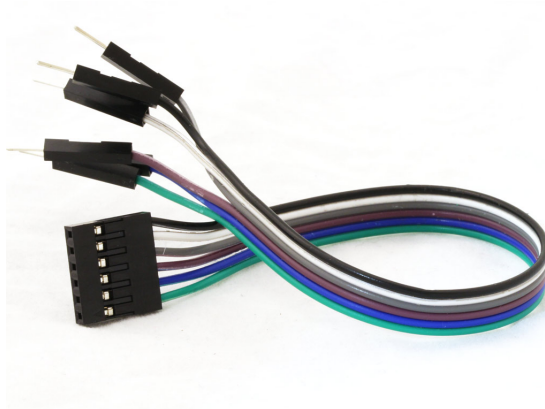


Fig. 107: Dupont 0.1"



Fig. 108: Tamiya Connectors (the metal connectors are fragile and will lead to random disconnects)

FTC Wiring Guide

FIRST has created a wiring guide to help teams with tasks like crimping cables, soldering connections, and ESD mitigation that won't be covered in this guide. Once you read up on electronics and wiring here, look at the FTC Wiring Guide for the best practices and more tips & tricks. In addition, FIRST has written an ESD mitigation whitepaper that is worth taking a look at.

[Robot Wiring Guide](#)³⁵

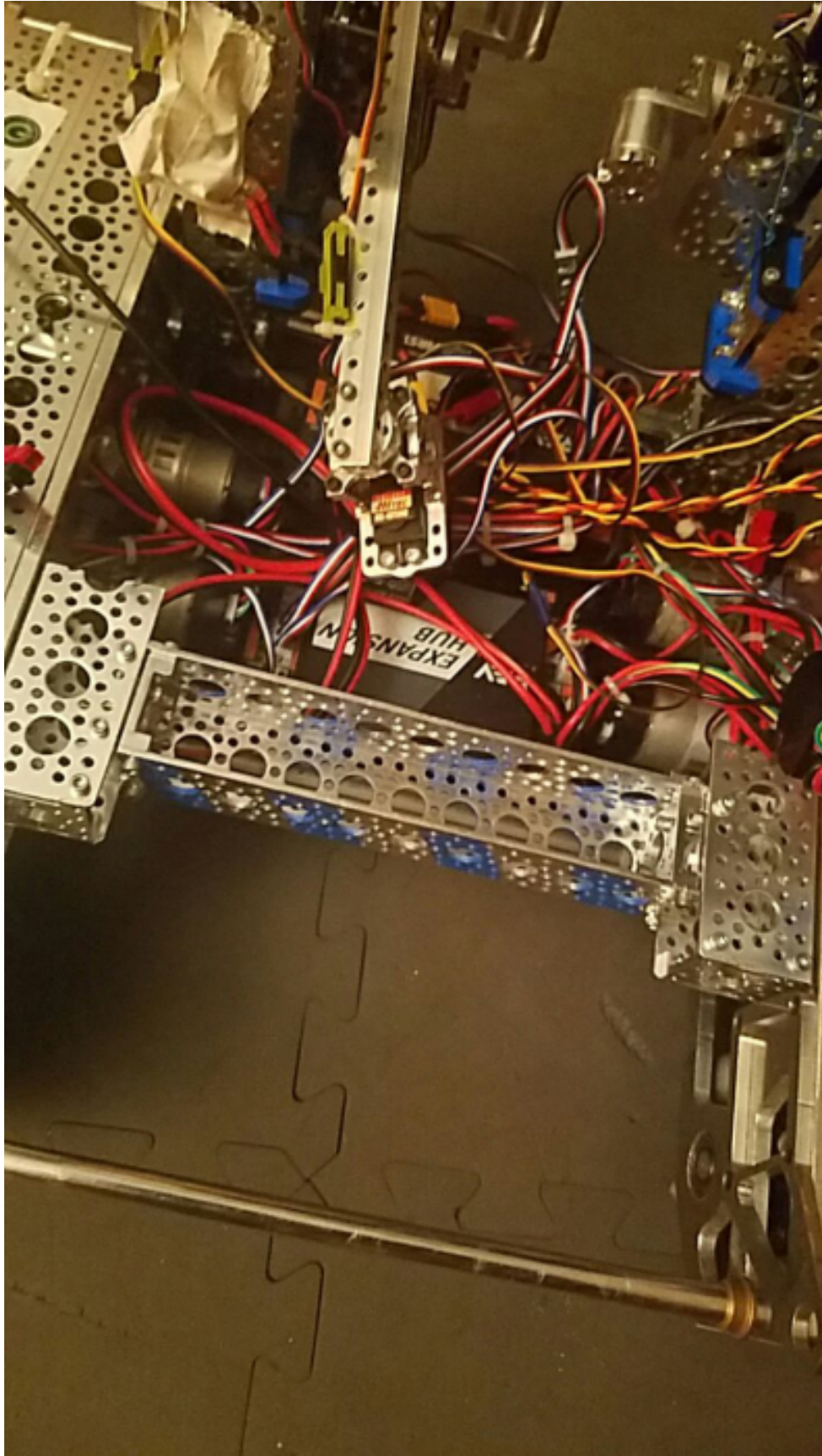
[ESD Mitigation Whitepaper](#)³⁶

General Advice

- **Always label wires! When bunched up, you may not know which wire goes into which port.**
- **Treat every wire connection as a point of failure.** Therefore, use electrical tape to tape up and insulate connections and utilize strain relief as much as possible.
- **Strain relief** should be used everywhere possible. It is highly recommended for teams to use products like the REV USB Retention Mount, as well as 3D printing strain relief methods for devices such as the Expansion Hub and robot controller phones.
- **DO NOT solder a wire before crimping it.** Solder can “creep” and losing connection is possible, possibly leading to fire.
- Keep all wire runs as short as possible to prevent entanglement and improve wire management.
- When using data/sensor cables, keep them away from motors. This will reduce electromagnetic interference (EMI). Add a ferrite bead if possible.
- Crimped connectors are generally better than soldered connectors, as solder joints can break easier than a crimped connection.
- Keep wires tucked away from moving mechanisms, and ensure that you will not be at risk of a mechanism snagging a wire. This is a proper application of materials such as acrylic, which allow drivers to see inside the robot while keeping wires out of the way of other robots/game pieces. It is advised for teams to purchase removable velcro ties or cable ties to aid with cable management.
- For power wires, lower gauge (larger size) wires are preferable. This means a lower resistance across the wire and higher power throughput. This is negligible for data wires, however.
- Small wires and cables are fragile. Treat them as such, and don't put them in an area where they will constantly be hit by another object. Larger power cables can take much more abuse.
- Ensure that your wires are kept out of pinch points where another mechanism could sandwich the wire. This is especially important in arms or mechanisms that are hinged.

³⁵ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/robot-wiring-guide.pdf

³⁶ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/analysis-esd-mitigation-echin.pdf



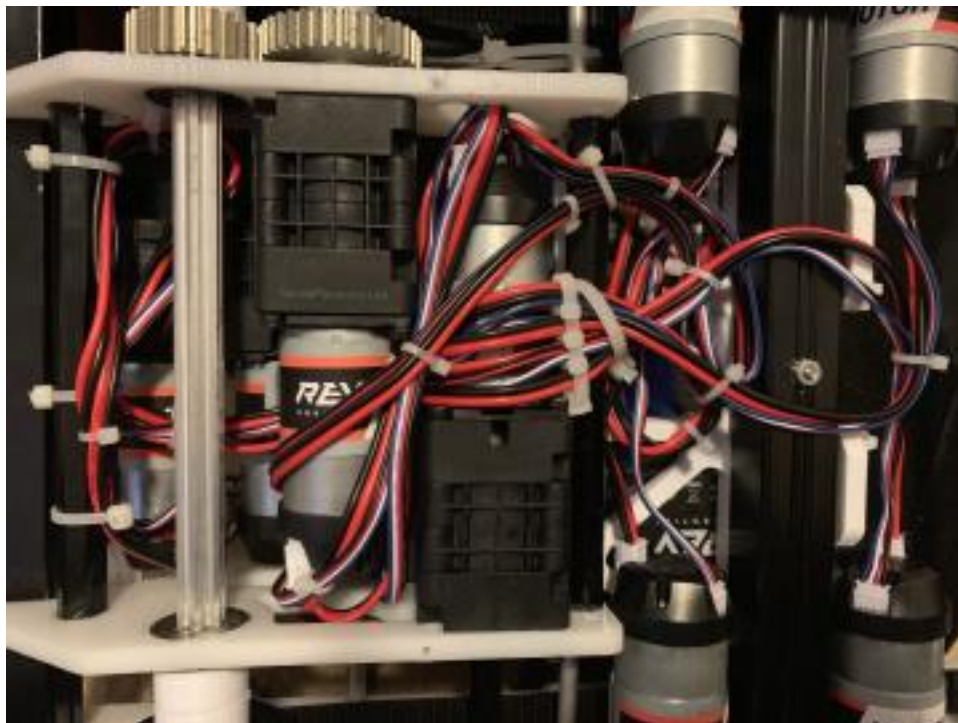
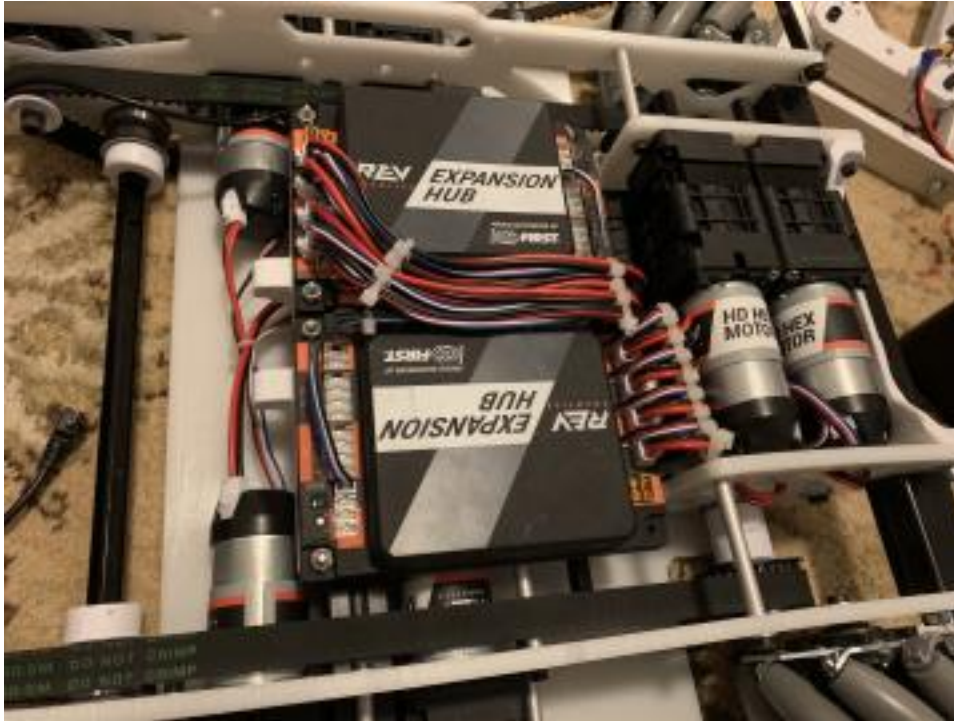


Fig. 110: Good example of wiring by 731 Wannabee Strange, Rover Ruckus

When wiring, also take the time to plan out a specific layout for your wires and how they will run throughout the robot. Take the time to lay out something like an electronics panel if necessary! When building the robot, devote adequate space to wiring. This could mean mounting a PVC pipe and running wires from the back end of the robot through it, or simply using velcro or zipties. **It is also recommended for electronics to be mounted on a nonconductive material such as wood to prevent ESD.**



Fig. 111: Good example of wiring by 8417 'Lectric Legends



Fig. 112: Good example of wiring by 7244 Out of the Box Robotics

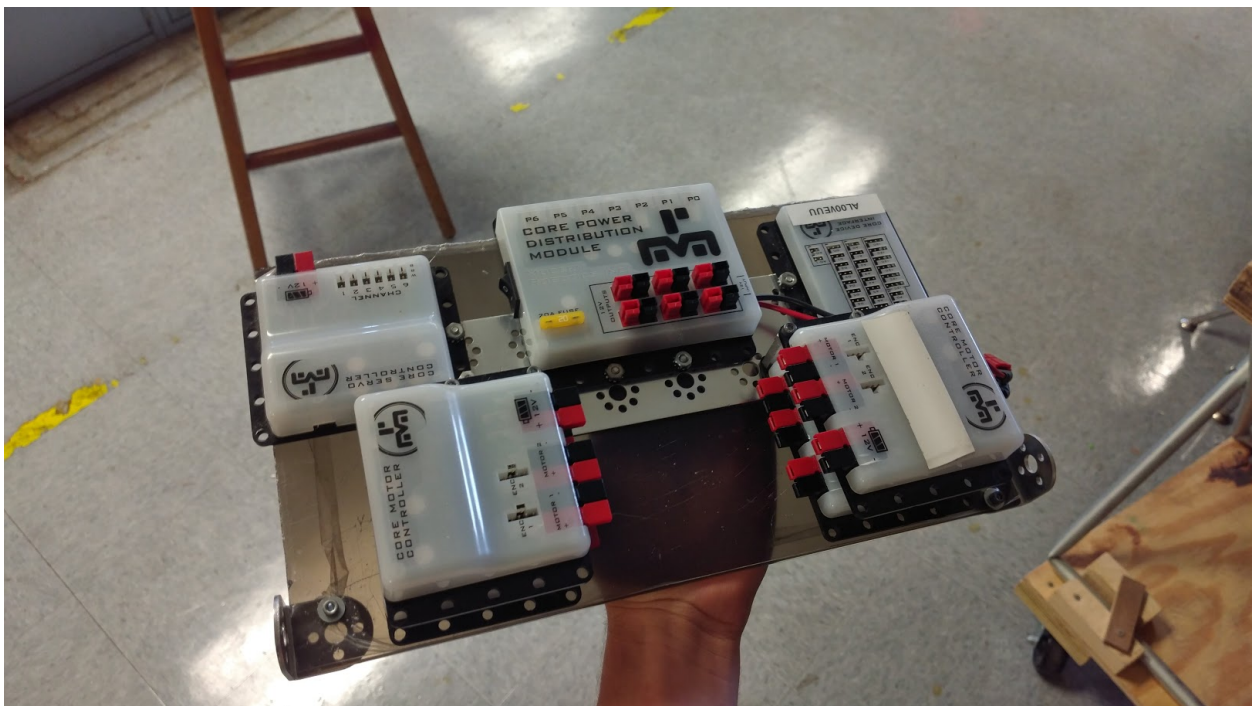


Fig. 113: 3736 Serious Business

Specific Recommendations

Module Power

Be aware, XT30 connectors can wear out significantly faster than PowerPole connectors. Additionally, because XT30 connectors are soldered, they can break much easier than PowerPole connections.

Cabling for module power should be at least 14awg, if not 12awg. Keep in mind that it must be stranded, not solid, wire.

Motor Power

Cabling for motor power should be between 16awg and 12awg. Again, stranded, not solid, wire.

Some motors (like the REV HD Hex and Core Hex) will have removable power connectors on the back, while other motors (like the Andymark NeveRests) will have a cable permanently soldered on the back. It is much more convenient to have a connector on the back, or failing that, have a very short plug on the back of the motor. Once your wires are run and secured, taking them out won't be fun.

Servo Wires

Using heavy-duty extension wires are recommended.

Tape the connections between extension wires and servo wires with electrical tape. This is as the connections can become loose over time and are easy to pull out.

The VEX Motor Controller 29 has the wrong gender connector on the 3-pin end. You are required to either use an adapter cable, or add the right connector to the wires (recommended). **Be sure to protect the MC29, as it is fragile and prone to failure if it takes impact from another object.**

USB

USB is generally a strong connector, but is prone to wearing out over time. Refrain from plugging/unplugging these cables more than necessary, especially on the RC/DS phones.

USB loves strain relief. To keep disconnects low, tie down cables to leave as little loose cabling on the robot as possible.

Sensor Wires/Encoder Wires

Sensor wires and their connectors can be incredibly fragile. Use caution when routing, and keep slack on the connector end when adding strain relief to the cable.

The JST data connectors on the REV Expansion Hub and Control Hub have +5v, GND, and two data pins. If you are using a digital or analog sensor that does not use I²C, you can use a Y cable that gives two sensors off of one port.

Miscellaneous

REV Grounding Strap The REV Grounding Strap is currently the only legal way to ground your robot. Attach the end to the metal part of your robot frame, and plug the XT30 connector end into a free XT30 port on your robot.

Power distribution blocks/panels The REV Power Distribution Block allows teams to have more than four XT30 connectors (2 on each Expansion Hub). The block can be connected to the Servo Power Module to boost voltage for servos or provide continuous rotation for the VEX 393 EDR.

Dryer sheet Dryer sheets can be used to wipe down the robot after every match in order to reduce static buildup. This is not directly recommended by FIRST or any vendor, but our empirical evidence throughout the years suggests that it helps, or at the very least, can't hurt to do so.

Staticide/static spray Staticide is a spray that helps to keep static off of the robot. Please be sure to spray your robot before an event and not during the event.

Common causes of static

- Every single contact point of your robot to the floor will increase the amount of static buildup.
- Too much turning scrub (or traction when wheels try to turn). This is possible if a 4WD or 6WD (no center drop) with all traction wheels is used.
- A conductive part dragging along the ground. For example, try not to have an intake touch the ground when the robot is moving as much as possible. Foam wheels and foam rollers are a common culprit.

Sensor Glossary

Sensors are used in a variety of applications within FTC. Sensors can give external feedback regarding the position of the robot (for example, relative to the field wall or to a vision target) or internal feedback (velocity, distance traveled, voltage, etc.). Sensors can also be used to determine the rotation of a mechanism and detect color.

Encoders/Potentiometers

- **Rotational**
 - **Absolute (+ Potentiometers)**
 - ★ MA3 ([am-2899](https://www.andymark.com/products/absolute-encoder-with-cable)³⁷)
 - ★ Potentiometer ([REV-31-1155](http://www.revrobotics.com/rev-31-1155/)³⁸)
 - **Relative**
 - ★ E4T ([am-3132](https://www.andymark.com/products/e4t-oem-miniature-optical-encoder-kit)³⁹)
 - ★ Generic ([Sparkfun Rotary Encoder](https://www.sparkfun.com/products/9117)⁴⁰)
 - **Both**
 - ★ CTRE (VEXpro) Mag Encoder ([217-5049](https://www.vexrobotics.com/217-5049.html)⁴¹)

³⁷ <https://www.andymark.com/products/absolute-encoder-with-cable>

³⁸ <http://www.revrobotics.com/rev-31-1155/>

³⁹ <https://www.andymark.com/products/e4t-oem-miniature-optical-encoder-kit>

⁴⁰ <https://www.sparkfun.com/products/9117>

⁴¹ <https://www.vexrobotics.com/217-5049.html>

- **Positional**

- Linear Potentiometers Slide Pot ([Sparkfun Slide Pot⁴²](#))

Contact

- **Physical**

- Endstops Generic ([Sparkfun⁴³](#))
- Touch Sensor REV ([REV-31-1425⁴⁴](#))

- **Magnetic**

- Hall Effect Sensor REV ([REV-31-1462⁴⁵](#))

Optical

- **Color**

- Adafruit RGB
- REV Color
- MR Color

- **Computer Vision**

- **Hardware**

- ✱ PixyCMU

- **Software**

- ✱ OpenCV/DogeCV/EasyOpenCV
- ✱ Vuforia
- ✱ TFLite

Distance

- ToF
- Ultrasonic

Other

- **IMU**

- Accelerometer
- Gyroscope

⁴² <https://www.sparkfun.com/products/9119>

⁴³ <https://www.sparkfun.com/products/13013>

⁴⁴ <http://www.revrobotics.com/rev-31-1425/>

⁴⁵ <http://www.revrobotics.com/rev-31-1462/>

- Compass
- Magnetometer

Logic Level Converter

The old Modern Robotics system run on 5v sensor logic. The new Rev Robotics system uses 3.3v. For most off the shelf sensors, this doesn't cause any problems, but for some existing FTC sensors it does. To solve this Rev sells boards, called [logic level converters](#)⁴⁶, that convert the sensor data to be readable by the Rev hubs. The [Rev Expansion Hub](#)⁴⁷ guide has a chart detailing what adapters are needed for what sensors. According to Rev testing, goBILDA, Rev and TorqueNado motors don't need logic level converters, but only some NeveRest motors worked with no discernable reason why.

It is ideal to not use logic level converters to simplify your wiring. If you need to, there is a best practice. Electrical tape the connectors on either end, this helps with static, and it keeps it from being physically disconnected. This does produce a very noticeable effect with encoders on fields with lots of static. The second tip is to never tape over the middle or LED. The board generates a very small amount of heat, and it's very easy to overheat if it can't ventilate, also don't fully enclose it in any cases without holes.

Control Systems

There are three possible control systems that can be run on an FTC robot legally:

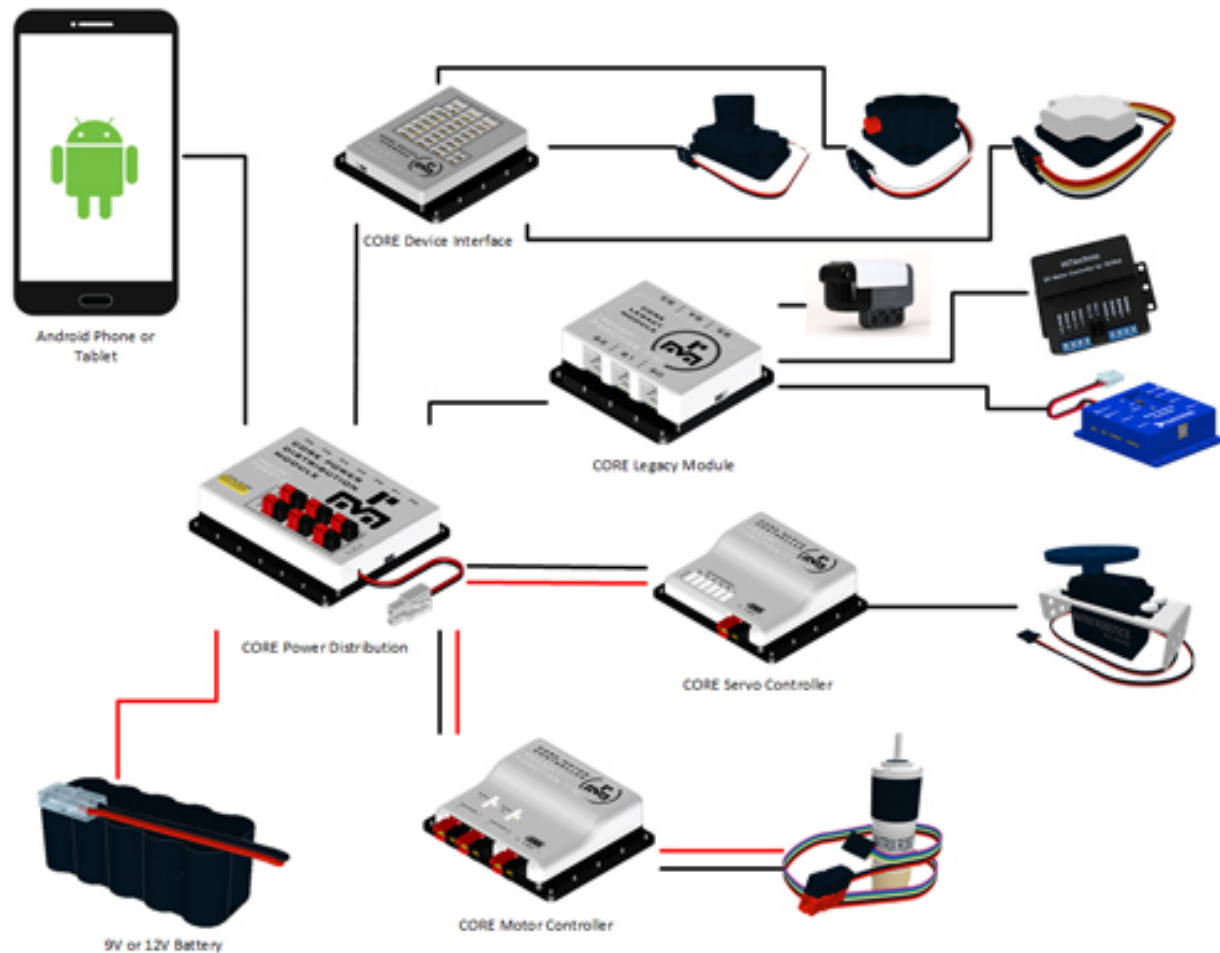
- RC Phone + Modern Robotics
- RC Phone + REV Expansion Hub(s)
- REV Control Hub + REV Expansion Hub

RC Phone + Modern Robotics

(Not included in the starter Electronics Kit). Modern Robotics is quickly being replaced by REV's control system, as it is less expensive and more reliable than the older Modern Robotics products. It is very likely that you can ignore anything Modern Robotics related, especially if your team is starting from scratch.

⁴⁶ <http://www.revrobotics.com/rev-31-1389/>

⁴⁷ <http://www.revrobotics.com/content/docs/REV-31-1153-GS.pdf>



RC Phone + REV Expansion Hub(s)

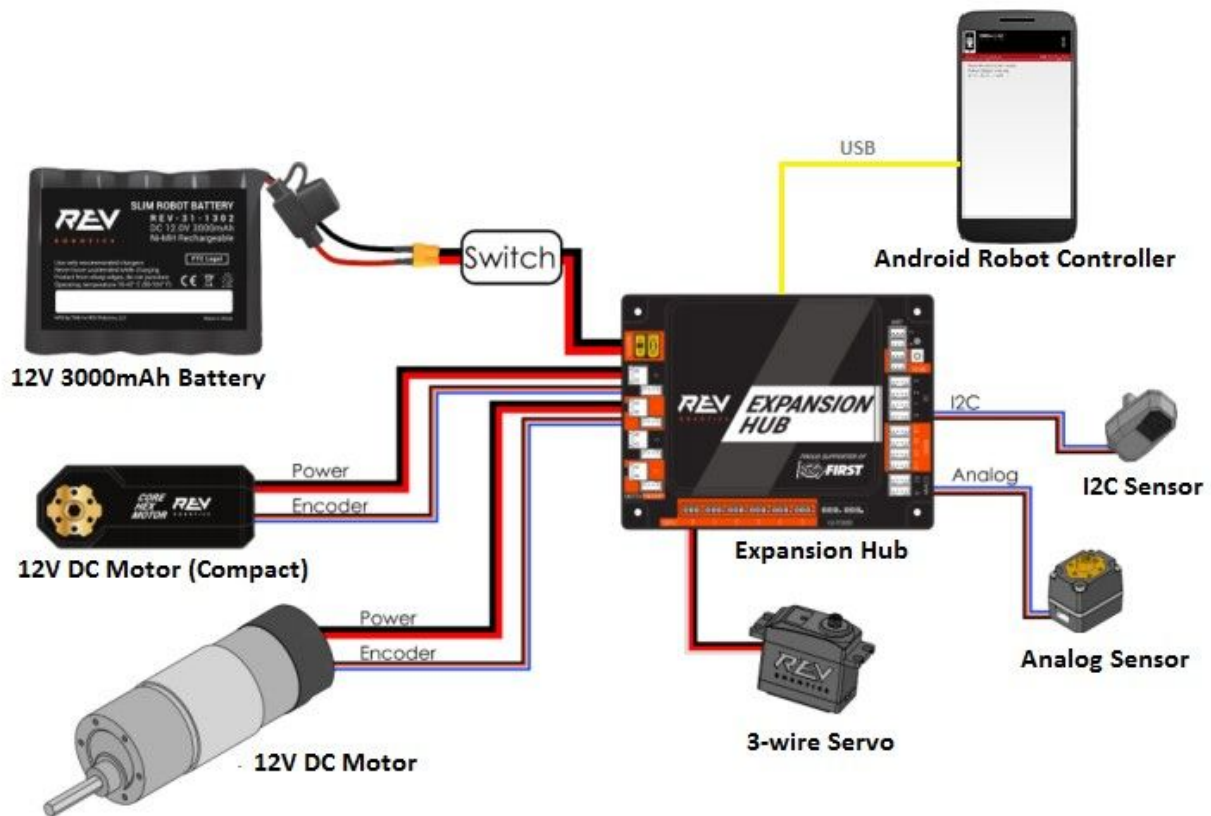
This is the standard control system for teams starting out in FTC. The REV Expansion Hub is reliable, as long as proper strain relief and wiring is carried out. This includes the USB Retention Mount, as well as 3D printing XT30 stress relief mounts. The Expansion Hub connects to the Robot Controller phone through the mini USB port, and the RC phone is linked to the DS (Driver Station) phone through WiFi Direct. For more information on setting up the Expansion Hub and configuring the robot, head to REV Robotics' Technical Resources page on their website.

- [USB Retention Mount](#)⁴⁸
- [XT30 Stress Relief](#)⁴⁹
- [REV Robotics Technical Resources](#)⁵⁰

⁴⁸ <http://www.revrobotics.com/rev-41-1214/>

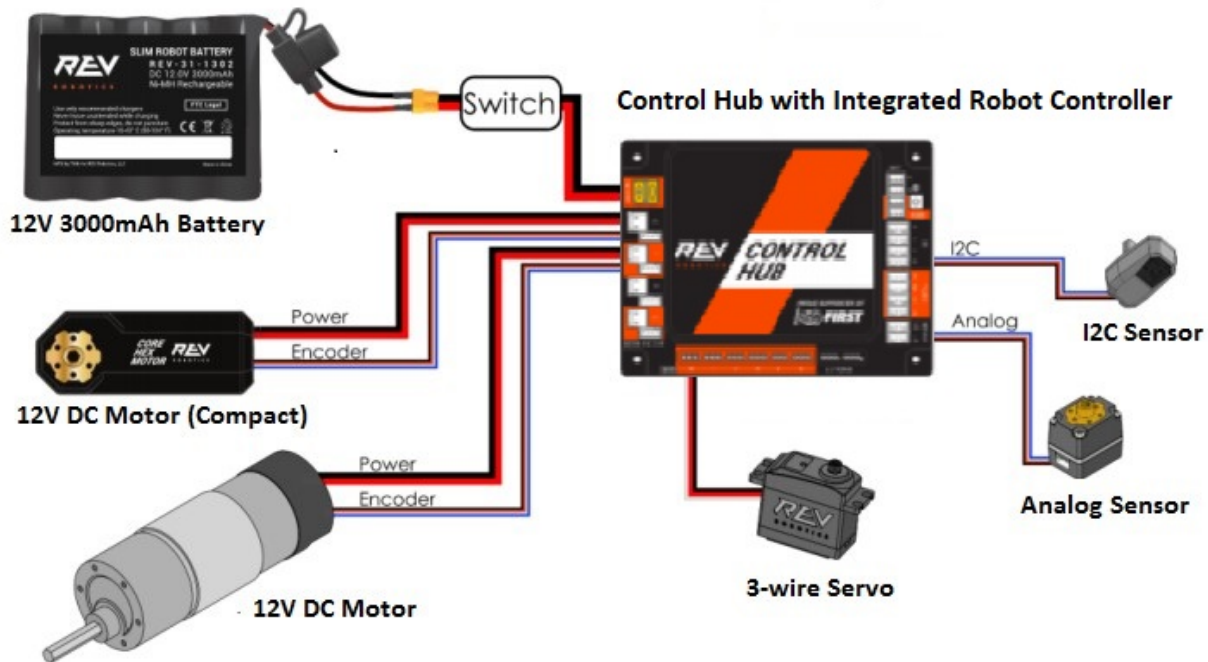
⁴⁹ <https://www.thingiverse.com/thing:2887045>

⁵⁰ <https://www.revrobotics.com/resources/>



REV Control Hub + REV Expansion Hub

Note: The Control Hub is currently undergoing a pilot program and is only legal for competition in specific regions throughout the US. It is not legal for all FTC teams for the 2019-2020 season.



1.16 Software

1.16.1 Fundamental Concepts of Programming

For almost any programming language, whether it's Java, Python, or Blocks, there are concepts in coding that transfer across languages. These ideas are foundational when learning to program and should be applicable in FTC and beyond.

This section is primarily for people with limited Java experience. However, even if you are more experienced, it may still be helpful to skim through the section, as you might find concepts that have not yet been introduced to you.

Examples will mostly be in Java, where `//` indicates a comment which the program ignores and is used for people to read.

```
int number; // Declaring that number will contain an integer.
number = 5; // Setting a value so that the variable holds something.
int secondNumber = 6; // Doing both above.
int total = number + secondNumber; // Math.
System.out.println(total); // Printing, it will show up as 11.
```

Java-Specific Exploratory Questions

- If I didn't set a value for number and then I printed it, what would it print?
- What other operations can I do with number and secondNumber?
- Can I set a decimal to number? If not, what happens?
- What is `System.out.println();`?
- Delete one character in the code. Remember the error (if any), and then undo it. Delete another part. How many different errors can you get?

There are different types of variables

- Numbers (Integers, Floats, Doubles)
- Strings (Text) or characters
- And a lot more depending on the language (Ex: Arrays)
- They help tell the program know the basis of what it should do with a variable.

```
String coolName = "Gluten Free";
String restOfSentence = " is epic.";

// Prints out the sentence by combining the strings, unlike adding if they // were
// integers
System.out.println(coolName + restOfSentence);

// Fun fact: Using + to add strings is called String Concatenation
```

Java-Specific Exploratory Questions

- Replace the text in coolName to something else. Your name, a phone number, your favorite anime. What about emotes and copypastas? What about characters in other languages?
- Try adding a number and a string, what happens?
- Is it possible to add multiple strings and numbers together?

Important Control Structures

Be sure to familiarize yourself with basic control structures (if/else statements, for loops, while loops, and for-each loops). These control structures are by far the most commonly encountered, and thus, familiarizing yourself with these principles is extremely important (not just for FTC, but programming in general). However, there are a few control structures that are far less common that are extremely useful in FTC.

Finite State Machines and Enums

A Finite State Machine (FSM) is a control structure that executes one of several code segments depending on its state. The logic can be thought of using if-statements:

```

if (state == 0) {
    doThing1();
}
else if (state == 1) {
    doThing2();
}
else if (state == 2) {
    doThing3();
}

```

In this example, the state variable can have 1 of 3 values: 0, 1, and 2. Depending on what value state has, different code segments are executed. However, instead of using primitive types or strings to store this information, the data type known as an Enum can be used instead.

Enums are a user-defined data structure that allow you to easily create and keep track of states in a state machine. **Each state should represent one unique, independent action of the robot.** Here's one simple example when a state machine is applied to a scoring lift:

This is how the state machine is defined:

```

public enum LiftState {
    IDLE,
    EXTEND,
    RETRACT
}

```

Then, you can initialize the a LiftState enum like this:

```
LiftState liftState = LiftState.IDLE;
```

And here is how to use the enum LiftState to create a state machine:

```

// anything inside the brackets after switch(autoState) defines the robot's
// action at each state
switch (liftState) {
    //case IDLE is equivalent to if(liftState == LiftState.IDLE)
    case IDLE:
        if(gamepad1.b) {
            liftState = LiftState.EXTEND;
        }
        if(gamepad1.a) {
            liftState = LiftState.RETRACT;
        }
        //break tells the program it should no longer execute the code within the
        ↪ IDLE state and can move on to check the next state
        break;
    case EXTEND:
        if(topTouchSensor.isPressed()){
            liftMotor.setPower(0);
            liftState = LiftState = LiftState.IDLE;
        }
        else {
            liftMotor.setPower(1);
        }
        break;
    case RETRACT:
        if(bottomTouchSensor.isPressed()) {

```

(continues on next page)

(continued from previous page)

```

        liftMotor.setPower(0);
        liftState = LiftState = LiftState.IDLE;
    }
    else {
        liftMotor.setPower(-1);
    }
    break;
}

```

Data Structures (Arrays)

Data structures are a method of organizing and storing large amounts of data. There are a lot of different types of data structures that mostly differ in the relationships between data points, and we would recommend that you read into them. We will only go over a few here.

Arrays⁵¹ Arrays are the most basic and simple data structure. When an array is initialized, its size must be set, and it cannot be changed. If you wish to expand an array, a new one must be created and all of the old data copied over. Elements of an array are stored adjacent to each other in memory, so when they are accessed the number you want to access times the amount of bits in the object in the array is added to the starting address, and data is accessed from there. This means that arrays are incredibly efficient at reading data in a nonlinear order.

ArrayList⁵²

Object Oriented Programming in Java⁵³

1.16.2 Options for Programming

There are three recommended options for teams to use when programming a robot to do various tasks: Blocks, OnBot Java, and Android Studio.

Blocks

The Blocks Programming Tool is a simple way to code simple actions for the robot. It has a colorful and lego-like design and includes sounds of block pieces of snapping together.

Using a device and a robot controller phone, connect your device to the phone's Wi-Fi by clicking the three dots in the top right of the RC and select "program / manage". This will display the WiFi SSID and password for you to connect to. Go onto your browser (preferably Chrome) and enter the address shown on the RC.

Advantages

- Good for beginners: one of the easiest and simplest options to use.
- Easy to set up: requires a device (laptop, chromebook, tablet) and a phone in the Program & Manage screen
- **Programs can be saved directly to the phone.**

⁵¹ <https://www.geeksforgeeks.org/arrays-in-java/>

⁵² <https://www.geeksforgeeks.org/arraylist-in-java/>

⁵³ <https://www.geeksforgeeks.org/classes-objects-java/>

- Changes can be made quickly
- Doesn't require an external Wi-Fi connection

Disadvantages

- **Not recommended if you're already familiar with some type of programming**
 - Primarily a teaching tool for people with no programming experience
- This will change your primary Wi-Fi network to the Robot Controller's Direct Wi-Fi network, and as such, you will be unable to access the internet while connected to this particular network.
 - Teams can get around this by purchasing an external Wi-Fi module that allows computers to run 2 Wi-Fi networks at once.
- Sacrifices flexibility and application for simplicity
- Don't ever use it on a phone, unless you're out of options. (Not phone-friendly) * You may make a bigger mess if you try to.

OnBot Java

OnBot Java uses a similar method of using a browser to code and save directly to the phone. The difference being that a programming language called Java is used instead.

Advantages

- Recommended if you're learning or have learned some programming, even better if you know a little Java.
- Greater flexibility than Blocks
- There are a lot more resources available in case you need help
- More applicable to the real-world than Blocks
- Maintains most of the advantages of Blocks

Disadvantages

- Connecting to the robot controller's Wi-Fi network will prevent you from using anything on the internet including video tutorials and online communication
- Using external libraries is difficult and borderline impossible
- Steeper learning curve than Blocks

Android Studio

Android Studio is a comprehensive Integrated Development Environment (IDE) that uses Java to program the phones. Instead of using a browser to upload code, Android Studio will compile your Robot Controller code into a .apk file (an app installer), and install that on the phone.

Advantages

- Recommended if you're learning or have learned some programming, even better if you know a little Java.
- Much greater flexibility than Block
- Much easier to integrate libraries like OpenCV, DogeCV, Road Runner, OpenFTC, etc.
- Can use plugins like [Road Runner](#)⁵⁴
- Can use either a USB connection to the RC phone, or a wireless connection to upload code
- Can debug in real-time
- Many resources for Java, Android Studio, and IDEA
- Can use other programming languages

Disadvantages

- Connecting to the robot controller's Wi-Fi network will prevent you from using anything on the internet including video tutorials and online communication, unless you have a second Wifi adapter (cheap and easy)
- Relatively easy setup process, but time consuming and is a hefty install (3GB of files between Android Studio, ftc_app, and other libraries)
- Issues can be difficult to diagnose and solve

Other Programming Languages

Kotlin

Kotlin is a relatively new and rapidly growing programming language from JetBrains, the creator of the IntelliJ IDE, which Android Studio is based off. Kotlin was made to be completely compatible with Java but be easier to work with. Google recently announced it as an official Android language, then announced they are going "Kotlin first" but still keeping Java support.

Advantages

- Concise, readable, easy to edit code
- Easy to write as it takes much less code to do the same thing
- Both optional type inference and a stronger type system than Java
- Null safety
- Thread safety
- Functional programming
- Seamless integration with Java code and libraries
- Very easy to transition from Java

⁵⁴ <https://github.com/acmerobotics/road-runner>

Disadvantages

- Not widely used in FTC yet
- New and has fewer community resources for training
- Not recommended for programmers who need large amounts of help from other teams

C and C++

C and C++ are native programming languages compatible with Android. Very few teams have used C++. This is typically used for only part of the code, with the majority being Java or Kotlin.

Advantages

- Fast execution for extremely resource-intensive applications.
- Supports more libraries

Disadvantages

- Rarely needed
- Very difficult to set up
- Difficult to debug code
- Very few teams can help you
- Very few online resources

1.16.3 Using the FTC SDK

LinearOpMode vs OpMode

There are two OpMode classes within the FTC SDK: OpMode and LinearOpMode. The one you use affects how you write the program. For examples of how to use OpMode and LinearOpMode, [refer to the example OpModes in the sdk](#)⁵⁵.

LinearOpMode Methods

- `runOpMode()`: Code inside this method will run exactly once after you press the INIT button. This is where you should put all code for the OpMode.
- `waitForStart()`: This method pauses the Op-Mode until you press the START button on the driver station.
- `isStarted()`: returns `true` if the START button has been pressed, otherwise it returns `false`.

⁵⁵ <https://github.com/FIRST-Tech-Challenge/SkyStone/tree/master/FtcRobotController/src/main/java/org/firstinspires/ftc/robotcontroller/external/samples>

- `isStopRequested()`: returns true if the STOP button has been pressed, otherwise it returns false.
- `idle()`: puts the thread to sleep
- `opModeIsActive()`: returns `isStarted()` && `!isStopRequested()` and calls `idle()`.

OpMode Methods

- `init()`: Code inside this method will run exactly once after you press the INIT button on the driver station.
- `init_loop()`: Once the code in `init()` has been run, code inside this method will run continuously until the START button is pressed on the driver station.
- `start()`: Code inside this method will run exactly once after you press the START button on the driver station.
- `loop()`: Once the code in `start()` has been run, code inside this method will run continuously until the STOP button is pressed on the driver station.
- `stop()`: Code inside this method will run exactly once after you press the STOP button on the driver station.

Note that unlike LinearOpMode, all methods in OpMode must be overwritten to be used.

Reading and Writing to Hardware

When using the FTC SDK, there are a variety of built in hardware classes. Objects can be created for these classes and then instantiated. These objects can then be used to communicate with hardware on the robot such as DC Motors, Servos, and Sensors.

Creating and Instantiating Hardware Objects

The first thing required to properly create an object is to import its class. In Android Studio, if the class is referenced without being imported Alt+Enter can be pressed to automatically import it. After it is imported, the next step is to create the object:

```
private DcMotor liftMotor;
```

After the object is created, it must be instantiated. Part of the opmode superclass is something called hardwareMap. hardwareMap is used in the FTC SDK to instantiate objects rather than calling a constructor. It contains all of the information entered into the configuration on the Robot Controller, such as names of hardware and what port it is plugged into. Here is an example of instantiating the motor we created above:

```
liftMotor = hardwareMap.get(DcMotor.class, "Lift Motor");
```

Whatever sensor you are using, you will pass that class into the spot where `DcMotor.class` is. For example, if liftMotor was a Servo, `Servo.class` would be passed instead. For the second argument, you pass whatever the device is named in the Robot Controller configuration. hardwareMap will then go find what port the device with that name is plugged into, which allows the hardware to be accessed.

Examples of Using Common Hardware Components

Built in to the FTC SDK are many examples of using things such as Color Sensors, Distance Sensors, Servos, Motors, etc. Here, we would like to give more of an explanation of using Motors and Servos because there are a few things the examples do not teach.

DC Motor

```
DcMotor leftMotor = hardwareMap.get(DcMotor.class, "Left Motor");
DcMotor rightMotor = hardwareMap.get(DcMotor.class, "Right Motor");

DcMotor elevatorMotor = hardwareMap.get(DcMotor.class, "Elevator Motor");
DcMotor intakeMotor = hardwareMap.get(DcMotor.class, "Intake Motor");
```

After a DcMotor is instantiated, there are a few variables you can set to affect how the DC Motor runs. The first of these is direction:

```
leftMotor.setDirection(DcMotor.Direction.REVERSE);
rightMotor.setDirection(DcMotor.Direction.FORWARD);
```

Changing the direction of the motor does exactly what should be expected, it changes the direction. If a power of 1 is applied to the motor while it is in forward mode, it will turn one direction. If it is in reverse, a power of 1 will spin it in the other direction. If you face the shaft of the motor towards you, forward is counterclockwise (with the exception of NeveRest motors).

Next, there are two zero power behaviors that can be adjusted:

```
leftMotor.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);
rightMotor.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.FLOAT);
```

Changing this variable affects how the DC Motor behaves while a power of 0 is applied. BRAKE will cause the motor to try and slow itself down if it is moving (it will NOT cause the motor to hold its position if not already moving), while FLOAT causes the motor to glide to a stop, letting friction do all the work.

Finally, there are four different run modes that can be used with DC motors:

```
leftMotor.setMode(DcMotor.RunMode.RUN_WITHOUT_ENCODER);
rightMotor.setMode(DcMotor.RunMode.RUN_USING_ENCODER);

elevatorMotor.setMode(DcMotor.RunMode.RUN_TO_POSITION);
intakeMotor.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);
```

It is important to note that encoder values can be read in any of these modes provided an encoder is properly plugged in. These modes just change how the motor reacts to these encoder values. RUN_WITHOUT_ENCODER makes the motor behave as if there is no encoder plugged in. When setPower() is called, it sets the output voltage to the motor directly.

Using RUN_WITH_ENCODER, the power set takes a more indirect route to the motor. It first goes through a velocity PID, and the output from that controller is output to the motor. This effectively means that setPower() sets the speed of the motor, not the power. If a power of .2 were fed while this mode is active, the motor will attempt to turn the same speed by fluctuating the output voltage depending on the load on the motor. This mode has one significant disadvantage, however. The max speed of the motor is somewhat significantly decreased, so it is recommended to use RUN_WITHOUT_ENCODER if possible if maximum speed is the goal; however, RUN_USING_ENCODER will provide more consistent results.

The final mode is RUN_TO_POSITION. To make the motor move with this mode, the function `setTargetPosition()` must be called. When a power is applied to the motor, a control loop will use that as the max power and try to drive the encoder position to the target position. This can be useful to newer teams for autonomous, as it can be an easy way to have accurate driving functions.

Servo

```
Servo relicServo = hardwareMap.get(Servo.class, "Release Servo");
```

After instantiating a Servo, there are two main functions that can be called: `setPosition()` and `getPosition()`.

```
releaseServo.setPosition(0.75);
telemetry.addData("Release Servo Target", releaseServo.getPosition());
```

`setPosition()` sets the position of the servo. The SDK will use a built-in control loop with the servo's potentiometer to drive the servo to that position and hold that position. `setPosition()` takes in a double between 0 and 1, where 0 is the servo's lower limit of rotation and 1 is the servo's upper limit of rotation. Everything between is directly proportional, so 0.5 is the middle, 0.75 is 3/4 the way up, etc.

`getPosition()` does not return the servo's current position, rather its current target position. If a variable for the servo's current target position is stored properly, this function should never be needed.

Continuous Rotation Servo

```
CRServo intakeServo = hardwareMap.get(CRServo.class, "Intake Servo");
```

A CRServo has one main method; `setPower()`. This works very similarly to DcMotor's `setPower()`, meaning that passing it 0 makes it stop, passing it 1 makes it go forward at full speed, passing it -1 makes it go backwards at full speed, and everything in between.

```
intakeServo.setPower(0.75);
```

Gamepad Input

A very important aspect of programming a driver controlled opmode is taking driver controls. Thankfully in the FTC SDK, this is very easy to do. Inside of every opmode, there are already 2 working gamepad objects, `gamepad1` and `gamepad2`. `gamepad1` is the controller that is connected using start+a, while `gamepad2` is the controller connected using start+b. To get input, no functions need to be called, rather static variables need to be accessed. Here are a few examples:

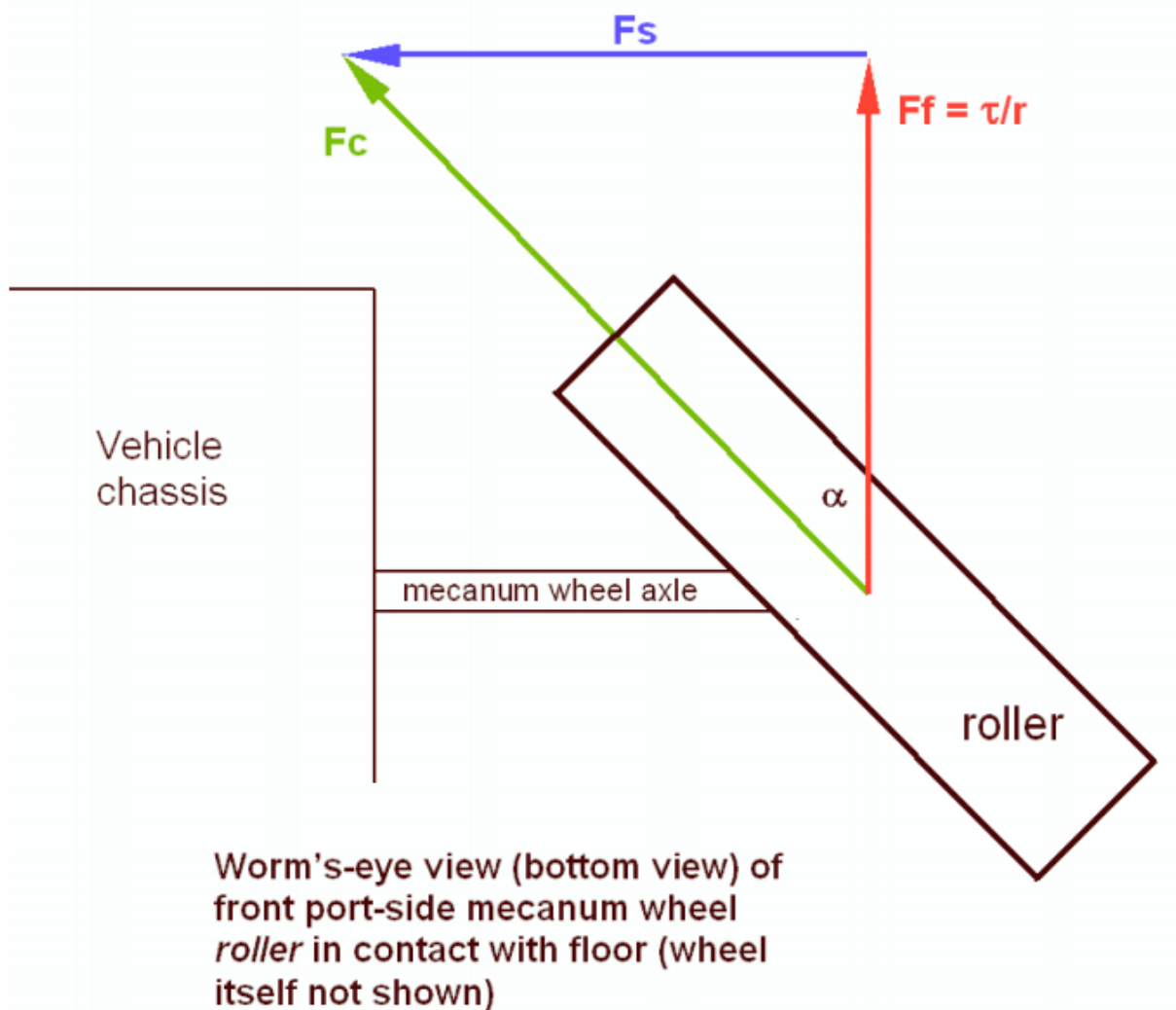
```
leftMotor.setPower(-gamepad1.left_stick_y);
rightMotor.setPower(-gamepad1.left_stick_y);

if (gamepad2.a) {
    intakeServo.setPower(-1.0);
}
else if (gamepad2.b) {
    intakeServo.setPower(1.0);
}
```

1.16.4 Programming Tutorial - Mecanum Drivetrain

Mecanum Physics

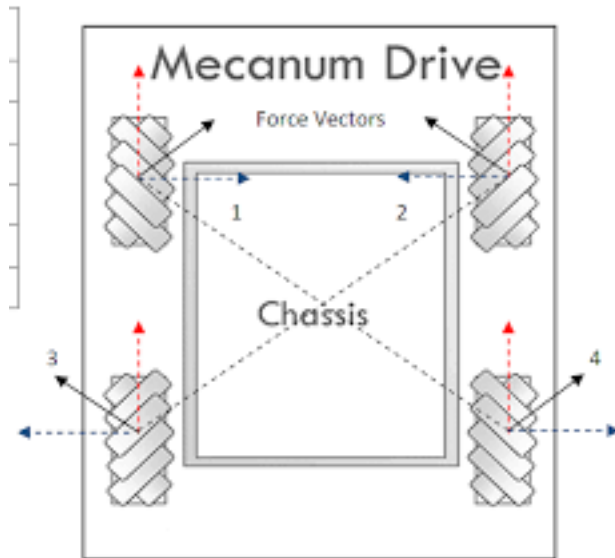
A very common choice for drivetrain in FTC is mecanum drive, but we have noticed that not a lot of people know how they work or how to properly program them. Mecanum wheels have rollers at a 45° angle to the rest of the wheel. Since these are in contact with the ground instead of something solid like in a traction wheel, instead of the wheel creating a force parallel to the orientation of the wheel, it creates one 45° from parallel. Depending on how the wheels are driven, X or Y components of the force vectors can cancel which allows movement in any direction.



Using Vectoring to Create Omnidirectional Movement

As stated earlier, depending on how the wheels are driven, movement in any direction can be created. This is because with 4 wheels on a robot oriented in an X orientation, the vectors can be cancelled such that movement is created in any direction. It is important to note that because of friction, perfect movement isn't

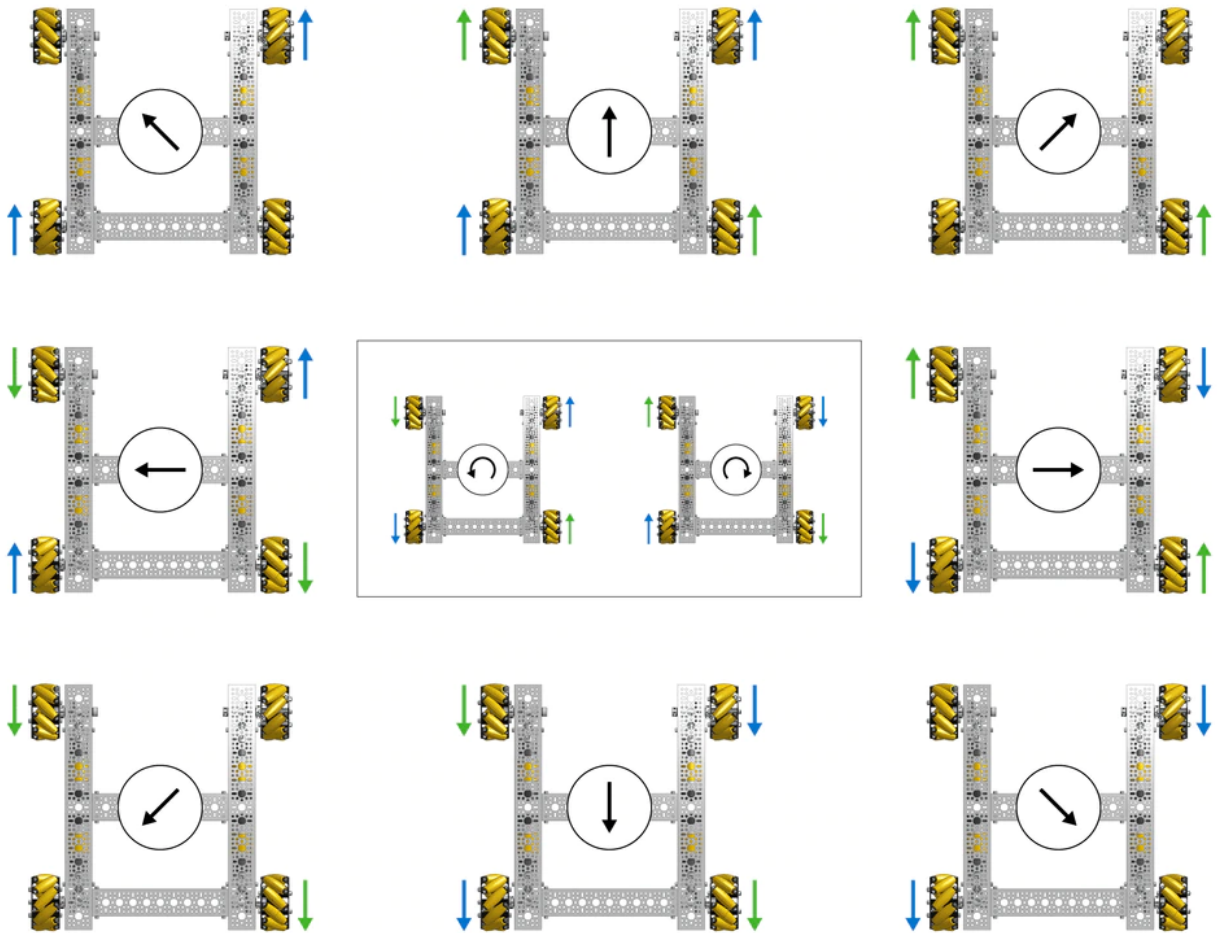
possible in every direction, so a mecanum drivetrain will be able to drive slightly faster forwards/backwards than any other directions.



In that image, 1, 2, 3, and 4 are the force vectors created by the mecanum wheels when driven towards the top of the image. The blue and red lines are their X and Y components, respectively. Here are a few examples of how the wheels must be driven to achieve different movements (It is not recommended to hard code these in; there is a much better mathematical way to program mecanum that allows for true omnidirectional movement and is much cleaner):



PRODUCT INSIGHTS



Deriving Mecanum Control Equations

Before thinking about mecanum, envision a scenario where you have a 2 motor tank drivetrain which you want to control using the left stick Y axis for forward/backward movement, and the right stick X axis for pivot turning. The motors are configured so that positive is clockwise for the right motor when the body is facing away from you, and the left motor is the opposite. To control only forward/backward movement, you simply need to set the motor powers to the Y stick value (flip the sign since Y is reversed):

```
double y = -gamepad1.left_stick_y; // Remember, this is reversed!
```

(continues on next page)

(continued from previous page)

```
leftMotor.setPower(y);
rightMotor.setPower(y);
```

Although at first adding rotation might seem like a difficult task, it's actually super simple. All you need to do is subtract the x value from the right side, and add it to the left:

```
double y = -gamepad1.left_stick_y; // Remember, this is reversed!
double x = gamepad1.right_stick_x;

leftMotor.setPower(y + x);
rightMotor.setPower(y - x);
```

Here, if the Y stick is pressed upwards, both of the motors will be fed a positive value, causing the robot to move forward. If it is pressed downwards, both of the motors will be fed a negative value, causing the robot to move backwards. A similar principle applies for rotation: if the X stick is pushed rightward, the left wheels will spin forward while the right spin backward, causing rotation. The opposite applies for pushing the stick left. If both sticks are pushed at the same time, say the Y stick is at 1 and the X stick is also at 1, the value of the left wheels will be $1 + 1 = 2$ (which gets converted to 1 in the SDK) and the right wheels will be $1 - 1 = 0$, which causes a rightward curve.

Applying omnidirectional movement with mecanum wheels operates under the same principle as adding turning into the tank example. The left stick X values will be added or subtracted to each wheel depending on how that wheel needs to rotate to get the desired movement. The only difference between adding turning is that rather than wheels on the same side being the same sign, wheels diagonal to each other will be the same sign.

We want a positive X value to correlate to rightward strafing. If we refer back to the vectoring image, this means that the front left and back right need to rotate forward, while the back left and front right need to rotate backwards. So, we should add the x value to the front left and back right and subtract it from the back left and front right:

```
double y = -gamepad1.left_stick_y; // Remember, this is reversed!
double x = gamepad1.left_stick_x;
double rx = gamepad1.right_stick_x;

frontLeftMotor.setPower(y + x + rx);
backLeftMotor.setPower(y - x + rx);
frontRightMotor.setPower(y - x - rx);
backRightMotor.setPower(y + x - rx);
```

This is the same as the tank example, except now with 4 motors and the strafing component added. Similarly to the tank example, the Y component is added to all wheels, and the right x (rx) is added to the left and subtracted from the right. Now, we have added another component that will allow us to strafe rightward. In doing that, however, we have actually allowed for strafing in any direction. If you think about it, pressing the joystick to the left will do the same thing in reverse, which is what is needed to strafe left. If it is pressed at 45 degrees, the x and y components of the joystick will be equal. This will cause two diagonal motors to cancel, allowing for diagonal movement. This same effect applies to every angle of the joystick.

Now that we have a functioning mecanum driving program, there are a few things that can be done to clean it up. The first of these would be multiplying the left X value by something to counteract imperfect strafing. Doing this will make the drive feel more accurate on non axis aligned directions, and make field centric driving more accurate. In this tutorial, we will use 1.5, but it's really up to driver preference.

```
double y = -gamepad1.left_stick_y; // Remember, this is reversed!
double x = gamepad1.left_stick_x * 1.5; // Counteract imperfect strafing
double rx = gamepad1.right_stick_x;
```

The other improvement we can make is scale the values into the range of -1 to 1. Since the SDK simply rounds if the input is out of that range, we can lose the ratio we are looking for unless we proactively put all the numbers back in that range while still maintaining our calculated ratio. For example, if we calculate values of 0.4, 0.1, 1.1, and 1.4, plugging those into the motors they will become 0.4, 0.1, 1.0, and 1.0, which is not the same ratio. Instead, we need to divide all of them by the largest number (absolute value):

```
// Put powers in the range of -1 to 1 only if they aren't already (not
// checking would cause us to always drive at full speed)

if (Math.abs(frontLeftPower) > 1 || Math.abs(backLeftPower) > 1 ||
    Math.abs(frontRightPower) > 1 || Math.abs(backRightPower) > 1 ) {
    // Find the largest power
    double max = 0;
    max = Math.max(Math.abs(frontLeftPower), Math.abs(backLeftPower));
    max = Math.max(Math.abs(frontRightPower), max);
    max = Math.max(Math.abs(backRightPower), max);

    // Divide everything by max (it's positive so we don't need to worry
    // about signs)
    frontLeftPower /= max;
    backLeftPower /= max;
    frontRightPower /= max;
    backRightPower /= max;
}
```

1.16.5 Control Loops

Control loops are software used to operate power transmission systems (such as a drivetrain or linear slide) in a fast and controlled fashion. Not only do control loops let you run mechanisms quickly without fear of losing control, in many cases, they help preserve the longevity of mechanisms by reducing rapid change of applied motor voltage.

What is Error?

The first thing that must be defined when discussing control loops is the concept of error. Error is defined as the difference between where you are and where you want to be. For instance, say you tell your drivetrain to drive at 30 inches per second, but in actuality, at a time, the drivetrain is driving at 28 inches per second. Since $30 - 28 = 2$, the error of the drivetrain's speed at this time T is 2 inches per second. In other words, at a time $t = T$, $e(t) = 2$.

PID

A PID controller (or Proportional Integral Derivative controller) is a control loop that solely uses error to control the system. PID is a form of a **feedback control loop**, or **closed loop control**. This means that data about the variable you are controlling is required in order for the loop to control that variable. In this case, information about the **error** of the system is required to control the system with a PID controller.

The Optional Calculus

The following equation represents the rigorous mathematical definition of the output of a PID controller f at any given time t :

$$f(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

where K_p , K_i , and K_d are constants and $e(t)$, as previously mentioned, is the error in the system. If you have no experience with calculus, don't worry; while PID is fundamentally rooted in calculus, you do not need any calculus experience to be able to understand it, only basic algebra. However, you are still urged to read the rest of the section regardless of calculus experience, as the formula alone doesn't tell you why it works.

Simplification of the PID formula

Here is a simplified version of the PID formula: $f(t) = K_p P(t) + K_i I(t) + K_d D(t)$

All we have done is simply take the full formula and replaced part of the terms with functions: $P(t)$, $I(t)$, and $D(t)$.

The Proportional Term

The first component of the function, $K_p P(t)$, is by far the most simple and easy to understand, as $P(t) = e(t)$. For the sake of example, let's pretend that $K_i = 0$ and $K_d = 0$ (a PID controller with only a proportional constant is known as a **P controller**). How will the system behave? Well, if the error is large, the output will be large. Likewise, if the error is small, the output will be small. Also, ideally, given enough time, the system always approaches its destination, assuming K_p is of the correct sign.

Say we apply this to a drivetrain. You want to drive a distance D , and you decide to set your motor powers using a P controller to accomplish this. In this case, your error is how far away the robot is from the desired location. As you start to drive forward, your error is large, so you drive forward quickly, which is desirable. After all, you aren't concerned with overshooting the target yet if you are far away from it. But as the robot's distance to the target approaches 0, you will start to slow down, gaining more control over the robot. Once the error is zero, ideally, the robot will stop, and you have reached your destination. If you happen to overshoot, the error will become negative, and the robot will backtrack, repeating the process.

The Derivative Term

This term, $K_d D(t)$, is intended to dampen the rate of change of the error. In other words, it tries to keep the error constant. How is this done? Well, for those of you with calculus under your belt, $D(t) = \frac{de(t)}{dt}$. For those without calculus experience, it represents how fast the error is changing. Graphically, $D(t)$ is simply the slope of the error at any given time t . This slope can be calculated by keeping track of the error over successive iterations of the control loop. One iteration occurs at time t_n with an error of $e(t_n)$. At the next iteration, the time is t_{n+1} with an error of $e(t_{n+1})$. Thus, to find $D(t)$, simply find the slope of $e(t)$ given these two points.

The Integral Term

Admittedly, the integral term is the least important term for FTC PID control loops. With a properly tuned K_p and K_d , you often can just set K_i to 0 and call it a day. However, it can still be useful in some cases. Just

like the derivative term, the integral term intends to correct for overshoot. if the system thinks it reached its destination, it will stop, even when, in fact, the error is not yet 0. Perhaps the motor is no longer being supplied enough power to move. Well, given enough time, the integral term will increase the output (in this case, motor power), causing movement towards the destination. To explain without calculus, the integral term essentially sums the error over a specific interval of time. To do this, error in each loop iteration is added to a variable (in this case, $I(t)$). However, summing error this way has an unfortunate side effect: the longer the loop takes to complete one iteration, the more slowly this sum increases, which is obviously not desirable, as we don't want lag to affect how the robot moves. To compensate for this, before the error is added to $I(t)$, it is multiplied by how long the previous loop took to complete, or $t_{n+1} - t_n$, preventing lag from making the system sum more slowly.

So say the robot stops short of the target. The P and D combination aren't strong enough to move it forward to the destination. You can either tune K_p and K_d to compensate (**this is recommended**), or you can add the integral term to increase output (**this works too, but requires more attention and tuning to achieve the same result**).

PID Pseudocode

```
while True:
    current_time = get_current_time()
    current_error = desire_position-current_position

    p = k_p * current_error

    i += k_i * (current_error * (current_time - previous_time))

    if i > max_i:
        i = max_i
    elif i < -max_i:
        i = -max_i

    D = k_d * (current_error - previous_error) / (current_time - previous_time)

    output = p + i + d

    previous_error = current_error
    previous_time = current_time
```

Tuning a PID Loop

The most important thing to know while tuning a PID loop is how each of the terms affects the output. This can allow you to see which gains need to be adjusted. For example, if the target is not reached but instead the setpoint begins to oscillate around the target, it means there is not enough D gain. If the target is eventually reached, albeit very slowly, that means there is not enough P gain or the D gain is too high. In brief, the P variable drives the error towards zero, the I variable corrects for steady state error, and the D variable dampens the effects of the P variable, more so as error approaches zero, which prevents overshoot.

The most common method for tuning a PID controller is as follows:

1. Set the I and D gains to zero
2. Increase the P gain until there are oscillations around the target
3. Increase the D gain until no overshoot occurs

4. If there is steady state error, increase the I gain until it is corrected

An important thing to note is that most systems do not need both I and D control. Generally, systems without a lot of friction do not need an I term, but will need more D control. Systems with a lot of friction, on the other hand, generally do not need D control because the friction facilitates deceleration but need I control because the friction prevents the system from reaching the target otherwise.

For a more in-depth explanation, [click here](#)⁵⁶

Feedforward Control

One less popular but equally useful control loop is the feedforward controller (sometimes unofficially referred to in FTC as the PVA controller, or Position-Velocity-Acceleration controller). For those without a physics background, velocity is the speed and direction something is moving and acceleration is how fast velocity is increasing or decreasing. Unlike PID, feedforward controllers require you to input not only where you want to go and where you are, but how fast you want to be moving at all times. Unlike feedback control loops such as PID, feedforward control loops don't require information about the variable you want to control. Instead of controlling a variable directly, it controls how fast that variable changes.

Conceptually, the controller is made up of 2 separate P controllers (remember, a P controller is made up of just the proportional term of a PID loop). Each of these P controllers are added together to create a feedforward controller.

Just like we did with the PID formula, we can define the function like this: $f(t) = K_v * V(t) + K_a * A(t)$.

In most FTC applications, $f(t)$ controls the position of the output. As the name PVA suggests, the first term relates to velocity, and the second relates to acceleration. Just like in a P controller, each term contains a constant multiplied by an error term (in this case, $V(t)$ and $A(t)$). However, unlike a PID controller, each term has their own setpoints and endpoints, meaning error is calculated differently for each term.

Unlike desired position, your desired velocity is likely to change throughout the control loop. After all, the entire point of using control loops are to try to create a balance of speed and control of a system. Remember, in most situations, you want to approach your destination quickly if you are far away and slow down if you are close for more control. For the sake of example, let's say $v(t)$ is a magic function that could tell you exactly how fast you should be going at any point. To calculate velocity error, subtract your current velocity from the magic function $v(t)$. This magic function can also be used to create another magic function: $a(t)$. This magic function tells you how exactly how fast the velocity should change in order to get to the next magic velocity at any specified time.

The last step is finding the magic functions $v(t)$ and $a(t)$, **which can be obtained using motion profiles** (discussed next).

Feedforward Pseudocode

```
while True:
    current_time = get_current_time()

    current_velocity = (current_position - previous_position) / (current_time -
↪previous_time)
    current_velocity_error = desired_velocity - current_velocity

    current_acceleration = (current_velocity - previous_velocity) / (current_time -
↪previous_time)
```

(continues on next page)

⁵⁶ <https://blog.wesleyac.com/posts/intro-to-control-part-two-pid-tuning>

(continued from previous page)

```
current_acceleration_error = desired_acceleration - current_acceleration
output = F + k_v * current_velocity_error + K_A * current_acceleration_error

previous_velocity = current_velocity

# end of feedforward code

previous_error = current_error
previous_time = current_time
```

Motion Profiles

Motion profiling is a technique popularized in FRC that is starting to find its way to FTC. A motion profile is a function used to change the speed of a power transmission system in a controlled and consistent way by changing desired speed gradually rather than instantaneously. Let's illustrate this with an example: say you want your drivetrain, which is initially unmoving, to drive forward at full speed. Ordinarily, you would set all drivetrain motors to full power in the code. However, this can be problematic because even though you tell the motors to move at full speed instantaneously, the drivetrain takes time to get to full speed. This can lead to uncontrolled movements which have the potential to make autonomous less consistent and, perhaps more importantly, damage mechanisms. Motion profiling attempts to solve this issue.

Advantages

- More controlled and predictable movements
- Reduces rapid change of applied motor voltage

Disadvantages

- Can be slower

There are two main types of motion profiles: **Trapezoidal** profiles and **S-Curve** profiles. Trapezoidal profiles accelerate the system at a constant rate, and S-Curve profiles assume jerk (the speed acceleration changes) is constant. Given that S-Curve profiles are not optimal for controlling 2d trajectories (such as driving) and exist to reduce slippage (which usually only occurs when driving in FTC), trapezoidal profiles are recommended for most FTC applications.

Trapezoidal profiles get their name from the shape of the graph of velocity over time:

Here is some pseudocode for a trapezoidal profile:

```
while True:
    current_velocity = get_current_velocity()
    current_time = get_current_time()

    direction_multiplier = 1

    if position_error < 0:
        direction_multiplier = -1

    # if maximum speed has not been reached
```

(continues on next page)

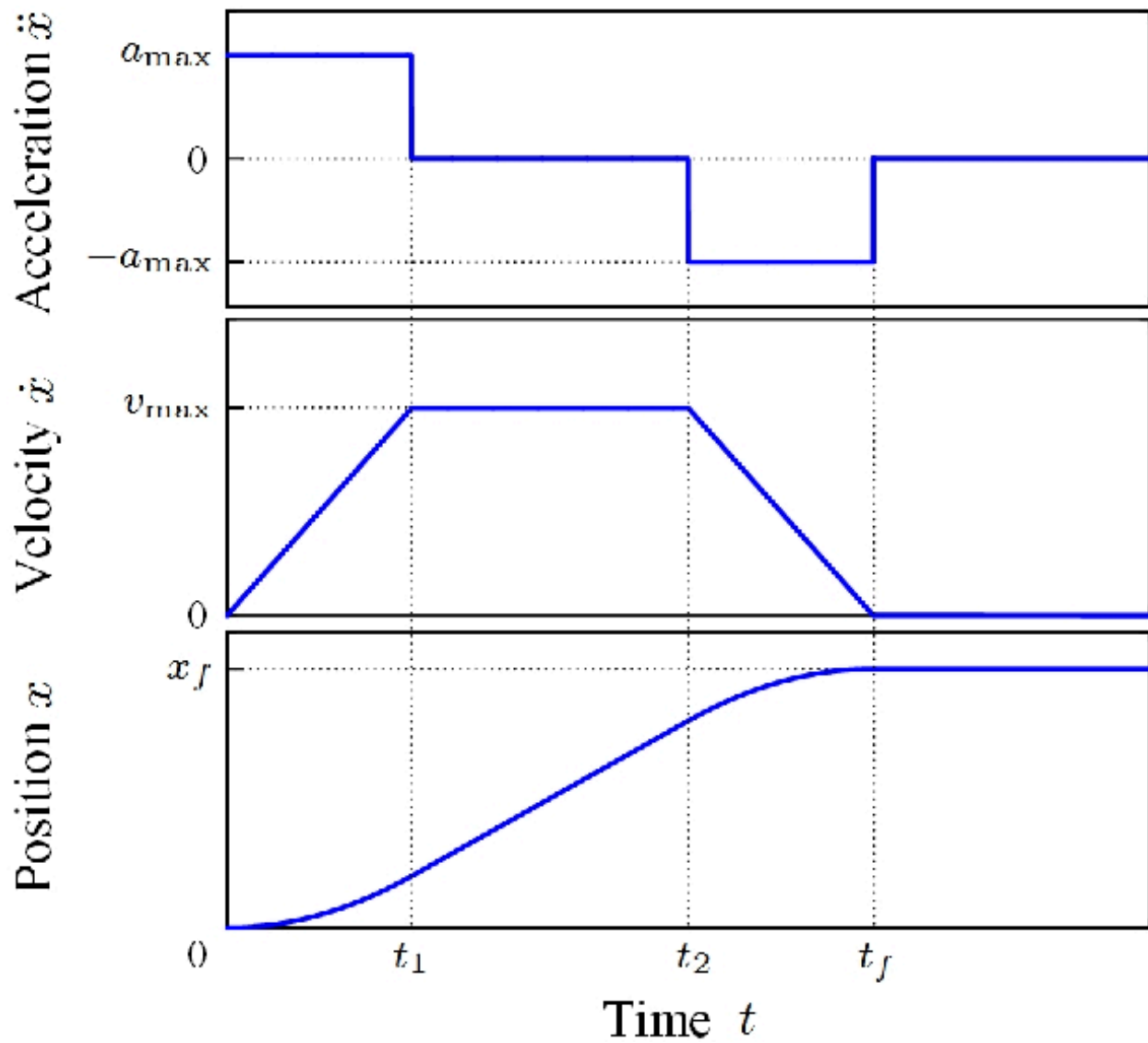


Fig. 114: These are the “magic functions” for velocity and acceleration over time alluded to in the feedforward section.

(continued from previous page)

```

if MAXIMUM_SPEED > abs(current_velocity):
    output_velocity = current_velocity + direction_multiplier * MAX_ACCELERATION_
↪ * (current_time - previous_time)
    output_acceleration = MAX_ACCELERATION

    #if maximum speed has been reached, stay there for now
else:
    outputVelocity = MAXIMUM_SPEED
    outputAcceleration = 0

    #if we are close enough to the object to begin slowing down
    if position_error <= (output_velocity * output_velocity) / (2 * MAX_
↪ ACCELERATION)):
        output_velocity = current_velocity - direction_multiplier * MAX_ACCELERATION_
↪ * (current_time - previous_time)
        output_acceleration = -MAX_ACCELERATION

previous_time = current_time

```

1.17 Appendix

1.17.1 Vendor List

- **Actuonix**⁵⁷
 - Actuonix sells linear actuators and linear motion components. Expensive, but robust. Teams can apply for a FIRST sponsorship.
- **AndyMark**⁵⁸
 - AndyMark sells the official game field and game sets, as well as individual game parts and the SoftTiles foam tiles.
 - AndyMark also sells NeveRest and NeveRest Sport motors, TileRunner, compliant, stealth, and mecanum wheels, as well as many other items.
- **goBILDA**⁵⁹
 - goBILDA sells its own build system, complete with Yellow Jacket motors, channel, motion components, and battery. Note that the cheapest batteries are found here, the MATRIX 12V Batteries are \$39.99, with the team discount they are \$29.99, nearly half the price of the \$50 batteries sold elsewhere.
 - Teams can get a 25% Team Discount from goBILDA.
- **McMaster-Carr**⁶⁰
 - McMaster-Carr sells hardware and raw materials in bulk quantities. They stock nearly every type of bolt, screw, and nut possible, as well as washers, bearings, springs, etc. Purchase from them for bulk quantities of hardware, as well as the times you need a very obscure part.

⁵⁷ <https://www.actuonix.com/>

⁵⁸ <https://www.andymark.com>

⁵⁹ <https://www.gobilda.com>

⁶⁰ <https://www.mcmaster.com>

- Don't be turned off by the hidden shipping. Generally, McMaster-Carr's shipping is around the same price as other vendors, and shipping is usually next-day.
- **MiSUMI**⁶¹
 - MiSUMI is a Japanese company specializing in industrial and manufacturing components. They sell bulk 15mm anodized extrusion similar to the REV Robotics extrusion. The 15mm extrusion can be cut to length as well.
 - MiSUMI also sells aluminum drawer slides that are popular for linear extensions. They are available in different lengths, but the most common is 400mm.
- **Pitsco**⁶²
 - Pitsco sells the Tetrix kit with channels, TorqueNado motors, and their own motion system.
- **REV Robotics**⁶³
 - REV Robotics sells the REV build system, which is an extrusion-based ecosystem complete with motors (HD Hex, HD Planetary, Core HEX), extrusion, servos (Smart Robot Servo), brackets, and battery.
 - REV also sells the control system for FTC (Expansion Hub and Control Hub).
 - REV offers various sensors (Magnetic Limit Switch, Color Sensor, Touch Sensor, Distance Sensor, Potentiometer, etc.)
 - Additionally, electronic components such as the Servo Power Module, SPARKmini, or Blinkin may be purchased.
- **Servocity/Actobotics**⁶⁴
 - Servocity sells the channel-based Actobotics build system with robust motion and structure. They also offer the Servoblock, a highly recommended part.
 - Servocity sells X-rail extrusion, lead screw actuator kits, and linear extension kits.
 - Servocity sells a wide range of servos, from Hitec to Futaba, at all price points.
 - Teams can get a 25% team discount from Servocity,
- **VEX/VEXPro**⁶⁵
 - VEXPro sells parts angled toward FRC use, but many of their parts, such as Thunderhex, can be made compatible with FTC robots.
 - VEX sells the 393 Motor, as well as the Motor Controller 29 in order for it to adapt to the Expansion Hub.
 - VEX offers the VersaPlanetary motor which allows teams to customize a specific gear ratio for their needs.
- **West Coast Products**⁶⁶
 - Also known as WCP, West Coast Products sells products exclusively aimed toward FRC use. However, much like VEXPro, many of their parts can be used in FTC, especially in custom robots. In fact, VEXPro and West Coast Products sell many of the same parts.

⁶¹ <https://us.misumi-ec.com/>⁶² <https://www.pitsco.com>⁶³ <https://www.revrobotics.com>⁶⁴ <https://www.servocity.com>⁶⁵ <https://www.vexrobotics.com/vexpro/ftc>⁶⁶ <https://www.wcproducts.net>

1.17.2 Vendor Guide

Vendor Identification

When you read some Bill of Materials (BOMs) or part numbers that are references in this guide, it might be confusing to find what vendor the part you need comes from. The obvious thing to do is Google the part number, and generally it can identify what you want and where it comes from. But if that fails, here are some quick tips to identify the vendor selling what you need.

Prefix Identification

- AM- signifies AndyMark (i.e. AM-0447)
- REV- signifies REV Robotics (i.e. REV-31-1155)
- WCP- signifies West Coast Products (i.e. WCP-0117)
- 217- signifies VEXpro (i.e. 217-6194)
- 276- signifies VEX EDR (i.e. 276-2193)

SKU Identification

- Actuonix: 3-4 digit SKU, alphanumeric, in the form of XX00 or X00 (i.e. PQ12)
- goBILDA: 12 digit SKU, numerical, in the form of 0000-0000-0000 (i.e. 3213-3606-0001)
- Pitsco/Tetrix: 6 digit SKU, W + 5 numbers, in the form of W00000 (i.e. W44260)
- Servocity/Actobotics: 6 digit SKU, numerical, in the form of 000000 (i.e. 615190)

Where can I get ...?

- **Motors and Gearboxes**
 - **Spur Gear Motors (not recommended)**
 - * AndyMark NeveRest Classic
 - * REV HD Hex & Core Hex
 - * goBILDA 5201 Spur Gear
 - * TETRIX TorqueNado
 - **Planetary Gearbox Motors (recommended)**
 - * AndyMark NeveRest Orbital
 - * REV HD Hex Planetary & *UltraPlanetary*
 - * goBILDA 5202 YellowJacket Planetary
 - * Vex *VersaPlanetary Lite* (gearbox only)
 - * (*italicized* = customizable gearbox ratio)
 - **Servos**
 - * HiTec (from ServoCity)

- * REV Smart Robot Servo
- * goBILDA Dual Mode Servo
- * Futaba (from ServoCity)
- * Savox heavy-duty servos
- * Actuonix (linear actuators)
- * VEX 393 EDR
- **Sensors**
 - * AndyMark
 - * REV
- **Hardware & Raw Materials**
 - * McMaster-Carr
 - * OnlineMetals
 - * Amazon
 - * Local hardware store
- **Aluminum Components**
 - * Actobotics channel, mini channel, X-rail extrusion
 - * REV extrusion, C channel, U channel
 - * goBILDA channel, goRAIL extrusion
 - * TETRIX channel
 - * MiSUMI extrusion
- **Gears/Sprockets/Pulleys**
 - * REV gears, sprockets, pulleys
 - * Actobotics gears, sprockets, pulleys
 - * goBILDA gears, sprockets, pulleys
 - * TETRIX gears and pulleys
 - * AndyMark gears and pulleys
 - * VEX gears, sprockets, pulleys
- **Wheels**
 - * AndyMark traction, compliant, mecanum wheel
 - * REV grip, traction, omni, mecanum wheel
 - * ServoCity traction, omni wheel
 - * goBILDA traction, omni, mecanum wheel
 - * VEX colson traction and mecanum wheel
 - * TETRIX traction, omni, mecanum wheel
 - * Nexus mecanum wheel
- **Linear Slide Kits**

- ★ REV 15 mm extrusion slide kit
- ★ Actobotics linear slide kit
- ★ goBILDA linear slide kit
- **Drawer Slides**
 - ★ MISUMI SAR2 or SAR3 aluminum slide
 - ★ REV UltraSlide
 - ★ Hafele cabinet slide
 - ★ Steel-rolled drawer slide
 - ★ Igus

1.17.3 Useful Resources

Here is a page dedicated to useful resources around the interwebs. Enjoy! **The links with an asterisk are resources that are especially pertinent or helpful. For further reading, we highly suggest you check out those links.**

- **This Guide!**
 - [Game Manual 0](#)⁶⁷
- **Official FIRSTInspires Resources**
 - [2019-2020 Game and Season Information](#)⁶⁸
 - [2019-2020 Game Manual Part 1](#)^{*69}
 - [2019-2020 Game Manual Part 2](#)^{*70}
 - [FTC Forum - USFIRST](#)⁷¹
 - [Robot Inspection Checklist](#)⁷²
 - [Field Inspection Checklist](#)⁷³
 - [DIY Field Perimeter Build Guide](#)⁷⁴
 - [Robot Building Resources](#)⁷⁵
 - [Robot Programming Resources](#)⁷⁶
 - [Team Management Resources \(Budget, Engineering Notebook, etc.\)](#)⁷⁷
 - [Mentor Manual](#)⁷⁸
- **General Team Resources**

⁶⁷ <http://gm0.copperforge.cc>

⁶⁸ <https://www.firstinspires.org/resource-library/ftc/game-and-season-info>

⁶⁹ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/game-manual-part-1.pdf

⁷⁰ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/game-manual-part-2.pdf

⁷¹ <https://ftcforum.usfirst.org/forum?s=2b3d68a255bd1bdd36e0831c9aa9078a>

⁷² https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/robot-inspection-checklist.pdf

⁷³ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/field-inspection-checklist.pdf

⁷⁴ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/low-cost-field-perimeter-guide.pdf

⁷⁵ <https://www.firstinspires.org/resource-library/ftc/robot-building-resources>

⁷⁶ <https://www.firstinspires.org/resource-library/ftc/technology-information-and-resources>

⁷⁷ <https://www.firstinspires.org/resource-library/ftc/team-management-resources>

⁷⁸ https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/mentor-manual.pdf

- KNO3 Software/Mechanical/CAD Resources⁷⁹
- Simbotics FRC Philosophy and Game Analysis^{*80}
- **Building Resources - Principles**
 - Simbotics FRC Mechanical Design Principles^{*81}
 - Simbotics FRC Drivetrain Design (designed for FRC)^{*82}
 - FTCKey - FTC wiki (slightly outdated)⁸³
- **Building Resources - Applications**
 - goBILDA with TETRIX (Blog for interfacing goBILDA with TETRIX)⁸⁴
 - **SDP-SI Timing Belt and Pulley Guide**⁸⁵
 - ★ SDP-SI Designing Belt Drives⁸⁶
 - KHK Guide to Gears⁸⁷
 - Gears Educational Systems Guide to Chain Drive Systems⁸⁸
 - 3D Printing Practical and Safety Course⁸⁹ - Tom Bladykas, FTC 3736
 - 8644 Brainstormers Tips and Tricks⁹⁰
- **Building Resources - CAD**
 - SOLIDWORKS Student Sponsorship for FIRST (SolidWorks Download)⁹¹
 - 10650 Hazmat Robotics CAD Tutorials⁹²
 - 10650 Hazmat Robotics CAD Library, Various Vendors⁹³
 - Actobotics (ServoCity) Official CAD Library⁹⁴
 - REV Robotics Official CAD Library⁹⁵
 - Guide to Rendering in CAD⁹⁶ - Justin, FTC 9656
- **Building Resources - Calculators**
 - JVN for FTC Design Calculator^{*97}
 - Sprocket Center-to-Center Calculator⁹⁸

⁷⁹ <https://www.kno3.net/resources>

⁸⁰ <http://www.simbotics.org/files/pdf/robotdesign.pdf>

⁸¹ <http://www.simbotics.org/files/pdf/mechanical.pdf>

⁸² <https://www.simbotics.org/files/pdf/drivetraindesign.pdf>

⁸³ <http://ftckey.com/>

⁸⁴ <https://gobildatetrix.blogspot.com>

⁸⁵ <https://www.sdp-si.com/PDFS/Technical-Section-Timing.pdf>

⁸⁶ <https://www.sdp-si.com/Belt-Drive/Designing-a-miniature-belt-drive.pdf>

⁸⁷ https://www.khkgears.co.jp/kr/gear_technology/pdf/gear_guide_060817.pdf

⁸⁸ http://gearseds.com/documentation/deb%20holmes/2.5_Chain_drive_systems.pdf

⁸⁹ <https://docs.google.com/presentation/d/1EmkYcIlHyltXlu7-TJMrwAawMWSspljUsFFP4Se32l8/edit?usp=sharing>

⁹⁰ https://www.youtube.com/playlist?list=PLoX10e-f5UglWtNA3mlb_SS0zS5w-eAIB

⁹¹ <https://app.smartsheet.com/b/form/6762f6652a04487ca9786fcb06b84cb5>

⁹² https://www.youtube.com/watch?v=NsFmFiC0D6g&list=PLQesWhH_pYWJhEFtDG39RZnApo4vaZh7c

⁹³ <https://workbench.grabcad.com/workbench/projects/gcpgZgLBwhldL0FfUKJJfM75cqa9RW1ncXaL-IQ4K0l1wa#/space/gcSzacmSel-119BYQNpm422pSHLenRxOxVtmaD-Pzynwsq/folder/6578524>

⁹⁴ <https://www.servocity.com/step-files>

⁹⁵ https://workbench.grabcad.com/workbench/projects/gcEvgrMnw6kRPx7OR6r45Gvb2t-iOdLiNG3m_ALpdGYzK_#/space/gcFd6nwp5Brrc3ks-92gagLZCV2FkceNTX3qGzaMvy2wQD/folder/2906404

⁹⁶ <https://drive.google.com/file/d/1t8Ke626MCedOHR4kzaNYtMdG7IC0bhGs/view>

⁹⁷ <https://www.chiefdelphi.com/uploads/default/original/3X/1/6/16e019399060799a45f54f4d75a8aa5fee1f394f.xlsx>

⁹⁸ <http://www.botlanta.org/converters/dale-calc/sprocket.html>

- Pulley Center-to-Center Calculator⁹⁹
- Pulley Center-to-Center Calculator, alternate version¹⁰⁰
- **Coding and Software Resources**
 - Java in 7 Days Tutorial¹⁰¹
 - Advanced Java Tutorial¹⁰²
 - Introduction to Control Systems¹⁰³
 - REV Robotics Expansion Hub Guide¹⁰⁴
- **Vendor Resources**
 - REV Robotics Technical Resources¹⁰⁵
- **Discord**
 - FTC Discord¹⁰⁶
 - The FIRST Tech Challenge Discord server is a discussion-based community server that has teams ranging from first-year rookies to Winning Alliance Captains at the World Championships. It also has a channel for direct access to vendors.

1.17.4 License

Game Manual 0 is licensed under CC BY-NC 2.0.

CC BY-NC 2.0 License Text

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONSPUBLIC LICENSE ("CCPL" OR "LICENSE").
THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW.
ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.
BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
"Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement,

(continues on next page)

⁹⁹ https://www.engineersedge.com/calculators/Pulley_Center_Distance/toothed_pulley_center_distance_calculator_12900.htm

¹⁰⁰ <https://www.sudenga.com/practical-applications/figuring-belt-lengths-and-distance-between-pulleys>

¹⁰¹ <https://www.guru99.com/java-tutorial.html>

¹⁰² <http://enos.itcollege.ee/~jpoial/allalaadimised/reading/Advanced-java.pdf>

¹⁰³ <https://blog.wesleyac.com/posts/intro-to-control-part-zero-whats-this>

¹⁰⁴ <http://www.revrobotics.com/content/docs/REV-31-1153-GS.pdf>

¹⁰⁵ <http://www.revrobotics.com/resources/>

¹⁰⁶ <https://discordapp.com/invite/first-tech-challenge>

(continued from previous page)

dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License.

For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.

"Licensor" means the individual or entity that offers the Work under the terms of this License.

"Original Author" means the individual or entity who created the Work.

"Work" means the copyrightable work of authorship offered under the terms of this License.

"You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
- to create and reproduce Derivative Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
- to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats.

All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(d) and 4(e).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder.

You may not sublicense the Work.

You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner

(continues on next page)

(continued from previous page)

inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any reference to such Licensor or the Original Author, as requested.

You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation.

The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work

(e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author").

Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

For the avoidance of doubt, where the Work is a musical composition:

Performance Royalties Under Blanket Licenses.

Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

Mechanical Rights and Statutory Royalties.

Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.

Webcasting Rights and Statutory Royalties.

For the avoidance of doubt, where the Work is a sound recording, Licensor

(continues on next page)

(continued from previous page)

reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions),
if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability.

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses.

Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with

(continues on next page)

(continued from previous page)

respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensors shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.