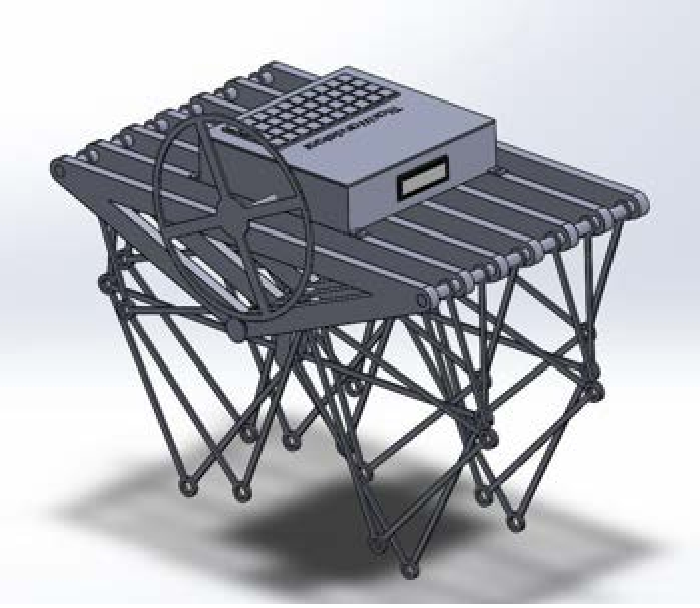


TitanWandelaar

Youssef Abdelhalim
Under supervision of: Shail D.

Table of Contents

List of Figures

List of Tables

Executive Summary

All of the rovers designed by NASA, starting all the way from the Mars Pathfinder in 1996 to the Perseverance rover in 2020, were designed for planets such as Mars with more of a rocky surface. These rovers and all the ones in between were built with 6 wheels which are extremely flexible while still being really strong so that they can navigate jagged surfaces while still being able to maintain the weight of the rover. These designs are not too effective for sandy surfaces; if one of them was placed on Saturn's biggest moon, Titan, they would not be able to move since wheels are not designed to operate on coarse surfaces. In fact, on May 1st, 2009, the Spirit rover got stuck in sand on Mars [1].

I analyzed pictures, videos, and measurements taken on the mechanisms of the wheels used on Mars rovers and made preliminary determinations about where these rovers may encounter issues especially on sandy or coarse terrains. To further extend my analysis and understanding on why these wheels were chosen, I contacted engineers from NASA and reviewed research papers published by NASA explaining their design choices. To gain a better understanding of different approaches to this problem, I reached out to automobile manufacturers that design vehicles for sandy surfaces as well.

The innovation in my design involves a new type of mechanical walker, which is formed using a 4-bar linkage. This linkage is comparable to the Jansen linkage, Trotbot linkage, Klann linkage, and the Strider linkage. It focuses mainly on allowing a rover to be able to transverse on an inclined coarse surface, such as a sand dune. This report does not just showcase this type of linkage, but it also presents it in application by developing a Solidworks model. To emulate a rover and show the effectiveness of my design on a smaller scale, I modeled the following:

- **Linkage model:** A special software was utilized to experiment with different bar lengths and explore the different types of loci that could be formed. Thus, I was able to produce just the right bar lengths to produce the desired locus.
- **Circuit**: An Arduino system was designed to emulate the electrical components on a Mars Rover. This allows the model to do things such as read temperature, control a motor to rotate the crankshaft, and read the level of humidity based on extremity.
- **Solidworks model**: A full model of a rover was created on Solidworks to show how the linkage could be applicable in a full rover system.
- **Animation**: An animation shows how this rover would work from multiple different angles.

Introduction

Dynamic systems of Mars rovers use the Rocker-Bogie design (Appendix A). This suspension design was developed by Don Bickler in 1988 for use in NASA's Mars rover Sojourner. The design worked well on rocky surfaces and became NASA's favored design. It was later used in rovers such as the Spirit, Opportunity, Curiosity, and lastly in 2020 in the Perseverance [2].

Though the innovative design showed great potential, it did not address all the obstacles that these rovers may face on the surface of Mars. On May 1st, 2009, the Spirit rover got stuck in sand on Mars [1]. The Spirit rover was a part of an $800 million program called the Mars Exploration Rover. However, due to this unexpected incident, it was deemed unusable.

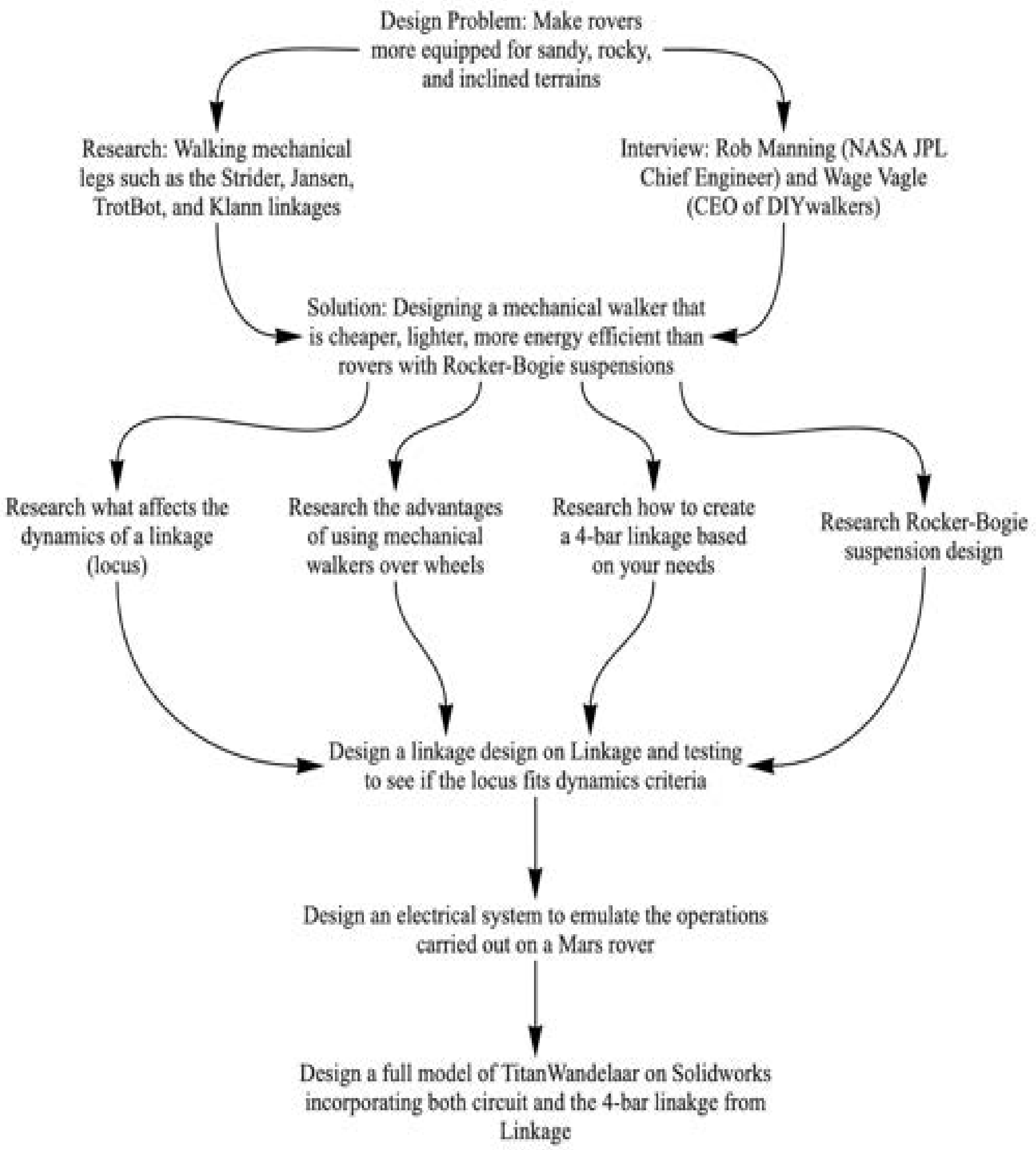
A few more instances shed light on the need for an updated suspension system, one that addresses sandy, rocky, and inclined surfaces. There has been a lack of development in this area and NASA chose to stick with variations of the Rocker-Bogie design due to cost expedience. According to Rob Manning, who is the Chief Engineer at the NASA Jet Propulsion Lab (JPL) in Caltech, they already had the design [the Rocker-Bogie design], they had really accurate models of how it worked, how different variations of it could affect its dynamics, and how it performed on different terrains (Appendix B). This left a large amount of ground for innovation in the types of dynamic systems that could be used on these rovers.

Based on these facts, I have concluded that the main focus here is to create a mechanical system that allows rovers to transverse on both rocky and sandy terrains with any elevation. A few other secondary requirements were added to ensure that the goal of this project was actually noble and could be feasibly executed:

- **Cost**: The cost to manufacture and build a rover using my system has to be within the same price range used to build a rover using the Rocker-Bogie suspension design.
- **Weight**: The structure has to be within the same weight as a normal rover. This is important to ensure that NASA could send this rover to space without needing a shuttle with more fuel or a more advanced shuttle.
- **Material**: The material used to build these linkages has to be widely available to ensure ease of manufacturing and lack of harm to the environment
- **Energy**: The new rover can't use up more energy than the current rovers

In this report, we will discuss my proposed design, the TitanWandelaar, which will address the issues presented by current designs, while adhering to the formed requirements. Through the report, we discuss the rationale we used and we present results to support our decisions.

Our Approach



[3]

User & Major Requirements

Users

TitanWandelaar is meant to be a rover that is designed to traverse planets, moons, and extraterrestrial bodies. Our main users would be any private or public institutions, such as SpaceX or NASA, respectively, who would be open to exploring a more efficient and effective design for a rover that has minimal to no design sacrifices. Though each would be interested in different aspects of the design, we made TitanWandelaar in such a way that it could benefit both private and public institutions.

Private Institutions

Institutions like SpaceX or BlueOrigin focus on cutting-edge innovations while keeping in mind cost-efficiency. These companies have more arduous requirements placed on them by the US Department of Defence, and thus have to undergo a lot of testing. This means that they will undergo a lot of failing, manufacturing, and prototyping. Thus, they need a design that is easy to manufacture and put together and is also cheap since their inventory is limited by the investors' money.

Public Institutions

Institutions like NASA are not as worried about cash flow since they get funding directly from the government, thus, the most cost-efficient option is not necessarily their main pursuit. However, manufacturing a structure more accurately using less man-power would eliminate a lot of errors in the building process. Moreover, they are more inclined to accept a proposal on a design flaw that cost them more than $800 million in the past.

Major Requirements

The major requirements placed have a really simple underlying rule: the new design needs to perform equally as good, if not better than, the current designs we are using for rovers in all categories except for mobility. For mobility, the new design has to perform better than the current design, since this is the main problem we are trying to solve.

Mobility

The new design has to offer a wider range of mobility for Mars rovers. They have to be able to transverse over a wider range of terrains and elevations. The current design seems to allow maneuvering on rocky and inclined terrains but it does not perform as well on coarse or sandy terrains. The new design needs to allow smooth maneuvering on sandy, rocky, and inclined terrains without sacrificing any of the other main functions of the rover.

Manufacturing

Building the new design needs to have a simple manufacturing process. It can not be more arduous than the processes being used right now to build the current rovers. It can not use a technology that is not widespread or hasn’t been explored enough yet.

Cost

Implementing the new design should not cost more than implementing the current ones. The cost endured includes the cost of the materials used to build it, as well as the manpower used to put the rover together. Thus, it can’t use any rare materials that are hard to acquire since that would drive up the cost of manufacturing, and it can’t be too heavy or fragile that it would need more workers putting it together. Usually the development and launch of a Mars rover costs around $700 million.

Material

The main frame can not be build with a material that is not too rare to ensure that there aren’t any major environmental damages or unethical methods taking place to acquire this material

Weight

Current rovers weigh around 2260 pounds [4]. The new design should be just as dense, if not less dense than current rovers to ensure that they are not too heavy to be launched up into space. Being too heavy would also require more work to move the rover, and we want to ensure that the new rover design is more energy-efficient.

Design Concept & Rationale

This section of the report will discuss the decisions made during the design process along with the steps that were taken to implement these design decisions and the various softwares that were used. These decisions were made in close correspondence to our requirements.

Mechanical Walkers

One of the main aspects of this design is the mechanism that the rover will be using to traverse various terrains. Mechanical walkers are really strong candidates. The vast majority of available walker designs allow for mobility on sandy or rocky terrains, or even for both. For example, The Strandbeest uses the Jansen Linkage to maneuver smoothly on sand without moving the actual main body laterally. This gives the illusion that the walker is gliding on sand rather than walking. This is because the locus of the Jansen Linkage has a flat bottom as seen in figure 1, which ensures that the strandbeest stays on the same level (Appendix C).

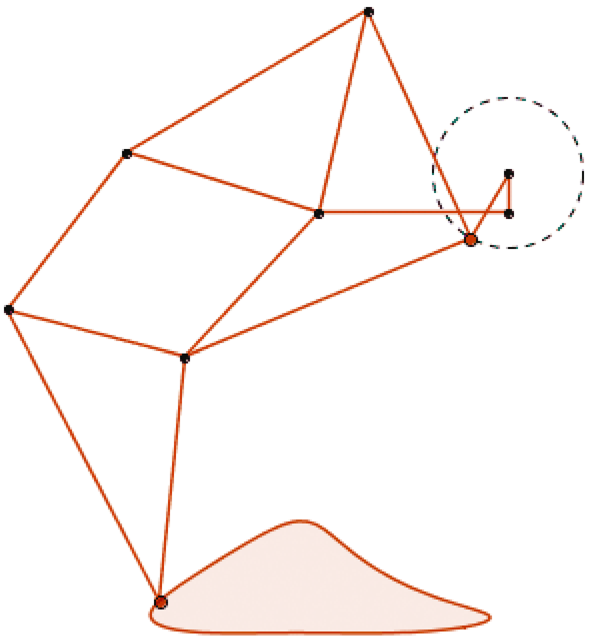


Figure 1: The locus of the Jansen Linkage [5]

Other linkages like the Strider Linkages were meant to provide smooth movement on rocky surfaces.

Figure 2: Strider Linkage on rocky surface [6]

There are many ways you can design a 4-bar linkage to create a crank-rocker movement due to Grashof's Law (Appendix D). Adding various linking bars around the main 4-bar linkage frames can allow you to create different loci as desired. The combinations of different bar lengths that you can have in a 4-bar linkage and how these different lengths combinations affect the behavior of the linkage as a whole is demonstrated in “Analysis of Four-Bar Linkage: Its Applications to the Synthesis of Mechanisms” by John A. Hornes and George L. Nelson, which is an atlas featuring around 700 pages of different bar length combinations.

These countless combinations along with the different shapes that could be created by adding closed links outside of a 4-bar linkage explains how these existing linkages are really different and perform better on different terrains. However, of the well-known linkages that we researched (Klann Linkage, Jansen Linkage, TrotBot Linkage, and the Strider Linkage), none of them allow for mobility on inclined surfaces due to the loci they produce. Thus, the problem here calls for producing a 4-bar linkage with a combination of surrounding frames that result in a locus that allows the linkage to climb on an inclined surface when paired up with an identical leg on the same crankpin journals of the crankshaft.

We drew diagrams and created quick, low-fidelity mockups to try and figure out the possible shape of a locus that would allow a rover to move on a sandy, rocky, and inclined surface. The following shape or variations thereof seemed to be optimal, figure 3. The idea is, the front leg (the left locus in figure 3) would dig slightly into the sand and push that sand behind it, creating a small hill behind it. The second leg then starts high and grabs the top of that small hill and uses it to climb. This design creates grip, even on coarse surfaces, which allows the rover to climb on an inclined surface, even if the angle of inclination is fairly steep.

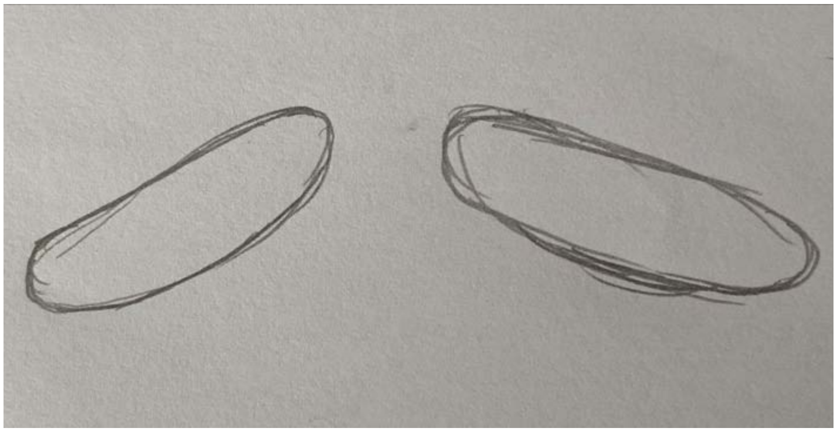


Figure 3: Most Optimal Locus

Then, we had to work backwards to produce the right design that would yield that locus. We used the following expressions [7] to predict the motion of the rocker leg in the crank-rocker 4-bar linkage.

We determined the angular position of using equation (1):

$$\psi = a_1 + a_2 \tag{1}$$

Where a_1 and a_2 are expressed through equations (2) and (3):

$$a_1 = tan^{-1}\left(\frac{sin(\theta)}{C+cos(\theta)}\right) \tag{2}$$

$$a_2 = cos^{-1}\left(\frac{K^2+2Ccos(\theta)}{2BL}\right) \tag{3}$$

Where K and L are expressed through equations (4) and (5):

$$K^2 = 1 + B^2 + C^2 - A^2 \tag{4}$$

$$L^2 = 1 + C^2 + 2C\cos(\theta) \tag{5}$$

Thus, ψ can be simplified as equation (6):

$$\psi = \tan^{-1}\left(\frac{\sin(\theta)}{C+\cos(\theta)}\right) + \cos^{-1}\left(\frac{K^2+2C\cos(\theta)}{2BL}\right) \tag{6}$$

And the angular velocity of the rocker leg was expressed as follows:

$$\frac{d\psi}{dt} = \frac{d\theta}{dt}\left[\frac{1}{L^2}\left(C\cos(\theta) + 1\right) + \frac{C\sin(\theta)}{S^2}\left(2 + \frac{M^2}{L^2}\right)\right] \tag{7}$$

Where M and S are:

$$M^2 = K^2 + 2C\cos(\theta) \tag{8}$$

$$S^2 = \sqrt{4B^2L^2 - M^4} \tag{9}$$

Taking the second derivative of ψ yields the angular acceleration of the rocker leg, which further defines the movements of the rocker leg and gives us a better prediction of the system's dynamics:

$$\begin{gathered}\frac{d^2\psi}{dt^2} = \frac{d^2\theta}{dt^2}\left[\frac{1}{L^2}\left(C\cos(\theta) + 1\right) + \frac{C\sin(\theta)}{S^2}\left(2 + \frac{M^2}{L^2}\right)\right] + \\ \left[\left(2 + \frac{M^2}{L^2}\right)\left(\frac{2C^2\sin^2(\theta)(2B^2-M^2)}{S^6} + \frac{C\cos(\theta)}{S^2}\right) - \frac{2C^2\sin^2(\theta)}{L^2S^2}\left(1 - \frac{M^2}{L^2}\right) - \frac{C\sin(\theta)}{L^2}\left(1 - \frac{2C\cos(\theta)+1}{L^2}\right)\right]\left(\frac{d\theta}{dt}\right)^2\end{gathered} \tag{10}$$

The parameters for the angular position, angular velocity, and angular acceleration of the rocker leg of the crank-rocker 4-bar linkage were taken from figure 4:

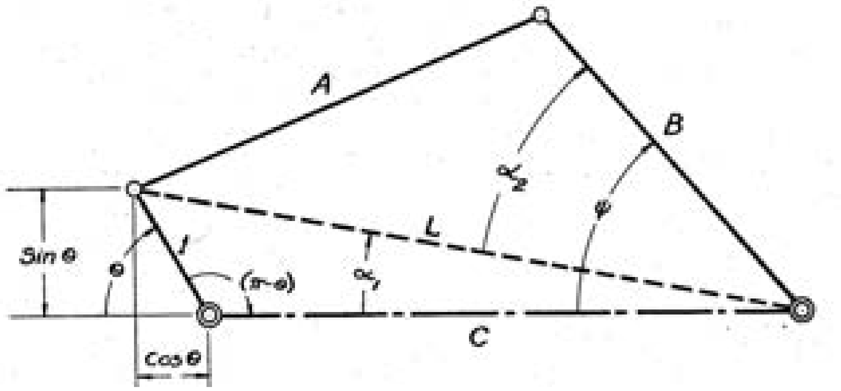


Figure 4: Motion of Rocker Leg of Crank-Rocker 4-Bar Linkage [7]

Using these equations along with a special software called Linkage, which was designed to simulate the motion of linkage designs (Appendix E), we were able to develop the following design for one leg, figure 5:

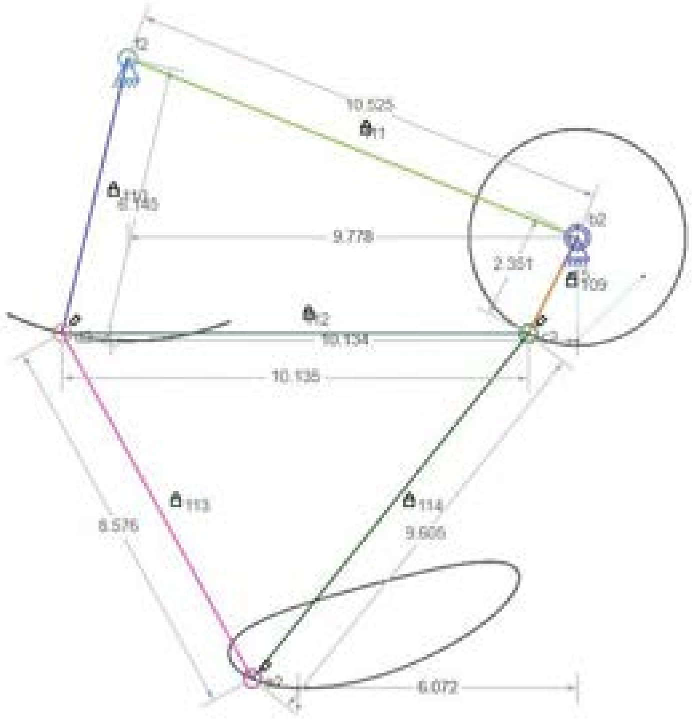


Figure 5: Single Leg of TitanWandelaar

And the following design for a pair of legs that are connected by a crankpin journal:

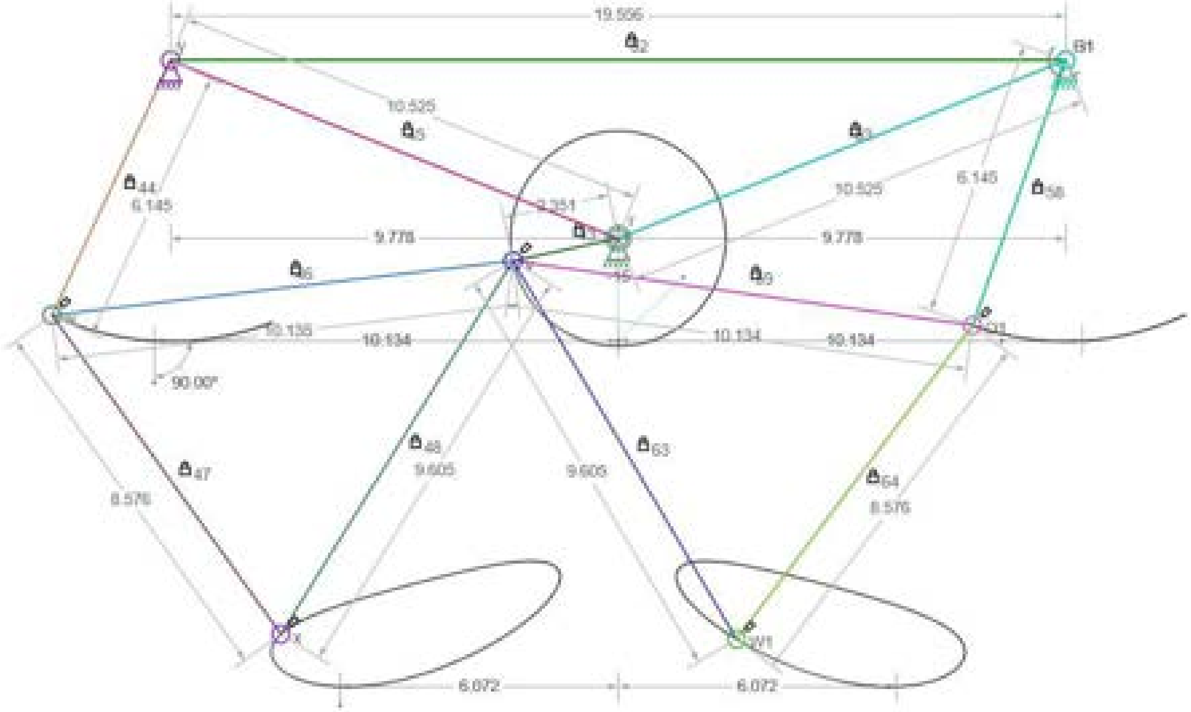


Figure 6: A pair of legs of TitanWandelaar sharing a crankpin journal

A similar model, combining these two approaches more accurately, was modeled on Mathematica and represented the Jansen Linkage. In Appendix F, We present Sandor Kabai's Mathematica notebook, which features the effects of changing the side lengths on the locus created by the Jansen Linkage, which is the more rigorous approach [8].

The main reason we chose the given lengths to the 4-bar linkage is to create a crank-rocker movement. Different dimensions could've yielded a crank-crank or a rocker-rocker movement. This crank-rocker movement allows us to grip onto something and then push off it, which is ideal for our application. The two additional bars were the ones that got experimented the most to create our desired locus.

We found the most optimal locus, figure 3, using low-fidelity models, some Mathematica code, and some intuition about what a pair of legs need to climb an incline. We then used a guess-and-check approach to get the bar lengths for the triangular piece of the leg. We kept experimenting with different combinations until we got really close to that optimal locus that we desired.

Electrical System

Mars rovers are mounted with batteries, cameras, motors, temperature controls, inverters, antennas, microphones, humidity sensors, and many more electrical components. These electrical systems' main focus is to detect the environment and communicate with Earth about their findings. To emulate these electrical systems, we designed one to be mounted on the TitanWandelaar that masks some of the functions carried out by Mars rovers.

For simplicity and ease of applicability and accessibility, our electrical system consists of an Arduino, which is a microprocessor (the main brain of the electrical system), a motor to rotate the crankshaft and thus move the legs, a temperature sensor accompanied by a display, and a humidity sensor accompanied by an extremity indicator.

Our electrical system will be presented below and it was designed on TinkerCAD. The schematic of the electrical system was also exported from TinkerCAD. The code that controls the system was written for the Arduino IDE (Appendix G).

While closed, the system looks like this:

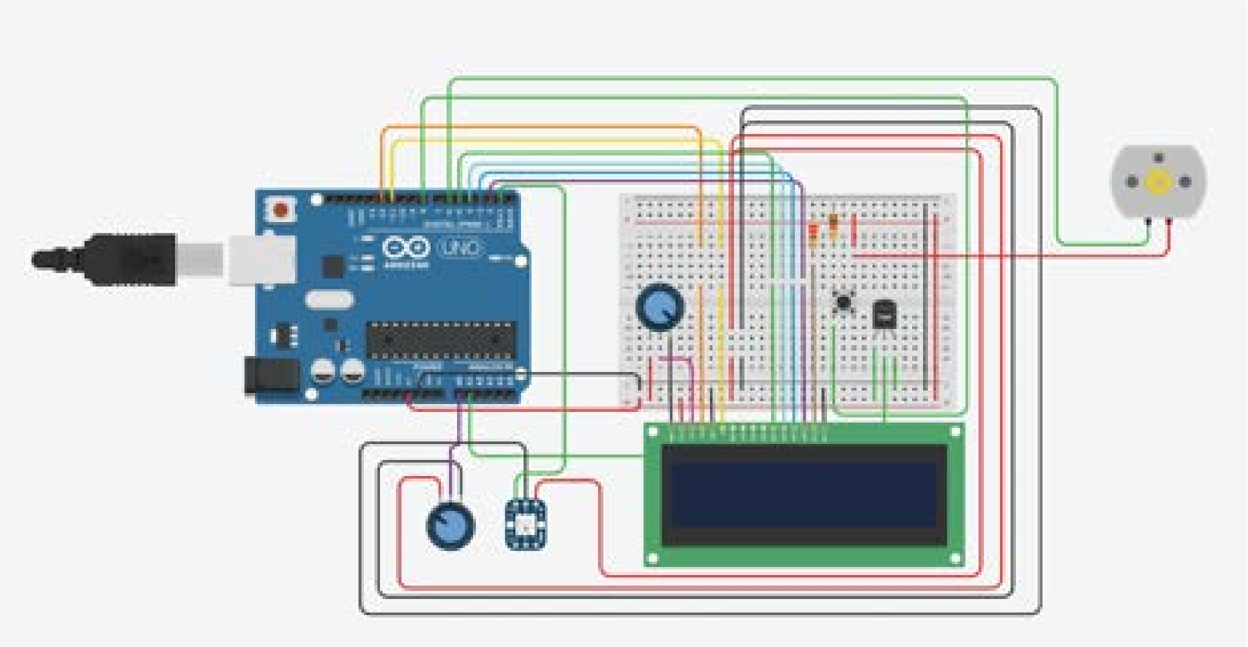


Figure 7: Electrical system in component form

The following schematic represents the system:

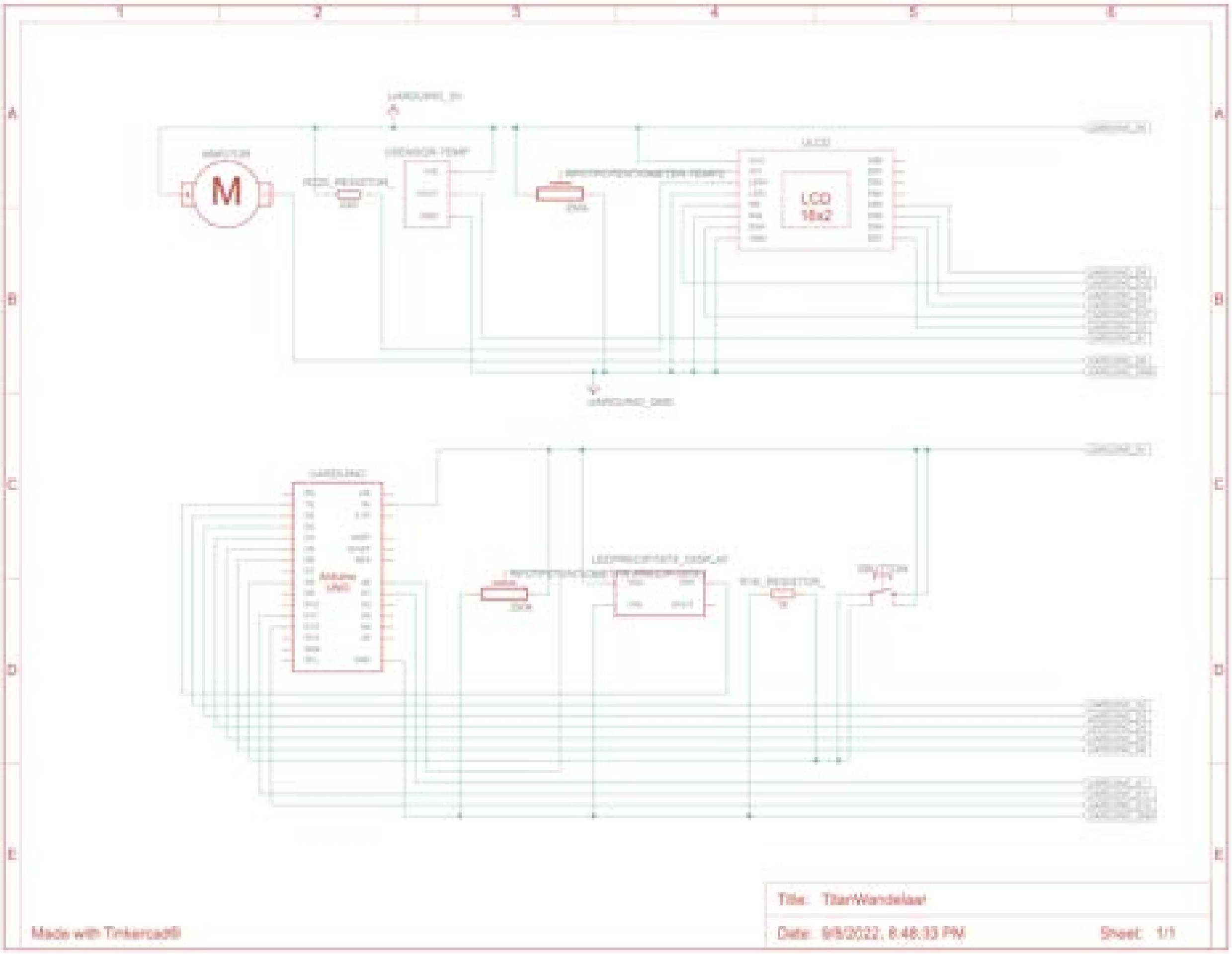


Figure 8: Schematic of Electrical System

Motor

The motor can be controlled easily using an on/off switch. A more complex model would include an inverter and multiple speeds in order to demonstrate a more accurate representation of a Mars rover, but due to the limitations of TinkerCAD, we had to pick a more simple and straightforward model.

The motor is connected in series with the Arduino. When the button is in its off state, the circuit is opened and the motor's shaft does not rotate, as seen in figure 9:

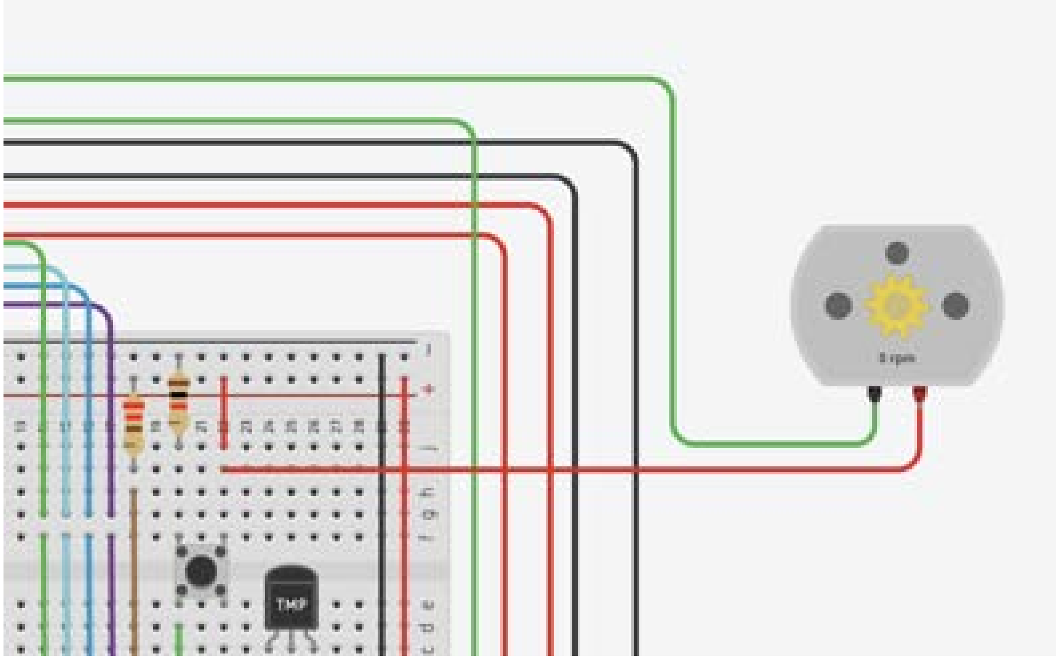


Figure 9: Motor in its "off" state

When the button is pressed, the circuit is closed, and the motor turns on:

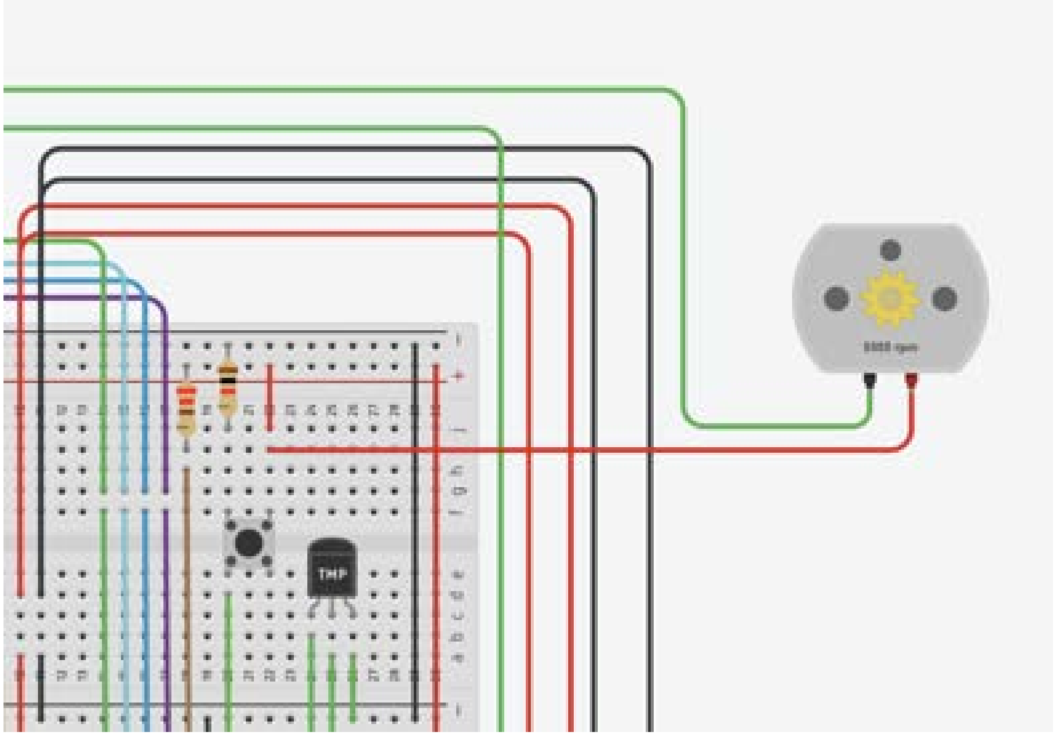


Figure 10: Motor is in its "on" state

In the case of this model, the motor's highest speed is 5555 rotations per minute (rpm)

Temperature Detection

To detect the temperature, we have a potentiometer that controls the “on” and “off” states of the sensors, or when the circuit will be complete or not. The temperature sensor detects the temperature. This value is then read on the display in degrees Celsius.

The following two figures represent how the temperature detection subsystem works. As you slide the temperature, the display reads off the detected temperature:

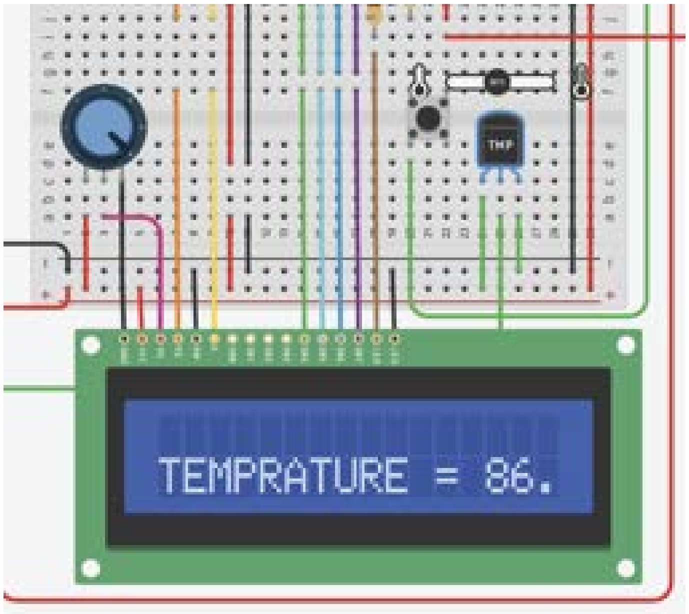


Figure 11: Temperature detection system at $86^{o}C$

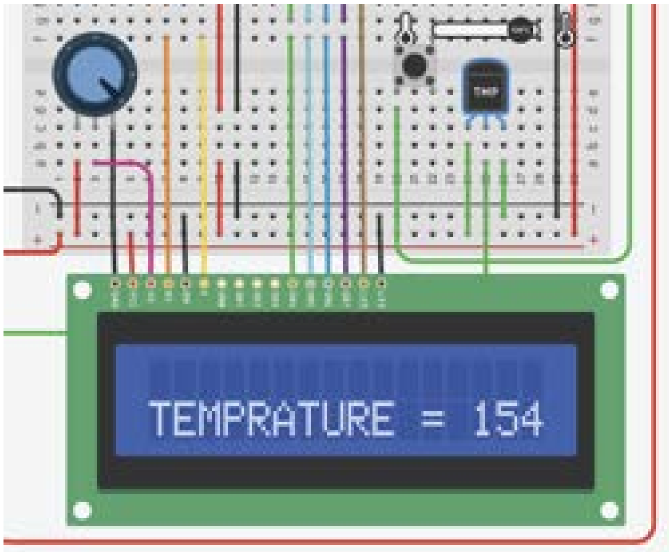


Figure 12: Temperature detection system at $154^{o}C$

Humidity Detection

The humidity detection system works a little differently than the temperature detection system. Instead of using an LCD display, we use a NeoPixel, which is an LED light that produces different colored lights. The potentiometer detects the amount of humidity, based on how much voltage is passing through it and passes that signal through to the Arduino, which then sends that signal to the NeoPixel.

If the voltage through the potentiometer is at its highest, this means that the conditions are really humid, and thus the NeoPixel lights up blue:

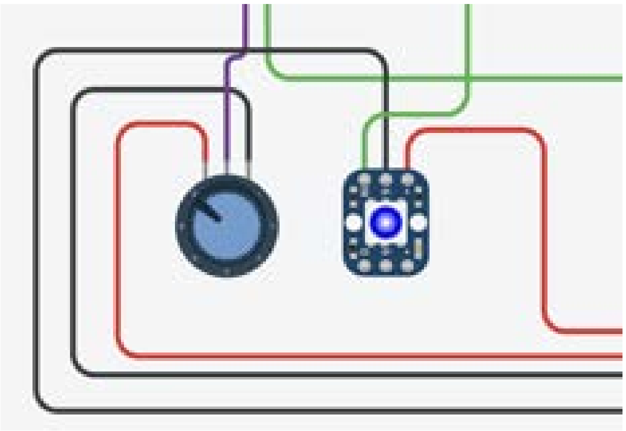


Figure 13: Humid/wet conditions

If there is no voltage through the potentiometer, this means the conditions are dry and thus the Neopixel lights up red:

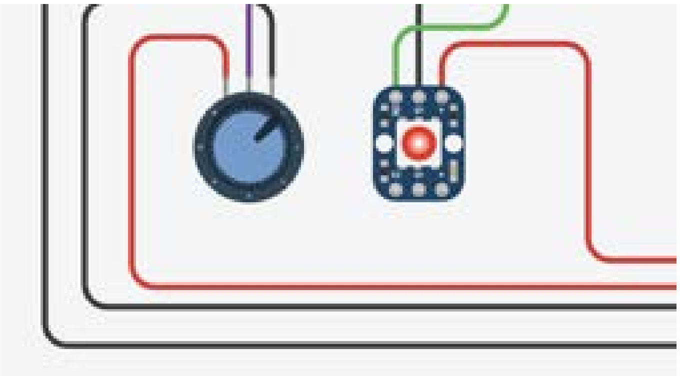


Figure 14: Dry conditions

If the voltage through the potentiometer is right between its maximum and minimum readable values, the Neopixel will light up green since the conditions are neither wet nor dry:

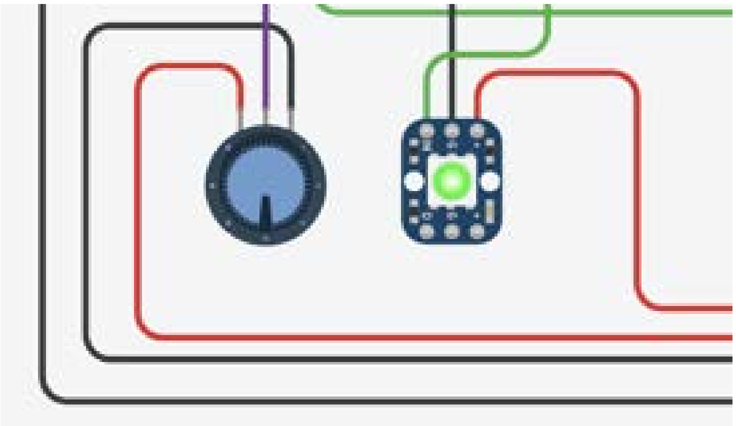


Figure 15: Neither dry not wet

Moreover, since these changes in humidity levels are gradual and not abrupt, the lights change colors a little more gradually, so the NeoPixel is not only blue when it is only at its maximum, but when it is also considerably humid. And the NeoPixel is not only red when it is only at its minimum, but also when the conditions are considerably dry.

Solidworks Model

We then put together a complex Solidworks assembly to present how the whole system will work together with multiple pairs of legs along with the electrical system. In the Solidworks model, we also include gears to be able to animate this model and collect accurate data from it.

In the model, we place the electrical components in an electronics container, however, we omit connecting these electrical components in Solidworks since it does not impact the effectiveness of the model or any animations, and we already demonstrated all the different features of the electrical system on TinkerCAD.

In this section, we will walk through the Solidworks designing process and explain the rationale behind some of our decisions. More details about each part individually and the mates used can be found in Appendix H

For the first part of the assembly, we will need the following parts:

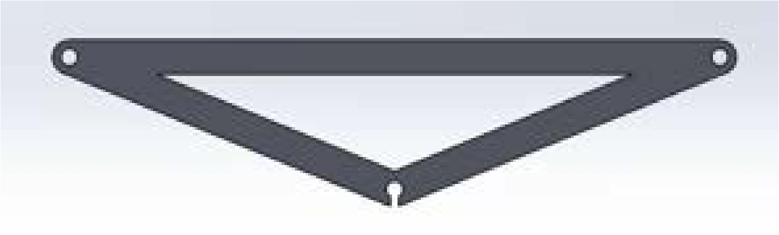


Figure 16: Main frame (part)

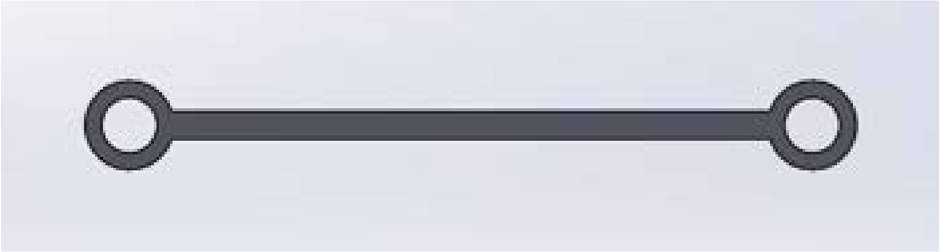


Figure 17: 6.145” leg section (part)

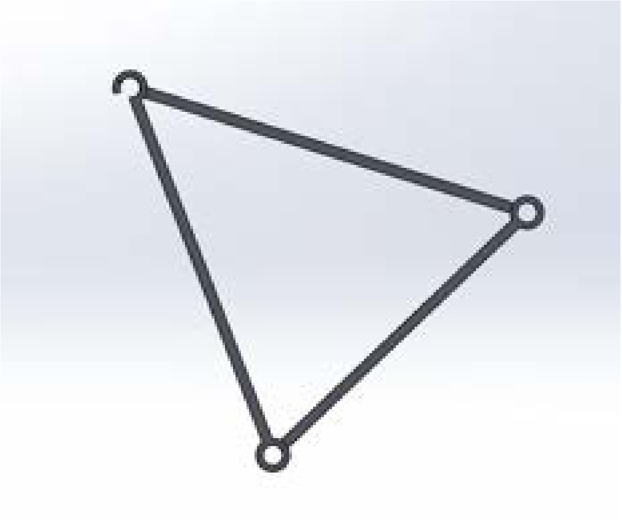


Figure 18: Assembly of the bottom triangle of the leg

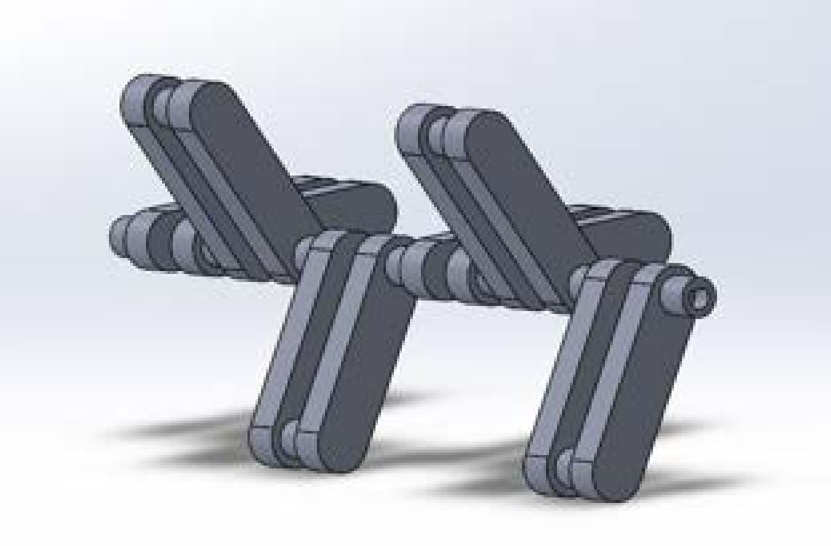


Figure 19: Crankshaft

Note that the crankshaft has 6 crankpin journals, which means that there will be 6 pairs of legs. Also, it has a D-shaped dowel hole to which the corresponding D-shaped dowel pin of the gear could fit into it and rotate it.

We then assemble these components as such to create the first leg:

Figure 20: Assembly with one leg

By adding another leg to the other side of the same crankpin journal, you will effectively create the first pair of legs:

Figure 21: First pair of legs

By repeating these past two steps, you can create all 6 pairs of legs:

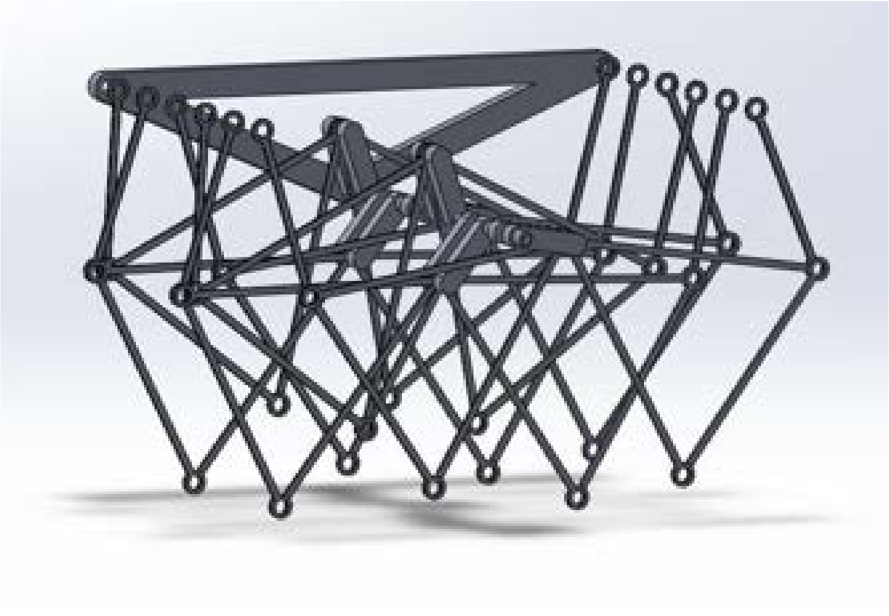


Figure 22: All legs assembled

We then add in the main frame to the other end of the crankshaft and we also add a mainframe to the main journal on the crankshaft in such a way that each pair of leg is sandwiched between two main frames:

Figure 23: All main frames added to assembly

We then add two rods to the assembly to hold all the main frames and legs together and to prevent unwanted movement since the top part of the 6.145” leg needs to be anchored for the 4-bar linkage to work as intended:

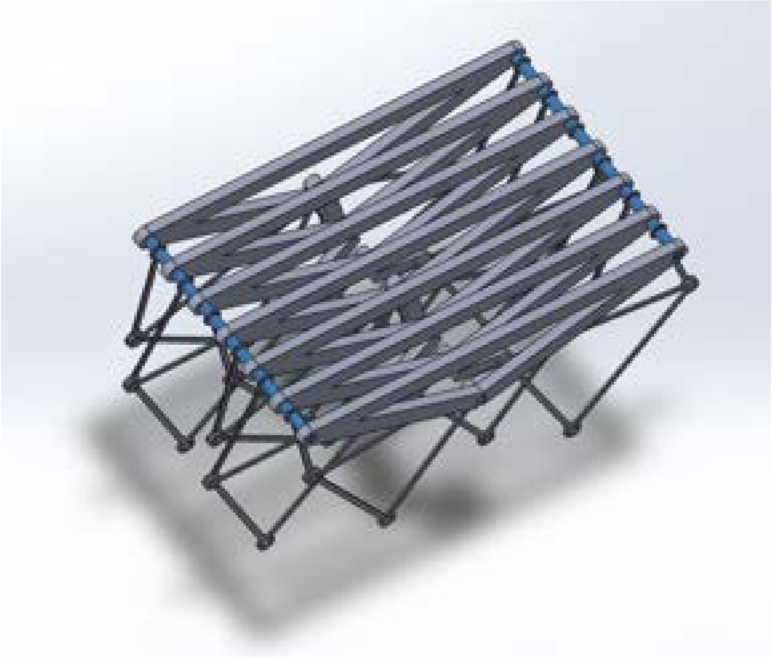


Figure 24: Two rods added to assembly

We then add the electronics box on top of the main frame and anchor it using 4 screws. Note that the main frames are intentionally flat to provide a surface to mount the electronics box on. Also note that the electronics box has two holes, a circular hole outside the motor shaft, and a rectangular hole to the front of the box.

These holes are to allow us to connect the gear to the motor shaft when the top of the box is closed and the rectangular hole is to allow us to read the temperature even when the lid is closed:

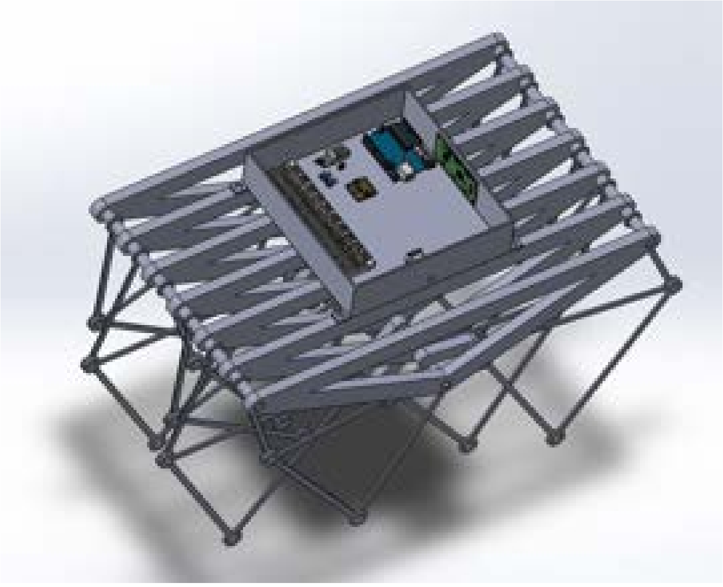


Figure 25: Electronics box assembled to the main frame

We then add a lid to house and protect all the electronics safely and to prevent from the harsh environment of the extraterrestrial body the rover is on:

Figure 26: Lid added, labeled “TitanWandelaar”

We then attach a small gear, of radius 0.5" to the crankshaft. The D-shaped dowel pin of the gear is the same exact size as the D-shaped dowel hole in the crankshaft to ensure snug fit:

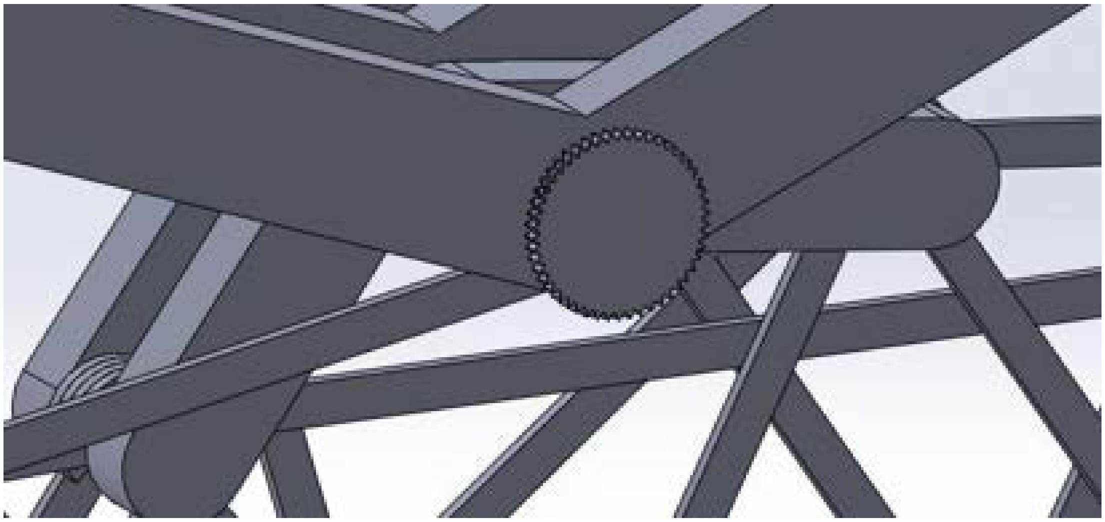


Figure 27: Small gear attached to the crankshaft

We then attach the bigger gear to the motor and ensure that its teeth are intertwined with the teeth of the small gear. We removed a lot of the volume of the bigger gear to get rid of unnecessary weight:

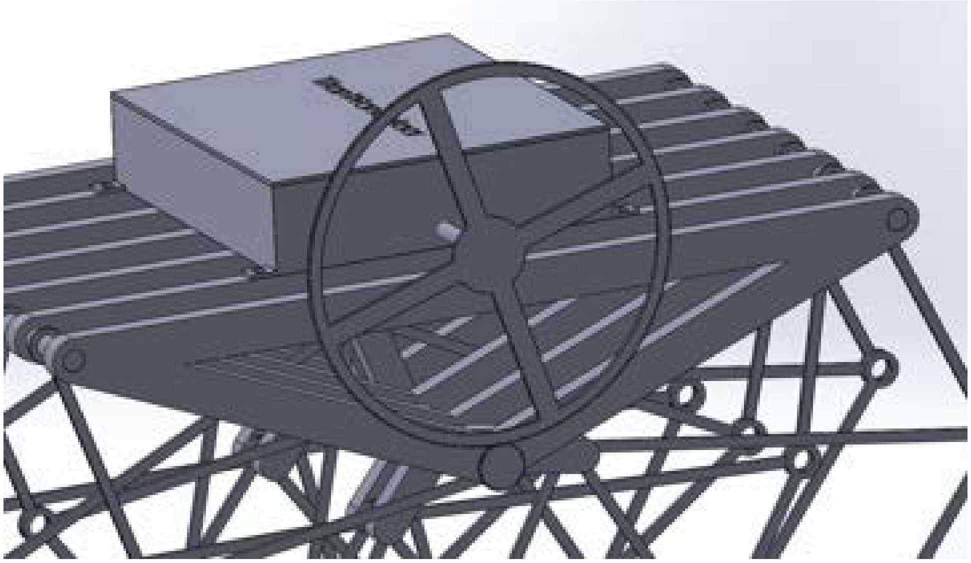


Figure 28: Bigger gear attached to the motor

We purposely make the gear that is attached to the motor significantly bigger, around 8 times bigger, to create a mechanical advantage and to move the legs more times per a motor shaft revolution. By doing this, we get around using a more powerful motor.

To make sense of this, consider a small movement of the big gear. This movement would cause the small gear to complete lots of rotations, and thus move the crankshaft more times. However, if we flipped this relation, the motor shaft would have to rotate more times for the bottom gear to complete one rotation.

The completed, full assembly is presented in figure 29, where we also add a solar panel as a power source for the electrical components in the electronics box:

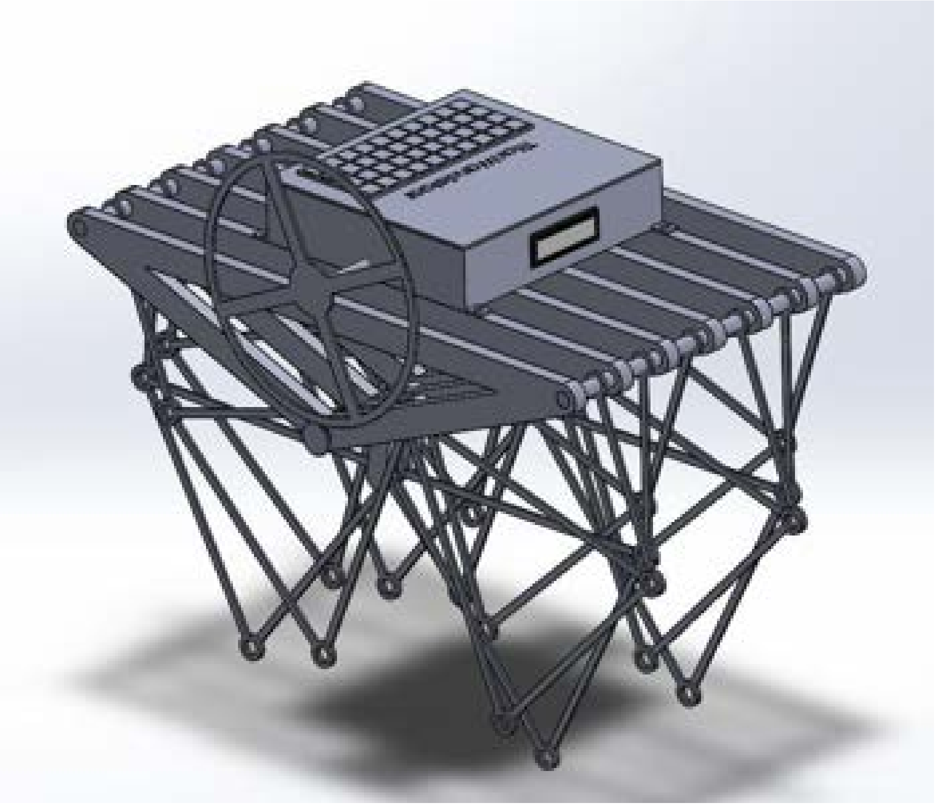


Figure 29: Full assembly of TitanWandelaar

Rationale Behind Mechanical Walkers

Mechanical walkers address two major problems with the Rocker-Bogie design that is used on all current Mars rovers. The first one has been mentioned exhaustively before, which is, being stuck in sand. Though the legs are pointy and intuitively would cause someone to think that their low surface area makes it easy for them to sink in the sand, the way the legs move in unisense along with the movement of each leg with respect to the sand makes it hard for the legs to get stuck in sand traps as what happened with the Spirit rover in 2009

Figure 30: Final images from the spirit rover after it got stuck in a sand trap [9]

Another problem that started showing up recently was wearing out of wheels due to the rugged Mars terrain. Recent reports about the Curiosity rover, which was sent to Mars in 2011, surfaced showing how heavily corroded the wheels of the rover were.

Figure 31: Wheels of Curiosity rover damaged due to rugged terrain [10]

Mechanical walkers address this problem in two different ways. First, these wheels need to be made out of thin metal so they could be malleable enough to curve into a wheel shape, which means they are weaker. The “feet” of the mechanical walkers are not made of the same material as the entire body, it does not have to be thinner or more malleable, which makes it stronger than the point of contact in current rovers.

Moreover, current rovers have 6 wheels. Losing one wheel makes it effectively dysfunctional since it can’t maneuver anymore. However, TitanWandelaar’s has 6 pairs of legs, which means 4 pairs can get damaged and it would still be able to function since all it needs is 2 functioning pairs of legs.

Limitations & Directions for Future Development

Due to the time and budget constraints that were allotted to this project, there were a lot of steps that could have improved the quality of this design that had to get cut out.

One of the initial goals that were set for this project was to end up manufacturing a prototype by 3D printing using plastic filaments (Appendix I) and then testing how it behaves in the real world under non-negligible forces such as gravity and wind and friction, since models are do not take these into account and thus, are not always the most accurate representations of a work. So we wanted to compare our experimental data with our expected data to see how well our design performed.

Moreover, constructing a small testing rig where we can simulate multiple forces like gravity, friction, and storms would provide valuable information that could be used to improve the design. Softwares like Solidworks or Ansys are better at simulating vector forces in one direction. So a force produced due to a storm or a tornado can not be simulated on such softwares. We do need to account for such forces when designing a full scale model and thus a testing rig would be a necessary step in a more rigorous design process where time isn't a limitation.

We would then improve the design, run Finite Element Analysis (FEA) and topology optimization to get rid of any structural weight that may not be necessary for the rigidity or the strength of the structure and metal 3D printing it using Additive Manufacturing (AM) which would be the way we would advise to manufacture this rover if it were to be built on a larger scale. This would save a large amount of materials and would save a great deal of weight without sacrificing the structural integrity of the rover. The material we would use would be Ti-6Al-4V, since it is easily accessible, easy to 3D print, light, and strong (Appendix J).

A few other details could be improved about the overall design. For example, as Wade Vagle, the CEO of DIYwalkers mentioned in one of our correspondences (Appendix K), If the main object is to climb inclined surfaces, then we need to make sure that the center of gravity of the rover is as low as possible, which could be achieved by assembling 3 pairs of legs, having blank some space to place the electronics box lower, and then adding the 3 pairs of remaining legs.

Also, since this rover is meant for extraterrestrial bodies with sandy and rocky surfaces, errorsion will be a factor that we need to consider (Appendix K). Since this design relies on mechanical walking mechanisms, there are a lot of joints, which means all of these joints need to be well preserved from sand and pebbles getting into them. One way this could be addressed is by adding a shrink wrap around all the joints. This will allow for flexibility and protection at any given point during the motion of the legs.

Another limitation we had was the inability to produce a code that picks the right combinations of 7 bars of a leg based on the desired locus, which is the approach that Theo Jansen did. This is also similar to the approach outlined by the Mathematica code in Appendix F. This would be a more accurate method than just guessing a close enough combination.

Another consideration that could drastically improve the performance of the linkage and the rover as a whole on inclined sandy and rocky surfaces is to consider a more intricate feet design. We considered having the feet have a shovel shape to scoop up the sand. However, this may cause some sand to remain in the bottom of these mini shovels and thus throw the whole structure's balance off. Thus, we thought that we should not just add a shovel shape just out of intuition without testing it or simulating it beforehand. But definitely considering the foot shape affects the movements of a mechanical walking linkage (Appendix C).

If a grant was received for this project, along with a long timeline, these goals could be accomplished. After that, the structure should be put under scaled strenuous testing to prove its integrity and functionality. When it passes these tests, this proposal could then be pitched to private and public institutions that are involved in building rovers.

Conclusion

This report and proposal address existing issues with the current suspension model, the Rocker-Bogie design, that is being used on all the Mars rovers but provided an alternate mechanism that these rovers can use to transverse a wider array of terrains and inclinations.

The proposed design costs less to manufacture and operate, since it won't get stuck in sand traps and cost institutions the entire price tag of the rover, as in the case with the Spirit rover, which cost NASA around $800 million. It also is more energy efficient to operate mechanical legs as opposed to wheels. The proposed material to construct the frame would be Ti-6Al-4V, which is not too rare that it would cause any significant environmental damage and could be easily 3D printed through Additive Manufacturing. By running FEA and topology optimization, we can further eliminate unnecessary weight to ensure that it is much lighter than existing models.

Another issue that this design addresses is possible longevity of the mission time of these rovers. Recent reports have shown the wheels of the curiosity rover which have been severely "eaten up" by the martian terrain. If one of these wheels become dysfunctional, the whole rover will be out of balance and will not be able to drive on the rugged terrain anymore. However, this is not much of an issue with the TitanWandelaar. This is because having 12 legs allows for a lot of room for error and damage. As a matter of fact, only 2 out of the 6 pairs of legs need to work for the rover to move. This wider safety margin would be advantageous when on an unexpected, rugged terrain.

References

[1] M. W. published, “Rugged Mars has taken big bites out of the Curiosity rover’s wheels (photos),” Space.com, Feb. 08, 2022. https://www.space.com/nasa-mars-rover-curiosity-wheel-damage-photos (accessed Sep. 09, 2022).

[2] “Rocker-bogie,” Wikipedia. Jul. 04, 2022. Accessed: Sep. 09, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Rocker-bogie&oldid=1096398935

[3] “Sign up | Miro | Online Whiteboard for Visual Collaboration,” Miro. https://miro.com/signup/ (accessed Sep. 09, 2022).

[4] mars.nasa.gov, “Learn About the Rover.” https://mars.nasa.gov/mars2020/spacecraft/rover/ (accessed Sep. 09, 2022).

[5] “In the left, the original Jansen’s mechanism with... | Curiosidades no geométricas | Mechanical design, Jansen, Kinetic art,” Pinterest. https://www.pinterest.com/pin/229050331020915086/ (accessed Sep. 09, 2022).

[6] Strider Robot v3 on the Rocks. Accessed: Sep. 09, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=K_7Hy7oulDQ

[7] J. A. (John A. Hrones and G. L. Nelson, Analysis of the four-bar linkage; its application to the synthesis of mechanisms. Cambridge: Published jointly by the Technology Press of the Massachusetts Institute of Technology, and Wiley, New York, 1951.

[8] “A Theo Jansen Walking Linkage - Wolfram Demonstrations Project.” https://demonstrations.wolfram.com/ATheoJansenWalkingLinkage/ (accessed Sep. 09, 2022).

[9] “Will This Mars Rover Ever Rove Again? Spirit Gets Stuck in the Sand,” Discover Magazine. https://www.discovermagazine.com/the-sciences/will-this-mars-rover-ever-rove-again-spirit-gets-stuck-in-the-sand (accessed Sep. 09, 2022)

[10] M. W. published, “Rugged Mars has taken big bites out of the Curiosity rover’s wheels (photos),” Space.com, Feb. 08, 2022. https://www.space.com/nasa-mars-rover-curiosity-wheel-damage-photos (accessed Sep. 09, 2022).

[11] *Design of Rocker-Bogie Mechanism - IJISRT*. https://ijisrt.com/wp-content/uploads/2017/05/Design-of-Rocker-Bogie-Mechanism-1.pdf.

[12] “Leg mechanism,” *Wikipedia*. Jun. 01, 2022. Accessed: Sep. 10, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Leg_mechanism&oldid=1090999901

[13] “Strandbeest.” https://www.strandbeest.com/evolution (accessed Sep. 10, 2022).

[14] SQB, “What is the significance of the numbers in ‘The Eleven Holy Numbers of the Mechanical Soul’?,” *Science Fiction & Fantasy Stack Exchange*, Sep. 18, 2016. https://scifi.stackexchange.com/q/139557 (accessed Sep. 10, 2022).

[15] "Strider Linkage Plans," *DIY Walkers*. https://www.diywalkers.com/strider-linkage-plans.html (accessed Sep. 10, 2022).

[16] "Now Scientists Can Accurately Guess The Speed Of Any Animal," *Animals*, Jul. 17, 2017. https://www.nationalgeographic.com/animals/article/Animal-speed-size-cheetahs (accessed Sep. 10, 2022).

[17] "Klann linkage," *Wikipedia*. Aug. 30, 2022. Accessed: Sep. 10, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Klann_linkage&oldid=1107565633

[18] *TJ-leg prototype*. Accessed: Sep. 10, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=i5Z6y9A5kCg

[19] "Mechanical Behaviour of Additively Manufactured Cellular Structures." *Metal Additive Manufacturing*, 13 Aug. 2020, https://www.metal-am.com/articles/modelling-the-mechanical-behaviour-of-3d-printed-cellular-structures/.

[20] Nazir, Aamer et al. "A State-of-the-Art Review on Types, Design, Optimization, and Additive Manufacturing of Cellular Structures." *International journal of advanced manufacturing technology* 104.9-12 (2019): 3489–3510. Web.

Appendices

Appendix A: Rocker-Bogie Suspension

NASA has been using the same suspension system on Mars rovers since the very first one due to its reliability and simplicity. The suspension system is called the rocker-bogie design and it was designed in 1988 by NASA JPL engineer Don Bickler.

This suspension system mainly works without the need of any springs and it allows rovers to climb inclined surface and navigate around rocks and other rugged surfaces

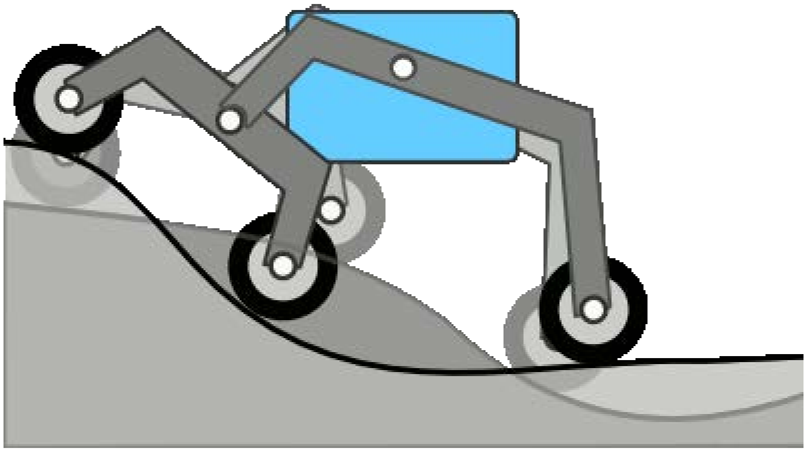


Figure 32: Rocker-Bogie on uneven surfaces [11]

The term “rocker” describes the rocking aspect of the larger links present on each side of the suspension system and balances the bogie as these rockers are connected to each other and to the vehicle chassis through a modified differential. [11]

In the system, “bogie” refers to the conjoining links that have a drive wheel attached at each end. Bogies were commonly used to bear loading as tracks of army tanks as idlers distributing the load over the terrain. Bogies were also quite commonly used on the trailers of semi trailer trucks as that very time the trucks will have to carry much heavier load. [11]

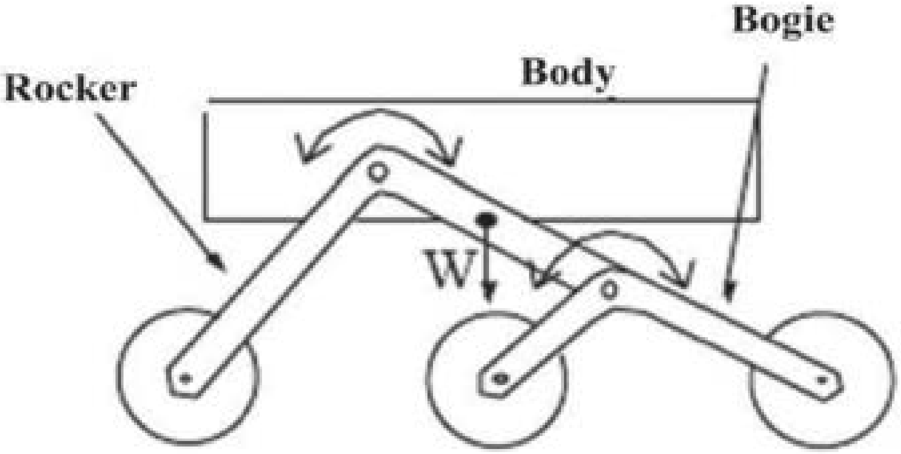


Figure 33: Rocker-Bogie labeled diagram [11]

The rocker-bogie design has no springs or stub axles for each wheel, allowing the rover to climb over obstacles, such as rocks, that are up to twice the wheel's diameter in size while keeping all six wheels on the ground. As with any suspension system, the tilt stability is limited by the height of the center of gravity. Systems using springs tend to tip more easily as the loaded side yields.

Some of the main features of the Rocker-Bogie design as mentioned in the "Design of Rocker-Bogie Mechanism" which was posted in the International Journal of Innovative Science and Research Technology [11]:

1. The mechanism allows it to climb over high obstacles, while keeping all the six wheels in contact with the ground. This is only true at the operational speeds of rovers like Curiosity which is around 10 cm/s.
2. The two sides (left and right) move independently, and hence the rover can traverse terrains where the right and left rockers go over different types of obstacles.
3. The mechanism is designed such that due to the independent motion of right and left rockers, the pitching of the chassis or the rover body remains an average of the two rockers.
4. Systems with spring suspensions are susceptible to tip-over sideways more easily than rocker-bogie. Curiosity, by design, can sustain over 50 deg tilt in any direction.
5. The design incorporates independent motors for each wheel. There are no springs or axles, making the design simpler and more reliable.
6. The design reduces the main body motion by half, compared to any other suspension. The jerk experienced by any of the wheels is transferred to the body as a rotation via the differential connecting the two rockers, not as translation like conventional suspensions.

The Rocker-Bogie design was designed to fit the following criteria for the Mars Exploration Rover [11]:

1. Stow in an extremely small space and deploy the mobility into a stance that would provide the rover with 45 degree stability.
2. Absorb a large percentage of the impact loads the rover would experience during lander egress and surface traverse.

Appendix B: Correspondence with Rob Manning

This appendix highlights all the email correspondences that happened between me and the NASA JPL Chief Engineer, Rob Manning.

Greetings Mr. Manning,

I hope you are doing well. My name is Youssef Abdelhalim and I am the US Presidential Scholar that Merri Anne Stowe talked to you about.

I first want to thank you for agreeing to do this and that it is truly an honor to be talking to someone who has helped send rovers to Mars.

I am a rising sophomore at Northwestern University and I am studying Mechanical Engineering with a concentration in Aerospace Engineering. I am currently working on a personal project to keep myself busy during the summer. My project focuses on improving the design of rovers, especially ones that operate on bodies with coarse inclined terrains. My approach was using 4-bar linkages to create mechanical walkers. This would prevent incidents such as the one where the Spirit rover got stuck in quicksand and went out of service and would make it easier to maneuver on sand and around obstacles such as rocks. Also, I have seen multiple research papers showing how mechanical walkers are more energy efficient than wheels.

I was reaching out to get some feedback on my idea, and more importantly, I was wondering why NASA stuck with the 6 wheel design for all its rovers. Is there a reason NASA has never tried walkers? Was it something that was considered in the past and got scrapped for some reason?

If you have some time, we can discuss this on this email thread or possibly meet over Zoom if you are interested.

I would once again like to thank you so much for your time and consideration.

Sincerely,
Youssef Abdelhalim.

Hi Youssef!

Great to virtually meet you. Congratulations on being a Presidential Scholar!!!!

I can answer some of your questions but not all of them.

Are the 4 bar linkages you are talking about are similar to those that Theo Jansen developed for his Strandbeests? (See https://www.strandbeest.com We invited him to JPL to talk and I was able to spend time with him over dinner. Very interesting, creative and talented person.)

The reason we selected Pathfinder's Sojourner rover's Rocker-Bogie design (designed by Don Bickler … another interesting and talented person … now retired) is because it worked well from a static stability and mobility (and very low power) point of view. It's kinematic design without stored energy has very low losses. While we don't use it that way now, the Rocker-Bogie design allows a (any) wheel to be able to drive over a rock one wheel diameter in height. I have not seen a mobility / stability comparison with multiple 4-bar linkages (walkers) but maybe some at JPL have. So I can't answer that part of your question. I can ask though. I am far from being the expert.

The reason we have stuck with variations on the Rocker-Bogie design on subsequent rovers (MER, MSL, M2020) was mostly because of cost expedience. We had the design, we had models of how it worked and how to morph it into different sizes and model its performance on different terrain classes. It was easy to do cost and risk comparisons. Does that help?

Do you have references for walkers that you are thinking about?

-Rob

Hello Mr. Manning,

I would first like to apologize for taking forever to respond. I was going through a family emergency and had to step back from some of my projects for a week or two until everything was sorted out.

Actually, yes! My love for 4-bar linkages was inspired by Theo Jansen. I have been following his work for the past few years and I found his designs to be really marvelous and elegant. I knew that NASA invited him over to speak about his designs through one of his interviews with BBC, but I was never able to find any sort of recording of that anywhere on the internet which is unfortunate.

His Strandbeests are moreso designed to walk on flat surfaces, while mine are meant more for inclined surfaces, which is something that none of the currently well-known walkers do (like the Klann or the Strider, or the TrotBot).

I did look into the Pathfinder's Sojourner rover Rocker-Bogie design and if I am being honest, it is truly a marvel of engineering. It is truly elegant and clever. I personally did not know it was low power. From my research, I found out that mechanical walkers can usually save up to 25% more energy than normal wheels. But this is in reference to normal car wheels, not Rocker-Bogie design.

It would be great if you can connect me to someone that has possibly explored comparing the stability and mobility of mechanical walkers compared to wheels or specifically the Pathfinder's Sojourner rover Rocker-Bogie design.

The walker I am talking about is called the TitanWandelaar (Titan refers to Saturn's moon, and I thought this was appropriate since Titan is sandy and hilly, which is the perfect surface for my walker. Mars and Venus work well too but I was trying to make the name sound more Sci-Fi. Wandelaar is the Dutch word for walker and I chose Dutch in honor of Theo Jansen since he inspired me a lot as a kid) and it was designed by me. I will attach a recording to this email to show you how it works. The front leg (the one on the left) basically pushes the sand behind and creates a tiny hill behind it, the back leg then grips on that mini hill and pushes on it to allow it to climb over inclined surfaces. These mini hills are basically the things that allow them to do that, so the locus of the TitanWandelaar was designed specifically to create a mini hill behind it. All the walkers that I have encountered end up tracing a nice, smooth surface behind them, which makes them unsuitable for climbing on inclined surfaces.

Again, sorry for the delayed response and thank you so much for your precious time.

Sincerely,
Youssef Abdelhalim.

Hi again Youssef,

I'm sorry about your family emergency, I hope all is well now.

I really am impressed with your design! I think I see the idea you are aiming for with the non linear motion of the feet trajectories compared with Theo's which allow for non-planar mobility.

Have you read about JPLs Athlete mobile platform? It’s a really a multi-DOF set of arms with wheels on each. (I have no idea about it’s performance compared with linkage kinematic mobility. I’m sure someone does though.)

I sent a note around asking who might know and have interest in alternative mobility designs. It’s a fun topic!

-Rob

It is really flattering to receive such a compliment from you Mr. Manning.

I did not know about the ATHLETE platform until now but I have been reading about it for the past few hours and it is really impressive. I had a similar approach with my project.

Thank you so much for asking around, please connect me with anyone who may know more/ would be interested in helping out.

P.S. I plan on applying for an internship at JPL for next summer, I hope I could get to work with you or talk to you in person if that works out for me.

Again, thank you so much for your precious time.

Sincerely,
Youssef Abdelhalim.

Appendix C: Walkers

There could be thousands of different walkers that create different locusts and have different amounts of connections since there are 4 parameters that could be altered when forming a 4-bar linkage. For example, Analysis of the Four Bar Linkages by John A. Hrones and George L. Nelson, which is an atlas for hundreds of different dynamics of 4-bar linkages depending on the side length of each bar and which bar is the fixed one.

However, there are well established walkers or mechanisms that have craved their way into the mechanical walkers fields and formed their own names and rules, for example, Theo Jansen, who invented the Jansen Linkage, discovered the “11 Holy Numbers”, which are proportions that he came up with to ensure the smooth gliding motion of his Strandbeests^{TM}.

This appendix will discuss the most widely known and established mechanisms and types of linkages.

The four most well-known walkers are the Strandbeest, TrotBot, Strider, and Klann walkers. All of these walkers use 4-bar linkages in their core to control their movements. The fact that they all have only one input link and one output link makes the design and manufacturing process way simpler since there is only one degree of freedom.

Some of the goals that these 4-bar linkage walking mechanisms try to achieve are as follows [12]:

- Horizontal speed to be as constant as possible while touching the ground
- While the foot is not touching the ground, it should move as fast as possible
- Constant torque/force input
- Stride height to be enough for clearance but not too big to avoid using up too much energy
- The foot has to touch the ground for at least half of the cycle for 3 or 6 leg mechanism
- Minimized moving mass
- Vertical center of mass always inside the base of support
- The speed of each leg or group of legs should be separately controllable for steering
- The leg mechanism should allow forward and backward walking

Strandbeest - Jansen’s Linkage

The Strandbeest was invented by Theo Jansen in 1990 and it has been through 12 stages of evolution since its inception [13]. The main innovation that is housed by the Strandbeest is the Jansen Linkage, which is the main mechanism that is used to control the locus, or the footpath. The Jansen Linkage has allowed the Strandbeests to have a really fluid and smooth movement on the beaches, giving off an illusion that they are gliding atop the sand rather than walking on the sand. As seen in figure 34, the locus seems to be flat on the bottom, this explains why the Stranbeest glides on the sand instead of pushing on it like a human's or an animal's foot would.

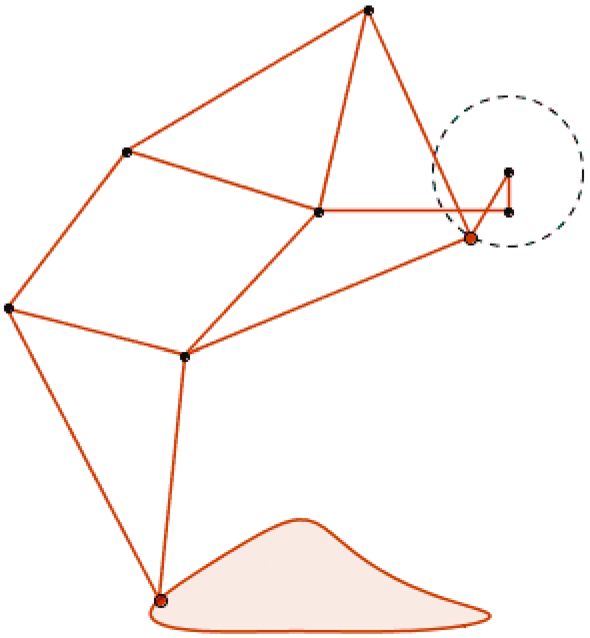


Figure 34: The locus of the Jansen Linkage

The proportions of the bars involved in the Jansen Linkage are called the "11 Holy Numbers." The lengths of these bars don't matter as long as they follow the proportions discovered by Jansen. These proportions lead directly to the path derived by the locus, changing the bars' length will alter the locus. A Mathematica program was prepared to explore how these alterations affect the locus and will be discussed more in-depth in a separate appendix.

The "11 Holy Numbers" were developed by Theo Jansen after running simulations on the Atari to figure out the perfect proportions that produced the locus that he wanted. He purposely avoided modeling the locus based on nature or animals, he wanted to generate his ratios in much the same way that nature itself had most likely done so, natural history understood in this context as a vast sort of calculating algorithm [13].

The "11 Holy Numbers" are just a ratio in dimensions and can be converted into any unit or scale. It should be noted that there are 13 numbers, however, 2 of these numbers represent the dimensions of the fixed frame, so they tend to be left out since they don't have much of an impact on the dynamics of the leg. They are presented in table 1:

Table 1: Bar, dimension, and length in inches [14]

Bar	Ratio	Length (in)
a	38.0	4.3379
b	41.5	4.7374
c	39.2	4.4863
d	40.1	4.5776
e	55.9	6.3699
f	39.4	4.4977
g	36.7	4.1895
h	65.7	7.5000
i	49.0	5.5936
j	50.0	5.7078
k	61.9	7.0662
l	7.8	0.8904
m	15.0	1.7123

These ratios or lengths correspond to the the bars labeled in figure 35:

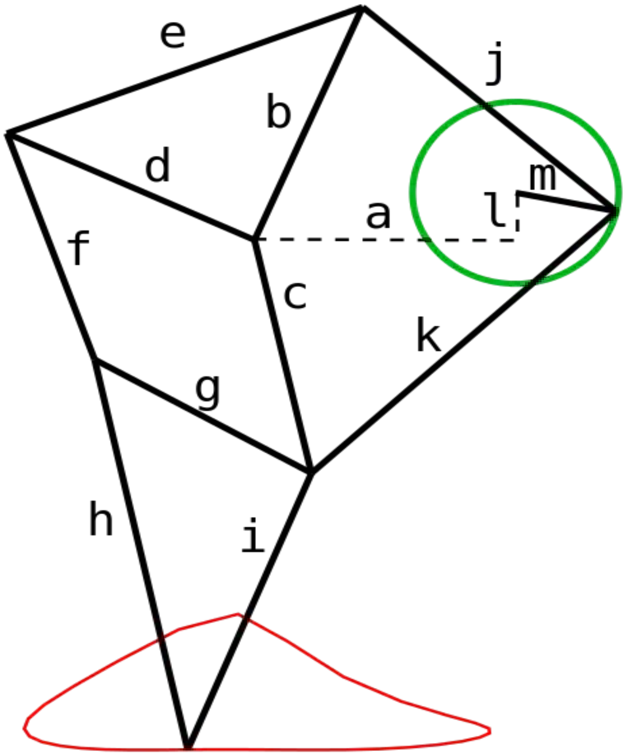


Figure 35: Labeled Jansen Linkage with “11 Holy Numbers” [14]

TrotBot - TrotBot's Linkage

The TrotBot was designed by Team TrotBot/ DIY Walkers, more specifically Ben Vagle and Wade Vagle. The TrotBot was inspired by the gait of a galloping horse and it was designed explicitly to walk on jagged surfaces such as rocky terrain that have pebbles of many different sizes. It lifts its feet high so they don't get jammed on obstacles [15].

Each linkage in the TrotBot has 8 bars and they are placed as follows:

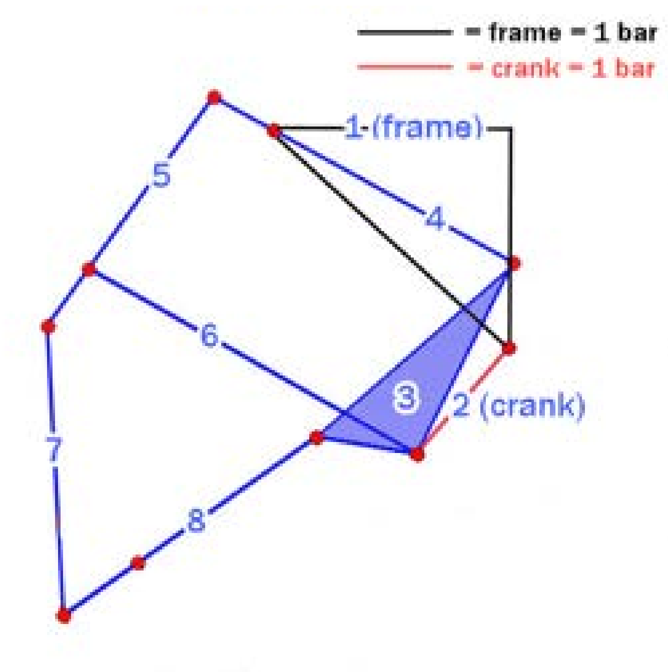


Figure 36: TrotBot Linkage's bar count [15]

The TrotBot has a more ovular locus compared to the strandbeest:

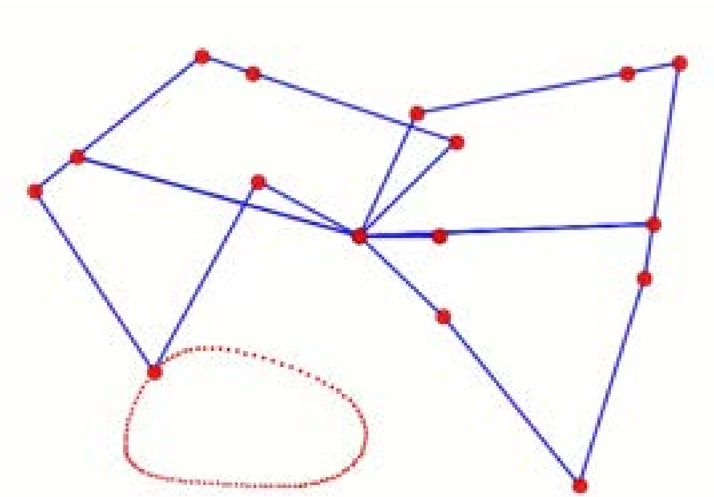


Figure 37: TrotBot's locus [15]

The TrotBot walks more efficiently with the addition of a "heel" on both smooth and rough terrains. The TrotBot's heel strikes before the main foot, taking the weight while pushing backward to continue driving the robot forward. The resulting gait reduces both torque and power consumption. This would be convenient if you were to use a power source to move the rover as opposed to using wind to move the rover as seen with the Strandbeests.

Adding a heel also causes the TrotBot to step higher on the backside of the locus, allowing the rear legs to step about as high as the front legs to avoid getting stuck when meeting obstacles. This is similar to the way cheetahs' have their back legs go as high as their front legs momentarily when they are running [15].

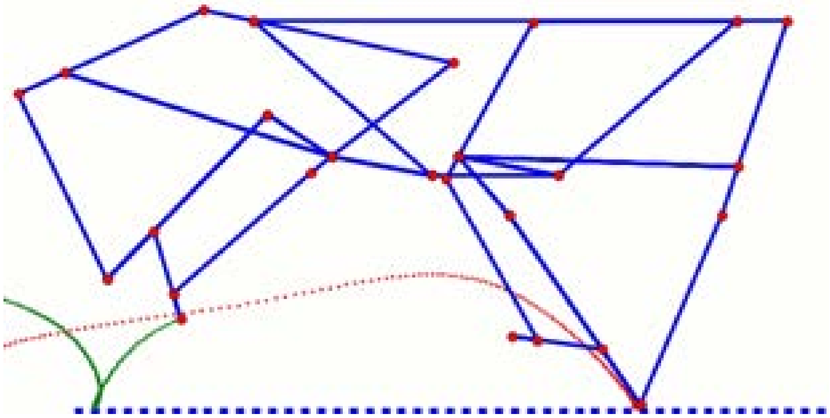


Figure 38: TrotBot with heel [15]

Figure 39: Cheetah running with back heel as high as front heel to avoid obstacles [16]

Strider's Linkage

The Strider's Linkage was also developed by DIY Walkers, Wade and Ben Vagle. They intended to build this linkage for more rugged terrains, just as they intended for the TrotBot Linkage. They formed it by attaching two 4-bar linkages to form a 10-bar linkage

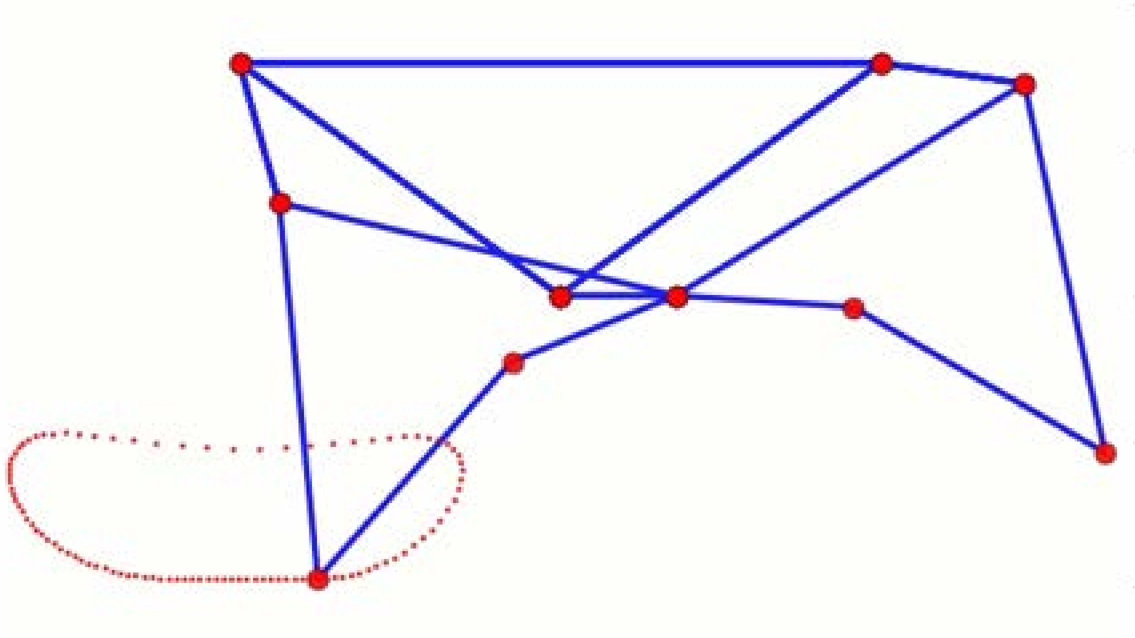


Figure 40: Strider's 10-bar linkage [15]

They designed both the TrotBot and the Strider while keeping in mind that obstacles such as rocks and pebbles are important to take account of. To address that, they made sure that the crank in both linkages lifted high enough to increase the step-height for rugged terrains.

The boat shape of the locus of the Strider's linkage results in a more consistent speed and less skidding when stepping on obstacles [15].

Klann's Linkage

Similar to how the TrotBot was modeled after a galloping horse or a speeding cheetah, Klann's linkage was modeled after the gait of a legged animal. It was developed by Joe Klann in 1994 as an expansion of the Burmester curves which are used to develop 4-bar double-rocker linkages. Klann's linkage is categorized as a Sephenson type III kinematic chain [17].

The linkage consists of the frame, a crank, two grounded rockers, and two couplers all connected by pivot joints. The proportions of each of the links are defined in such a way that optimizes the linearity of the foot for one-half of the rotation of the crank.

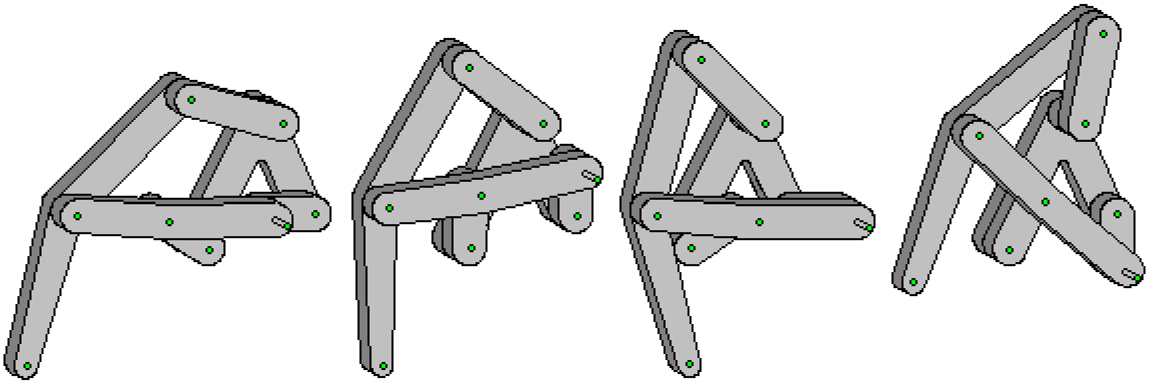


Figure 41: Klann's linkage in 4 separate stages of motion [17]

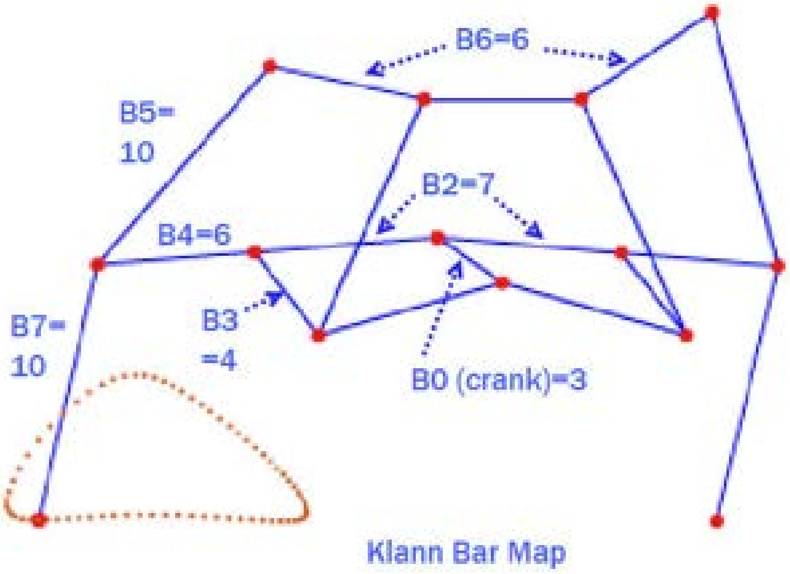


Figure 42: Klann's linkage locus and bar map [15]

Appendix D: Grashof's Law

This appendix will address the parts of Gashof's Law that are applicable to creating a 4-bar linkage that can be used for the TitanWandelaar's legs.

Introduction

Grashof's Law addresses the dynamics and the possible outcomes of any 4-bar linkage system. By applying Grashof's Law, one can design a crank-crank, crank-rocker, or a rocker-rocker 4-bar linkage.

Results

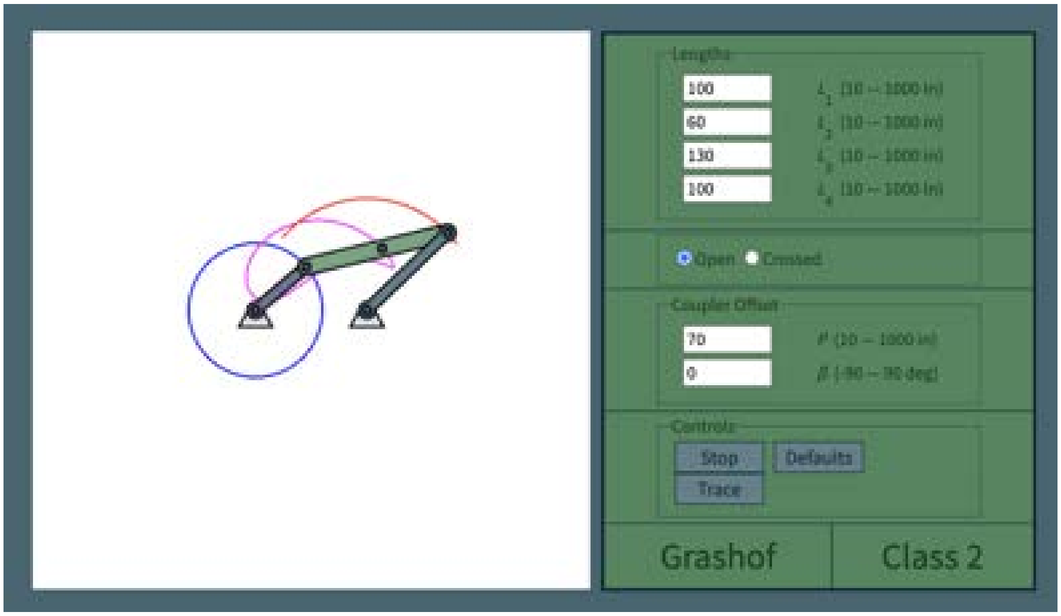
If you have a 4-bar linkage, where l is the longest link, s is the shortest link, and p and q are assigned randomly to the other links whose lengths fall in between:

l = longest link
s = shortest link
p = remaining link
q = other remaining link

By Gashof Law, for one link to be able to make one full revolution:

$$s + l \leq p + q$$

As seen in the figure 43, $s = 60\ in$, $l = 130\ in$, $p = 100\ in$, and $q = 100\ in$. Since $s + l = 60 + 130 = 190$, which is less than $p + q = 100 + 100 = 200$, then using Grashof's law, we can conclude that this 4-bar linkage will form a crank-rocker movement where the shortest link will crank and the longest link will rock.



Lengths

100 L_1 (10 – 1000 in)
60 L_2 (10 – 1000 in)
130 L_3 (10 – 1000 in)
100 L_4 (10 – 1000 in)

● Open ○ Crossed

Coupler Offset

70 P (10 – 1000 in)
0 β (-90 – 90 deg)

Grashof

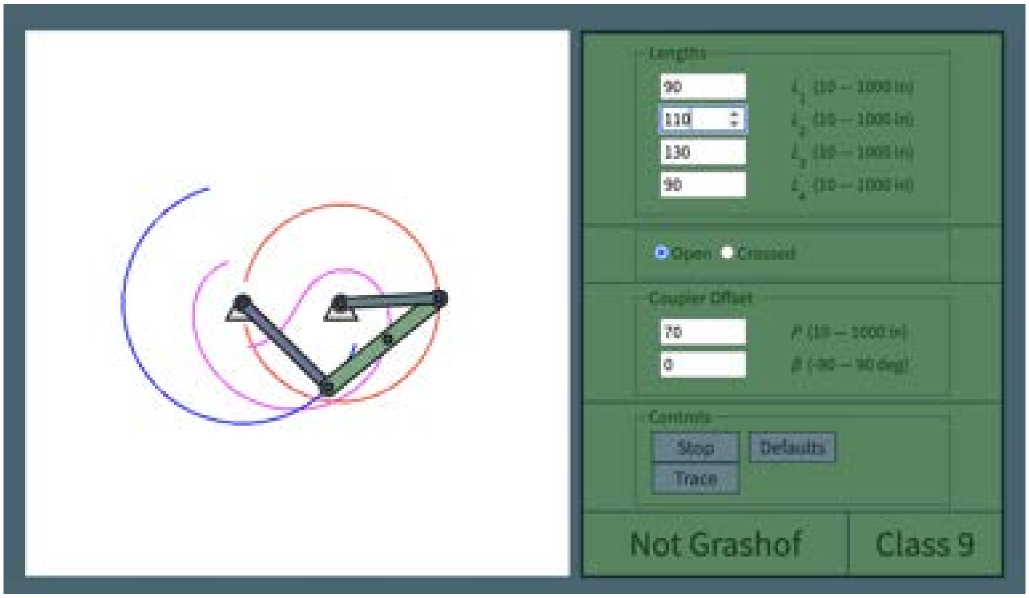
Class 2

Figure 43: Crank-rocker 4-bar linkage

By Gashof Law, neither link will be able to make a complete revolution:

$$s + l > p + q$$

As seen in the figure 44, $s = 90\ in$, $l = 130\ in$, $p = 90\ in$, and $q = 110\ in$. Since $s + l = 90 + 130 = 220$, which is less than $p + q = 90 + 110 = 200$, then using Grashof's law, we can conclude that this 4-bar linkage will form a rocker-rocker movement where no link completes a full revolution.



Lengths

90 L_1 (10 – 1000 in)

110 L_2 (10 – 1000 in)

130 L_3 (10 – 1000 in)

90 L_4 (10 – 1000 in)

● Open ○ Crossed

Coupler Offset

70 P (10 – 1000 in)

0 β (-90 – 90 deg)

Not Grashof

Class 9

Figure 44: Rocker-rocker 4-bar linkage

There are three cases under the conditions $s + l < p + q$ that can lead to different dynamics:

Case 1:
- Shortest link is adjacent to fixed link
- Crank-Rocker mechanism
- Shortest link is the crank link

Case 2:
- Shortest link is the fixed link
- Double crank mechanism

Case 3:
- Shortest link is opposite to the fixed link
- Double-rocker mechanism

Inversion:

Created by grounding a different link in a kinematic chain. There are as many inversions of a given linkage as it has links

So for $s + l < p + q$, we have 4 inversions for which 3 are distinct inversions:

1. Crank Rocker (2)
    a. Obtained by grounding either of the links adjacent to the shortest link
2. Double Crank
    a. Obtained by grounding the shortest link
3. Double Rocker
    a. Obtained by grounding the link opposite to the shortest link

When $s + l = p + q$, the links become collinear at least once per revolution of input crank

- If all side lengths are distinct, all the inversions obtained are the same as in the case $s + l < p + q$
- In the case length of 2 links have the same side length:
    - Equal links opposite to each other:
        - Parallelogram linkage
    - Equal links adjacent to each other:
        - Deltoid linkage
    - All inversions are either crank rocker or double cranks for this case.
    - When the links become collinear, both the linkages suffer from “change point condition” i.e. the output behavior becomes indeterminate. The linkage may assume either double crank or crank rocker configurations at these positions

Appendix E: Linkage

This appendix will address a really useful and simple software developed by David Rector, an enthusiast of linkages and wanted to build a software where you can create and modify linkages. Most of this appendix will borrow information that he mentioned on his website regarding how the software runs.

Linkage is a CAD program that is used for designing and prototyping mechanical linkages. The linkage could be easily edited and animated right on the same window, which makes this software spectacular for quick analysis and modification while working on a design.

Linkage changed the course of action of my project and allowed me to cut multiple phases of the prototyping and testing stages, allowing me to save time and resources. If it weren't for Linkage, I would have had to create drawings of the linkages that I thought would satisfy the locus that I think would fit best for the problem I am trying to solve. Then, I would need to create mock-ups of these drawings, and use a pencil attached to the feat to make sure that the desired locus is indeed achieved, as seen in figure 45:

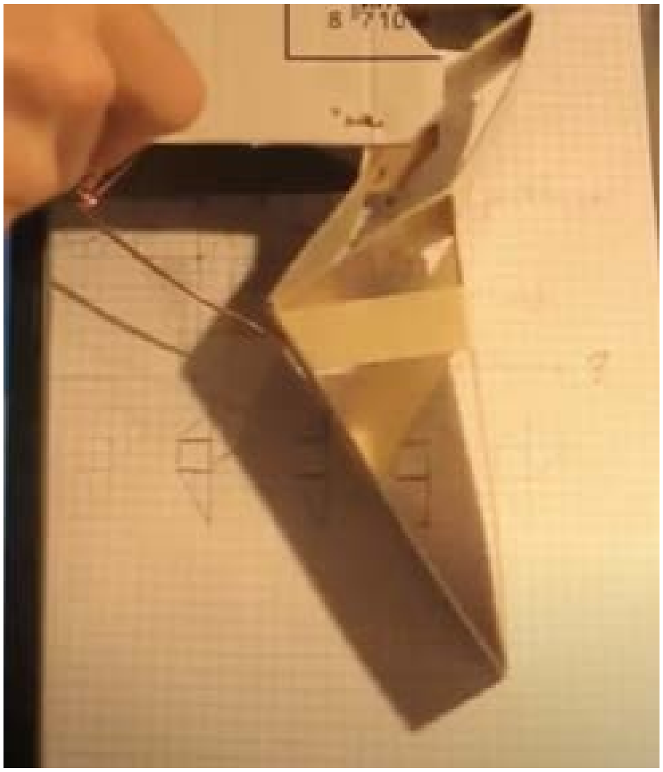


Figure 45: Screenshot of YouTuber riklmr producing a mock-up of a Jansen Linkage using tape and cardboard pieces [18]

Moreover, the software alerts you if you create a faulty or an erroneous linkage where one of the bars would have to get bent or stretched for the linkage to work. This is a great feature because again, it would be wasteful to find out such information during the mock-up phase.

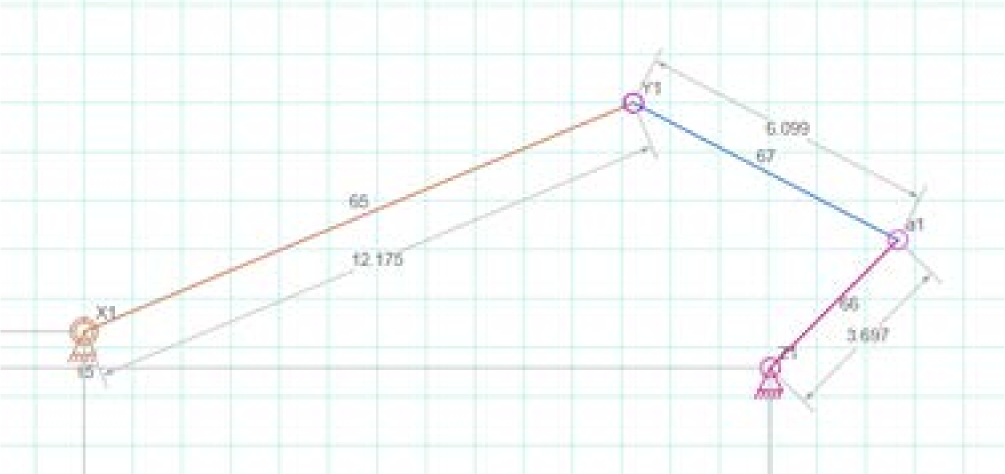
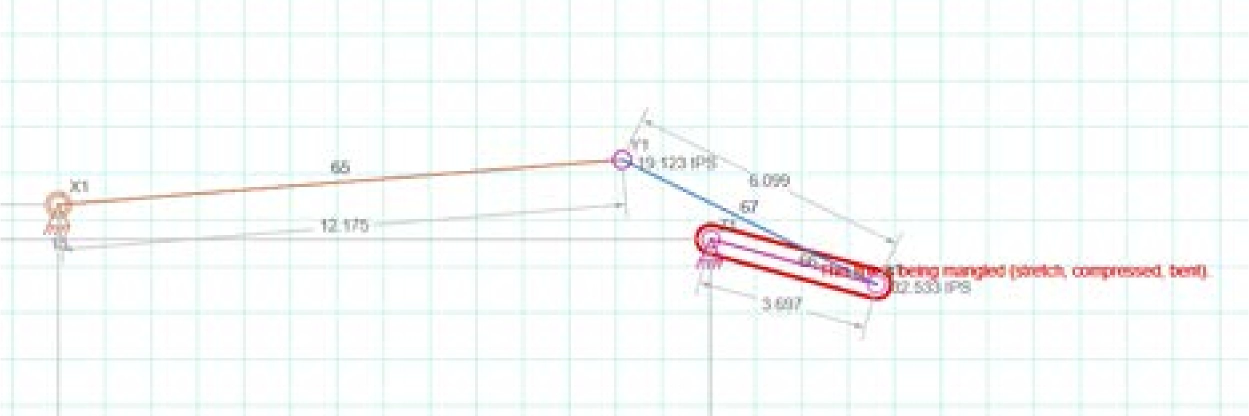


Figure 46: Random linkage I created to demonstrate the error message



[illegible] being mangled (stretch, compressed, bent).

Figure 47: Error message demonstrated after trying to animate a faulty linkage

While the mock-up phase is indeed meant for experimenting your ideas and seeing if something works or not, and that everything working should not be the expectation during the mock-up phase, lots of time and money could be saved if we could eliminate ideas that won't work before we even get to the mock-up stage. This program allowed me to narrow down my options to linkages that all worked, and I had to make a decision based on the locus.

In fact, for this project, I had borrowed scraps from the fabrication shop at the Ford Motor Company Engineering Design Center in order to simulate how some of the linkage I design will behave. This was before I found out about Linkage. Many of the initial linkages that I designed on Linkage were faulty. If I hadn't found Linkage and had to create mock-ups for all of the linkages just to end up finding that each one of them was faulty, I would have been set back plenty and with a short, 10-week project like this one, time is really valuable, so there isn't much of it to be poured into the prototyping phase.

In Linkage, mechanisms can be designed with pivot connectors or sliding connectors that also pivot. Inputs to drive the mechanism can be rotary or linear. The number of connections on a link and the number of links that one wants to input into the software is virtually unlimited, though adding links should be done with extreme care. Creating a functioning 4-bar linkage is already hard as is, creating higher order linkages gets even more tricky.

Linkage is a Windows program that has been developed and tested on Windows 7, 8, and 10. It has also run on some other Windows operating systems such as Windows XP. Unfortunately, Linkage is not Macintosh Operating System compatible, however, one way to get around this is by using VMware or Parallels Desktop. These are both Windows simulators that you can download on a MacBook in order to simulate a controlled Windows environment.

According to Dave's Blog, some of the features on Linkage are:

- Works like a vector drawing program.
- Has a modeless interface with no mouse tool selection for any operation or action.
- Lets the user create any configuration of links, connections, gears, and chains. There is no limit to using specific types of linkages and mechanisms.
- Gear and chain mechanisms can have gears on moving links.
- Has a visual style that matches mechanisms shown in many books.
- Runs at 30 frames per second when simulating the mechanism.
- Reads and writes .linkage2 files that use the XML format.
- Can move, rotate, scale, stretch, cut, copy, and paste, any set of selected connectors and links.
- Can align selected connectors in many ways including at right angles, any angle, in a parallelogram or rectangle, etc.
- Will optionally snap connectors to a grid and to other objects during editing.
- Has a zoom and pan.
- Has unlimited levels of undo of all operations (depending on available memory). Also has a Redo feature.
- Will play, stop, pause, and step the simulation, at any time during editing.
- Uses pivoting connectors as well as less common sliding connectors.
- Allows for any number of rotating and/or linear (actuator/hydraulic) inputs.
- Allows control of input positions manually during the simulation, if desired.
- Will print hard copies of the mechanism on one page and on multiple pages at 1:1.
- Lets you record the simulation in an HD video file.
- Lets you save a picture of a mechanism in JPEG or PNG format in a variety of sizes. The image can also be copied and pasted into other programs and apps.

- Allows you to assign drawing capability to any connector to visualize its path during simulation.
- Will open and simulate a wide variety of included sample mechanisms.
- Automatically displays dimensions of parts in mm or inches in a way that is suited to manufacturing individual parts.
- Will draw dimension/measurement lines manually.
- Will draw points and lines separate from the simulated mechanism.

Features in Linkages

Linkage has a lot of tools that allow you to create linkages that you can easily design in the real world. Linkage is very much a concrete program, designed by an engineer for engineers with the understanding that it is meant to design linkages that will later on get prototyped or fabricated. By right-clicking on the main window, the following taskbar pops up.

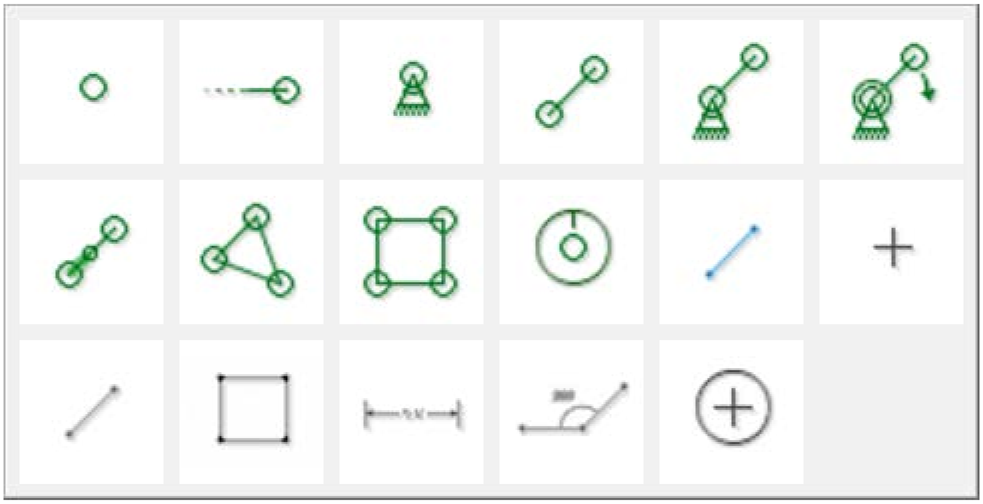


Figure 48: Pop-up Taskbar in Linkage

Table 2: Descriptions of each of the options from the pop-up taskbar

Item #	Item name	Description
1	Connector	Add a single lone connector
2	Add linked connector	Add a connector and create a link from the previously selected connector to the new connector
3	Anchor	Add a single lone anchor
4	Link	Add a link with 2 connectors

5	Anchor with link	Add a link with an anchor and a connector
6	Input with link	Add a link with a rotating input anchor and a connector
7	Linear Actuator	Add a linear actuator
8	3-connector link	Add a link with 3 connectors
9	4-connector link	Add a link with 4 connectors
10	Gear	Add a gear
11	Guideline	Add a guideline to the drawing layer
12	Point	Add a point to the drawing layer
13	Line	Add a line to the drawing layer
14	Polygon	Add a polygon to the drawing layer
15	Insert Measurement	Add a measurement to the drawing layer
16	Angle Measurement	Add a polyline to the drawing layer showing the angle between the segments
17	Circle	Add a point with a circle to the drawing layer

Appendix F: Effects of Bar Lengths on Jansen’s Locus

This appendix will demonstrate how changing the length of a bar in the Jansen Linkage can change the locus and thus change the functionality of the mechanism on a given surface.

This is demonstrated using Mathematica, an application that uses the Wolfram language. To effectively see the effects, copy the following code, and paste it into Mathematica and click on Shift and Enter at the same time and then play around with the dials to visually see how it impacts the locus.

The Wolfram code is as follows [8]:

```
Manipulate[

 \[Alpha] =
  ArcCos[(-a^2 + b^2 - c^2 - d^2 + 2 a d Cos[\[Phi]] + (
      4 a^4 Sin[\[Phi]]^2)/(
      4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
       4 a^2 Sin[\[Phi]]^2) - (4 a^2 b^2 Sin[\[Phi]]^2)/(
      4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
       4 a^2 Sin[\[Phi]]^2) + (4 a^2 c^2 Sin[\[Phi]]^2)/(
      4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
       4 a^2 Sin[\[Phi]]^2) + (4 a^2 d^2 Sin[\[Phi]]^2)/(
      4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
       4 a^2 Sin[\[Phi]]^2) - (8 a^3 d Cos[\[Phi]] Sin[\[Phi]]^2)/(
      4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
       4 a^2 Sin[\[Phi]]^2) + (a Sin[\[Phi]] \[Sqrt](a^2 c^2 (4 a^2 -
              4 b^2 + 4 c^2 + 4 d^2 -
              8 a d Cos[\[Phi]])^2 Sin[\[Phi]]^2 -
           4 c^2 (a^4 - 2 a^2 b^2 + b^4 + 2 a^2 c^2 - 2 b^2 c^2 +
              c^4 + 2 a^2 d^2 - 2 b^2 d^2 - 2 c^2 d^2 + d^4 -
              4 a^3 d Cos[\[Phi]] + 4 a b^2 d Cos[\[Phi]] +
              4 a c^2 d Cos[\[Phi]] - 4 a d^3 Cos[\[Phi]] -
              4 a^2 c^2 Cos[\[Phi]]^2 +
              4 a^2 d^2 Cos[\[Phi]]^2) (4 d^2 - 8 a d Cos[\[Phi]] +
              4 a^2 Cos[\[Phi]]^2 +
              4 a^2 Sin[\[Phi]]^2)))/(c (4 d^2 - 8 a d Cos[\[Phi]] +
          4 a^2 Cos[\[Phi]]^2 + 4 a^2 Sin[\[Phi]]^2)))/(c (-2 d +
        2 a Cos[\[Phi]]))];
 \[Beta] =
```

```
 ArcCos[(-a^2 + b^2 - c^2 - d^2 + 2 a d Cos[-\[Phi]] + (
    4 a^4 Sin[-\[Phi]]^2)/(
    4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
     4 a^2 Sin[-\[Phi]]^2) - (4 a^2 b^2 Sin[-\[Phi]]^2)/(
    4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
     4 a^2 Sin[-\[Phi]]^2) + (4 a^2 c^2 Sin[-\[Phi]]^2)/(
    4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
     4 a^2 Sin[-\[Phi]]^2) + (4 a^2 d^2 Sin[-\[Phi]]^2)/(
    4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
     4 a^2 Sin[-\[Phi]]^2) - (8 a^3 d Cos[-\[Phi]] Sin[-\[Phi]]^2)/(
    4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
     4 a^2 Sin[-\[Phi]]^2) + (a Sin[-\[Phi]] \[Sqrt](a^2 c^2 (4 a^2 \
- 4 b^2 + 4 c^2 + 4 d^2 - 8 a d Cos[-\[Phi]])^2 Sin[-\[Phi]]^2 -
         4 c^2 (a^4 - 2 a^2 b^2 + b^4 + 2 a^2 c^2 - 2 b^2 c^2 +
           c^4 + 2 a^2 d^2 - 2 b^2 d^2 - 2 c^2 d^2 + d^4 -
           4 a^3 d Cos[-\[Phi]] + 4 a b^2 d Cos[-\[Phi]] +
           4 a c^2 d Cos[-\[Phi]] - 4 a d^3 Cos[-\[Phi]] -
           4 a^2 c^2 Cos[-\[Phi]]^2 +
           4 a^2 d^2 Cos[-\[Phi]]^2) (4 d^2 -
           8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
           4 a^2 Sin[-\[Phi]]^2)))/(c (4 d^2 -
        8 a d Cos[-\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
        4 a^2 Sin[-\[Phi]]^2)))/(c (-2 d + 2 a Cos[-\[Phi]]))];
vA = {a Cos[\[Phi]], 0, a Sin[\[Phi]]};
vB = {-c Cos[\[Alpha]] + d, 0, c Sin[\[Alpha]]};
vC = {d, 0, 0};
vD = {0, 0, 0};

vE = {-c Cos[\[Beta]] + d, 0, -c Sin[\[Beta]]};
pivotE = {Red,
  Cylinder[{{-c Cos[\[Beta]] +
     d, -0.5, -c Sin[\[Beta]]}, {-c Cos[\[Beta]] + d,
    0.2, -c Sin[\[Beta]]}}, 0.1]};
pivotEa = {Yellow,
  Cylinder[{{-c Cos[\[Beta]] +
     d, -0.25, -c Sin[\[Beta]]}, {-c Cos[\[Beta]] +
     d, -0.15, -c Sin[\[Beta]]}}, 0.16]};
pivotEb = {Yellow,
  Cylinder[{{-c Cos[\[Beta]] +
     d, -0.45, -c Sin[\[Beta]]}, {-c Cos[\[Beta]] +
```

```
      d, -0.35, -c Sin[\[Beta]]}}, 0.16]};
vF = {-c Cos[\[Alpha] + addF] + d, 0, c Sin[\[Alpha] + addF]};
pivotF = {Red,
  Cylinder[{{-c Cos[\[Alpha] + addF] + d, -0.32,
     c Sin[\[Alpha] + addF]}, {-c Cos[\[Alpha] + addF] + d, 0.1,
     c Sin[\[Alpha] + addF]}}, 0.1]};
pivotFa = {Yellow,
  Cylinder[{{-c Cos[\[Alpha] + addF] + d, -0.3,
     c Sin[\[Alpha] + addF]}, {-c Cos[\[Alpha] + addF] + d, -0.1,
     c Sin[\[Alpha] + addF]}}, 0.17]};
vG = {((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
    2 + ((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/2 -
   d, 0, ((-c Sin[\[Beta]]) + c Sin[\[Alpha] + addF])};
pivotG = {Red,
  Cylinder[{{((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
       2 + ((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
       2 -
      d, -0.33, ((-c Sin[\[Beta]]) +
       c Sin[\[Alpha] +
          addF])}, {((-c Cos[\[Beta]] +
           d) + (-c Cos[\[Alpha] + addF] + d))/
       2 + ((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
       2 - d, 0.2, ((-c Sin[\[Beta]]) + c Sin[\[Alpha] + addF])}},
   0.1]};
pivotGa = {Yellow,
  Cylinder[{{((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
       2 + ((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
       2 -
      d, -0.3, ((-c Sin[\[Beta]]) +
       c Sin[\[Alpha] +
          addF])}, {((-c Cos[\[Beta]] +
           d) + (-c Cos[\[Alpha] + addF] + d))/
       2 + ((-c Cos[\[Beta]] + d) + (-c Cos[\[Alpha] + addF] + d))/
       2 - d, -0.1, ((-c Sin[\[Beta]]) + c Sin[\[Alpha] + addF])}},
   0.16]};

vH = {-c Cos[\[Beta]] + d + h Cos[leg \[Alpha]],
  0, -c Sin[\[Beta]] - h Sin[leg \[Alpha]]};
```

```
barA = Cylinder[{vD, vA}, 0.05];
barB = Cylinder[{{a Cos[\[Phi]], -0.2,
   a Sin[\[Phi]]}, {-c Cos[\[Alpha]] + d, -0.2, c Sin[\[Alpha]]}},
  0.05];
barC = Cylinder[{vC, vB}, 0.05];
barD = Cuboid[{-0.2, 1, -0.3}, {2.7, 1.3, 0.3}];
barCE = {Yellow, Translate[Cylinder[{vC, vE}, 0.05], {0, -0.2, 0}]};
barAE = {Red, Translate[Cylinder[{vA, vE}, 0.05], {0, -0.4, 0}]};
barCF = {Yellow, Translate[Cylinder[{vC, vF}, 0.05], {0, 0, 0}]};
barBF = {Red, Translate[Cylinder[{vB, vF}, 0.05], {0, 0, 0}]};
barFG = {Yellow, Translate[Cylinder[{vF, vG}, 0.05], {0, -0.2, 0}]};
barEG = {Yellow, Translate[Cylinder[{vE, vG}, 0.05], {0, 0, 0}]};
barEH = Translate[Cylinder[{vE, vH}, 0.05], {0, 0, 0}];
barGH = Translate[Cylinder[{vG, vH}, 0.05], {0, 0, 0}];

w1 = {RGBColor[0.4, 0.5, 0.9],
  Cylinder[{{0, 0.1, 0}, {0, 0.2, 0}}, 1.1 a]};
shaft1 = {Green, Cylinder[{{0, -0.2, 0}, {0, 1, 0}}, 0.14]};
shaft1a = {Yellow, Cylinder[{{0, -0.05, 0}, {0, 0.3, 0}}, 0.2]};
shaft2 = {Green, Cylinder[{{d, -0.4, 0}, {d, 1, 0}}, 0.1]};
shaft2a = {Yellow, Cylinder[{{d, -0.08, 0}, {d, 0.1, 0}}, 0.16]};
shaft2b = {Yellow, Cylinder[{{d, -0.15, 0}, {d, -0.3, 0}}, 0.16]};
pivot1 = {Yellow,
  Cylinder[{{a Cos[\[Phi]], -0.1,
    a Sin[\[Phi]]}, {a Cos[\[Phi]], -0.3, a Sin[\[Phi]]}}, 0.15]};
pivot1b = {Yellow,
  Cylinder[{{a Cos[\[Phi]], -0.5,
    a Sin[\[Phi]]}, {a Cos[\[Phi]], -0.34, a Sin[\[Phi]]}}, 0.15]};
pivot1a = {Red,
  Cylinder[{{a Cos[\[Phi]], 0.2, a Sin[\[Phi]]}, {a Cos[\[Phi]], -1,
    a Sin[\[Phi]]}}, 0.09]};

pivot2 = {Red,
  Cylinder[{{-c Cos[\[Alpha]] + d, -0.33,
    c Sin[\[Alpha]]}, {-c Cos[\[Alpha]] + d, 0.05,
    c Sin[\[Alpha]]}}, 0.1]};
pivot2a = {Yellow,
  Cylinder[{{-c Cos[\[Alpha]] + d, -0.3,
    c Sin[\[Alpha]]}, {-c Cos[\[Alpha]] + d, -0.1,
    c Sin[\[Alpha]]}}, 0.16]};
```

```
foot = {RGBColor[1, 0.3, 0.6],
  Cylinder[{{-c Cos[\[Beta]] + d +
     h Cos[leg \[Alpha]], -0.4, -c Sin[\[Beta]] -
     h Sin[leg \[Alpha]]}, {-c Cos[\[Beta]] + d +
     h Cos[leg \[Alpha]],
    0.1, -c Sin[\[Beta]] - h Sin[leg \[Alpha]]}}, 0.12]};
par = ParametricPlot3D[{-c Cos[
     ArcCos[(-a^2 + b^2 - c^2 - d^2 + 2 a d Cos[-\[Phi]] + (
        4 a^4 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) - (4 a^2 b^2 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) + (4 a^2 c^2 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) + (4 a^2 d^2 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) - (
        8 a^3 d Cos[-\[Phi]] Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) + (a Sin[-\[Phi]] \[Sqrt](a^2 c^2 \
(4 a^2 - 4 b^2 + 4 c^2 + 4 d^2 -
              8 a d Cos[-\[Phi]])^2 Sin[-\[Phi]]^2 -
            4 c^2 (a^4 - 2 a^2 b^2 + b^4 + 2 a^2 c^2 -
              2 b^2 c^2 + c^4 + 2 a^2 d^2 - 2 b^2 d^2 -
              2 c^2 d^2 + d^4 - 4 a^3 d Cos[-\[Phi]] +
              4 a b^2 d Cos[-\[Phi]] + 4 a c^2 d Cos[-\[Phi]] -
              4 a d^3 Cos[-\[Phi]] - 4 a^2 c^2 Cos[-\[Phi]]^2 +
              4 a^2 d^2 Cos[-\[Phi]]^2) (4 d^2 -
              8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
              4 a^2 Sin[-\[Phi]]^2)))/(c (4 d^2 -
            8 a d Cos[-\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[-\[Phi]]^2)))/(c (-2 d +
          2 a Cos[-\[Phi]]))]] + d +
    h Cos[
     leg ArcCos[(-a^2 + b^2 - c^2 - d^2 + 2 a d Cos[\[Phi]] + (
         4 a^4 Sin[\[Phi]]^2)/(
         4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
          4 a^2 Sin[\[Phi]]^2) - (4 a^2 b^2 Sin[\[Phi]]^2)/(
         4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
          4 a^2 Sin[\[Phi]]^2) + (4 a^2 c^2 Sin[\[Phi]]^2)/(
```

```
        4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
         4 a^2 Sin[\[Phi]]^2) + (4 a^2 d^2 Sin[\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
         4 a^2 Sin[\[Phi]]^2) - (
        8 a^3 d Cos[\[Phi]] Sin[\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
         4 a^2 Sin[\[Phi]]^2) + (a Sin[\[Phi]] \[Sqrt](a^2 c^2 \
(4 a^2 - 4 b^2 + 4 c^2 + 4 d^2 - 8 a d Cos[\[Phi]])^2 Sin[\[Phi]]^2 -
             4 c^2 (a^4 - 2 a^2 b^2 + b^4 + 2 a^2 c^2 -
              2 b^2 c^2 + c^4 + 2 a^2 d^2 - 2 b^2 d^2 -
              2 c^2 d^2 + d^4 - 4 a^3 d Cos[\[Phi]] +
              4 a b^2 d Cos[\[Phi]] + 4 a c^2 d Cos[\[Phi]] -
              4 a d^3 Cos[\[Phi]] - 4 a^2 c^2 Cos[\[Phi]]^2 +
              4 a^2 d^2 Cos[\[Phi]]^2) (4 d^2 -
              8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
              4 a^2 Sin[\[Phi]]^2)))/(c (4 d^2 -
            8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[\[Phi]]^2)))/(c (-2 d +
          2 a Cos[\[Phi]]))]], -0.4, -c Sin[
    ArcCos[(-a^2 + b^2 - c^2 - d^2 + 2 a d Cos[-\[Phi]] + (
        4 a^4 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) - (4 a^2 b^2 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) + (4 a^2 c^2 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) + (4 a^2 d^2 Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) - (
        8 a^3 d Cos[-\[Phi]] Sin[-\[Phi]]^2)/(
        4 d^2 - 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
         4 a^2 Sin[-\[Phi]]^2) + (a Sin[-\[Phi]] \[Sqrt](a^2 c^2 \
(4 a^2 - 4 b^2 + 4 c^2 + 4 d^2 -
              8 a d Cos[-\[Phi]])^2 Sin[-\[Phi]]^2 -
            4 c^2 (a^4 - 2 a^2 b^2 + b^4 + 2 a^2 c^2 -
              2 b^2 c^2 + c^4 + 2 a^2 d^2 - 2 b^2 d^2 -
              2 c^2 d^2 + d^4 - 4 a^3 d Cos[-\[Phi]] +
              4 a b^2 d Cos[-\[Phi]] + 4 a c^2 d Cos[-\[Phi]] -
              4 a d^3 Cos[-\[Phi]] - 4 a^2 c^2 Cos[-\[Phi]]^2 +
              4 a^2 d^2 Cos[-\[Phi]]^2) (4 d^2 -
```

```
                 8 a d Cos[-\[Phi]] + 4 a^2 Cos[-\[Phi]]^2 +
                 4 a^2 Sin[-\[Phi]]^2)))/(c (4 d^2 -
              8 a d Cos[-\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
              4 a^2 Sin[-\[Phi]]^2)))/(c (-2 d +
           2 a Cos[-\[Phi]]))]] -
    h Sin[
      leg ArcCos[(-a^2 + b^2 - c^2 - d^2 + 2 a d Cos[\[Phi]] + (
           4 a^4 Sin[\[Phi]]^2)/(
           4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[\[Phi]]^2) - (4 a^2 b^2 Sin[\[Phi]]^2)/(
           4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[\[Phi]]^2) + (4 a^2 c^2 Sin[\[Phi]]^2)/(
           4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[\[Phi]]^2) + (4 a^2 d^2 Sin[\[Phi]]^2)/(
           4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[\[Phi]]^2) - (
           8 a^3 d Cos[\[Phi]] Sin[\[Phi]]^2)/(
           4 d^2 - 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
            4 a^2 Sin[\[Phi]]^2) + (a Sin[\[Phi]] \[Sqrt](a^2 c^2 \
(4 a^2 - 4 b^2 + 4 c^2 + 4 d^2 - 8 a d Cos[\[Phi]])^2 Sin[\[Phi]]^2 -
                4 c^2 (a^4 - 2 a^2 b^2 + b^4 + 2 a^2 c^2 -
                 2 b^2 c^2 + c^4 + 2 a^2 d^2 - 2 b^2 d^2 -
                 2 c^2 d^2 + d^4 - 4 a^3 d Cos[\[Phi]] +
                 4 a b^2 d Cos[\[Phi]] + 4 a c^2 d Cos[\[Phi]] -
                 4 a d^3 Cos[\[Phi]] - 4 a^2 c^2 Cos[\[Phi]]^2 +
                 4 a^2 d^2 Cos[\[Phi]]^2) (4 d^2 -
                 8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
                 4 a^2 Sin[\[Phi]]^2)))/(c (4 d^2 -
               8 a d Cos[\[Phi]] + 4 a^2 Cos[\[Phi]]^2 +
               4 a^2 Sin[\[Phi]]^2)))/(c (-2 d +
            2 a Cos[\[Phi]]))]]}, {\[Phi], 0, 2 Pi}][[1]];
 polBCF = {RGBColor[0.4, 0.1, 0.4], Polygon[{vB, vC, vF}]};
 polEGH = {RGBColor[0.2, 0.6, 0.4], Polygon[{vE, vG, vH}]};
 Graphics3D[{polEGH, polBCF, pivotE, pivotEa, pivotEb, pivotFa,
   pivotG, pivotGa, pivotF, par, foot, barGH, barEH, barEG, barFG,
   barBF, barCE, barAE, barCF, pivot2, pivot2a, barD, pivot1, pivot1a,
    pivot1b, shaft1, shaft2, shaft2a, shaft2b, shaft1a,
   w1, {RGBColor[0.1, 0.1, 0.2], Specularity[0.7], barA, barB, barC}},
   Boxed -> False, ViewAngle -> Pi/48, ViewPoint -> {-4, -10, 5},
  ImageSize -> {400, 400},
```

```
PlotRange -> {{-2, 5}, {-1, 2}, {-5.4, 2.5}}],
{{\[Phi], 0, "rotate"}, 0, 2 Pi},
{{leg, 0.95, "adjust leg"}, 0.5, 1.5},
{{addF, 1, "advance pivot F"}, 0.8, 1.2},
{{a, 0.5, "bar a"}, 0.4, 0.6, ImageSize -> Tiny,
 ControlPlacement -> Left},
{{b, 2.3, "bar b"}, 1.8, 2.3, ImageSize -> Tiny,
 ControlPlacement -> Left},
{{c, 1.5, "bar c"}, 1.5, 1.7, ImageSize -> Tiny,
 ControlPlacement -> Left},
{{d, 1.6, "bar d"}, 1.6, 1.7, ImageSize -> Tiny,
 ControlPlacement -> Left},
{{h, 2.6, "leg"}, 0, 3, ImageSize -> Tiny, ControlPlacement -> Left},
 TrackedSymbols -> Manipulate, AutorunSequencing -> {1, 2, 4, 5}]
```

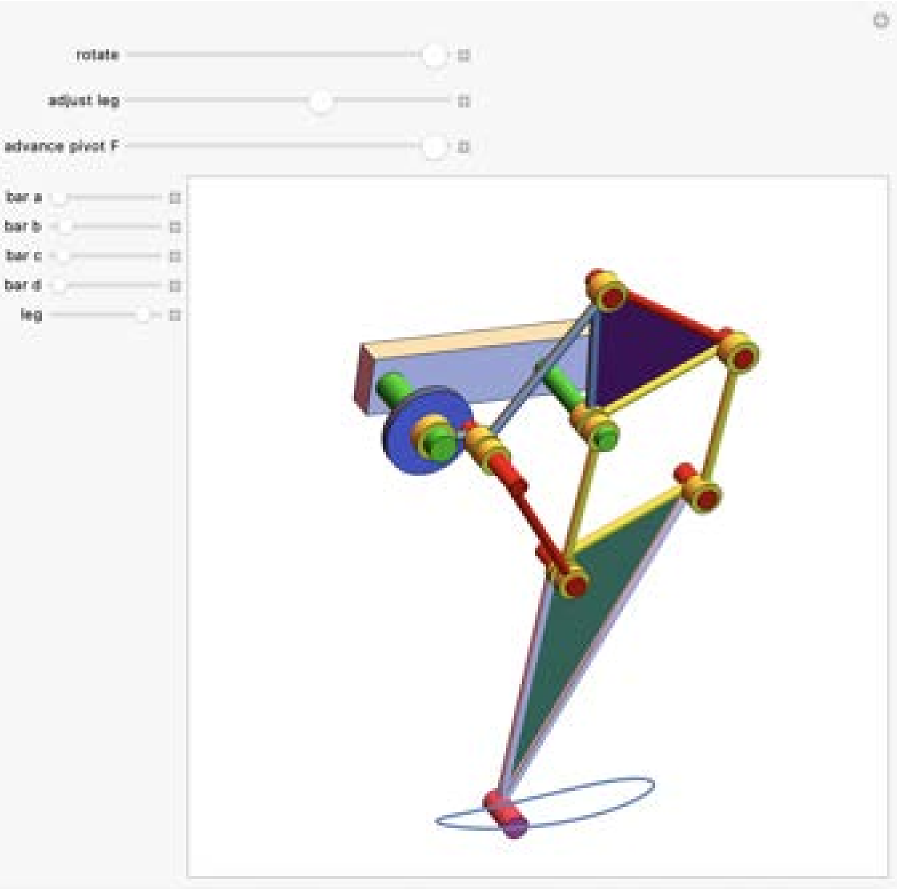


Figure 49: Visualization produced by running the Wolfram Code [8]

Appendix G: Arduino Code

This Appendix will feature the code that was uploaded to the Arduino to control the inputs and the outputs of the system. A different appendix will feature the exact configuration of the circuit.

```
/*
 LiquidCrystal Library - Temperature from a heat sensor

 Demonstrates the use of a 16x2 LCD display.  The LiquidCrystal
 library works with all LCD displays that are compatible with the
 Hitachi HD44780 driver. There are many of them out there, and you
 can usually tell them by the 16-pin interface.

 This sketch prints "Temperature in Degree Celsius !" to the LCD
 and shows the time.

  The circuit:
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * LCD VSS pin to ground
 * LCD VCC pin to 5V
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)

 http://www.arduino.cc/en/Tutorial/LiquidCrystal
 */

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int val;
```

```
int tempPin = 1;

#include <Adafruit_NeoPixel.h>

#define PIN 1      // pin neopixel is attached to
#define SENSOR A0     // input pin for Potentiometer
#define NUMPIXELS 1   // number of neopixels in strip

/* values to consider as wet or dry*/
#define dryThreshold 50    // below this value, begin alerting dry, turn red
#define wetThreshold 200   // above this value, begin alerting wet, turn blue
#define thresholdCenter (dryThreshold + wetThreshold)/2  // brightest green point
#define crossFade 20       // how much blue and red should fade in to green

Adafruit_NeoPixel pixels = Adafruit_NeoPixel (NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);

int redColor = 0;
int greenColor = 0;
int blueColor = 0;

int sensorValue = 0;
int transitionValue = 0;

int boton = 8;
int motor = 6;

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  //Serial.begin(9600);
  pixels.begin();
  pinMode(SENSOR, INPUT);
  pinMode(boton , INPUT);
  pinMode(motor , OUTPUT);
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
```

```
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  //lcd.print(millis() / 1000);
  val = analogRead(tempPin);
  float mv = ( val/1024.0)*5000;
  float cel = mv/10;
  float farh = (cel*9)/5 + 32;

  lcd.print("TEMPRATURE = ");
  lcd.print(cel);
  lcd.print("*C");
  lcd.println();
  //Serial.println();
  delay(1000);

  sensorValue = analogRead(SENSOR);
  transitionValue = map(sensorValue, 0, 1023, 0, 255);
  setColor();
  // pixels.Color takes RGB value, from 0,0,0 up to 255, 255, 255
  pixels.setPixelColor(0, redColor, greenColor, blueColor);

  // this sends the updated pixel color to the hardware
  pixels.show();

  // delay for a period of time (in milliseconds)
  delay(100);
  int estadoBoton = digitalRead(boton);

  if (estadoBoton == 1){
  digitalWrite(motor , LOW);
  }
  else {
  digitalWrite(motor , HIGH);
  }
}

void setColor(){
  // red value greater towards higher resistance/drier
  redColor = ((transitionValue <= dryThreshold + crossFade) && (transitionValue >= 0 )) ?
map(transitionValue, 0, dryThreshold + crossFade, 255, 0) : 0;
```

```
  // blue value greater towards lower resistance/wetter
  blueColor = ((transitionValue >= wetThreshold - crossFade) && (transitionValue <= 255)) ?
map(transitionValue, wetThreshold - crossFade, 255, 0, 255) : 0;

  // green value towards middle resistance
  if(transitionValue >= dryThreshold && transitionValue <= thresholdCenter) {
  greenColor = map(transitionValue, dryThreshold, thresholdCenter, 0, 255);
  }
  else if(transitionValue > thresholdCenter && transitionValue < wetThreshold){
  greenColor = map(transitionValue, dryThreshold, thresholdCenter, 255, 0);
  }
  else {
  greenColor = 0;
  }
}
```

Appendix H: Solidworks - Parts & Mates

This appendix will provide a detailed step-by-step process on how each part was created and how all the parts were mated together. These details were omitted from the main body of the report for conciseness.

Leg

The first piece to design would be the bar of the leg that is 6.145” long. We draw a 6.145” guideline and then a 0.25” radius circle and a concentric 0.4” radius circle at each end point. We then connect the two shapes and extrude the whole sketch by 0.10” to make it into a solid object.

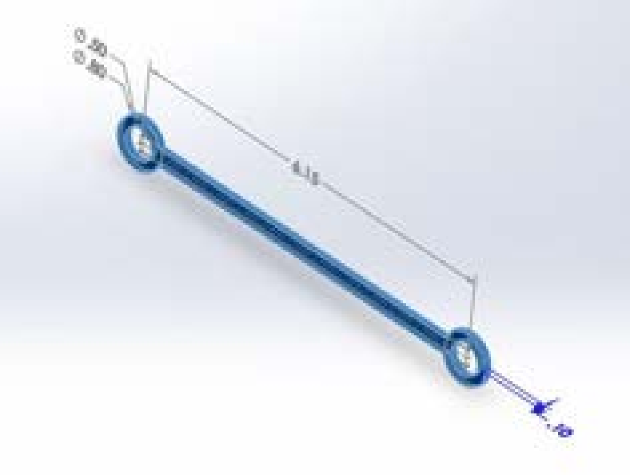


Figure 50: 6.145” Leg

We then use the same exact method to form the 8.576” leg

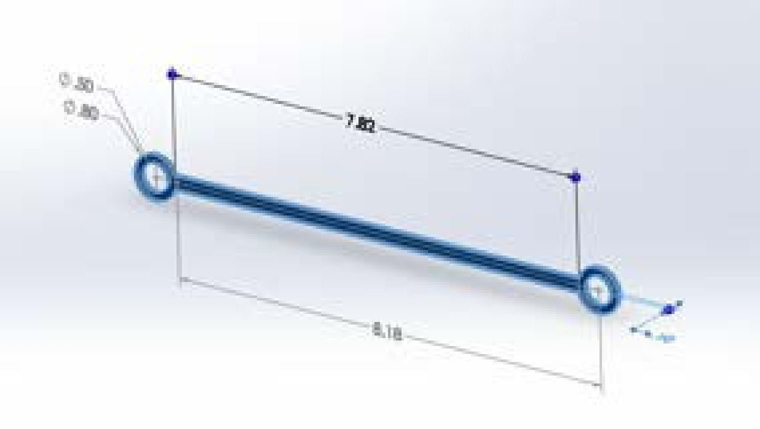


Figure 51: 8.576” Leg

We then create one of the two members of the leg that hook onto the crankpin journals of the crankshaft. We use the same process for this part, except that for the part that hooks onto the crankshaft, we use the centerpoint arc to have an open shape. We arbitrarily use the angle 130.02^o

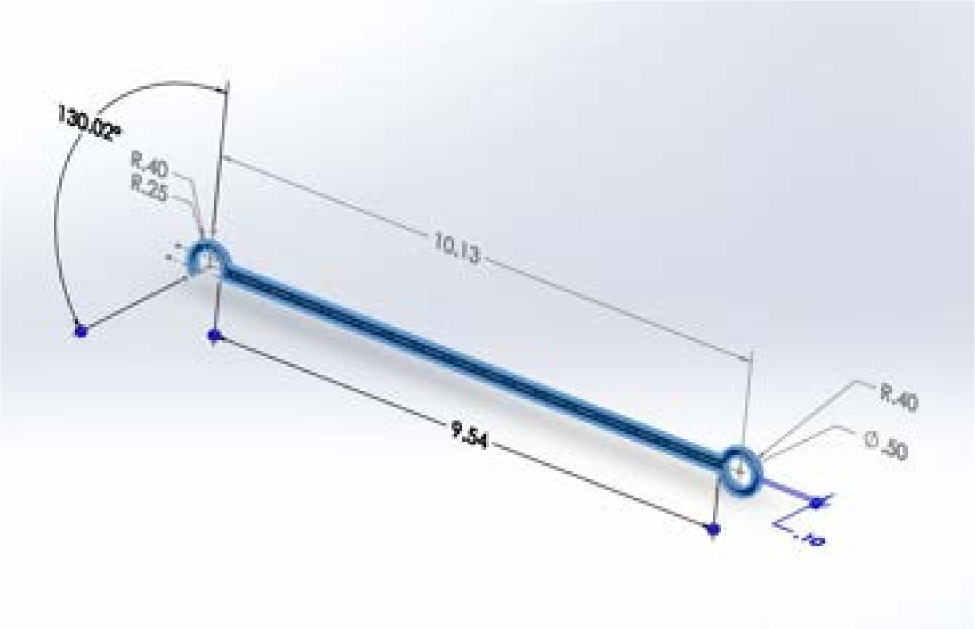


Figure 52: 10.134” Leg

We then use the same exact process to create the 9.605” leg

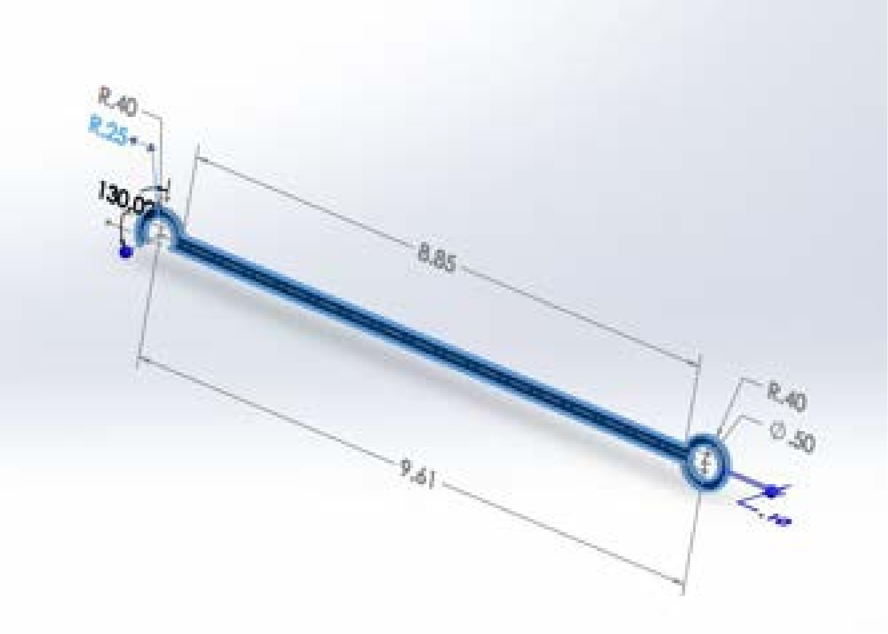


Figure 53: 9.605” Leg

We then create an assembly using these three leg parts. We do not include the first leg part (the 6.145” one) because adding that into this assembly instead of the main assembly restricts it from moving.

We use incident mates to constrain the 2 circles. This ensures that they will remain tight as if they were screwed together. We then use the concentric mate for the hook sides of the 9.605” and 10.134” legs. This is because these parts can not be incidentally mated to each other and they will be incidentally mated to the crankshaft itself.

Forming this assembly individually saves us a lot of time when we are putting together the main assembly. This is because, without this assembly, we would need to form 36 additional mates. Since this triangle shape itself does not change, this assembly could be formed independently. This would be a different case however, if we were creating a linkage like the Jansen Linkage, where triangles tend to change angles.

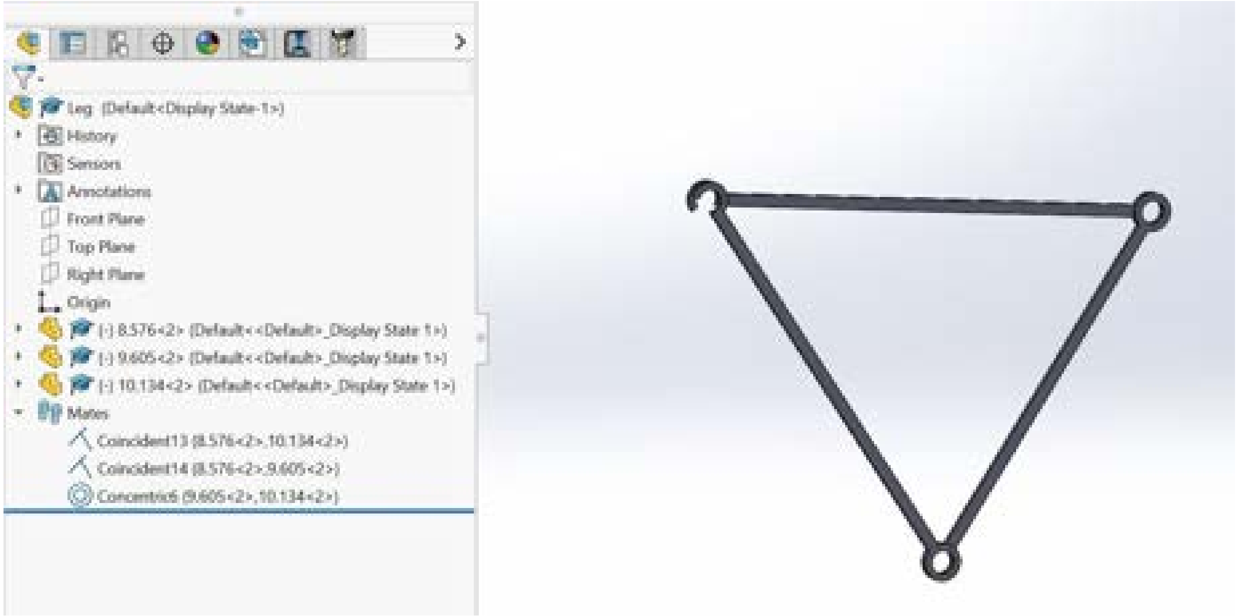


Figure 54: Assembly of 8.576”, 10.134”, and 9.605” leg

Main Frame

We then create the mainframe using the dimensions acquired from Linkage:

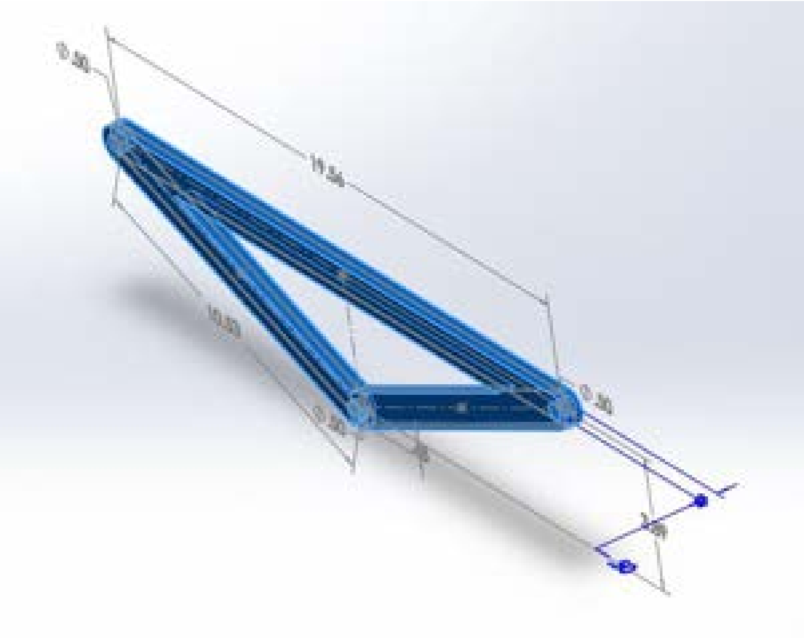


Figure 55: Main Frame

Crankshaft

We then start designing the crankshaft. We first start by making this simple cylinder:

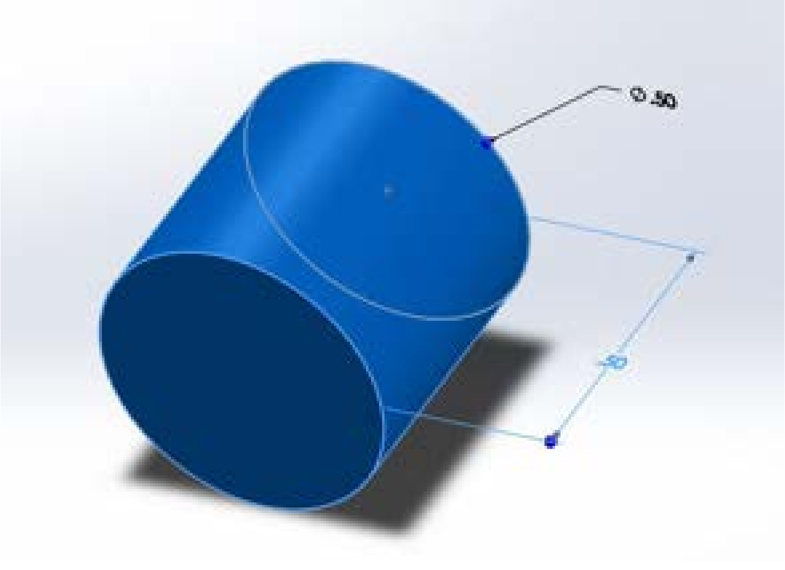


Figure 56: Main journal

We then create the counterweights. Note here that the width and radius of the counterweights is arbitrary, but the length between the two centers of radii is not arbitrary, in fact, this is the size of one of the bars of the 4-bar linkage, and thus, this size is crucial and important to the dynamics of the whole rover.

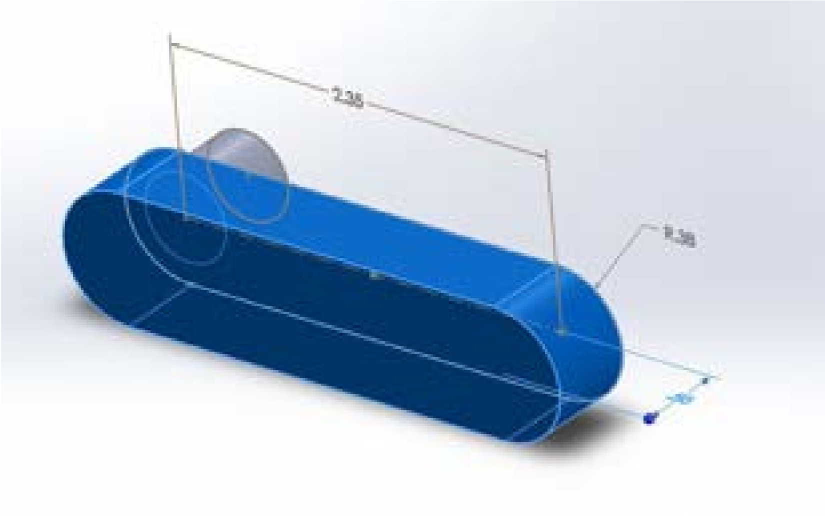


Figure 57: Counterweights

We then create our first crankpin:

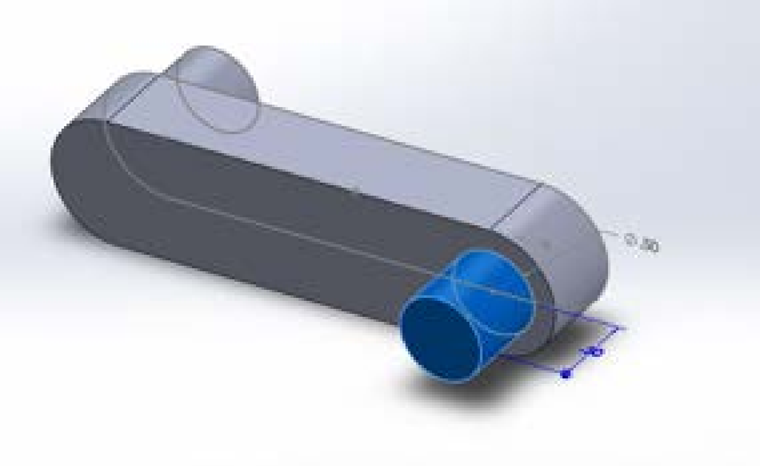


Figure 58: Crankpin

We then sandwich that crankpin with another counterweight:

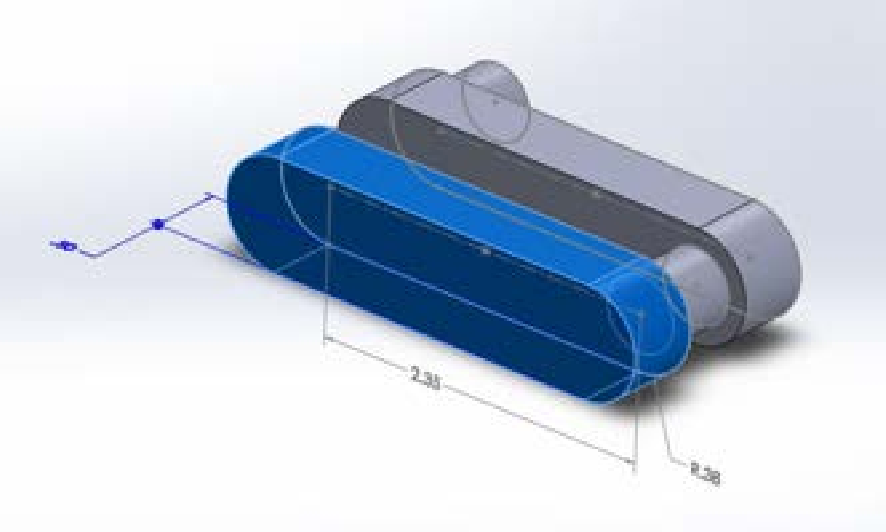


Figure 59: Counterweight

We proceed to add a main journal. By that, we are done with our first section of the crankshaft

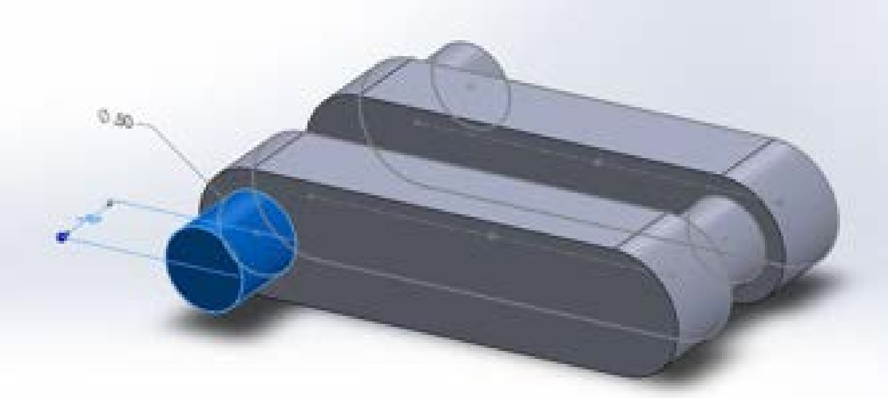


Figure 60: Main journal

We then offset by 120 degrees to create our second section of the crankshaft. To move to the next section, just rotate 120 degrees and repeat the past few steps

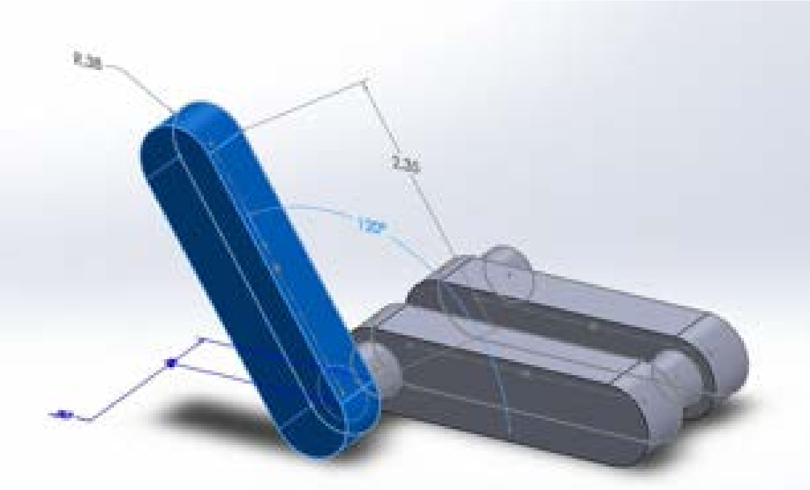


Figure 61: Second section of crankshaft

At the end, our crankshaft will look like this:

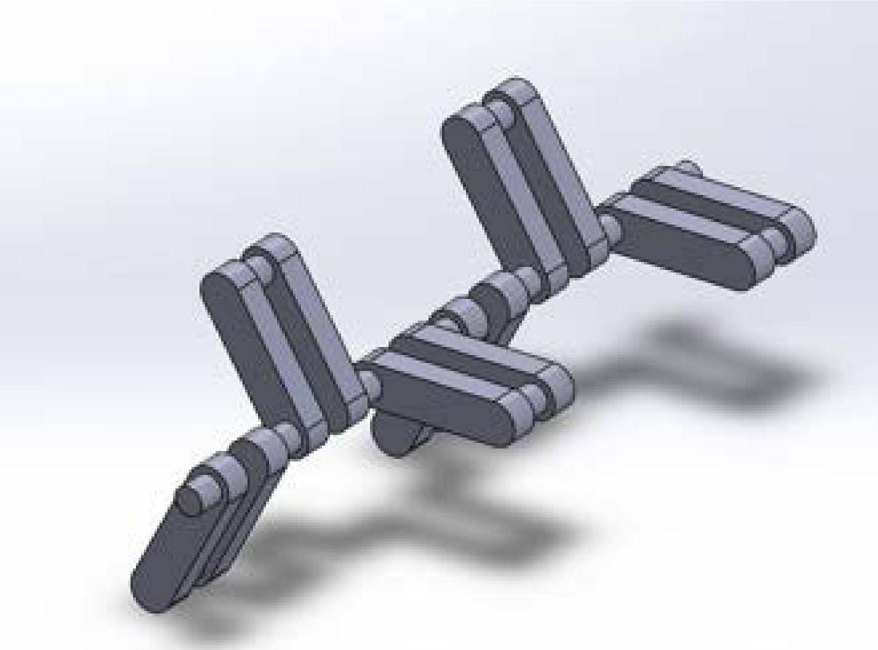


Figure 62: crankshaft

We finally add a D-shaped dowel hole so that the small gear’s D-shaped dowel pin can fit perfectly in and rotate the crankshaft:

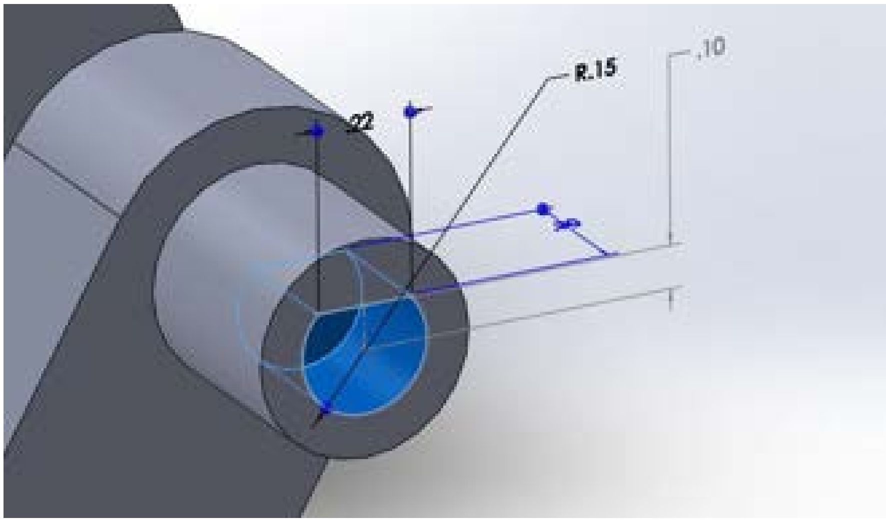


Figure 63: D-shaped dowel hole in crankshaft

Rod

We then create this simple rod that will go through the top two holes of the main frame. This is just to add stability and hold the structure together:

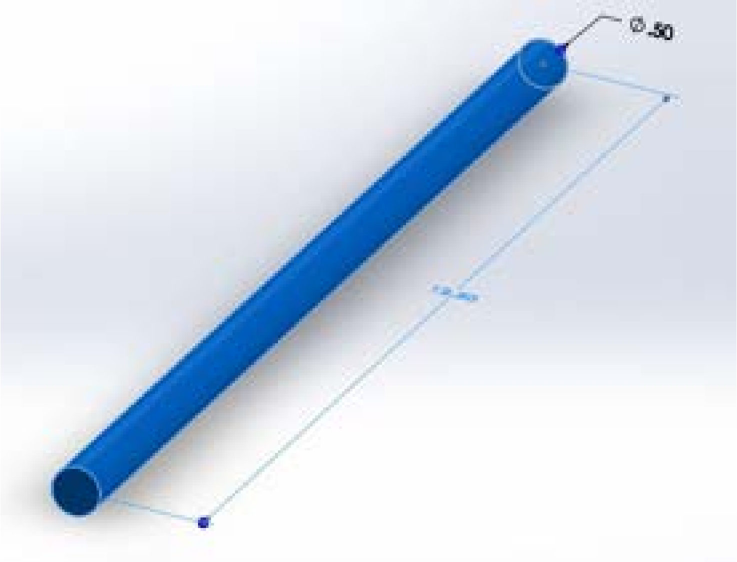


Figure 64: Rod

Small Gear

We then create the small gear. We start by creating a simple cylinder:

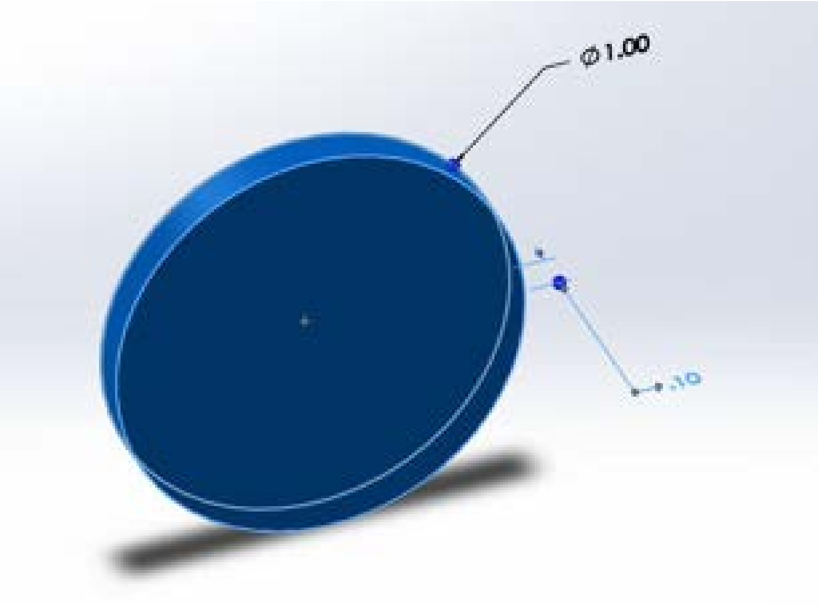


Figure 65: Base of small gear

We then extrude the D-shaped dowel pin, which will fit inside the D-shaped dowel hole in the crankshaft:

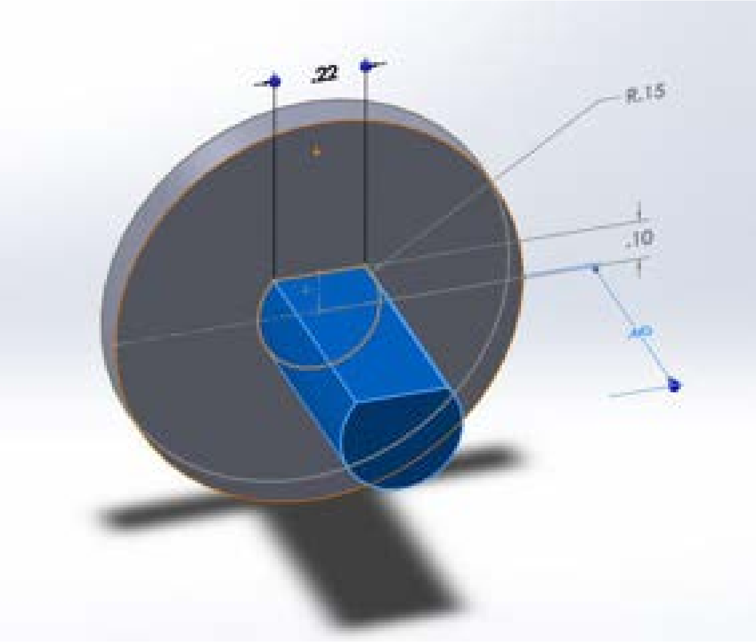


Figure 66: D-shaped dowel pin

We then sketch the shape of the gear teeth. The side lengths of this triangle were chosen arbitrarily. We create a circular pattern of 50 triangles that are equally spaced throughout a full circle:

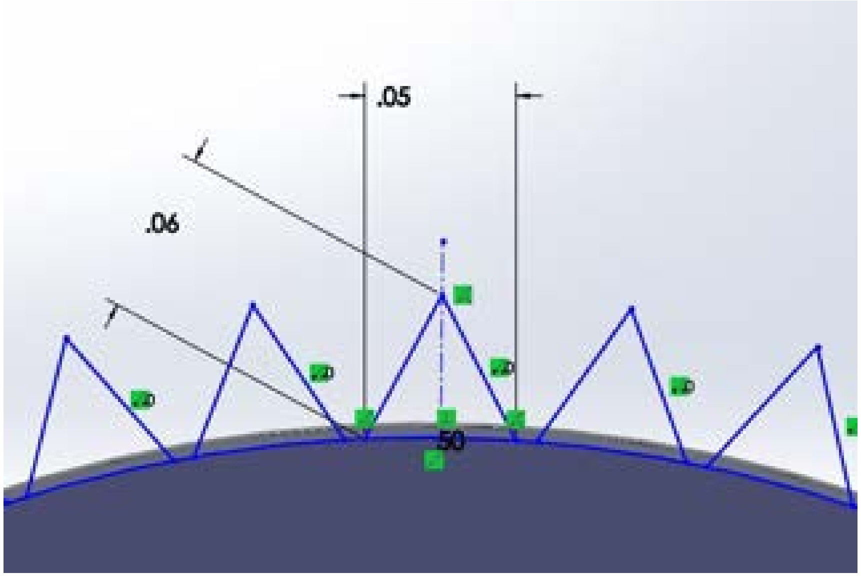


Figure 67: Sketch for small gear teeth

By extruding this sketch, we get the gear teeth:

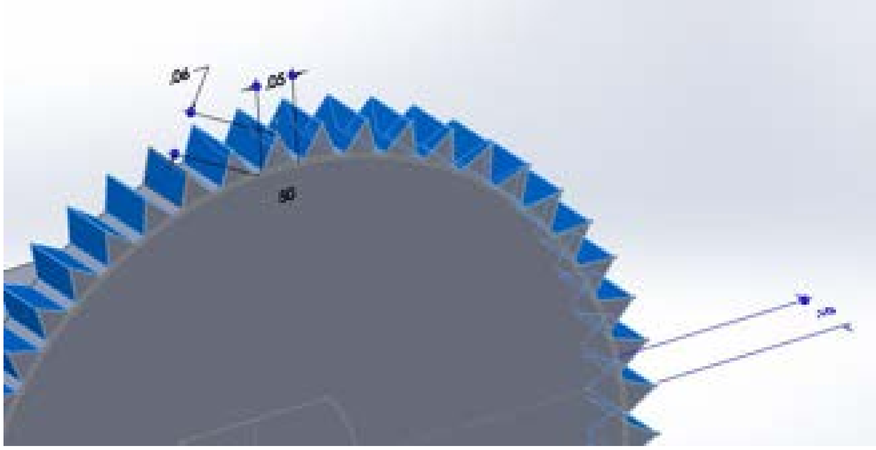


Figure 68: Extruded small gear teeth

We then add a 0.01” filet to the edge of the teeth to ensure we don’t leave any “razor edges” and so the gears don’t get jammed

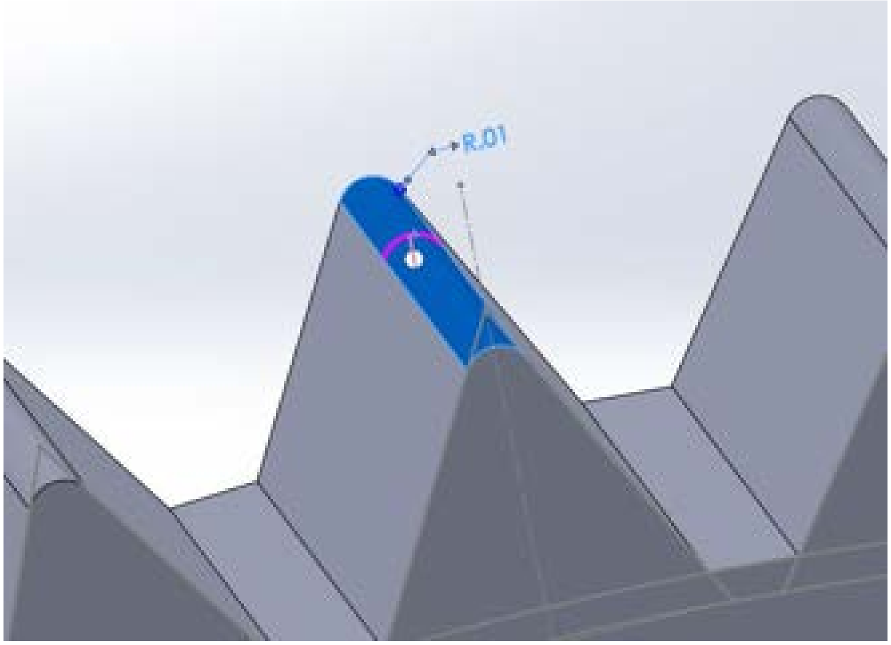


Figure 69: Small gear teeth filet

Big Gear

We now go on to create the big gear. We start off by making a cylinder where the motor shaft could fit into. The diameter of the hole is the same as the diameter of the motor shaft:

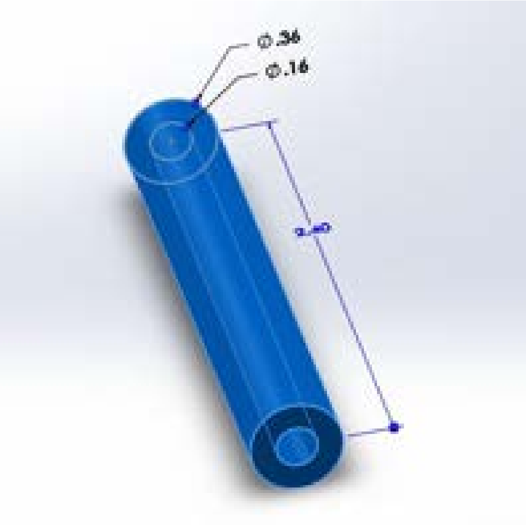


Figure 70: Big gear motor shaft connection

Next, we create the base feature for the main sketch. This will be a 4.25” radius cylinder. This number is not arbitrary, it is the distance between the center of the motor shaft and the highpoint of the small gear. This ensures that the two gears actually come in contact.

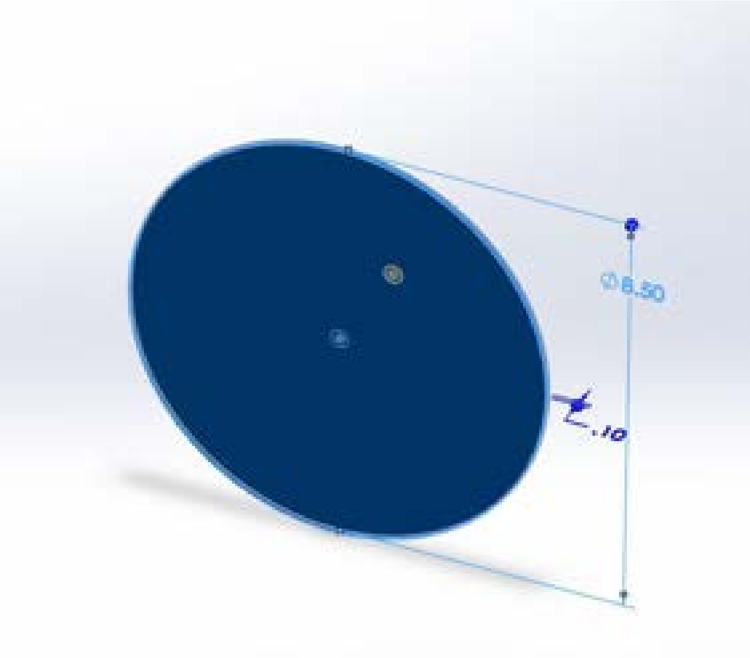


Figure 71: Base feature for big gear

We then create a tooth for the big gear. These dimensions were chosen arbitrarily but mindfully to ensure that the two gears actually interlock:

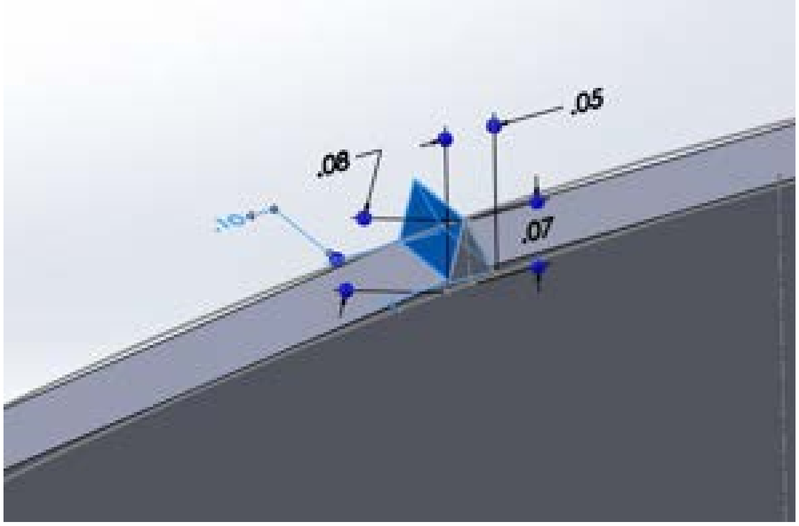


Figure 72: Single tooth of big gear

We then add a 0.01” radius filet:

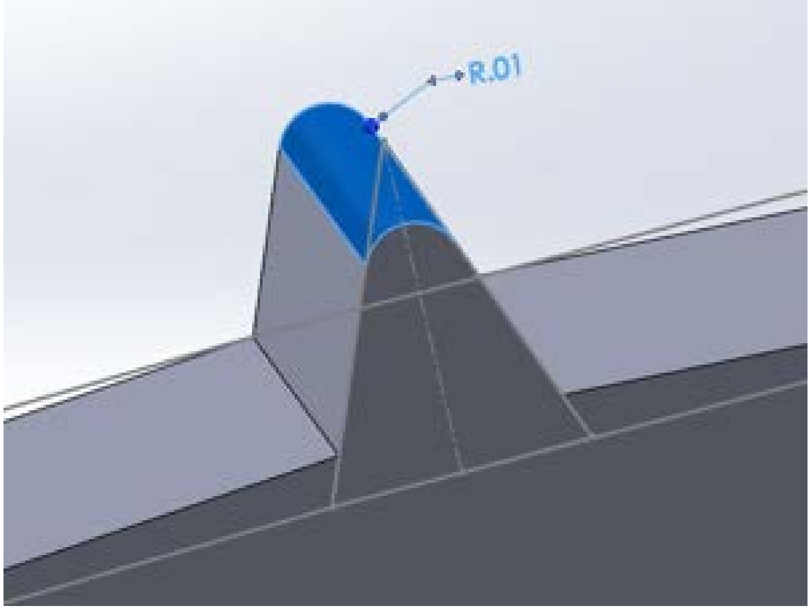
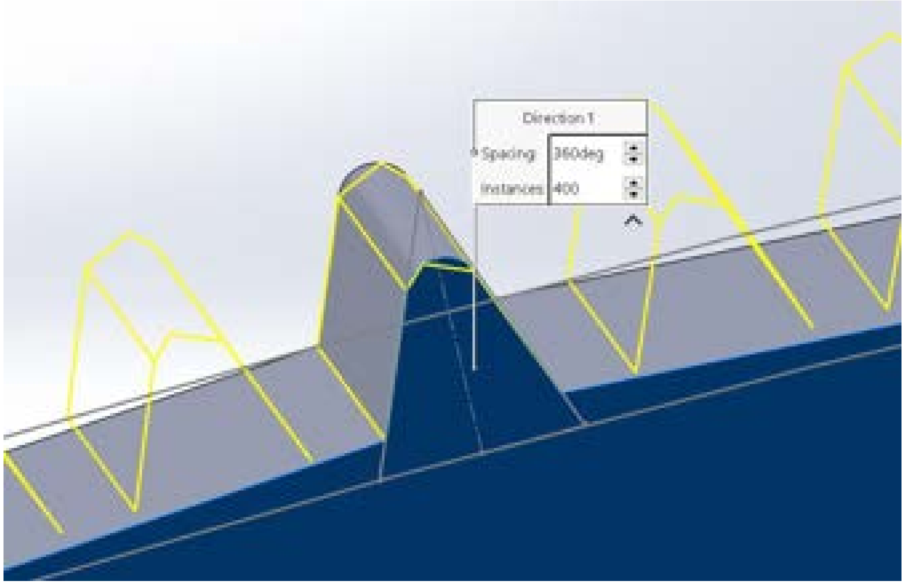


Figure 73: Big gear teeth filet

We then use a circular pattern of 400 teeth all around the gear base feature:



Direction 1
Spacing 360deg
Instances 400

Figure 74: Big gear teeth

To remove unnecessary material, we decided to remove some of the gear's base features. It also looked more appealing when assembled on the TitanWandelaar compared to a normal one. We first create a shape formed of two concentric arcs and connect them with straight lines, and then cut extrude that sketch. After that, we create a circular panel of 4 of these shapes to yield the following shape:

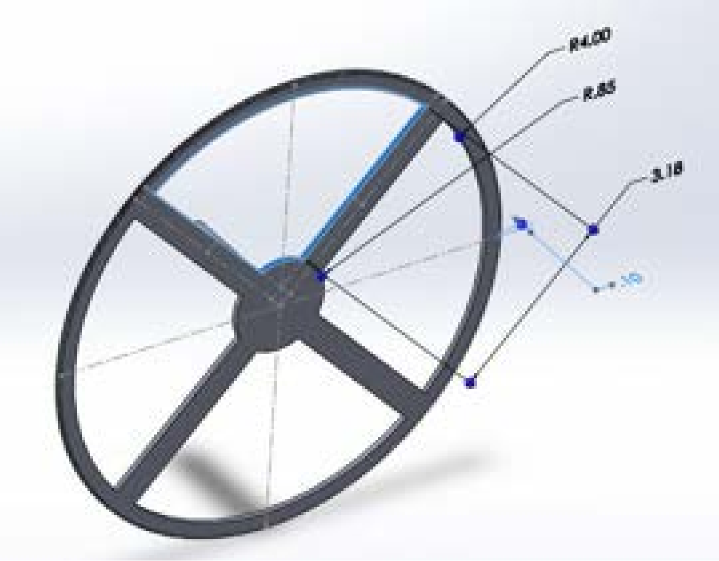


Figure 75: Sketch for cutting unnecessary material off big gear

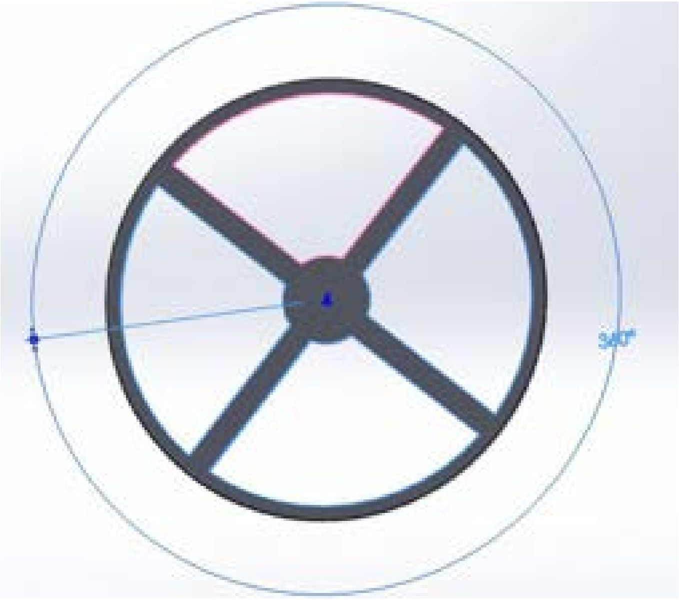


Figure 76: Circular pattern of 4

Electronics Box

We then create the electronics box. To do, we start off by making an 8.50”x8.00”x2.00” box:

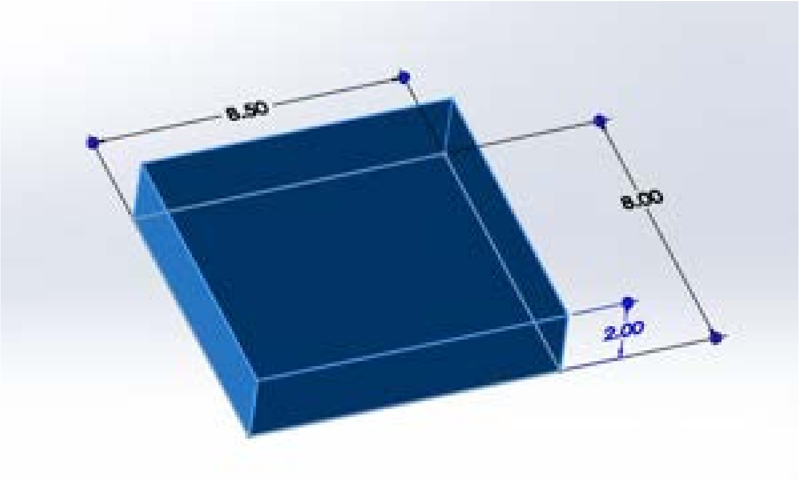


Figure 77: Base feature of electronics box

We then apply a 0.10” shell to hollow out the box:

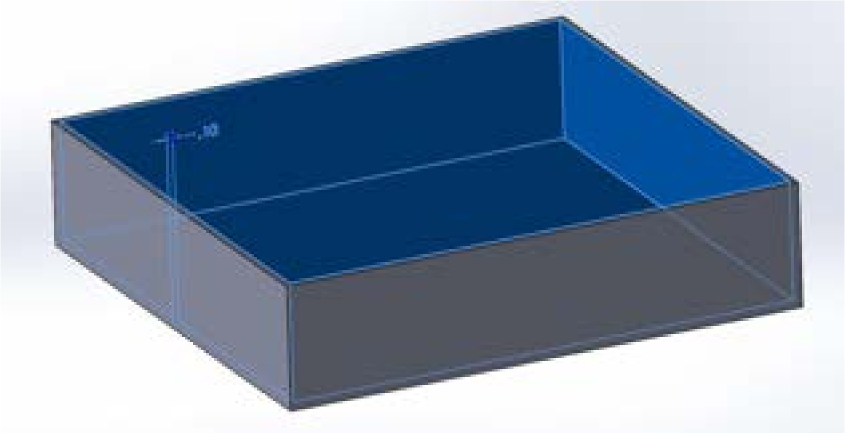


Figure 78: Shelled out box

To mount the electronics down to the main frame, we need to make mounting brackets. The dimensions of this mounting bracket are mostly arbitrary except for the 0.50”, since that is meant to be the same thickness as the thickness of the main frame to avoid any stubs:

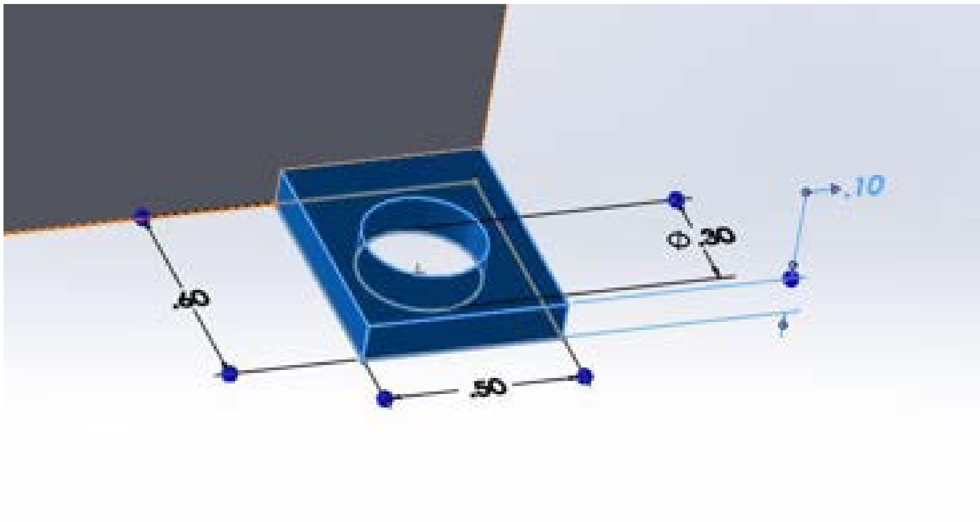


Figure 79: Mounting bracket

We then create a hole for the big gear to be able to connect to the motor shaft:

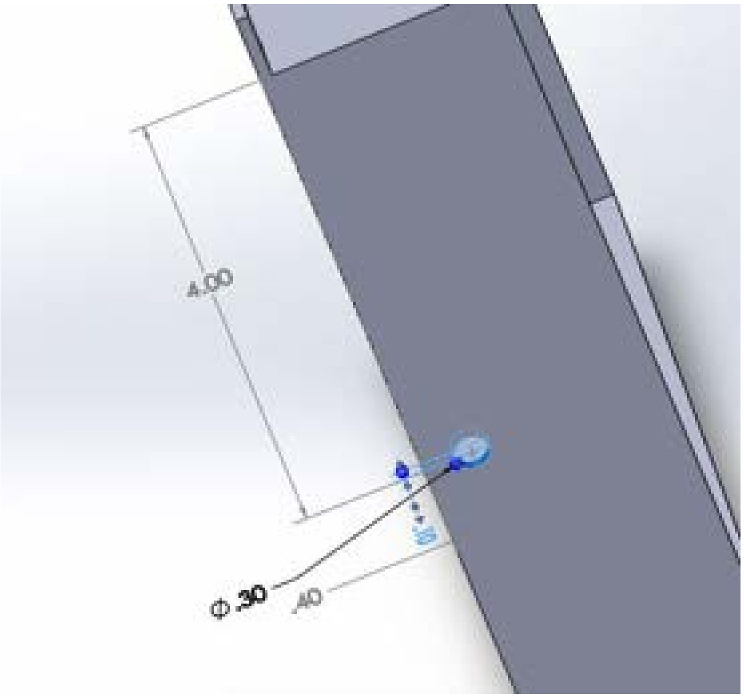


Figure 80: Hole to allow for motor shaft to big gear connection

We then create a rectangular hole to fit the LCD screen that is a part of the temperature detection system:

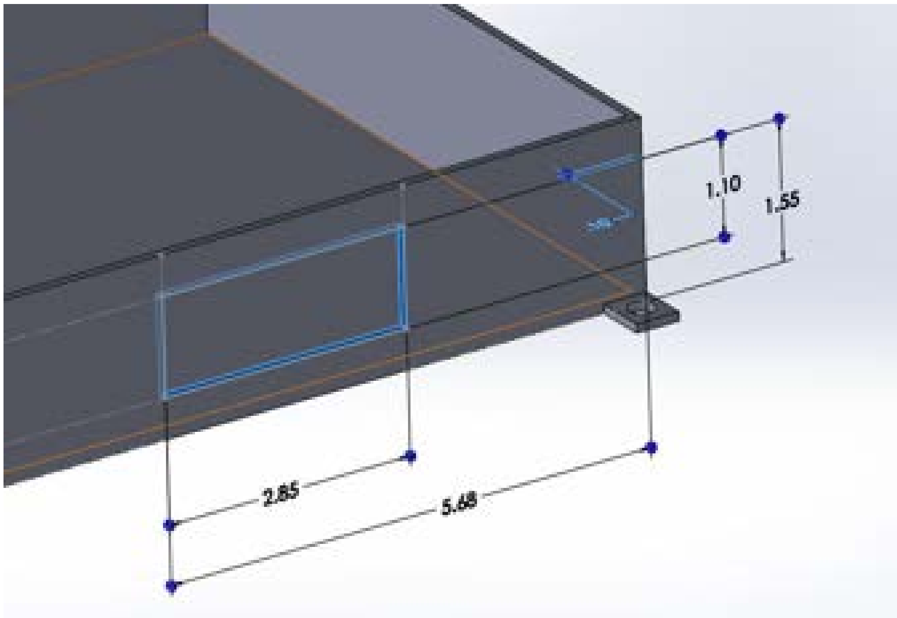


Figure 81: Hole for temperature detection LCD

The parts of the electronics assembly were found on GrabCAD. Not much thought was put into the dimensions of these items since we are not actually connecting the circuit in Solidworks. We are just placing them inside the box to show that they all fit in there, with even a little bit of extra room for a possible payload.

Figure 82: Electronics box assembly

We then create a lid for the electronics box with a solar panel on it:

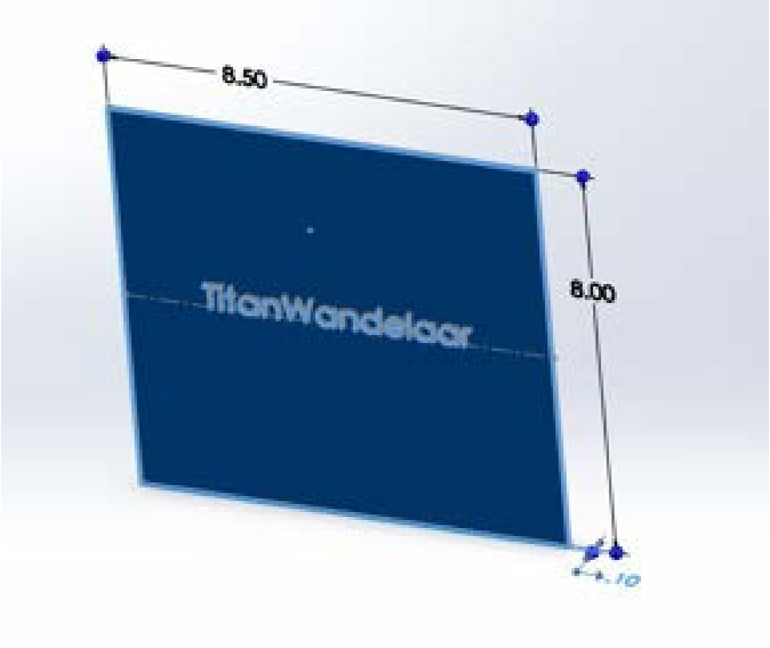


Figure 83: Electronics box lid

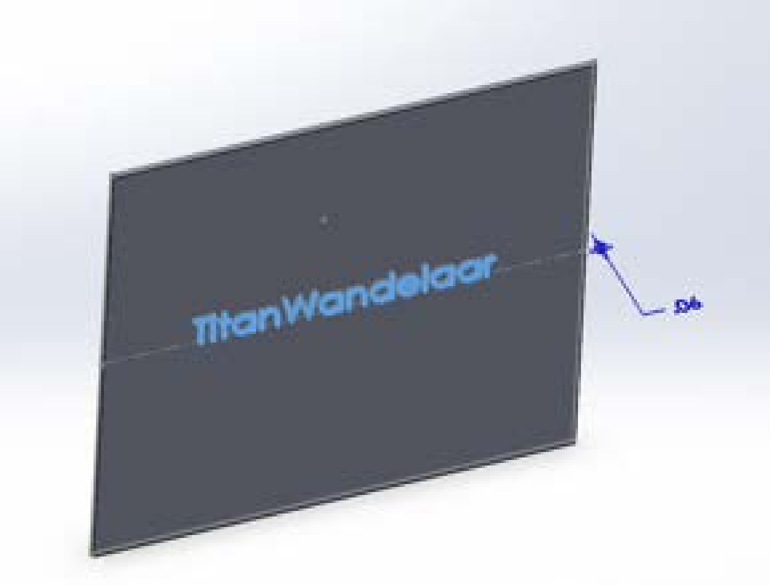


Figure 84: TitanWandelaar engraving

Solar Panel

We also created a solar panel that is to be mounted on top of the electronics box’s lid. This is to power the electrical components

We first create a 3.00”x7.00” panel that has 0.10” thickness:

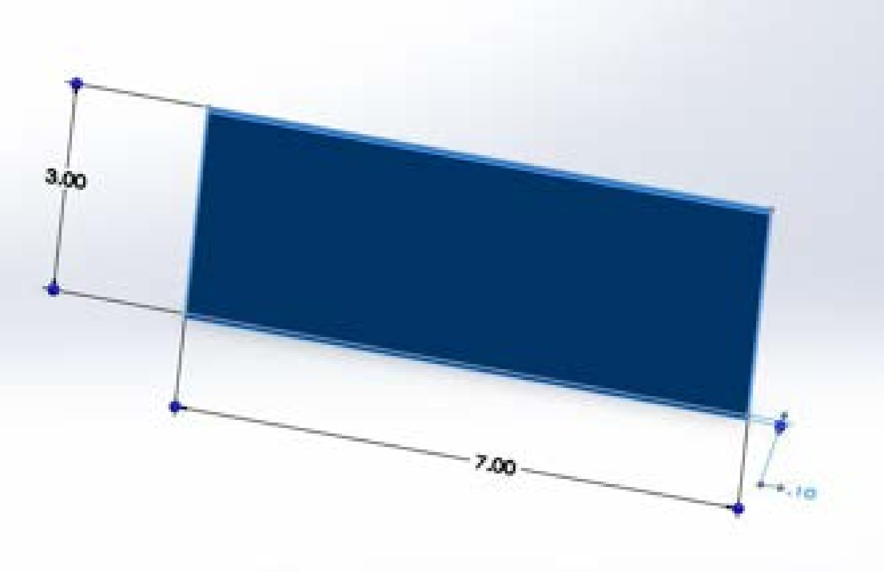


Figure 85: Solar panel base feature

We then add in the solar cells on top of the solar panel’s base. Each cell was 0.625”x0.59” with a 0.075” filet around the corners. We use the linear pattern feature to create 40 of these that were 0.10” away from each others in both the x and y directions:

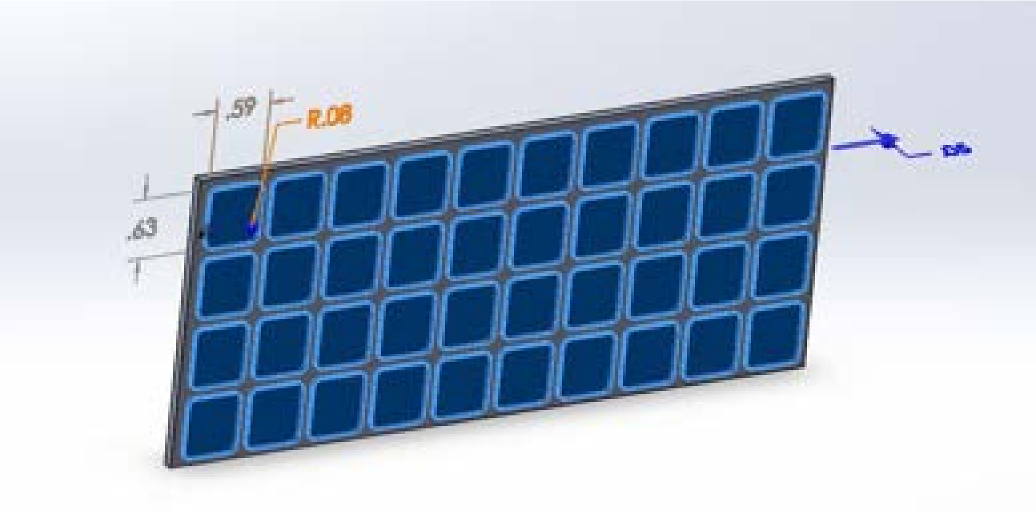


Figure 86: Solar cells

Appendix I: Bill of Materials

This appendix will include the materials that were used in the production of the prototype of the TitanWandelaar, as well as the price and the suggested purchase store for each item. A small description will accompany each item to give some context as to why that item was chosen.

Table 3: Bill of Materials for TitanWandelaar

Item(s)	Description	Qty	Vendor	Part Number	Unit cost	Total cost
Electrical components						
Arduino	The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.	1	Arduino	7630049200050	27.60	27.60
Temperature sensor	DS18B20 Temperature Sensor Simplify connecting a waterproof temperature sensor to your project with this adapter module kit Pull-up resistor included on the adapter module, an external resistor is not required to connect directly to the GPIO of the Raspberry Pi Waterproof digital temperature sensor DS18B20 on 100cm cable with adapter module	1	Amazon	41112200	8.99	8.99

	Measure the temperature directly in water or soil DS18B20 Wiring: Red = VCC, Yellow = Data, Black = GND					
Precipitate sensor	HiLetgo 3pcs LM393 Rain Drops Sensor Weather Moisture Monitor Sensor Humidity Sensitivity Module Nickeled Plate 3.3-5V for Arduino The sensor uses high quality FR-04 double-sided materials, large area of 5.0 * 4.0CM, and nickel-plated surface, with oxidation resistance, conductivity, and life expectancy superior performance Comparator output, the signal clean, good waveform, strong driving ability, more than 15mA The raindrop board and the control board are separate, easy to lead the line TTL level output, TTL output valid signal is low. Drive capacity of about 100MA, direct drive relays, buzzers, small fans, and so on. Adjust sensitivity by potentiometer	1	Amazon	49101600	2.06	6.19
NeoPixel	The ultra bright ws2812b smart full-color LED pixel is individually addressable. It has 256 brightness	1	Amazon	4334419429	0.1599	15.99

	display and full 24-bit (16,777,216) color display. You can set to any color or animation. It’s put on a small PCB board that allows you to solder it in whatever configuration you want. It’s easy to wire up and control. You just need one digital pin plus 5V and ground to control as many LEDs. The chainable design means no crossed threads. Compatible with Arduino, Raspberry Pi, Teensy, T1000S,K1000C etc programmable controllers. It can also be controlled by SP105E SP108E SP110E pre-programmed app controllers and SP106E SP107E music controllers, SP501E and MHCTRWF5V smart WiFi Controllers.					
Potentiometer	These variable resistors are easy to connect and have a long grippy adjustment knob, can easily plug into breadboards or perforated boards. There are 12 pieces of breadboard potentiometers in total, enough quantity to satisfy your various needs, bringing convenience.	1	Amazon	B09G9TBY38	0.8325	9.99

	These potentiometer assortment kits are made of quality material, lightweight and sturdy, safe and stable, you can use them for a long time with confidence.					
Breadboard + jumper wires	WHAT YOU GET – Package comes with 2 pieces 830-Points large bread boards, 2 pieces 400-Points mini breadboards and 120 pieces jumper wires for use in electronic and electrical experiments projects. HIGH QUALITY EXPERIMENT BREADBOARD – The solderless breadboard is made of ABS plastic and can be fixed on a flat surface with adhesive tape at the back. The 830 Points breadboard is about 6.5” x 2.1” x 0.3“in size, and the 400 Points breadboard is about 3.2” x 2.1” x 0.3” in size. JUMPER WIRES RIBBON CABLES – The multicolored insertion wire is made of high-grade copper core, and is about 8” (20cm) in length. Totally 120 single axes that includes 40 pieces male-male, 40 pieces male-female and 40 pieces female-female. FITS FOR ARDUINO PROTO SHIELD – The	1	Amazon	4330118957	14.99	14.99

	breadboard and jumper wires kit is ideal for setting up a trial circuit, prototype or experimenting. NOTE – If you have any problem with our product, you can click the “seller contact” button in your Amazon account to contact us. We will be delighted to help you and reply as soon as we can					
Electrical tape	HEAT RESISTANT: This tape can handle temperatures from 14 Degrees Fahrenheit (-10 Degrees Celsius) to 194 Degrees Fahrenheit (90 Degrees Celsius) UL LISTED: With this stamp of approval, you’ll know that this tape has gone through independent testing to ensure electrical safety FLEXIBLE: This tape is ideal for stretching out and wrapping around electrical wiring, keeping it fully insulated from the elements STRONG ADHESIVE: This tape is meant to permanently adhere to your subjects, so you won’t have to worry about it coming loose any time soon MADE IN THE UNITED STATES OF AMERICA Commercial grade, 7 mil vinyl electrical tape	1	Amazon	4330118957	1.99	1.99

	Ideal for holding, protecting and insulating Highly flexible, stretchy and conformable Rated for temperatures up to 194-Degree UL LISTED					
LCD Monitor	GeeekPi 2-Pack I2C 1602 LCD Display Module 16X2 Character Serial Blue Backlight LCD Module for Raspberry Pi Arduino STM32 DIY Maker Project Nanopi BPI Tinker Board Electrical IoT Internet of Things The LCD1602 is an industrial character LCD that can display 16x2 or 32 characters at the same time. The principle of the LCD1602 liquid crystal display is to use the physical characteristics of the liquid crystal to control the display area by voltage, that is, the graphic can be displayed. The 1602 uses a standard 16-pin interface, and our display module is a module that provides I2C functionality. I2C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL),pulled up with resistors. Typical voltages used are +5 V or +3.3 V although systems with other	1	Amazon	B07S7PJYM6	5.495	10.99

	voltages are permitted. It can be operated as long as it supports the I2C development board. For example, the common Ar-duino, raspberry pi, Stm32 and so on. Features: Easy to use; Less I/O ports are occupied; Support IIC Protocol; The I2C LCD1602 library is easy to get; With a potentiometer used to adjust backlight and contrast; Blue backlight; Power supply: 5v; I2C address is: 0x27.					
Solar Panel	AOSHIKE 10Pcs 5V 30mA Mini Solar Panels for Solar Power Mini Solar Cells DIY Electric Toy Materials Photovoltaic Cells Solar DIY System Kits 2.08"x1.18"(5V 30mA 53mmx30mm) Adaptation: Solar yard lighting,small household lighting systems,Solar street lighting. Suitable for all kinds of low-power electrical appliances,emergency lights, advertising lights,household lights, electric fans, such as solar water pumps,small solar systems.	1	Amazon	B07BMMHMSJ	1.599	15.99

	Features: 1. High conversion rate, high efficiency output 2. All data are actually measured under the condition that the solar light is sufficient. In full sunlight, the voltage will be higher than 5V 3. Unique technology to make components beautiful and strong anti-snow, easy to install 4. The unique technology to avoid freezing water within the framework and deformation					
Motor	Gikfun 1.5V-6V Type 130 Miniature DC Motors for Arduino Hobby Projects DIY (Case Pack of 6) EK1450 Type 130 Mini DC Motors for 1.5V to 6V, 3V is recommended Come with 6 inch (15cm) black and red wire leads for easy connection Motor size: 15 x 20 mm, Shaft diameter: 2.0 mm, Shaft length : 9 mm Reference current: 0.35-0.4A, Speed : 16000 RPM (at 3V) Perfect for robotics projects, solar/battery-powered cars, brush-bots, fans/windmills, and more.	1	Amazon	39120000	1.61	9.68

Build						
3D Printer Filament	Ultimaker 2 ABS Filament - Yellow	1	Amazon	8718836374135	37.46	37.46
3D Printer	Resume Printing Function: Ender 3 has the ability to resume printing even after a power outage or lapse occurs. Easy and Quick Assembly: It comes with several assembled parts, you only need about 2 hours to assemble 20 nuts well. Advanced Extruder Technology: Upgraded extruder greatly reduces plugging risk and bad extrusion; V-shape with POM wheels make it move noiseless, smoothly and durable. Safety Protected Power Supply: Only needs 5 minutes for the hot bed to reach 110 degrees. Strict Test: Strict testing for key components before delivery and life-time technical support available.	1	Amazon	0796862079093	189.00	189.00

Appendix J: Granta - Materials Selection

Granta is a software developed by Ansys that is mainly used by Material Scientists for materials selection and comparison. It gives extensive information about any given materials and allows the user to produce graphs and tables to pick the best fitting materials.

Building the structure that has been optimized through Finite Element Analysis (FEA) and topology optimization is best done with Additive Manufacturing. This technology has allowed engineers to increase stiffness-to-weight ratios, energy absorption and thermal performance by taking out unnecessary material as seen from simulations [19]. This means we can decrease the amount of materials being used to decrease the weight without affecting the strength or stiffness of a structure.

From the results of a paper that was published in The International Journal of Advanced Manufacturing Technology, "A State-of-the-Art Review on Types, Design, Optimization, and Additive Manufacturing of Cellular Structures," we were able to see that Ti-6Al-4V was the most used material in additive manufacturing, figure 85:

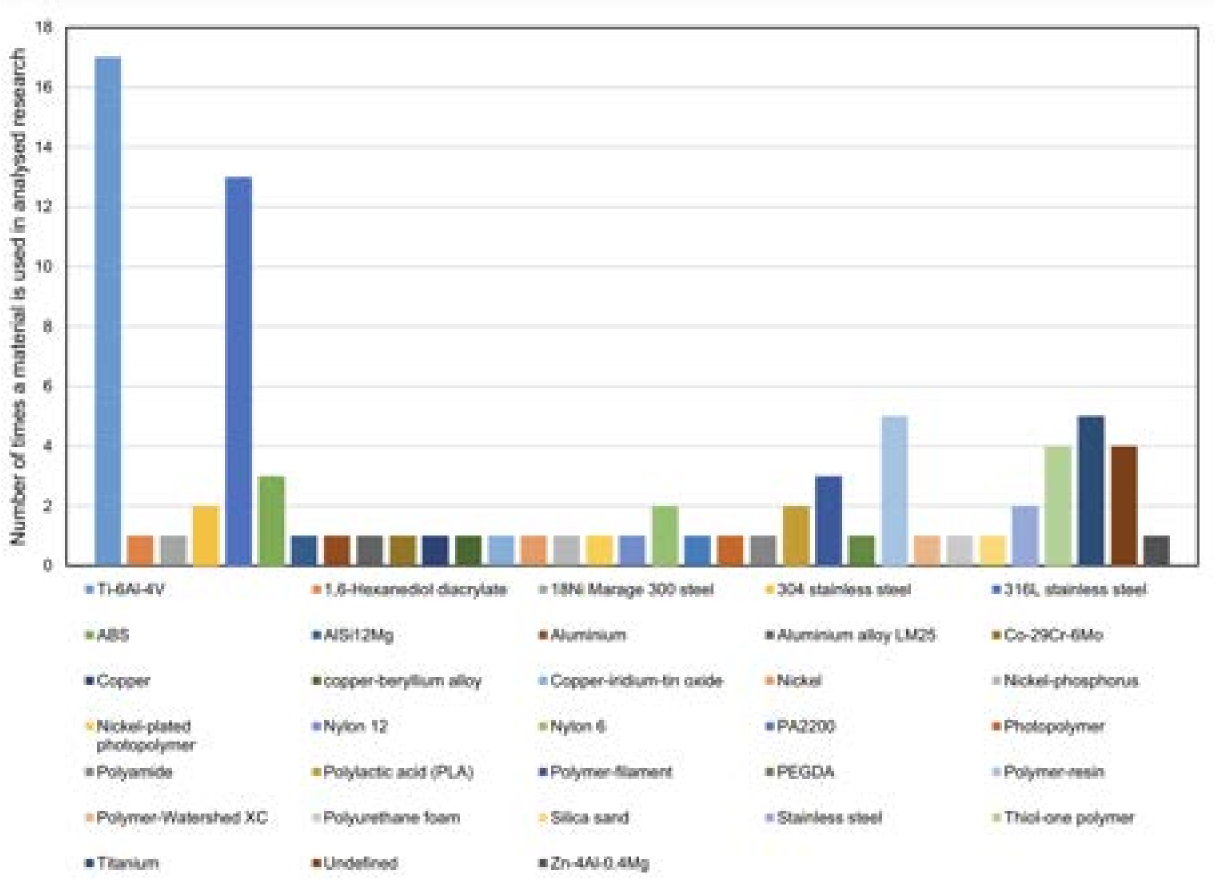


Figure 87: Ti-6Al-4V used the most number of times in analyzed research [20]

We then use Granta to further compare Ti-6Al-4V to other materials. We compose the following two graphs:

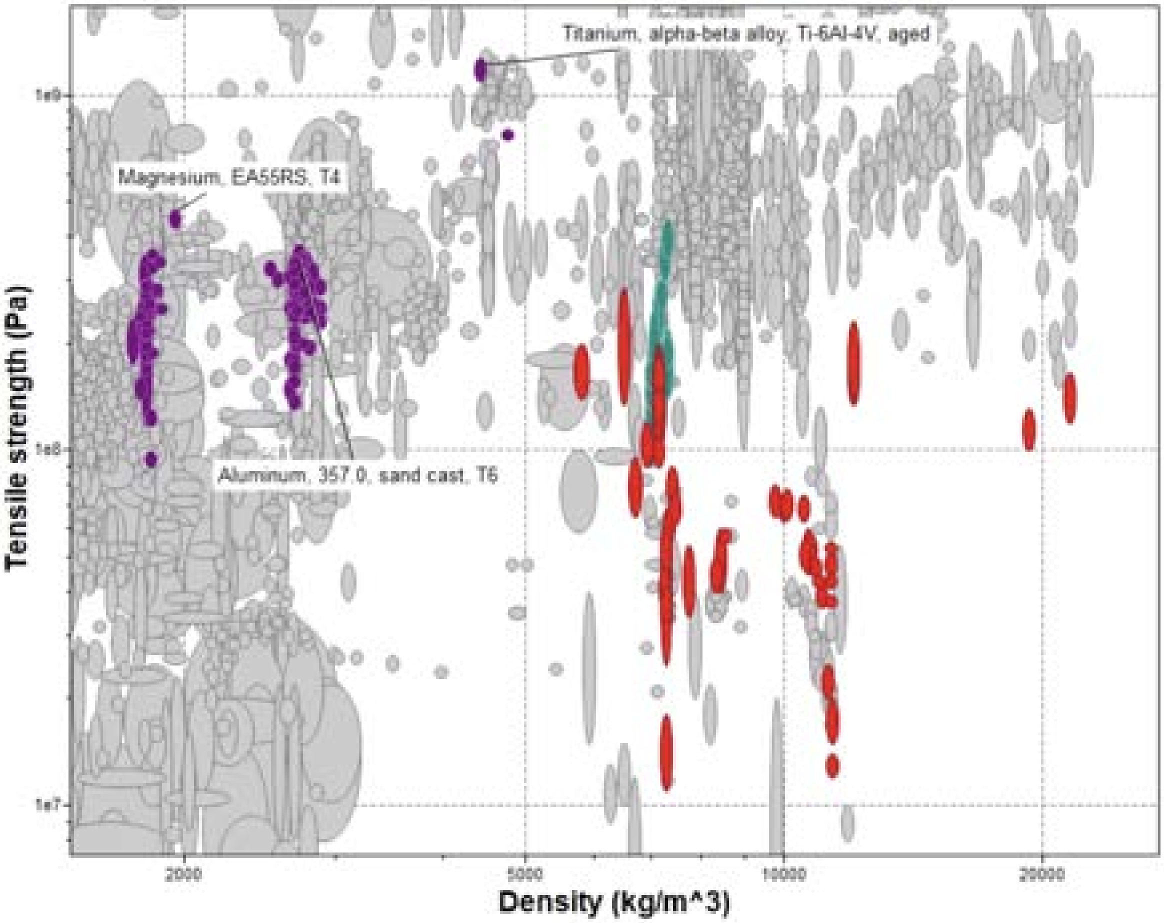


Figure 88: Tensile Strength (Pa) vs Density $\left(\frac{kg}{m^3}\right)$

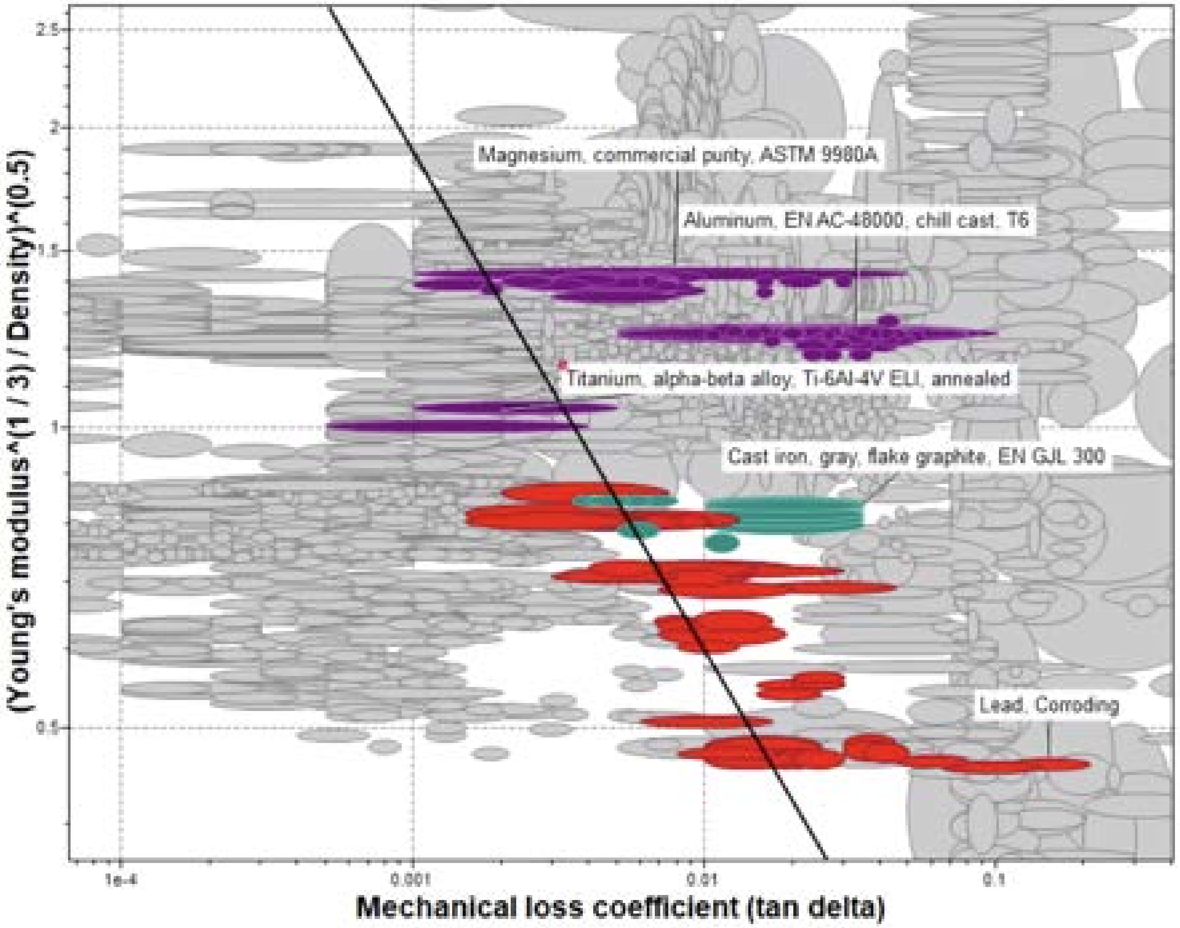


Figure 89: $\frac{(Young's\ Modulus)^{\frac{1}{3}}}{(Density)^{\frac{1}{2}}}$ vs Mechanical loss coefficient (tan delta)

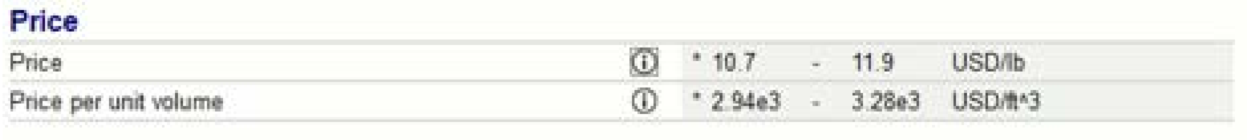
From these three graphs, we can see that Ti-6Al-4V is the strongest material with fairly low density, though not the lowest. Though from figure 87 it does not seem to be a super superior material, the results of figure 86 are more important for our purposes.

The commonly used metal in Mars rovers right now is Al-T62. Comparing the prices Al6061-T62 and Ti-6Al-4V:

Price

Price	ⓘ	* 0.885	-	0.989	USD/lb
Price per unit volume	ⓘ	* 149	-	168	USD/ft^3

Figure 90: Price and price per unit volume of Al6061-T62



Price

Price	ⓘ	* 10.7	-	11.9	USD/lb
Price per unit volume	ⓘ	* 2.94e3	-	3.28e3	USD/ft^3

Figure 91: Price and price per unit volume of Ti-6Al-4V

Though the price of Ti-6Al-4V is a lot more expensive, it is way stronger than Al6061-T62. The price of Ti-6Al-4V is still in the low price range of metals:

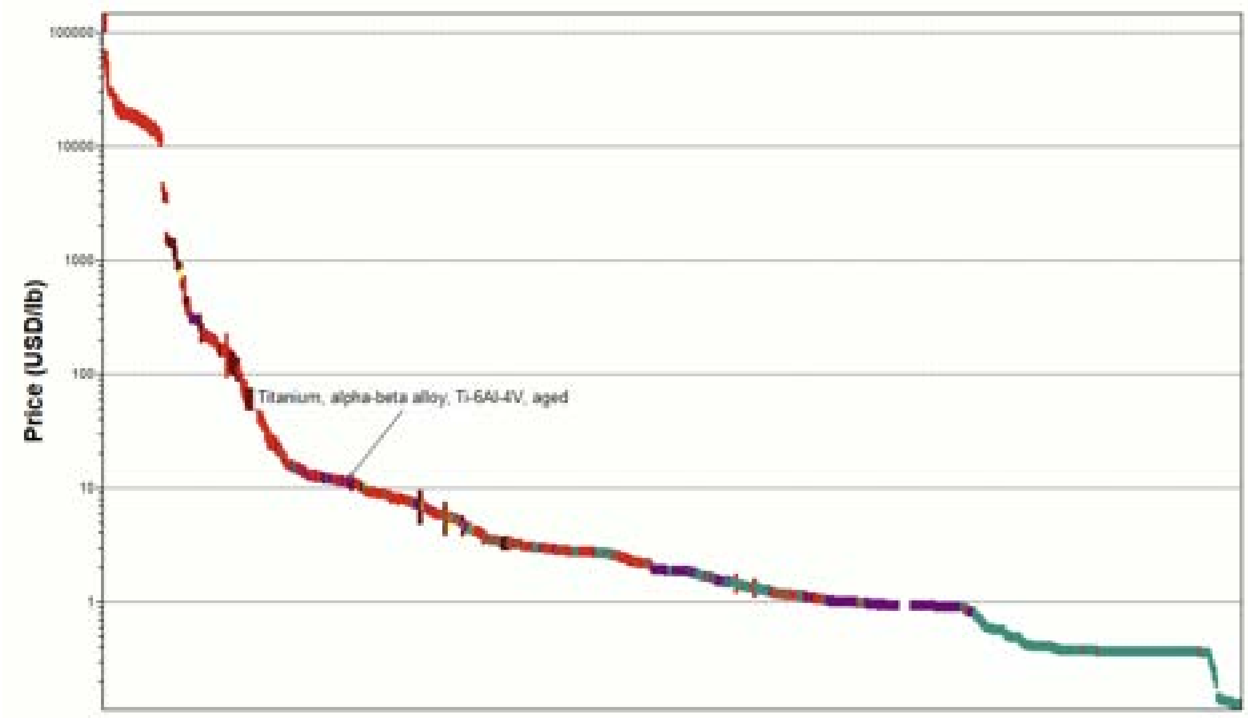
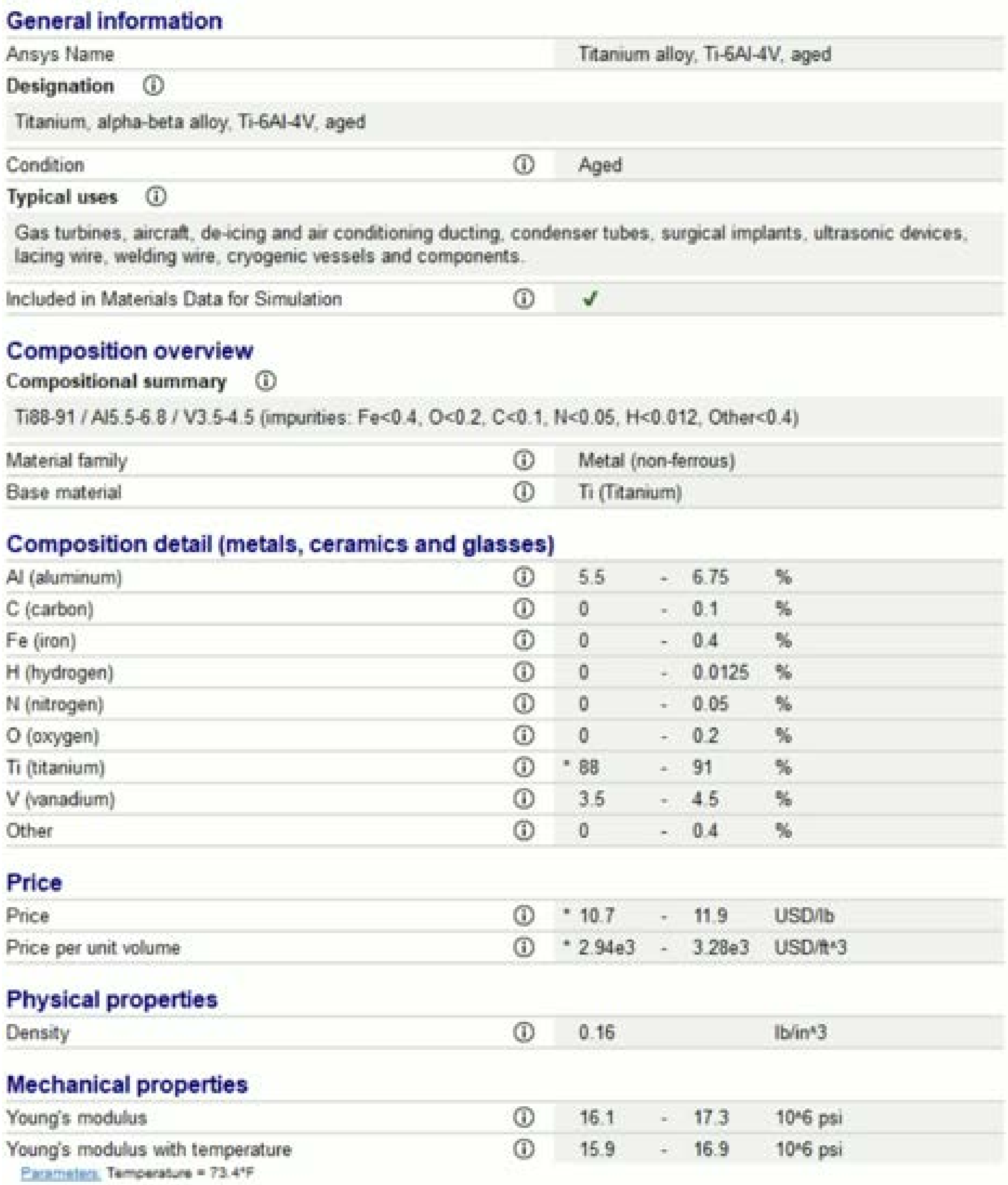


Figure 92: Price of Ti-6Al-4V compared to other metals

It is also easier to manufacture Ti-6Al-4V using Additive Manufacturing compared to Al6061-T62, which means the decreased Ti-6Al-4V can be optimized for stress and weight density while Al6061-T62 can not. Though more data is needed, this may mean that at the end Ti-6Al-4V will end up requiring way less material and thus will incur an overall smaller cost even though it costs more per pound. It may also mean that the structure may be less dense even though Ti-6Al-4V is more dense than Al6061-T62.

More data about Ti-6Al-4V will be inserted below for more context about all the properties regarding this material. All this data was acquired through Granta:



General information

Ansys Name	Titanium alloy, Ti-6Al-4V, aged

Designation

Titanium, alpha-beta alloy, Ti-6Al-4V, aged

Condition	Aged

Typical uses

Gas turbines, aircraft, de-icing and air conditioning ducting, condenser tubes, surgical implants, ultrasonic devices, lacing wire, welding wire, cryogenic vessels and components.

Included in Materials Data for Simulation	✓

Composition overview

Compositional summary

Ti88-91 / Al5.5-6.8 / V3.5-4.5 (impurities: Fe<0.4, O<0.2, C<0.1, N<0.05, H<0.012, Other<0.4)

Material family	Metal (non-ferrous)
Base material	Ti (Titanium)

Composition detail (metals, ceramics and glasses)

Al (aluminum)	5.5	-	6.75	%
C (carbon)	0	-	0.1	%
Fe (iron)	0	-	0.4	%
H (hydrogen)	0	-	0.0125	%
N (nitrogen)	0	-	0.05	%
O (oxygen)	0	-	0.2	%
Ti (titanium)	* 88	-	91	%
V (vanadium)	3.5	-	4.5	%
Other	0	-	0.4	%

Price

Price	* 10.7	-	11.9	USD/lb
Price per unit volume	* 2.94e3	-	3.28e3	USD/ft^3

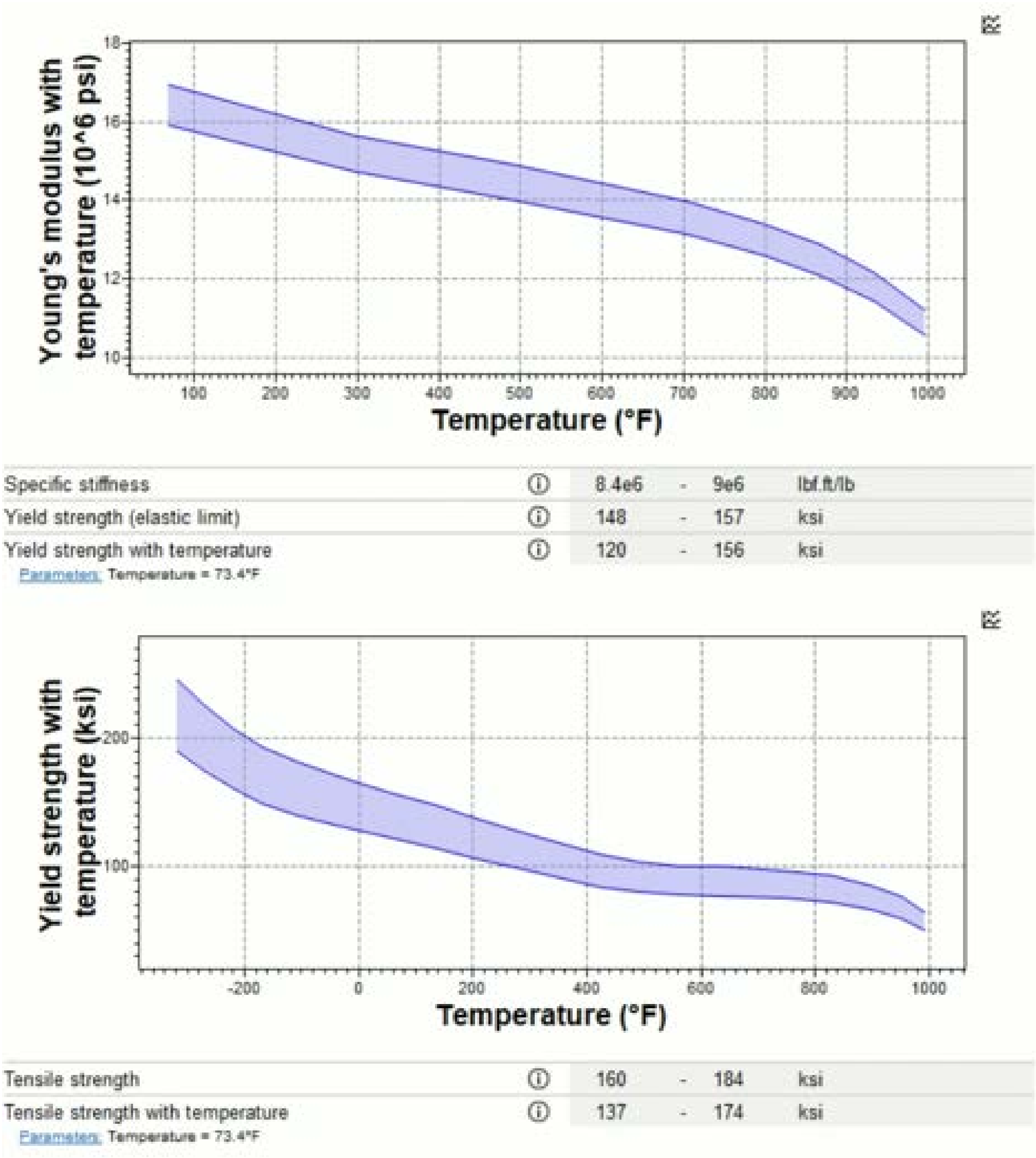
Physical properties

Density	0.16			lb/in^3

Mechanical properties

Young's modulus	16.1	-	17.3	10^6 psi
Young's modulus with temperature	15.9	-	16.9	10^6 psi

Parameters: Temperature = 73.4°F

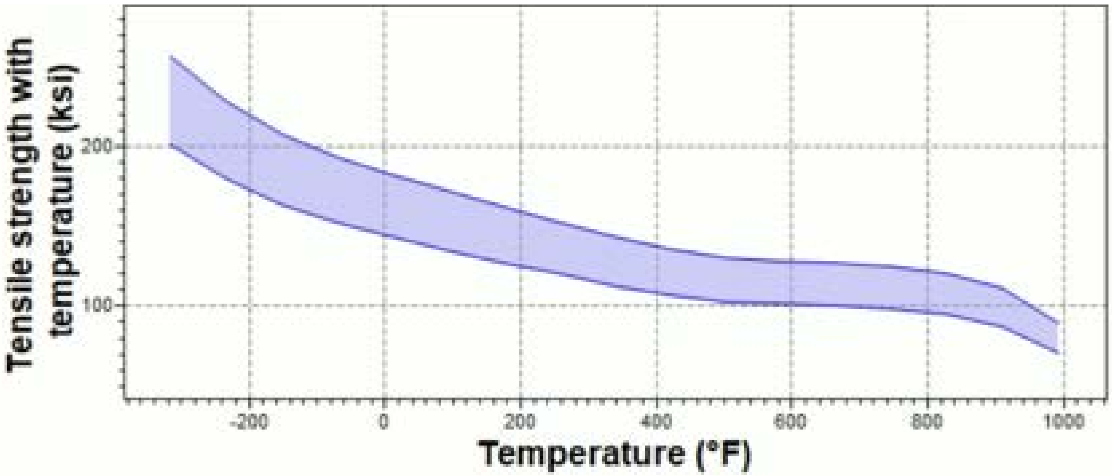


Specific stiffness	ⓘ	8.4e6	-	9e6	lbf.ft/lb
Yield strength (elastic limit)	ⓘ	148	-	157	ksi
Yield strength with temperature	ⓘ	120	-	156	ksi

Parameters: Temperature = 73.4°F

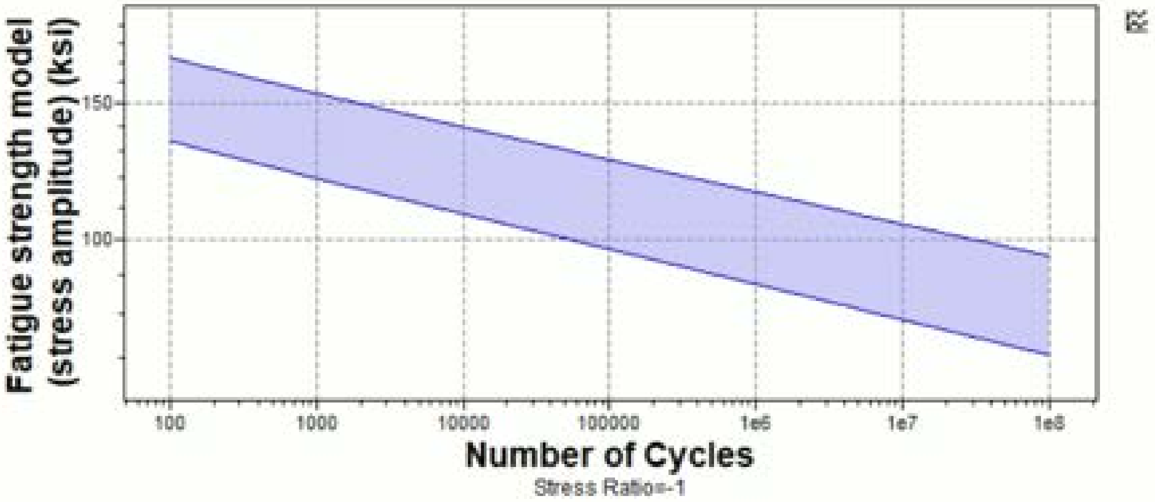
Tensile strength	ⓘ	160	-	184	ksi
Tensile strength with temperature	ⓘ	137	-	174	ksi

Parameters: Temperature = 73.4°F



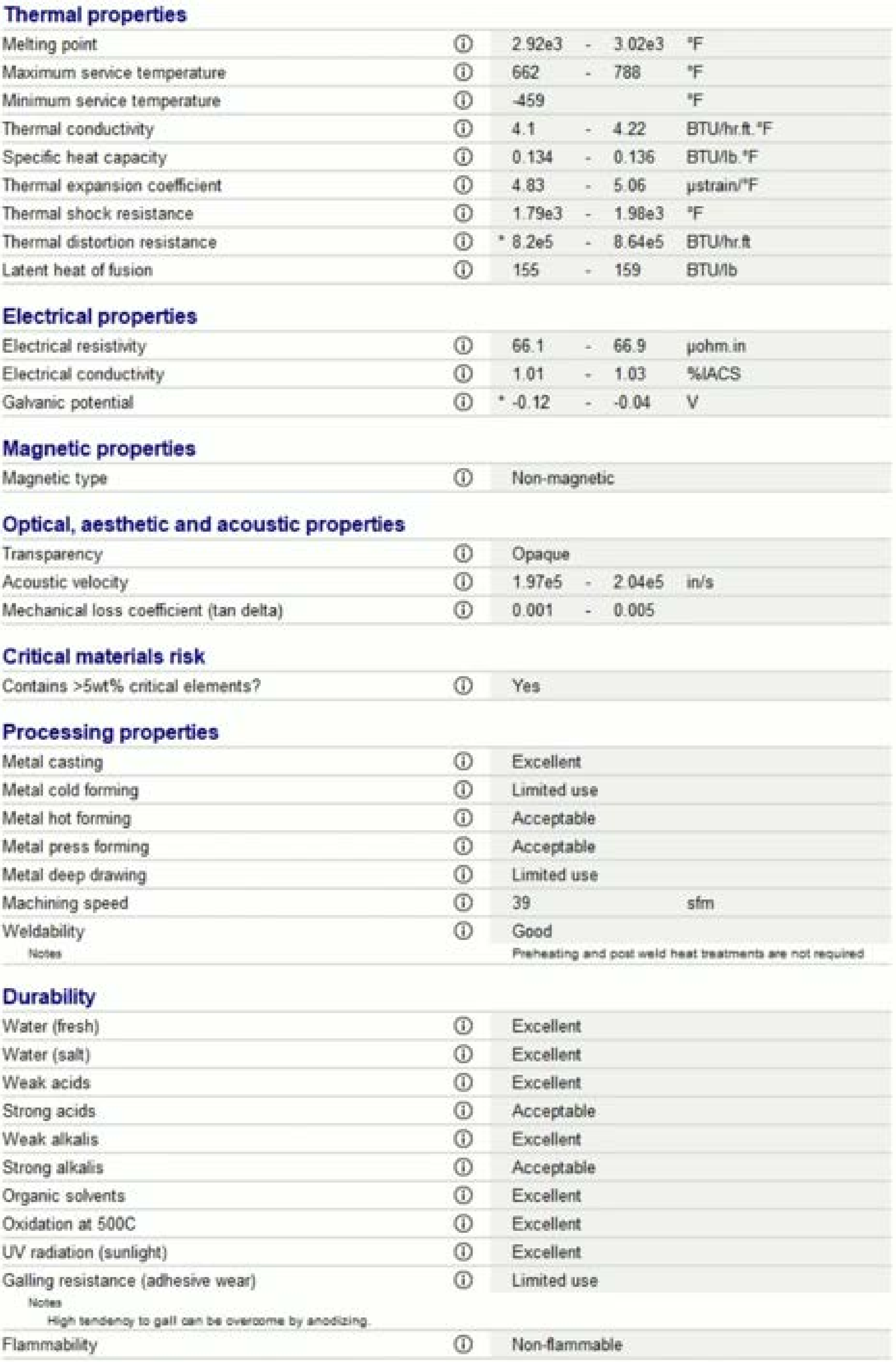
Specific strength	ⓘ	7.73e4	-	8.16e4	lbf.ft/lb
Elongation	ⓘ	8	-	13	% strain
Tangent modulus		328			ksi
Compressive modulus	ⓘ	* 16.1	-	17.3	10^6 psi
Compressive strength	ⓘ	* 160	-	167	ksi
Flexural modulus	ⓘ	* 16.1	-	17.3	10^6 psi
Flexural strength (modulus of rupture)	ⓘ	148	-	160	ksi
Shear modulus	ⓘ	5.8	-	6.53	10^6 psi
Bulk modulus	ⓘ	17.8	-	22.2	10^6 psi
Poisson's ratio	ⓘ	0.35	-	0.37	
Shape factor	ⓘ	11			
Hardness - Vickers	ⓘ	380	-	420	HV
Hardness - Brinell	ⓘ	361	-	400	HB
Elastic stored energy (springs)	ⓘ	54.3	-	61.9	ft.lbf/in^3
Fatigue strength at 10^7 cycles	ⓘ	* 88.9	-	92.5	ksi
Fatigue strength model (stress amplitude)	ⓘ	* 78.6	-	105	ksi

Parameters: Stress Ratio = -1, Number of Cycles = 1e7cycles



Impact & fracture properties

Fracture toughness	ⓘ	74.6	-	91	ksi.in^0.5
Toughness (G)	ⓘ	28	-	41.1	ft.lbf/in^2



Thermal properties

Melting point	2.92e3	-	3.02e3	°F
Maximum service temperature	662	-	788	°F
Minimum service temperature	-459			°F
Thermal conductivity	4.1	-	4.22	BTU/hr.ft.°F
Specific heat capacity	0.134	-	0.136	BTU/lb.°F
Thermal expansion coefficient	4.83	-	5.06	µstrain/°F
Thermal shock resistance	1.79e3	-	1.98e3	°F
Thermal distortion resistance	* 8.2e5	-	8.64e5	BTU/hr.ft
Latent heat of fusion	155	-	159	BTU/lb

Electrical properties

Electrical resistivity	66.1	-	66.9	µohm.in
Electrical conductivity	1.01	-	1.03	%IACS
Galvanic potential	* -0.12	-	-0.04	V

Magnetic properties

Magnetic type	Non-magnetic

Optical, aesthetic and acoustic properties

Transparency	Opaque			
Acoustic velocity	1.97e5	-	2.04e5	in/s
Mechanical loss coefficient (tan delta)	0.001	-	0.005	

Critical materials risk

Contains >5wt% critical elements?	Yes

Processing properties

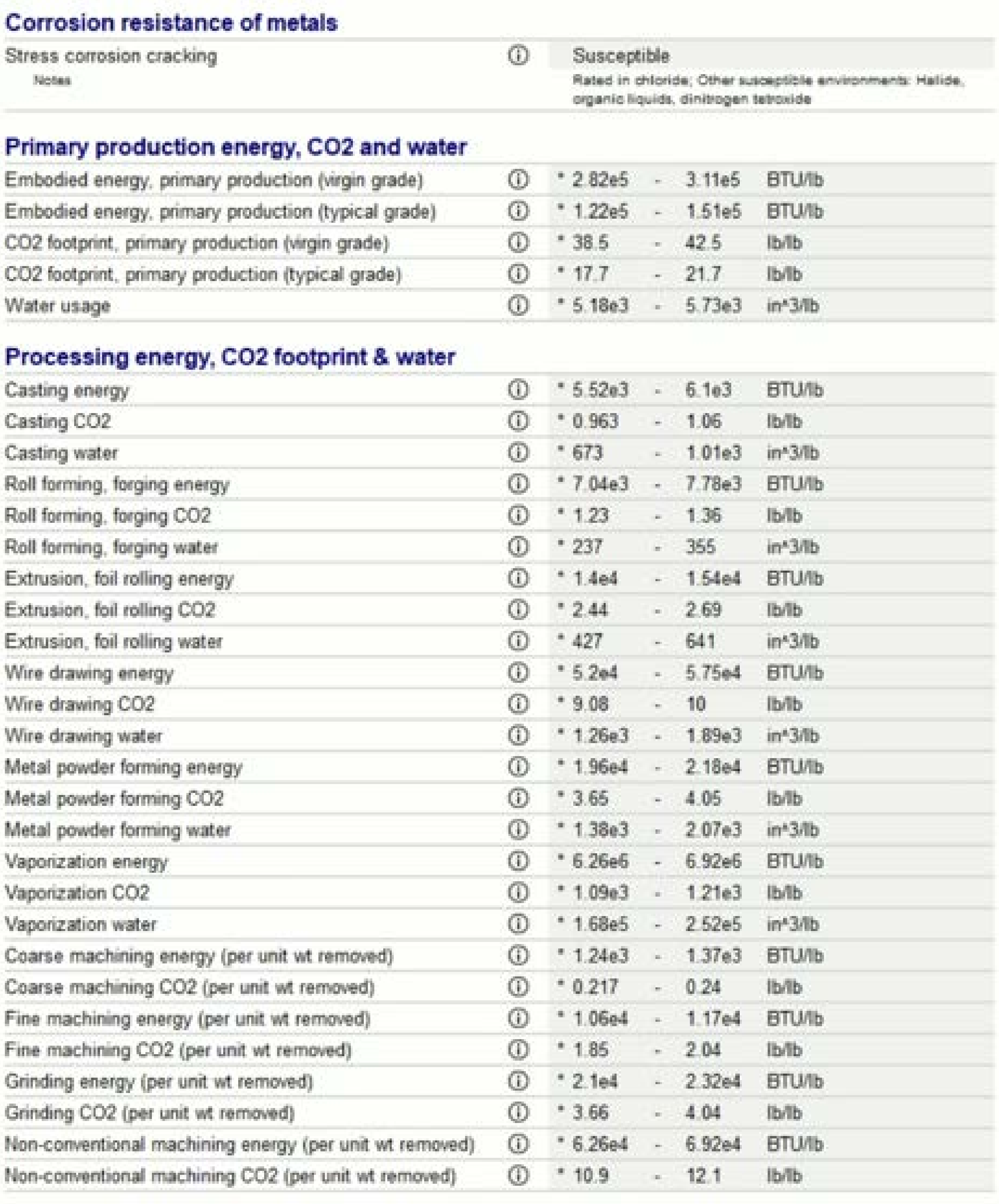
Metal casting	Excellent	
Metal cold forming	Limited use	
Metal hot forming	Acceptable	
Metal press forming	Acceptable	
Metal deep drawing	Limited use	
Machining speed	39	sfm
Weldability	Good	
Notes	Preheating and post weld heat treatments are not required	

Durability

Water (fresh)	Excellent
Water (salt)	Excellent
Weak acids	Excellent
Strong acids	Acceptable
Weak alkalis	Excellent
Strong alkalis	Acceptable
Organic solvents	Excellent
Oxidation at 500C	Excellent
UV radiation (sunlight)	Excellent
Galling resistance (adhesive wear)	Limited use

Notes
High tendency to gall can be overcome by anodizing.

Flammability	Non-flammable



Corrosion resistance of metals

Stress corrosion cracking	ⓘ	Susceptible
Notes		Rated in chloride; Other susceptible environments: Halide, organic liquids, dinitrogen tetroxide

Primary production energy, CO2 and water

Embodied energy, primary production (virgin grade)	ⓘ	* 2.82e5	-	3.11e5	BTU/lb
Embodied energy, primary production (typical grade)	ⓘ	* 1.22e5	-	1.51e5	BTU/lb
CO2 footprint, primary production (virgin grade)	ⓘ	* 38.5	-	42.5	lb/lb
CO2 footprint, primary production (typical grade)	ⓘ	* 17.7	-	21.7	lb/lb
Water usage	ⓘ	* 5.18e3	-	5.73e3	in^3/lb

Processing energy, CO2 footprint & water

Casting energy	ⓘ	* 5.52e3	-	6.1e3	BTU/lb
Casting CO2	ⓘ	* 0.963	-	1.06	lb/lb
Casting water	ⓘ	* 673	-	1.01e3	in^3/lb
Roll forming, forging energy	ⓘ	* 7.04e3	-	7.78e3	BTU/lb
Roll forming, forging CO2	ⓘ	* 1.23	-	1.36	lb/lb
Roll forming, forging water	ⓘ	* 237	-	355	in^3/lb
Extrusion, foil rolling energy	ⓘ	* 1.4e4	-	1.54e4	BTU/lb
Extrusion, foil rolling CO2	ⓘ	* 2.44	-	2.69	lb/lb
Extrusion, foil rolling water	ⓘ	* 427	-	641	in^3/lb
Wire drawing energy	ⓘ	* 5.2e4	-	5.75e4	BTU/lb
Wire drawing CO2	ⓘ	* 9.08	-	10	lb/lb
Wire drawing water	ⓘ	* 1.26e3	-	1.89e3	in^3/lb
Metal powder forming energy	ⓘ	* 1.96e4	-	2.18e4	BTU/lb
Metal powder forming CO2	ⓘ	* 3.65	-	4.05	lb/lb
Metal powder forming water	ⓘ	* 1.38e3	-	2.07e3	in^3/lb
Vaporization energy	ⓘ	* 6.26e6	-	6.92e6	BTU/lb
Vaporization CO2	ⓘ	* 1.09e3	-	1.21e3	lb/lb
Vaporization water	ⓘ	* 1.68e5	-	2.52e5	in^3/lb
Coarse machining energy (per unit wt removed)	ⓘ	* 1.24e3	-	1.37e3	BTU/lb
Coarse machining CO2 (per unit wt removed)	ⓘ	* 0.217	-	0.24	lb/lb
Fine machining energy (per unit wt removed)	ⓘ	* 1.06e4	-	1.17e4	BTU/lb
Fine machining CO2 (per unit wt removed)	ⓘ	* 1.85	-	2.04	lb/lb
Grinding energy (per unit wt removed)	ⓘ	* 2.1e4	-	2.32e4	BTU/lb
Grinding CO2 (per unit wt removed)	ⓘ	* 3.66	-	4.04	lb/lb
Non-conventional machining energy (per unit wt removed)	ⓘ	* 6.26e4	-	6.92e4	BTU/lb
Non-conventional machining CO2 (per unit wt removed)	ⓘ	* 10.9	-	12.1	lb/lb

Recycling and end of life

Recycle	ⓘ	✓			
Embodied energy, recycling	ⓘ	* 3.56e4	-	3.93e4	BTU/lb
CO2 footprint, recycling	ⓘ	* 6.5	-	7.18	lb/lb
Recycle fraction in current supply	ⓘ	58.9	-	65.1	%
Downcycle	ⓘ	✓			
Combust for energy recovery	ⓘ	✗			
Landfill	ⓘ	✓			
Biodegrade	ⓘ	✗			

Notes

Other notes ⓘ

Most widely used of all titanium alloys. Bars and forgings may be heat treated to a range of strength levels and is used in highly stressed structures. Weldable alloy.
Brinell Hardness, 3000kg load, 10mm ball

Appendix K: Correspondence with Wade Vagle

This appendix highlights all the email correspondences that happened between me and the CEO of DIYWalkers, Wade Vagle.

Greetings,

I hope this email finds you well. My name is Youssef Abdelhalim, I am a rising sophomore at Northwestern University, studying Mechanical Engineering. I recently started working on a personal project, just to keep me busy over the summer, that involves the use of 4-bar linkages as walkers. During my research, I came across your website, which was TREMENDOUSLY helpful, it is the first website I came across that was focused on education and explaining how these walkers work and why they work.

I am working on developing a new type of walker that performs well specifically on sandy inclined terrains, think sand dunes. I was wondering if you have any advice for me since you're an expert in the field. We can talk over email or we can get on a zoom call if you have some free time and would like to discuss this more.

Thank you so much for your time.

Sincerely,
Youssef Abdelhalim.

Greetings Mr. Wade,

I wanted to follow up on my initial email. I would like to iterate how helpful and significant your feedback would be. You and Ben seem to be the experts in the field and looking up anything, anywhere always leads to your website.

A conversation with you would be greatly appreciated.

Sincerely,
Youssef Abdelhalim.

Hi Youssef,
By "sand dunes" do you mean the loose sand in dunes like the sahara? If so, I would advise using feet with broad surface area so that the feet don't sink, perhaps by putting pads on the feet like this Strider "snowbot" test:
https://www.youtube.com/watch?v=TRQBp0A9RSM

I would also try to shield the joints from sand to avoid friction/wear.

If the sand is uneven/bumpy, then you may want a high foot-path. If walking on flat sand like on Jansen's beaches, then stepping high isn't critical.

Hope this helps,
Wade

Thank you so much for your response Mr. Vagle.

So I was thinking more about coarse and smooth dunes, but ones with steep inclines. Most of the walkers that I have seen are meant for level-ish surfaces. I am focusing my project on soarse surfaces that are moreso inclined.

I did not consider the surface area of the feet so that is definitely a great call. I also did not consider covering the joints to prevent friction. Do you know of any methods of covering the joints without inhibiting their motion?

Again, thank you so much.

Sincerely,
Youssef Abdelhalim.

Youssef,
If you want to climb steep grades, then be sure to put most weight toward the bottom center of the robot for a lower center of gravity. Also, you may want to put space between the front and rear legs. Finally, you'll probably want at least 8 legs for more stability, such that 4 corners of the robot are in contact with the ground during most portions of the axle's rotation.

Good luck!
Wade

That is all really helpful. Thank you so much Mr. Vagle. I will implement all your suggestions in my next iteration of my design and update you on how things look.

Again, thank you so much for all the great work that you have done on the DIYWalkers website. This website was the perfect guide when I first started working on my project and it helped me understand a good amount of the concepts related to mechanical walkers and their dynamics.