SPARTAN SUPERWAY

A Solar-Powered Automated Public Transportation System

San Jose State University

Mechanical Engineering Department

ME195A/B Final Report

June 2, 2017

Advisors:

Burford Furman

Eric Hagstrom

Ron Swenson

Authors:

2016-2017 Spartan Superway Team

Brean Aquino

Ernest Theo Cabreza

Randy Castillo

Shengson Cho

Neil Dey

Patrick Ding

Luis Escamilla

Jezreel Gajardo

Kathlyn Garces

Christopher Hansen

Matthew Hsiung

Craig Hudacko

Michael Kemp

Franklin Kha

Zhiwei Li

Kevin Maligaya

Qihuan Miao

Claude Michel

Steven Mou

Kha Pham

Moralma Rodriguez

Winter Saeedi

Andrew Snytsheuvel

Jorge Soto

Derick Wong

Kevin Yoshihara

Abstract

Spartan Superway is a multi-year project that fulfills Senior Project requirements for Mechanical Engineering majors at San Jose State University College of Engineering. Run by faculty from SJSU as well as partners in local industry, it provides a basis for students to learn about sustainable transportation in a real-world application.

Continuing work from previous semesters, students this year focused on three major models. The 1/12th scale model was updated with a new track and podcar system, complete with a solar charging and integration with a mobile application. The half-scale model received an improved propulsion and control system, as well as new suspension and wayside power. Finally, the full-scale model was addressed by mocking up small models of beam and railing design.

Over the two semesters, the team worked to improve on previous designs while creating new solutions for problems. Students interacted with local vendors and retailers to achieve their goals, gaining valuable soft skills to complement their technical education. Additionally, deliverables were set for each subteam, providing deadlines for certain parts of the project to be complete. Most of these dates coincided with our two major events throughout the year, Paseo Prototyping Festival and Maker Faire. These two events allowed the subteams to display their work to others, engaging the public and testing their project designs at the same time.

After completing each part of the project, students have moved Spartan Superway forward in both technology and ease of use. While files and documents are available from each subteam in the Superway Archive, this report serves as a record of the work done over the 2016-2017 academic year.

Acknowledgements

Spartan Superway would like to thank the following sponsors and partners for their contributions and cooperation during the 2016-2017 academic year:

ASSOCIATED STUDENTS
SAN JOSÉ STATE UNIVERSITY

EQUIPMENT
TOOL SHED
INC. RENTALS
Family Owned Since 1945

BARRY SWENSON
BUILDER
TRADITION • INNOVATION • INTEGRITY

BEAMWAYS

Bryan Burlingame

SJSU | CHARLES W. DAVIDSON
COLLEGE OF ENGINEERING

Table of Contents

List of Figures

List of Tables

Executive Summaries

1/12th Scale

Bogie Improvement

The 1/12th scale mechanical bogie allows Spartan Superway to be represented in a smaller scale and give people a better idea on how the full-scale model works. The 1/12th scale model allows the public to visualize fullest potential of Spartan Superway and allow other subteams predict possible issues that other scale models can encounter. The main issue that we addressed last semester was fixing the design of the switching mechanism, different design solutions were suggested last semester and during this semester we produced and tested the different design to find the best solution. This semester the team no longer uses the piano wires to link to the servo arms, making the switching mechanism more reliable. The final design of the switching mechanism can be easily 3D printed and it allows the bogies to run more consistently. The design of the cabin allowed the necessary sensors to be placed within the cabin and continue to have the bogie compatible with the track. Also, the bogie parts were reduced and many separate parts were combined into just one piece.

The team last semester decided to 3D print all of the parts to reduce the weight of the bogie, cost and production time. The bogie parts and cabin parts were modified several times to have the bogie best fit the track. It was easy to reproduce the iterations because all that was needed was access to a 3D printer. Different designs were made for the switching mechanism until one was found that made the bogie run consistent.

The team used many different printers and both ABS and PLA until the best parts were produced. The following 3D printers were used: Lulzbot Taz 4 in the Spartan Superway Design Center, The MP Select Mini from the track improvement team's member Jezreel Gajardo, and the Fortus 250 mc from San Jose State's Dr. Ken Youssefi. The best results came from Professor Youssefi and those parts were made with ABS. The links for the switching mechanism were printed with the MP Select Mini Printer because they were too small for the Fortus printer to reproduce. The rapid production of the 3D printed bogie parts and cabin allowed us to rapidly test and make the necessary changes to have the most compatible bogie with the track.

Position Sensing

Spartan Superway, a solar powered automated transportation network, is a sustainable transportation solution to the problems of climate change, energy shortage and traffic congestion. The purpose of the 1/12 scale model of the Spartan Superway project is to create small scale and show a big picture of the Spartan Superway and also to lower the cost of designing the components. The small scale could test the ideas and designs in a much lower cost before implementing them to the full-scale model. Previously, the Spartan Superway small scale model did not have a completed position sensing system. Previous teams implemented the barcode system and the Reed sensor to define the position of switching tracks. However, there is never a position system that is able to provide real time position information of each bogies on the track. The small-scale model is developing a mobile application to interact with users. An accurate real time positon sensing system become extremely important for monitoring the condition podcars and controlling the podcars. The position sensing subteam is a team focus on developing an accurate, reliable and cost-effective position sensing system in order to achieve and develop a “smart” and autonomous multiple-podcar system.

The idea of the new position sensing system is to attach a position sensor on the bogies and there are corrugated ferrous strips attached around the whole track. The position sensor is programed to count the number of peaks the bogie passed. With the number of the peaks, the distance that the bogie has passed can be easily calculated. There are checkpoints on the track for position reference. The position sensor is made of a piece of magnet, a Hall Effect sensor and a bracket which holes the magnet and sensor in a fix separation distance.

When a piece of magnet, a Hall Effect sensor are lined up and separate with a fix distance, the Hall Effect sensor is able to detect the magnetic field. When the valleys of the corrugated ferrous strips line up with sensor, the magnetic field will not be affected. When the peaks of the corrugated ferrous strips line up with sensor, the magnetic field of the magnet will be affected by the ferrous strip, and the Hall Effect sensor will experience a stronger magnetic field. Therefore, the combination of the corrugated ferrous strips, a unipolar Hall Effect sensor and a piece of Neodymium magnet will be able to distinguish the peaks and valleys of the corrugated ferrous strips in the designed configuration. The position sensing team should design a position sensing system based on the working theory.

Solar Improvement

Previous year’s solar racks were damaged. The wiring for charging component storage solution was a fire hazard and potential obstruction to pod cars traveling on the track. The joint joining the two halves of the solar rack failed due to normal stress. Pedestals were required to elevate the charging boxes to reach the solar panels. Wiring connecting the solar panels located above the track to the charging boxes are intruding the path of pod cars. Fabric-lined wooden boxes stored charging components and hid the charging operation to the audience. Opposite polarity stripped wiring connections are placed near each other which can potentially cause shorting.

The solar rack was redesigned using similar aspects of previous year’s rack. Similarly, the rack would help form a curvature on the solar panel, span the entire length of the solar panel, and comprise of two halves. Aluminum was chosen as a material to be used due to appearance, availability, lightweight, and strength. A curvature was formed with aluminum bar stock. These curved aluminum bars span the length of the rack like ribs across a torso. The solar panel are placed above the ribs is clamped down to confirm to the curvature.

Wiring between charging components were extended or shortened to create a cleaner and organized appearance. Spade connectors were crimped on to each end of the wirings for a secure connection between components. Components were mounted on a polycarbonate board via screws and nuts. The box with a hinged cover was placed over the board. The box is mounted on drawer sliders attached to the underside of the racking system.

The solar team designed, modeled, and fabricated a practical solar racking system. It is self-contained, forms a curvature of along thin film solar panels, can be assembled and disassembled in minutes, and does not interfere with the operations of pod cars on the tracks below.

Track Fabrication and Process Development

The main purpose of the Track Fabrication and Process Development subteam was to provide adequate fabrication capabilities of sufficient quality to permit in-house fabrication where possible. The first way that this purpose was manifested was to produce sections of track as required. A significant issue with the track produced by the 2015-2016 small-scale team was that corner sections did not have proportionally similar radii, resulting in bogies being unable to traverse corners without risk of falling off or binding.

In order to remedy the issue, the focus of the 2016-2017 Track Fabrication and Process Development subteam was to create a bender to reliably and repeatedly create corner sections of track for uniformity. Background research was performed early during the Fall 2016 semester to investigate the viability and capabilities of commercial machine available to the common consumer, namely Hossfeld and DiAcro benders. Neither a Hossfeld nor DiAcro bender appeared to be able to deliver the specific functionality required, but certain elements could be highlighted and incorporated into the design of the bender that the team would create.

Several problems needed to be addressed during bender creation; die dimensions, the bender arm, and the roller. Die dimensions were calculated with equations (as shown in Appendix A4-1) that accounted for material strength springback. The bender arm was created from a single aluminum bar with sufficient length to allow material to be drawn around the die when sufficient force was applied. The final version of the roller utilized aluminum round stock as the pin and wheel, providing sufficient strength to resist deformation applied to the wheel by the bending motion.

The second way that the team met fabrication requirements was through aid given to the Solar subteam in the form of fabrication and assembly help. Ribs and tabs to hold a solar panel to the rack had sharp bends made with a form meant to be utilized in a vise, with larger bends coming from the larger material bender. Further detail concerning the solar panel rack can be found later in this report and in the Solar subteam report.

There are several recommendations that can be made for any who would take part in following years' Track Fabrication and Process Development subteams. First, the team should work with the Bogie Improvement subteam and the Track Improvement subteam to work towards a design that allows larger tolerances to be accepted, especially when the bogie needs to traverse corners. Second is that frequent and focused inter-subteam communication be prioritized, as understanding the status of each subteam and how it would relate to your own work is necessary. Third is that integration of parts can often prove troublesome, so compatibility should be tested for early and often if possible; understanding the status of other groups can greatly aid in this process. Fourth is that precision is key, and shoddy work will very quickly create errors that can result in overall failure, so care should be taken to provide adequate time to allow for high-quality work to be performed. Fifth and finally is that testing requires forethought, not only to account for what materials will be handled but to allow for processes to be learned if necessary, as correct procedure is no less important than correct production.

Track Improvement

Spartan Superway is an automated transit network that is on suspended rails to be out of the way of pedestrian and automobile traffic. The idea is to provide a method in which passengers could travel from point A to point B without having to make stops in between locations and without having to share a commute with strangers. The 1/12th scale team is developing a model to demonstrate how the larger network could be designed and automated. The track improvement subteam is to provide a usable track for the bogies to move around without complications. The track has been built and re-designed over several years, however, improvements were still needed. In the last year, the track improvement team was able to complete a fully functional track.

Objectives were to: Design and manufacture the track to provide proper alignment and clearance for the bogie; design a track assembly that would be easy to break down, transport, and reassemble; communicate with other subteams about sensor placement and any necessary changes; remove any need for tapping aluminum stock.

The first objective is being addressed by building and designing the track in such a way that alignment issues are significantly minimized. A few brackets have already been redesigned and manufactured. A U-shape bracket has been designed and is planned to be 3D-printed. Its purpose is to make sure the two lower tracks remain aligned and at proper distance of 2.00 inches. The existing L-shape brackets will be re-used but a few will have to be modified to allow them to mount to the vertical supports.

The brackets for the corners have been mounted in such a way that the guideway track should not bind with the bogie. This was done using wooden shims during the manufacturing process to keep the guideway within .9375 inches from the bottom rails. The corners were built to become a single piece, as well as the stations. A second iteration of “U”-brackets were made so that pieces of the track could be connected properly. There is also a separate straight component made to connect the then assembled loops.

The “U”-brackets worked on the first design, but showed some deformation due to the internal load created by the aluminum tracks. The distance between the two mounting surfaces also came out the 2.49 inches due to the thickness of the 3D filament. This made it difficult for the bogie to travel through. A second iteration was manufactured that was longer, allowing for more material to take the same internal stress. It was also manufactured to have a width of 2.55 inches so that the bogie could easily clear. The old L-shape brackets have been re-used to weld onto the lower track, rather than bolted. This was done with a TIG welder using 4043 filler aluminum to 6061 aluminum. The guideway was too small to weld. Long 2.5 inch bolts were used as well as a nut to clamp the guideway to the top of the old L-bracket.

The track as it exists now is much easier to work with. The entire process of disassembly can be completed with two, ½ inch wrenches, a Phillips screwdriver, and a 3mm hex key. Once broken down, the track only exists as three unique parts that are easy to keep track of. In concept, these parts would be interchangeable, however, due to manufacturing shortcuts, each component must go back in the same place. The vertical supports are easy to erect into the concrete bases and have slots created to make different parts of the track height adjustable.

All of the initial revisions were manufactured and implement several weeks before the final deadline. This allowed numerous tests to be performed to fine tune the design of the track, as well as other complementary projects. Additional revisions were made, such as making the guideway on the corners thicker to reduce the tilt of the bogie. Brackets were altered to provide a better-quality track. The track works great, but could use some fine tuning. Every time the track is erected, small alterations need to be made. The process of fine tuning takes numerous attempts of trial and error. A more robust design could reduce the amount fine tuning required. Alterations could also be made to allow construction to be completed faster.

Vehicle Controls

Spartan Superway aims to create an automated transit network (ATN) powered by renewable energy. The goals are to reduce traffic congestion, reliance on foreign oil, and to provide a point-to-point transportation system that is highly efficient. The 1/12th scale model allows for an end user to interact with the entire system at a smaller scale to understand the scope of the project. The Vehicle Controls team in particular is responsible for the entire automation of the model, ranging from electronic components to coding.

During the first semester of Spartan Superway, the basis of a control system was developed using two microcontrollers (MCUs) interfaced via I^{2}C communication. This methodology was intended to prevent interference between sensors, but ended up being problematic due to the complex interaction between the two MCUs. A new system was derived during Winter Break, consisting of a single Arduino Mega. This version required no device-to-device communication, and was easily interfaced with the various sensors and controls already chosen for the podcar.

At the heart of the system is an Arduino Mega, which is an MCU that drives outputs based on logic from inputs. The inputs consist of: two switches to control the podcar, two hall effect sensors to detect magnets on the track, sonar to detect a podcar in front, encoders to count track displacement, and an RFID reader to detect unique cards mounted at each station or checkpoint. There are three outputs on the system: drive motors to provide propulsion, a servo to actuate the switching arms, and an RGB LED to show status of the podcar.

Finally, there is two-way communication via an XBee module, which allows the Arduino to wirelessly communicate with a central computer also equipped with an XBee module. This system creates a method for sending the pods messages via the Software Team's mobile application.

After deciding on the MCU and final system architecture, code was written to create functions for each of the system's outputs and inputs. While the code to interface with the individual electronic components is not increasingly complex, the logic and algorithms for processing the data and making decisions based on input requires considerable effort. As such, the code for routing the podcars from station to station was a joint effort between the Vehicle Controls team and the Software Team.

Before the Spring semester, the necessary electronics for a total of six podcars was ordered from various online retailers as well as Halted Supply Co. Turnaround time for the components was relatively quick, which made construction of a preliminary bench-top model for the Software Team possible. This system allowed them to test and experiment with their code very early in the project.

The components chosen for the project required a large amount of wiring, so the team decided to create an expansion board for the Arduino Mega based on an off-the-shelf board from AdaFruit. This board mounts on top of the Mega, with a breadboard-type construction in the middle of the input/output pins. Due to the high number of components, a modular system was desired. This was accomplished by using header pins as plugs on the expansion board, which connected to harnesses for each component.

Upon testing the first expansion board, several changes were determined to improve upon the system. A removable XBee module allowed the Arduino to be programmed without taking off the entire expansion board, and improved header pins made the connections more robust. Additional pins were also added to directly plug in the servo to the expansion board.

The completed systems were then installed into the bogie and cabins created by the other 1/12th scale subteams. The sonar, Arduino, switches, LED, and RFID reader were mounted neatly in the cabin, with the hall effect sensor and motors mounted on the bogie. To minimize wasted effort and time, the team elected to use existing barcode scanner mounting shoes for the hall effect sensors.

In all, six complete podcars were built, each with their own identical expansion boards, electronic components, and wiring harnesses. The Vehicle Controls team displayed these models at both Paseo Prototyping Festival and Maker Faire 2017, showcasing the future of automated transit network technology in the Bay Area.

Half-Scale

Active Suspension

When cars make a sharp turn or sudden stop, if there is no force to oppose the momentum within the system, then the car could flip over or give a harsh ride experience. The same thing applies to a pod car of a suspended transportation system. As a suspended transportation system is the first of its kind then there is a technological need where mechanical engineers can address by designing technology that will give the pod car the ability to react to outside forces, self-level itself and nullify ride vibrations.

The challenge with the Spartan Superway Active Suspension System is to design the needed technology to deliver a ride quality that people would enjoy experiencing while using the Superway. The plan is to deliver on these requirements by designing three systems. The first system is a tilt mechanism. This is needed in order to keep the pod car level to the ground. By keeping the pod car level to the ground the rider never feels that they are in an uncomfortable riding position. The second is a self-leveling system. This is needed to keep the floor of the pod car level with the platform that riders use to disembark. By keeping the floor level, it provides a seamless ridership experience rather than a blatant tripping hazard. The third system is a mechanical suspension system. This is needed to nullify any harsh vibrations of the Superway rail system. As with any transportation system the more comfortable and enjoyable the experience is the more people will be inclined to use it.

The first system that was designed and revised to work with the following systems was the mechanical suspension. Inspiration was taken from F1 racing teams and their shock placement and double wishbone designs. It was CAD modeled with those components in mind. It was later seen that the double wishbone design was not applicable but the shock design was still relevant. The shock design was then chosen and it was decided that the bottom of the entire system would house the mechanical suspension. This was done so that any vibration from tilt or scissor lift mechanism would be absorbed by the shocks. The links for the mechanical suspension are appropriately sized for the expected load. The shocks and springs will be chosen once the other systems are complete.

The second system that was designed was the tilt mechanism. It started with concept drawings to first spatially realize the system and to brainstorm ideas. Then mechanisms were considered to be used to tilt the pod car. The choices were narrowed down to a four-bar mechanism. A force analysis was done on the four-bar mechanism to determine the link widths and pin sizes that a half-scale prototype would require. Then based on the force analysis the torque was calculated and a motor sized for the application was chosen.

The third system that followed the tilt mechanism was the self-leveling mechanism. Concept drawings and possible designs of this system were done in order to explore the integration of this system with the tilt mechanism. The design is a scissor lift mechanism. The advantages of using this design is that it can expand and retract a large distance while moving in straight line motion. However, the design of the self-leveling mechanism uses a linear actuator in the middle of the system. This makes it difficult to occupy that space with another mechanism.

A bill of materials has been completed and all mechanisms can be built. In retrospect, it would have been easier to go through this process with a better understanding of how it will be used and displayed.

Bogie, Steering, and Failsafe

A cabin will carry a maximum of 5 persons and their respective belongings. The cabin will need a foundation to carry the weight and accurately steer along the railway which contains different angles at specific locations. A half scale design for the system is necessary to predict and implement in the full-scale system. Previous design ideas exist. Many problems with the previous design originated. One problem was the inability to accurately travel along the 17-degree angle. The bogie came in contact with the roof of the rail of the system which prevented the system from moving. Another problem was the steering which did not work as precisely as expected. The design was complicated, frail, and was unable to withstand the forces applied by the weight of the cabin. Also, the wheels contain relatively large rotational friction as they are mounted on a threaded shaft. The friction on the wheels decrease the speed of the bogie thus causing a problem when going uphill. The connecting rod was also a problem as it constrained the system to only 1 degree of freedom.

In order to solve the multitude of problems, some design changes were implemented. The single bottom wheel was changed to two wheels. This change maintains the bogie perpendicular to the rails at all times and will also prevent the bogie from coming in contact with the roof. The connecting rod is now positioned at the bottom instead of the middle to the bottom of the bogie. The connections of the rod were also changed to a bracket which will allow for 2 degrees of freedom. Some widths were increased to prevent any forces from distorting the bogie and ultimately impacting the functionality of the design. The slot that was originally for the upper steering arm was removed as it was not necessary and the metal now there was used to mount the new gearbox for steering. A major change was in the steering system. All long threaded connecting rods were removed to prevent the steering from failing and decreasing stress concentrations. The new steering system consists of gears and pulleys which will provide enough torque to opposing all dynamic forces. A single motor will be used to control the entire steering system.

All changes to the design were tested with a wood prototype and adjusted accordingly. With the usage of the prototype, adjustments were made before the construction of the final bogie. The mechanics of the project can be verified thus decreasing the possibility of mistakes during the construction of the final product. Also, dynamics and resilience of our design were verified by utilizing the features provided by solid works. Some FEA, Finite Element Analysis, verified the endurance of components. Appendix B2-2 demonstrates FEA validation for failsafe. After testing and ensuring that every aspect of the system worked as expected, all parts were ordered and fabrication began. The Water jet machine was used to complete major parts. After, parts were manually and CNC milled. Next, shafts were lathe. Finally, specific holes were tapped and assembly started. As assembly took place, parts were secure and soldered. The bogie was also powder coated to prevent any corrosion. Once testing begin, some adjustments took place. For example, guiding wheels were replaced by bigger diameter wheels to improve reliability. Also, reducing shafts lengths provided a better fit. All processes enable the completion of the project.

Braking and Propulsion

The development of braking and propulsion are critically important components for the bogie system to perform the function of moving along the guideway and being able to stop at stations. The propulsion design will make use of 24V motors to drive the main wheels supporting the entire bogie by translating the motor rotation using two miter gears attached to the motor shaft and one attached to the wheel shaft. The wheels will make use of the normal force provided by the weight of the bogie on the wheels to move along the rail. Movement is designed for the bogie to move along an incline of 17°. Furthermore, the motor includes a gearbox with gear ratio of 160:1 to provide maximum torque. Braking will consist of using spring and lever arm to rub chocks against the driving wheels. Additionally, the stepper motor will pull the lever arms back enough that the chocks will no longer contact the driving wheels. Each half bogie has their own propulsion and braking system.

Mechatronics and Integration Systems

In the ATN (Automated Transportation Network), public transportation is automated to avoid excessive need of human input. The mechanical bogie needs certain input to move to a certain degree of location while maintaining the safety of the passengers. An Arduino is configured to carry out specific tasks given by the user with minimal amount of data, which in this case the user inputs the destination and the system automates the movement of the bogie. The goal of the subteam was met and the system was able to move the bogie to its destination safely and reliably.

This subteam, mechatronics and bogie integration systems team is responsible for maintaining the controls for the bogie movement. When a human input is received on the control panel, the bogie control system must register the steps needed to move each electrical component. Each sensor on the bogie is responsible for relaying data to the microcontroller and algorithms uploaded to the microcontroller. This is important because the bogie must iterate any possible safety checks on its own when it encounters any danger. Failsafe is also one of the main elements in the system as the bogie must be able to assess any standard failures before human interference is needed.

The microcontroller controls the timing of switching on and off the electrical components. Motors and actuators are turned on and off using 16 channel relays and H-bridges. Other electrical components such as fuses and resistors are useful when handling the circuit. Safety fuses and resistors protect the microcontroller and electrical components from shorted out and thus saving time and money replacing those components.

Wayside Power

Our job as the wayside team is to ensure that energy can be supplied to the vehicle so that onboard operation can continue smoothly and all motors will be able to operate when necessary. Some components needed to be powered such as propulsion motor, Hall effect sensors, actuators, gyroscopes, Arduino etc. Therefore, our goal was to provide a stable power input for the electrical components while bogie is in operation. The conductive system has to be safe, efficient, and effectively carry a current high enough to sustain the bogie with stability. The system has to also be accessible throughout the entire transit and remain active throughout operating hours without interruption. Moreover, the system has to be detachable for transporting to demonstration location, feasible for larger scale production, and maintainable. All of these key features are crucial to allow the system to run consistently which is why we've worked to provide a solution to each of these specifications.

Our design utilizes a fourth rail system that is attached parallel to the track; a fourth rail system is essentially 2 lines of conductive material that are attached to a power source, with one rail reserved for current and the other for return (ground). These rails will have a specified current and voltage derived from the power source and will be tapped into using a collector shoe that is attached to the bogie. The collector shoe is an arm that essentially protrudes from the bogie onto the track - making contact will allow for the onboard circuitry to complete and power will flow from the upper rail - through the bogie and down to the return bottom rail.

Under careful consideration, we chose a fourth rail design since it Isolates current from running rails (electricity only flows in conductive rail which is housed and safer).

It can also be easily implemented and maintained as opposed to other wayside systems that use the running rails to conduct electricity. This would mean that issues on the running track would cause an interruption in energy flow to the vehicle. If the track has a disconnect in our design, the power source will still be accessible; this also allows for maintenance to the track without blackouts. The materials required for this type of system allowed us to build the entire half-scale wayside system feasible as well, the main mass of materials dedicated to this system was copper and PVC tubes.

Our fourth rail establishes a reliable and safe design. Housing the conductive rail in an insulated material and having the conductive strip placed on the wall allows for benefits such as preventing build up from rain and dirt, as opposed to being placed on bottom concave which was considered due to the application of gravity based contact, where its weight would allow for the shoe to contact the rail by simply resting it. The design aids in protecting passengers from making contact with the conductive rail; an exposed conduit would be lethal if directly touched. It also uses cost effective material supply as well as maintain an ease of manufacturing due to its simplicity.

Our design is also created around a system that can deliver over 60V. Our power source will be simulated using an equivalent battery. In future designs, this power source can be replaced with a solar array configuration. However, this capacity is a reasonable choice to work under to maintain the efficiency of the overall ATN; anymore and the consumption rate would be unreasonable from a conservative standpoint.

Friction issues and maintenance were mitigated using conductive lubricant; since the fourth rail design requires a shoe to drag alongside the track, this friction would affect the travel of the bogie. The dry contact friction coefficient is directly proportional on dry contact. This friction would also cause wear over time and require replacement in certain sections of the conductive rail or even collector shoe. Our solution to mitigate this was to utilize a conductive lubricant specific to our amperage rating and had a guaranteed operating temperature above our intended range. Since combustion often occurs if the wrong lubricant is selected; a silicon-based lubricant was chosen with an operating temperature maximum of 400 degrees F. The silicon base assures that flaring doesn't occur from arcs and the operating temperature ceiling is to assure the heat from friction doesn't cause combustion.

A commercial collector shoe was implemented for a stable contact to the rail. The rating for the collector shoe was listed as 50 - 60A a rating higher than the load that will be placed on the copper strip ensuring its stability and compatibility with the rail. The commercial collector also used a spring four bar linkage mechanism that allows contact to be maintained throughout the operation.

Overall the system functioned as specified and all base goals were met, however certain issues can still be alleviated; however, this requires further spending and future consideration of other teams; further detail into this can be read in the conclusion section.

Full-Scale

Railings and Supports

This report was written to document the work done by the Support and Railings Subteam in contribution to the Spartan Superway. The Spartan Superway is an ongoing project whose goal is to create Automated Transit System (ATN) for the Silicon Valley which utilizes green energy. Establishing such a transit system would aid in the problems associated with today's current public transportation solutions, such as long wait times, lengthy commutes, and noise. It will also alleviate the area's traffic congestion and cut down on carbon emissions produced from transporting people. Currently there are only five proper ATN systems in the world and none have reached the full functionality possible for ATNs. By creating an on-demand elevated podcar system powered by solar energy, the Spartan Superway aims at creating a unique and convenient system for the Silicon Valley.

The focus of the Support and Railings Subteam this semester was on the support system of the guideway upon which the podcars will travel. The support system includes the columns, footings, crossbars, and the connections between each of these pieces. This semester focused on the structures and connections to X-beams developed in the previous semester.

The structure created was required to have a general palm-tree design and be able to integrate solar racking systems. In addition, the structure had to be attached using methods that would not compromise the experimental CFT's strength. This created a unique set of challenges for the team to solve. In addition to the structure design, the team also continued work on the X-beam and actually manufacturing of a physical prototype.

The final resulting solutions for design was the creation of three different profiles: two cat's eye shapes and a T-shape. The first cat's eye shape takes advantage of an upper arc to pull the weight of the cantilevered load onto the column. The second cat's eye included an under arc to support the loading through compression, such that the load rests onto the vertical column. Ultimately, the prime design the team chose to work on was the T-shape profile as it was the most stable and versatile of the designs.

An architectural model was built for each of the three models. The first cat's eye was created by cutting the pattern from wood. The wooden model also showcased the X-beam shape that would be used in the full-scale system. The second cat's eye was created from heat-bent PVC pipes and fittings. The structure was then covered in PVC insulation. This model better represents the desired round cross section of the column and structure that the public would see. The final T-shape model was built from a variety of materials. The main T-shape structure was modeled using laser-cut acrylic while the actual column structure was modeled with a PVC pipe. Clamps for the system were 3D printed using school resources. Footings for each of the models were created using standard PVC flanges and wooden bases.

At the same time, the team stayed in contact with their South African contact company to get final dimensions for a 4-foot stub of the X-beam. There were some delays in communication, but ultimately blueprints were sent to the manufacturing company Vander-Bend bent. The stub was completed in time for Maker's Faire. The overall cost for all four models for the Spring semester was $623.40.

Overall, the team met its goals over the course of the semester. Aesthetically pleasing models were successfully created and a stub column was completed according to schedule. This is the first year the team was established so there is still much work to be done in terms of supports and railings. Next semester's work can include bring the current or new designs up to full scale and include physical testing of models. Work completed this semester however, was meaningful and can serve as a base for future teams.

Introduction

In the Silicon Valley Bay Area, road congestion has become a serious concern in the past 5-10 years. The number of people traveling to and from jobs in city centers is constantly increasing, and there are more cars on the road than ever before. This presents a problem in regards to transportation, and the current traffic situation is not sustainable. The benefits of the carpool lane have been negated with the increase in the number of electric vehicles with carpool stickers, and public transit is not advanced or integrated enough to become an option for most workers in the surrounding area. Thus, the need for a sustainable and accessible transit network is at an all-time high. However, the desire for improved transportation does not just end with the user; gasoline usage is also increasing and costing billions yearly. Our society needs to shift towards a structure of transportation that runs on renewable energy and is accessible to users of all backgrounds.

One such solution is an automated transit network (ATN), in which there is no driver and the system operates itself with pre-determined routes or paths. This can be combined with the concept of a public rapid transit (PRT), which results in a lucrative combination of technology and societal contribution that could provide a means of transportation for millions of people in city centers worldwide. Spartan Superway plans to take this concept a step further by using renewable energy methods like solar to take the entire system off the grid of power consumption. Additionally, Superway also utilizes suspended railways, avoiding vehicles and obstacles on the surface road, whereas most public rapid transit systems involve traveling underground, which increases costs, or on the road, which contributes to road congestion. Therefore, Spartan Superway inherently provides more appeal for cities to implement as well as for commuters to utilize. Commuters often choose carpool or to operate personal vehicles rather than public transportation as their primary method of transportation due to convenience. However, owning a personal vehicle or carpooling can be expensive, inconvenient, or even inaccessible. Therefore, Spartan Superway improves access to transportation for the poor, elderly, and disabled by providing a public transportation that is both reliable and relatively inexpensive. Fortunately, students at San Jose State University have been working on Spartan Superway for four years now to make ATN a reality for cities worldwide. As of now, a 1/12th scale model and sections of both half- and full-scale models have been built to prove the concept of Spartan Superway to show what can and will be done.

Objectives

The driving force behind Spartan Superway is the implementation of a suspended, autonomous transportation network that is powered solely through solar energy. Three scale models — full-scale, half-scale, and 1/12th scale — are all aimed to demonstrate these concepts in their respective models to prove that this system is possible in a real-world environment. Each subteam within each scale model had different objectives that will all be defined in each subteam's respective sections. These objectives needed to be met by each subteam to present a fully functional prototype at the end of the academic year. This year, the Spartan Superway project had the following objectives:

1. Demonstrate the functionality of the bogie system on the half-scale model
2. Redesign each scale model so that each model relies only on solar energy to power the whole system
3. Build a 1/12th scale prototype model to demonstrate an autonomous, suspended system that navigates a podcar on a track
4. Build a structural model of a section of the track to demonstrate the foundation and construction of various sections of the track
5. Present a fully functioning prototype at the Paseo Prototyping Challenge and Maker Faire

Chapter 1: 1/12th Scale

Bogie Improvement

Background and context

Is important for the bogie improvement team to work on improving the switching mechanism and work on the clearance with the track because the bogie will be able to easily move around the track. By properly placing of sensors in the right location that will allow the bogie to be more compatible with the track. The original plan was for the bogie to be made more narrow than it was finally made but since the track is not made uniform at all the bends we were not allowed to make the bogie as narrowed as the team had planned. However, the width of the bogie was narrow enough to have the bogie properly run through the track. The compatibility of the bogie with the track was also influenced by the switch the controls team made to hall effect sensors. Enough testing was done with the 3D printed bogies to solve all the issues the bogies had from previous years.

Description of the Subteam and Objectives

The purpose of the twelfth scale model is to present the bigger picture of Spartan Superway on a smaller scale. The twelfth scale bogie is a simplified version of the full-scale mechanism of the bogie. Though the mechanism is different, the concept is the same. The figure below compares the mechanism between the small-scale bogie and the full-scale bogie. The small-scale bogie's simpler design allows the full picture of Spartan Superway to be showcased. It also allows all the other subteams in small scale to showcase their work. With the bogie, many aspects of Spartan Superway are demonstrated that aren't demonstrated in the other teams. This includes the Spartan Superway app and podcar communication.

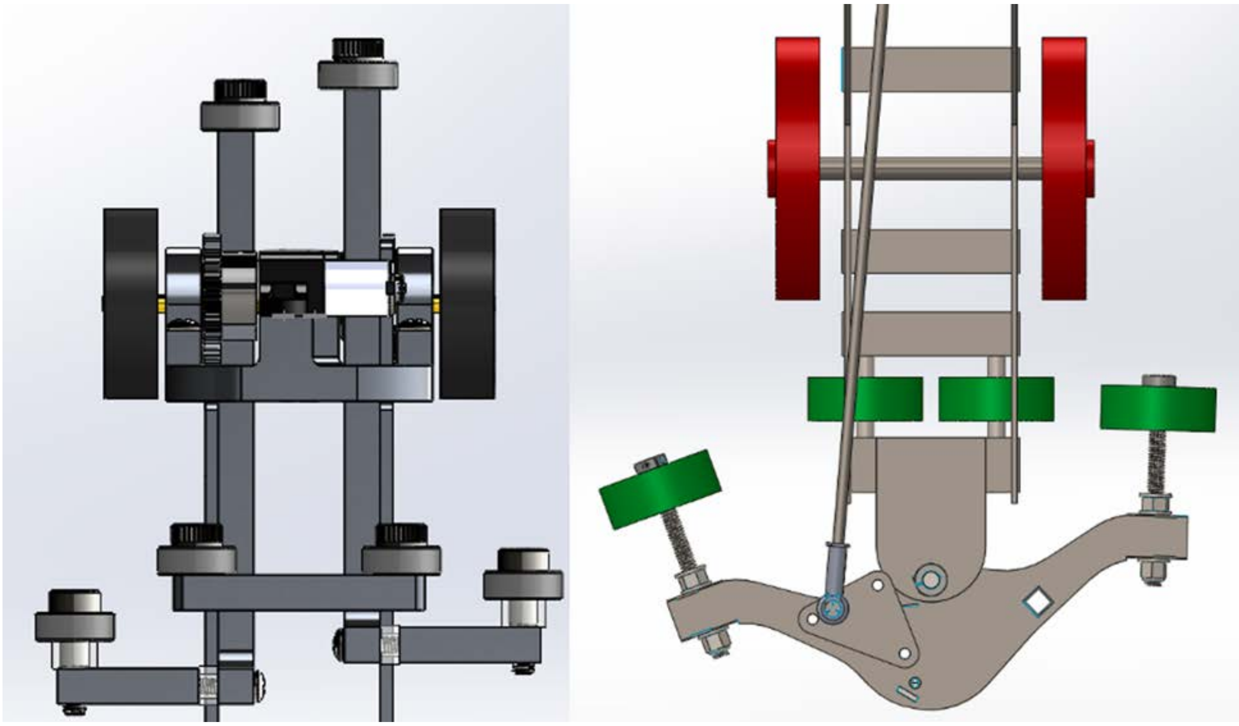


Figure 1-1. 1/12 Scale Bogie compared to Full Scale Bogie.

In order for the Spartan Superway's full capabilities to be demonstrated, the bogie needs to be able to run smoothly through the track. For this to happen, there were objectives put forth for the twelfth scale bogie. The bogie needed to be lighter but robust. The previous design was made of aluminum which added a lot of weight. The bogie also needed to be easier and cheaper to manufacture. The old bogie had to be CNC'd or milled. Parts and hardware also needed to be reduced. Reducing the number of parts and hardware will make a more solid model that is robust and effective. The switching mechanism of the bogie also needed to be improved. The older design used piano wires which were easily malleable and hard to maintain. Finally, the width of the bogie needed to be narrower in order to accommodate for the corrugated metal strips that will be implemented to the track by the Positioning Team.

In addition to improvements needed to be done to the twelfth scale bogie, improvements were also needed on the twelfth scale cabin. The cabin serves as the housing for the electrical components of the bogie. It also serves as a visual aid. The cabin allows the entire twelfth scale model to be recognized for what it is. A cabin is a familiar aspect that allows viewers to recognize what the model is.

The previous design of the scale cabin was bulky and hard to maintain. It consisted of 3 separate 3D printed parts that held the previous iterations of the electrical components. The design was heavy and failed to make an efficient use of space. The new cabin needs to be simpler but still maintain its structural integrity. It needs to be able to withstand an inevitable fall. It also needs to house the new electrical components that the controls team will be utilizing. Lastly, the new cabin needs to be easily recognized as a cabin for Spartan Superway.

Design Requirements and Specifications

Bogie Design Requirements and Specs

The priority specification requirements for the bogie are to be able to make clearances on the track. Making the clearances on the track allows the bogie to be able to run smoothly. By setting forth specifications, it also allows the bogie design to have something to strive for once it moves from CAD to physical model. The following figures show the specifications for the bogie design. The top rail as well as the bottom rail are cleared by the bogie with the given specifications. It also allows the bogie to run close enough to the track that it runs smoothly. Another specification that is required in order to allow the bogie to make clearances is the length of the link. The link connects the servo horn to the bogie arm. A length of 0.85 inches will achieve the specified clearances with only a 90-degree rotation from the servo.

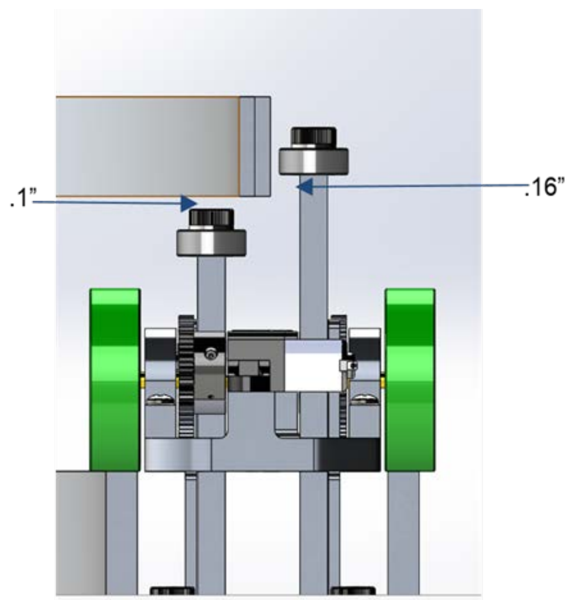


Figure 1-2. Bogie clearances with the track.

Another requirement for the bogie is for it to be easily manufactured. To do this it was decided that the bogie will be entirely 3D printed. Doing so will also change the design of the bogie. Number of hardware and parts of the bogie can also be reduced due to the fact that the design will be entirely 3D printed. Parts can now be combined together, decreasing the number of parts and the need for some of the screws. A 3D printed bogie will have a design that can easily replace spacers and easily combine parts which reduce the number of hardware.

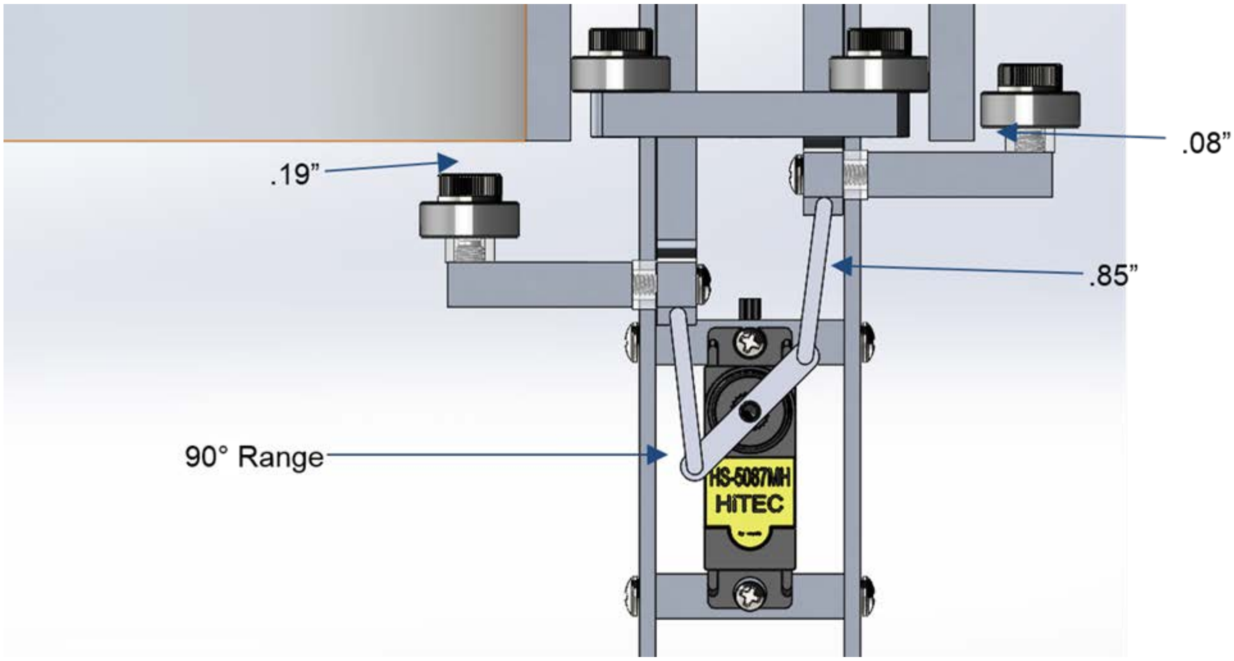


Figure 1-3. Overall clearances.

Cabin Design Requirements and Specs

This semester the team made a lot of changes to the cabin used in previous years. During previous years, the cabins were made into more than two parts and it was hard to access the electronics inside. The requirements of the cabin were that it needed to be big enough to hold the ultrasonic sensor, battery packs, Arduino, a shield, an RFID module, LEDs, buttons and the necessary connecting wires. The cabin had to easy attach to the bogie and make it easier to access the electronics. The overall size of the cabin was limited by maximum size the 3D printers at the Tech Shop are allowed to print since most of the cabins were printed in the Tech Shop. The printer in the design center can print a piece as large as 10 x 11 x 12", however, since the printer is not as reliable as the ones at the Tech Shop. Therefore, the team decided to not change the overall size of the cabin in case it was required to print at the Tech Shop again. The final size of the cabin is about 5 x 5 x 3".

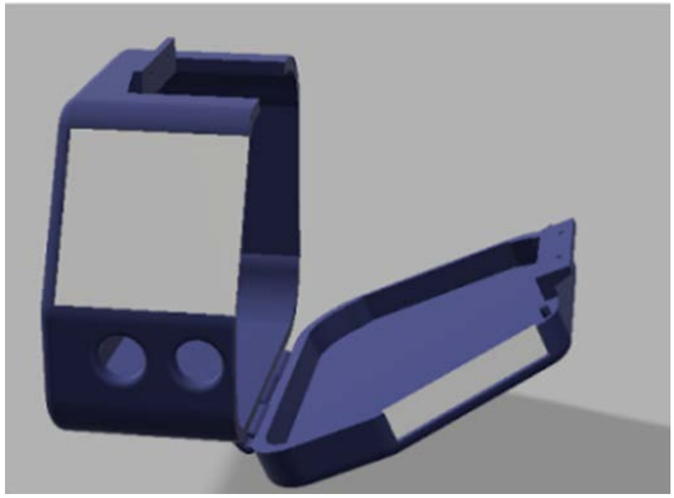


Figure 1-4. Cabin Design for Paseo Prototyping.

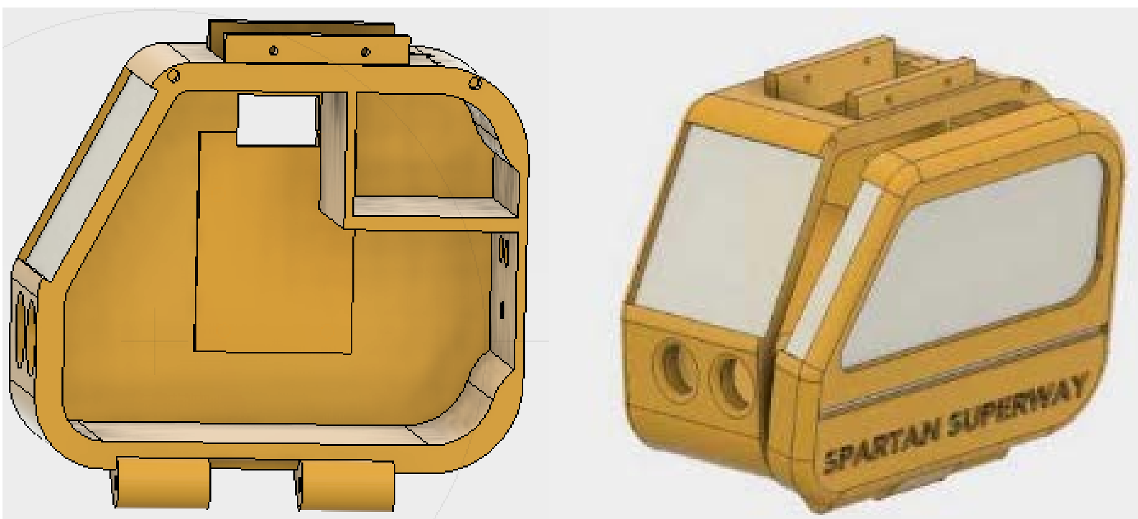


Figure 1-5. Cabin design for Maker Faire.

During Paseo Prototyping, some of the bogies fell due to people stopping them while running on the track and also because of the wind. When the bogies fell some parts of the cabin broke specially the doors at the top and at the bottom of the hinge. The controls team also suggested that the biggest part of the cabin be fully attach to the bogie and just have the other part open and close, that way nothing had to be unscrewed when the inside needed to be accessed. Due to the falls and some parts of the cabin breaking the wall thickness was increased to .3 mm all around except on the wall where the RFID is located to not interfere with the signal. The part where the hinge is located was also increased to .2 mm all around. The App team also suggested to another battery pack to the bogie and a slot on the top was added for that battery back to have the pack not interfere with the wire connections to the shield. The final design of the cabin worked well at the Maker Faire, when the bogie falls due to collisions the cabins no longer brake. Two cabins were able to be printed with Professor Youssefi and they are printed with ABS material at 100 percent density, and they are more durable. Everything works fine with the cabins now the only thing that they need to improve on is the snap closing technique, it snaps open and close successfully with the ABS cabin but sometimes that snapping mechanism breaks off from the PLA cabins and they have to be held together closed with tape. The cabins can be printed as fast as 5 hours for the bigger piece and 2 hours for the smaller piece.

State-of-the-Art/Literature Review

An Automated Transit Network or ATN is a network of vehicles, which does not require a driver, and are operated by separated guideways. The goal of ATN is to improve the effectiveness of transportation. The most common use of transportation, bus or train, must stop at each station on the route, but one of an ATN's goal is to only at stop when the passenger needs it. Typical transportation systems travel on a set schedule and on a set travel loop or line, but an ATN system bring a lot to transportation. An ATN brings a lot of benefits to a city, including less car congestion on the streets, faster transportation for passengers and an environment friendly transportation system. Experts that have done studies about ATN systems claim that it would be cheaper to build these new systems compared to the traditional rails. These, environment friendly networks would be cheaper to build because they would require smaller stations, tracks and vehicles.

ATN Technology Around the World

In the beginning of 2010, there were a couple of ATN systems already running and others under construction throughout the world. Some of the ATN systems that were already functioning were at the London-Heathrow Airport and in Masdar City, in United Arab Emirates. In the following years, agreements were signed to start building ATN systems in Suncheon, South Korea and in India.

Figure 1-6. ATN system in London-Heathrow Airport.

ATN Technology in San Jose

Since the November 2000, citizens of Santa Clara County approved a measure which allowed a 30-year half cent sales tax, that would go towards the improvement of public transportation. In the summer of 2010, the City Council allowed the city to hire a team to conduct a study on the possibility of building an ATN in the city, and that study was finalized in the year 2012. The analysis demonstrated the possibility to build an ATN, which will allow passengers to travel from major transit stations around the city to the city's International Airport. The analysis, funded by the Santa Clara Valley Transportation Authority, showed that even though there would be a lot of benefits from creating an ATN in San Jose, there still more tests to be done and many regulations that the system has to pass. According, to the study done it was estimated that building an ATN system in the city of San Jose would cost $758 million. That amount is cheaper, compared to the $967 million, it would cost to construct a more traditional Automated People Mover (APM) system, that already exist in some airports across the country.

Description of Your Design

Bogie Design

The final bogie has a design that is simple, functional, as well as robust. Due to the addition of corrugated metal strips on the track, the bogie was made narrower for a safe measure. It was narrowed by 0.07 inches on each side that ensured that enough clearance for the corrugated metal strip. The final design also features less parts and less hardware. This was enabled by the 3D printing technology and would've been difficult and expensive to perform on aluminum.

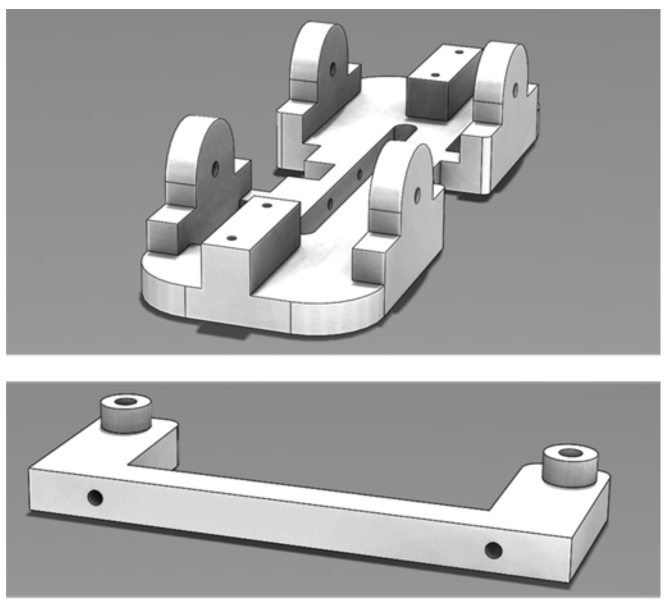


Figure 1-7. Improvements of 3D printed parts.

The top motor plate is now one solid piece instead of 5 pieces held by screws. Parts that needed spacers were also combined into one, removing the need for adding spacers while assembling. The final shows a decrease of 30% in weight from the aluminum bogie to the plastic bogies.

Final design of the switching mechanism was also able to function successfully. It was entirely made of 3D printed parts; each part takes less than a minute to print. It consists of a simple link designed to length. The link is connected to the bogie's switching arm with a 3D printed hinge that inserts and cocks into position. This helps secure the linkage. The link is connected to the servo horn via a 1 mm diameter paper clip cut to length with soldered ends to secure into place.

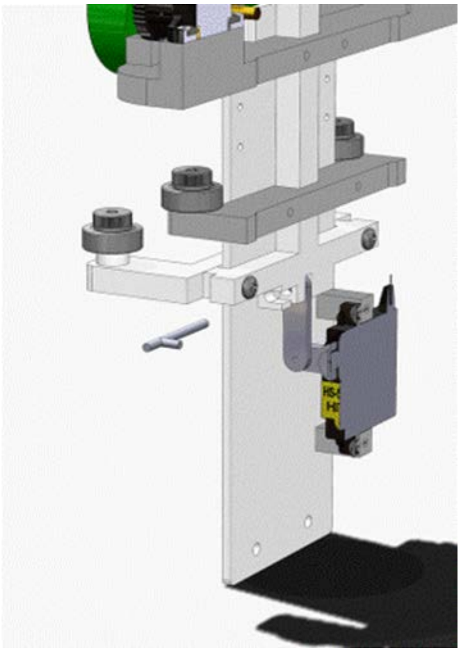


Figure 1-8. Switching mechanism.

Cabin Design

The final design of the cabin consists of two parts that attach to the bottom of the bogie. The cabin width is designed precisely to be within range of RFID cards placed on the track for the controls team. The cabin features a slot for the RFID reader outside of the cabin that allows the RFIDs on the track to be read. The RFID reader is also connected to the microcontroller through a clearance slot on the cabin. The front of the cabin also features clearance holes for the ultrasonic sensors. This allows for the sensors to know when there is something in front of it. The top of the cabin also contains an opening for the wires from the sensors, servos, and motors to connect to the microcontroller. The back of the cabin also features specified slots for the buttons LEDs to control the bogie and indicate its status. On the bottom, the cabin contains a hinge design that allows the two parts to connect but also to open up for easy access to the electrical components inside the cabin.

Figure 1-9. Left side of the cabin: RFID.

Figure 1-10. Back side of the cabin: buttons.

Analysis/Validation/Testing

In order to make sure that 3D printing the parts will be able to hold up, the accuracy of the features and the structural integrity of the 3D printed parts need to hold up. A test print of part was made and essential features were compared to dimensions in the design and the aluminum part.

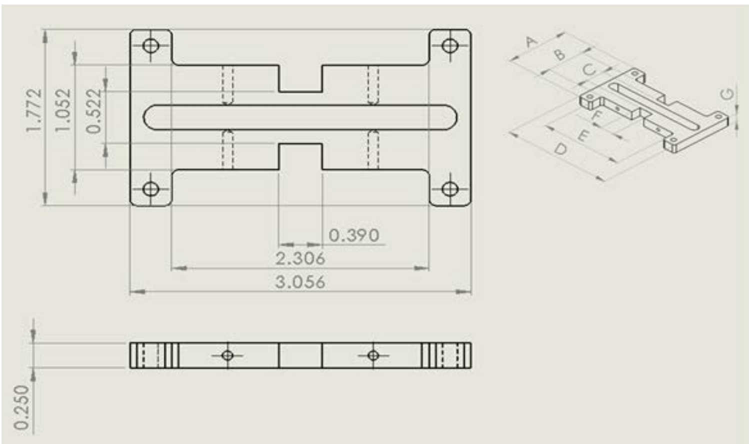


Figure 1-11. Detailed drawing of the Mid Plate.

Table 1-1. Detail comparison of 3D printed parts vs CNC parts.

Feature	Target Dimension	CNC Dimensions	3D Print Dimensions
A	1.772 in.	1.775 in.	1.778 in.
B	1.052 in.	1.149 in.	1.154 in.
C	0.522 in.	0.611 in.	0.623 in.
D	3.056 in.	3.063 in.	3.079 in.
E	2.306 in.	2.299 in.	2.291 in.
F	0.390 in.	0.392 in.	0.366 in.
G	0.250 in.	0.251 in.	0.244 in.

The table shows essential dimension measured on the 3D printed part and the aluminum part. They are then compared to the original dimensions from the design. After getting the average deviation from each part it is concluded that the 3D printed parts' dimension will hold up just as well as the aluminum CNC'd parts.

The next test needed to be done was to make sure that the 3D printed parts will be able to hold up with the given loads. Finite Element Analysis was done on 3D parts to make sure that the parts will not fail.

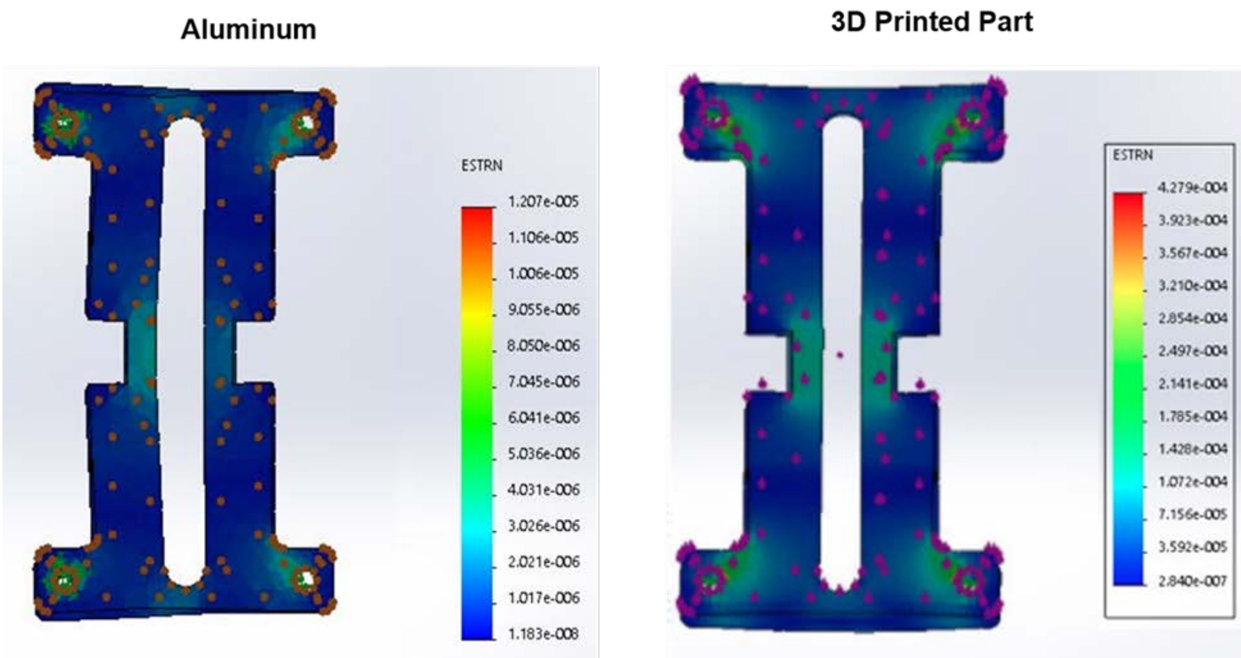


Figure 1-12. FEA of Aluminum vs 3D printed part.

The results show that deformation of the 3D printed parts will be minimal with given forces. Thus, it was concluded that 3D printed parts will be hold up.

Assembly Procedure

Since the new bogies have fewer parts and fewer screws it is easier to assemble it. As long as the appropriate screws are used for the bogie parts then it should easily be assembled. Each part is held with 4-40 screws. The only parts held with 2-56 screws are the motors and the hall effect sensors.

Another thing that needs to be assembled are the switch arms to the servo horn. They are connected with a paperclip approximately 1mm in diameter and cut to length. The ends of the paperclips are then soldered in order to prevent them from slipping off.

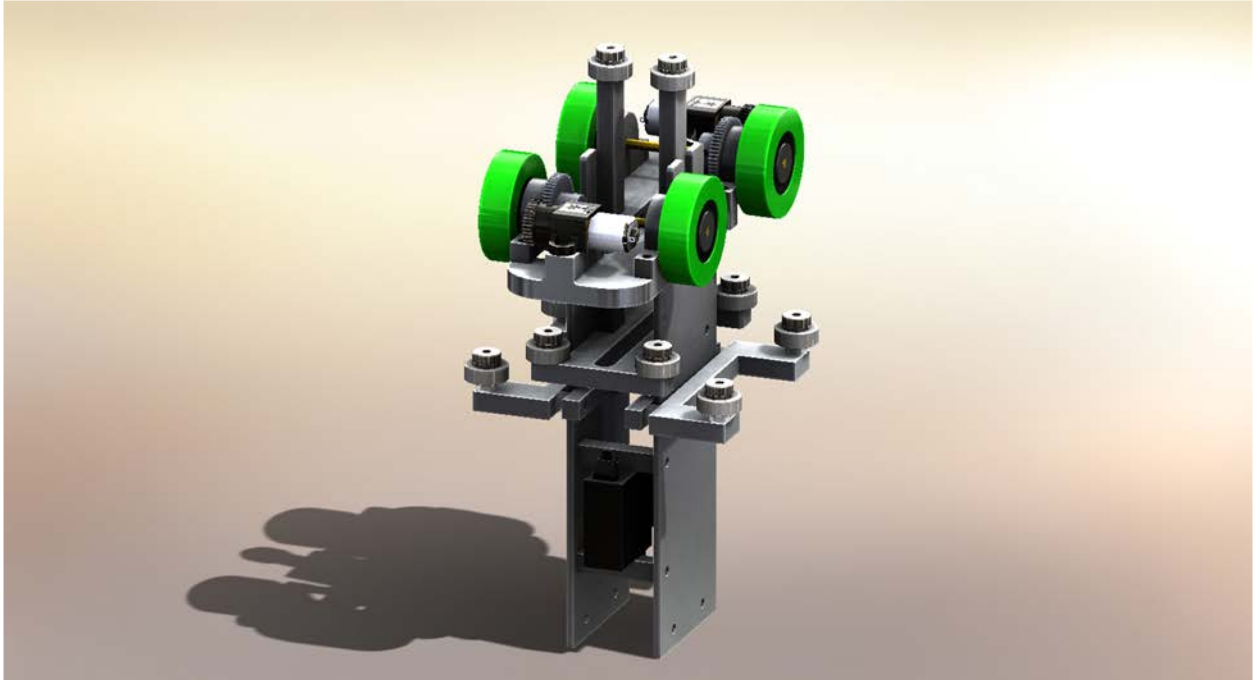


Figure 1-13. Complete assembled bogie parts on Solidworks.

Expenses

A detailed list of items bought are shown in the table below. It is arranged in the order they were purchased. The subteam had a budget of $1,428.64 to build 6 fully functioning bogies and cabins.

Table 1-2. Bill of Materials.

Site	Description	Price	Qty.	Total + S&H
SDP-SI	Module 0.4, 36 Teeth, 20° Pressure Angle, Acetal / Brass Insert Spur Gear	$12.97	10	$144.38
SDP-SI	Module 0.4, 60 Teeth, 20° Pressure Angle, Acetal / Brass Insert Spur Gear	$13.63	8	$109.04
McMaster Carr	18-8 Stainless Steel Shoulder Screw 5 mm Diameter x 5 mm Long Shoulder, M4 x 0.7 mm Thread Size	$2.57	60	$176.70
McMaster Carr	Nylon Unthreaded Spacers 1/8" OD, 1/8" Length, for Number 0 Screw Size, Packs of 25	$7.77	1	$7.77
McMaster Carr	316 Stainless Steel Phillips Rounded Head Screws Super-Corrosion-Resistant, 4-40 Thread Size, 5/8" Long, Packs of 50	$6.97	1	$6.97

McMaster Carr	316 Stainless Steel Phillips Rounded Head Screws Super-Corrosion-Resistant, 4-40 Thread Size, 1/4" Long, Packs of 50	$3.73	2	$7.46
McMaster Carr	General Purpose Tap for Closed-End Hole Threading, M4 x 0.7 Thread Size	$6.67	1	$13.48
McMaster Carr	General Purpose Tap for Closed-End Hole Threading, 4-40 Thread Size	$6.13	1	$33.30
McMaster Carr	General Purpose Tap for Closed-End Hole Threading, 2-56 Thread Size	$9.72	1	$9.72
Amazon	0.020" Diameter Music Wire, 1/4 Pound Coil	$6.67	1	$7.25
McMaster Carr	18-8 Stainless Steel Phillips Rounded Head Screw 2-56 Thread Size, 1/4" Long, Packs of 100	$4.49	1	$11.19
McMaster Carr	Low-Strength Steel Hex Nut Zinc-Plated, 2-56 Thread Size, Packs of 100	$0.97	1	$0.97
McMaster Carr	General Purpose Tap for Closed-End Hole Threading, 2-56 Thread Size	$9.72	1	$16.81
McMaster Carr	Captive Pin M1 Diameter, 4mm Length, Packs of 25	$10.05	1	$17.57
McMaster Carr	18-8 Stainless Steel Phillips Rounded Head Screw 4-40 Thread Size, 3/8" Long, Packs of 100	$4.30	1	$4.30
McMaster Carr	M4 x 0.7 mm Thread Size, 4.7 mm Installed Length, Heat-Set Inserts for Plastics	$14.25	1	$21.77
McMaster Carr	4-40 Thread Size, 0.219" Installed Length, Heat-Set Inserts for Plastics	$11.65	2	$25.47
McMaster Carr	2-56 Thread Size, 0.115" Installed Length, Heat-Set Inserts for Plastics	$9.67	1	$10.57
				Total
				$624.71

Results and Discussion

In the end the team was able to achieve all of the objectives on time and under budget. Six fully functioning bogies and cabins were manufactured. The bogie was able to run smoothly through the track and the cabin was able to effectively and efficiently house all the electrical components needed by the controls, positioning, and app team. In the end the team was able to produce the bogie and cabin from design to final physical product.

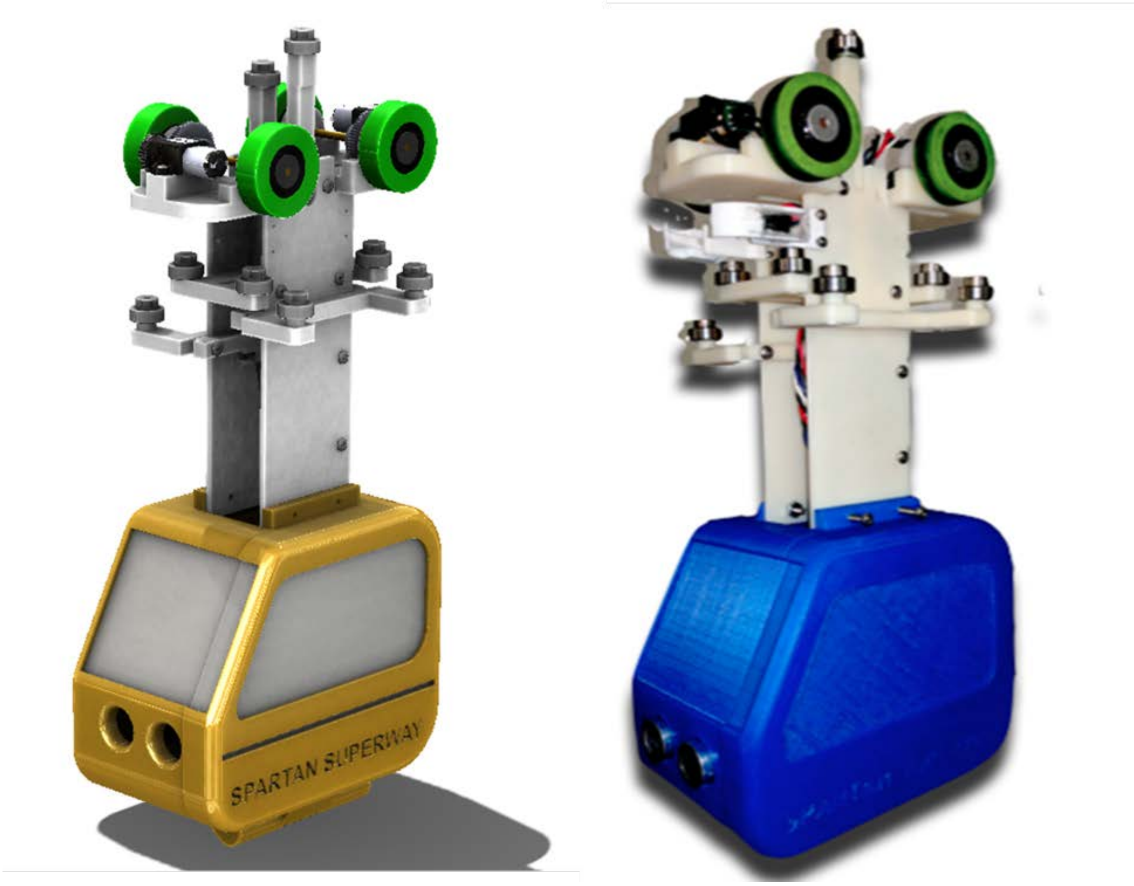


Figure 1-14. Assembled bogie on Solidworks (left) and 3D printed bogie (right).

Conclusions

In conclusion, all objectives were achieved and the final products were able to function properly. A lot was learned by the team over the duration of the project. A main aspect of the project was 3D printing. The capabilities of 3D printing technology are advancing even more and this project has allowed the team to become familiar with the technology. Becoming familiar with 3D printing technology is one of the major accomplishments of the group. This skill can be taken into the real world where many companies utilize the technology to make their own products.

Suggested Future Works

Though the bogie and cabin perform the way they were designed to, there are still ways to further improve. Most of the holes on the bogie parts are tapped. Some have threaded metal inserts placed in them in order to increase the longevity of the parts. This can be done for all of the bogies but appropriate length screws need to be bought to make them effective.

Position Sensing

Background and context

There was no method to determine the accurate real-time position of the podcars on the track for the small-scale model. The small-scale position sensing team is a team focusing on designing a new position sensing system for the small-scale model. The new position sensing system should accurately provide real-time location information of the podcars on the track to the system. The position sensing system should be developed cost-effectively. The primary method to develop the positon sensing system is to use a Neodymium rare earth cylinder magnets and a Hall Effect sensor to create a position sensor, and use the position sensor to detect the peaks of the corrugated ferrous strips on the track. The team should evaluate the feasibility of developing a position sensor, manufacturing the corrugated ferrous strip and attaching the corrugated ferrous strips on the track.

Description of the Subteam and Objectives

The purpose of the 1/12 scale model of the Spartan Superway project to lower the cost of designing the components and simulate the full-scale Superway project. The Position Sensing team is supposed to work with other vehicle control teams to develop a "smart" and autonomous multiple-podcar system. The main focus of the team is to solve the positioning problem to create an accurate and reliable podcar positioning system for the twelfth scale model this year. The team is supposed to create a position sensing system with linear magnetic encoders systems. The linear magnetic encoder system will be created by implement Hall Effect sensors and neodymium magnets with a corrugated ferrous strip on the guide ways. The objectives of the position sensing subteam will be:

1. Design a position sensing system with corrugated ferrous strips, Hall Effect sensors, and magnets.
2. Design a device to manufacture the corrugated ferrous strips.
3. Design a position sensor with Hall Effect sensors and magnets to detect the peaks of the corrugated metal strips.
4. Program the linear encoder in order to provide an incremental position signal to the position sensing system.
5. Integrated the linear encoder and the corrugated metal strips with the one-twelfth scale model.

Design Requirements and Specifications

The design of the position sensing system must meet the following requirements and specifications:

1. The design of the position sensing system must use a Hall Effect sensor as the signal sensing components.
2. The available space between the track and the bogie was determined to be 9.5 mm. Both the corrugated ferrous strip and the position sensing should be fit in the 9.5mm gap.

3. The corrugated ferrous strip must be able to be mounted on the track and reserve space between the track and the bogie for the clearance of the bogie and the sensors from the control team.
4. The design will use a magnet which is ⅛ in x ⅛ in Cylindrical Neodymium rare earth cylinder magnet and an SE022 Hall Effect sensor. The sensor bracket must be able to hold both the magnet and the Hall Effect sensor in a constant separation distance.
5. The surface area of the corrugation on the corrugated metal strip must be at least twenty-five square millimeters to allow the Hall Effect sensor to sense the magnetic field.
6. The corrugated ferrous strips must be manufactured in a cost-effective way.

State-of-the-Art/Literature Review

For modern ATN system, the most common method of tracking railcars in transit is using a Global Positioning System (GPS) receiver and a satellite transceiver on board the locomotive, and a local area network of railcar tracking units. For some of the ATN system, in other to increase the accuracy and efficiency of tracking system, the railcar tracking units transmit their unique identifications (IDs) to one of the tracking units in the local area network acting as an administrator tracking unit. The administrator tracking unit maintains a list of IDs and periodically transmits the list of IDs to the locomotive. The locomotive locates itself using the GPS system and periodically transmits its position and velocity to the central station (Pradeep).

Also, the cell phone is one of the indispensable necessities of life. TRAC-IT is a mobile phone application that records travel behavior by collecting real-time GPS data and requiring minimal input from the user for data such as trip purpose, mode of transportation, and vehicle occupancy (A, Paola).

To create the most efficiency position sensing system for the 1/12 scale model of Spartan Superway, the system should qualify high accuracy and modernization features. Therefore, the system involves the features of GPS and mobile phone application. Instead of only tracking the bogie at some specific position, the goal of the system is available to track the bogie at all of the positions of the track. Therefore, by using Hall Effect sensor to sense the corrugated metal strip can achieve this goal. Another advantage of this design is low cost. Moreover, after all of the data collecting by Arduino, it will send the data to the Cell Phone, then customers can know where the bogies are locating at.

Description of Design

Position Sensor

The position sensor consists of a corrugated ferrous strip. After testing, the team decided that the minimum separation between the magnet and the Hall Effect sensor should be 6 mm. If distance between the magnet and the sensor is less than 6 mm, the magnetic field will be to strong and the sensor cannot detect any change in the magnetic field strength. The Hall Effect sensor must be place at a maximum distance of 2 mm away from the peaks of the corrugated ferrous strip (Figure 1-15). Two millimeter is the maximum distance that the magnetic field of the magnetic can be affected by the peaks of the corrugated ferrous strip. If the separation of the corrugated ferrous strip and the Hall Effect sensor is further than 2 mm, the magnetic field will not be affected by the peaks of the corrugated ferrous strips.

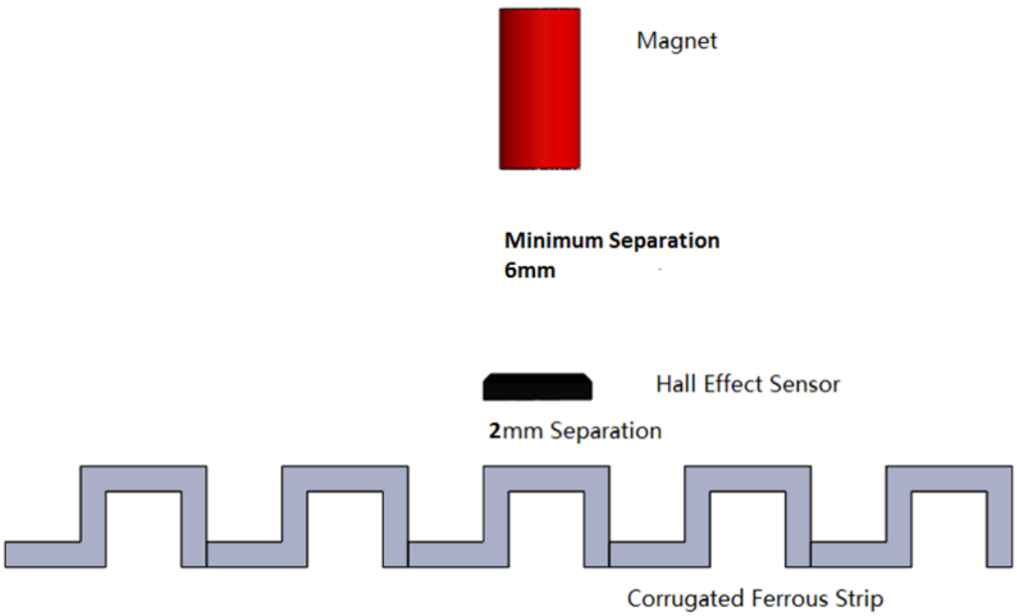


Figure 1-15. Combination of Corrugated Ferrous Strip, Hall Effect Sensor and Magnet.

The SE022 Hall effect sensor is a Hall Effect sensor with hysteresis circuit to reduce signal noise. The volume of the hysteresis integrated circuit is too large for the bogie. Hence, the Hall Effect sensor will be de-soldered and put on the bracket, the integrated circuit board will be placed in the cabin, and they will be rewired with long wires.

Base on the test result, the sensor bracket is designed, the complete view of the position sensor is shown in Figure 1-16.

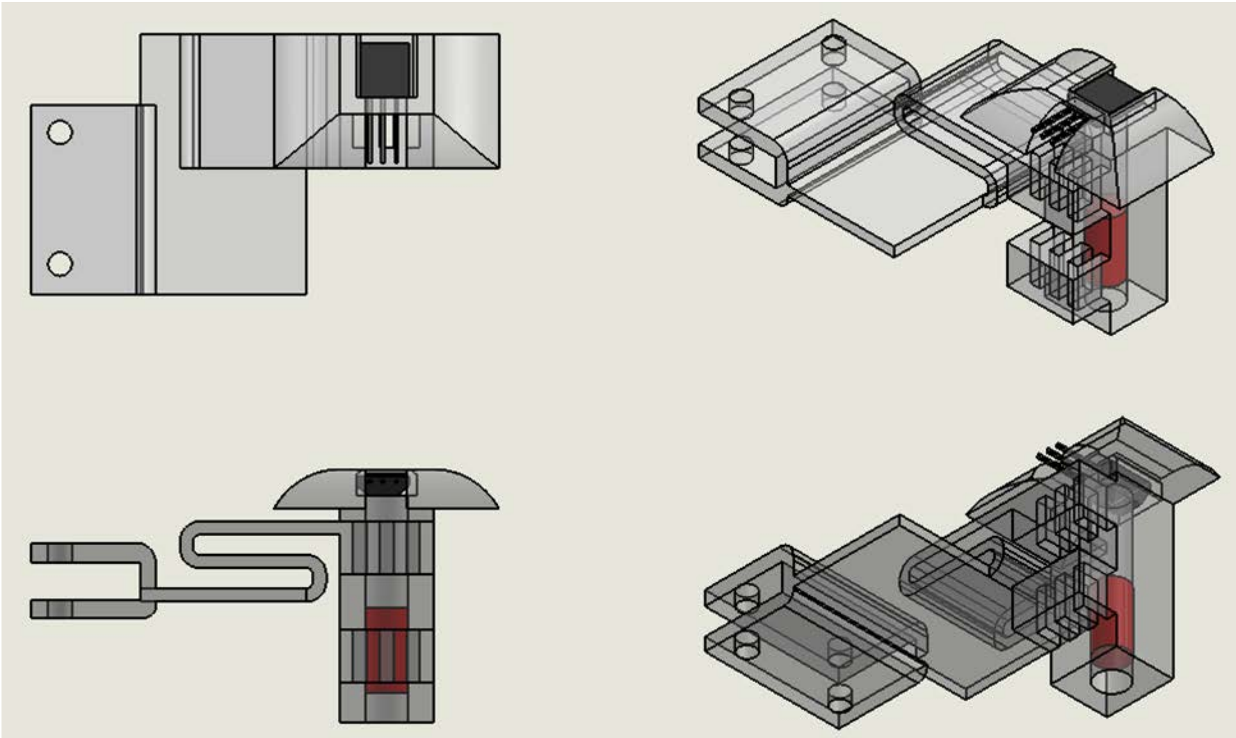


Figure 1-16. The completed view of the Position Sensor

The sensor brackets are 3D printed. The minimum 3D printing resolution for the bracket should be 100 microns. The designed position of the position sensor is on the rear of the bogie.

Corrugated Ferrous Strip

The prime design of the corrugated ferrous strips is shown is Figure 1-17. The corrugated ferrous strips were designed to have a 5-mm corrugation pitch and 4-mm height. To manufacture the corrugated ferrous strip, the team designed a set of gear which is able to corrugate ferrous strips. The gears were 3D printed with ABS (Figure 1-18). The team bought a 30-gauge steel sheet. The steel sheet was cut in into strips first, then corrugated by the gear set. The designed width of the corrugated ferrous strip should be larger than 5mm in order to have enough surface area to interfere the magnetic field, but not too large so that it will not affect the other sensors on the bogie. Because of human error during cutting the steel sheet into strips, the width of the ferrous strips should be 5.5 ± 0.5 mm.

The corrugated ferrous strips were attached on the track with superglue. It is currently the strongest adhesive the team could find. However, during assembly and disassembly of the track, a portion of the corrugated ferrous strips might be detached from the track.

Due to vehicle tilting, the separation of the bogie and the top part of the track would be widened on the corners of the track. The positon sensor will not be able to detect most of the peaks because of the wider separation. The solution to this problem was to add a layer of heavy duty double side tape between the track and the corrugated ferrous strips (See Figure 1-19). The double side tape brings the corrugated ferrous strips closer to the position sensor and also works as an adhesive for the corrugated metal strips and the track.

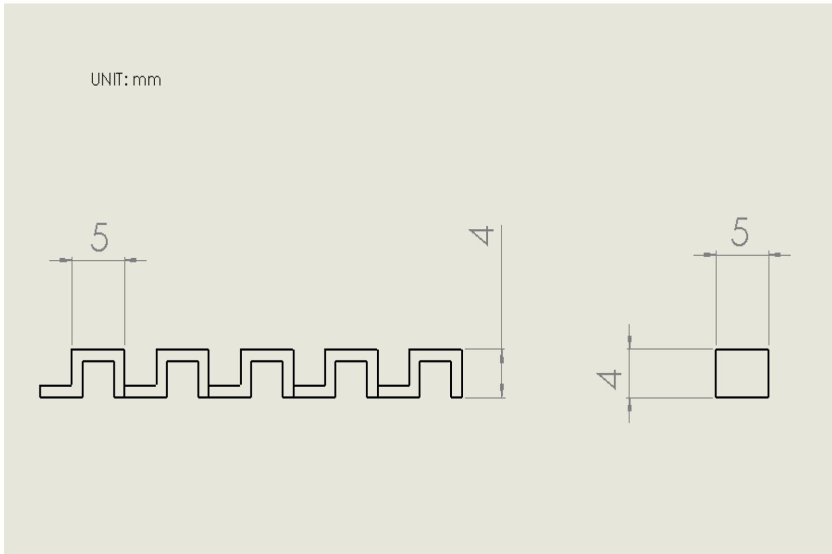


Figure 1-17. Profile view of the corrugated ferrous strip.

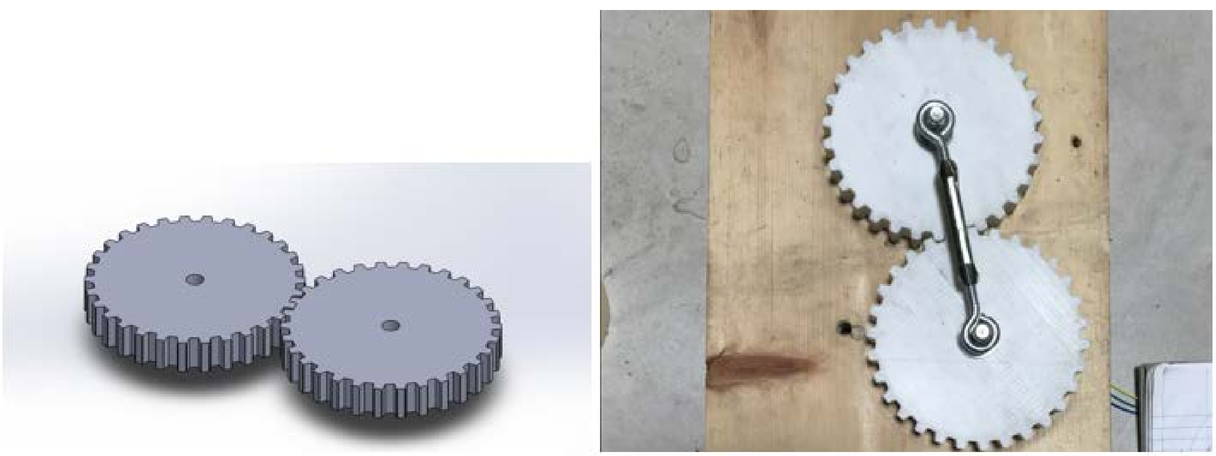


Figure 1-18. 3D Drawing and Prototype of the Gear Set

Figure 1-19. Double Side Tape works as adhesive for corrugated ferrous strips and the track.

Programming

After the corrugated ferrous strips were installed on the track and the position sensors are completed. The position sensing system will be programmed with the program in Appendix A2-2. The position sensors were connected to the analog pins. The program has to be calibrated for every bogie individually. The calibration requires to run the bogie on the track, and get the analog reading of both sensors reading the peaks and the valleys on the track. The team has to change the values in the int sensor (int sensorValue) function in order to convert it to digital signals.

Analysis/Validation/Testing

Corrugated Ferrous Strip

Because the corrugated ferrous strip should have enough surface area to interfere the magnetic field, but not too large so that it will not affect the other sensors on the bogie, the width of the metal strip has to be larger than 5 mm. After cutting a set sample ferrous strips, the team found that because of human error and the tolerance level of the sheet metal shear while cutting the steel sheet into strips, there will be a 0.5 mm tolerance on the ferrous strips. As a result, the width of the ferrous strips should not aim at 5.5 mm instead of 5 mm.

Position Sensing System

The team created a test model of the position sensing system (see Figure 1-20). The team attached a piece of the corrugated on a wood wheel. There was a position sensor placed at a fix distance to the corrugated ferrous strips. The system was an analogy of the bogie with the position sensor running on the track.

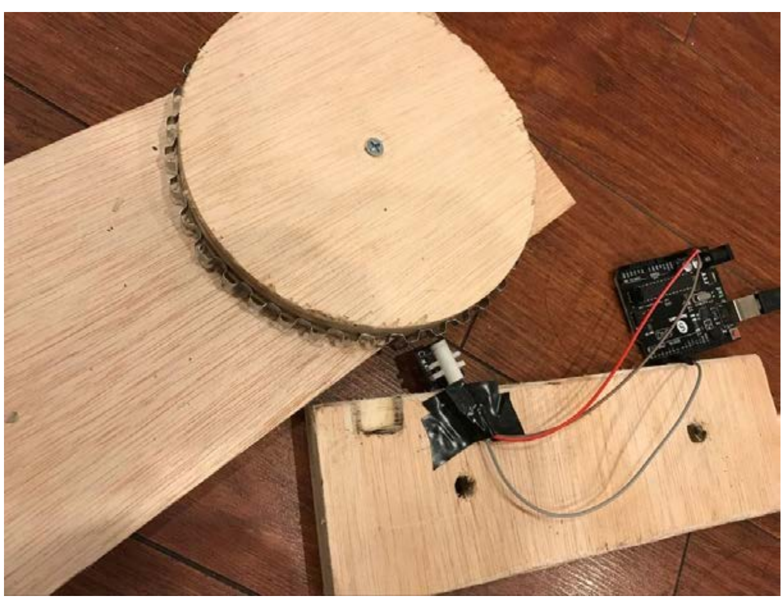


Figure 1-20. Demonstration of the Position Sensing System.

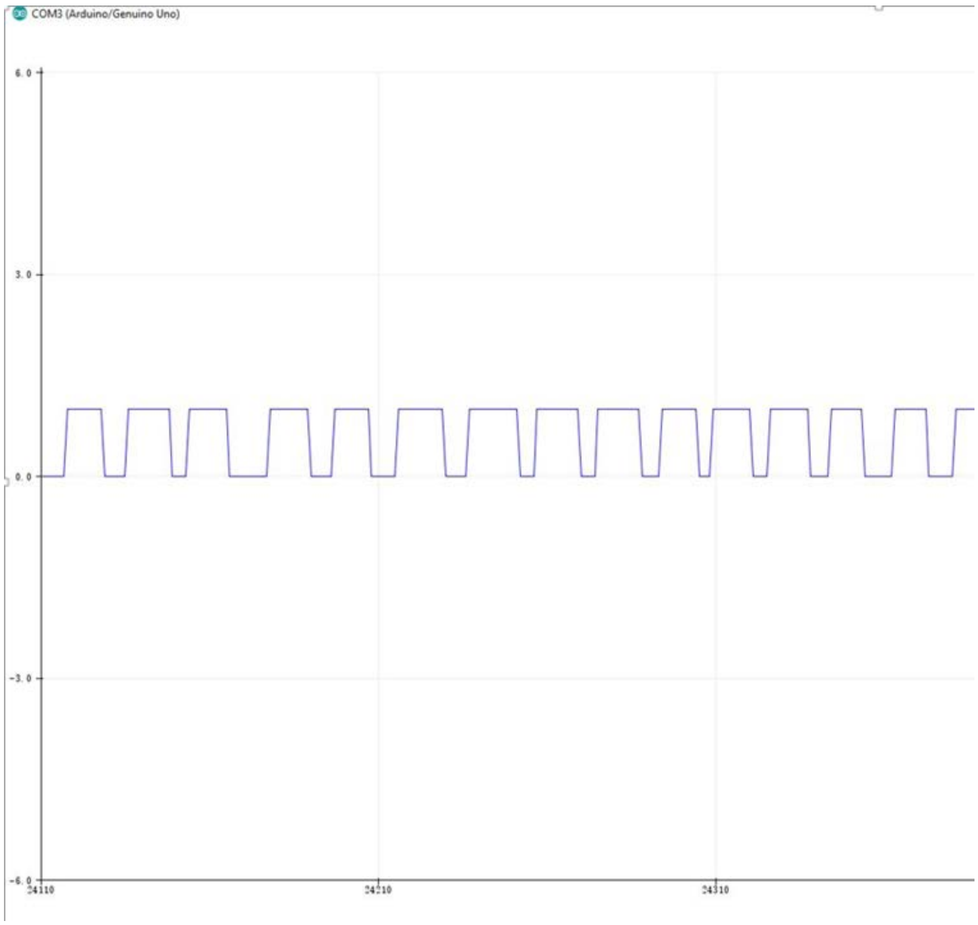


Figure 1-21. Testing result from the demo.

The result of the demonstration was shown in Figure 1-21. The analog signal was converted to digital signal. 0 means peaks and 1 means valley. The model outputted a perfect square wave with the program. With the square wave, the team will be able to count the peaks.

Procedure / Instruction Manual

Gear set

The gear set should be operated by two people. One person rotates both gears simultaneously. The other person keeps both sides of the ferrous to avoid vertical displacement. The corrugation process should be slow and with a steady speed.

Position sensor

The program has to be calibrated for every bogie individually. The calibration requires to run the bogie on the track, and get the analog reading of both sensors reading the peaks and the valleys on the track.

The team has to change the values in the int sensor (int sensorValue) function in order to convert it to digital signals.

Money Spent on Project

Table 1-3. Bill of Materials.

Name of Part	Pricing Quotes (Dollars/each)	Lead Time of Service (days)	Quantity	Purchased From
Steel Sheet (36 in. x 48 in.)	17.2	0	1	Home Depot
Hall Effect Sensors (2Pcs)	6.99	2	6	Amazon
Gears (3D Print)	11.99	1	2	3D HUBS
Sensor Brackets	3.65	1	6	3D HUBS
Super Glue	5.59	0	2	Home Depot
Double-Side Tape	5.97	0	1	Home Depot
Total Cost:	122.17			

Results and Discussion

The position sensing system was tested with the overall control program on the track. However, the first test shows that the accuracy of the position sensor was only 20%. The problem was found. The program took too long to loop; the speed of the program was not fast enough to detect all the peaks. The team commented out the code for the ultrasonic sensor; the accuracy increased dramatically to 80%. The result proved that the program was not fast enough to detect all the peaks. The control team moves the Hall Effect sensors and the ultrasonic sensors to the interrupt pin. However, it did not solve the problem.

The team successfully designed a device and manufactured the corrugated ferrous strips, created linear encoders with Hall Effect sensors and magnets, programmed the encoder to output position sensing signals and integrated and demonstrate the position sensing system. However, the accuracy did not improve. Because of the low accuracy, the position sensing system was not able to be implemented to the control system.

Conclusions and Suggestions for Future Work

In order to provide accurate position information of each podcar, the position sensing team proposed, designed and implemented a new position sensing system. The position sensing system consists of a position sensor which is attached to the track and corrugated ferrous strips which are attached to the whole track. The team designs and create a gear set to corrugated the metal strips. The position sensing system was programmed integrated to the control program with Arduino. The corrugated ferrous strips were successfully and efficiently manufactured with the gear set and the available sheet metal shear in the shop, and the installation of the corrugated metal strips was completed. The position sensors were created. The position sensing program was completed. However, the accuracy of the position system only reached 20%. As a result, the position sensing system was not able to be implemented.

Future improvement of the position sensing system must be made to create an accurate position sensing system.

The position sensor must be connected to interrupt pins on the controller instead of analog pins, and the corresponding program for the interrupt pin connections must be rewritten and tested.

The device to manufacture corrugated ferrous strips was not simple enough. Future improvements such as adding a handle and a track to hold the strips in place. However, since there will be only a few of the ferrous strips on the track needed to be replaced, the device will not frequently be used.

Several team members were cut and injury during assembly and disassembly because of the sharp edges of the corrugated ferrous strips. The edges of the corrugated ferrous should be processed for safety.

Solar Improvement

Background and context

Previous year’s Solar Team designed a working charging system to charge batteries to power pod cars. Examination of the design work revealed several flaws that will need be addressed.

The previous year’s racking system was constructed with wood. The overall racking design was sound except for the jointing piece that held both halves together. The joint failed when the exceeding unsupported weight of the racks cause the halves to split apart.

The charge box containing the charging components were also constructed with wood. The cover was lined with fabric with excess fabric folded and hot glued underneath. Holes were made on the cover to fit and hold components. 16-gauge wiring were used throughout the system. The ends of the wiring that connected components are stripped and plugged into terminal blocks which are narrow to support a 16-gauge wire. Strains of positively charged wiring could easily contact one and another.

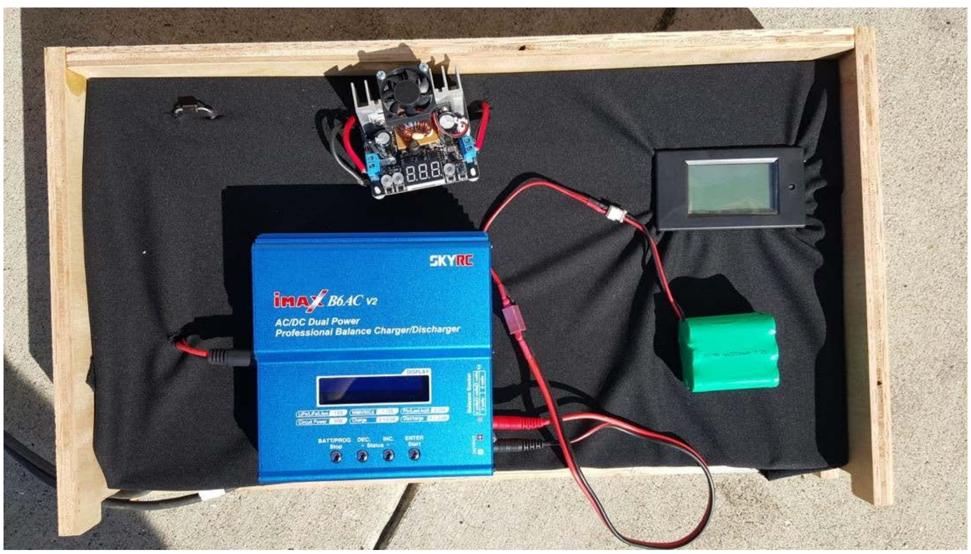


Figure 1-22. Previous year's charging box.

The placement of the rack above the track created a challenge for the wiring to connect to the solar panel and charge box together. Chairs were used to elevate the charge boxes to reach the solar panels.

Figure 1-23. Chairs used to elevate charge boxes.

Description of the Subteam and Objectives

Solar energy is the backbone of the Spartan Superway project. By choosing to be powered by solar energy will differentiate Spartan Superway from any other transportation system. Mass-transit systems are traditionally powered by electricity provided by local utilities which are often produced by burning fossil fuels or coal. Solar energy can provide renewable energy at a low cost to users while combating greenhouse gas emissions and reduce our dependence on fossil fuels.

The 1/12th Scale Solar Improvement team will demonstrate the functionality and future possibility of a full-scale Spartan Superway automated transportation network. The goal of the team is to provide solar energy to power pod cars and being able to charge batteries that power pod cars.

The main objective of the project is to refurbish the solar panels and associated components to ensure functionality while improving performance and aesthetics. The objective of the solar team within the 1/12th scale teams is to be able to provide energy to power mechanical bogies for demonstrations of sustainability of solar power.

A redesign and fabrication of the racking system for the solar panels will replace previous year's wooden structure. The design will be modular and easily assembled or disassembled with basic tools.

Charging components will be reorganized and rewired. A design for the mounting of components will accommodate accessibility, safety, and aesthetics. Research, design, calculation, and testing will be provided to the implementation of the capacity of the two solar panels to store energy in batteries that power mechanical bogies. Optimization of placement of the solar panels was explored for the best performance.

Design Requirements and Specifications

The main design requirement for the project is to provide energy to power the pod cars for the 1/12 scale team.

To optimize the energy output, the racking system must be able tilt to the optimal tilt angle to ensure maximum performance of the photovoltaics.

The project will be presented publicly which requires the entire system to be easily disassembled and reassemble multiple times without issues.

State-of-the-Art/Literature Review

The 3 main types of solar panels used are: monocrystalline, polycrystalline, and thin film. Each type has its advantages and disadvantages. In the case of monocrystalline, it has been the number one choice for many years due to its efficiency and dependable way of producing solar energy. The modules are made from a pure single silicon crystal. To accomplish this, a process like semiconductors is used. The typical efficiency of the monocrystalline solar panels is from 15-20%. Although these solar panels produce the highest power outputs and require the least amount of space, it is the most expensive of the three.

Figure 1-24. Monocrystalline silicon solar panel.

Polycrystalline solar panels are made of similar silicon material but instead of making a single crystal, they are melted into a mold. The result has random crystal boundaries. The process of making polycrystalline is much simpler and therefore less expensive. Also, the tolerance to heat of the polycrystalline solar panel is less than that of the monocrystalline solar panel. The typical efficiency of the polycrystalline solar panel ranges from 13-16%.

Figure 1-25. Polycrystalline solar panel.

The efficiency of thin film solar panels range from 7-13% but future modules are expected to increase to 10-16%.

Figure 1-26. Thin film solar panel.

Description of Your Design

The design for the solar racking system revolves around the physical dimensions of the solar panels.

Table 1-4. Dimensions of solar panel.

PHYSICAL CHARACTERISTICS

Solopower SFX1-i

Length	120.1 in / 3.05 m
Width	11.5 in / 0.292 m
Thickness	0.1 in / 2.0 mm
Weight	5.0 lbs / 2.27 kg
Roof Load From Module	0.53 lbs/ft² / 2.61 kg/m²

The racking system was designed to be 10 feet long to accommodate the length of the solar panels. The racking system foundation comprise of 2 rectangular box frames that jointed together in the middle by (2) M8 x 1.25 bolts and nuts. To form the rectangular box frames, aluminum bar stock measuring 1.5 inch by 0.25 inch were cut to form (4) 5 feet pieces. These 4 pieces form the sides of the box frame. Using the same aluminum bar stock, (10) 10.5 inch pieces were cut to be used as width of the box frames and mounting supports for the tilting mechanisms and charge boxes. Welding originally was proposed to be the method of joining these pieces together. After much thought, a simpler method would be to utilize corner brackets as all these pieces are perpendicular to each other.

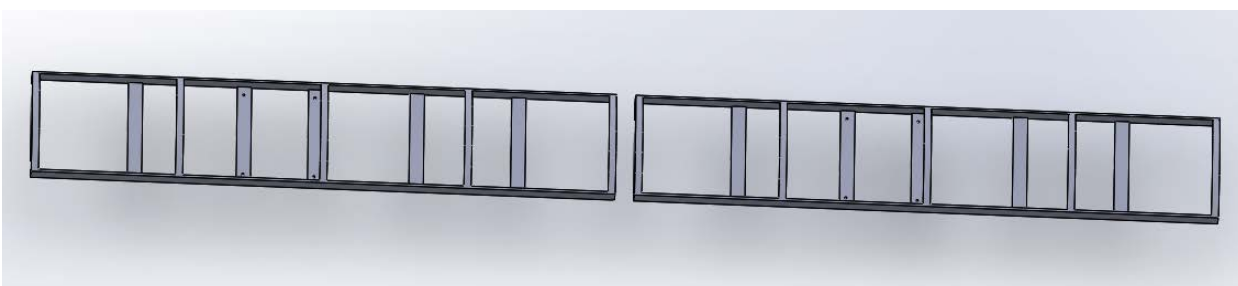


Figure 1-27. Rectangular box frames.

A design was required to from the curvature of the solar panels. The natural tendency of the solar panel is lay flat. To form a curvature, the solar panel must be forced into a narrower width. Several options were explored throughout this design process. The first idea was to create a channel with the width slightly narrower than 11.5 inches. The problem with this design is the panel possibly falling out when tilted or becoming misaligned easily.

Dr. Burford Furman suggested a design concept like that of the ribs of a human body. The solar panel would conform to the shape of the rib. With this idea, several tests were conducted to determine the optimal curvature. After testing and measuring the curvature of the panel while forcing it into several different lengths, the design dimensions for the ribs are:

Table 1-5. Dimensions of rib.

Arc Length	11.5 inches
Arc Radius	24 inches
Chord Length	11.4 inches
Length of Sides	1.5 inches

10 ribs are placed across equally across the rack's length of 10 feet. Initially, welding was proposed to attach the ribs to the box frame. Extending the sides 1.5 inch to sit flush with the sides of the frames then bolting the rib to the frame was a simpler solution. Holes were placed on ribs and along the sides of the box frame. The material used for the ribs are aluminum bar stock leftover from the Track Improvement team, measured to be 0.125 inch by 0.750 inch.

Figure 1-28. Rib.

With the solar panel being placed above these ribs, a design to force the solar panel to confirm to the curvature was required. Clamps were designed to be placed on both ends of the ribs. The area in contact with the solar panel could not impede the functionality of the solar panel. The border between the photovoltaic cells and the edge of the solar panel was measured to be 0.5 inch. The clamp design used this area to apply pressure to the solar panel. Like the ribs, the sides will sit flush against the sides of rib underneath and utilize a hole to bolt the 3 components (rib, clamp, and box) together. A slot was designed to allow adjustment from fabrication inconsistencies and tolerances.

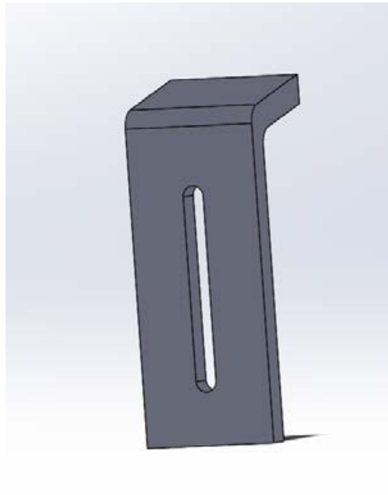


Figure 1-29. Clamp.

The tilt mechanism from last was carried over from previous year's design. The 2-bar linkage allows tilting of 0^{o} to 90^{o}. The support structure that attaches to the support poles of the track was also carried over.

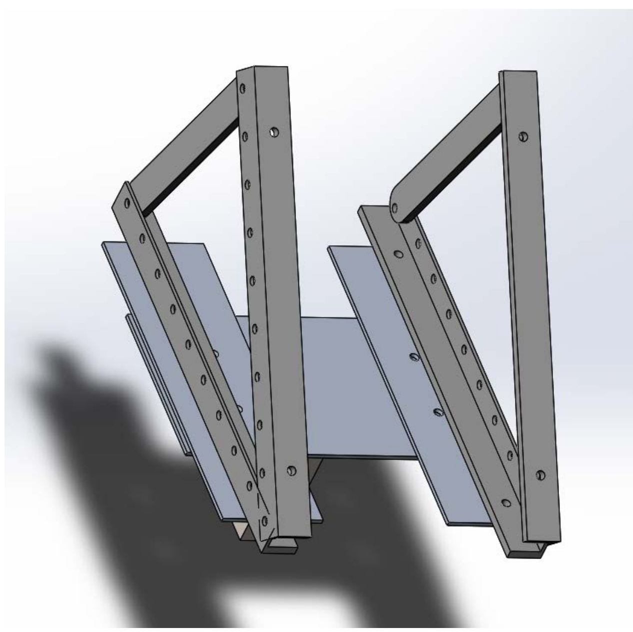


Figure 1-30. 2-bar linkage for tilt.

A charge box of 13 inches by 11 inches by 2.5 inches was designed to contain all the charging components. A hinge was placed on the top of the box. The components are mounted on the top panel of the box. This allows box to be accessible regardless of the orientation of the racking system.

Figure 1-31. Components mounted to top panel of charge box.

The charging components of the system was carried over from previous year's design. The wiring between components were reorganized. The ends of each wires were cut, stripped, and spade connectors were crimped on. Using spade connectors, connections were more secure and safe. The lengths of each wire were shortened or lengthened to accommodate the new charge box.

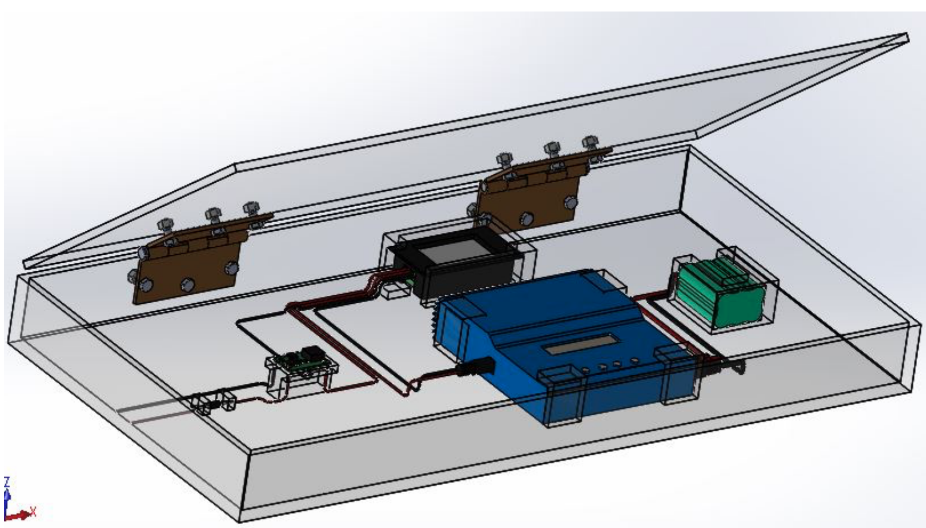


Figure 1-32. Organization of battery box.

The electrical components from previous year was carried over. The solar panels used are SoloPower SoloPanel SFX-i70. A voltage regulator is used to drop the 24 volts produced by the solar panel to 18 volts which is the maximum voltage the charge controller can accept. A digital multi-meter is placed inline of the circuit to display measurements. A diode placed on positive wire from the solar panel to the voltage regulator prevents current traveling the opposite direction.

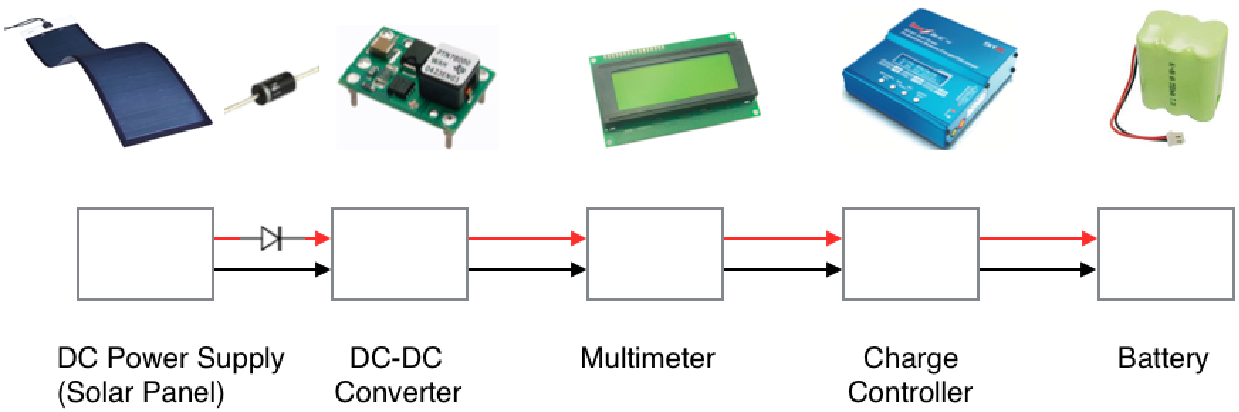


Figure 1-33. Diagram of charging components.

The batteries from last year were 6-cells 7.2 volts and 2500 mAh. At the request of the Vehicles Control team, 2 of the 6 cells were removed to lower the voltage. The conversion of the batteries required each battery pack to be unwrapped, have tabs cut, re-soldered, rewrapped, and new connectors soldered on. (6) new batteries (5-cell 6 volt 2500 mAh AA NiMH) were purchased in additional to having (7) modified 4-cell 4.8 volt 2200 mAh AA NiMH batteries.

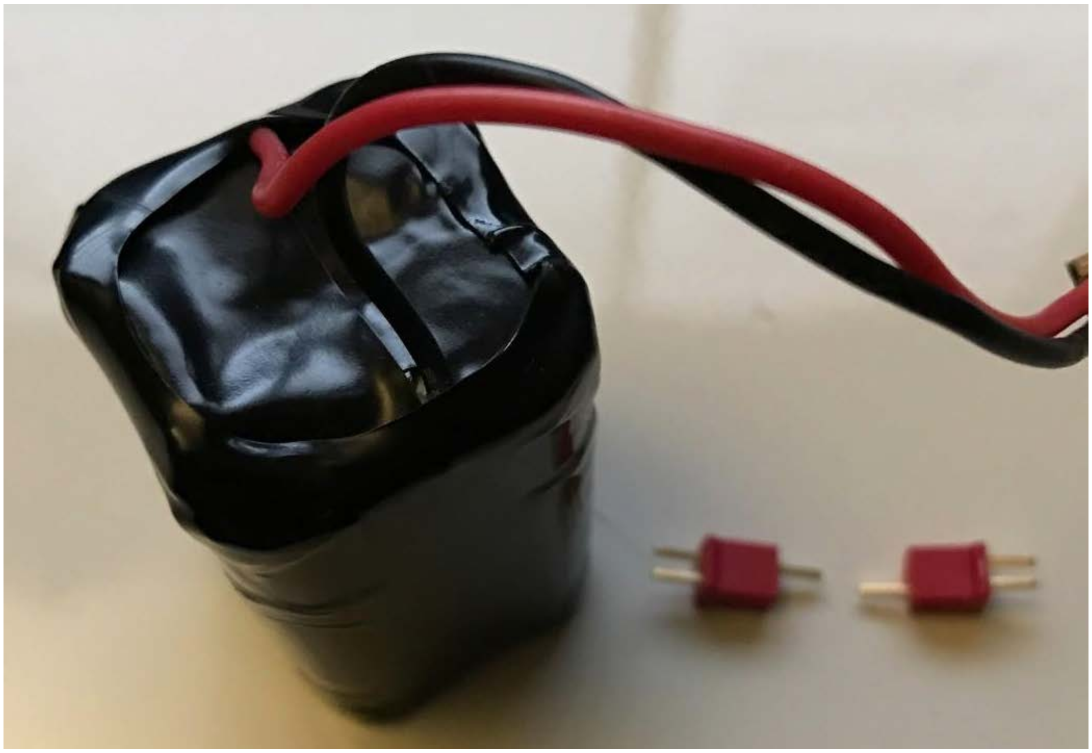


Figure 1-34. Modified 4.8 volt 4 cell batteries.

Analysis/Validation/Testing

The distributed load from the weight of the solar panel is ignored due to the solar panel weighing 5 lbs. The design placed the support poles in the center line of the rack to reduce any moments about the support poles. Analyzing along the length of the rack, the largest moment would be produced at the middle where both halves of the rack met. To ensure structure stability, the sections where the halves met were fabricated so that the two surfaces would sit flush against each other.

Tests were conducted to determine the efficiency of the system to charge batteries. On days with unobstructed sunlight, several discharge and recharge cycles were conducted with the charge controller. The 4-cell 6V batteries took the average of 55 minutes to charge from empty to full.

Figure 1-35. Time to fully charge 4 volt batteries.

New batteries were requested by the Vehicle Controls subteam. After extended testing during Paseo Public Prototyping Fair, it was reported that the 4 volt batteries' run time on the pod cars were approximately 20 minutes. Numerous discussions led to the assumption of the 4 volt batteries being stored for an extended amount of time without any proper care or maintenance. A proposal for new batteries were made to procure newer batteries. A budget of roughly $100 for batteries was approved. Chris Hansen (from the Vehicle Controls subteam) and I explored and considered several options which includes purchasing Lithium-ion batteries along with voltage regulators or larger capacity AA Nickel-metal hydride batteries. Voltage regulators would be required for each pod car bogie as the motors specification sheet called for a maximum input voltage of 6 volts. The additional cost of voltage regulators excluded the option of Lithium-ion batteries. A 5 cell 2500 mAh AA 6 V Nickel-metal hydride was determined to be most cost efficient option for purchasing.

Figure 1-36. New 6 volt 5 cell batteries.

“Curved Photovoltaic Surface Optimization for BIPV: An Evolutionary Approach Based on Solar Radiation Simulation” by Sheng Cheng was used to analyze the efficiency of curved and tilted solar panels. Research into the subject matter also did not provide evidence to support the theory of improved efficiency of curved panels. Tests were conducted to come to a conclusive answer to but there were too many uncontrollable variables while testing. Power, voltage, and amperages were constantly changing.

Tilting a solar panel to south-southeast towards the equator to optimize solar radiation was supported several articles including the dissertation mentioned previously. Using radiation data provided by the National Aeronautics and Space Administration’s (NASA) Atmospheric Science Data Center, the panels can be tilted to maximize the solar radiation incident on the panel. Theoretical energy output from this data can be calculated.

Table 1-6. Monthly optimal tilt angle and theoretical energy output.

Parameters for Tilted Solar Panels:

Maximum Radiation Incident On An Equator-pointed Tilted Surface ($kWh/m^2/day$)

Lat 37.339 Lon -131.889	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Annual Average
SSE MAX	2.39	3.61	4.65	5.84	6.51	6.73	6.64	6.15	5.28	4.26	2.72	2.11	4.74
K	0.50	0.58	0.57	0.59	0.58	0.58	0.58	0.59	0.60	0.63	0.53	0.49	0.57
Diffuse	0.85	1.01	1.42	1.79	2.11	2.26	2.16	1.87	1.46	1.01	0.89	0.77	1.47
Direct	3.89	5.21	5.54	6.10	6.30	6.38	6.35	6.27	6.13	6.01	4.32	3.60	5.51
Tilt 0	2.36	3.50	4.60	5.81	6.47	6.68	6.59	6.12	5.20	4.23	2.68	2.10	4.70
Tilt 22	3.33	4.57	5.37	6.14	6.40	6.43	6.42	6.28	5.87	5.43	3.71	3.04	5.25
Tilt 37	3.77	5.00	5.55	5.97	5.96	5.87	5.91	5.99	5.94	5.89	4.17	3.48	5.29
Tilt 52	4.00	5.14	5.43	5.50	5.22	5.03	5.11	5.40	5.70	6.00	4.39	3.73	5.05
Tilt 90	3.57	4.21	3.89	3.28	2.70	2.46	2.54	3.01	3.80	4.79	3.84	3.40	3.45
OPT	4.03	5.14	5.55	6.14	6.53	6.68	6.62	6.31	5.95	6.01	4.41	3.78	5.60
OPT ANG	60.0	52.0	38.0	22.0	9.00	3.00	6.00	16.0	33.0	50.0	58.0	62.0	33.9

NOTE:	*Diffuse radiation, direct normal radiation and tilted surface radiation are not calculated when the clearness index (K) is below 0.3 or above 0.8.*

Solar Cell Dimensions (meters)		Average Daily Available Solar Energy (kWh/m^2/day)		Advertised Efficiency	Average Energy Output (kWh/day)	
Length	2.85	Jan	4.03	0.095	Jan	0.295
Width	0.27	Feb	5.14		Feb	0.376
		Mar	5.55		Mar	0.406
		Apr	6.14		Apr	0.449
Total Area (m^2)	0.77	May	6.53		May	0.478

Procedure / Instruction Manual

The entire system is comprised of three parts:

1. Right half of the rack (contains the charging box)
2. Left half of the rack
3. Solar panel

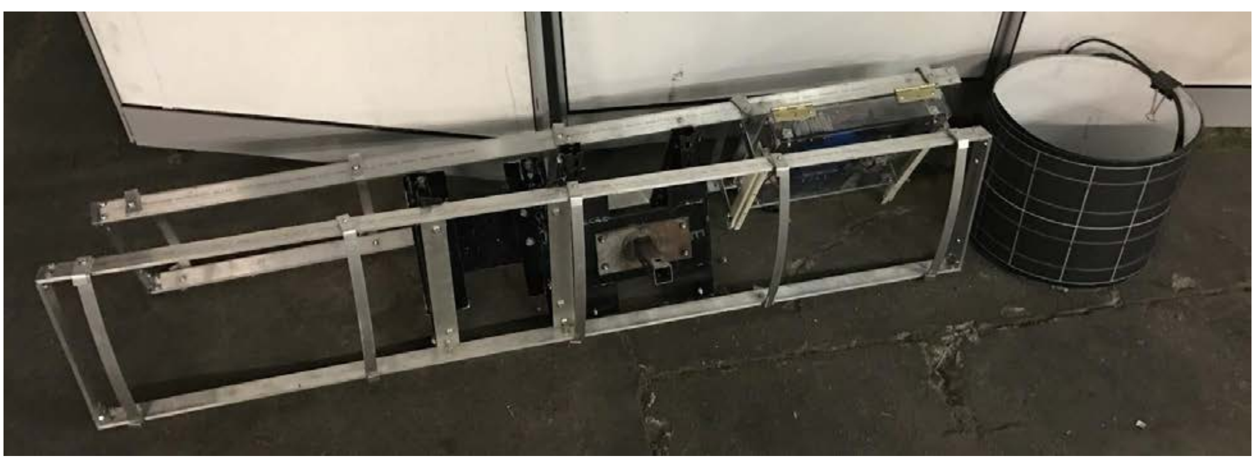


Figure 1-37. Components of system.

To Assemble:

1. Slide the right half of the rack's support beam over on the support pole above the track. The side with the holes to connect the halves together should be facing the center.
2. Slide the left half of the rack's support beam over on the support pole above the track. The side with the holes to connect the halves together should be facing the center.
3. Using the (2) M8x1.25 bolts and nuts fasten and secure the two halves of the rack together.
4. Unroll the solar panel and slide the panel across the rack underneath each clamp. The end with connectors should be at end with the charge box.
5. Pushing down on clamps and ribs, tighten down the screws with an impact driver. Using a screwdriver will require a wrench to hold the nut underneath.
6. Connect the solar panel to the charge box.

To Disassemble:

1. Ensure the charging box is locked with the security bolt.
2. Disconnect the connection between the solar panel and charge box.
3. Loosen the clamps by untighten the screws.
4. Slide the solar panel off the racking system and roll the solar panel up for transportation. A large binder clip will keep the solar panel rolled up.
5. Ensure there are enough threads to keep the nut from falling off the screw.
6. Unbolt the (2) M8x1.25 bolts and nuts. Secure the bolts and nuts onto one side of the rack to prevent losing them.
7. Pull both halves of the rack off the support pole.

To charge battery:

1. Unlock the security bolt from the charge box and slide box out.
2. Connect the battery to be charged.
3. Select battery chemistry. Batteries are NiMH.
4. Select mode of operation and select amperage.
5. Hold the start button to begin charge.

Money spent on your project

Aluminum bar stock was sourced from the Track Improvement subteam which significantly reduced the total cost of the project. The major cost for the project were the new batteries.

Table 1-7. Budget spent on project.

Quantity	Item	Unit Price	Subtotal
5/1/2017 Maxxpacks.com			
6	#2505H 6V NiMH AA Pack	$16.00	$96
4/27/17 Lowes			
1	6 ft. x 1.25 in. Plated Steel Perforated Angle	$14.98	$14.98
3/1/17 Tower Hobbies			

11	1222 Micro 2R Plug	$1.49	$16.39
12/2/16 TAP Plastics			
1	Cut to Size	$18.35	$18.35
4/29/17 TAP Plastics			
1	Cut to Size	$20.32	$20.32
1	Cut to Size	$4.06	$4.06
1	Cut to Size	$3.12	$3.12
3/5/17 Fry's			
1	1 1/2" Black Heat Shrink, 4 ft.	$9.99	$9.99
4/19/17 Home Depot			
2	Machine Screw #8-32x1/2"	$1.18	$2.36
9	Brace, Corner 3/4" Zinc 4pk	$1.97	$17.73
2	Brace, Corner 4" Zinc 2pk	$2.98	$5.96
1	Machine Screw Nut 8/32	$3.92	$3.92
1	Metric Flange Nut M8-1.25	$0.62	$0.62
1	Metric Flange Bolt M8-1.25X20mm	$1.28	$1.28
1	SAE Washer 8	$4.24	$4.24
1	Machine Screw Round Head #8-32x1"	$5.60	$5.60
4	Hex Bolt 1/4x1	$0.12	$0.48
8	Cut Washers 1/4 in.	$0.11	$0.88
4	Hex Nuts 3/8x2	$0.06	$0.24
2	Hex Bolt 3/8x2	$0.34	$0.68
4	Cut washers 3/8 in.	$0.14	$0.56
2	Hex Nuts -USS 3/8	$0.12	$0.24
4/2/17 Home Depot			
1	12" Self-Closing Drawer Slide 1pair	$5.24	$5.24
2	Machine Screw #8-32x1/2"	$1.18	$2.36
2	Brace, Corner 1.5" Zinc 4pk	$2.80	$5.60
4/2/17 Home Depot			
3	Machine Screw Round Head #8-32x1"	$1.18	$3.54
3	Machine Screw Round Head #8-32x3/4"	$1.18	$3.54
4/5/17 Home Depot			
1	Brace, Corner 4" Zinc 2pk	$2.98	$2.98
1	Hinge, UTL NOREM NRRW 3" SB 2pk	$3.43	$3.43
3/28/17 Home Depot			
7	Brace, Corner 3/4" Zinc 4pk	$1.97	$13.79
1	Machine Screw #8-32x1/2"	$1.18	$1.18
2	#8 Zinc Washers	$1.18	$2.36
3/27/17 Home Depot			
5	Brace, Corner 3/4" Zinc 4pk	$1.97	$9.85
1	Machine Screw Nut 8/32	$3.92	$3.92
1	#10 Zinc Washers	$1.18	$1.18
	Total including taxes and shipping		$305.84

Results and Discussion

The results of this project improved the functionality and aesthetics of the solar system of the 1/12th scale model. The new racking system choice of material matches the track below created a sense of cohesion between the track and racking system.

Assembly and disassembly was reported by team mates to be simple and intuitive as I could not be present at every setup and disassembly for each event. The operation of the charging function was also commented by team members to be easy.

The solar charging portion of the system were a main attraction for event attendees. Questions regarding the source of power for the project were commonly brought up.

Conclusions and Suggestions for Future Work

Currently, the solar energy harvested by the system is insufficient in sustaining the rate of discharge of the batteries depending on the amount of pod cars on the track. The 4 cell batteries last for approximately 20 minutes during testing before the motor speeds are reduced to an unacceptable rate. From the charge and discharge test sessions, the 4 cell batteries and require approximately 55 minutes to be recharged from a complete discharge. To sustain the rate of usage, the time it takes to charge batteries will need to be shorter than the runtime of the batteries.

Future work should include exploring ways to charge multiple batteries with each panel at the same time. A new charge controller should accommodate this feature as much of the power produced by the solar panel is converted into wasted heat via the several voltage reductions, 24 volts to 18 volts and 18 volts down to 6 volts of the batteries.

The suggestion of using a larger size battery as storage by Ron Swenson would be an idea worth considering. The idea is to place a large capacity battery in line with the solar panel and have a charge controller operate this function. The smaller batteries would be feed off the larger battery.

Track Fabrication and Process Development

Background and Context

The shortcomings of the track fabrication from the 2015-2016 year prompted the creation of the Track Fabrication and Process Development subteam for the 2016-2017 year. Prominent issues from the preceding year featured imprecise in-house fabrication, poor design choices with lack of foresight, and widespread use of ramshackle stopgap solutions, with an additional highlight on the cost of outsourced fabrication work. Some solutions were reached through work performed by the Track Fabrication and Process Development subteam, while other solutions were found through build recommendations made to the Track Improvement subteam.

The first of the issues was that sections of track did not have a proportionally similar radius, meaning that distance between tracks was not uniform and create problems for the bogie attempting to traverse said sections (see Figure 1-38). The second issue was that mounting brackets for the track were not made well, resulting in the use of spacers as a means to shore up imperfections, and threaded aluminum with a limited lifespan as a means of permanent affixture (see Figure 1-39). The third issue was that alignment and smooth connection of track sections was often an issue (see Figure 1-40).

Figure 1-38. Disproportionate bends. Closeup highlights inconsistency plaguing previous year's build.

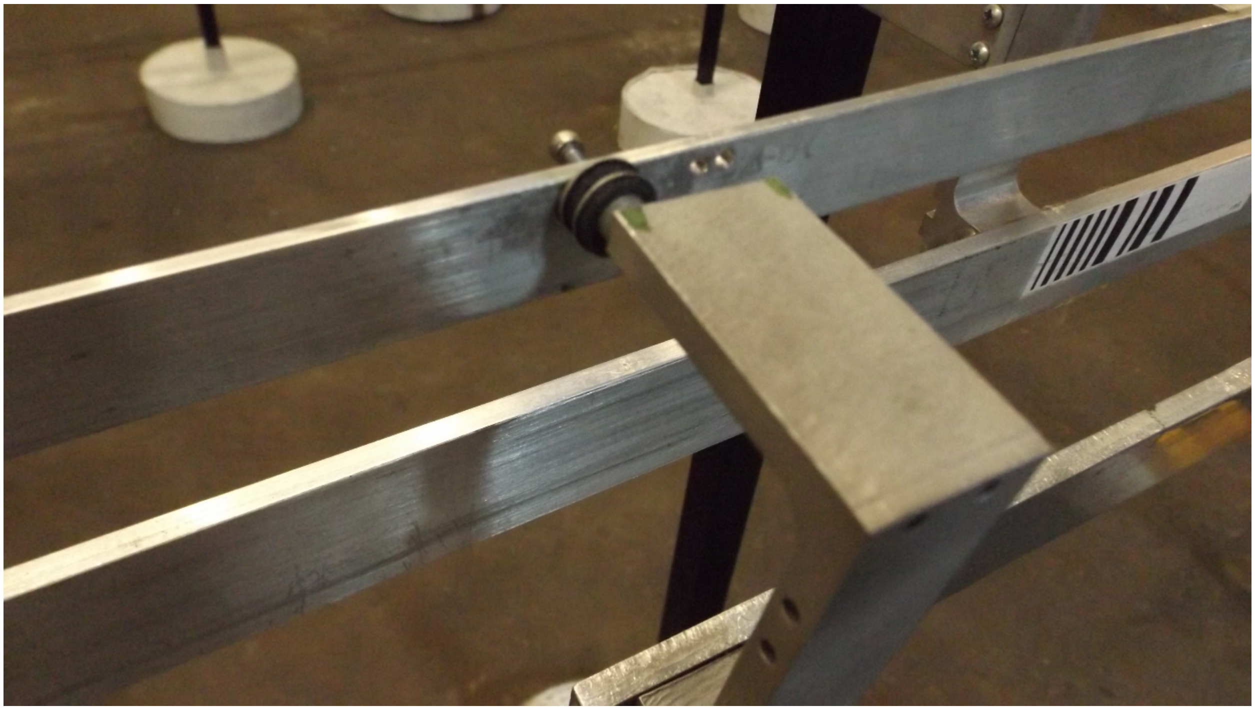


Figure 1-39. Build imperfections. Close-up highlights issue plaguing previous year's build.

Figure 1-40. Track misalignment. Close-up highlights issue plaguing previous year's build.

Description of the Subteam and Objectives

The Small-Scale Track Fabrication and Process Development subteam was formed for the first time for the 2016-2017 year to investigate the viability of in-house fabrication and to aid in additional fabrication where possible. The objectives of the subteam were to:

1. Reduce monetary and temporal costs related to outsourced fabrication.
2. Design and create a bending mechanism to allow for fabrication to desired specifications.
3. Achieve reliable, repeatable bends on flat aluminum bar stock.

An additional objective was added during the course of the year for the subteam to address when possible:

1. Aid in fabrication requests from other small-scale subteams.

Design Requirements and Specifications

The Small-Scale Track Fabrication and Process Development subteam, responsible for fulfilling the fabrication requirements of other small-scale subteams, required those aforementioned subteams to determine their build requirements before the Small-Scale Track Fabrication and Process Development subteam could proceed with any fabrication. These fabrication requests manifested as two separate builds; fabrication of corner sections of track for the Track Improvement subteam and production and assembly of solar panel rack sections to aid the Solar subteam.

The Track Improvement subteam required small size aluminum bar stock to be bent into arcs of varying radii and angle. The final set of build requirements were for two corner sections of track to be fabricated to specifications as follows; 0.25-inch by 1.50-inch 6061 aluminum bar stock bent into a 90-degree arc with a radius of 19.69-inches, and 0.125-inch by 0.750-inch 6061 aluminum bar stock bent into a 90-degree arc with a radius of 20.87-inches. Additionally, each corner section was to have a 9.84-inch length of straight section on either end of the bend. Figure 1-41 below is the bar stock bent to specifications and prepared for handoff to the Track Improvement subteam for assembly.

Figure 1-41. Stock bent to specification. Measurement made against mater die to confirm dimensions.

The Solar subteam required small size aluminum bar stock to be bent into tabs and ribs to directly support a solar panel, and requested additional aid for a new solar panel post support and fabrication of one of the two solar panel racks. Tabs and ribs required 1.5-inch lengths to provide space for through-holes for #10 machine screws for mounting, and featured sharp bends to allow for curvature of the solar panel. The new support panel post support was modeled after the existing solar panel post supports, utilizing a rectangle of 0.125-inch thick aluminum plate 9.75 inches by 10.0 inches, a 5.25-inch length of 1.25-inch square aluminum tube with 0.125-inch wall with sections of 1-inch angle stock for mounting, and 1-inch angle stock for support rails to connect the post support to the rack. See Figure 1-42 through Figure 1-46 for coverage of the various builds for the solar panel rack. More explicit details pertaining to the build not found here are available in the final report by the Solar subteam.

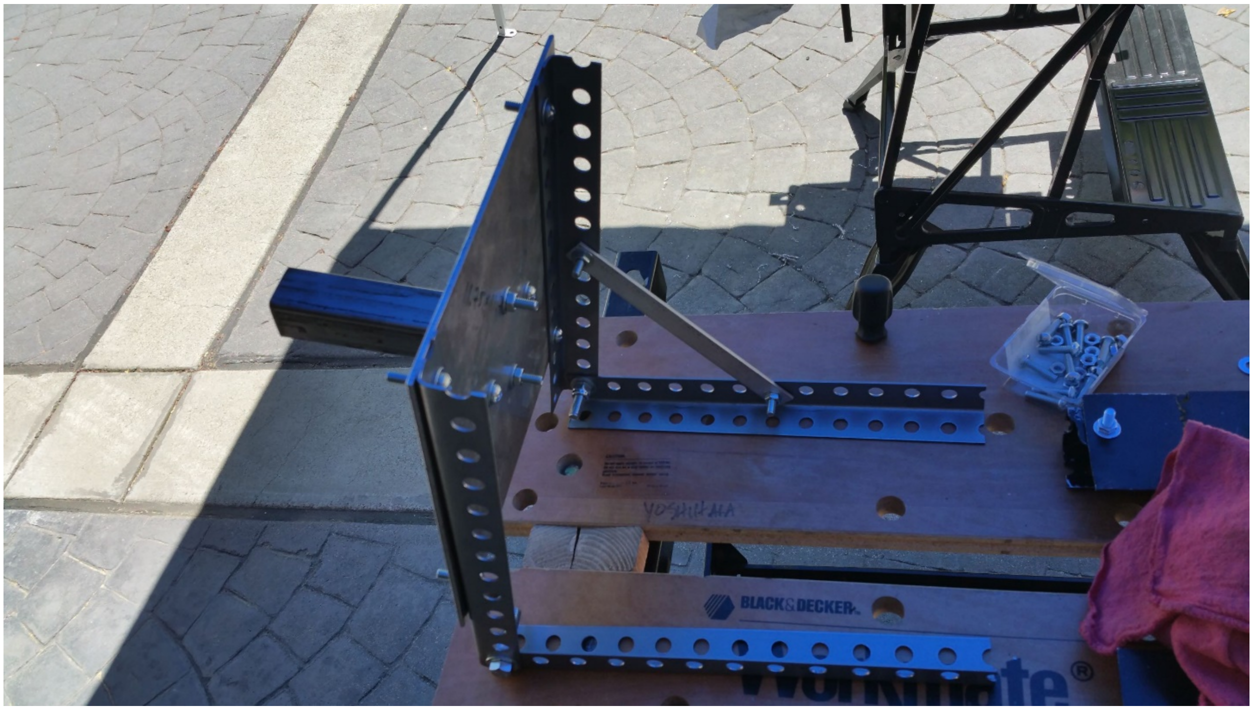


Figure 1-42. Solar panel rack and post support. Modeled after existing supports.

Figure 1-43. Solar panel ribs. Production before handoff to Solar subteam.

Figure 1-44. Solar panel tabs. Production before handoff to Solar subteam.

Figure 1-45. Solar panel rack assembly. State prior to assembly.

Figure 1-46. Solar panel rack test fit. State after assembly showed solar panel fit.

State-of-the-Art / Literature Review

Aside from fabrication available by common and normal methods, the Small-Scale Track Fabrication and Process Development subteam was required to obtain the capability to reliably and repeatedly bend small-size aluminum bar stock. To achieve said capability, investigation into common methods and machines available to the average consumer were researched. Research results pointed to two common benders; Hossfeld and DiAcro, which were highlighted previously in the executive summary.

Research initially proved promising; existing machines that could be purchased by the common consumer could be used to meet our requirements. Additional research, however, found that the minimum buy-in cost for either bender started at $1000, with an average additional cost of $300 more for any common die, with custom-sized dies averaging an even higher price. Furthermore, the arcs required by the Track Improvement subteam had large angles of 90 degrees and radii of 19.69-inches and 20.87-inches, and with both benders only supporting the ability to bend small sections several inches in length of such large arcs, reliability and repeatability could prove problematic. The decision was made to identify key elements of the Hossfeld and DiAcro benders, and to utilize those key elements to aid in the creation of our own bender.

Description of The Design

The bender that the Track Fabrication and Process Development subteam created incorporated external design requirements and additional internal design considerations. The Track Improvement subteam developed their track designs (mentioned in the Design Requirements and Specifications earlier in this report), which mandated certain sizing requirements. Internal considerations were to allow for many dies to be created and fit to the bender to allow for modularity and account for fluid requirements from the Track Improvement subteam. The following information will explain how these details manifested themselves in the bender design.

Several key steps were made during the process of the bender design to incorporate both requirements and considerations. It began with clarification of goals; what needed to be created, and what modularity could be integrated to accommodate fluid requirements. Next was to investigate viability; what products or methods were available to achieve those creations, and would need to be done to create a product or method ourselves if necessary. What followed was to generate our design, then validate the design through creation and testing.

As aforementioned, our bender needed to bend small-size aluminum bar stock into large arcs of different radii. Investigation into commercially available methods allowed for identification of key elements of our bender. To achieve the necessary capability, a pin and roller located on an arm rotating about a center point of the die would draw material around the die, while a material hold block would prevent one end of the material from moving while the bend was being formed, all of which sat atop an old lathe stand that provided a heavy support that would resist movement. Material springback, the degree to which a material resists permanent elastic deformation, was an extra consideration for bender sizing, but was addressed with a series of calculations. Refer to Appendix A4-1 for the die sizing chart generated for the bender. Note that only die sizes #3 and #6 were utilized.

With regards to the tabs and ribs of the solar panel rack support, a block of aluminum was cut into a V-shape to subsequently be used in a vise to create sharp bends in the aluminum bar stock. The remaining details of the solar panel rack design can be found in the Solar subteam's final report. Figure 1-43 and Figure 1-44 show the ribs and tabs that were created, while Figure 1-47 shows the solar panel rack post support in more detail. The red arrow points to aluminum plate used as a base. The orange arrow points to the tilt arm, which allowed the solar panel to be angled in one direction as required. The green arrow points to the rib that connects the support to the rack. The blue arrow points to the post sleeve, a tube that is bolted to the aluminum base plate with sections of angle brackets acting as L-brackets.

Figure 1-48 highlights the various parts of the bender. The green arrow points to the material hold block, used for holding the material in place at one side of the die. The orange arrow points to the die that the material is bent around. The blue arrow points to the arm of the bender that rotates about a pin located at the center of the die, utilized to move the roller. The red arrow points to the roller and pin, used to draw material around the curvature of the die.

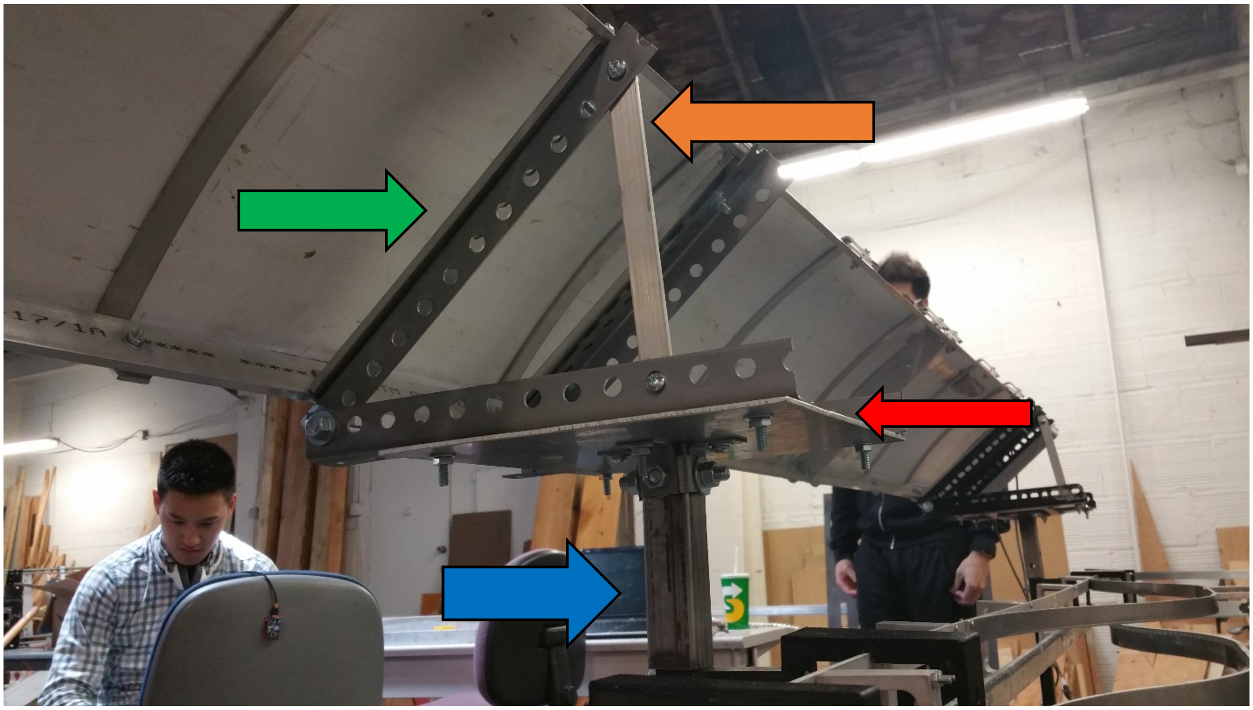


Figure 1-47. Solar panel rack support reference. Arrows highlight points of interest.

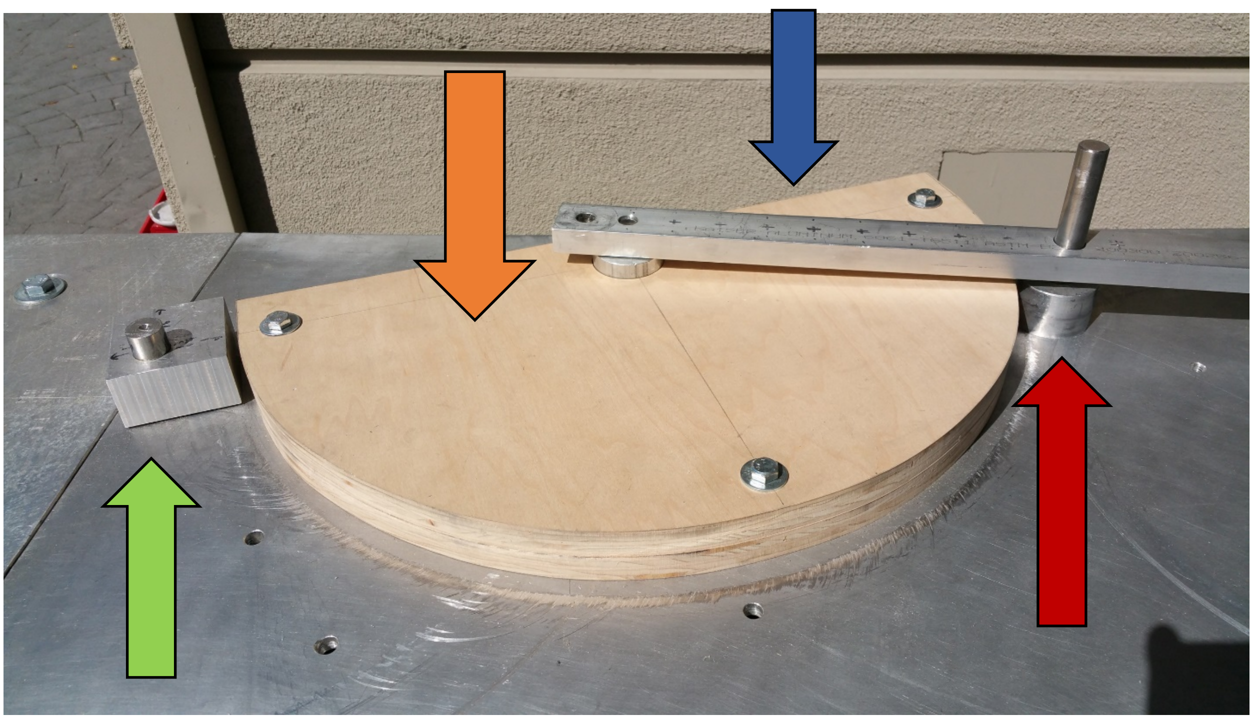


Figure 1-48. Material bender reference. Arrows highlight points of interest.

Analysis / Validation / Testing

The bender was validated through results-based analysis, where actualized results were matched to theoretical results that guided the initial design. Each bend was compared against a hand-crafted master, with each master created to the design specifications of the Track Improvement subteam. Refer to the Design Requirements and Specifications part of this report for more information relating to design specifications. Figure 1-49 below showcases a sample bend and comparison.

Figure 1-49. Bent material compared to master. All materials compared to masters before handoff.

The solar panel rack was validated through results-based analysis, where actualized results were matched to theoretical results that guided initial design. The solar panel rack was placed on top of the support posts for fit, and a solar panel was also mounted to verify the fit and quality of the production. Refer to the Design Requirements and Specifications part of this report for more information relating to design specifications, and to the final report by the Solar subteam if further details are sought after. Additionally, refer to Figure 1-46 for an image of a sample test fit for a solar panel on the rack, and Figure 1-50 below for a sample test fit for the rack onto the track supports (supports highlighted with red arrows).

Figure 1-50. Solar panel rack on track. Test fit performed in-shop after fabrication.

Procedure / Instruction Manual

Use of the material bender requires several simple steps before and during use to ensure uniform results are achieved:

1. Select and secure die of choice to correct location (Appendix A4-2 for hole and bolt locations).
2. Properly locate the bender arm, pin, and roller (Figure 1-51).
3. Insert the material to the desired location between material hold block and die (Figure 1-52).
4. Pull on the bender arm while ensuring that the material remains flat (Figure 1-53).

5. Ensure the material remains flat against the die while rolling the arm back.
6. Perform the bend a second time, starting at the opposite end of the bend (Figure 1-54).
7. Compare the material to your required dimensions.

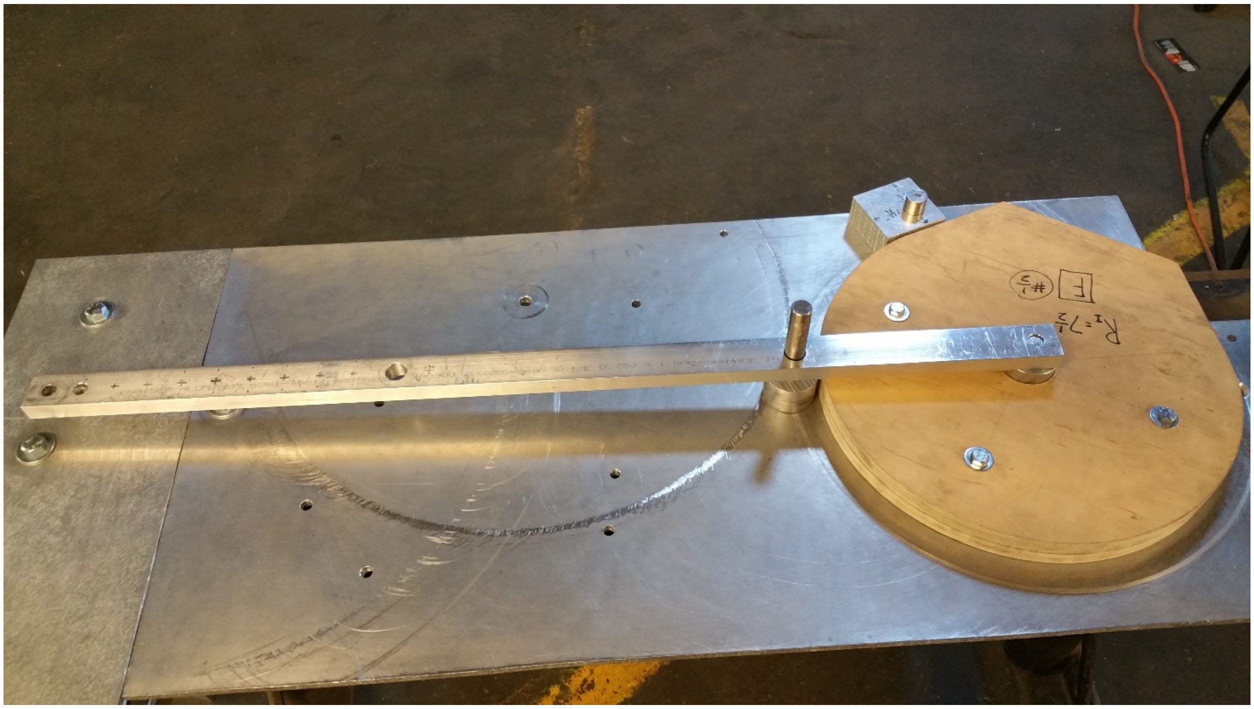


Figure 1-51. Bender setup. Die, roller, material hold block, arm are all in place.

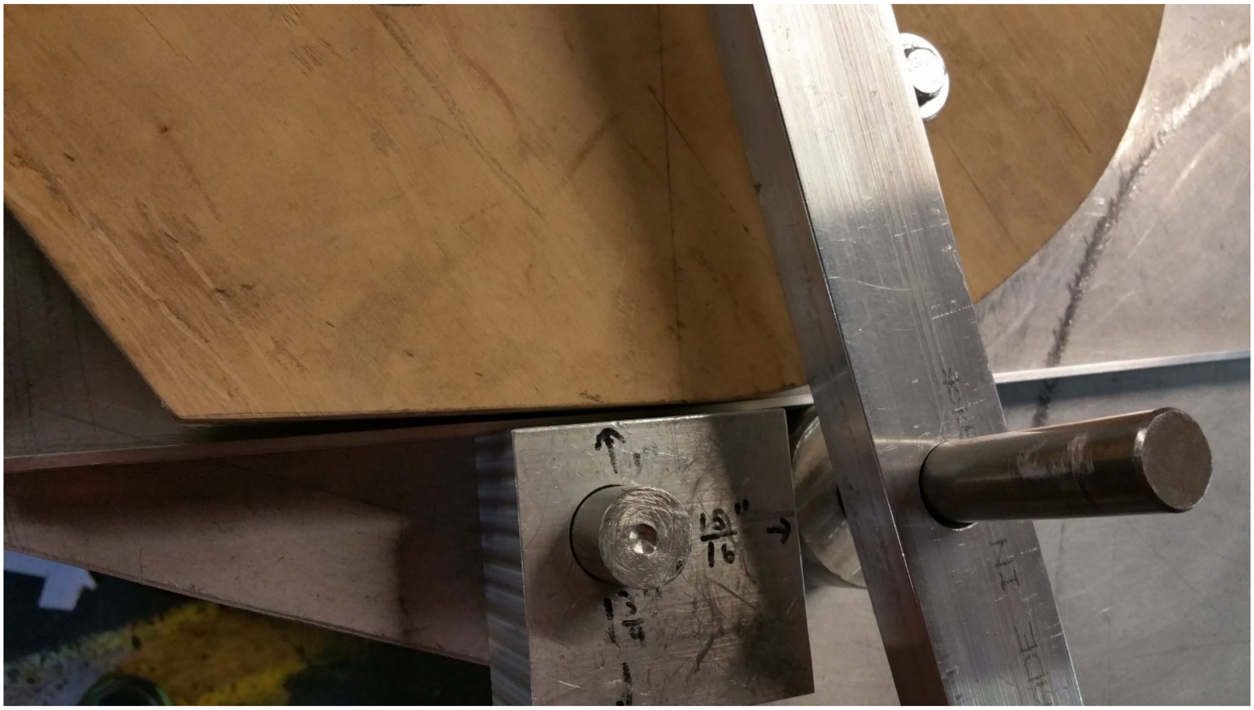


Figure 1-52. Material placement. Mark on material indicates beginning of curve.

Figure 1-53. Bender mid-bend. Roller is pressing material around die into desired shape.

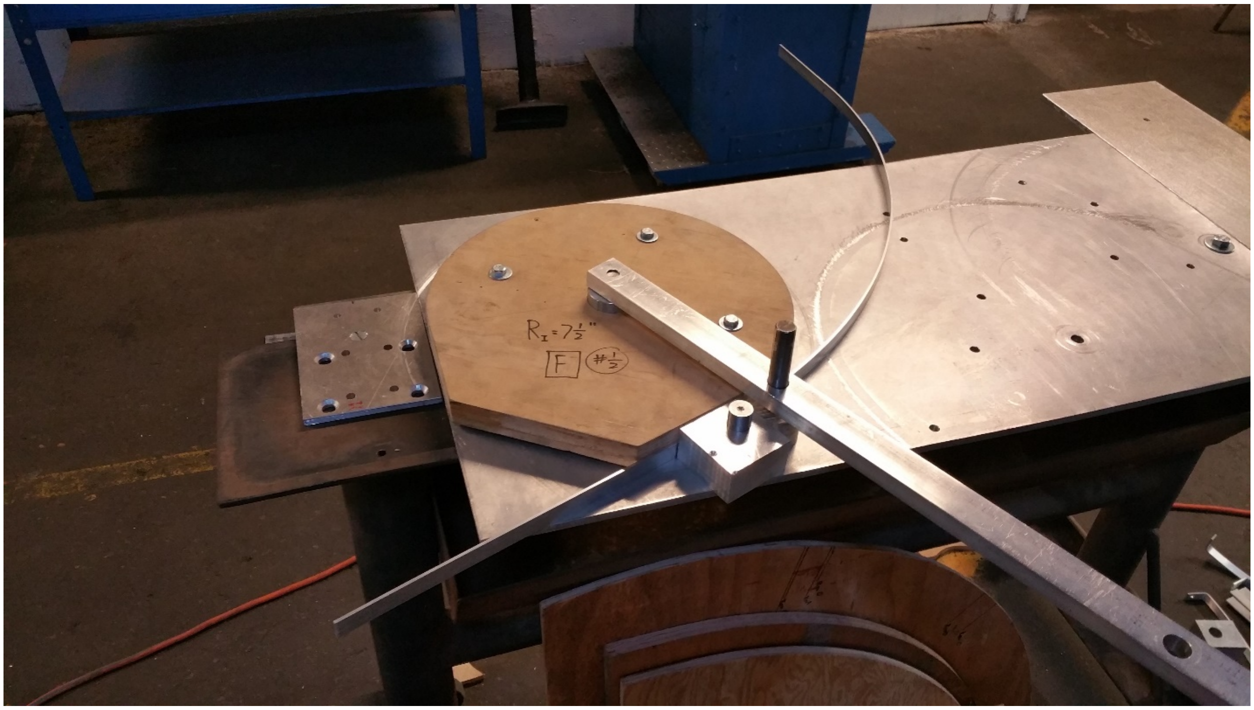


Figure 1-54. Bend from reverse end. Necessary step to ensure uniform bending.

In order to create the sharp bends required on the solar panel rack and tabs, the aluminum V-block should be utilized in a vise as follows:

1. Determine the point at which the bend is wanted.
2. Ensure both halves of the V-block are flat against the jaws of the vise (Figure 1-55).
3. Place the bend point at the V and tighten the vise until the material is held.
4. Tighten the vise until the wanted bend is achieved (~4 revolutions) (Figure 1-56).
5. Loosen the vise, remove the material, and verify the bend (Figure 1-57).

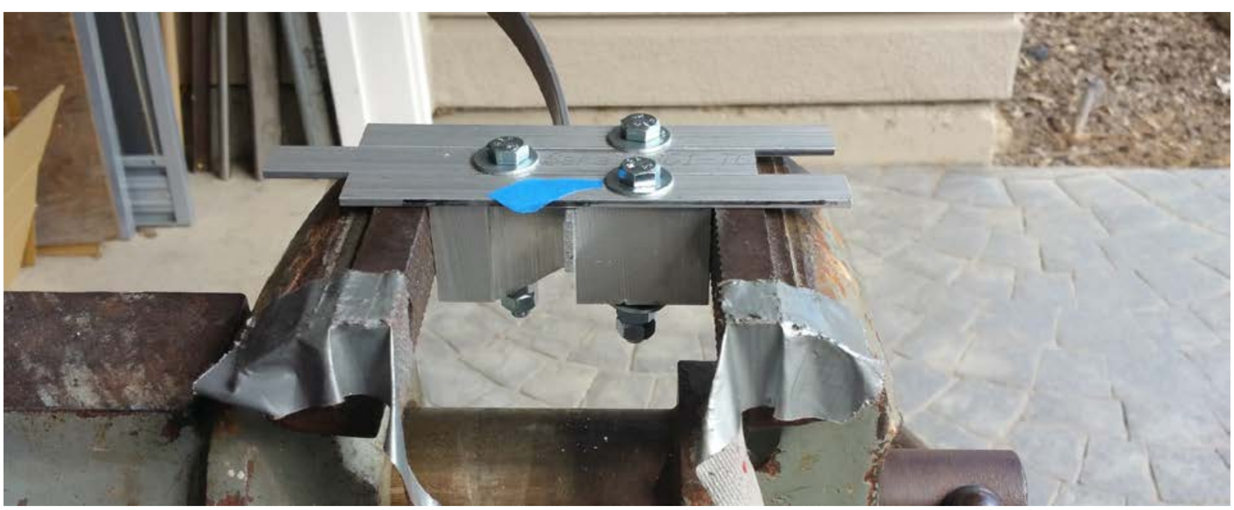


Figure 1-55. Rib in V-block. Tighten vise enough to verify material bend location.

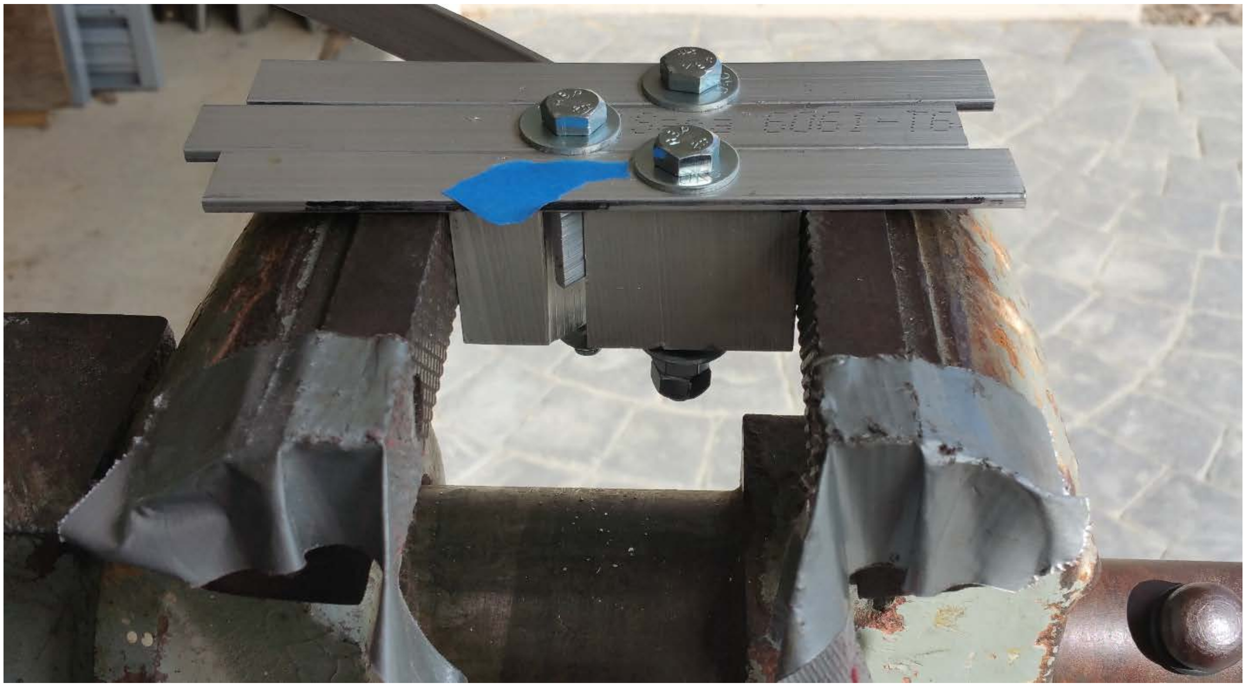


Figure 1-56. Material bent in V-block. Vise tightened until material was bent enough.

Figure 1-57. Sharp bend checked. Necessary angle achieved without tab cracking and falling off.

Money Spent on the Project

The monetary cost of the Track Fabrication and Process Development subteam to the Spartan Superway project was zero dollars. Materials were obtained and utilized with no attached cost, mainly as leftovers, spare parts, or scraps from previous years' project builds, and minimally from Craigslist and donations to the project. Refer to the chart in Appendix A4-3 for the full breakdown of materials used.

Results and Discussion

Successfully achieving our goal of reliable and repeatable track fabrication has provided several benefits to the project overall. The first benefit of fabricating all items in-house is the reduced monetary cost associated with fabrication. According to the costs of outsourced fabrication by previous years, as well as estimates made during the 2016-2017 year, the reduction in cost gained by in-house fabrication was between \$300 and \$400. The savings are very significant, especially when compared to the overall costs of up to \$800 for work that was considered this semester and the \$300 that was actually spent. Another benefit gained from performing fabrication in-house was that designs could be tested early and verified quickly, before seeking to outsource any potential mass production. One final benefit gained was that any changes necessary could be performed rapidly, with a new die made and bend tested within a matter of days, as compared to outsourcing work that would only be completed and delivered many weeks later. With the success of the Track Fabrication and Process Development subteam this year, the hope is that the Spartan Superway sees the benefits gained and will continue to employ fabrication teams in subsequent years to continue providing these benefits.

Conclusions and Suggestions for Future Work

In total sum, the Track Fabrication and Process Development subteam achieved all pertinent objectives and design requirements set forth by the Spartan Superway project. The team was able to reliably and repeatedly bend sections of small aluminum bar stock into arcs as required by the Track Improvement subteam's design. The team was also able to aid in the additional support requested by the Solar subteam by creating tabs and ribs, as well as providing general fabrication and assembly. The Track Fabrication and Process Development team was successful in the 2016-2017 iteration and addressed previously outstanding issues.

Through the course of working on the project over the months, several skills that have showcased their value time and again can be highlighted. Communication with other teams has been paramount, especially with regards to any work that depends on or requires inter-team interaction; in our case, this manifested in the form of requiring track and solar panel rack design to be complete prior to bender and rack creation. Allowing extra time to address any issues also provided particular advantage for the Track Improvement subteam and afforded them extra time to assemble the track before Paseo Public Prototyping. Working with many different individuals and multiple presentations have aided in the development of interpersonal skills, which are necessary in any environment.

As a message to any that would look to our year with the intent to improve upon the success found, there are several topics that can be addressed to provide immediate benefits. First, a reduction in the necessary level of precision required would greatly ease fabrication requirements; with the current design, success of the bogie in traversing the track is permitted within limitations as small as tenths of an inch. Second, know exactly what kind of material you will work with; the materials used this year had 2% different hardness properties, which if unnoticed would have created many more hours of work investigating the reasons for the lack of uniformity. Lastly, counter-sinking the heads of bolts on dies and shortening of the central column would allow the bender arm to rotate while sitting flat atop the die; such a modification could increase the accuracy of the bender and reduce the chance of material being bent unequally with the roller.

Track Improvement

Background and Context

There are numerous issues to work on in order for the track to be useable. The problems that were found from last year's design will be fixed to help the bogie run smoothly without falling off the track or getting stuck within the bottom rails. Some problems that exist are uneven planar surfaces. The track sits on an uneven surface, causing some of the rails to bend and alter the shape it is designed to be. This uneven surface can cause the bogie to struggle through the tracks. The bottom rails were never consistent. This can also cause the bogie to fall or struggle through the bottom tracks. In order to keep the bottom rail widths the same, a bracket was designed to hold it 2" apart. This bracket also helped align the top guideway rails. This ensures that the bogie will have adequate contact with guideway rail. The track was put together by tapping into the material and screwed in with small 4-40 screws. This made sections droop where tracks were mated. The design of the new bracket helped fix these issues.

Description of the Subteam and Objectives

The 12th scale Spartan Superway team is to create a feasible prototype in order to showcase the functionality of the project. Some exhibits include Paseo Public Prototyping and Maker Faire. The 12th scale team will showcase automated point to point transportation. The track does not replicate a city setting since it is only intended to show the potential of what Sparta Superway is. This year's 12th scale team focused on improving what has done in the previous years. For exhibits, the team was to have 6 functional bogies running on the track at the same time. Along with a functional track, being solar powered, and being able to track a bogies position. These objectives were as a whole 12th scale team. Each subteam had their own objectives to face. For Track Improvement team, the main objectives are to create a track that is portable, easily assembled and disassembled, and compatible with the other 12th scale subteams. Designs will be discussed in the next section.

Design Requirements and Specifications

The design requirements and specifications that the track needed were based on what was needed to get the idea to people and potential sponsors. The track will show point to point transit by the means of having four stations. Before the bogie can go point to point, the track has specifications that need to be met. The lower track, bottom rails, are to be two inches plus or minus about a tenth of an inch to have clearance for the bogie to run through. The guide rail, top rail, needs to low enough for the bogie's top bearings to be able to catch and guide through turns. A big design requirement is slotted posts. This will aid with uneven surfaces that may occur at exhibit locations. The track is to be made up of three different sections. This will make the track feasible for assembly and disassembly. This also helps with the portability of getting the track to places. The longest piece is about eight feet long which can easily fit in a van to be transported.

State-of-the-Art/Literature Review

There are only five automated transit networks around the world. They exist in Virginia (1975), Rotterdam (1999), Abu Dhabi (2010), London (2011), and South Korea (2014) according to “Automated Transit Networks (ATN): A Review of the State of the Industry and Prospects for the Future” by Burford Furman, Lawrence Fabian, Sam Ellis, Peter Muller, and Ron Swenson. Some of the greatest differences between the Spartan Superway and these other ATN is: Rotterdam does not have an exclusive guideway, Abu Dhabi and Korea continue to contribute to congestion, and London’s is not suspended which still consumes drivable space (Furman, Fabian, Ellis, Muller, & Swenson, 2014, p. 12). San Jose State University has developed the ATN by making it solar-powered and suspended so that it would reduce pollution and congestion. Spartan Superway also has an exclusive network that allows for origin-to-destination service which will save tons of time. Spartan Superway revolutionizes the ATN by combining the best attributes of these systems into a single system.

The 12th scale track team has come a long way since 2012. Figure 1-58 represents one of the first iterations of the track. And has since then been improved numerous times, which finally reaches our current and most updated track (Figure 1-59).

Figure 1-58. One of the first track iterations.

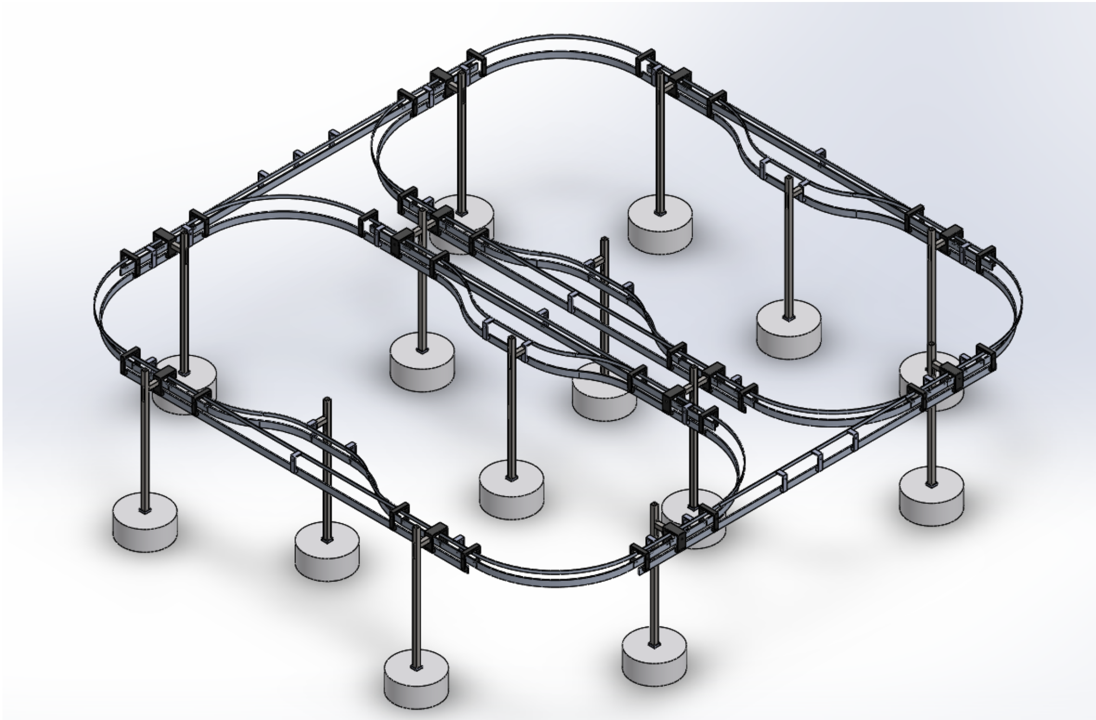


Figure 1-59. Most updated track.

Our team is the 12th scale track improvement team which does not really have state-of-the-art technology, but our team will be incorporating sensors and corrugated metal strips from the other 12th scale teams to monitor the location and to keep the bogie on course. The solar team will also be integrating their solar panels onto the track's poles, which will not directly affect the track, rather, it will affect the poles that hold up the track. Our job is to make sure that the poles that hold up the track are evenly spaced so that the solar team could use the poles. Next, we need to make sure that the track width is wide enough for the bogie to sail smoothly and for the corrugated metal strip to fit, while not wide enough for the bogie to slow or fall. Lastly, the track will have to be manufactured to be able to incorporate magnets for the controls team. These state-of-the-art technologies require a bit more work, but will be well worth it to produce a great product.

Description of Your Design

The design that the team went with is a three-section modular track. This means, the track consists of three types of sections to put the whole track together. The first part that will be explained is the U-bracket. This bracket is designed to solve the issues that the track had when we first worked on it. Figure 1-60 & Figure 1-61 shows a front and isometric view of the design. The slot seen at the top of the bracket is for the guide rail to fit in. The guide rail will be press fit into this slit to keep the alignment. The two bottom tracks will sit in the inner part of the bracket which will keep the right consistent width. The bottom rail is mounted to the bracket using an M5 nut and bolt. The top hole is where the bracket will be mounted to a support post. All part drawings can be found in

Appendix A5-2.

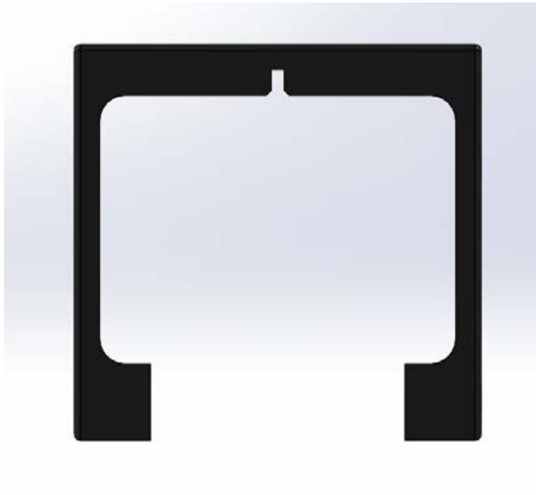


Figure 1-60. Front view of U-bracket.

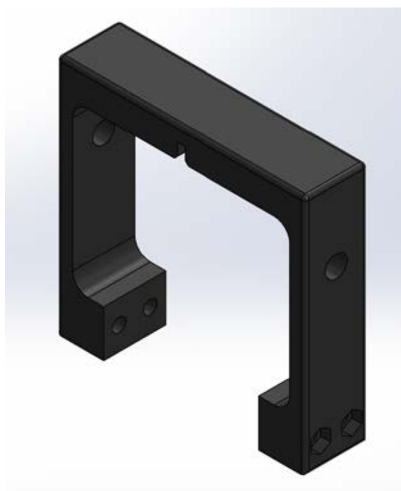


Figure 1-61. Iso view of U-bracket.

The next bracket is the U-Bracket V2. This bracket is exactly the same as the U-bracket. The only difference is the change of width. Figure 1-62 shows the changes of the U-bracket. This bracket is for mating points which will be discussed later - same function as the U-bracket, just with added support.

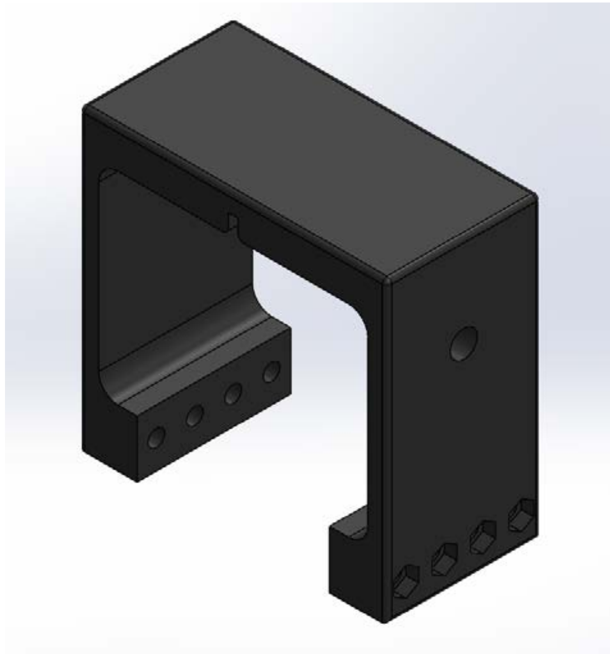


Figure 1-62. Iso view of U-bracket V2.

The last bracket, L-Bracket, is used as a support for the guide rail. Figure 1-63 shows the design of the bracket. This bracket is to be welded onto the bottom rails and screwed through the guide rail. These were machined out of aluminum. The top uses screws because during fabrication, found that welding the thin ⅛" thick guide rail was a challenge. These brackets help to create the foundation of the three sections.

The first section is the station. This consist of four U-brackets, two U-bracket V2, and three L-brackets. Along with bottom rails and guide rails. Figure 1-64 shows how the station is put together.

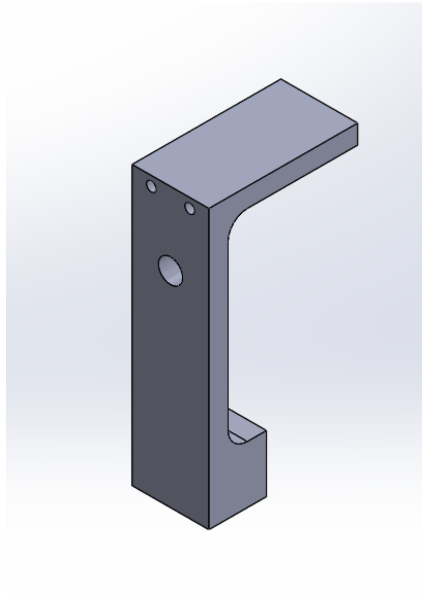


Figure 1-63. Iso view of L-bracket.

Figure 1-64. View of station assembly.

The next section is the corners. The corners are held together by welds and screws. This section consists of a top and bottom rail and two L-brackets. Figure 1-65 shows the corner section. The width between the two rails needed to be a certain dimension apart. Otherwise the bogie will fall and struggle to drive up the next track.

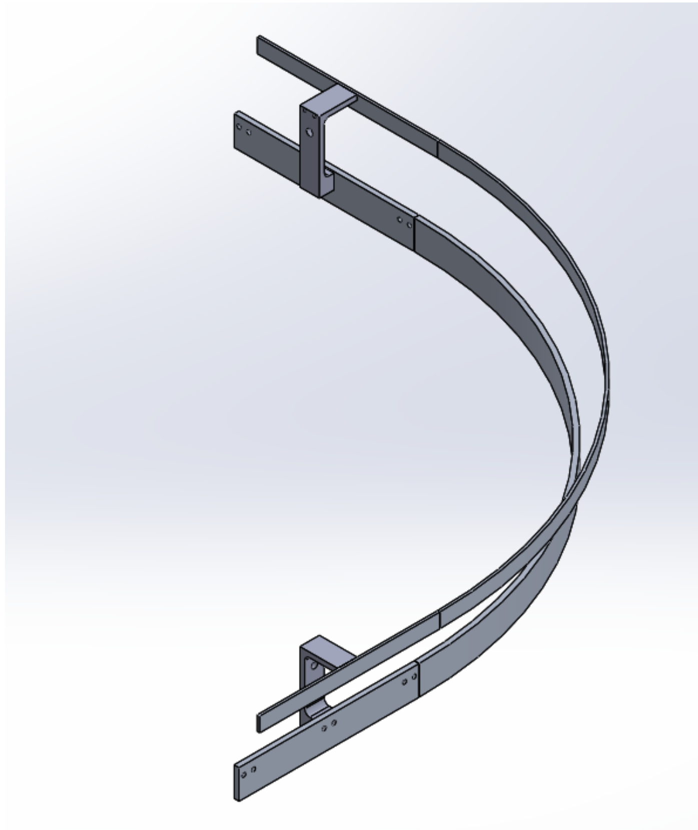


Figure 1-65. Iso view of corner assembly.

The last section is the straight section. Shown in Figure 1-66. This section is the key to connecting loops together. This allows the bogie to go from one loop to the next loop. These three sections are then mated together to create a loop. If more loops are needed, simply use the straight section to connect the loops.

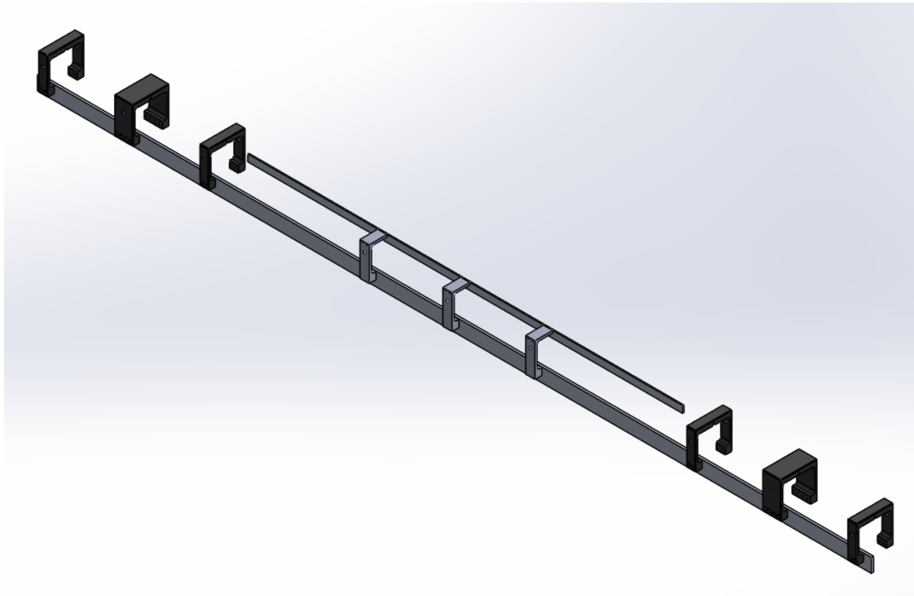


Figure 1-66. Iso view of straight section assembly.

Analysis/Validation/Testing

To test the track, Finite Element Analysis was done. FEA has been done to two parts and the track itself. FEA shows what type of displacements occur when under a certain load. Most of the figures shown will look scary but in reality, they are not. For the track, there are a total of 8 fixed points that are shown in green. These are where posts will be in order to support the track. The purple arrows show the a 10 lbf acting on the top of the track. FEA was done on this to see how many posts are actually needed for the track to be supported. The table is hard to read in Figure 1-67 but the red portion shows a displacement of about 0.02 millimeters. This displacement is very small and will not affect the track. This means there are enough posts on the track to support the loop. The next FEA part is the U-bracket. Looking at Figure 1-68, the max deformation reached is 0.3 millimeters. This is a small enough number to cause a big concern over. Last FEA part is the U-bracket v2 shown in Figure 1-69. This will be carrying a lot of load. This only reaches a max deformation of 0.003 millimeters which does its job perfectly. Some other tests that were done to prove the design specifications were met was to check the track level using a level. The slotted posts help to achieve this. Moving the track up or down in certain areas will ensure proper leveling. For the corners, blocks were made to ensure proper spacing and proper height for the top rail. The blocks made sure that there was proper clearance for the bogie to fit through and also that the top rail was low enough for the bogies top bearings to sit on.

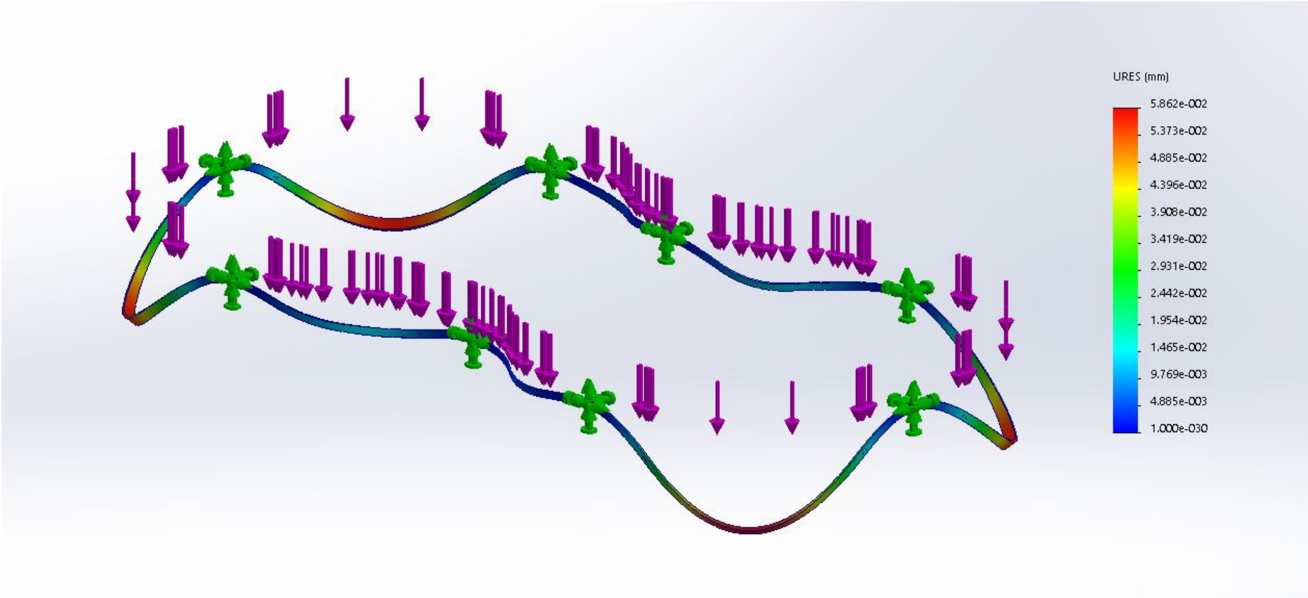


Figure 1-67. Finite element analysis of loop.

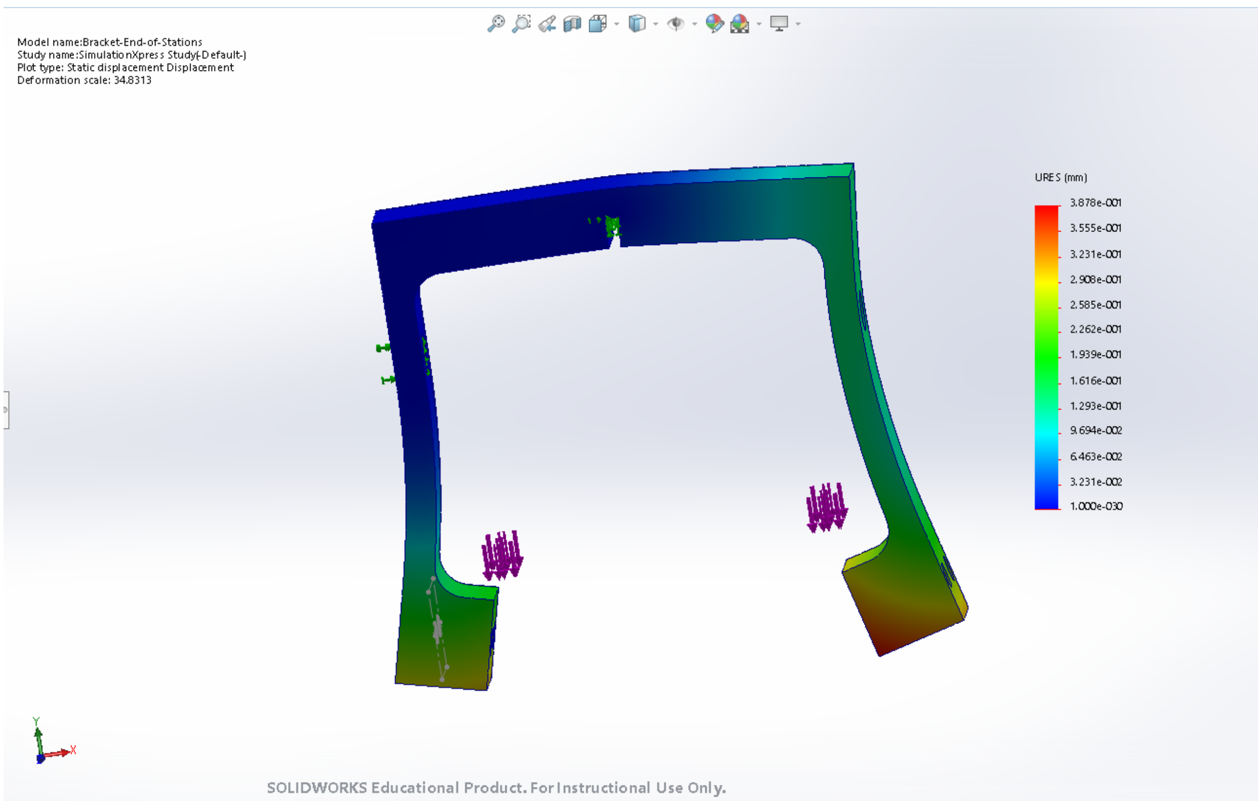


Figure 1-68. Finite element analysis of U-bracket.

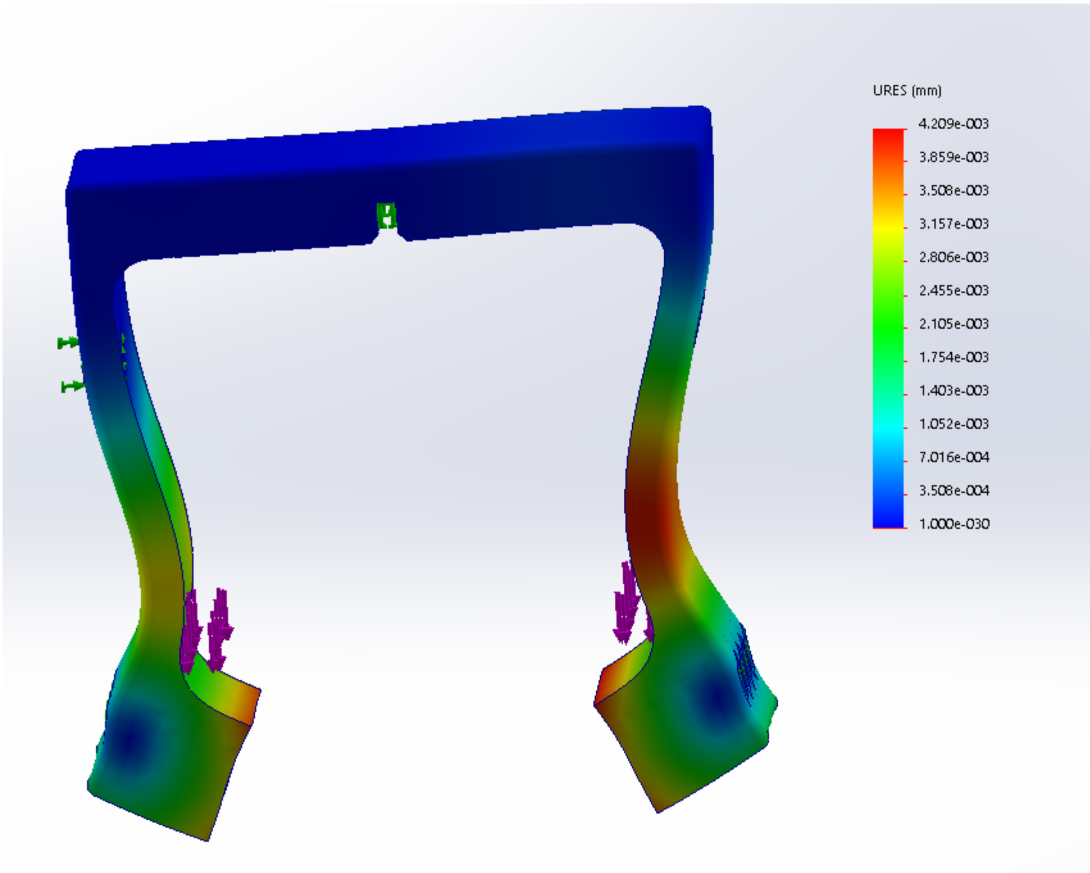


Figure 1-69. Finite element analysis of U-bracket V2.

Procedure / Instruction Manual

The disassembly and assembly procedure is pretty simple. Four tools (3-millimeter hex key, screwdriver, and two wrenches) will be needed as shown in Figure 1-70. Adjustable chairs are also of great help when assembling (not necessary for disassembling) - more chairs or a scissor lift make it easier. We will be using the terms found in Figure 1-71 throughout the instruction manual.

Figure 1-70. The four tools for (dis)assembly.

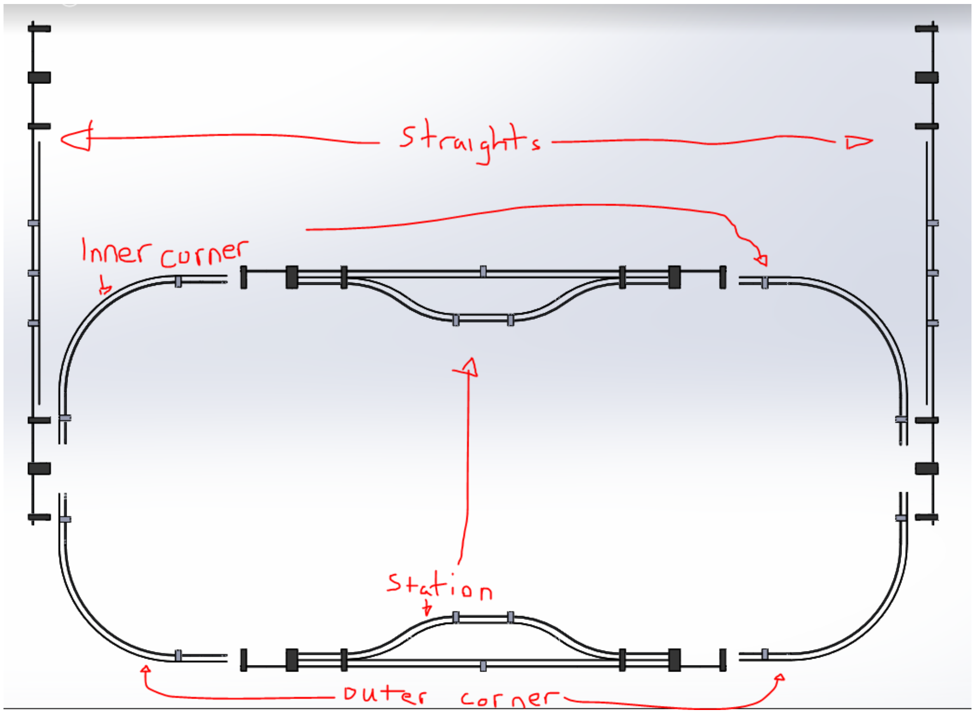


Figure 1-71. Picture of terms.

Disassembly

1. Loosen the black M5 hex screws of the U-brackets for the outside corners as shown in Figure 1-72. It is not necessary to remove the screws on the straights, as you can leave them on for less work to do during assembly process (as shown in Figure 1-73).

Figure 1-72. Unscrewing the U-bracket.

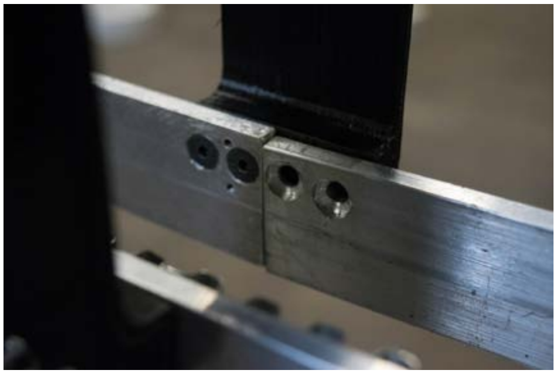


Figure 1-73. Notice how only two of the screws are not there - more screws removed = more work when assembling.

2. Once you take out all of the screws, hold onto the corner and you can just give it a gentle smack on each side of the corner and it will pop right off like in Figure 1-74.

Figure 1-74. Corner piece.

3. Repeat steps 1-2 for all outside corners.
4. For inside corners (those connected to the long straights) you will need to take out the screws attaching the inside corners to the straights (shown in Figure 1-75) with a screwdriver and then repeat steps 1 and 2 for the inside corners.

Figure 1-75. Screws connecting inner corner to long straights.

5. The stations can be easily removed with the two wrenches by holding one of the nuts and then rotating the other wrench to take out the second nut as shown in Figure 1-76. ONLY LOOSEN.

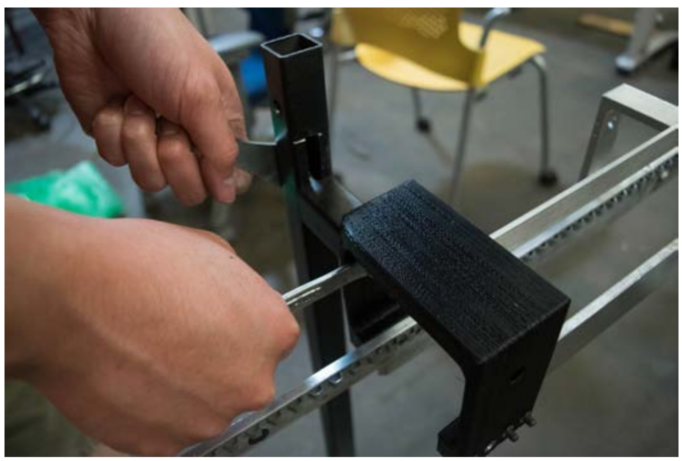


Figure 1-76. Loosening the nuts on the vertical support.

6. Once you loosen all three vertical supports holding the station, (with the help from some friends/chairs holding the station), unscrew the nuts, and gently take each one out. You're done!

Assembly

If at any point, screws don't line up, just use a little force to bring holes together.

1. Start with assembling the middle of the two track loops by first putting two concrete blocks with vertical posts in desired location as a starting point.
2. Put chairs at desired height of track and set the two middle stations (2 and 3) on top of the chairs while placing the 7-inch threaded aluminum tubing into the vertical support post (Figure 1-77).

Figure 1-77. Middle two stations are assembled.

3. On each side of the vertical support posts, put the spacers on the aluminum tubing and then the tubing should go through the U-brackets as shown in Figure 1-78. Place the washers and then screw in the nuts afterward for both sides. Hand-tighten and then tighten just a little bit with the wrenches as shown in Figure 1-76 from above.

Figure 1-78. Spacers on both sides of vertical posts.

4. Attach the inside corners by gently pushing it into the press fit of the U-bracket and line up the holes to add the black M5 hex screws.
5. Screw the M5 hex screws in with the 3-millimeter hex wrench. Avoid letting corners hang (adds stress to brackets) by using chairs to hold up corners.
6. Align the vertical support post with the U-bracket and put the 4-inch threaded aluminum tubing with the spacer on it into the U-bracket and vertical support.
7. Place washer and nut on both sides of the threaded tubing. Hand-tighten the nuts and then tighten with the two wrenches just a little bit.

8. Add the straights to the inner corners of the two tracks. Stick the long screws into the holes that connect the inner corners to the straights as shown in Figure 1-75 above. Use screwdriver and nut to secure. Avoid letting straights hang (adds stress to brackets) by using chairs to hold it up.
9. Secure the straights to the inside corners by screwing in the M5 hex screws with the 3-millimeter hex keys.
10. Add the outer corners by repeating steps 4-5.
11. Add the outside stations, similar to step 2 but with 4-inch tubing. Avoid letting stations and corners hang (adds stress to brackets) by using chairs to hold up.
12. Repeat step 6. You're done!

Money spent on your project

Overall, our team spent $583.04 as seen in Table 1-8. A third of that was obtaining more aluminum, which is actually super cheap since we purchased it from CoastAluminum. Because they are located in Hayward, shipment date is negotiable to whenever fits one's schedule. The second third of our spending was on PLA (polylactic acid) filament which we used to produce all of our U-brackets and items for the Bogie and Controls Teams. The final third was for screws, nuts, bolts, etc. Most of our material was purchased at Lowe's, Amazon, or McMaster. Our recommendation for McMaster is to purchase a few more screws or nuts than calculated because mistakes happen or miscounting occurs - their shipping is expensive, even for smaller purchases, thus one single shipment is best.

Table 1-8. Money spent.

Item	Price	Location
PLA x 1	$ 30.51	Amazon
PLA x 2	$ 43.98	Amazon
PLA x 3	$ 59.97	Amazon
Nuts and Bolts	$ 19.01	McMaster
Nuts and Bolts	$ 12.15	McMaster
Aluminum	$233.82	CoastAluminum
Dremel stuff	$ 11.03	Lowe's
loctite stuff	$ 20.88	orchard
Nuts and Bolts	$ 31.49	McMaster
Drill bit stuff	$ 48.45	lowe's
Blurry receipt	$ 12.01	home depot
HM 5/16IN-18 x 3 stuff	$ 12.69	lowe's
welder stuff	$ 47.05	praxair
TOTAL	$583.04	

Results and Discussion

Our project was a success. With the help from the previous years' numerous designs and iterations, our team has successfully created a track compatible with the other twelfth-scale teams. It has few to no problems remaining and functions as intended. It is compatible with the other subteams and the bogie runs through it smoothly. One point worth noting is that the corners are not entirely identical and thus each corner has a specific location with respect to the track and so it is important to keep each piece at its designated spot. This track will be useable and durable - any extra pieces that will be needed can easily be reproduced with either a 3D printer or the Track Manufacturing Team's bending station.

Conclusions and Suggestions for Future Work

The twelfth-scale track had many improvements to be implemented. We wanted to make it portable, leveled, evenly spaced, compatible with the other teams, and functional. We accomplished most of what we wanted to do for the year. One regrettable thing would be the nearly identical corners and nearly identical drilled holes. These two close but not precise implementations required the corners to be at specific locations. Aside from that, our track worked smoothly. One suggestion would be to come up with a method to flip the screws so that it would go from the outside of the U-bracket to the inside - easier to assemble. Increasing the height of the guideway will also increase the stability and tolerances of the bogie's upper bearings at the corners. The future teams could also create a method to speed up the assembly and disassembly process (screws are time-consuming) or to simply move the track as fewer pieces. Unlike other teams, the successfulness of our project is based on how unobvious our accomplishments are. Simply put, if we did our job correctly, then the track will be invisible because it is simply the foundation for the other subteams to showcase their accomplishments. Ergo, the more invisible our track seems, the better of a job we have done.

The track was important on a microscopic level because it acted as a base for the project. However, it has a macroscopic impact as well. The 12th-scale team's purpose is to portray the entirety of the project on a smaller scale. It is far easier to carry and transport than the full-sized project. It also helps to obtain publicity, viewership, and possibly sponsorship. Finally, the Spartan Superway will revolutionize public transportation - solar-powered, autonomous, and suspended. Since it is solar-powered, it mitigates pollution, is sustainable, and will be affordable. The autonomous component of it allows people to be transported in their own podcar at their own discretion (where and what time to meet the customer). Since each podcar is individualized, it will also go straight from pickup to destination without making any extra stops unlike most public transportations. Also, because it is suspended, it will avoid traffic. These qualities will motivate people to take public transportation and reduce pollution output. And the twelfth scale track will have helped progress the project one step forward.

Vehicle Controls

Background and context

Spartan Superway aims to have multiple small scale models of the transit network as proofs of concept to show potential investors and customers how the system would work. As of now, there are full-scale, half-scale and 1/12th scale models of the system, each with their own individual components and operations.

The 1/12th scale team uses a small bogie with a servo-actuated rocker to switch tracks. Previously, the team used a barcode reading system which ended up being unreliable in operation. This year, the system will incorporate a hall effect sensor and magnets to detect when to switch. Similarly, the previous team used an open source programming language and IDE called Processing to control the podcars. The methodology for using this software is very complex, and will be replaced by a mobile app for user interaction. The mobile app will be developed by the Software Team.

Last semester's objectives involved creating a working breadboard setup of the podcar with code that supports travel from station to station. The team was also required to make changes and provide a design that improved the system dynamics. These changes included a shift in components based on our decisions to move from the less reliable barcode scanner to a solid-state hall effect sensor. This semester's objectives involved rapid prototyping, code for pathing, integrating all code, along with developing functions in conjunction with the Software Team's requirements.

Description of the Subteam and Objectives

The Spartan Superway 1/12th scale Vehicle Controls team was tasked with designing a control system that will navigate a podcar around a track with minimal human input. The team aimed to create an autonomous system through integration of various sensors, electronics, and software to demonstrate this concept on the 1/12th scale model.

During the academic year, the Vehicle Controls team improved upon the prior method of switching the lever arms on the bogie of each podcar and created a reliable method of determining position of each podcar on the track. Although the 1/12th scale Positioning Team developed a fine method of determining the position of each podcar, the Vehicle Controls team created a coarse system that consisted of various checkpoints and stations around the track. The team had the following objectives for the project:

1. Redesign the method of determining when switching of the lever arms should take place
2. Simplify the prior control system design both in hardware and software
3. Improve the positioning system to reliably determine the position of each podcar on the track at various predetermined locations
4. Integrate the Vehicle Controls software and hardware with the Software Team's mobile app to demonstrate the overall functionality of the transportation system

Design Requirements and Specifications

The Vehicle Controls team had various design requirements and specifications that needed to be met to create an autonomous, reliable, and robust system. The previous controls system was complex and used software that was out of scope for a typical mechanical engineering student. A software package called Processing was used in conjunction with Arduino to control a podcar around the track by means of a user interface (see Figure 1-90 later in this report). In Processing, a user can specify the speed of a specific podcar, designate a podcar to a station, and switch the lever arms of a podcar manually. The positioning and pathing methods used a combined system of barcodes and nodes, where barcodes were used to determine when the switching of the lever arms should take place and the nodes were used to determine position around the track (Figure 1-79 below). These barcodes made the system unreliable when the podcar would occasionally misread a barcode or skip it entirely. Thus, the pathing algorithm and the method of determining when the lever arms need to be switched must be simplified.

```
<nodes>

103 280 420
104 350 260
304 650 260
402 780 420
403 780 620
404 700 760
204 370 760
102 280 620
203 510 620
202 510 420
302 550 620
303 550 420

#Stations
101 245 520
401 810 540
201 480 540
301 580 520
</nodes>
```

Figure 1-79. System of barcodes and nodes used for pathing and positioning.

These aspects of the former controls system did not qualify it as an autonomous system and required a major redesign. Aside from software, the hardware must be modular and expandable if additional sensors are required to create an efficient controls system. This requires that any microcontroller or any expansion board used to be readily and easily modifiable in the case that any hardware changes need to be made to the system.

There were many specifications that the controls team aimed to achieve in both the hardware and the software aspects of the project. As for software, the pathing algorithm required the use of magnets, hall effect sensors, RFID cards, and readers. The magnets and hall effect sensors would also be used to determine when switching should take place based on the path that a podcar is designated to take. The software required an interface between the Arduino software and the mobile app team's application to demonstrate minimal human input, which allows a user to request a podcar and navigates the podcar to any station that the user specifies.

For the hardware aspect of the controls system, the system required a suitable microcontroller that can handle all the inputs and outputs for the entire system. To avoid having various breadboards to connect all the electronics to the microcontroller, a shield must be designed or chosen to have all the electronics plug onto the board. The system also required a reliable anti-collision system that would be able to avoid any collisions using ultrasonic sensors. The controls system also requires an emergency button in the case that any podcar must abort its operation for any purpose. To determine the status of any podcar, an LED indicator is used. During the previous summer, a former controls team used a system of magnets and reed switches. These switches constrained the movement of the podcar and caused wear, as shown below in Figure 1-80. Thus, an alternative to this sensor must be chosen to perform the same function as it was intended. To determine position around the track and to identify each station on the track, RFID cards and readers must be used to differentiate the stations and to determine a rough location of a podcar on the track. For wireless communication, XBees must be used to connect the podcars and the mobile application. Finally, the podcar must be able to run at a target speed of 0.65 to 0.82 ft/sec where magnetic encoders mounted on the bogie will give feedback on the actual speed of the podcar. With the integration of these sensors and electronics, there must be six podcars running autonomously on the track with intelligent coordination using the mobile application.

Figure 1-80. The reed switch sensor which caused issues with track clearances.

State-of-the-Art/Literature Review

What is an Automated Transit Network?

An ATN (automated transit network) is a transportation system in which its vehicles are fully automated and run on specifically designed separate guideways, while operating on an on-demand basis. Unlike conventional transportation systems such as BART (Bay Area Rapid Transit) or VTA (Valley Transportation Authority) where passengers must board according to a time schedule and vehicles are constantly making stops at each intermediate station, ATN systems provide point to point travel for its passengers as soon as vehicles are requested.

Key Features of ATN Technology Around the World

The most pertinent examples of ATN technology can be found in the Morgantown Personal Rapid Transit and ULTra (Urban Light Transit) rapid transit (shown in Figure 1-81 and Figure 1-82 respectively).

Figure 1-81. Morgantown Personal Rapid Transit system podcar.

The Morgantown Personal Rapid Transit is an automated transit network located in Morgantown, West Virginia that connects the three campuses of West Virginia University to the downtown area. The Morgantown Personal Rapid Transit system consists of five stations and three different modes of operation depending on the hour of the day (Raney, 2000). Unfortunately, only during non-peak hours of the day does the system truly operate as an automated transit network in which passengers call for a car on demand and travel directly to the desired station. The routing of the Morgantown Personal Rapid Transit involves allowing the car to continue straight or to turn into the desired station. For example, if the passengers within the car have decided to go from ‘station 1’ to ‘station 3,’ then the car would skip ‘station 2’, continuing straight until it reaches ‘station 3.’

The ULTra PRT system, located in London at Heathrow Airport, is also an ATN that was developed to connect Terminal 5 to the passenger car park (ULTra PRT, 2013). This system was meant to extend throughout the whole airport, but the plan never came to fruition. Therefore, the system only consists of three destinations between Terminal 5 and the passenger car park. Unlike other automated transit networks, the ULTra PRT system uses concrete barriers as guideways rather than railways (ULTra PRT, 2013). The podcars can operate and steer themselves by shining lasers on the concrete barriers to measure the distance and to calibrate how much the podcar needs to turn to avoid collision. The podcar is also able to navigate itself based on an internal map that is implemented into its program (ULTra PRT, 2013).

Figure 1-82. ULTra PRT system podcars navigating their guideways.

Common key features of both existing ATNs are that they are automated, consist of specially built guideways, and provide point to point travel. Both are automated in the sense that the vehicles can operate themselves, rather than passenger input. Once passengers input their desired destinations, the vehicles are then able to navigate themselves to the corresponding station. Both systems also consist of guideways that lead to specific stations, with the Morgantown Personal Rapid Transit utilizing railways and the ULTra PRT utilizing elevated platforms with concrete barriers. In addition, both systems provide point to point travel where passengers can move from one station to another without stopping at intermediate stations.

Work Done on ATN at SJSU Since 2012

At San Jose State University, students and faculty have been working on the Spartan Superway project since 2012, which is an ATN that is intended to be implemented in San Jose and the Bay Area. The project is split into three scales: 1/12th scale, half-scale, and full-scale. The 1/12th scale model is intended to show how the Spartan Superway functions and to provide insight into any potential issues that may arise in the larger scale implementations. Alternatively, the half-scale and full-scale projects are intended to show that the Spartan Superway functions properly for a small portion of the track guideway. Work done on Spartan Superway includes developing a suspended guideway that elevates the podcar to avoid traffic on the road or ground surface. This was done with the intent of decreasing road congestion and commute times. Additional work includes developing a bogie system that can safely connect the podcar with the guideway such that the podcar can navigate and switch tracks safely. Work has also been done on developing a suspension system that allows the podcar to travel through inclines such that it stays parallel to the ground, ensuring passengers experience no discomfort. Most importantly, work has been done on developing a controls system that would tie all the mechanical aspects together with the electronics. Four years in progress, the Spartan Superway is a project in which improvements are continuously being made every single year. Teams can research external information or previous teams' work and progress, and either build off that information or develop an entirely new and improved method that satisfies the project requirements. Therefore, Spartan Superway is an automated transit network project that is intended to be implemented into society for public use.

Description of Your Design

General Design

The current controls system utilizes components from the previous controls system such as the Pololu micro gearmotors, HC-SR04 ultrasonic ranging modules, XBee Pro Series 1 modules, and Hitec HS-5087MH servos. Unlike the previous team's utilization of an Arduino Uno and Arduino Nano combination, the 2017 Vehicle Controls team opted for the Arduino Mega due to the increased number of input and output pins. This option simplified the system and removed the communication issue between the two microcontrollers.

The Pololu micro gearmotors were used for propulsion, in conjunction with the standard Arduino motor driver library. Initially, the Timer 3 library was used to control the motors but changes to the design required that the motors not occupy the interrupt pins on the Arduino Mega. However, the Timer 3 library was still used for ultrasonic purposes since the Software Team required a quicker response from these sensors. For convenience, the Timer 3 library can be found in the Spartan Superway Archive under the folder for the 2016-2017 1/12th scale Vehicle Controls team. In addition, each motor was connected to the DRV8838 motor driver board where the motors required 6V of power supply. A wiring schematic of the whole controls system can be found in Appendix A6-1 and

Appendix A6-2. Each motor is equipped with rotary encoders, which were originally going to be used for feedback on the actual speed of the podcar. The controls team was not able to successfully acquire a reasonable number from the sensor that resembled a speed and thus the encoders were disregarded.

One of the design requirements that the controls team aimed to meet was a reliable and efficient anti-collision system. The previous team used an ultrasonic sensor to prevent forward collisions and this aspect of that system was integrated to the current anti-collision system. Unfortunately, the podcars do exhibit blind spots that the forward-facing ultrasonic sensor does not detect, as shown in Figure 1-83. For example, collisions would occasionally occur on the left or right side of the podcar in sections of the track that merge together. To resolve this issue, a second sensor was placed on the left side of the podcar to prevent collisions from the podcar that is merging into traffic. Due to time constraints, this solution was not fully tested and was left as a suggestion for improvement in the anti-collision system aspect of the controls system.

Figure 1-83. Blind spots exist on the right and left side of the podcar.

In addition to collision detection, the Vehicle Controls team created a method of halting the motion of a podcar in the case of an emergency or for any other purpose. To achieve this, a simple momentary push button was used. To avoid the need to press and hold the button, Boolean logic was used to create a latching button where if the button was pressed, the podcar would stop and stay in that state. Since this event occurred and no other code responded to multiple presses of the button, a second momentary push button was added to reset the system to the previous state before the kill button was pressed. These two buttons are shown in Figure 1-84 below.

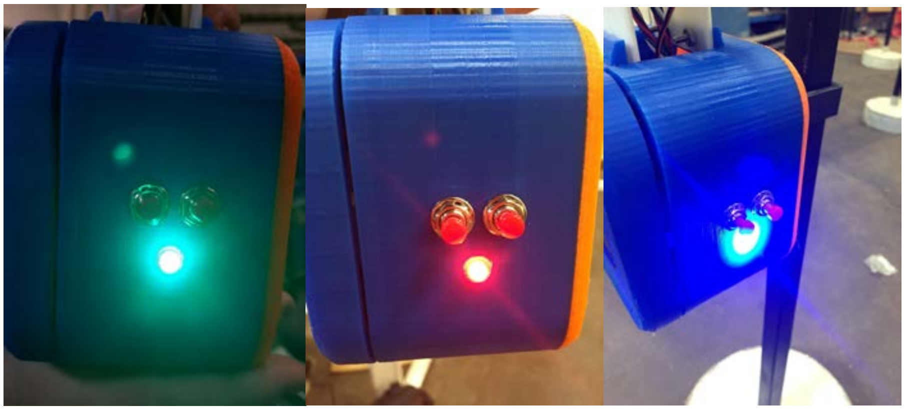


Figure 1-84. The two switches and LED on the podcar.

One accommodating feature that was added to the controls system was the use of an LED indicator which represented different podcar states. This indicator helped greatly during the debugging phase of the project. There are three main colors that were used to indicate different states of the podcar as shown in Figure 1-84 (above): a green light indicated that the podcar is in motion, a red light indicated that the podcar has stopped, and a blue light indicated that the podcar successfully read a magnet or an RFID card. There were two other variations that were used to indicate other statuses. For example, a blinking red light indicated the emergency state whereas a blinking blue light indicated that the podcar is navigating a desired station-to-station route. The full Arduino sketch for the controls system can be found in Appendix A6-3 and includes the methods that were used to control each sensor and other electronics of the control systems.

Finally, the only form of communication that was present in the controls system was through the use of the XBee Pro modules. In contrast to last year's controls system, these modules use an adapter (see Figure 1-88 later in the report) instead of the use of an XBee shield that was placed on top of the Arduino Uno. These modules were mainly used to provide communication between the Software Team's mobile application and the six podcars. These modules are mainly in the design of the mobile app and thus it recommended that one references their report if further exploration of how these modules work is desired.

Method of Switching

Hall effect sensors were used to determine when switching of the lever arms should occur. These sensors, shown in Figure 1-85 below, were small enough to incorporate into the side of the bogie. The sensors change their output based on an incident magnetic field.

Figure 1-85. Hall effect sensors were used to determine when switching should occur.

Similarly, magnets were placed at different locations of the track (Figure 1-86), and a pathing algorithm was developed to determine when the lever arms should switch. In addition to the hall effect sensors, the servo was connected to the lever arms of the podcar which would actuate based on the detection of a magnet and based on the route that was assigned to a specific podcar. The hall effect sensors were mounted on both the right and left sides of the bogie so that the podcar would be able to detect magnets on both sides of the track.

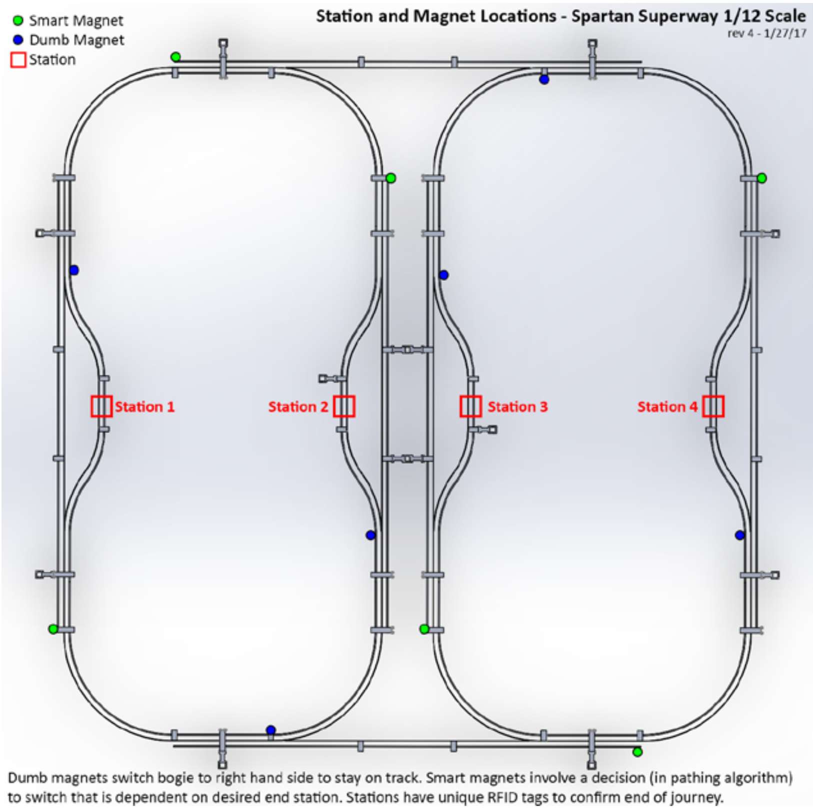


Figure 1-86. The location of the magnets was crucial to the pathing algorithm.

Pathing Algorithm

As mentioned in the previous section, hall effect sensors were used to determine when switching should occur and magnets were placed at different locations on the track. There were two different magnets that were placed on the track, labeled “smart” and “dumb” magnets. “Dumb” magnets consisted of magnets that were placed on the right side of the rails on the track. These magnets would cause the podcar to switch the lever arms to the right arm up position to make sure that the podcar would remain on the track regardless if a station was assigned to the podcar or not. “Smart” magnets consisted of magnets that would prompt the podcar to decide on whether to turn into a station or to continue straight on the track. These magnets were situated on the left side of the rails on the track. The pathing algorithm consisted of creating 12 different path possibilities that a single podcar could take and then creating a stack array that holds values of 0’s and 1’s based on the path that a podcar is assigned. To use the stack array, a library must be included in the Arduino IDE which can be found in the Spartan Superway Archive. This array would subtract one value from the right of the array whenever a magnet is read (along with the short blink of a blue light from the indicator), and would keep on deleting values until the array is blank. When the array reached an empty state, the podcar has arrived at its destination. This pathing algorithm is incorporated in the Arduino sketch and can be found in Appendix A6-3.

Identification of Podcars and Stations

To differentiate one podcar from one another, the podcars were labeled 1-6 and unique sketches loaded to each individual Arduino Mega. This convention also aided the Software Team in sending packets of data to a specific podcar. Furthermore, four RFID cards were mounted on the track to differentiate each station. Four additional RFID cards were placed near the corners of the outer loop as checkpoints for a rough indication of where the podcar was relative to the nearest station of the corresponding card. An image of an RFID card and reader is shown in Figure 1-87. Software for these RFID readers can also be found in the Arduino sketch that is included in Appendix A6-3. It’s important to note that the RFID readers require a special library designed for the Arduino Mega that can also be found in the Spartan Superway Archive.

Figure 1-87. RFID card and reader used for station identification.

Expansion Board

Since the current controls team had more electronics connected to a microcontroller compared to the previous year, a prototype expansion board was suitable in gathering all the sensors and electronics onto one board without the need of any breadboards or unnecessary wires. This expansion board was an alternative to a PCB since the controls system was never tested on an Arduino Mega prior to this year's design. For this reason, a prototype of the board was necessary before deciding on a permanent solution, such as PCB, where changes would be difficult to make after production. Although the team planned to minimize the use of jumper wires on the bottom of the expansion board to make the connections between pins, this could not be avoided since changes were made to the board during the last stretch of the year. Initially, the team attempted to add a jumper connection between the XBee adapter and power since Arduino does not allow any connections to the TX and RX pins on the microcontroller when uploading code to the board. It was assumed that the jumper would not provide power to the adapter if the jumper was in the cutoff position, but it was found that the adapter was still powered through the communication pins. Thus, instead of soldering the adapter to the expansion board, headers were added so that the XBee could be removed when uploading code to the Arduino Mega. This same method was applied to all the sensors and other electronics so that if any of the electronics or sensors were faulty, they could easily be replaced without having to de-solder the connection on the board. Figure 1-88 below shows the current expansion board that fits on top of the Arduino Mega.

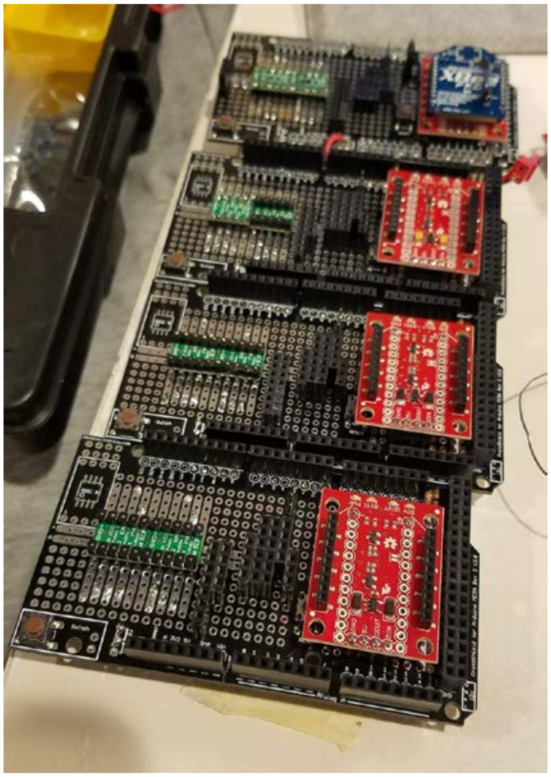


Figure 1-88. Expansion boards that show headers as well as connections for the XBee adapters.

Mounts and Brackets

Since the controls team utilized hall effect sensors and RFID technology, various brackets and mounts needed to be designed to mount the RFID cars and to house the hall effect sensors. Figure 1-89 shows the bracket and mount that were used. As mentioned earlier, the previous controls team employed a system of barcodes and nodes where they used an optical sensor. These sensors were housed in a bracket that they had designed and 3D printed and were mounted on the bogie. Since there were plenty of these brackets readily available, they were repurposed for the hall effect sensors. These sensors were placed inside the brackets, and hot glue was used to fill in the gaps between the sensor and the bracket (Figure 1-89, right). A mount for the RFID cards was designed so that it would be adjustable to account for inconsistencies in track dimensions and podcar width. These mounts were fastened onto the poles that supported the track using an M3x8 screw. The CAD model for the mount can be found in the Superway Archive.

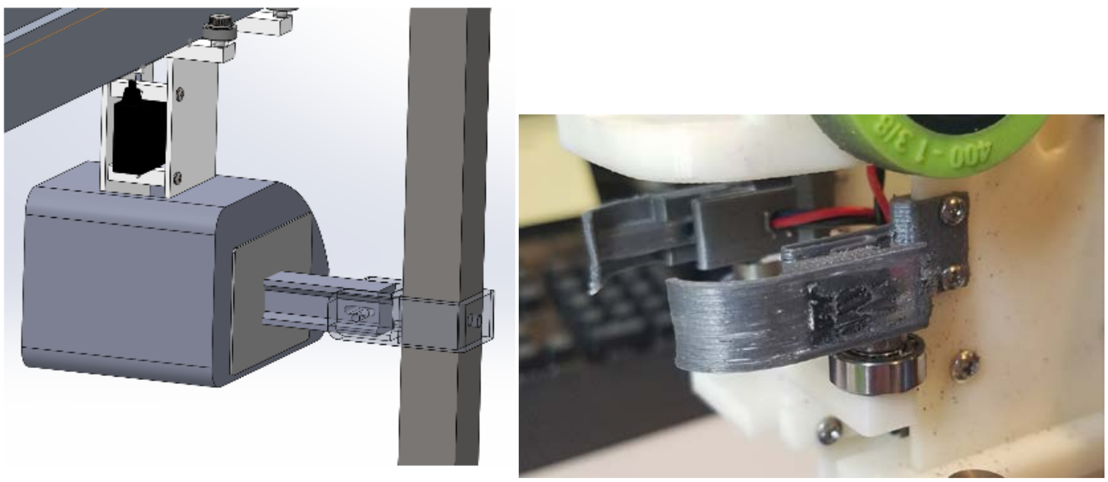


Figure 1-89. RFID card mounts (left) and hall effect sensor brackets (right).

Analysis/Validation/Testing

This semester, the team made sure that the design met the requirements and specifications through numerous iterations of testing and validating. Sensors were implemented in a top-down fashion by creating specific functions for each to validate that they work individually. For example, the function, 'isSomethingInFront' was created to ensure that the ultrasonic sensor operates as a standalone unit. Other functions were created as well to test other components such as the hall effect sensors, RFID card reader, and servo motor. Once it was validated that all sensors or components could operate on their own, all the functions were then integrated into a larger single code to test that all the sensors and functions could operate together without any issues. Overall, the team's process of testing and validating ensured that sensors and code could operate individually, and then confirm that they could operate integrated in the entire system. The same method was used for building the expansion boards, where one was tested and validated for proper functionality before building the rest. Once all the expansion boards were soldered, small revisions were then implemented as changes to the controls system were made.

Testing was also conducted alongside the Software Team, as the Vehicle Controls team's system was required to be integrated with the mobile application. Therefore, much of the later testing was conducted with the podcars on the track using the mobile application. Since the mobile application did not fall within the expertise of the team, the Vehicle Controls team took on more of a support role in this later phase of testing. The team provided assistance with the podcar whenever there were mechanical issues while the Software Team conducted testing on their mobile application. However, the procedure for testing and validating remained the same. For example, the mobile application was tested with all individual sensors and components to validate that they all operate individually. Once the sensors and components were validated with the mobile application, they were then tested and re-validated together as a whole system. After it was established that the mobile application and the controls system could operate together, the podcar was then tested on the track for pathing. Pathing was again tested and validated using the mobile application where a pick-up point and destination would be chosen. Once chosen, the podcar travels to the pick-up point, stops for passenger pick-up, and then travels to the destination where it stops for passenger drop-off. Changes and fixes were then implemented as issues and bugs were discovered.

Procedure / Instruction Manual

The Arduino code developed this year can be found on the Superway Archive, with the file path specified below:

2016-2017 ME195A/B Project Folder / 2016-2017 1/12th Scale / Vehicle Controls / Code

In this folder is all the associated code, commented and described as thoroughly as possible. The podcars run on code that allows them to be controlled via the mobile application, and specific sketches which make the podcars run on the inner and outer loop. All the files are labeled within the folder, and Table 1-9 below shows the meaning of the various code files.

Table 1-9. Final 2017 Arduino Mega code folder contents.

File Name	Description
Mega_2017_Scheduling_New_Expansion_Board	Arduino Sketch that allows the podcar to be controlled via the mobile application for 'rides'. The podcar will default to outer loop after the ride is complete. Input/output pins designed for use with reworked expansion board (V1.1)
Mega_2017_Scheduling	Arduino Sketch that allows the podcar to be controlled via the mobile application for 'rides'. The podcar will default to outer loop after the ride is complete. Input/output pins designed for use with original expansion board (V1)

Mega_2017_Outer_Loop_New_Expansion_Board	Arduino Sketch that sets the podcar to continuously drive around the outer loop. Input/output pins designed for use with reworked expansion board (V1.1)
Mega_2017_Outer_Loop	Arduino Sketch that sets the podcar to continuously drive around the outer loop. Input/output pins designed for use with original expansion board (V1)
Mega_2017_Inner_Loop_New_Expansion_Board	Arduino Sketch that sets the podcar to continuously drive around the inner loop. Input/output pins designed for use with reworked expansion board (V1.1)
Mega_2017_Inner_Loop	Arduino Sketch that sets the podcar to continuously drive around the inner loop. Input/output pins designed for use with original expansion board (V1)

Use of this code requires the use of various Arduino libraries, which are also included in the same folder. Before loading the sketch to the Arduino, these libraries should be downloaded and placed in the appropriate Arduino libraries folder. Restarting the Arduino IDE will then automatically load these libraries, and allow the user to program the Arduino normally.

Operating the podcars built this semester is quite simple; there are two momentary push-button switches located on the rear of the cabin, as well as an RGB LED. The left button is the kill switch, which stops the bogie during operation. Additionally, the kill switch is used to manually flip the switching arms, allowing the user to place the podcar on the track easily. Note that the podcar starts up in a kill state when the battery is plugged in. The right button is for restart, which starts the bogie while in a kill state. It is used for starting the bogie after it has been correctly mounted on the track. Table 1-10 below shows the various color indicators for the podcars.

Table 1-10. Podcar indicator status interpretation.

LED State	Podcar State
Red	Obstacle Detected
Green	Moving / OK
Blue	Station Read
Red Flashing	Kill

Purple Flashing	Station-to-Station Navigation

The Arduino Mega expansion boards were built following the wiring schematic for our design, with various changes worked in throughout our testing cycle. The pinouts for the boards were recorded and meticulously tracked with various documents and spreadsheets. The pinout for the wiring harnesses and the podcar build status documents will be the most useful for future groups to reference. These are available using the following file path:

2016-2017 ME195A/B Project Folder / 2016-2017 1/12th Scale / Vehicle Controls / Documentation

The 2015-2016 Spartan Superway controls team as well as the 2016 Summer Korean team both used a computer programming language and IDE called Processing. With this software, a user could specify the speed of a podcar, the podcar destination (station), request the status of a podcar, and manually switch the lever arms of a podcar. This software was removed from use from the current 2016-2017 Vehicle Controls team because a different user interface and pathing algorithm was incorporated into the system.

For any future teams that wish to use the older software, a brief set of instructions have been developed by the Vehicle Controls team to save time in the long run. First, the software must be downloaded from the Processing website and a folder will be created under the directory specified by the user where the libraries, examples, and other folders will exist after installation. Next, the external Processing libraries that were created for the Spartan Superway project may be found in the Spartan Superway Archive under the 2016 Summer Small Scale folder. These libraries must be saved under the Processing libraries folder where the Processing software folder was created (under the directory specified earlier). After successfully downloading and saving the libraries to the Processing libraries folder, all five sketches that are in the 2016 Summer Korean team's controls folder (.pde file type) must all be saved under one of the sketches. That is, after downloading all five sketches and saving them, four of the five sketches must be moved to one of the sketches folder. Consequently, all five sketches will be under one sketch folder. Under that same folder that all five sketches exist, the Spartan Superway logo .png file, the Spartan Superway track .txt file, and the track .png file must all be saved under that same folder. Finally, there are 13 Java files that must be downloaded and saved under that same folder where all five sketches exist. These files for Processing can be found under the 2016 Summer Korean Small Scale team's folder in the Spartan Superway Archive. After these steps, the user clicks 'play' in the Processing IDE and the same window as in Figure 1-90 below should appear.

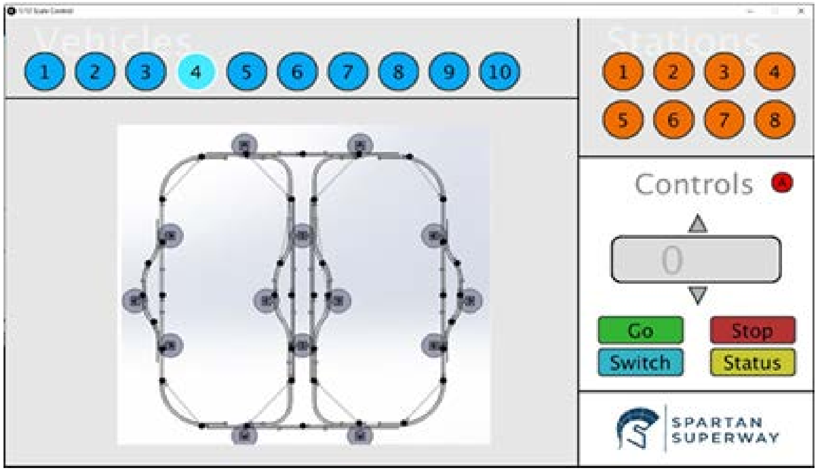


Figure 1-90. Processing software used to control podcars in previous semesters.

Expenses and Costs

At the end of the first semester of senior project, the Vehicle Controls team developed a Bill of Materials (BOM) using the components of the designed system that totaled around $855. This BOM (Figure 1-91) was conditionally approved, on the pretense that the team must get in contact with HSC to see how much of the electronics could be purchased using Professor Furman's credits. The team visited HSC in January, and while they did not have all the items in stock, they were willing to order from SparkFun, a vendor that carries most of the Arduino-based electronics that were required for the project.

Item	Purpose		Product Link	Alternative	Approx. Lead Time	Qty in house	Total quantity for 6 podcars	Additional qty. req. for 6 podcars	Price (ea)	Ext. Price for 6 podcars
Hall Effect Sensor	Detection of magnets on track	A1104LUA unipolar	Jameco	HSC?	1-2 days	2	12	10	$1.39	$13.90
Motor driver board	Propulsion		Polulu		3-5 days	8	12	4	$3.49	$13.96
Sonar	Forward collision detection		Amazon	Sparkfun	2-3 days	7	6	0	$1.79	$0.00
Arduino Uno	Processing of motor & sensor data		Amazon	DigiKey	2-3 days	3	7	4	$16.06	$64.24
Arduino Uno Xbee shield	Arduino/Xbee communication		Amazon		2-3 days	1	7	6	$17.98	$107.88
Arduino/OSEPP Nano	Processing of sonar & sensor data		DigiKey	Fry's	2-3 weeks	2	6	4	$21.49	$85.96
RFID Reader + Cards	Detection of station ID + station ID		Amazon		2-3 days	6	6	0	$7.99	$0.00
RGB LED	Podcar status	Common cathode	Jameco	HSC?	1-2 days	2	6	4	$0.39	$1.56
XBee Pro	Wireless communication	Xbee Pro Series 1	Sparkfun	Amazon	2-6 days	2	7	5	$37.95	$189.75
Momentary button	Kill switch for each podcar	SPST momentary	Amazon	HSC?	2-3 days	0	6	6	$0.70	$4.20
Servo	Track switching	HS-5087MH	Amazon		1-2 weeks	1	6	5	$36.95	$184.75
Drive motors	Propulsion	Polulu 2215 75:1	Polulu		3-5 days	4	12	8	$16.95	$135.60
Drive motor encoders	Podcar speed		Polulu		3-5 days	6	12	6	$8.95	$53.70
										$855.50
										Total 6

Figure 1-91. Original BOM for vehicle controls team prime design.

During winter break, the team worked on simplifying the system structure. In doing so, the team negated the use of two microcontrollers, instead opting to use one Arduino Mega. This eradicated communication issues between the two devices, while simultaneously providing more input/output pins to work with.

After confirming the electronic components for the system, several orders were placed at HSC between January and March of 2017, well ahead of the intended podcar build date. This was done to ensure that the team wouldn't have a lack of parts when the time came to start assembly.

As mentioned, the original BOM was about $855. However, this did not include many consumable items that were not accounted for, such as header pins, cabling, magnets, resistors, and additional momentary switches. With these items added, the total was roughly $980 (Figure 1-92). More than $578 of that amount was purchased at HSC with Professor Furman's credit, leaving only $400 to be spent out-of-pocket for the entire Vehicle Controls team project.

Item	Purpose	Product Link	Alternative	Approx. Lead Time	Qty in house	Total quantity for 6 podcars	Additional qty. req. for 6 podcars	Price (ea)	Ext. Price for 6 podcars
Hall Effect Sensor	Detection of magnets on track	Jameco		1-2 days	2	12	10	$1.39	$13.90
Motor driver board	Propulsion	Polulu		3-5 days	8	12	4	$3.49	$13.96
Sonar	Forward collision detection	Amazon	Sparkfun	2-3 days	7	6	0	$1.79	$0.00
RFID Reader + Cards	Detection of station ID + station ID	Amazon		2-3 days	6	6	0	$7.99	$0.00
Servo	Track switching	Amazon	AeroMicro?	1-2 weeks	1	6	5	$36.95	$184.75
Drive motors	Propulsion	Polulu		3-5 days	4	12	8	$16.95	$135.60
Drive motor encoders	Podcar speed	Polulu		3-5 days	6	12	6	$8.95	$53.70
								Out of pocket:	**$401.91**
RGB LED	Podcar status	HSC	1/6/2017	< 2 wks	2	6	4	$1.95	$7.80
XBee Pro	Wireless communication	HSC	1/6/2017	< 2 wks	2	7	5	$37.95	$189.75
Momentary button	Kill & reset switch for each podcar	HSC	1/6/2017		0	12	12	$1.09	$13.08
10k resistors	Needed for every hall effect sensor	HSC	1/6/2017				24	$0.02	$0.48
Xbee regulator board	Regulator for Xbee	HSC	2/2/2017	< 2 wks			6	9.95	$59.70
Arduino Mega 2560 R3	Main microcontroller	HSC	2/2/2017	< 2 wks			6	45.95	$275.70
Wire (22/26awg)	For building expansion boards	HSC	2/2/2017				4	7.95	$31.80
								HSC:	**$578.31**
								Total:	**$980.22**

Figure 1-92. Final BOM for prime design, including portion spent at HSC.

The affiliation with HSC was invaluable in purchasing the parts for the designed system. Their cooperation and Professor Furman's credit allowed the team to build six bogies, each with an identical full stack of electronics.

Results and Discussion

The Vehicle Controls team successfully built 6 fully functional expansion boards for each of the podcars. Although there were not six podcars running simultaneously on the track, three podcars were exhibited at both major events. It was deemed unnecessary to have the full six on the track, due to congestion in the small track layout and mobile application scheduling issues. Due to some software issues and modifications to the expansion boards, not all podcars could be assigned a station through the mobile application. Thus, the team has six variations of the code, four of which have predetermined routes that the podcar takes once the code is uploaded to the Arduino Mega. The other two sketches contain code that is strictly for scheduling through the mobile application. These sketches can all be found in the Spartan Superway Archive under the 1/12th Scale 2016-2017 Vehicle Controls folder.

As mentioned, not all 6 podcars had the ability to have stations assigned to them through the mobile application, but important conclusions can be drawn from the work that the team accomplished. A controls system that is integrated with the mobile application demonstrates that a full-scale version of this 1/12th scale model is possible in a real-world environment. Although the batteries were a factor for how long the system ran, a transit network that runs solely on solar power is possible if the run time of the batteries matches the charge time of the batteries. In addition, as the batteries depleted, various electronic components, such as the ultrasonic sensors, slowly began to not function with the rest of the system. The batteries did not supply enough power to the entire system because the voltage of the battery depleted as the batteries were in use. Overall, the concepts of a fully automated, solar powered, suspended transit network was successfully demonstrated on a 1/12th scale model.

Conclusions and Suggestions for Future Work

The team was able to effectively meet the design requirements of improving switching and pathing with multiple podcars running on the track. Rather than utilizing barcodes, which were unreliable, the team took advantage of using hall effect sensors and magnets on the track to develop a robust switching system for the podcars. In addition, pathing was simplified in terms of expandability and comprehensibility for future mechanical engineering teams. Barcodes are an uncommon tool for mechanical engineering majors so figuring out how they work would be time consuming for individuals who are not familiar with them. Therefore, the team utilized stack arrays and RFID card readers instead, which are significantly easier to understand and proved to be very reliable in application.

Overall, the team did very well in improving the controls system from last semester. Through working on the Spartan Superway, the team developed technical skills involving coding, soldering, computer-aided design, and working with various electronic components in general. In addition to technical skills, the team learned the importance of communication after having to moderate discussions between various cross-functional teams to complete tasks on time and with minimal issues. Although the team was successful in meeting the design requirements, there are always improvements that can be made.

Suggestions for future teams include utilizing and testing encoders for speed control feedback, as well as positioning. In addition, either a second ultrasonic sensor or new method can be implemented for collision detection to avoid potential accidents at the merging points within the track. In addition, once a permanent controls system is created, the expansion boards should be replaced with custom PCBs to avoid complex wiring. Lastly, the current code contains over 2000 lines of code, which leads to long file loading times and headaches when trying to decipher certain functions. Thus, it is recommended that libraries are created for each of the sensors and electronics to greatly reduce the length of the code and better understand how the controls system functions.

Chapter 2: Half-Scale

Active Suspension

Background and context

One of the problems that the Spartan Superway faces, is the pod car going on an incline or decline railing. The active suspension will accommodate the different kinds of railing by utilizing its tilt mechanism, which will tilt the pod car so that it will stay level no matter what so that the passengers will be tumbling all over the place inside the pod car. Without the tilt mechanism, the Spartan Superway would have the same ride quality as a roller coaster which could potentially injure the riders.

In addition to the tilt mechanism, a self-leveling system is needed within the active suspension. Since we don't know if the platforms are all going to be the same height, we need to be prepared for all different scenarios. The self-leveling system will allow the pod car to be level with every platform to ensure that it is accessible to all passengers.

The last aspect that the active suspension must address is the suspension system itself. Just like the suspension in car, the suspension for the pod car must serve the same purpose to prevent the pod car from binding or flipping over when making a sharp turn or coming to an abrupt stop.

Description of the Subteam and Objectives

What the active suspension subteam is focusing on is making sure that the Spartan Superway is to improve ride quality to make sure it as comfortable as possible for passengers. Like in the suspension of a car, it absorbs the shock when the car goes over a bumpy or uneven road. It also prevents the car from flipping forward or tipping over when making an abrupt stop or making a sharp turn. The team's number one goal is ride quality, so in addition to making the active suspension like how a regular suspension is in a car, we have to incorporate systems like a tilt mechanism and self-leveling mechanism to our overall system.

The team approached the problem by researching different kinds of suspension like the ones in motor vehicles. Although, the research was not applicable to the system because the suspension or a car are different than what is needed for an active suspension for a suspended transportations system. In addition, there is no information at all regarding an active suspension. Therefore, the team hand sketched ideas from scratched and critiqued them until mutual ideas were agreed upon. Subsequently, Brean applied force, vibration, and torque analysis to the system, to be able to come with the needed materials to operate at the level the Superway requires. Neil, the team leader, then utilized Solidworks to CAD the final sketch to check if there were any clearance or mechanical issues.

After the research, calculations, and design the suspension team went into the manufacturing phase. This means making sure the team got the proper materials per their calculations and used proper technique when using tools to promote safety. To minimize the risk of injury when using tools, the suspension team took classes at TechShop such as milling, welding, and water jetting.

Design Requirements and Specifications

1. The tilt mechanism must be able to keep the pod car level on an incline or decline of 17 degrees.
2. The self-leveling mechanism must be able to move the pod car up and down in a straight-line motion.
3. The active suspension must be able to dampen vibration seen by pod car from the bogie and other systems.

The proposed design solution the active suspension team intends to make can be seen in Figure 2-1 through Figure 2-3. Referring to Figure 2-1 at the top is the tilt mechanism, then the self-leveling system, and at the bottom is the suspension.

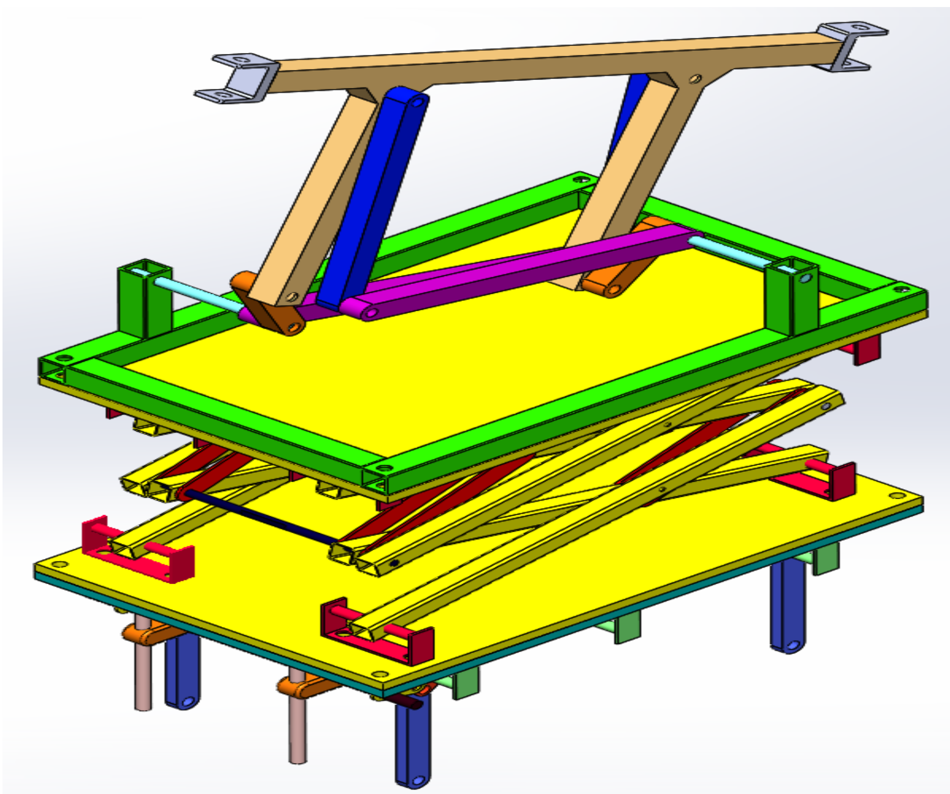


Figure 2-1. Final Design.

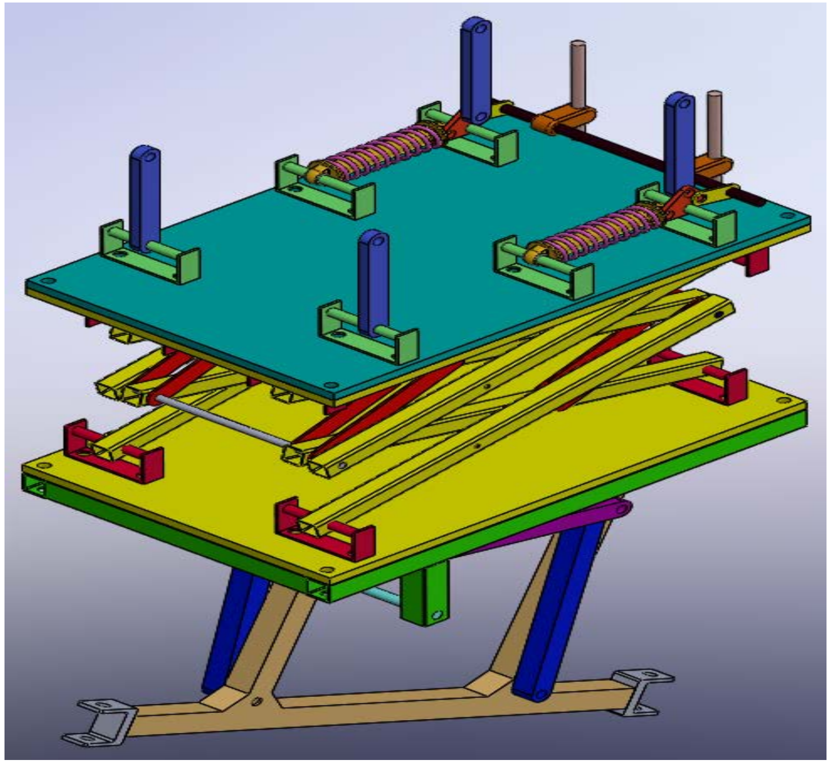


Figure 2-2. Final Design, upside down.

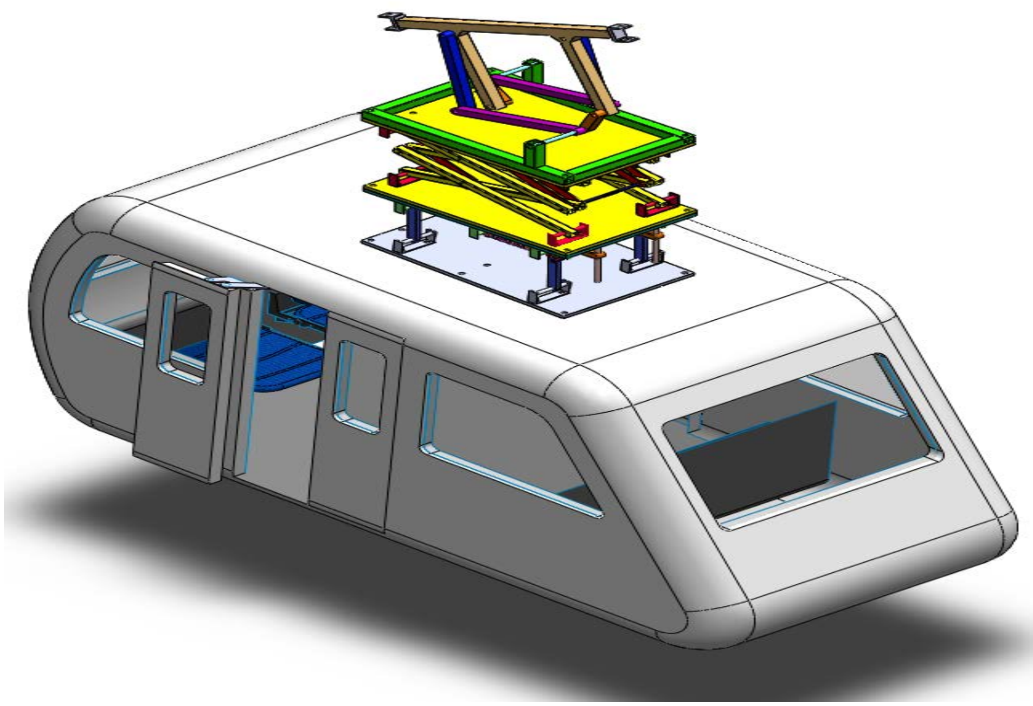


Figure 2-3. Final Design attached to Pod Car.

State-of-the-Art/Literature Review

There have been many cities across the globe that have tried to implement a suspended transportation in their town, but only a few stood out. For example, the Wuppertal Schwebebahn (refer to Figure 2-4), located in Wuppertal Germany, it stood out to us because it's a small-scale design that doesn't require a large base like that of railway stations or BART.

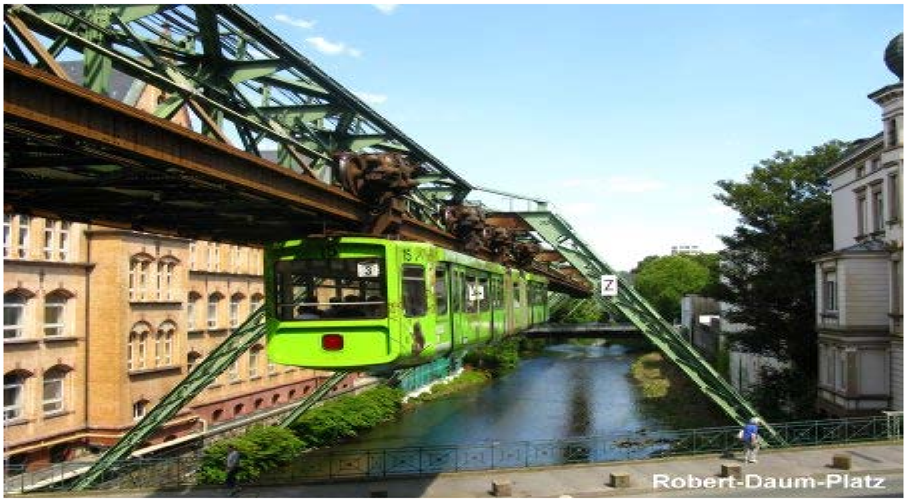


Figure 2-4. Suspended Transportation System in Wuppertal, Germany.

Another suspended transportation system that is unique and the Spartan Superway would like to model after is the Metropolitan Individual System of Transportation on Elevated Rail or MISTER, which can be seen in Figure 2-5 and is located in Poland. The pod cars are small and compact and if it is completed and it will have the ability to go up or down on an incline or decline railing.

Figure 2-5. Metropolitan Individual System of Transportation on Elevated Rail.

Even though there are currently suspended transportation system actively working, none of them utilize an active suspension that the Spartan Superway needs to operate. MISTER looks like it uses one, but there is currently no information on the suspension system for it.

Description of Your Design

The Self-Leveling mechanism, as shown in Figure 2-6, is composed of steel square tubes, steel pins, steel links, steel brackets, shaft collars, nuts, bolts, and a linear actuator. The square tubes, pins, and links were all bought from a metal wholesaler and the team used a chop saw to make it to the required length per design sketches and drilled half-inch or quarter inch holes where necessary on the links and square tubing.

Figure 2-6. Self-Leveling Mechanism on Top of Pod Car.

The bracket, refer to Figure 2-7, which is located on all eight corners of the Self-Leveling mechanism was composed of mild steel U-channel 4 inches wide, 1.5 inches tall, and approximately 17in long. The team then utilized a mill to accurately drill the holes (2, quarter-inch holes on the bottom and 1, half-inch holes on each side) in their proper places. A horizontal band saw was then used to cut each individual piece out at a 1 inch thickness. This gave us 17 pieces.

Figure 2-7. Bracket Located on all eight corners of the Self-Leveling Mechanism.

The last part that makes up the Self-Leveling Mechanism is the linear actuator. It's what will make it act like a scissor lift. The team put the linear actuator in the middle of the system (Figure 2-8) and a put a pin through both sides.

Figure 2-8. Inside of the Self-Leveling Mechanism.

The tilt mechanism couldn't be assembled to what the team designed it to be due to moment restriction and because of time constraints it was impossible to come up with a new design and manufactured by Maker Faire. Refer to Figure 2-9.

Figure 2-9. Tilt Mechanism.

Analysis/Validation/Testing

In order to prove that the design meets the requirements and specifications the team utilized vibration analysis, force analysis, and Finite Element Analysis. To find out the needed spring and damping coefficient, it is necessary to apply vibrational analysis to the suspension system, refer to Appendix B1-1 for the full calculations of the vibrational analysis. This is important because, by inputting the damping ratio (ζ) needed for the system to operate effectively, it is possible to calculate the spring and damping coefficient to achieve that ratio.

In addition to vibrational analysis, force analysis was also applied to ultimately determine the proper size of the links and pins so that it can carry the weight of the passengers without breaking apart. Furthermore, the needed torque for the motor was also gained from the force analysis (Appendix B1-2 and Appendix B1-3) applied to the system.

Through Finite Element Analysis the force analysis equations could be checked. Refer to Appendix B1-5 for full FEA results. Three different parts were tested. These three parts will hold the majority of the load in the assembly. Each part has three fringe plots: displacement, von Mises stress, and strain energy. Also included is the convergence for each FEA test. This is done to show the accuracy of the FEA.

If the mesh grows beyond a polynomial order of 6 and cannot converge usually that means a FEA cannot be completed.

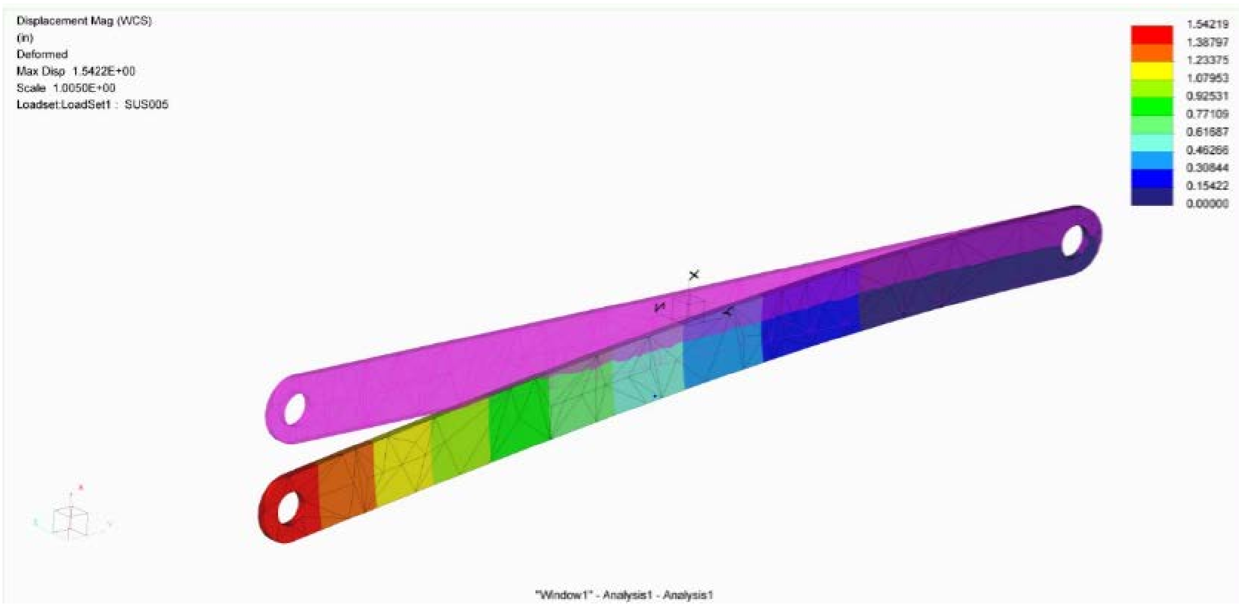


Figure 2-10. SUS005 FEA Fringe Plot of Displacement (Before).

Figure 2-10 is the fringe plot for the displacement of SUS005. This is a link that is used in the tilt mechanism. The max displacement it sees from a force equaling 150 lbs-f at a thickness of .125 inches is 1.54 inches. The material used is aluminum 6061. After review, the analysis it was deemed that the thickness of the link was inadequate. The thickness was changed to 1 inch. The waterjet cuts a maximum thickness of a ½ inch. The plan is to cut the links at ½ inch then bolt them together to create a 1 inch thick link.

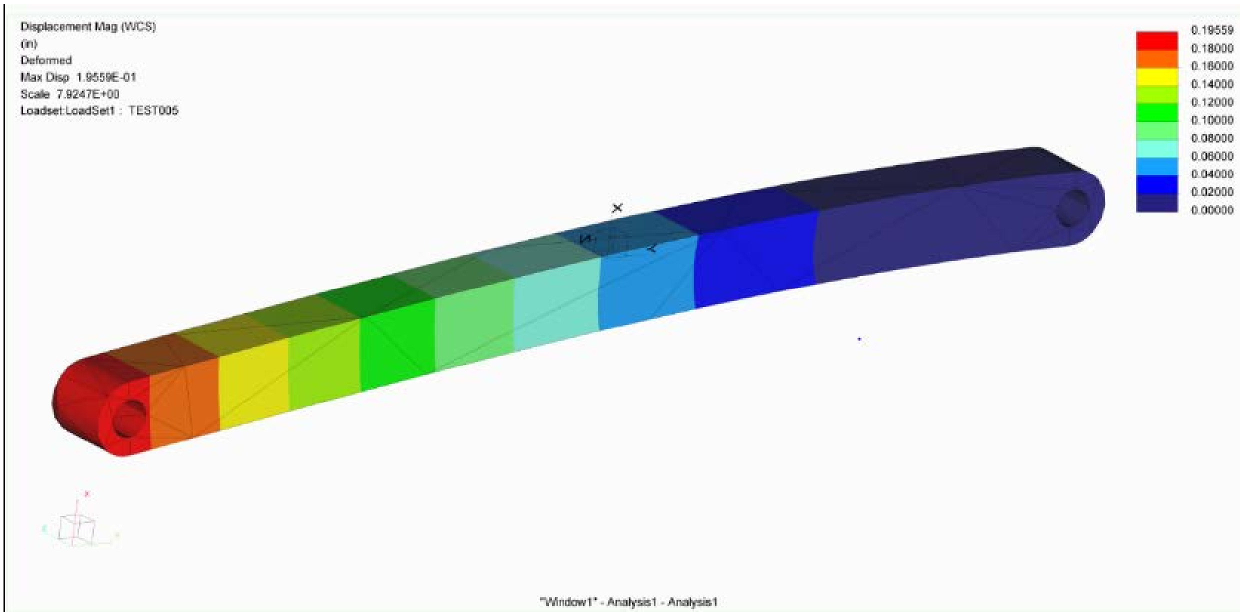


Figure 2-11. SUS005 FEA Fringe Plot of Displacement (After).

Figure 2-11 is the fringe plot for the displacement of SUS005. This is a link that is used in the tilt mechanism. The max displacement it sees from a force equaling 150 lbs-f at a thickness of 1 inch is .195 inches. The material used is aluminum 6061. After review the analysis it was deemed that the thickness of the link was adequate.

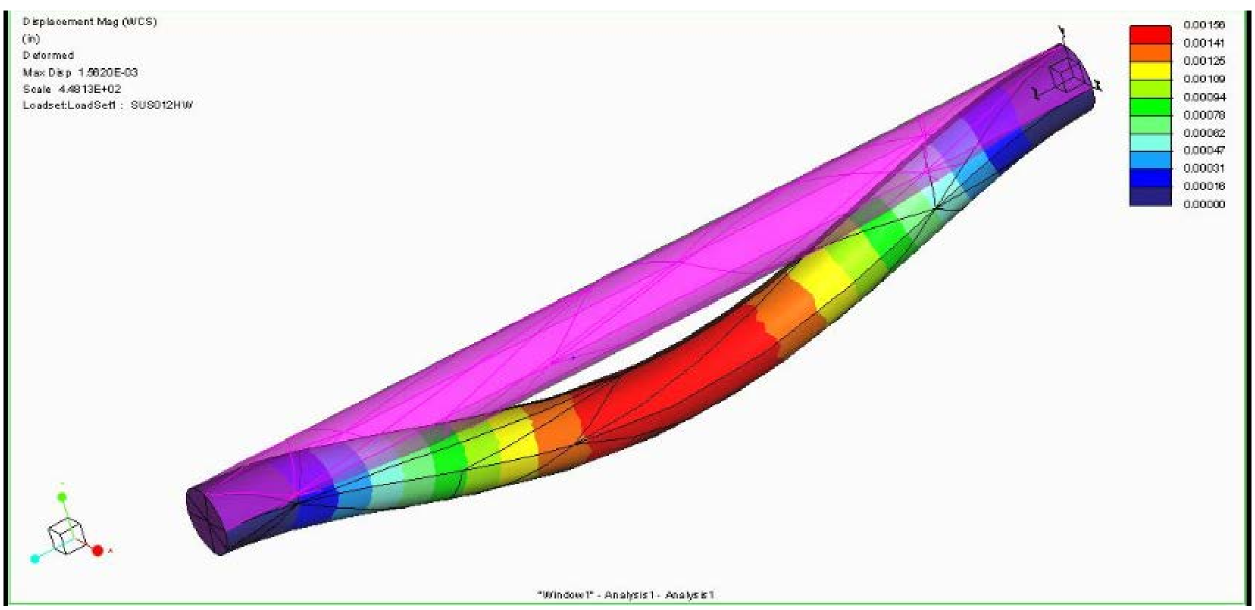


Figure 2-12. SUS012 FEA Fringe Plot of Displacement.

Figure 2-12 is the fringe plot for the displacement of SUS012. This is a pin that is used in the tilt mechanism. The max displacement it sees from a force equaling 150 lbs-f at a diameter of ½ inch is .002

inches. The material used is steel. After review, the analysis it was deemed that the diameter of the pin is sized properly.

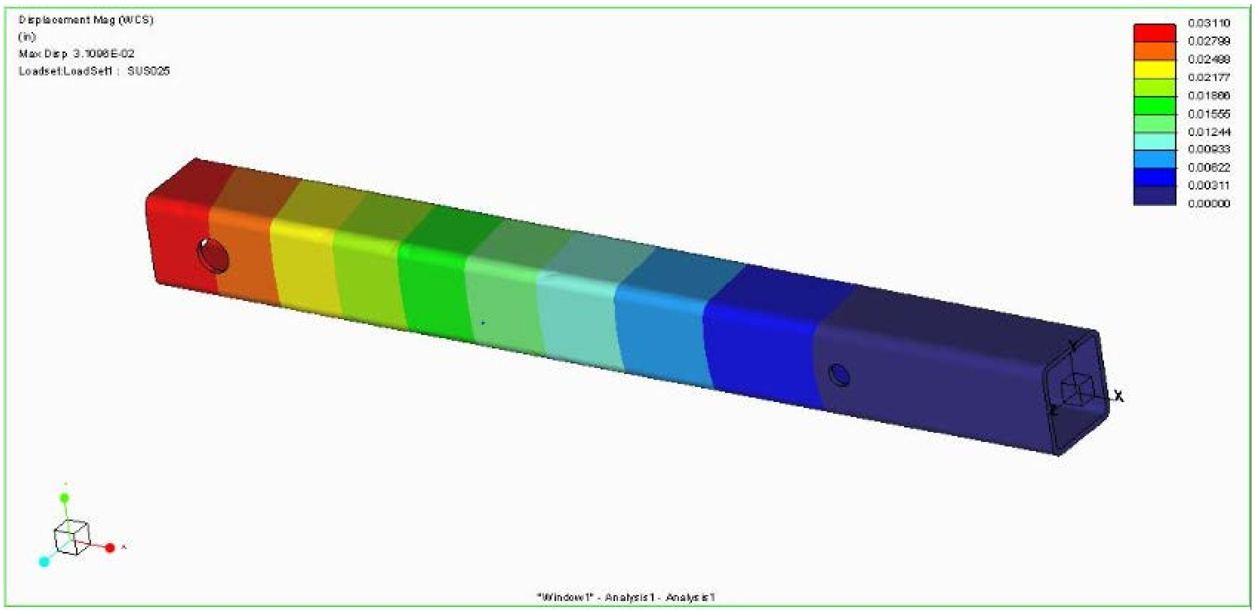


Figure 2-13. SUS025 FEA Fringe Plot of Displacement.

Figure 2-13 is the fringe plot for the displacement of SUS025. This is a link used throughout the scissor lift mechanism. The max displacement it sees from a force equaling 75 lbs-f at the ½ diameter hole in this 1 inch square tube is .0311 inches. The material used is steel. After review, the analysis it was deemed that the part will not fail.

Procedure / Instruction Manual

There are two systems that we have completed that can be assembled.

Tilt Mechanism

The tilt mechanism has six links. Each one has a different length. All the pins are made out of half-inch diameter steel round bar. There are two 7-inch pins to connect the chassis to the main follower beam.

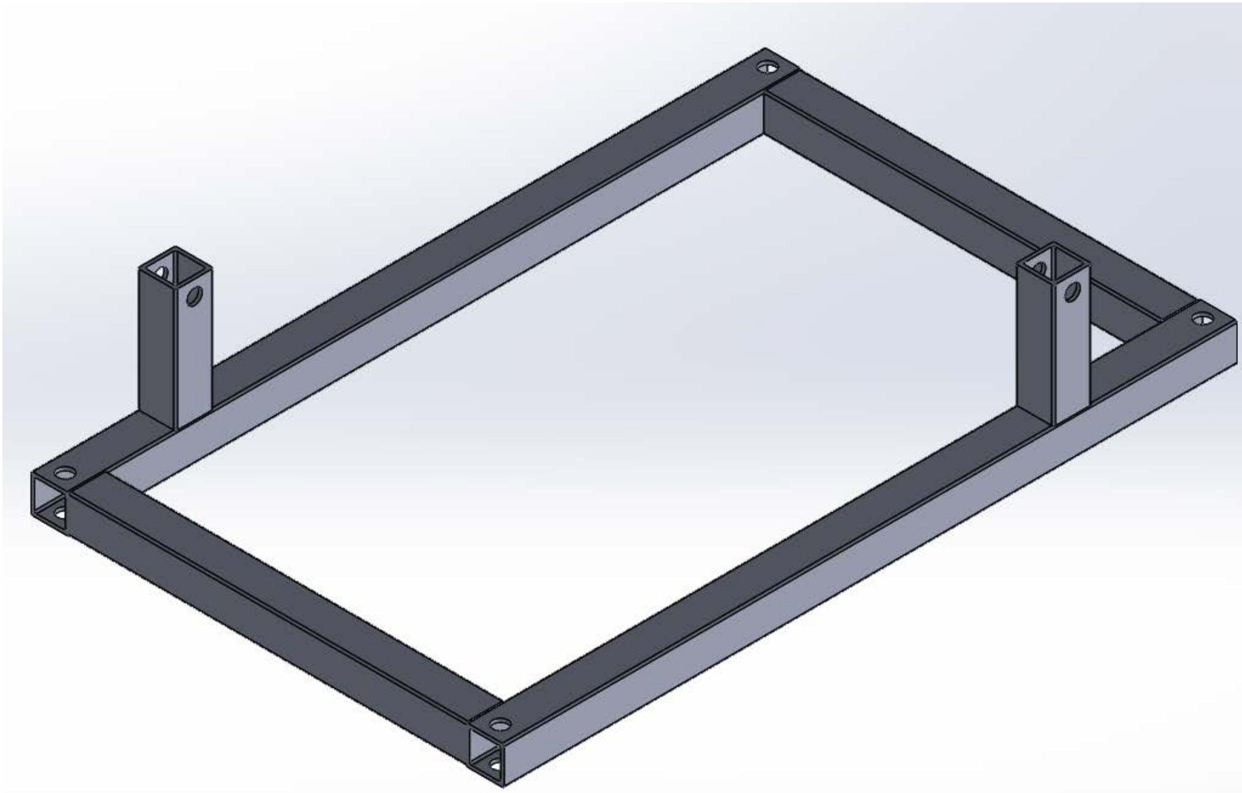


Figure 2-14. Chassis.

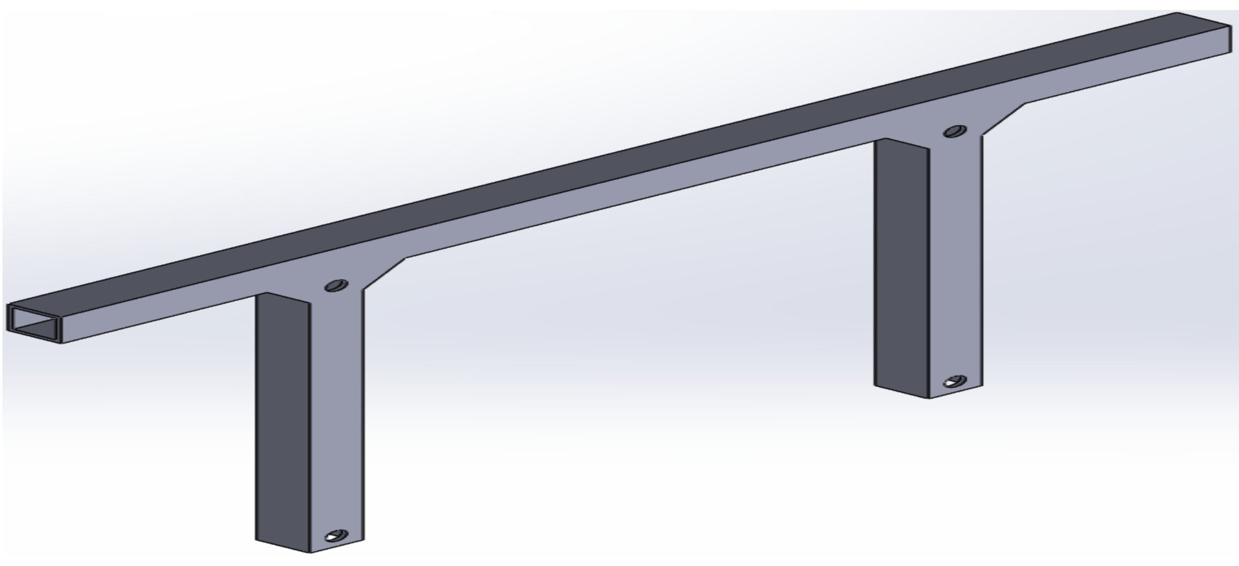


Figure 2-15. Main Follower Beam.

In order to construct it you will need six links and six pins. Four pins can be as short as 2 inches but two pins that are longer than 7 inches are needed.

First connect the long 7-inch pin to the opposite pole. Match this with the other side's pole. In order to keep them in place put a shaft collar on both sides of the pole.

Then connect the longest and shortest link to this pin. The longest link is first followed by the shortest link. I would recommend putting a thrust needle bearing in between the links to make the rotating action smoother.

Mirror these connections on the other side.

Now connect the last link to the top hole of the main follower beam and the bottom hole of that link should connect to the longest link. To hold the links together use shaft collars. It's also recommended to use thrust bearings to have a smooth rotating action.

Mirror these connections on the other side.

The following instructions are a way to mount the groschopp motor to the tilt mechanism in order for it to be motorized.

Items needed:

- 1 steel block with a ⅜ inch hole and a ¾ inch broached with a 3/16 keyway with a set screw to hold it onto the motor shaft
- 1 steel block with a ⅜ inch hole and a ½ inch hole welded onto the crank pin. Also weld this block to the shortest link
- 1 stranded steel cable wire
- 1 tensioner pulley with slotted mounting brackets.
- 1 motor mount either welded or bolted into the main follower beam
- 4 ¼ 20 bolts 1.5 inches long for the motor

Attach the motor mount to main follower beam. Then attach the groschopp motor to the motor mount using the four ¼ 20 bolts.

Attach the motor block to the motor using the set screw to hold the block to the motor shaft.

Attach the crank block to the lower hole on the main follower beam where the shortest link connects.

Attach the tensioner pulley to the side leg of the main follower beam by welding the brackets.

Run the steel cable through the motor block and secure the end with a wire nut.

Run the other end of the cable along the tensioner pulley and through the crank block.

Secure that end with a wire nut.

Scissor Lift Mechanism

The scissor lift mechanism consists of (8) 21.-inch square tubes. Each square tube has a half-inch hole drilled into it at about 0.75 inches from the edge. They have been modified to be shorter because the brackets we made caused a clearance issue.

Items needed for assembly:

- 8 21.5-inch 1 inch square tubes
- 8 5.5-inch ½ diameter steel pins
- 4 8.5-inch links with ¼ inch holes with a spacing of 7.5 inches
- One hole being ½ diameter and the second hole ¼ inch diameter
- 4 ¼ 20 2-inch bolts
- 20 ¼ 20 nuts
- 20 ½ inch shaft collars
- 1 linear actuator with a minimum stroke of 15 inches and mounting bracket
- 2 aluminum plates 24 by 15.5 inches waterjet cut to part match SUS027LB
- 16 ¼ 20 bolts ½ inch long
- 8 brackets made to match part SUS016HW
- 2 13.5-inch ½ diameter steel pins

The model shows eight links but those are thin and takes longer to make if you are using a mill. We assembled it using thicker ⅜ inch flat bar and cut down on the amount of links by half.

Using the ½ inch long ¼ 20 bolts bolt up the brackets to the plate.

You will need two 7/16 wrenches to tighten them down.

Once they are on slide the ½ diameter pins a quarter through the holes.

Take each square tube and slide those through the pin to place them close to the center of the bracket.

Make sure that the tubes on the inside have a ¼ hole for the link. The hole should be closer to the pin on the bracket.

Slide the pin to the other side.

Use the ½ inch shaft collars on both sides to hold the pins in place.

Next connect the links with the ¼ 20 bolt and nut. Do not tighten them down as that will cause binding. Just make sure that the nut has sufficient thread so it does not back off.

Then go to the other side and attach two more square tubes, linear actuator, and the two links. Use a long 13.5 inch ½ inch diameter steel pin to do this. Put the shaft collars on both ends.

When done, you should have on one side a square tube, link, square tube, and the linear actuator in the middle.

Take the two square tubes left and attach them on the other side. When done, you should have on one side a square tube, link, square tube, and linear actuator mounting bracket in the middle. Using the ½ inch long ¼ 20 bolts bolt up the brackets to the plate.

Once they are on slide the ½ diameter pins a quarter through the holes.

Take each square tube and slide those through the pin to place them close to the center of the bracket.

Use the ½ inch shaft collars on both sides to hold the pins in place.

Now the scissor lift is assembled.

Money spent on your project

Referring to Appendix B1-6, the active suspension team spent a total of $1,106.58 for parts which also include the linear actuator. Although the funding from Associated Students covered $870.54, $236.04 was payed out of pocket. In addition, a substantial amount of money was saved by having TechShop as one of the sponsors because the team had free access to their facility and tools to make their parts for the whole semester rather than paying for the parts to be made by someone else.

Results and Discussion

The implications are that we the team was able to assemble a system that will improve ride quality to the riders. Referring to Figure 2-16 and Figure 2-17, the scissor lift is able to retract and compress so that it will be accessible to all riders and it will minimize the risk of injuries due to tripping. The main focus of the active suspension team is to improve ride quality and the scissor lift is a step in that direction. The scissor lift is also something that next year's team can utilize as well and all other future active suspension teams.

Figure 2-16. Whole System with Scissor Lift Compressed.

Figure 2-17. Whole System with Scissor Lift Extended.

Conclusions and Suggestions for Future Work

We set out to build three different assemblies and combine them together to make an active suspension. However, the thing we needed most was more time to work on the project. This was mainly due to the timeline for applying for AS funding and the acquisition of needed materials after that funding was approved. Despite this we did build the tilt mechanism and the scissor lift mechanism. The mechanical suspension did not get completed because we tried to use the shocks from another project and they did not fit with the parts we had made. There was a lapse in foresight on that part but the CAD model has been updated with the dimensions of the shocks so a future team could easily produce our design if they desire to do so.

The tilt mechanism was built and tested in a couple positions to check its stability. It was seen that if it hung under gravity the weight of the pod car and other connected components would cause the shafts connected the mechanism to cantilever. This was expected but what we didn't account for was the twisting motion that would bind the links (Figure 2-18). The twisting motion makes the mechanism unstable and chaotic to operate. This design needs a balancing force to prevent twisting. The tilt chassis can be modified to reinforce the cantilever pins, however that adds weight and does not prevent the twisting motion. After seeing the pod car go through the track it would be better to not use a four-bar mechanism. The best design would be a central pin on a free hub that has one degree of freedom (able to rotate about the hub). For example, this design is seen on ski gondolas and as the gondola ascends or descends the incline the cabin tilts automatically to keep itself level with the weight of its load. I have made a concept design of how it could be implemented. The suspension system is able to be mounted underneath the central hub so it does not need to be redesigned. See Figure 2-19.

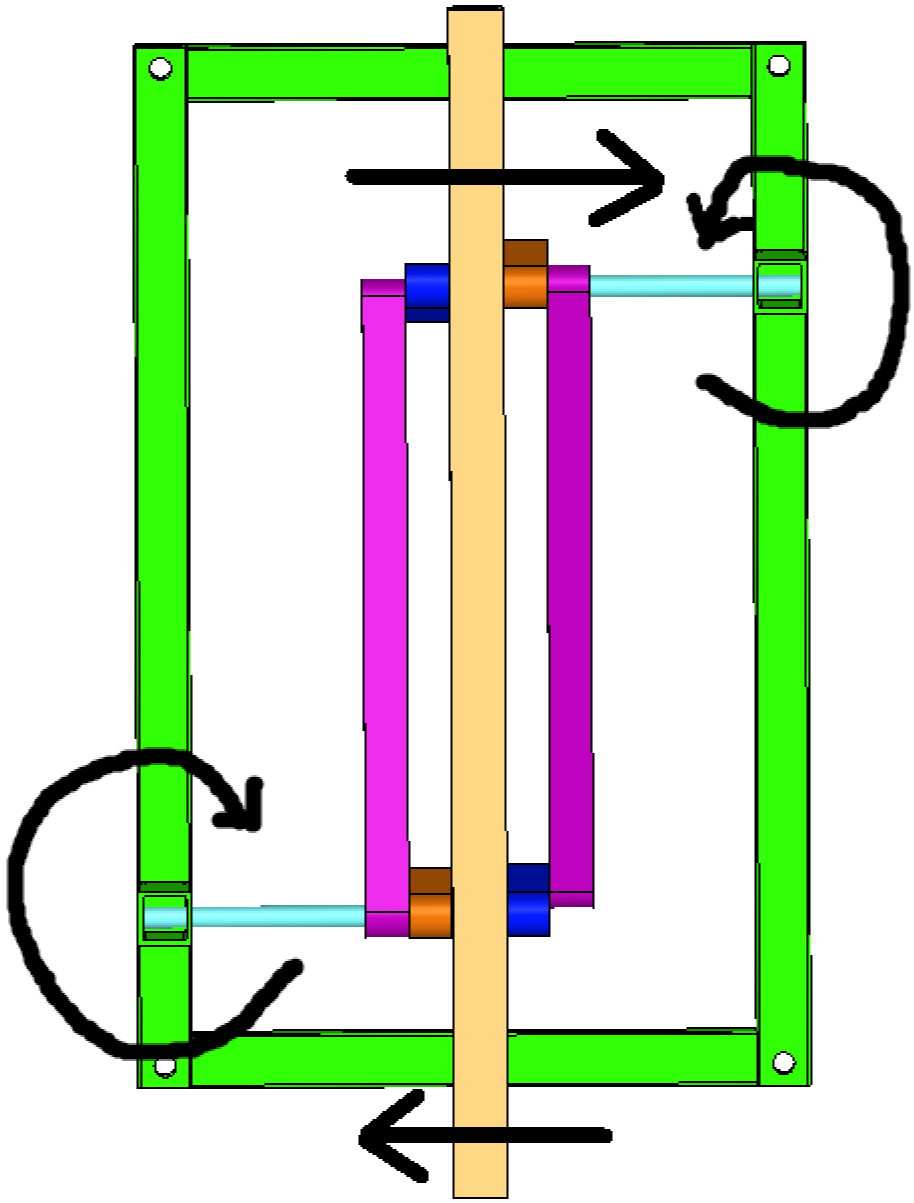


Figure 2-18. Picture of twisting.

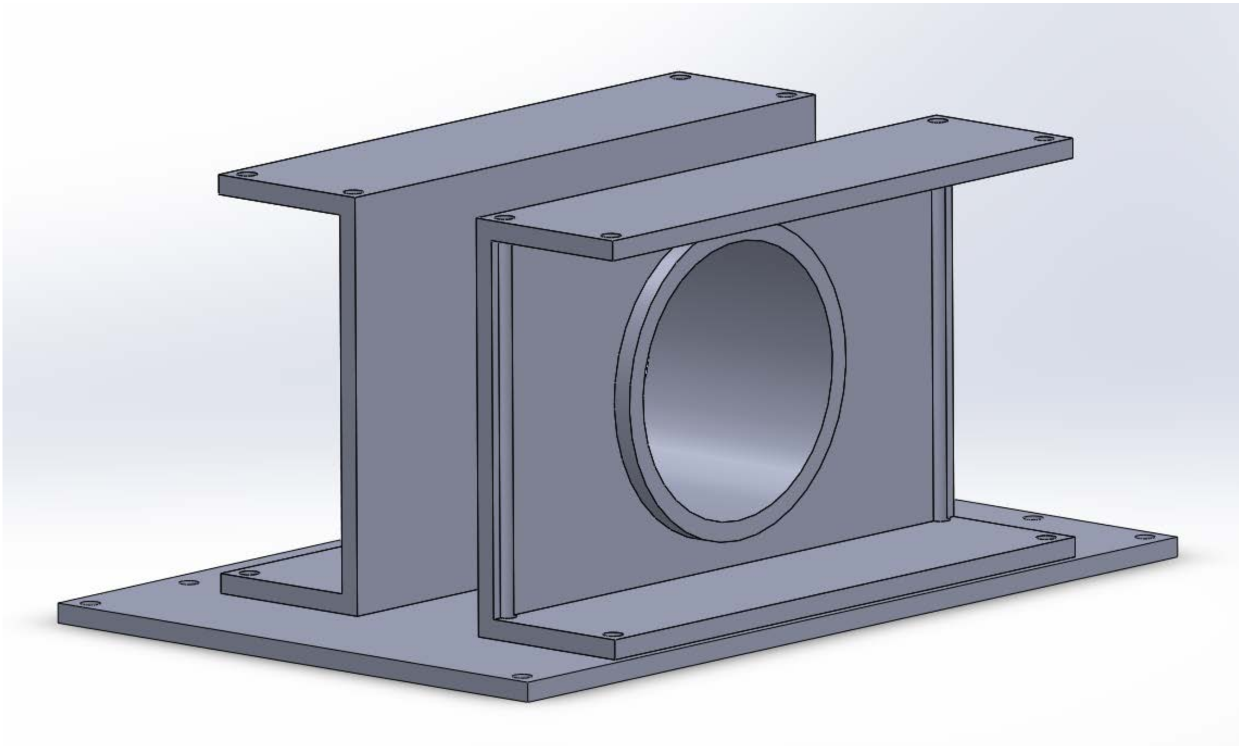


Figure 2-19. Revised Tilt Concept.

The scissor lift mechanism works as expected. It is rare to be able to build something for the first time and have it work as expected. The bars and brackets for this mechanism were the difficult parts of producing the mechanism. However, the only thing that needs to be changed is to make the mechanism out of lighter parts. The aluminum plates for the top and bottom can be cut to save weight and space. The one inch square tubing could be made out of a thinner gauge thickness. The connecting links could be made out of 0.125-inch steel plate instead of the 0.375 flat bar we used.

The mechanical suspension did not get built. Our team came up with a design but it needed more thought and time to make a suspension system that would work as intended. The biggest obstacle was the brackets we made to hold the suspension weren't made tall enough to accommodate the size of the shock. The latest design revision uses a ball joint and pull-rod suspension setup to make the suspension actuate. See Figure 2-20. This solves the folding problem of have four links with two degrees of freedom. The ball joint can only go down or up. The links that are not connected to the shocks still connect to similar pivot points. That way they can mimic the side connected to the shocks. The anti-roll bar connects both shocks together and prevents the pod car from swaying.

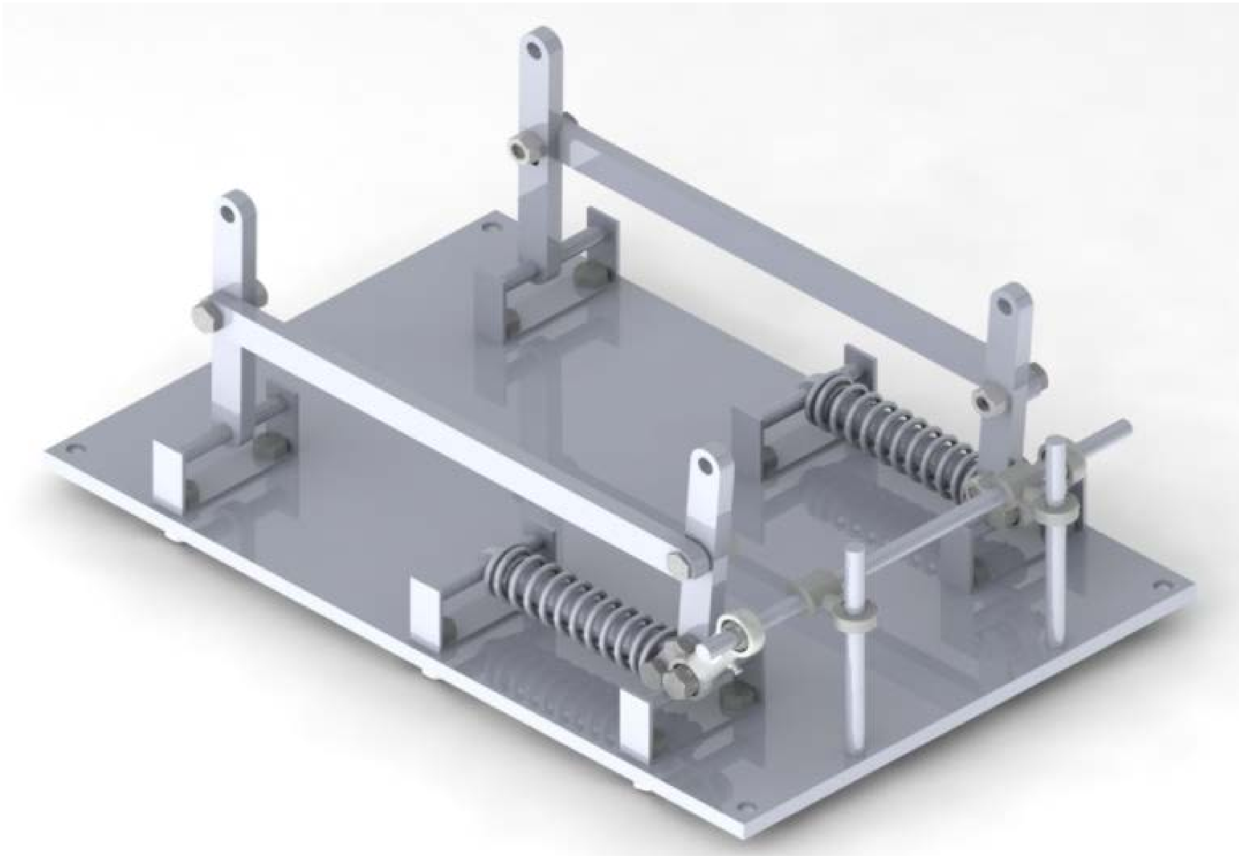


Figure 2-20. Last Updated Suspension System.

Despite having challenging obstacles, the team learned a lot of new skills that will help our engineering careers down the road. For example, the team learned how to weld part together, how to use a mill effectively, and as well as fabricating parts through water jetting. In addition to the shop skills the team was able to communicate with a cross-functional team to reach a common goal, which means communicating not only within the suspension team, but the other subteams to make sure the team's designs doesn't interfere with their designs and vice versa.

Like mentioned earlier, a great accomplishment from this project is the team going through the whole designing, verifying, and manufacturing process for the first time and be able to engineer an actual working system such as the scissor lift, while using the new techniques that were just learned this semester.

Bogie, Steering, and Failsafe

Background and context

Previous year bogie contained many flaws in the design thus changes were implemented. Certain parts presented unreliability and provided potential fail zones. The first flaw observed in the previous design was the single wheel located at the bottom did not accurately maintained the bogie oriented horizontally at all times. Also, much of the design of the front plates produced a fragile bogie. Thin strip of metal can be appreciated near the guiding wheels. If any stress concentrates in that specific spot, failure of such thin strip would be eminent. Also, the joint connecting the bogie only provides 1-degree of freedom thus constraining the bogies while it moves throughout the railway. There exist many imperfections throughout the guide way which will prevent the system from moving as expected. The same bar connecting the bogies together was located in the middle which took much of the space which can be used for other components. Next, the steering system showed unreliability. The previous bogie possesses a relatively complex and fragile design. It consisted of a four-bar linkage connected to a long, threaded rod. The problem with this design was the inconsistency of how the steering arms will turn. It is likely that the steering will not work properly and essentially fail to complete the intended motion. Also, using a long, threaded rod is not ideal for the system as fatigue stress, which starts at small notches, can become an important factor. The rod will experience a massive cyclic load and will fail prematurely at an unpredictable time. Finally, many rotating components experienced substantial amount of friction as contact of metal to metal occurred. There was not any type of lubricant or bearings to ease the friction and allow the wheels to rotate. The previous year bogie is illustrated in Figure 2-21.

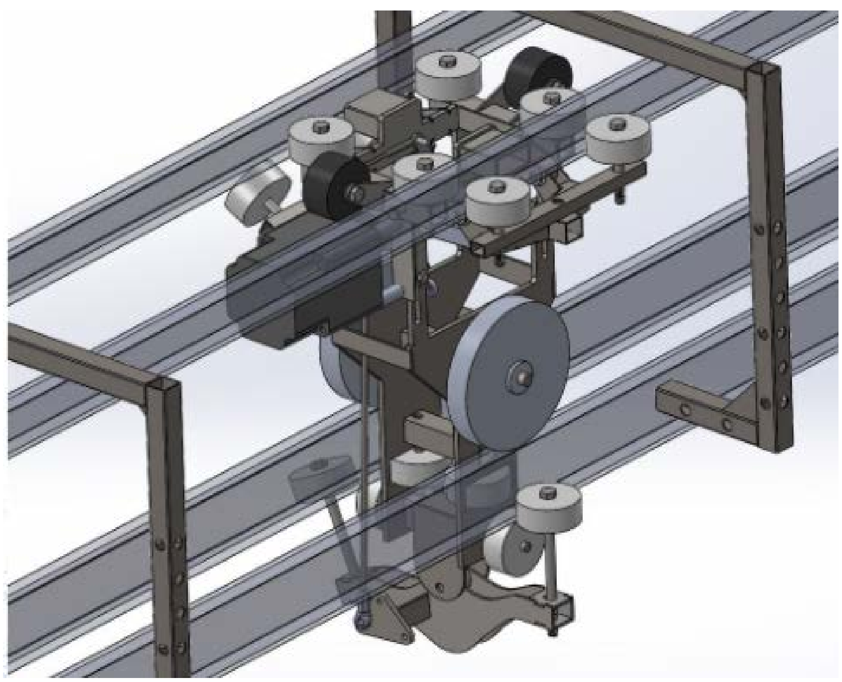


Figure 2-21. Previous bogie design. Obtained from the Spartan Superway website.

Description of the Subteam and Objectives

The current objectives are to improve the existing design of the bogie, main frame, and steering system for the solar automated transit system. A current track exists thus the specifications will be based on such a track illustrated in Figure 2-22. The bogie carrying a cabin will transverse along the track and be able to accurately switch between rails to arrive to different locations. Such locations will be the designated stations. The steering should be designed to switch within a timeframe while exerting enough force to keep the bogie on the right track. The bogie itself will accommodate other subsystems, for example propulsion and braking, thus a design with enough spacing will be produced. A failsafe system is necessary in case of any accident that may occur. The failsafe should enable the bogie to stay on the rail system if any failure may occur. Ultimately, the bogie will show the features which can be expected from the full-scale model in future implementation.

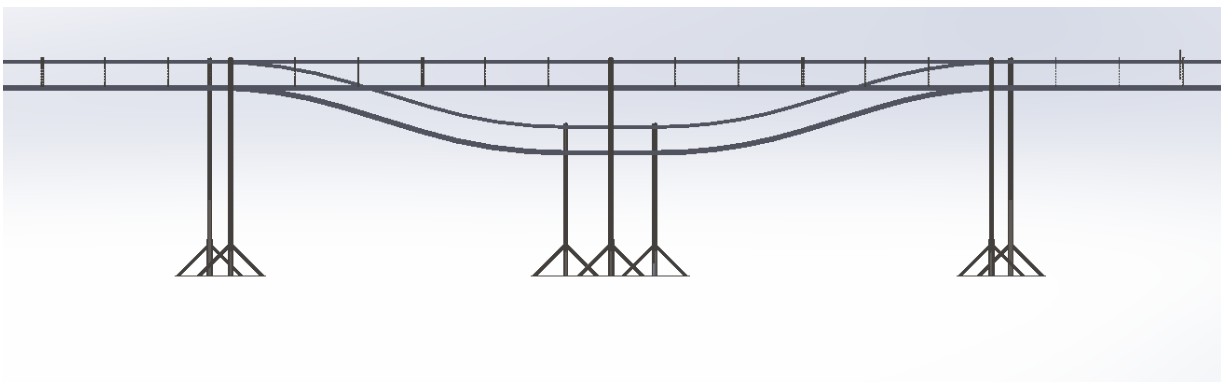


Figure 2-22. Complete half-scale track.

Design Requirements and Specifications

The proposed design ideas were implemented considering the given specifications. The failsafe system should be able to withstand a vertical force of approximately 14,400 lbs if the main wheels may ever fail as shown in Figure 2-23. Also, both bogies simultaneously should carry the weight of all components, implemented by the other subteams, which exerts a vertical force of approximately 800 lbs. The connecting bogies and weight of the different components is shown in Figure 2-24. Next, the steering design requires to undergo the full motion in approximately a 2-10 second interval. The switching interval is due to the pot car traversing along the rails at approximately 2-3 mph. The switching system is shown in Figure 2-25. Finally, the entire bogie should be able to traverse a 17-degree angle incline while allocating all other components. A connection between the bogies should be able to freely allow the required movements as the bogies travel across inclines and horizontal paths. The incline is shown in Figure 2-26.

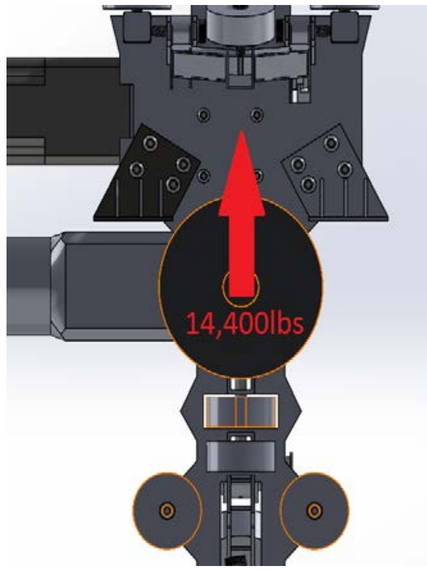


Figure 2-23. Failsafe Requirement. Image from current design.

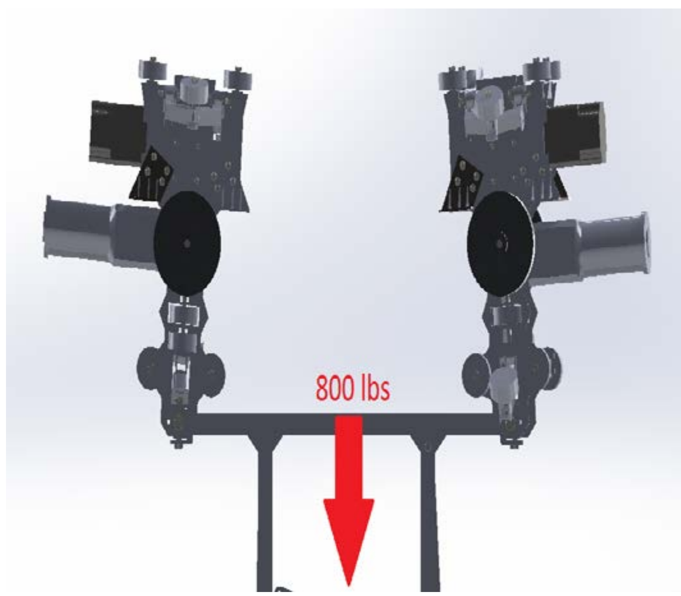


Figure 2-24. Joint design specification. Image from current design.

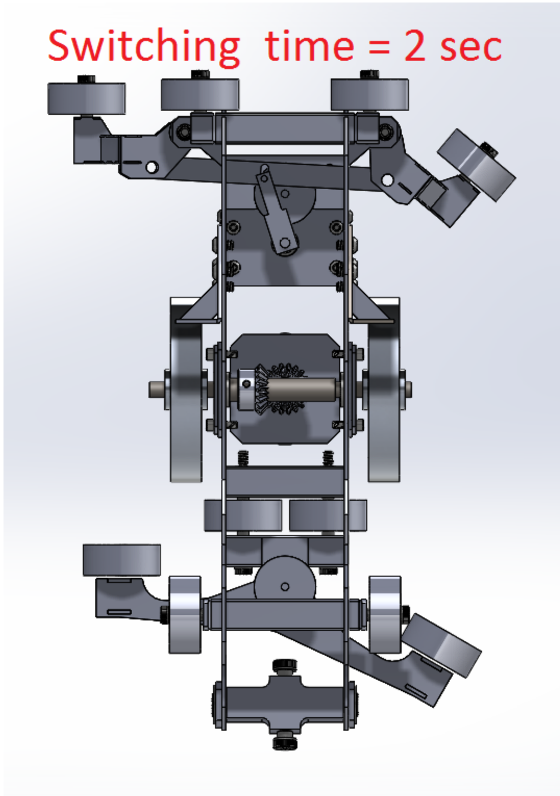


Figure 2-25. Specification for the steering design. Image from current design.

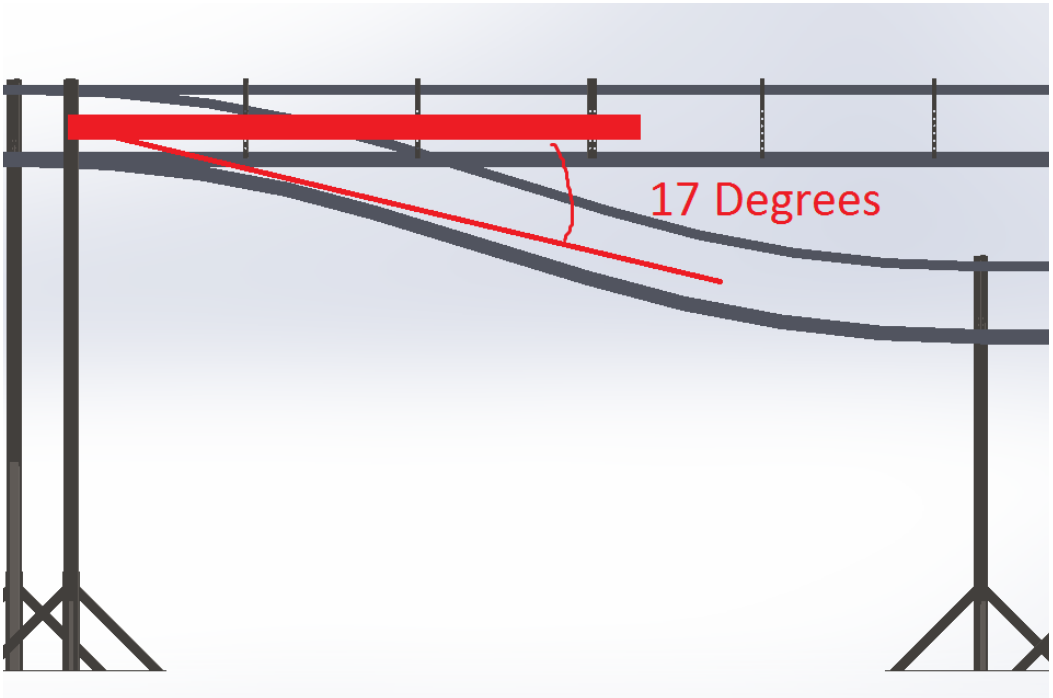


Figure 2-26. Incline specification. Screenshot from Spartan Superway, n.d, Retrieved November 30, 2016, from http://spartansuperway.blogspot.com/. Adapted with permission.

State-of-the-Art/Literature Review

An Automated Transit Network (ATN) is essentially a network of automated vehicles which travel from one station to another. The vehicle travels without making any stops along the way. Also, as mentioned by Ellis, Fabian, Muller, Swenson and Dr. Furman (2014), the network does not depend in a strict schedule and can be accessed at any time convenient for the customer (pg. 7). The main purpose of such transportation system is to relieve the massive traffic congestions which exist in urban cities. There have been many attempts in providing an Automotive Rapid Transportation system. The main idea is to create and improve the current public transportation in place. The most notable example of such system is the Cabintaxi PTR system created in Germany more than 30 years ago. The Cabintaxi was created to innovate the transportation system. The system consists of a cabin suspended above the ground on a rail system. Extensive tests were done to investigate the short comings of the new invention before implementation. All tests were successful, but according to the University of Washington (2012), the project was not implemented due to lack of budget. Another similar design was the sky train built in Dortmund, Germany. The train travels from an airport to a train station. According to H-Bahn (n.d.), in 2002 the train reduced the massive traffic congestions that generally exists in every airport. Spartan Superway, currently in process at San Jose State University, is the next step to revolutionize transportation for the future. The project will achieve implementation in urban cities by using all current accessible technological advances. Since 2012, Superway has been researching and developing models to gather all the available knowledge and construct the full-scale model that will eventually be implemented in Silicon Valley. Extensive work has been done and will continue to be executed in order to improve the design and ensure the best possible outcome. Ultimately, Automated Transit Networks will innovate how transportation is carried out while reducing traffic, reducing dependence of scarce fossil fuels, and protecting the environment.

For our subteam, research has been done in accordance to inventions with similar features. The most notable inventions that were researched were the train and rollercoaster. Knowing how both systems work will determine how the bogie will achieve the motion anticipated. Trains and roller coasters move in a rail system similar to the Superway. According to Coaster101 (2011), the different wheels ensure for a smoother ride and regulate how fast the ride will travel. Guiding wheels will ensure that the cabin will stay in place at all times to ensure the safety of the passengers. Also, a major factor is the contact friction on the rails to ensure that the bogie is able to travel up and down the inclined section of the track. The many similarities between the roller coaster and the bogie determined the materials that may be used as well as the design of our system.

Description of Your Design

The proposed solution for our system is to keep the majority of the design while improving any parts that may cause any problems. In the new bogie design, two wheels were implemented on the bottom of the bogie. This change will ensure that the bogie will stay on the rails at all times. Also, some geometry changes on the faces were made to ensure that the bogie will not be as fragile or break as easily. The top steering wheels were changed to a single wheel in order to remove any constraints that may occur during turns. Next, the joints that hold the two bogies together were changed to a 2-degree of freedom joint to allow for more room of movement as the track will never be perfect. In addition, the joining bar was relocated under the bogie to allow more spacing as well as decrease stress. Next, steering was entirely redesigned to meet specifications. For the new design, a gearbox, pulleys, and timing belt are implemented. A gear ratio is chosen to effectively increase output torque while maintaining a reasonable switching speed. The gearbox and pulleys are located at the top and bottom steering arms. Both steering arms are connected by a timing belt. The steering design ensures that the bogie stays on the track, thus increasing safety and reliability. Finally, the failsafe component is designed to ensure that the bogie does not fall through in case of any complications. A failsafe is placed on the side of the bogie to withstand an impact force in case the main wheels break by shearing stress. The failsafe will withstand the impact force and maintain the bogie on the track, thus preventing any injuries the accident may cause to the passengers at the moment of failure. Figure 2-27 and Figure 2-28 illustrate a closer look at the steering gear box and failsafe design.

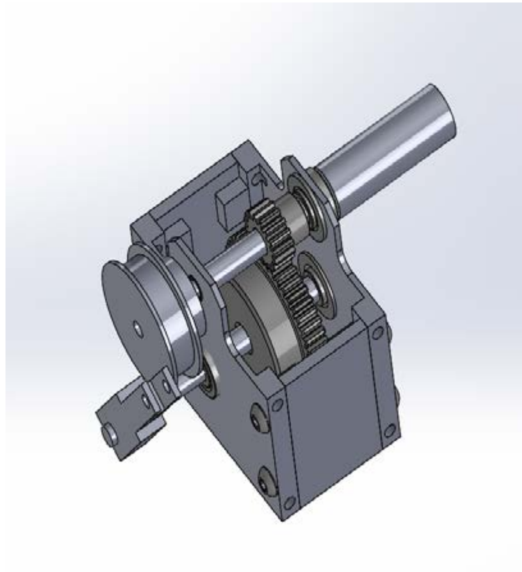


Figure 2-27. Gearbox system.

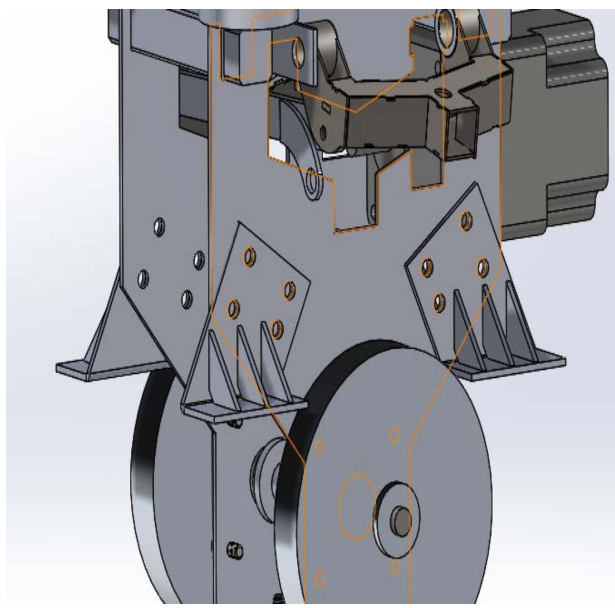


Figure 2-28. Failsafe system.

Analysis/Validation/Testing

First, the analysis perpetrating the steering was based on Newton's second law (F=ma). This enabled us to find the lateral force acting on the bogie due to the weight and motion of the system. Once the force was found, demonstrated by Figures B2-15 and B2-17 (Appendix B2-2), the steering design was calculated according to the value of such force to overcome and maintain the system on the rails. The steering will be required to overcome such force and thus the gears were implemented. All gear equations and calculations are shown in Figures B2-16 and B2-18 (Appendix B2-2). A motor, used in the previous design, has a designated holding torque as illustrated in Figure 2-21 (Appendix B2-3). According to Norton (2006), gear ratio will be able to increase the amount of torque output by the system according to the equations. The idea of the new design is to be able to increase the torque while still obtaining a permissible switching speed. In other words, as the torque increases, the speed decreases. Also, we aspire for the two steering arms to reach the rail simultaneously, thus the gear ratios are different for both arms as they will travel a different angular distance. The bottom steering arm will travel 2 times the distance compared to the top arm, thus a greater speed is required in that location. For the failsafe, the analysis was done using FEA demonstrated in Figure B2-19 (Appendix B2-2). A force was applied to the current design to visualize and understand whether the design choices will withstand the impact force. Such force was found by applying Newton's second law. The mass used was the entire system mass while the downward acceleration is the value of gravity. The up-stop wheel of the bogie was implemented to keep the bogie perpendicular to the track at all times. Using one wheel will only constrain the upward translation thus allowing the bogie to freely rotate. Adding another wheel and separating them to a desired distance will constrain the upward motion as well as the rotational motion to some degree, thus preventing the bogie from coming in contact with the roof of the track. For the joining bar connecting both bogies, bending stress was used. The bending forces on the beam are caused by the weight of the cabin and any sub-system connected directly. The formulas used for such design choices is pictured in Figure B2-20 (Appendix B2-2). If the forces provide a large enough tensile stress on the joint, the cabin will fall, therefore the beam is designed to withstand and prevent any failure. Testing was also done using the wood prototype. Dynamics of the system were tested and later adjusted to ensure that the design was correct. Also, once the new was constructed and assemble, all components were tested manually to ensure all specifications were accomplished.

Procedure / Instruction Manual

The following instructions explain the assembly and disassembly of the bogies. Figure 2-29 shows most components, except for the gear box, required for the construction of the new design. The entirety of the design can be assembled and disassemble using Hex L Wrenches. There is not a right way to start as the parts can be assemble and disassemble in any order. One suggestion to avoid some difficulty is by assembling the gearbox and securing it in the bogie before assembling the failsafe. If the failsafe is in place before the gearbox, interference occurs and it is fairly difficult to secure the gearbox in place. If the gearbox is forced to fit in place while failsafe is secured, the pulleys may break. Figure 2-29 to Figure 2-33 show some of the components and how they assemble. The rest of components are self-explanatory and assemble in a similar manner. Appendix B2-1 contains sample detail drawings of every component.

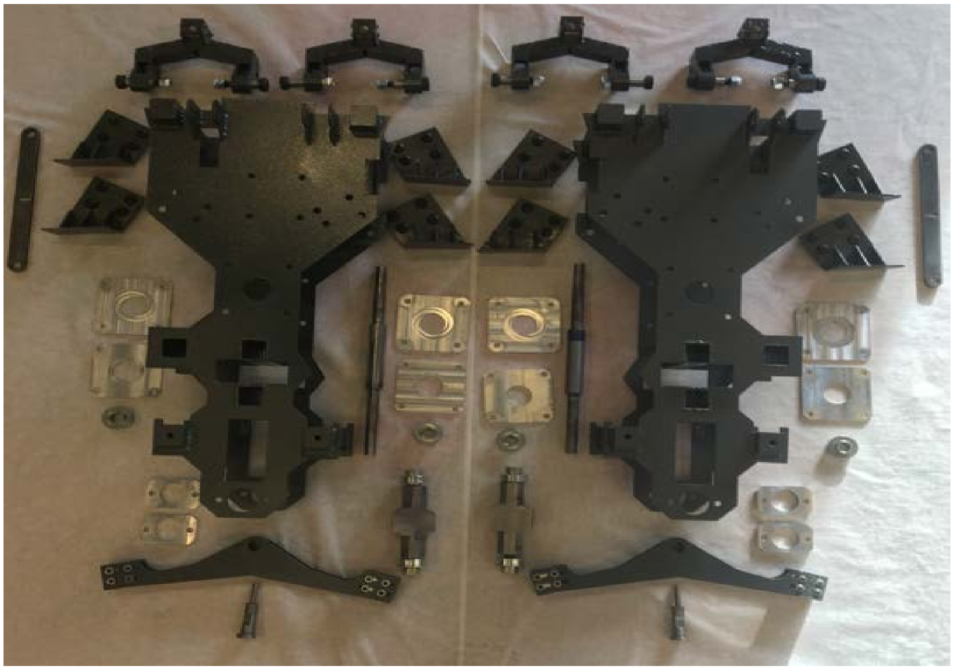


Figure 2-29. Disassembled bogie.

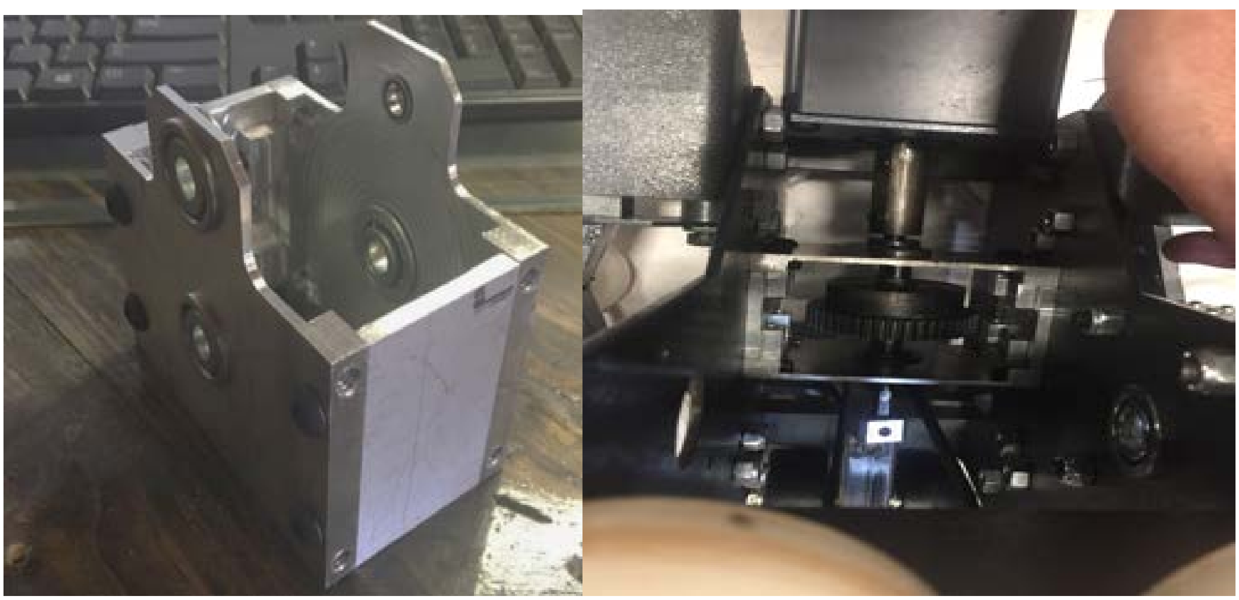


Figure 2-30. Gear box assembly.

The gear box contains four screws in each face and can be easily removed from with the tools. The bearings inserted in the holes can be removed by hand and no tools are necessary. Pulleys and connection to the bogie are secure by a screw which secures the shafts via a keyway.

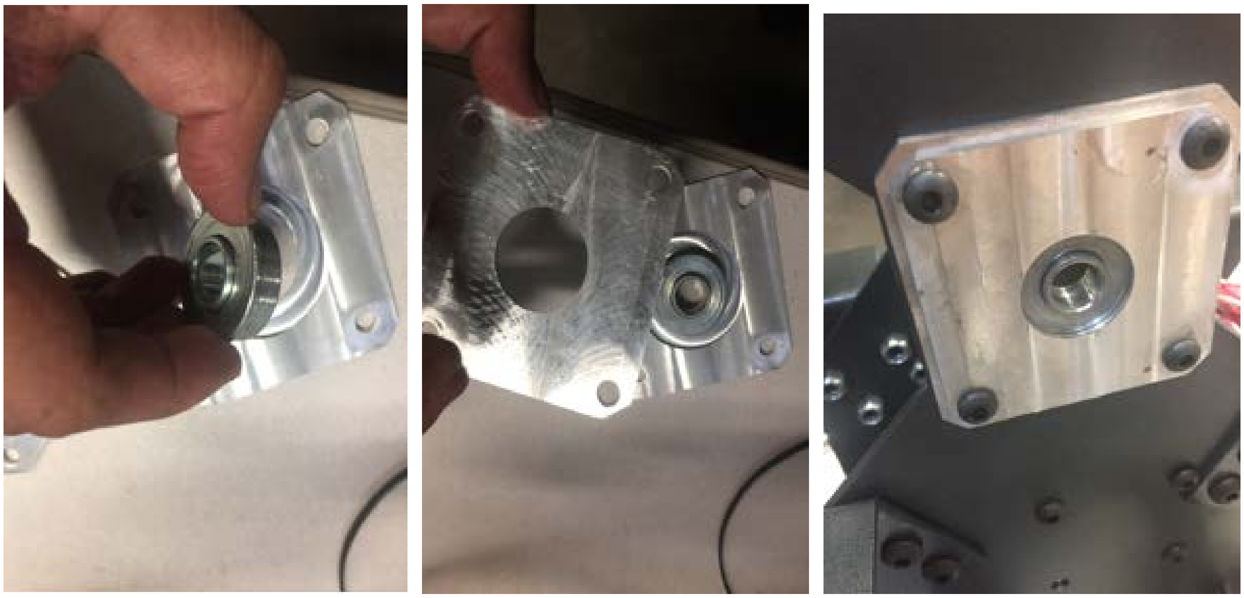


Figure 2-31. Main wheel bearing assembly.

The bearing always enters the hole of the bogie from the outside. First, the big plate is positioned, After, the bearing is situated on the plate. Finally, the small plate is situated on the bearing and secured onto the bogie.

Figure 2-32. Failsafe assembly.

Failsafe is secure by Allen key bolts and nuts. The only concern when securing this specific component is to secure it in place as designed. The bottom of the part should be vertical to ensure that the bogie would not fall if any failure may occur.

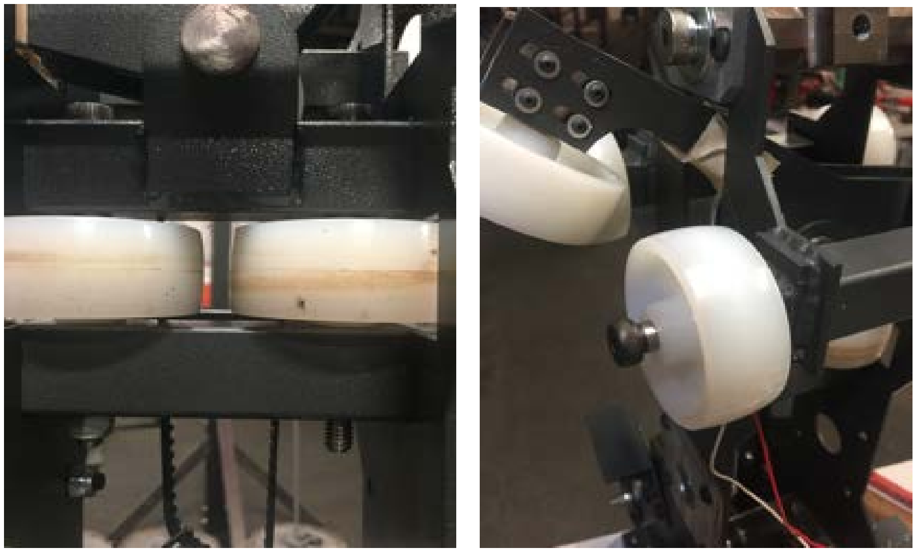


Figure 2-33. Wheel assembly.

The wheels are connected by shoulder bolts. Most wheels are connected by small bolts except for the guide wheels. The guide wheels contain a longer shoulder bolt and a lock nut to maintained in place as secure as possible. The rest of the wheels are secure in holes which have been threaded.

Money spent on your project

Table 2-1 shows the expenses in order to construct the design envisioned. Approximately 30% of expenses went towards fasteners, tools and machine process. The rest of the of expenses were used for all materials. Some money was saved by utilizing the same motors from previous years. The total money spent was $1403.8 dollars.

Table 2-1. Project expenses.

Part Description	Product ID number	Pricing (each)	ad Time of servi	Quantity	Why is Important?
48 Tooth Gear	**6325k860**	$41.37	Not necessary	2	Prevets bogie from Falling
16 Tooth Gear	**6325k820**	$17.26	Not necessary	2	Prevets bogie from Falling
16 Tooth Pulley	**1277N46**	$9.44	Not necessary	2	Prevets bogie from Falling
28 Tooth Pulley	**1277N53**	$14.73	Not necessary	2	Prevets bogie from Falling
Sheet metal (12x24x 1/8 in)	**1388K474**	$97.44	Not necessary	4	Main boigie Frame
Timing Belt	**6484K241**	$7.94	Not necessary	2	Steers the Bogie
Connecting bogie beams	**6527K174**	$17.76	Not necessary	6 Ft	Main bogie Frame
Bearings	**6383K234**	$8.72	Not necessary	6	Holds main hub wheel on
Bearings	**6383K215**	$8.72	Not necessary	2	holds shaft for gear box
Bearings	**6384K344**	$7.96	Not necessary	1	holds shaft for gear box
WaterJet Cut Time (mins)		$3.00	Not necessary	120	all sheet metal parts are cut
Billet Steel (Lbs)		$0.80		50	need for joint, etc.
Total		**$1,066.72**			

Fasteners (Jorge mcmaster)		146.46
Fasteners (Jorge Fastenal)		73.2
Gears and Pulleys (Jorge)		181.48
drills bits		43.49
endmills/taps		60.89
deburring tool		13.04
		337.08

Results and Discussion

Achieving all projected goals means the project will be able to move on to the next steps. The working bogies and steering successfully proves the intended concepts. By achieving all goals, half-scale is able to demonstrate the advantages of the project as a whole and new sponsors will be able to visualize the concepts used. Any desired demonstrations can be schedule to exhibit the mechanics to all possible sponsors. The working bogie and steering enables future groups to improve a working system which already achieves all specifications. Future teams do not have to redesign the entire system and will be able to simplified certain aspects of the design to reduce failure. The system can now be simplified and components may be added to improve reliability. With the successful design of bogie and steering, testing can be done which will enable to locate any failures that can occur. All dynamic aspects may be working, but other unexpected failures may appear. For example, some parts of the bogie or steering possibly fails due to unforeseen stresses can be located by testing. Once the failure zones are identified, redesign of such parts can be performed accordingly. Finally, other working bogies can be manufactured if necessary. Ultimately, some characteristics of the half scale bogie and steering can be used and implemented in the full-scale model.

Conclusions and Suggestions for Future Work

Overall, the Half-Scale Bogie, Steering and failsafe team achieved all objectives. The bogies are able to traverse the entirety of the rail system. One of the main concerns was the 17-degree angle incline. The new bogie design can accurately traverse up and down the incline without experiencing any interference. Also, the bogie is able to allocate all the other subteams components. We were able to provide enough spacing which is used to connect other functioning parts such as suspension. Next, the bogies are able to carry the entirety of the weight. The steering system switches accordingly to specifications. The gearbox used provides enough movement in order for the steering arms to traverse the rotation necessary and switch between rails. Also, the gearbox, provides sufficient torque to maintain the bogies on the rail system. It is necessary to provide enough torque as the other subteam components exert a lateral force which may cause the bogie to fall off the railway. Finally, the failsafe proves accurate as it will prevent the bogie from falling presuming the main wheels fail. All specifications were achieved and a fully functional bogie was constructed.

Although the specifications were accomplished, many improvements can be implemented. While testing the bogie in Makers Faire, one of the shafts from steering fail. The shaft broke as the top steering attempted to move above the expected limit. The failure was due to the torque exerted by the gearbox which caused a fatigue failure. The full motion of the top steering should be adjusted to prevent such failure. As the gears wanted to rotate a greater distance, a massive the stress was exerted on the shaft causing failure. Also, many components should be simplified. The aluminum blocks can be replaced by steel blocks as the threads in aluminum are not reliable. If possible, the connections should be replaced entirely to a simpler design. One of our goals was to provide a simple bogie which can be assemble and disassemble easily. The easy assembly also introduced many more failure zones. Many of the connections can be improved which will increase reliability. There is definitely space for improvement.

Ultimately, this project developed many of our skills. One of the main skills is to communicate with everyone in the group. By communicating with other subteams, it ensures all components and ideas fit well together. Each idea is dependent of one another. Also, time management is of upmost importance. Planning and arranging deadlines makes sure the job gets done in a timely manner. Finally, many technical skills were learned. Solidworks skills were enhanced. We familiarized with the usage of different tools and machine processes. The entirety of the project did not only teach us technical and non-technical skills, but portrayed the future of transportation. Spartan Superway will improve the transit system which will enhance the lifestyle of our society.

Braking and Propulsion

Background and context

The purpose of forming a braking and propulsion team this year was to solve issues from previous years' half-scale team. Last year's direct drive hub motor design for half-scale propulsion couldn't traverse along the railway incline. Additionally, the braking discs from the previous year didn't actuate appropriately as well. As a result, the half-scale braking and propulsion team was formed to solve both issues of making sure the bogie can move and stop adequately along the guideway designed.

Description of Subteam and Objectives

As a team, we were in charge of first designing a system that would be able to move the bogies along the track, which specifically means being able to climb the 17-degree incline of the track's station. Additionally, we used a target of a 600lb load, which the previous year seemed to use as their targeted weight. For braking, we decided that we would want to be able to stop the bogie down the same slope, while still developing a failsafe for the system that would activate when the system suffers a power loss. At the same time, the braking system would have to be able to stop the same 600lb load within 50 feet. Just to keep the stopping range comparable to a car traveling at 10mph.

Our team's goal for propulsion is to design a mechanical system that could climb the 17-degree incline of the track with a 600lb load. Additionally, we aimed to make sure that the motors would operate under 400 watts of power, drawing as little current as possible to theoretically run them just off the solar panels in the shop. The brakes did not have the same power consumption requirements that the propulsion motors had; the brakes in turn aimed to stop the bogies while they were descending the 17-degree slope at 10mph and within 50 feet. Additionally, because the brakes should act like a failsafe, the aim was that the in the event of power loss the brakes would activate, stopping the wheels from turning. For this reason, we felt that the braking system should be spring loaded, only being released when a stepper motor pulls a cord that would release the brake.

Design Requirements and Specifications

There are several objectives for our subteam, the main purpose of the propulsion and braking team is to make sure that the pod car can move and stop in a safe manner. Specifically, for propulsion of the bogie system, the main goal was to be able to move a 600-lb. load up a 17-degree slope. Within the scope of this project, speed at which the bogie moves aren't necessarily all that important. However, the power required for the bogie to move along the incline was an important factor to consider. The inverter that had been used for the bogie could only produce 1500 watts. Of the 1500 watts, power is needed for propulsion, steering, braking, suspension, and any other electronics that are needed for the pod car to move along the track. For this reason, it was decided that no more than 400 watts would be dedicated to propulsion, limiting the amount the voltage and current to something that is both reasonable and safe to work with. Additionally, the system was to be made in a way that could allow for regenerative braking to be implemented later. For braking, we wanted to be able to stop the same 600lb load within

50 feet while traveling down a 17-degree slope at 10 mph. Propulsions goal is certainly achievable and measurable by loading the bogie to weight and allowing it to move freely along the track. However, the objective set for braking proved to be a difficult task as the bogie system space limitations doesn't allow for existing, basic braking systems to be implemented within the bogie.

State-of-the-Art/Literature Review

Automated Transit Networks are not a new idea. One of the earliest systems, the Morgantown personal rapid transit system, has been around since 1975 and continues to operate to this day (Raney 2004). Morgantown in particular is an electric transit system with 5 stations and several miles of track going through West Virginia University and Morgantown West Virginia, with a picture of a station seen in Figure 2-34. It was proposed initially as a demonstration of a personal transit system, but has be in continuous operation since 1979 due to its success (Raney 2004). Since then, the next system that would be considered an ATN is the Parkshuttle Rivium metro-feeder outside Rotterdam, developed in 1999 (Furman 2014). While the rest of the ATNs are more modernized, being developed in 2010 and onward. These systems are electrical, comprising of one main control station, have several passenger stations, and are powered with batteries or using a wayside system (Furman 2014). Because of the automation, and relative stability of the system, small teams are needed to maintain and control the system compared the large number of people involved with conventional transportation solutions. Control is key for these systems to be successful, being able allowing the passenger cars to go down the system's guideways, keep from running into other cars, and being able to pull into a station and go the correct direction as it leaves the station. Additionally, the stations allow for quick loading and unloading of passenger as well as direct passenger to as to what stations the cars are going to. Additionally, due to ATNs being electrical transportation, they have the potential of being eco-friendly by using renewable energy compared to conventional fossil fuel powered vehicles.

Figure 2-34. Morgantown Personal Rapid Transit.

Since 2012, the SJSU has been working on an ATN called the Spartan Superway, which is a hanging pod car personal rapid transport system. The Superway also a ATN that is solar-powered, using solar panels that are mounted to the top of the track in order to power the podcars though a wayside system. The first three years were dedicated to the implementation of the 12th scale model as well as the full-size model (Furman 2016). With the four year (last year) introducing the half-scale model. In the first three years, more development was put into the small-scale model, for controls systems and software systems to control multiple cars. While the Full-scale team worked on the pod-car design, suspension and steering. Last year in particular, active suspension was added to the full-scale model and the twelfth scale expanded in size, requiring an additional robustness to the system. While the half-scale team was presented with the challenge of making a new system, loyal to the design of the current full scale design, while compacting everything. Unfortunately, not many of the half-scale systems worked in the first year, forcing a redesign of all the major systems.

Due to this project being focused on the braking and propulsion for the half-scale model, more focus will be put on their developments. There is nothing particularly special about the design of the propulsion and braking systems for the bogie. The original braking ideas were centered on using a disc brake which can be found on most conventional cars. Yet, the design evolved to something using a standard brake caliper on the rails that support the bogie. This way avoiding the tightness between the bogie's sides and giving more room for propulsion and steering to have their gearing. For propulsion, initially different

DC motor were searched for through various sites for their torque output and overall wattage. The team initially looked at using the same system that the propulsion team use last year, as seen in Figure 2-35, but that system was dismissed due to problems with power requirements, torque requirements, and overall ease of implementation. For this reason, the design was changed to use a DC motor that had been gear properly to move the system. The advantage being that a DC motor could potentially be used for regenerative braking to help recharge the bogie as it descends downhill instead of relying solely on solar power and the batteries. In general, there isn't much that can be considered as state of the art when dealing with a standard car braking system and a DC motor for propulsion.

Figure 2-35. The previous year's propulsion design.

Description of Design

The design for propulsion was determined prior to the beginning of the spring semester. Compared to the previous year, the propulsion system was switched to a two-motor system, directly driving the weight bearing wheels of the bogie. This design was to maximize the amount of traction that the wheels driving the system will have through the forces already present, instead of using a system that artificially increases the normal force on the wheels. The propulsion design consists of using two 24 volt, 120-watt motor between the two walls of the bogie. This motor has a gearbox reducing the motor speed 160:1. The motor is mounted perpendicular to the drive shaft allowing for miter gears to change the direction of motion allowing for the wheels to be driven. The miter gears are adjustable on both the motor and drive shaft, allowing for adjustment of the gears so they can mesh correctly. (Figure 2-39). There will be 90-degree miter gears interfaced to translate rotation of the motor shaft to the driving shaft.

Additionally, failure calculation for the miter gears transmitting the force resulted in a safety factor of 1.2, which is appropriate within the design.

Teaming up with the bogie design team, for the fall semester, gears and motor mounts were 3D printed to provide a proof of concept of spacing within the bogie in preparation of manufacturing and integrating the propulsion design, shown in Figure 2-36. A few components of the propulsion design, such as the key-stock and wheel bearings weren't shown in the prototype design as they would be time-consuming and negligible in determining the spacing limitations within the bogie.

Fabrication for propulsion consisted of cutting steel to create mounts (Figure 2-38) and lathing a keyway for the driving shaft (Figure 2-39) and output motor shaft for the miter gears. Furthermore, the design for propulsion was already determined early in design phase, however there were various issues in attaining the miter gears and propulsion motors. The propulsion motors are the most expensive components within our design, mostly due to the gear reduction needed to provide torque for the bogie to climb along the guideway. The expense was driven higher with the miter gears included, and ultimately funding became an issue within our designs. As a result, the motors and miter gears arrived late in April and ultimately delayed testing of the bogie system on the rail.

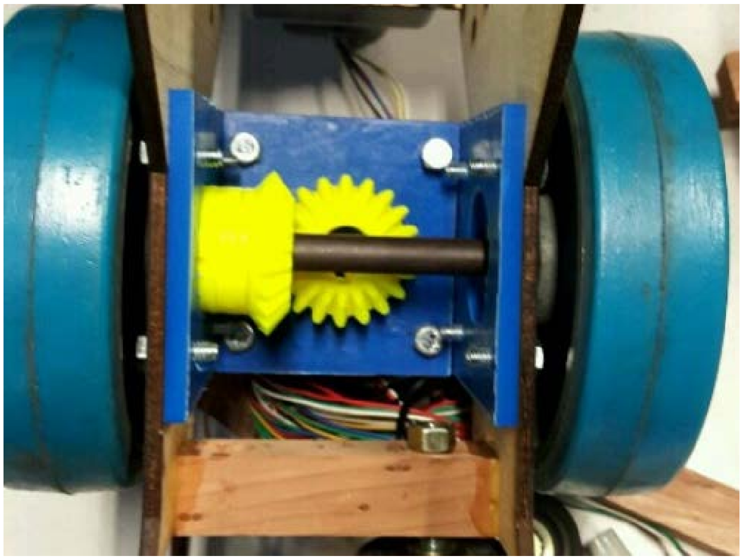


Figure 2-36. Miter gear prototype integrated to wheel shaft.

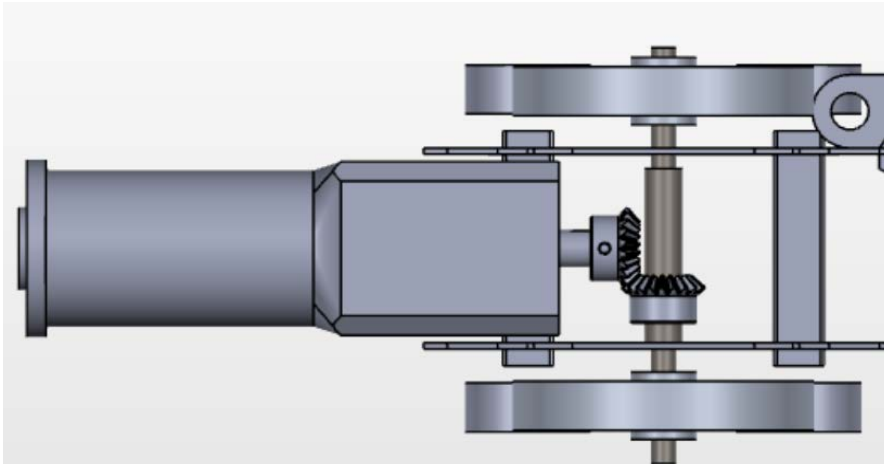


Figure 2-37. CAD Top view of motor and driving shaft interfaced with miter gears.

Figure 2-38. CAD Top view of motor and driving shaft interfaced with miter gears.

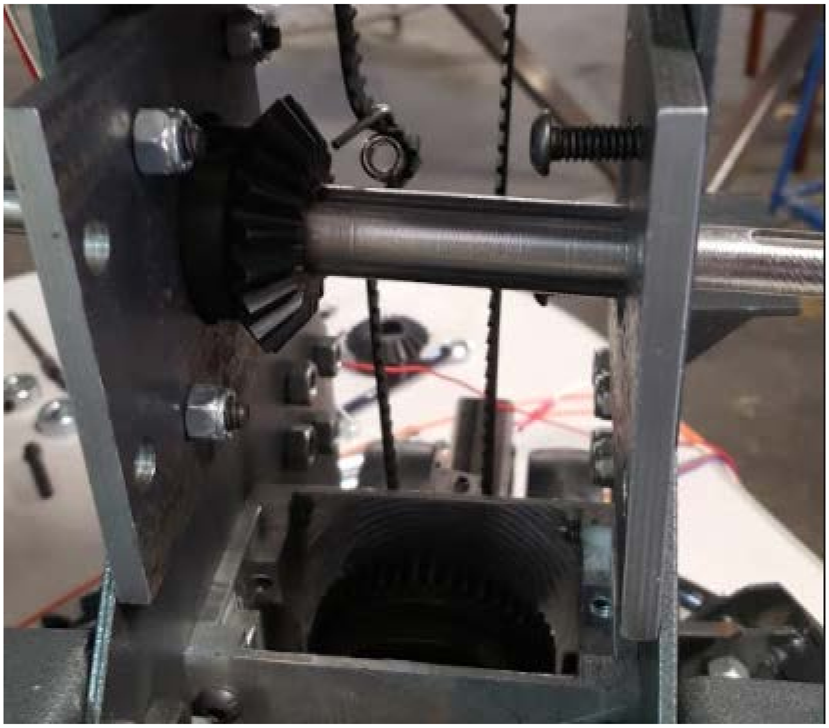


Figure 2-39. Miter gear attached to driving shaft.

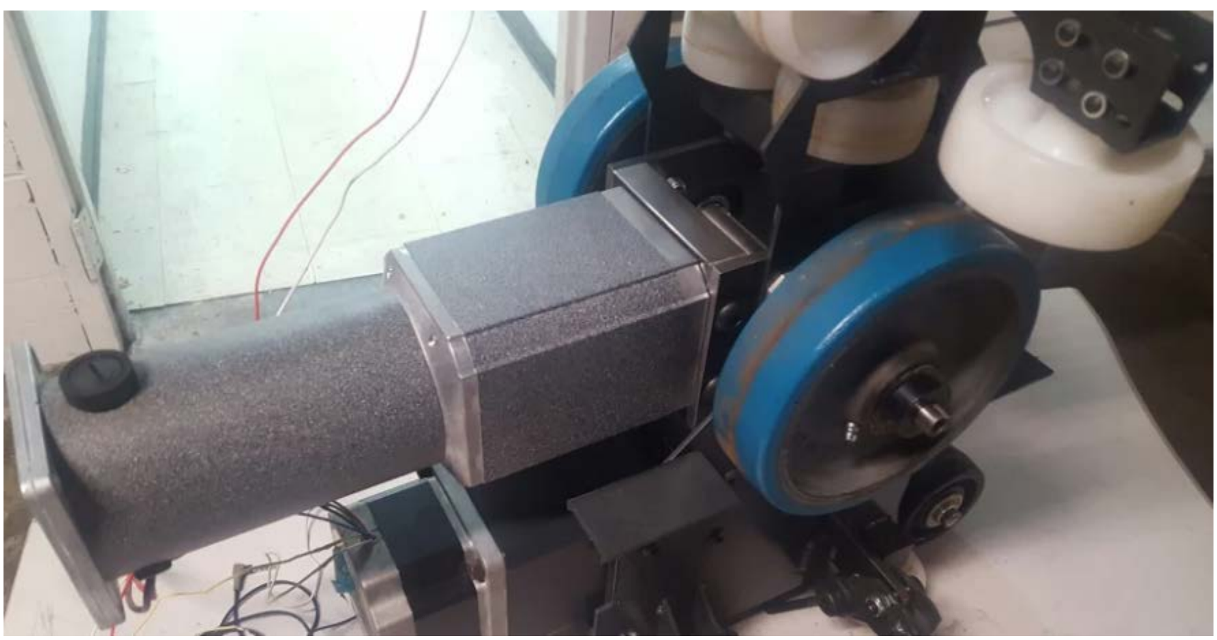


Figure 2-40. Propulsion motor attached to bogie along with driving wheels.

The design for braking was quite time consuming as our propulsion design and the bogie design left very little space to implement a viable braking mechanism for the bogie. There were numerous design ideas for braking but ultimately, we decided to go along with a spring-loaded lever arm to rub against the driving wheel. The spring would push the lever arm, and the chock attached, to rub against the driving wheel and slow it down eventually to a stop, shown in Figure 2-44. In order to disengage the lever arm, a stepper motor would reel the paracord attached to the lever arm and compress the spring, as seen in Figure 2-43. Each of the braking mechanisms have a mounting location for the stepper motors. There would be two braking mechanisms attached to the entire bogie system, one for each half bogie, for all driving wheels. Furthermore, the integration of a gearbox was considered due to the low torque the stepper motors could provide to work against the strong springs needed to tightly squeeze against the driving wheels. However, due to the allotted time, only one gearbox was fabricated in time for Maker Faire.

The reason we decided to go along with this brake design was due to the simplicity of providing friction force and placed in a location within the bogie that wouldn't interfere with any other sub-systems. All the components for the braking mechanism were personally fabricated at the TechShop using steel square inch tube, ½ inch steel stock, and aluminum. The team made use of the CNC, mill, and the lathe machines in creating the components for both braking and propulsion. Additionally, one of the braking mechanisms included mounting plates for the wayside team to mount their shoe collector that would power the bogie system, as seen in Figure 2-42 and Figure 2-44.

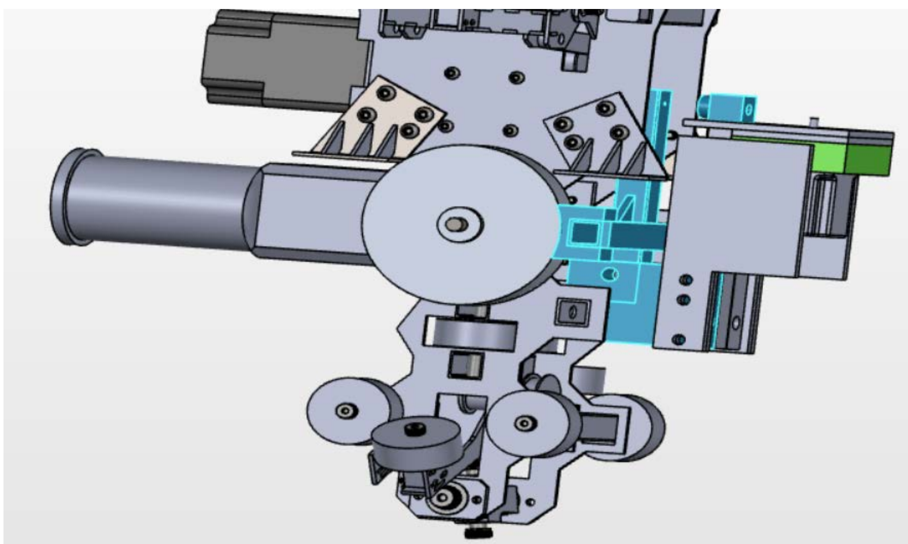


Figure 2-41. Half bogie with Braking highlighted in blue.

Figure 2-42. Braking mechanism with mounting plates for Wayside team.

Figure 2-43. Paracord and spring for braking mechanism.

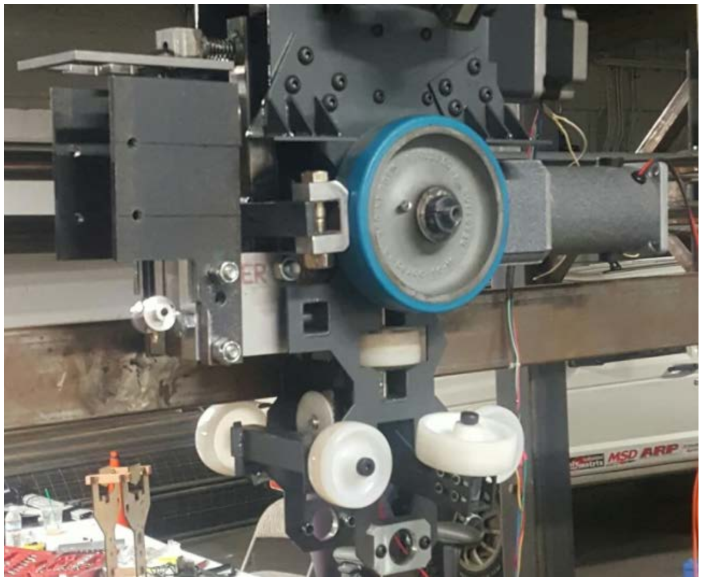


Figure 2-44. Bogie mounted on guideway with braking chock rubbing against drive wheel.

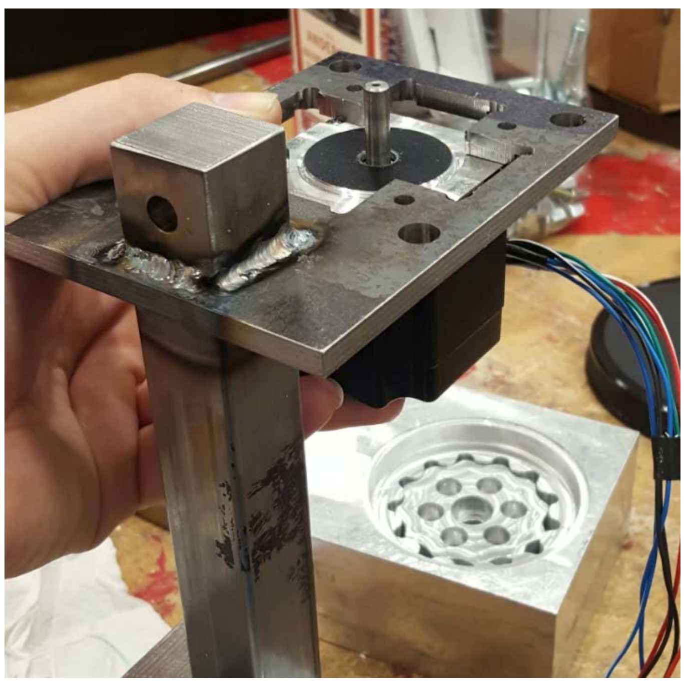


Figure 2-45. Stepper motor attached to mount for braking mechanism.

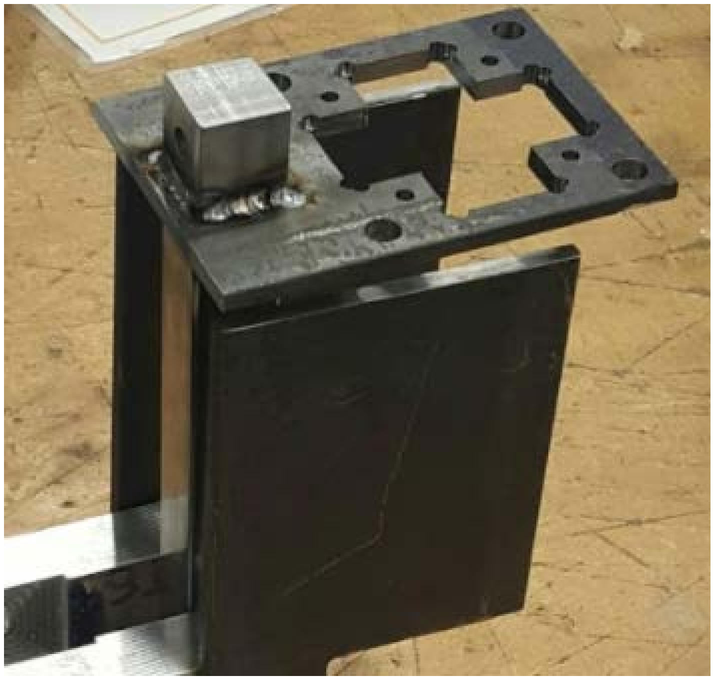


Figure 2-46. Mounting plate for stepper motor welded to brake mechanism.

Analysis/Validation/Testing

Once all components for both propulsion and braking were assembled, testing of each mechanism began. For propulsion, the miter gears were correctly aligned with the motor shaft and driving shaft and as the mechatronics team actuated the propulsion motors, driving wheels rotated as intended. The important test was making sure that the bogie can move along the guideway, especially the incline. However, due to the propulsion motors being heavily geared, the bogie couldn't move without the motors being actuated by the mechatronics team. Manual movement of the bogie along the guideway was only possible if one of the miter gears was disengaged. The mechatronics team had a few difficulties in assembling the electrical hardware along with correcting the programming code, and as a result movement of the bogie along the guideway was delayed. Ultimately, at Maker Faire, the bogie system moved along the guideway and more importantly could traverse the incline with ease.

To test out the braking system, the braking mechanism was mounted to the bogie then attempted to rotate the driving wheels. The chocks and lever arm were successful in locking the driving wheels in place, as rotation of the wheels wasn't possible. The important test for braking, was making sure that the bogie could come to a complete stop along the decline section of the guideway. The bogie was placed right before the decline and tested to see if it would slide down the slope. The bogie stayed in place, meaning it was successful in braking the bogie, as seen in Figure 2-47. Upon further observation, we noticed that the force on the wheels was too strong for the stepper motor to pull back. The springs were cut down to lessen the force on the wheels, yet they remained too strong as we couldn't pull back the braking lever manually. Two different springs were ordered within the last week to test out with the stepper motors but eventually couldn't test them appropriately. In addition, the gearbox was tested along with the stepper motor. We observed that the gearbox would keep binding and eventually wouldn't move correctly. Due to timing constraints, it was decided that the gearbox would be omitted from the brake design. The next step would be to make sure that the stepper motor could pull the lever arm away from the wheels. The braking mechanism seemed to successfully lock the wheels in place, however, the team couldn't further test braking due to complications on mechatronics programming and electronics.

Figure 2-47. Entire bogie with brake mechanism locking wheels.

Procedure/Instructions

Figure 2-48. Brake assembly parts.

To start assembling the brakes, first the pieces in Figure 2-48 of this section need to be collected and organized. These parts are labeled in the grabcad as bm_bm_1, bm_bm_15, bm_bm_14, and the welded pieces attached to bm_bm_2. All these parts should be found in the existing model of the bogies that can be found in the braking folder on the google drive. But otherwise, it is very simple to put these parts together; using 2 3/8-16 X 4.5 inch bolts and 2 nylon-locking nuts, every part slips easily together and can be tightened accordingly as seen in Figure 2-49. It should be mentioned that this process needs to be completed twice for each bogie to have their own brakes.

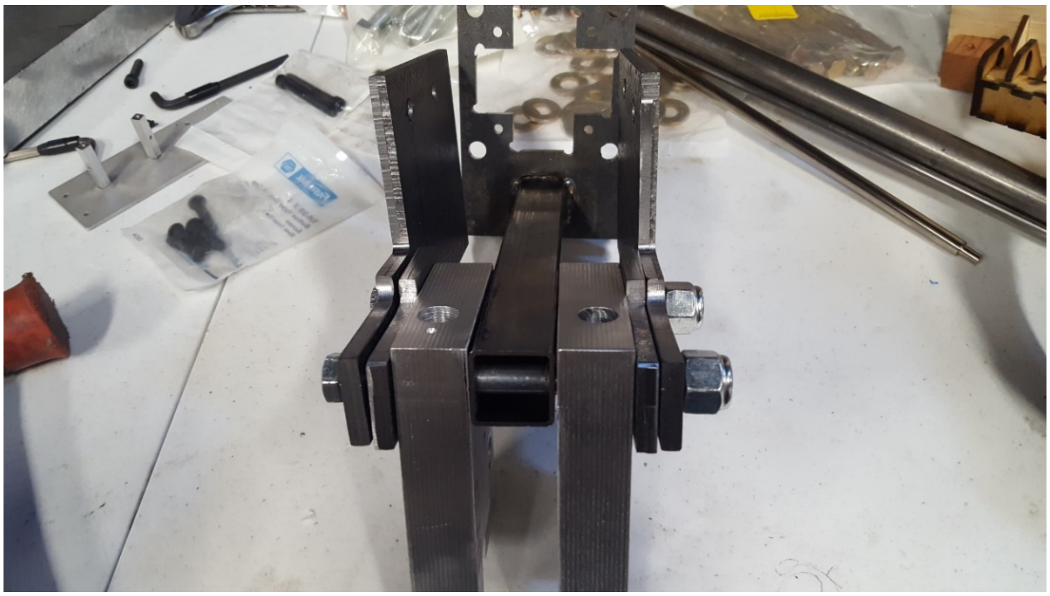


Figure 2-49. Partially assembled brake design.

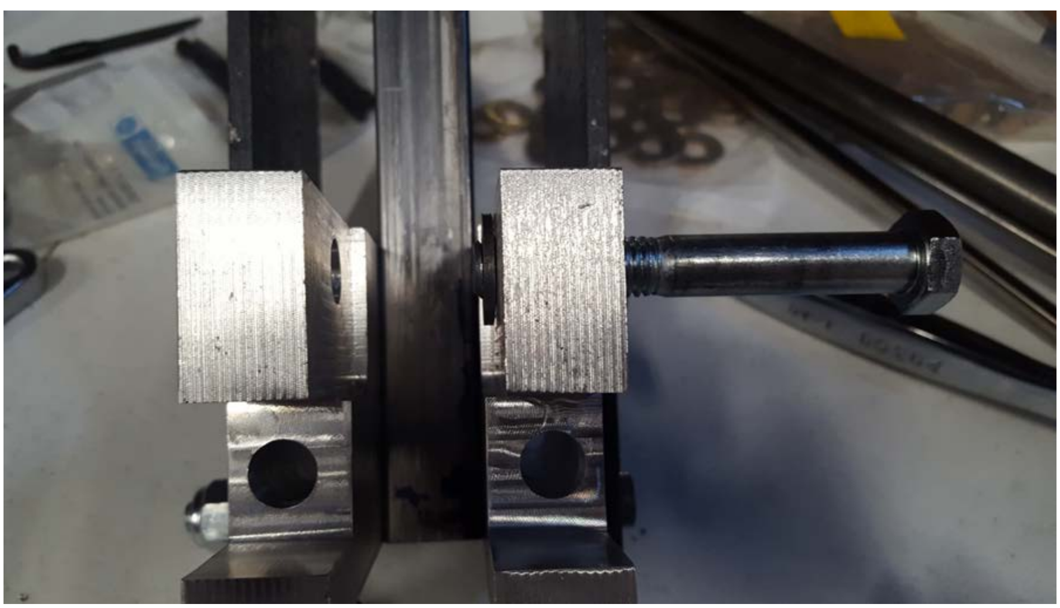


Figure 2-50. 1/2-13 X 3.5-inch bolt being pushed through the current assembly.

After making the general assembly for the back of the brake, a ½-13 X 3.5-inch bolt is slipped into place on one side of the assembly. As seen in Figure 2-50. The braking lever as well as 2 spacers are pushed into place and locked down using a nut. The spacers and lever arm are seen in Figure 2-52 and the overall assembly looks like Figure 2-51.

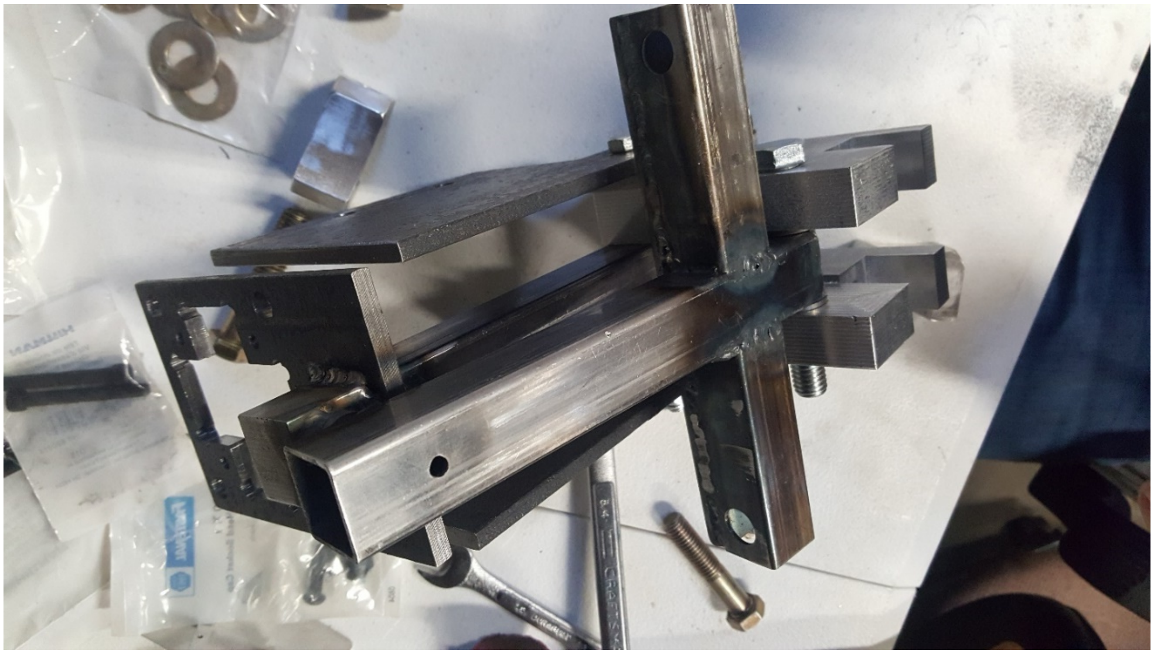


Figure 2-51. An image of the braking lever arm in place using the 1/2-13 X 3.5-inch bolt.

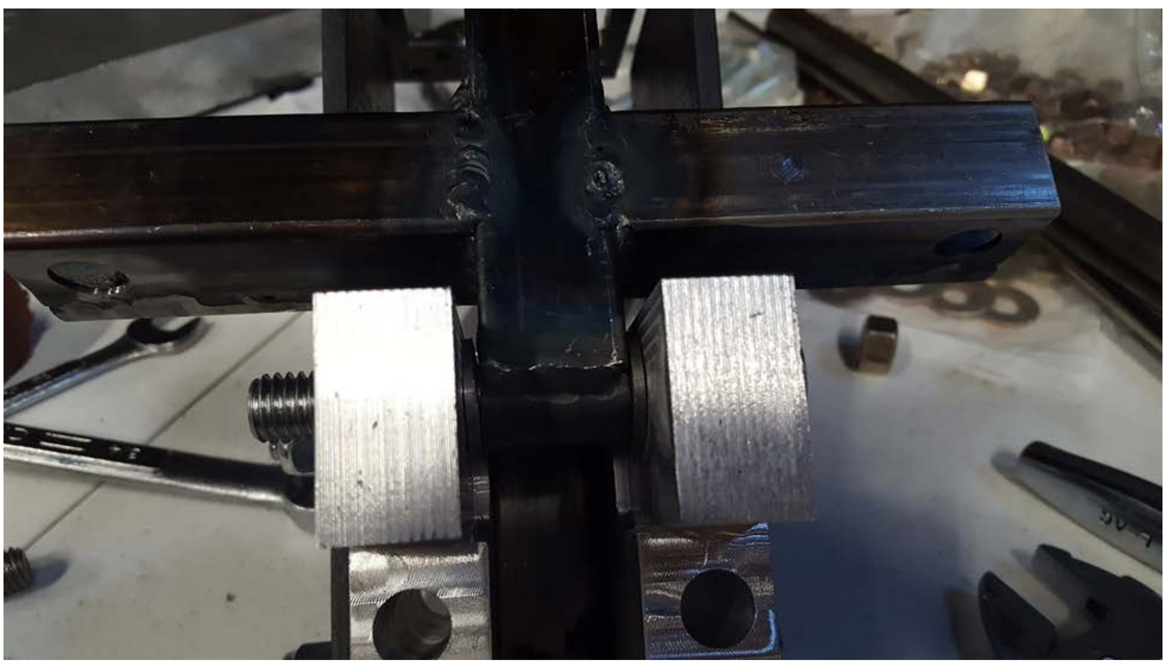


Figure 2-52. It should be noted that there are spacers in between the lever arm and the walls of the braking assembly.

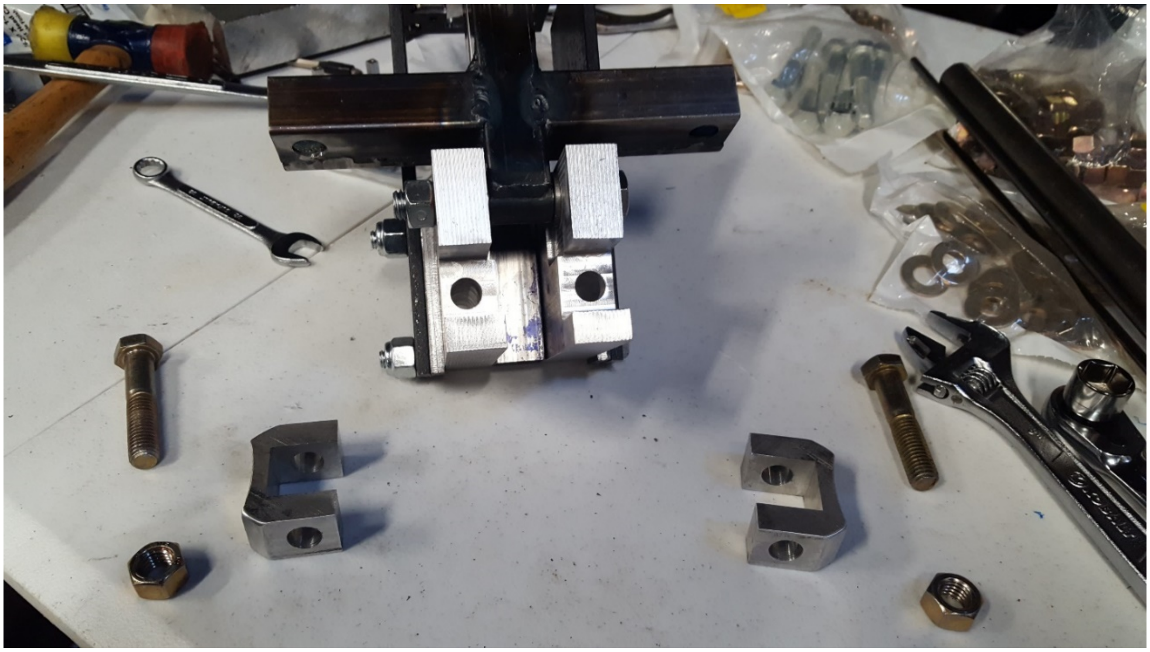


Figure 2-53. The next step is attaching the chocks to the bogie brakes.

The next step of assembling the brakes is to use 2 ½-13 X 2.5-in bolts and nuts to put the brake chocks into place as seen in Figure 2-53. It is simple to slip the chocks over the brake lever arms and then push the bolts through the holes on top, bolting them down on the other side.

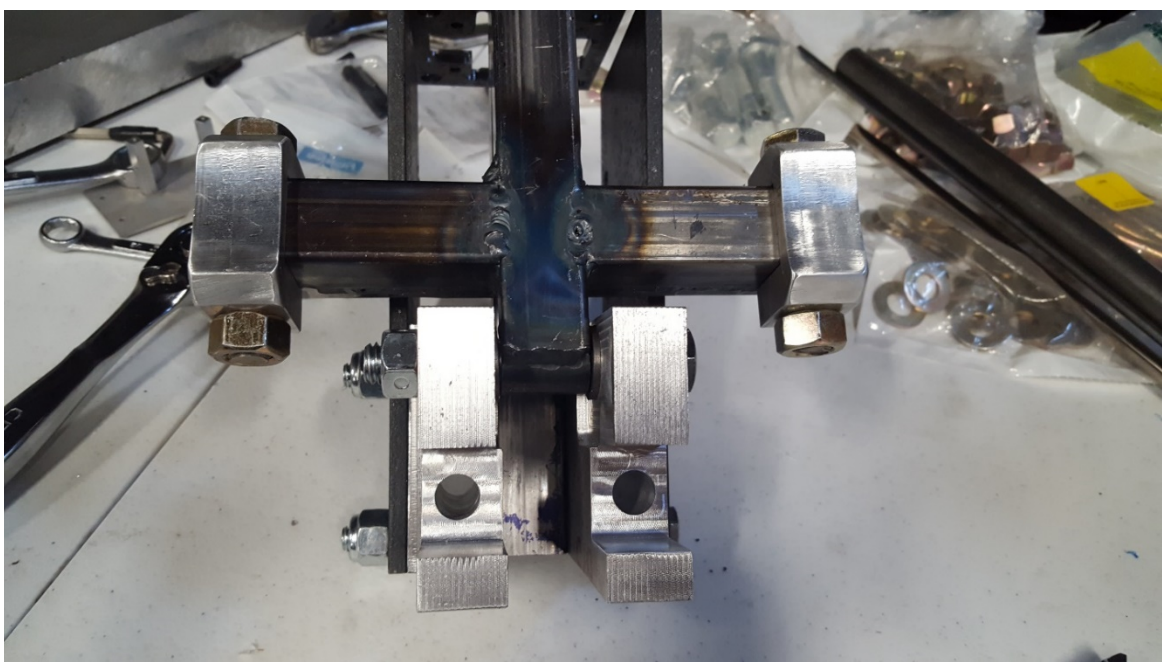


Figure 2-54. With the brakes in place, the near completed assembly should look like this.

Figure 2-55. Next, the winding string is put through the lever arm.

The next part of the assembly uses a sting that slips through the braking lever arm towards the vertical support as seen in Figure 2-55. A spring is then slipped around the spring and slid into the cup on the braking lever arm as seen in Figure 2-56. The string is then put through a hole in the top of the support structure and the spring is pushed into place. As seen in Figure 2-57.

Figure 2-56. A spring is placed into the appropriate cup with the line going through the center of the spring.

Figure 2-57. The line is then fed through the other cup towards the stepper motor mounting point.

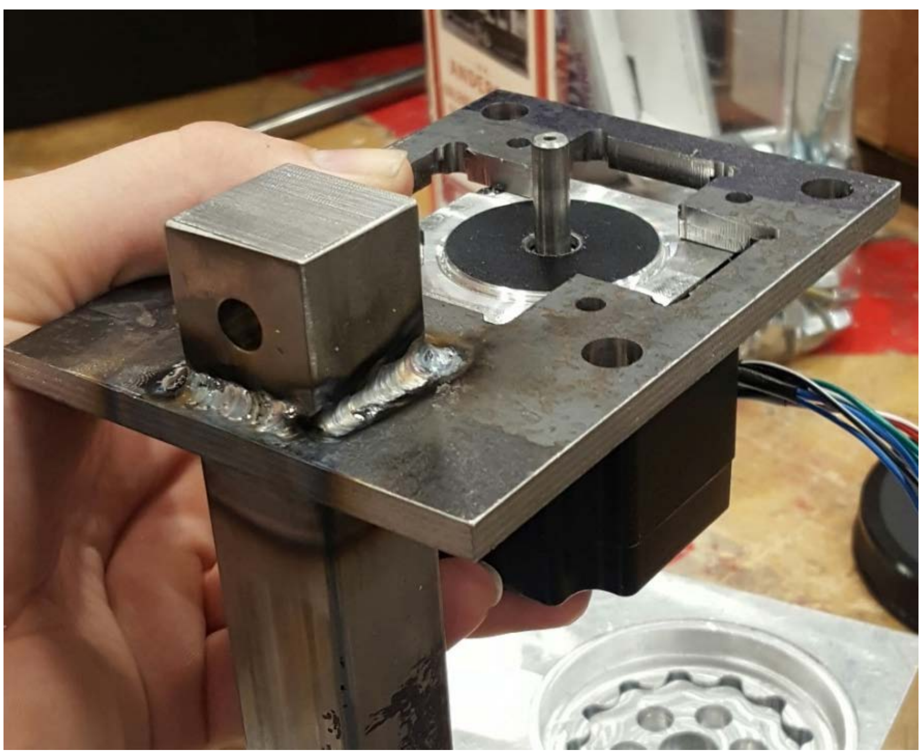


Figure 2-58. Due to a bit of a mix up, this is where the stepper motor will be mounted, with 10-32 screws holding it in place.

Unfortunately, there are a few missing pictures for the last parts, but a stepper motor is attached to the top of the vertical support using 10-32 screws as seen in Figure 2-57. And then an aluminum winding coil is attached to the top of the stepper motor with a set screw. The string it the pulled through the winding coil and wrapped around, giving the stepper a means to pull back the brake lever arm and release the brake. Finally, the brake can be attached to the bogies using 2 ½-13 X 5.5-inch bolts as seen in Figure 2-59. It is recommended that the brake is put on last after the propulsion to more easily access and adjust the miter gears without having to reach around the brake.

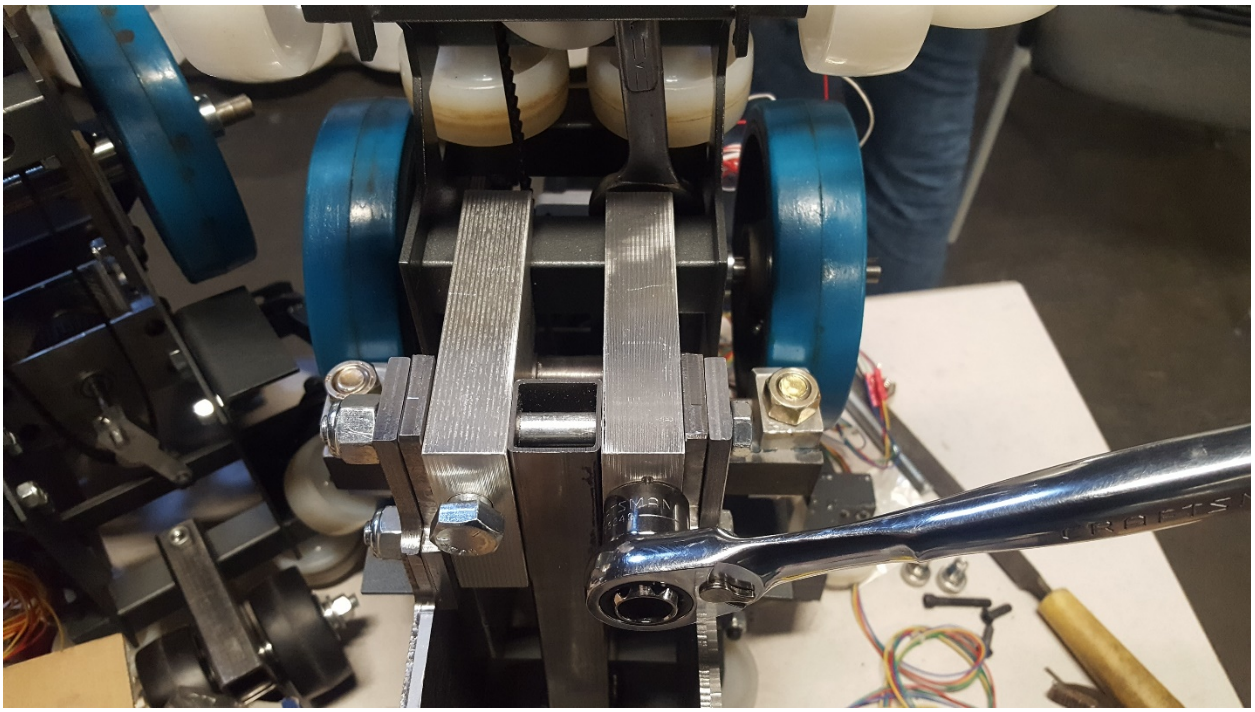


Figure 2-59. Finally, the brake assembly is mounted to the bogies using 2 1/2-13 X 5.5 in. bolts with nuts on the other side. It is highly recommended to do this after propulsion is put in place.

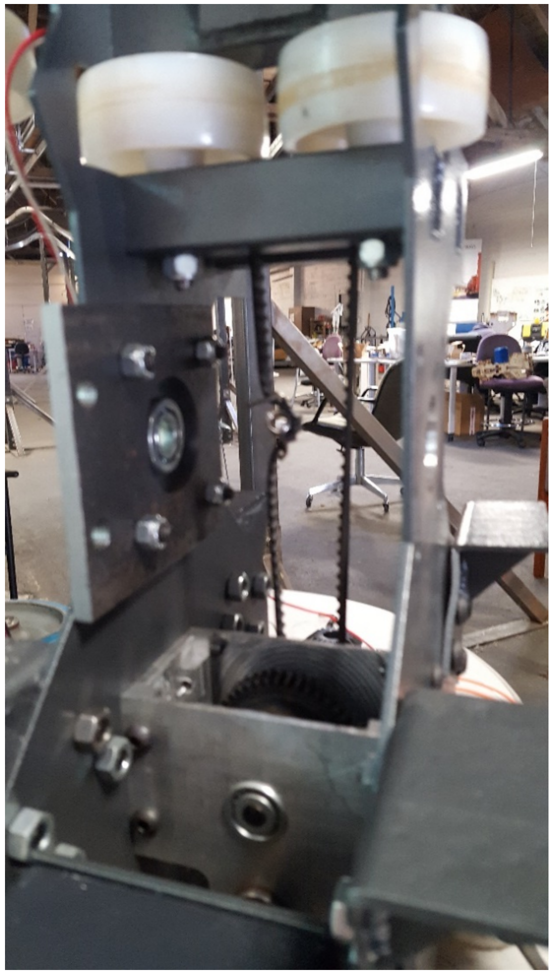


Figure 2-60. To begin propulsion, one of the mounting plates, labeled mb_motor 2 is attached to the same 1/4-20 half in bolts holding the main shaft's bearings.

Assembling the propulsion is a little trickier that the brake, there is a very specific order that needs to be followed to make sure every part slips into place. First, the part mb_motor_1 is attached to one side of the bogie, making sure that the mounting screw holes are not in line with the pulley controlling the steering as seen in Figure 2-60. The ¼-20 screws that attach the bearing plate on easily slip through mb_motor_1, allowing for the nylon locking nuts to be screwed into place against the plate. The next step is a bit trickier as the next motor mount plate is rested against the other wall of the bogie as seen in figure 14, a screw can be used to help keep it in place, but it needs to be there for the main axle to be put in place. So, as the axle is pushed into place, the miter gear needs to slip on before getting pushed through the bearing on the other side. As seen in Figure 2-63 and Figure 2-64. A piece of key stock can then be place between the axle and the miter gear to make sure they turn in unison. Before the axle is slipped through the bearing on the far side.

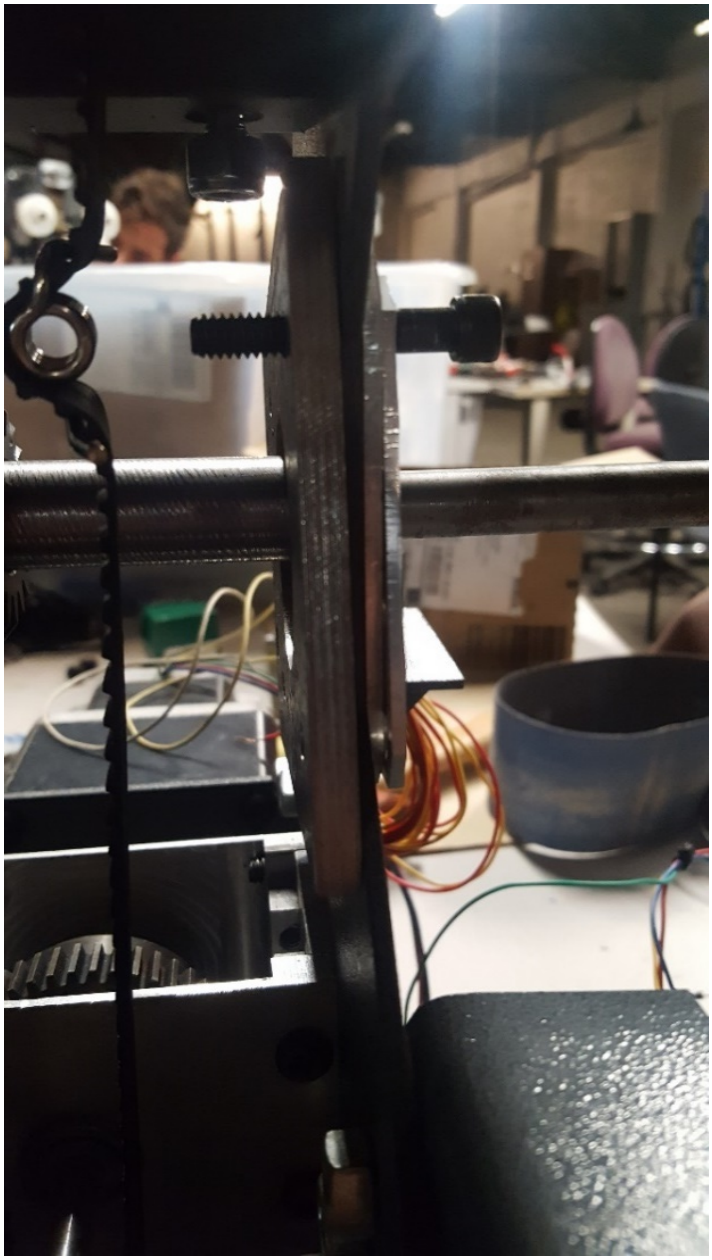


Figure 2-61. The next step is to put the other motor mount in place, but do not tighten it down until the main axle is through with the miter gear.

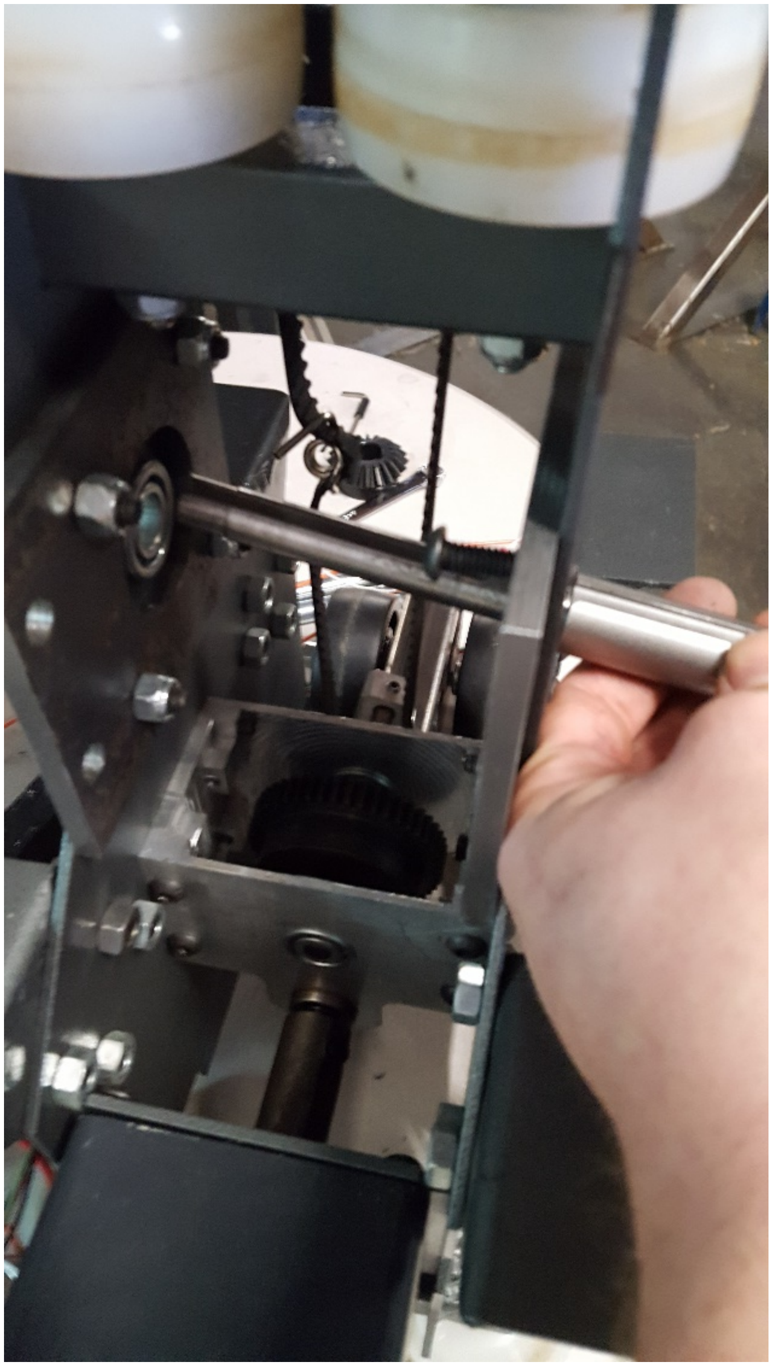


Figure 2-62. An image of the main axle being guided into place.

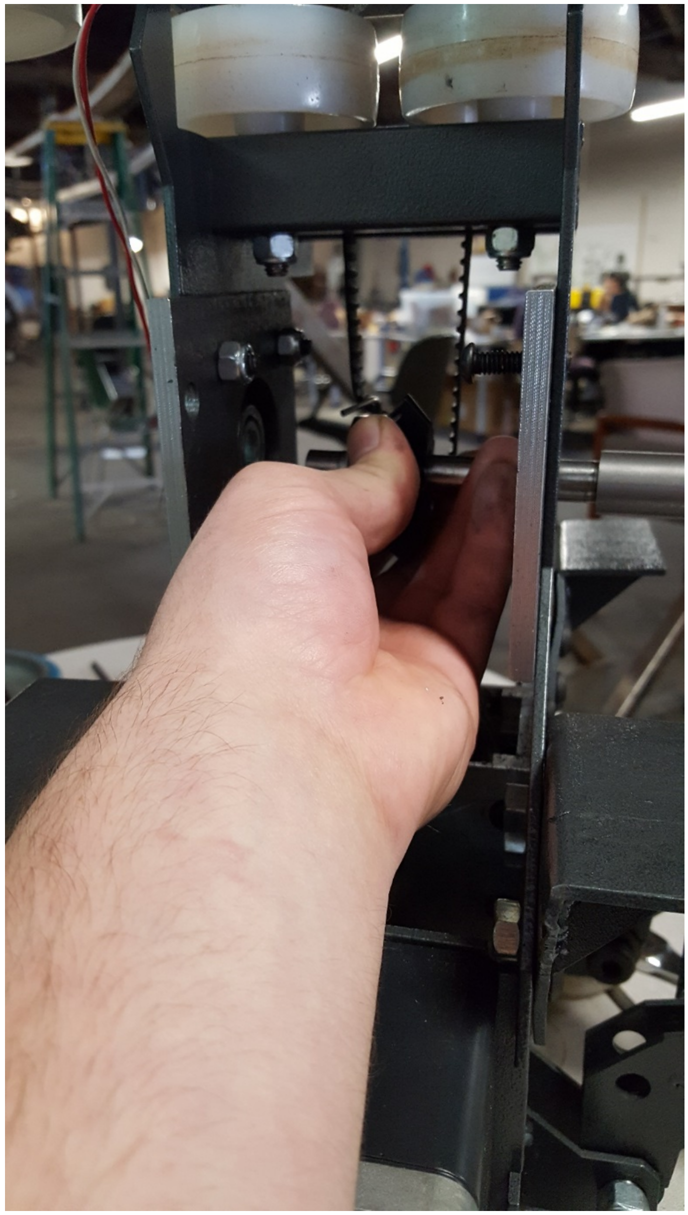


Figure 2-63. Before the axle makes its way through the other side, the miter gear and key stock need to be put in place so they can slip onto the shaft.

With the axle in place, the other bearing and holding plate can slip over the axle. As seen in Figure 2-65. Then, the ¼-20 X 1 inch screws can be pushed through and tighten down with nuts on the other side. At this point the propulsion motor will have the mounting plate attached to the front of it using ¼-20 X 1.75 in bolts as seen in Figure 2-67. At this stage the miter gear is slipped over the shaft of the motor aligning the keystock with the keyway on the miter gear and not tightening it down quite yet to make sure the miter gear on the main shaft and motor can align.

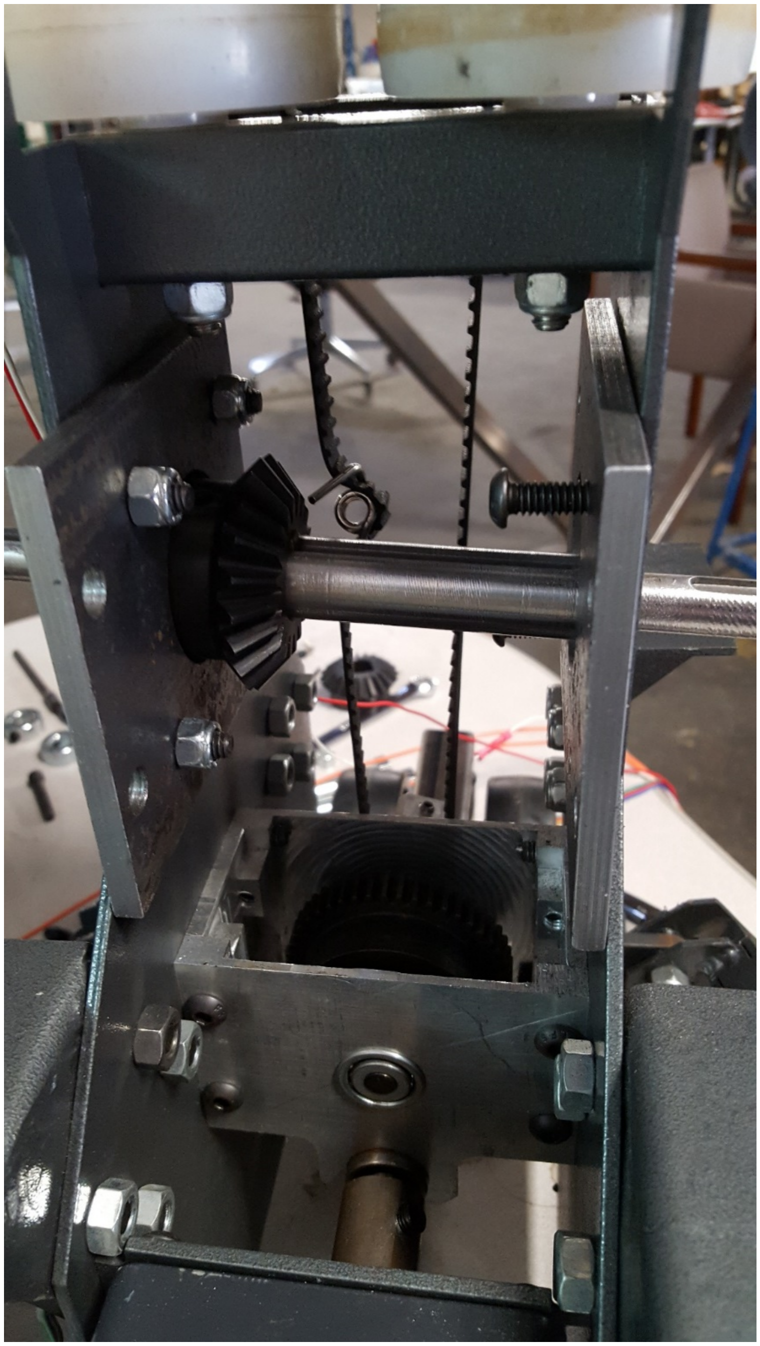


Figure 2-64. The main axle with the miter gear close to being in place.

Figure 2-65. The other bearing plate and bearing being put into place with over the axle.

Figure 2-66. To give you an idea of how the middle part of the motor mount attaches there are 2 3/8-16 button head screws holding it in place.

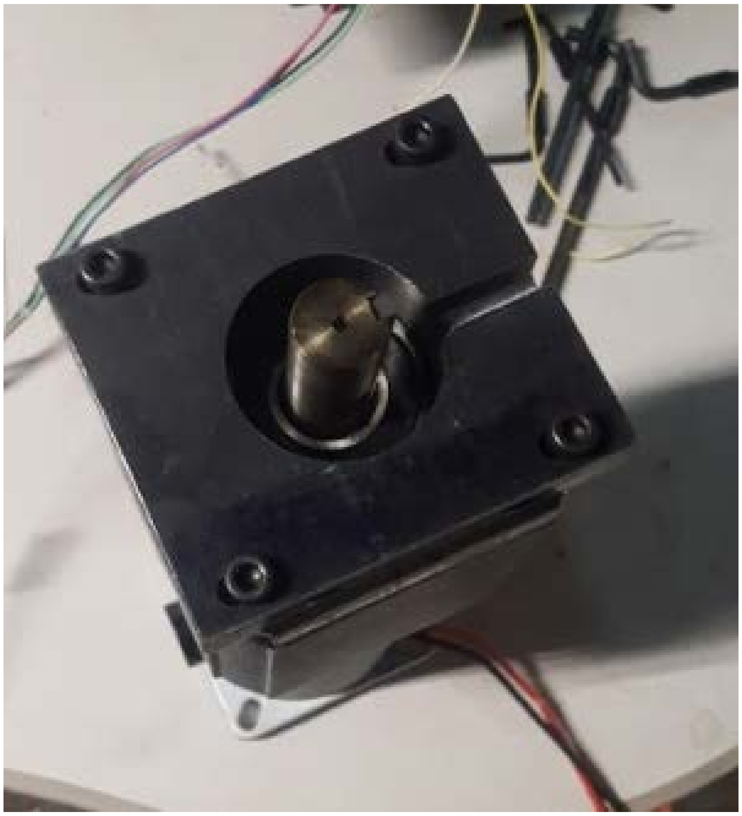


Figure 2-67. The motor is attached to piece mb_motor_1 using 4 1/2-20 X 1.75 inch screws, it is at this stage that the miter gear needs to be placed onto the motor shaft.

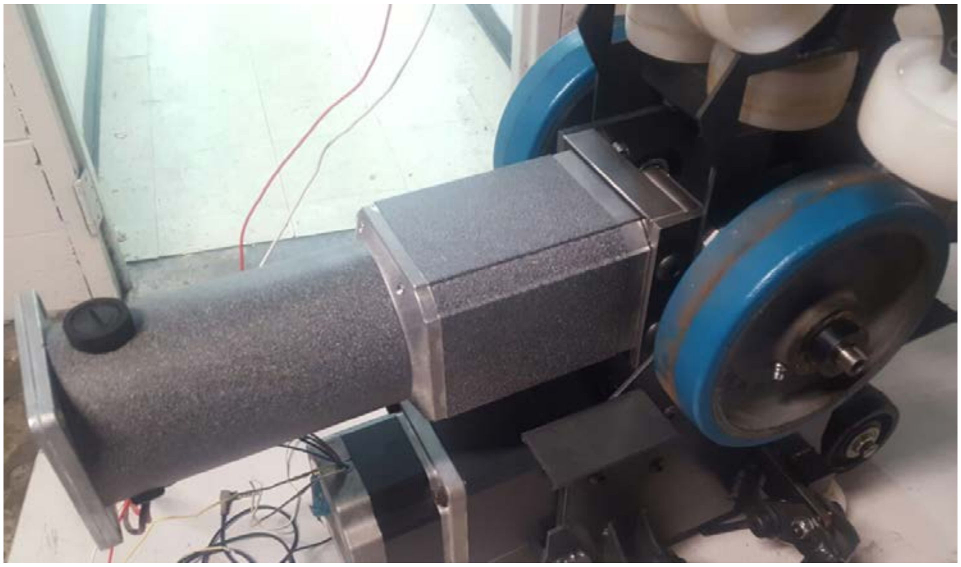


Figure 2-68. The motor attached to the mounting plate will slip between the two mb_motor_2 plates which have been bolted into place. Then the 3/8-16 screws are used to attach it.

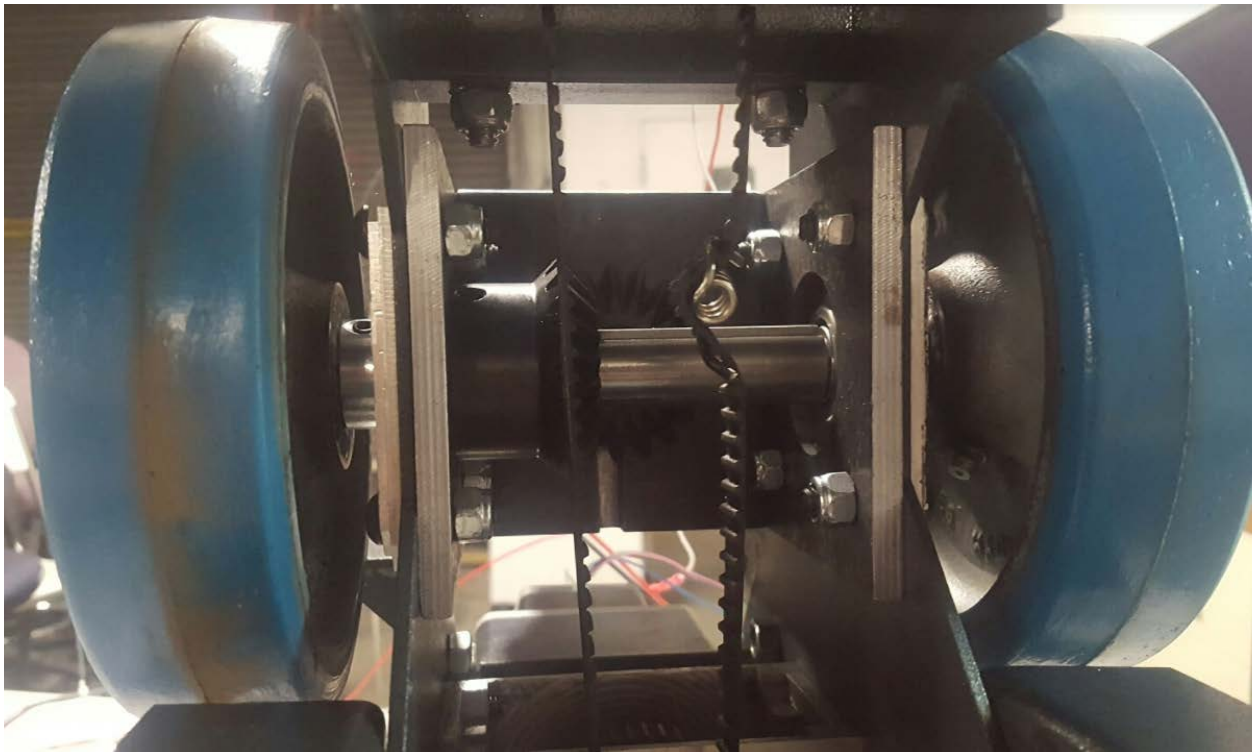


Figure 2-69. The miter gears can then be slipped into mesh and their set screws tighten using a 1/8 in Allen wrench so they will stay in place.

With the mounting plate attached to the motor, the mounting plate could then be put in between the two mounting plates that are already attached to the bogies as seen in Figure 2-68. Then 4 3/8-16 button head screws can be used to secure the motor from the side, fixing it in place. At this stage, the miter gears can be moved to properly mesh with each other. Then their set screws can be tightened with a 1/8th inch Allen wrench. Then on the outside of the bogie, two retaining rings are added on the shafts to keep them in place relative to the bogie. The wheels are then slipped on, and keystock added between the wheels and the shaft to keep them turning together. Then one more retain ring set is added on the outside of the wheels to keep them on. If the gears properly mesh and the set screws are tightened properly, then propulsion should function well as mechatronics spins the motor to speed. After the wheels are put on, the brake can put on in the same method as seen in Figure 2-59. Since there is a spring between the upright support and the lever arm, it will take a little bit of effort to push the brake on, for the spring will be compressed. For this reason and due to some tolerancing issues, the bolt should screw through the support post on the bogies instead of sliding right into place, but that will help to keep it on as it is tightened down. Then that will complete the assembly of the braking and propulsion parts to the bogie structure.

Money Spent on Project

Table 2-2. Money spent on parts for Braking and Propulsion.

Vendor	Description	Quantity	Price Per Item	Total Cost
McMaster	Miter Gears (6843K19)	4	$53.56	$214.24
McMaster	Compression Spring (9657K94)	1 (Pack of 6)		$7.18
McMaster	Music-Wire steel Precision compression spring (9434K167)	1 (pack of 5)		$5.71
Amazon	Paracord Grade 550lb Type III Nylon Paracord 10 feet Black	10 ft.	$5.99	$5.99
Amazon	Steel Key Stock, Standard Tolerance, 3/16" Thickness, 3/16" Width	1 ft.	$9.15	$9.15
Amazon	CNC Stepper Motor Nema 23 Bipolar 2.8A 269oz.in/1.9Nm CNC Mill Lathe	1	$39.99	$39.99
	Stepper motor	1		Provided
Groschopp	24V DC motor with 160:1 Gear ratio (PM8014-PL73160)	2	$261.27	$522.54
Western Tool and Supply	¼ 4 flute end mill, double sided	1	$11	$11
	#4 center drill	1	$24.95	$24.95
	1/4 in 4 flute end mill	1	$8.24	$8.24
	2 carbide turning inserts	2	$8.45	$16.90
	3 titanium turning inserts	3	7.44	22.32
	3/16 4 flute end mills	2	15.78	31.56
	2 ½ inch stub drill bits	2	7.30	14.60
	#2 center drill	3	4.24	12.72
	½ rougher endmill	1	45.01	45.01
Jem Tool Supply	⅜ 4 flute endmill	1	28.30	28.30
Fastenal	½-13 X 5 bolt	5	.98	4.90

	¼-20 X 1 screws	10	.30	3.00
	#6-32 set screw	10	.30	3.00
	¼-20 X1 screw	20	.3	6.00
	¼-20X1.75	10	.30	3.00
Sims Metal Management	1.25-inch round stock	6	5.7$/ft	34.20
	Usable Ferrous metal	35	.5$/lb	17.50
	1X 16-gauge square tubing	10	.73$/ft	7.30
	Usable Ferrous metal	65	.5$/lb	32.50
	usable ferrous metal	20	.65$/lb	13.00
	¼ X4 inch square stock	10	2.52$/ft	25.20
	Metal		$103.83	$103.83
			Total Cost	$1,199.74

Results and Discussion

The braking and propulsion systems were successful in completing the objectives set forth from the Fall semester. The propulsion was successfully displayed at Makers Faire; however, braking wasn't successfully actuated and integrated with mechatronics due to time limitations. Furthermore, since the propulsion motors were heavily geared due to the gearbox, if the motors weren't powered to move, the bogie would come to a complete stop. The gears functioned as brakes for the system. Ultimately, the braking mechanism built would function as failsafe braking. If the power was cutoff, the springs would push the chocks and lock the driving wheels in place.

The propulsion mechanism works well enough that it can carry over for next year's Spartan Superway team. Since most the entire braking mechanism is bolted on to a section of the bogie, it allows for next year's team to work on the existing model or rework on a new model. The progress from this year's braking and propulsion team allows for next year's team to refine the current designs and continue important work on other sub-systems of the project, such as mechatronics, and steering.

Conclusion & Suggestions for Future work

Ultimately, the propulsion design met the specifications set beginning of the year. Braking design partly satisfied the specifications but because of integration issues wasn't completed in time. Testing the integration of all sub-systems was the most difficult task to accomplish for our team. Time for fabrication wasn't estimated appropriately and even further delayed due to receiving parts and funding for parts so late in the year. The reason for such small amount of time to test the designs was largely due to the funding situations delaying all subteams to get the parts in time for early fabrication. Since the fabrication was done by a team member with extensive machining work, the team saved money on fabrication. Overall, we were satisfied with the progress made for both braking and propulsion design and hope that it carries on for next year's team so that they may improve the Spartan Superway half-scale model.

There are a few suggestions for next year's team to consider in further improving both propulsion and braking. For propulsion, we ran into issues with the miter gears disengaging. The reason was that the set screws weren't enough to hold them in place after a few runs. Suggestion would be to find a better way to secure the miter gears in place. For braking we suggest continuing testing out the brake design we worked on and find a balanced manner to have a strong spring locking the wheels from moving, while also operating a strong enough motor to pull the spring back. We also highly suggest managing time wisely, especially in the spring semester. There were moments in which we were stuck with braking design because the space limitations didn't allow for integrating an existing braking model.

Mechatronics and Integration Systems

Background and context

Maintaining the bogie in motion throughout the tracks is a challenge, especially when the bogie is constantly reading measurements taken from the sensor and relays the values to the microcontroller. Mechatronic system is present to provide necessary corrections to the bogie for self-adjustment and efficient power distributions. Without the necessary corrections and feedback in the system, the bogie will not be able to move throughout the track in a safely manner.

Since the tracks are spread out in an urban environment, it is important for the bogie to be able to sense any changes such as obstacles. Maintenance will also be carried out on the tracks as well as the bogie, and to manage the bogie to automate its movement the microcontroller must register any input or danger into its system.

Design Requirements and Specifications

1. Utilize the power provided (less than 1000W) to power the electronics and major electrical components.
2. Motors, actuators and sensors must be able to coordinate within themselves to bring the bogie through a complete loop.
3. For safety reasons, the operator must be able to gain control anytime when the bogie is in motion.

The following Figure 2-70 illustrates the overall system of the control circuit. Power supply and the microcontroller are to be separated safely using relay switches and diodes.

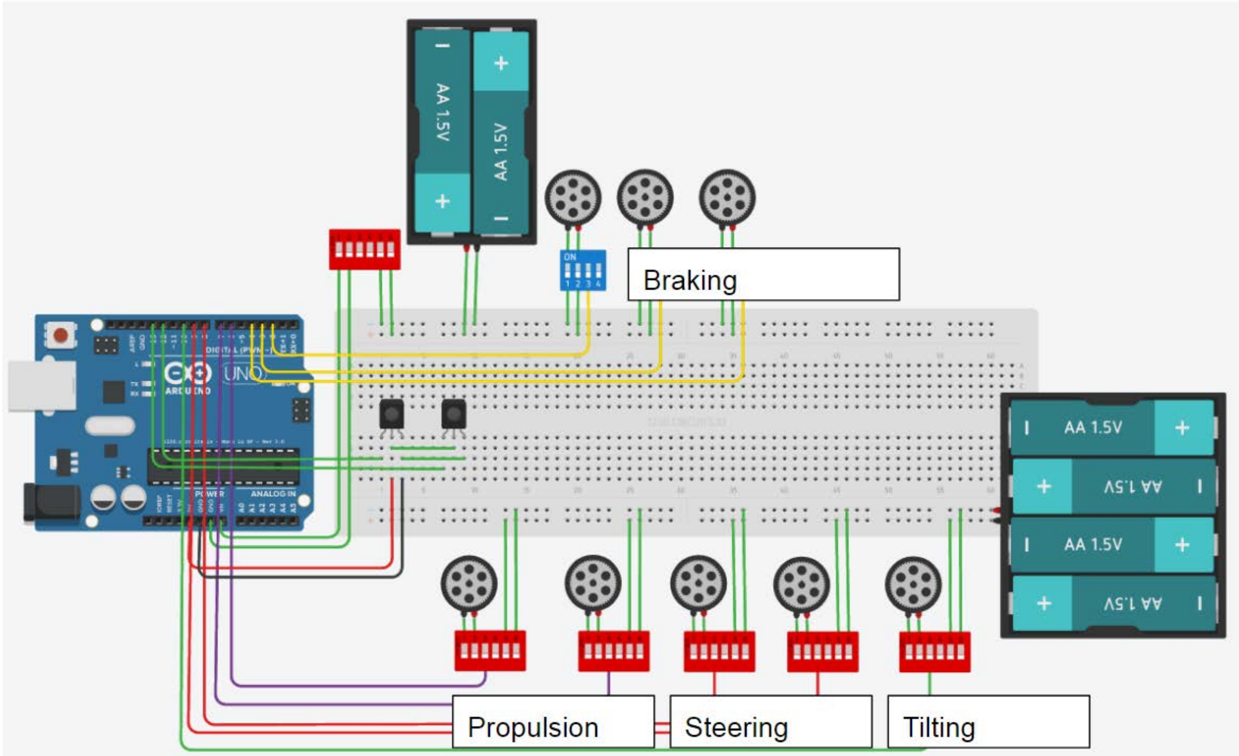


Figure 2-70. Circuit layout of the control system.

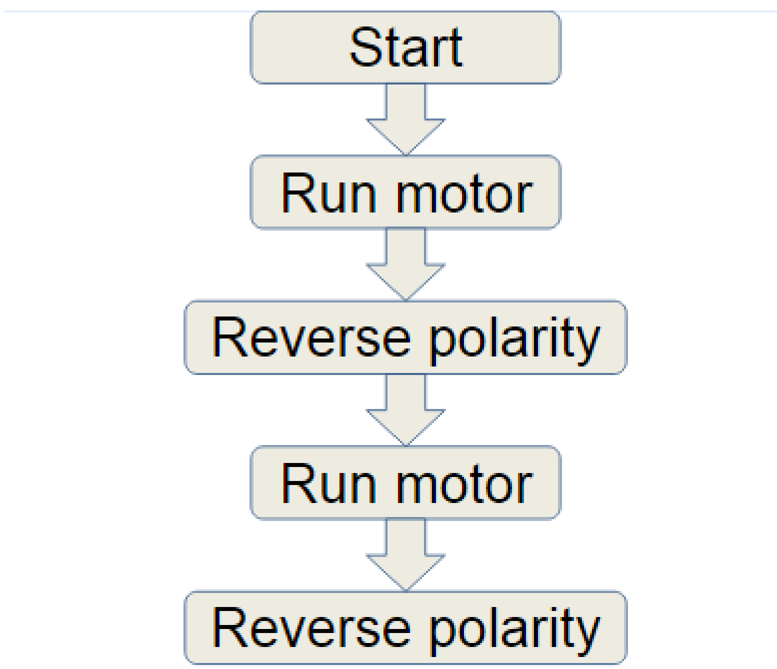


Figure 2-71. Diagram of propulsion control system.

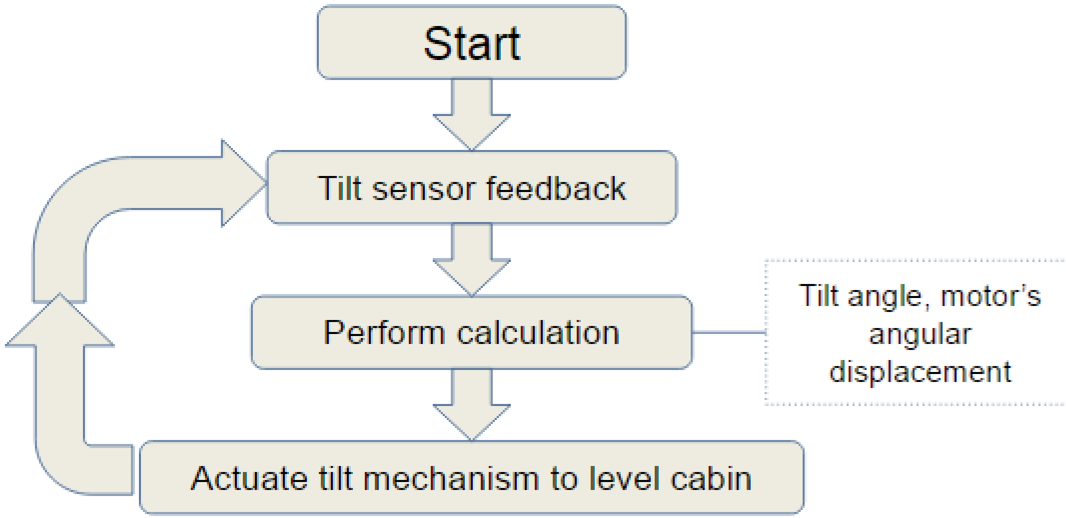


Figure 2-72. Diagram of tilting control system.

Literature Review

Masdar City is one of the places where Automated Transit Network (ATN) has launched. The network consists of podcars at selected stations both underground and on the road. These cars carry up to 4 passengers and allow them to select the desired destinations. While the network is similar to driverless cars, congestion still may occur and the problem with that would never be solved.

Figure 2-73. Masdar City Podcars.

The ATN at SJSU from 2012 to 2016 had their automated podcar system based on Arduino. According to the Spring 2016 report of Spartan Superway, PID controlling was used with specific and thorough configuration. The specific Sparkfun gyroscope was used to output feedback and data to the Arduino for PID controlling. Sensors such as hall effect sensors are configured to give feedback such as light on both sides of the mechanical bogie.

Description of Design

The design specifications include hauling no more than 400lbs of weight with 800 watts of power available. It must take about 5 seconds and less than 2 meters to reach its maximum velocity. The power must be sufficient enough to attain speed as well as powering the electronics.

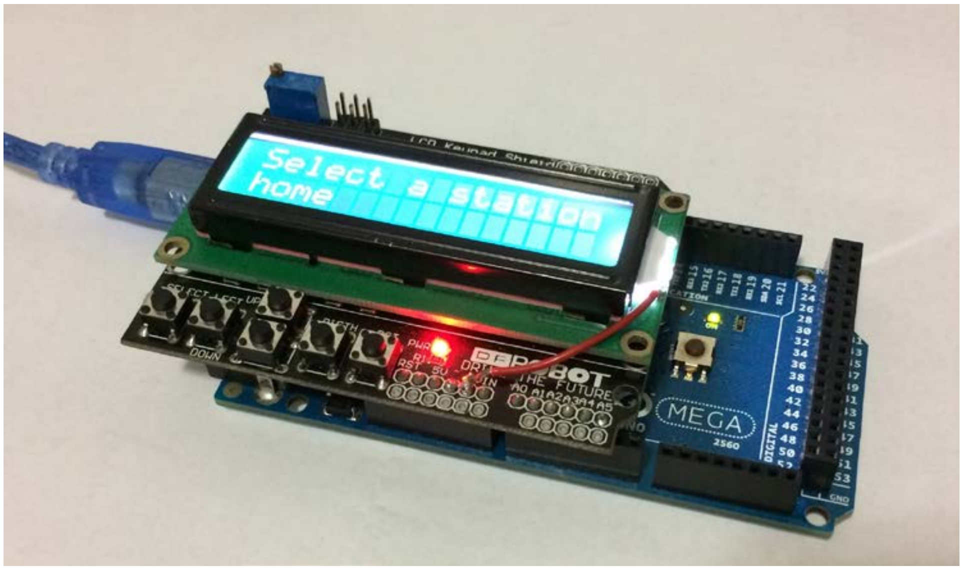


Figure 2-74. LCD control panel for central communication.

The LCD control panel will show the status of the bogie if it is running, decelerating or at a specific position on the track. User input is fed through the LCD keypad and into the Arduino, allowing the Arduino to run specific codes to control the movement of the motors, brakes as well as the switching arms. The Arduino will also specify how much power is drawn to supply the electronic components. Two hall effect sensors and a gyroscope is situated in front of the mechanical bogie. This is because the front of the mechanical bogie will tilt first as compared to the back of the bogie. This allows the gyroscope to make changes to the pitch of the bogie so that the bogie maintains level at all times. The two hall effect sensors are situated in front of the bogie to detect the location and the cues to decelerate at a given time. Once the first pair of magnets is detected by either one of the hall effect sensors, the system will determine that the bogie will decelerate within 5 seconds. If the sensors detect the second pair of magnets, then the brakes will be fully applied to get the bogie to a complete stop. Two seconds is the suggested time to take for the switching arm to switch the bogie to the secondary track. Once the bogie reaches to its destination, brakes are fully applied to ensure that the bogie does not move involuntarily. In the event of failsafe, the brakes will be fully applied and the gyroscope will tilt the bogie if necessary to ensure that the bogie is safely stopped. In a real-life situation, this will allow the evacuation of passengers from the cabin in a safe manner.

The allocated time for the bogie to reach its peak velocity is about 5 seconds. 1.25m was the calculated distance for the bogie to reach its maximum velocity. While the total power needed to power the electronics remain unclear at this stage, a minimum of 300 watts of power is required and 800 watts of power is proposed to power the electronics.

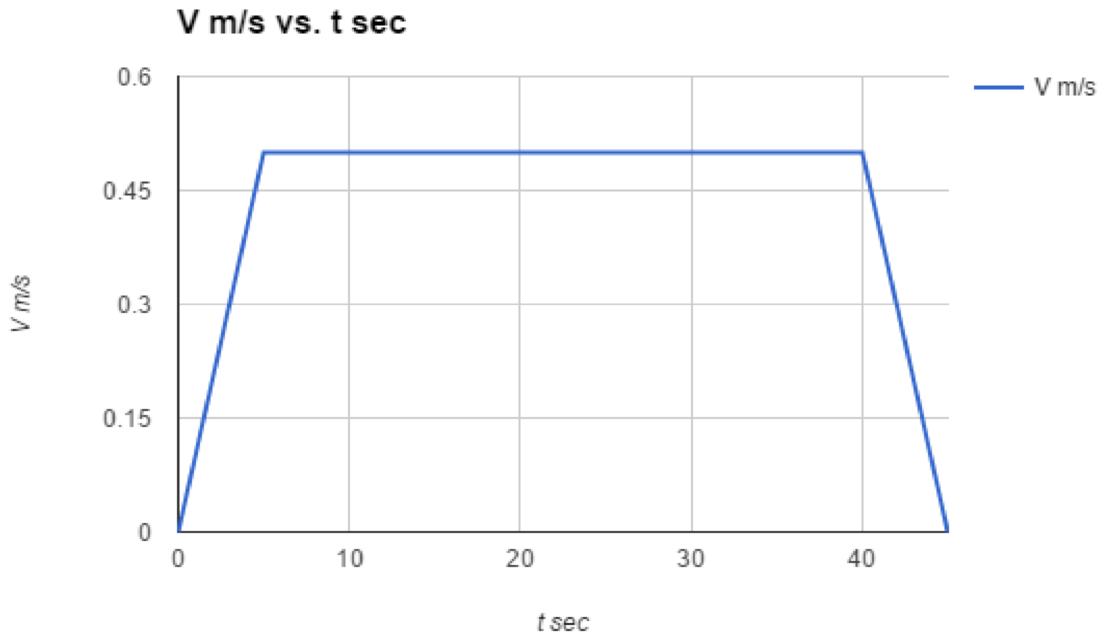


Figure 2-75. Graph of velocity vs time of the bogie.

The codes for fail safe, track loop demonstration and debugging are to be done over the winter. This subteam will wait for the half scale team to fully gather the specifications of the motors and required power to ensure that the bogie moves smoothly. In the beginning of Spring 2017, the purchasing and selecting of the motor drivers, batteries (if necessary) will be done.

Analysis Validation/Testing

The power needed for the entire system is estimated to be about 800 watts. Since the power inverter can safely take about 1500 watt of power, the inverter is used to convert DC to AC power. The motor was first tested by supplying a steady 12V from the power supply and the speed of the motor was evaluated. It was found that the motor speed was very slow, it was estimated to be about 20 RPM when running on 12V power supply. The current however was never tested because it was not a concern at the testing stage. Since a 10A fuse was connected to the motor, there was no more than 10A of current drawn from the motor at the starting phase. To counter the voltage spike when starting the motor, Dr. Furman advised to embed diodes that are able to take up any excess counter flow in the circuit. Voltage was not a concern at this point as the current on the motor was the main reason why electronics fail.

The stepper motors used on braking were running on 12V as well, together with L298N drivers. The motors were tested and their speed could be configured using the stepper library on Arduino IDE. Once the motors were tested using the L298N drivers, a 12:1 gear ratio gear reduction box was used. Propulsion and braking team tested the gear box and made some refining on the gears before it was tested the second time. Caution was taken when handling the L298N because the heat sink on the driver FET could get extremely hot when the 5V regulator was plugged in.

The stepper motors on steering were paired with the CW230 drivers. The drivers take a maximum of 3A when all the dip switches are configured properly. 12V was initially plugged into the driver to power the motors but there was no response on the motor. It was later observed that the stepper motors output high torque on their shafts and must be powered by 24V. The power supply that was used in the system could not completely power the motors because the voltage measured was lower than expected. A steady 19-21V was measured instead of 24. Necessary stepping function on the Arduino was reduced so that the motors could provide more torque with a slightly lower voltage. The motors were able to move with the limit switches in place.

The linear actuator on the platform levelling was tested by stripping the cable. A negative and positive terminal was found on the cable and was tested using the same 12V power supply. Because the linear actuator runs on 12V with a lead screw mechanism, the torque output is extremely high. When the 12V supply was plugged into the linear actuator with the platform levelling attached, sparks on the power supply were observed and the safety turn off relay on the power supply was activated. It was observed that the platform levelling mechanism was jammed and the actuator was pulling a lot of current. Because of this, a 4A fuse was used to limit the current into the linear actuator.

Limit switches were tested out using the N.O. configuration. The official Arduino website was used as a reference and a pull-down resistor was used in the circuit in respect to ground. Designated pins were used to connect to the limit switches. However, Dr. Furman had suggested that the Arduino Mega itself has a 10k pull-up resistor on each pin that could eliminate the need for an external resistor. INPUT_PULLUP was used in the code to enable the resistors on the pins and the rest of the limit switches were used without external resistor.

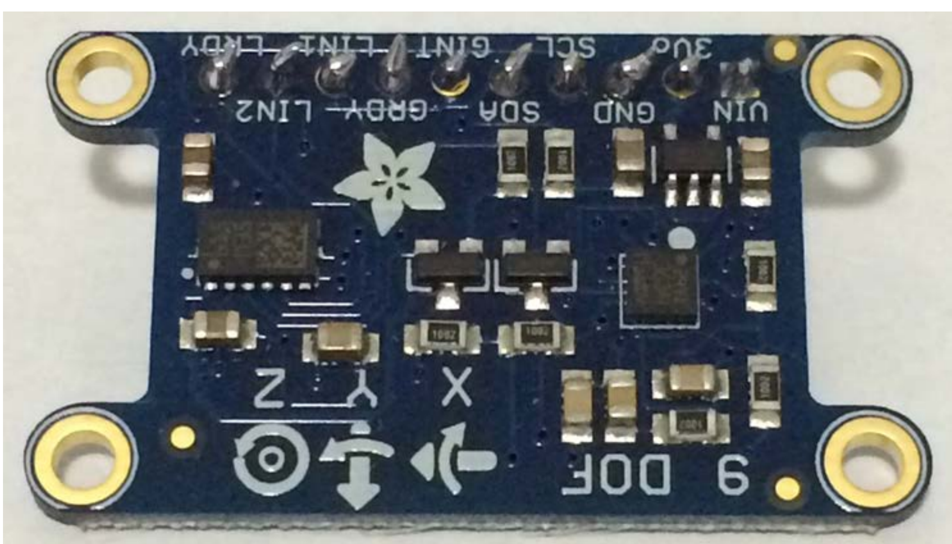


Figure 2-76. Gyroscope used to implement tilting.

The PID controller was used in the tilting mechanism based on the Adafruit's gyroscope. The PID is responsible for maintaining the bogie levelled to the ground as the bogie goes through the 17-degree slope on the track. Initially, PID_v1.h from Arduino was used to configure the system but deemed complicated and was then scrapped. The test was performed on a smaller geared motor using Pololu drivers. It was later determined that the Groschopp motor was not ideal in switching directions with fast speeds especially in real-life applications, but in this test the Groschopp motor was fast enough to turn the shaft in either directions. A new mathematical function was used instead and said to be more robust and easy to configure especially during Maker Faire.

Budget

Table 2-3. Budget for the team.

Parts	Name	Qty	*Price total $	Source
Gyroscope	Adafruit 9-DOF IMU Breakout - L3GD20H + LSM303	1	19.95	https://www.adafruit.com/products/1714
Hall effect sensor	Hall effect sensor - US5881LUA	1	2	https://www.adafruit.com/products/158
buzzer	Large Enclosed Piezo Element w/Wires	1	0.95	https://www.adafruit.com/?q=buzzer&p=1
Adafruit Shipping	Shipping		4.42	
magnets	N52 Strong Disc Rare Earth Neodymium Magnet	10	8.75	https://goo.gl/wdx7J3
In line fuse	in line fuse 10A, 4A	4	9.58	HSC
Miscellaneous	Miscellaneous	1	44	HSC
			89.65	

Spending was well within budget as all the items used on prototyping were ready in-house. Other special components such as gyroscope was bought via Adafruit and eBay. Most of the components needed to construct the circuit were bought from HSC electronic supply as suggested by Dr. Furman.

Results and Discussion

The figures below indicated and proved that the system had successfully powered the motors to climb the 17-degree slope. The mechanical bogie had carried the electrical components with no signs of wear and danger at first. However, the steering and the gearbox on the propulsion had broken and track switching were not able to perform. Wires were tied to the bogie to maintain the angle of the steering arm to make sure the bogie stayed on track. The bogie initially climbed to the top after the 17-degree slope, without entering the cross-rails where steering was essential. The bogie stopped safely after given a time to proper the motor through iteration in the Arduino code. After a few hours of testing on the bogie movement, the slipping and backlash on the gearbox or motor was speculated to occur, therefore the bogie could not climb to the top efficiently. The control system on the LCD was able to perform as expected and manual human input was successful in moving the bogie according to the operator's command.

Figure 2-77. Bogie seen climbing the 17-degree slope.

Figure 2-78. Bogie seen propelling up the slope.

Conclusion and Suggestions for Future Work

The initial prototype of the control system was based on station switching and LCD panel controls. It is learned that the importance of making the electrical components work and power distribution maintain top priority in this team. Delays in getting the components make testing harder and mostly unpredictable as small scale motors do not scale their strength and power consumption linearly. More circuitry knowledge is required for future work as there are countless possibilities in making the bogie move as well as making it safe to operate. Circuits building is essential when showcasing in Maker Faire and presentation, as to show that professional work is present. PCB circuitry layout is recommended for future work as well as the fabrication of it.

Some of the power systems tested did meet the team's expectations but components may fail over time. Testing out different system is essential but the simplest always work best for a small-scale prototype. Future team should consider the number of ways that could power the electrical circuits, electrical components as well as the LCD control panel. The team learned that wayside could not power the bogie entirely because connections between the shoe and the rail may fail. Backup batteries are recommended to address this problem. Switches and sensors are also important when testing the bogie movement, they often needed to go as a pair as the bogie moves along the track. Switches that fail may pose a safety threat to the system and the problem must be addressed first before any test is carried on.

Non-electrical components are also needed to be considered. Cases should be 3d printed for safety and aesthetic reasons. Refer to Appendix B4-1. Though these cases may take time to design but should be considered when the system is completed ahead of time. Many young audience in Maker Faire will need good practices and examples on how to manage electrical components, in this case 3d printed cases are desired. In conclusion, future team is also recommended to get their hands on other subteams' components before the official testing dates so that a window of problem solving time could be scheduled.

Wayside Power

Background and context

Half-scale wayside team focuses on the improvement of the power pickup system and provide stable current for mechatronics system of the bogies. The existing wayside conductive rails were not able to carry a continuous current safely, and the power pickup system was not able to pick up the voltage from the rail and transfer to the bogie's mechatronics. This year wayside team has significantly improved the power pickup system by using industrial rating current collector shoes. Moreover, conductive rails were also redesigned to improve the safety factor and mobility in case of transporting to exhibition sites.

Description of the Subteam and Objectives

Firstly, the power must be available for a moving vehicle; this means it must remain in contact with the power source at all times. Other railway systems employ electrification rails for this issue.

1. Create a mobile source of power
    a. Must be stable without gaps or drops in current – Stability within the entire system must be accounted for since any disconnection must be brief; reliance on the onboard battery must be minimized for emergencies only.
    b. Must uphold specified power output – The nominal voltage is above 50 volts; the design must account for this and avoid ohmic heating
    c. Must be resilient
2. Implement a wayside rail
    a. Must be modular – If the entire rail system is to be moved, the wayside rail must also be modular to ensure site construction and demonstration is possible.
    b. Must be safe, insulate and isolate current from passengers – At nominal voltage, there is a risk of electrocution if passengers contact the rail; safety must be assured above all things.
    c. Must be cost effective – Since the track will be spanning over a long distance, manufacturing these rails will become costly with every increase in distance; minimizing cost for fabrication will mitigate high expenses.
    d. Must conduct past 60V – If arcing or ohmic heating occurs at the specified voltage, the rail system will defect.
    e. Must be resilient
3. Implement a collector system
    a. Must maintain contact with rail at all times – A spring mechanism will allow for contact to be stable and maintain contact throughout travel.
    b. Must conduct total rail current/voltage – Again, ohmic heating and arcing must be avoided.

c. Must intuitively jump power through to bogie integration – Intuitive wiring should be supplied to the integration aspect of the overall bogie connections; ours simply outputs positive and ground.

Design Requirements and Specifications

Wayside power system for the half scale model consists of 4 conductive rails (2 on each side of the track), and current runs constantly in the conductive rails. Current collectors are attached directly to the bogie that will slide along the conductive rails to pick up the current and provide electricity for the electrical components within bogie and pods. The rails will be hooked up to a 48V 20Ah LiFePO4 battery and provide a stable voltage to be picked up by current collector arms as bogies slide along the track. The specifications of the battery used can be seen in Table 2-4 below.

Table 2-4. Battery Specifications (LiFePO4).

Item	SPEC	Remark
Nominal Capacity	20Ah	0.2C Discharge
Minimum Capacity	19Ah	0.2C Discharge
Nominal Voltage	48V	
Charging Voltage	58.4V	
Cell Overcharge Voltage	3.75V	
Cell Overdischarge Voltage	2.7V	
Charging/Discharging Current (Std.)	4A	Ambient and Battery Cell temperature -20~+65℃ (efficiency α T cycle life α 1/T)
Continuing Charging Current(Max.)	75A	Ambient temp:0~+45℃
Peak Charging Current(Max.)	100A	＜3mins
Continuing Discharging Current(Max.)	75A	Ambient temp:0~+45℃
Peak Discharging Current(Max.)	100A	＜3mins

The cross-sectional area of the conductive rail inline was also calculated to specifications to match the correct gage (8 AWG) of ampacity as defined by the nominal voltage requested (Table 2-5).

Table 2-5. Gage comparison to specified current.

American Wire Gauge	Diameter	Diameter	Cross Sectional Area		Cross Sectional Area of Copper Strip
(#AWG)	*(inches)*	*(mm)*	*(mm^2)*		*(mm^2)*
4	0.204	5.19	21.1		9.6774
5	0.182	4.62	16.8		
6	0.162	4.11	13.3		
7	0.144	3.67	10.6		
8	0.129	3.26	8.36		

State-of-the-Art/Literature Review

Automated transit networks (ATN) systems is an emerging type of transportation option for individual or small group travel, typically ranging from 3 to 6 passengers in each vehicle. Topological networks are organized for guideway which allow for the vehicles to follow a laid-out network with the stations located on sidings. The methodology in the arrangement of the tracks involves frequent merging and diverging points to allow for nonstop travel by bypassing all intermediate stations.

To this date there are 5 fully integrated advanced automated transit network (ATN) systems. Boeing manufactured the Morgantown personal rapid transit (PRT) in Morgantown, West Virginia, US with up to 20 passengers per vehicle qualifying it as a group rapid transit system as well. The guideway spans approximately 13.2 km with 5 stations and over 73 vehicles. During low usage periods the system disables point-to-point rides (Gibson, 2008). ParkShuttle manufactured by 2getthere is a group rapid transit (GRT) built in Rivium, Netherlands; the system is a 2nd generation GRT that can accommodate up to 24 passengers. The vehicle operates on schedule during peak hours with intervals between each other only approximately 2.5 minutes apart. Off of peak hours, the GRT can operate on demand it is currently being secured for its future operation beyond 2016 (Morge, 2009). CyberCab is another PRT that was manufactured by 2getthere, built in Masdar City, Abu Dhabi. This PRT system was planned to be a solution to all transportation means with cars to be banned upon finishing; the initial network was not completed and did not expand enough to follow through with the plans. This ATN is one of the only type that has designated freight vehicles but have not been fully utilized yet (Cottrell, 2008). ULTra PRT created by ULTra in Heathrow, England for the Heathrow Airport was established and operational in 2011. It connected Terminal 5 with a long-term car park. A 5-year draft was made in 2014 to extend the system further throughout the airport. The current design spans across 3.8 km along the airport (Lowson, 2001). Skycube, designed and built by Vectus, was formed in Suncheon, South Korea. It travels a total of 4.64 km and connects the site of the Suncheon Garden Expo Korea (2013) to a station in the wetlands to allow access to the Suncheon Literature Museum. The line runs along the Suncheon-dong Stream with the stations directly connected on the line (Gustafsson, 2009).

Currently the design for the wayside power on the Spartan Superway incorporates the use of a 4th rail system. Our design utilizes a fourth rail system that is attached parallel to the track; a fourth rail system is essentially 2 lines of conductive material that is attached to a power source, with one rail reserved for current and the other for return. These rails will have a specified current and voltage derived from the power source and will be tapped into using a collector shoe that is attached to the bogie. Fourth rail design was chosen because it isolates current from running rails (electricity only flows in conductive rail which is housed and safer), can be easily implemented and maintained as opposed to other wayside systems that use the running rails to conduct electricity and it builds on previous design; utilizing available material. Housing the conductive rail in an insulated material and having the conductive strip placed on the wall allows for benefits such as preventing build up from rain and dirt, as opposed to being placed on bottom concave, and protecting passengers from making contact with the conductive rail. The conductive material, copper was chosen due to its cost efficiency, aluminum was considered however copper has higher conductivity, it is more ductile, it has relatively high tensile strength, and it can be easily soldered, with copper having a specific resistance 37% less than aluminum.

Description of Your Design

The wayside system consists of two parts conductive rails and current pickup arms. Conductive rails for each track (straight track and sloped track) consist of a forward current and returning current rails. The two rails are placed in parallel and separated by wooden brackets that clamped into the track structure. The rails on top of each track (positive) are wired at the end of the track as well as the bottom rails (negative). In other words, the conductive rails of each tracks are wired in parallel and connect directly into the battery as in Figure 2-79.

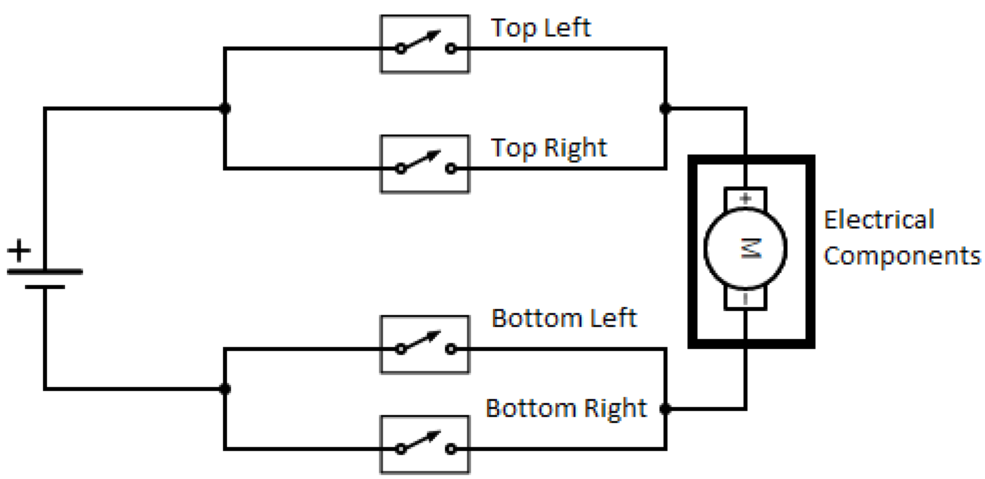


Figure 2-79. Schematic of the wayside conductive rails.

Conductive rails are made with copper wires glued inside the lining of schedule 40 PVC pipes. The conductive rails are made by smaller section from 6-10 ft to reduce the difficulty of disassembly and transport them to the exhibition sites (Figure 2-80). Initially, we propose to use 1 inch wide, 0.02-inch thick copper strips. The mentioned copper strip will enable us to fabricate more reliable conductive rails. However, the proposed budget was not approved; therefore, we proceeded to use the existing conductive rails and manufacture more rails to replace some of the defective sections. The existing rails use 4 AWG wire flattened into 6 gauge, since the wire are too thick fabrication has been difficult to work with.

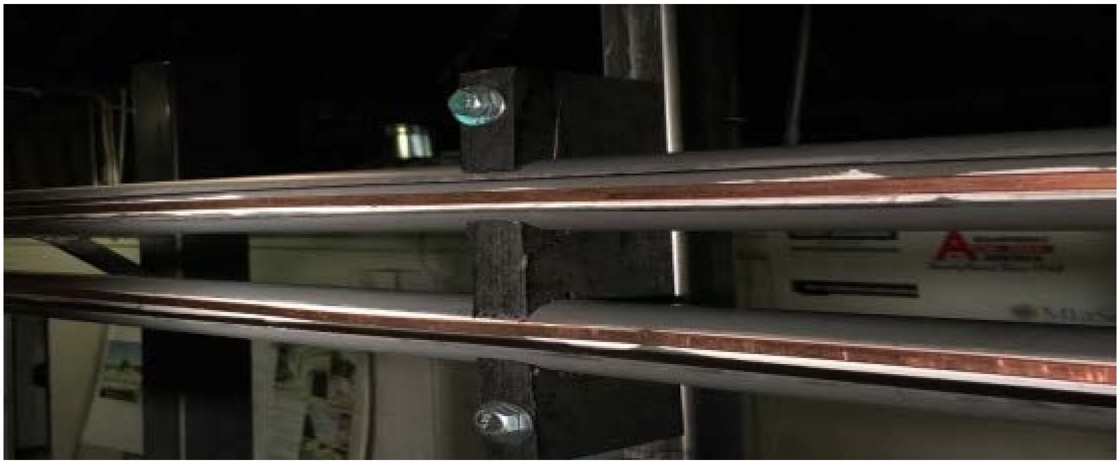


Figure 2-80. Overview of conductive rails and brackets clamped onto track structure.

On the other hand, the current will be picked up by current collective arms. The arms consist of copper shoes as shown in Figure 2-81 which make primary contact with the conductive rails, and four-bar linkage arms to provide more degree of freedom when sliding along the track.

Figure 2-81. Copper shoes of the current collector arms.

Current collector shoes are attached on the designed plates on the bogies. The wires from the current collector arm are wired to the inverter of the mechatronics system (Figure 2-82).

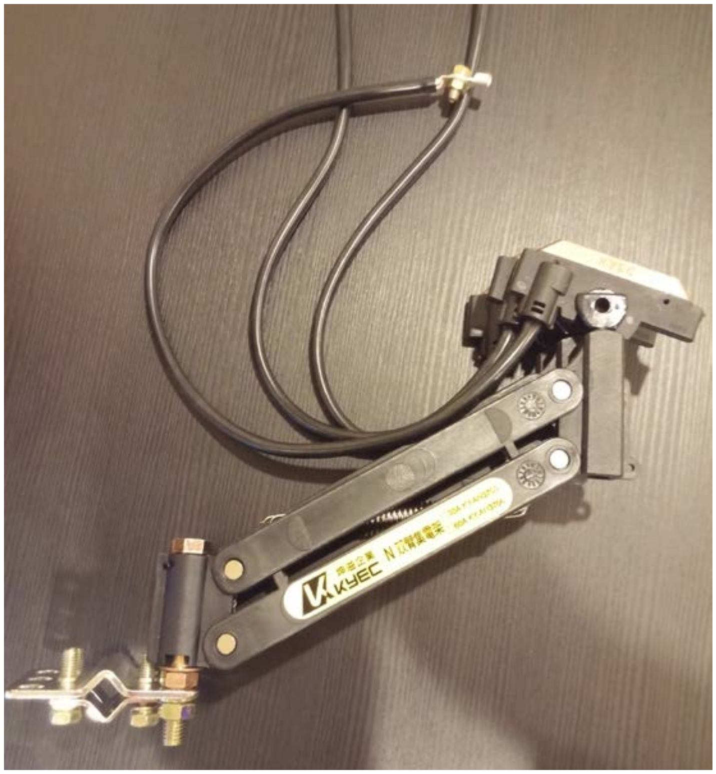


Figure 2-82. Current collector arms overview.

Analysis/Validation/Testing

Testing voltage of the rails and current collector arms:

The battery is fully charged and connected to DMM which shows a value between 54-56 V. Then it is connected to the conductive rails system, and we proceed to connect the DMM to the rail to detect the voltage drop. The voltage measured by DMM on the rail is exactly as the voltage we measure directly from the battery (Figure 2-83).

Figure 2-83. Testing voltage through rails.

Testing of the battery:

In the simulation of the characteristic graph of the battery, Simulink have used a generic dynamic model for the battery blocks. Thus, since every type of battery contains different characteristic, the formula using to plot the characteristic graph is different. The equation below (Figure 2-84) is used in the Lithium-ion battery dynamic simulation.

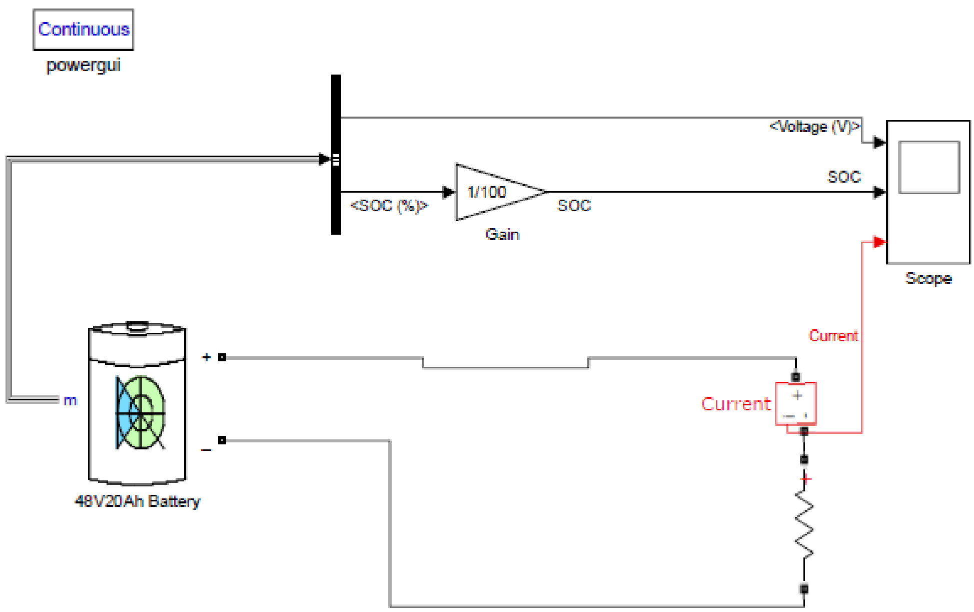


Figure 2-84. Discharging Battery Circuit in Simulink.

Result of the simulation:

The 1C current rate could provide the constant voltage in a longest time, which is about 50 minutes. When the current rate is decreasing, the time will be shorter to remain the steady voltage.

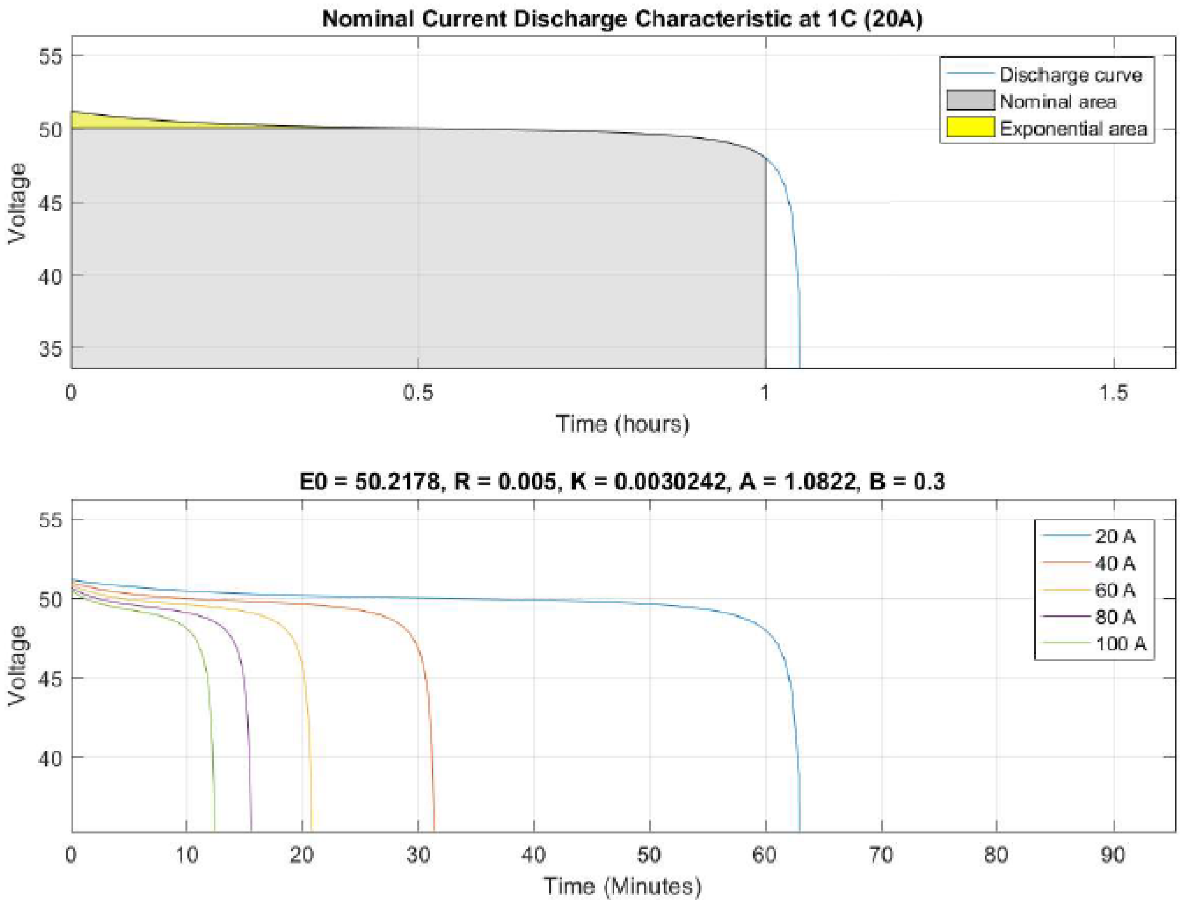


Figure 2-85. Nominal Current Discharge Characteristic Graph in Time.

In Figure 2-86 below, the capacity of the battery could be observed when the battery is discharging. The most efficient current rate is 1C because it will contain largest capacity when it remains the same voltage. In addition, the discharge characteristic is plotted when applying the Simulink discharging formula, and data of E0, R, K, A and B shown in figures are used in the formula, which are calculated by the program.

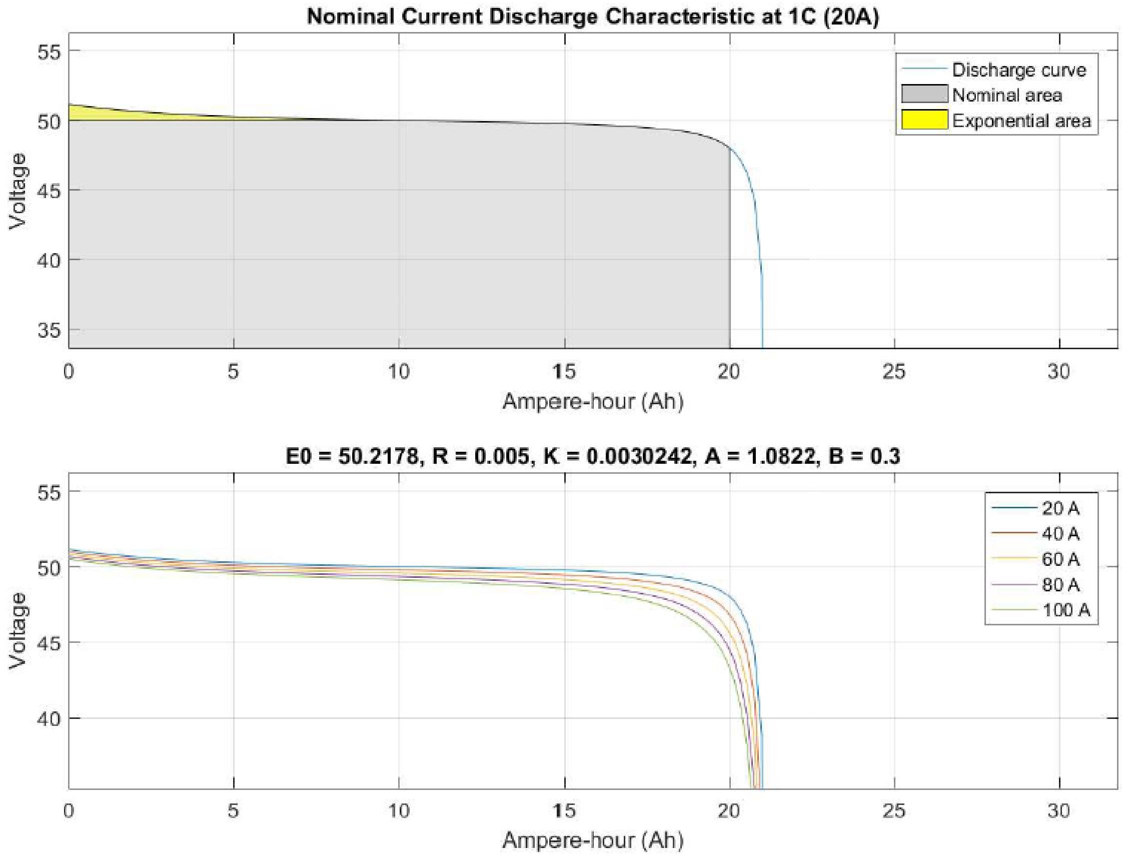


Figure 2-86. Nominal Current Discharge Characteristic Graph in Capacities.

The next graph (Figure 2-87) represents the change of voltage, state of charge and the current. When battery is discharging, the battery will provide a steady voltage during a period of time. Since the voltage is steady, the current will be steady, too. After a certain time, the state of charge is low, and the voltage started to drop. The point where the voltage will drop drastically is when the battery reach the nominal voltage of 48V. Since the voltage is dropped, the current will be decreased as well, due to the simulation having a constant RLC load. In this simulation, state of charge is calculated to last approximately 1.4 hours for the battery to go from 100% to 0%.

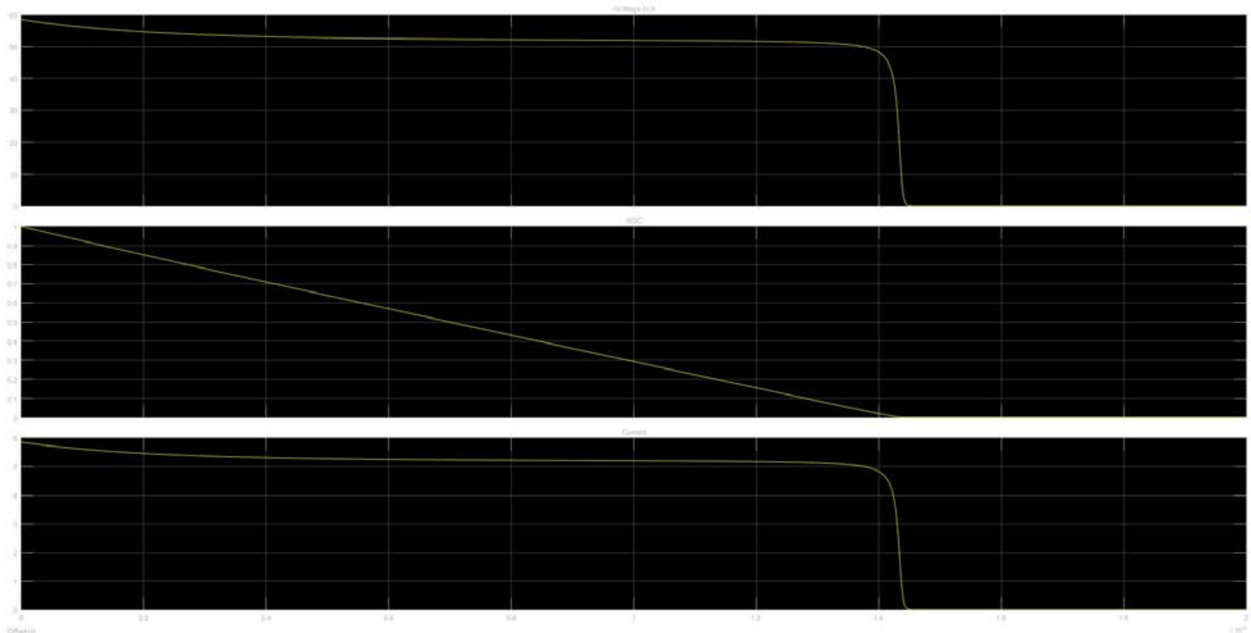


Figure 2-87. Voltage, State of Charge and Current Behavior during the Simulation.

Procedure / Instruction Manual

Conductive rail

To utilize this conductive rail, the shoe collector must maintain in contact with the copper strip at all times. Connecting the battery plug acceptor to the battery output will supply power to the rails. This connection can be converted to accept other power sources by simply connecting positive and negative leads to the terminals.

To replicate the conductive rail, the process requires PVC piping, Copper strips, adhesive (silicon based preferably) and 1/8th screws and nuts. Take the PVC piping and cut out an arc length that spans a fourth of the circumference along the tube so that the inner side is exposed through this opening. Apply adhesive while inlaying the copper strip from end to end. Once the materials have set, drill a hole through the ends so that the screws can connect with other rails modularly. Bumps can be remedied by grooving out the PVC in areas of connection.

The rail is to sit alongside the track in place held by the brackets. Slip the rail in the piping according to their incremental number labeling (all rails have been numbered with their associative place on the half scale model - this can be seen on the track and on the rails). Connect the rails and attach the terminals to the power source.

Brackets

To utilize the brackets, they must sit alongside the track and uphold the rail accordingly. The size of the hole is dependent on the size of the PVC pipe used.

The model for the brackets can be found in Appendix B5-1. Bore two holes into a plank of the denoted dimensions and drill 2 holes through the sides. The holes must be drilled through the face of the bracket for only 2 sections of the rail (lower stop section)

Append the brackets to the rail by bolting them to the holes drilled into the track support system. The lower section of the track requires bolts through the face of the bracket.

Shoe Collector

Usage

The shoe collector model we used is sourced by KYEC, to use this collector, simply attach the item to the plates marked on the bogie and adjust the height before tightening the screws. Maintain contact with the conductive rails and identify any gaps to adjust the track or the collector.

Fabrication

Commercially sourced products can be found in the budget section of this report.

Installation

The shoe collector requires a single fastener to implement onto the bogie. Using the L shaped plate, there are holes that correspond to where the collector can be fitted onto. Affix the collector and position appropriately. The leading wire can then be attached to an inverter and a power source. Take precautions as arcing may occur with the power supply box.

Money spent on your project

Table 2-6. Money spent on project.

Item Requested	Specifications	Quantity	Unit Cost	Shipping	Amount
20 Mil Copper Strip 0.75 X 100’	100 ft, 20mm, 0.75”	4	129.99	0.00	519.96
Current Collector	60A, Single Arm	6	13.20	58.00	137.20
Conductive Lubricant	10 ounces, T_{max} = 400°F	2	10.00	5.39	25.39
Total (only includes items in blue)					162.59

NOTE: ITEMS IN RED NOT PURCHASED

1. The copper strip will be lined alongside the track in pairs as a 4^{th} rail design. The estimated length required is approximately 280 ft. according to the current half-scale model. It is suggested that this type of copper strip be used in future systems, to increase contact area.

2. The current collector is a single arm attachment that will be able to transfer the necessary power requirements to the bogie. This is of high priority to continue designing an appropriate integration onto the chassis.

Importance to Project

The copper strip is necessary to line the rail housing with conductive material so that electricity can be delivered; this is crucial to the entire wayside system, otherwise there is no way electricity can be brought to the bogie.

The collector is necessary to retrieve the energy running in the conductive rail and connect to the 4^{th} rail ground. A commercially rated shoe is highly recommended since testing has been done to confirm specifications to a degree that cannot be achieved in the workshop.

Conductive lubricant is necessary to extend the life of the copper strip and the shoe collector by mitigating corrosion. It will also improve the conductivity of the entire system as well as reduce the operating temperature due to friction. The minimization in friction will also prevent a dragging effect due to the fourth rail design.

Availability in Shop

There is copper in the shop, however it is not copper that can be used for the design that we are implementing; due to the dimensions (thickness mainly, as well as the copper tube being unable to fit appropriately in the current insulation housing) we will require a thinner, and wider strip that is malleable enough to appropriately fit the curves that the half scale track runs on.

The shoe and lubricant are not present in the shop.

Results and Discussion

The current was stable throughout the half scale model and the rail conductivity was tested once installed. Firstly, we tested the maximum amperage along the track with a multimeter to assure the correct output was being supplied. We then determined the reliability of this current by finding gaps in the rail that would cause the circuit to terminate unintentionally - these gaps were filled either by adjusting the fit of the rail or by using copper tape lining to connect the gap.

The collector was also tested to determine if power was able to be picked up from our fourth rail design; we had to assure that there would be a correct power rating that goes onto the wire terminals that exited the collector shoe as this would be the contact where the bogie would draw power from. The collector was also found to be capable of conducting the maximum power we intended to output onto the system to determine no bottlenecking would occur with the shoe.

Conclusions and Suggestions for Future Work

Although we achieved our goals, there are specifications we would have liked to surpass and even design types would have liked to implement. The next team can hopefully pick up on these things. The first would be to implement a more uniform design for the rail; order in a conductive copper rail inlay that has a greater area to maximize contact stability, and create a more uniform housing design as opposed to a cutout PVC tubing insulation. The second would be to begin converting the power source from the battery connected to solar array sourcing. The solar panels can be set up and routed to the conductive rail, bringing the overall system closer to the envisioned minimized footprint of the overall Spartan Superway goal. The brackets can also be replaced using a different material; an insulating plastic may be easier to fabricate in bulk. The design of the brackets can also be altered so that their profile is less intrusive as it is now. Optimizing the modular aspect of the rail is also recommended; this can be done by creating a connector for the rails that can easily be latched in for ease of setup. These are just a few things that can improve the current system, further considerations can be suggested by the upcoming team.

Chapter 3: Full-Scale

Railings and Supports

Background and context

The current model for the Spartan Superway - half scale and full scale; do not actually have a fully developed column system for structural integrity. Creating a structurally sound column is important for any structure, especially since the Superway is going to be suspended above ground. Looking at both half scale and 1/12th scale teams, it is too obvious that there wasn't much effort into equipping a properly designed column. This was our team's main goal in the Fall semester. Continuing our work in the Spring semester, we were given a new task - create an architectural model for potential clients. With the flexibility our team brought for the Superway, we were able to change our original direction and get working on this new goals for the Spring semester.

Description of the Subteam and Objectives

Our focus this semester largely revolved around an aesthetic point of view. Whereas last semester focused on structural calculations and strength of materials, we tried to be more artistic this semester. Our main focus was the architectural model in which we showcased plausible structures to be implemented onto the full-scale Spartan Superway model at some point in the near future. This was done in three main parts as mentioned earlier: A Palm Tree model shown in Figure 3-1, a wooden Cat's Eye Model shown in Figure 3-2, and a PVC Cat's Eye model shown in Figure 3-3. Note that these are progressive images, as final copies will be included later in the report.

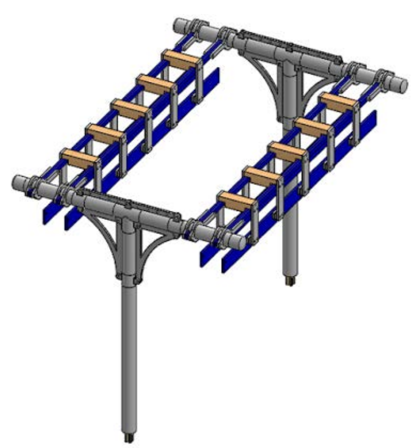


Figure 3-1. CAD Image of Palm Tree Model Including Laser Cut Acrylic Guideway.

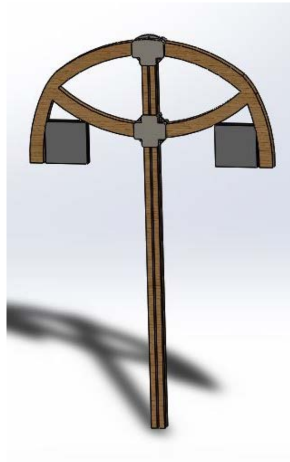


Figure 3-2. Wooden Cat's Eye CAD Model for Purpose of Designing Clamps.

Figure 3-3. PVC Cat's Eye Design after Being Molded to Shape.

The subteam was mainly tasked with the general concept of improving the support structure of the entire Spartan Superway Full Scale test track. Because that was too daunting of a task for our group, since we had no prior experience, our finished product resulted in working on the 12th-scale figure. For this semester, our main focus was the architectural model, though we also reflected on the X-beam from the first semester. We had been in contact with our contact from South Africa who visited us and claimed that his team could design a small section of column for us to gain experience by manufacturing it. Unfortunately, due to miscommunication and time constraints, this effort was extremely delayed, but we expect it to be done before Maker Faire. Ultimately, we finished all our tasks on time.

Specific objectives for subteam:

- Create an architectural model of the support structure featuring 3-4 original designs (Palm Tree, wooden Cat's Eye, PVC Cat's Eye)
- In the architectural model, include:
    - Wooden X-beam columns as main supports

- 3D printed clamps to avoid welding and bolting into column
    - Laser cut acrylic to show guideway section
    - 3D printed brackets to demonstrate where the guideway would be attached on the other model designs
    - Some type of original footing (PVC flanges on a wooden box, the wooden Cat's Eye model also included a concrete base)
    - If welding or bolting would be necessary, explain where it would be and why
- Create a 4-foot tall stub X-beam column of 1/8th inch thickness folded steel, manufactured by Vander-Bend
- Communicate with other subteams within Spartan Superway, though our project didn't interfere or coincide with anyone else's scope of work

Design Requirements and Specifications

The Supports and Railings team was tasked with creating a structure profile to attach the Superway guideway to. Requirements were fairly loose. The team was given a basic palm-tree shaped profile that clients found aesthetically pleasing and some general dimensions that the final design should be sized to (See Figure 3-4). The structure should allow for the integration of a guideway for travel in two directions. The structure should allow for the ability to easily install a solar racking system to provide the energy needed for the system. Additionally, it would have to integrate Milotek's X-beams and would therefore need to be attached without the use of riveting, welding, drilling or other methods that would compromise the X-beam's strength. The weight of the system should be supported and force of the guideway and cabins must be effectively transferred onto the X-beam.

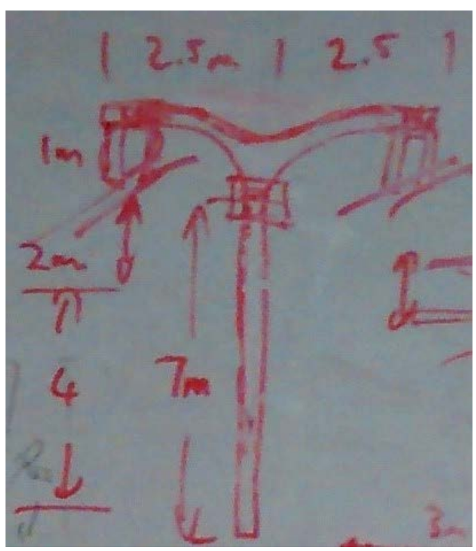


Figure 3-4. The Requested Parameters and the Palm Tree design.

State-of-the-Art/Literature Review

ATN Technology

The designation of an automated transit network (ATN) is a fairly recent concept. Prior to 2010, the ATN concept was referred to as under the more general umbrella idea of personal rapid transit (PRT) (Furman, Fabian, Ellis, Muller, Swenson). There are seven established key features that qualify a transit mode as an ATN according to the Advanced Transit Association (ATRA) guidelines:

1. Direct origin-to-destination service with no need to transfer or stop at intermediate stations
2. Small vehicles available for the exclusive use of an individual or small group traveling together by choice
3. Service available on demand by the user rather than on fixed schedules
4. Fully automated vehicles (no human drivers) that can be available for use 24 hours a day, seven days a week
5. Vehicles captive to a guideway that is reserved for their exclusive use
6. Small guideways, usually elevated but also at or near ground level or underground
7. Vehicles able to use all guideways and stations on a fully connected network

ATN is unique in that it integrates ideas from both PRT and automated guideway transit (AGT). The network consists of a closed guideway reserved solely for the use of the transit system. Like PRT, the system runs similarly to a taxi in that is on-demand and goes directly to the passenger's desired destination. Functioning like a AGT, the system is automated and driverless, relying on a central control management to operate vehicles.

There are currently only five operational ATN systems in the world that meet all of ATRA's guidelines. They are as follows: Morgantown PRT, ParkShuttle, Masdar PRT, Heathrow T5, and Skycube.

The earliest instance is the Morgantown PRT at West Virginia University. The system has connected the University's three Morgantown campuses since 1975 in a simple linear fashion along a railway with five stations (West Virginia University). Each vehicle can carry 8 to 20 passengers.

The ParkShuttle system in the Netherlands utilizes autonomous vehicles. They appear bus-like but are driverless and are guided by magnetic guidewires laid in the road (2getthere, 2016). Each vehicle can carry 12 seated passengers.

The Masdar PRT has been servicing the carbon neutral, car-free city of Masdar City, Abu Dhabi since 2010 (2getthere, Masdar). The 10 on-demand vehicles are powered by rechargeable lithium batteries and also utilize inlaid magnetic guideway strips. Like the Superway, the Masdar PRT is based upon a system that runs entirely on clean energy. Each can carry up to 4 adults and 2 children. Although the system is technically an ATN, it is made of a single linear route and functions much like a shuttle.

The ULTra PRT system in Terminal 5 of Heathrow Airport in London runs between the terminal and its business passenger car park 2.4 miles away (ARUP). Eighteen pods run on a largely elevated guideway.

Each pod can carry up to 4 passengers and their luggage. With the exception of a fork at one end of the route, the system is also limited to shuttle-like functionality.

The most recently completed network is the Skycube system in Suncheon Bay, South Korea. Forty vehicles follow a largely elevated linear guideway between the two stations at Suncheon Bay International Wetlands Center and Suncheon Literature Museum (Tebay, 2013). Each vehicle can transport 6 passengers. Again, the single guideway route limits functionality to that of a shuttle-like system. This system is most similar to the Spartan Superway in that it is located on its own elevated guideway, however the Superway guideway is inverted.

Overall, the functionality of current systems is not at full potential. Many run on restrictive routes that essentially make the system an eco-friendly shuttle. Improvement can be made with expansion of routes and planning of a comprehensive network of guideways.

Interlocking Columns

Work completed this semester is based upon X-beams created by the team's contact company Milotek. These beams are an advanced variation of Concrete Filled Tubes (CFT), made by bending high-strength steel to shape and filling them with concrete. So long as the bent steel remains unaltered (drilled, welded, or otherwise) the properties of the steel effectively improve the strength of the beam. This results in columns that are much stronger than traditional CFTs. In application, the usage of these columns will allow for thinner columns that are spaced farther apart which ideal in city settings.

Description of Your Design

The first model we wanted to create was a 4-foot stub of the interlocking X-beam to give perspective and a means for future groups to design clamp systems, as well as letting Vander-Bend practice manufacturing them. Based on the dimensions from an official document (which had nominal dimensions of 160mm for the sides, flange width of 70mm, and 3mm sheet thickness), we chose the dimensions shown with 6.5in for the sides, 3.5in flange width, and 1/8in sheet thickness (see Figure 3-5). We asked Milotek if they could create the laser cutter pattern for Vander-Bend, but due to unforeseen complications, the pattern arrived very late in the semester, and Vander-Bend did not have time to make it before this report was written. However, it was completed just before Maker Faire, and resembles like a larger version of the beam in Figure 3-6.

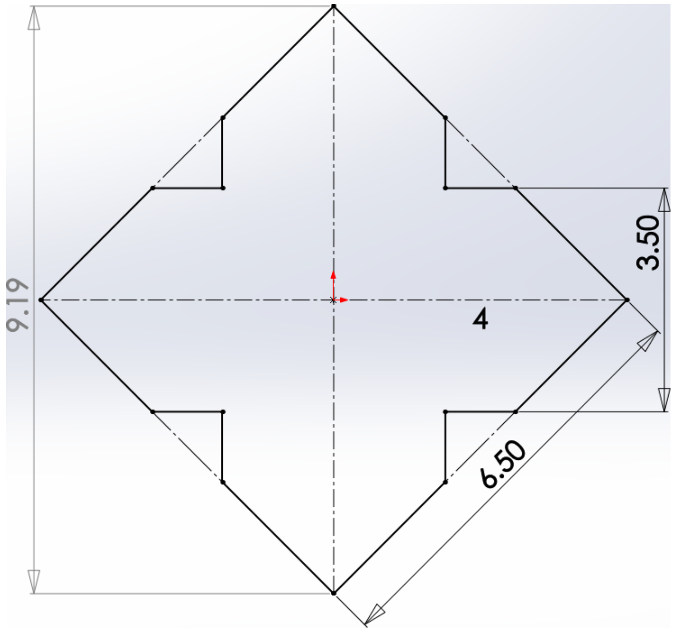


Figure 3-5. Cross Section Dimensions for our Stub.

Figure 3-6. A Donated X-beam Stub.

In response to the palm tree sketch from Figure 3-4, we began studying its viability. Unfortunately, we realized quickly that this design branching component would not be a practical as it would create a major weak point and fails to distribute the load properly. As such, we came to two different designs that attempted to maintain a similar aesthetic while fully taking advantage of the vertical X-beam's strength; the Cat's Eye and the T-support. In both of these models, multiple clamps are used to redistribute the loads and to maintain a better connection with the vertical X-beam.

Shown in Figure 3-7, Figure 3-8, and Figure 3-9, The Cat's Eye is a general design named after its shape (looking like an eye with a vertical pupil), and consists of the vertical X-beam and three support arc (made with curved I-beams or CFTs). It works by having one large upper arc connected directly to the top of the X-beam (preventing vertical motion) and two convex lower arcs connected to the side (preventing horizontal motion). The design fully takes advantage of the weight of the beams by redistributing the load to the much stronger X-beam, lowering the requirements that the clamps and the arcs will need to endure, which will additionally lower cost. It should also be noted that the upper arc is also ideal for mounting a curved solar roof. We created two models (Figure 3-10 and Figure 3-11), one primarily from wood and the other from PVC pipes, for these, the exact (relative) dimensions were not particularly significant, instead the objective was to showcase the unorthodox geometry and ways to connect a guideway bracket.

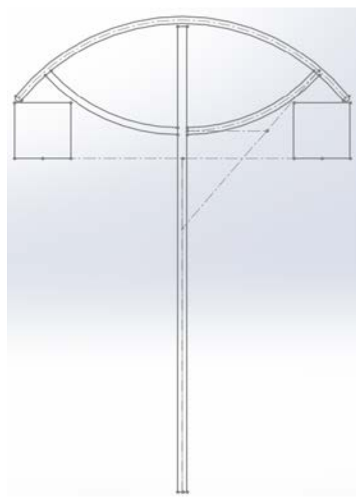


Figure 3-7. Different Types of Cat's Eye Designs (1).

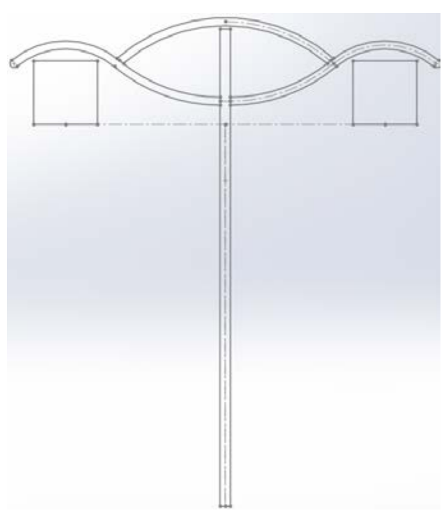


Figure 3-8. Different Types of Cat's Eye Designs (2).

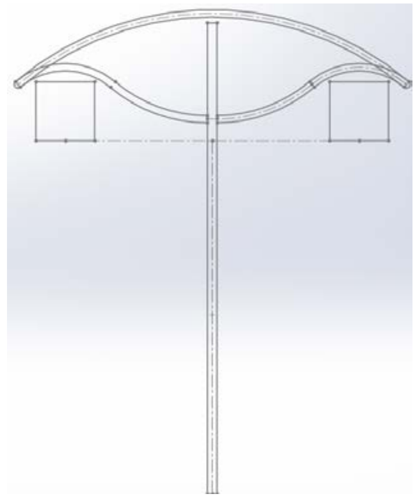


Figure 3-9. Different Types of Cat's Eye Designs (3).

Figure 3-10. The wood model.

Figure 3-11. The PVC Cat's Eye Model.

The T-support design may appear simple at first, but due to its extremely modular nature it can be very versatile. As shown in Figure 3-12, in this design there is an additional X-beam column placed horizontally above the vertical X-beam to support the rails. There are at least four main pairs of clamps, one at the intersection, two on the sides, and one below, which are interconnected through the standard I-beams or CFTs. Of these, the diagonal connection can be done with a concave shape to fulfill the Palm Tree criteria, and the upper beam can be modified to accommodate customizable attachments such as a solar panel system or a suspension cable tower for additional guideway support. For the model representing it we made sure to keep it as close to 1/12th scale as possible and produced two such supports. Wooden internal beams similar to those in the Wood Model to represent the X-beams. PVC pipes were also used for additional structural support and a smooth look (though they would only be cosmetic sheet metal covers in the real design). The clamps and end caps were made with 3D printed ABS. The secondary beams were made with laser cut acrylic, (with the upper beams having holes every inch, to allow for an unfinished solar panel assembly). In addition, we made a guideway system based on one of Milotek's systems that would be able to hold the 12th scale group's bogie. It is made primarily from laser cut acrylic, with the exception of the upper wooden spacers. This guideway uses two upper support beams that are connected perpendicularly to the T-support (in the real design there would be a special clamp to connect it to the beam, but in the model, it is connected to the PVC cover), and has a series of brackets along it that holds the rails up and keeps everything held a certain distance. The completed model is shown in Figure 3-13.

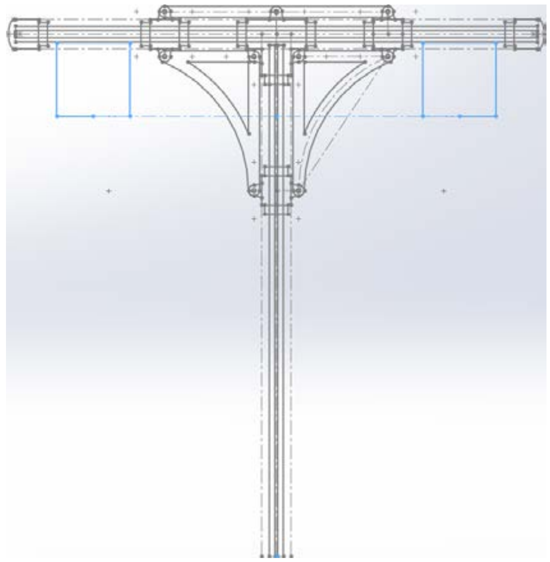


Figure 3-12. The T-Support design.

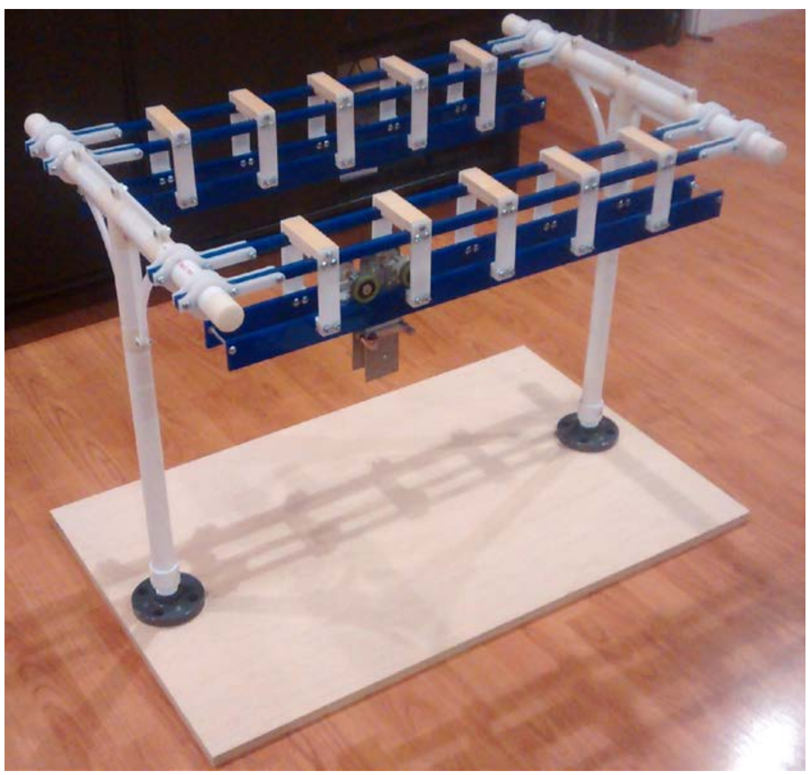


Figure 3-13. The Completed T-Support Model.

Analysis/Validation/Testing

As the aesthetics, parameters, and chosen material properties were still very fluid at this stage of development, we chose not to perform conventional mathematical testing or simulations. Instead, we continuously made modifications to the designs and their respective models to more easily explore different possibilities.

At the beginning of the semester, we looked at the initial palm tree design and immediately saw that it would not only be difficult to manufacture, but also be structurally weak. We considered multiple solutions to overcome these, such as a large multipart clamp with a chain of curved CFTs (Figure 3-14) and a using bridging beam to keep two separate branches together (Figure 3-15). However, even these were designs are fairly unstable as they still had only one point of contact with the X-beam, and the upper portion take the entire load of the guideways and bogie on their own, meaning that they would have to be needlessly expensive to meet requirements.

Figure 3-14. Multi Part Clamp.

Figure 3-15. Bridge for Branches.

To remedy these shortcomings, the Cat's Eye design was created. However, our supervisors cautioned us that this design was beginning to stray from the original request, as such we began development on the T-design.

Figure 3-16. Solid version of T-Support.

Figure 3-17. Truss version of T-Support.

We showed the Cat's Eye pattern and the Truss version of the T-support to Andries Louw, our contact, in a teleconference, and he pointed out a few flaws in each. For the Cat's Eye, having the upper arc be curved takes away from its strength significantly. For the T-support, the cover would make the system look too bulky, and more importantly the upper protrusion with holes was not cost effective.

Following this, we began planning and producing the models. For the Cat's Eye models, the general design was not significantly changed, and exact scaling was deemed to not be important as the goal was simply to showcase the basic concepts and potential aesthetics. Effectively, the main test was to confirm that the custom manufactured parts would be usable and aesthetically pleasing.

All of the components in the wood model, were custom built. The first was a miniature X-beam, which was made with a table saw and a router bit so that the wood had a square cross section with a side length of 1in and a triangular cutout 1/8in deep on each side. Second was the support beams which hand milled and had to let the x-beam through. Third, was the 3D printed clamps and guideway brackets which had to hold the aforementioned components together.

For the PVC pipe model, the main custom parts were the heat bent 1in PVC pipes, which had to successfully pass through several 1in PVC connectors. In addition, there were foam covers for the bent pipes that were used to cover the bent pipes and an unbent 1in PVC pipe, both of which had to be cut to lengths that would make the assembly look elegant.

For both of these models there were two additional requirements. First, they had to attach the 1/12th brackets. Second the base was made with a large wood block, and with some pipe flanges, and needed to be able to hold their models firmly.

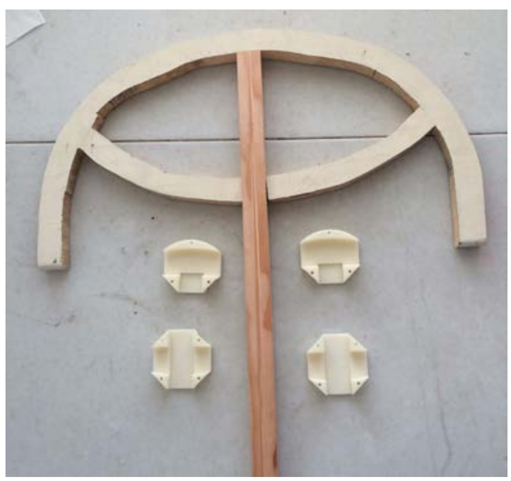


Figure 3-18. Custom Wood Model Components.

Figure 3-19. Heat Bent PVC Pipes.

The T-Shaped design had a few more tests. For one, the ratio of dimension had to remain as close to Figure 3-4's ratio. In addition, it used internal wood beams similar to those in the wood model, but with sides of 5/8in and 1/16 in deep cutouts and rounded edges for the clamps. Second, was the 3D printed clamps, which had to be able to fit firmly into 1 in schedule 40 pipes (internal diameter of 1.049 in). Third, the acrylic parts template had to fit into two 18x24in sheets (originally one), and each part had to firmly connect to other parts: whether beams on the support structures, or the guideway components. The guideway also had to be just the right size to hold the 1/12th scale bogie and to have standardized parts for versatility. Lastly, the completed assembly had to be connected to a wooden base through 1in PVC flanges. These requirements lead to many iterations of the design: for the clamps, the accuracy of the 3D printer lead to the option of making the clamp connections have 1 in diameters, with blue tape wrapped around as an intermediate between them and the PVC pipe. For the acrylic components, the original plan of using one sheet had to be changed when the mechatronics lab's Epilog Helix laser cutter failed to work, causing a complete redesign for the ME 101 Boss laser cutter than required 2 sheets. In addition, there was a miscount of the acrylic parts required, so four parts had to be modified, and four more had to be made entirely. As the flanges were the wrong type (having large bolt holes and a threaded section for the PVC), modifications were required - three holes were drilled in its base to allow for wood screws and adapters were purchased to allow a connection between the pipe and the flange.

Fortunately, after all of the hard work, we had successfully completed the models which passed all of the requirements.

Procedure/Instruction Manual

Since most of the team's prototypes are mainly physical designs, there isn't much need to understanding how they work - but more how they are designed; which is already addressed above. The Wooden and PVC Cat's Eye Models are inert and fully assembled. The X-beam and the T-support model require construction or assembly.

X-beam Construction

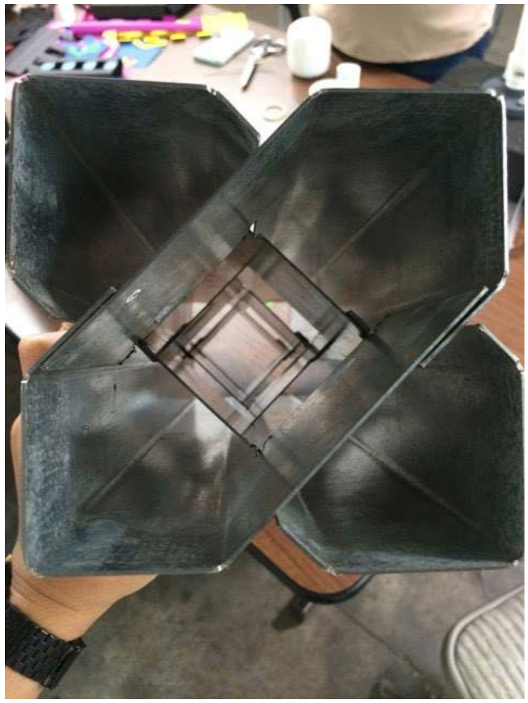


Figure 3-20. Interlacing Design of Donated Model.

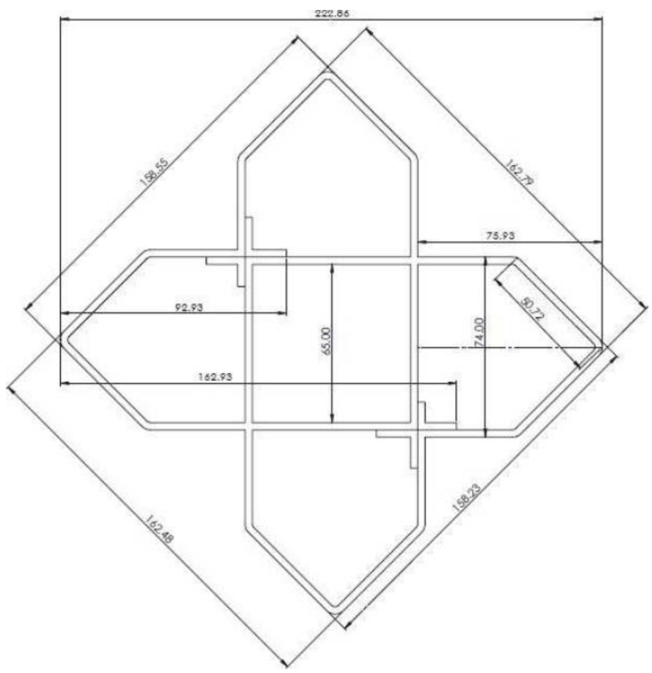


Figure 1: Domex 3*mm* column cross section filled with concrete.

Figure 3-21. Original X-beam Blueprint.

The X-beam stub will be created out of 1/8th in or 3mm sheet metal and fabricated through Vanderbend. Following Figure 3-21's blueprint and advice from Andries, we learned that the sides of the folded tabs must be collinear and that there is an offset and gap for the tab arrangement. These were overlooked on our team's part from last semester's models, but are important going forward.

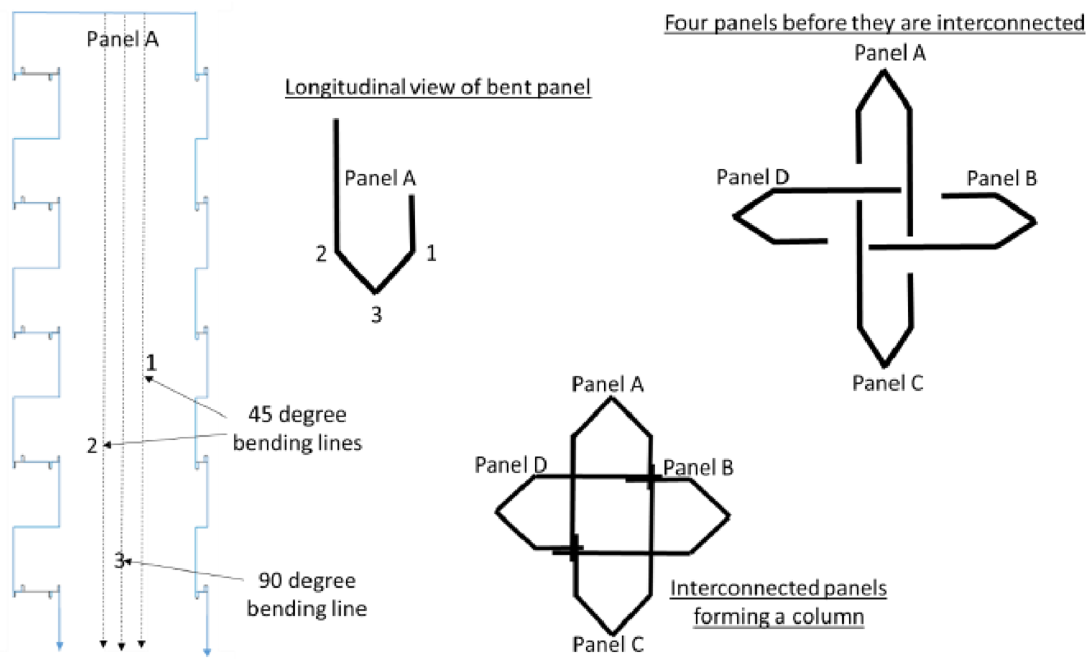


Figure 3-22. X-Beam Panels.

For the manufacturing process, four panels (similar to those in Figure 3-22) are cut from the sheet metal. There should be small gaps at the ends of the lines marked 1, 2, and 3 which are used to show the bending locations. Lines 1 and 2 are bent to 45°/135° angles and line 3 to 90° such that the bent panel forms a symmetric J shape as shown. Two of these bent panels are placed alongside each other so that the slots on the tabs line up with the smaller portion is on the outside, then two other bent panels are slid in from the opposite orientation (with the smaller tab again facing on the outside) and placed into the slits. The final cross should resemble that of Figure 3-20 or Figure 3-21. Afterwards, Concrete is poured into the gaps, resulting in an extremely strong column.

T-Model Assembly

For the following, steps, it is assumed that the truss portions of the T-Support and the guideway are already assembled (if not - refer to the Solidworks in the spring. there are several steps that must be done.

1. Check that the hex nuts on the truss portion and the guideway are properly tightened. Also, using a segment of the PVC check that the connection with the blue tape on the ABS parts will be snug, if not add or replace with additional 0.5 in blue tape (note that no tape is need for sections of the clamps surrounded by the acrylic beam or one of the four end caps which has a slightly larger connection diameter than the others).

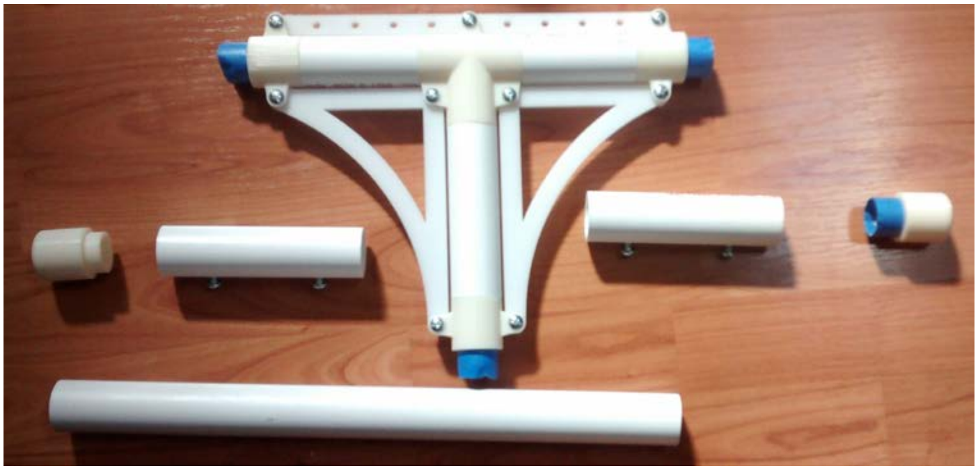


Figure 3-23. T-Support Assembly Step 1.

2. Slide the 23.5in beam into the horizontal portion of the clamp, and the 24.5in beam into the vertical section. Combine the cap parts with the PVC sections that have screws, then slide them across the wood beam to the truss structure. Note that the wood beams will likely get stuck on the edge of the clamps - just rotate the cap and gently wiggle the beams until they fall into place. Finally, slide the large PVC section along the vertical to fit on the lower clamp. The horizontal beam section should not have any exposed beam, while the vertical section will only show a small length.

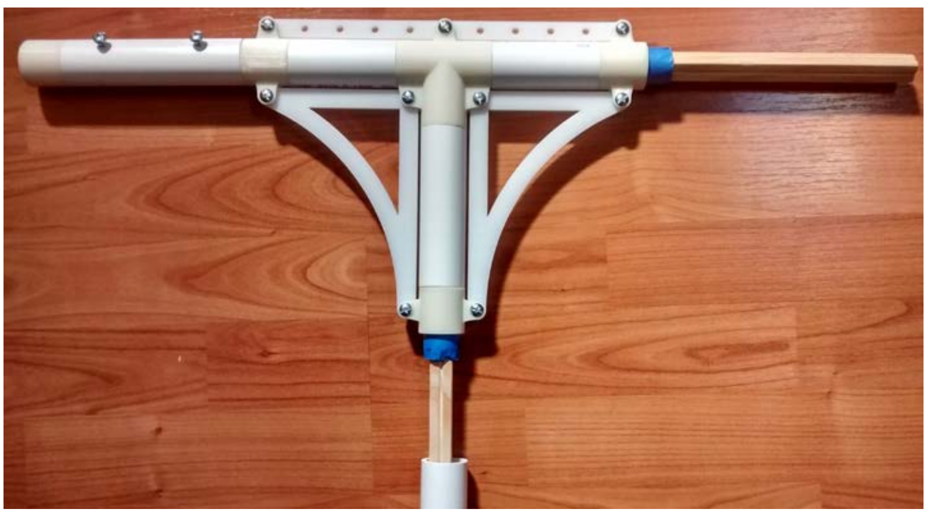


Figure 3-24. T-Support Assembly Step 2.

3. Repeat steps 1 and 2 for the other T-Support. Holding the vertical wood beam in place, put the two T-Supports into the adaptors on the ground panel. Remove all eight of the screws on the horizontal PVC sections, then slide both guideways so the gaps in their connectors surround the holes. Finally, put the screws back in.
4. To attach the 12th bogie, select one end of a guideway rail and remove both outer hex nuts from the screw there. Gently push open the rails and remove the end screw one end at a time.

Slide the bogie in (making sure that there are no components that will touch the screws heads along the rail) and replace the screw and two hex nuts.

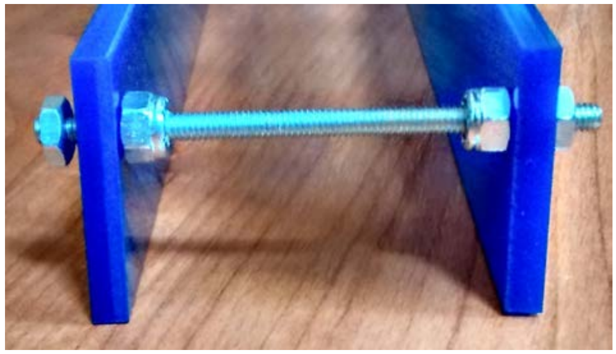


Figure 3-25. End Screw.

Money spent on your project

This Spring semester we spent a total of $590.57 and a grand total throughout the school year of only $623.40. This semester consisted of three architectural models which we had to fabricate and design ourselves and also a 1/8in thick steel column, developed last semester. The fabrication of the three architectural models costed the team less because we were able to use various free resources: SJSU's Machine shop, access to a free 3D printer and leftover materials around the Superway shop. The most expensive purchase for the Spring semester was the X-beam, which came out to $441.71. The X-beam was manufactured with the help of Vander-Bend's facilities and staff.

Table 3-1. Spring Semester Bill of Materials.

ITEM NO.	DESCRIPTION	QTY	UNIT COST	TOTAL COST
6	MDF Panel	1	$0.00	$0.00
7	E887Q Stainable Wood Filler, 3.25 Ounce	1	$6.15	$6.15
8	3D Printed Clamps (SJSU) - pairs	13	$0.00	$0.00
9	3D Printed Brackets	4	$0.00	$0.00
10	1in Schedule 40 PVC pipe (10ft)	2	$2.40	$4.80
11	1in PVC Tee Connector	3	$1.22	$3.66
12	1in PVC Cross Connector	1	$1.22	$1.22
13	1in PVC Sch. 40 Male Adapter	3	$0.79	$2.37
14	1in PVC Sch. 80 Pipe Flange	3	$6.93	$20.79
15	1.5in PVC Sch. 80 Pipe Flange	1	$7.90	$7.90
16	1/4"x18"x24" in Acrylic Sheet	2	$26.16	$52.32
17	#6 x 1" Sheet Metal Screws (bag of 10)	2	$1.29	$2.58
18	8/32 0.75" Machine Screw (box of 100)	1	$6.29	$6.29
19	8/32 1.25" Machine screw (box of 100)	1	$9.02	$9.02
20	8/32 Hex Nut (box of 100)	1	$3.89	$3.89
21	8/32 Lock Nut (box of 100)	1	$5.79	$5.79
22	.709"x23.75"x47.5" Fir panel	1	$22.02	$22.02
23	3mm thick X-Beam (Vanderbend)	1	$441.77	$441.77
			Spring Total	**$590.57**
			Grand Total	**$623.40**

Results and Discussion

One of the most significant principle we used while making our models was that a design is only as strong as its weakest part. As such, finding ways to redistribute the load to the stronger X-beam without bolting or welding became critical. There were several techniques were developed and expanded upon by our group this semester that proved successful in this task. The first was to have a sturdy beam perpendicular to the vertical X-beam, which prevented vertical motion. The second was to also redistribute a portion of the load into the side of the beam as the resulting compression would cancel out a portion of forces and prevent horizontal motion. The Third was to use multiple smaller clamps (connected by a standard I beam or CFT) - this lowers the load that each clamp will have to bear at any given times, and they will be simpler to replace or repair as needed.

The resulting Cat's Eye and T-Support models proved to be quite effective. With further refinements, both could become valid support structures for the Spartan Superway or other suspended rail systems where only one vertical column is to be used. Nonetheless, it must be stated that we also learned that the optimal design would likely be a combination of the two - using the horizontal X-beam and the secondary upper support beam from the T-Support and the Convex support beams of the Cat's Eye, though this may be less aesthetically pleasing.

In any case the models will serve as good starting point for the next group to expand on. The guideway design in particular provides a more developed alternative on how to use brackets and support beams. Finally, although we were not able to finish the stub in time to analyze it, it will be invaluable for future groups who will want to test its properties or design clamps and other attachments, as well as giving Vander-Bend an opportunity to practice for further beam production.

Conclusions and Suggestions for Future Work

Ultimately, we reached our design specifications. Our main goal was to create an architectural model, which we did successfully. The specifications were rather loose, as we had full design freedom, though the client did request the Palm Tree model with Dr. Furman's advice. The architectural model featured three different designs as alternative options, as well as one section of guideway to showcase that alteration. The creation of the architectural model went smoothly with no roadblocks, as we allotted plenty of time for ourselves to finish it with great detail.

For our column stub, however, we ran into multiple problems along the way. What should've been perhaps a couple weeks maximum of waiting for Andries to give us the column design turned into about 4 months, even with a Skype meeting call to discuss its progress. Because of this, our communication with Vander-Bend was delayed, and the estimated 3 weeks it would take to fabricate instead took closer to 4 weeks, including rush pricing. From this experience, we learned that we either need to not depend on outside resources and instead design everything ourselves, or we need to become more comfortable with repetitive reminders. If we had called Andries and Vander-Bend more often, we might've been made a higher priority for them. However, it was still completed for Maker Faire, and will be very helpful for future groups.

As a recap of our major accomplishments, we created an architectural model of the Spartan Superway support structure featuring three main models: The Palm Tree, wooden Cat's Eye, and PVC Cat's Eye models. This architectural design featured 3D-printed clamps and brackets, laser-cut acrylic pieces, wooden footings with PVC flanges and concrete, and the interlocking column as the base structure. We also manufactured the column stub so that future teams would better understand the geometry and integrity of its intricate design.

For next year's group of Spartan Superway engineers, they should focus on bringing the work of the Supports and Railings subteam to the Full-Scale size. Our work was preliminary brainstorming necessary to move onto the large scale, so we feel that we've adequately prepared the next group for actual prototyping. This would've been too much work for us to handle in addition, so it would be recommended that future groups take our work to apply it to the bigger model. We feel that the Supports and Railing team that we've become doesn't need to be a subteam next year as well.

Conclusions and Recommendations

Spartan Superway is a new concept, despite being in the works for about four years now. However, the individual aspects of the project are already proven in industry: automated transit already exists in the form of self-driving cars, point-to-point travel such as small trams and shuttles move people through airports, and renewable solar technologies have already been integrated into some passenger vehicles. The unique part about Superway is that it combines each of these aspects into a functional network of efficient transportation. Often, new ideas such as this are often looked at with fear and suspicion, but small-scale models allow the concept to be proven with much less investment of time and resources.

The scale models created by students for Spartan Superway each provide a basis for interpretation of these new concepts. The full-scale model offers aesthetic observation on a 1:1 scale, the half-scale model explores the limits of bogie propulsion and braking, and the 1/12th scale model shows a working network of podcars throughout a city. Additionally, the various models can provide recommendations to existing modes of transportation to improve efficiency. Solar panels can be implemented in cityscapes with minimal intrusion to aid in energy recovery, and public transit track design can be improved to minimize wasted travel time by stopping at unnecessary stations. Ultimately, these models allow us to engage the public and ensure that Spartan Superway is a concept that is real, possible, and inevitable.

References

1/12th Scale

A, Paola & Jeremy, G. (n.d.). Automating mode detection using neural networks and assisted GPS data collected using GPS-enabled mobile phones. Florida: University of South Florida.

"Automated Transit Networks (ATN): A Review of the State of the Industry and Prospects For the Future." Mineta Transportation Institute. Sept. 2014. Web. 10 Oct. 2016. <http://transweb.sjsu.edu/PDFs/research/1227-automated-transit-networks.pdf>

Automated Transit Systems - 2getthere. (n.d.). Retrieved December 03, 2016, from http://www.2getthere.eu/

Automated transport systems. (2012, February 2). Retrieved December 03, 2016, from http://adciv.org/Automated_transport_systems

Cheng, S. "Curved Photovoltaic Surface Optimization for BIPV: Anud Evolutionary Approach Based on Solar Radiation Simulation." CORE. UCL (University College London), 01 Jan. 1970. 24 May 2017.

Clifton Park, A., Delanson, D., Glenville, G., & Albany, S. (n.d.).Railcar location using mutter networks and locomotive transmitter during transit. General Electric Company.

Department of Transportation: Automated Transit Network (ATN). sanjoseca.gov. 1 May 2016. <http://www.sanjoseca.gov/index.aspx?NID=3706>

Furman, Burford, Ph.D, Lawrence Fabian, Sam Ellis, Peter Muller, and Ron Swenson.

http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001485305-da-01-en-IDUINO_SE022_ANALOG_HALL_SENSOR_MODUL.pdf

Insist Library <http://www.inist.org/library/>

Mark, A et al. (2016). Spartan Superway (Rep).

Model No. 2 Standard Bender. (2016). Retrieved November 5, 2016, from http://www.hossfeldbender.com/model-no-2-standard-bender/

MODEL 4 BENDER. (2016). Retrieved November 6, 2016, from https://www.diacro.com/product/model-4-bender/

Morgantown Personal Rapid Transit. (n.d.). Retrieved December 14, 2016, from https://en.wikipedia.org/wiki/Morgantown_Personal_Rapid_Transit

O. (n.d.). Greenhouse Gas Emissions from a Typical Passenger Vehicle. Retrieved December 1, 2016, from https://www.epa.gov/sites/production/files/2016-02/documents/420f14040a.pdf

Rodrigue, D. J. (n.d.). Urban Transport Challenges. Retrieved December 12, 2016, from https://people.hofstra.edu/GEOTRANS/eng/ch6en/conc6en/ch6c4en.html

Spartan Superway, A Solar Powered Automated Public Transportation System. Final Report (2015-16 Spartan Superway Team). (2016, Dec 13).

"Surface Meteorology and Solar Energy." NASA. NASA, n.d. Web. 24 May 2017.

Ultra PRT. (2013). Retrieved December 02, 2016, from http://www.ultraglobalprt.com/wheres-it-used/heathrow-t5

U.S. Energy Information Administration - EIA - Independent Statistics and Analysis. (n.d.). From https://www.eia.gov/electricity/monthly/epm_table_grapher.cfm?t=epmt_1_1

Half-Scale

Bigelow, P. (2016, May 15). America has the worst traffic in the world. Retrieved December 10, 2016, from http://www.autoblog.com/2016/03/15/america-worst-traffic-congestion-cities-smart/

Carnegie, Jon et al. Viability Of Personal Rapid Transit In New Jersey. 1st ed. New Jersey: N.p., 2007. Web. 7 Dec. 2016.

Carryer, J., Ohline R., and Kenny T. (2011). Introduction to Mechatronic Design. New Jersey: Pearson Education, inc.

City of San Jose Population. (2016). Retrieved December 08, 2016, from http://www.sanjoseca.gov/index.aspx?NID=2044

Critical Issues In Transportation. 1st ed. Washington: Transportation Research Board, 2013. Web. 7 Dec. 2016.

Cottrell, W., & Mikosza, O. (2008). New-generation personal rapid transit technologies: overview and comparison. Transportation Research Record: Journal of the Transportation Research Board, (2042), 101-108.

Coasr101. (2011). “Coasters-101: Wheel Design”. Coaster101. Retrieved from: http://www.coaster101.com/2011/10/24/coasters-101-wheel-design/

Ellis, S., Fabian, L, Muller, P., Swenson, R. & Dr. Furman, B.(2014). “Automated Transit Networks (ATN): A Review of the State of the Industry and Prospects for the Future”.

Furman, B., Ellis S., Fabian L., Muller, P., and Swenson R. (2014). Automated Transit Networks (ATN): A Review of the State of the Industry and Prospects for the Future. [pdf.] Retrieved from: http://transweb.sjsu.edu/PDFs/research/1227-automated-transit-networks.pdf

Furman, B. (advisor). (2016). Spartan Superway. A Solar Powered Automated Public Transportation System. [Word Document]. Retrieved from doc.google.com

Gibson, Tom (2008). U.S. Patent No. 5,842,666. Washington, DC: U.S. Patent and Trademark Office.

Gustafsson, J., & Lennartsson, S. (2009). VECTUS PRT Concept and Test Track Experience. ASCE Automated People Movers.

H-Bahn.(n.d). “Sky-Train Düsseldorf”. H-Bahn. Retrieved from: http://www.h-bahn.info/en/skytrain.php

J. Glennon. (2016).“calculating Critical Speed”. Crash Forensics. Retrieved from: http://www.crashforensics.com/papers.cfm?PaperID=42

Lowson, M. (2001). The ULTra PRT System. In Automated People Movers: Moving Through the Millennium (pp. 1-14). ASCE.

Meacham, Jody. (2016). “San Jose’s traffic congestions holding steady at moderately awful, in U.S and World Ranking”. Silicon Valley Business Journal. Retrieved from: http://www.bizjournals.com/sanjose/news/2016/03/23/san-joses-traffic-congestion-holding-steady-at.html

Mineta Transportation Institute. Retrieved from: http://transweb.sjsu.edu/PDFs/research/1227-automated-transit-networks.pdf

MISTER (Metropolitan Individual System of Transportation on Elevated Rail) LTD. (n.d.). Retrieved December 1, 2016, from https://misterltd.wordpress.com/about/

Mogge, John, The Technology of Personal Transit, "Figure 6. MASDAR Phase 1A Prototype Passenger PRT." Paper delivered at the World Future Energy Summit, Jan. 20, 2009. Available in WFES online media center.

NC State University Engineering. (n.d). “Gear Ratio”. NC State University. Retrieved from: https://www.engr.ncsu.edu/mes/media/pdf/gears

Norton, R. (2014). Machine Design: An Integrated Approach. 5th edition. New Jersey: Pearson Education, inc.

Oriental motors. (2016). “Data sheet for PK296A2A-SG36, Stepper Motor”. Retrieved from: http://catalog.orientalmotor.com/item/stepping-motors/pk-series-stepping-motors/pk296a2a-sg36?&seo=110

Parker, Josh. (n.d). “Normal Stress, Bending Stress, and Shear Stress”. StruCalc. Retrieved from http://www.strucalc.com/engineering-resources/normal-stress-bending-stress-shearstress/

Public Transportation Benefits. (n.d.). Retrieved October 1, 2016, from http://www.apta.com/mediacenter/ptbenefits/Pages/default.aspx

Raney, S. and Young S. (2004). Morgantown People Mover - Update Description. [pdf.] Retrieved from: http://www.cities21.org/morgantown_TRB_111504.pdf

Richards, G. (2016, October 3). Bay Area’s 10 most congested freeways (No. 3 is a surprise). Retrieved from http://www.mercurynews.com/2016/10/03/bay-areas-10-worst-commutes-unveiled-no-3-is-a-surprise/

Spartan Superway, A Solar Powered Automated Public Transportation System. Final Report (2015-16 Spartan Superway Team). (2016, Dec 13).

“Superway: Project Information”. (n.d.). Retrieved from http://www.engr.sjsu.edu/smssv/projectInfo.html

University of Washington.(n.d) “Cabintaxi PRT System”. University of Washington. Retrieved from: http://faculty.washington.edu/jbs/itrans/cabin.htmte

Wuppertal. (n.d.). Retrieved December 1, 2016, from http://www.urbanrail.net/eu/de/w/wuppertal.htm

Full-Scale

2getthere. (2016, September 11). Driverless ParkShuttle. Retrieved from <http://www.2getthere.eu/driverless-parkshuttle/>

2getthere, Masdar PRT. Retrieved from <http://www.2getthere.eu/projects/masdar-prt/>

ARUP, Heathrow Personal Rapid Transit system. Retrieved from <http://www.arup.com/projects/heathrow_personal_rapid_transit_prt>

Furman, B., Fabian, L., Ellis, S., Muller, P., Swenson, R. (2014, September). AUTOMATED TRANSIT NETWORKS (ATN): A REVIEW OF THE STATE OF THE INDUSTRY AND PROSPECTS FOR THE FUTURE. Retrieved from <https://www.inist.org/library/2014-09.Furman.Automated%20Transit%20Networks.MTI%201227.pdf>.

San Francisco Business Times (2004, March 2). S.F. has among nation's highest public transit use among commuters. Retrieved from <http://www.bizjournals.com/sanfrancisco/stories/2004/03/01/daily22.html>

San Jose City Hall. Department of Transportation (N.d.). Public Transportation. Retrieved from <https://www.sanjoseca.gov>.

Tebay, Andy (2013, February 13). PRT System to Open for Suncheon Garden Expo. Retrieved from <http://kojects.com/2013/02/13/prt-system-to-open-in-april-for-suncheon-bay-garden-expo/>

Transportation Research Board of National Academies (2012). Track Design Handbook for Light Rail Transit (2nd ed.). Retrieved from <http://onlinepubs.trb.org/onlinepubs/tcrp/tcrp_rpt_155.pdf>

Weisbrod, Glen, and Arlee Reno (2009). Economic Impact of Public Transportation Investment. American Public Transportation Association. Retrieved from <http://www.apta.com/resources/reportsandpublications/Documents/economic_impact_of_public_transportation_investment.pdf>.

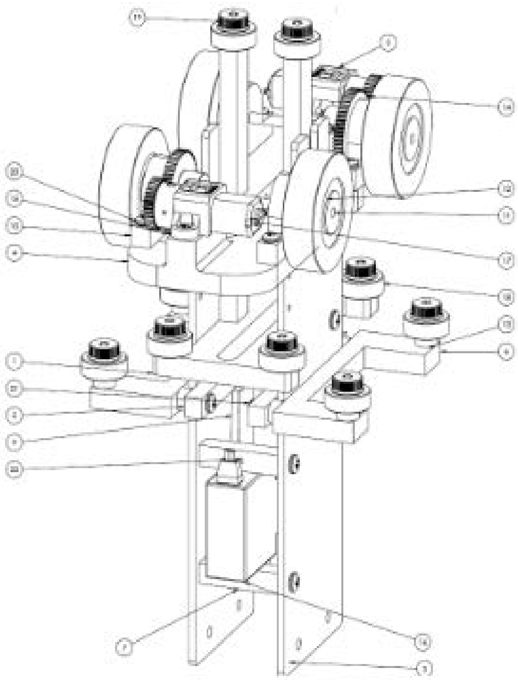
West Virginia University (N.d.) Transportation and Parking: PRT Facts. Retrieved from <http://transportation.wvu.edu/prt/prt-facts>

Appendix A: 1/12th Scale

Bogie Improvement

Appendix A1-1

Detailed Description of Bogie Parts:



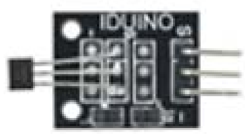
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	wheel holder plate	3D Printed PLA	1
2	Micro Metal Gear Motor With Bracket	Motor	2
3	side plate	3D Printed PLA	2
4	top motor plate	3D Printed PLA	1
5	switch side	3D Printed PLA	2
6	lower switch wheel bar	3D Printed PLA	2
7	Servo Mount	3D Printed PLA	2
8	servo horn	3D Printed PLA	1
9	Link	3D Printed PLA	2
10	Pillowball Bearing Bracket	3mm shaft	4
11	Rod	3mm x 7 mm	2
12	Wheel	35mm x 10mm	4
13	spacer4x7	4mm x 7mm	8
14	Gear, Plastic, SDP A1P2MYD04036A	Wheel gear	2
15	Gear, Plastic, SDP A1P2MYD04060A	Motor gear	2
16	HS-5087MH	Digital Metal Gear Micro Servo	1
17	Magnetic Encoder	12 CPR, 2.7-18V	2
18	7804K109	Bearing	10
19	90265A126	Shoulder Screw 5mm x 5 mm	10
20	91735A104	4-40 Screw, .375 in	20
21	91735A109	4-40 Screw, .625in	4
22	94070A077	2-56 Screw, ..375 in	6

Position Sensing

Appendix A2-1

IDUINO for Maker's life

Hall Effect Sensor (SE022)



1 Introduction

This module is analog hall sensor module, it can output an analog voltage at the signal pin of this module. This module is different from hall magnetic sensor(Module 31), which just output digital signal, like a magnetic switch.

Specification

- Operation voltage: 5V
- 3Pin
- Size:25*12mm
- Weight: 8g

2 Pinout

Pin	Description
S	Analog output pin, real-time output voltage signal
+(middle pin)	Power
-	Ground

3.Example

In this example, this module will read the value of magnetic and print on the Serial Monitor. These value can be reflect the intensity of environment magnetic.

Appendix A2-2

```
int pos;
bool peak = true;
const int peak_gap = 7;
int Peak_Num = 0;

int sensor(int sensorValue)
{
  if (sensorValue < 485)
    pos = 0;
  else if (sensorValue > 488)
    pos = 1;
    return pos;
}
void setup() {
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:

void loop() {
  int sensorVal = analogRead(A0);
  pos = sensor(sensorVal);

  if (pos == 0 && peak)
  {
    Peak_Num++;
    peak = false;
  }
  else if (pos == 0) {
  }
  else {
    peak = true;
  }
  Serial.println(Peak_Num);
}
```

Solar Improvement

Appendix A3-1

See Superway Archive (Google Drive) for the following files:

1. Product specification sheet for SoloPower SoloPanel SFX-i70
2. Product specification sheet and instruction manual for iMAX B6AC v2 AC/DC Dual Power Professional Balance Charger/Discharger
3. CAD files for racking system

Track Fabrication and Process Development

Appendix A4-1

Given Ri [in]		thickness [in]	Y [psi]	E [psi]	Rf arc length
10.21091683		0.25	40800	10000000	30.92898137
Rf [in] =		Given Ri [in]			Die angle
19.69000108		10 3/16			173.5495574

Insert thickness, modulus of elasticity, yield stress of material.

Input initial radius or final radius of material.

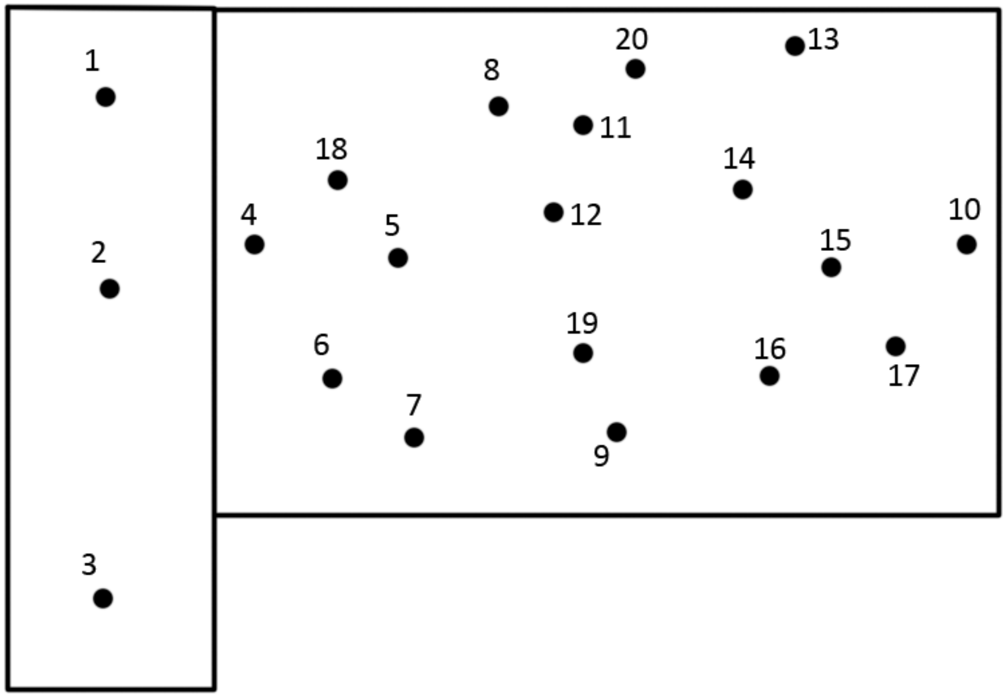
Run Solver to obtain Ri for a given Rf.

**All radii are inside radii.

Die Size Sheet, Old Material (T6?)				December 15th, 2016			
Elastic Modulus		40000 psi		Yield Stress		10000000 psi	
Die #	Matl. Thick	R-final	R-final	Arc Length	R-initial	R-initial	Die Angle
1	0.25	16.37	16 6/16	10 14/16	9.29	9 5/16	66
2	0.25	16.62	16 10/16	10 6/16	9.37	9 6/16	64
3	0.25	19.69	19 11/16	30 15/16	10.3	10 5/16	172
4	0.13	13.99	14	14	6.28	6 4/16	129
5	0.13	18.99	19	19	7.2	7 3/16	152
6	0.13	20.87	20 14/16	32 12/16	7.84	7 13/16	252
Cumulative sheet of numbers found with Solver implementation for die size.							
All measurements in inches or degrees.							

Appendix A4-2

Representative Hole Chart of Bender Surface



Die Placement Chart				Key	
Die Size	0.5" bolt	0.375" bolt		(d)	bolt holds down die
7.50"	4, 10, 15[c]	14(d)[t], 16(d), 17(d), 13[M]		[c]	center bolt for arm pin center
10.3125"	1, 2, 8[c], 10	4[M][t], 18(d), 19(d), 20(d)		[t]	extra length of bolt is threaded
				[M]	material hold block

Readme: This die placement chart correlates bolt location to the holes on the bener to allow for bender use. The Key explains the codes used to refer to special use or needs of any individual bolt. This chart is inteded for use in conjunction with the Representative Hole Chart of Bender Surface.

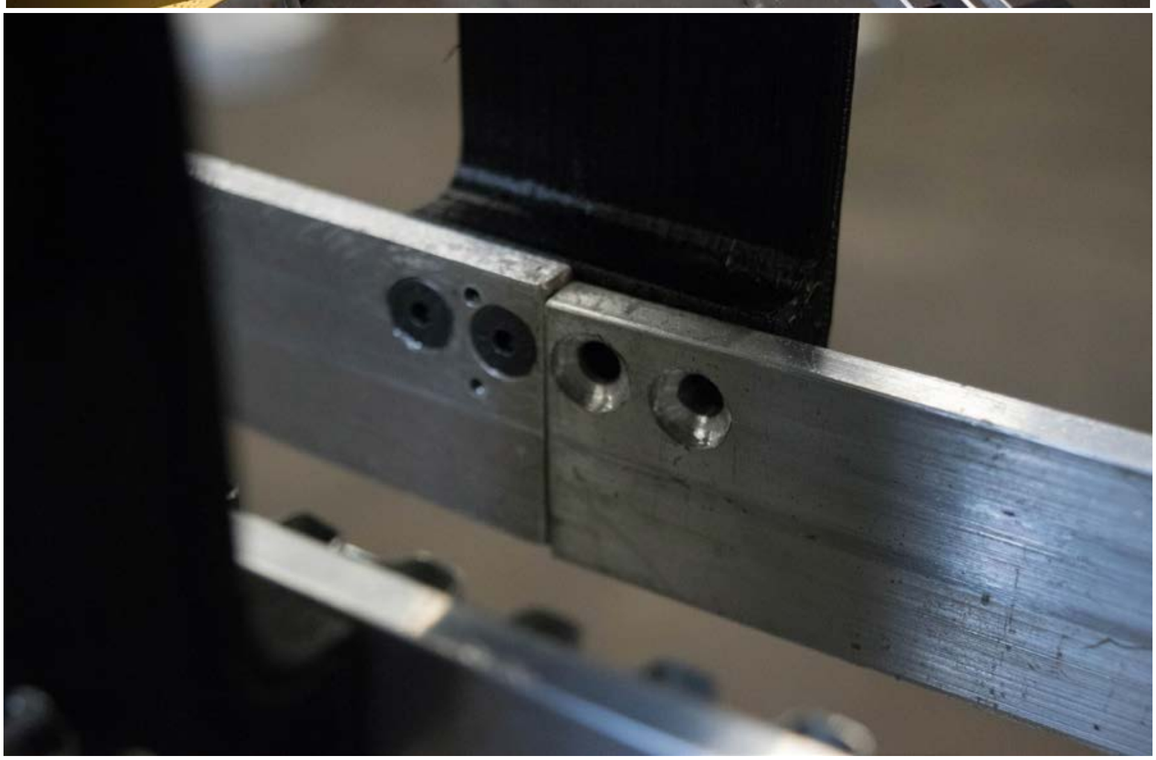
Appendix A4-3

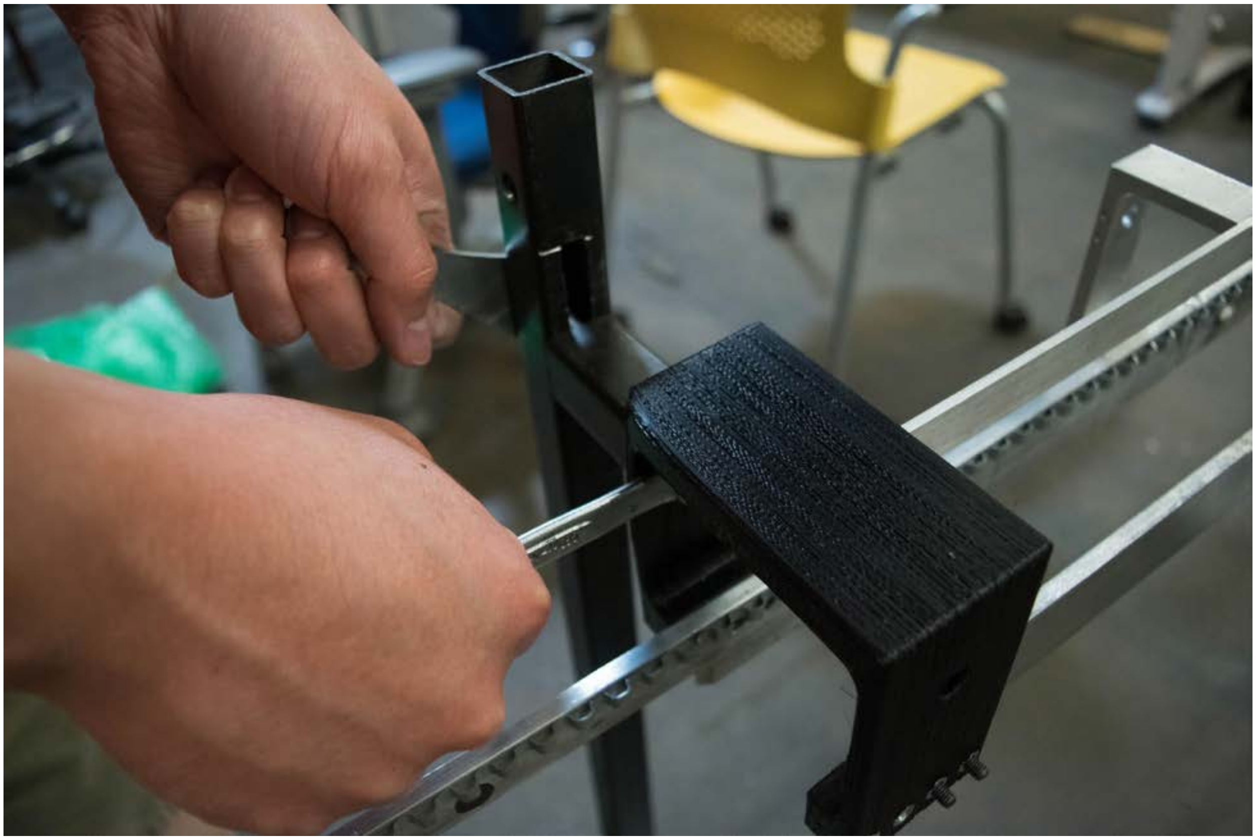
Material	Dimension	Quantity	Cost	Quotes	Lead Time	Sponsorship	Need	Time Table
Plywood	0.750" x 14.000" x 48.000"	1	Free (Shop)	N/A	N/A	N/A	Die, Tabletop, Measurement Mold	N/A
	0.750" x 16.500" x 48.000"	1	Free (Shop)	N/A	N/A	N/A	Die, Tabletop, Measurement Mold	N/A
	0.750" x 30.000" x 96.000"	2	Free (Craigslist)	N/A	N/A	N/A	Die, Tabletop, Measurement Mold	N/A
	0.750" x 48.000" x 96.000"	1	Free(Donated)	N/A	N/A	N/A	Die, Tabletop, Measurement Mold	N/A
Aluminum plate	20.000" x 39.000"	2	Free (Shop)	N/A	N/A	N/A	Tabletop	N/A
Aluminum bar stock	2.000" x 3.4375"	1	Free (Shop)	N/A	N/A	N/A	Center Pin Hold, Roller	N/A
	0.875" x 5.125"	1	Free (Shop)	N/A	N/A	N/A	Material Hold Center Pin	N/A
	1.000" x 9.000"	1	Free (Shop)	N/A	N/A	N/A	Roller Center Pin	N/A
Steel bar stock	0.500" x 6.000"	1	Free (Shop)	N/A	N/A	N/A	Roller Arm Center Pin	N/A
Aluminum rect. stock	0.250" x 1.500" x 96.000"	4	Free (Shop)	N/A	N/A	N/A	Test Bend Material	N/A
	0.125" x 0.750" x 96.000"	4	Free (Shop)	N/A	N/A	N/A	Test Bend Material	N/A
	3.000" x 3.000" x 1.500"	1	Free (Shop)	N/A	N/A	N/A	Material Hold Block	N/A
	2.375" x 2.375" x 2.750"	1	Free (Shop)	N/A	N/A	N/A	Material Hold Block	N/A
	0.750" x 1.500" x 38.000"	1	Free (Shop)	N/A	N/A	N/A	Bender Arm	N/A
"Two by Four"	1.500" x 3.500" x 96.000"	2	Free (Craigslist)	N/A	N/A	N/A	Table Cross-Support	N/A
Steel Bolts	0.3125" x 2.500"	12	Free (Shop)	N/A	N/A	N/A	Table/Tabletop Support	N/A
	0.375" x 1.500"	1	Free (Shop)	N/A	N/A	N/A	Material Hold Pin Support	N/A
	0.375" x 3.000"	3	Free (Shop)	N/A	N/A	N/A	Die Support	N/A
	0.500" x 1.500"	3	Free (Shop)	N/A	N/A	N/A	Tabletop Support	N/A
Steel Nuts	0.3125"	12	Free (Shop)	N/A	N/A	N/A	Anchor Support	N/A
	0.375"	4	Free (Shop)	N/A	N/A	N/A	Anchor Support	N/A
	0.500"	2	Free (Shop)	N/A	N/A	N/A	Anchor Support	N/A
Steel Washers	0.3125"	24	Free (Shop)	N/A	N/A	N/A	Anchor Support	N/A
	0.375"	8	Free (Shop)	N/A	N/A	N/A	Anchor Support	N/A
	0.500"	5	Free (Shop)	N/A	N/A	N/A	Anchor Support	N/A
Propane Torch	N/A	2	Free(Donated)	N/A	N/A	N/A	Material Manipulation Aid	N/A

Track Improvement

Appendix A5-1

Miscellaneous Photos

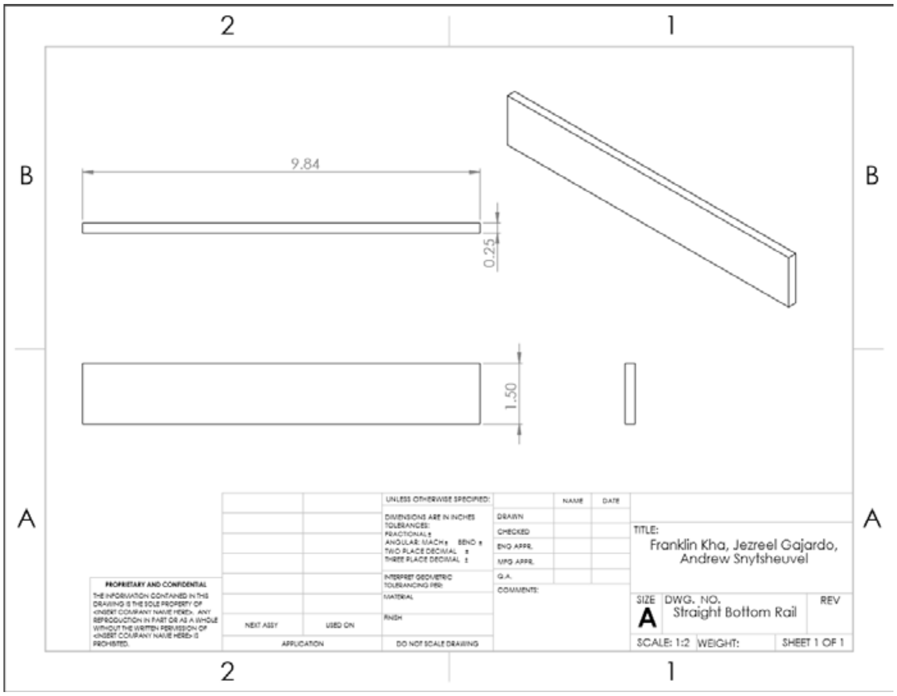


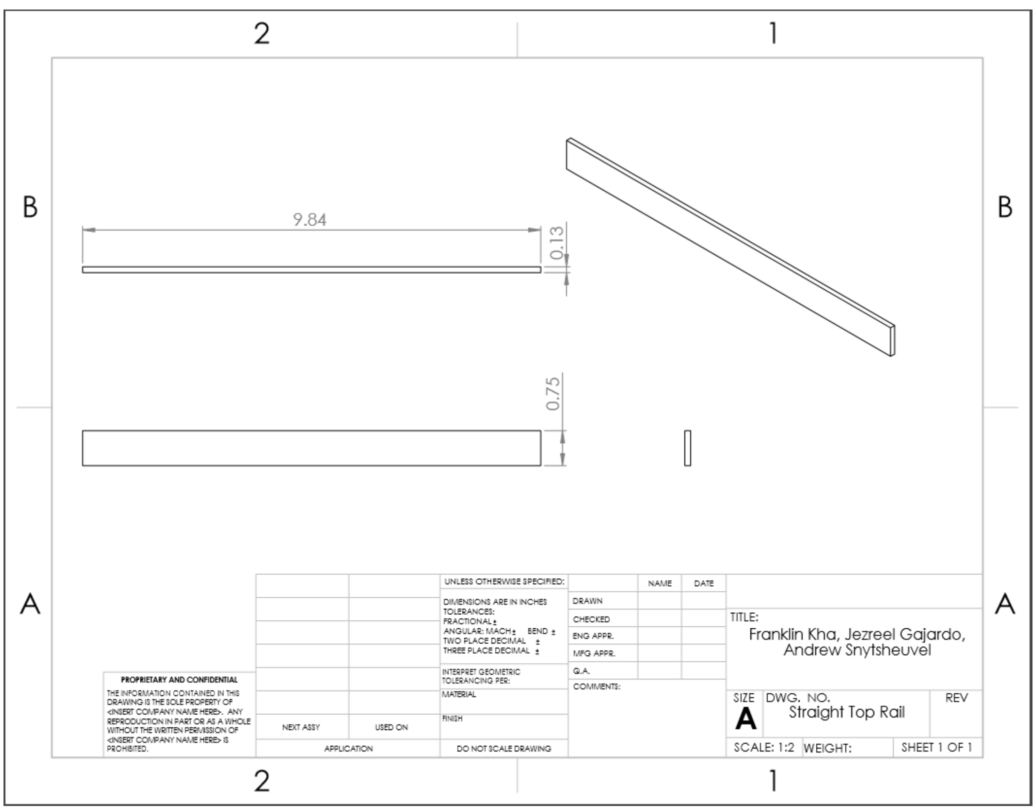


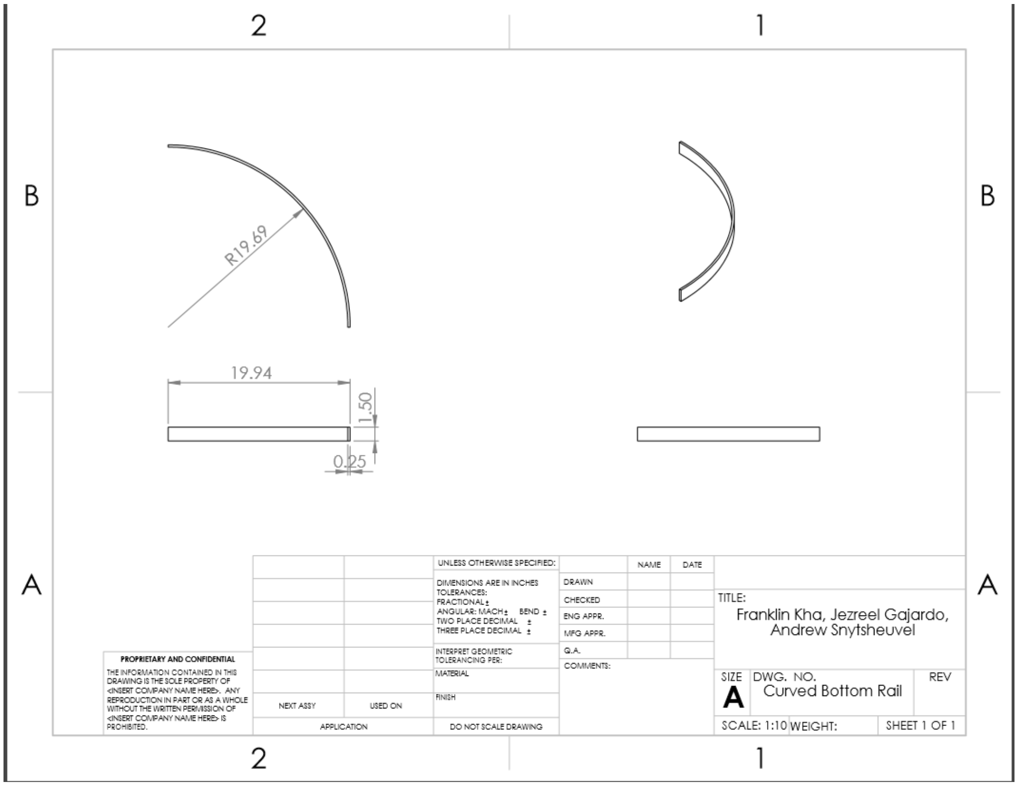

North

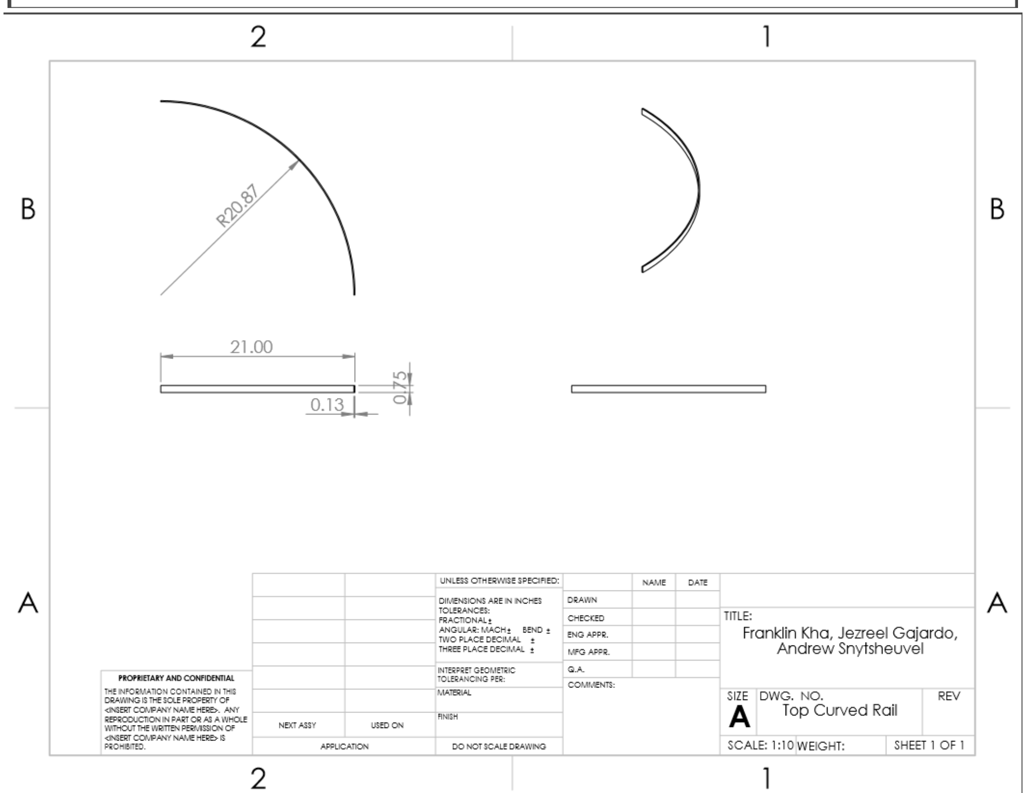
Appendix A5-2

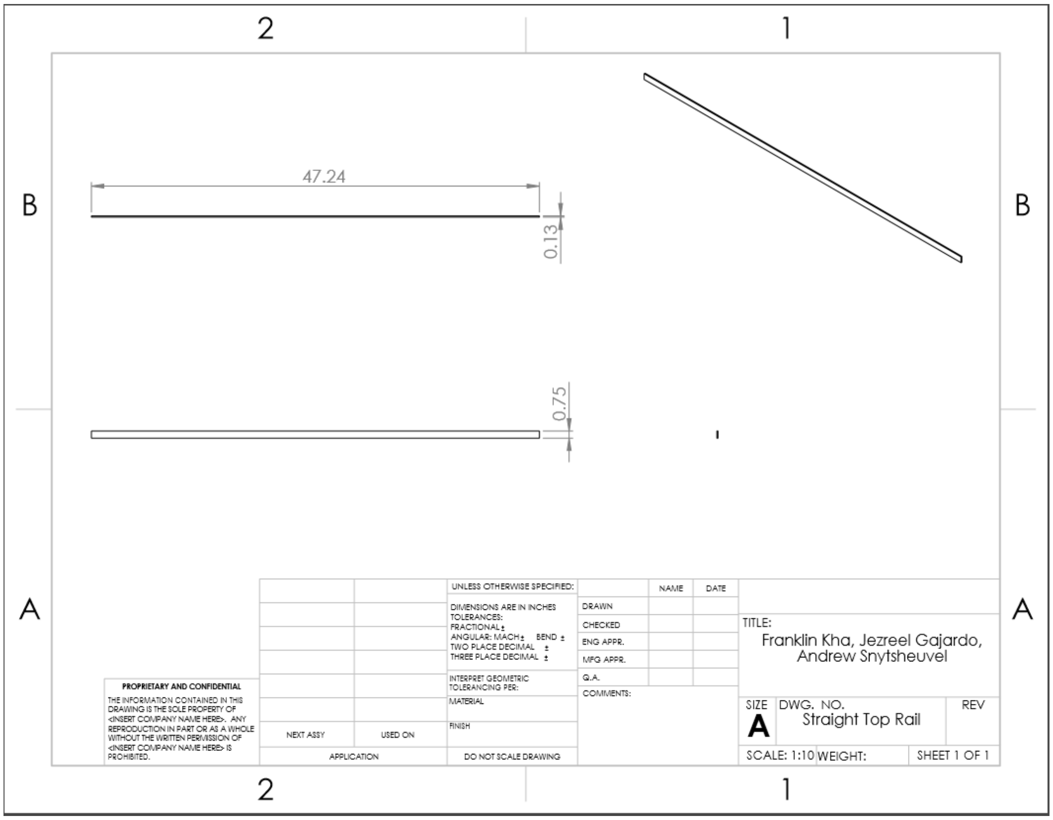
Part Drawings

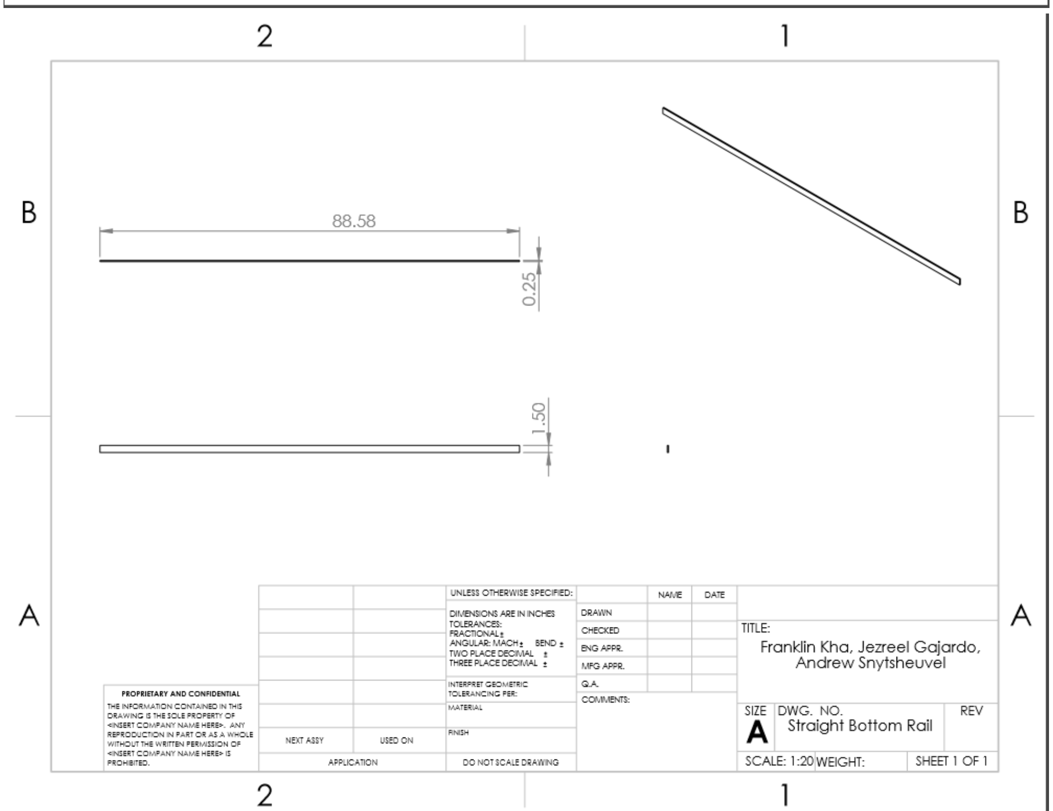


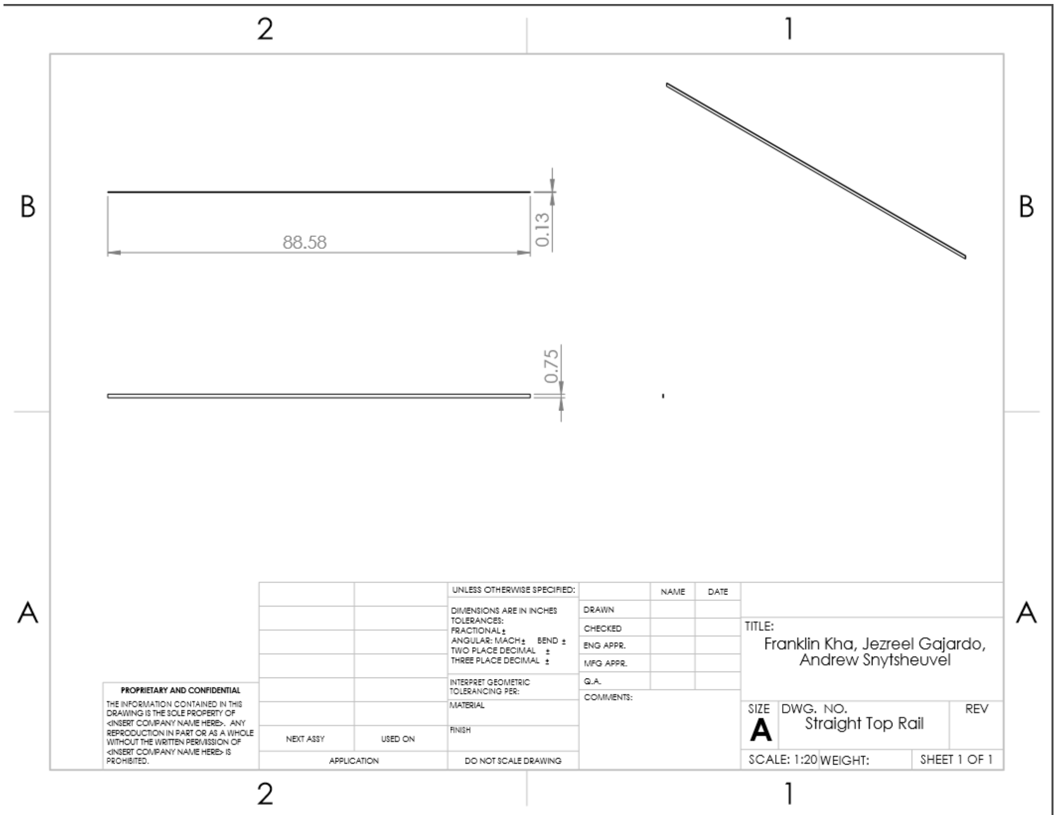


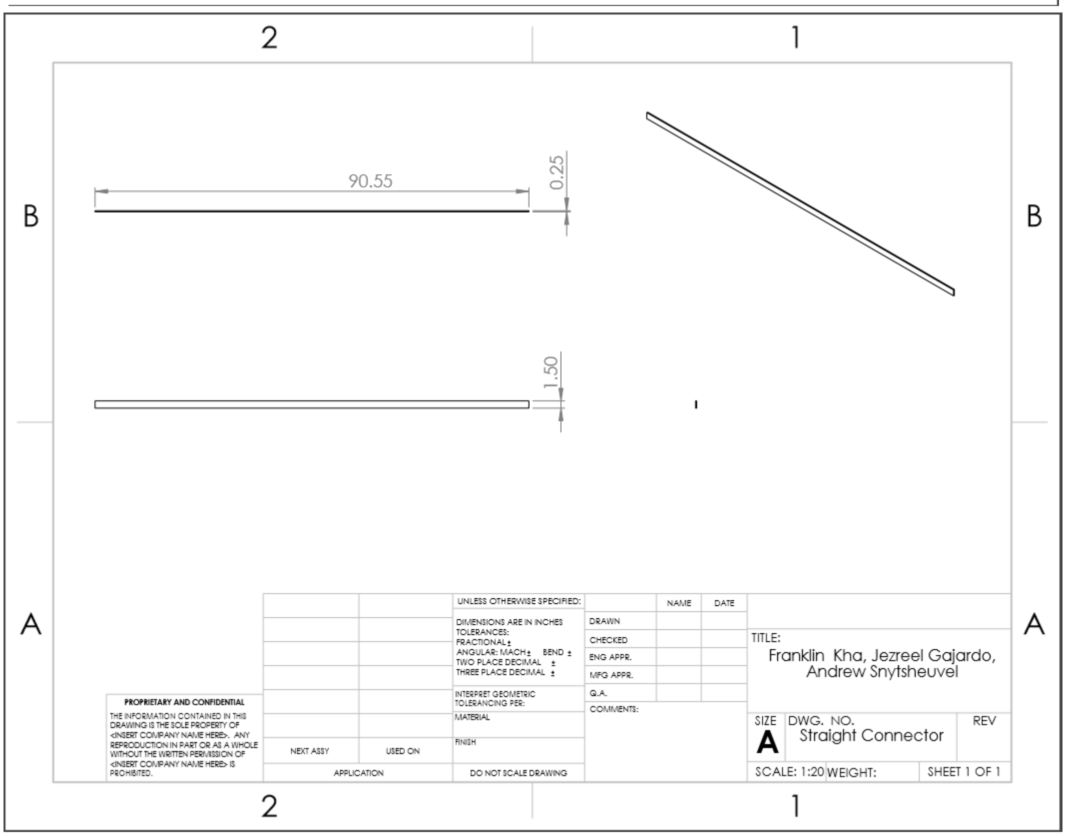

R19.69
19.94
1.50
0.25
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE
A
DWG. NO.
Curved Bottom Rail
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

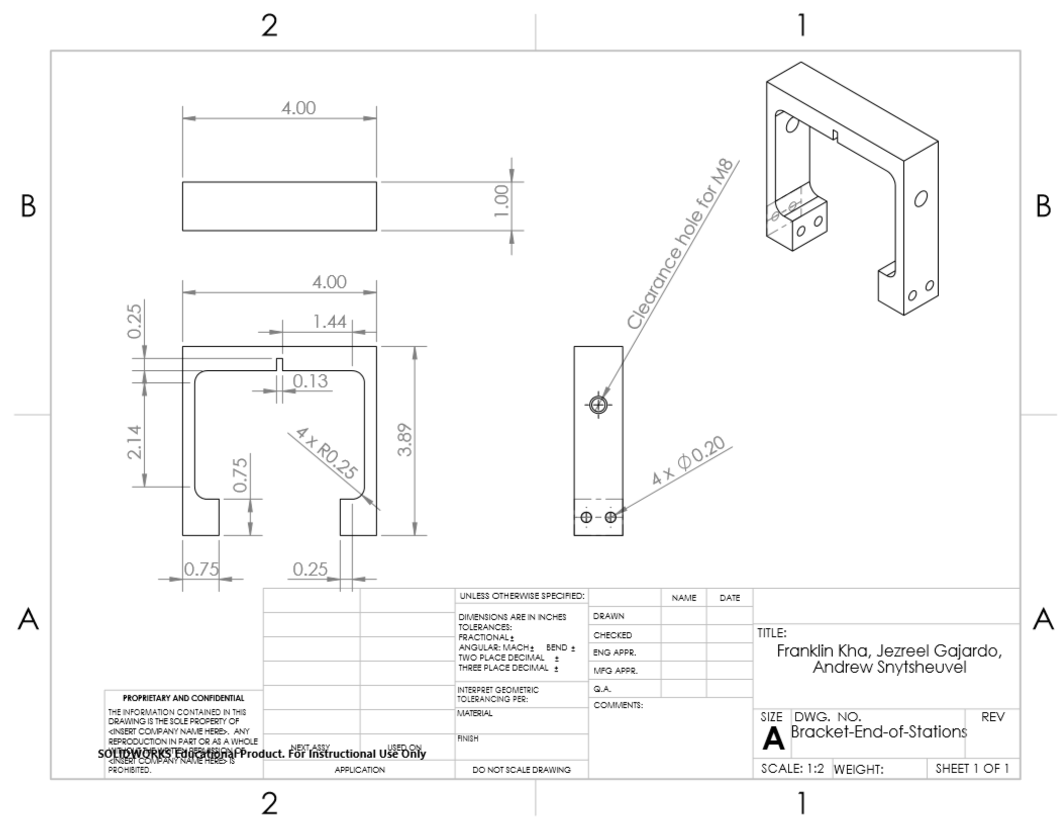

R20.87
21.00
0.75
0.13
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE
A
DWG. NO.
Top Curved Rail
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

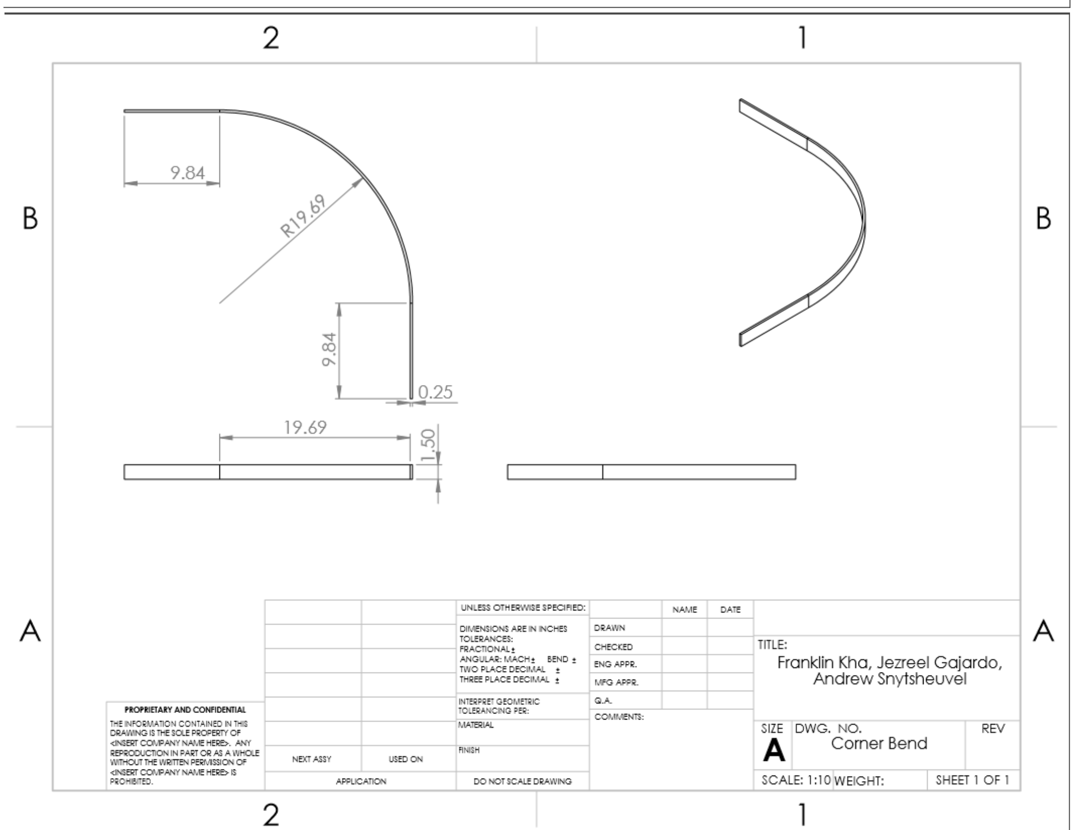

47.24
0.13
0.75
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE
A
DWG. NO.
Straight Top Rail
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NEXT ASSY
USED ON
APPLICATION

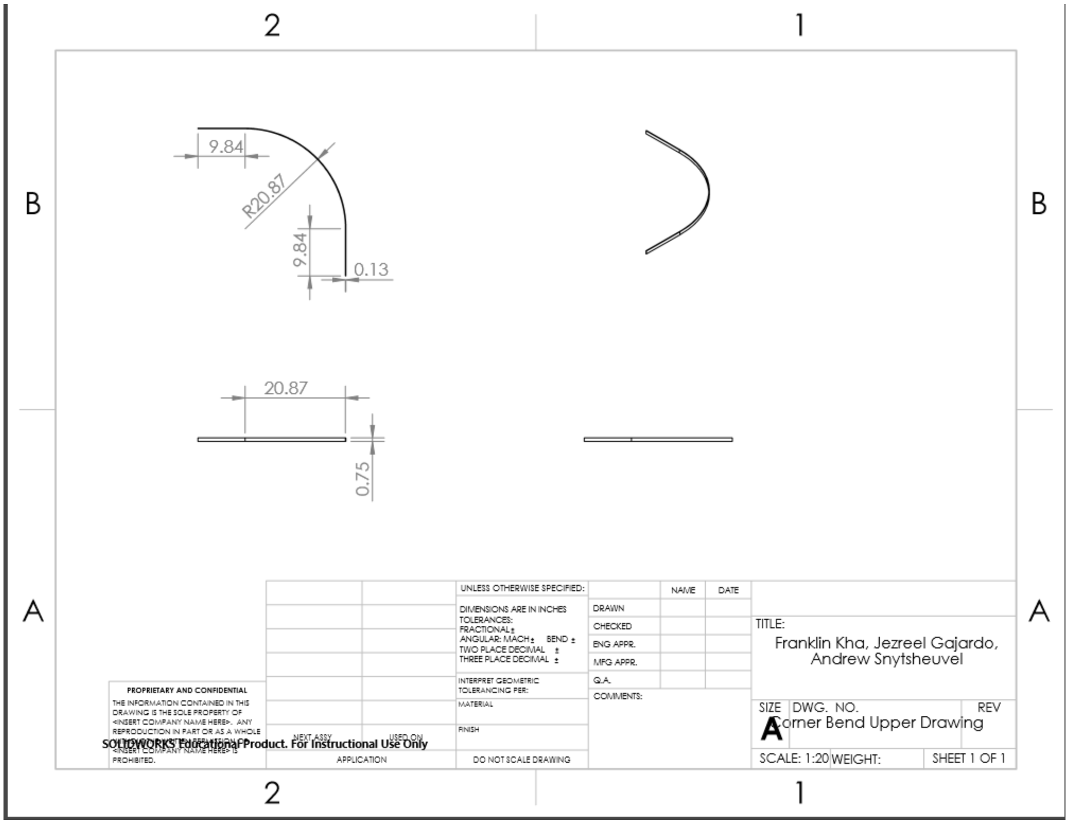

88.58
0.25
1.50
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE
A
DWG. NO.
Straight Bottom Rail
REV
SCALE: 1:20
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NEXT ASSY
USED ON
APPLICATION

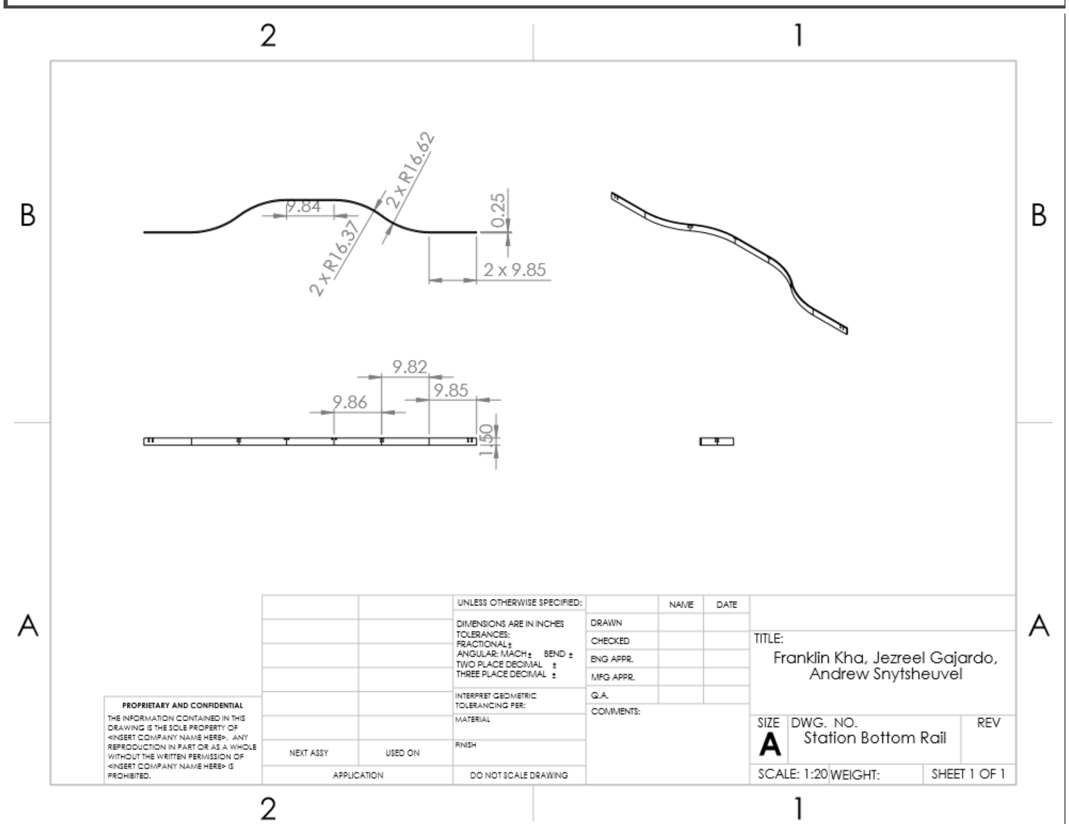

88.58
0.13
0.75
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE A
DWG. NO. Straight Top Rail
REV
SCALE: 1:20
WEIGHT:
SHEET 1 OF 1

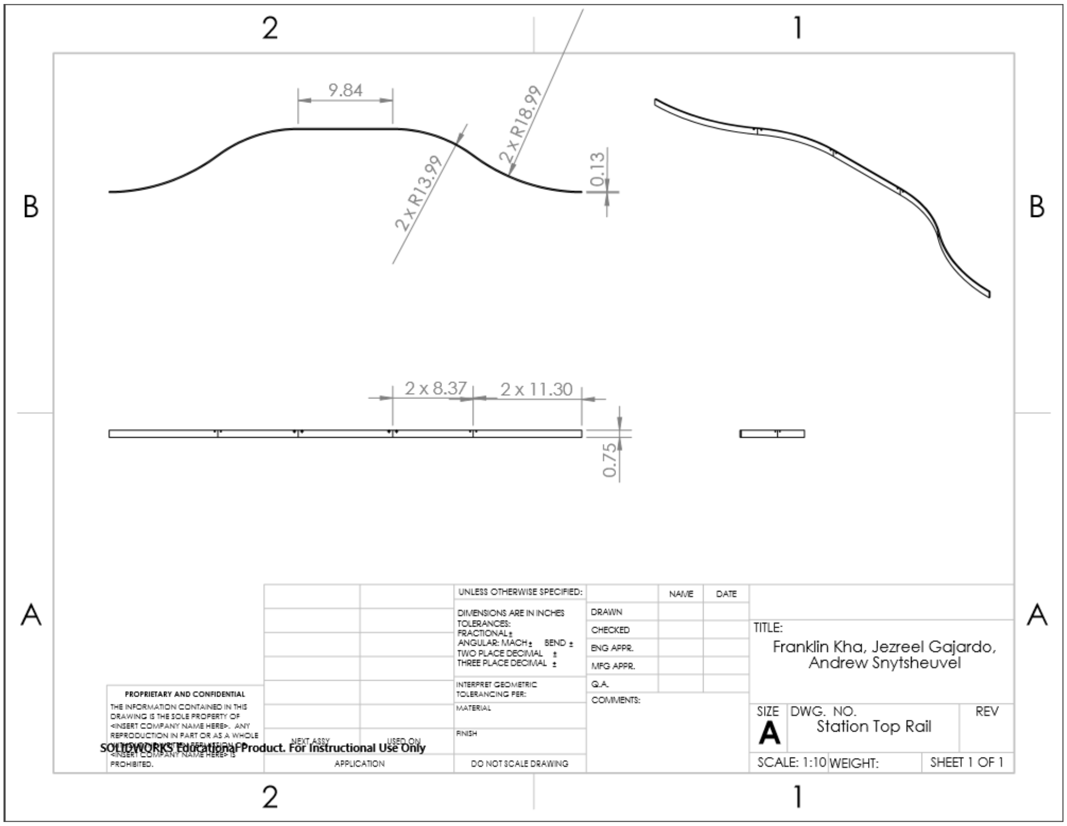

90.55
0.25
1.50
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE A
DWG. NO. Straight Connector
REV
SCALE: 1:20
WEIGHT:
SHEET 1 OF 1

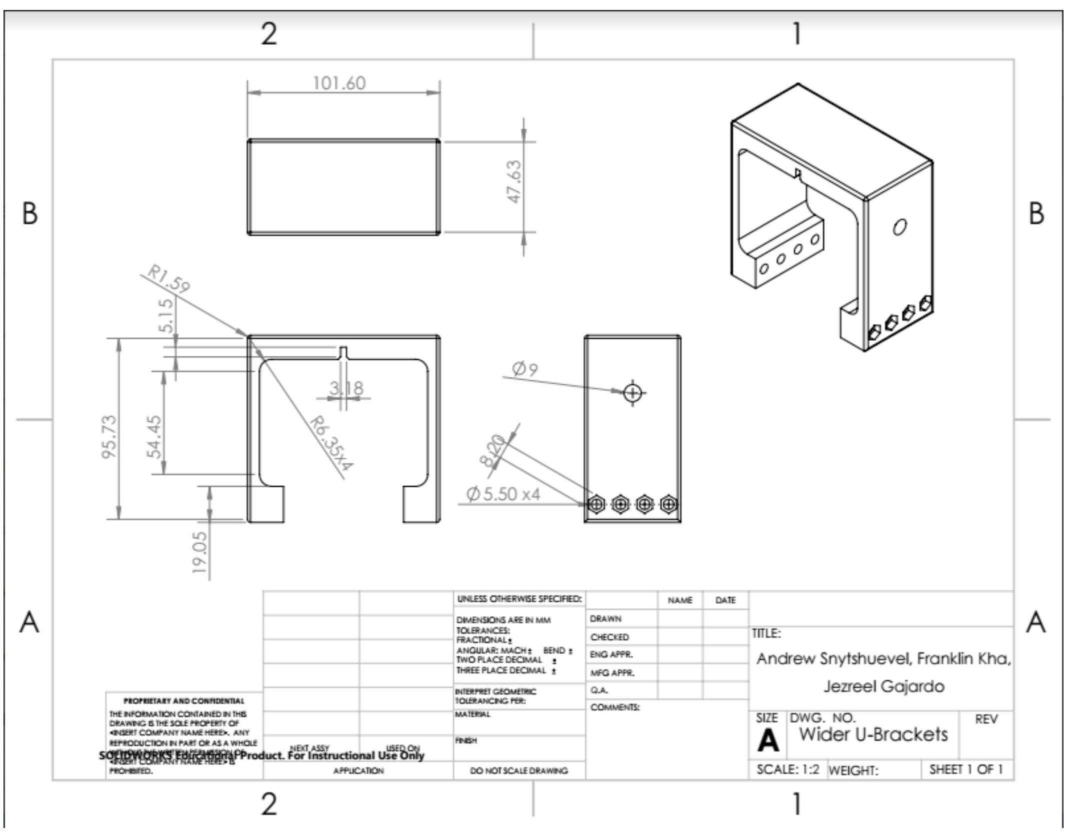

4.00
1.00
4.00
1.44
0.25
0.13
2.14
0.75
4 x R0.25
3.89
0.75
0.25
Clearance hole for M8
4 x Ø0.20
SOLIDWORKS Educational Product. For Instructional Use Only
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE A
DWG. NO.
Bracket-End-of-Stations
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
PROPRIETARY AND CONFIDENTIAL


9.84
R19.69
9.84
0.25
19.69
1.50
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE A
DWG. NO.
Corner Bend
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
PROPRIETARY AND CONFIDENTIAL


9.84
R20.87
9.84
0.13
20.87
0.75
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE
A
DWG. NO.
Corner Bend Upper Drawing
REV
SCALE: 1:20
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NEXT ASSY
USED ON
APPLICATION
SOLIDWORKS Educational Product. For Instructional Use Only


9.84
2 x R16.62
2 x R16.37
0.25
2 x 9.85
9.82
9.86
9.85
1.50
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE
A
DWG. NO.
Station Bottom Rail
REV
SCALE: 1:20
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
NEXT ASSY
USED ON
APPLICATION

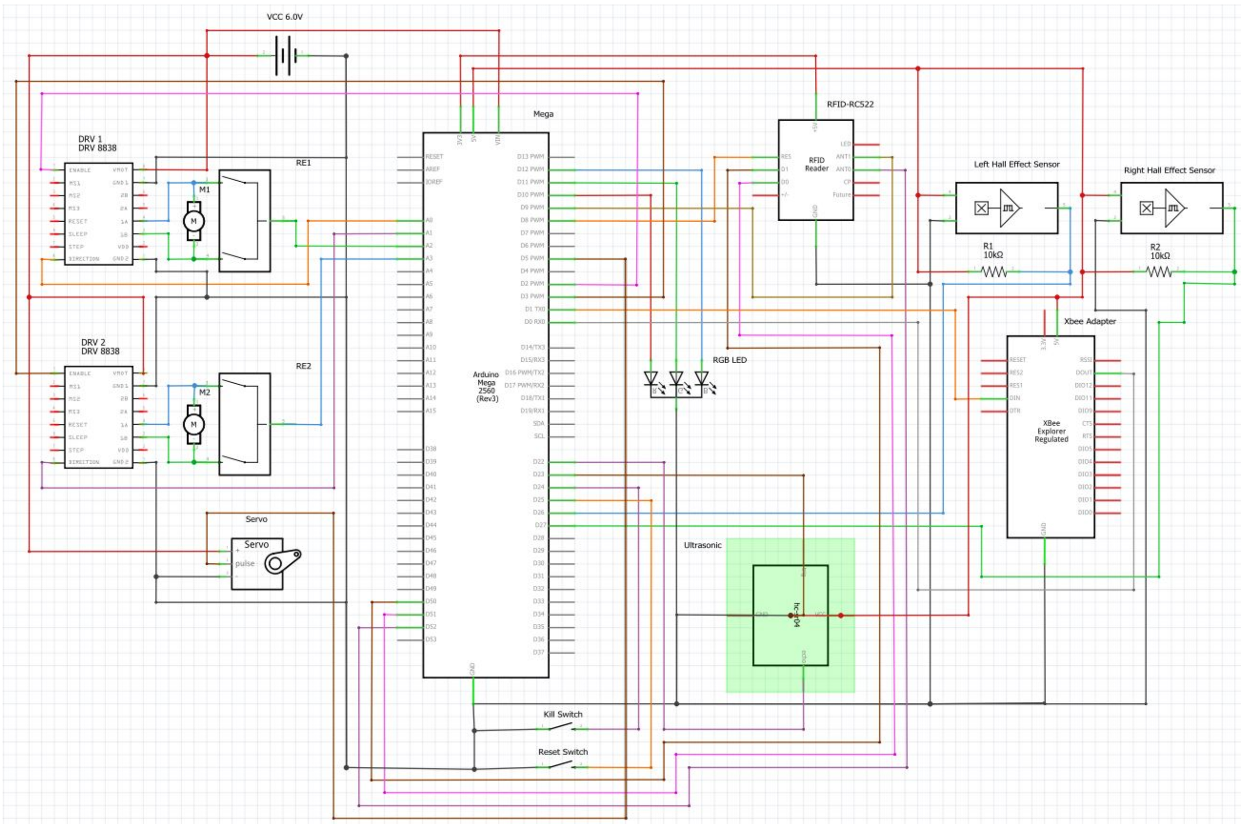

9.84
2 x R18.99
2 x R13.99
0.13
2 x 8.37
2 x 11.30
0.75
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Franklin Kha, Jezreel Gajardo, Andrew Snytsheuvel
SIZE A
DWG. NO.
Station Top Rail
REV
SCALE: 1:10
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
NEXT ASSY
USED ON
APPLICATION
SOLIDWORKS Educational Product. For Instructional Use Only


101.60
47.63
R1.59
5.15
3.18
R6.35x4
95.73
54.45
19.05
Ø9
8.20
Ø5.50 x4
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MM
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Andrew Snytshuevel, Franklin Kha, Jezreel Gajardo
SIZE A
DWG. NO.
Wider U-Brackets
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
NEXT ASSY
USED ON
APPLICATION
SOLIDWORKS Educational Product. For Instructional Use Only

Vehicle Controls

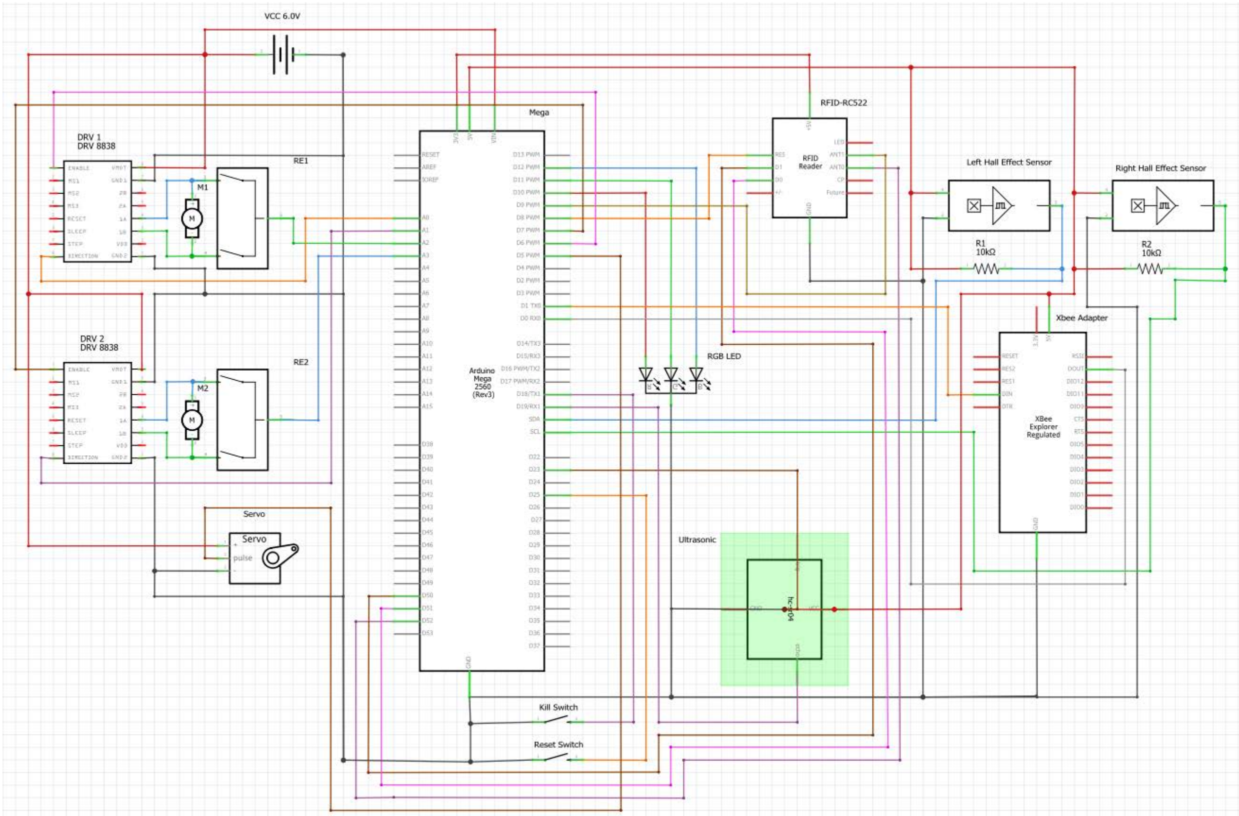
Appendix A6-1

Wiring Diagram for Old Expansion Board



Appendix A6-2

Wiring Diagram for Updated Expansion Board



Appendix A6-3

Code for Vehicle Controls System 2016-2017

```
/////////////////////////////////////// LIBRARIES ///////////////////////////////////////////////////////////

// Libraries for the RFID, pathing algorithm, and for the motors and servo
#include <RFID.h>
#include <SPI.h>
#include <TimerThree.h>
#include <StackArray.h>
#include <Servo.h>
#include <SoftwareSerial.h>

//////////////////////////////////// PIN DEFINITIONS ////////////////////////////////////////////////////////
/* Note: For XBee Pro, Channel CH = C, Pan ID = 2017
 *
// */
//// Pins 0 and 1 on the Mega are used for XBee Comms
///* Note: For RFID reader - SDA (PIN 9), SCK (PIN 52), MOSI (PIN 51),
// *  MISO (PIN 50), RST (PIN 8).
// */
// Pins 0 and 1 on the Mega are used for XBee Comms
/* Note: For RFID reader - SDA (PIN 9), SCK (PIN 52), MOSI (PIN 51),
 *  MISO (PIN 50), RST (PIN 8).
 */
#define MOTOR1                6   // Interrupt 0
#define MOTOR2                7   // Interrupt 1
#define SERVO                 5   // Servo pin
#define RED                   10  // STOP and Error status
#define GREEN                 11  // GO status
#define BLUE                  12  // Status of the podcar
#define KILL_SWITCH           18  // Interrupt 5
#define ECHOFRONT             19  // Echo pin
#define TRIGFRONT             23  // Trig pin
```

```
#define RESET                 25  // Resets the system after error status has been acknowledged
#define HALL_SENSOR1          20  // Switch servo motor using hall effect sensor mounted on left of podcar
#define HALL_SENSOR2          21  // Switch servo motor using hall effect sensor mounted on right of podcar


// For ultrasonic interrupts
#define TIMER_US 30
#define TICK_COUNTS 4000
#define ECHOINTFRONT 4


////////////////////////////////// SYSTEM VARIABLES ////////////////////////////////////////////////////////
// For button state
#define NOT_PRESSED           LOW
#define IS_PRESSED            HIGH


// For ultrasonic and LED
#define OFF                   LOW
#define ON                    HIGH


/* Define the DIO used for the SDA (SS) and RST (reset) pins. */
#define SDA_DIO               9
#define RESET_DIO             8


/* Create an instance of the RFID library */
RFID RC522(SDA_DIO, RESET_DIO);


// Latching button using software
boolean kill_latch_state = true;
boolean reset_latch_state = true;


// Stores the state of each button
int reset_state;
int killStatePressedTwice;


// Additional values for the two switches
bool checkForSecondKillPress;
bool needToFlush;
bool killStateNotRefreshed;


// Used for the ultrasonic sensor
int maximumRange = 200;       // Maximum range needed
int minimumRange = 0;         // Minimum range needed
volatile long distanceFront;                    // Duration used to calculate distance in ultrasonic 1
volatile long distanceSide;             // Duration used to calculate distance in ultrasonic 2
int distance;                 // Stores the distance
int countForUltrasonic;       // Count the time to trigger ultrasonic
bool stillInRangeUltrasonic;  // Checks if the sensor is still in range after 2 seconds


// Variables used to store the readings from the hall effect sensors
volatile int hall_1_state = 1;
volatile int hall_2_state = 1;


/* Note: RGB LED- Green (GO status), RED (STOP), BLUE (Hall effect sensor reading)
 *  Blinking BLUE (Station arrival), Blinkin RED (Error)
 */


// Used for LED purposes
int redState = OFF;           // Initially, all LED states should be off
int greenState = OFF;
int blueState = OFF;
unsigned long currentBlink = 8;
unsigned long previousBlink = 4;
const long interval = 10;


// Stores the station number and checkpoint number
int  STATIONID = 0;
int  CHECKPOINT = 0;
long SSN = 0;
int howLongToStayAtStation;
bool stillAtStation;
bool serverGoFromStation;


// Station and Checkpoint Numbers
int STATIONONE;
int STATIONTWO;
int STATIONTHREE;
int STATIONFOUR;

int CHECKPOINTONE;
int CHECKPOINTTWO;
int CHECKPOINTTHREE;
int CHECKPOINTFOUR;
```

```
// To tell if a packet is sent or not
bool didSendStationPacket = true;


// The isCard result value
bool RFIDisCard;


// Check if the magnets continue to be true
bool isMagnetNotRead;


// Global set values
long SPEED;
int maximumSpeed;
int STATEOFCAR;
int STATEOFCARFROMSERVER;
int SERVOINT;
int incrementByFifty;

// Positioning Variables
const int peakgap = 7;
int LastRFID;
int Peak_Num;
int Position;
int tickCounterStation = 0;
bool isFirstPosTickThrough;
bool isSecondPosTickThrough;


// Create Array to be stored as message
int message[1];


// Allows the XBee to communicate
SoftwareSerial XBee(0,1); //RX, TX


// The Packet int array that will be used for communication
int commands[8];


// Variables used for pathing
#define SEND_PATH1           6
#define SEND_PATH2           7         // buttons and are temporary


// The array for the 12 paths
volatile StackArray <int> scheduleArray;


// 8 different paths a podcar can take from anywhere to a station
int exactlyAfterStationOneFour[] = {1,1,1,0};
int exactlyBeforeStationTwoThree[] = {0,0};

int afterStationOneFourToTwoThree[] = {1,1,0,0};
int passMiddleIntoStationOneFour[] = {1,0};

int fromMiddleIntoStationOneFour[] = {1,1,0};
int fullCircleIntoStationTwoThree[] = {1,1,1,0,0};

int halfCircleIntoStationTwoThree[] = {1,0,0};
int switchIntoClosestStation[] = {0};


// Before or After magnet value
volatile bool isBeforeMagnet;


// 12 different paths a podcar can take from station to station
int path1to2[]={0, 0}; //0 for a
int path1to3[]={1, 1, 0, 0}; //1 for a
int path1to4[]={1, 0}; //2 for a

int path2to1[]={0}; //3 for a
int path2to3[]={1, 1, 1, 0, 0}; //4 for a
int path2to4[]={1, 1, 0}; //5 for a

int path3to1[]={1, 1, 0}; //6 for a
int path3to2[]={1, 1, 1, 0, 0}; //7 for a
int path3to4[]={0}; //8 for a

int path4to1[]={1, 0}; //9 for a
int path4to2[]={1, 1, 0, 0}; //10 for a
int path4to3[]={0, 0}; //11 for a

bool firstChoiceForInitialPath;
bool isNotAtLastStationInDestination;
int lastStationInDestination;


// Initialize the servo
volatile Servo leverArm;
```

```
// Podcar number
int podNumber = 2;
bool testStack;


//testing new ISR ultrasonic
volatile long echoStartFront = 0;
volatile long echoEndFront = 0;
volatile long echoDurationFront = 0;
volatile int triggerTimeEventFront = 0;
volatile int ultrasonicCounterFront = 0;

volatile long echoStartSide = 0;
volatile long echoEndSide = 0;
volatile long echoDurationSide = 0;
volatile int triggerTimeEventSide = 0;
volatile int ultrasonicCounterSide = 0;


///////////////////////////////////////// SETUP ///////////////////////////////////////////////////////////////
void setup(){
  // Outputs
  pinMode(MOTOR1, OUTPUT);
  pinMode(MOTOR2, OUTPUT);
  pinMode(TRIGFRONT, OUTPUT);
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);

  // Inputs
  pinMode(ECHOFRONT, INPUT);
  pinMode(KILL_SWITCH, INPUT_PULLUP);
  pinMode(RESET, INPUT_PULLUP);
  pinMode(HALL_SENSOR1, INPUT);
  pinMode(HALL_SENSOR2, INPUT);

  // Temporary
  pinMode(SEND_PATH1, INPUT_PULLUP);
  pinMode(SEND_PATH2, INPUT_PULLUP);

  // Set TimerThree to signal at 20Hz frequency (20 times/sec)
  Timer3.initialize(TIMER_US);
  Timer3.attachInterrupt(ISRpulseFront);

  // Change this to 57600 for XBee Comm
  Serial.begin(57600);
  XBee.begin(57600);

  /* Initialize the RFID reader */
  SPI.begin();
  RC522.init();

  killStatePressedTwice = 0;
  checkForSecondKillPress = false;

  // initialize SPEED and set initial STATE
  // Divide by 4 to translate from old numbers
  SPEED = 600 / 4;
  maximumSpeed = 700 / 4;
  incrementByFifty = 50 / 4;

  STATEOFCAR = 5;
  SERVOINT = 140;
  howLongToStayAtStation = 0;
  stillAtStation = false;
  serverGoFromStation = false; // change false when using the server

  // For the track
  STATIONONE = 35;
  STATIONTWO = 110;
  STATIONTHREE = 103;
  STATIONFOUR = 132;

  CHECKPOINTONE = 107;
  CHECKPOINTTWO = 23;
  CHECKPOINTTHREE = 176;
  CHECKPOINTFOUR = 213;

  // change to bool function later
  // initialize the before/afer magnet value
  isBeforeMagnet = true;
  isMagnetNotRead = true;

  // initialize the ultrasonic values
  countForUltrasonic = 0;
  stillInRangeUltrasonic = false;

  // Setup the servo
  leverArm.attach(SERVO);
  leverArm.write(SERVOINT);

  // Interrupts
  attachInterrupt(digitalPinToInterrupt(HALL_SENSOR1), IfSmartMagnetDetected, FALLING);
```

```
  attachInterrupt(digitalPinToInterrupt(HALL_SENSOR2), IfDumbMagnetDetected, FALLING);
  attachInterrupt(digitalPinToInterrupt(KILL_SWITCH), killState, FALLING);
  attachInterrupt(echoIntFront, ISRechoFront, CHANGE);

  // Needed to flush the initial XBee Serial so it does not get interrupted when turned on.
  // Only set true in the initial kill state.
  needToFlush = true;

  // This is how the vehicle performs the killed state
  inKilledState();

  if (needToFlush){
          while (Serial.available() > 0) {
          Serial.read();}}

  killStateNotRefreshed = true;

  // Set false after initial kill state.
  needToFlush = false;

  isNotAtLastStationInDestination = false;
  lastStationInDestination = 0;

  testStack = true; // To check the schedule

  isFirstPosTickThrough = true;
  isSecondPosTickThrough = true;}

//////////////////////////////////////// MAIN CODE ///////////////////////////////////////////////////////////
void loop(){
  // reattach kill switch interrupt if it was detached inside inKilledState method
  if(killStateNotRefreshed) {
          attachInterrupt(digitalPinToInterrupt(KILL_SWITCH), killState, FALLING);
          killStateNotRefreshed = false;}

  // System controls
  controls();}

//////////////////////////////////////// CONTROLS ////////////////////////////////////////////////////////////
// Function that controls the functions of the system
void controls(){

          //testRFID();
          RFIDisCard = RC522.isCard();

          // Checks that there are no RFID cards being read
          if (!RFIDisCard) {
          RFIDisCard = RC522.isCard();}

          // LED is blinking red if button is pressed
          if (killState()){
          // This is how the vehicle performs the killed state
          inKilledState();}

          // Changed distance from 30 cm to 15 cm due to pole obstacle
          // LED is Red when statement is true
          else if (isSomethingInFront()){
          STATEOFCAR = 1;
          setStateLED(STATEOFCAR);
          SPEED = 0;
          setSpeedOfMotors(SPEED);}

          // Checks to see if podcar passes an RFID card
          else if (RFIDisCard){
          STATEOFCAR = 3;
          setStateLED(STATEOFCAR);

          // Checks if the RFID is a station or not
          if (isAtStation()) {
          STATIONID = getStationNumber();

          if (lastStationInDestination == STATIONID) {
          isNotAtLastStationInDestination = false;}

          else {
          isNotAtLastStationInDestination = true;}

          // Decide to stay or go
          if (isStoppedAtStation()){
          speedUpFromStop();}

          else{
          setSpeedOfMotors(0);}}
          else{
          CHECKPOINT = getCheckpointNumber();
          isBeforeMagnet = true;
          speedUpFromStop(); }}

          // LED should be green
          else{
          STATEOFCAR = 5;
          setStateLED(STATEOFCAR);
          isMagnetNotRead = true;
          speedUpFromStop();
          leverArm.write(SERVOINT);
```

```
          howLongToStayAtStation = 0;
          getTicks();

          didSendStationPacket = true;
          serverGoFromStation = false; // uncomment when using the server}}

//////////////////////////////////// RESOLVE PACKET DATA ////////////////////////////////////////////////////////////
// Does action to the packet received from the XBee
void resolveReceivedPacketData() {

  // Checks if the Podcar number is the correct one that the server sent it to
  if (podNumber == commands[1]) {

          // Initialize the ActionId
          int actionId;

          // initialization of ActionId
          if ((String(commands[2]) + String(commands[3])).toInt() > 0) {
          actionId = (String(commands[2]) + String(commands[3])).toInt();}

          // initialization of variables for the last 4 digits in the int array when needed
          int firstPos;
          int secondPos;
          int thirdPos;
          String ticks;

          int closestStationNumberToThisCar;

          // switch based on which command was received from server
          switch (actionId) {

          // getStationNumber function
          case 1:

          // Create a packet to send to the server
          createPacket(commands[0], commands[1], commands[2], commands[3],
                    0, 0, 0, getStationNumber());

          // Sends the packet to the server
          sendXBee();
          break;

          // setStateLED function
          case 2:

          // Sets the LED color of the Pod. Mainly for testing purposes as there is no way to lock it down to one color at the
moment
          STATEOFCAR = (String(commands[6]) + String(commands[7])).toInt();
          setStateLED(STATEOFCAR);
          break;

          // getStateLED function
          case 3:

          // Create a packet to send to the server
          createPacket(commands[0], commands[1], commands[2], commands[3],
                    0, 0, 0, getStateLED());

          // Sends the packet to the server
          sendXBee();
          break;

          // setSpeed function
          case 4:

          // Sets the Speed of the Podcars. Might be useful for future iterations of this code
          SPEED = (String(commands[5]) + String(commands[6]) + String(commands[7])).toInt();
          setSpeedOfMotors(SPEED);
          break;

          //getSpeed function
          case 5:

          // First, we have to break up each spot of the speed to send to the server
          firstPos = String(SPEED).substring(0,1).toInt();
          secondPos = String(SPEED).substring(1,2).toInt();
          thirdPos = String(SPEED).substring(2).toInt();

          // Create a packet to send to the server
          createPacket(commands[0], commands[1], commands[2], commands[3],
                    0, firstPos, secondPos, thirdPos);

          // Sends the packet to the server
          sendXBee();
          break;

          // setDestination function
          case 6:

          // Use the values from the packet to create a set track from one station to the next
          setStackArray(commands[5], commands[7]);
          toStation(commands[5]);
          lastStationInDestination = commands[7];
          isNotAtLastStationInDestination = true;
          break;
```

```
        // getLocation function
        case 7:

        // Tick Value
        ticks = String(Peak_Num);

        // Location in ticks away from each checkpoint
        firstPos = ticks.substring(0,1).toInt();
        secondPos = ticks.substring(1).toInt();

        // Create a packet to send to the server
        createPacket(commands[0], commands[1], commands[2], commands[3],
                  0, getCheckpointNumber(), firstPos, secondPos);

        // Sends the packet to the server
        sendXBee();
        break;

        //setNextStation
        case 8:
        break;

        case 9:

        // Get the closest station number
        closestStationNumberToThisCar = getClosestStationNumber();

        switch (closestStationNumberToThisCar) {
        case 1:
        createPacket(commands[0], commands[1], commands[2], commands[3],
                  1, 2, 4, 3);
        break;

        case 2:
        createPacket(commands[0], commands[1], commands[2], commands[3],
                  2, 4, 3, 1);
        break;

        case 3:
        createPacket(commands[0], commands[1], commands[2], commands[3],
                  3, 1, 2, 4);
        break;

        case 4:
        createPacket(commands[0], commands[1], commands[2], commands[3],
                  4, 3, 1, 2);
        break;}

        sendXBee();

        break;

        case 10:

        // reserved actionId for sending stationNumber to server
        break;

        case 11:

        // Tell car to go from server when stopped
        serverGoFromStation = true;

        break;}

        // Resets the ActionId to zero to let the pod know to accept other commands from the server

        commands[2] = 0;
        commands[3] = 0;}

  // Reset the ActionId to zero if the pod number is incorrect so this can accept a new job
  else{
        commands[2] = 0;
        commands[3] = 0;}}

//////////////////////////////////// RECEIVE XBEE MESSAGE /////////////////////////////////////////////////////////////
// Receives the message from the server
void receiveXBeeMessage() {

  char in;
  int additive;
  int counterForCommands = 0;

  // Fill the array with the list of commands.
  while(Serial.available()) {

        // Read and translate message to int in order to store the values into an int array
        in = (char)Serial.read();

        // Translates the read message to something readable
        additive = (int)in - 48;

        // Stores the message to the int array and increments to receive a new message
        commands[counterForCommands] = additive;
        counterForCommands++;
```

```
        // Testing to see if less of the controls methods will be blocked by receiving messages
        controls();}

  // Checks if command was sent to correct XBee
  if (podNumber != (String(commands[0]) + String(commands[1])).toInt()) {
        commands[3] = 0;}}

///////////////////////////////////// SEND XBEE ////////////////////////////////////////////////////////////////
// Function to send all requested information through XBee??
void sendXBee(){

  // Initialize the complete message
  String completeMSG;

  // Check the ActionId value whether to accept values or not
  if((String(commands[2]) + String(commands[3])).toInt() > 0) {

        // Since the int array has 8 values, simply for loop through 8 times.
        for (int checkcounter = 0; checkcounter < 8; checkcounter++) {

        // Store the whole array into one String to send
        completeMSG += String(commands[checkcounter]);}

        // Send the int array as a String
        Serial.print(completeMSG + 'e');}

        // Set the ActionId to zero to accept new information
        commands[3] = 0;}

///////////////////////////////////// CREATE PACKET ////////////////////////////////////////////////////////////
void createPacket(int podNumOne, int podNumTwo, int aIdPosOne, int aIdPosTwo,
                    int contentOne, int contentTwo, int contentThree, int contentFour){

  // Fill the array to send data back to the server
  commands[0] = podNumOne;
  commands[1] = podNumTwo;
  commands[2] = aIdPosOne;
  commands[3] = aIdPosTwo;
  commands[4] = contentOne;
  commands[5] = contentTwo;
  commands[6] = contentThree;
  commands[7] = contentFour;}

// Sets the next station number
// [ActionId = 1, lengthOfArray = 3, the next station number between 1-4]
void setNextStation(int station_number){
        STATIONID = station_number;
        SSN = 0;}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////START GET FUNCTIONS/////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

int getClosestStationNumber() {

  int lastCheckpoint = getCheckpointNumber();
  bool magnetCheck = isBeforeMagnet;

  switch(lastCheckpoint) {
        case 1:
        if (magnetCheck) {
        return 2;}
        else {
        return 4;}
        break;

        case 2:
        if (magnetCheck) {
        return 4;}
        else {
        return 3;}
        break;

        case 3:
        if (magnetCheck) {
        return 3;}
        else {
        return 1;}
        break;

        case 4:
        if (magnetCheck) {
        return 1;}
        else{
        return 2;}
        break;}}

///////////////////////////////////// GET CHECKPOINT /////////////////////////////////////////////////////////////
int getCheckpointNumber() {

  /* If so then get its serial number */
        RC522.readCardSerial();

        /* Get the Serial number of the tag */
```

```
          SSN = RC522.serNum[0];

  // Next four statements are for checkpoint cards
          if (SSN == CHECKPOINTONE){
          CHECKPOINT = 1;}

          if (SSN == CHECKPOINTTWO){
          CHECKPOINT = 2;}

          if (SSN == CHECKPOINTTHREE){
          CHECKPOINT = 3;}

          if (SSN == CHECKPOINTFOUR){
          CHECKPOINT = 4;}

          return CHECKPOINT;}

//////////////////////////////////// GET STATION NUMBER /////////////////////////////////////
// Retrieves the Station ID
int getStationNumber(){
          /* If so then get its serial number */
          RC522.readCardSerial();

          /* Get the Serial number of the tag */
          SSN = RC522.serNum[0];

          // For round tags SSN = 85, For Cards SSN = 35
          if (SSN == STATIONONE){
          STATIONID = 1;}
          // For round tags SSN = 101, For Cards SSN = 110
          if (SSN == STATIONTWO){
          STATIONID = 2;}
          // For round tags SSN = 149, For Cards SSN = 103
          if (SSN == STATIONTHREE){
          STATIONID = 3;}
          // For round tags SSN = 21, For Cards SSN = 132
          if (SSN == STATIONFOUR){
          STATIONID = 4;}

          return STATIONID;}

//////////////////////////////////// GET TICKS //////////////////////////////////////////////////////////
int getTicks(){

  if (!RFIDisCard) {

          int pos1 = analogRead(POS_SENSOR1);
          int pos2 = analogRead(POS_SENSOR2);

          if(pos1 < 285 && isFirstPosTickThrough){
          Peak_Num++;
          isFirstPosTickThrough = false;}

          else if (pos1 < 285){
          // do nothing
          }

          else {
          isFirstPosTickThrough = true;}

          if(pos2 < 315 && isSecondPosTickThrough){
          Peak_Num++;
          isSecondPosTickThrough = false;}

          else if (pos2 < 315) {
          // do nothing
          }

          else{
          isSecondPosTickThrough = true;}}

          else{

          if (getCheckpointNumber() != tickCounterStation) {
          tickCounterStation = getCheckpointNumber();
          Peak_Num = 0;}}

  return Peak_Num;}

//////////////////////////////////// GET POSITION //////////////////////////////////////////////////////////
int getPosition(){
  // In cm
  return Peak_Num * peakgap / 10;}

//////////////////////////////////// GET LOCATION //////////////////////////////////////////////////////////
void getLocation(){
  // Don't know what to do with this just yet
  sendXBee();}

//////////////////////////////////// GET STATE LED //////////////////////////////////////////////////////////
// Gets the state of the LED
// Used for mobile app purposes
int getStateLED(){
  return STATEOFCAR;}
```

```
//////////////////////////////////// GET SPEED OF MOTORS ///////////////////////////////////////////////////
//// Retrieves the speed of the motor based on the encoders
double getSpeedOfMotors(){
  return SPEED;}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////START SET FUNCTIONS/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////// SET SPEED OF MOTORS ///////////////////////////////////////////////////
// Function that sets the speed of the motors
void setSpeedOfMotors(double motor_speed){

  analogWrite(MOTOR1, motor_speed);
  analogWrite(MOTOR2, motor_speed);}

// Turns on the LED based on one of the specified states
// [ActionId = 3, sets the state between 1-4 atm or 0 if not on]
void setStateLED(int colorState){

  // LED is red (Station arrival or collision prevention)
  if (colorState == 1){
        digitalWrite(RED, ON);
        digitalWrite(GREEN, OFF);
        digitalWrite(BLUE, OFF);}

  // LED is blinking red (error)
  if (colorState == 2){
        if (currentBlink - previousBlink >= interval){

        previousBlink = currentBlink;

        if (redState == OFF){
        redState = ON;
        greenState = OFF;
        blueState = OFF;}

        else{
        redState = OFF;
        greenState = OFF;
        blueState = OFF;}

        // Set the state of the LED
        digitalWrite(RED, redState);
        digitalWrite(GREEN, greenState);
        digitalWrite(BLUE, blueState);}

        // Checks to see if the reset button has been pressed
        reset_state = digitalRead(RESET);}

  // LED is blue (checkpoint detected)
  if (colorState == 3){
        digitalWrite(RED, OFF);
        digitalWrite(GREEN, OFF);
        digitalWrite(BLUE, ON);}

  if (colorState == 4){
        if (currentBlink - previousBlink >= interval){

        previousBlink = currentBlink;

        if (blueState == OFF){
        redState = OFF;
        greenState = OFF;
        blueState = ON;}

        else{
        redState = OFF;
        greenState = OFF;
        blueState = OFF;}

        // Set the state of the LED
        digitalWrite(RED, redState);
        digitalWrite(GREEN, greenState);
        digitalWrite(BLUE, blueState);}}

  // LED is green (podcar in motion)
  if (colorState == 5){
        digitalWrite(RED, OFF);
        digitalWrite(GREEN, ON);
        digitalWrite(BLUE, OFF);}

  if (colorState == 6){
        if (currentBlink - previousBlink >= interval){
        previousBlink = currentBlink;

        if (greenState == OFF){
        redState = OFF;
        greenState = ON;
        blueState = OFF;}

        else{
        redState = OFF;
        greenState = OFF;
        blueState = OFF;}
```

```
        // Set the state of the LED
        digitalWrite(RED, redState);
        digitalWrite(GREEN, greenState);
        digitalWrite(BLUE, blueState);}}

  // LED is purple (podcar in motion)
  if (colorState == 7){
        digitalWrite(RED, ON);
        digitalWrite(GREEN, OFF);
        digitalWrite(BLUE, ON);}

  if (colorState == 8){
        if (currentBlink - previousBlink >= interval){
        previousBlink = currentBlink;

        if (redState == OFF && blueState == OFF){
        redState = ON;
        greenState = OFF;
        blueState = ON;}

        else{
        redState = OFF;
        greenState = OFF;
        blueState = OFF;}

        // Set the state of the LED
        digitalWrite(RED, redState);
        digitalWrite(GREEN, greenState);
        digitalWrite(BLUE, blueState);}}

  // LED is yellow (podcar in motion)
  if (colorState == 9){
        digitalWrite(RED, ON);
        digitalWrite(GREEN, ON);
        digitalWrite(BLUE, OFF);}

  if (colorState == 10){
        if (currentBlink - previousBlink >= interval){
        previousBlink = currentBlink;

        if (redState == OFF && greenState == OFF){
        redState = ON;
        greenState = ON;
        blueState = OFF;}

        else{
        redState = OFF;
        greenState = OFF;
        blueState = OFF;}

        // Set the state of the LED
        digitalWrite(RED, redState);
        digitalWrite(GREEN, greenState);
        digitalWrite(BLUE, blueState);}}

  checkBlinks();}

void toStation(int startingDestination){

  int magnetCheck = isBeforeMagnet;

  // Check if pod just left station 2 or 3
  if ((STATIONID == 2 && CHECKPOINT == 1) || (STATIONID == 3 && CHECKPOINT == 3)){

        // Check if just left station 2
        if (STATIONID == 2){

        // Give the podcar an array directing it towards its starting destination
        switch (startingDestination) {

        case 1:

        for(int i = (sizeof(switchIntoClosestStation)/2)-1; i > -1 ; i--){
                scheduleArray.push(switchIntoClosestStation[i]);}

        break;

        case 2:

        for(int i = (sizeof(halfCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
                scheduleArray.push(halfCircleIntoStationTwoThree[i]);}

        break;

        case 3:

        for(int i = (sizeof(fullCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
                scheduleArray.push(fullCircleIntoStationTwoThree[i]);}

        break;

        case 4:

        for(int i = (sizeof(fromMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
```

```
                    scheduleArray.push(fromMiddleIntoStationOneFour[i]);}

        break;}}

        if (STATIONID == 3){

        switch (startingDestination){

        case 1:

        for(int i = (sizeof(fromMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
                    scheduleArray.push(fromMiddleIntoStationOneFour[i]);}

        break;

        case 2:

        for(int i = (sizeof(fullCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
                    scheduleArray.push(fullCircleIntoStationTwoThree[i]);}

        break;

        case 3:

        for(int i = (sizeof(halfCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
                    scheduleArray.push(halfCircleIntoStationTwoThree[i]);}

        break;

        case 4:

        for(int i = (sizeof(switchIntoClosestStation)/2)-1; i > -1 ; i--){
                    scheduleArray.push(switchIntoClosestStation[i]);}

        break;}}}

  else{

        switch (startingDestination){

        case 1:
        goToStationOne();
        break;

        case 2:
        goToStationTwo();
        break;

        case 3:
        goToStationThree();
        break;

        case 4:
        goToStationFour();
        break;}}}

void goToStationOne(){

  int checkpoint = getCheckpointNumber();
  int magnetCheck = isBeforeMagnet;

  switch (checkpoint){

        case 1:

        for(int i = (sizeof(fromMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
        scheduleArray.push(fromMiddleIntoStationOneFour[i]);}

        break;

        case 2:

        if(magnetCheck){
        for(int i = (sizeof(fromMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
        scheduleArray.push(fromMiddleIntoStationOneFour[i]);}}

        else{
        for(int i = (sizeof(passMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
        scheduleArray.push(passMiddleIntoStationOneFour[i]);}}

        break;

        case 3:

        for(int i = (sizeof(switchIntoClosestStation)/2)-1; i > -1 ; i--){
        scheduleArray.push(switchIntoClosestStation[i]);}

        break;

        case 4:

        if(magnetCheck){
        for(int i = (sizeof(switchIntoClosestStation)/2)-1; i > -1 ; i--){
        scheduleArray.push(switchIntoClosestStation[i]);}}
```

```
          else{
          for(int i = (sizeof(exactlyAfterStationOneFour)/2)-1; i > -1 ; i--){
          scheduleArray.push(exactlyAfterStationOneFour[i]);}}

          break;}}

void goToStationTwo(){

  int checkpoint = getCheckpointNumber();
  int magnetCheck = isBeforeMagnet;

  switch (checkpoint){

          case 1:

          for(int i = (sizeof(fullCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(fullCircleIntoStationTwoThree[i]);}

          break;

          case 2:

          if(magnetCheck){
          for(int i = (sizeof(fullCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(fullCircleIntoStationTwoThree[i]);}}

          else{
          for(int i = (sizeof(afterStationOneFourToTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(afterStationOneFourToTwoThree[i]);}}

          break;

          case 3:

          for(int i = (sizeof(halfCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(halfCircleIntoStationTwoThree[i]);}

          break;

          case 4:

          if(magnetCheck){
          for(int i = (sizeof(halfCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(halfCircleIntoStationTwoThree[i]);}}

          else{
          for(int i = (sizeof(exactlyBeforeStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(exactlyBeforeStationTwoThree[i]);}}

          break;}}

void goToStationThree(){

  int checkpoint = getCheckpointNumber();
  int magnetCheck = isBeforeMagnet;

  switch (checkpoint){

          case 1:

          for(int i = (sizeof(halfCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(halfCircleIntoStationTwoThree[i]);}

          break;

          case 2:

          if(magnetCheck){
          for(int i = (sizeof(halfCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(halfCircleIntoStationTwoThree[i]);}}

          else{
          for(int i = (sizeof(exactlyBeforeStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(exactlyBeforeStationTwoThree[i]);}}

          break;

          case 3:

          for(int i = (sizeof(fullCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(fullCircleIntoStationTwoThree[i]);}

          break;

          case 4:

          if(magnetCheck){
          for(int i = (sizeof(fullCircleIntoStationTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(fullCircleIntoStationTwoThree[i]);}}

          else{
          for(int i = (sizeof(afterStationOneFourToTwoThree)/2)-1; i > -1 ; i--){
          scheduleArray.push(afterStationOneFourToTwoThree[i]);}}

          break;}}
```

```
void goToStationFour(){

  int checkpoint = getCheckpointNumber();
  int magnetCheck = isBeforeMagnet;

  switch (checkpoint){

          case 1:

          for(int i = (sizeof(switchIntoClosestStation)/2)-1; i > -1 ; i--){
          scheduleArray.push(switchIntoClosestStation[i]);}

          break;

          case 2:

          if(magnetCheck){
          for(int i = (sizeof(switchIntoClosestStation)/2)-1; i > -1 ; i--){
          scheduleArray.push(switchIntoClosestStation[i]);}}

          else{
          for(int i = (sizeof(exactlyAfterStationOneFour)/2)-1; i > -1 ; i--){
          scheduleArray.push(exactlyAfterStationOneFour[i]);}}

          break;

          case 3:

          for(int i = (sizeof(fromMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
          scheduleArray.push(fromMiddleIntoStationOneFour[i]);}

          break;

          case 4:

          if(magnetCheck){
          for(int i = (sizeof(fromMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
          scheduleArray.push(fromMiddleIntoStationOneFour[i]);}}

          else{
          for(int i = (sizeof(passMiddleIntoStationOneFour)/2)-1; i > -1 ; i--){
          scheduleArray.push(passMiddleIntoStationOneFour[i]);}}

          break;}}

/////////////////////////////////// SET STACK ARRAY /////////////////////////////////////////////////////////
void setStackArray(int from, int to){

  // Generate the value to compare to each of these
  int fromTo = (String(from) + String(to)).toInt();

  if (scheduleArray.isEmpty()){

          // Create the stack array to process the course that the pod car has to take
          switch(fromTo){
          case 12:
          for(int i = (sizeof(path1to2)/2)-1; i > -1 ; i--){
          scheduleArray.push(path1to2[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 13:
          for(int i = (sizeof(path1to3)/2)-1; i > -1 ; i--){
          scheduleArray.push(path1to3[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 14:
          for(int i = (sizeof(path1to4)/2)-1; i > -1 ; i--){
          scheduleArray.push(path1to4[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 21:
          for(int i = (sizeof(path2to1)/2)-1; i > -1 ; i--){
          scheduleArray.push(path2to1[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 23:
          for(int i = (sizeof(path2to3)/2)-1; i > -1 ; i--){
          scheduleArray.push(path2to3[i]);}

          break;

          case 24:
          for(int i = (sizeof(path2to4)/2)-1; i > -1 ; i--){
          scheduleArray.push(path2to4[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
```

```
          break;

          case 31:
          for(int i = (sizeof(path3to1)/2)-1; i > -1 ; i--){
          scheduleArray.push(path3to1[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 32:
          for(int i = (sizeof(path3to2)/2)-1; i > -1 ; i--){
          scheduleArray.push(path3to2[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 34:
          for(int i = (sizeof(path3to4)/2)-1; i > -1 ; i--){
          scheduleArray.push(path3to4[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 41:
          for(int i = (sizeof(path4to1)/2)-1; i > -1 ; i--){
          scheduleArray.push(path4to1[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;

          case 42:
          for(int i = (sizeof(path4to2)/2)-1; i > -1 ; i--){
          scheduleArray.push(path4to2[i]);}

          break;

          case 43:
          for(int i = (sizeof(path4to3)/2)-1; i > -1 ; i--){
          scheduleArray.push(path4to3[i]);}

          helpSetStackArray(firstChoiceForInitialPath);
          break;}}}
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/////////////////////////////////////////////START HELPER FUNCTIONS//////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

///////////////////////////////// CHECK BLINKS //////////////////////////////////////////////////
void checkBlinks(){

  if (currentBlink > 20){
          currentBlink = 10;
          previousBlink = 5;}

  currentBlink++;}

///////////////////////////////////// IS AT STATION ///////////////////////////////////////////////////////////
// Checks if the RFID card is at a station or not
bool isAtStation(){

  /* If so then get its serial number */
  RC522.readCardSerial();

  /* Get the Serial number of the tag */
  SSN = RC522.serNum[0];

  // Checks for each Station Number to see if the Podcar is at a Station
  if (SSN == STATIONONE || SSN == STATIONTWO || SSN == STATIONTHREE || SSN == STATIONFOUR){
          return true;}

  else{
          return false;}}

//////////////////////////////////// IS SOMETHING IN FRONT ///////////////////////////////////////////////////
// Detects if anything is in front of the podcar
bool isSomethingInFront(){

  distanceSide = 20;

  // Check for if either the front or side ultrasonic have something in front of them within a certain amount of time.
  if ((ultrasonicCounterFront > 6 && distanceFront < 16) || (ultrasonicCounterSide > 5 && distanceSide < 7)){
          return true;}

  // Check if something is in front of either of them
  else if (distanceFront < 16 || distanceSide < 7){

          ultrasonicCounterFront++;
          ultrasonicCounterSide++;
          return false;}

  // If nothing is in front, set the counts back to zero.
  else{
          ultrasonicCounterFront = 0;
          ultrasonicCounterSide = 0;
          return false;}}
```

```
//////////////////////////////////// INTERRUPT SERVICE ROUTINE FOR ECHO PIN 1 ///////////////////////////////////////////////////
// When a pulse is received, this function interrupts and resolves the action in the method.
void ISRechoFront(){

  // do a certain value based on the pulse sent.
  switch(digitalRead(ECHOFRONT)){

        // On HIGH, record the time at the start of the pulse with millis() and reset the end of the pulse by setting it to 0.
        case HIGH:
        echoEndFront = 0;
        echoStartFront = micros();
        break;

        // On LOW, record the end of the pulse time with millis(), get the difference of the start and end value, and calculate
the distance in cms.
        case LOW:
        echoEndFront = micros();
        echoDurationFront = echoEndFront - echoStartFront;

        // calculating distance in cms here
        distanceFront = echoDurationFront / 58;
        break;}}

//////////////////////////////////// INTERRUPT SERVICE ROUTINE FOR PULSE PIN 1 ///////////////////////////////////////////////////
// Sends a pulse every 30 microseconds through the Timer3 library. This will help free up some time in the loop() function
void ISRpulseFront(){

  // Make them change state based on certain criteria
  static volatile int state = 0;

  // Change to state 1 when the timer runs all the way down
  if (triggerTimeEventFront-- < 0){
        triggerTimeEventFront = TICK_COUNTS;
        state = 1;}

  switch(state){

        // do nothing state
        case 0:
        break;

        // send pulse to the echo pin with HIGH value and change state to 2
        case 1:
        digitalWrite(TRIGFRONT, HIGH);
        state = 2;
        break;

        // send pulse to the echo pin with LOW value and change state back to 0 to start counting again
        case 2:
        default:
        digitalWrite(TRIGFRONT, LOW);
        state = 0;
        break;}}

////////////////////////////////// KILL STATE ////////////////////////////////////////////////////////////////
// Checks the status of the button
bool killState(){

  if (digitalRead(KILL_SWITCH) < 1){
        return true;}

  else{
        return false;}}

//////////////////////////////////// MAGNET READ /////////////////////////////////////////////////////////////
// Retrieves the readings from the hall effect sensors
void magnetRead(){
  // Checks if any magnet is read on the left of podcar
  hall_1_state = digitalRead(HALL_SENSOR1);

  // Checks if any magnet is read on the right of podcar
  hall_2_state = digitalRead(HALL_SENSOR2);}

//////////////////////////////// IF SMART MAGNET DETECTED //////////////////////////////////////////////////
void IfSmartMagnetDetected(){
        // Check for the magnet's value
        magnetRead();

  if (hall_1_state == 0){ //If smart sensor detects a magnet
        SERVOINT = 20;
        leverArm.write(SERVOINT);}}

//////////////////////////////// IF DUMB MAGNET DETECTED ///////////////////////////////////////////////////
// Switches the lever arm if a dumb magnet is detected
void IfDumbMagnetDetected(){
  // Check for the magnet's value
  magnetRead();

  //If dumb/right sensor detects a magnet
  if (hall_2_state == 0){

        // Switch to right arm
        SERVOINT = 140;
        leverArm.write(SERVOINT);}}
```

```
///////////////////////////////// SPEED UP FROM STOP ////////////////////////////////////////////////////
// Makes the Car accelerate to a certain maximum speed
void speedUpFromStop(){

  if (SPEED < maximumSpeed){
          SPEED += incrementByFifty;}

  setSpeedOfMotors(SPEED);}

// example of interrupt kill state
void kindaKillState(){

  int stay = 1;

  while (stay == 1){
          STATEOFCAR = 2;
          setStateLED(STATEOFCAR);
          SPEED = 0;
          setSpeedOfMotors(SPEED);

          if (digitalRead(RESET) < 1){
          stay = 0;}}}

///////////////////////////////// INSIDE THE KILLED STATE ////////////////////////////////////////////////////
// This is how the vehicle performs the killed state
void inKilledState(){

  // Creates a latching button
  int stayInKill = 1;
  bool killPressedOnce = true;
  bool isSwitchArms = false;
  int lastServoInt = SERVOINT;

  // Make it so that the kill switch interrupt pin is reattached later.
  killStateNotRefreshed = true;

  // detach the kill switch for the kill state
  detachInterrupt(digitalPinToInterrupt(KILL_SWITCH));

  // Continues to blink the LED red while latching button is in effect
  // Only way to stop this cycle is to reset Arduino or press reset button
  while(stayInKill == 1){
          // Makes sure that the arms dont switch when entering kill state
          if (lastServoInt == 140){
          killPressedOnce = false;
          lastServoInt = 0;}

          // Check for the presses after the kill switch is pressed. Switch arms after the first button press to get into the
kill state.
          if (digitalRead(KILL_SWITCH) < 1 && killPressedOnce){
          killPressedOnce = false;
          isSwitchArms = !isSwitchArms;}

          else if (digitalRead(KILL_SWITCH) < 1){
          // This makes sure that if the button is held down too long, it does nothing.
          }
          else{
          // On unpress of the button, let the button get the initial press function back.
          killPressedOnce = true;}

          // Switch arms based on kill switch presses.
          if (isSwitchArms){
          SERVOINT = 20;
          leverArm.write(SERVOINT);}

          else{
          SERVOINT = 140;
          leverArm.write(SERVOINT);}

          STATEOFCAR = 2;
          setStateLED(STATEOFCAR);
          SPEED = 0;
          setSpeedOfMotors(SPEED);

          if (digitalRead(RESET) < 1){
          stayInKill = 0;}

          delay(35);}

  leverArm.write(SERVOINT);
  killStatePressedTwice = 0;}


void switchArms(){

  if (SERVOINT == 140){
          SERVOINT = 20;
          leverArm.write(SERVOINT);}
  else{
          SERVOINT = 140;
          leverArm.write(SERVOINT);}}

///////////////////////////////// STOP AT A STATION ////////////////////////////////////////////////////
// Stops the car at a station for a bit
```

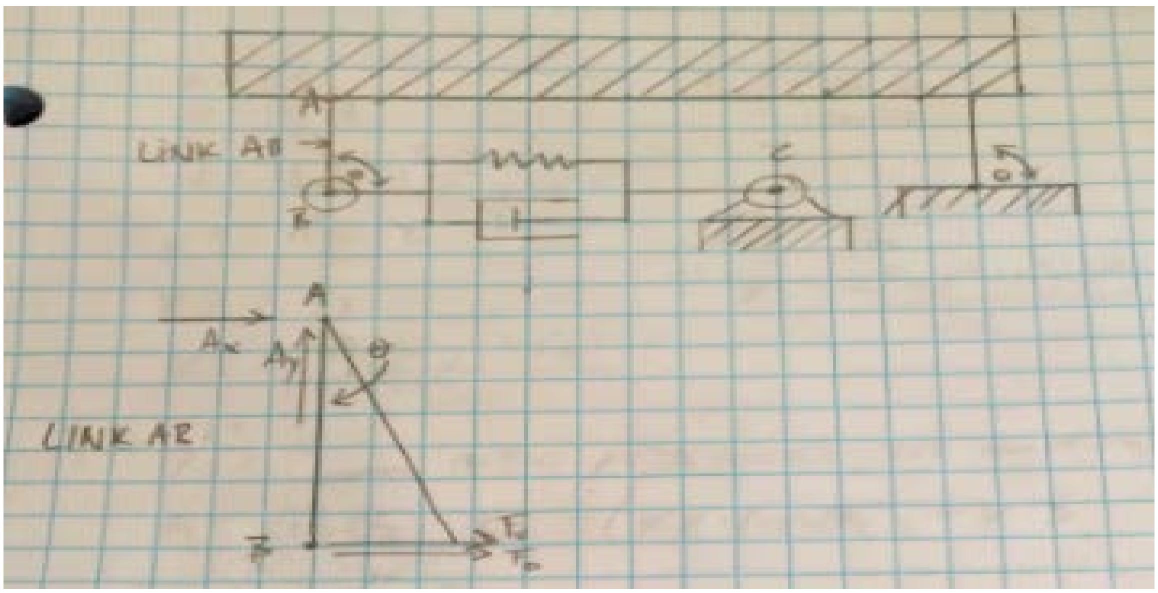
```
bool isStoppedAtStation(){

  if (howLongToStayAtStation > 100){
          return true;}

  else{
          howLongToStayAtStation++;
          return false;}}
```

Appendix B: Half-Scale

Active Suspension

Appendix B1-1

Vibrational Analysis



Free-Body Diagram on Suspension System for Vibrational Analysis

Referring to above, after constructing a FBD for the suspension system the sum of all moments was applied to it.

$$\Sigma M_A = I_A \frac{d^2\theta}{dt^2} \curvearrowright_+$$

$$-F_d\ (\alpha) - F_s\ (\alpha)\ = I_a\theta$$

$$F_s = k\delta_B\ \&\ F_d = c\delta_B * \frac{d\ \ \theta}{dt}$$

Therefore,

$$F_s = k * sin\theta\ \&\ F_d = c(\frac{d}{dt}\alpha * sin\theta)$$

$$I_A \frac{d^2\theta}{dt^2} + c\frac{d}{dt}(\alpha * sin\theta) * \alpha + k * (\alpha * sin\theta) * \alpha\ = 0$$

$$sin\theta \ = \theta \ \& \ cos\theta \ = 1$$

$$I_A \frac{d^2\theta}{dt^2} + c\alpha^2 \frac{d \quad \theta}{dt} + k\alpha^2\theta = 0$$

Comparing this equation with:

$$m \frac{d^2x}{dt^2} + c \frac{d \quad x}{dt} + kx = 0$$

Then,

$$I_A = m$$

$$c\alpha^2 = c$$

$$k\alpha^2 = k$$

After applying the governing equations,

$$\omega_n = \sqrt{\frac{k\alpha^2}{m}}$$

$$\zeta = \frac{c}{2m\omega_n}$$

Appendix B1-2

MATLAB code for Force Analysis of Tilt Mechanism

```
%%Phase 1 Angle Calculations
clear all
close all
link1 = 3.51;
link2 = 12;
link3 = 14.49;
link4 = 18.26;
k1 = link4/link1;
k2 = link4/link3;
k3 = ((link1^2)-(link2^2)+(link3^2)+(link4^2))/(2*link1*link3);
k4 = link4/link2;
k5 = ((link3^2)-(link4^2)-(link1^2)-(link2^2))/(2*link1*link2);
inputangle = 5;
inputradians = ((inputangle)*pi)/180;
angle2 = inputradians;
a = cos(angle2)-k1-k2*cos(angle2)+k3;
b = -2*sin(angle2);
c = k1-(k2+1)*cos(angle2)+k3;
d = cos(angle2)-k1-k4*cos(angle2)+k5;
e = -2*sin(angle2);
f = k1+(k4-1)*cos(angle2)+k5;
angle4root1radians = 2*(atan((-b+(b.^2-4.*a.*c).^(1/2))/(2.*a)));
angle4root2radians = 2*(atan((-b-(b.^2-4.*a.*c).^(1/2))/(2.*a)));
angle3root1radians = 2*(atan((-e+(e.^2-4.*d.*f).^(1/2))/(2.*d)));
angle3root2radians = 2*(atan((-e-(e.^2-4.*d.*f).^(1/2))/(2.*d)));
angle4root1degrees = ((angle4root1radians)*180)/pi;
angle4root2degrees = ((angle4root2radians)*180)/pi;
angle3root1degrees = ((angle3root1radians)*180)/pi;
angle3root2degrees = ((angle3root2radians)*180)/pi;
angle2degrees = inputangle;
%% Phase 2
M = 300;
```

```
N = 18.66; %distance between the first link and third link

inputangle4 = angle4root2radians;
input4degrees = ((inputangle4)*180)/pi;
angle5degrees = input4degrees - 90;
angle5radians = ((angle5degrees)*pi)/180;

x1 = 1.755*cos(angle2);
y1 = 1.755*sin(angle2);
x2 = 7.245*sin(angle3root2radians);
y2 = 7.245*cos(angle3root2radians);
x3 = 6*sin(angle5radians);
y3 = 6*cos(angle5radians);

%% Phase 3 simultaneous solver
syms A B C D E F G H I J K L
eqn1 = A+C == 0;
eqn2 = B+C+M == 0;
eqn3 = (C*y1)-(D*x1)-(A*y1)+(B*x1)+(M*x1) == 0;
eqn4 = E+G == 0;
eqn5 = F+H+M == 0;
eqn6 = (E*y2)-(F*x2)+(H*x2)-(G*y2)-(M*x2) == 0;
eqn7 = I+K == 0;
eqn8 = J+L == 0;
eqn9 = (L*x3)+(K*y3)-(I*y3)-(J*x3) == 0;
eqn10 = C+K+E+A+G+I == 0;
eqn11 = B+D+F+H+M+J+L == 0;
eqn12 = -(D*N)-(M*(x3+2*x2))-(A*2*y1)-(E*2*y1)-(B*(x3+2*x2))-(F*(x3+2*x2))-(H*2*x3)-(J*2*x3)-(I*2*y3)-(G*2*y3) == 0;

sol = solve([eqn1, eqn2, eqn3, eqn4, eqn5, eqn6, eqn7, eqn8, eqn9, eqn10, eqn11, eqn12 ], [A, B, C, D, E, F, G, H, I, J, K, L]);
ASol = sol.A;
BSol = sol.B;
CSol = sol.C;
DSol = sol.D;
ESol = sol.E;
FSol = sol.F;
GSol = sol.G;
HSol = sol.H;
ISol = sol.I;
JSol = sol.J;
KSol = sol.K;
LSol = sol.L;

A = double(ASol)
B = double(BSol)
C = double(CSol)
D = double(DSol)
E = double(ESol)
F = double(FSol)
G = double(GSol)
H = double(HSol)
I = double(ISol)
J = double(JSol)
K = double(KSol)
L = double(LSol)
```

Output of MATLAB Code:

```
A = 164.3815

B = -135.6185

C = -164.3815

D =  135.6185

E =  151.5916

F = -188.4405

G = -151.5916

H = -111.5595

I =      0

J =      0

K =      0

L =      0
```

Appendix B1-3

MATLAB code for Force Analysis of Scissor Lift Mechanism

```
syms A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A1 B1 C1 D1...
    E1 F1 G1 H1 I1 J1 K1 L1 M1 N1 O1 P1 Q1 R1 S1 T1 U1 V1
eqn1 = A+C == 0;
eqn2 = B+D == 0;
eqn3 = -(B*20)+(75*20) == 0;
eqn4 = 900+E+G+K+I == 0;
eqn5 = F+H+L+J-75 == 0;
eqn6 = (6.17*H)+(G*4.27)+(J*16.45)-(I*11.25)-(900*11.25)+(L*16.14)+...
    (K*11.25) == 0;
eqn7 = M+P+Q-900 == 0;
eqn8 = O+R+M-75 == 0;
eqn9 = -(75*16.41)-(N*16.41)-(M*11.25) == 0;
eqn10 = S+U+W-900 == 0;
eqn11 = T+X+V == 0;
eqn12 =(T*2.71)+(S*7) == 0;
eqn13 = D1-900+Y+A1 == 0;
eqn14 = Z+B1+C1+F1-75 == 0;
eqn15 = -(75*16.45)-(F1*16.45)-(E1*11.38)+(C1*10.29)+(D1*7) == 0;
eqn16 = I1+G1+900+K1 == 0;
eqn17 = H1+J1+L1 == 0;
eqn18 =(K1*11.38)-(L1*16.45) == 0;
eqn19 = Q1+M1+O1+900 == 0;
eqn20 = N1+P1+R1 == 0;
eqn21 = (Q1*7)+(R1*2.71) == 0;
eqn22 = Q1+M1+O1+900 == 0;
eqn23 = N1+P1+R1 == 0;
eqn24 = (Q1*7)+(R1*2.71) == 0;
eqn25 = S1+U1 == 0;
eqn26 = T1+V1-150 == 0;
eqn27 = -(T1*20)+(75*20) == 0;

sol = solve([eqn1, eqn2, eqn3, eqn4, eqn5, eqn6, eqn7, eqn8, eqn9,...
    eqn10, eqn11, eqn12, eqn13, eqn14, eqn15, eqn16, eqn17, eqn18,...
    eqn19, eqn20, eqn21, eqn22, eqn23, eqn24, eqn25, eqn26, eqn27],...
    [A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U,...
    V, W, X, Y, Z, A1, B1, C1, D1 , E1, F1, G1, H1, I1, J1, K1, L1,...
    M1, N1, O1, P1, Q1, R1, S1, T1, U1, V1]);
ASol = sol.A;
BSol = sol.B;
CSol = sol.C;
DSol = sol.D;
ESol = sol.E;
FSol = sol.F;
GSol = sol.G;
HSol = sol.H;
ISol = sol.I;
JSol = sol.J;
KSol = sol.K;
LSol = sol.L;
MSol = sol.M;
NSol = sol.N;
OSol = sol.O;
PSol = sol.P;
QSol = sol.Q;
RSol = sol.R;
SSol = sol.S;
TSol = sol.T;
USol = sol.U;
VSol = sol.V;
WSol = sol.W;
XSol = sol.X;
YSol = sol.Y;
ZSol = sol.Z;
A1Sol = sol.A1;
B1Sol = sol.B1;
C1Sol = sol.C1;
D1Sol = sol.D1;
E1Sol = sol.E1;
F1Sol = sol.F1;
G1Sol = sol.G1;
H1Sol = sol.H1;
I1Sol = sol.I1;
J1Sol = sol.J1;
K1Sol = sol.K1;
L1Sol = sol.L1;
M1Sol = sol.M1;
N1Sol = sol.N1;
O1Sol = sol.O1;
P1Sol = sol.P1;
Q1Sol = sol.Q1;
R1Sol = sol.R1;
S1Sol = sol.S1;
T1Sol = sol.T1;
U1Sol = sol.U1;
V1Sol = sol.V1;
```

```
A = double(ASol)
B = double(BSol)
C = double(CSol)
D = double(DSol)
E = double(ESol)
F = double(FSol)
G = double(GSol)
H = double(HSol)
I = double(ISol)
J = double(JSol)
K = double(KSol)
L = double(LSol)
M = double(MSol)
N = double(NSol)
O = double(OSol)
P = double(PSol)
Q = double(QSol)
R = double(RSol)
S = double(SSol)
T = double(TSol)
U = double(USol)
V = double(VSol)
W = double(WSol)
X = double(XSol)
Y = double(YSol)
Z = double(ZSol)
A1 = double(A1Sol)
B1 = double(B1Sol)
C1 = double(C1Sol)
D1 = double(D1Sol)
E1 = double(E1Sol)
F1 = double(F1Sol)
G1 = double(G1Sol)
H1 = double(H1Sol)
I1 = double(I1Sol)
J1 = double(J1Sol)
K1 = double(K1Sol)
L1 = double(L1Sol)
M1 = double(M1Sol)
N1 = double(N1Sol)
O1 = double(O1Sol)
P1 = double(P1Sol)
Q1 = double(Q1Sol)
R1 = double(R1Sol)
S1 = double(S1Sol)
T1 = double(T1Sol)
U1 = double(U1Sol)
V1 = double(V1Sol)
```

Output:

```
A =     0
B =    75
C =     0
D =   -75
E =  -3.2712e+03
F =    75
G =   2.3712e+03
H =     0
I =     0
J =     0
K =     0
L =     0
M =   900
N = -692.0018
O =  -825
P =     0
Q =     0
R =     0
S =     0
T =     0
U =   900
V =     0
W =     0
X =     0
Y =   900
Z =  -44.8980
A1 =    0
B1 =    0
C1 =  119.8980
D1 =    0
```

```
E1 =    0
F1 =    0
G1 =  -900
H1 =    0
I1 =    0
J1 =    0
K1 =    0
L1 =    0
M1 =  -900
N1 =    0
O1 =    0
P1 =    0
Q1 =    0
R1 =    0
S1 =    0
T1 =    75
U1 =    0
V1 =    75
```

Appendix B1-4

Torque Calculation

Using the results from Appendix B1-2 we can calculate the torque needed to move the mechanism.

$$\boldsymbol{\tau} = \mathrm{r \ x \ F}$$

$$\boldsymbol{\tau} = 3.51\mathrm{in \ x \ 164 \ lbf}$$

$$\boldsymbol{\tau} = 575.64 \mathrm{\ lbf \ x \ in} = 65.04 \mathrm{\ N \ x \ m}$$

Appendix B1-5

Calculation Results

SUS005 FEA Results

Creo Simulate Structure Version P-10-38:spg

Summary for Design Study "Analysis1"

Tue Dec 13, 2016   20:23:58

Run Settings

Memory allocation for block solver: 512.0

Parallel Processing Status

Parallel task limit for current run: 8

Parallel task limit for current platform: 64

Number of processors detected automatically: 8

Checking the model before creating elements...

These checks take into account the fact that AutoGEM will

automatically create elements in volumes with material

properties, on surfaces with shell properties, and on curves

with beam section properties.

Generate elements automatically.

Checking the model after creating elements...

No errors were found in the model.

Creo Simulate Structure Model Summary

Principal System of Units: Inch lbm Second (Pro/E Default)

Length: in

Mass: lbm

Time: sec

Temperature: F

Model Type: Three Dimensional

Points: 51

Edges: 192

Faces: 227

Springs: 0

Masses: 0

Beams: 0

Shells: 0

Solids: 87

Elements: 87

Standard Design Study

Static Analysis "Analysis1":

Convergence Method: Multiple-Pass Adaptive

Plotting Grid: 4

Convergence Loop Log: (20:23:58)

>> Pass 1 <<

Calculating Element Equations (20:23:58)

Total Number of Equations: 126

Maximum Edge Order: 1

Solving Equations (20:23:58)

Post-Processing Solution (20:23:58)

Calculating Disp and Stress Results (20:23:58)

Checking Convergence (20:23:58)

Elements Not Converged: 87

Edges Not Converged: 192

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 100.0%

Resource Check (20:23:58)

Elapsed Time (sec): 0.31

CPU Time (sec): 0.14

Memory Usage (kb): 582871

Wrk Dir Dsk Usage (kb): 0

>> Pass 2 <<

Calculating Element Equations (20:23:58)

Total Number of Equations: 648

Maximum Edge Order: 2

Solving Equations (20:23:58)

Post-Processing Solution (20:23:58)

Calculating Disp and Stress Results (20:23:58)

Checking Convergence (20:23:59)

Elements Not Converged: 54

Edges Not Converged: 117

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 96.3%

Resource Check (20:23:59)

Elapsed Time (sec): 0.39

CPU Time (sec): 0.20

Memory Usage (kb): 583927

Wrk Dir Dsk Usage (kb): 0

>> Pass 3 <<

Calculating Element Equations (20:23:59)

Total Number of Equations: 2121

Maximum Edge Order: 4

Solving Equations (20:23:59)

Post-Processing Solution (20:23:59)

Calculating Disp and Stress Results (20:23:59)

Checking Convergence (20:23:59)

Elements Not Converged: 33

Edges Not Converged: 47

Local Disp/Energy Index: 63.8%

Global RMS Stress Index: 24.6%

Resource Check (20:23:59)

Elapsed Time (sec): 0.52

CPU Time (sec): 0.31

Memory Usage (kb): 583927

Wrk Dir Dsk Usage (kb): 1024

>> Pass 4 <<

Calculating Element Equations (20:23:59)

Total Number of Equations: 3582

Maximum Edge Order: 4

Solving Equations (20:23:59)

Post-Processing Solution (20:23:59)

Calculating Disp and Stress Results (20:23:59)

Checking Convergence (20:23:59)

Elements Not Converged: 0

Edges Not Converged: 0

Local Disp/Energy Index: 6.0%

Global RMS Stress Index: 3.3%

RMS Stress Error Estimates:

Load Set	Stress Error	% of Max Prin Str
LoadSet1	1.12e+05	2.2% of 5.12e+06

Resource Check (20:23:59)

Elapsed Time (sec): 0.69

CPU Time (sec): 0.45

Memory Usage (kb): 589431

Wrk Dir Dsk Usage (kb): 3072

The analysis converged to within 10% on

edge displacement, element strain energy,

and global RMS stress.

Total Mass of Model: 1.458175e+00

Total Cost of Model: 0.000000e+00

Mass Moments of Inertia about WCS Origin:

Ixx: 2.76227e+01

Ixy: -4.63760e-08 Iyy: 6.08574e-01

Ixz: -1.87509e-08 Iyz: 8.30680e-09 Izz: 2.72591e+01

Principal MMOI and Principal Axes Relative to WCS Origin:

Max Prin	Mid Prin	Min Prin
2.76227e+01	2.72591e+01	6.08574e-01

WCS X: 1.00000e+00	5.15784e-08	1.71673e-09
WCS Y: -1.71673e-09	3.11693e-10	1.00000e+00
WCS Z: -5.15784e-08	1.00000e+00	-3.11693e-10

Center of Mass Location Relative to WCS Origin:

(-3.17744e-10, -9.51927e-10, 5.00000e-01)

Mass Moments of Inertia about the Center of Mass:

Ixx: 2.72581e+01

Ixy: -4.63760e-08 Iyy: 2.44030e-01

Ixz: -1.89826e-08 Iyz: 7.61276e-09 Izz: 2.72591e+01

Principal MMOI and Principal Axes Relative to COM:

Max Prin	Mid Prin	Min Prin
2.72591e+01	2.72581e+01	2.44030e-01

WCS X: -1.89577e-05	1.00000e+00	1.71674e-09
WCS Y: 2.81829e-10	-1.71673e-09	1.00000e+00
WCS Z: 1.00000e+00	1.89577e-05	-2.81797e-10

Constraint Set: ConstraintSet1: TEST005

Load Set: LoadSet1: TEST005

Resultant Load on Model:

in global X direction: -5.791329e+04

in global Y direction: -8.908331e-07

in global Z direction: 1.008107e-06

Measures:

Name	Value	Convergence
max_beam_bending:	0.000000e+00	0.0%
max_beam_tensile:	0.000000e+00	0.0%
max_beam_torsion:	0.000000e+00	0.0%
max_beam_total:	0.000000e+00	0.0%
max_disp_mag:	1.955901e-01	0.4%
max_disp_x:	-1.955901e-01	0.4%
max_disp_y:	9.563459e-03	0.2%
max_disp_z:	-1.870352e-04	0.5%
max_prin_mag:	5.115037e+06	0.5%
max_rot_mag:	0.000000e+00	0.0%
max_rot_x:	0.000000e+00	0.0%
max_rot_y:	0.000000e+00	0.0%
max_rot_z:	0.000000e+00	0.0%

max_stress_prin: 5.115037e+06 0.5%

max_stress_vm: 6.948304e+06 6.9%

max_stress_xx: -2.322589e+06 8.2%

max_stress_xy: -3.878826e+06 7.1%

max_stress_xz: -9.674206e+05 2.9%

max_stress_yy: 4.808287e+06 6.8%

max_stress_yz: 6.263410e+05 18.2%

max_stress_zz: -1.398694e+06 1.9%

min_stress_prin: -5.037829e+06 0.9%

strain_energy: 5.386562e+03 0.4%

Analysis "Analysis1" Completed (20:23:59)

Memory and Disk Usage:

Machine Type: Windows 7 64 Service Pack 1

RAM Allocation for Solver (megabytes): 512.0

Total Elapsed Time (seconds): 0.72

Total CPU Time (seconds): 0.47

Maximum Memory Usage (kilobytes): 589431

Working Directory Disk Usage (kilobytes): 3072

Results Directory Size (kilobytes):

1762 .\Analysis1

Maximum Data Base Working File Sizes (kilobytes):

3072 .\Analysis1.tmp\kel1.bas

Run Completed

Tue Dec 13, 2016   20:23:59

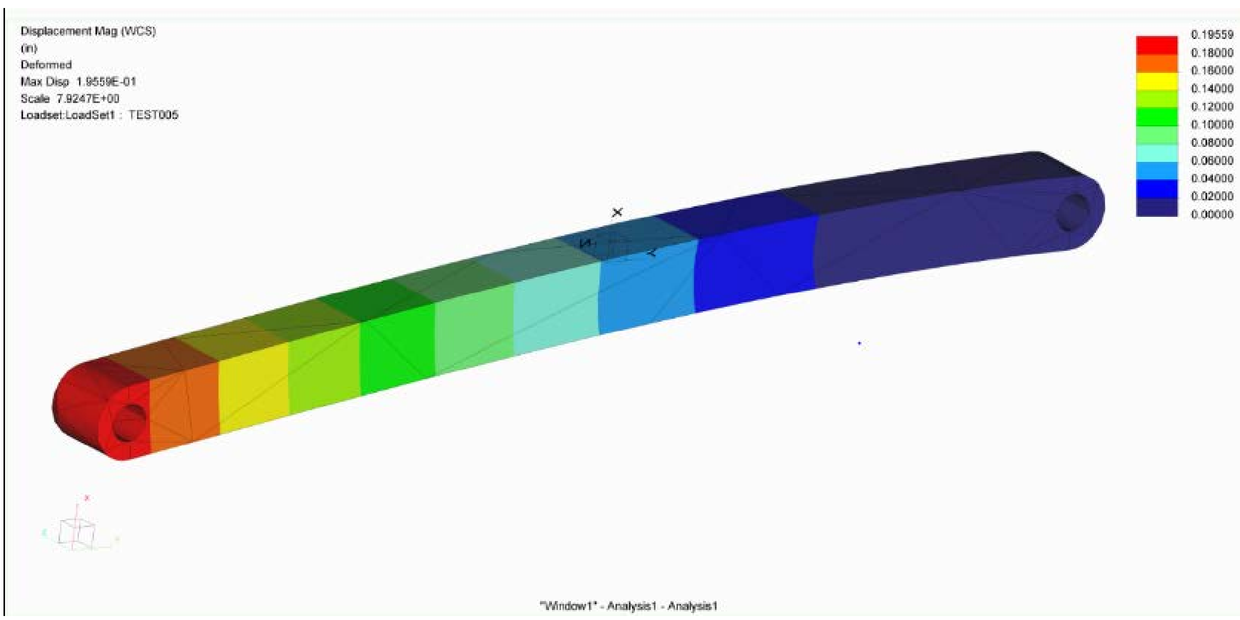


Figure B1-1. SUS005 Fringe Plot of Displacement

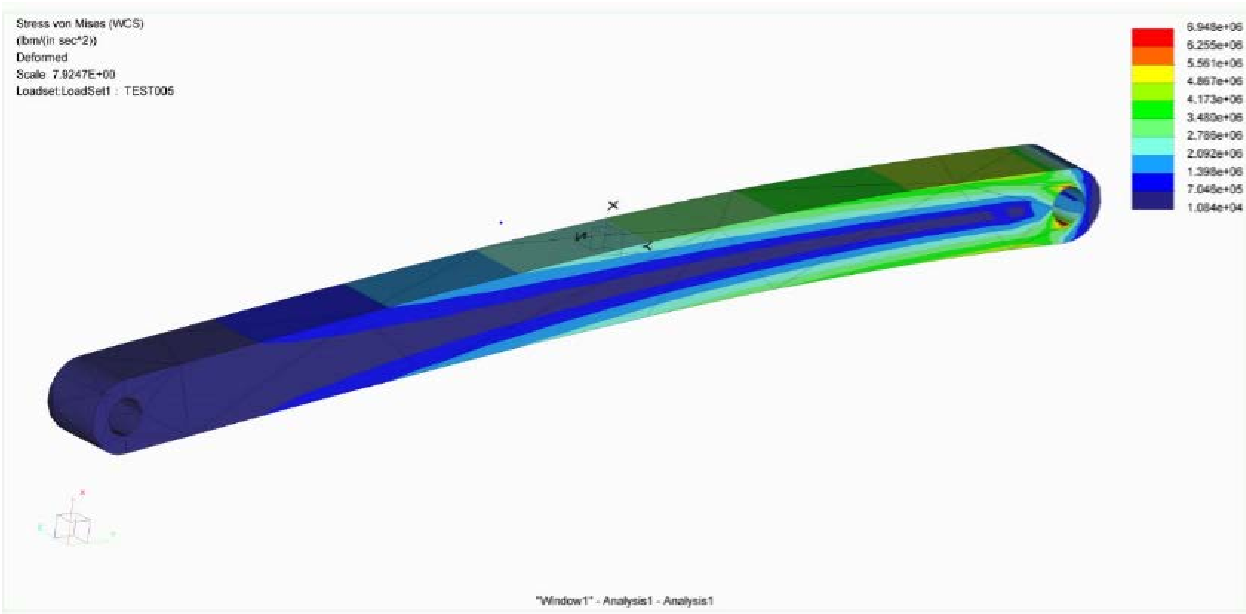


Figure B1-2. SUS005 Fringe Plot of Max Stress Von Mises

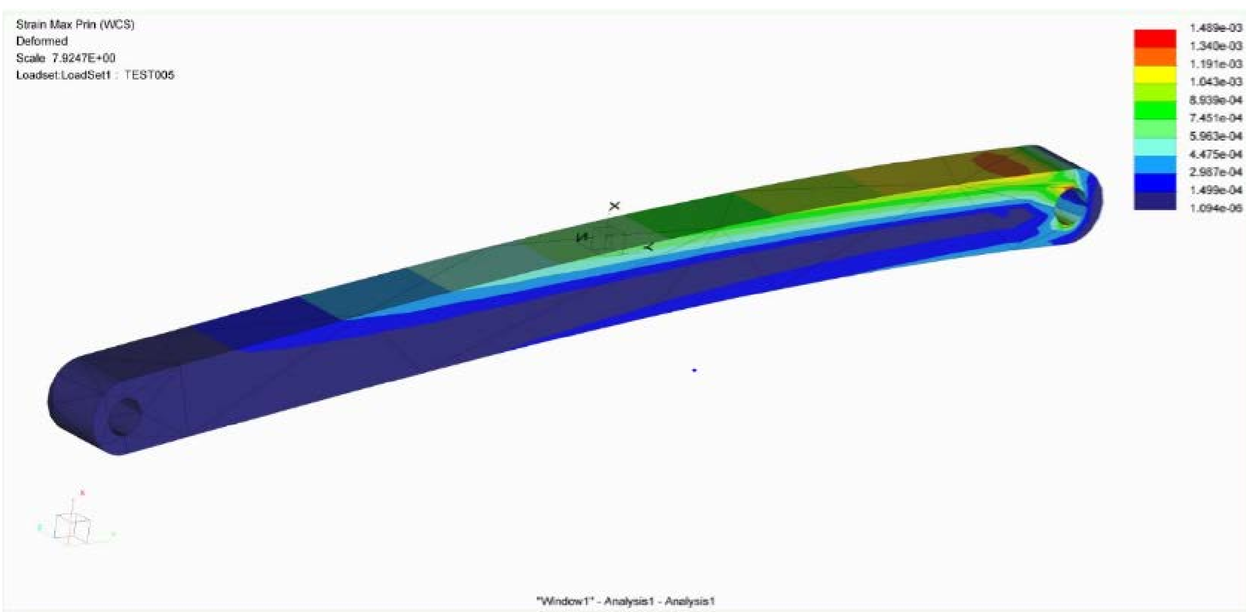


Figure B1-3. SUS005 Fringe Plot of Max Principal Strain

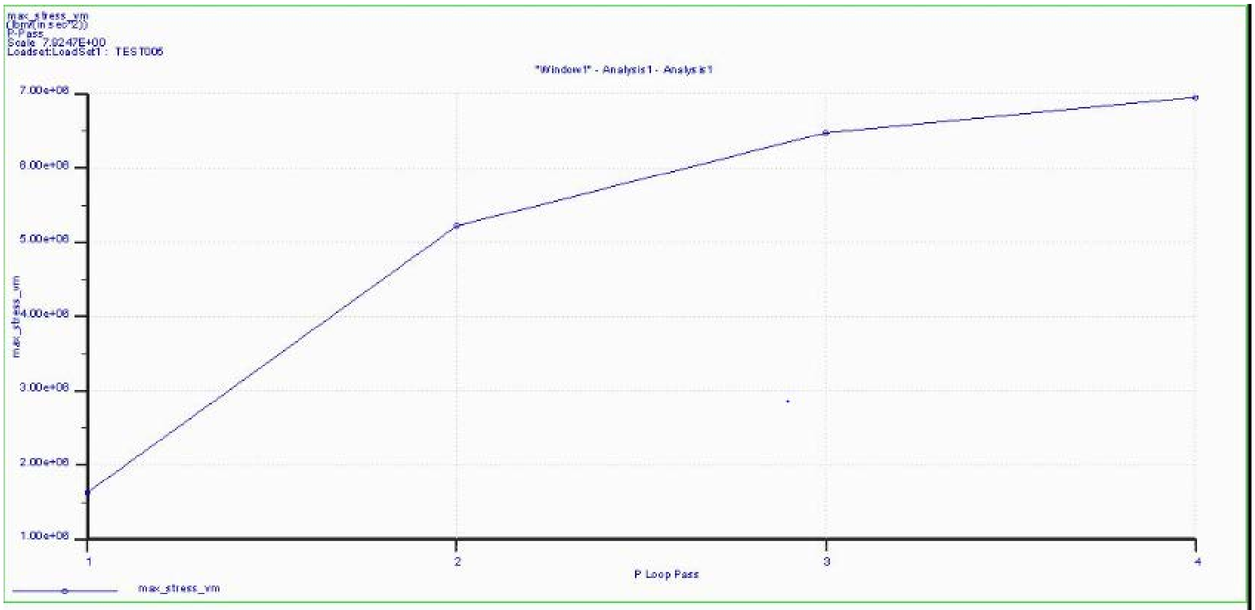


Figure B1-4. SUS005 Convergence shown on the Max Stress Von Mises graph

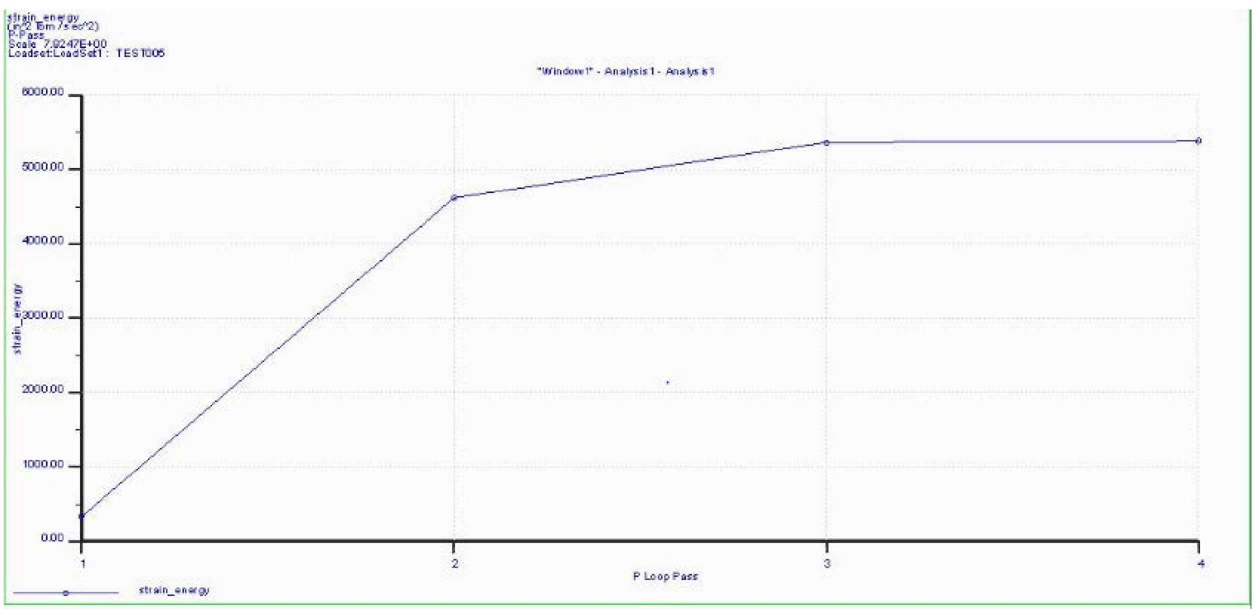


Figure B1-5. SUS005 Convergence shown on the Strain Energy graph

SUS012 FEA Results

Creo Simulate Structure Version P-10-38:spg

Summary for Design Study "Analysis1"

Tue Dec 13, 2016 19:28:47

Run Settings

Memory allocation for block solver: 512.0

Parallel Processing Status

Parallel task limit for current run: 8

Parallel task limit for current platform: 64

Number of processors detected automatically: 8

Checking the model before creating elements...

These checks take into account the fact that AutoGEM will

automatically create elements in volumes with material

properties, on surfaces with shell properties, and on curves

with beam section properties.

Generate elements automatically.

Checking the model after creating elements...

No errors were found in the model.

Creo Simulate Structure Model Summary

Principal System of Units: Inch lbm Second (Pro/E Default)

Length: in

Mass: lbm

Time: sec

Temperature: F

Model Type: Three Dimensional

Points: 29

Edges: 107

Faces: 134

Springs: 0

Masses: 0

Beams: 0

Shells: 0

Solids: 55

Elements: 55

Standard Design Study

Static Analysis "Analysis1":

Convergence Method: Multiple-Pass Adaptive

Plotting Grid: 4

Convergence Loop Log: (19:28:48)

>> Pass 1 <<

Calculating Element Equations (19:28:48)

Total Number of Equations: 51

Maximum Edge Order: 1

Solving Equations (19:28:48)

Post-Processing Solution (19:28:48)

Calculating Disp and Stress Results (19:28:48)

Checking Convergence (19:28:48)

Elements Not Converged: 55

Edges Not Converged: 107

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 100.0%

Resource Check (19:28:48)

Elapsed Time (sec): 1.81

CPU Time (sec): 0.45

Memory Usage (kb): 581957

Wrk Dir Dsk Usage (kb): 0

>> Pass 2 <<

Calculating Element Equations (19:28:48)

Total Number of Equations: 312

Maximum Edge Order: 2

Solving Equations (19:28:49)

Post-Processing Solution (19:28:49)

Calculating Disp and Stress Results (19:28:49)

Checking Convergence (19:28:49)

Elements Not Converged: 55

Edges Not Converged: 87

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 97.1%

Resource Check (19:28:49)

Elapsed Time (sec): 1.89

CPU Time (sec): 0.48

Memory Usage (kb): 583013

Wrk Dir Dsk Usage (kb): 0

>> Pass 3 <<

Calculating Element Equations (19:28:49)

Total Number of Equations: 972

Maximum Edge Order: 4

Solving Equations (19:28:49)

Post-Processing Solution (19:28:49)

Calculating Disp and Stress Results (19:28:49)

Checking Convergence (19:28:49)

Elements Not Converged: 52

Edges Not Converged: 85

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 82.4%

Resource Check (19:28:49)

Elapsed Time (sec): 1.99

CPU Time (sec): 0.56

Memory Usage (kb): 583013

Wrk Dir Dsk Usage (kb): 0

>> Pass 4 <<

Calculating Element Equations (19:28:49)

Total Number of Equations: 2157

Maximum Edge Order: 5

Solving Equations (19:28:49)

Post-Processing Solution (19:28:49)

Calculating Disp and Stress Results (19:28:49)

Checking Convergence (19:28:49)

Elements Not Converged: 28

Edges Not Converged: 42

Local Disp/Energy Index: 29.4%

Global RMS Stress Index: 16.2%

Resource Check (19:28:49)

Elapsed Time (sec): 2.11

CPU Time (sec): 0.69

Memory Usage (kb): 585829

Wrk Dir Dsk Usage (kb): 2048

>> Pass 5 <<

Calculating Element Equations (19:28:49)

Total Number of Equations: 3972

Maximum Edge Order: 5

Solving Equations (19:28:49)

Post-Processing Solution (19:28:49)

Calculating Disp and Stress Results (19:28:49)

Checking Convergence (19:28:49)

Elements Not Converged: 0

Edges Not Converged: 0

Local Disp/Energy Index: 2.3%

Global RMS Stress Index: 2.8%

RMS Stress Error Estimates:

Load Set	Stress Error	% of Max Prin Str
LoadSet1	2.49e+04	0.7% of 3.61e+06

Resource Check (19:28:49)

Elapsed Time (sec): 2.37

CPU Time (sec): 0.95

Memory Usage (kb): 591589

Wrk Dir Dsk Usage (kb): 5120

The analysis converged to within 5% on

edge displacement, element strain energy,

and global RMS stress.

Total Mass of Model: 3.886541e-01

Total Cost of Model: 0.000000e+00

Mass Moments of Inertia about WCS Origin:

Ixx: 6.35409e+00

Ixy: 4.18814e-10 Iyy: 6.35409e+00

Ixz: 1.69568e-07 Iyz: -2.22912e-08 Izz: 1.21454e-02

Principal MMOI and Principal Axes Relative to WCS Origin:

Max Prin	Mid Prin	Min Prin
6.35409e+00	6.35409e+00	1.21454e-02

WCS X: 1.00000e+00 9.39794e-17 -2.67376e-08

WCS Y: 0.00000e+00 1.00000e+00 3.51488e-09

WCS Z: 2.67376e-08 -3.51488e-09 1.00000e+00

Center of Mass Location Relative to WCS Origin:

(-1.45508e-07, 3.93674e-08, 3.50000e+00)

Mass Moments of Inertia about the Center of Mass:

Ixx: 1.59308e+00

Ixy: 4.18811e-10 Iyy: 1.59308e+00

Ixz: -2.83653e-08 Iyz: 3.12599e-08 Izz: 1.21454e-02

Principal MMOI and Principal Axes Relative to COM:

Max Prin	Mid Prin	Min Prin
1.59308e+00	1.59308e+00	1.21454e-02

WCS X: 8.99258e-01 -4.37418e-01 1.79422e-08

WCS Y: 4.37418e-01 8.99258e-01 -1.97731e-08

WCS Z: -7.48553e-09 2.56293e-08 1.00000e+00

Constraint Set: ConstraintSet1: SUS012HW

Load Set: LoadSet1: SUS012HW

Resultant Load on Model:

in global X direction: 1.461103e-09

in global Y direction: -5.791329e+04

in global Z direction: -2.772140e-09

Measures:

Name	Value	Convergence
max_beam_bending:	0.000000e+00	0.0%
max_beam_tensile:	0.000000e+00	0.0%
max_beam_torsion:	0.000000e+00	0.0%
max_beam_total:	0.000000e+00	0.0%
max_disp_mag:	1.562043e-03	0.4%
max_disp_x:	6.512440e-06	100.0%
max_disp_y:	-1.562043e-03	0.4%
max_disp_z:	-1.650413e-04	0.3%
max_prin_mag:	-3.611341e+06	8.7%
max_rot_mag:	0.000000e+00	0.0%
max_rot_x:	0.000000e+00	0.0%
max_rot_y:	0.000000e+00	0.0%
max_rot_z:	0.000000e+00	0.0%
max_stress_prin:	3.506821e+06	6.0%

max_stress_vm: 2.498759e+06 5.9%

max_stress_xx: 1.252658e+06 2.1%

max_stress_xy: 1.635183e+05 7.4%

max_stress_xz: -4.851552e+05 13.8%

max_stress_yy: 1.261447e+06 2.7%

max_stress_yz: 6.877143e+05 25.2%

max_stress_zz: -3.387230e+06 6.5%

min_stress_prin: -3.611341e+06 11.4%

strain_energy: 2.427671e+01 0.5%

Analysis "Analysis1" Completed (19:28:49)

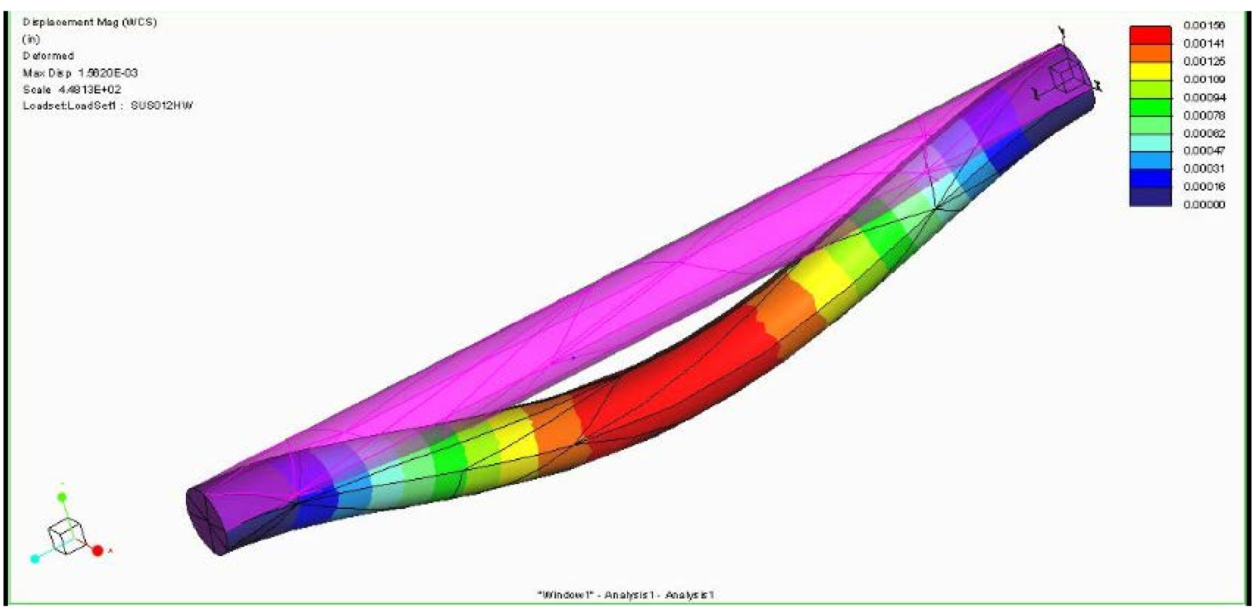


Figure B1-6. SUS012 FEA Fringe Plot of Displacement

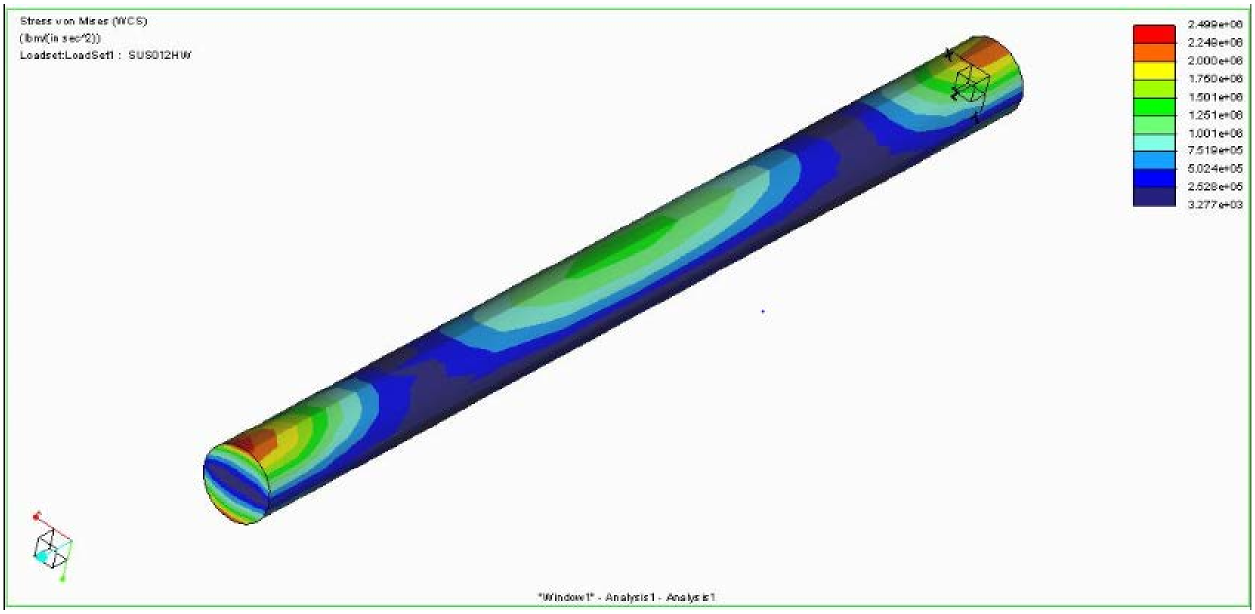


Figure B1-7. SUS012 Fringe Plot of Max Stress Von Mises

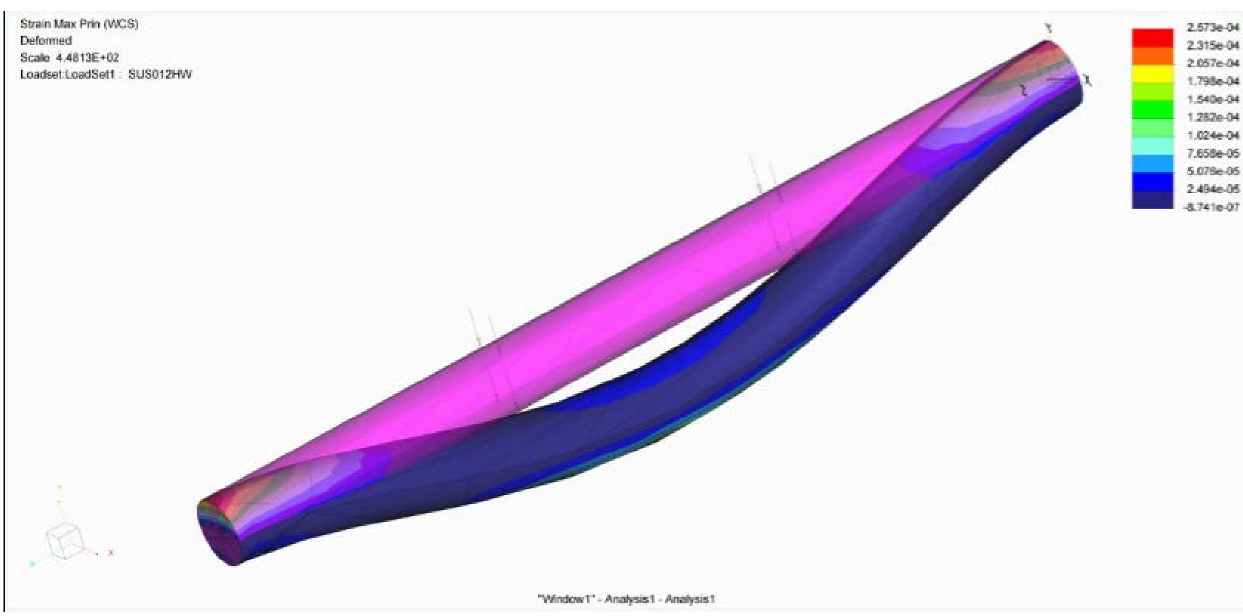


Figure B1-8. SUS012 Fringe Plot of Max Principal Strain

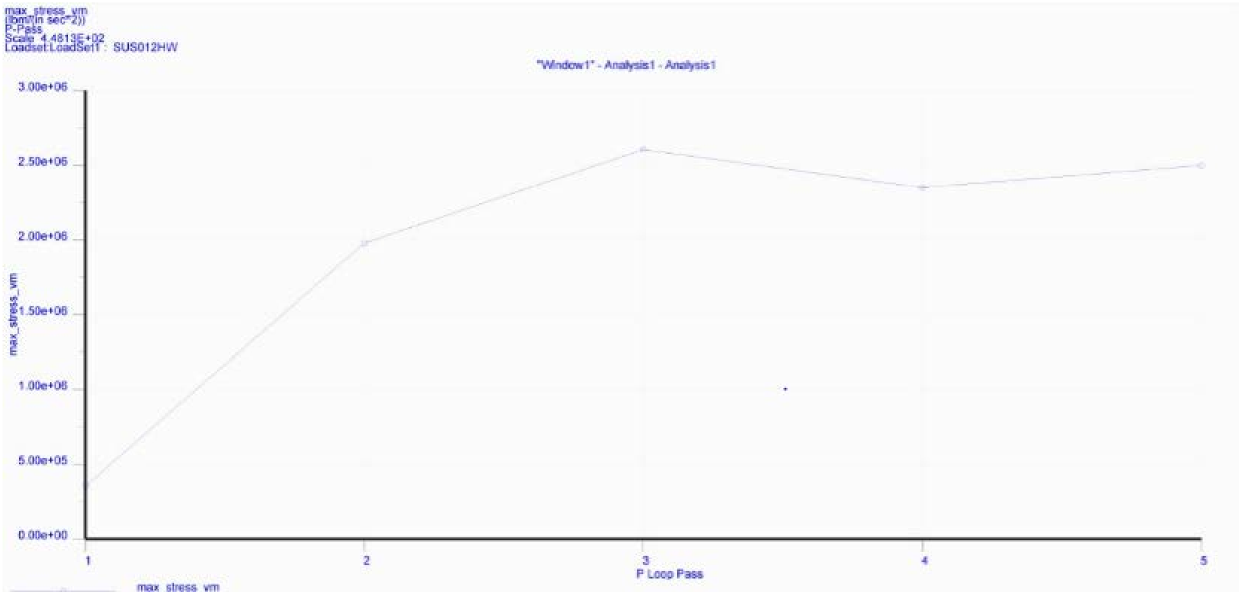


Figure B1-9. SUS012 Convergence shown on the Max Stress Von Mises graph

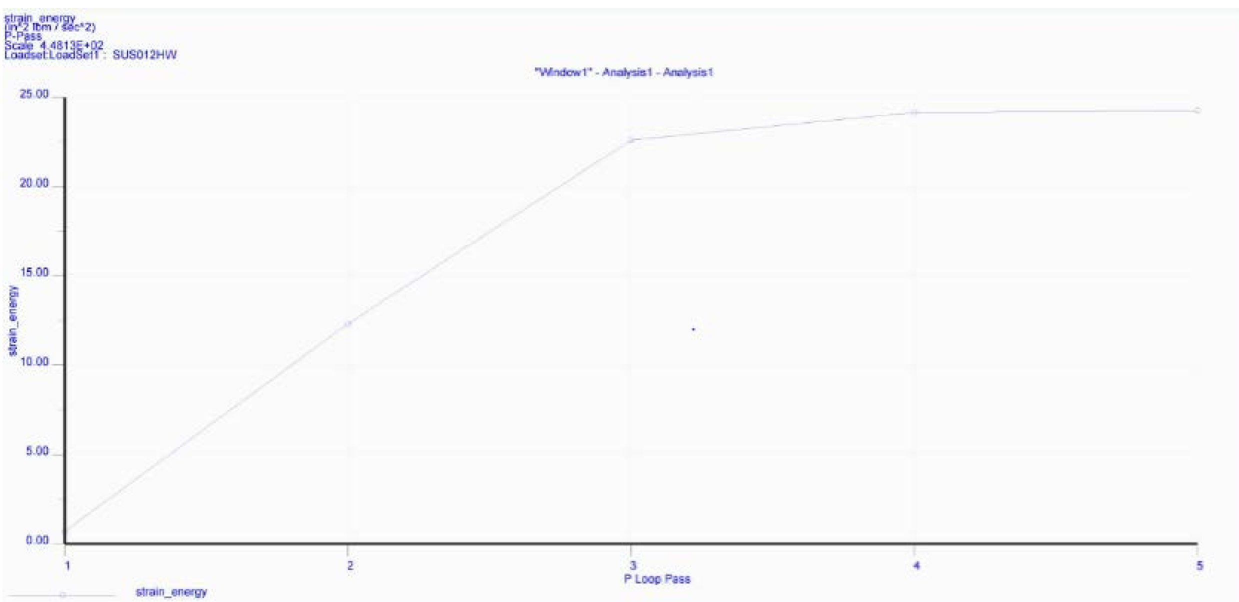


Figure B1-10. SUS012 Convergence shown on the Strain Energy graph

SUS025 FEA Results

Creo Simulate Structure Version P-10-38:spg

Summary for Design Study "Analysis1"

Tue Dec 13, 2016   20:46:45

Run Settings

  Memory allocation for block solver: 512.0

Parallel Processing Status

  Parallel task limit for current run: 8

  Parallel task limit for current platform: 64

  Number of processors detected automatically: 8

Checking the model before creating elements...

These checks take into account the fact that AutoGEM will

automatically create elements in volumes with material

properties, on surfaces with shell properties, and on curves

with beam section properties.

  Generate elements automatically.

Checking the model after creating elements...

No errors were found in the model.

Creo Simulate Structure Model Summary

Principal System of Units: Inch lbm Second (Pro/E Default)

Length: in

Mass: lbm

Time: sec

Temperature: F

Model Type: Three Dimensional

Points: 1217

Edges: 5951

Faces: 8274

Springs: 0

Masses: 0

Beams: 0

Shells: 0

Solids: 3544

Elements: 3544

Standard Design Study

Static Analysis "Analysis1":

Convergence Method: Multiple-Pass Adaptive

Plotting Grid: 4

Convergence Loop Log: (20:46:46)

>> Pass 1 <<

Calculating Element Equations (20:46:46)

Total Number of Equations: 3558

Maximum Edge Order: 1

Solving Equations (20:46:46)

Post-Processing Solution (20:46:46)

Calculating Disp and Stress Results (20:46:46)

Checking Convergence (20:46:48)

Elements Not Converged: 3544

Edges Not Converged: 5951

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 100.0%

Resource Check (20:46:48)

Elapsed Time (sec): 2.58

CPU Time (sec): 2.03

Memory Usage (kb): 644221

Wrk Dir Dsk Usage (kb): 26624

>> Pass 2 <<

Calculating Element Equations (20:46:48)

Total Number of Equations: 21225

Maximum Edge Order: 2

Solving Equations (20:46:48)

Post-Processing Solution (20:46:48)

Calculating Disp and Stress Results (20:46:49)

Checking Convergence (20:46:50)

Elements Not Converged: 2903

Edges Not Converged: 4426

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 82.9%

Resource Check (20:46:50)

Elapsed Time (sec): 4.96

CPU Time (sec): 3.78

Memory Usage (kb): 659406

Wrk Dir Dsk Usage (kb): 26624

>> Pass 3 <<

Calculating Element Equations (20:46:50)

Total Number of Equations: 82503

Maximum Edge Order: 4

Solving Equations (20:46:51)

Post-Processing Solution (20:46:52)

Calculating Disp and Stress Results (20:46:53)

Checking Convergence (20:46:55)

Elements Not Converged: 304

Edges Not Converged: 0

Local Disp/Energy Index: 100.0%

Global RMS Stress Index: 7.4%

Resource Check (20:46:55)

Elapsed Time (sec): 10.13

CPU Time (sec): 10.39

Memory Usage (kb): 685641

Wrk Dir Dsk Usage (kb): 92160

>> Pass 4 <<

Calculating Element Equations (20:46:55)

Total Number of Equations: 147090

Maximum Edge Order: 5

Solving Equations (20:46:57)

Post-Processing Solution (20:47:00)

Calculating Disp and Stress Results (20:47:01)

Checking Convergence (20:47:03)

Elements Not Converged: 0

Edges Not Converged: 0

Local Disp/Energy Index: 7.7%

Global RMS Stress Index: 1.4%

RMS Stress Error Estimates:

Load Set	Stress Error	% of Max Prin Str
LoadSet1	8.72e+04	0.9% of 9.67e+06

Resource Check (20:47:04)

Elapsed Time (sec): 18.97

CPU Time (sec): 27.25

Memory Usage (kb): 720640

Wrk Dir Dsk Usage (kb): 194560

The analysis converged to within 10% on

edge displacement, element strain energy,

and global RMS stress.

Total Mass of Model: 6.762335e-01

Total Cost of Model: 0.000000e+00

Mass Moments of Inertia about WCS Origin:

Ixx: 1.93814e-01

Ixy: 1.44624e-10 Iyy: 2.59613e+01

Ixz: -1.03904e-08 Iyz: -1.64644e-09 Izz: 2.59625e+01

Principal MMOI and Principal Axes Relative to WCS Origin:

Max Prin	Mid Prin	Min Prin
2.59625e+01	2.59613e+01	1.93814e-01

WCS X: -4.03219e-10	-5.53333e-16	1.00000e+00
WCS Y: -1.37229e-06	1.00000e+00	0.00000e+00
WCS Z: 1.00000e+00	1.37229e-06	4.03219e-10

Center of Mass Location Relative to WCS Origin:

(-5.35568e+00, -6.24826e-10, -1.16714e-09)

Mass Moments of Inertia about the Center of Mass:

Ixx: 1.93814e-01

Ixy: 2.40755e-09 Iyy: 6.56474e+00

Ixz: -6.16342e-09 Iyz: -1.64644e-09 Izz: 6.56594e+00

Principal MMOI and Principal Axes Relative to COM:

Max Prin	Mid Prin	Min Prin
6.56594e+00	6.56474e+00	1.93814e-01

WCS X: -9.67248e-10	3.77895e-10	1.00000e+00
WCS Y: -1.37229e-06	1.00000e+00	-3.77896e-10
WCS Z: 1.00000e+00	1.37229e-06	9.67247e-10

Constraint Set: ConstraintSet1: SUS025

Load Set: LoadSet1: SUS025

Resultant Load on Model:

in global X direction: 4.827701e-08

in global Y direction: -2.895664e+04

in global Z direction: -1.054563e-07

Measures:

Name	Value	Convergence
max_beam_bending:	0.000000e+00	0.0%
max_beam_tensile:	0.000000e+00	0.0%
max_beam_torsion:	0.000000e+00	0.0%
max_beam_total:	0.000000e+00	0.0%

max_disp_mag: 3.109556e-02 0.1%

max_disp_x: -2.055653e-03 0.1%

max_disp_y: -3.104572e-02 0.1%

max_disp_z: -2.103593e-04 0.0%

max_prin_mag: 9.671454e+06 13.5%

max_rot_mag: 0.000000e+00 0.0%

max_rot_x: 0.000000e+00 0.0%

max_rot_y: 0.000000e+00 0.0%

max_rot_z: 0.000000e+00 0.0%

max_stress_prin: 9.671454e+06 13.5%

max_stress_vm: 6.754364e+06 16.7%

max_stress_xx: 8.879426e+06 13.3%

max_stress_xy: 2.021837e+06 15.4%

max_stress_xz: 1.686673e+06 11.5%

max_stress_yy: 3.444803e+06 8.2%

max_stress_yz: -1.032757e+06 1.9%

max_stress_zz: 3.275423e+06 8.3%

min_stress_prin: -8.502188e+06 9.2%

strain_energy: 4.045012e+02 0.1%

Analysis "Analysis1" Completed (20:47:04)

Memory and Disk Usage:

Machine Type: Windows 7 64 Service Pack 1

RAM Allocation for Solver (megabytes): 512.0

Total Elapsed Time (seconds): 19.11

Total CPU Time (seconds): 27.39

Maximum Memory Usage (kilobytes): 720640

Working Directory Disk Usage (kilobytes): 194560

Results Directory Size (kilobytes):

46693 .\Analysis1

Maximum Data Base Working File Sizes (kilobytes):

167936 .\Analysis1.tmp\kel1.bas

26624 .\Analysis1.tmp\oel1.bas

Run Completed

Tue Dec 13, 2016 20:47:04

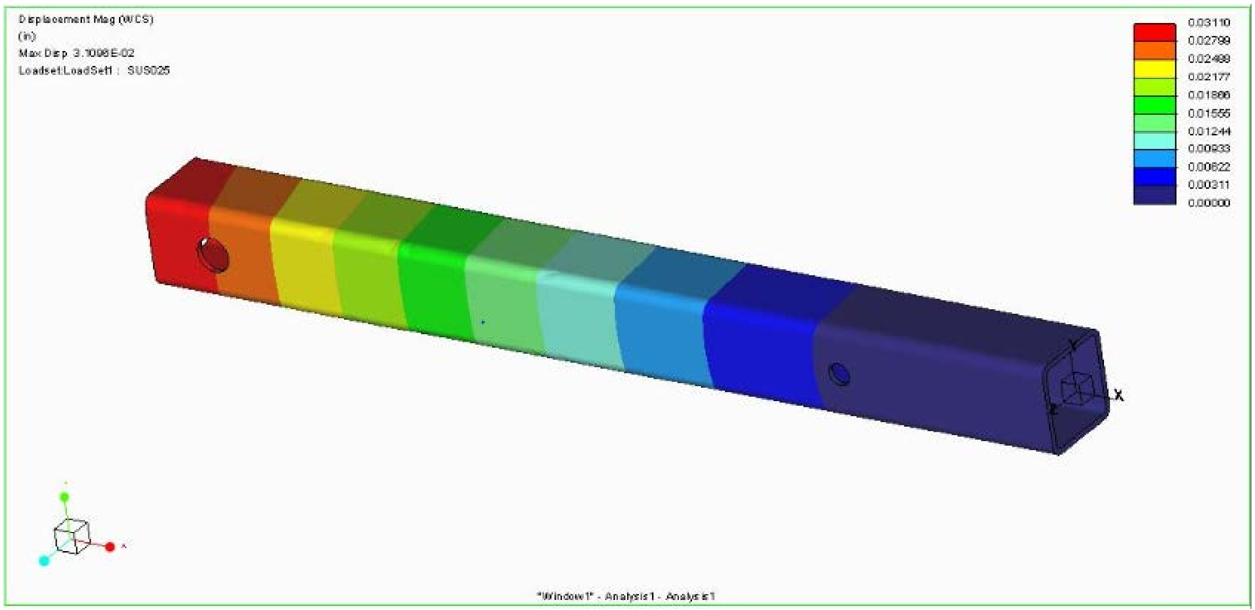


Figure B1-11. SUS025 FEA Fringe Plot of Displacement

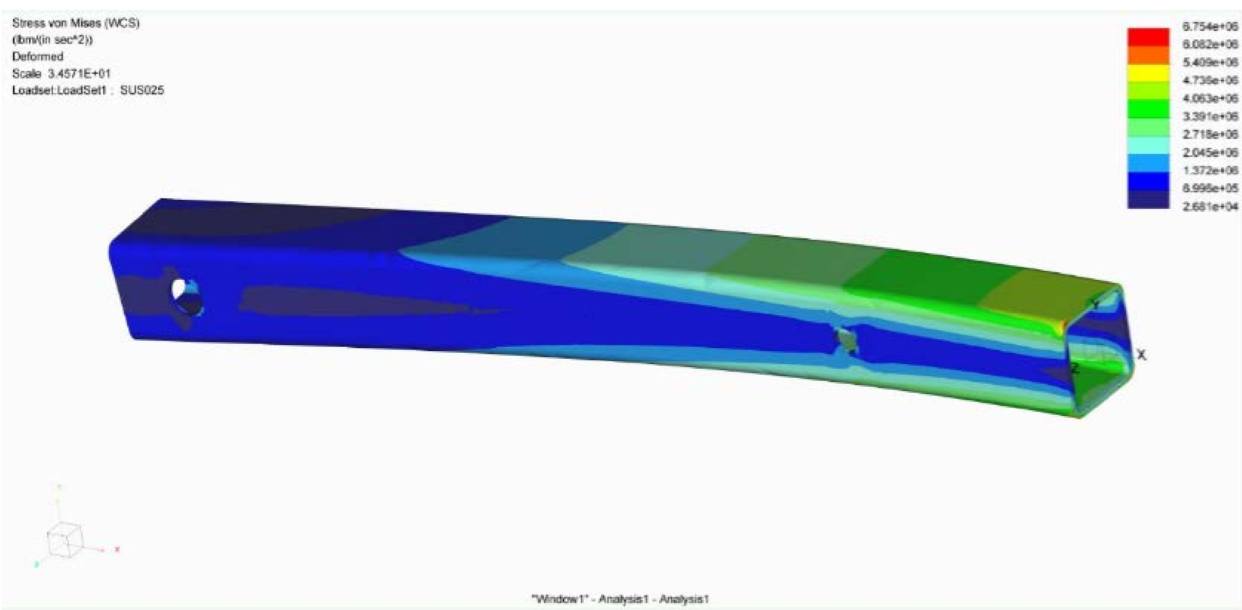


Figure B1-12. SUS025 Fringe Plot of Max Stress Von Mises

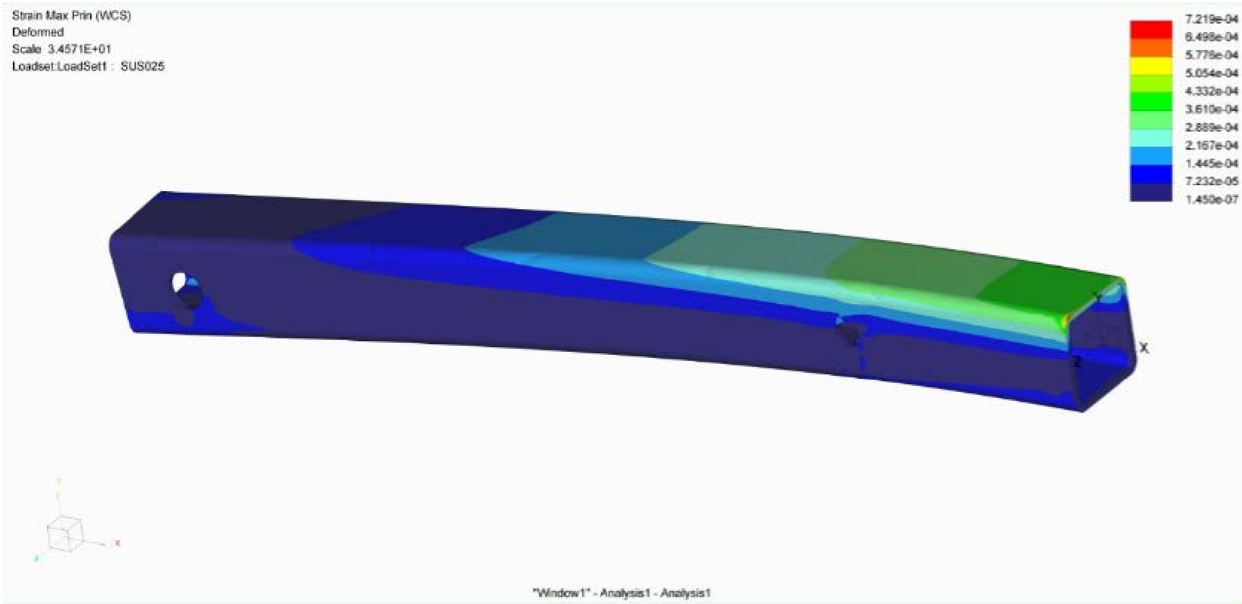


Figure B1-13. SUS025 Fringe Plot of Max Principal Strain

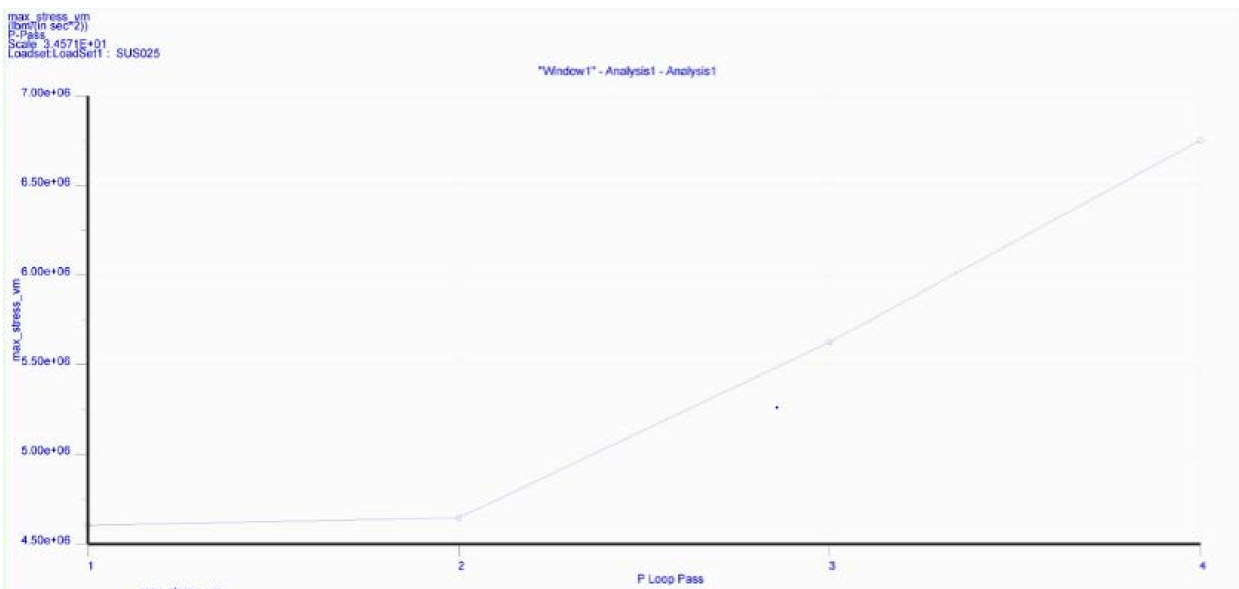


Figure B1-14. SUS025 Convergence shown on the Max Stress Von Mises graph

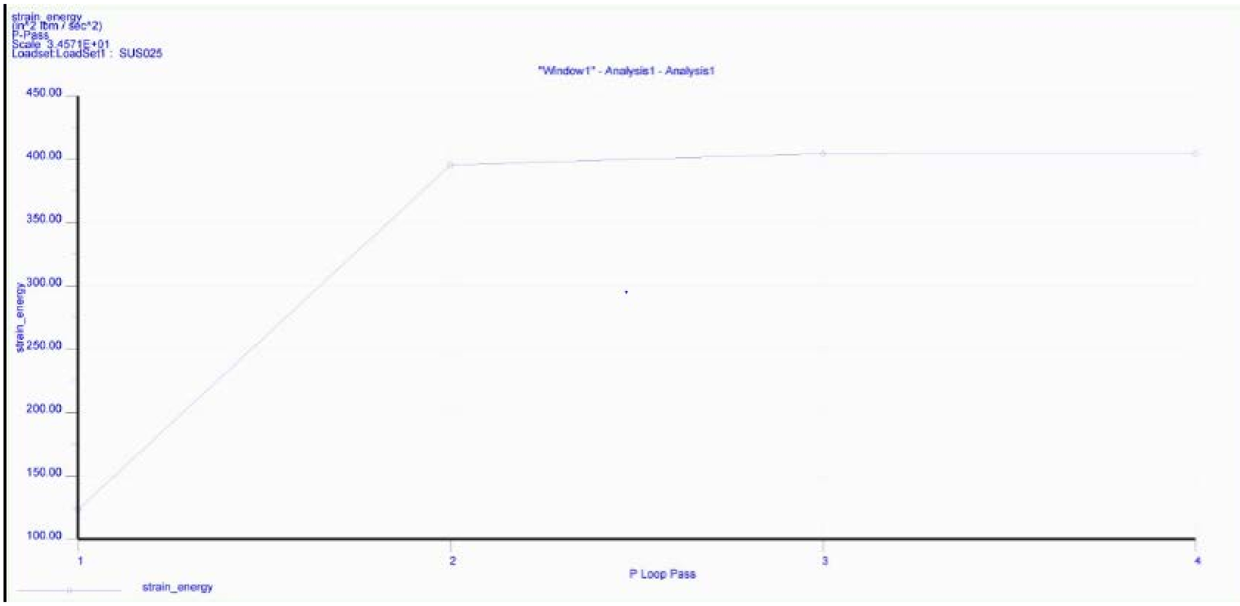


Figure B1-15. SUS025 Convergence shown on the Strain Energy graph

Appendix B1-6

Bill of Materials

Part Description	Part number	Price	Lead Time of service	Quantity	Why it's Important?	Item in the shop?	Sponsored item?	Date needed
Channel A36		$4.32	Not necessary	5	Needed for Construction	No	No	12/31
3/8 X 1-1/2 HR FLAT BAR A36		$1.60	Not necessary	20	Needed for Construction	No	No	12/31
MISC STEEL BLOCKS		$1.00	Not necessary	6.5	Needed for Construction	No	No	12/31
1/2 in. Bore Zinc-Plated Mild Steel Set Screw Collar	203025022	$1.32	Not necessary	40	Needed for Construction	No	No	12/31
1" Square Tube 30' long		$14.70	Not necessary	1	Needed for Construction	No	No	1/1
3/8-16 Grade 8 Bolt (10 pack)	91257A638	$8.45	Not necessary	1	Needed for Construction	No	Yes	1/1
Linear Actuator	PA-18	$185.99	Not necessary	1	Needed for Leveling	No	Yes	1/30
1/2 20 Bolt (10 pack)	91247A358	$10.15	Not necessary	2	Needed for Construction	No	Yes	1/1
Groschopp Gear Motor	PM8018-PL7300	$406.20	Not necessary	1	Needed for Construction	No	Yes	1/2
Alum Metal plate (48X24X1/4 in) (gorilla metals)		$211.92	Not necessary	1	Needed for Suspension Frame	No	Yes	1/1
1/4-20 Bolt (50 pack)	91257A546	$7.60	Not necessary	1	Needed for Construction	No	Yes	1/1
KOYO THRUST NEEDLE ROLLER BEARING ID 1/2in	NTA-815	$2.56	Not necessary	18	Ensures smooth operation of Tilt	No	No	1/1
Steel, 1/2" Diameter, 10 ft Long (sims metal)		$10.10	Not necessary	1	Needed for Construction	No	No	1/1
WaterJet Cut Time (min)		$3.00	Not necessary	12.31	all metal parts will be fabricated	No	No	1/1
1/4 in washer grade 8 (100 pack)	98023A029	$6.36	Not necessary	1	Needed for Construction	No	Yes	1/1
3/8 in washer grade 8 (50 pack)	98023A031	$5.72	Not necessary	1	Needed for Construction	No	Yes	1/1
1/2 in washer grade 8 (25 pack)	98023A033	$6.00	Not necessary	1	Needed for Construction	No	Yes	1/1
1/4 in nut grade 8 (100 pack)	94895A029	$3.22	Not necessary	1	Needed for Construction	No	Yes	1/1
3/8 in nut grade 8 (100 pack)	94895A031	$7.46	Not necessary	1	Needed for Construction	No	Yes	1/1
1/2 in nut grade 8 (50 pack)	94895A825	$8.20	Not necessary	1	Needed for Construction	No	Yes	1/1
Total		$1,106.58						

Figure B1-16. Bill of Materials

Bogie, Steering, and Failsafe

Appendix B2-1

Engineering Drawings

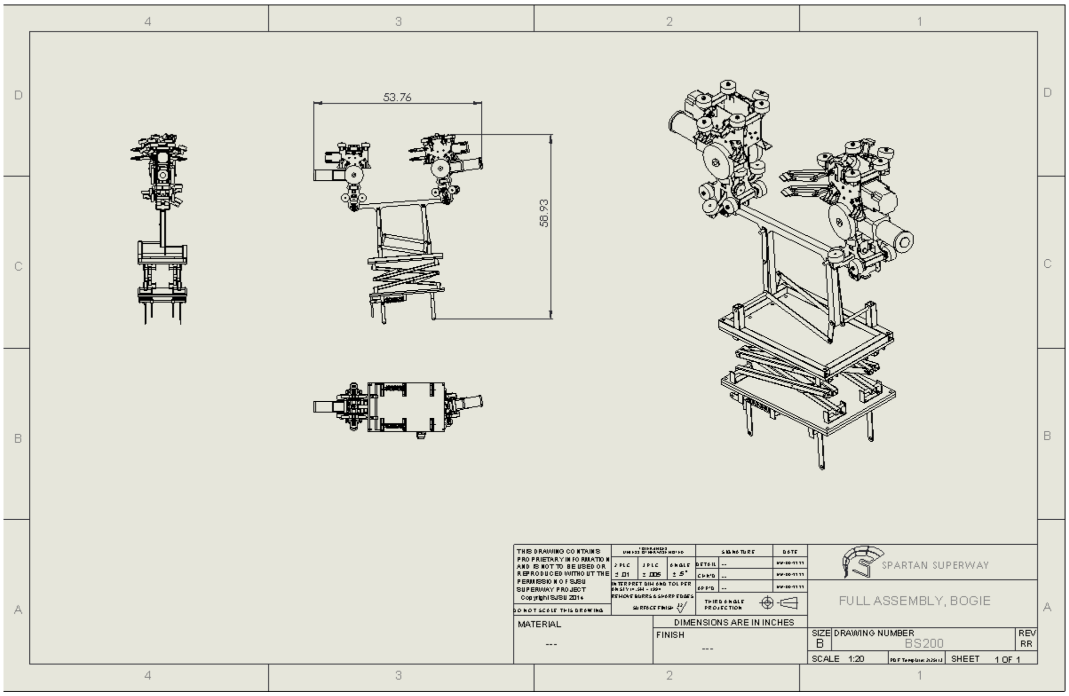


Figure B2-1. Full Assembly drawing of bogie design with Suspension

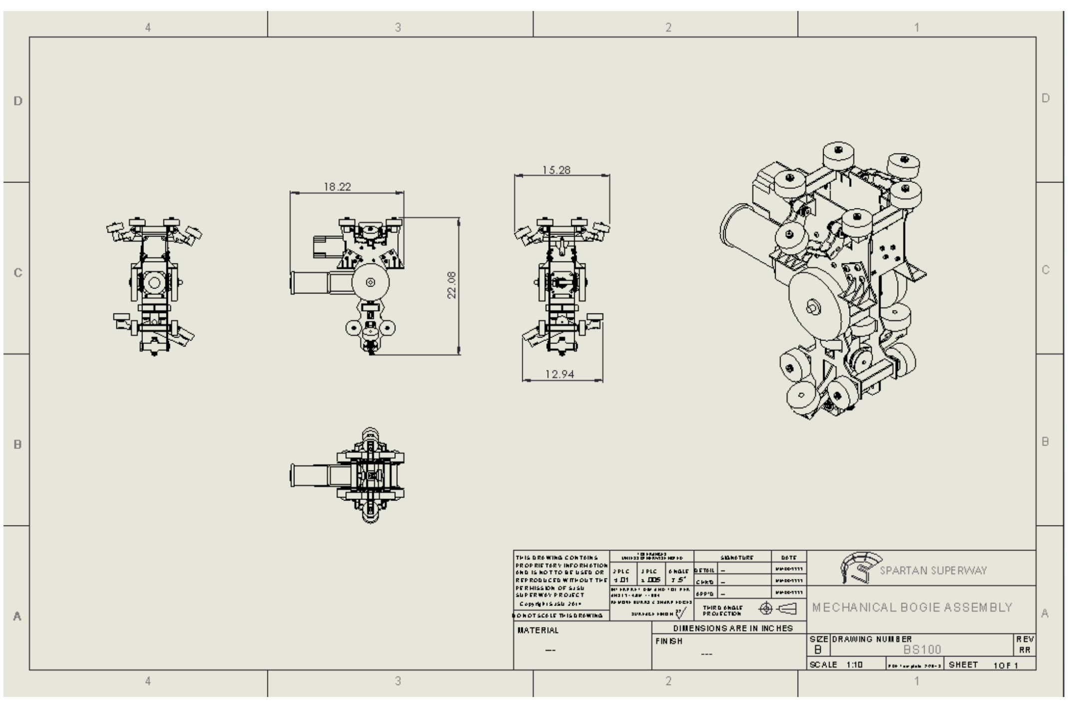


Figure B2-2. Individual Mechanical Bogie

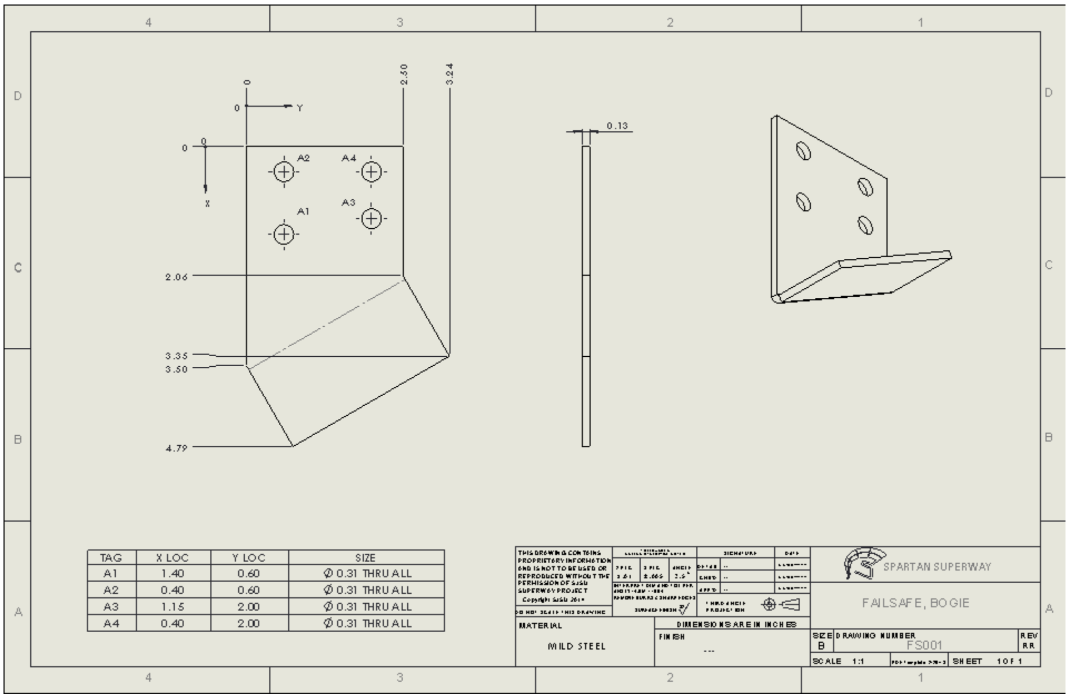


Figure B2-3. Bogie Failsafe

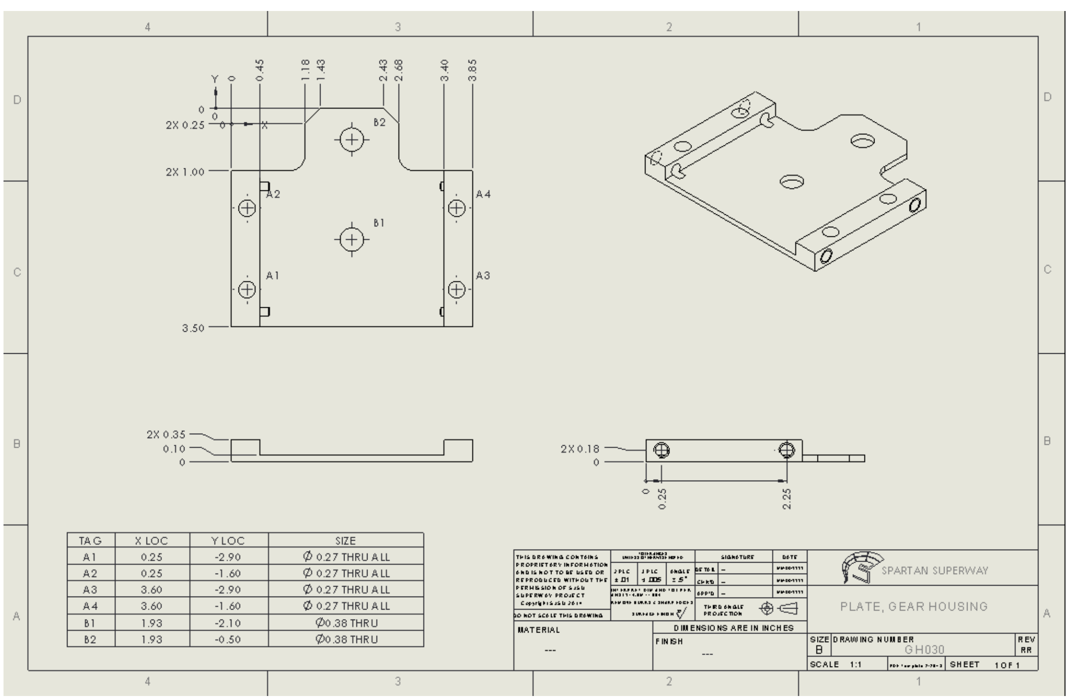


Figure B2-4. Gear housing plate

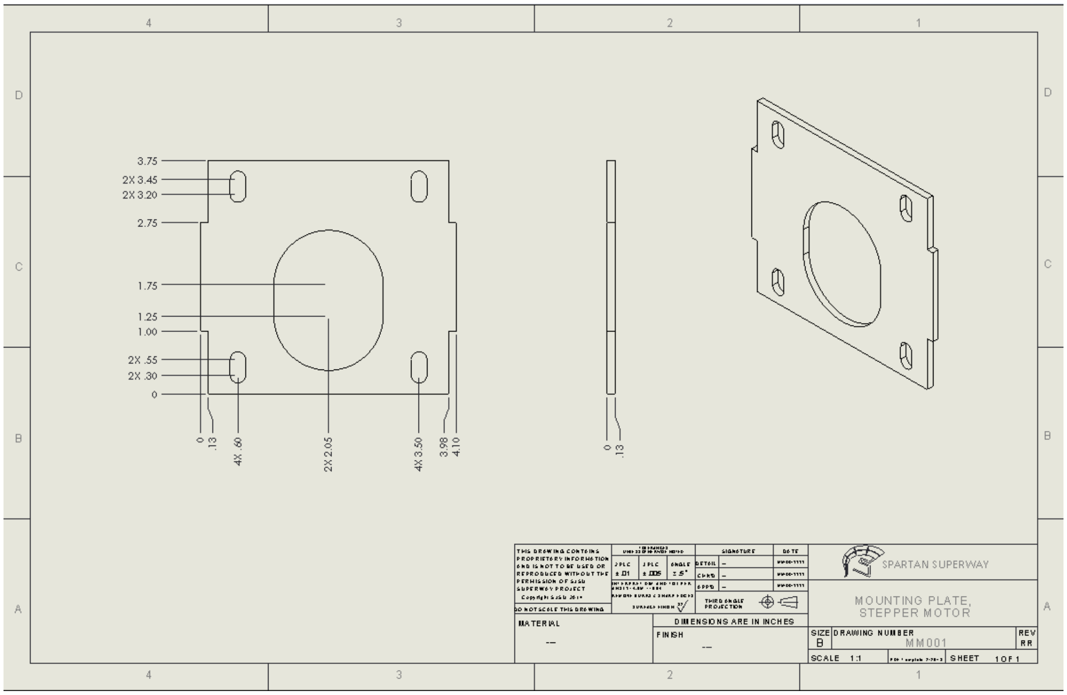


Figure B2-5. Stepper motor mounting plate

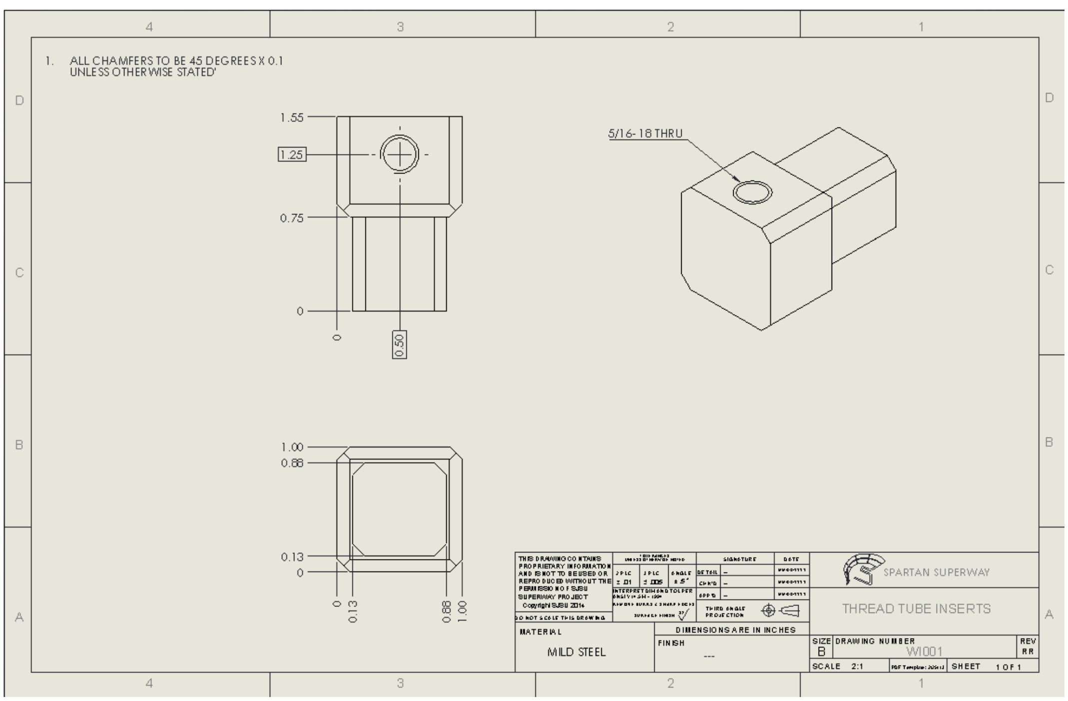


Figure B2-6. Threaded tube inserts

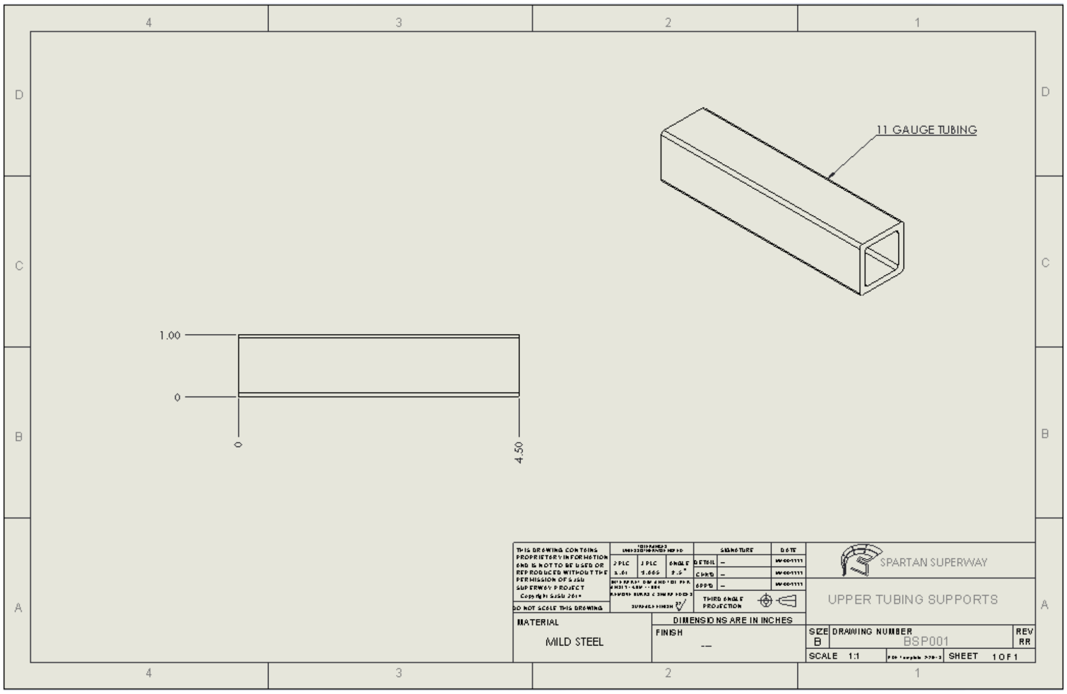


Figure B2-7. Upper tubing support members

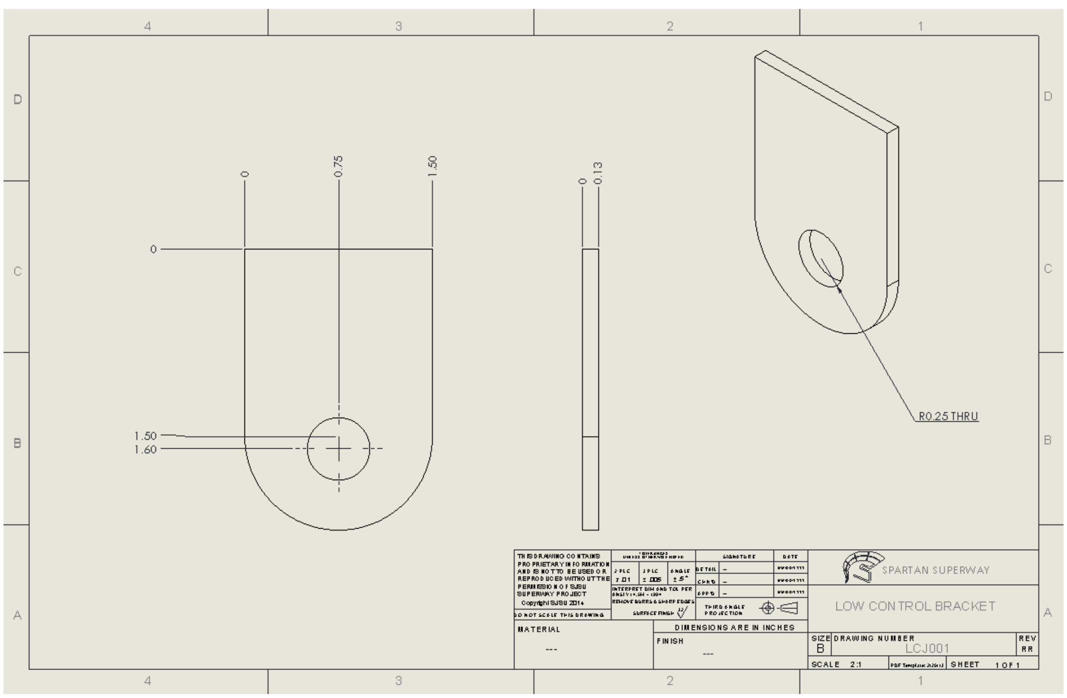


Figure B2-8. Low control arm bracket

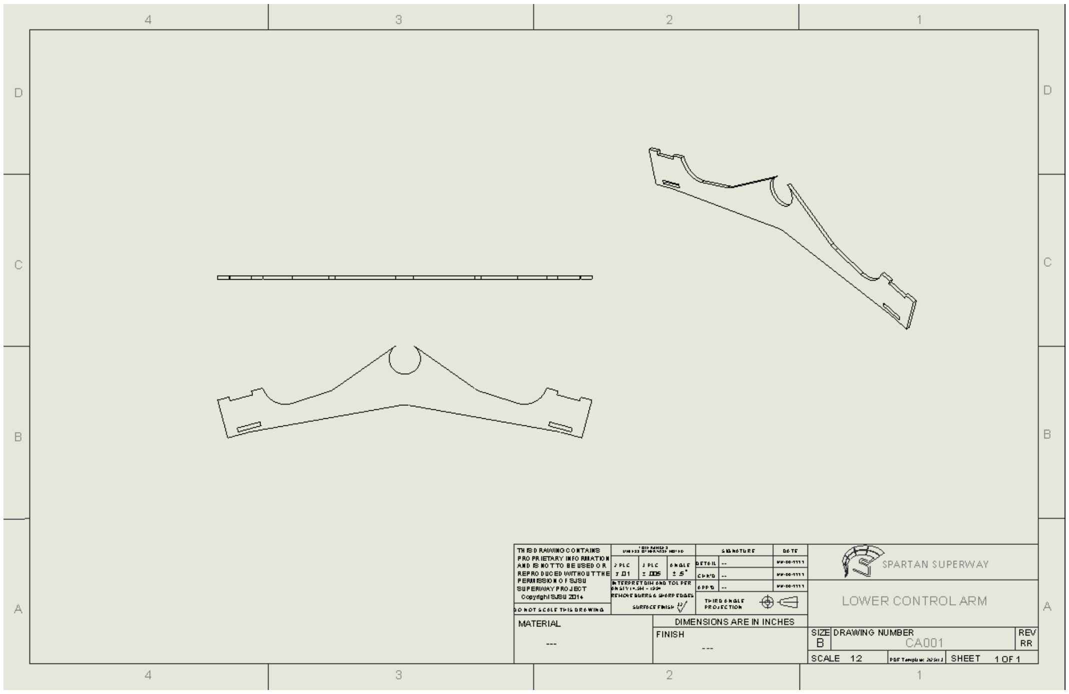


Figure B2-9. Lower control arm plate

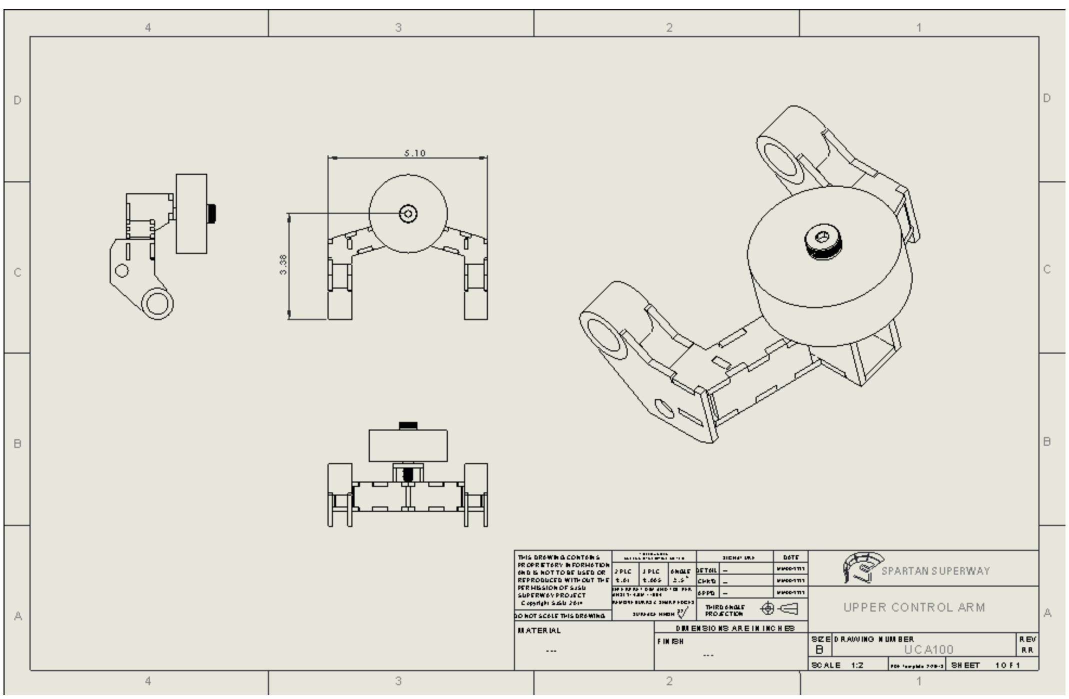


Figure B2-10. Upper control arm assembly

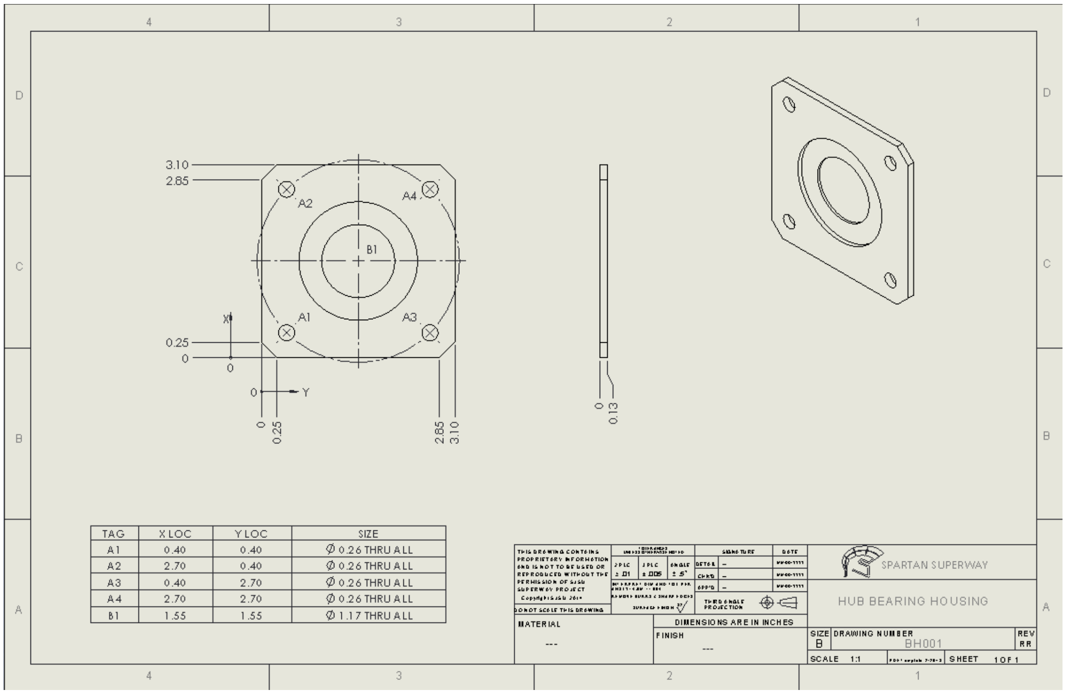


Figure B2-11. Hub bearing housing

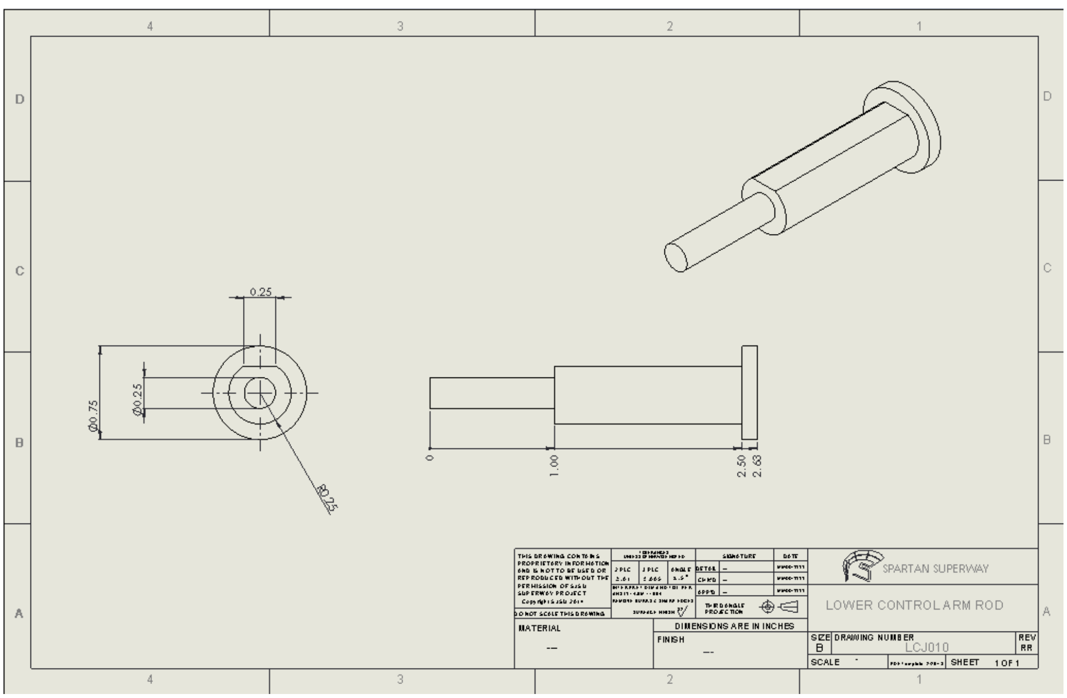


Figure B2-12. Lower control arm rod

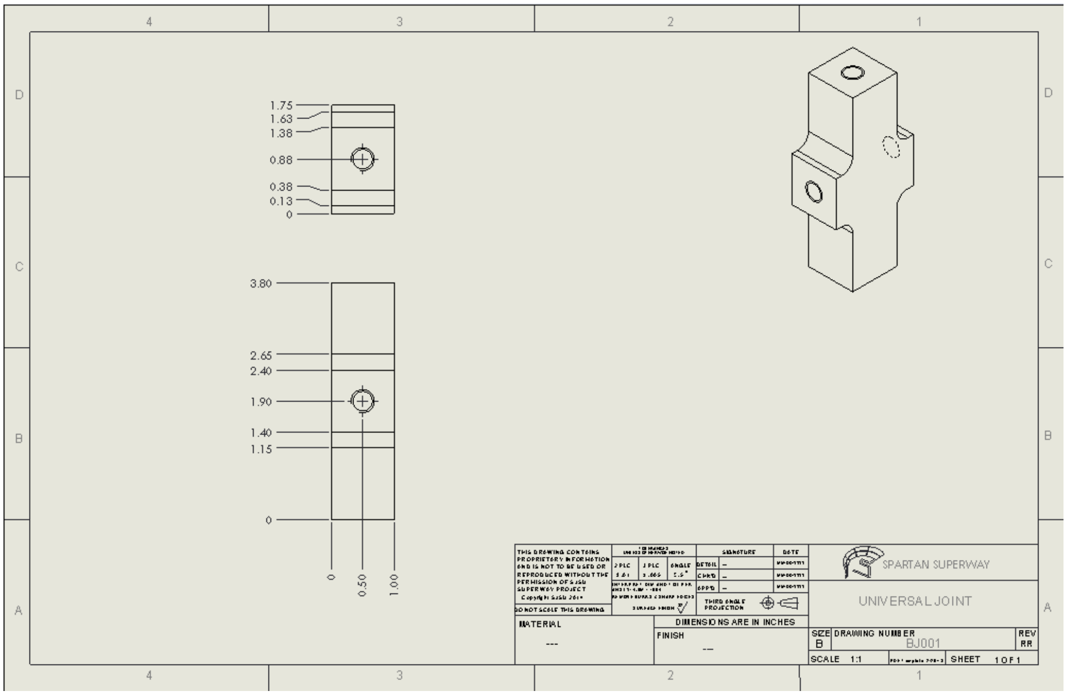


Figure B2-13. Universal Joint

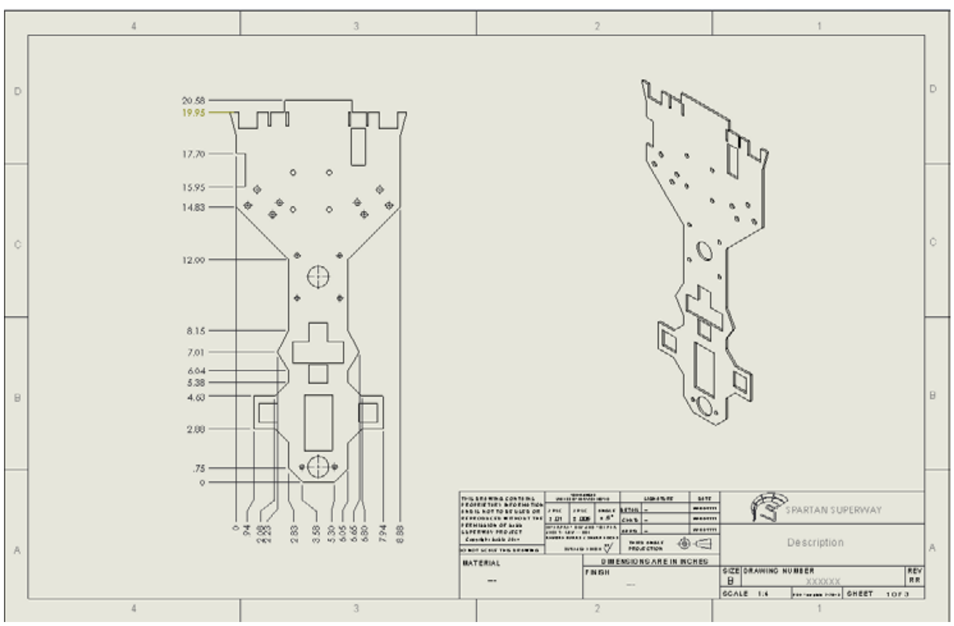


Figure B2-14. Bogie side plate

Appendix B2-2

Calculations

Lateral Force Calculations

Assumptions:

Load = 1000lb, R = 12 ft., g = 32.17 ft/ sec^2 , V = 3.6 ft/sec (2.5 MPH)

$$a = \frac{v^2}{R} \qquad (1)$$

$$F = ma \qquad (2)$$

or

$$F = \frac{Load * a}{g}$$

Combine 1 and 2 to get equation 3.

$$F = \frac{Load*v^2}{g*R} \qquad (3)$$

$$F = \frac{(1000) * (3.6)^2}{(32.17) * (12)} = 33.57\ lb$$

Even assuming V = 5 MPG, F = 138.04 lb. This is the force that the steering arms will need to overcome.

Sources:

J. Glennon, "Calculating Critical Speed", Crash Forensics, 2016. Retrieved from: http://www.crashforensics.com/papers.cfm?PaperID=42

Figure B2-15. Calculations for lateral force.

Steering Design

The motor available has a holding torque= 106 lb-in and a **speed**= 50 RPM

Lateral force due to weight of the bogie=33.57 lbf

Chosen gears and pulley → **48 teeth gear**, **16 teeth gear**, and **24 teeth pulley**.

Gear Ratio= $\frac{Teeth\ of\ driven}{Teeth\ of\ Driver}$ x $\frac{Teeth\ of\ Driven}{Teeth\ of\ Driver}$

Gear ratio of top linkage= $\frac{48}{16}$=**3**

Gear ratio at pulley= $\frac{24}{16} = \mathbf{1.5}$

Speed= RPMout= $\frac{RPMin}{Ratio}$

Speed at top= $\frac{50}{3} = \mathbf{16.6\ RPM}$

Speed at pulley= $\frac{50}{1.5} = \mathbf{33.3\ RPM}$ $(need\ greater\ speed\ as\ it\ rotates\ a\ greater\ distance)$

Torque(out)= Torque(in) x Ratio → **Torque(out) top**= (106)(3)=**318 lb-in**

Torque(out) at pulley= (106) (1.5) = **159 lb-in**

Figure B2-16. Calculations for gear box implemented in steering.

$$F = ma$$

or $$F = \frac{Wa}{g} \qquad \text{(Eq.2)}$$

where F = force (ft–lbs.)

m = mass (W/g)

W = weight of body (lbs)

g = acceleration of gravity (ft/sec^2)

substituting Equation 1 into Equation 2, the following can be written:

$$F = \frac{Wv^2}{gR} \qquad \textit{(Eq. 3)}$$

Figure B2-17. Equations used to find the lateral force. Published by j. Glennon.(2016). "calculating Critical Speed". Crash Forensics. Retrieved from: http://www.crashforensics.com/papers.cfm?PaperID=42

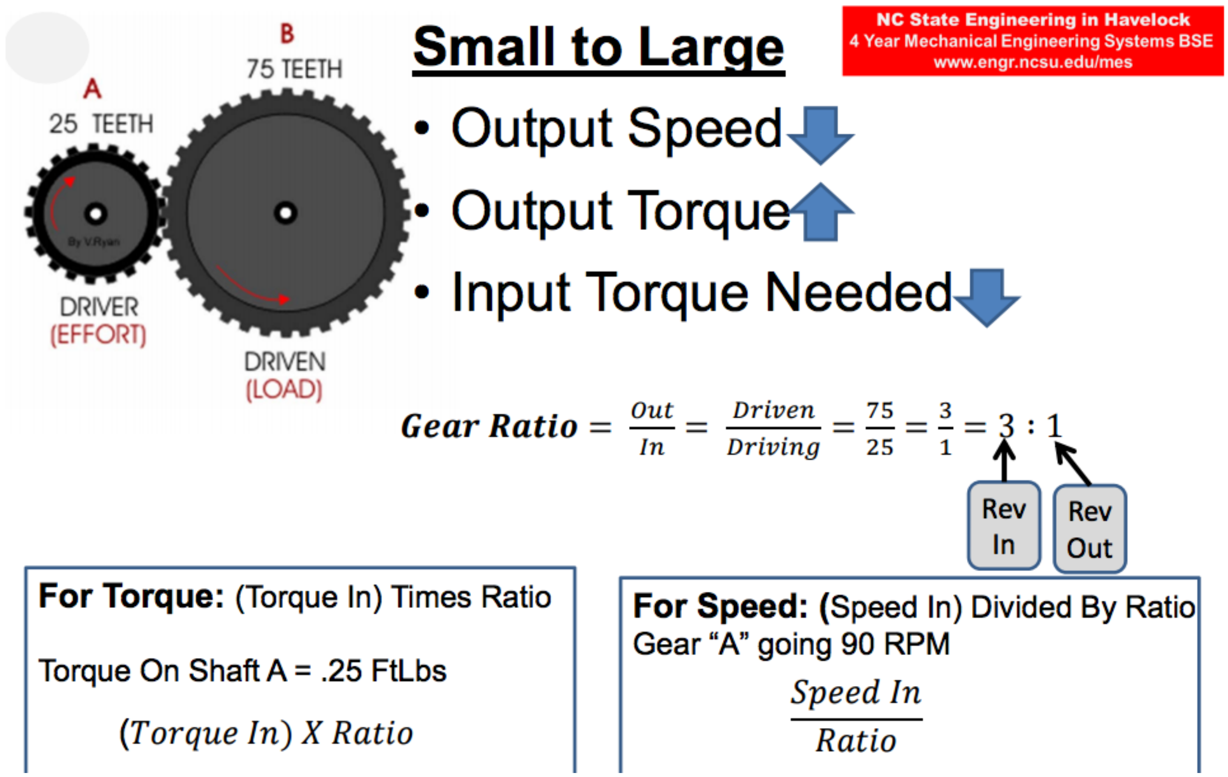


Figure B2-18. Concepts of gears. Published by NC State University Engineering. "Gear Ratio". NC State University, n.d. Retrieved from: https://www.engr.ncsu.edu/mes/media/pdf/gears

$$F = \frac{(600\ lbs) * (3\ in)}{(0.125\ in)} = 14400\ lb$$

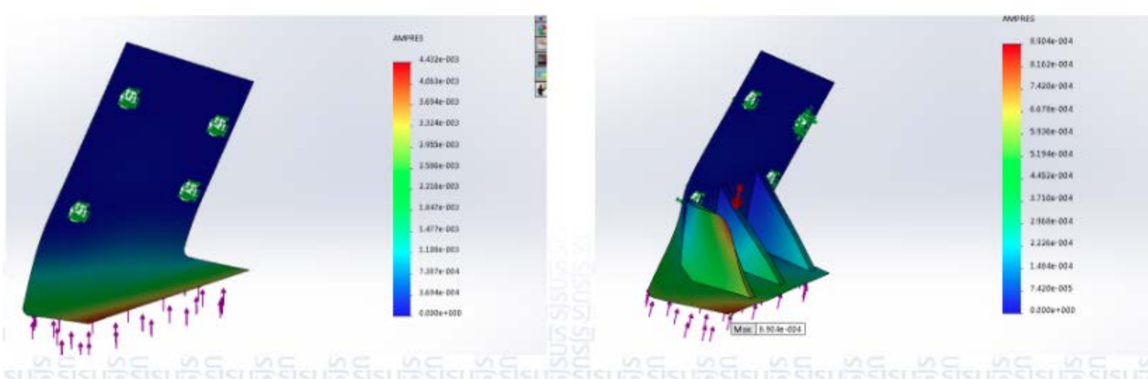


Figure B2-19. FEA analysis for failsafe.

$$\sigma = \frac{P}{A}$$

σ – Normal Stress

P – Axial Force

A – Cross Sectional Area

A normal stress will occur when a member is placed in tension or compression. Examples of members experiencing pure normal forces would include columns, collar ties, etc.

Bending Stress

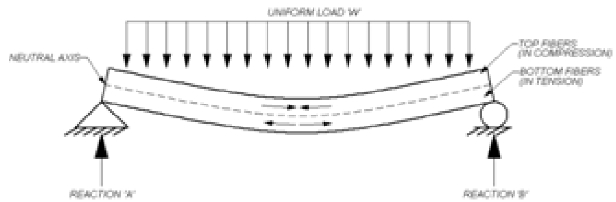


Figure 1-Beam diagram

Figure B2-20. Bending stress formula used for the Joint connecting the two bogies. Published by Parker, Josh. (n.d). “Normal Stress, Bending Stress, and Shear Stress”. StruCalc. Retrieved from: http://www.strucalc.com/engineering-resources/normal-stress-bending-stress-shear-stress/

Appendix B2-3

Data Sheets

larger image

Item # PK296A2A-SG36, Stepper Motor
Web Price $268.00

Incorporating the SH gears with high permissible torque delivers high resolution, high torque and smooth low-speed rotation.

RoHS

Unit of Measure: Imperial | Metric

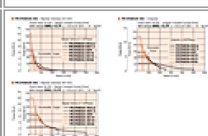
Lead Time · **Specifications**

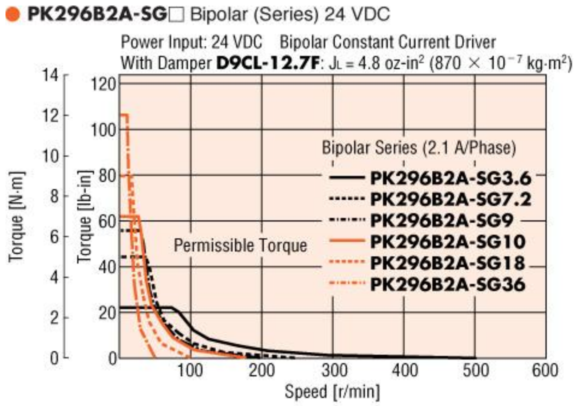
Lead Time

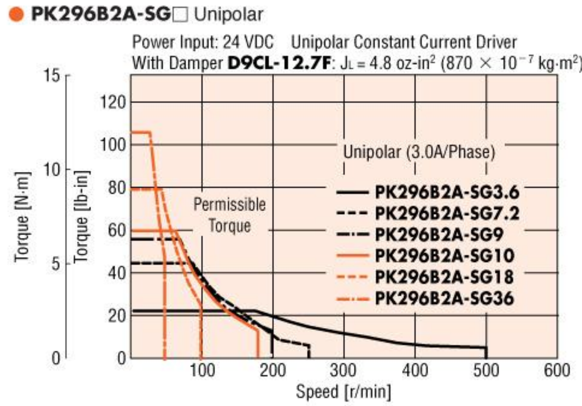
Available to Ship[1]	10/25/2016

[1] Quoted Ship Date for orders placed before 12:00 pm PST. Quantities may affect Shipping Date.

Specifications

Motor Type	2-Phase
Frame Size	3.54 in
Motor Length ?	4.96 in.
Speed-Torque Characteristics	 Speed - Torque Characteristics larger image
Holding Torque ?	1696 oz-in
Shaft/Gear Type	Spur Gear
Gear Ratio (X:1)	36 :1
Shaft	Single
Type	Geared





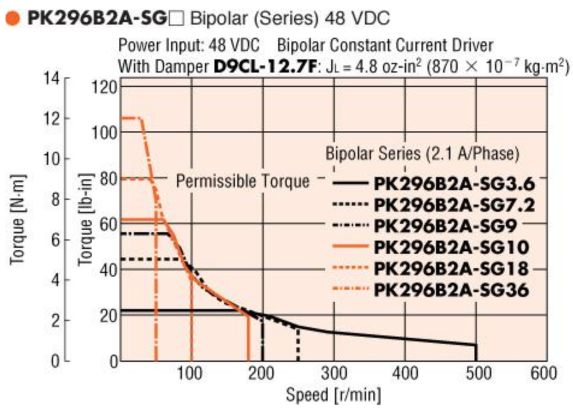


Figure B2-21. Steering motor datasheet. Published by Oriental motors. (2016). “Data sheet for PK296A2A-SG36, Stepper Motor” [data sheet]. Retrieved from: http://catalog.orientalmotor.com/item/stepping-motors/pk-series-stepping-motors/pk296a2a-sg36?&seo=110

Braking and Propulsion

Appendix B3-1

Propulsion motor Specs

MOTORTEC™ DATA SHEET

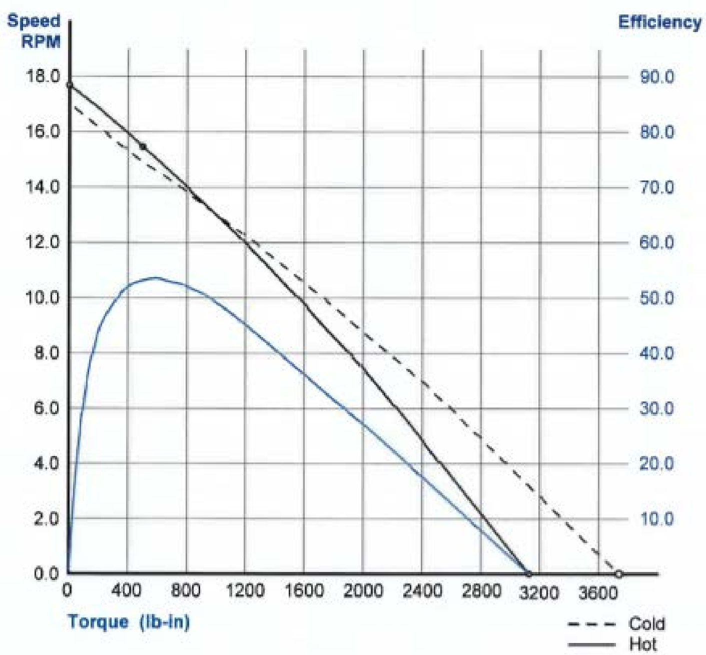
Customer: Craig Hudacko
Application: Pod Transportation System
Prepared by: Turner Stammer · 712-722-4135

Quote No: ---
Date: 11/03/16

Product ID: 62772

Motor: PM 8014
Voltage: 24 v DC
Speed: 2500 rpm 2 Poles
Gearbox: PL 7300 160:1 Ratio

Speed, Efficiency Vs. Torque



Notes:

SPECIFICATIONS

Rating:
Speed: 15.5 rpm
Torque: 498.4lb-in
Current: 7.10 amps
Output: 91 watts
Output: 0.1223 HP

Thermal Capacity:
Gearbox: 34.0 %
Motor: 100.0 %

Duty Cycle:
On: Continuous
Off:

Efficiency:
Gearbox: 72.4 %
Motor: 73.3 %
System: 53.1 %

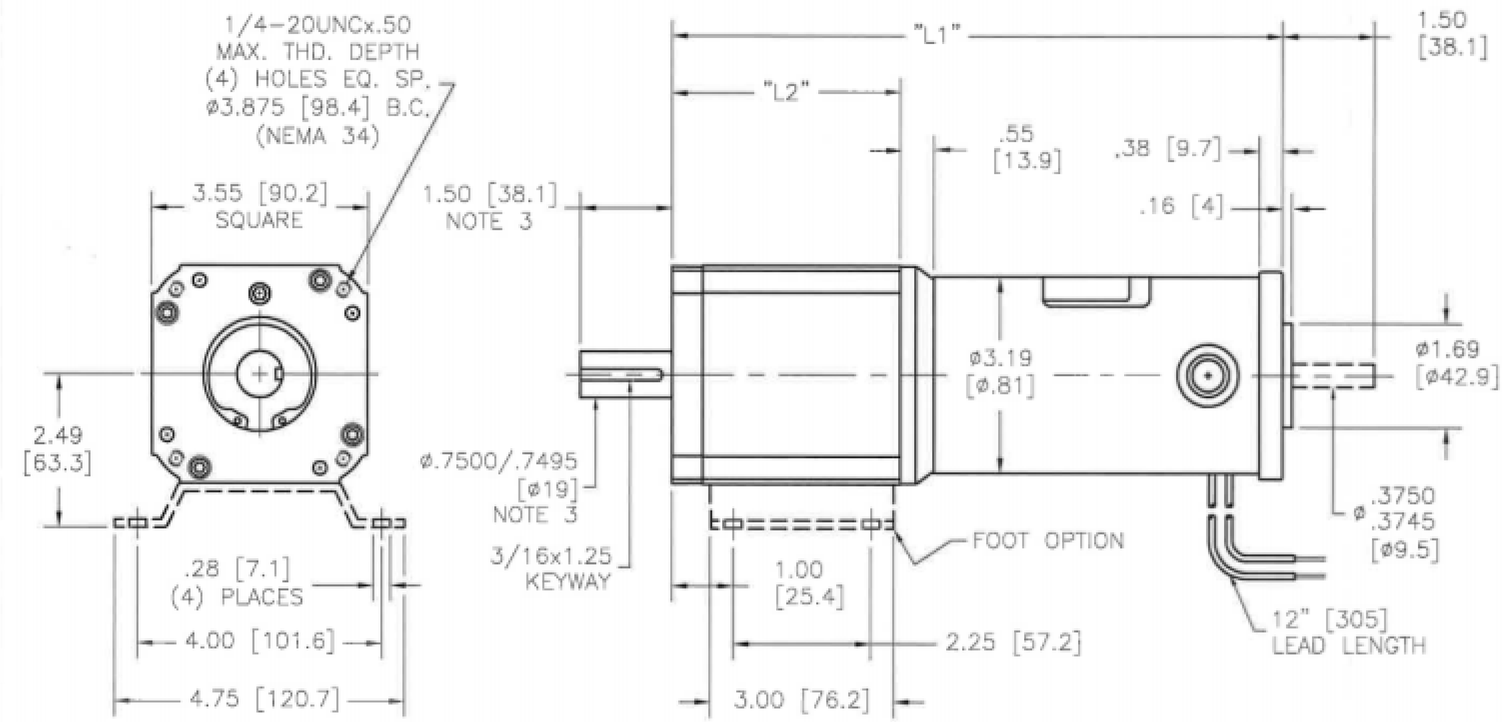
Start/Stall Conditions:
Current: 44.41 amps
Torque: 3738.48 lb-in
Yield: 5800.00 lb-in

Constants:
Ke: 8.20 v/krpm
Kt: 0.69 lb-in/amp

Design Notes:
Motor thermal capability limits design.

PM8014-PL73160
PM8014-209-024
PL7300-160

GROSCHOPP
Groschopp, Inc., 420 15th Street NE, Sioux Center, IA 51250-2100 USA
(712) 722-4135 Phone · (800) 829-4135 Toll Free · (712) 722-1445 FAX
www.groschopp.com



MODEL	RATIO	"L1"	"L2"
PM8014-PL73__	5:1 – 61:1	9.99 [254]	3.73 [95]
	90:1 – 614:1	10.60 [269]	4.34 [110]
	1000:1	11.20 [284]	4.94 [125]
PM8018-PL73__	5:1 – 61:1	11.49 [292]	3.73 [95]
	90:1 – 614:1	12.10 [307]	4.34 [110]
	1000:1	12.70 [322]	4.94 [125]

NOTES:

1. ALL DIMENSIONS ARE NOMINAL.
2. [mm] = MILLIMETERS.
3. ADDITIONAL STANDARD SHAFT SIZES AVAILABLE UP TO ø1¼" IN DIAMETER & 3" IN LENGTH.

GROSCHOPP
ELECTRIC MOTORS • REDUCERS • GEAR MOTORS • MOTOR SETS

PM801_–PL73__
CATALOG STANDARD

REV: 3
DATE: 12-2-13

Appendix B3-2

Braking calculations

Stopping Distance:

$$x = 50\text{ft} = 15.24\text{m}$$

Initial, Top Velocity:

$$V_i = 20\text{mph} = 8.94\frac{\text{m}}{\text{s}}$$

Final velocity:

$$V_f = 0$$

Kinematics Equation:

$$V_f^2 = V_i^2 + 2ax$$

Solve for a:

Deceleration required for 50ft:

$$a = 2.62 \text{ m/s}$$

Weight of Half Bogie:

$$W = 300\text{lbs} = 1335\text{Newtons}$$

Mass of Half bogie:

$$m = \frac{1335 \text{ N}}{9.81 \text{ m/s}^2} = 356.9 \text{ kg}$$

Newton's Second Law:

$$F = ma$$

Solve for F, using deceleration calculated previously:

Braking force required for half bogie

$$F = 357 \text{ N}$$

Torque Equation:

$$T = F \cdot r$$

Radius of wheel:

$$r = 6\text{in.} = 0.1524\text{m}$$

Solve for T, using braking force:

Braking torque required per half bogie

$$T = 54.39\ N \cdot m$$

Appendix B3-3

Propulsion Calculations

Torque to oppose gravity (Nm): T = sin(17)* F1* r (wheel)

F1 = 1500N
T = sin(17)*1500N*.0762m = 30Nm
Rolling Friction (Nm) : T = f * cos(17)*F1
T = .0077 (Coefficient of rolling friction of hard rubber on steel) *cos(17)
*1500N = 14Nm
Bearing Friction (Nm) : T = F * f * (d/2) approximately .03Nm
T = 1500N*.0015(Coefficient of friction for ball bearings * (.019/2)
Where the diameter for ball bearing is .019m. Approximation at best.
Total Torque needed to move the wheel is 44.03Nm

The motor that was selected has a Ke constant of 8.20 V/krpm and Kt constant of .69 lb-in/amp
It also has a gearbox with a 160:1 ratio at 73% efficiency meaning that the torque the motor would produce is .38Nm or 3.375 lb-in (before the gearing)
Knowing the torque and the value of Kt allows for us to calculate the approximate current, being 4.9 amps
Given that the motor itself is 24 volts, the power needed for each motor would be 120 Watts of power.
While running approximately at 13.3 rpm at the wheel as it goes up an incline
For 2 motors, a total of 240 watts are required

Appendix B3-4

Design Iterations

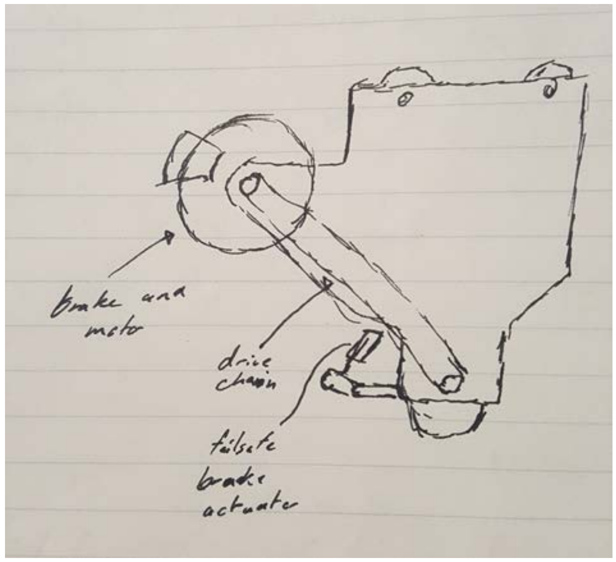


Figure B3-1: Initial design with offset disc brake and motor, using a chain to drive the wheels

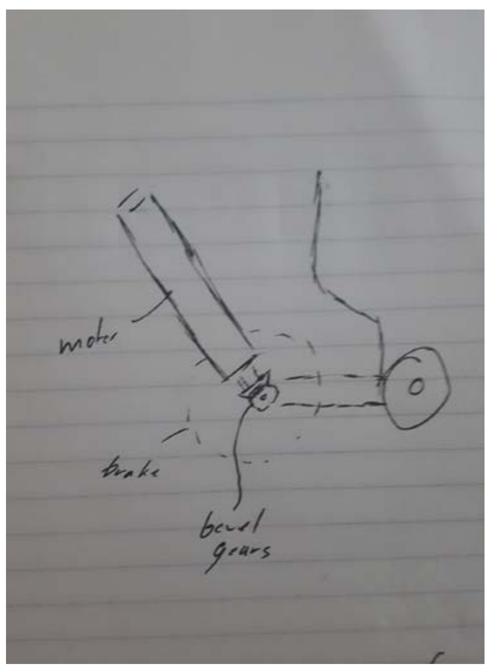


Figure B3-2: Secondary design with heavily geared motor with miter gear and chain mechanism to drive the wheels. Thr Disc brake is also offset

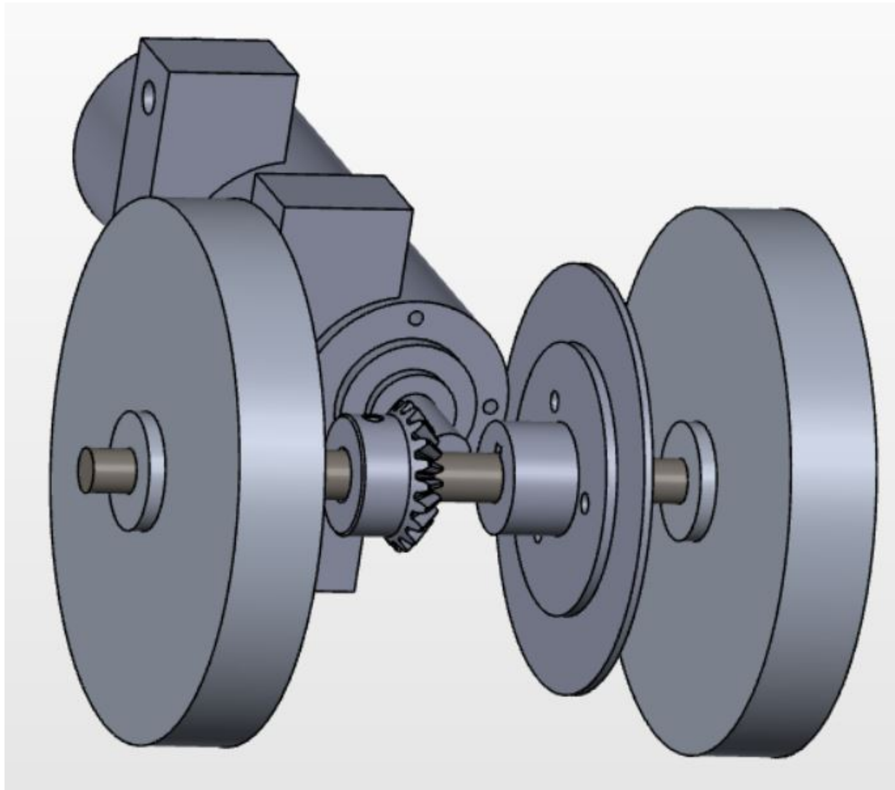


Figure B3-3: Third Design iteration with geared dc motor using a direct drive and a small disc brake attached to the main shaft.

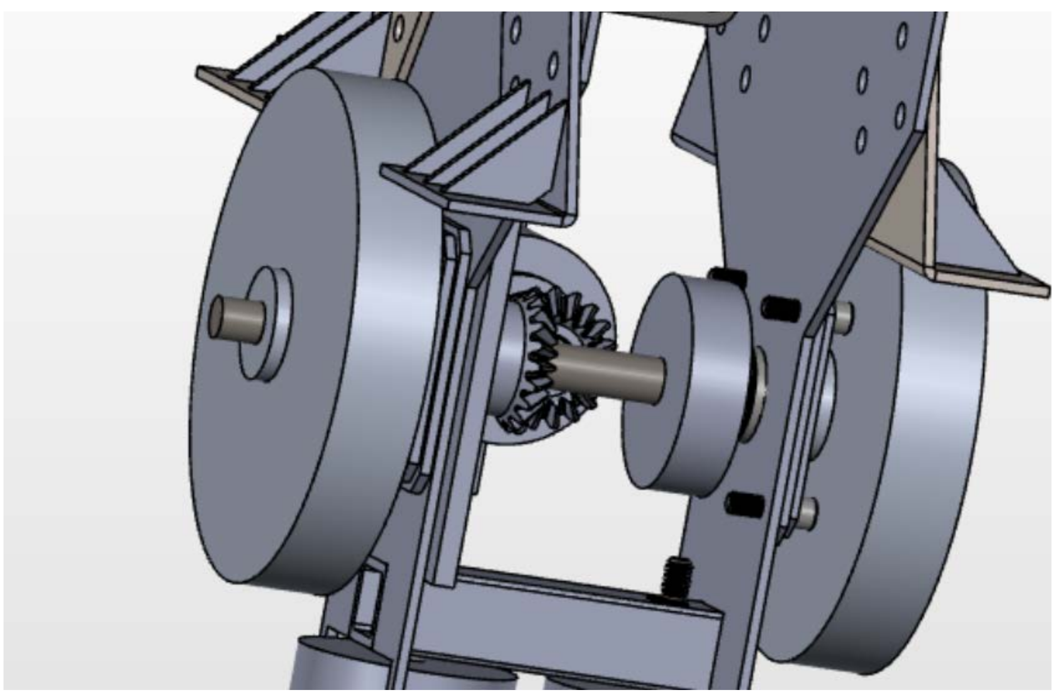


Figure B3-4: Fourth iteration, using the same DC motor and a band brake disc on the main drive shaft

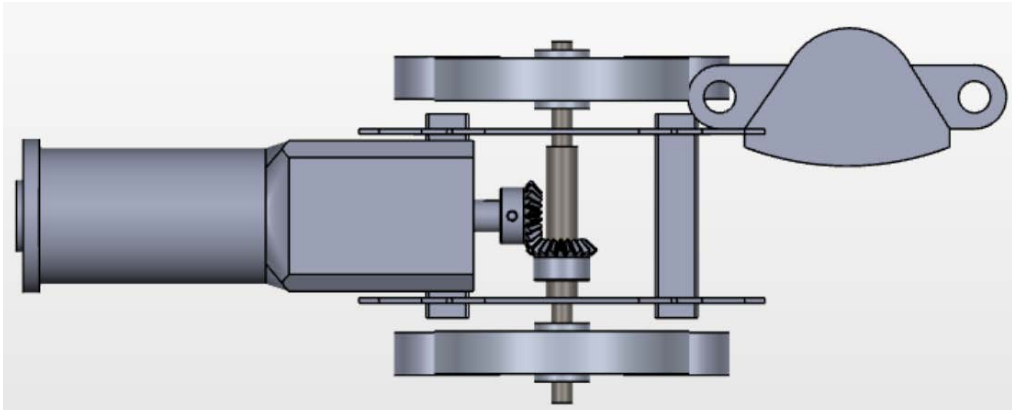


Figure B3-5: Fifth design iteration, using an update motor model that was quoted and a caliper from a car that will be used on the main rail of the system

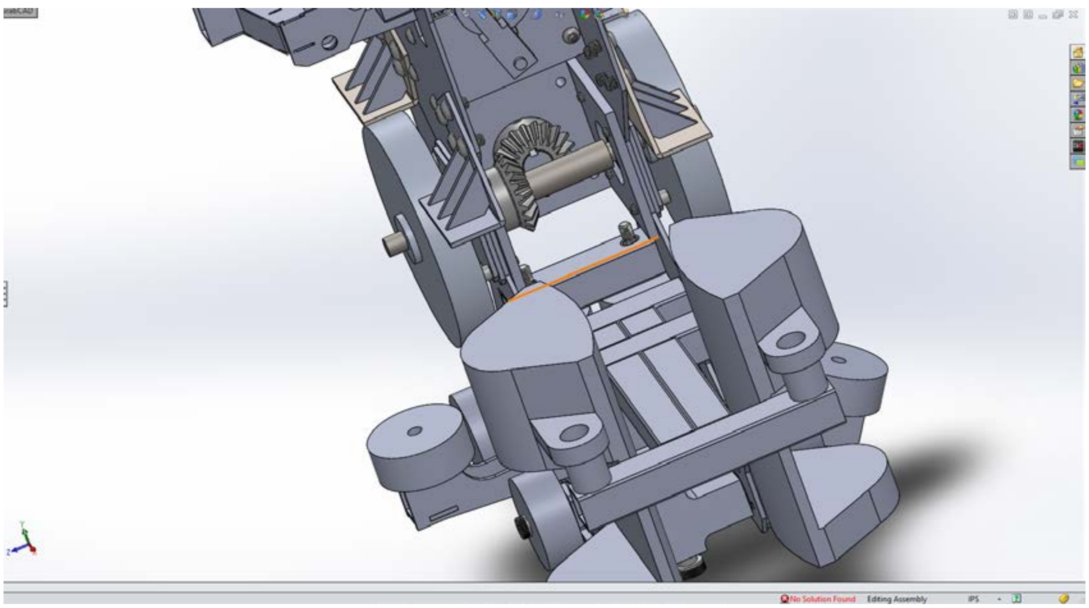
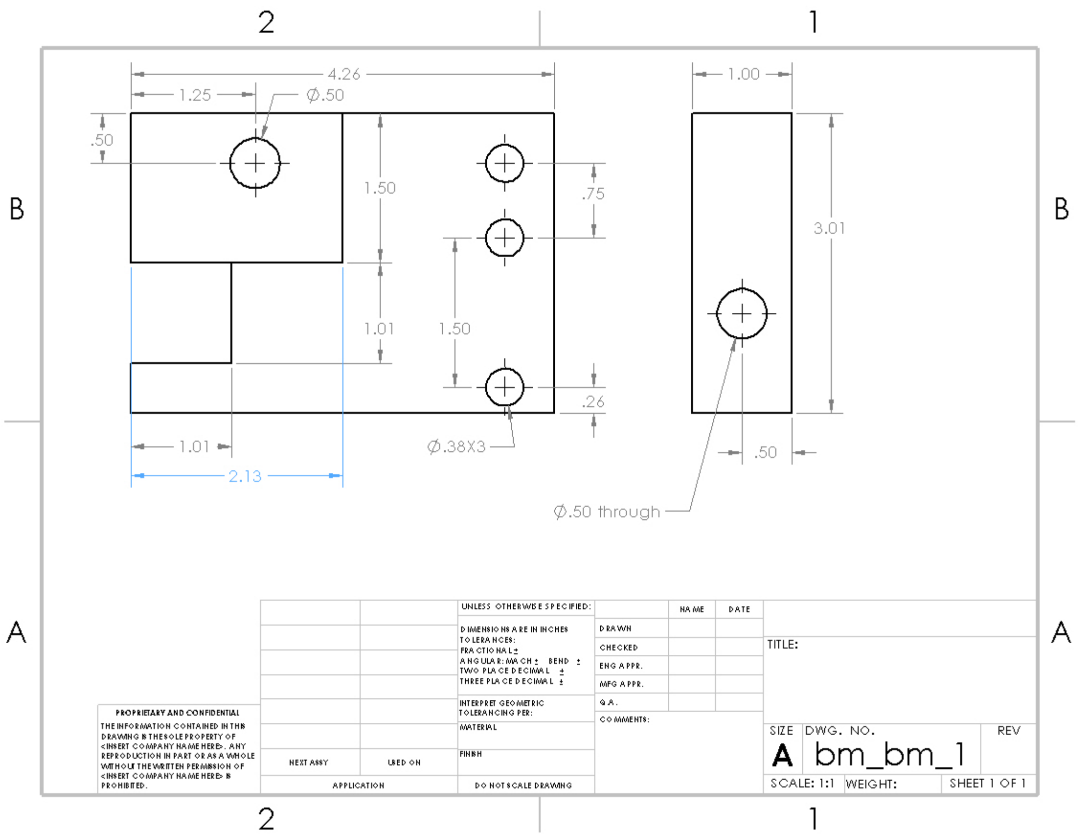
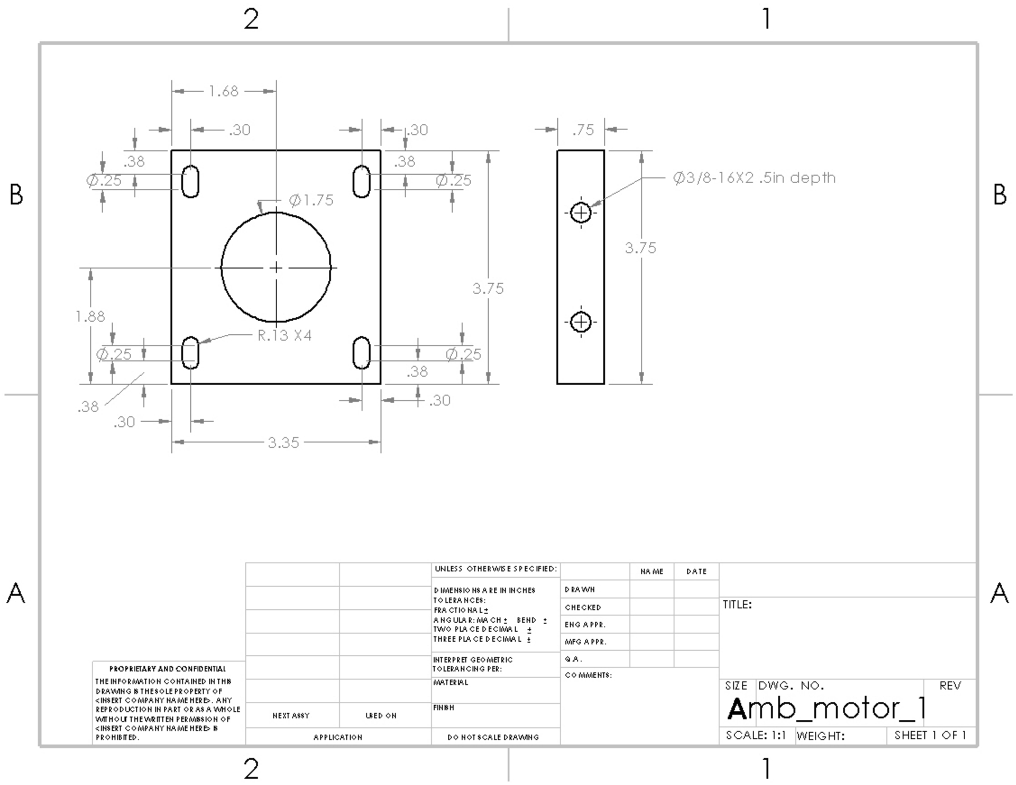


Figure B3-6: Most up to date model with the mounting brackets for both the brakes and the motor

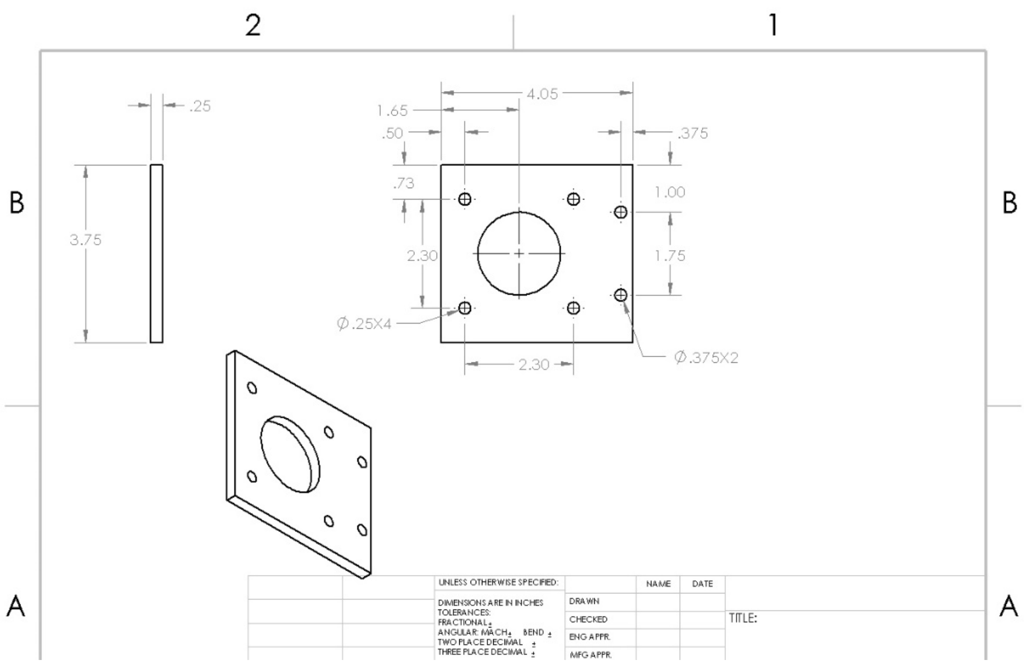
Appendix B3-5

Detailed Drawings





1.68
.30
.38
Ø.25
Ø1.75
1.88
R.13 X4
3.75
3.35
.75
Ø3/8-16X2 .5in depth
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
SIZE
DWG. NO.
REV
Amb_motor_1
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING



.25
3.75
4.05
1.65
.50
.375
.73
1.00
2.30
1.75
Ø.25X4
Ø.375X2
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
TITLE:

Mechatronics and Integration Systems

Appendix B4-1

$$Vo = 0$$
$$V1 = \frac{0.5m}{s}$$
$$a = \frac{dV}{dT} = \frac{0.5}{5} = \frac{0.1m}{s^2}$$
$$t = 5s$$
$$V1^2 = Vo^2 + 2 * a * d$$
$$d = 1.25m\ distance\ to\ reach \max V$$
$$\frac{5}{1024} \sim 5seconds$$

Figure B4-1. Equation for calculating timing

Components	Driver	Power rating	Power
1 x Arduino	12V power supply (5V desired output)	5VDC, 1A	5W
2 x PM8014 motor	MOSFET	24VDC, 8.42A*	405.6W*
2 x PK296A2A	CW230	24VDC, 3A	144W
1 x PM8014 motor	MOSFET	24VDC, 8A	192W
1 x Lin Egr Stepper	L298N	12VDC, 1A	12W
1 x PA18 linear act.	CW230	12VDC, 8A max	96W
2 x NEMA23 motor	L293D	12VDC, 1A	24
Total			**878.6W***

Figure B4-2. Power consumption of the system

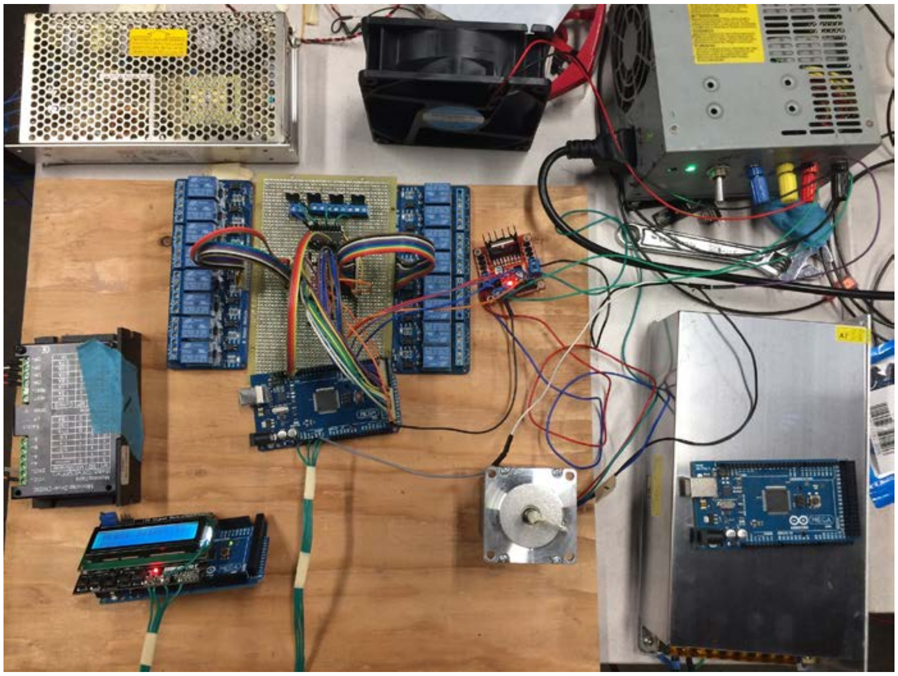


Figure B4-3. Layout of electrical components

Figure B4-4. Case for LCD control panel with emergency stop button installed

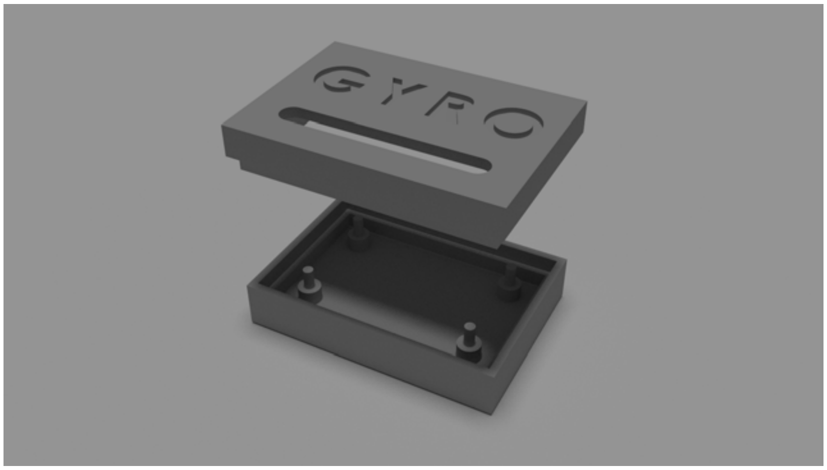


Figure B4-5. Case design for housing the gyro

Appendix B4-2

```
/*******************************************************

  Spartan Superway F2016-S2017
  Author: Patrick Ding
  Team: 1/2 scale Mechatronics
  All Copyrights Reserved.
  Date created: 23rd January 2017

********************************************************/
#include <Wire.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <Stepper.h>

// select the pins used on the LCD panel
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

// define some values used by the panel and buttons
int lcd_key     = 0;
int adc_key_in  = 0;
#define btnRIGHT  0
#define btnUP     1
#define btnDOWN   2
#define btnLEFT   3
#define btnSELECT 4
#define btnNONE   5
const int stepsPerRevolution = 200; // change this to fit the number of steps per revolution

//int hardware
int motor1 = A13;
int steering = 28;
int hallLeft = A10;
int hallRight = 46;
int limitSwitch = 45;
Stepper myStepper(stepsPerRevolution, 21, 22, 23, 24);

int read_LCD_buttons()
{
  adc_key_in = analogRead(0);      // read the value from the sensor
  // my buttons when read are centered at these valies: 0, 144, 329, 504, 741
  // we add approx 50 to those values and check to see if we are close
  if (adc_key_in > 1000) return btnNONE; // We make this the 1st option for speed reasons since it will be the most likely result
  // For V1.1 us this threshold
  if (adc_key_in < 50)   return btnRIGHT;
  if (adc_key_in < 250)  return btnUP;
  if (adc_key_in < 400)  return btnDOWN;
  if (adc_key_in < 600)  return btnLEFT;
  if (adc_key_in < 850)  return btnSELECT;

  return btnNONE;  // when all others fail, return this...
}

//int states
int brakes = 3;
int trackNumber = 0;
int motorSpeed = 0;
int hallLeftValue = 0;
int hallRightValue = 0;
int limitValue = 0;

void setup()
```

```
{
  myStepper.setSpeed(30);   //****************30RPM FOR SPEED****************//
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Spartan Superway");
  lcd.setCursor(3, 1);
  lcd.print("2016-2017");
  delay(1000);
  lcd.clear();
  delay(1000);

  Serial.begin(9600);
  pinMode(motor1, OUTPUT);
  pinMode(hallLeft, INPUT);
  pinMode(hallRight, INPUT);
  pinMode(limitSwitch, INPUT);
  limitValue = digitalRead(limitSwitch);
  pinMode(13, OUTPUT);  //***********CHANGE PIN ACCORDINGLY****************//
}

void loop()
{
  lcd.setCursor(0, 0);
  lcd.print("Select to run");


  adc_key_in = analogRead(0);
  delay(100);// read the value from the sensor
  lcd_key = read_LCD_buttons();  // read the buttons

  switch (lcd_key)               // depending on which button was pushed, we perform an action
  {
    case btnSELECT:
      {
        normalOperation();    //*********************MAIN OPERATION***********************//
        break;
      }
  }
}

void normalOperation() {      //***************1. MAIN OPERATION STARTS HERE @trackNumber = 1*****************//

  trackNumber = 1;
  int parkingBrake = 0;
  lcd.clear();
  delay(500);
  lcd.setCursor(0, 0);
  lcd.print("Running");
  delay(300);

  normalMotion();                  //**************2. FIRST RUN****************************//

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Switching tracks");    //**************8. SWITCH TRACKS*********************//

  steeringMotion(); //switching to track 2

  lcd.print("Switching complete");
  delay(500);
  lcd.clear();
  lcd.print("Running 2");

  normalMotion();                  //**************9. SECOND RUN @trackNumber = 2***************************//

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Switching tracks");

  steeringMotion(); //switching to track 1

  analogWrite(motor1, 0);          //**************13. Motor shuts off**************************************//
  lcd.clear();                     //**************14. LOOP COMPLETE****************************************//
}

void normalMotion() {
  lcd.clear();
  for (motorSpeed = 0; motorSpeed <= 2048; ++motorSpeed) {//***************3. ACCELERATION**************************//
    analogWrite(motor1, motorSpeed);
    delay(5);
    Serial.println(motorSpeed);
    lcd.setCursor(0, 0);
    lcd.print("Motor Speed:");
    lcd.setCursor(12, 0);
    lcd.print(motorSpeed);
  }
  lcd.clear();
  delay(20000);
  while (hallLeftValue == 1 || hallRightValue == 1 || limitValue == 1) {
    analogWrite(motor1, 0);
    delay(1000);
    break;
  }
}
```

```
void steeringMotion() {
  if (trackNumber == 1) {
    myStepper.step(stepsPerRevolution); //****************DEFAULT AT 1 REVOLUTION*****************//
    delay(2000);
    trackNumber = 2;
    Serial.println("On track 2");
    delay(500);
  } else if (trackNumber == 2) {
    myStepper.step(-stepsPerRevolution);
    delay(2000);
    trackNumber = 1;
    Serial.println("On track 1");
    delay(500);
  }
}
```

Wayside Power

Appendix B5-1

Design Drawings

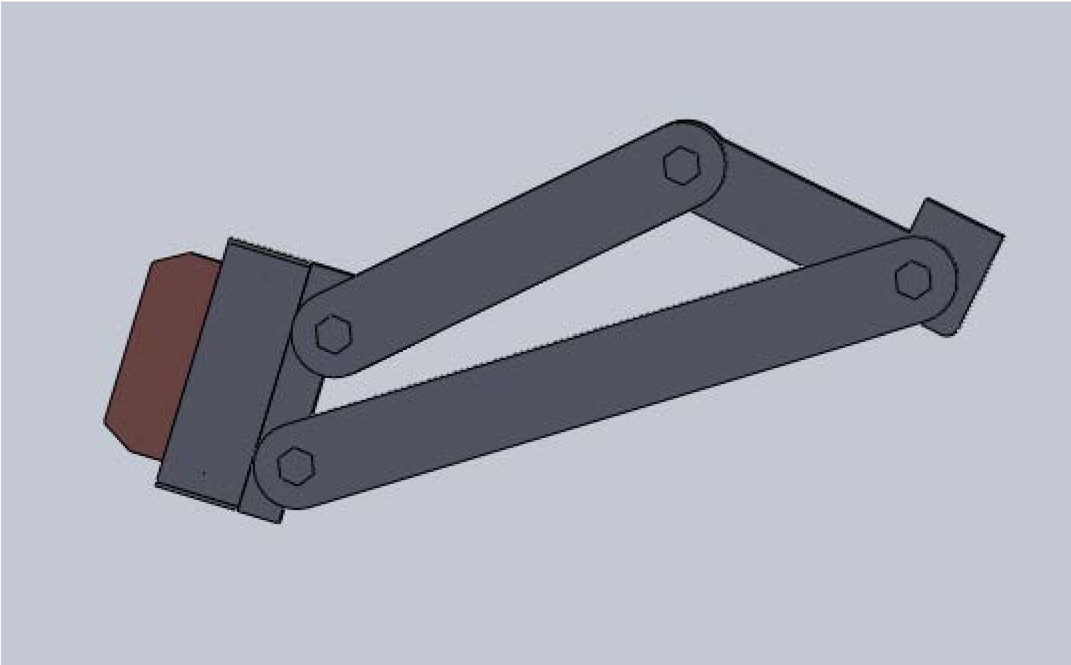


Figure B5-1. Shoe Collector Model

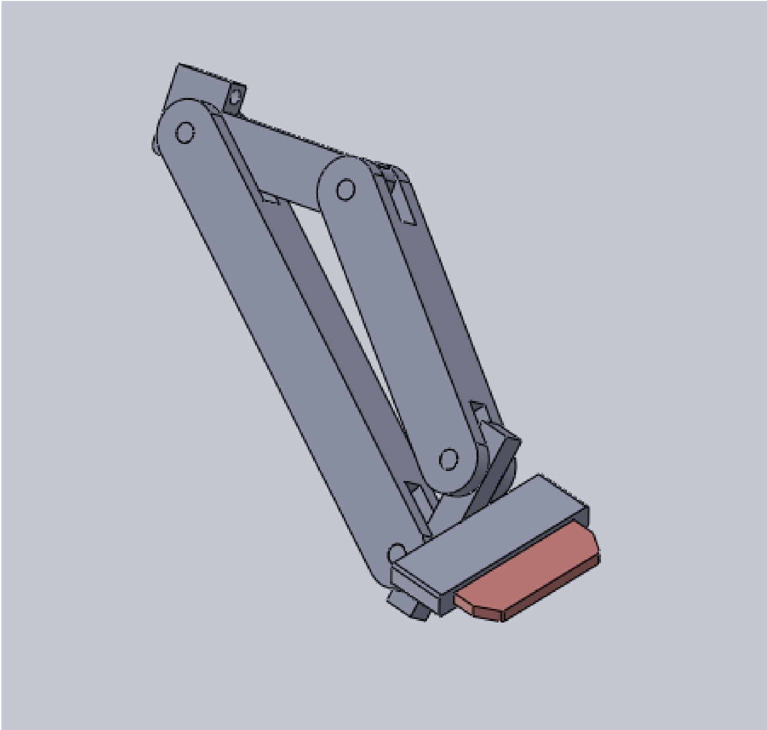


Figure B5-2. Shoe Collector Model (Extended)

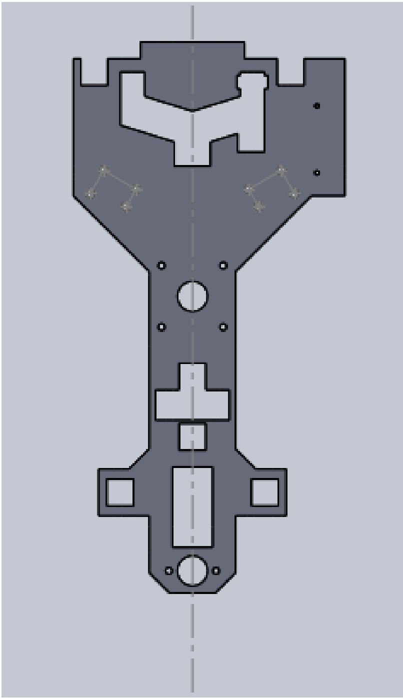


Figure B5-3. Sideways Junction

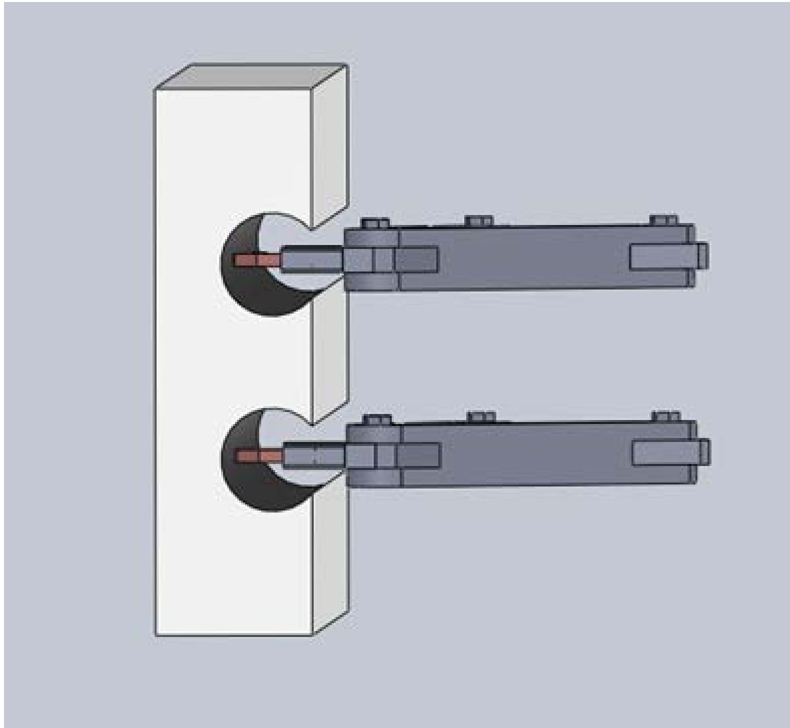


Figure B5-4. Shoe Collector on Rail

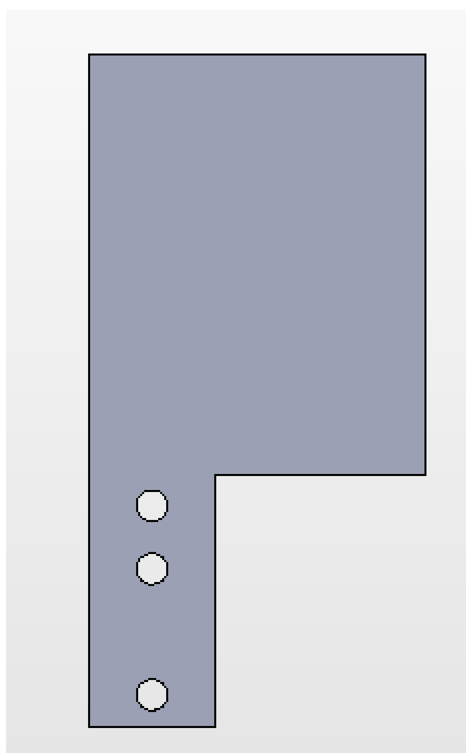


Figure B5-5. Bogie Attachment

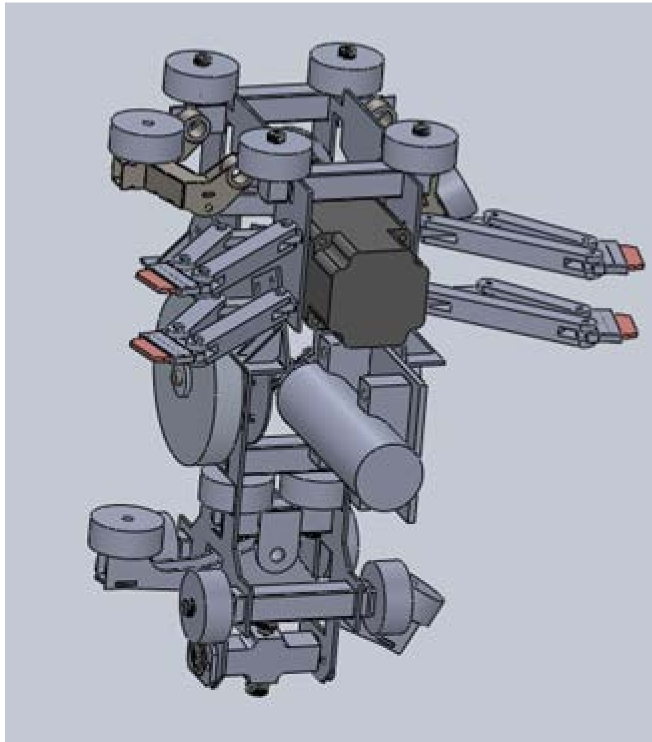


Figure B5-6. Bogie with shoe collectors attached

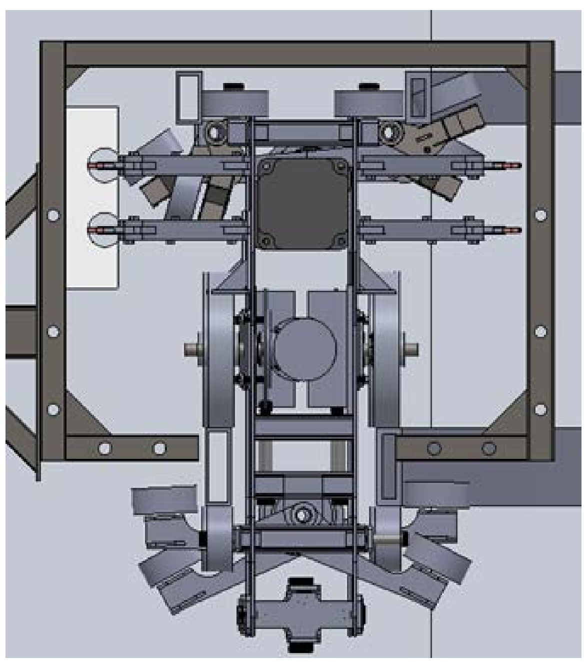


Figure B5-7. Overall Implementation

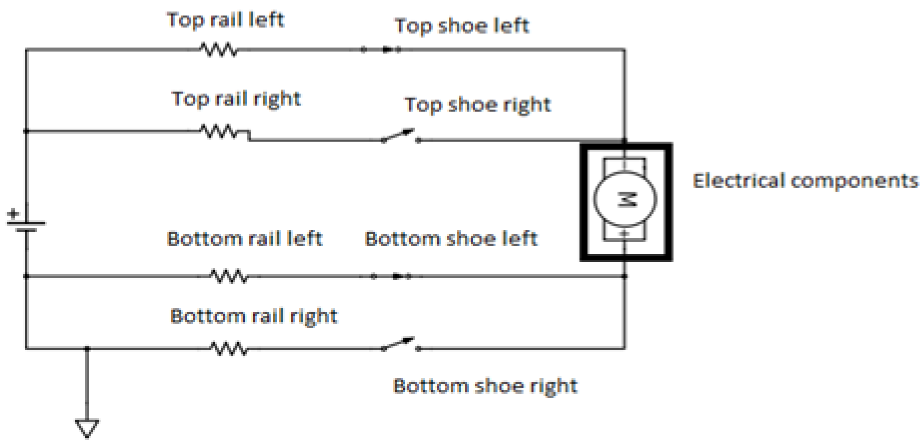


Figure B5-8. Circuit Diagram of Wayside Connection

Appendix B5-2

Calculations & Analysis

	Components	Quantity	Specifications	Voltage (V)	Current (A)	Power (W)
Propulsion	24 V 4A motor	2	4A, 24V	24	4	192
	5V Hall Effect sensor	1	5mA, 5V	5	0.005	0.025
Breaking	Stepper motor	1	330mA, 12V	12	0.33	3.96
Intergration	4in Actuator	1	4.6A, 12V	12	4.6	55.2
	6in Actuator	2	5A, 12V	12	5	120
	Arduino Mega	1	50mA, 7-12V	12	0.05	0.6
	Gyro sensor	1	5mA, 5V	5	0.005	0.025
	5V Hall Effect sensor	3	5mA, 5V	5	0.005	0.075
	4 channel motor driver	1	5A, 12V	12	5	60
					Total Power (W)	431.885
					Fusion current (A)	8.997604

Figure B5-9. Fusion Calculation

Appendix B5-3

Final System Pictures

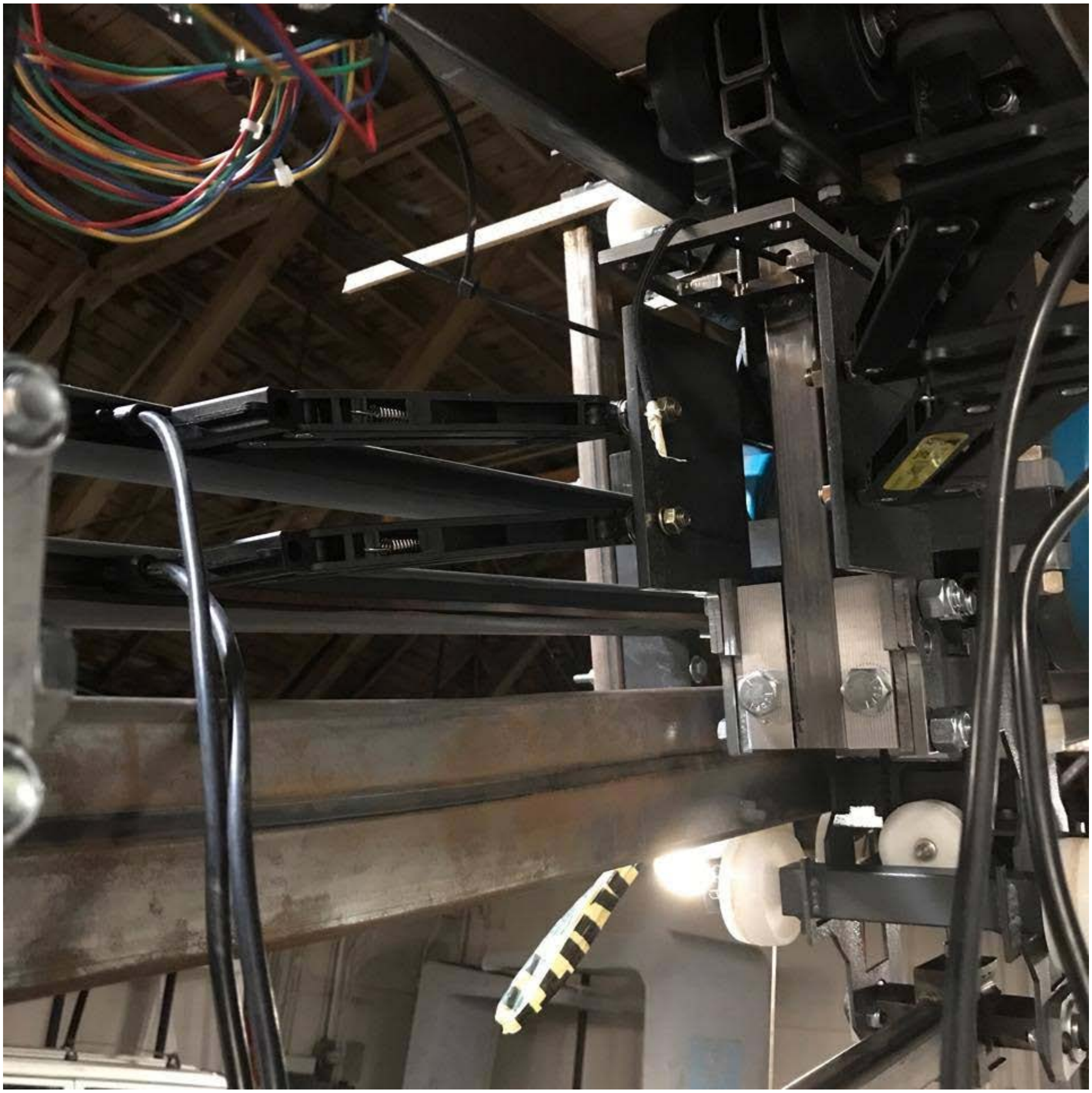


Figure B5-10. Collector to rail

Figure B5-11. Modular screw attachment