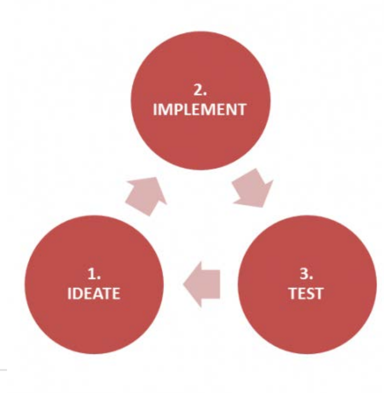
Unit 1: Introduction to Engineering

This unit introduces students to what engineering is.

Students will learn what the engineers do. They will also learn what the different types of engineering are and the specific tools used by engineers during their work. This unit will show students how to begin and use and engineering notebook. This notebook will be used to document their progress throughout the semester.



Curriculum Lesson Content up 1.1: Introduction

1.1: Introduction

When asked “Do you know what engineering is?” most people will emphatically reply that they do; however, when they’re then asked what does an engineer do? What is something you have used today that has been engineered?”, they struggle.

WHAT IS ENGINEERING?

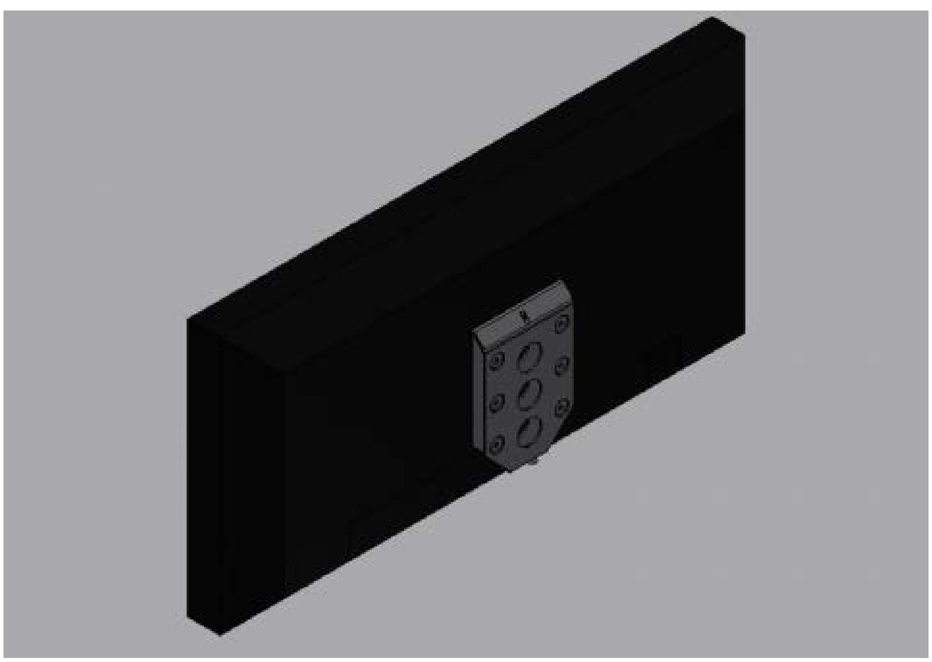
Engineering can be defined as the application of practical and scientific knowledge to the solving of a problem through the use of a methodical process.

More simply put, at its most fundamental level engineering is problem solving.

The solutions to these problems can take many forms. Maybe the engineer is creating a physical thing to solve a problem, maybe they are creating or improving a process for doing something, or maybe they are determining why something happened the way it did. The many different types of problems are all related to engineering, especially if they are solved using a methodical process (more on this later.)

The above definition mentions that engineers apply practical and scientific knowledge to their problem solving. Some would say that engineers take scientific knowledge and find a practical application for it to benefit humanity. To be successful, engineers must become familiar with a wide variety of fields, and experts in some specific fields.

As an example, one could look at this product design problem:



In order to design the above bracket to mount a TV to a wall, an engineer would need a wide variety of background knowledge. They might need to draw on knowledge of classical mechanics to determine the forces the TV would exert on the bracket. They would need knowledge of structural design to figure out a bracket capable of withstanding the forces. The engineer would need background in the manufacturing processes that are required to make the bracket, and design it so it can actually be produced. They would need some background on the types of surfaces it would mount to, and the screws & nuts that would be used to mount it so they can determine what screws to use and how many screws are required to keep the bracket from falling off the wall. The engineer would need some basic idea of how the bracket would be installed by the final customer and create a good user experience, as well as knowledge of how the bracket would be used after it is installed (to be a good product, does it need to tilt up and down? Does it need to tilt left and right?). The engineer would also need some knowledge of TVs; to be a good product, the bracket will hopefully be able to mount a wide variety of television sets – are there any standards TV manufacturers design for which would allow for universal mounting?

This knowledge is all necessary to engineer and design the best possible solution. The example above includes a variety of the different types of scientific knowledge. It also has quite a bit of practical knowledge. So where do engineers get all this knowledge? Much of it would be learned through higher education – to become an engineer, one must complete a degree program from an accredited engineering college in the field they which to practice in. The rest of the knowledge would be learned through training, or on-the-job experience. For example, a young engineer may be taught about types of fastening hardware (nuts & bolts) from a more experienced engineer during their first few years of work.

WHAT IS DESIGN?

The term “design” was listed above, but what exactly does this mean? When it is said “engineers design a bracket” what is being described?

Design is defined in the dictionary as follows:

- To conceive of fashion in the mind; invent
- To formulate a plan for; devise
- To plan out in a systematic, usually graphic form
- To create or contrive for a particular purpose or effect
- To create or execute in an artistic or highly skilled manner

A simpler definition might be*: Design is thinking of and creating something new, or adapting something old to solve a problem and/or satisfy a need*. One should note that this definition has the key words “problem solving” again.

DISCIPLINES OF ENGINEERING:

There are many different types of engineers, each specializing in a different field of knowledge, each with a specific set of problems they specialize in solving. There are almost as many fields of engineering as there are fields of scientific inquiry! Some examples are listed below. Keep in mind that this list is not all-inclusive.

- Acoustical Engineering
- Aeronautical Engineering
- Aerospace Engineering
- Agricultural Engineering
- Architectural Engineering
- Automotive Engineering
- Biological Engineering
- Biomechanical Engineering
- Biomolecular Engineering
- Ceramic Engineering
- Chemical Engineering
- Civil Engineering

- Computer Engineering
- Control Engineering
- Electrical Engineering
- Electronic Engineering
- Energy Engineering
- Environmental Engineering
- Heating, Venting, Refrigerating & Air-Conditioning Engineering
- Industrial Engineering
- Manufacturing Engineering
- Materials Engineering
- Mechanical Engineering
- Mechatronics
- Metallurgical Engineering
- Mining Engineering
- Molecular Engineering
- Nano Engineering
- Naval / Ocean / Marine Engineering
- Nuclear Engineering
- Optical Engineering
- Paper Engineering
- Petroleum Engineering
- Plastics Engineering
- Power Engineering
- Process Engineering
- Structural Engineering
- Systems Engineering
- Thermal Engineering
- Transportation Engineering

This curriculum will discuss a few concepts which are relevant to ALL of these types of engineering, and touch on others more specifically related to the field of robotics including Robotics Engineering, Mechanical Engineering, Electrical Engineering, Computer Engineering, and others.

So what happens if the problem being solved is so large and complicated that it involves more scientific and practical knowledge than any one engineer can understand? This happens all the time! This is why engineers commonly work together as part of a Design Team.

Unit 1: Introduction to Engineering | up | 1.2: Engineering/Design Teams

1.2: Engineering / Design Teams

Most problems require more than one engineer to solve. Take for instance the design of an automobile.

This is a hugely complex system that includes thousands of smaller problems that need to be solved. Some examples of the engineering disciplines required can be seen in the list below.

Acoustical Engineers might work on minimizing road noise within the car, or improve the design of the car's stereo and speaker system, or even work to improve the sound of the cars engine.

Aeronautical Engineers would be involved in improving the aerodynamics of the car to reduce drag and maximize gas mileage.

Automotive Engineers are a specialized type of engineers who utilize the skills of many of the other branches of engineering listed here, and would be involved in most aspects of the car design.

Ceramic Engineers work with inorganic, non-metal materials, and might develop special ceramic composites (combinations of multiple materials) for use in heat shielding, or bearings. Some high-end cars use specially engineered ceramic brakes.

Computer Engineers would be involved in creating the firmware (software embedded in the car's microchips) of the car.

Control Engineers, Electrical Engineers, and Electronic Engineers would work on designing & integrating the car's electrical system, software, and sensors.

Environmental Engineers would be involved in making sure the car meets all emissions requirements.

Heating, Venting, Refrigerating & Air-Conditioning Engineers might be involved in creating the car's heat and air conditioning systems.

Systems Engineers & Industrial Engineers would be involved in the management and supervision of the car creation process.

Manufacturing Engineers would determine how to make the individual components of the car.

Materials Engineers would help create new materials for use in the car construction.

Mechanical Engineers would work on the design of the mechanical aspects of the car; anything from the transmission to the engine to the suspension to the design of the snaps that hold the seats onto the frame.

Optical Engineers work on lenses and other optical instruments. They would design the car's mirrors and windows.

Plastics Engineers would create plastic types for use in the car's construction.

Process Engineers would be required to determine the best way to make the car and to ensure it is built correctly.

Structural Engineers might be involved in the creation of the car's chassis and frame.

Thermal Engineers would work on the complex heat transfer systems, such as the engine cooling and exhaust.

That is only part of the huge team required! Diversity of knowledge isn't the only reason for engineering teams. Even if there were one engineer out there who possessed all the knowledge listed above, more than one engineer would still need to be used. All individuals are shaped by their backgrounds and their experiences; these experiences serve to create their viewpoints. Multiple viewpoints are hugely beneficial when trying to find creative solutions to complex problems, as expressed by the cliché "two heads are better than one." The more viewpoints looking at a problem, the more likely the problem will be solved. Of course the most obvious reason one would need multiple engineers working to design a car is that designing a car is a LOT of work for one person; the workload needs to be spread out among multiple people to get it done in a timely fashion!

WORKING ON A DESIGN TEAM:

Every student involved in competition robotics will have the opportunity to work on a design team at some point. There are a number of considerations they should keep in mind to achieve success:

- One should always keep an open mind. It is important to allow crazy ideas to develop. The most likely time for a creative solution to be found is early in the design process when wild ideas are expressed.
- No one should become overly attached to any single idea - especially one they created. It is easy to become blinded to other ideas simply because "they aren't mine."
- One should not become defensive regarding the opinions of others. Defend one's own opinions and ideas but always focus on the ultimate goal of providing the best solution possible.
- One should always stay positive, even when discussing negatives.
- Engineering is based in logic. One should focus on factual arguments, not those based on opinions. Emotion should not be allowed to interfere with the process.
- It is important not to be offended if disagreements occur, even if things get heated and criticisms are overly harsh. Most engineers get passionate during design discussions and will often be very blunt. It is important not to take this personally.
- An unjustified opinion is not useful. Team members must be able to describe WHY they like or dislike something.
- *This is NOT rhetoric, it is engineering.* In rhetoric, the person who argues best will be most persuasive. In engineering, the person who has the best argument will be most persuasive. It is not the one who can speak the best but the one who can provide

quantitative proof that will win an argument and prove their idea is better! It is important to be quantitative wherever possible.

The term "quantitative" is used a lot when discussing engineering arguments or justifications, but what does this really mean?

quan·ti·ta·tive (adj.)

- Expressed or expressible as a quantity.
- Of, relating to, or susceptible of measurement.
- Of, or relating, to number or quantity.

Quantitative arguments are simply ones that can be measured! In a design discussion, these are extremely valuable. As stated above, it is important to be quantitative whenever possible. For instance, the statement "that option is heavier, so I don't think the extra functionality it provides is worth it," is not nearly as valuable as saying, "That option weighs 50% more based on my initial estimates. Do we want to accept this additional weight for the functionality it provides?"

1.1: Introduction

up

1.3: What is the Engineering Design Process?

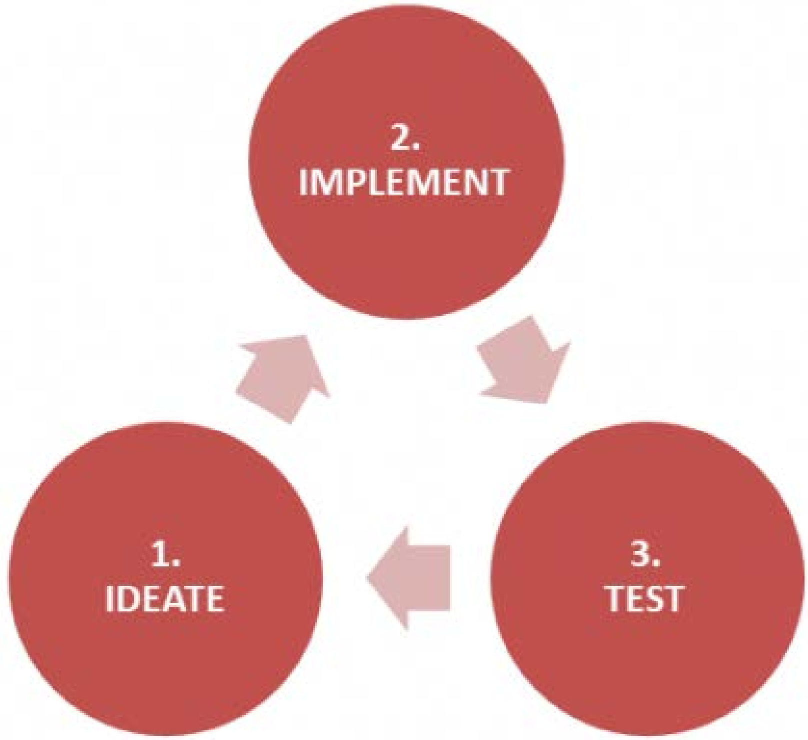
1.3: What is the Engineering Design Process?

Engineering was defined at the beginning of this unit as the application of practical and scientific knowledge to the solving of a problem through the use of a methodical process. Discussed above were some different types of knowledge that engineers apply to solving a problem, but the process itself has not been mentioned. What methodical process do engineers use to solve problems?

The engineering design process is a series of steps that engineers follow when they are trying to solve a problem and design a solution for something; it is a methodical approach to problem solving. This is similar to the “Scientific Method” which is taught to young scientists. There is no single universally accepted design process. It seems as though most engineers have their own twist for how the process works. The process generally starts with a problem and ends with a solution, but the middle steps can vary.

One can think of the engineering design process as a recipe for banana bread; it can be made a lot of different ways but it’s usually going to start with bananas and it’s going to end with a loaf of bread. One such “recipe” for the engineering design process will be outlined in this unit; this is not the only correct version of the process, it is just one example. It will provide a good starting point for students to explore the engineering process.

The design process in its simplest terms can be seen as a 3-step loop:



In this simple design loop an idea is generated (1). This idea is implemented (2). After the idea is implemented, the design group would test the product or evaluate the result of the implementation through testing (3). Typically, during this testing and evaluation, additional ideas are generated, and the process starts over again. This cycle and repetition is why it can be said that design is an iterative process.

Iteration is the act of repeating something over and over again in order to improve the process and eventually achieve a desired goal.

Obviously this process could go on forever (or until the design group stops thinking of new ideas and stops finding problems with the design). There is a saying sometimes used by veteran engineers: "At some point in every design process someone needs to get rid of the engineer and just build the thing!"

USING THE ENGINEERING DESIGN PROCESS:

As discussed above, there is no single engineering design process. Throughout this course students will use an 11-step design process as they conceptualize, design, and create a robot to compete in head-to-head robotics competition. The process used is seen below.

Step 1 – UNDERSTAND – Define the Problem

Step 2 – EXPLORE – Do Background Research

Step 3 – DEFINE – Determine Solution Specifications

Step 4 – IDEATE – Generate Concept Solutions

Step 5 – PROTOTYPE – Learn How Your Concepts Work

Step 6 – CHOOSE – Determine a Final Concept

Step 7 – REFINE – Do Detailed Design

Step 8 – PRESENT – Get Feedback & Approval

Step 9 – IMPLEMENT – Implement the Detailed Solution

Step 10 – TEST – Does the Solution Work?

Step 11 – ITERATE

STEP 1: UNDERSTAND

In this step engineers will define the problem they are trying to solve.

This is the single most important step in the design process. Without fully understanding the problem how can an engineer solve it successfully? This step is often done incorrectly or incompletely and results in a failure of the design. It is important to define the true problem one is solving, not just the symptoms of the problem or the perceived problem.

When trying to define the real problem, remember the elevator riddle, as follows.

There is a story about a skyscraper which is told to young engineers to emphasize the importance of this step in the design process. The story goes that there was a skyscraper in a major city and the occupants of the building were complaining that the elevator ride times were too long. The owners of the building wanted to fix this, so they put out a call to several local engineering firms asking them for proposals.

1. One firm put in a bid to renovate the office and add two additional elevators. They speculated that adding more elevators would cut down on elevator stops and decrease the average ride time. They estimated this would cost some ludicrous amount of money (details vary based on the telling.)

2. Another engineering firm suggested renovating the building and adding some brand new, state of the art, high-speed elevators. These faster elevators would also reduce ride time. This suggestion didn't cost as much as the first proposal, but was still a ridiculous amount of money.

3. The third engineering firm came back with a proposal to upgrade the elevator software. They claimed that they had devised a new algorithm that would more effectively utilize the elevators already in place to cut down on average ride time. This proposal was still somewhat expensive, but much cheaper than the other two.

The owners of the building were just about to hire the third firm when a fourth proposal was presented. After detailed review, the fourth proposal was immediately implemented. The fourth engineering firm suggested that full-length mirrors be installed in every elevator. When the building residents were in front of a mirror, they fidgeted and adjusted their ties, checked their make-up, and so forth and didn't notice the length of the elevator ride. This proposal didn't cost the owners very much at all and was dubbed a great success. The fourth company understood that the real problem wasn't that the elevators were too slow, but that the residents thought the ride times were too long.

UNDERSTAND for Competition Robotics

In competitive robotics there are typically numerous problems that need to be solved by the design team. The further designers get in their robot design, the more problems come up (the main problem is often broken down into smaller problems). Early in the robot design the problems may be more "big picture" and later they will become more "detail oriented."

Some sample problems a designer may encounter that need to be solved, and questions that need to be answered are below.

What is the most effective strategy for playing the game? How do we win matches?

How can the robot score the most points during the match? How do we score more than our opponents?

How fast does the robot need to move?

How can the robot pick up the game object?

How can the robot pick it up quickly?

How many game objects does the robot need to hold?

These problems and questions all have many answers; some answers are better than others. How does a designer go about finding the "correct" solution or the "correct" answer? That is where the rest of the process comes into play, but until the correct problem is defined it can never be solved!

STEP 2: EXPLORE

In this step engineers will do background research on the problem their solving. They will investigate the ways others have tackled similar problems. Engineers will also gather details on the environment they're dealing with, the situations their solution will be used in, and the ways it will be used.

EXPLORE for Competition Robotics

Students involved in competition robotics will also need to explore their challenge. They should investigate challenges from the real world similar to the one they are solving. Students can also look to see if any other robotics competitions have utilized similar challenges in the past. This section is all about gathering data from other sources to help student robot designers create a successful solution.

STEP 3: DEFINE

In this step engineers will specify WHAT the solution will accomplish, without describing HOW it will do it. They do this through the use of specifications.

What are specifications? A specification is defined as an explicit set of requirements to be satisfied by a material, product, or service. In this case, specifications are requirements for the solution of the problem defined in Step 1 of the design process.

Specifications typically come from two places:

1. Design Constraints

2. Functional Requirements

What are constraints? A constraint can be defined as a condition that a solution to a problem must satisfy. Constraints, in short, are restrictions. What are functional requirements? Functional requirements describe how well the finished solution must perform.

Again, specifications outline WHAT the solution will do and how WELL it will do it, not HOW it will do it. In Competitive Robotics the specs would describe WHAT the robot does, not HOW it does it. Thinking too much about "how" at this stage in the process can be counter-productive and may stifle creativity. At the same time, designers need to keep the "how" in the back of their minds because they need to have a basic understanding of what is possible. (For example, specifying that a new laptop computer will run continuously for one year off a single AA battery is not reasonable.)

DEFINE for Competition Robotics

In competitive robotics, designers are presented with some challenge or game in which their robot will compete. This challenge often includes a manual containing a series of restrictions and requirements that every robot must fulfill; these are design constraints. This is the first type of specification a designer encounters during the process. Some examples of this type of spec are “maximum robot starting size” and “maximum number of motors allowed.”

Some specifications are also due to the resources available to the designer. Since the first set listed above are present in the competition rules they are apparent to all designers. This second set of restrictions is not always as obvious but it is equally as important to consider during the design process. Some of these may be self-imposed design constraint type specifications. Two examples are “robot must fall within designer’s budget” and “robot must utilize parts the designer already has.”

Another self-imposed design constraint revolves around the team’s capabilities.

One of the most important parts of successfully generating design constraints in competition robotics is to understand one’s limitations. Many teams are tempted to overstretch their capabilities by exciting designs.

Every team needs to understand exactly what they are capable of so they don’t end up missing a target. Capabilities often depend on manpower, resources, budget, experience, and more. It is important to focus on the big picture when determining whether a design is achievable. When divided up, each piece may seem doable while the overall system is “too much.”

Teams will often be more successful by choosing a simple design and executing it very well than by choosing a complex design that they are not capable of executing! For instance, consider that two teams are trying to build robots to put a soccer ball in a soccer goal. One team decides to build a simple plow to push the ball into the goal. The other team tries to build a kicking mechanism to kick the ball into the goal. The kicking mechanism may seem like the better solution, but what if the second team can’t actually finish building their kicker? In this case, the simple solution would win! The second team should have considered their ability to complete the project before deciding.

The next group of specifications comes from the designer’s functional requirements for the robot. These are things the designer believes the robot should be capable of and are performance based. Many of these are related to the challenge placed before the designer (i.e., robot can hold 10 game objects, robot can lift game objects one meter, etc.)

It may be difficult or even impossible to generate this third type of spec early in the design process, as most of them are dependent on the nature of the design and how it progresses. These are more common during some of the sub-design processes than for the overall system.

Specifications Ranking

All specs are not created equal, some are more important to the design than others. Designers need to think about what is most important, and why. Specifications are often ranked in some way to denote their importance. One such scale is:

W = Wish (not that important, but it would be nice if it is possible)

P = Preferred (important, but the project won’t fail without it)

D = Demand (critical to the project, MUST be included)

With these, a designer would go through and rank the specifications. These provide a good “check” for the designer at the end of the project. It is easy to go back down the list of specifications and see how well the design fulfilled them.

Designers must make decisions about what is most important when they apply these rankings. Ranking the specifications in this way will also make it clearer in the designer’s mind what to focus on. Some rankings are easier than others, for instance the constraints

REQUIRED by the design challenge itself are obviously ranked as "Demand."

When creating specifications some designers will list several similar specifications at different rankings to show varying degrees of importance. An example of this can be seen below:

Robot can hold 5 game objects – Demand.

Robot can hold 10 game objects – Preferred.

Robot can hold 15 game objects – Wish.

In this example the specifications make it clear that the robot MUST hold 5 game objects, if possible it should hold 10, and the designer would be very happy if it held 15 game objects. Through the use of good specifications and ranking it is possible to outline exactly what requirements the design team should follow and what goals the design team should strive to meet.

STEP 4: IDEATE

Ideate means to formulate, imagine, or conceive of an idea.

Now that the engineer knows WHAT the solution will do, he or she must determine HOW it will do it.

Two words: "Napkin Sketches." This phrase refers to the habit of jotting down ideas whenever and where ever they occur - even if you have to jot them down on a napkin.

Everyone does the same thing when faced with a problem or a decision to make: they think of alternative courses of action, even if they do this subconsciously. Formally documenting this intuitive action may help when solving complex engineering problems.

This is a step that requires some creativity. Some of the questions most commonly asked of engineers are, "How did you come up with that?" and "Where do you get your ideas?" Ideas come from everywhere! Inspiration can come from anywhere!

The keywords here are: "imagination" and "think." This is where the designer needs to brainstorm multiple ways to fulfill the specifications. It is important to remember to look for inspiration everywhere. A common mantra is, "*steal from the best, then invent the rest.*" Good designers will look in the world around them and try to find solutions to adapt to their problem and build off of. Innovation is also important early in the design process (don't wait to innovate, always put innovation first); there is a good balance to be found between "thinking outside the box" and "using pre-made designs."

Often combining two ideas or compromising between two different suggestions may yield a good concept. Again, improvements and innovations early in the process will yield better results later in the process.

It is important not to settle for mediocre concepts and to strive to find the "right" solution. Often this "right" solution reveals itself. Designers will often comment, "It just feels right." The "right" solution will just seem elegant. Unfortunately it is not always this easy, and elegance is not always so apparent.

Engineers should record ALL ideas in their engineering notebook!

(It is important for engineers to copy their napkin sketches into their engineering notebooks so they have an organized record of their thought process and ideas.)

IDEATE for Competition Robotics

In competition robotics there are a number of concepts that need to be generated. Teams need to generate concept strategies, concepts for the overall system, and concepts for individual subsystems and mechanisms. Some of these systems will be dependent on and influence each other. The team's strategy will affect the overall system design, which in turn affects the different subsystems, but each of the subsystems will also affect the overall system.

These concepts are typically generated in brainstorming sessions involving the whole competition team. Concepts are recorded as diagrams, sketches, and descriptions into individual team member's engineering notebooks.

Brainstorming – Group Creativity Technique

This stage in the engineering design process requires great creativity and the generation of a number of options for the problem's solution. To accomplish this, one must use an engineering tool known as BRAINSTORMING. Brainstorming is an exercise in which groups of individuals work together to generate large numbers of ideas.

Some important rules for brainstorming:

1. When brainstorming, teams focus on the quantity of ideas generated, not the quality. The premise is that from lots of ideas will come a few great ones!

2. Reserve judgement. There are no bad ideas during the brainstorming session, because even the most outlandish concept could inspire someone else to come up with something great. Crazy ideas may also be improved and developed during the collaborative process and become feasible ideas.

3. Record everything. Student designers should document all the ideas generated during brainstorming in their engineering notebooks.

STEP 5: PROTOTYPE

In this stage of the process engineers takes some of their concepts from the previous step and make mock-up versions of them. The goal of this stage is to learn how each concept solution will function in "real life" and how it interacts with the real environment. This is also where a designer will start to determine which design concept will work the best. These prototypes are designed to be crude, but functional enough to be educational to the designer. The keyword here is "LEARN."

Designers don't need to prototype everything, just the things they want to work!

PROTOTYPE for Competition Robotics

In competitive robotics, the robots must often interact with their environment and designers must learn the nature of these interactions to be successful. Designers should test in "real world conditions" to see how things interact, and find places for improvement EARLY in the design. The VEX Robotics Design System is perfectly suited to the kinds of quick prototypes designers need to perform in competition robotics. It is easy to quickly build something up, test to see how it works, and tweak it – all without any manufacturing capabilities.

Teams should be very meticulous during their prototyping. Students should take detailed notes in their engineering notebooks during the prototyping process, recording what they see, trying to figure out why some things work better than others, and then creating additional prototypes to test these ideas. Gathering data is an important part of prototyping.

STEP 6: CHOOSE

At this point in the process the designer or design group has several different potential solutions for the problem. This step is where the designers will use the lessons learned from their prototyping to determine which concept is best and go forward with it. This is not always an easy decision. Sometimes the "right" solution just reveals itself. Other times it is difficult to even define "best." Teams can compare how each concept fulfills the specifications from step three in the process and see if one is significantly better than the others. Designers should look for the simple and elegant solution.

In the event that there is no obvious solution, a more methodical approach must be used to make the decision.

When choosing concepts as a design group, it is tempting to rely on a vote. However, a vote is nothing but an unjustified opinion, and an unjustified opinion isn't worth much in an engineering discussion. When it comes to design decisions it is better to talk through things and make a logical decision by building consensus. As discussed previously, it is important to be as quantitative as possible; one shouldn't just say something is "better," they should say it is "14.8% lighter" and then prove why that makes it better.

In some cases the decision-making is not made by the whole design group, but by a smaller leadership group or even by a single leader. In this situation the leadership is responsible for impartially comparing each of the alternatives and then choosing the course of action. This method does not always work well, especially if the rest of the design group does not recognize the authority of the leadership and questions the final decision. However, this method can be useful in preventing stalemate situations where no consensus can be reached. To help get the group's approval, some leaders will try to use a form of consensus building, leading up to the final decision.

CHOOSE for Competition Robotics

The challenges faced by student designers in competition robotics are almost identical to the ones faced by real engineers. Students must work together with their teammates to figure out which concept best fulfills the design specifications, which really means, "Which concept works best to solve the problem?" Students should rely on prototypes to help them make this determination. It is important for students to remember to use quantitative arguments to show how one option is better than another; the easiest way to do this may be to prove it with a prototype.

Decision Making Tool: Weighted Objectives Tables

One tool used to help during the concept selection stage of the design process is the weighted objectives table (WOT), sometimes referred to as a decision matrix. The weighted objectives table can be used to help designers choose between options based on how they are ranked on several criteria. The WOT is an especially effective tool because of how it helps a designer compare alternatives based on what is most important to the final solution.

Subsystem 1 - Drivetrain Comparision

Comparison Criteria	Weight	Slide Drive Score	Slide Drive Weighted Score	Swerve Drive Score	Swerve Drive Weighted Score	Skid Steer Score	Skid Steer Weighted Score
Cost	5	6	30	3	15	9	45
Maneuverability	15	10	150	10	150	4	60
Weight	10	5	50	2	20	8	80
Motor Usage	20	5	100	2	40	8	160
TOTAL:	50		330		225		345

Subsystem 2 - Gripper Comparision

Comparison Criteria	Weight	Pinchy Claw Score	Pinchy Claw Weighted Score	Roller Claw Score	Roller Claw Weighted Score	Scoop Score	Scoop Weighted Score
Cost	5	5	25	5	25	9	45
Speed of Grab	20	7	140	9	180	2	40
Grip Strength	15	10	150	10	150	1	15
Weight	10	6	60	4	40	8	80
TOTAL:	50		375		395		180

Subsystem 3 - Lift Comparision

Comparison Criteria	Weight	Elevator Score	Elevator Weighted Score	2-Jointed Arm Score	2-Jointed Arm Weighted Score	Linkage Arm Score	Linkage Arm Weighted Score
Cost	5	3	15	7	35	9	45
Difficulty	20	2	40	7	140	9	180
Vertical Distance	15	8	120	7	105	4	60
Packaging	10	9	90	8	80	3	30
TOTAL:	50		265		360		315

For more information on using a weighted objectives table to choose between concepts, refer to Appendix 7 – Using a Weighted Objectives Table.

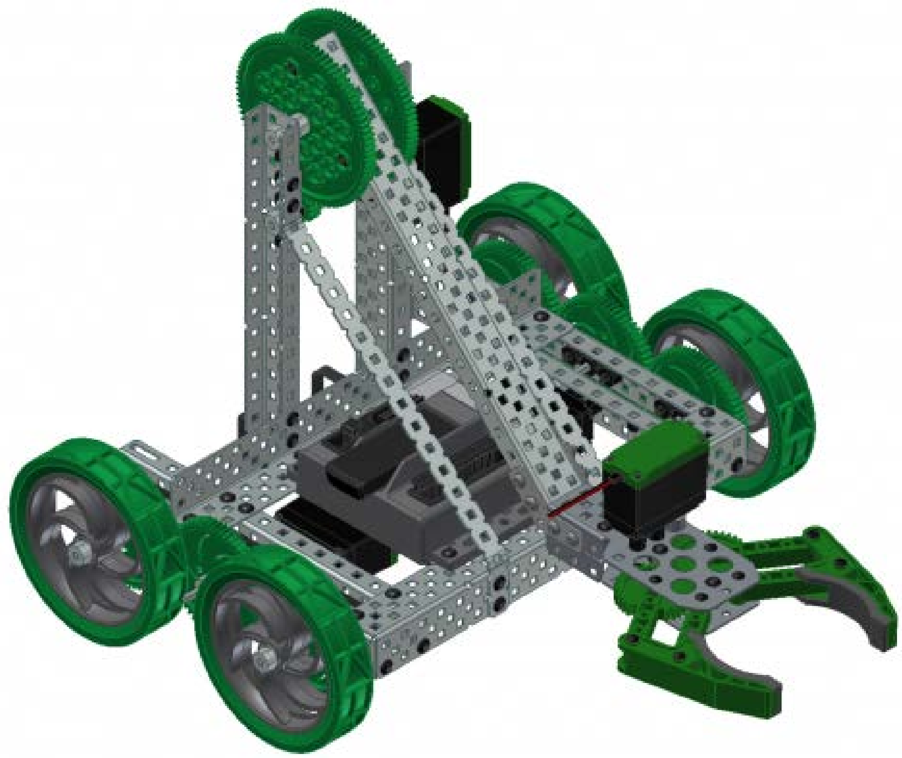
STEP 7: REFINE

This is the stage of the design process where engineers take their chosen concept and make it into something more "real." This stage is all about the details. At the end of this stage design teams should have everything necessary so that the full design can be constructed or implemented. Some of the pieces that may be generated during this step are CAD Models, Assembly Drawings, Manufacturing Plans, Bill of Materials, Maintenance Guides, User Manuals, Design Presentations, Proposals and more.

These designs will start off very basic and evolve as more details are added. It is not practical to start by detailing every piece of the solution until one sees how the pieces fit together. These basic pieces are then refined into more detailed pieces that are part of the final design.

REFINE for Competition Robotics

In competitive robotics it is a good idea to make a 3D model of the entire robot in Autodesk Inventor. This can be one of the longest stages in the design process, but the work pays off. The more work put into the design of the robot, the easier it will be to make it.



The devil is always in the details of the design.

By spending time CADing all the detail, any issues will be solved before they become serious problems.

This step in the process is also when design calculations are completed. These calculations can refer to optimizations of gearing, material strength, weight, cost, and more. In an abbreviated design process it is not always possible to fully optimize all aspects of the design. Depending on the project, sometimes getting it "close enough" is all that is needed. Many designers can plan things using simply their prior experiences and intuition rather than calculating every detail; this kind of thinking may work fine for a high school robotics project, but wouldn't be acceptable when designing pieces of a spacesuit where optimization is important. Rather than optimize each piece, just ensure it can do its job. It is okay to "over build" so long as none of the specifications are violated.

STEP 8: PRESENT

The detailed design must often go through some sort of [design review] or approval process before it can be implemented. A design review can come in many forms. Some reviews occur as a simple conversation between two of the designers. Some reviews are done as a meeting of the Design Group where they recap and check the work that has been completed and try to find any errors. Many reviews involve presenting the detailed design to a customer, manager, or some other decision-maker for final approval.

PRESENT for Competition Robotics

In competition robotics, the robot designer or design group needs to present the final robot design to the rest of the team, to their class at school, or to the team leadership for final approval. Sometimes a team will do a very formal design review meeting and invite sponsors, school administrators and community members to participate.

Design Presentations are an important part of the engineering process. Many engineers believe that language arts type classes are not important to them, and that they have an excuse for poor spelling, bad grammar or poor communication skills. This could not be further from the truth. If an engineer has an idea, but cannot communicate it effectively, it is not helpful; if an engineer has an opinion but cannot express it, it is not going to help solve the problem. The ability to summarize, present, and defend ideas is a skill that is absolutely critical! This applies to verbal communication, written reports, slideshow presentations, engineering drawings, and other types of media.

The goal of a design review is not simply to approve the design; it is also to find any problems with the design or potential places where the design can be improved. During the design process, several alternative concepts were generated and one was chosen. There are many such choices made during the design process. Justifying these choices is one of the key parts of the design presentation. "*WHY did you do it like THAT instead of like THIS?*" The review group needs to ensure that the designers have done "due diligence", meaning they need to see that alternatives were investigated. They need to verify that the design is well thought out and is not just the first thing that popped into someone's head.

Designers should take detailed notes at Design Reviews in their engineering notebooks, including a list of action items to be accomplished later.

Common questions from a Design Review:

- Why was it done this way?
- Did you think of doing it a different way?
- Why did you rule out other alternatives?
- Does it fulfill our requirements and specs?
- How can we make it function better?
- How can we make it weigh less?
- How can we make it faster?
- How can we make it more robust?
- How can we make it smaller?
- How can we make it simpler?
- How can we make it more efficient?
- How can we make this cheaper?
- How can we make this easier to construct?
- What other functionality would be easy to add?

Cost-Benefit Analysis

When reviewing a design it is sometimes important to perform a cost-benefit analysis. When performing this kind of analysis, a designer will look at an aspect of the design to see two things: what it costs, and how much benefit it provides. The designer will then determine whether the benefit was worth the cost of implementation.

"Cost" does not always refer to money. A feature's cost refers to the resources that must be diverted to it; these could be time, personnel, money, space on the robot, weight, and more. It could also refer to items that must be sacrificed in order to implement the feature being analyzed. (i.e. "If we build a 2 jointed arm, we won't have room for a ball intake on the robot.")

Features that provide a BIG benefit at a small cost are the kind that should be added to the design (it is important to look for these at all stages of the process; a simple addition can often provide big results). High cost items should only be implemented if they provide a big benefit! These considerations are important ones, and designers need to keep them in mind.

STEP 9: IMPLEMENT

Once the design has been completed and approved, it needs to be implemented. Depending on the nature of the problem being solved, the solutions to the problem could vary wildly. Depending on the type of solution, the implementation could also vary. The implementation could consist of using a new process that was designed, or it could consist of following a manufacturing plan and producing some physical object. For instance, in the example of the elevator riddle discussed previously, there are a number of solutions proposed and these solutions all took different forms.

If an engineer is trying to solve how to tie shoes faster, they are designing a process for tying shoes. Their implementation would be to tell people about their new shoe-tying procedure. If an engineer is trying to design a better shoe, their implementation would the manufacture and sale of the new shoes. Implementations can take many forms.

IMPLEMENT for Competition Robotics

In competition robotics, this is the phase where students "build the thing." All the details done in Step 7 are used to create a finished, functional robot to compete with (or a subsystem that is part of a larger finished product)! This stage can involve purchasing components, cutting parts, assembly, and more - anything it takes to produce a final product.

This is also the stage where a team would produce marketing packets, award submissions, and other materials related to their competition, but not associated with the robot. These items are all part of the final implementation.

STEP 10: TEST

In this stage engineers will test their implemented solution to see how well it works. The implementation must be reviewed to see what worked, what didn't, and what should be improved. The testing procedures and results should be well documented. The main thing that should be determined during this stage in the process is whether or not the final implementation performs as expected and fulfills the specifications.

So what happens if the design is not found to be acceptable? The design group must find a way to make it acceptable! The design group needs to come up with a plan of improvement to get the solution up to snuff. Their plan may include starting over and going back to the drawing board to create a new plan entirely.

Once the solution has been implemented, the analysis completed, and the design has been found acceptable, the design process is complete.

TEST for Competition Robotics

In competition robotics this testing can occur during the competition. When the robot is on the field during a match, it is apparent exactly how well it functions! However, this is not a good situation. *Most* robotics teams would prefer to know how well their robot will function BEFORE it takes the field. This is why in an ideal situation teams complete their robots with plenty of time to test and improve it. Continuous improvement is the key to success. Planning ahead for this will allow for testing and adaptation before the competition.

STEP 11: ITERATE

There were several mentions during the design process of repeating certain steps multiple times until an acceptable result is achieved. This act of repetition is known as "iteration." This iteration results in a better end result and is one of the most important parts of design; this is why it is said that *design is an iterative process!*

Students may be familiar with iterative design from their language arts studies. When a student writes an outline, then a rough draft, then a final paper they are completing three iterations of their final paper, refining each one. These iterations make the final paper much better.

One important thing designers should note is that iteration does not just take place at the end of the process, it will happen during EVERY stage in the process.

The design process is NOT a linear thing; it is common to jump from step to step. Sometimes a design team may jump back and forth between steps one and two several times before ever moving onto step three. Design teams should NOT be afraid of going backward in the process. At any step in the process, a design team may find themselves skipping backwards to any other step. The ultimate goal is to create the best design possible by improving it over and over again. Repeat parts of the process to improve the final result.

The greater the number of iterations a design goes through, the better the final result will be, so why would a designer ever stop iterating? At first each repeat will result in large improvements to the design, but the longer the process goes on, the fewer problems there will be to fix and the smaller the improvements. This is known as the law of *diminishing returns*. Improvements to the design will get smaller with each successive improvement. Eventually a designer may decide that the next improvement is too small to be worth the effort, and the design is good enough.

Some designers take longer to call a design "finished" than others because they strive for perfection. Unfortunately, in the real world it is not always possible to achieve perfection. In the real world, if an engineering contractor misses a deadline, they may not get another chance, and they may have trouble finding other contracting jobs!

ITERATE for Competition Robotics

In competition robotics, how does one reconcile the benefits of continuous (and potentially never-ending) improvement with the need for project completion? Simple: each team needs to set a schedule, and then stick to it. This schedule will vary greatly from team to team depending on their circumstances. If a team has six weeks to design and build their robot before they must ship it, they should come up with some sort of schedule for this time period. This schedule can vary in detail greatly. Some teams will plan out each and every step in the process while others will just do a quick overview.

The schedule is not always set in stone; ultimately the only fixed dates are the project start date and the robot completion deadline (usually the date of a competition). Everything else is likely to shift as the process unfolds. Many teams in competitive robotics know these shifts will occur, so they don't even bother trying to plan a schedule in detail.

SUBSYSTEM DESIGN:

How is a design process like an onion? They both have layers! There are often smaller design processes within the main design process. One may end up using a "mini" design process for a small part of the overall design, and then using a smaller process for one aspect of that mini process. To make this easier, the overall design is sometimes broken down into smaller chunks that can be worked on independently. These are referred to as subsystems.

There may be several parallel processes occurring at the same time, each interconnected as part of the overall system. These different layers will probably depend on each other to a certain extent, if only at some interface point. The nature of this System Integration will be discussed later on.

1.2: Engineering/Design Teams
up
1.4: Design Documentation

1.4: Design Documentation

When solving a problem, almost everyone follows a process similar to the one outlined above, even if they only do it subconsciously. Every time they are asked to make a decision, they run through this process without even realizing it. The design process can be accomplished with varying degrees of formality, ranging from the subconscious process everyone does in their head to the highly documented process used in corporate engineering.

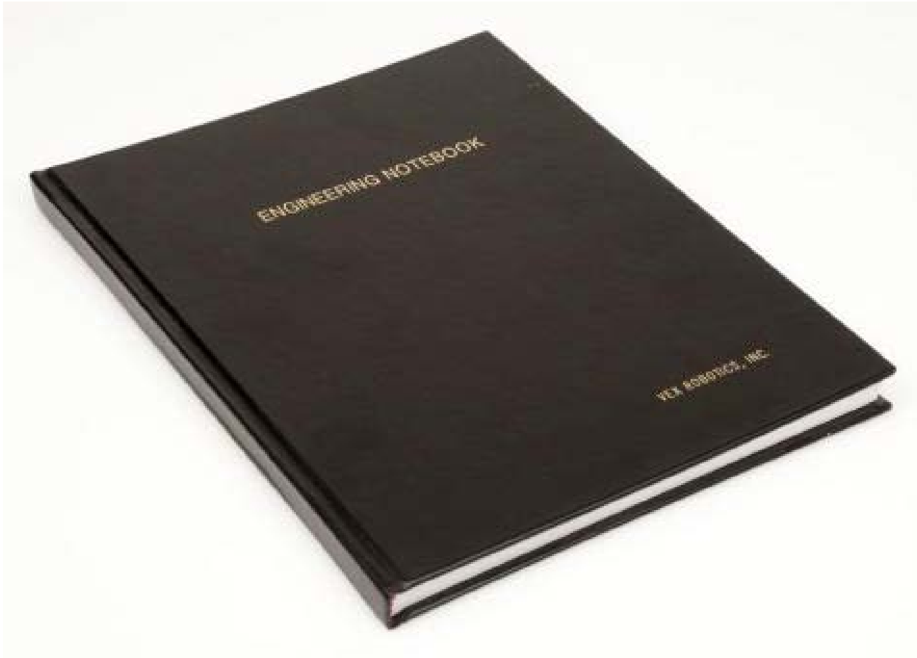
Designers must determine what degree of detail and documentation is needed for their specific process. Many designers are tempted to do everything in their head, thinking that documentation will only slow them down. In truth, a more formalized process will produce a better result. Formalization will promote thoroughness; additional documentation will help prevent mistakes.

In competition robotics it is useful to keep documentation of the design, though the extent of this documentation is sometimes limited by the time available. However, as described above a documented process is a more methodical process. The notes can also be useful when explaining the design to competition judges and they will serve as good documentation for future team members who want to understand the process used.

For the purposes of this class, students should document almost everything in their engineering notebooks.

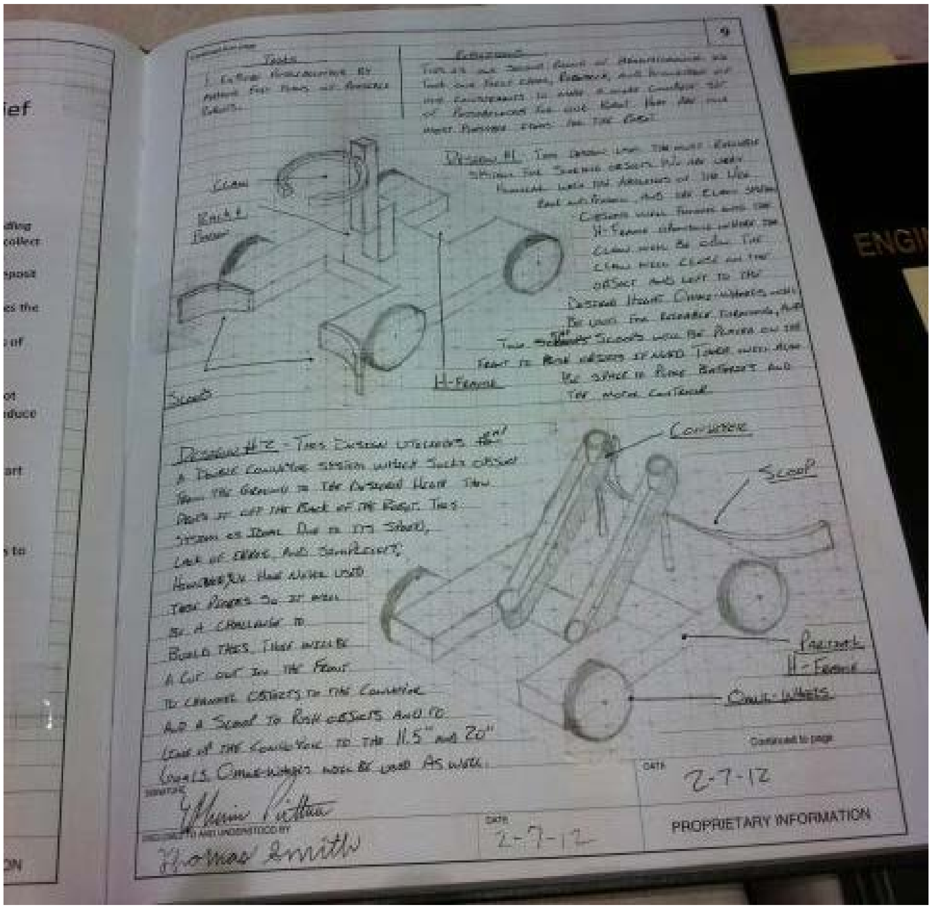
Engineering Notebooks

An Engineering Notebook is a record of the design process; it is basically a “diary” that designers keep as they progress through the process.



Engineering Notebooks come in many different formats, but they should detail each step of the process. They should combine a narrative of the progress, concept sketches, engineering calculations, pictures of prototypes, test procedures, and more. Some of the most important things to record are the decisions made, and the reasoning behind these decisions. Later on in the design process, if a designer runs into a problem and does not remember why something was done a certain way, the notebook will provide a good reference. A Design Notebook should serve as a roadmap such that any outsider can follow the designer's process, understand the choices made by the designer, and end up with the same result.

If a designer gets transferred to a different team in the middle of a project, someone else should be able to read his or her Engineering Notebook and pick up right where they left off.



Every notebook entry should be dated and signed by the designer to provide proof of when the documented work was done; this comes in handy during any patent or intellectual property debates that occur over the design (obviously this typically doesn't apply to the work done by competition robotics teams).

ENGINEERING TOOLS:

Engineers use a variety of tools to help them during the completion of a design process and the solving of a problem. One of these tools is the Engineering Notebook, which was described previously. If robotics teams want to take their Engineering Notebooks to the next level, they can use digital design tools to enhance their work and save valuable production time. One useful tool for digital design creation and documentation is Sketchbook. Autodesk® SketchBook® software allows students to document their robot design ideas, not only in the classroom, but also wherever and whenever they like using the mobile app. During the modeling, testing and developing stages of the design process, students are able to revisit their original concepts in Sketchbook, and quickly modify and save as a new design. This produces a comprehensive log for their Engineering Notebook that records their decisions throughout the design process.

Engineers often also use something called Computer Aided Design (CAD) software to help them in the virtual creation and visualization of their designs. Autodesk's industry leading CAD programs and mobile apps make the modeling, simulation and visualization of competition robot designs quick and easy. Autodesk® Inventor®, Autodesk® Fusion 360™, and Autodesk® ForceEffect™ software are available to students and educators at no charge. Inventor will be used by students in this class, with Fusion as an optional tool for flexibility.

Autodesk Inventor

Autodesk® Inventor® Professional software simplifies the transition from 2D and 3D mechanical design, using intuitive sketching, direct manipulation, product simulation, and design communication. Inventor takes you beyond 3D to Digital Prototyping by enabling a designer to produce an accurate 3D model that can help with the design, visualization, and simulation of the robot before it's built. Digital Prototyping with Inventor will help students design a winning robot, by guiding them through assembly creation and ensuring that all parts and components fit correctly.

Autodesk Fusion 360

Autodesk® Fusion 360™ software is a cloud-based 3D CAD/CAM tool that enables fast and easy exploration of design ideas with an integrated concept-to-production platform. You can quickly iterate on design ideas with sculpting and modeling tools, create assemblies, and produce photorealistic renderings and animations. Make ideas a reality by creating toolpaths to machine your components or using the 3D printing workflow to create a prototype. Robotics teams can work together anytime, anywhere, using collaboration functionality in the cloud.

A mobile app which will support the design of a competition robot is Autodesk ForceEffect.

Autodesk ForceEffect & ForceEffect Motion

Autodesk® ForceEffect and ForceEffect Motion™ engineering apps are purpose-built tools used to quickly and easily simulate design options. ForceEffect enables students to perform static systems analysis using free body diagrams. ForceEffect Motion is ideal for developing mechanical systems with moving parts. Unlike the traditional approach of using paper, pencil, and a calculator to develop equations for design options, Autodesk ForceEffect and ForceEffect Motion do all the simulation and engineering calculations right on your mobile device, enabling designers to quickly and easily simulate options during the concept phase to determine if a design will work.

Students, educators and schools can download the same software used by professionals today, at no charge, by visiting the Autodesk Education Community at www.autodesk.com/education.

CONCLUSION:

As seen in this unit, Engineering and the Engineering Design process are both integral to the development of competition robots. Students will gain practical knowledge in topics related to robotics, and apply them using the engineering design process to design their competition robot.

1.3: What is the Engineering Design Process? | up | 1.5: Design Challenge

1.5: Design Challenge

Students should divide up into design teams and use their new knowledge of engineering and the engineering process to complete the following challenge:

Using nothing but ten letter size sheets of paper, students must create a freestanding tower as tall as possible in 30 minutes. Students are required to spend 5 minutes planning and designing their tower before they receive any materials. Students will then be given ten sheets of paper and allowed ten minutes for prototyping; at the end of the prototyping period ALL paper and prototypes will be collected. Teams will then be given 15 minutes to implement their final tower design. The tower must remain freestanding for at least 30 seconds for its height to count.

Not all the steps in the engineering design process are appropriate for this challenge, however each design team should follow the simplified process shown here:

Step 1 – UNDERSTAND – Define the Problem

Step 2 – DEFINE – Determine Solution Specifications

Step 3 – IDEATE – Generate Concept Solutions

Step 4 – PROTOTYPE – Learn How Your Concepts Work

Step 5 – CHOOSE – Determine a Final Concept

Step 6 – REFINE – Do Detailed Design

Step 7 – IMPLEMENT – Implement the Detailed Solution

Step 8 – TEST – Does the Solution Work?

Step 9 – ITERATE

All students must document the process their group followed in their engineering notebook while including as much detail as possible.

1.4: Design Documentation

up

1.6: Engineering Notebook

1.6: Engineering Notebook

Answer the following questions in your Engineering Notebook:

1. What does an engineer do?
2. What is something that you have used today that was designed by an engineer?
3. Why is classical mechanics such an important part of engineering?
4. How does having constraints placed on a design change the engineering process?
5. Why is making a prototype so important in the design process?
6. What have you learned from the iterative process?

1.5: Design Challenge
up
Unit 2: Introduction to Robotics

Unit 2: Introduction to Robotics

In this unit students will learn about robotics in our world, and how the different aspects of STEM are all used in the field of robotics. This unit will also provide an introduction to the VEX Robotics Design System, students will get an overview of the different subsystems within the VEX system and how they interact together. Students will then put this knowledge into practice as they follow step-by-step directions to build their first robot.

1.6: Engineering Notebook
up
2.1: What is Robotics?

- Curriculum Lesson Content
  - Unit 1: Introduction to Engineering
  - Unit 2: Introduction to Robotics
    - 2.1: What is Robotics?
    - 2.2: VEX Robotics Design System
    - 2.3: Building the VEX Clawbot
    - 2.4: Engineering Notebook
  - Unit 3: Introduction to VEXnet
  - Unit 4: Introduction to Autodesk Inventor
  - Unit 5: THE GAME!
  - Unit 6: Object Manipulation
  - Unit 7: Speed, Power, Torque & DC Motors
  - Unit 8: Mechanical Power Transmission
  - Unit 9: Drivetrain Design
  - Unit 10: Lifting Mechanisms
  - Unit 11: Systems Integration
  - Unit 12: Testing and the Iteration Process
  - Unit 13: Design your Own Part [optional]
- Appendices

- Teacher Materials

2.1: What is Robotics?

A robot is a programmable mechanical device that can perform tasks and interact with its environment, without the aid of human interaction. Robotics is the science and technology behind the design, manufacturing and application of robots.

The word robot was coined by the Czech playwright Karel Capek in 1921. He wrote a play called “Rossum's Universal Robots” that was about a slave class of manufactured human-like servants and their struggle for freedom. The Czech word *robota* loosely means "compulsive servitude.” The word robotics was first used by the famous science fiction writer, Isaac Asimov, in 1941.

Basic Components of a Robot

The components of a robot are the body/frame, control system, manipulators, and drivetrain.

Body/frame: The body or frame can be of any shape and size. Essentially, the body/frame provides the structure of the robot. Most people are comfortable with human-sized and shaped robots that they have seen in movies, but the majority of actual robots look nothing like humans. (NASA's Robonaut, pictured in the previous section is an exception.) They are typically designed more for function than appearance.

Control System: The control system of a robot is equivalent to the central nervous system of a human. It coordinates and controls all aspects of the robot. Sensors provide feedback based on the robot's surroundings, which is then sent to the Central Processing Unit (CPU). The CPU filters this information through the robot's programming and makes decisions based on logic. The same can be done with a variety of inputs or human commands.

Manipulators: To fulfill their purposes, many robots are required to interact with their environment, and the world around them. Sometimes they are required to move or reorient objects from their environments without direct contact by human operators. Unlike the Body/frame and the Control System, manipulators are not integral to a robot, i.e. a robot can exist without a manipulator. This curriculum focuses heavily on manipulators, especially in Unit 6.

Drivetrain: Although some robots are able to perform their tasks from one location, it is often a requirement of robots that they are able to move from location to location. For this task, they require a drivetrain. Drivetrains consist of a powered method of mobility. Humanoid style robots use legs, while most other robots will use some sort of wheeled solution.

Uses and Examples of Robots

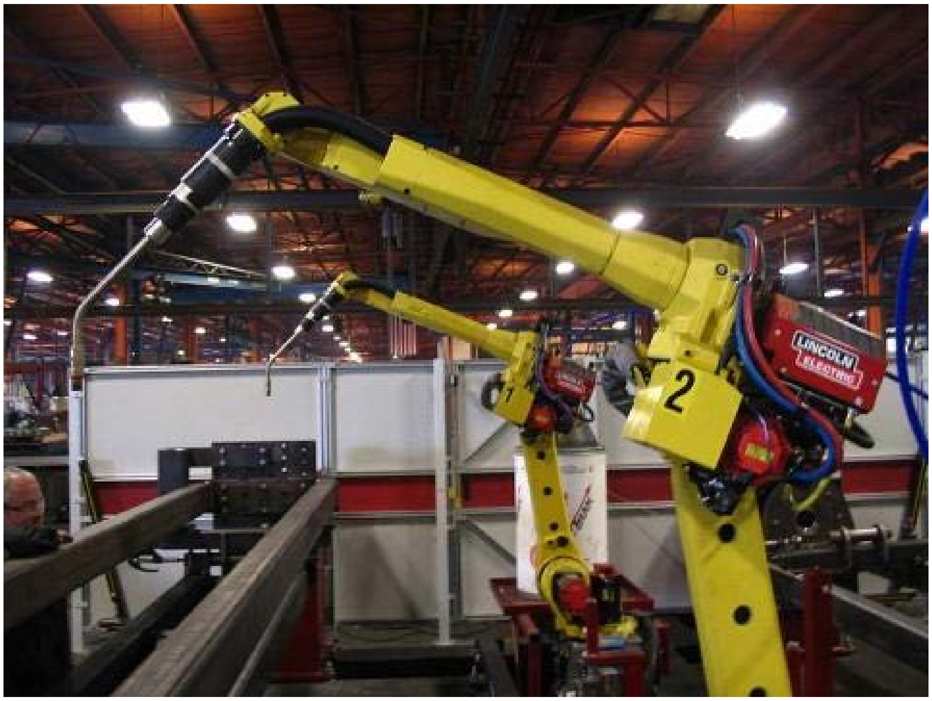
Robots have a variety of modern day uses. These uses can be broken down into three major categories:

- Industrial Robots
- Robots in Research
- Robots in Education

Industrial Robots

In industry, there are numerous jobs that require high degrees of speed and precision. For many years humans were responsible for all these jobs. With the advent of robotic technology, it became evident that many industrial processes could be sped up and performed with a higher degree of precision by the use of robots. Such jobs include packaging, assembly, painting, and palletizing. Initially robots in industry only performed very specialized and repetitive jobs that could be executed with a simple yet precise set of instructions.

However as technology has improved, we now see industrial robots that are much more flexible, making decisions based on complex sensor feedback. Vision systems are now common on industrial robots. By the end of 2014, the International Federation of Robotics predicts that there will be over 1.3 million industrial robots in operation worldwide!



Robots can be designed to perform tasks that would be difficult, dangerous, or impossible for humans to do. For example, robots are now used to defuse bombs, service nuclear reactors, investigate the depths of the ocean and the far reaches of space.

Robots in Research

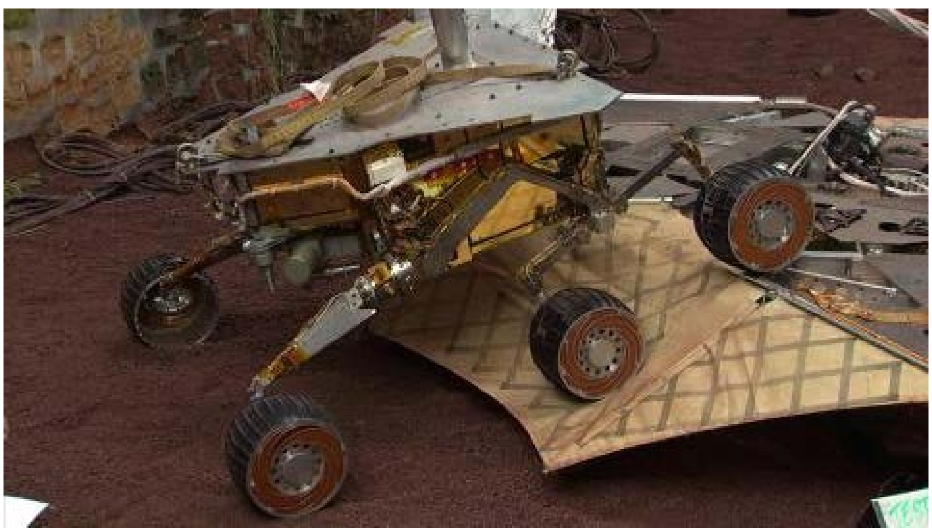
Robots come in very handy in the world of research, as they often can be used to perform tasks or reach locations that would be impossible for humans. Some of the most dangerous and challenging environments are found beyond the Earth. For decades, NASA has utilized probes, landers, and rovers with robotic characteristics to study outer space and planets in our solar system.

Pathfinder and Sojourner

The Mars Pathfinder mission developed a unique technology that allowed the delivery of an instrumented lander and a robotic rover, Sojourner, to the surface of Mars. It was the first robotic roving vehicle to be sent to the planet Mars. Sojourner weighs 11.0 kg (24.3 lbs.) on Earth (about 9 lbs. on Mars) and is about the size of a child's wagon. It has six wheels and could move at speeds up to 0.6 meters (1.9 feet) per minute. The mission landed on Mars on July 4th, 1997. Pathfinder not only accomplished this goal but also returned an unprecedented amount of data and outlived its primary design life.

Spirit and Opportunity

The Mars Exploration Rovers (MERs), Spirit and Opportunity, were sent to Mars in the summer 2003 and landed there in January 2004. Their mission was to search for and characterize a wide range of rocks and soils that hold clues to past water activity on Mars in hopes that a manned mission may someday follow. Although initially planned for a lifespan of 90 days, the elapsed mission time surpassed six years, discovering unimaginable amounts of geological information about Mars.



Space Shuttle Robotic Arm

When NASA scientists first began the design for the space shuttle, they realized that there would have to be some way to get the enormous, but fortunately weightless, cargo and equipment into space safely and efficiently. The remote manipulator system (RMS), or Canadarm, made its first flight into space on November 13, 1981.

The arm has six joints, designed to simulate the joints of the human arm. Two are in the shoulder, one is at the elbow, and three are in the highly dextrous wrist. At the end of the wrist is an end effector which can grab or grapple the desired payload. In the weightless environment of space, it can lift more than 586,000 pounds and place it with incredible accuracy. Its total weight on earth is 994 lbs.

The RMS has been used to launch and rescue satellites and has proven itself invaluable in helping astronauts repair the Hubble Space Telescope. The Canadarm's final shuttle mission took place in July of 2011, marking the 90th time it was used on a shuttle mission.

Mobile Servicing System

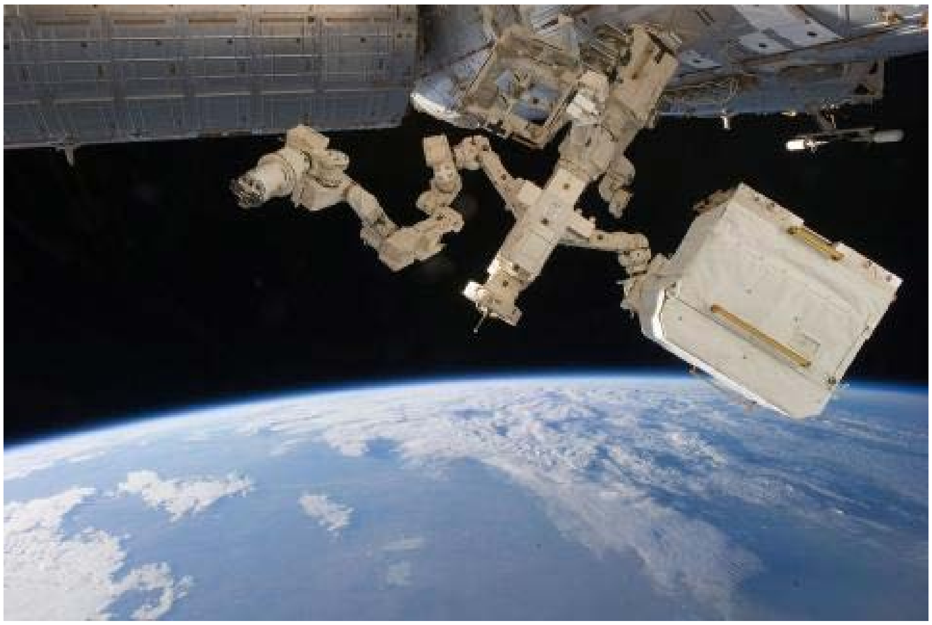
A similar device to the RMS, the Mobile Servicing System (MSS) otherwise known as Canadarm2 was designed to provide manipulation functions for the International Space Station. The MSS is responsible for servicing payloads and instruments attached to the International Space Station, while also assisting with the transport of supplies and equipment around the station.

Dextre

As part of the Space Shuttle mission STS-123 in 2008, the shuttle Endeavour carried the final part of the Special Purpose Dexterous Manipulator, or "Dextre."

Dextre is a robot with two smaller arms. It is capable of handling the delicate assembly tasks currently performed by astronauts during spacewalks. Dextre can transport objects, use tools, and install and remove equipment on the space station. Dextre also is equipped with lights, video equipment, a tool platform, and four tool holders. Sensors enable the robot to "feel" the objects it is dealing with and automatically react to movements or changes. Four mounted cameras enable the crew to observe what is going on.

Dextre's design somewhat resembles a person. The robot has an upper body that can turn at the waist and shoulders that support arms on either side.



Robots in Education

The field of robotics has become an exciting and accessible tool for teaching and supporting science, technology, engineering, mathematics (STEM), design principles, and problem solving. Robotics enables students to use their hands and minds to create like an engineer, artist, and technician does, all at once. It allows for instantaneous application of scientific and mathematical principals.

In today's education system with its budgetary constraints, middle and high schools are on a constant search for cost-effective exciting ways to deliver high-impact programs that integrate technology with multiple disciplines while preparing students for careers in the twenty-first century. Educators quickly see the advantages that robotics projects and curriculum provide by linking in a cross-curriculum method with other disciplines. Additionally, robotics can provide more affordability and reusability of equipment as compared to other prepackaged options.

Today more than ever, schools are adopting robotics in the classroom to revitalize curriculum and meet ever increasing academic standards required for students. Robotics not only has a unique and broad appeal throughout various teaching fields, but it is quite possibly the technical field that will have the largest influence upon our society throughout the next century.

Why is Robotics Important?

As we saw in the uses and examples of robotics section, robotics is an emerging field with applications in many facets of our lives. It is important for all members of society to have an understanding of the technology that surrounds us. However, robotics is important for more than that reason. Robotics provides a unique combination of the pillars of STEM: science, technology, engineering and math. When taught in schools, it allows students to experience a true interdisciplinary lesson while studying a cutting edge and exciting topic. Also, the aesthetics which go into the design and creation of robots allow students to experiment with an artistic side, while working through technical principals. This combination rewards participants on a plethora of different learning levels.

2.2: VEX Robotics Design System

The VEX Robotics Design System, which was created by Innovation First, Inc., is recognized as a leading classroom robotics platform. It has been designed to nurture creative advancement in robotics and knowledge of science, technology, engineering, and math (STEM) education. The VEX system provides teachers and students with an affordable, robust, and state-of-the-art robotics system suitable for both classroom use and for use on the competition playing field. VEX's innovative use of premade and easily formed structural metal, combined with a powerful and user-programmable microprocessor for control, leads to infinite design possibilities.

Beyond science and engineering principles, a VEX Robotics project encourages teamwork, leadership and problem solving among groups. It also allows educators to easily customize projects to meet the level of students' abilities. The affordable VEX platform is expanding rapidly and is now found in middle schools, high schools and university labs around the globe.

For more information on the VEX Robotics Design System, please visit: http://www.vexrobotics.com/

VEX Product Subsystems

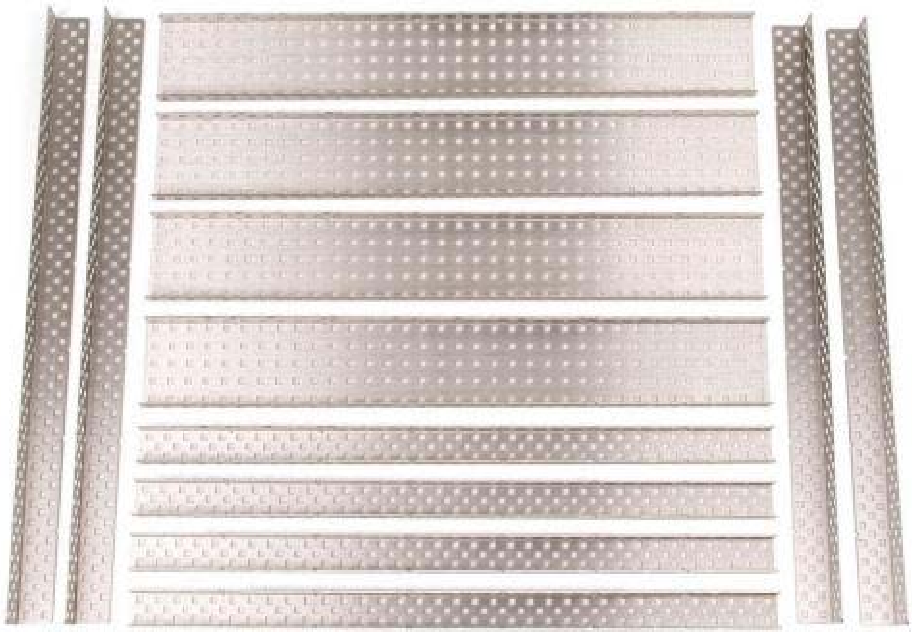
Sub-System	Description
Structure	All metal, fasteners, and structural/mechanical plastic VEX parts
Motion	Motors, Servos, Gears, Chain and Sprockets, Tank Treads and Associated Items
Power	Batteries, Chargers and Associated Items
Sensors	Bumper and Limit Switch, Ultrasonic, Line Follower, Shaft Encoders, Potentiometer
Logic	Microcontroller, PWM Cables, Programming Kits
Control	Joystick, Transmitter, Receiver, Crystals, Signal Splitter, Tether Cables

Structure Subsystem

The parts in the VEX Structure Subsystem form the base of every robot. These parts are the "skeleton" of the robot to which all other parts are attached. This subsystem consists of all the main structural components in the VEX Robotics Design System, including all the metal components and hardware pieces. These pieces connect together to form the "skeleton" or frame of the robot. The Structure and Motion subsystems are very tightly integrated to form the chassis of the robot.

In the VEX Robotics Design System the majority of the components in the Structure Subsystem are made from bent aluminum or steel sheet metal. These pieces come in a variety of shapes and sizes and are suited to different applications on a robot.

The VEX structural pieces all contain square holes (0.182" sq) on a standardized 0.5" grid. This standardized hole-spacing allows for VEX parts to be connected in almost any configuration. The smaller diamond holes are there to help users cut pieces using tin snips or fine toothed hacksaws without leaving sharp corners.



VEX square holes are also used as an alignment feature on some components. These pieces will snap into place in these square holes. For example, when mounting a VEX Bearing Flat there are small tabs which will stick through the square hole and hold it perfectly in alignment. Note that hardware is still required to hold the Bearing Flat onto a structural piece!

Hardware is an important part of the Structure Subsystem. Metal components can be attached together using the 8-32 screws and nuts which are standard in the VEX kit. These screws come in a variety of lengths and can be used to attach multiple thicknesses of metal together, or to mount other components onto the VEX structural pieces. Allen wrenches and other tools are used to tighten or loosen the hardware. There are also smaller 6-32 screws in the VEX Robotics Design System, which are used only for the mounting of VEX Motors and Servos.

When using screws to attach things together, there are three different nuts which can be used.

- Nylock nuts have a plastic insert which will prevent the screw from loosening. These nuts are harder to install; an open-ended wrench is needed to tighten them. However, these nuts will not shake loose due to vibration or movement.
- KEPS nuts have a ring of teeth on one side of them which will grip the piece they are being installed on. An open-ended wrench is not needed to tighten them, but it is recommended. These nuts are installed with the teeth facing the structure. They can loosen up over time if not properly tightened, but they work well in most applications.
- Regular nuts that have no locking feature. These basic hex nuts require a wrench to install and may loosen up over time, especially as a result of vibration or movement. They are thin and can be used in certain applications where a Nylock or KEPS nut would not fit.

Another useful structural component are the 8-32 threaded standoffs; these standoffs come in a variety of lengths and add a great deal of versatility to the VEX kit. Standoffs are useful for mounting components as well as for creating structural beams of great strength.

One of the key features of many VEX structural parts is their “bend-able” and “cut-able” nature. Users can easily modify many of these structural parts into new configurations better suited for their current needs. These parts were designed to be modified!

Motion Subsystem

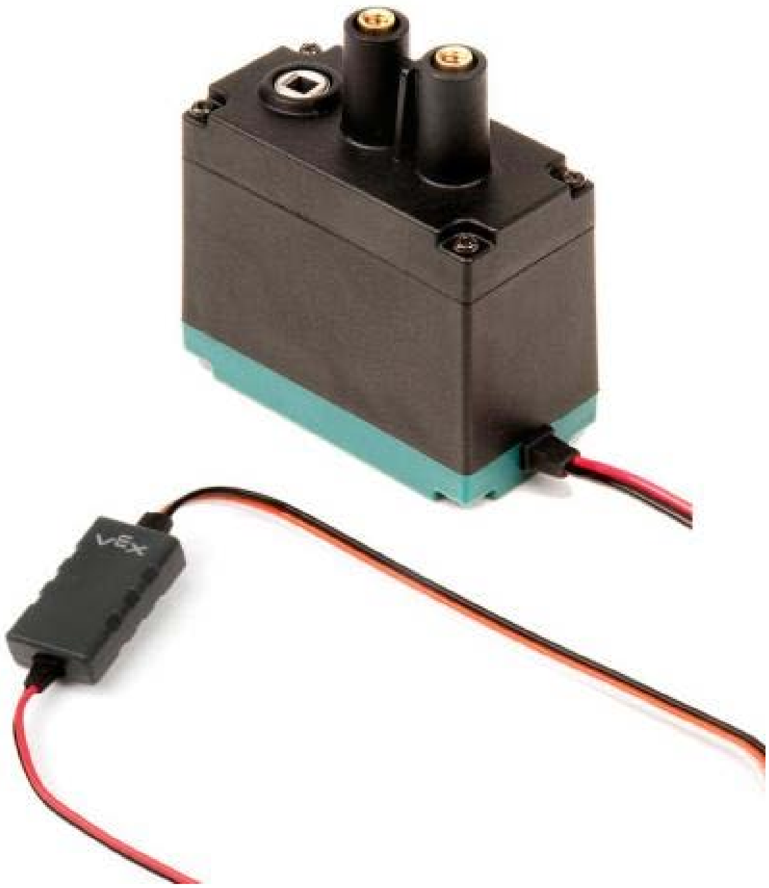
The Motion Subsystem comprises all the components in the VEX Robotics Design System which make a robot move. These components are critical to every robot. The Motion Subsystem is tightly integrated with the components of the Structure Subsystem in almost all robot designs. In the VEX Robotics Design System the motion components are all easily integrated together. This makes it simple to create very complex systems using the basic motion building blocks.

The most fundamental concept of the Motion Subsystem is the use of a square shaft. Most of the VEX motion components use a square hold in their hub which fits tightly on the square VEX shafts. This square hole / square shaft system transmits torque without using cumbersome collars or clamps to grab a round shaft.

The square shaft has rounded corners which allow it to spin easily in a round hole. This allows the use of simple bearings made from Delrin. The Delrin bearing will provide a low friction piece for the shafts to turn in.

These VEX Delrin bearings come in two types, the most common of which is a Bearing Flat. The Bearing Flat mounts directly onto a piece of VEX structure and supports a shaft which runs perpendicular and directly through the structure. Another type of bearing is a Bearing Block; these are similar to the “pillow-blocks” used in industry. The Bearing Block mounts on a piece of structure and supports a shaft which is offset above, below, or to the side of the structure. Some bearings can be mounted to VEX structural components with Bearing Pop Rivets. These rivets are pressed into place for quick mounting. These Rivets are removable; pull out the center piece by pulling up on the head of the Rivet to get it to release.

The key component of any motion system is an actuator (an actuator is something which causes a mechanical system to move). In the VEX Robotics Design System there are several different actuator options. The most common types are the VEX Continuous Rotation Motors and VEX Servos. The Motors can rotate infinitely, while the range of rotation of the Servos is restricted to 150 degrees. Each VEX Motor & Servo comes with a square socket in its face, designed to connect it to the VEX square shafts. By simply inserting a shaft into this socket it is easy to transfer torque directly from a motor into the rest of the Motion Subsystem.



The Motion Subsystem also contains parts designed to keep pieces positioned on a VEX shaft. These pieces include washers, spacers, and shaft collars. VEX Shaft Collars slide onto a shaft, and can be fastened in place using a setscrew.

There are several ways to transfer motion in the VEX Robotics Design System. A number of Motion Subsystem accessory kits are available with a variety of advanced options, including spur gears, sprocket & chains, bevel gears, etc. For a full listing of what is available, please visit: http://www.vexrobotics.com/products/accessories/motion

The VEX Motion Subsystem also contains a variety of components designed to help make robots mobile. This includes a re variety of wheels, tank treads and other options. . For a full listing of what is available, please visit: http://www.vexrobotics.com/products/accessories/motion

Power Subsystem

Power is vital to the operation of all the electronic parts on the robot, including the controller and the motors. With the structural subsystem as the robot's skeleton, and the motion subsystem as the muscle, the power subsystem is the circulatory system that provides the rest of the robot with energy.

There are two major power considerations for a VEX robot; robot power and joystick power. The robot is powered by a rechargeable 7.2V battery pack. The VEXnet Joystick is powered by 6 AAA batteries. For more information on all the power options and accessories available, please visit: http://www.vexrobotics.com/products/accessories/power


7.2V
3000 mAh
NiMH
POWER
PACK
VEX

Sensors Subsystem

The sensor subsystem gives the robot the ability to detect various things in its environment. The sensors are the “eyes and ears” of the robot, and can even enable the robot to function independently of human control. A robot senses its environment and adjusts its own behaviors based on that knowledge. A sensor will generally tell the robot about one very simple thing in the robot’s environment, and the robot’s program will interpret that feedback to determine how it should react.

There are a myriad of sensor options available in the VEX Robotics Design System. Some of these include ultrasonic range finders, gyroscopes, light sensors and optical encoders. For a full list of all sensors available, please visit: http://www.vexrobotics.com/products/accessories/sensors

Logic Subsystem

The Logic Subsystem major component is one of the VEX Microcontrollers. A microcontroller is the most integral component of the entire VEX system, because it coordinates and controls all the other components. The Logic Subsystem is effectively the robot’s brain.

The VEX Cortex Microcontroller comes preprogrammed with a default routine which allows users to get their robots up and running as quickly as possible. With the use of jumper pins, quick adjustments can be made to this default code for greater flexibility. For more advanced programming options, the microcontroller can be fully user programmed using one of the available programming options. More information on these options can be found here: http://www.vexrobotics.com/products/programming

Control Subsystem

The Control Subsystem enables a human operator to maneuver the robot. Commands are issued through joysticks and buttons on the VEXnet Joystick, and sent wirelessly to the robot. In this way, the robot can be controlled through a combination of manual and autonomous methods.

The VEXnet Joystick allows a human operator to control a robot in real time using the innovative VEXnet Wireless link. The joystick has two 2-axis analog joysticks, 4 trigger buttons and two 4-button directional pads.

[2.1: What is Robotics?](#) | [up](#) | [2.3: Building the VEX Clawbot](#)

5.1: Introduction

When faced with the task of designing and building a VEX robot, students (and teachers) will often immediately want to pick up their tools and materials and start building. This may be the most active and fun way to get started, but it is not the best path. Imagine introducing the sport of football to a group of students for the very first time. Naturally, the tendency would be to start throwing the ball around right away, but this would not be the best approach. To truly master the game, there are a series of steps to take before beginning the actual play.

- Learning the rules of the game. What are the objectives? What actions are allowed? What actions are prohibited?
- Analyzing potential strategies.
- Stretching to get into optimal shape.
- Practice.

These steps are all crucial elements to ensure success. Building a robot to play a competitive game is no different. There are a series of steps that make up the design process that are integral to ensuring the success of the robots, while giving plenty of opportunities for students to learn and develop crucial analytical skills.

Unit 5: THE GAME!

up

5.2: Strategic Design

5.2: Strategic Design

In Unit 1 the engineering design process was defined and thoroughly explained. In Competition Robotics, Steps 1-3 (Understand, Define, Explore) are often known as "Strategic Design". Strategic Design is the process of determining what a robot should be able to do. In this process you are not trying to solve the problems of what the robot will look like, or how the robot will complete its tasks. Strategic Design occurs before both of those problems can and will be solved. It is impossible to build a successful competition robot without knowing what the robot is supposed to accomplish. It would be like going for a walk without knowing the final destination. You would definitely be able to meander along, but the chances of actually reaching the destination are slim to none. Any attempt to build a competition robot without having done the Strategic Design would still result in a robot, but the chances of it succeeding would not be very high.

5.1: Introduction

up

5.3: Defining Objectives

5.3: Defining Objectives

Designing and building competition robots is a rewarding and fun task. However, the task becomes much more enjoyable when done successfully. This of course begs the question, “How do we define success?” In terms of competition robotics, a successful robot is defined as one that can regularly win matches based on its own merits and skills. Throughout the design process, this needs to be the overriding goal. When making any sort of decision or trade off, the team needs to ask themselves, “Will this choice help the team win matches?” If the answer is “no”, then the decision needs to be revisited. Of course, this is a very broad question. Later in the unit, this question will be broken down into more tangible and measurable items, many defined by the team itself. There are also other secondary objectives involved, which may not directly (or even indirectly) lead to the winning of matches. Considerations such as aesthetics, design elegance and even “shock value” are all potential aims. However, objectives must always remain secondary to prevent them from interfering with the goals of success. For example, a team should not sacrifice a tangible way for their robot to win matches in a pure effort to make the robot more aesthetically pleasing.

5.2: Strategic Design

up

5.4: Analyzing the Game

- Teacher Materials

5.4: Analyzing the Game

The overall goal in competition robotics is to win matches, but this says nothing about the goals within a specific game. When presented with a competition robotics game challenge, the first step is to analyze the game and determine the optimal way to win matches. Determining this optimal strategic design will allow a team then make the optimal physical design of the robot. The first step in analyzing the game is very simple; however, much like strategic design, it is often overlooked. This simple step is reading the game rules. The game rules are, in a sense, the design specifications for a competition robot. The rules explain the objectives of the game, what actions are permitted, and which actions are prohibited. Almost all competition robotics games are structured so that a team wins matches by outscoring their opponents. Assuming that the game being played is in this format, after completely reading the rules a team needs to ask the following questions.

1. What are the different ways of scoring points?

- Make a list of all potential ways to score points, no matter how obscure they may be. Competition robotics games can often be much more complicated than a conventional sport. For example, in soccer you score points by kicking the ball into a goal. That's it. However, in competition robotics there can be multiple ways to score points for performing an assortment of different tasks. In fact, sometimes these tasks are not fully obvious to the untrained eye. To avoid missing these scoring potentials, one must read every detail of the rules and also learn to read between the lines. Consider the following example:

A game has an objective of scoring points by being on a platform. The platform consists of an elevated surface with a tall flagpole in the center. The rules state that a robot is considered to be "on" the platform if it is touching any part of the platform, but not touching the ground. The obvious way to score points by being on the platform would be to have the robot climb onto the elevated platform. However, the less obvious way to score points on the platform would be to grab the flagpole and to climb it such that the robot is no longer touching the ground. Recall that the flagpole is part of the platform, and the rules simply state that you must be touching the platform and not touching the ground. It is vital that a team carefully searches the rules for all possibilities of this sort. Missing them will ensure that a team misses valuable opportunities to score points, thereby reducing the chances of winning a match.

2. What are the different ways of stopping your opponent from scoring points?

- Make a list of all potential ways to deny points, no matter how obscure they may be. In any competition robotics match that involves robots playing against each other, the skill of denying points (known as defense) is invaluable. Many teams only focus on scoring points, not realizing the value towards winning a match that is provided by denying points.

- Looking at the above two lists, what is the maximum possible score in a match? Is there a maximum possible score?
  - To determine the maximum score, look at all the potential ways to score points and determine if any of them are limited. For example, if the game involves scoring pop cans in containers, the limits could be the number of pop cans or the size of the container.

5.3: Defining Objectives

up

5.5: Cost Benefit Analysis

5.5: Cost Benefit Analysis

Now that the various ways to earn and deny points are understood, it's time to rank these options in terms of how they contribute to the overall goal: winning matches. A rudimentary way of ranking these options would be to say the ones that provide the most points are the most valuable. However, it is not that simple. The task that earns the most points may be incredibly difficult, while a task that earns less points may be very simple to accomplish. It is for this reason that a cost-benefit analysis must be performed. In a cost-benefit analysis, a comparison between the cost of an item and the benefit of the item takes place. In the case of a cost-benefit analysis within the Strategic Design of a competition robot, the "cost" is the level of difficulty of the given task, while the "benefit" is the number of points earned or denied by the same task. The goal is to identify the tasks which give the highest ratio of benefit to cost.

How does one determine the difficulty of performing a task? There are many factors to consider when assessing the difficulty of a task within a competition robotics game. Some of these include:

- How long does it take to complete the task? The more time spent, the more difficult the task.
- How much distance needs to be traveled to complete the task? This is similar to the previous item, since distance is proportional to time
- Many tasks within competition robotics require lifting and placing objects, thus the weight of the object is a factor. Lifting a ping pong ball is not as difficult as lifting a bowling ball. The heavier the object, the more difficult the task.
- Similar to weight is the height of the where the object is being placed. The higher the placement height, the more difficult the task.
- The precision required for the task. For example, parking a robot in a 12"x12" space compared to parking a robot in a 24"x24" space. The more precision required, the harder the task.
- Does the task require a specific mechanism to complete? Tasks that can be done in conjunction with other tasks involve less difficulty, since you can essentially do two tasks at once (or two tasks with the same robot feature.) This is why denying an opponent a point is often seen as not very difficult. While it may be difficult to defend an opponent during a match, it can typically be done without adding extra functionality to the robot; simply blocking the opposing robot's path can suffice. Thus, it is an easy feature to add to a robot.

Here's a simple exercise on how to perform a basic cost-benefit analysis based on what has been taught so far. The following exercise is a good way of taking the analysis from the previous steps and organizing it in such a way that choosing the optimal strategy becomes a much easier task.

1. From the "analyzing the game" portion of the unit, take the list of all the different possible robot tasks, whether they are to score points, or to prevent the opponent from scoring.
2. Assign a rank on a scale of 1-100 for each task, indicating the benefit towards winning a match. (100 being the most beneficial, 1 being the least.) Using a relative scale from 1-100 as opposed to the pure point values allows for a more uniform comparison between tasks, since some tasks will not have a clearly defined point value.
3. Assign a rank on a scale of 1-10 for each task, indicating how difficult each task will be to complete. (100 being the most difficult, 1 being the easiest.) When evaluating difficulty, consider both how difficult it would be to execute the task during a match, as well as how difficult it would be to build a robot to complete the task.
4. Take the ratio of benefit to difficulty for each task. The task with the highest ratio is the most optimal according to your analysis.

At this point, revisit the list of items. Do the most optimal tasks based on the analysis agree with personal assessments of the tasks? If not, it is important to determine why. A very common reason is usually that the initial rankings and assessments of cost-benefit were done incorrectly. However, sometimes the most optimal strategies are not intuitive. That is why this type of analysis is so crucial.

How does one avoid making incorrect assessments of cost and benefit? One of the most common errors that can take place in these types of assessments is the underestimation of the level of difficulty to complete tasks. As such, tasks which are harder can often be overvalued. This is very dangerous at it leads to overcomplicated robots that do not perform well. It is important to always be realistic and reasonable when evaluating the level of difficulty of any task.

5.4: Analyzing the Game

up

5.6: Prioritization of Tasks

5.6: Prioritization of Tasks

Now that a cost-benefit analysis of all the potential game tasks as been determined, it is time to make design priority lists. To do this, start with two separate lists. The first list covers robot qualities. This is the list of “What should the robot be like?” Examples of qualities include:

- Speed - “The robot should be fast.”
- Power – “The robot should be strong, able to push things easily.”
- Agility – “The robot should be easily maneuverable.”
- Low Center of Gravity – “The robot shouldn’t tip over easily.”

The qualities are based off the list of optimal tasks. The most desired qualities are the ones which are most critical to the success of the most optimal tasks. For example, if the most optimal task is placing baseballs in a goal on the opposite side of the field, speed would be very desirable in order to reduce the time spent traversing the distance across the field.

Below is an example priority list of robot qualities for a hypothetical competition robotics game. The objective in this game is to score baseballs in three goals at three different heights. More points are earned for baseballs scored in the higher goals. The goals are located on the opposite side of the field from the baseballs.

1. Speed. The robot needs to be fast to repeatedly cross the field to gather baseballs and score them.
2. Accuracy. The robot needs to reach desired positions in an accurate fashion, to have the precisions required to score the baseballs in the goals.
3. Agility. The robot needs to be quite maneuverable in order to avoid obstacles and opponent robots in an effort to gather the baseballs.
4. Low Center of Gravity. In order to reach the high valued goals, the robot must reach high in the air. Thus, the robot must not tip at these heights.

The second list covers robot functionalities. This is a list of the abilities that the robot should have. This list should just include the ability, and not the actual implementation. That comes at a later stage of the design process. If possible, it helps to quantify the abilities. Examples of abilities that have been quantified include:

- Traversing the field – “The robot should be able to traverse the full length of the field in less than five seconds.”
- Picking up objects – “The robot should be able to pick up three baseballs at a time from the ground.”
- Depositing objects – “The robot should be able to accurately lift one baseball to a height of 24” and release it, all in a time of four seconds.”

Once again, these qualities are based off the list of optimal tasks. The abilities that are prioritized higher are the ones that are most critical to the success of the optimal tasks. At this point, there should be natural parallels between the two lists. Below is an example priority list of robot abilities to play the same game described above.

1. Drive – The robot needs the ability to drive across the field in less than four seconds, to maximize the amount of baseballs scored.

The drive will also provide the ability to play defense against opposing robots.

2. Picking up baseballs – The robot needs to be able to instantly pickup any baseball it touches, with a minimal amount of lining up. A large tolerance, allowing for minimal accuracy in lining up is desired to make the process as fast as possible. Being able to hold two baseballs at a time is desired, as it minimizes the number of trips across the field that are needed.
3. Placing baseballs – The robot needs to be able to elevate to the height of the highest goal in less than two seconds and place two baseballs in the goal.

Notice that there is a sequential order to these sample priority lists. This is by design. You cannot place baseballs in a goal if you cannot pick up baseballs. You cannot pick up baseballs without being able to get to the baseballs. Thus the ability to place and pick up baseballs is completely dependent on the ability to get to the baseballs. Therefore, the ability to drive has to be your most important ability. Any time a specific ability is dependent upon another ability, this other ability must be a higher priority. After all, one cannot walk without first being able to crawl.

This example game is quite simple, hence the priority lists are also simple and straightforward. This is not the case with most competition robotics games. The lists are based on complex and detailed analysis, and will be much longer and more intricate than the ones presented here. The combination of these two lists will act as the Functional Requirements (See Unit 1, Step 3: Define, for more details) for the robot.

With the completed prioritized lists of abilities and qualities, it is now possible to start working on the new defined problem: “How do we build a robot that achieves the stated abilities and qualities?” With the new problem defined and identified, the next step is to proceed with the engineering process from Step 3. In other words, start deciding which qualities and abilities to pursue and how to pursue them based on the given design constraints. Please be sure to review the detailed information and lessons on this topic from Unit 1.

5.5: Cost Benefit Analysis
up
5.7: Engineering Notebook

- Teacher Materials

5.7: Engineering Notebook

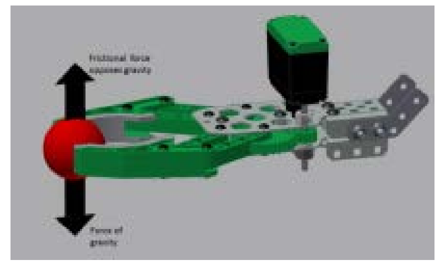
Answer the following questions in your Engineering Notebook:

1. How can you maximize the number of points you can score during the game?
2. How can you keep your opponent from scoring efficiently during the game?
3. How do you choose what features of the robot are needed to play the game?

5.6: Prioritization of Tasks up Unit 6: Object Manipulation

Unit 6: Object Manipulation

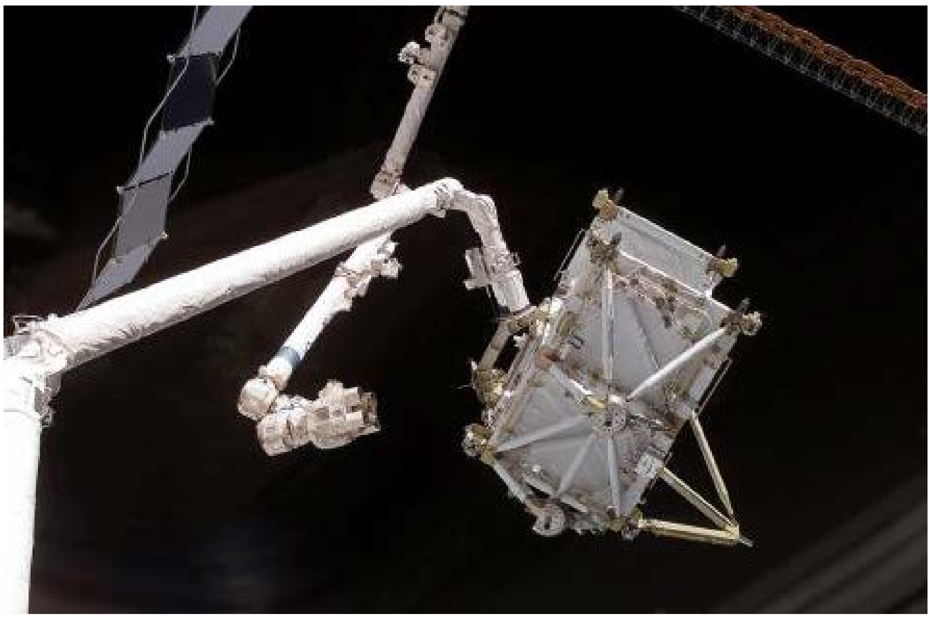
In this unit, students will learn about the different types and categories of robot manipulators. Students will be presented with robot manipulators from the real world, and shown the basic principles behind their operation. Students will then create their own object manipulator for use on their competition robot.



5.7: Engineering Notebook | up | 6.1: Introduction

6.1: Introduction

All robots are designed with a purpose in mind, and these purposes can vary greatly. Robots are traditionally used for tasks that would be unsuitable for a human to do, mainly because these tasks are dangerous, or inaccessible to humans. Dangerous tasks like bomb disposal or handling hazardous waste, as well as inaccessible tasks such as interplanetary exploration are all perfectly suited to robots.



To fulfill their purposes, many robots are required to interact with their environment, and the world around them. Sometimes they are required to move or reorient objects from their environments without direct contact by human operators.

The need for object manipulators applies in competition robotics as well. In the typical VEX Robotics Competition students build a robot to play head-to-head matches against other robots. These games traditionally include some sort of game object that robots must manipulate in such a way that they score points.

The above image shows goal and scored game objects from a past VRC game. Red and Blue colored rings were picked up by robots and placed over posts similar to the one above.

The above two images show a blue trough, and the yellow and green sacks that robots picked up and dumped in them.

The above two images show colored balls and barrels, and the cylindrical goals robots dumped them in.

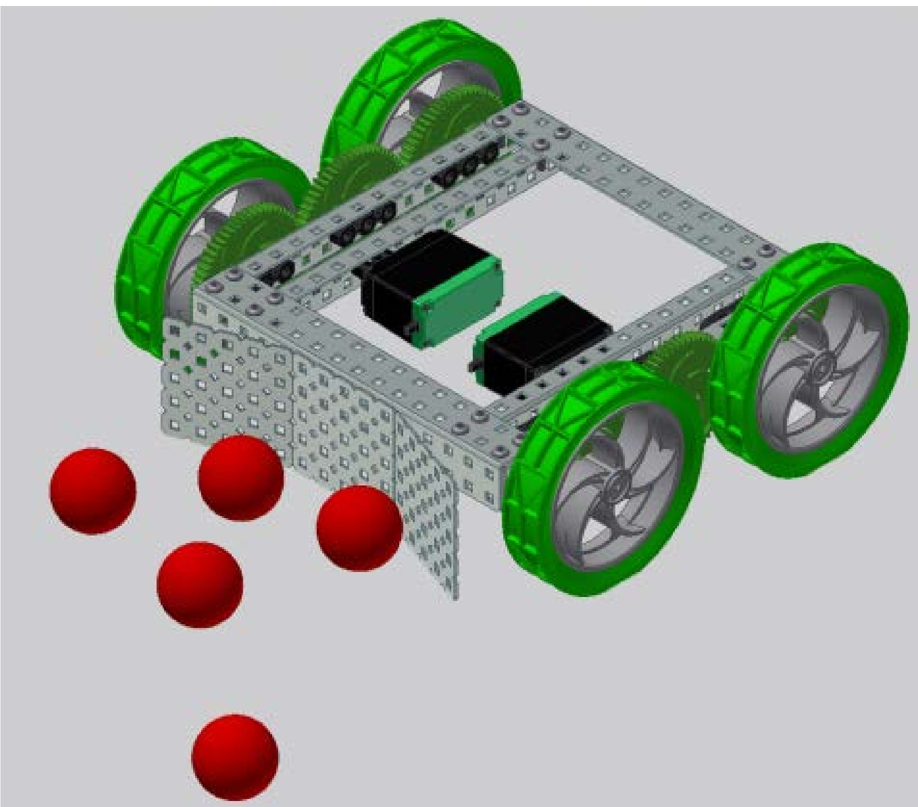
Unit 6: Object Manipulation
up
6.2: Manipulators

6.2: Manipulators

In Competition Robotics, one can classify most object manipulators into three basic categories: Plows, Scoops, and Friction Grabbers.

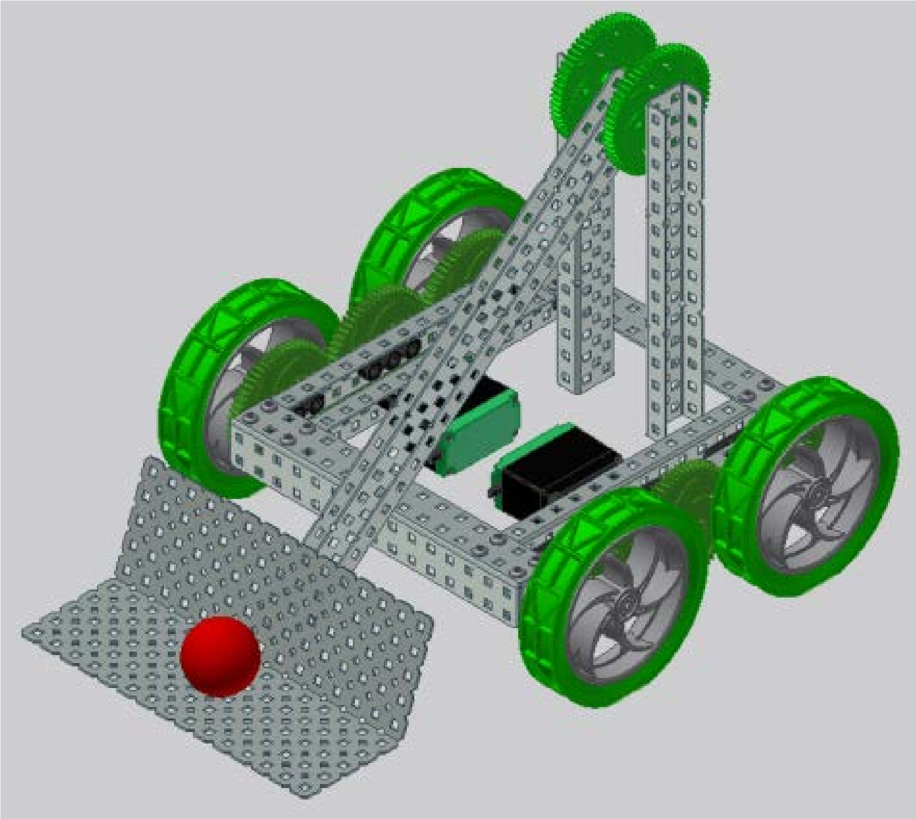
Plows:

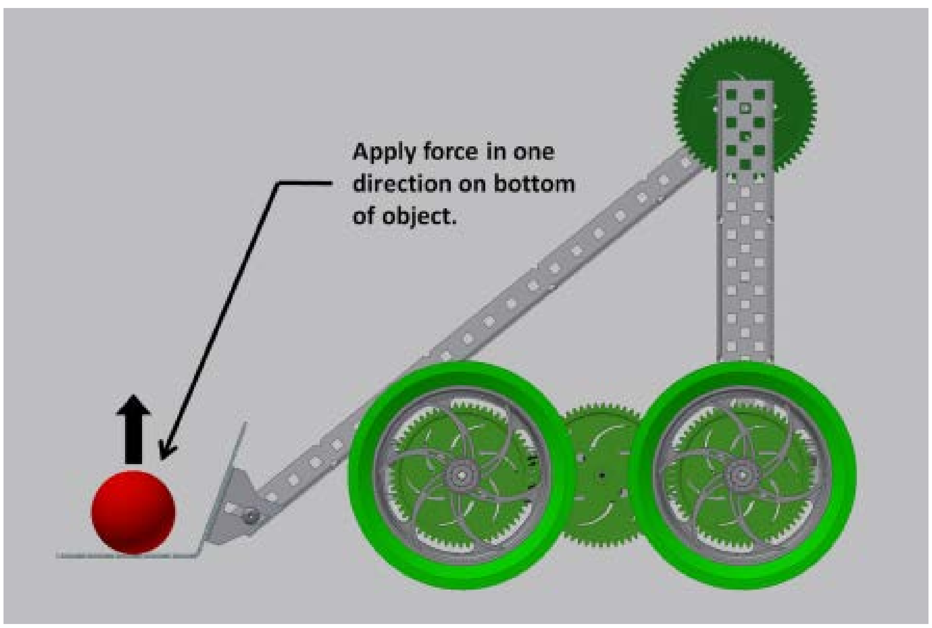
The first category consists of manipulators that apply a single force to the side of an object. They move objects without actually picking them up. One of the most common forms of this manipulator is a simple plow.



Scoops:

The second category comprises manipulators that apply force underneath an object such that the object can be elevated and carried. The most common form of this manipulator type is a scoop.





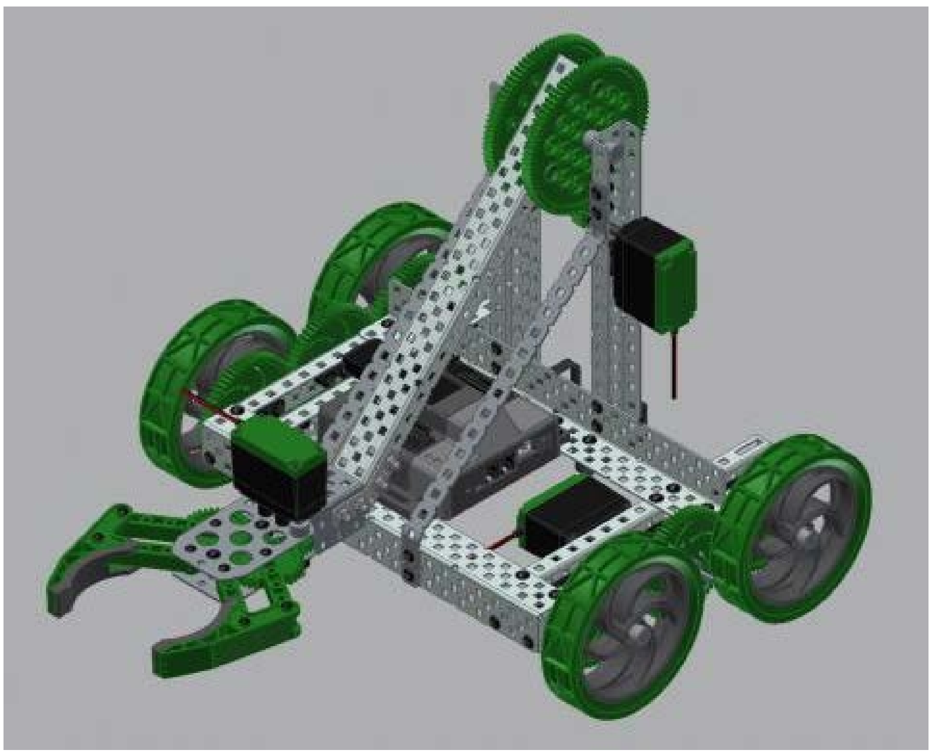
Scoop Manipulator

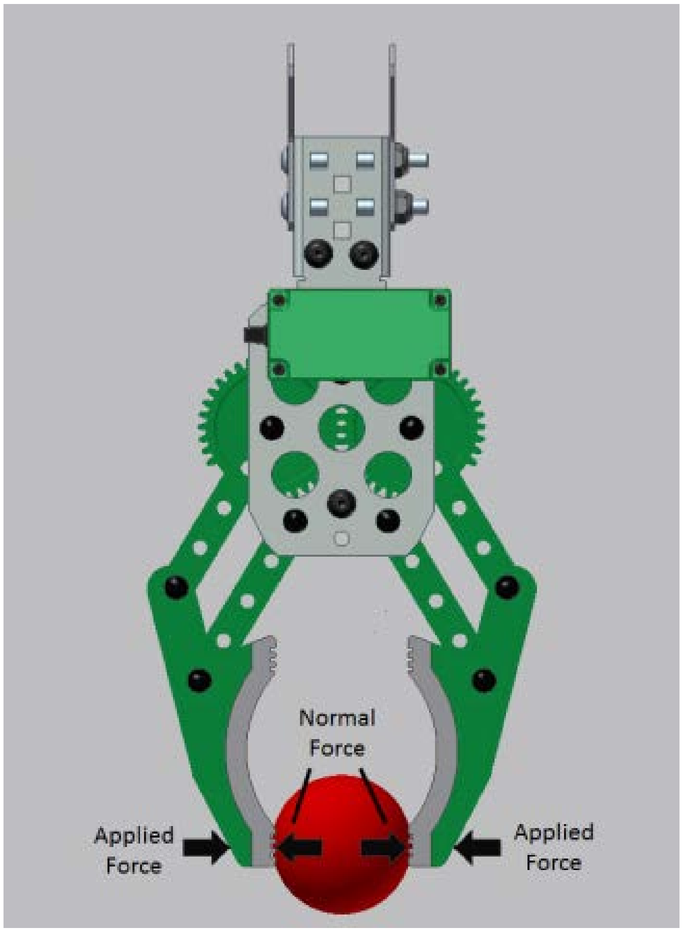
Click here to download this video.

This video shows an articulated scoop that has been modeled and animated in Autodesk Inventor. The step by step instructions to model the manipulator can be found later in this unit, section 6.5.

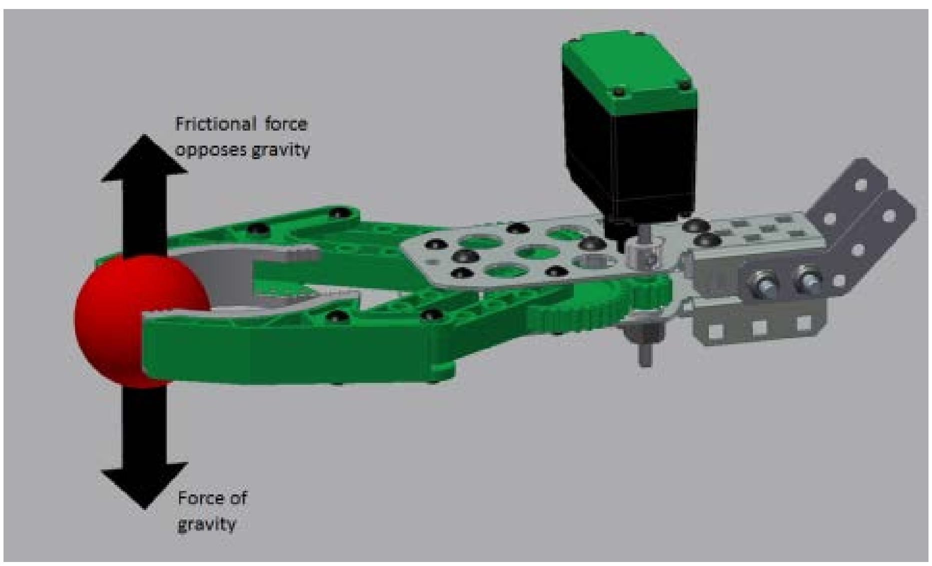
Friction Grabbers:

The third category comprises manipulators that apply a normal force between the object and a traction pad, and then rely on the frictional force between the object and this traction pad to manipulate the object. This manipulator category can be described as friction grabbers. The most common form of this manipulator is a claw which pinches an object – the claw fingers pressing against the object provides the normal force, and the friction between the fingers and the object allows for the object to be manipulated.





The above image shows the force exerted by the claw fingers on the ball, and the resulting normal forces.



The above image shows the friction force between the ball and claw fingers opposing the force of gravity pulling down on the ball.

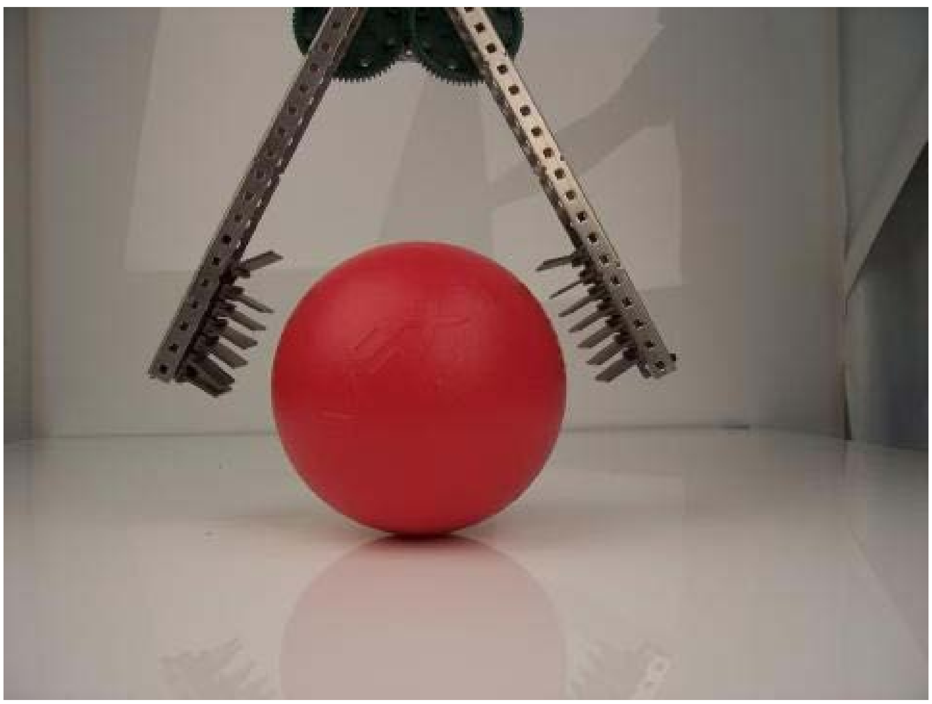
Gripper Manipulator

Click here to download this video.

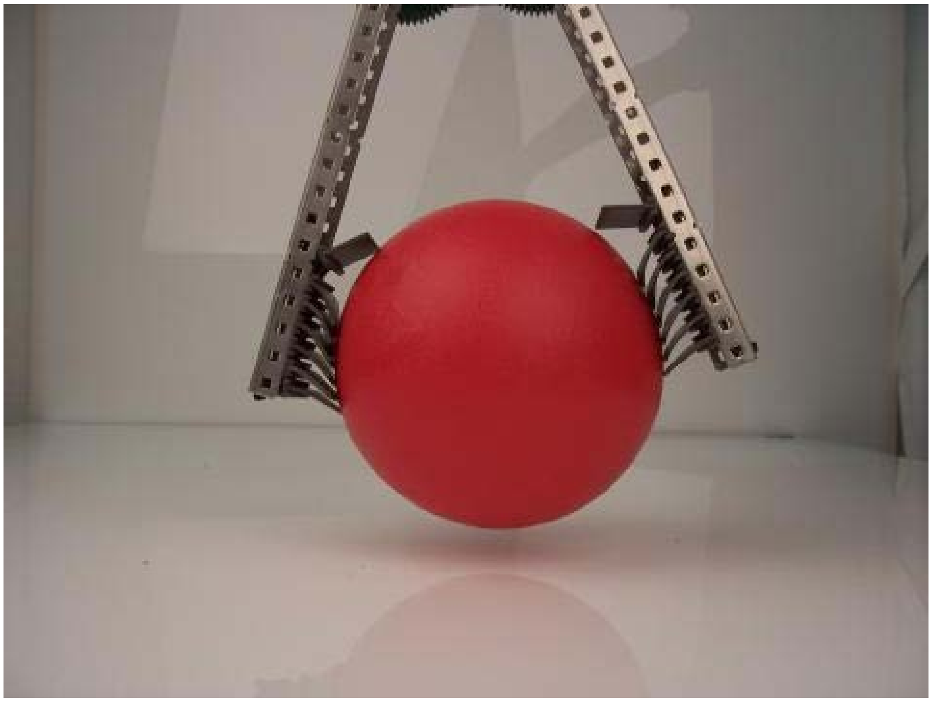
This video shows some applications of friction grippers that have been modeled and animated in Autodesk Inventor.These categories of manipulators are not mutually exclusive; object manipulators can utilize aspects of more than one but using these categories enables a designer to describe almost all types of manipulators. Some common manipulators from competition robotics can be seen below.

Pinching Claw:

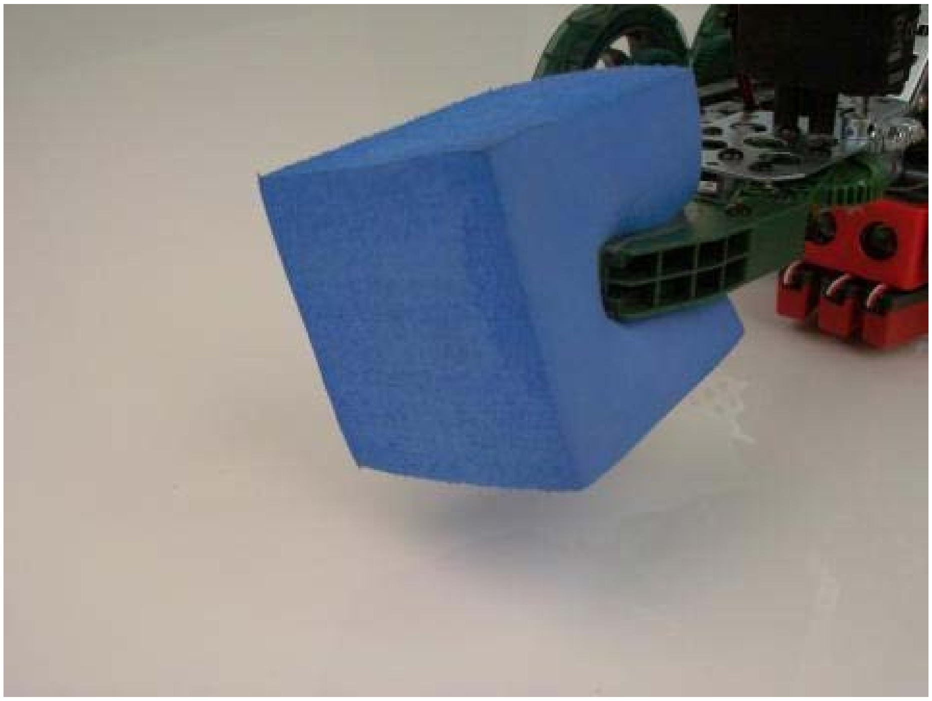
As mentioned above, pinching claws are great examples of friction grabbers. With just a few minor tweaks one can configure a pinching claw to accommodate a wide variety of objects; many different types are possible.



Designers creating a pinching claw should include some sort of elastic “give” in the design. This elasticity should be present in either the object being gripped (i.e. a squishy ball) or in the claw itself (i.e. flexible fingers). Including this elasticity will allow for a more consistent object grip since it will help to provide a more consistent normal force on the object. This consistent normal force will provide a consistent frictional force and result in a solid grip.

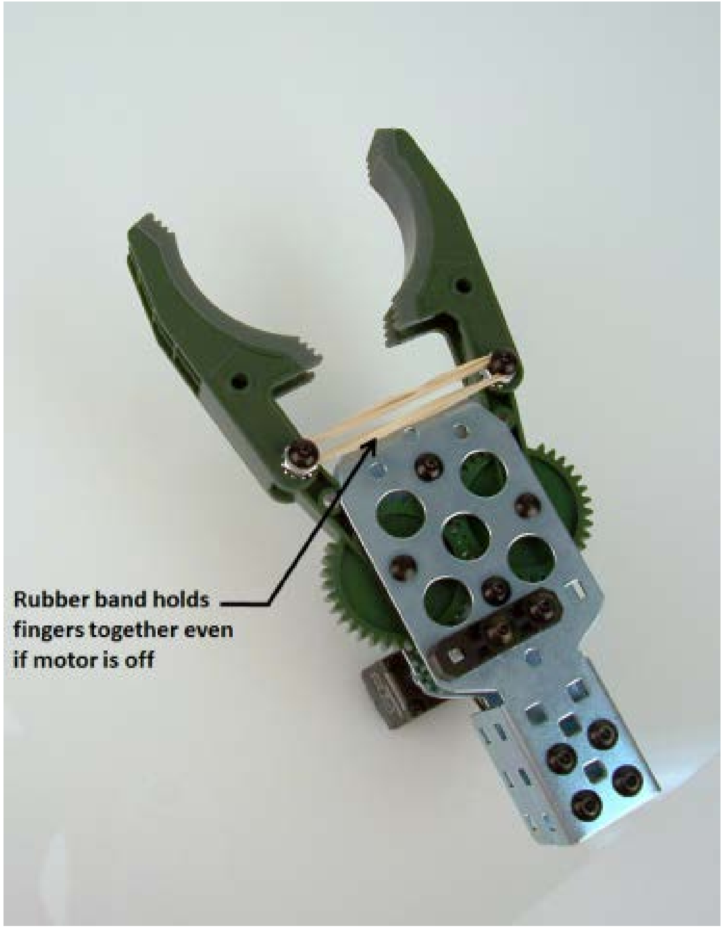


The above image shows an example of claw elasticity – the claw has flexible fingers on each gripping arm.

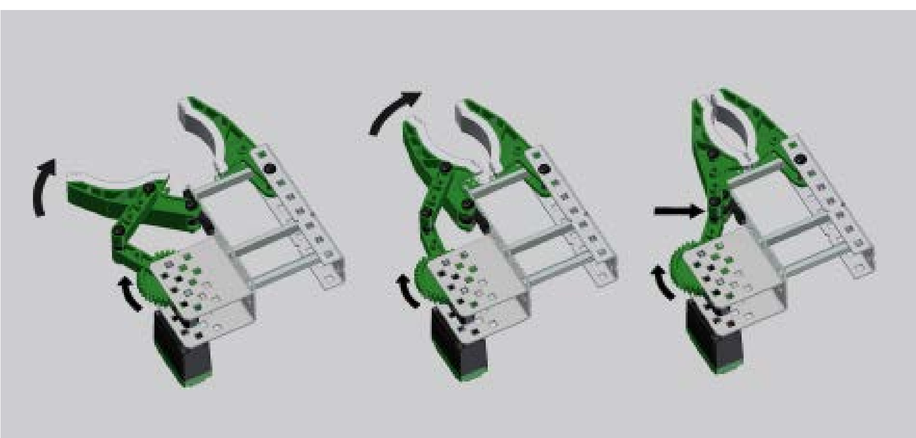


The above image shows an example of object elasticity – the foam cube is compressing as the claw pinches it.

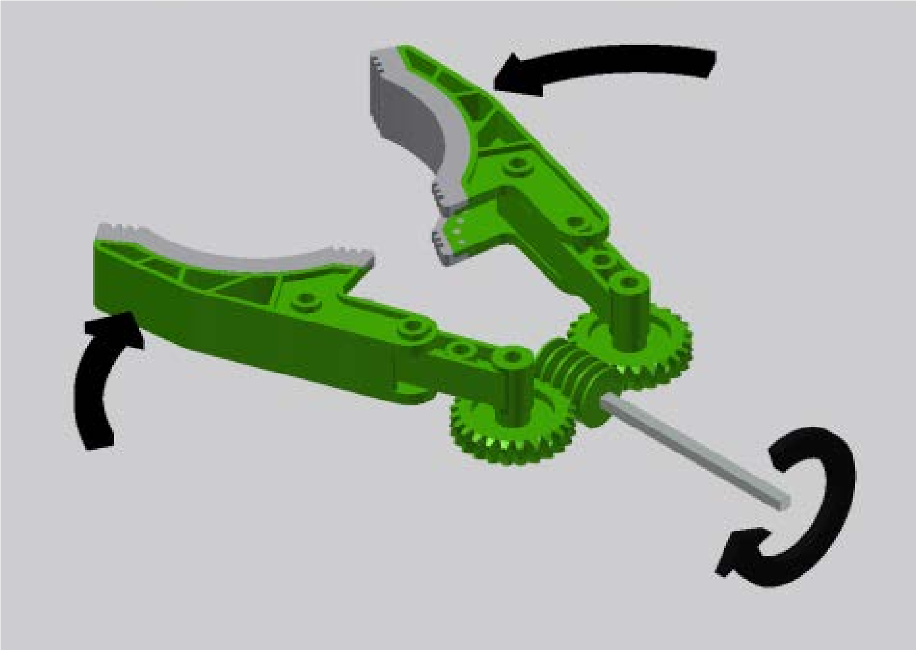
Good pinching claw designs don't require the motor to be continually powered to maintain grip on the object. Some pinching claws utilize internal springs to continually provide force, others have some sort of locking mechanism, and some simply include an actuation mechanism which cannot be back-driven (i.e. worm gear).



The above image shows an example of a claw which uses rubber bands to stay spring-loaded closed.



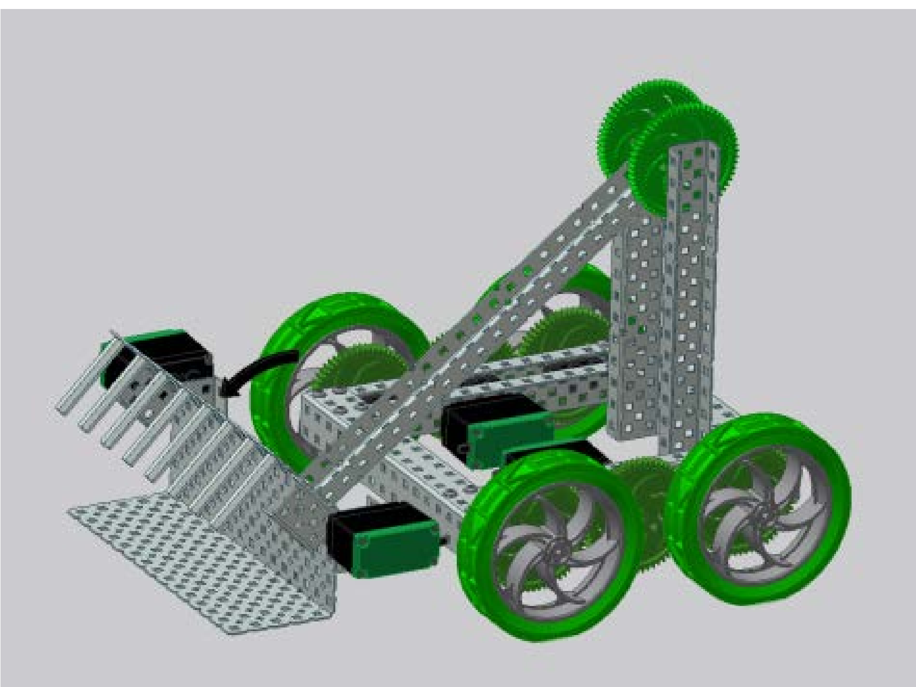
The above image shows an example of a claw that uses a mechanical linkage to lock closed.

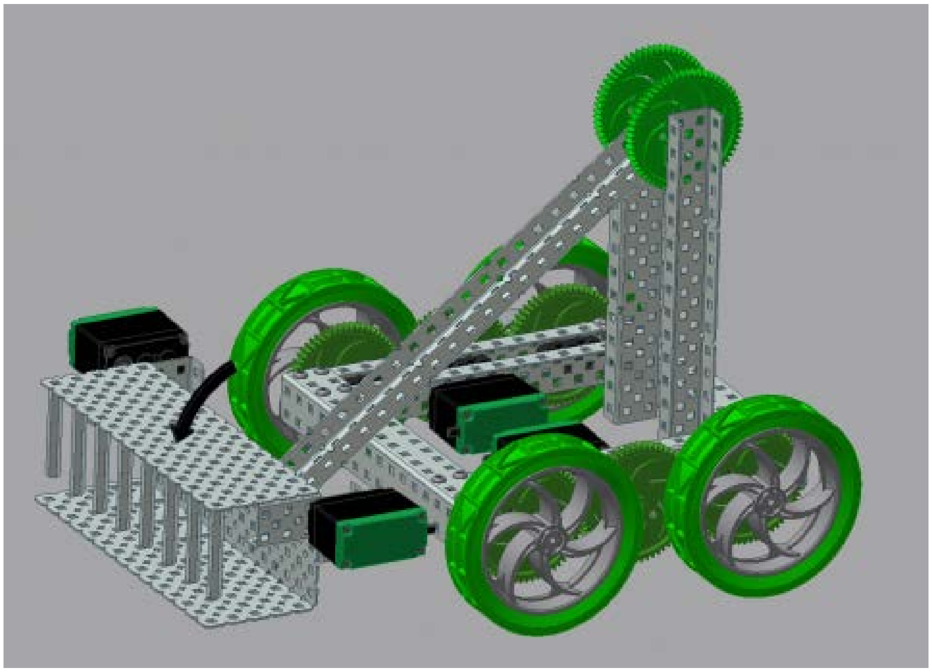


The above image shows an example of a claw driven by a worm gear.

Top-Jaw Grabber:
Another common type of competition object manipulator is a “top-jaw” type design.

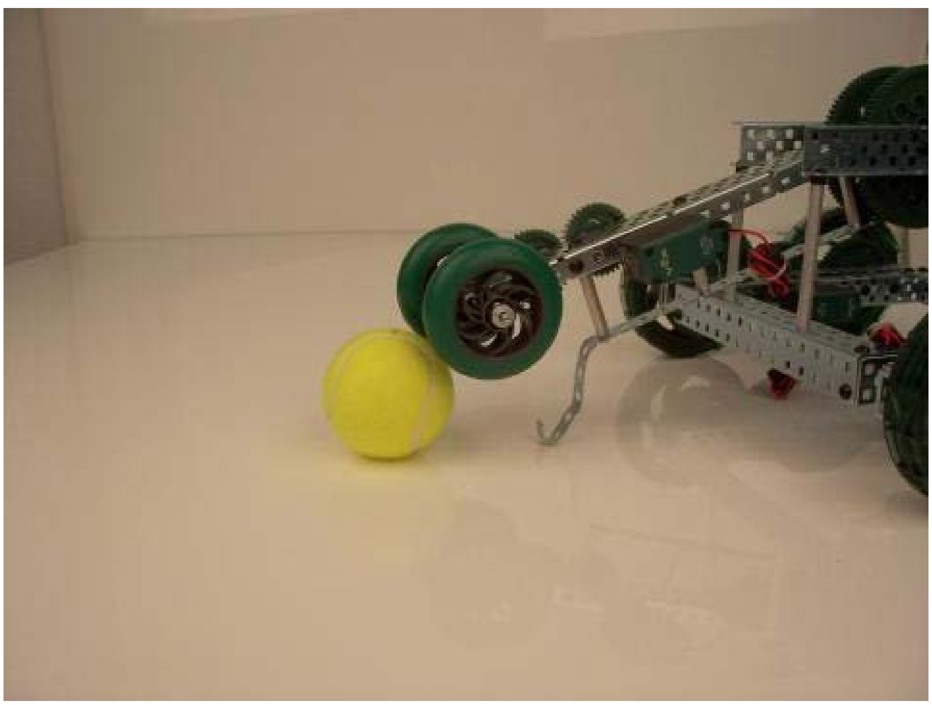
This type of manipulator consists of a stationary lower jaw with an actuated top jaw. It is a combination of a scoop and a friction grabber. The lower jaw provides upwards normal force under the object, while the top jaw pressing down provides friction-grabber type normal force which results in a frictional force that keeps the object from moving side to side.

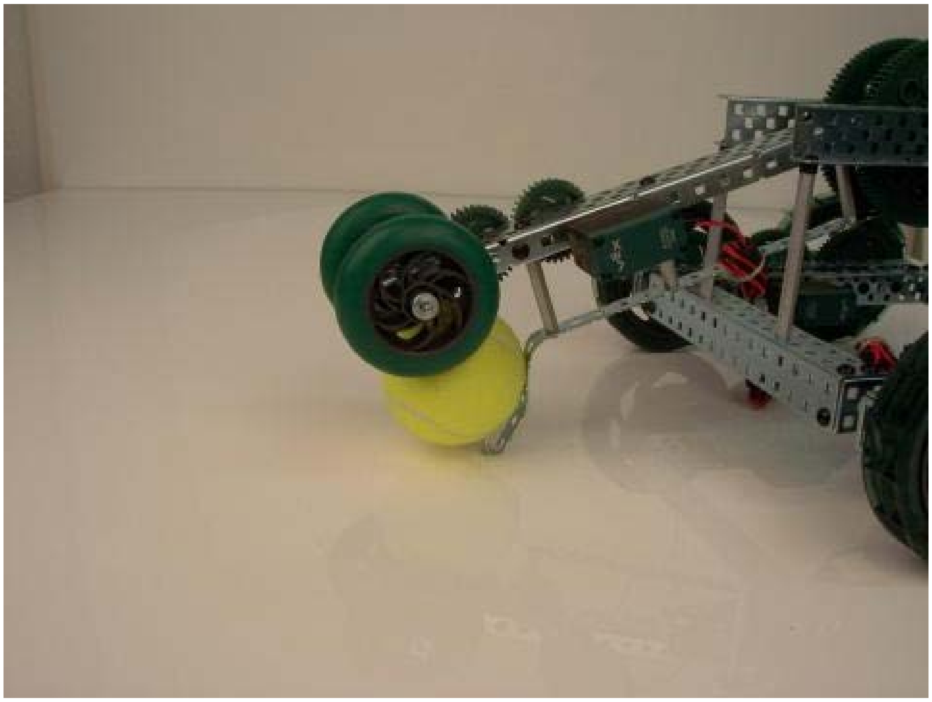




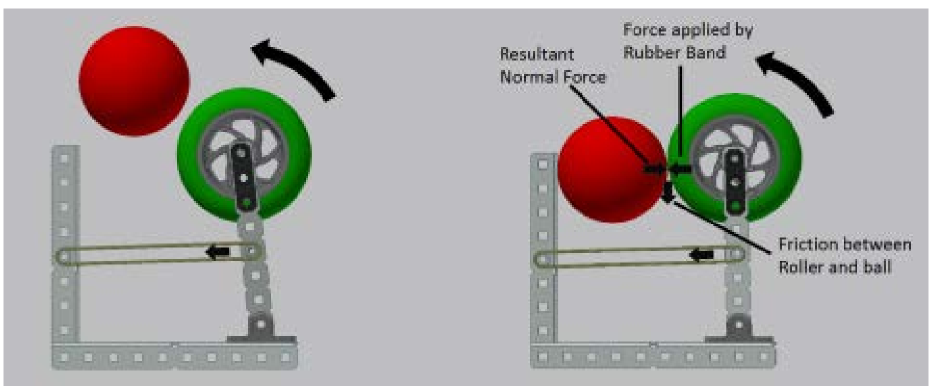
Roller Claws:

One of the most common and successful types of object manipulators used in competition robotics is the “roller claw.” This type of manipulator uses rotating wheels or rollers to “suck in” the game objects. This is an example of a friction grabber since the mechanism relies on of the friction between the rollers and the objects to hold them in place.



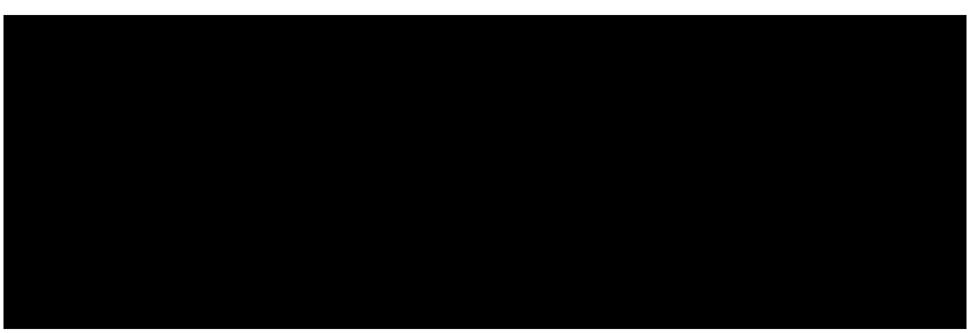


As discussed above, this claw relies on some sort of elasticity to maintain friction – either the object needs to give, or the claw does. This is an especially important consideration in a roller claw because the actuation mechanism isn't in the direction providing normal force (i.e. if the rollers are too far apart it doesn't matter how fast they spin since there is no elasticity to provide normal force on the object).



Roller Claws are extremely popular in competition robotics since they allow for very quick pickup of game objects without a requirement for fine positioning. Robot drivers just need to turn on the rollers, touch the object with the front of the roller claw, and it will suck the object in without any further control. Some designs are setup to automatically turn off the roller claw motor once the object has been grabbed (as previously described, good object manipulators don't require to be powered to maintain grip).

<u>Roller Manipulator</u>



Click here to download this video.

This video shows a roller claw mounted on a robotic arm that has been modeled and animated in Autodesk Inventor.

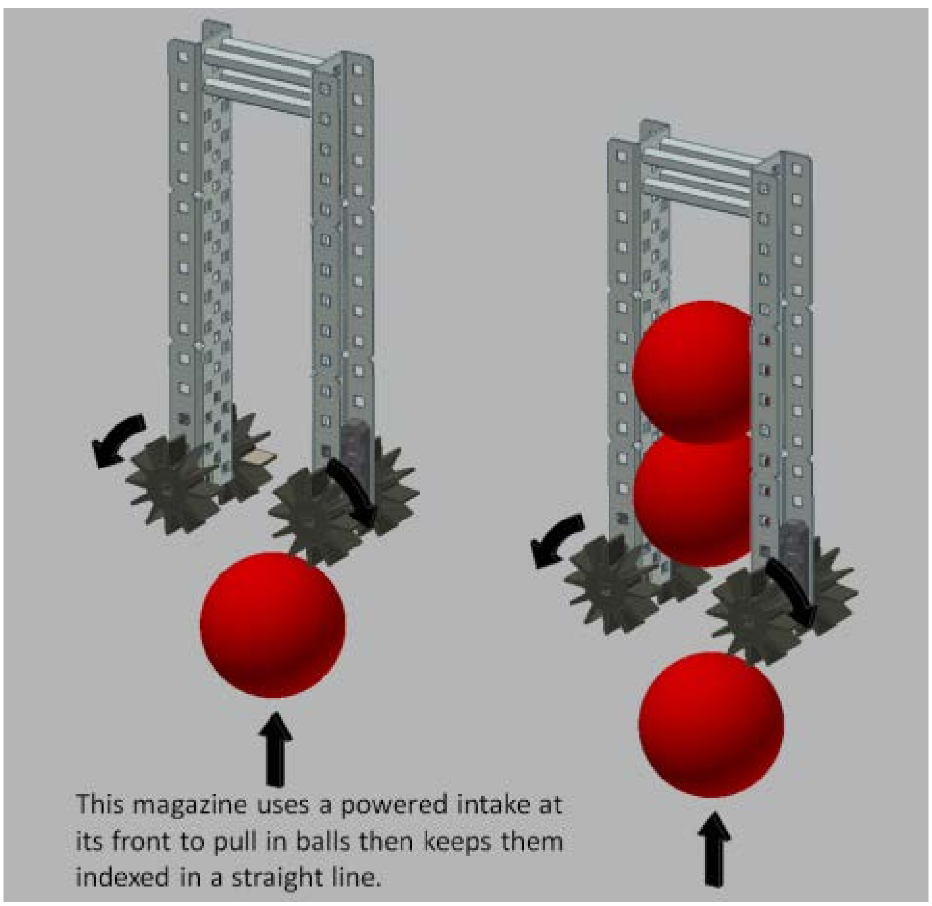
6.1: Introduction
up
6.3: Accumulators

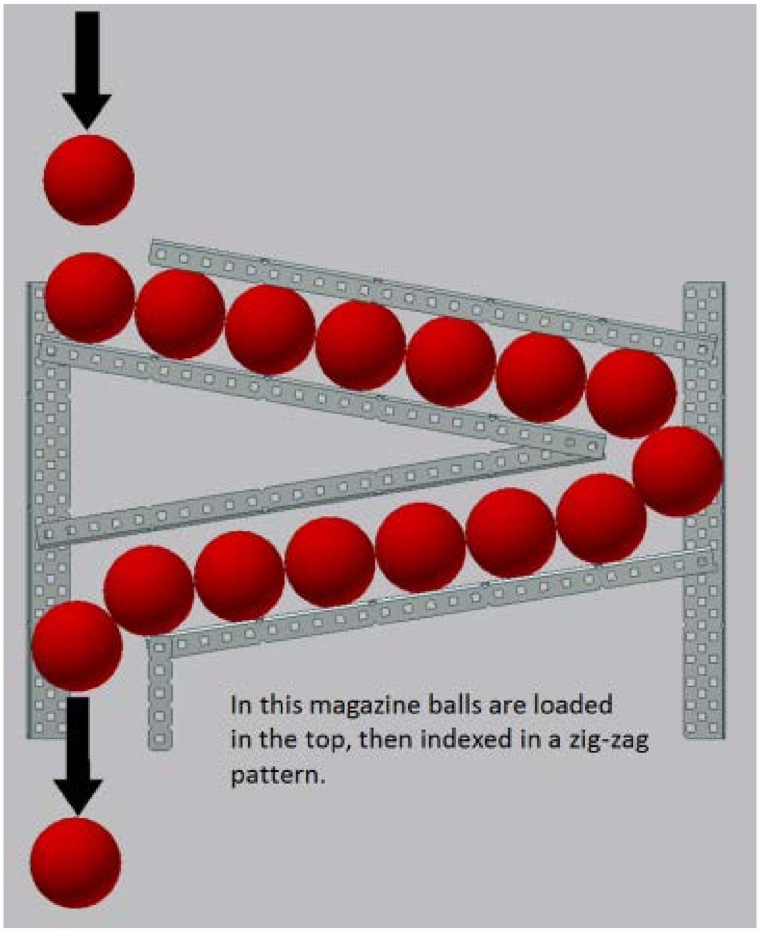
6.3: Accumulators

In competition robotics, it is often important to be able to collect multiple game objects at one time. This requires a specific type of object manipulator called an accumulator. An accumulator is a robot mechanism designed to pick up a large number of similar objects.

Magazines:

There are a few common types of accumulators. The first type is a magazine in which objects are loaded one at a time into some sort of storage area and held in a fixed orientation, in line (i.e. the first object in is the last object out). In this type of accumulator the objects are typically not actively manipulated once they are inside the storage area.

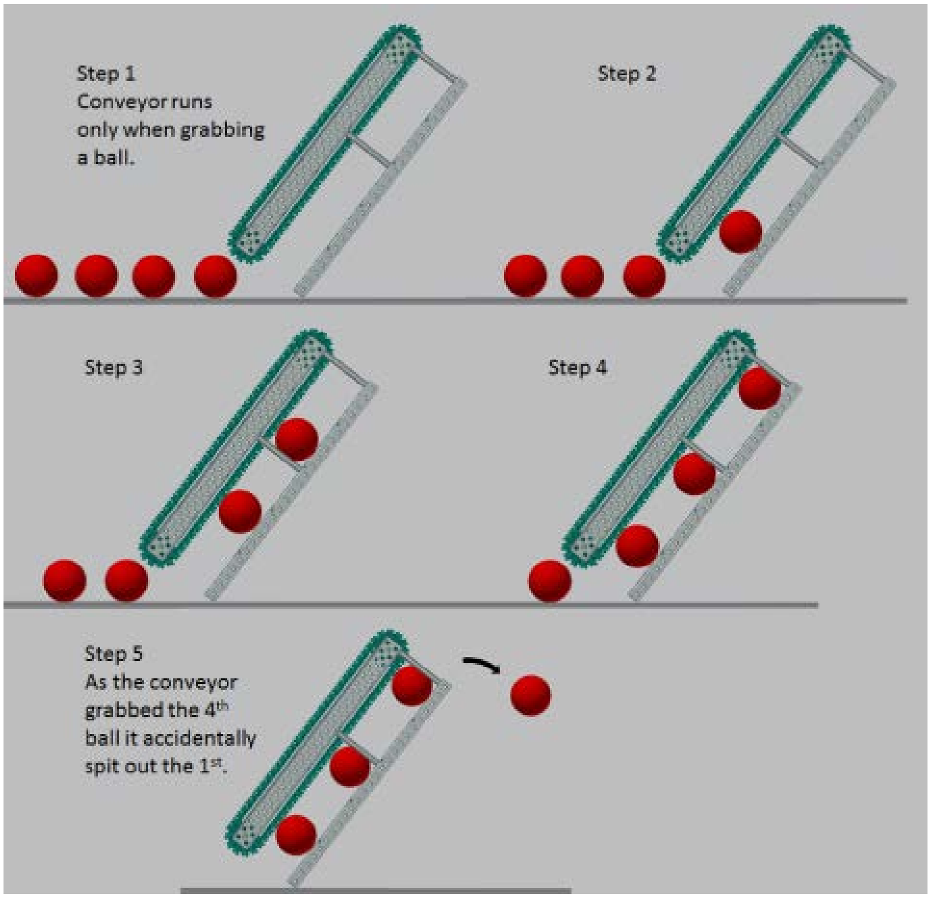




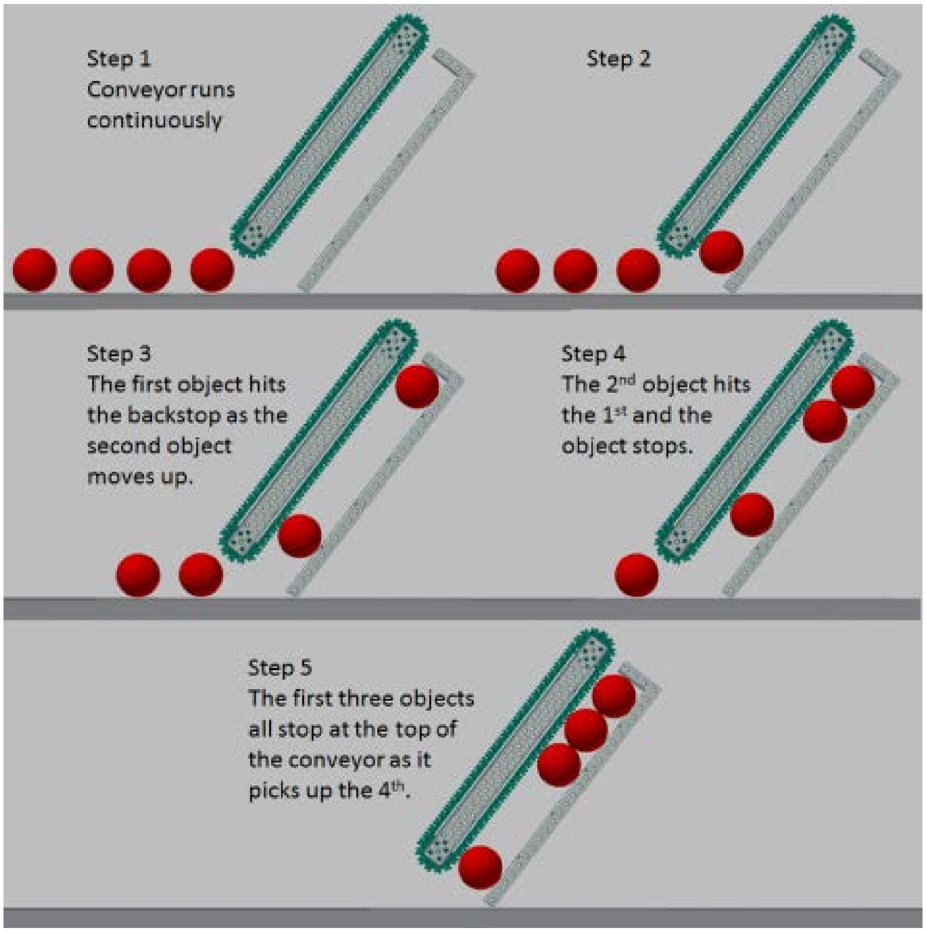
Conveyor Belts:

Another type of accumulator is a conveyor belt, which is actually a form of magazine. A conveyor belt type magazine uses some sort of belt to manipulate the objects even after they are inside. Conveyor belts are examples of frictional grabbers (they're like roller claws that hold more than one object) since they rely on the friction between the belt and object to accumulate. More details on using conveyance as part of accumulator design will be discussed later in this unit.

It is important to note that there are two main types of conveyor belts, indexing belts, in which the belt runs only when grabbing an object to enable good sorting, and non-indexing belts in which the belt continuously runs and the objects self-sort.



In the above examples the belt is run a little bit as each object is grabbed, this keeps the objects evenly spaced within the accumulator. This requires a high degree of control, either by the driver or by the robot's pre-programming and sensors. If the belt runs at the incorrect time or for too long the system can jam or may end up packing the objects very inefficiently.



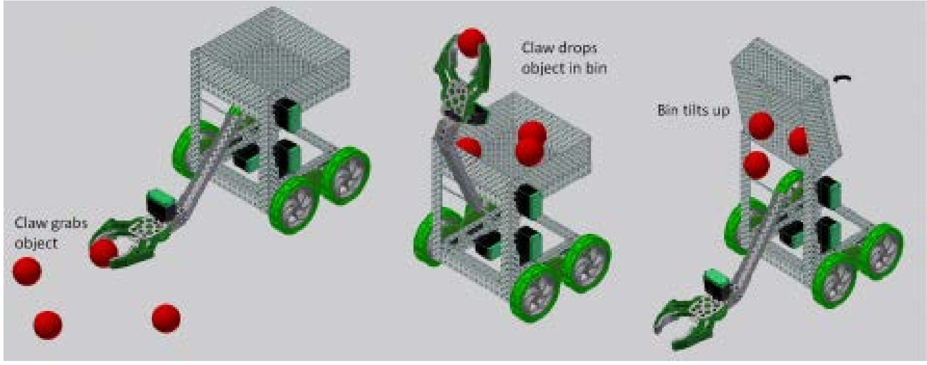
A non-indexing conveyor belt as shown above continuously runs. When the first object is grabbed it moves to the top of the belt until it hits the back of the accumulator where it hits a stop and the belts slide past it. When a second object is grabbed it moves up until it hits the first, and stops (at which point the belt now slides past both of these stationary objects). This type of accumulator is easier to use, but is more difficult to design. For this type of accumulator to function correctly the robot designer must ensure that the objects "self-sort" correctly, that is they don't bunch up into configurations where they will jam the belts. The designer must also ensure that the frictional force between the belt and the object is enough to pull the object in, but not too much so that the belt stops when the objects are resting against the stop. Once correctly configured this type of accumulator can be very successful in competition robotics.

Hoppers:

A third type of accumulator is a simple hopper, or bin. A hopper is a large storage area in which objects are placed – it differs from a magazine in that the objects are not held in any specific orientation, they're just thrown into a bin! As such designers of hopper type accumulators must be very careful to ensure the objects won't jam if they end up in the wrong orientation; some shapes of objects are more appropriate for hoppers than others.

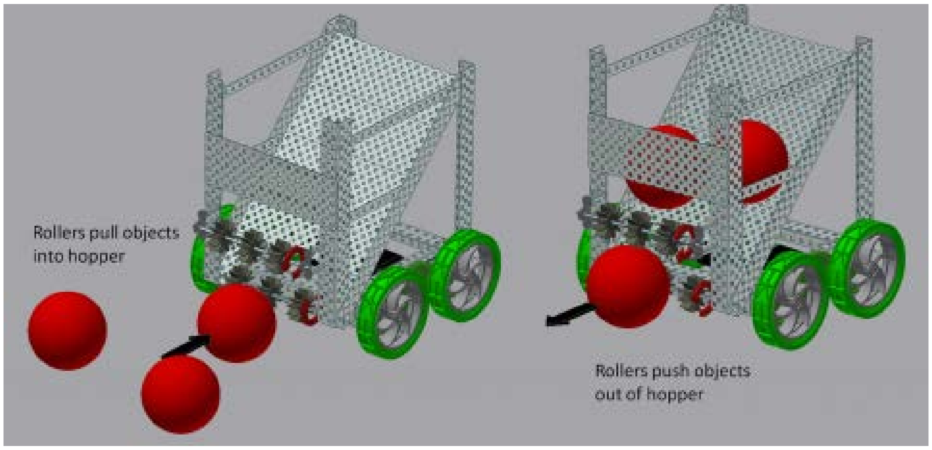
There are two types of hoppers.

Dumping hoppers are hoppers in which the way objects are released is different than the way they are gathered. One example of this is a robot can be seen below.



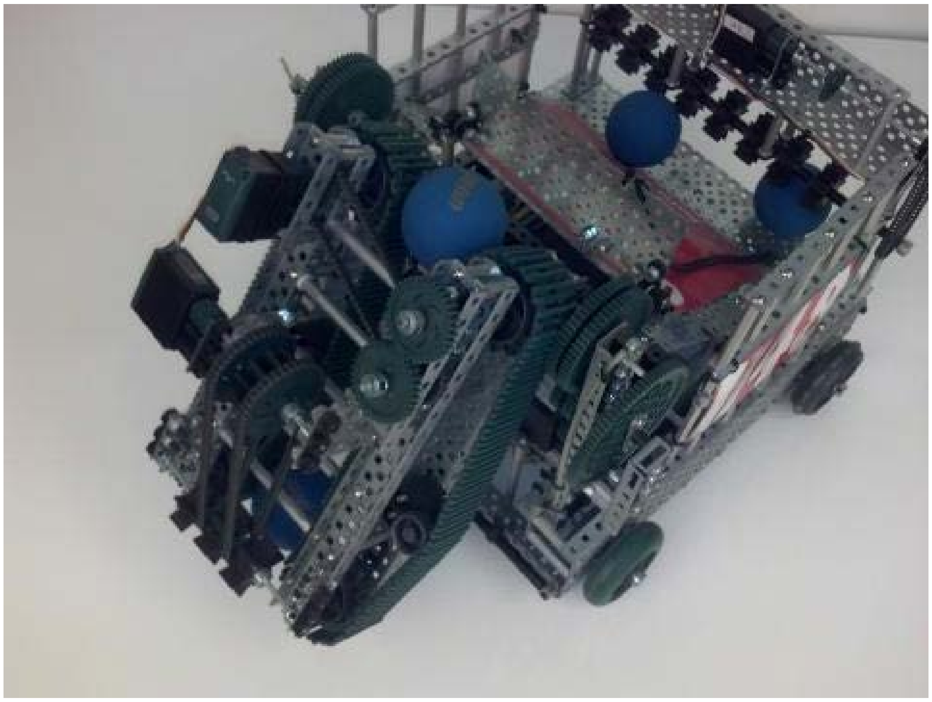
On this robot a claw arm picks up objects and dumps them into the hopper, then the hopper tilts back like a dump truck to score the objects in a goal.

Reversing hoppers are hoppers similar to magazines in which objects are released through the same mechanism with which they are gathered. An example of a reversing hopper can be seen below.



In this accumulator the objects are sucked into a large bin by two rollers. To score them the robot simply reverses the rollers and pushes the objects back out.

There are many successful designs which utilize multiple types of accumulators as part of a larger system. For example, refer to the robot below.



On the above robot, a non-indexing conveyor lifts up blue racquet balls off the floor and deposits them into a dumping hopper, where they are then dumped out the back through a separate roller mechanism into a goal.

Accumulator Design:

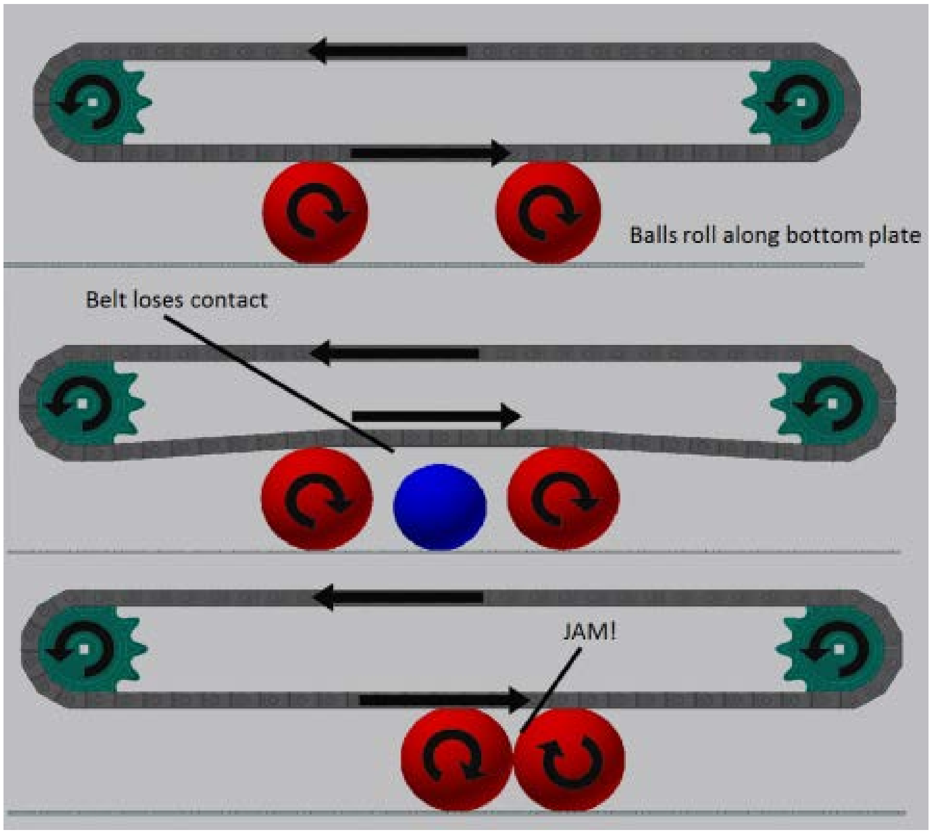
There are a number of design characteristics which are common among successful accumulators. The best competition robot accumulators will have the following characteristics:

- A wide intake "mouth" allowing for pickup over a wide area without precise robot positioning.
- A means to prevent jamming of objects after pickup.
- A high-speed intake that allows a robot to pick up an object even when it is driving at full speed.
- The ability to gather multiple objects at the same time.
- The ability to gather a large number of objects, one after another, without jamming or slowing down.
- The capability to pick up objects of various sizes.

Conveyance:

As discussed above, conveyors are very common in accumulator design and the concepts related to conveyance are important for all robot designers. There are several design characteristics of conveyor belts that are important for designers to consider.

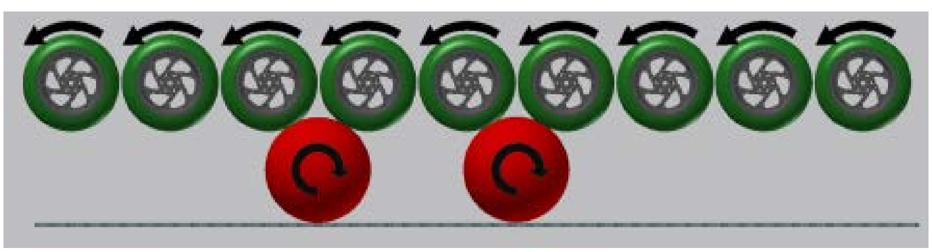
Conveyor Configuration - The configuration of the conveyor belts is a huge part of how the system behaves. There are several different types to consider. One simple configuration of this is a single conveyor belt in front of a flat wall.



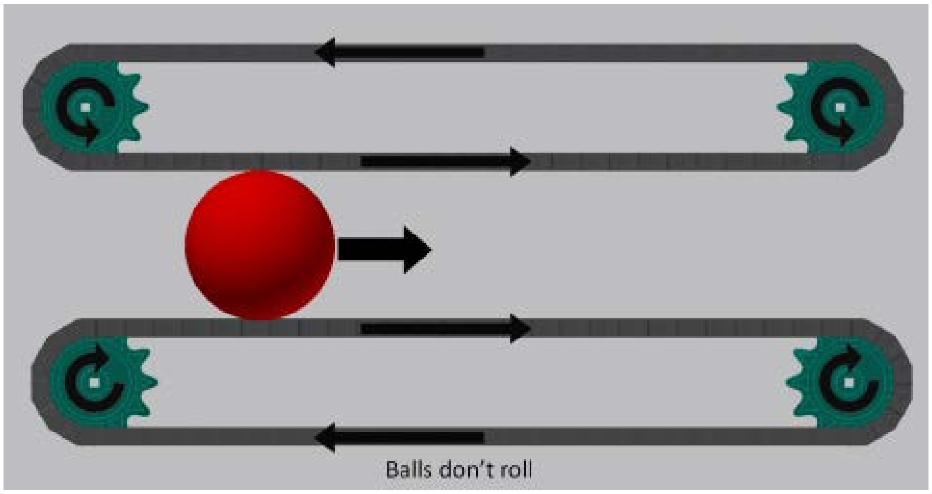
In this type of conveyor, the belt contacts the balls on one side and rolls them up the opposing surface. This relatively simple setup requires only one conveyor belt. However, this setup has some disadvantages. First, since the balls are rolling, they move through the accumulator at half the speed of the conveyor belt. Second, this setup is subject to jamming if two balls are picked up too closely together and they touch inside the conveyor. Because the back side of the ball in front is moving up while the front side of the ball in back is moving down, the balls can bind up and jam. This type of system needs to be tuned very carefully to ensure that one object doesn't push up on the belt such that it loses contact with the next object. It is difficult to ensure that the conveyor will be able to provide the correct amount of frictional force on all objects inside.

To help maintain even frictional force on all objects, a configuration with multiple independent rollers can be used.

In this setup, each of the rollers is linked to the power source. This system is less likely to jam, but is still subject to many of the problems found in the first setup. Also, using many rollers instead of a single belt adds significant complexity.

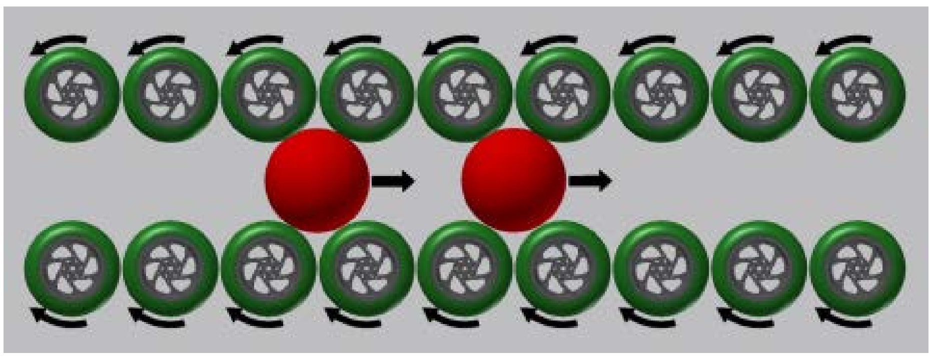


Another option is to utilize two belts.



This setup is much less likely to jam since the balls no longer roll; they more in a straight line up the conveyor. If two objects do touch they are much less likely to jam. This system is still susceptible to the varying friction of objects pushing the belts apart found in the first single-belt system above, but it performs much better at the expense of having two belts.

Why not combine the best of system 2 and system 3? What about a system with rollers on both sides?



This system should prevent jamming like the double belt setup, and should also provide consistent frictional force on the objects like the single sided roller setup, however – it is MUCH more complex than the above systems.

Design is all about tradeoffs: what is the simplest solution that will solve the problem? Designers will need to choose the configuration which best works to accumulate their specific object, in the manner most relevant to their competition match needs (i.e. maybe the robot only need to pick them up one at a time, so jamming isn't an issue).

Conveyor Gearing - Another important conveyance design consideration is the conveyor gearing. It is important to gear the accumulator appropriately. Ideally, the accumulator intake is geared so that it pulls an object in faster than the drivetrain at maximum speed. In a single-belt system, this means that the intake is geared in such a way that the linear belt speed is more than double the drivetrain's top speed. In a two-belt system, the intake's linear belt speed only needs to be more than the drivetrain's top speed. When it comes to accumulator gearing, faster is almost always better – just make sure the accumulator can overcome the friction caused by pulling in the objects. More details on this will be discussed in Units 7 & 8.

Compression & Elasticity – For belts and rollers to pull in an object, there must be some force pressing the belt onto the object. Often this force is caused by the compression or elasticity of some part of the system. Sometimes the conveyor belting bends backwards and this "spring" is what applies the force on the object. Other times, the object itself has some elasticity and deforms when it is sucked into the intake. Other times, additional elastic bands or springs are used to give the entire conveyor assembly some give, which enables it to deform when an object moves through it. In this case, the springs apply the normal force on the object. Finding the correct balance of grip on an object is sometimes difficult, especially when building an accumulator designed to pick up multiple objects at the same time – as discussed above, the conveyor configuration will play a large role in this.

[6.2: Manipulators] [up] [6.4: Design Considerations]

6.4: Design Considerations

When deciding what type of object manipulator or accumulator to use on a competition robot, a designer has a number of specific considerations.

The most important thing is to keep the specific scoring object in mind. It is easy to look at examples and think they will work, but every object has its own unique challenges and requirements. Designers need to focus on the object in question, and design accordingly. Designers can model and simulate design ideas using 3D design software such as Autodesk Inventor prior to building prototypes in order to save valuable build time. Prototypes and experiments will be very helpful as designers learn about the nature of an unknown game object.

Designers need to focus on how the object manipulator will be used in the real world (during a competition match) and take this into account. Some common questions to consider:

- What orientation will the object be in when picked up?
- Does the gripper need to be able to grasp the object from multiple orientations?
- How will the gripper deposit the object?
- Does it need to deposit the object in multiple orientations?
- What orientation change does the object need to make between pickup and deposit?

Any object manipulator or accumulator should maintain a firm grip on the object at all times during possession.

They should be able to quickly and efficiently complete their tasks with minimum wasted motion. Effective designs focus on speed and efficiency and, if possible, will not require precise positioning. As soon as the robot touches the object, it should grab it.

The robot should be designed to allow as little time to elapse as possible between picking up the object and scoring it. Speed & Efficiency are critical!

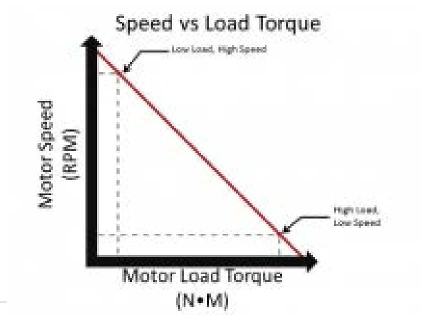
6.3: Accumulators

up

6.5: Modeling an Object Manipulator

Unit 7: Speed, Power, Torque & DC Motors

In this unit, students will learn about the physical principles of speed, power, and torque. They will also learn about DC motors and how these principles apply to them. Students will apply these concepts on a sample mechanical system to calculate key details of the design.



[6.7: Engineering Notebook](#) [up](#) [7.1: Introduction](#)

7.1: Introduction

Engineering is problem solving. Specifically, engineering is the application of practical and scientific knowledge to a methodical problem solving process. Unit 1 described a commonly used design process, but only briefly touched on the types of knowledge engineers need to utilize it effectively. This unit will provide budding robot designers with a beginners look at two topics helpful for solving problems in this field. To get more in-depth information on these topics, students should consider pursuing higher education in an engineering or science field.

Unit 7: Speed, Power, Torque & DC Motors

up

7.2: Classical Mechanics

7.2: Classical Mechanics

The field of Classical Mechanics deals with the study of bodies in motion, specifically the physical laws that govern bodies under the influence of forces. Much of the mechanical aspects of robotics design are heavily tied to the principles of this field. This unit will describe a few key concepts of Classical Mechanics that are particularly applicable.

SPEED – A measure of how fast an object is moving. Describes a change in position with time (or more simply put, how far an object will travel over a given period of time.) This measure is given in units of distance per time (i.e. miles per hour, or feet per second.)

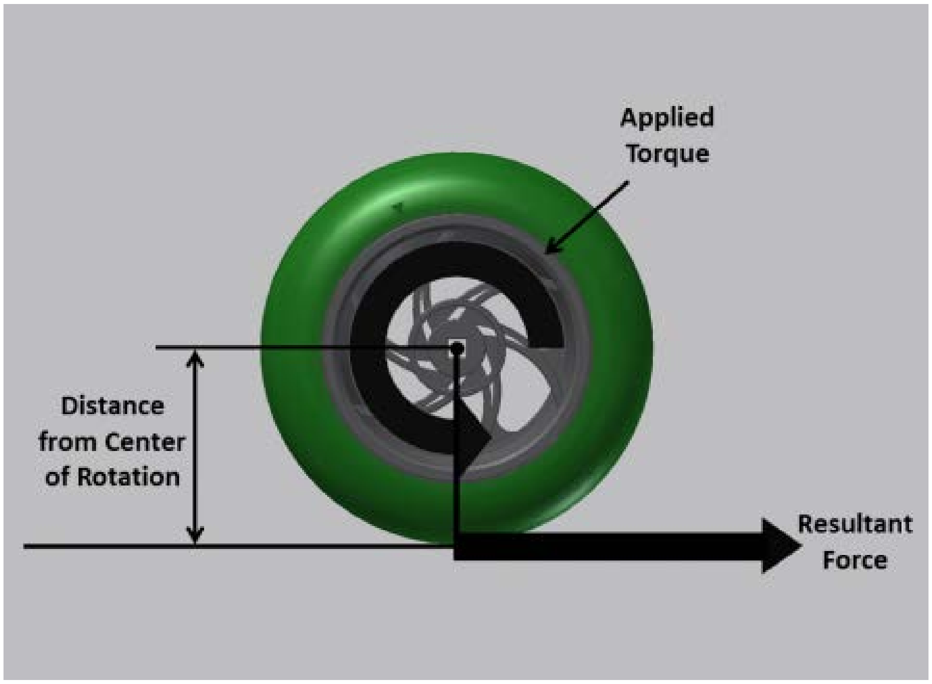
ROTATIONAL SPEED – Speed can also be expressed rotationally. This refers to how fast something is moving in a circle. It is measured in units of angular-distance per time (i.e. degrees per second) or rotational cycles per time (i.e. revolutions per minute.) When someone talks about "RPM" they are referencing rotational speed. When talking about the RPM of a car engine, one is describing how fast the engine is spinning.

ACCELERATION – A change in speed over a period of time is described as acceleration; the higher the acceleration the faster the change in speed. If a car goes from 0 miles per hour to 60 miles per hour in 2 seconds, it is a higher acceleration than if the car goes from 0 MPH to 40 MPH in 2 seconds. Acceleration is a rate of change of speed. No change means no acceleration – if something is moving at constant speed it is not accelerating.

FORCE - Accelerations are caused by forces; they are influences that cause a change of movement, direction or shape. When one presses on an object, they are exerting a force on it. When a robot is accelerating, it does so because of the force its wheels exert on the floor. Force is measured in units such as Pounds or Newtons.

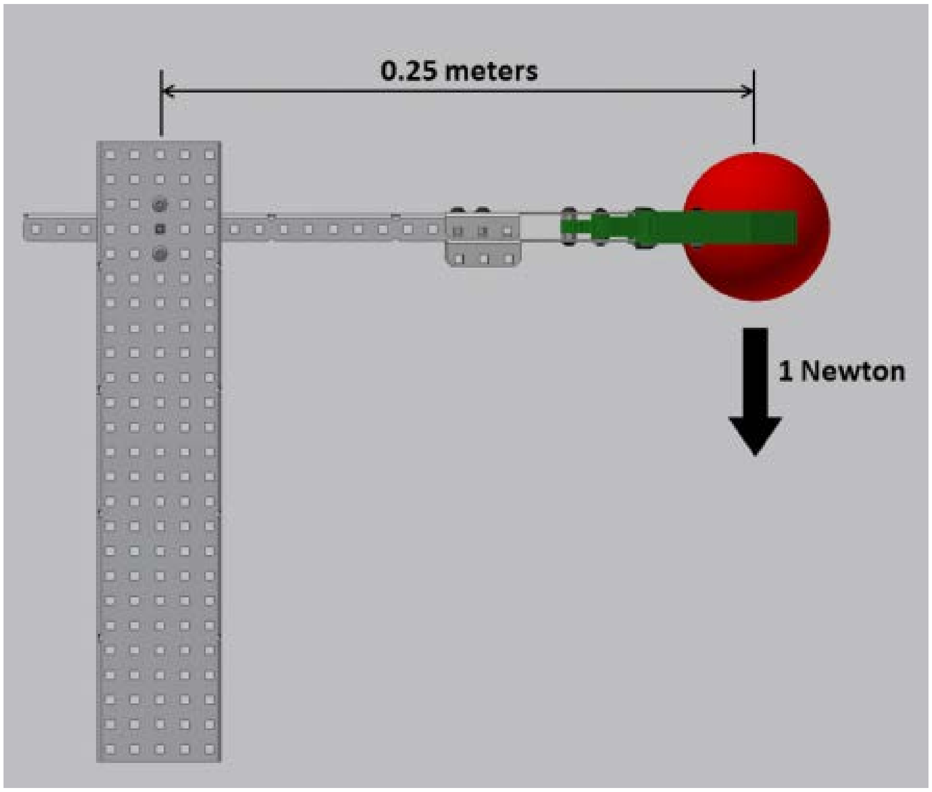
For instance, the weight of an object is the force on the object due to gravity (accelerating the object towards the center of the earth.)

TORQUE – Force directed in a circle (rotating an object) is known as torque. Torque is a spinning force. If torque is spinning an object, the object will create a linear force at its edge. In the instance of a wheel spinning on the ground, the torque applied to the wheel axle creates a linear force at the edge of the tire where it contacts the ground. This is how one defines torque, a linear force at the edge of a circle. Torque is described by the magnitude of the force multiplied by the distance it is from the center of rotation (Force x Distance = Torque). Torque is measured in units of force*distance, such as Inch-Pounds of Newton-Meters.



In the above example of a wheel on a surface, if we know how much torque is applied to an axle with a wheel on it, we can find out how much force the wheel is applying on the floor. In this case, the wheel radius would be the distance the force is from the center of rotation.

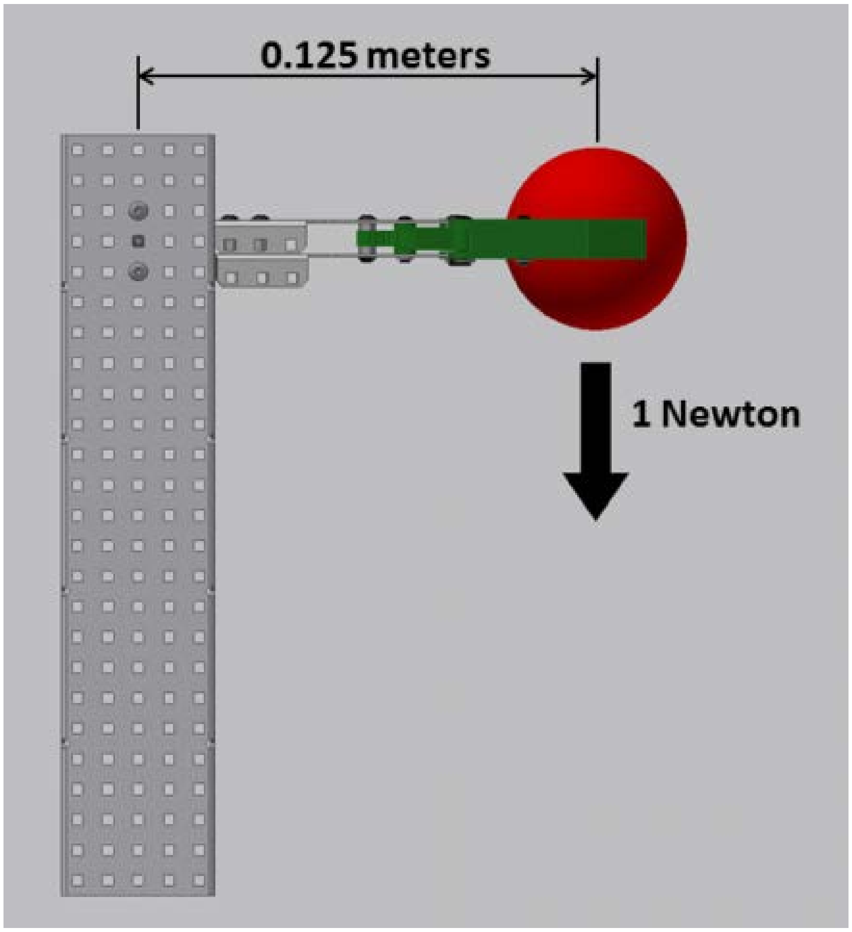
Force = Torque / Wheel Radius



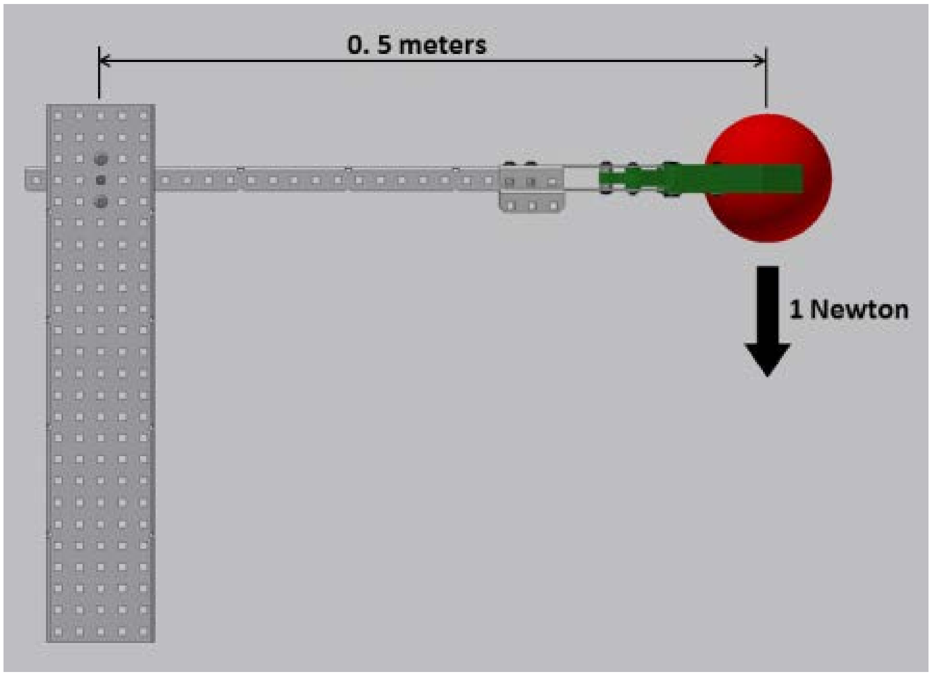
In the above example of a robot arm holding an object, we can calculate the torque required to lift the object. Since the object has a weight of 1 Newton, and the arm is 0.25 meter long (the object is 0.25 meters from the center of rotation), then

Torque = Force x Distance = 1 Newton x 0.25 Meter = 0.25 Newton-Meters.

This means the torque required to hold the object stationary is 0.25 Newton-Meters. In order to move it upwards the robot needs to apply MORE torque than 0.25 Newton-Meters to overcome gravity. The more torque the robot has, the more force it exerts on the object, the greater the acceleration on the object, and the faster the arm will lift it up.



Example 7.2



Example 7.3

For these examples, we can calculate the torque required to lift these objects as well.

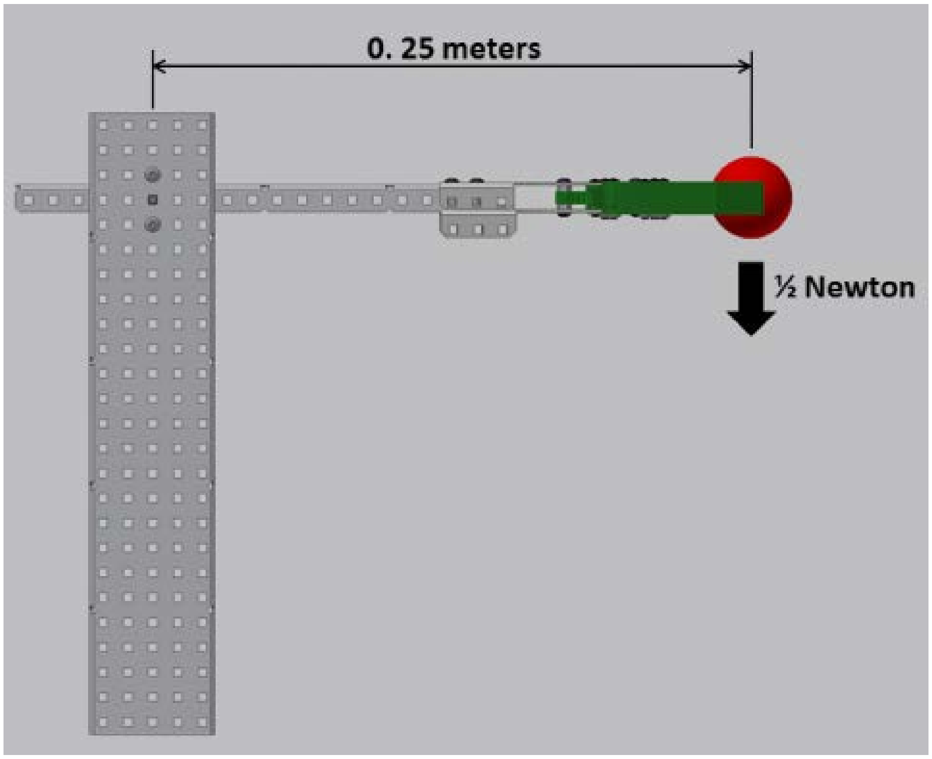
Example 7.2 – Torque = Force x Distance = 1 Newton x 0.125 meters= 0.125 Newton-Meters.

In this example, the arm is half the length of Example 1, and the torque required is also half. The arm length is proportional to the torque required; shorter arms require less torque to lift the same object as longer arms.

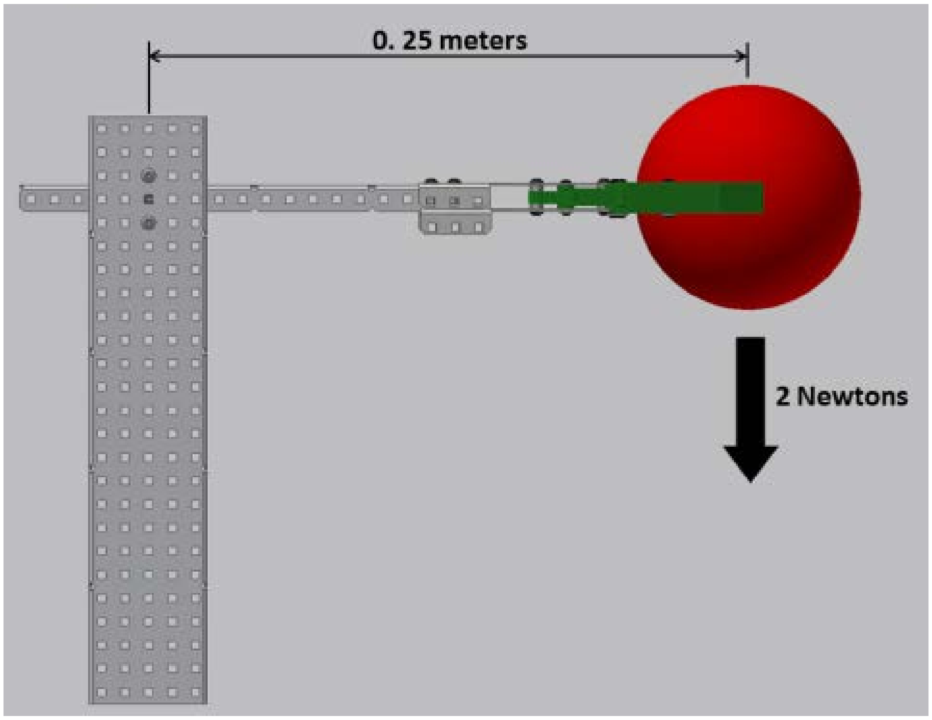
Example 7.3 – Torque = Force * Distance = 1 Newton x 0.5 meters = 0.5 Newton-Meters.

In this example, the arm length is double the length of Example 1, and the torque required is also double.

Another way to think of this is if a robot has limited torque at its arm joint, a shorter arm will be able to lift more weight than a long arm (but it can't lift it as high.)



Example 7.4



Example 7.5

These examples have the same robot arm lifting objects of different weights. How does this affect the required arm torque?

Example 4 – Torque = Force x Distance = ½ Newton x 0.25 meters = 0.125 Newton-Meters

Example 5 – Torque = Force x Distance = 2 Newton x 0.25 meters = 0.5 Newton-Meters

In these examples, as the weight of the object goes down, the torque required goes down as well. The applied weight is proportional to the torque required to lift it; heavier objects require more torque to lift.

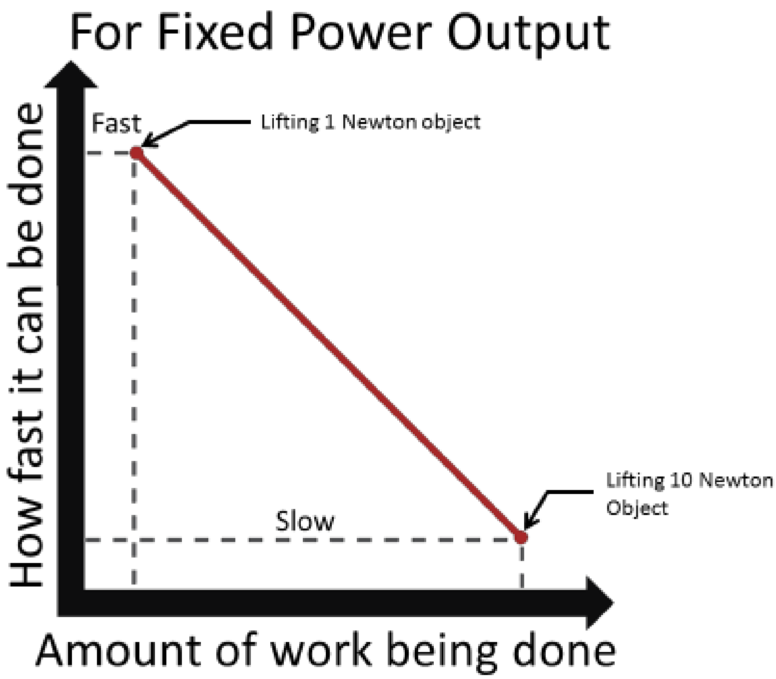
Budding robotics designers should note these key relationships between torque, arm length, and object weight.

WORK – The measure of force exerted over a distance is referred to as work. For instance, say it takes 10 pounds of force to hold an object. It would then take a certain amount of work to lift this object 10 inches, and it would take double that work to lift it 20 inches. Work can also be thought of as a change in energy.

POWER - Most people are more familiar with Power as an electrical term, but it is part of mechanical physics as well.

Power is defined as the RATE that work is performed. How fast can someone do their work?

In robotics design it is handy to think of power as a limit, since competition robotic systems are limited in the amount of power they can output. If a robot needs to lift a 2 Newton weight (exerting a 2 Newton force) the amount of power the robot can output limits how FAST (the rate) at which the robot can lift it. If the robot is capable of outputting lots of power, it will be able to lift it quickly. If it can only output a small amount of power, it will lift it slowly (or not at all!)



Power is defined as Force multiplied by Velocity (how fast one can push with a constant force), and is frequently expressed in units of Watts.

Power [Watts] = Force [Newtons] x Velocity [Meters / Second]

1 Watt = 1 (Newton x Meter) / Second

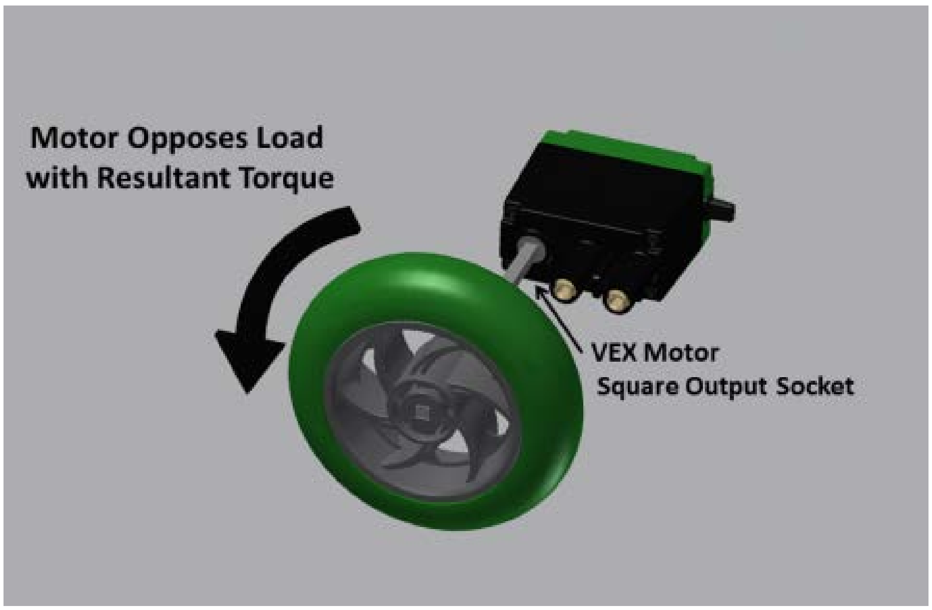
So how does this apply to competition robotics? Robot designers have certain power limitations they must remain below. Competition robotics designers using the VEX Robotics Design also need to consider the physical limitations presented by the motors. These motors have limited power, and they can only do so much work, so fast.

Note: these are basic descriptions of very advanced concepts. For more in-depth discussions of these physical properties, students should pursue higher education in the STEM fields.

[7.1: Introduction](#) up [7.3: DC Motors](#)

7.3: DC Motors

Actuators are mechanisms used to act upon an environment, usually for moving or controlling a mechanism or system. Actuators drive everything that moves on a competition robot. The most common type of actuator in this application is a motor; in particular, VEX Robots utilize Direct Current (DC) Motors.



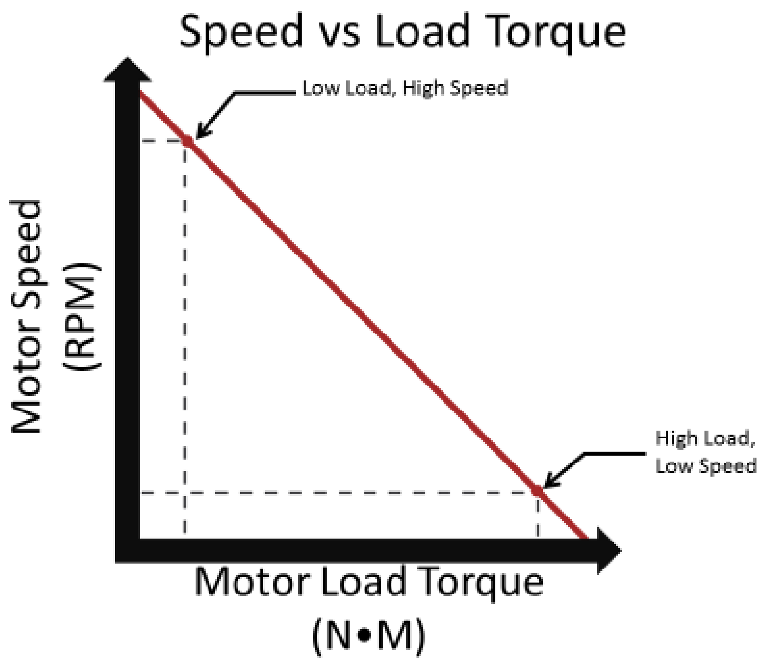
Motors convert electrical energy into mechanical energy through the use of electro-magnetic fields, and rotating wire coils. When a voltage is applied to a motor it outputs a fixed amount of mechanical power. The mechanical power is seen as the motor's output (usually some shaft, socket, or gear), spinning at some speed with some amount of torque.

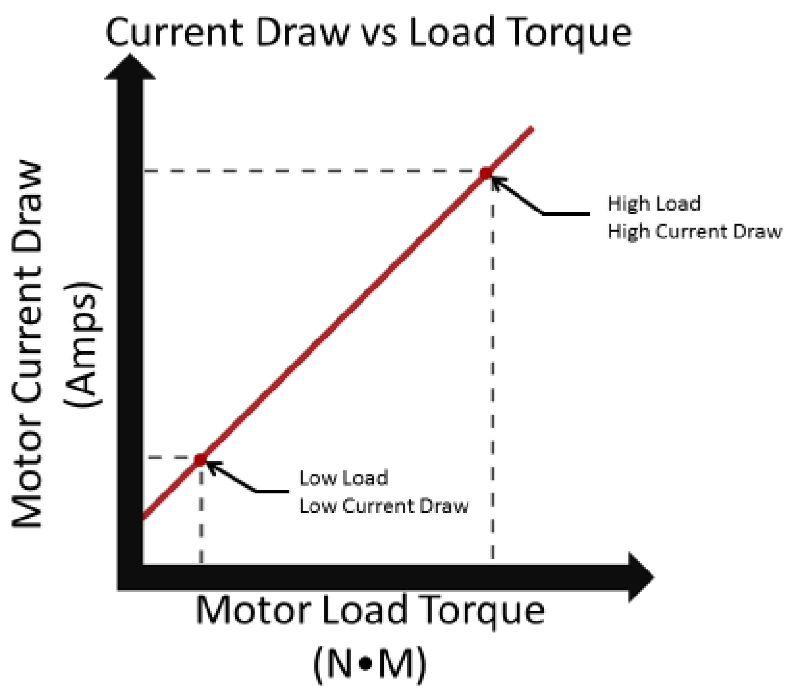
Motor Loading

Motors only apply torque in response to loading. Ideally, with no loading on the output the motor will spin very, very fast with no torque. This never happens in real life, since there is always friction in the motor system acting as a load and requiring the motor to output torque to overcome it. The higher the load placed on the motor output, the more the motor will "fight back" with an opposing torque. However, since the motor outputs a fixed amount of power, the more torque the motor outputs, the slower its rotational speed. The more work one makes the motor do, the slower it spins. If one keeps increasing the load on the motor, eventually the load overcomes the motor and it stops spinning. This is called a STALL.

Current Draw

The motor draws a certain amount of electrical current (expressed in units of amps) depending on how much load is placed on it. As the load increases on the motor, the more torque the motor outputs to overcome it and the more current the motor draws.





As seen in the graphs above, current and torque load are proportional. More torque load means more current draw, but current and rotational speed are inverse. The faster the motor spins, the less current it draws.

The “key” Motor Characteristics

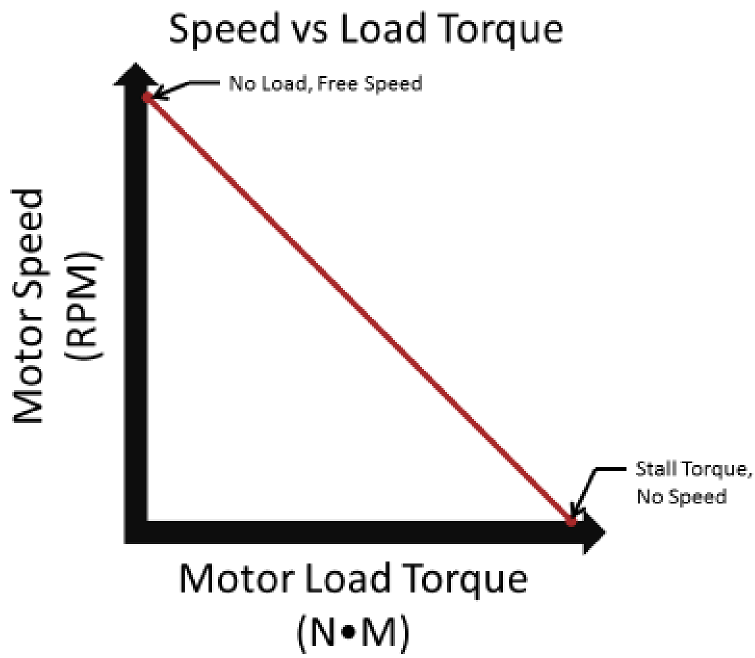
All motors are different, and their properties vary depending on their type, configuration, and manufacture. There are four main characteristics that govern the types of DC motors used commonly in competition robotics.

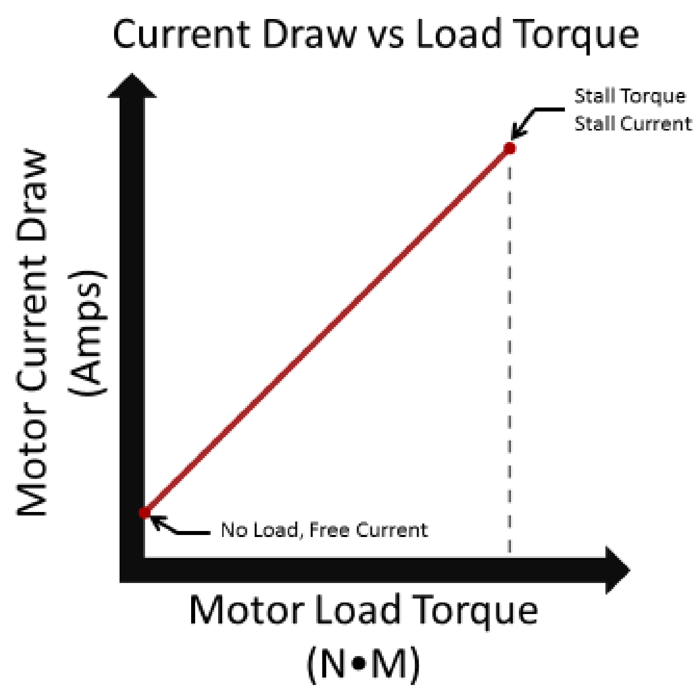
Stall Torque (N-m) – The amount of load placed on a motor that will cause it to stop moving.

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

Stall Current (Amp) – the amount of current a motor will draw when it is stalled.

Free Current (Amp) – The amount of current a motor will draw when it is under no load.





Based on the above relationships, one can see how the concept of power comes into play. With a given loading, the motor can only spin at a certain speed.

Since the relationships shown above are linear and proportional, it is a simple matter of plotting the torque-speed and torque-current graphs for any motor by experimentally determining two points on each graph.

Varying Power with Voltage

The Power output of a DC Motor varies with the voltage applied. This means that the more voltage is applied, the more power is available and the faster the motor can do work.

If a motor is under a fixed amount of load, and the voltage is increased (resulting in an increase of power), what will it do? It will spin faster! There is more power available to accomplish the same amount of work.

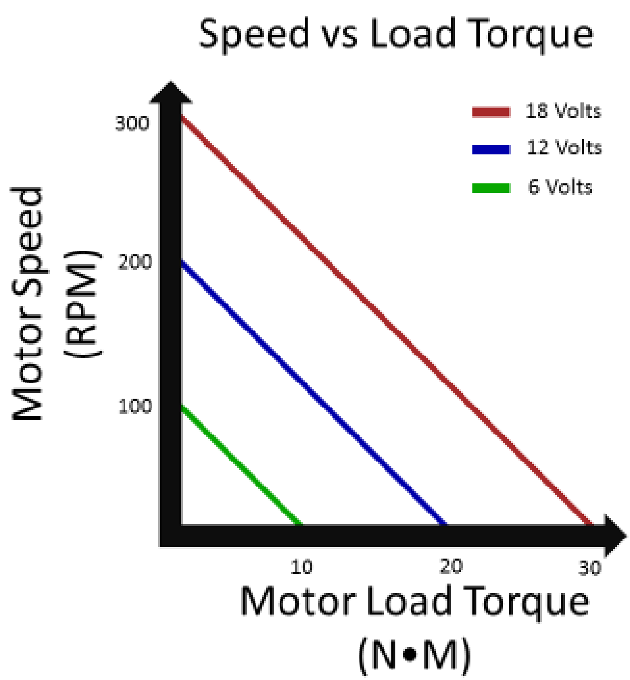
This means that the above motor characteristics change depending on the Voltage applied to the motor, and as such they need to be listed at a given Voltage (i.e. “Tested at 12V”.) In fact, the four characteristics above vary proportionally with applied voltage. For example, if one knows a motor has a free speed of 50 RPM at 6V, if the voltage is doubled to 12V the free speed doubles also, and can be calculated to be 100 RPM.

One can calculate the value of one of these characteristics at a specific voltage if one knows the characteristic at another voltage by multiplying the known value by the ratio between the voltages. Note, this does not apply to the motor's Free Current, which is the same at any voltage.

New Value = Spec Value x (New Voltage / Spec Voltage)

For instance, in the example described above a motor’s free speed is specified at 50 RPM at 6V. The designer is planning to run the motor at 8V – what will be the free speed at this voltage?

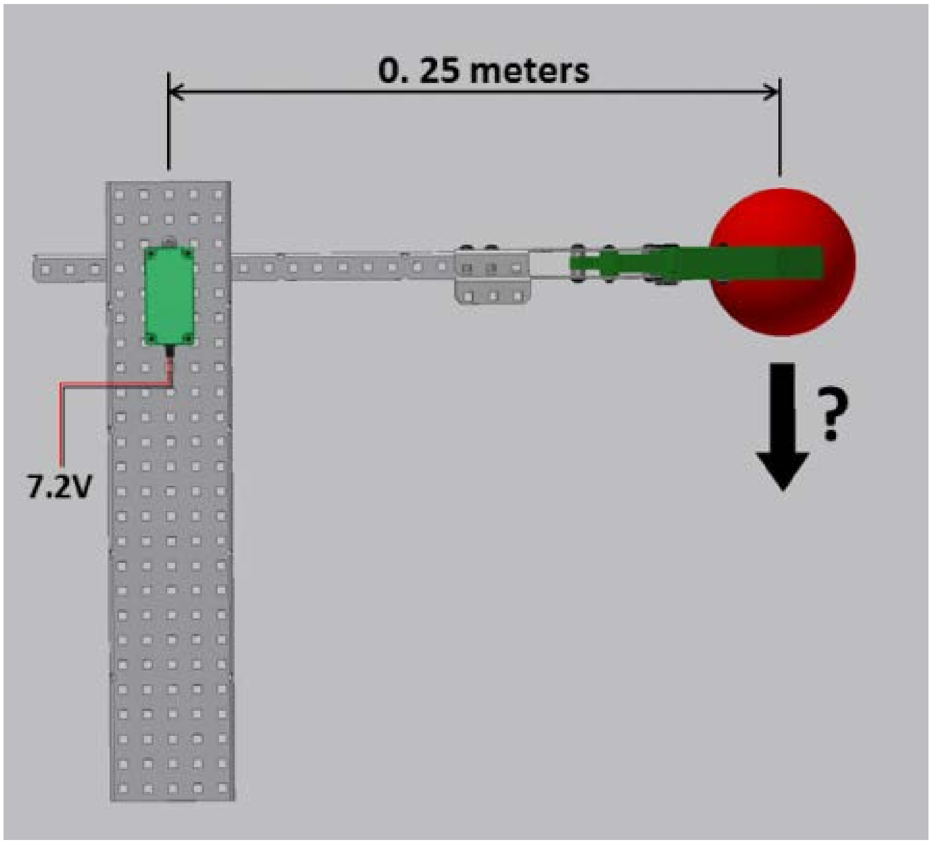
Free Speed @ 8V = Free Speed @ 6V x (8V / 6V) = 50RPM x (8/6) = 66.66 RPM



So what implications does varying voltage have for control? The motors on a robot aren’t just on / off devices. A robot designer can vary the voltage going to the motor under load to get different amounts of power, and varying speed. This is done using electrical devices known as motor controllers, which regulate the voltage supplied to the motors.

Motor Limits & Calculations

Does this mean a designer can keep adding voltage to a motor until it is capable of outputting the power needed for an application? Not quite, because motors have limits. At some point the power will be too much for the motor’s electrical windings to handle and it will fail (usually, with a puff of white smoke.) Luckily, this doesn’t happen with VEX motors – they all have internal thermal breakers that will cut the current being delivered to the motor if they get too hot. This is good because it means the motors won't burn out, but it presents a new constraint for designers: making sure the motors don't trip their breakers! How do designers do this? By designing their systems so that the motors don't draw more current than a specified amount by limiting the amount of load that will be applied.



Sample Motor Specs
Tested at 7.2 Volts

Free Speed	100 RPM
Stall Torque	1.0 N•M
Free Current	0.1 Amps
Stall Current	3.0 Amps

Arm Load Calculation

In the above example, a known motor at a known voltage is driving a known robot arm. In this scenario, what is the maximum weight the robot can hold stationary?

To solve this problem, a designer must understand that the maximum weight the robot can hold stationary occurs at the stall torque of the motor. If the motor is stalled, it is applying a torque of 1 N-m on the robot arm, which is 0.25 meters long. Torque = Force * Distance

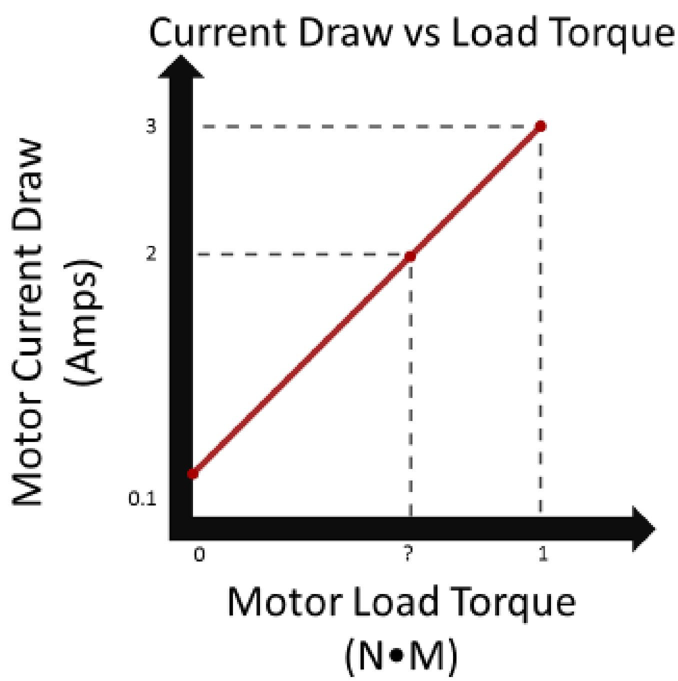
Force = Torque / Distance = 1 Newton-meter / 0.25 meters = 4 Newtons

The arm can hold up 4 Newtons at motor stall. Any more and the arm will drop down.

Torque Load Calculated from Current Limit:

This is simple, but things get more complicated when one needs to consider a current limit. Say for instance in the above example a current limiting breaker is present in the motor that will trip if the motor draws more than 2 Amps. What is the maximum weight the robot can hold without tripping this breaker?

Now, the motor is not operating at stall torque – if the motor is at stall it will draw the stall current of 3 amps and trip the circuit breaker. The designer needs to figure out what torque load the motor must be under to maintain current draw less than 2 amps. How does one go about doing this?



Looking at the above graph and remembering that the relationships are linear, an equation can be derived to calculate the torque loading at any specified current draw.

The equation for a line is y = mx + b where y is the value in the y-axis, x is the value in the x-axis, m is the slope of the line, and b is where the line intersects the y-axis (y-intercept).

The slope of the line can be expressed as m = (Change in Y / Change in X) = (Stall Current – Free Current) / Stall Torque

The Y-Intercept is the free current.

The Y value is the current at a given point on the line and the X value is the torque loading at that point.

The equation can be expressed as:

Current = ((Stall Current – Free Current) / Stall Torque) x Torque Load + Free Current

Rearranging to solve for Torque Load:

Torque Load = (Current – Free Current) x Stall Torque / (Stall Current – Free Current)

Plugging the parameters of the above example in, one can solve for the torque load which results in a current draw of 2 Amps.

Torque Load = (2 Amps - .1 Amps) x 1 N-m / (3 Amps - .1 Amps)

Torque Load = (1.9 Amps) x 10 N-m / (2.9 Amps)

Torque Load = .655 N-m

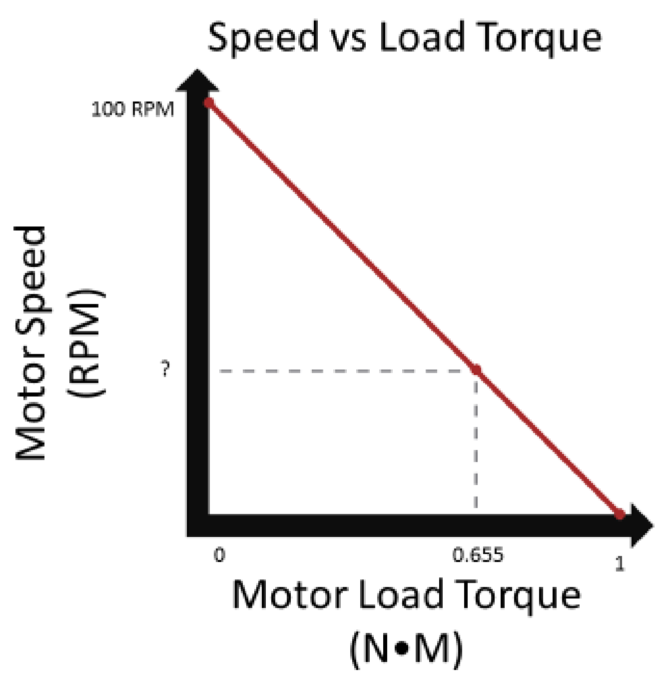
Based on this calculation, the designer knows that if the torque applied to the motor is above 0.655 N-m, the motor will exceed 2 amps of current draw, and the breaker will trip. All that is left is to calculate how much force can be placed on the arm.

Force = Torque / Distance = 0.655 N-m / 0.25 m = 2.62 N

If the robot arm picks up an object heavier than 2.62 N, it will trip the motor circuit breaker.

Motor Speed from Torque Load Calculation

In the above example, how fast does the motor move when it is at the current limit? Based on the calculation in the previous step, the designer needs to determine the motor speed at a load of 0.655 N-m.



Looking at the above graph, it is possible to derive an equation to calculate the motor speed at any given torque load in a way similar to the current draw equation from the last example.

In this case, the slope of the line is expressed as m = (Change in Y) / (Change in X) = -(Free Speed) / (Stall Torque).

Note: the slope is negative.

The Y-Intercept is the Free Speed

The Y value is the speed at a given point on the line, and the X value is the load torque at that point.

The equation is expressed as:

Speed = -(Free Speed / Stall Torque) x Torque Load + Free Speed

Plugging the parameters of the above example in, one can solve for the speed of the motor at a torque load of 6.55 in-lbs:

Speed = -(100 RPM / 1 N-m) x0.655 N-m + 100 RPM

Speed = -(100 RPM/N-m) x 0.655 N-m + 100 RPM

Speed = -65.5 RPM + 100 RPM = 34.5 RPM

The motor will spin at 34.5 RPM when it is under a torque load of 0.655 N-m, while drawing 2 amps and lifting an object weighing 2.62 N.

Multiple Motors

When an application requires more power than a motor can handle, a designer has three options:

1. Deal with it, and change the requirements of the problem so that lower power is acceptable.
2. Switch to a more powerful motor.
3. Use more than one motor in the application.

What happens when multiple motors are used on one application? Simple – they balance the torque load between them. If 2 N-m of torque is applied, each motor will have a torque load of 1 N-m, and react accordingly.

A simple way to think about it is that the motors take on the characteristics of one super-motor with combined specs of the individual motors. The stall torques, stall currents, and free currents add together but the free speed doesn't change.

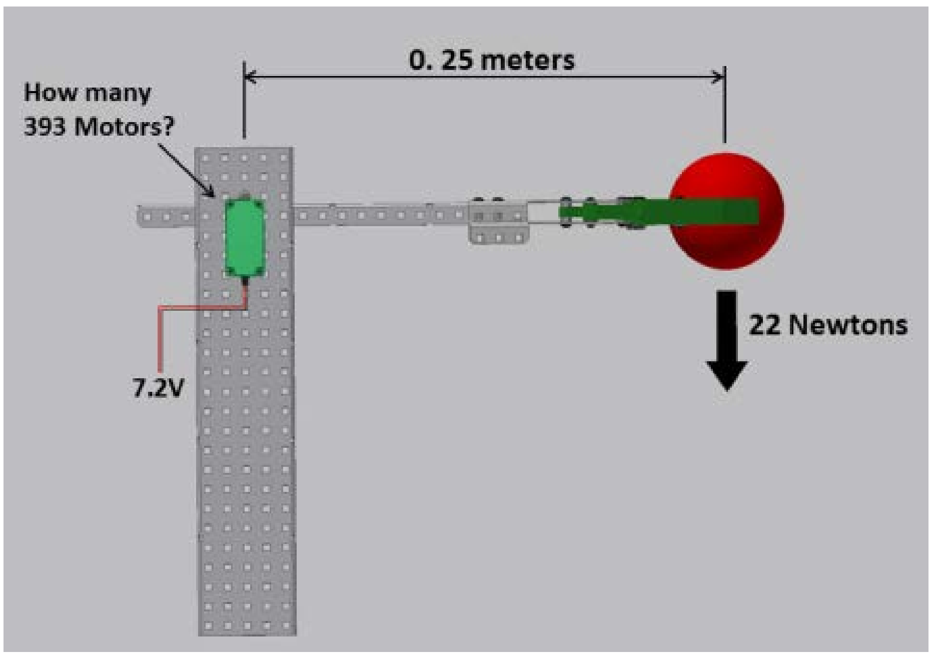
VEX 2-Wire Motor 393 (Single Motor)

	Default	High Speed
Free Speed	100 RPM	160 RPM
Stall Torque	1.67 N•M	1.04 N•M
Stall Current	4.8 Amps	4.8 Amps
Free Current	0.37 Amps	0.37 Amps

VEX 2-Wire Motor 393 (2 Motors)

	Default	High Speed
Free Speed	100 RPM	160 RPM
Stall Torque	3.34 N•M	2.08 N•M
Stall Current	9.6 Amps	9.6 Amps
Free Current	0.74 Amps	0.74 Amps

The above chart shows the specifications for the VEX 2-Wire Motor 393, and shows the specifications one would get when two are combined into the same application.



In the above example how many VEX 393 motors would it take to hold the object stationary?

The torque load on the motors can be calculated as follows:

Torque Load = Force x Distance = 22 N x 0.25 m = 5.5 N-m

Based on this torque load, one can compare it with the Stall Torque of the VEX 393 motor and determine how many would be needed.

5.5 N-m / 1.67 N-m = 3.29 motors

Thus, it would take 4 motors to hold up the arm in the example above.

7.2: Classical Mechanics up 7.4: Arm Design

7.4: Arm Design

As part of the design of a competition robot, it is possible a designer would use an arm similar to the ones shown above to pick up objects and dump them in a goal. Students should apply the lessons learned in this unit to the following example:

1. Determine the total weight of the object manipulator designed in Unit 6.

2. Determine the weight of one game object for the game described in Unit 5.

3. Assuming a simple arm system where the arm is 0.25m long, calculate how many VEX 393 motors would be required to hold the arm stationary with the object manipulator and game object at its tip.

4. Assume the 393 motors cannot draw more than 2.5 amps before their internal circuit breakers will trip; calculate how many motors are required for this application, then calculate the rotational speed of the arm.

CONCLUSION

Reviewing the above example, one could say that this is not a practical design because of how many motors it takes. One variable the designer can change to require fewer motors to accomplish the same task is the length of the arm. A shorter arm would place less torque load on the motors as described in the section on Torque above. This is known as mechanical advantage or leverage. A short arm is not practical for many designs – are there other ways to utilize mechanical advantage so that less power can be used to do the same amount of work? In the next unit, designers will see how this is possible through the use of gear ratios.

7.3: DC Motors | up | 7.5: Simulate and Size a DC Motor

7.5: Simulate and Size a DC Motor

The design team has asked you to verify that the motor raising the arm is capable of handling the torque applied through the gears. Using Autodesk Inventor Simulation, determine the force at the gear faces and then calculate the torque. This value is compared to the specifications of the motor supplied by VEX.

There are four videos to walk the student through the simulation and sizing of a DC Motor. Click on each video to review the content. Pause, rewind, fast forward, and stop features are available as the student reviews the content. The workflow in these videos includes:

- Creating a simplified model of the arm assembly.
- Setting up the assembly in the Dynamic Simulation environment.
- Running a simulation and analyzing the data.
- Calculating the torque.

To be able to complete this unit you should have a basic understanding of the Autodesk Inventor user interface, navigation, and know how to work with Assemblies. For review, please refer to Appendix 9, 'Basic Inventor Commands Overview' for further information on these."

Overview

Click here to download this video.

In this video, students will review the key phases required to calculate the maximum torque on a driveshaft. To determine the torque of a driveshaft, use the Dynamic Simulation environment in Inventor. The key phases in the required workflow are reviewed in this video.

Video 1: Review the Robot Assembly and Simplified Model

Click here to download this video.

In this video, students review the existing robot assembly and create a simplified model. To calculate the torque on the driveshaft, use the Dynamic Simulation environment in Inventor. To simplify the calculation, a new assembly is created. Start the Dynamic Simulation and review the settings.

Note the Metric equivalent of Imperial force shown 2lbf = 8.9N.

Note: The files required for this activity must be downloaded, and data sets in Imperial and Metric units are available. The data sets provided will work for Inventor version 2013 onward. Download and unzip these files and save them into a new project folder called 'DC Motor.'

Inventor 2013 - Imperial.zip

Inventor 2013 - Metric.zip

Video 2: Run the Simulation and Calculate the Torque

Click here to download this video.

In this video, students determine the torque on the driveshaft by running a simulation. The parameters of the rolling joint are modified and the simulation is run. The results are interpreted and the maximum force is determined. Using this value, the torque is calculated, and the value is compared to the specifications supplied by VEX.

Summary

Click here to download this video.

In this video, students Review the workflow required to calculate the maximum torque on a driveshaft. To determine the torque of a driveshaft, use the Dynamic Simulation environment in Inventor. The key phases in the required workflow are reviewed in this video.

Want to try more projects based on VEX robots and other exciting challenges? Click the link to access the Autodesk Design Academy.

academy.autodesk.com

7.4: Arm Design
up
7.6: Formulas

7.6: Formulas

Speed = Distance / Time

Rotational Speed = Rotational Cycles / Time

Rotational Speed = Degrees / Time

Torque = Force x Distance

Force = Torque / Distance

Where the Distance is the distance from the axis of rotation

Power = Force x Velocity

Motor Key Specs:

- Stall Torque
- Free Speed
- Stall Current
- Free Current

For determining motor characteristics as voltage varies:

New Value = Spec Value x (New Voltage / Spec Voltage)

For determining Motor Current Draw at a given Torque Load:

Current Draw = ((Stall Current – Free Current) / Stall Torque) x Given Torque Load + Free Current

For determining the Torque Load at a given Motor Current Draw:

Torque Load = (Given Motor Current – Free Current) x Stall Torque / (Stall Current – Free Current)

For determining Motor Rotational Speed at a given Torque Load:

Rotational Speed = -(Free Speed / Stall Torque) x Given Torque Load + Free Speed

For determining “Super” Motor Specs from Multiple Motors combined as one, geared to the same Speed:

Free Speed = SAME

Stall Torque = Sum of all Motor Stall Torques

Stall Current = Sum of all Motor Stall Currents

Free Current = Sum of all Motor Free Currents

7.5: Simulate and Size a DC Motor
up
7.7: Engineering Notebook

7.7: Engineering Notebook

Answer the following questions in your Engineering Notebook:

1. Why would you want to increase your speed and lower your power?

2. Why would you want to increase your power and lower your speed?

3. How does the change in the load affect your current draw?

4. How would the position of the load on the arm affect the load on the motor?

5. If the load was too great for the motor, how could we reposition our load and what effect would it have on the reach of the robot?

7.6: Formulas

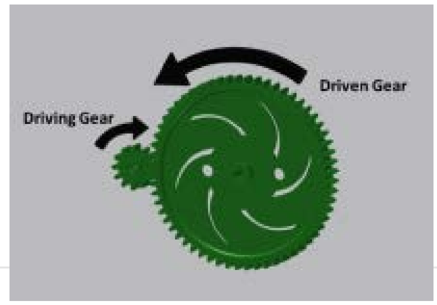
up

Unit 8: Mechanical Power Transmission

Unit 8: Mechanical Power Transmission

In this lesson students will learn about the different types of mechanical power transmission. They will learn about different gear types, and how to calculate gear ratios.

These principles will then be applied to the types of motor - arm systems seen in Unit 7.



7.7: Engineering Notebook | up | 8.1: Introduction

8.1: Introduction

In Unit 7 students were introduced to some of the concepts of classical mechanics, and also of DC motors. These concepts were then applied to the design of simple DC motor – torque arm systems.

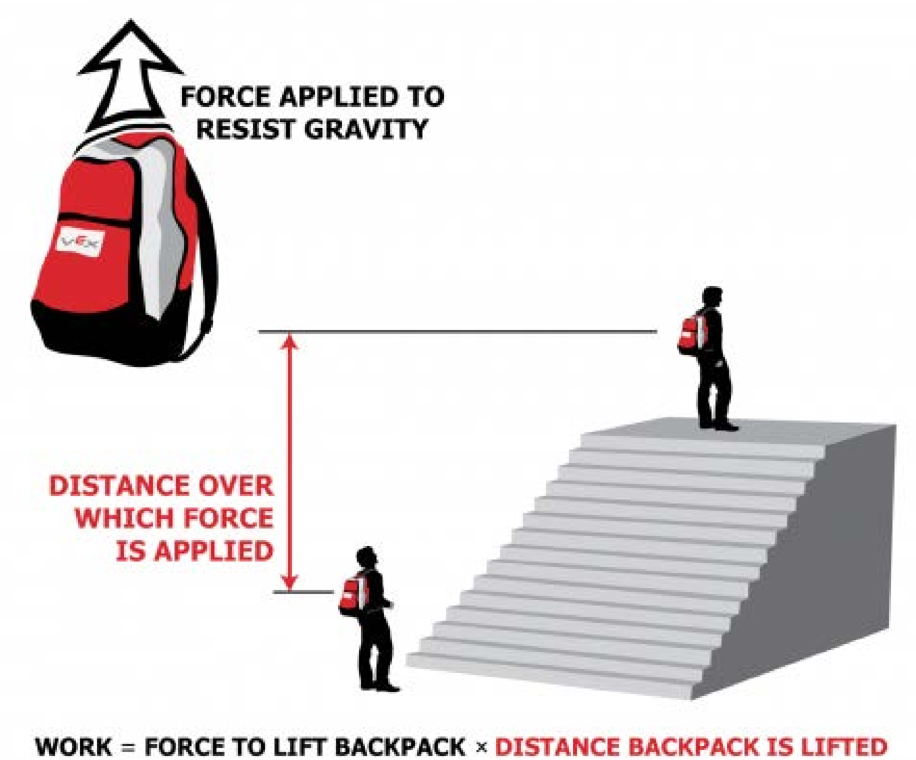
DC Motors have a limited amount of power that they can output, which means that for a specific amount of work (i.e. lifting a weight over a set distance), the motor can only do it so fast. To ensure the motor will spin at a reasonable speed and draw a reasonable amount of current, the motor load must be below a certain threshold. The motors in their default configuration are too fast for a typical applied load – they are trying to accomplish the work faster than their power limit will allow. So if the motor must lift a certain weight while still staying under the design threshold, mechanical advantage must be used.

This unit will show how gear ratios can be used to adjust mechanical advantage so that the motors can do the work more slowly, within their power limit.

Unit 8: Mechanical Power Transmission
up
8.2: Power Transmission

8.2: Power Transmission

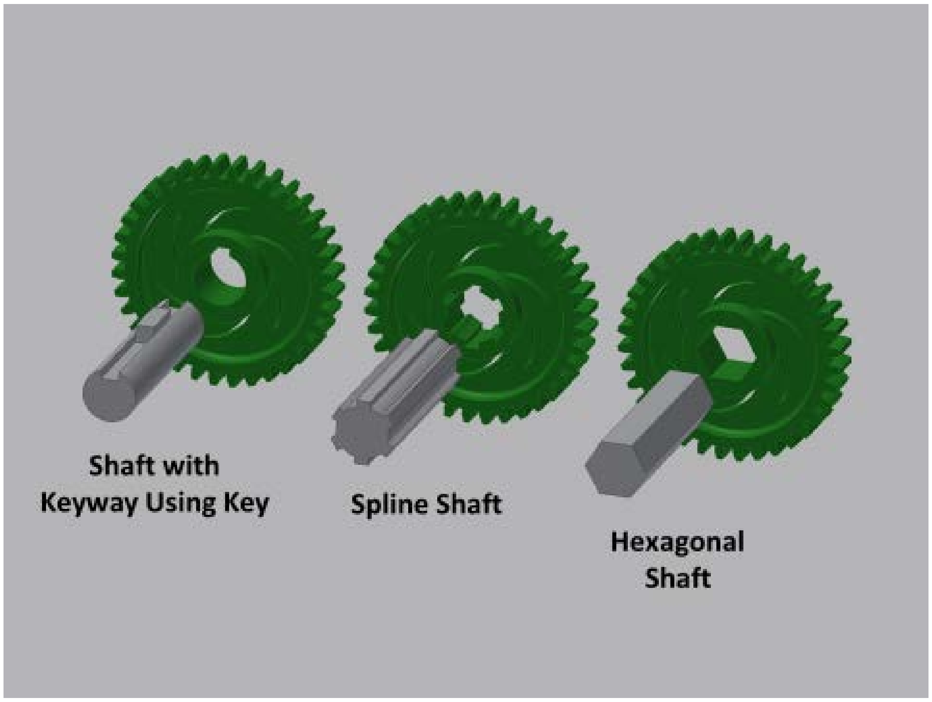
As described in Unit 7, power is the rate at which work is performed (i.e. how fast a student can carry a backpack holding 15 lbs of books up a flight of stairs.) Power can also be thought of as the rate that energy is transferred (i.e. how fast can a student transfer the chemical energy in their muscles into mechanical energy to lift the backpack up the flight of stairs.)



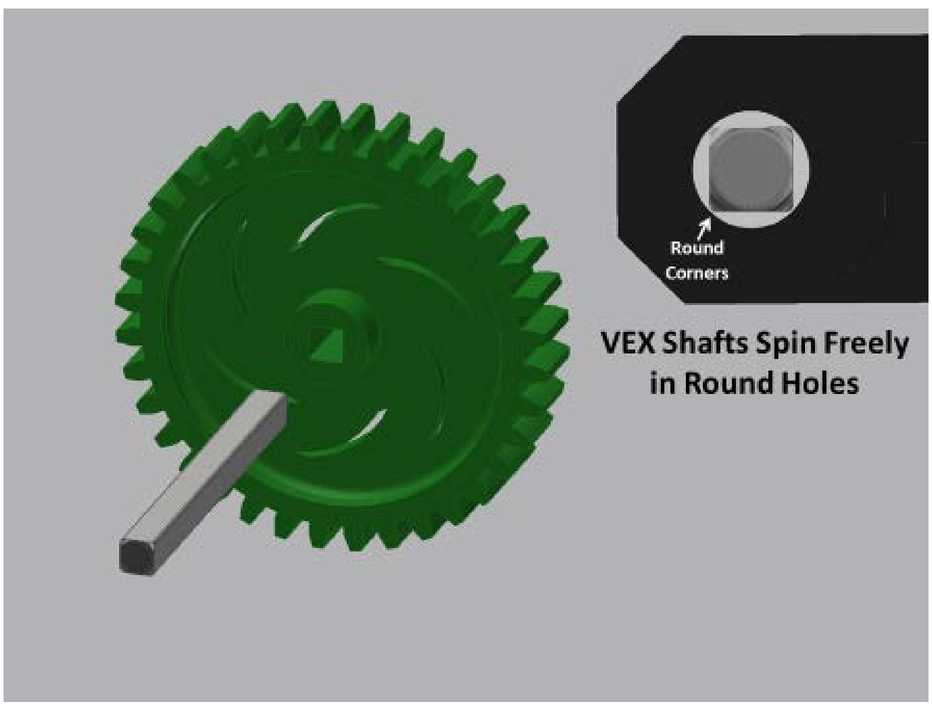
Power transmission is simply defined as the transfer of energy from its place of generation or storage to a location where it does work. Look at electricity: electrical energy is stored in a battery; it is then transmitted through wires to a motor where it is converted into mechanical energy to do work.

Mechanical power can be similarly transmitted across large distances in a variety of ways. This unit will focus on the transfer of mechanical energy in the form of rotational motion (i.e. one has an input spinning at a speed with some torque, and one needs to transfer the power from this input to a different output.)

Shafts transfer motion from point to point along their axis of motion. A common example of this is the drive axle of a car. Power is transferred into shafts via keys, splines, or polygonal shafts.



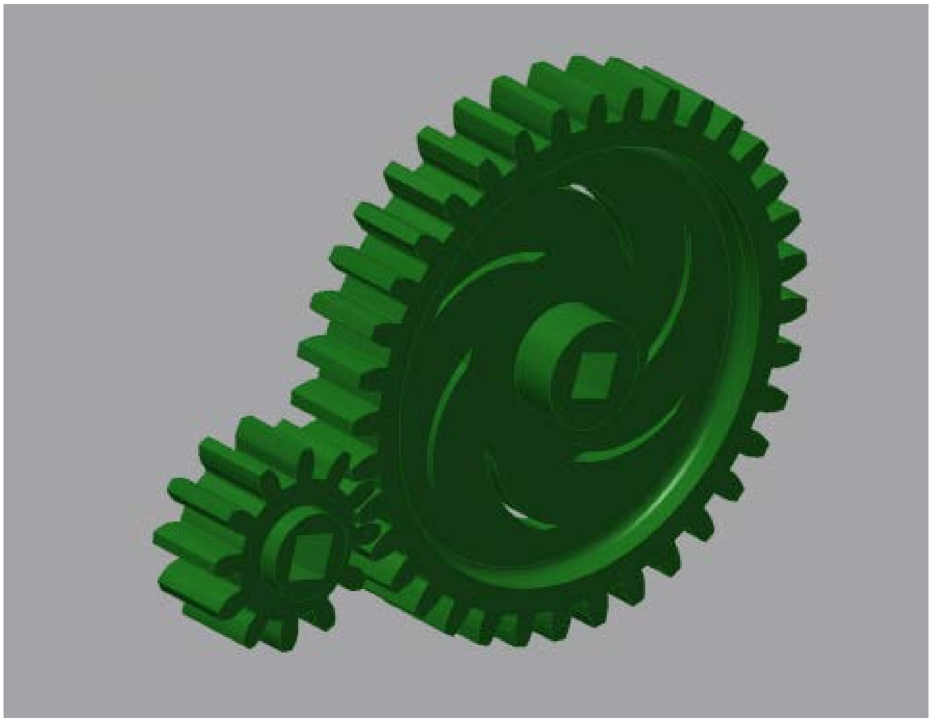
VEX uses a four-sided polygon (square) shaft as part of its motion system. This means that the shaft will transfer torque directly to anything which has a matching square hole. This square shaft also has rounded corners, which allow the shaft to spin freely in a larger-size round hole.



Gears are another method of mechanical power transmission. There are many different types of gears, and they are found very commonly in the world.

SPUR GEARS:

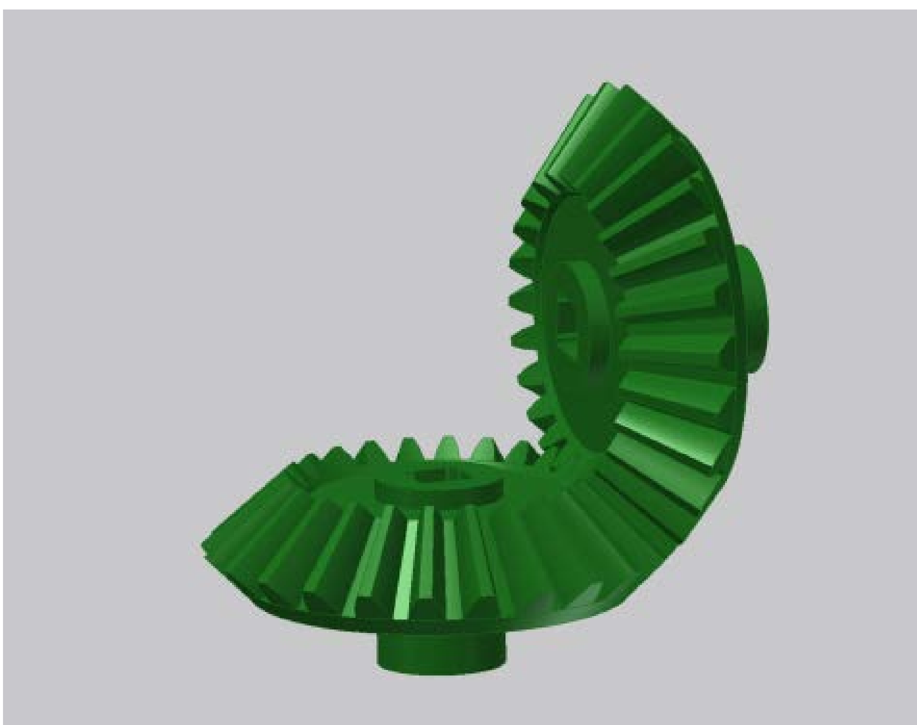
The most common type of gear is called a spur gear. When most people think of gears, they think of spur gears.



Spur gears transfer motion between two shafts running parallel to each other. Spur Gears are characterized by their teeth, which are straight and parallel to the gear's axis of rotation. These are the primary form of mechanical power transfer used in the VEX Robotics Design System. In addition, spur gears are found in the real world in everything from automobiles to the mechanism that opens the tray on a DVD player.

BEVEL GEARS:

Bevel gears are conically shaped, and transmit power between shafts that have intersecting axes of motion.



Bevel gears can transmit power between shafts at a variety of angles, but are most commonly used to transmit power 90-degrees as seen in the above example.

CROWN GEARS:

Crown gears are a type of bevel gear in which the teeth project perpendicular to the gear face.

Crown gears can mesh with other bevel gears and spur gears (as seen in the example above) so that motion is transferred between shafts with intersecting axes of rotation.

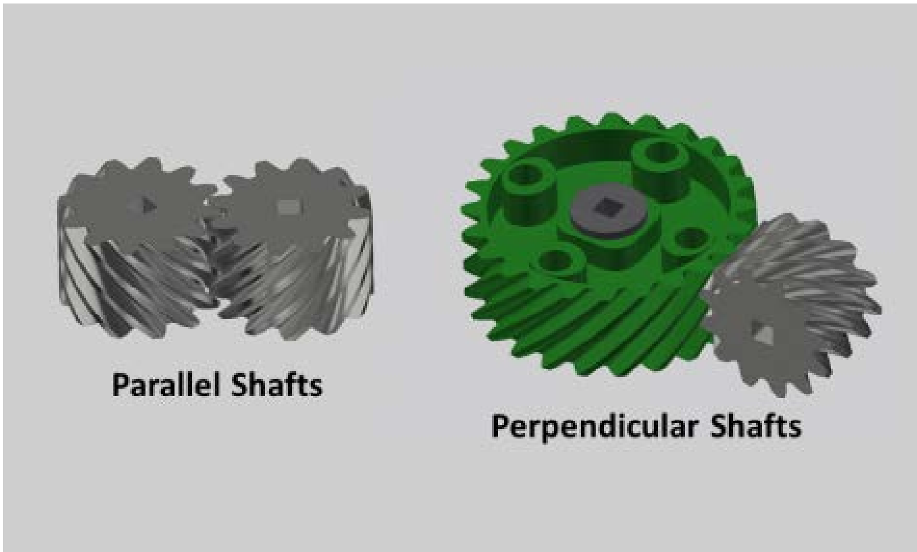
WORM GEARS:

Worm gears come in pairs: worm gears and worm wheels that mate together to transfer power between perpendicular shafts that have axes of rotation offset from each other.

Worm gears resemble screws; as they spin, they turn their mating worm wheel. This type of gear pair is very useful for creating a high mechanical advantage in a small form factor. In this type of gear pair, the worm gear can drive the worm wheel forward, but it is very difficult for the worm wheel to drive the worm gear. For this reason, these gears are useful for applications where the designer doesn't want a mechanism to be back-driven.

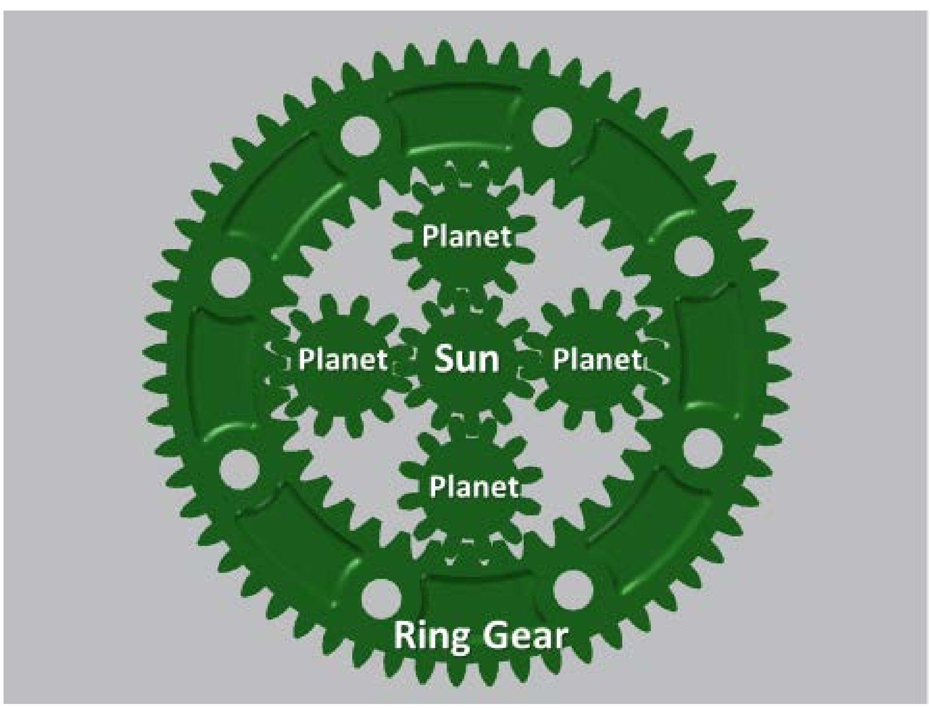
HELICAL GEARS:

Helical gears resemble spur gears, only their teeth are curved in the shape of a helix. These gears can be used to transmit power between two parallel axes of motion, or between perpendicular non-intersecting axes of motion.



EPICYCLIC (PLANETARY) GEARS:

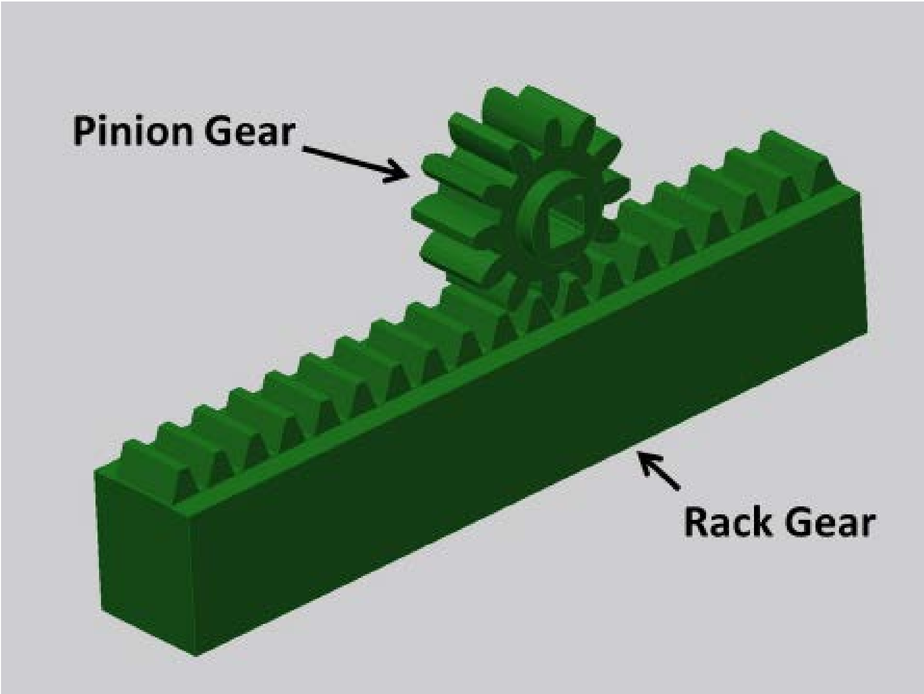
An epicyclic or planetary gear set consists of one or more planet gears moving along an outer ring gear as a central sun gear drives them. As the planet gears are driven, they typically move a planet carrier plate along with them.



Interestingly, planetary gears can be used in a variety of ways with different gears serving as the inputs and outputs. For example, one might use the sun gear as the input and the planet carrier as the output while the ring gear is held stationary, or one might use the ring gear as the input and the sun gear as the output while the planet carrier is held stationary. The overall mechanical advantage of a planetary gear set changes depending on the configuration used.

RACK GEARS:

A rack gear is a gear mounted to a straight rod, such that it moves in a linear fashion when torque is applied to it by a spur gear (known as the pinion gear).



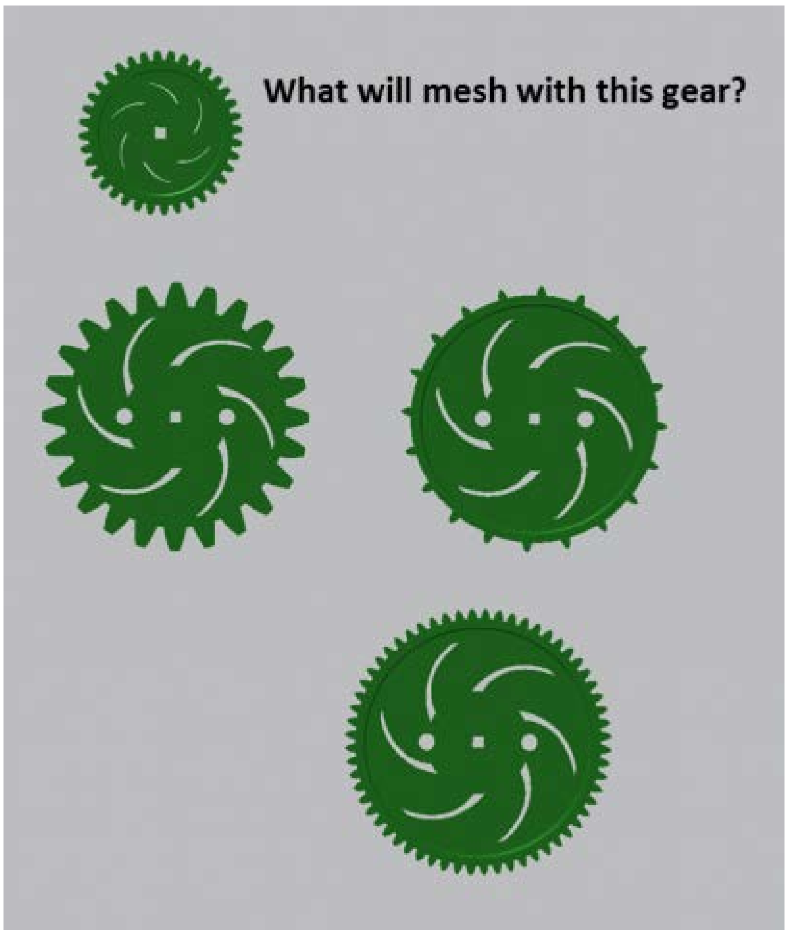
Rack and pinion gear sets are commonly used to convert rotational motion to linear motion. Cars utilize this type of gear set to convert the rotary motion of a steering wheel into a linear left/right motion required to steer the car. This is why it is called rack and pinion steering.

Within competition robotics there are many applications where rack gears can be used to create linear actuators for driving mechanisms.

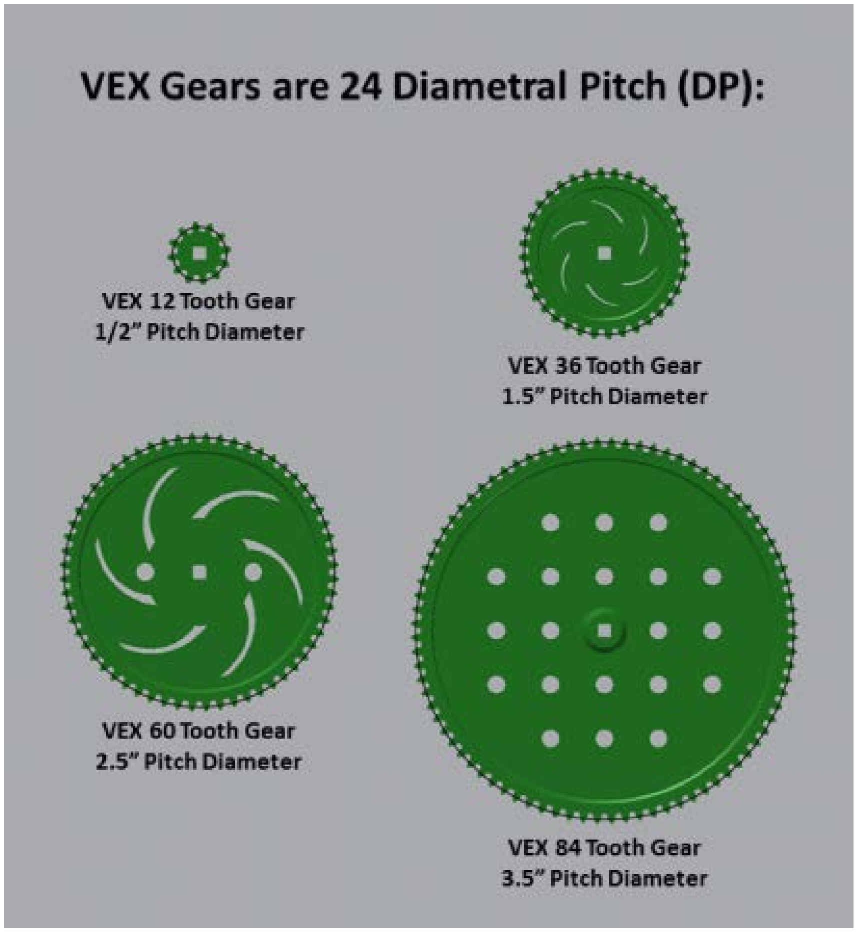
[8.1: Introduction](#) [up](#) [8.3: Gear Teeth & Pitch](#)

8.3: Gear Teeth & Pitch

In order for two gears to mesh, they need to have the same size and shape teeth on the same spacing.



As shown in the example above only one of the larger gears would correctly mesh with the smaller gear. It is said that these gears have the same pitch.



As seen above, for gear of the same pitch, tooth count and diameter are directly proportional. More teeth = bigger diameter.

8.2: Power Transmission

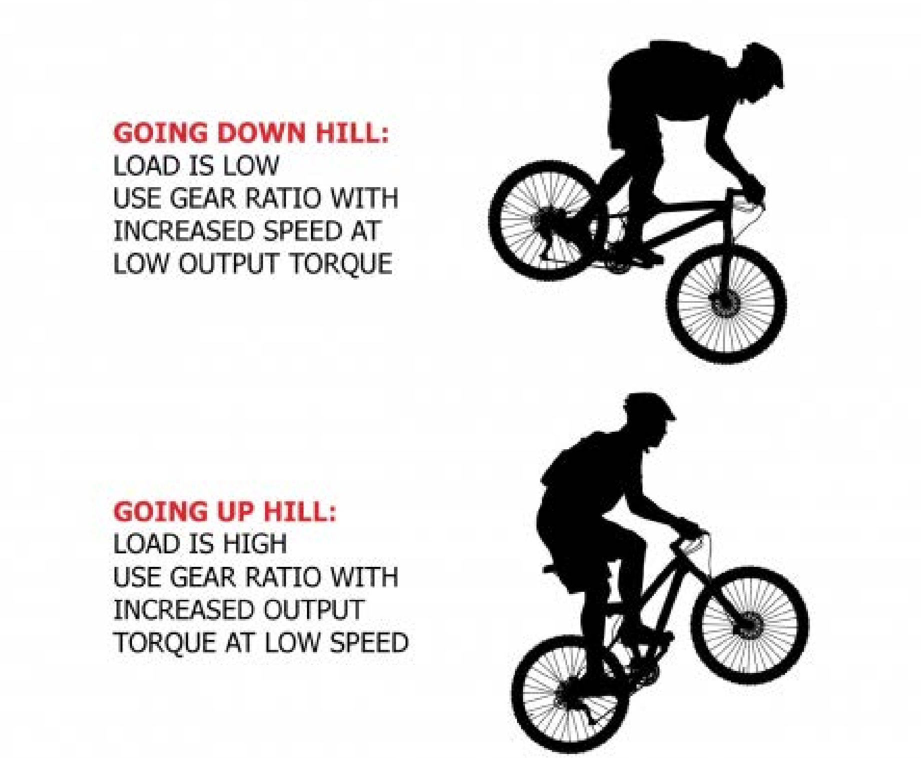
up

8.4: Gear Ratios

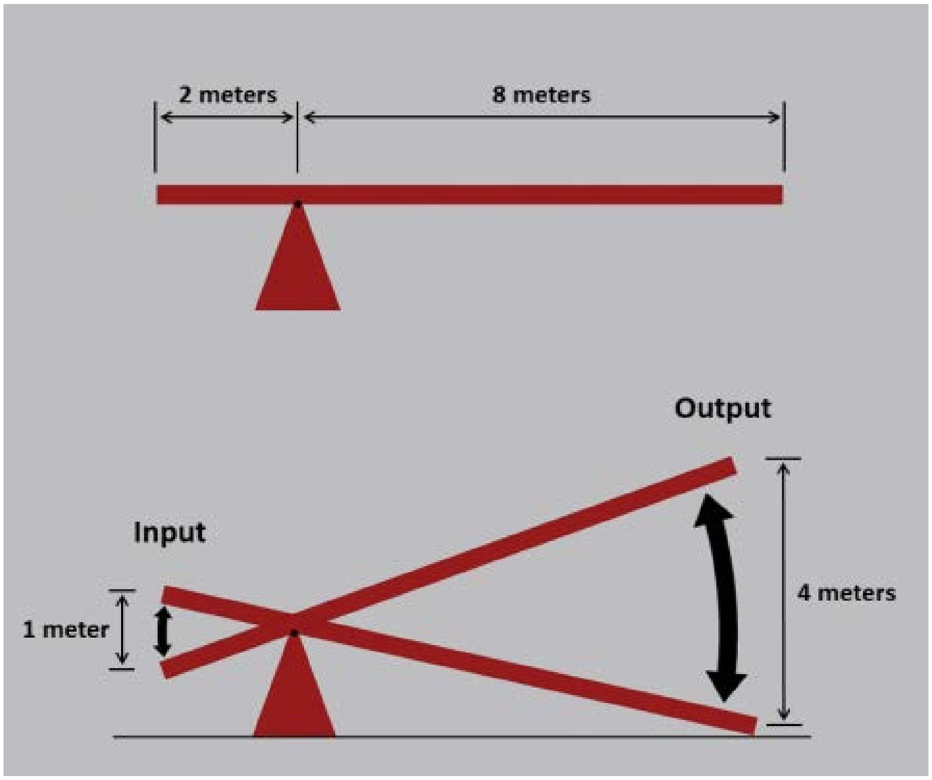
8.4: Gear Ratios

Gears are not just used to transfer power, they also provide an opportunity to adjust the mechanical advantage of a mechanism. As discussed in the introduction to this unit, there are cases where a motor itself is powerful enough for an application but the motor's output characteristics are not well suited to the application. A motor that is VERY fast but has only a little bit of torque would not be suitable to lift a heavy load; in these cases it is necessary to use gear ratios to change the outputs to a more appropriate balance of torque and speed.

Think of a bicycle: the rider has limited power, and wants to ensure the power gets harnessed as much as possible at all times.



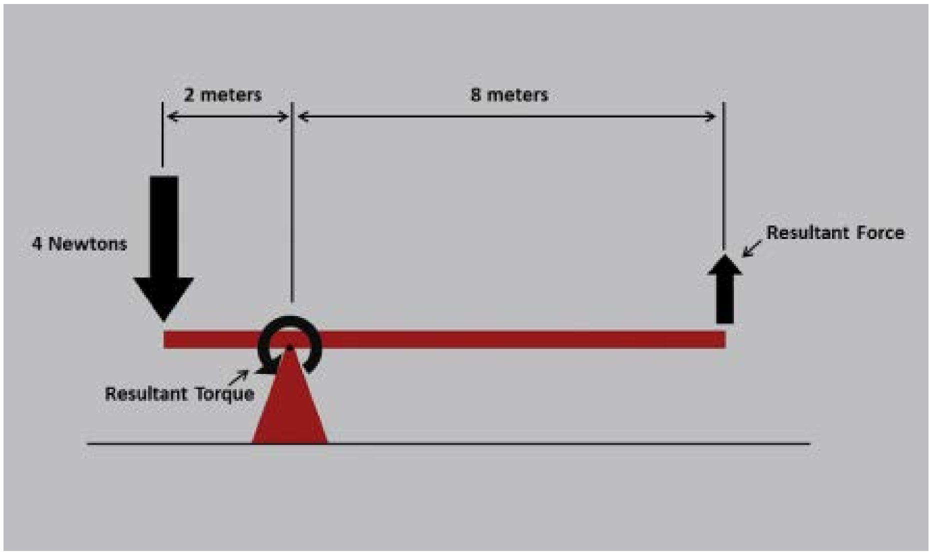
As the mechanical advantage changes, the speed of motion also changes. Power is the rate at which work is done. If the amount of work increases, the speed at which it gets done decreases.



Example 8.1

In example 8.1, one can see that if the input side of the lever moves 1 meter then the output moves 4 meters. This difference is proportional to the ratio between the lengths of the levers. Thus, output length / input length = 8 / 2 = 4.

The interesting thing about this is that it moves these distances at the same time. Let's say that it takes one second to move the input one meter, and the input is moving at one meter per second. At the same time, the output moves four meters in one second, and is moving at a speed of eight meters per second. The output is moving FASTER than the input, by the ratio of the lever lengths.



Example 8.2

In example 8.2, the same system shown in example 8.1 now has a 4 Newton force applied to the input. How much force then results at the output?

The first step is to calculate the applied torque on the center of rotation caused by the input force. Using the formulas from Unit 7:

Torque = Force x Distance from Center of Rotation = 4 N x 2 meter = 8 N-m

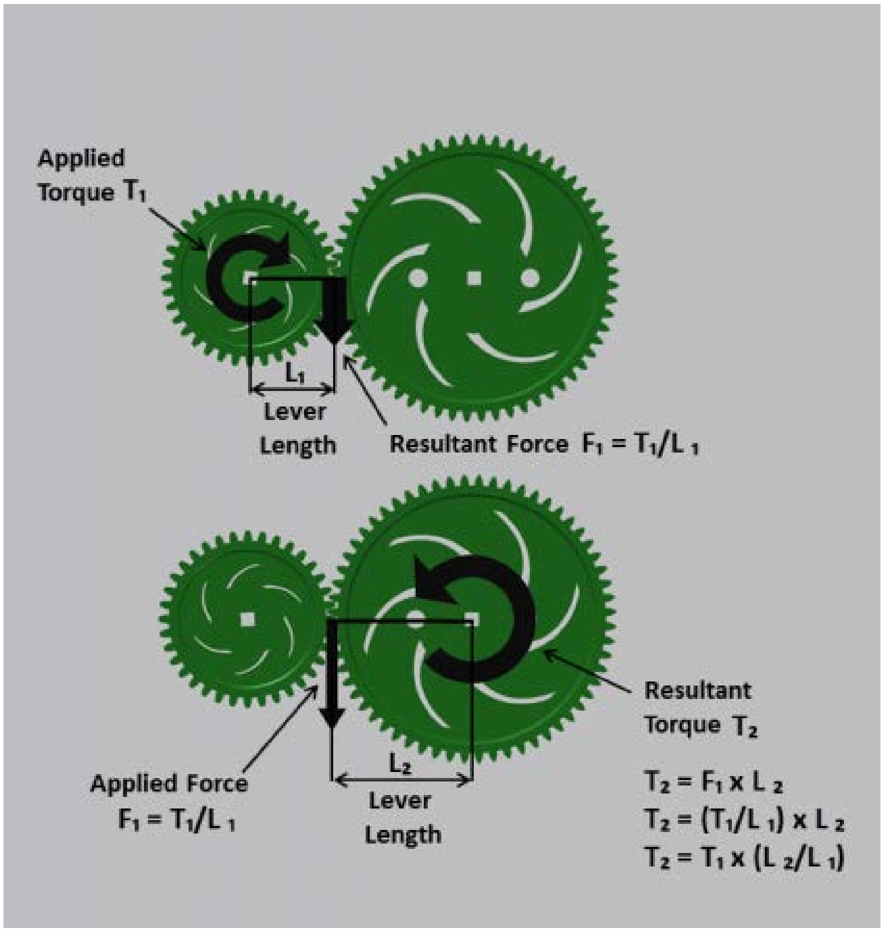
The second step is to calculate the resultant force that this torque now has on the output:

Force = Torque / Distance = 8 N-m / 8 meter = 1 Newton

So looking at the above two examples, if the lever system above has an input Force of 4 Newtons and moves 1 meter, the output will have a force of 1 Newton and moves 4 meters – it moves faster, with less force!

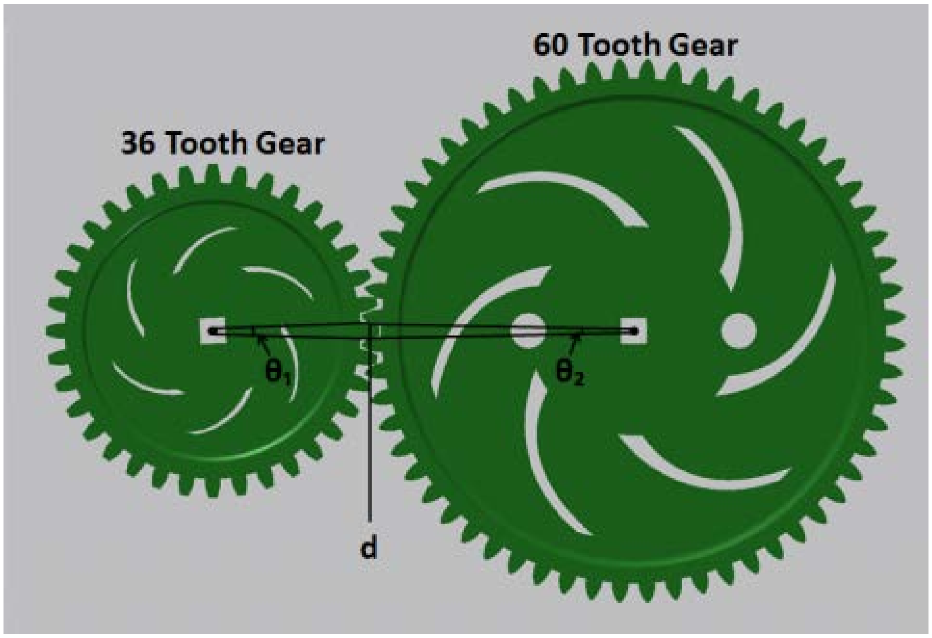
One can see how mechanical advantage (in the form of levers) can be used harness a fixed input force to accomplish a desired output. Gears work in the same manner.

A spur gear is basically a series of levers; the larger diameter the gear, the longer the levers.



Example 8.3

As seen in example 8.3, torque applied to the first gear results in a linear force at the tip of the gear teeth. That same force is applied onto the tip of the tooth of the gear it is mated with, which in turn results in a torque rotating this gear. The diameters of the gears become the lengths of the levers, and the resulting change of torque is equivalent to the ratio of the diameters. Small gears driving large gears result in a torque increase. Large gears driving small gears result in a torque decrease.



Example 8.4

In example 8.4, if the input 36-tooth gear is rotated 1 tooth (d = 1 tooth width) thenthe gear is rotating 1/36th of a revolution (a1 = 360 / 36 = 10 degrees). As it advances it moves the 60-tooth gear 1 tooth also. However, on the 60 tooth gear this is only 1/60th of a revolution (a2 = 360 / 60 = 6 degrees).

As the small gear turns a certain amount in a given time the larger gear turns a smaller amount. This means the larger gear is spinning slower than the small gear. This concept works both ways. Small gears driving big gears result in a speed decrease. Large gears driving small gears result in a speed increase.

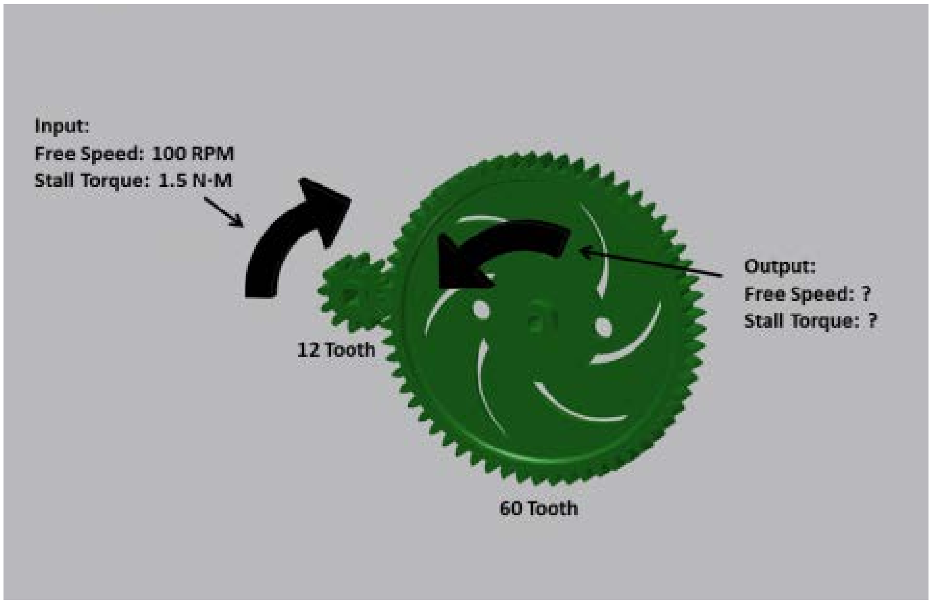
Combining the lessons of examples 8.1 through 8.4, one can see that the ratio between the sizes of two gears meshing is proportional to the resulting torque change and speed change between them. This is known as the Gear Reduction.

As discussed previously, the number of teeth a gear has is proportional to its diameter, so instead of using diameters to calculate Gear Reduction, one can just use tooth counts.

The Gear Ratio is denoted as (Driving Gear Teeth):(Driven Gear Teeth), so the above pair of gears could be described as 12:60 (or 36 to 60).

The Gear Reduction is calculated as Driven Gear Teeth / Driving Gear Teeth

So Gear Reduction = Driven Gear Teeth / Driving Gear Teeth = 60 / 36 = 1.67



As discussed above, the Gear Ratio is denoted as (Driving Gear Teeth):(Driven Gear Teeth), so the above pair of gears could be described as 12:60 (or 12 to 60).

The Gear Reduction is calculated as Driven Gear Teeth / Driving Gear Teeth

So Gear Reduction = Driven Gear Teeth / Driving Gear Teeth = 60 / 12 = 5

Looking at the above example…

The stall-torque of the second shaft can be calculated using the following formula:

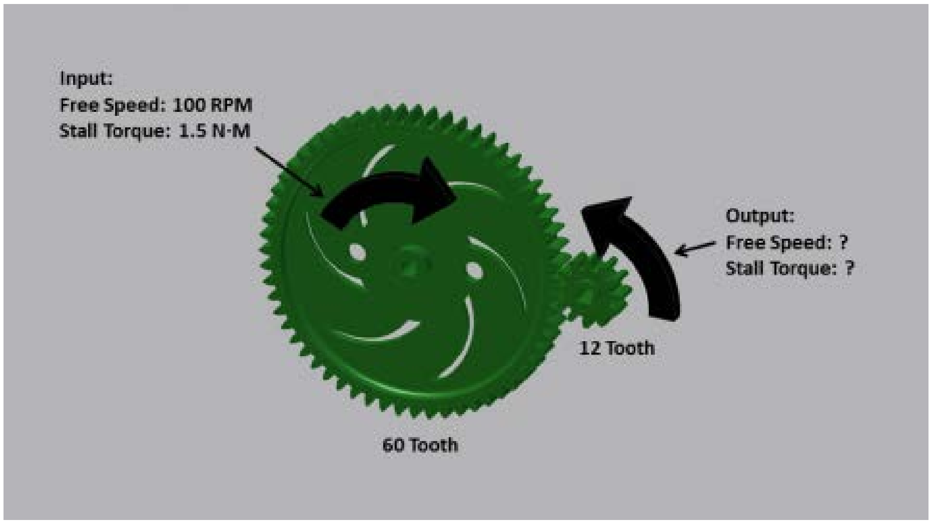
Output Torque = Input Torque x Gear Reduction

Output Torque = 1.5 N-m x 5 = 7.5 N-m

The free-speed of the second shaft can be calculated using the following formula:

Output Speed = Input Speed / Gear Reduction = 100 RPM / 5 = 20 RPM

So the secondary shaft spins with a free speed of 20 RPM and the stall torque is 7.5 N-m. The speed decreased, but the torque increased.



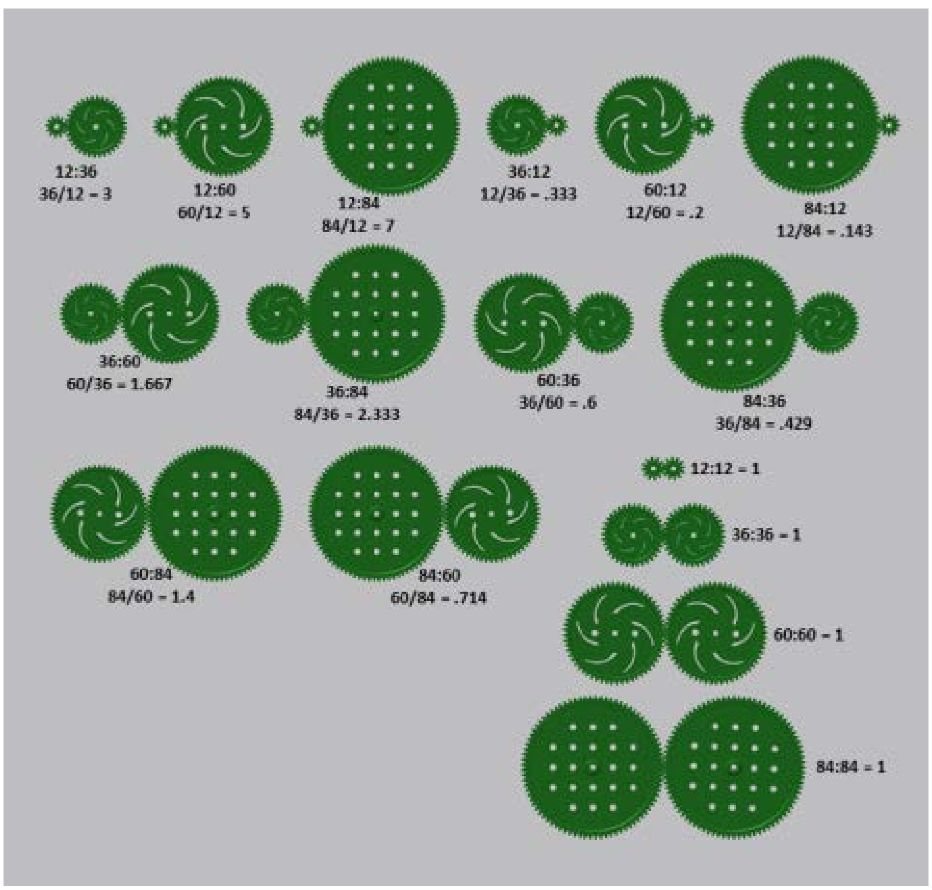
This second example can be calculated the same way.

Gear Reduction = Driven Gear Teeth / Driving Gear Teeth = 12 / 60 = 0.2

Output Torque = Input Torque x Gear Reduction = 1.5 N-m x 0.2 = 0.3 N-m

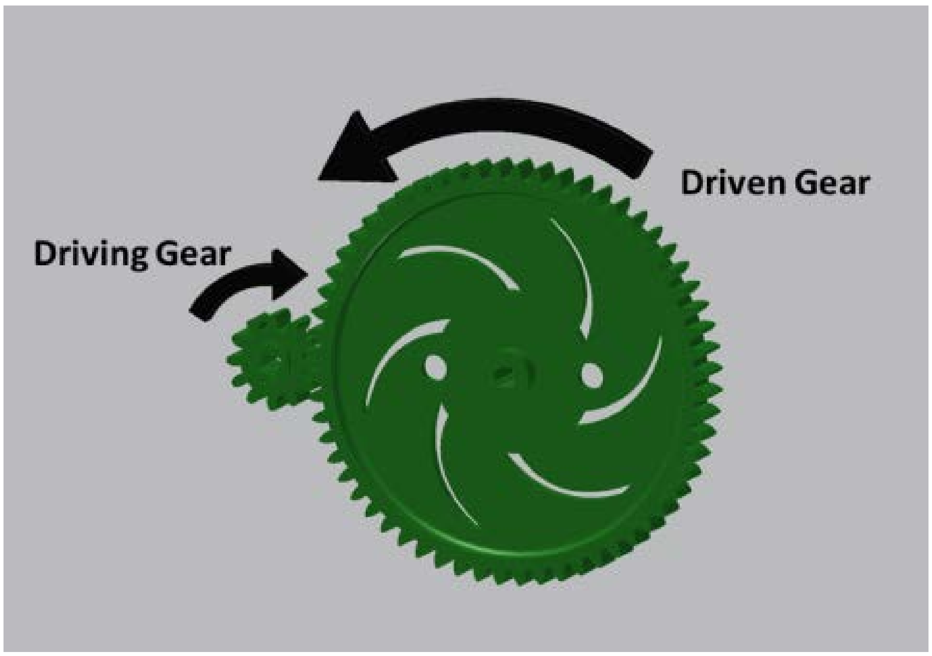
Output Speed = Input Speed / Gear Reduction = 100 RPM / 0.2 = 500 RPM

So the secondary shaft spins with a free speed of 500 RPM and the stall torque is 0.3 N-m. The speed increased, but the torque decreased.

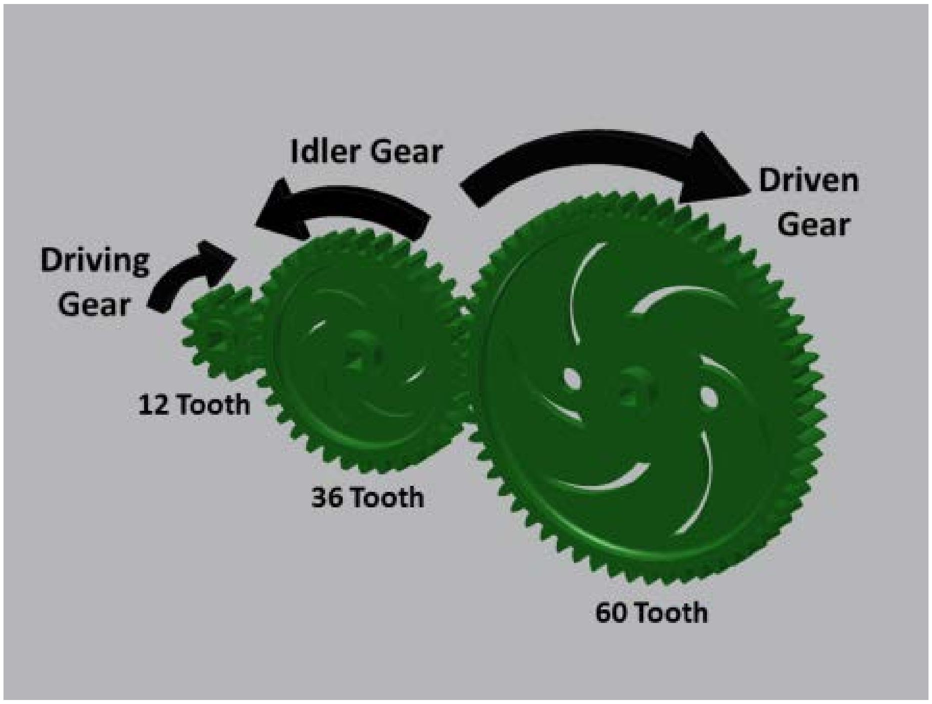


8.5: Motion Reversal & Idler Gears

One other aspect of gears designers must consider is how they reverse motion: the driven gear spins in the opposite direction as the driving gear.



To make the output gear spin in the same direction as the input gear, some designers will use something called an idler gear.



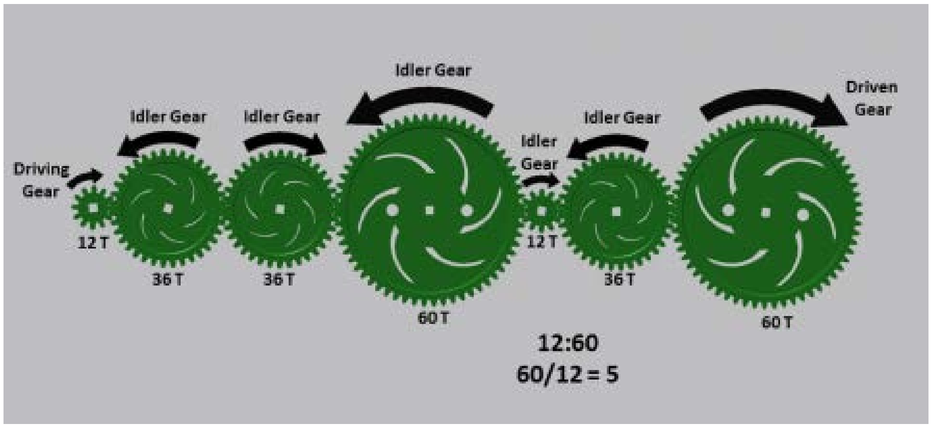
The idler gear has no effect on the overall reduction. In a previous example it was shown that a 12:60 ratio results in a gear reduction of 5. One can similarly calculate the gear reduction on the idler gear set in two stages.

Gear Reduction 1 = 36 / 12 = 3

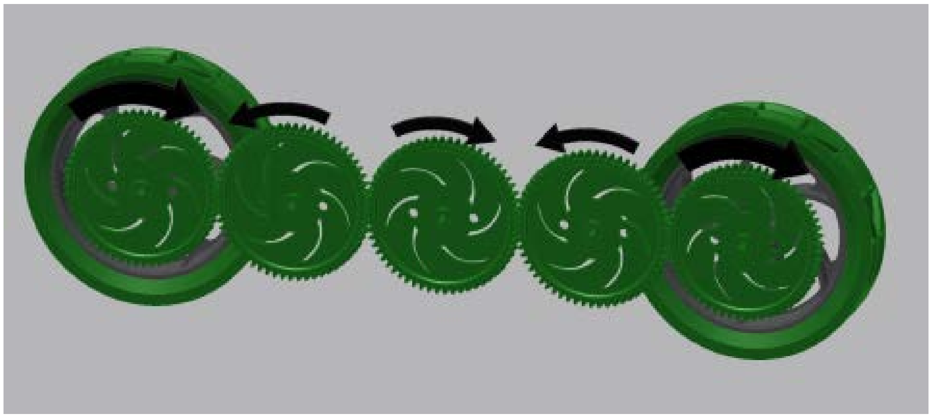
Gear Reduction 2 = 60 / 36 = 1.667

Overall Gear Reduction = Gear Reduction 1 x Gear Reduction 2 = 3 x 1.667 = 5

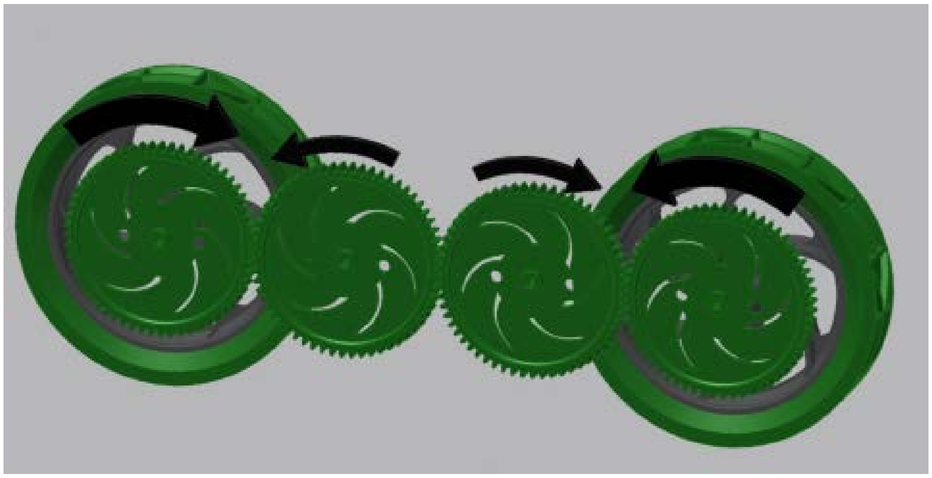
The only gears that matter to reduction in a system like this are the first gear and the last gear.



Idler gears can be very useful, especially for spanning long distances; the below example shows how they can be used in a competition robot drivetrain.



Note that if a designer uses the wrong number of idlers, the drivetrain won't function quite as well:

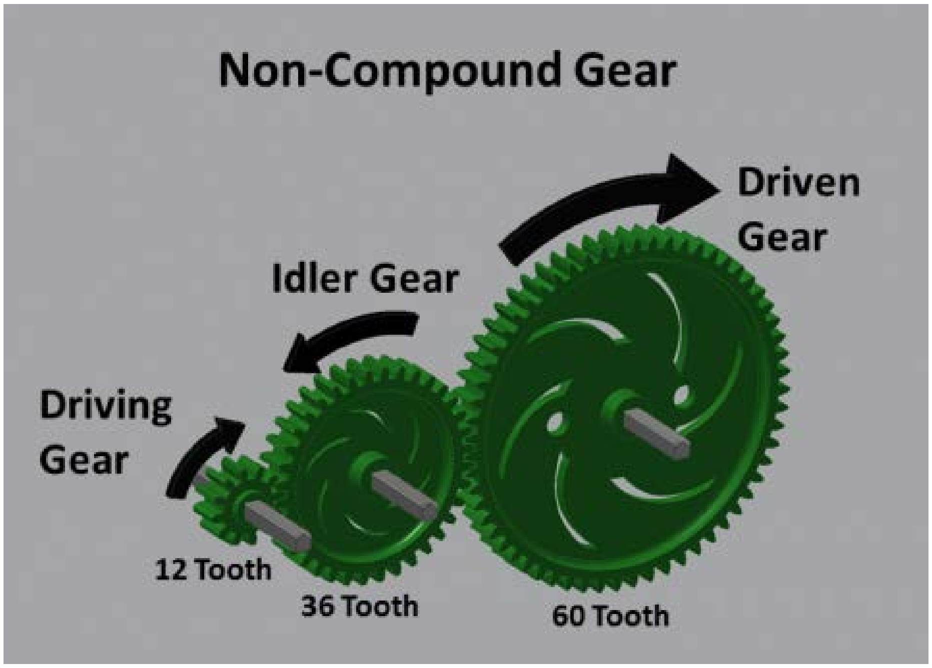


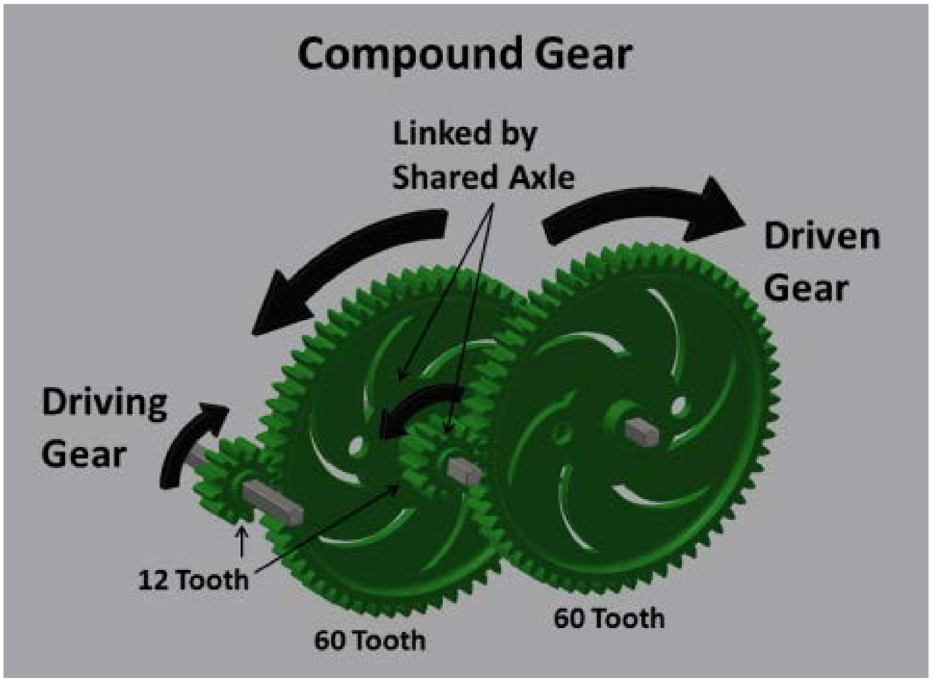
In the above example, the two wheels are spinning in opposite directions. Robots won't drive anywhere when built this way.

8.4: Gear Ratios
up
8.6: Compound Gear Reduction

8.6: Compound Gear Reduction

In certain situations, a design may require more mechanical advantage than a single gear ratio can provide or is otherwise impractical. For example, if a VEX Robot Design requires a 12:500 gear ratio it is a problem because there is no 500-tooth gear available. In this situation, a designer can use multiple gear reductions in the same mechanism. This is called a compound gear reduction.





In a compound gear system, there are multiple gear pairs. Each pair has its own gear ratio, and a shared axle connects the pairs to each other. The resulting compound gear system still has a driving gear and a driven gear, and still has a gear reduction (now called a “compound gear reduction”). The compound gear ratio is calculated by multiplying the gear reductions of each of the individual gear pairs.

For the above example the overall gear reduction is calculated as follows:

Compound Gear Reduction = Reduction 1 x Reduction 2 = (60 / 12) x (60 / 12) = (5) x (5) = 25

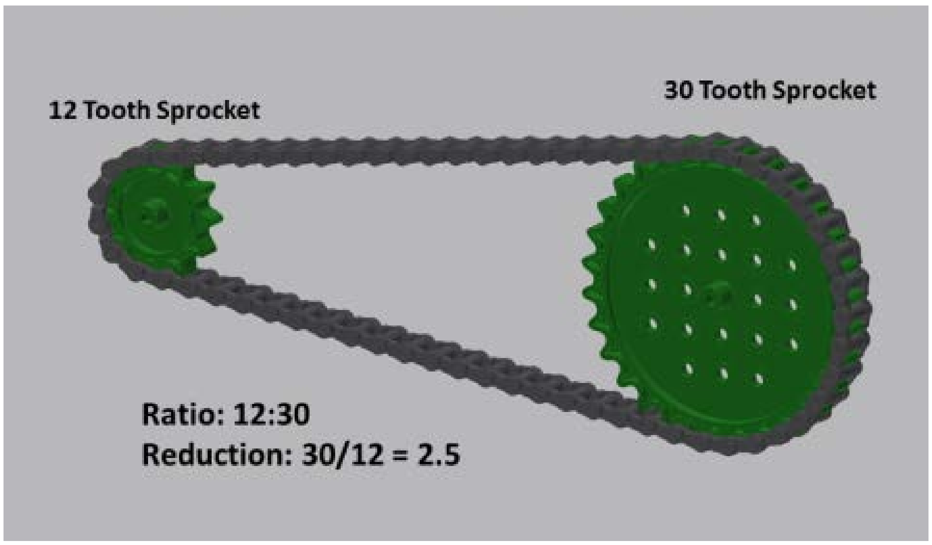
That means the output shaft is 25 times slower than the input shaft with 25 times as much torque. Compound gear ratios add up quickly!

The above example is a gearbox with twelve 12:60 reductions as part of one compound reduction. This produces an overall reduction of 244,140,625, almost a quarter of a billion to 1. This means someone would need to spin the input 244,140,625 times just to get the output to spin once! Fun Fact: spinning the input once per second, it would take approximately 7 years and 9 months before the output spun once.

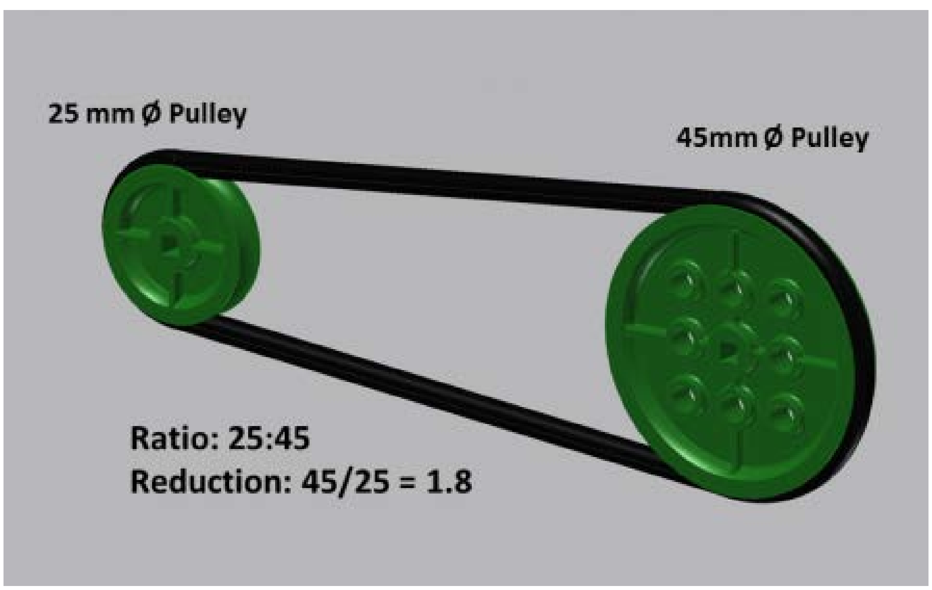
8.5: Motion Reversal & Idler Gears
up
8.7: Other Types of Reductions

8.7: Other Types of Reductions

Gears aren't the only mechanisms in competition robotics that provide gear reduction. The same principles apply to sprockets & chains and pulleys & belts.



Similar to gears, sprocket ratios can be calculated by counting their teeth.



Pulleys and belts don't have teeth, but their ratio can be calculated by comparing their diameters, as shown above.

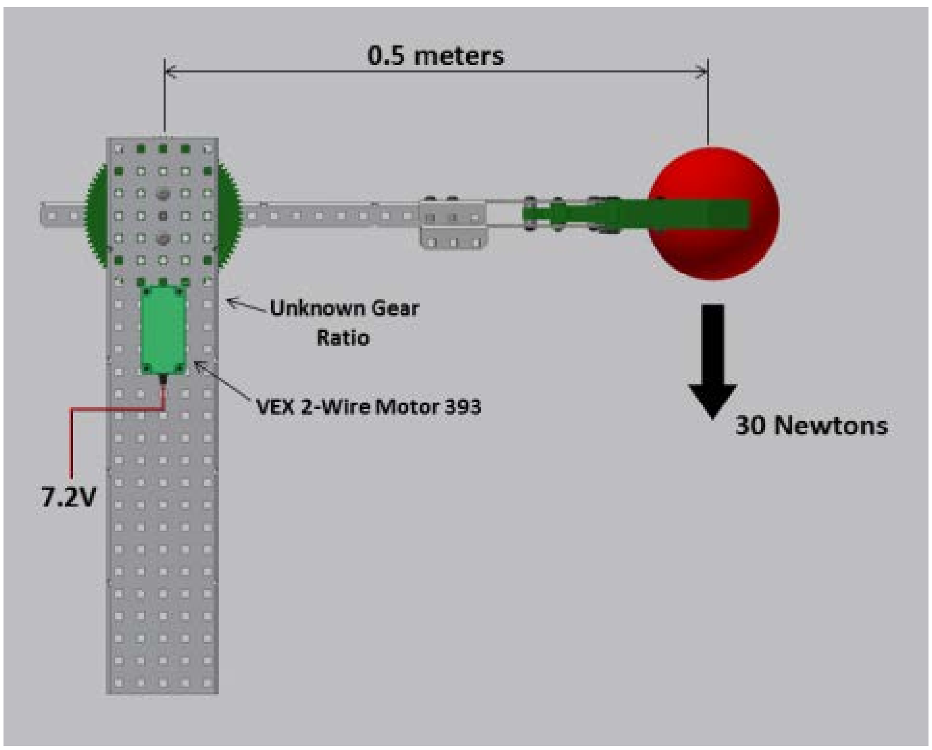
Both of these mechanisms provide more options to designers working with mechanical power transmission. These two options work great in situations where torque needs to be transferred over long distances. Unlike gears, these systems do not reverse the direction of motion.

8.6: Compound Gear Reduction
up
8.8: Applying Gear Ratios to DC Motor Systems

8.8: Applying Gear Ratios to DC Motor Systems

Based on the lessons learned in Unit 7, adjustments to mechanical advantage are important to the design of DC motor systems. DC motors sometimes have current limits they must stay under or other load limits. Designs sometimes require certain speeds, which motors must be geared up or down to achieve.

The first step in these types of design problems is to calculate the load the motor must be under to meet the design criteria. This is done using the motor characteristics, the design criteria, and the formulas and lessons used in Unit 7. After this, it is a matter of taking the output requirements and the input limits and calculating the ratio required.



VEX 2-Wire Motor 393

	Default
Free Speed	100 RPM
Stall Torque	1.67 N•M
Stall Current	4.8 Amps
Free Current	0.37 Amps

For the above arm configuration, assume the system has one VEX 2-wire Motor 393, and that this motor cannot draw more than 2.5 amps at any time. What gear ratio is needed to lift the 30 Newton object and not exceed this current draw?

The first step is to calculate the torque applied by the arm:

Torque = Force x Distance = 30 N x 0.5 m = 15 N-m

The next step is to calculate the torque load which will cause the motor to exceed 2.5 amps:

Torque Load = (Given Motor Current – Free Current) x Stall Torque / (Stall Current – Free Current)

Torque Load = (2.5 amps – 0.37 amps) x 1.67 N-m / (4.8 amps – 0.37 amps)

Torque Load = (2.13 amps) x 1.67 N-m / (4.43 amps)

Torque Load = 0.803 N-m

So if the torque load at the output of the gearbox is 15 N-m, and the load on motor can't be more than 0.803 N-m, what gear reduction is required?

Output Torque = Input Torque x Gear Reduction

Re arranging:

Gear Reduction = Output Torque / Input Torque

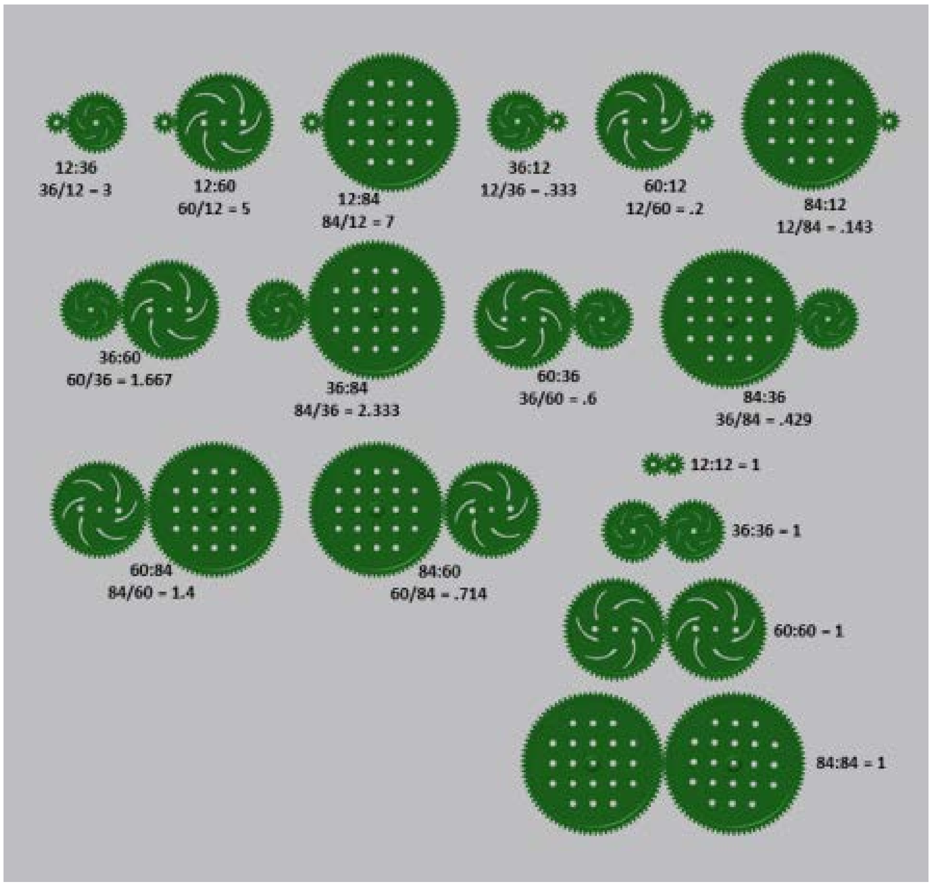
Gear Reduction = 15 N-m / 0.803 N-m = 18.68

Based on this example the gear reduction required is 18.68. However, as discussed above it is not always possible to achieve specific gear ratios with the gears on hand. Assuming the designer only has 12-tooth, 36-tooth, 60-tooth, and 84-tooth gears, calculate a compound reduction that creates the overall gear ratio required for the above example.

There are many different solutions possible; the most important thing is that the compound ratio chosen results in a reduction of more than 18.68 to achieve the design goals shown above.

Considerations: Designers should try to achieve the required reduction in as few stages as possible, and should try to get as close to 18.68 as possible without going under.

HINT: One can look at the big list of reduction options shown previously in this unit, and see which ones multiplied together as a compound reduction come close to 18.68.



How about 12:60, then 12:60?

60 / 12 = 5

60 / 12 = 5

5 x 5 = 25

25 is greater than 18.68, so it is acceptable – however, 25 is a lot more reduction than required, so the arm will move slower than is necessary to achieve the design goal. Is there another option which is closer to 18.68?

How about 12:36, then 12:84, as a simple 2-stage compound gear reduction?

36 / 12 = 3

84 / 12 = 7

3 x 7 = 21

21 is greater than 18.68, so 12:36 then 12:84 is an acceptable choice, this option is much closer to the minimum required reduction than 12:60, 12:60!

8.7: Other Types of Reductions

up

8.9: Arm Design

8.9: Arm Design

Students should revisit the arm design problem from the end of Unit 7. Designers should create a single motor, gearbox, and arm system that can lift the weight of a single game object - the manipulator from Unit 6. Designers must choose an arm length appropriate for the game, which fits within the 18” robot size requirement. The gear ratio should be calculated so that the motor can be loaded such that it draws no more than one amp of current. After calculating the necessary ratio, users must design a compound gearbox that achieves this ratio, and then calculate the final speed of the arm.

CONCLUSION

Mechanical power transmission systems are very important in the design and construction of competition robots. A designer’s ability to vary the gear ratio and the mechanical advantage in a system gives them the versatility necessary to accomplish whatever work needs to be done, with whatever motors they have (at the expense of speed.) At the simplest level, designers just need to determine their inputs and outputs, calculate the difference between them, and set their gear ratio accordingly. This simple method can be applied to any number of motor/gearing systems.

8.8: Applying Gear Ratios to DC Motor Systems

up

8.10: Modeling an Articulating Scoop

8.10: Modeling an Articulating Scoop

Testing of the scoop bucket prototype has determined that it is difficult to control the motion of the scoop. The direct drive mechanism from the motor to the lock bar is moving too quickly and the robot operator is having difficulty scooping up the game object. To resolve this design problem, a gear drive is designed and tested using Autodesk Inventor software.

In this project, there are seven videos including the Overview and Summary. To use the videos, click the link and review the content. Pause, rewind, fast forward, and stop features are available as the student reviews the content. The workflow in these videos includes:

- Modifying the scoop bucket frame.
- Modeling the driveshaft assembly.
- Modeling a gear assembly.
- Animating the assembly.

To be able to complete this unit you should have a basic understanding of the Autodesk Inventor user interface, navigation, and know how to work with Assemblies. For review, please refer to Appendix 9, 'Basic Inventor Commands Overview' for further information on these.

Overview:

Click here to download this video.

In this video, students will review the key phases required to design and model a gear assembly to drive the scoop bucket. To resolve a design issue with the motion of the scoop bucket, the frame is redesigned and a gear drivetrain is created.

Video 1: **Model a Spur Gear Assembly**

Click here to download this video.

In this video, students will review how a spur gear assembly is created using Autodesk Inventor design accelerators. Standard mechanical parts such as shafts, gears, v-belts, and bearings are easily modeled and inserted into an assembly using design accelerators. In this project, the workflow to create a spur gear assembly is shown.

Note: The files required for this activity must be downloaded, and data sets in Imperial and Metric units are available. The data sets provided will work for Inventor version 2013 onward. Download and unzip these files and save them into a new project folder called 'Manipulator.'

Inventor 2013 - Imperial.zip
Inventor 2013 - Metric.zip

Video 2: **Model a Spur Gear Assembly**

Click here to download this video.

In this video, students will modify the existing scoop bucket assembly. To accommodate the gear drivetrain, the scoop bucket frame is modified. A new frame member is added and two shafts with shaft collars are placed in the assembly.

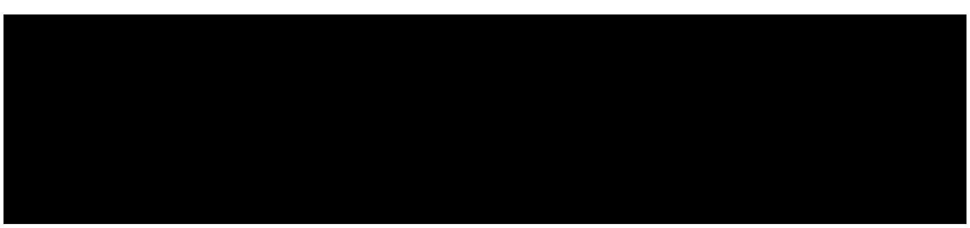
Note: Metric equivalent of Imperial offset shown -0.625" = -15.9mm and -0.25" = -6.35mm.

Video 3: Assemble the Gears

Click here to download this video.

In this video, students will place the gears in the assembly. The initial gear drivetrain consists of a 12-tooth gear and a 36-tooth gear. The gears are placed in the assembly, constrained to the driveshafts, and the gear teeth are aligned.

Video 4: Complete the Gear Assembly



Click here to download this video.

In this video, students will add bearing flats and bearing pop rivets to complete the gear assembly. To hold the shafts in place, bearing flats and shaft collars are placed in the assembly.

Video 5: Add the Motor and Animate the Assembly

Click here to download this video.

In this video, the motor is added to the shaft and the gear assembly is checked to make sure it is rotating correctly. With the gear drivetrain in place, the original motor is placed in the assembly and constrained to the driveshaft. Checks are made on the rotation of the gears and shafts. When all parts are rotating correctly, an animation is created by driving an angle constraint.

Summary:

Click here to download this video.

In this video, students will review the key phases required to design and model a gear assembly to drive the scoop bucket. To resolve a design issue with the motion of the scoop bucket, the frame is redesigned and a gear drivetrain is created.

Want to try more projects based on VEX robots and other exciting challenges? Click the link to access the Autodesk Design Academy.

academy.autodesk.com

Lesson Content click to view section

8.9: Arm Design
up
8.11: Formulas

8.11: Formulas

Gear Ratio = (Driving Gear Teeth):(Driven Gear Teeth)

Gear Reduction = Driven Gear Teeth / Driving Gear Teeth

Output Torque = Input Torque x Gear Reduction

Gear Reduction Required = Output Torque / Input Torque

Output Speed = Input Speed / Gear Reduction

Gear Reduction Required = Input Speed / Output Speed

Compound Gear Reduction = Gear Reduction 1 x Gear Reduction 2 x Gear Reduction 3 x (all other gear reductions)

8.10: Modeling an Articulating Scoop

up

8.12: Engineering Notebook

8.12: Engineering Notebook

Answer the following questions in your Engineering Notebook:

1. How do the different types of gears provide an advantage in your arm design?

2. How do the mathematical calculations help you to determine what type of gear ratio is needed in your design?

3. How would a larger gear on the motor shaft and the smaller gear on the scoop shaft affect the movement of the scoop compared to the motor?

4. How do you think a larger gear on the motor shaft and a smaller gear on the scoop shaft would affect the load on the motor?

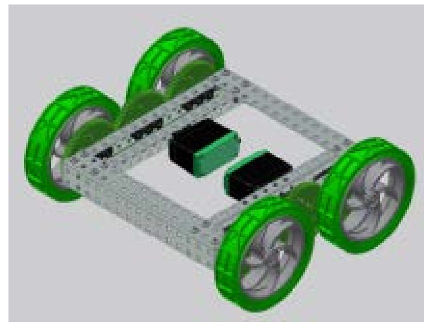
8.11: Formulas

up

Unit 9: Drivetrain Design

Unit 9: Drivetrain Design

In this unit students will be exposed to the physical principles of friction & traction while exploring the implications these principles have on robot drivetrain design. Students will be shown a variety of different robot drive system types and will learn the differences between them.



Students will then apply the lessons they've previously learned about DC motors & gear ratios to design the powertrain of their robot's drive system.

8.12: Engineering Notebook up 9.1: Introduction

9.1: Introduction

Competition robots will vary greatly depending on the game challenge they are designed to play. One thing common among them is that they usually have some method for moving. The subsystem which provides the ability to move around the field is known as a drivetrain.

Drivetrains may come in many different forms. The form which will be discussed in this chapter is one in which power is transmitted from a motor, through some sort of gear train, into a wheel, which applies force on the field surface to propel the robot forward. This form of wheeled, rolling drivetrain is the most common one found in competition robotics.

Unit 9: Drivetrain Design up 9.2: Friction and Traction

9.2: Friction and Traction

One of the most important principles students must learn before they can begin drivetrain design is that of Friction.

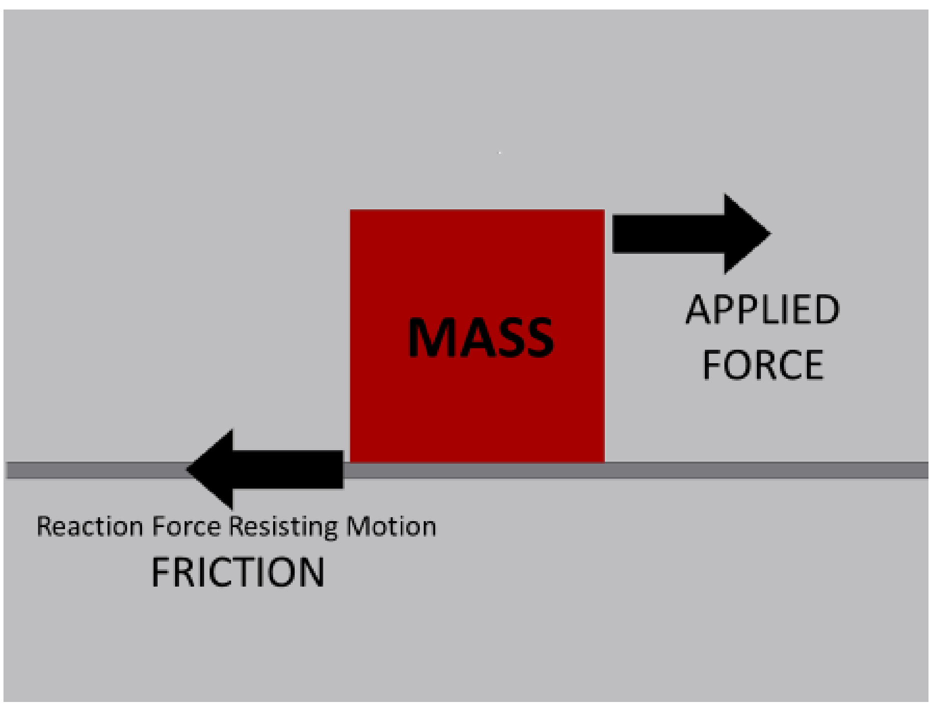
FRICTION is the force that opposes motion when two surfaces rub together. It is a reaction force only. It occurs when two surfaces are in contact and a force is applied such they slide along one another. If an object has no forces causing it to try to move, there can be no friction. No applied force, no reaction force.

There are two types of friction: Static Friction and Kinetic Friction.

Static Friction is the frictional force between two objects that are NOT moving relative to each other. It is the initial force that must be overcome in order for things to move. If the force trying to move the object is less than the force of static friction, the object cannot move.

Kinetic Friction is the frictional force between two surfaces that ARE moving relative to each other (sliding along each other).

Once an object has overcome static friction and has started moving, it has kinetic friction acting on it, resisting its motion.



In the above diagram one can see the opposing relationship between applied force and friction. As the applied force increases, the opposing frictional force also increases. Up until the mass starts sliding, the frictional force is static friction. Once the applied force exceeds the maximum static friction the mass will begin to move; after the mass begins moving kinetic friction acts upon it. Static friction is greater than kinetic friction, so once the mass begins sliding it takes less force to keep it sliding.

It is easy to duplicate both types of friction by simply pushing the palm of one hand against the palm of another and trying to move them in a sliding motion. This motion will be resisted by the texture of the skin and the magnitude of the applied force. The harder the hands are pushed together, the harder it is to move them. This is static friction.

As the sliding force increases, the hands begin to slide and they are moving relative to each other; now kinetic friction is present. One can note how after the hands break loose from static friction, it takes less force to keep them sliding.

There are two factors which determine the maximum frictional force which can occur between two surfaces: how “grippy” the surfaces are (known as the Coefficient of Friction of the surfaces), and how hard the two surfaces are being pushed together (known as Normal Force).

The maximum Force of Friction (Ff) between two surfaces is equal to the Coefficient of Friction (Cf) of those two surfaces multiplied by the Normal Force (N) holding those surfaces together.

Maximum Force of Friction = (Coefficient of Friction) x (Normal Force)

Ff = Cf x N

COEFFICIENT OF FRICTION:

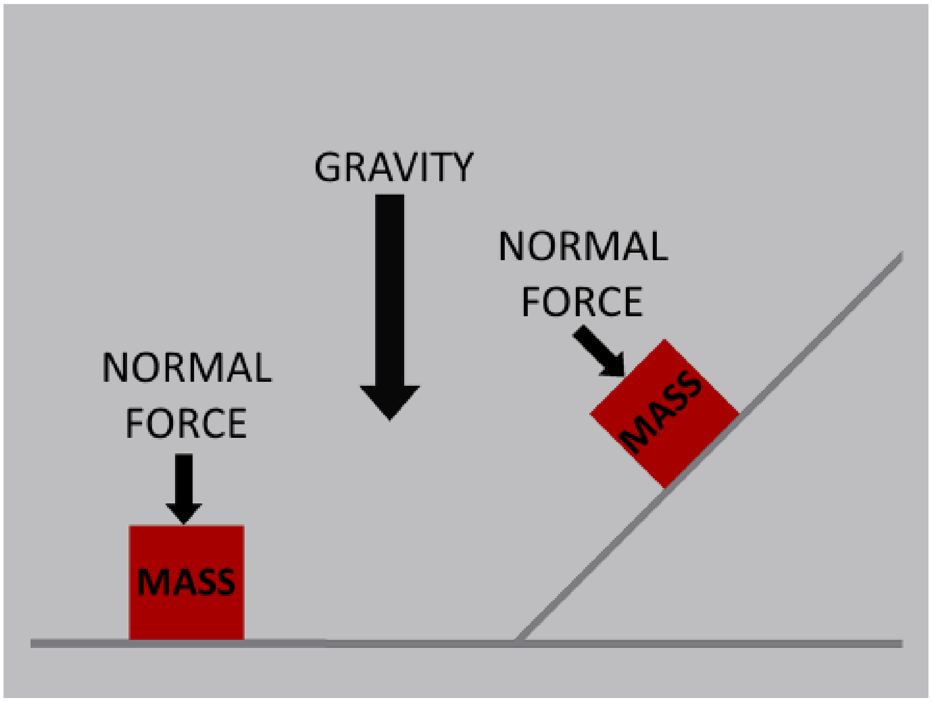
As stated above, a coefficient of friction is a constant that describes the “grippyness” of two surfaces sliding against one another. Note, that this is not function of a single surface, but of two surfaces. For example, a tire on its own has NO coefficient of friction, but a tire sliding on pavement DOES have a coefficient of friction.

Slippery objects have a very low coefficient of friction while sticky objects have a very high coefficient of friction. This constant is determined for a pair of surfaces (not a single surface.) Each pair of materials will have a coefficient of static friction, and a coefficient of kinetic friction.

One shouldn’t confuse pure friction with actual sticky surfaces like tape or high friction coatings that bind to the other surface. These surfaces almost need to be looked at as being joined as one. For instance, tape resists sliding even when there is no normal force, or even when there is a negative normal force.

NORMAL FORCE:

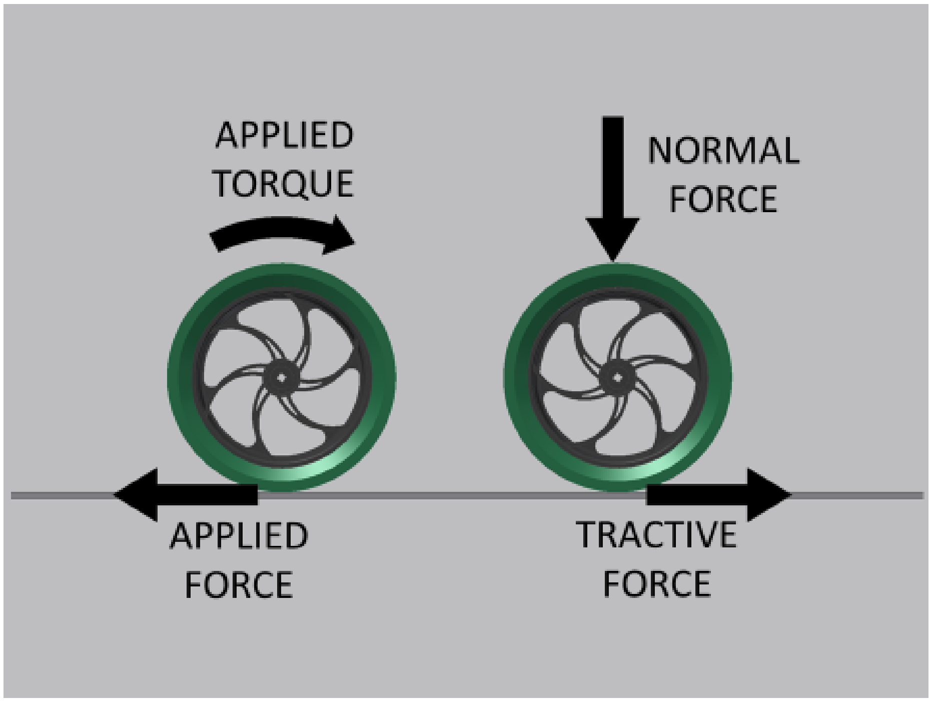
The force which presses two sliding surfaces together is referred to as NORMAL FORCE. This normal force is always perpendicular to the two surfaces (if an applied force is not perpendicular to the two surfaces, only a portion of it will act as normal force.) Often the normal force acting on two surfaces is simply the weight of one object resting on the other; in this case the normal force is caused by gravity.



As seen in the above diagram, when an object is on an inclined plane, gravity is not acting perpendicular to the sliding surfaces. In this case only a portion of the object's weight would act as normal force.

TRACTION:

TRACTION can be defined as the friction between a drive wheel and the surface it moves upon. It is the amount of force a wheel can apply to a surface before it slips. A wheel will have different traction on different surfaces; as described above, the coefficient of friction is based on pairs of surfaces.



As covered in Unit 7, and as seen in the diagram above, when a torque is applied to a wheel, it applies a force along the ground. However, one can imagine that if the wheel was spinning on ice, the wheel would slip and would not move forward. The friction between the wheel and the ground is necessary to make it move forward, this is the tractive force.

Note that the tractive force is equal to the frictional force between the wheel and the ground. If the wheel is rolling along and not slipping, it is equal to the static friction. If the applied force exceeds the maximum static friction then the wheel will start to slip, and now the tractive force is equal to the maximum kinetic friction.

Increasing Traction:

Since traction is dependent on the friction between the wheel and the surface, to increase traction one must maximize this friction. As seen above, the friction between two objects is dependent on the coefficient of friction between them (in this case between the wheel and the surface it drives on) and the normal force (the weight of the robot pressing the wheel to the surface). To increase traction, one must either increase the coefficient of friction (grippier wheels) or increase the normal force acting on the wheel (heavier robot, or more weight on drive wheels).

Building a Pushing Robot:

In order to build a robot capable of pushing or pulling with great force, the robot requires two things. It requires high traction wheels, and a significant amount of torque driving those wheels. Friction is a reaction force; if there is no applied force there will be no traction. To maximize traction, the torque applied to the wheels needs to be enough to reach the maximum static friction of the wheels.

A car can have all the traction in the world, but if it has a small engine it won't be able to push or pull anything. This is why small cars can't tow big trailers or boats.

Friction in VEX:

There are a variety of components in the VEX Robotics Design System which can be used to gain traction, including several types of wheels. Each of these has different characteristics on different surfaces. It is important for designers to experiment and determine which wheel is best for a given application.

Friction between the wheels and the floor is not the only friction relevant on VEX Robots. Friction also acts as a brake on the rotating components of the robot and reduces the amount of power which gets from a motor to its output. The VEX Robotics Design System has several parts designed to reduce friction in a robot design. Metal against metal contact is not desirable in moving systems. The plastic parts such as the bearing blocks, spacers, and washers allow for lower friction contact points.

9.1: Introduction | up | 9.3: Drivetrain Terminology

9.3: Drivetrain Terminology

Below is some of the terminology that will be important as student designers learn about the various types of drivetrains.

DRIVE WHEEL – a wheel to which power is transferred and is used to propel the robot forward. Not all wheels are drive wheels, some wheels do not help to move the robot.

TURNING POINT – the point around which a robot is turning.

TURNING SCRUB – the friction caused by wheels dragging "sideways" along the ground as the robot turns. Turning scrub resists the robot turning.

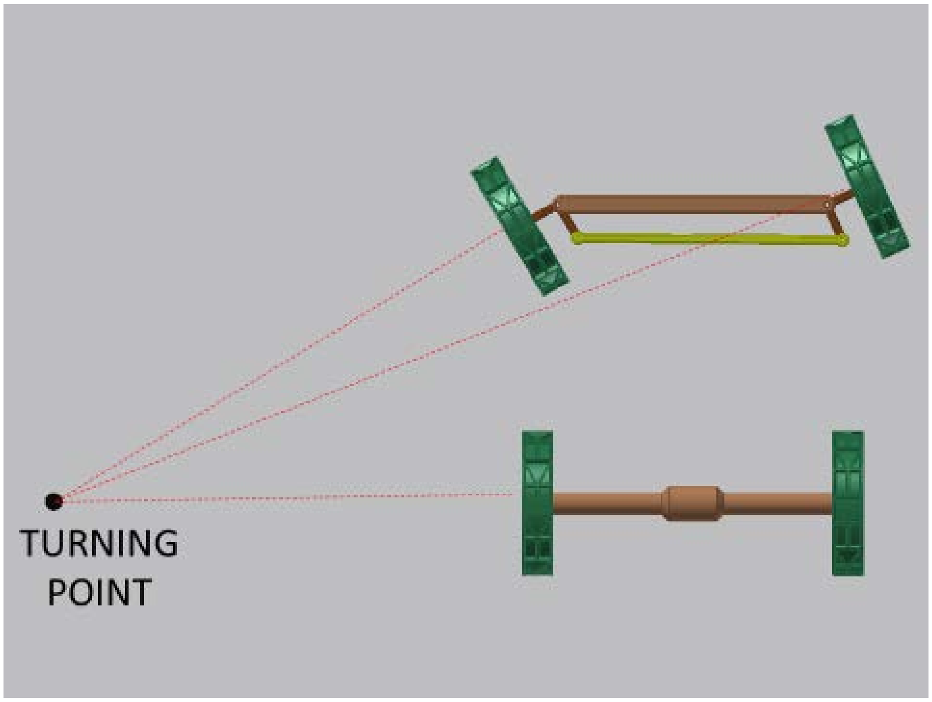
ZERO RADIUS TURN – A zero radius turn is when a robot turns in place without moving forward. In a zero radius turn, the turning point of the robot is at the center of the robot.

CHASSIS – The structure of a robot which holds the wheels, motors, and gear-train in place.

DRIVETRAIN TYPES:

There are a number of different drivetrain types commonly found in competition robotics. All of these types have their own benefits and drawbacks. A few common types are described below.

Ackermann "Car Style" Steering:

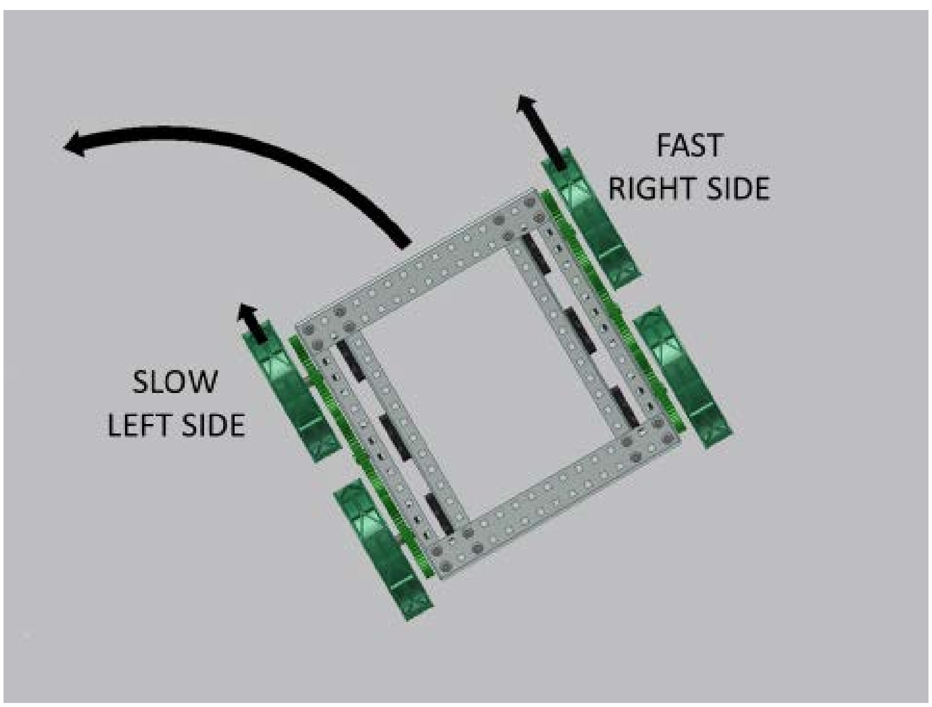


In this type of drive, all the wheels move in the same direction: forward or backwards. Steering is accomplished by turning the wheels such that all the wheels are positioned in an arc around a single turning point.

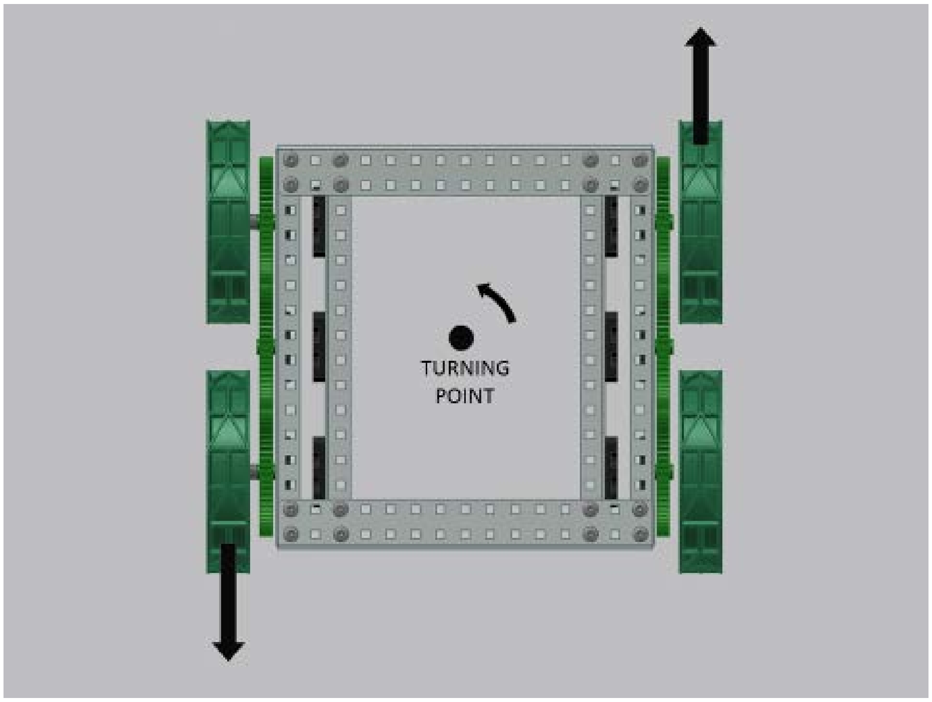
One of the benefits of this configuration is that there is no turning scrub when it is properly set up. However, a major drawback for this type of drive is its inability to perform a zero radius turn.

Skid Steer:

This is the most common type of competition robotics drivetrain. This style is sometimes referred to as “tank drive” since it is commonly used on tanks. In this type of drivetrain, the wheels on the right side and the left side of the drive are powered by separate motors. These wheels are locked pointing forward/backward, and do not steer. Steering is accomplished by varying the speed of the different sides (i.e. if the right side goes forward very fast, and the left side goes forward slowly – the robot turns left).

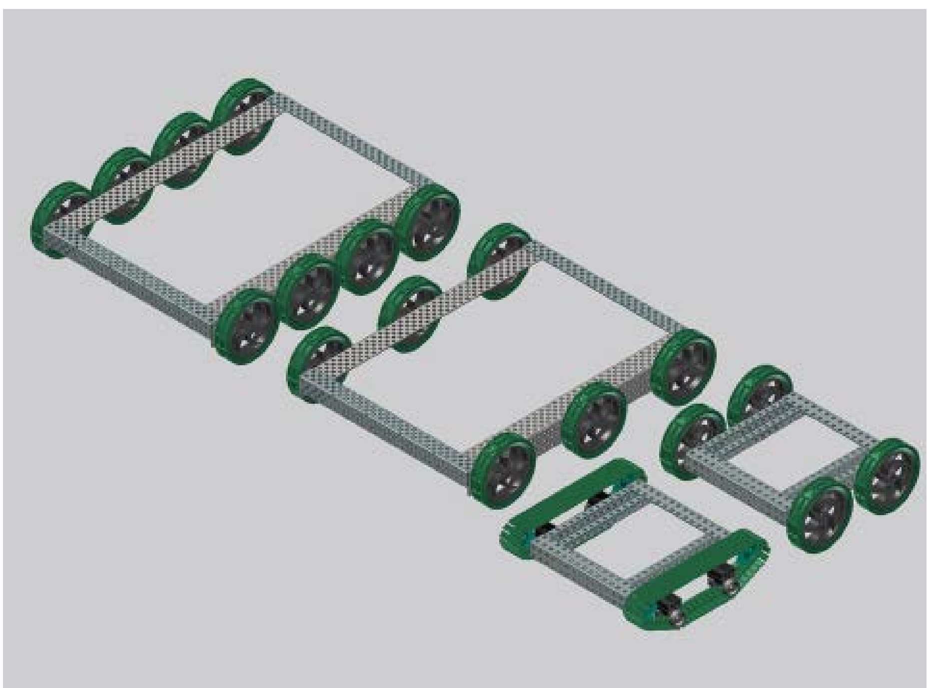


This type of drivetrain is capable of zero radius turns; the robot driver would simply power one side forward and the other side in reverse.



This type of drive does require two motors, but does not require a specific actuator for steering – both motors are used when it is going straight, so ALL power can be used for acceleration or pushing. When this drivetrain turns, it does have turning scrub (this will be described in more detail later).

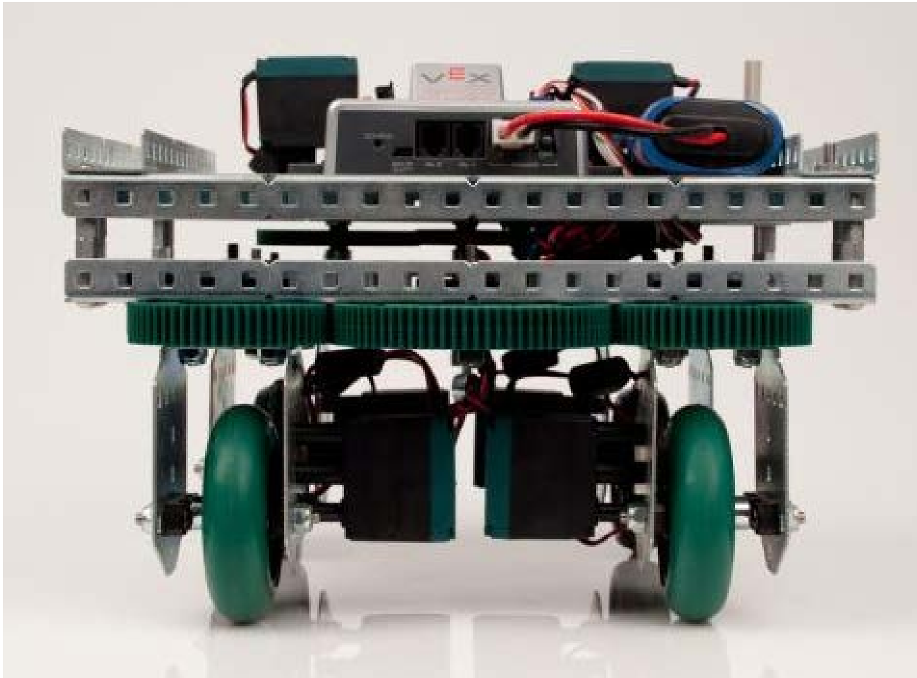
Skid Steer drivetrains can come in many different configurations, but they all function the same way.

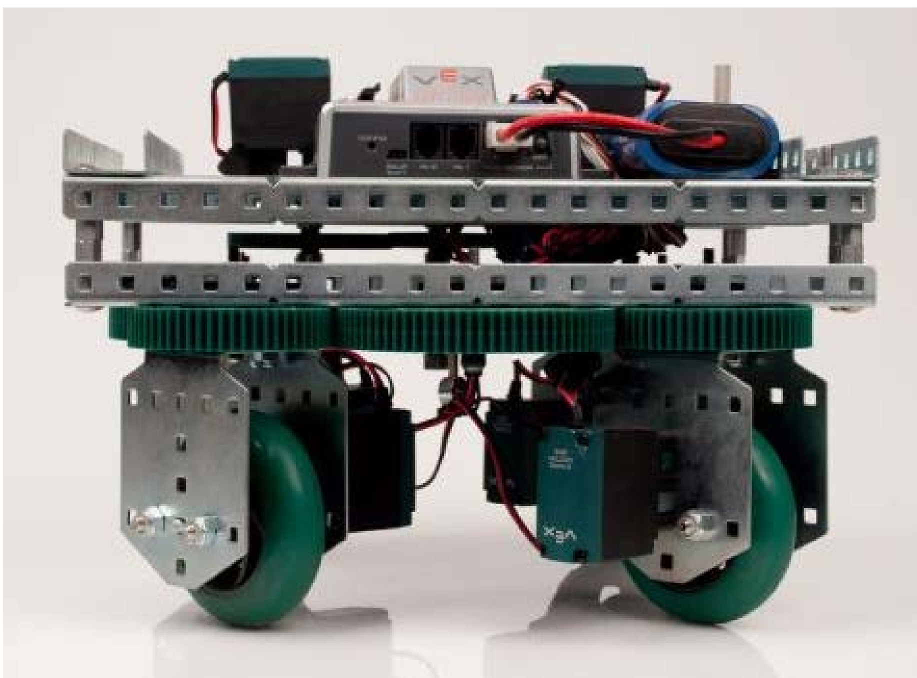


The rest of this unit will focus on Skid Steer type drivetrains.

Swerve Drive:

Another type of competition robotics drivetrain is known as a swerve drive. A swerve drive is one in which the wheels are not only powered forwards & backwards, but can also be independently steered. This means the robot can turn like a Skid Steer robot, but can also move in any direction by steering its wheels.





Crab Drive:

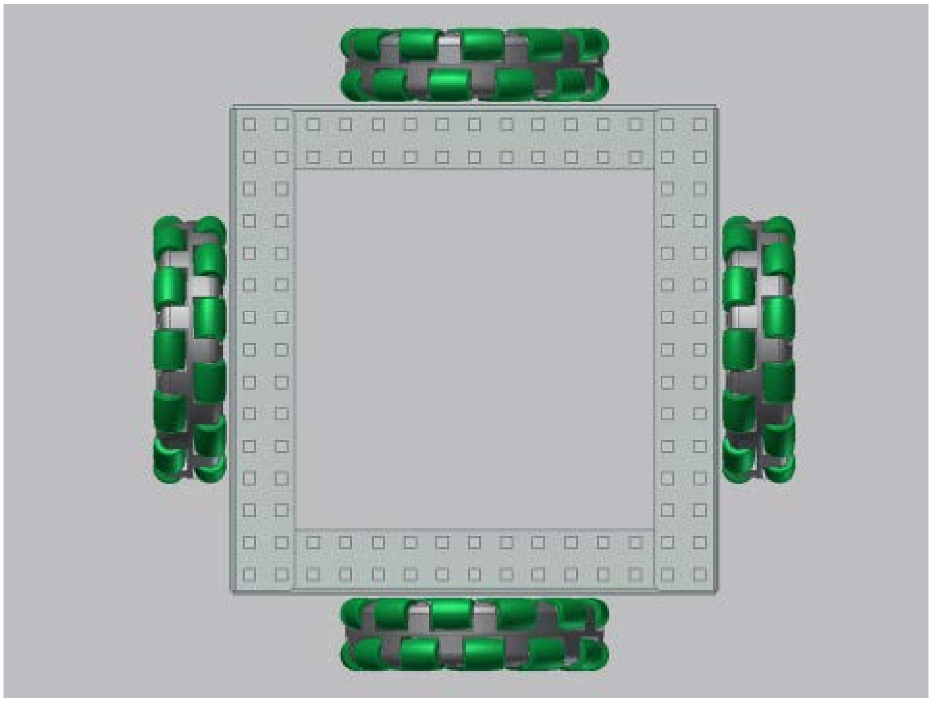
A Crab Drive system is one which utilizes two sets of skid-steer drivetrains, each pointed in different directions. Only one of these drives would be on the ground at a given time. For instance, a crab drive would have a primary skid-steer drive pointed forwards/backwards, and could drive around normally on this. Then it could drop a secondary drivetrain pointed right/left, lift its primary drive off the ground, and be able to move right or left (like a crab).

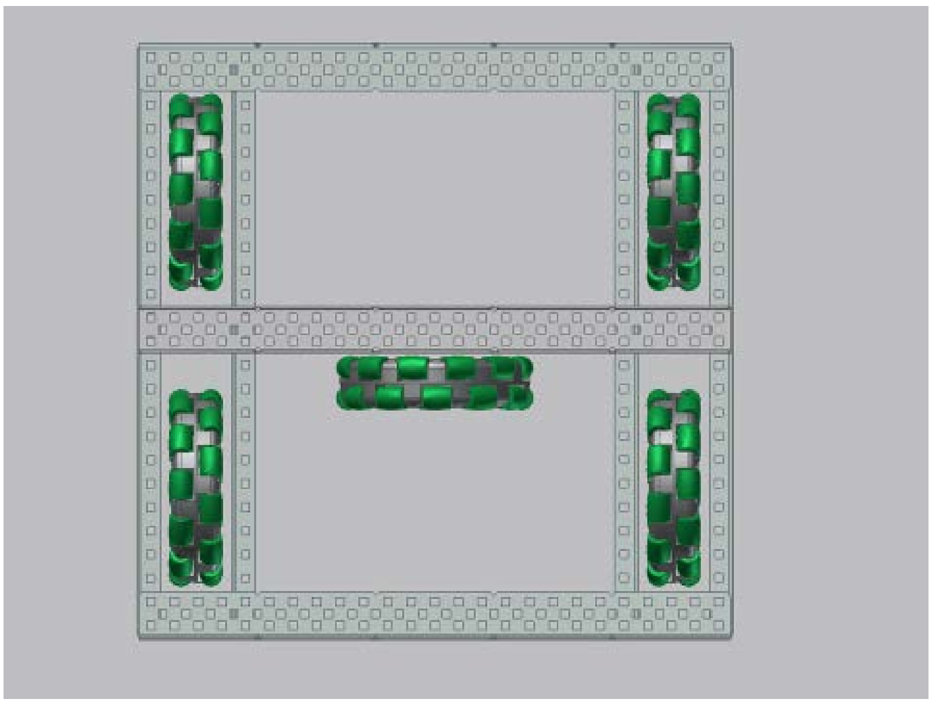
Omni-Directional Drivetrain:

A drivetrain that can move in any direction at a given moment, without waiting for wheels to steer is called an Omni-Directional Drivetrain. These drivetrains use special wheels, called “omni-wheels”. Omni-wheels are wheels with small rollers around the perimeter that freely spin perpendicular to the wheel’s rolling direction. This means that the wheels can slide sideways with very low friction.

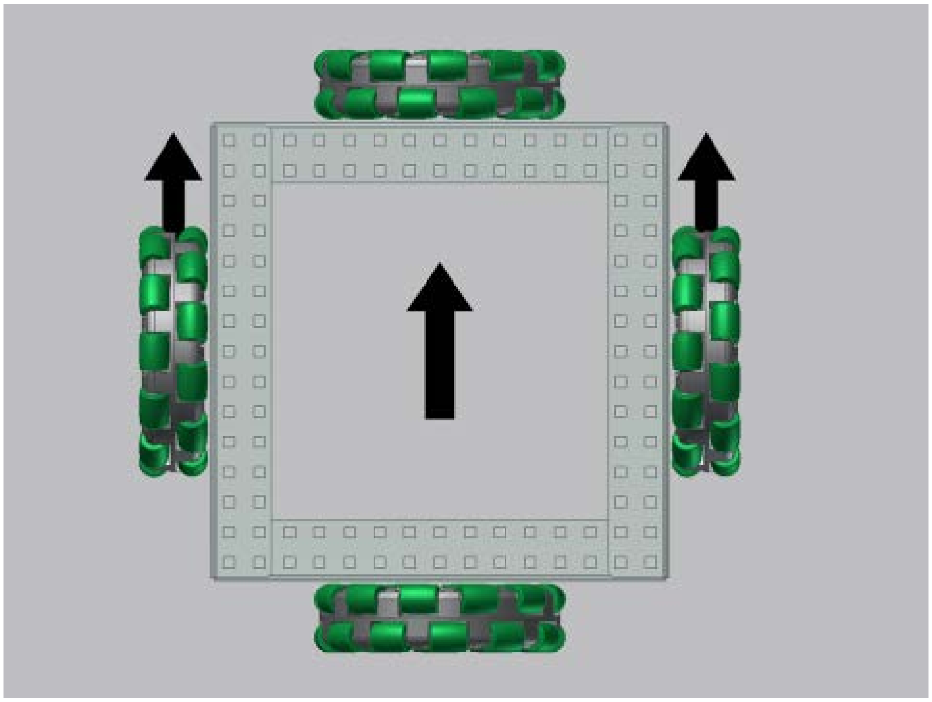
In the image above, the green rollers spin freely perpendicular to the way the wheel moves. The wheel can slide sideways on these rollers!

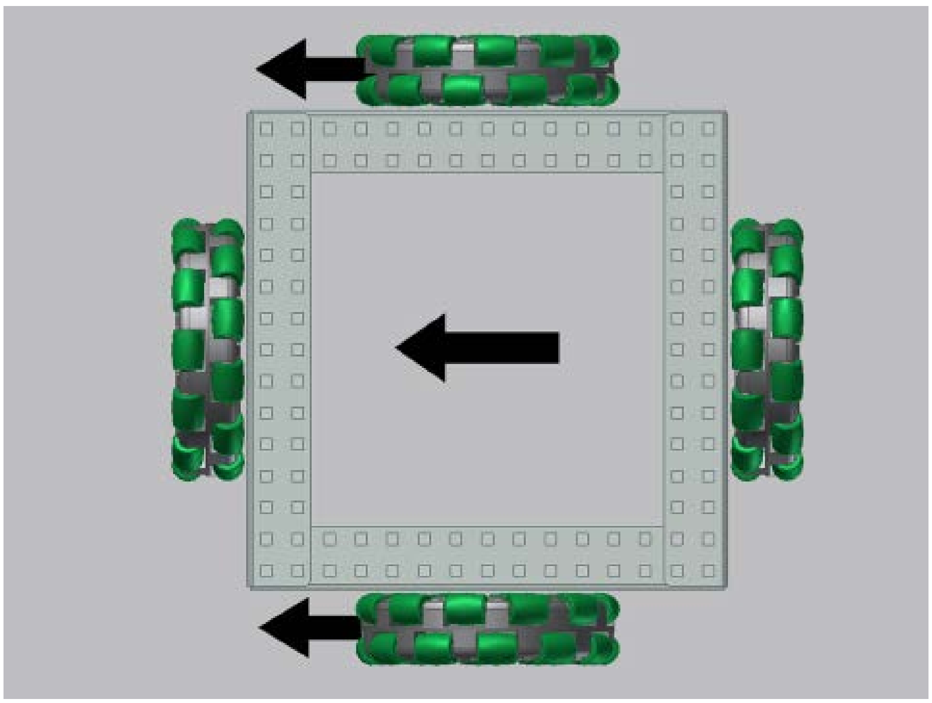
These wheels can be used in a variety of configurations to allow for omni-directional driving. Two common configurations are shown below:

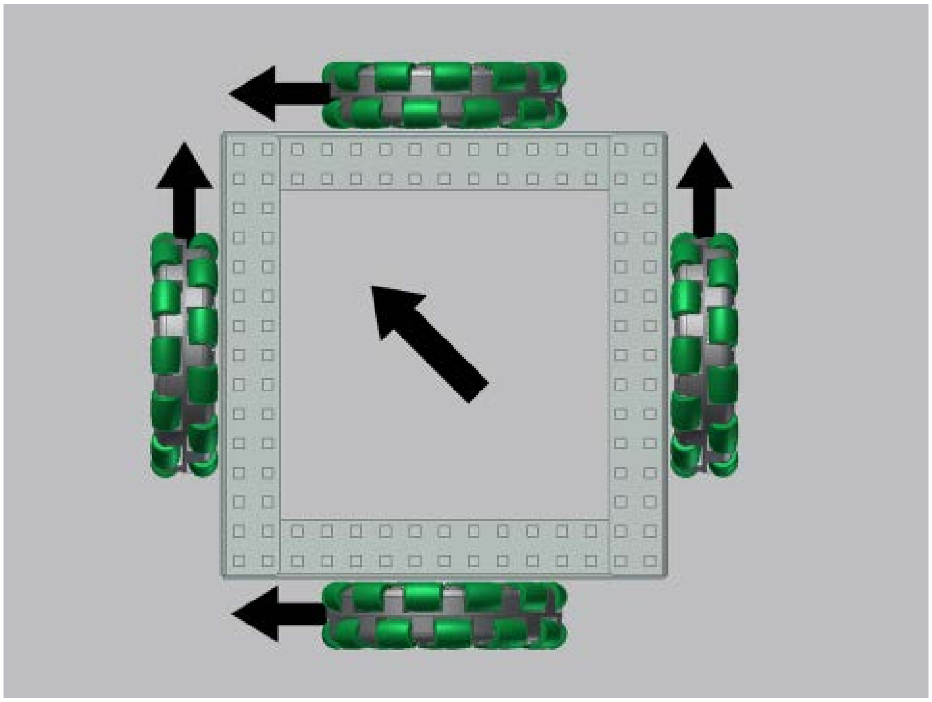




Since omni-wheels don’t have any sideways friction, the wheels facing forward/backward can drive without the wheels facing right/left dragging. By powering both sets of wheels, the robot can move in any direction.







One problem with omni-drives is the requirement for multiple motors, with only some of the motors contributing to the robots forward motion at most times. A simple skid-steer drive can be built with two motors, but a square omni-drive like the one above requires four motors.

9.2: Friction and Traction
up
9.4: Drivetrain Geometry and Turning

- Teacher Materials

9.4: Drivetrain Geometry and Turning

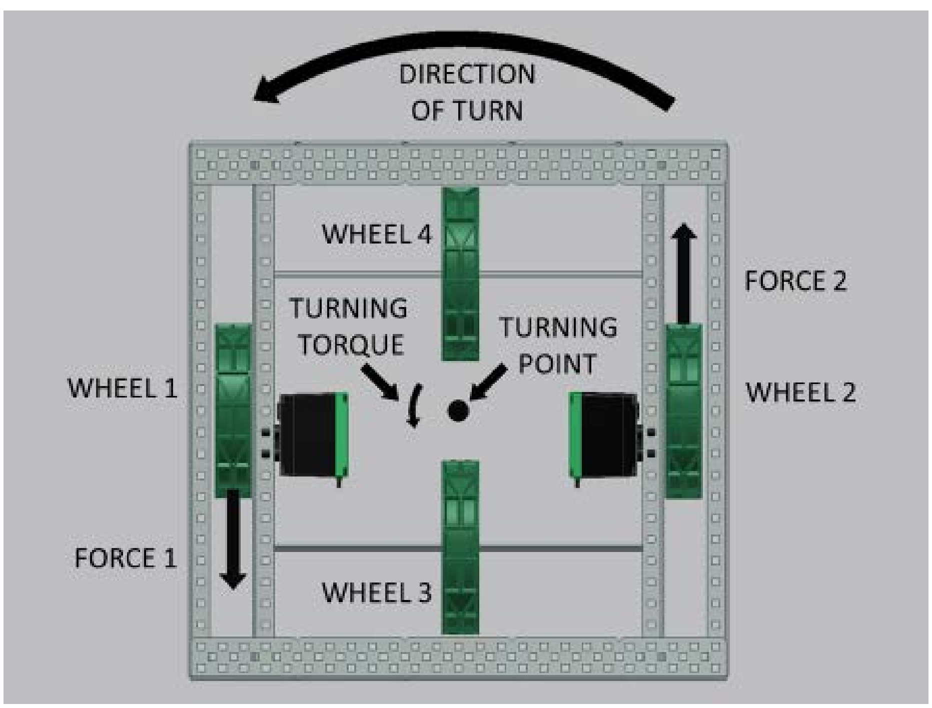
As mentioned above, one of the most common types of robot drivetrain is known as a Skid Steer drivetrain. This type of drivetrain consists of two independent sets of powered wheels, one on each side of its chassis. By running the sides of the drivetrain at different speeds, it is possible to steer the robot in arcs. This drivetrain is also capable of a zero-radius turn (it will spin in place) if the sides are run at the same speed in opposite directions.

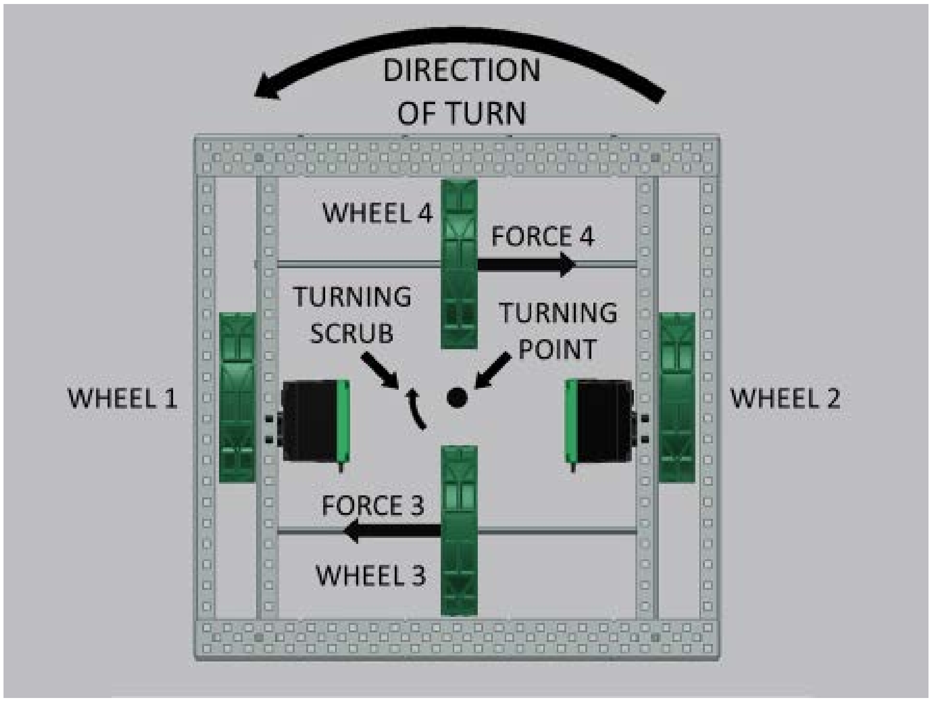
One of the major attributes that defines a drivetrain's performance is how well it turns. There are two main properties which affect drivetrain turning: Turning Torque and Turning Scrub.

Turning Torque is the torque about the turning point which causes the robot to turn.

Turning Scrub is the friction which resists the robot turning. This is caused by wheels dragging sideways on the ground as the robot turns, resisting the motion of the turn. Turning Scrub is also expressed as a torque about the turning point of the robot, opposing the turning torque.

In a typical skid-steer drivetrain (specifically one in which all the wheels are drive wheels), ALL the wheels will exert force that contributes to the turning torque, and they will ALL drag sideways and contribute to the turning scrub. To help visualize things more simply, one can think of a robot in the odd configuration seen below:

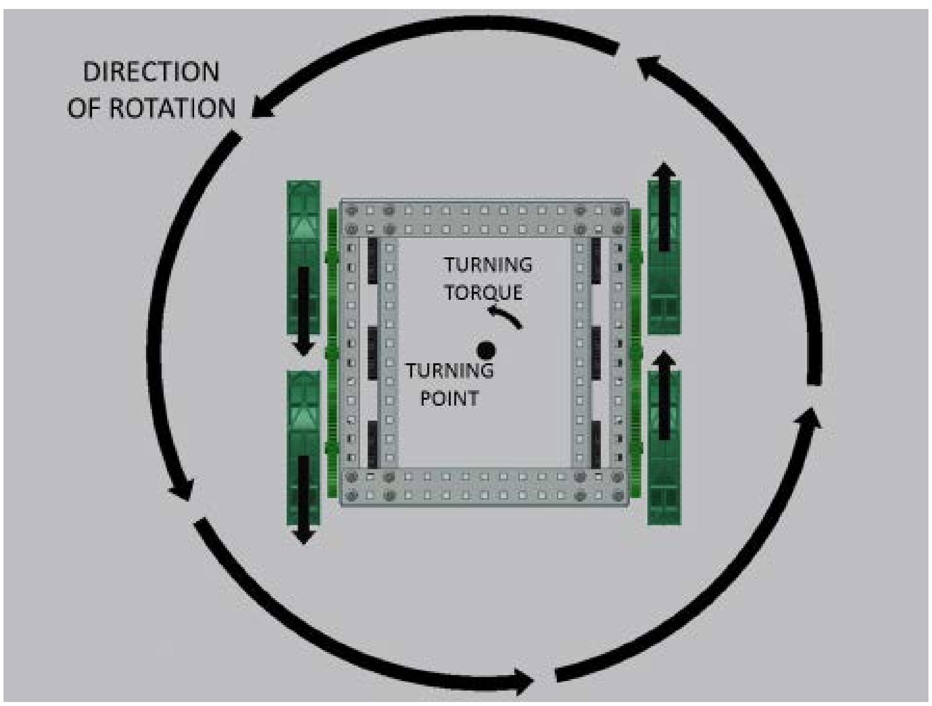


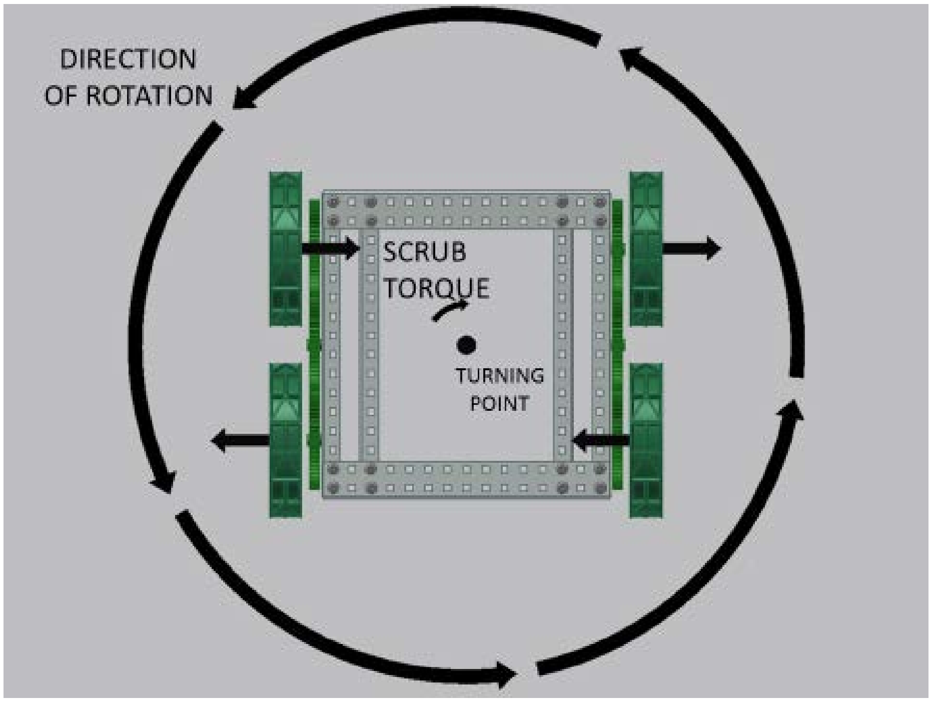


One can see that Wheel 1 and Wheel 2 contribute to the turning torque. They each exert a linear force (Force 1 & Force 2) that creates a torque about the turning point. These wheels do not drag at all as the robot turns, so they don't contribute to the turning scrub.

Wheel 3 and wheel 4 do not contribute to the turning torque, but they slide sideways as the robot turns and significantly contribute to the turning scrub. Force 3 and Force 4 are frictional forces from the wheels on the ground; this friction results in the turning scrub.

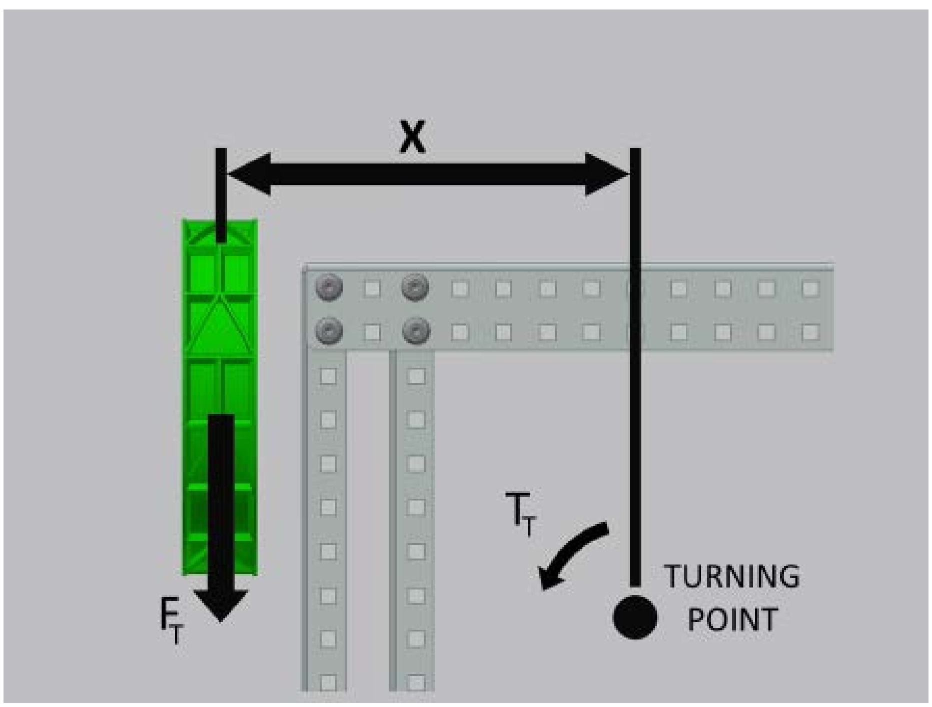
Now, in a more traditional drivetrain configuration, all the wheels would both contribute to the turning torque, AND the turning scrub:



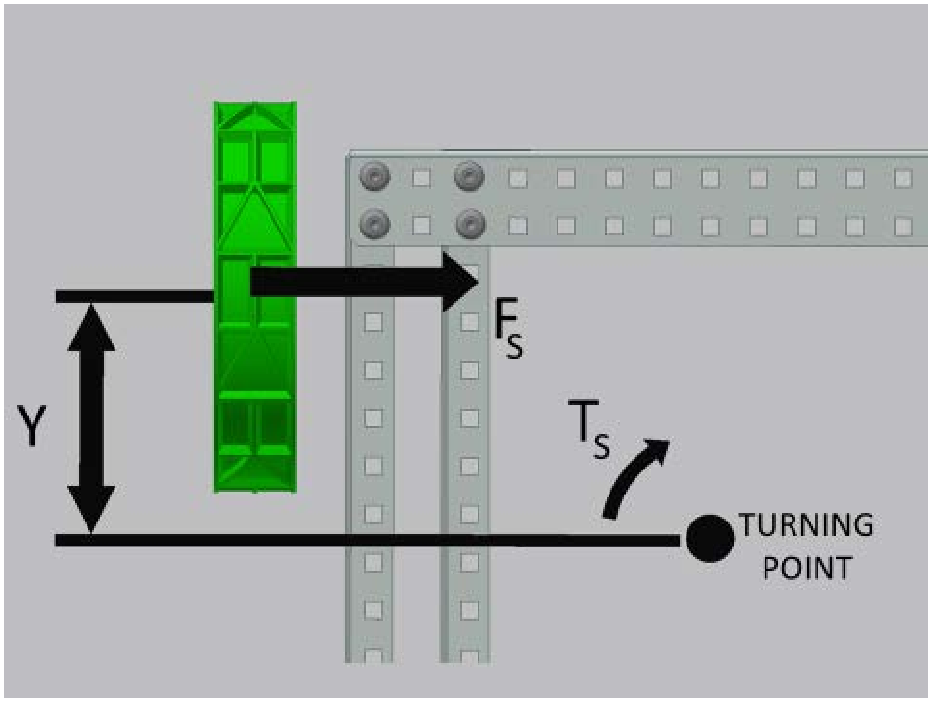


In the above case, all four wheels contribute to the Turning Torque, and all four wheels contribute to the Turning Scrub. Each wheel applies some force that contributes to turning, and each wheel needs to slide sideways and contributes some friction to scrub.

Turning Torque and Turning Scrub are both torques about the robot's turning point. As discussed in Unit 7: a torque is a turning force, defined by a linear force at a distance from some center of rotation. The below diagrams show how the frictional force of the wheel rolling forward contribute turning torque of the robot, and the frictional force of the wheel sliding sideways contributes to the turning scrub of the robot.



As seen above, the turning torque is contributed to by the friction force of the wheel at a distance from the turning point.



As seen above, the turning scrub is contributed to by the force of the wheel, around the robot turning point.

If a drivetrain has multiple wheels on the ground, all of these wheels will contribute based on their location in the drivetrain relative to the turning point.

DESIGNING A TURNING DRIVETRAIN:

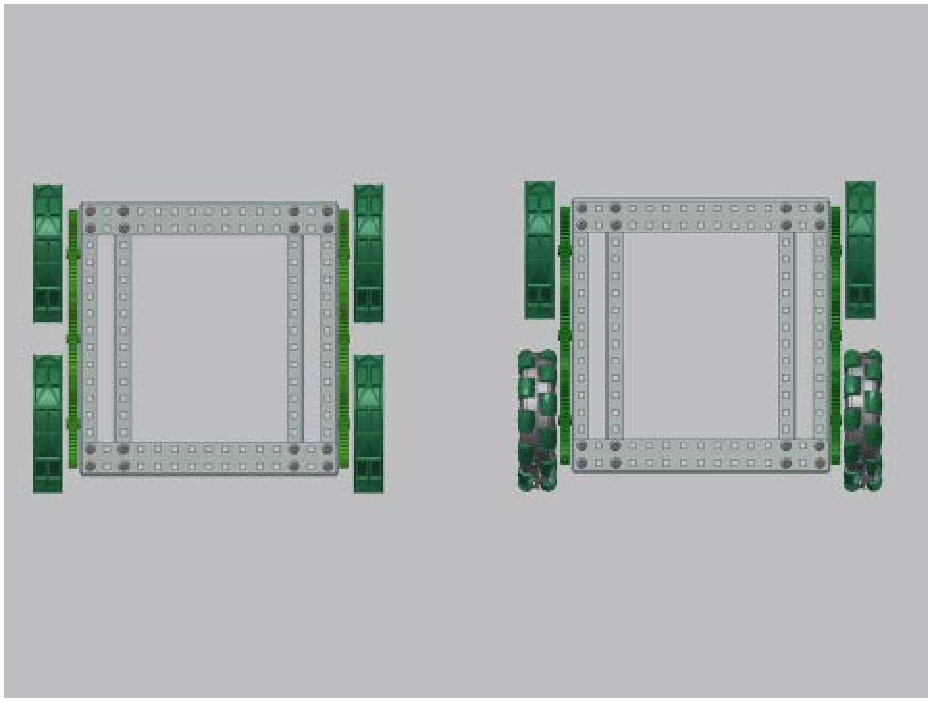
Once one understands the concepts of turning torque, turning scrub, and how they affect robot turning, one can begin to understand how to alter them to make a robot turn more effectively.

How does one reduce turning scrub?

Turning scrub is driven by the force of friction of the wheel sliding sideways on the floor. By reducing this frictional force, one reduces the turning scrub. One may also decrease the distance the wheel is from the turning point.

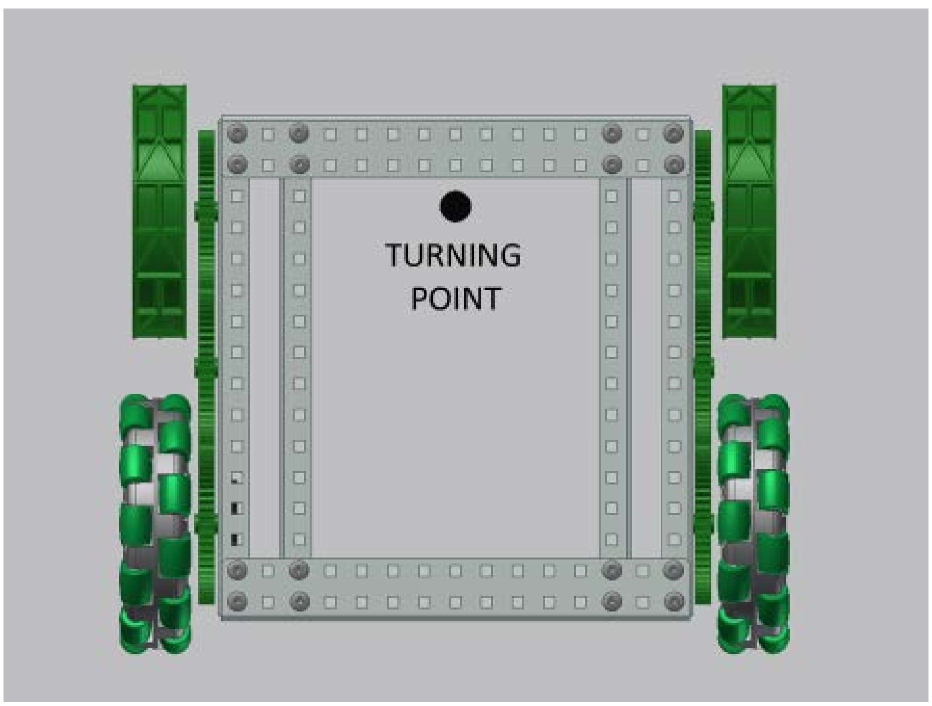
Similarly, one could increase turning torque with the opposite approach; by increasing the frictional force, or increasing the distance from the turning point.

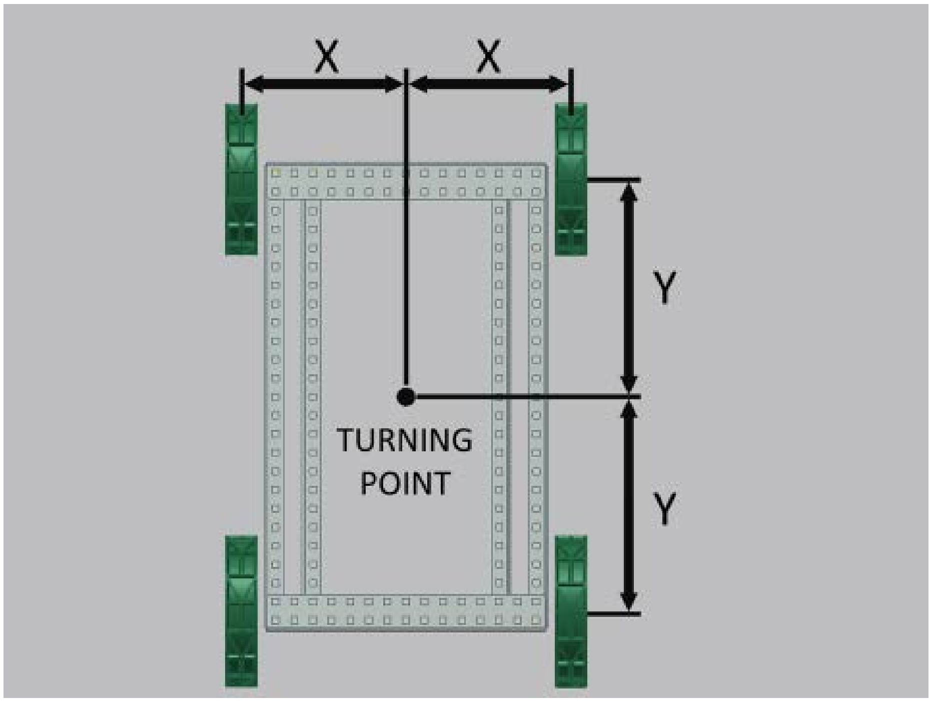
Notice that to decrease turning scrub, one must decrease the wheel friction in the left/right direction. To increase turning torque, one must increase the friction of the wheel in the front/back direction. It is difficult to modify the friction of a wheel in one direction without affecting the other, so it is usually best to modify the geometry of the robot chassis to help improve robot turning.



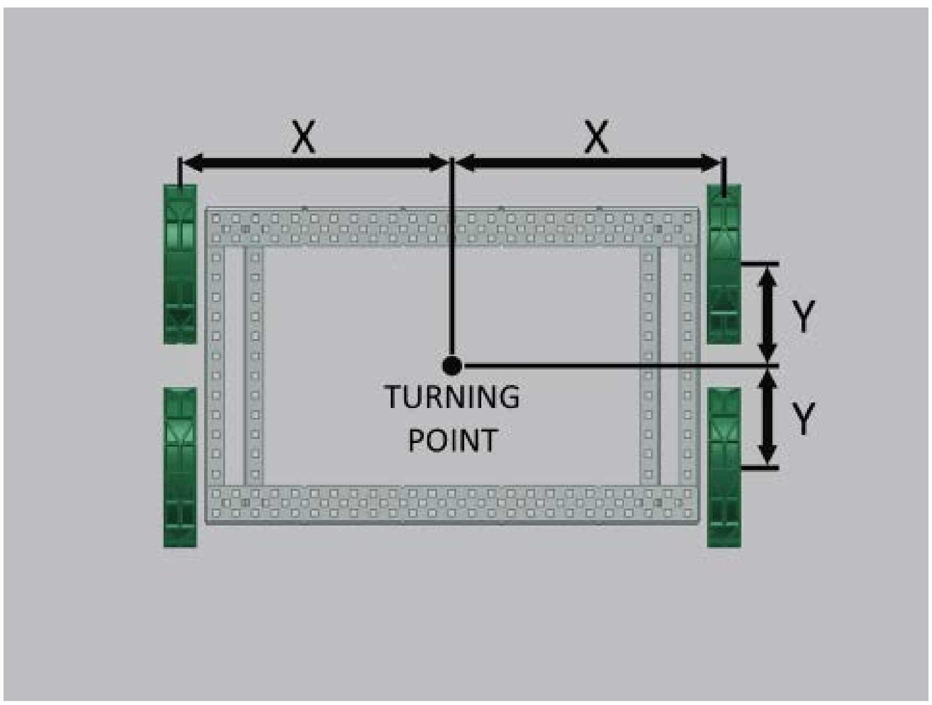
However, designers should note that omni-directional wheels have ZERO sideways friction. This means a drivetrain with an omni-wheel would have NO turning scrub caused by that wheel. A drivetrain with ALL omni-wheels would have almost ZERO turning scrub!

It is interesting to note that a drivetrain with two omni-wheels and two traction wheels would have a turning point directly between the two traction wheels. This drivetrain would also have no turning scrub, since the traction wheels would not need to slide sideways.





The above example shows a drivetrain configuration that is long and narrow. This configuration would likely have poor turning characteristics because of its low turning torque and high turning scrub.



The example above shows a drivetrain configuration that is short and wide. This configuration would likely have very good turning characteristics because of its high turning torque and low turning scrub.

All of the examples discussed so far have been simplified to help illustrate their major underlining concepts. There is another important consideration that will change the dynamics of these systems – the location of the turning point. In all the examples seen so far, the turning point has been in the exact center of the robot; this is not always the case.

The turning point will often vary based on the differences between the wheels (front vs. back, or left vs. right). This is primarily based on the friction between the individual wheels and the floor. As discussed previously, this friction is dependent on the weight resting on the wheels, and the coefficient of friction of the wheels. This means that if most of the weight is towards the front of the robot, the turning point would be towards the front.

The traction of the different wheels and the location of the weight of the robot will greatly affect where the turning point is, and will affect the turning torque and turning scrub of the robot.

To recap – in order to make a robot turn better one should primarily adjust three things: the chassis geometry (wide vs narrow, long vs short), the difference in coefficient of friction between the various wheels (primarily front vs back) and the location of the robot's center of gravity.

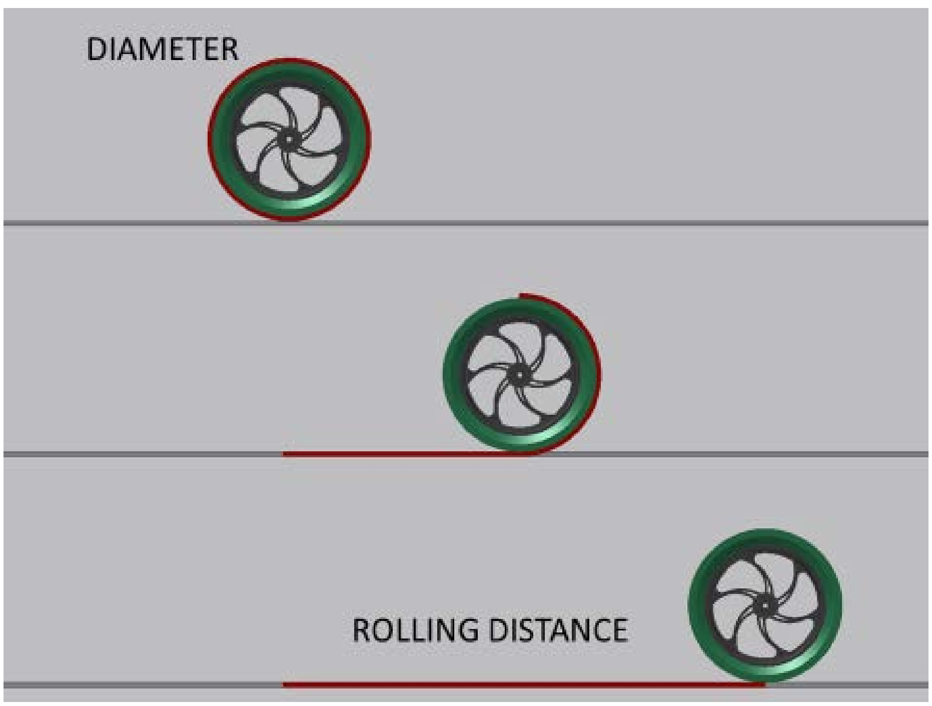
9.3: Drivetrain Terminology
up
9.5: Gear Train Design

9.5: Gear Train Design

The gear train represents the part of the drivetrain that transmits power from the motor, to the wheel.

Wheel Speed:

The first concept to understand is how to figure out how fast the robot moves across the field based on how fast the wheel is spinning. For each time the wheel makes a full rotation, it will roll forward a distance equal to its circumference. So if one calculates the circumference of the wheel, one knows how far the robot goes per wheel revolution.



The circumference of a wheel is equal to its diameter multiplied by Pi (a mathematical constant which is about 3.14).

Once one knows the circumference of the wheel, it is possible to calculate how fast the robot is travelling based on the wheel's rotational speed. In the example above, the wheel diameter is 101.6 mm [4 inches] and is spinning at 100 RPM (revolutions per minute). So based on this, how fast is the robot travelling (in mm/second)?

Circumference = Diameter x Pi

Circumference = 101.6 mm x 3.14

Circumference = 319.024 mm

So the robot moves 319.024 mm per 1 wheel revolution. The wheel is moving at 100 revolutions per minute, or 100 revolutions per 60 seconds.

From this, one can calculate the linear ground speed of the robot:

$$\frac{319.024\ \text{mm}}{1\ \text{revolution}} \times \frac{100\ \text{revolutions}}{60\ \text{seconds}} = \frac{531.707\ \text{mm}}{1\ \text{second}}$$

So the robot is moving at about 532 mm/second or 0.532 meters/second.

Armed with this method, and knowing the specifications for VEX motors, one can determine the necessary gear ratio for a VEX robot to hit a desired top speed.

EXAMPLE – Calculating Gear Reduction to hit a Desired Top Speed:

One can consider the case of a robot that has a wheel with a diameter of 69.85mm (2.75 inches) and a motor that has a free speed of 100 RPM. In this case, if the designer wants the robot to have a desired free speed of 900 mm / s, what gear reduction is required? (One needs to use the knowledge of Gear Ratios discussed in Unit 8).

The first step in calculating this is to determine what RPM the wheel is required to spin at to achieve the desired top speed of 900 mm / s.

Circumference = Diameter x Pi

Circumference = 69.85 mm x 3.14

Circumference = 219.329 mm

So the robot moves 219.329 mm per 1 wheel revolution. Converting the goal speed into RPM based on this circumference results in the following:

$$\frac{900\ \text{mm}}{1\ \text{second}} \times \frac{1\ \text{revolution}}{219.329\ \text{mm}} = \frac{4.103\ \text{revolutions}}{1\ \text{second}}$$

$$\frac{4.103\ \text{revolutions}}{1\ \text{second}} \times \frac{60\ \text{seconds}}{1\ \text{minute}} = \frac{246.18\ \text{revolutions}}{1\ \text{minute}}$$

If the wheel needs to spin at 246.18 RPM, and the motor spins at 100 RPM one can calculate the required gear reduction using an equation from Unit 8:

Gear Reduction Required = Input Speed / Output Speed

Gear Reduction Required = 100 RPM / 246.18 RPM

Gear Reduction Required = 0.4062

So the designer needs to use a gear reduction of 0.4062 or less to achieve a top speed of greater than 900 mm / s.

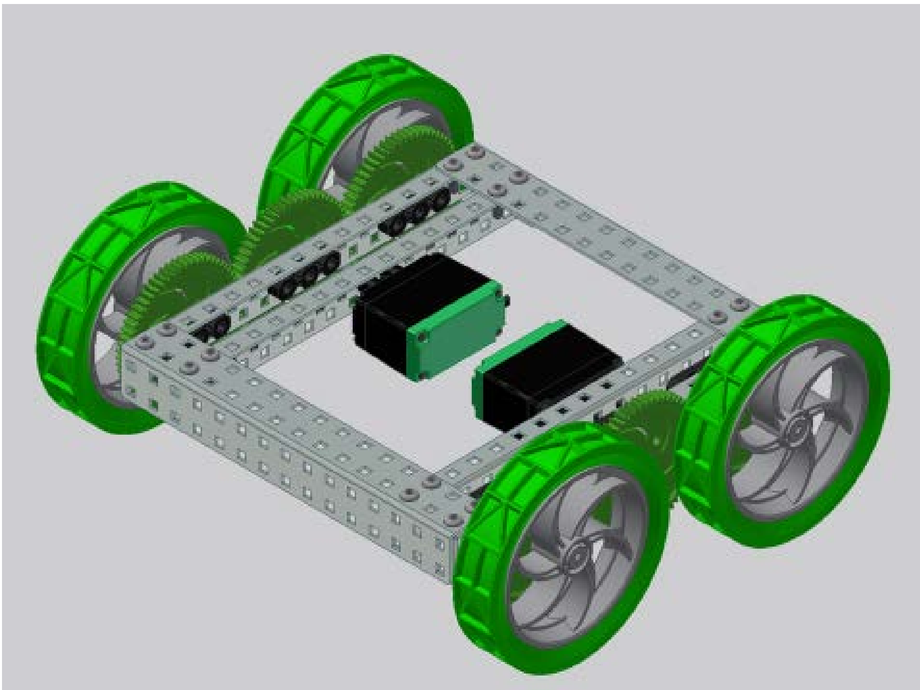
Motor Loading & Gearing:

The second concept designers must consider when designing drivetrains is how motor-loading affects gear train design. In particular, it is important to consider the maximum load applied to the motor by the drivetrain. This occurs during a situation where a robot is pushing against a stationary (immovable) object, and is running full throttle into it. In this situation the wheels should slip on the floor, and the friction between the wheels and the floor will act as a brake on the motor.

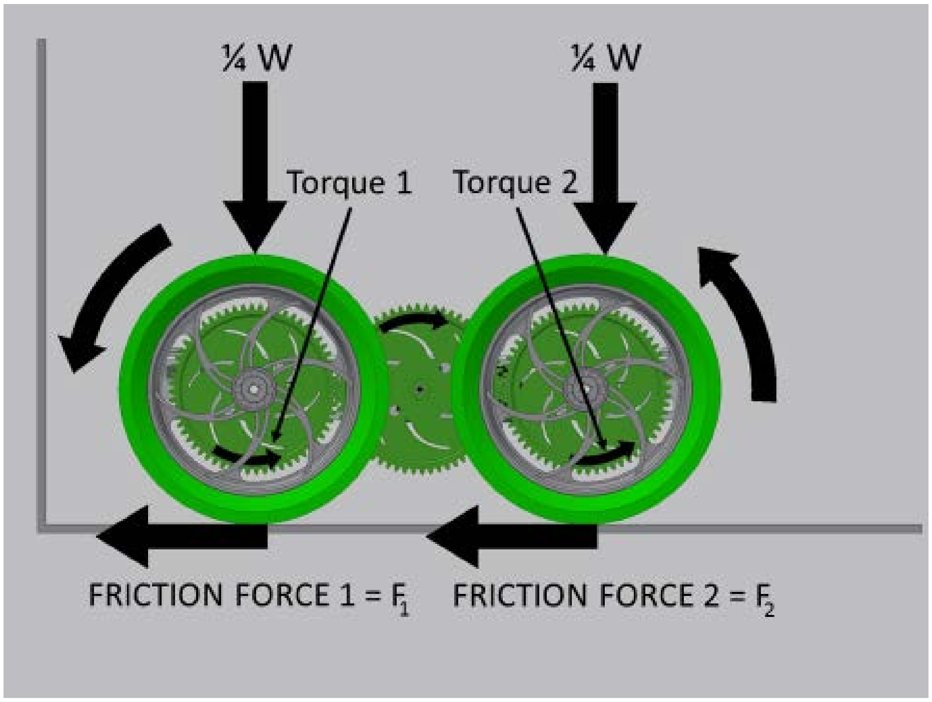
The first step is determining how many wheels are acting as a brake on the gearbox. Only wheels directly linked through gearing or chain will apply load to the gearbox and motor.

The second thing to consider is to determine how much of the robot's weight is resting on each of those wheels. As discussed earlier, the traction between the wheels and the floor is dependent on the normal force pressing them together.

As an example, one could consider the robot below:



In this case, the robot's weight is evenly divided among the 4 robot wheels, and each motor is directly linked through gearing to two of the wheels (right vs. left). This means that each motor has 1/2 the robot's traction acting on it as a brake.



As shown in the above image, the friction of each wheel creates a torque which opposes the motion of the motor. Both of the torques contribute load on the motor.

If the gear train has multiple motors associated with it (i.e. two motors driving one geared set of wheels), then the torque will be divided evenly between them.

It is important to design the gearing such that the load applied on each motor is not higher than the motor limit (as described in Unit 8.) Designers should use the principles of gear reduction to ensure that the motor limit is not exceeded; when in doubt, gear the robot slower for less loading.

Using the two concepts discussed above, along with those from Unit 7 and Unit 8, designers should be able to gear a robot so it moves at a desired speed. They should also ensure there is no excessive load on the motors.

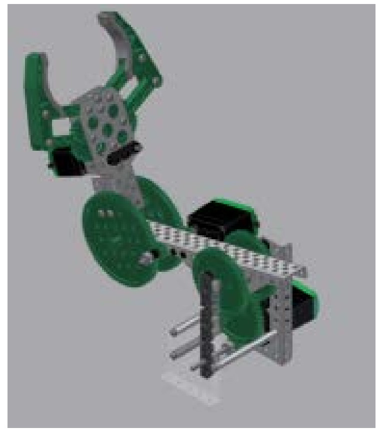
9.4: Drivetrain Geometry and Turning

up

9.6: Design Activity

Unit 10: Lifting Mechanisms

In this unit students will learn about different types of lifting mechanisms which are useful on competition robots. Students will then do preliminary design work on a mechanism for their robots.

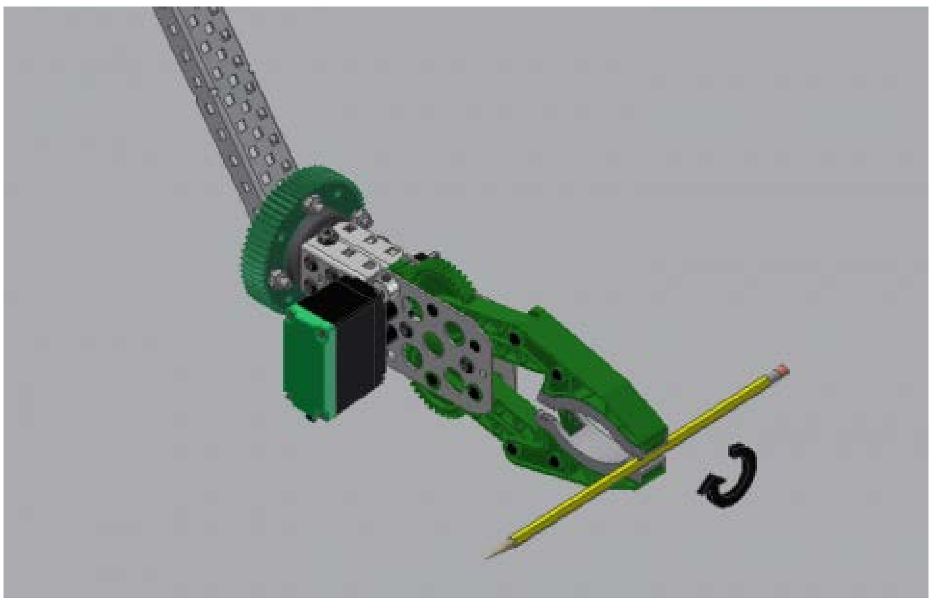


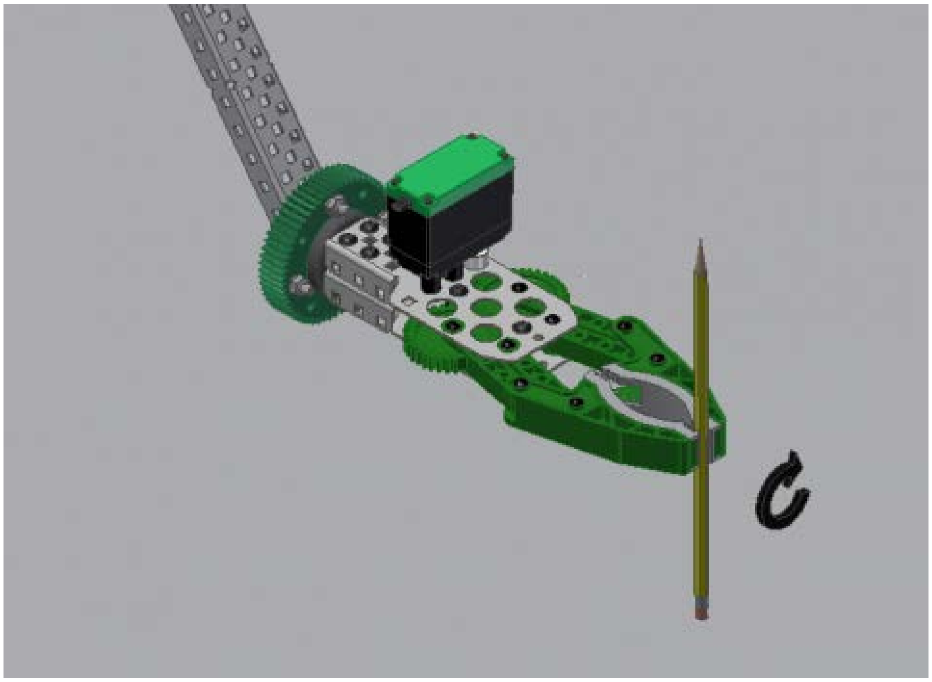
9.7: Engineering Notebook up 10.1: Introduction

10.2: Degrees of Freedom

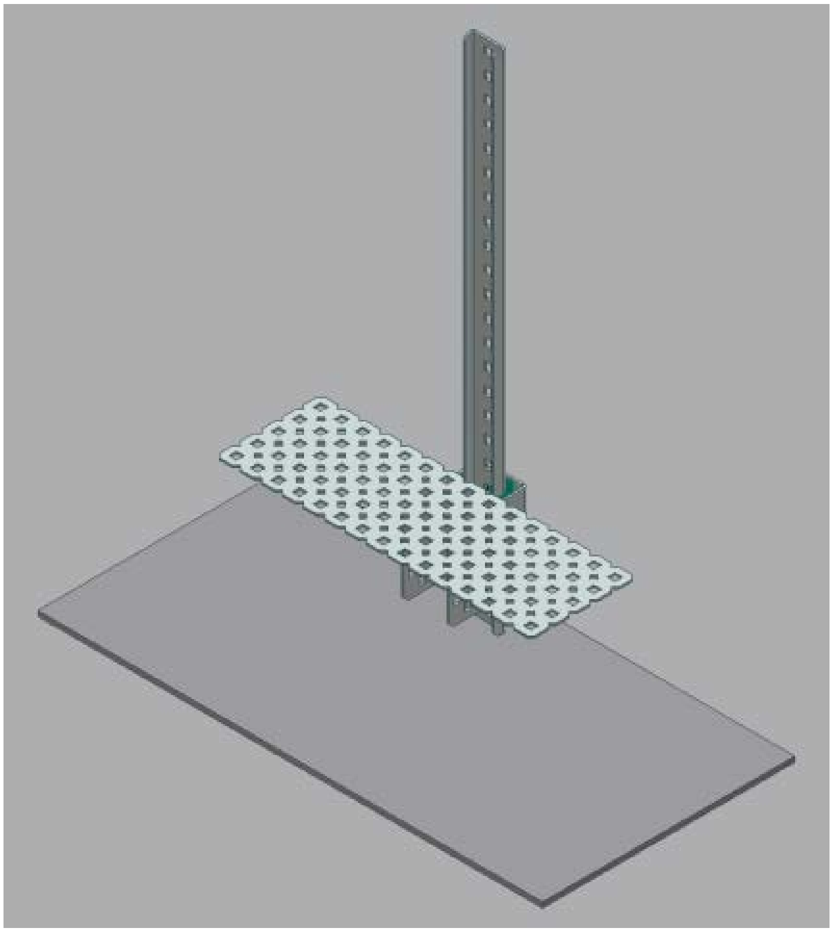
A degree of freedom refers to a something's ability to move in a single independent direction of motion. To be able to move in multiple directions means to have multiple degrees of freedom. Moving up & down is one degree of freedom, moving right & left is another; something that can move up/down and left/right has TWO degrees of freedom.   In the context of competition robotics, three basic degrees of freedom will be discussed.

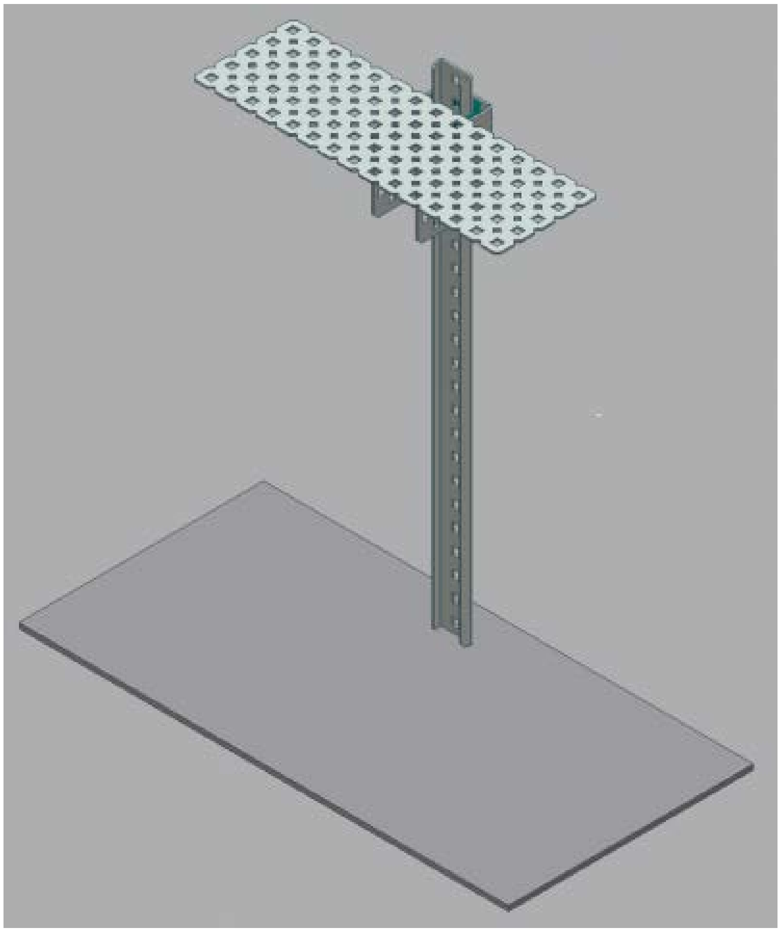
The first type of degree of freedom is one in which the robot's arm can rotate about an axis parallel to the arm. This degree of freedom is found in the human wrist. One can imagine someone holding a pencil in his or her fist so it is parallel to the floor (horizontal). Now this same person twists his or her wrist such that the pencil is pointed straight up at the ceiling (vertical). This "twisting" is one degree of freedom.



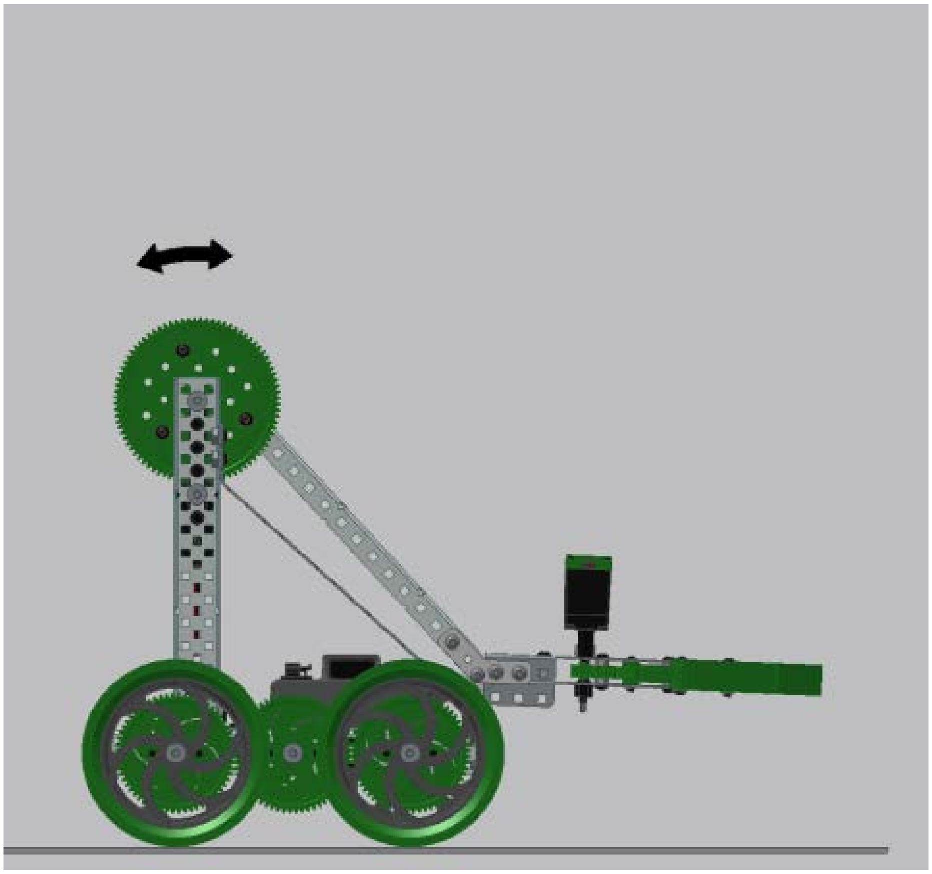


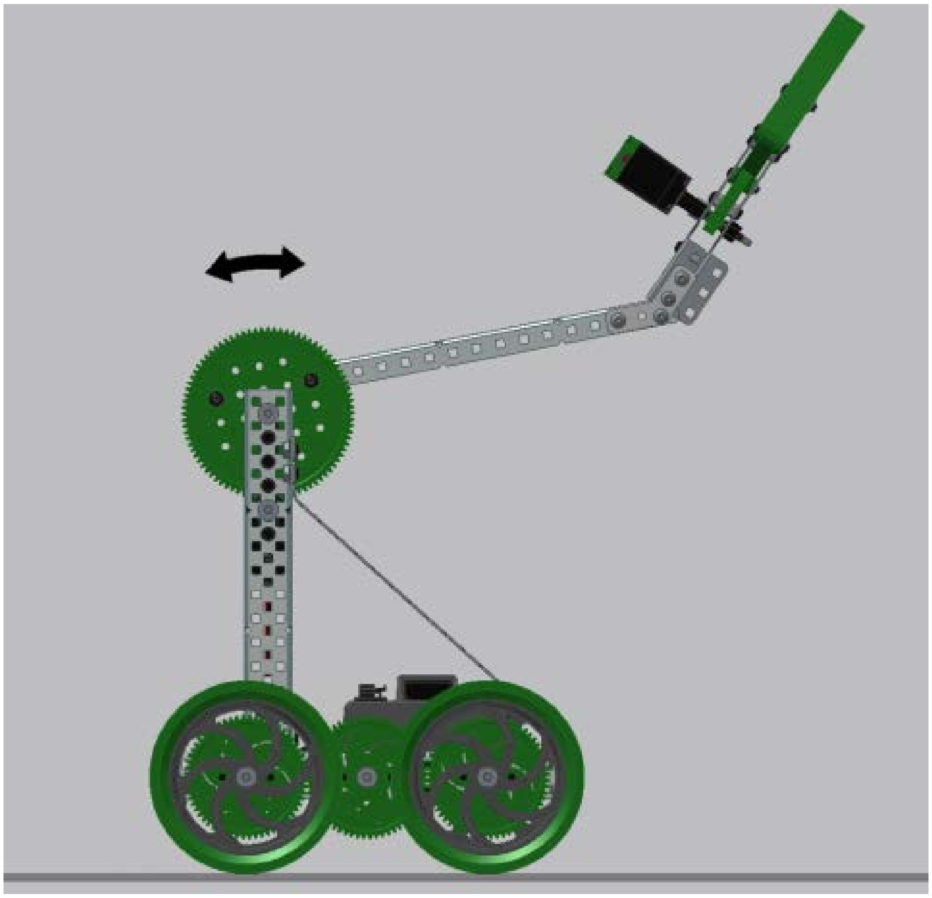
The second type of degree of freedom is linear movement. In this case, a component on a robot can slide in and out (or up and down, or left and right). An elevator represents this linear degree of freedom (moving up and down), as does a common desk drawer (moving in and out).





The third type of degree of freedom is one in which there is rotation about an axis perpendicular to an arm. The human elbow is an example of a joint with this degree of freedom. The arm-mass systems discussed in Unit 7 and Unit 8 are also examples of this degree of freedom.





Exercise: Students should try to determine how many degrees of freedom the human arm has. Hint: some of the joints have multiple degrees of freedom!

Competition robots typically only require a few degrees of freedom, since they don't need to be as versatile or dexterous as a human arm. Best practices for competition robot lifting mechanism design will be discussed in more detail in this unit.

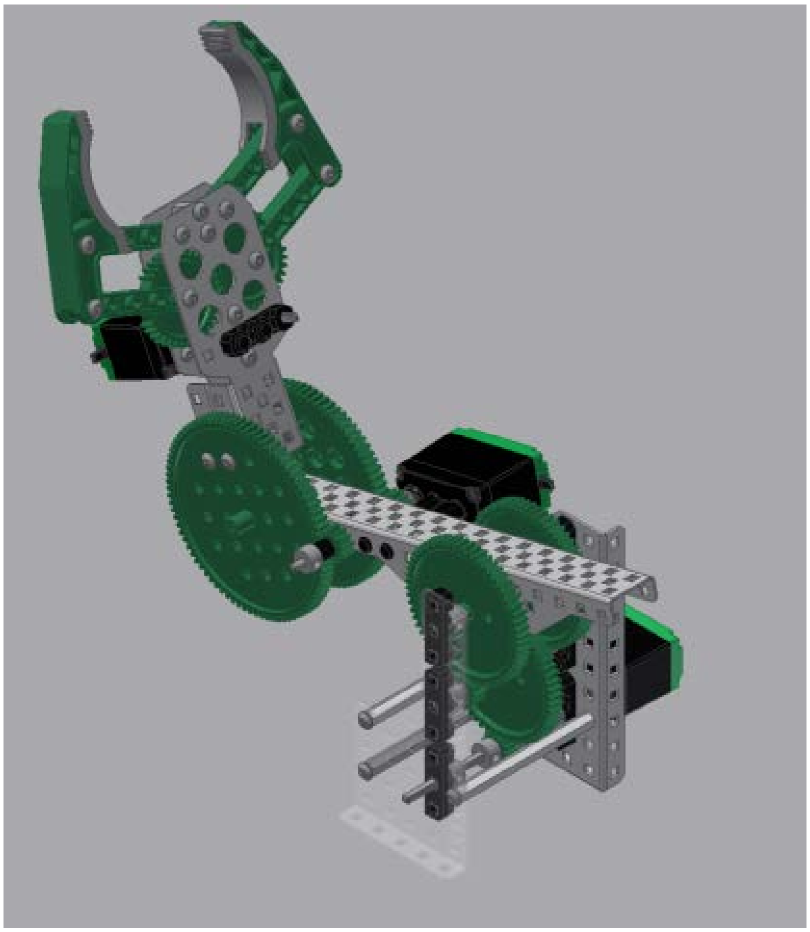
10.1: Introduction

up

10.3: Rotating Joints

10.3: Rotating Joints

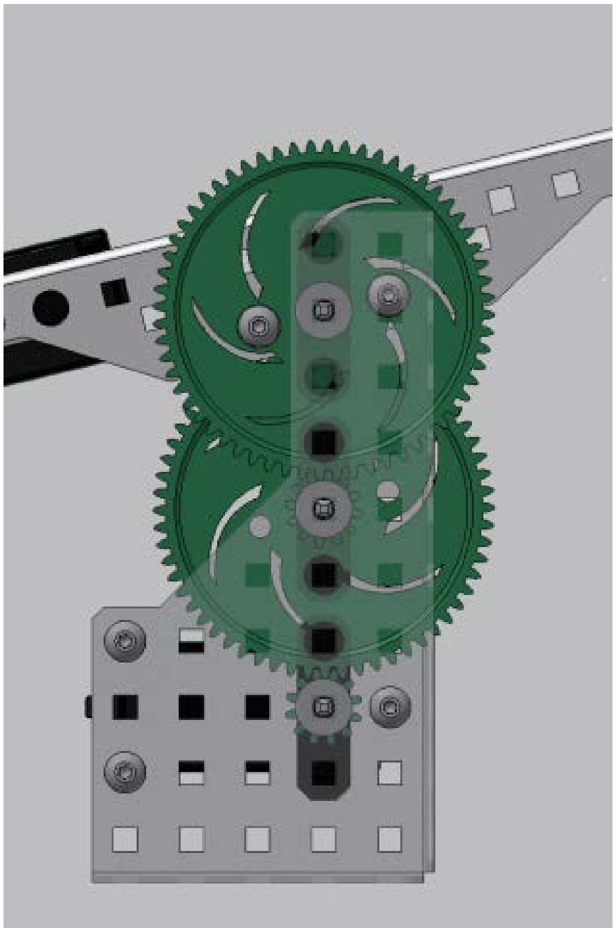
The most frequently used lifting mechanism in competition robotics is a rotating joint.



The above example shows a 2-jointed arm, which includes a shoulder joint and a wrist joint (note that joints on a robot arm are often named similarly to a human arm). These joints are constructed by locking part of the robot structure onto part of a motion system. In the above case, the shoulder joint includes an arm locked to the gears of the shoulder gearbox – as these gears turn, the arm turns as well. Similarly, the claw is locked to the gears of the gearbox attached to the end of the arm.

One might notice in the above example that the shoulder joint has a much higher gear reduction than the wrist joint. This is done because of motor loading. The shoulder joint needs to lift the weight of the arm, the weight of the wrist joint, the weight of the claw, and whatever object the claw has picked up. The wrist joint only needs to lift the weight of the claw and the object it has grabbed. The shoulder joint also needs to lift this weight with a much longer lever arm than the wrist joint, so it is under significantly more torque. Each joint is designed to handle the loads which will be applied to it. These concepts were discussed in Unit 7 and Unit 8.

An example of a double reduction rotation joint is seen below.



This joint has two stages of 12:60 gear reductions. The second stage is attached directly attached to the robot arm. The second stage also has two of the same gear reduction running in parallel; this means that the load is divided evenly over these two sets of gears. By reducing the load on individual components, the joint is less likely to have a failure (broken gear, etc).

A SHOCK LOAD is an instantaneous spike of loading on a mechanical system. Imagine someone grabs onto the arm of a robot and pushes down on it as hard as they can in a jerking motion. This would apply a shock load to the arm gearbox. It is always a good idea to design robot joints robustly enough to withstand these shock loads. Running multiple parallel gear reductions will spread shock loads and prevent damage.

The first stage of gear reduction does NOT utilize parallel reductions; it has only one set of meshing gears. This is because the impact of a shock-load is lowest at the part of the gearbox closest to the motor. Typically gear stages closest to the motor do not need to be as robust as stages further into the gearbox.

Joint Loading:

In order to effectively utilize a motor, one needs to adjust the load applied to its motor such that it runs within acceptable parameters. Methods for using mechanical advantage and gear reduction to optimize motor loading were discussed in Unit 7 and Unit 8.

If there is some force applied at the end of the robot arm (as discussed in Unit 7) it will apply a torque on the joint equal to the magnitude of the force multiplied by its distance from the joint. This applied force is partially caused by the weight of the arm itself, as well as any forces the arm will encounter during operation (weight of objects it is lifting, etc.)

When choosing the appropriate gear reduction for an arm, it is a good idea to add a Factor of Safety to these forces to ensure the joint can handle any unanticipated loads it encounters. A factor of safety, also known as a margin of safety, describes the amount of overage the system can handle. Basically if the robot needs to be able to lift with a force of 10 N, one might design the robot to handle 12 N. This would be a factor of safety of 1.2 (10 * 1.2 = 12). This factor of safety will take care of any unexpected loads the arm encounters. It is always a good practice to take the unexpected into account when designing.

Joint Speed:

Often it is important for a joint to move as quickly as possible; however, this is not always practical. Designing a joint to be too fast may make it uncontrollable without advanced software.

There are two approaches to choosing the gear reduction required for a rotating joint. Either method will work for any rotating joint design. If the amount of load the joint can lift is critical, designers should use Approach 1. If the speed of the joint is more important than what will be lifted, designers should use Approach 2.

Approach 1 – Start with Loading:

Step 1 – Determine the applied load on the end of the arm, and the length of the arm.

Step 2 – Determine the maximum load which can be applied on the motor. (Refer to Unit 7).

Step 3 – Determine the required gear reduction which results in the desired motor loading.

Step 4 – Calculate how fast the arm will be rotating with this gear reduction.

Step 5 – Determine if this is a “good” speed.

Step 6a – If it is “good”, build it.

Step 6b – If it is too fast, determine how fast the joint needs to move, and then calculate the gearing required for this speed (it will be slower than the previously calculated speed). Then build it.

Step 6c – If it is too slow, add additional power to the system so it can handle this load at a faster speed (add additional motors to this joint). Then recalculate.

Step 6d – If the end result does not meet expectations, the constraints of the system need to be reevaluated and a redesign should be considered.

Approach 2 – Start with Speed:

Step 1 - Determine the desired speed of the joint, i.e. 90 degrees per second.

Step 2 – Determine the required gear reduction to achieve this desired speed. (Refer to Unit 8).

Step 3 – Determine the maximum load which can be applied on the motor. (Refer to Unit 7).

Step 4 – Determine the maximum load which can be lifted by the arm based on the gear reduction from step 2 and the maximum motor load from step 3.

Step 5 – Determine if this maximum lifting load is acceptable based on what the load on the joint will likely be (including safety factor).

Step 6a – If the load is “good”, build it.

Step 6b – If the load is too low, and the designer is willing to reduce the speed of the joint to accommodate this load, he or she should redesign based on Approach 1, above.

Step 6c – If the load is too low, and the designer is NOT willing to reduce the speed of the joint, he or she should add additional power to the joint and then recalculate.

Also note: reducing the length of the arm attached to the joint will reduce the amount of load on the motor (as discussed in Unit 7). This is one of the options a designer can choose if the output result does not meet his or her expectations.

Both of the above approaches will work well for designing a rotating joint. Each process requires iteration to be successful, and designers may find themselves doing and redoing calculations in an effort to find a design which works best for their given situation.

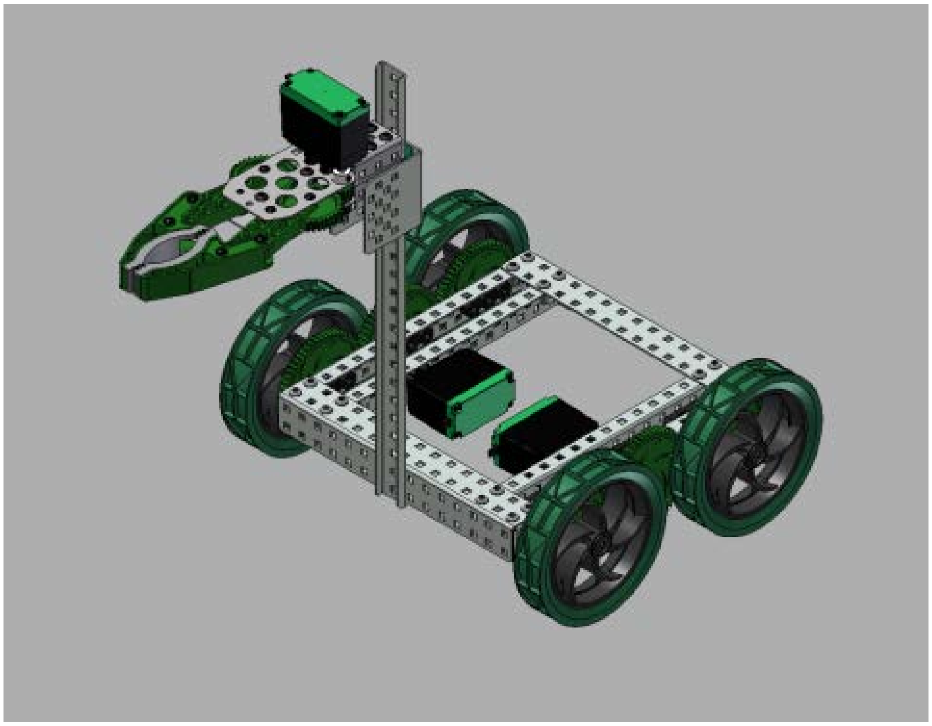
10.2: Degrees of Freedom

up

10.4: Elevators

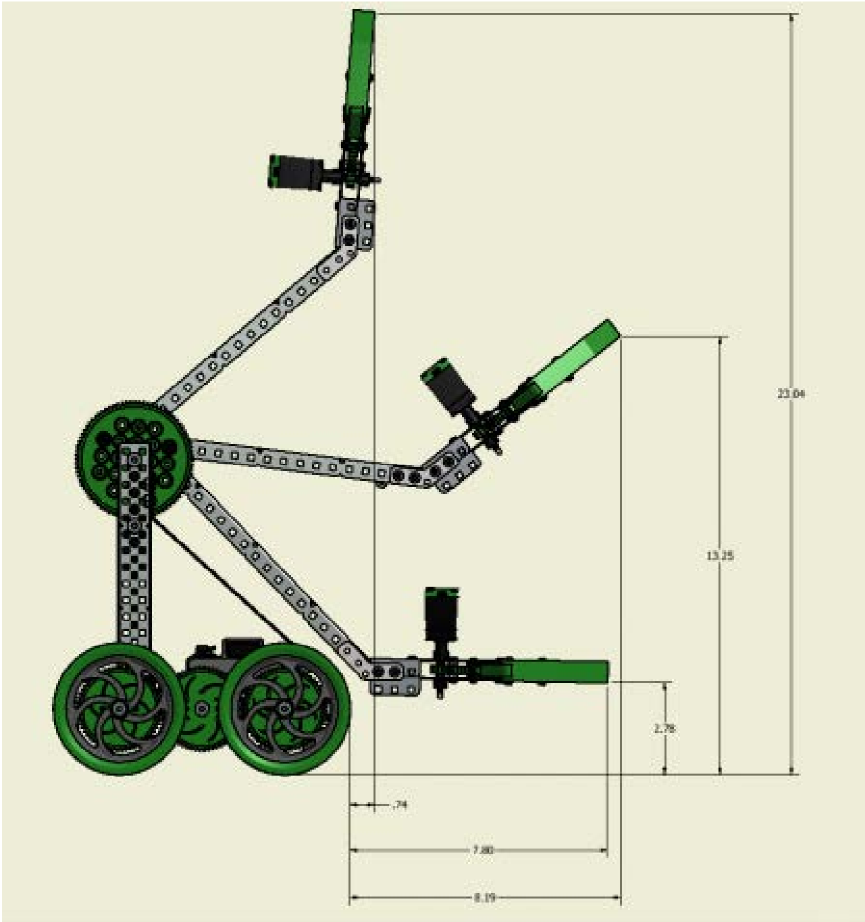
10.4: Elevators

Another lifting mechanism commonly used in competition robotics is a linear elevator.



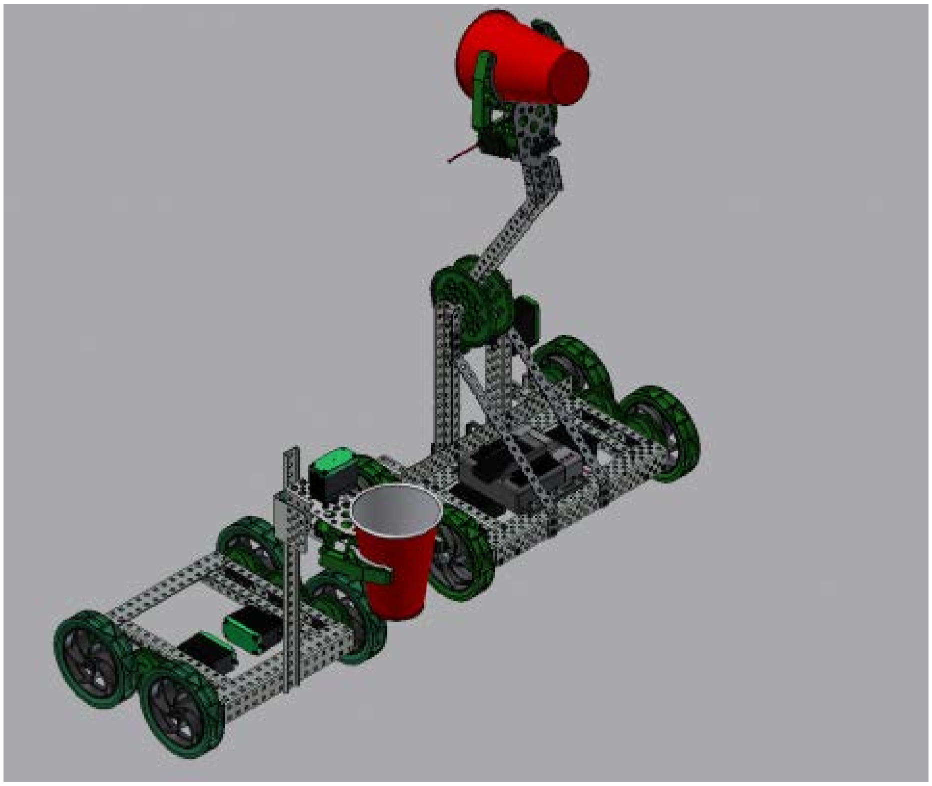
The VEX Robotics Design System has many different ways to create rotational mechanisms and low-friction rotating joints but has only a few methods for low-friction linear motion. As a result, this type of mechanism is less common than a rotating joint on competition robots; however, it does have some advantages.

As seen in the example below, as the rotating joint raises upwards the robot's claw will swing in an arc about the point of rotation. This means the distance between the claw and the drive base varies depending on the height the claw is at. If a driver is trying to score in goals of different heights, the driver must position the robot the correct distance away based on the height of the goal.



A claw mounted on a linear elevator is always a consistent distance from the drivebase, so regardless of the goal height, the driver will need to move the robot to the same distance from the goal.

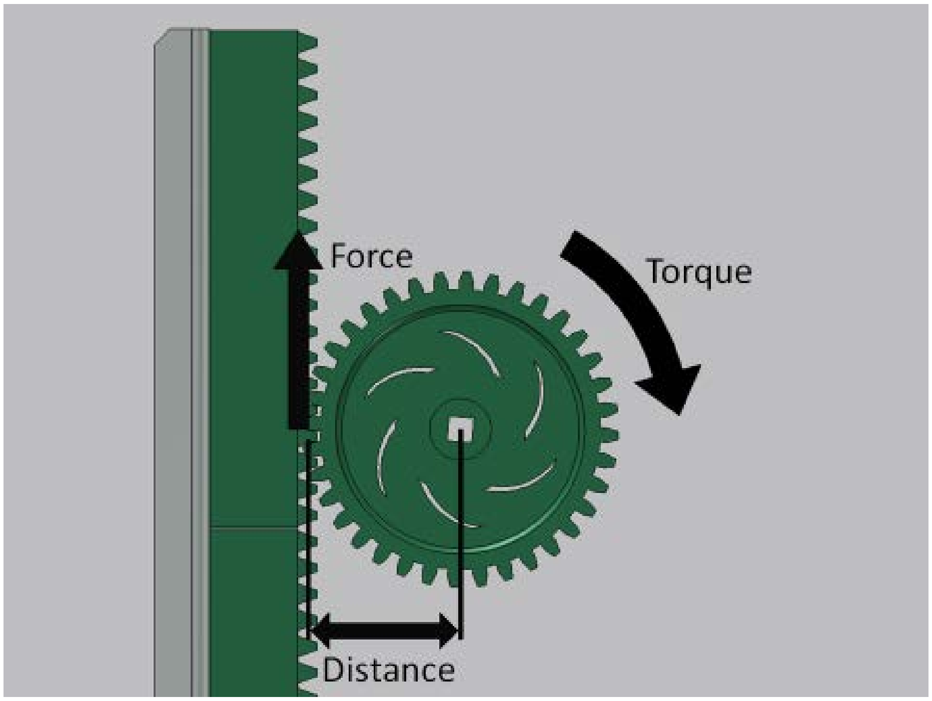
Another difference between an elevator and rotating joint is the way the object changes orientation. In a rotating joint, as the joint lifts the arm changes orientation. A linear elevator does not change orientation as it elevates.



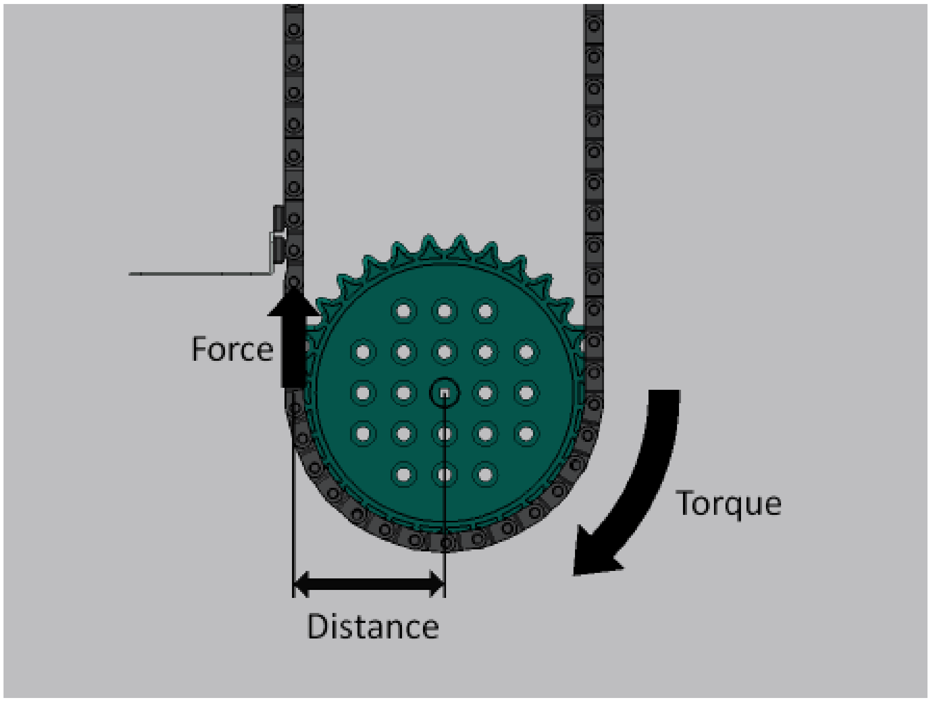
In the above example, as the rotating joint lifts the cup, it spills it! The elevator lifts the cup straight up without any spilling.

Elevator Actuation & Loading:

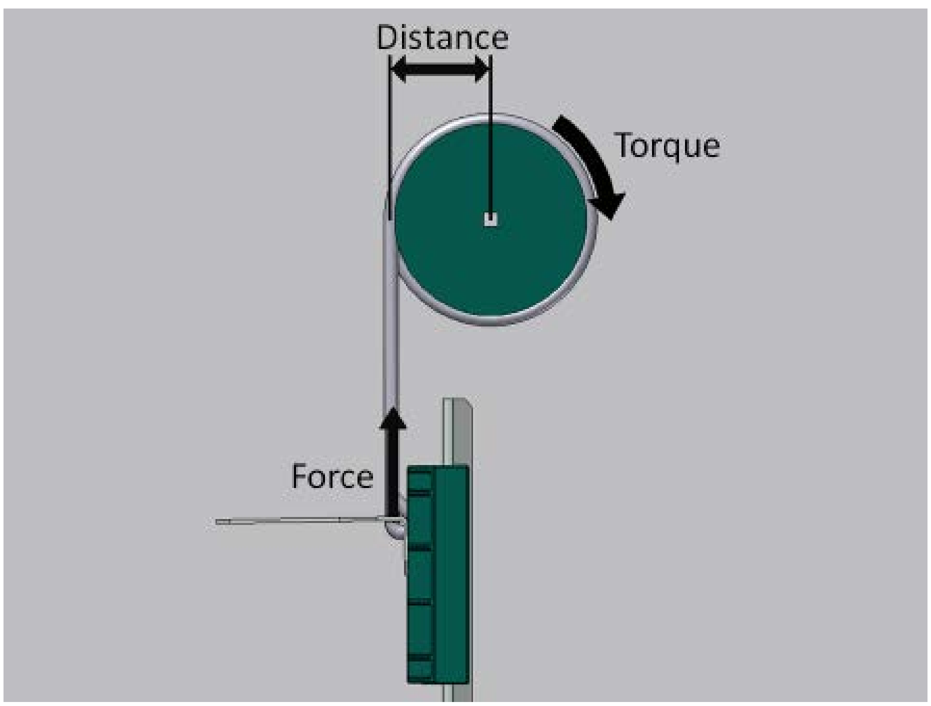
Actuation of a linear elevator is a little different than a rotating joint. Unlike a rotating joint, an elevator somehow must convert the rotary motion of a motor into the linear motion required to drive the elevator. There are several ways this is commonly accomplished.



In the above example a rack and pinion mechanism is used to drive the elevator. The pinion gear spins at some speed with an applied torque. This torque then applies a linear force at the gear's pitch circle, which drives the mechanism.



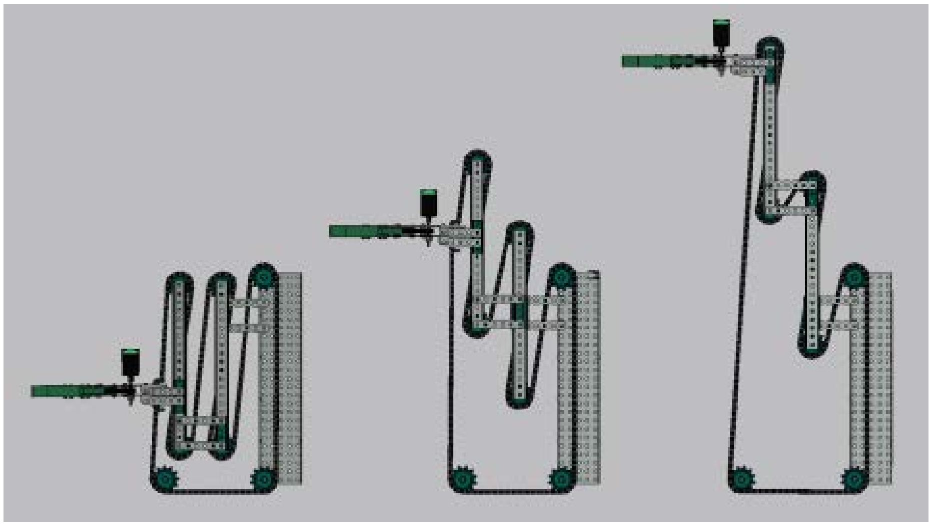
The above image shows how chain and sprockets can be used to drive a linear elevator. The torque on the driving sprocket spins the chain. The chain is attached to the linear elevator, and exerts a linear force to drive the mechanism.



In the above example, a winch is used to drive a linear elevator. A motor applies torque to the winch which provides a linear force along a rope, which then drives the mechanism.

Multiple Stage Elevators:

Single stage elevators are able to lift up only one extension of their length. That is, if the elevator is one meter tall, it will be able to lift a claw from the bottom to the top, a total of one meter. However, multiple stage mechanisms are also possible. By stacking multiple linear elevators together, one can create a mechanism which will reach up much higher than their own height.



10.3: Rotating Joints
up
10.5: Linkages

10.5: Linkages

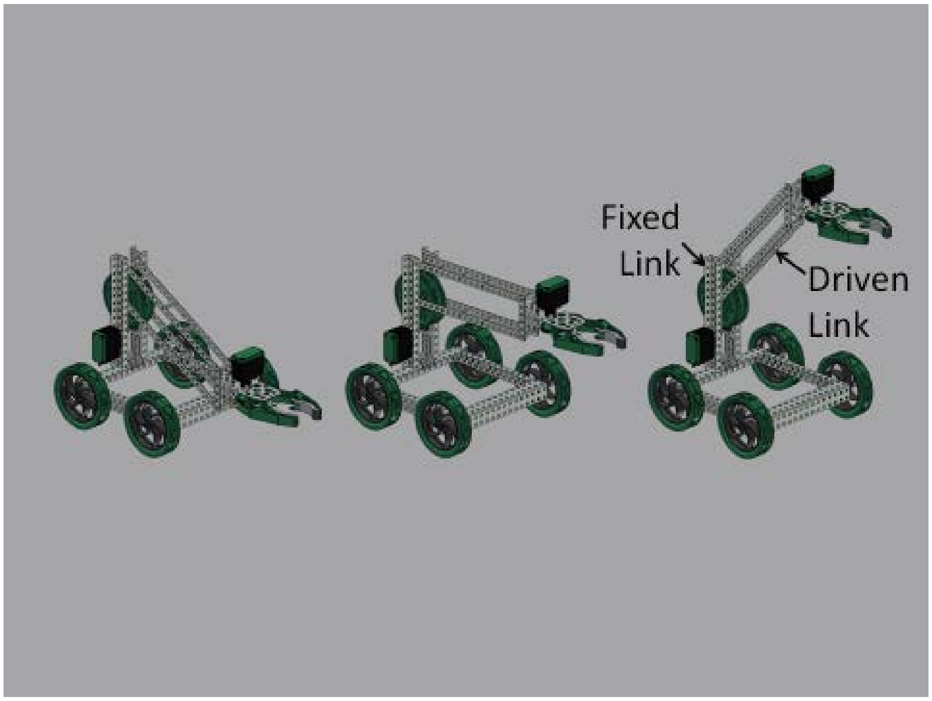
Linkages are designed to convert some input motion into a different output motion. A linkage typically consists of a series of rigid bodies called links, connected together by freely rotating joints. Typically, one link is fixed and cannot move, and one link is driven in some input motion. Linkages are a fundamental part of machine design because of their ability to create such a wide variety of output motions and their ability to alter the path, velocity, and acceleration of the input. Very precise and somewhat complicated motions can be designed using a simple linkage design. Linkage motions are extremely repeatable.

Linkages are found in many places throughout the world. Below is an example of a simple linkage found on a pair of Vice Grips.

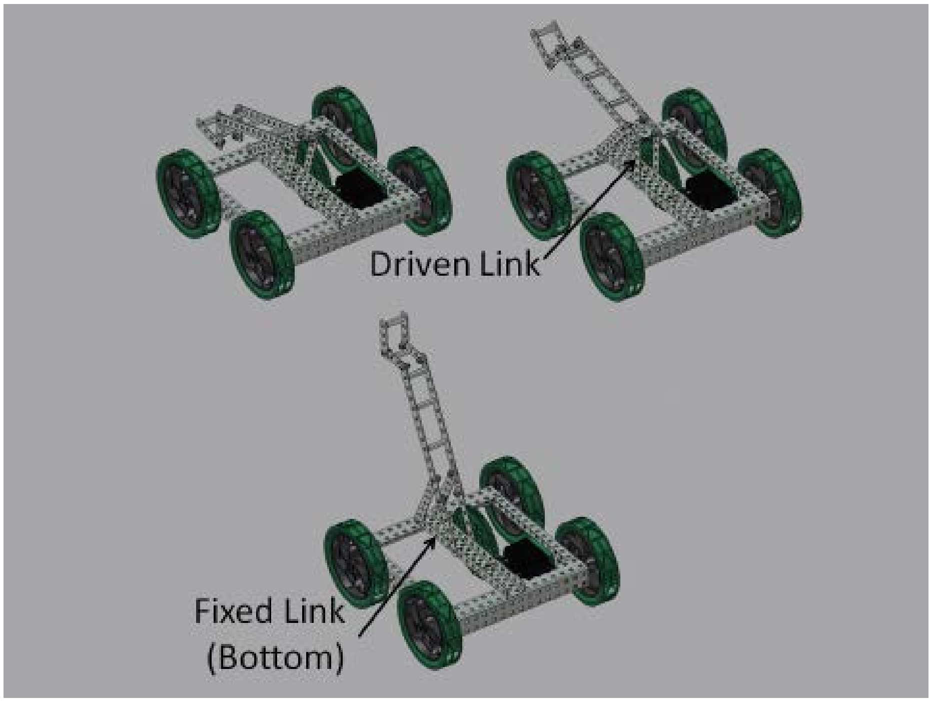
The first picture above shows the linkage in one configuration. The link at the bottom is the input link which is driven, and the link at the top is the fixed link. The second picture shows the linkage at the other end of its motion. This is a linkage with four links, each link has two joints. This is one of the most common types of linkage systems.

Four Bar Linkages:

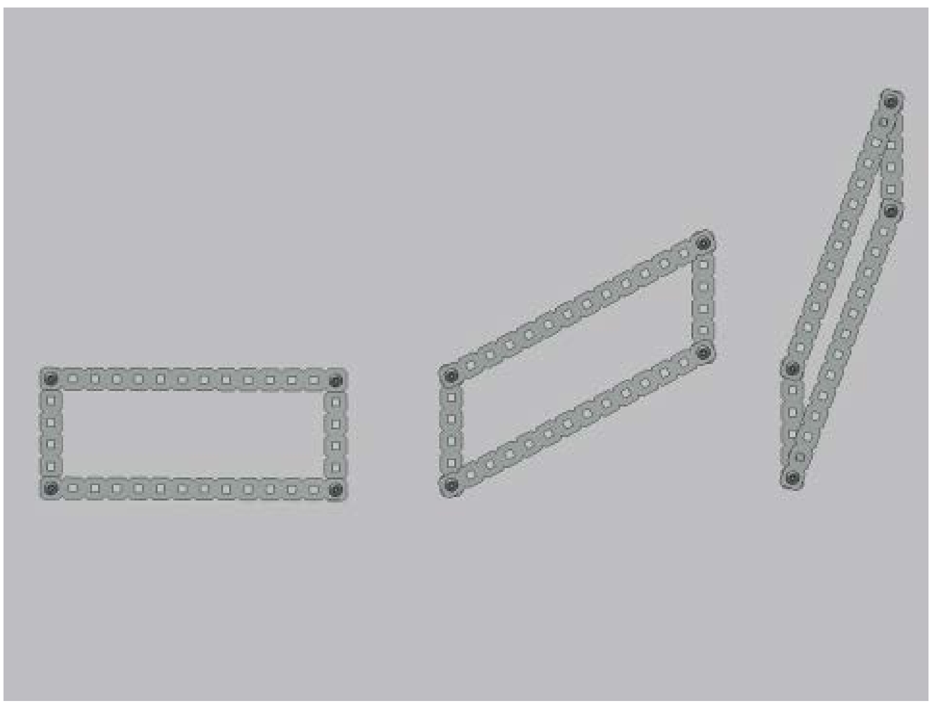
One of the simplest and most common linkage types is the four bar linkage. This is a linkage system which can provide a wide variety of motions depending on its configuration. By varying the length of each link, one can greatly change the output motion. The most basic type of 4-bar linkage is one where the links are equal length and parallel to each other, as seen below:

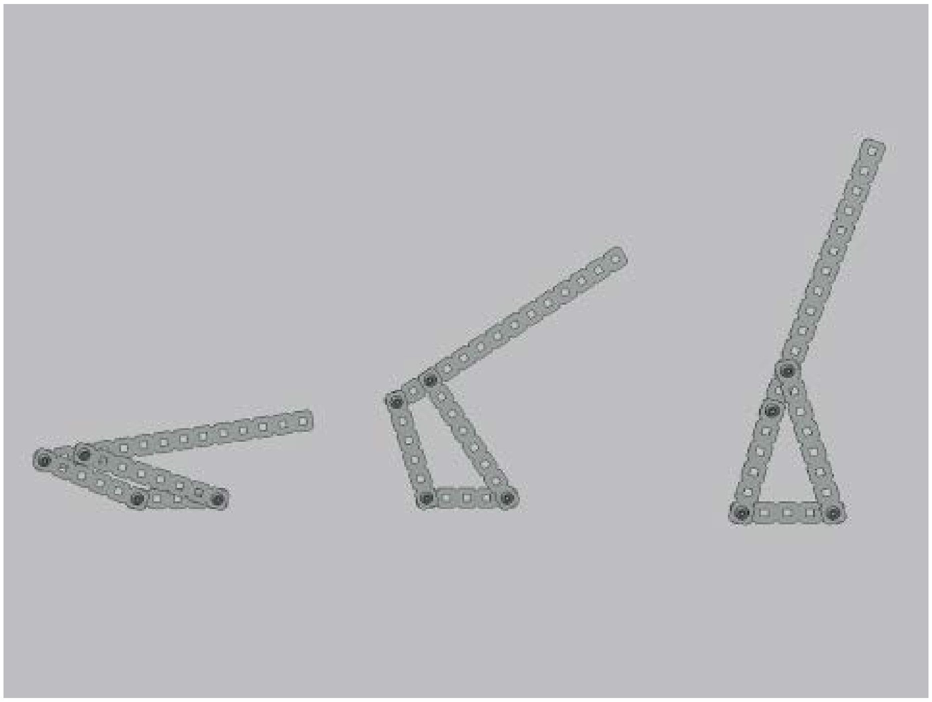


In this particular setup, as the linkage travels through its motion the output link will remain parallel to the fixed link. As such, the claw remains in a consistent configuration. This would be useful for an application where the object manipulator must remain in the same orientation relative to the ground. However, not all 4-bars take this configuration.



In this example of a four-bar linkage, the output motion is very different from the previous example! By tweaking the lengths of the different links, a designer can create many different designs. Using the VEX Robotics Design System it is easy to experiment with linkages to test their different motions.





The above linkage prototypes can be created with only a few VEX metal pieces, and a few screws.

10.4: Elevators

up

10.6: Design of Lifting Mechanisms

- Teacher Materials

10.6: Design of Lifting Mechanisms

This unit described a number of different lifting mechanisms, but which one will work best on a given robot?

There are a few key considerations, discussed below, which designers should take into account when designing a lifting mechanism.

Elevation Required:

The first and most important consideration a designer needs to take into account is the elevation required. At what height does the robot pick up the object? At what height does the robot need to lift the object? Are there multiple heights the object will need to be scored at? What mechanism will accomplish these changes in elevation?

Object Orientation:

Another important consideration for a designer is the change of orientation the object will go through, if any. What orientation(s) will the objects be in when picked up? What orientation(s) will the object be in when scored? Are these orientations different at different elevations? Often, the lifting mechanism can be used to accomplish these orientation changes. Some lifting mechanisms will be better suited to an application than others due to required orientation changes. For instance, if the object needs to stay in the same orientation from when it is picked up until it is scored, a rotating joint may not be as good a choice as a linear elevator.

Starting Configuration & Other Size Limitations:

VEX Robotics Competition robots must start each match within a limited size, as described in the Game Rules released each year. This starting size may restrict which lifting mechanisms can be used. For instance, a single jointed arm which starts within an 18” x 18” x 18” box cannot reach up four feet – though a multiple jointed arm might be able to do it. There are often other size limitations involved in the creation of a competition robot. What if there is a bar on the field that the robot must drive under? In this situation a designer may choose to design so his or her robot’s lifting mechanism can fold down under the height of this bar. Size limitations play a large role in the design of lifting mechanisms, especially when combined with elevation requirements as discussed above. It is easy to make a robot reach up four feet; it is more difficult to make a robot start in an 18” cube and then reach up four feet.

Some lifting mechanisms may take up more space than others. If the robot has a large hopper-style object manipulator, and it takes up almost all of the space in the robot’s starting configuration, the designer will need to find a lifting mechanism that will package into the space available. Some lifting mechanisms may “fit” better than others.

Complexity:

In some situations, multiple lifting mechanisms may all work. However, these mechanisms can vary greatly in complexity. It is usually a good idea to choose the simplest mechanism possible which accomplishes the design goals. Sometimes, it is even beneficial to make tradeoffs and reduce the design goals. For example, if a simple mechanism is “almost good enough” to achieve the design goals, and if fully accomplishing the goals would require a VERY complex mechanism, it is probably better to choose the simple mechanism. Simple mechanisms have fewer moving parts, are more robust, and are less likely to fail; in these ways, they are usually better.

Motors Required:

Competition robots are limited in the number of motors and other actuators that they are allowed to use. Designers should be cognizant of how many motors are required for each lifting mechanism option. A 2-jointed arm requires two motors, one for each joint, whereas a single jointed arm only requires one motor. In this case, the designer could still use two motors (as described in Units 7 & 8) and the mechanism would be able to handle its load twice as fast!

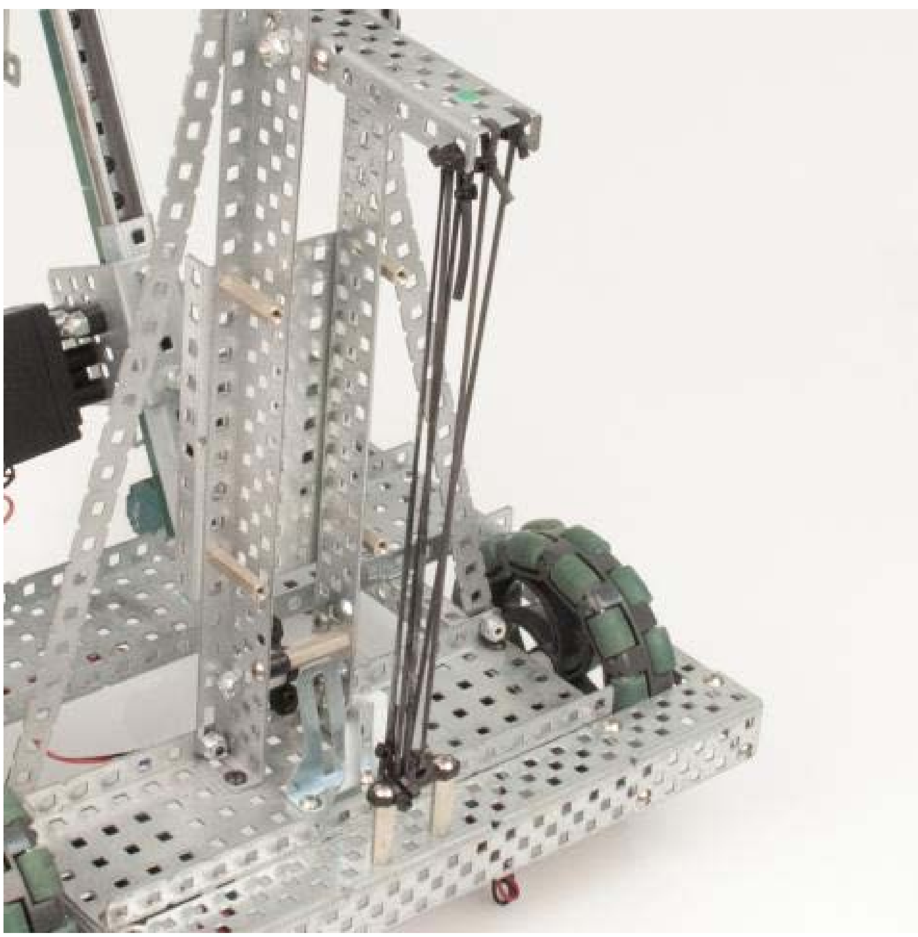
10.5: Linkages | up | 10.7: Passive Assistance

10.7: Passive Assistance

One “expert trick” used by many competition robotics teams is based on the concept of passive assistance. Passive assistance is when a designer adds something to the lifting mechanism to assist the actuator in lifting the load. Usually these take the form of counter-balances or springs. If the lifting mechanism is trying to lift a claw with a 1 Newton weight in it, and the designer adds in 1 N worth of spring force, the motor will only need to lift the weight of the claw! Using passive assistance can GREATLY reduce the load on the motors and allow for much faster mechanisms that lift much greater load. In certain situations, it may even be beneficial to use so much passive assistance that the motor must work to pull the lifting mechanism down.

Consider a robot that is lifting a ball which exerts a force on the end of an arm of 4 Newtons (for this example, one can assume the arm and claw weigh nothing). The designer could add 2 Newtons of passive assistance springs to the arm. This means when the arm is lifting up, the motor is exerting 2 Newtons and the spring is adding 2 Newtons to lift the 4 Newton ball. After dropping the ball into a goal, the motor would then need to fight against the springs to lower the arm, and would exert 2 Newtons of force.

The most common kinds of passive assistance used in the construction of VEX Robots are elastic surgical tubing and rubber bands. These can be stretched in such a way that they contribute to the upward motion of a lifting mechanism.



There are tradeoffs involved in using passive assistance, but clever designers can use it to reduce the load on their motors, and make their lifting mechanisms much quicker and more robust.

10.6: Design of Lifting Mechanisms
up
10.8: Design Activity

10.8: Design Activity

10.8: Design Activity

Lifting Mechanism Construction

Using the lessons learned in this unit, students should follow a design process similar to the one described in Unit 1 to determine a lifting mechanism for their competition robot. Students should brainstorm multiple solutions and record the entire process in their engineering notebooks. Students should choose one concept and design it in Autodesk Inventor and/or build it using VEX Components.

When designing your Lifting mechanism, ask yourself the following questions to get started:

- What do you want your lifting mechanism construction to achieve?
- Will a simple Elevator lift achieve the Design objective or would it be better to use a 4 Bar?
- Will you need to use a Rotating Joint?
- Do you need to use Linkages to transfer the motion?
- How many motors are you going to have to use?
- Can you use Passive Assistance?

To be able to complete this activity you should have a basic understanding of the Autodesk Inventor user interface, navigation, and know how to Create Parts and work with Assemblies. For review, please refer to Appendix 9, 'Basic Inventor Commands Overview' for further information on these.

10.7: Passive Assistance

up

10.9: Engineering Notebook

10.9: Engineering Notebook

Answer the following questions in your Engineering Notebook:

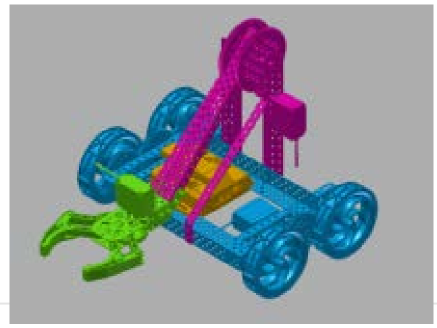
1. Explain how the degrees of freedom will allow you to design a robot that is able to transfer motion as it manipulates game objects.
2. Explain how a linkage system allows a robot to score on a high goal in a game situation.
3. Explain how passive assistance can provide your robot with a mechanical advantage.

10.8: Design Activity

up

Unit 11: Systems Integration

Unit 11: Systems Integration



In this unit students will learn techniques for successfully integrating a number of disparate subsystems into one cohesive whole.

Students will take the lessons learned earlier in the semester and their existing designs to create their overall robot.

10.9: Engineering Notebook

up

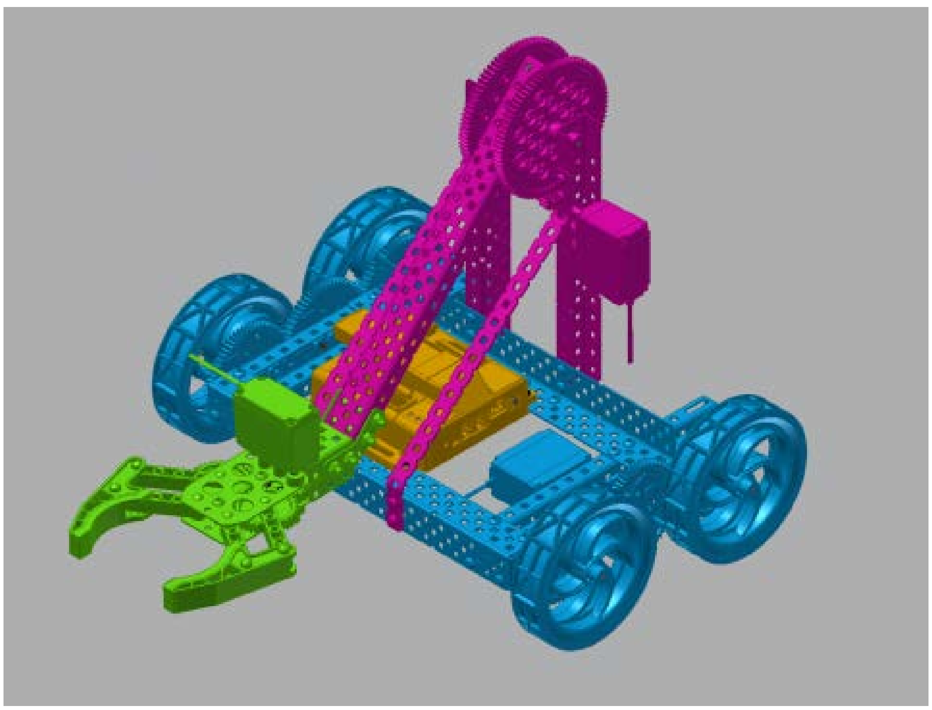
11.1: Introduction

- Curriculum Lesson Content
  - Unit 1: Introduction to Engineering
  - Unit 2: Introduction to Robotics
  - Unit 3: Introduction to VEXnet
  - Unit 4: Introduction to Autodesk Inventor
  - Unit 5: THE GAME!
  - Unit 6: Object Manipulation
  - Unit 7: Speed, Power, Torque & DC Motors
  - Unit 8: Mechanical Power Transmission
  - Unit 9: Drivetrain Design
  - Unit 10: Lifting Mechanisms
  - Unit 11: Systems Integration
    - 11.1: Introduction
    - 11.2: Systems Integration
    - 11.3: Design Activity - Finish Competition Robot
    - 11.4: Engineering Notebook
  - Unit 12: Testing and the Iteration Process
  - Unit 13: Design your Own Part [optional]
- Appendices

- Teacher Materials

11.2: Systems Integration

One of the most important concepts in design is that of systems integration. It was discussed earlier how a design is often divided into subsystems. Systems integration refers to the way that these individual subsystems are combined during the design process into one cohesive product. This is not something that happens at the end of the design process after each subsystem has been designed, but rather something that is integral to the process. Ideally each subsystem works well and supports the other subsystems of the robot in a way that the whole becomes greater than the sum of the parts. During the design process, it is important to think about the ways each subsystem will affect and interface with each of the others.



In competitive robotics the robots are typically divided down into subsystems that perform different functions. Some of these systems will stretch throughout the robot while others will consist of only a single mechanism. An example list of subsystems is shown below:

- Power
- Control
- Sensors
- Pneumatics
- Drivetrain
- Lifting Mechanism
- Object Manipulator

Each of these subsystems could be designed independently, but each is dependent on all the others. In order for the overall robot to function effectively, each of these systems must work together. In order to design any one of these systems, one must have knowledge of the others. Each of these subsystems will have its own individual design process as part of the overall design process; any requirements on the way the subsystems interact as part of the systems integration would be treated as Specifications (design constraints) in Step 3 of the design process described in Unit 1.

For example, if the robot has a maximum size requirement, each subsystem may need to nest together such that the overall system "fits in the box." In this scenario each subsystem may be given a specified amount of "robot real estate" that it must stay within and specified ways they would interface with the others. For example, the claw must start within an imaginary 3" x 3" x 5" tall box located at the very front of the robot. It is 2" off the ground, and it attaches to the robot arm at the lower, rear center of this imaginary box using a specified hole pattern.

Tips for Integration:

Whatever types of subsystems are being integrated, it is important to keep several things in mind. Remembering these basic principles will help create a better overall system.

1. Look for changes that can be made to the individual subsystems that will improve the performance of the overall system. (For instance, a simple "funnel" design in the front of the chassis of the drivetrain subsystem could significantly improve the performance of an accumulator collection subsystem.)
2. Try to reduce components used wherever possible. If possible, share components between subsystems.
3. Try to utilize components that can provide multiple functions in the overall robot system.
4. Design the system so that it can be easily assembled, disassembled, and maintained.
5. Design the system so that it requires fewer motors. Try to share motors if possible.
6. Improve the speed of the overall system. Speed is often the measure of effectiveness.

By remembering these tips it is possible to create a well functioning overall system. System Integration is best performed throughout the entire robot design process.

11.1: Introduction | up | 11.3: Design Activity - Finish Competition Robot

11.3: Design Activity - Finish Competition Robot

Finish Competition Robot

Student design teams should redesign or enhance their Object Manipulator from Unit 6, Drivetrain from Unit 9, and Lifting Mechanism from Unit 10. Students will combine these systems using the principles of Systems Integration described above to complete a full competition robot to play the game described in Unit 5.

Design teams should fully model their robot in Autodesk Inventor, and then build it out of VEX Robotics Design System components in real life. Once built, students should use their knowledge of the VEX System learned in Unit 3 to wire & configure the robot to make it fully functional and ready for competition play.

When designing your Competition Robot, ask yourself the following questions to get started:

- What Drivetrain do you want your robot to have?
- How is your robot going to manipulate the game objects?
- Will you need a lifting mechanism construction to achieve the competition Objective?
- How many motors are you going to have to use?
- Can you use Passive Assistance?
- Does your robot need speed or strength to be competitive?

To be able to complete this activity you should have a basic understanding of the Autodesk Inventor user interface, navigation, and know how to work with Assemblies. For review, please refer to Appendix 9, 'Basic Inventor Commands Overview' for further information on these.

11.2: Systems Integration

up

11.4: Engineering Notebook

12.2: Testing and Iteration

After the first version of the robot is complete it is time to test it. Some questions for the students to consider while testing:

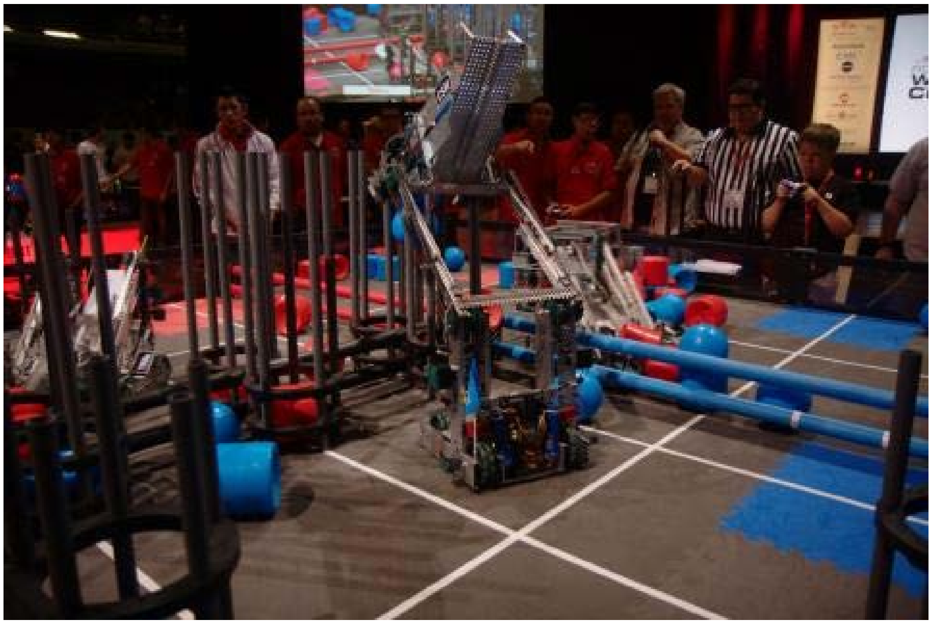
- Does the robot complete tasks in the desired manner?
- Is it fast enough?
- Is it robust enough, or are parts of it breaking or coming apart during use?
- Does the robot achieve the goals that have been set forth by the design team?
- Can it score enough points to consistently win matches?
- Is it easy to control?

After answering these questions, students should identify what areas they need to improve upon. At this point they should prioritize the improvements and begin working on the most pressing changes. Beware, it may seem like a good idea to change multiple things at once, but this could introduce a whole new set of problems.

It is best to approach improvements in a systematic and logical way; students should take detailed notes in their engineering notebooks during this part of the process.

1. What is being tested?
2. What results were observed?
3. What could be done to improve the results?
4. Are the results acceptable or is it worthwhile to make improvements?
5. After completing this process, students should test the robot again asking the same questions.
6. Repeat: Design is an iterative process!

Once the students are satisfied with their completed robot, it's time to get ready to compete.



12.1: Introduction

up

12.3: Competition

12.4: Analysis and Reflection

Once the competition is over, students should provide analysis and reflection on their experiences by completing the following essay type questions in their engineering notebook.

- What was your favorite part of the competition robotics experience?
- Did the matches play out like you expected?
- What would you improve about your robot design? Would you make any major changes or only minor changes?
- What would you improve about the design process if you had to start over? Were there things you wish you had spent more time on?
- What was the most important life skill you learned during this process? Why was it important to your experience?

12.3: Competition

up

12.5: Engineering Notebook