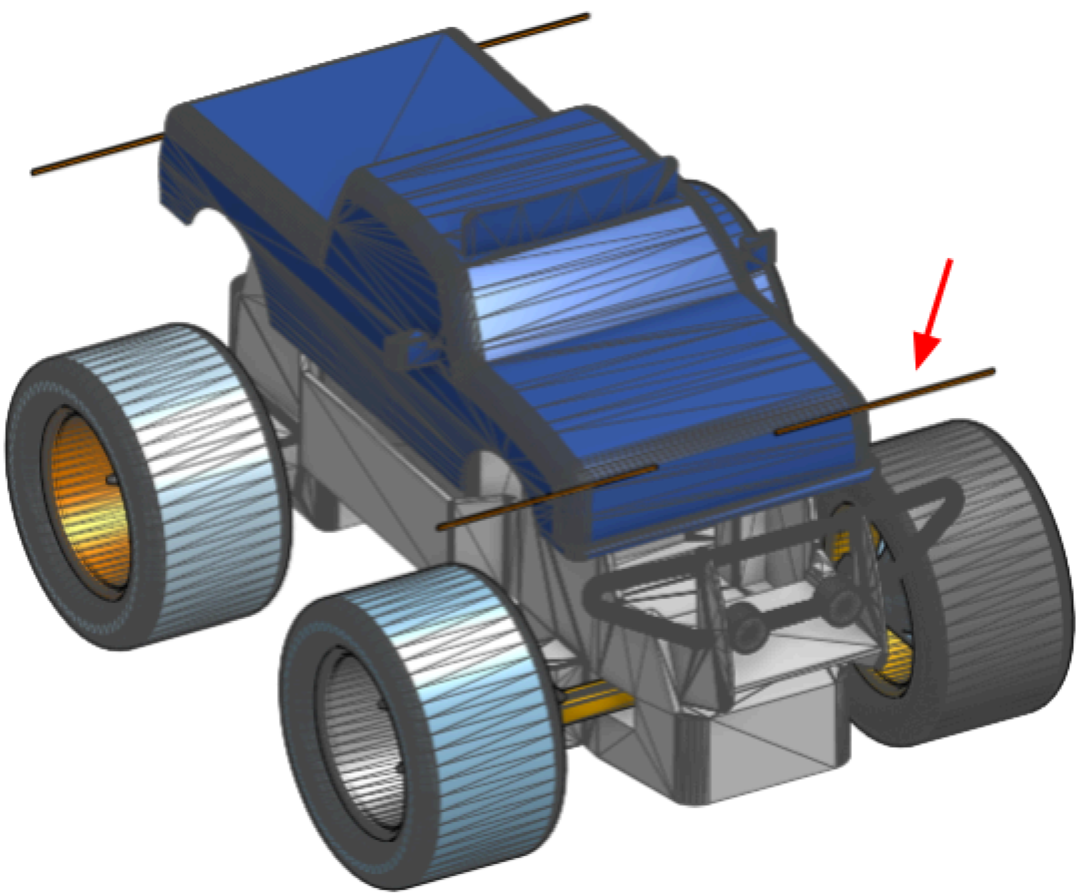
Addition of limiters

As I am writing this(July 14, 2024), there has been an increased interest in humanoid, bipedal robots, partly spurred on by advancements in AI. For example, Boston Dynamics created a new version of its bipedal robot Atlas, now powered by electric motors as opposed to hydraulics.[1] OpenAI invested in the robotics company Figure, which has demonstrated its humanoid robot Figure 01.[2] How do these robots differ from cars? For one, these robots use legs to walk, whereas cars use wheels to roll. Without going too much into how walking is different from rolling, one thing we can suppose is that for something to walk, the appendages which touch the ground(legs, wheels, etc.) must have a limited degree of motion in its main direction. For example, a car's wheels mainly turn in the direction parallel to the car doors, and are able to turn a full 360 degrees and beyond in this direction.

In order to add these restrictions, you need to add limiters to your RC car. These limiters will prevent the legs(which you will eventually add) from turning a full 360 degrees. Ideally, they will only let it turn around 180 degrees. For the limiters, you can either use the materials provided(**with the number 5**) or use your own materials, such as a pencil or pen. You can attach them to the RC car with a sticking material such as tape, glue, etc. An example is shown below. The limiters are the thin orange things sticking out of the car sides.

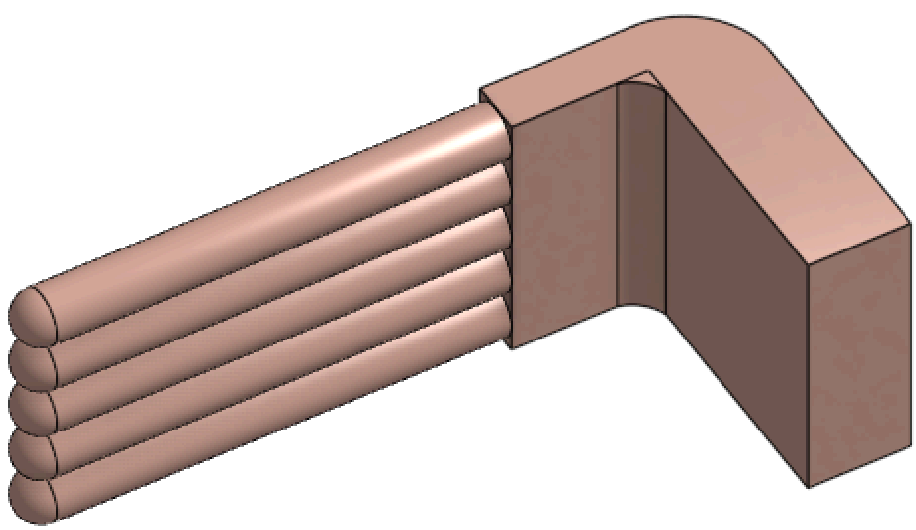


Measurement of the COM

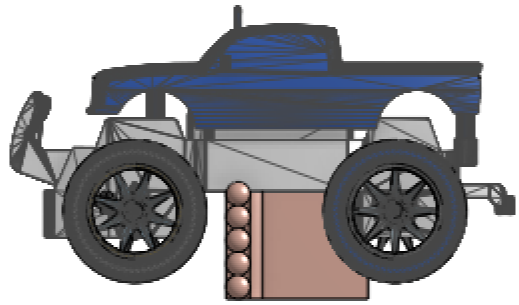
The COM — which stands for the center of mass — is a way of describing an object as a single point based on its mass. Imagine that for any object you could see, for any point or part of the object, how much mass that part had. A ball or sphere made up of pure iron will have its center of mass at its center, because it looks the same no matter which way you look at it. If you cut that ball in half(called a hemisphere), the center of mass will be on the line of symmetry of the hemisphere, because if you rotate it around the line of symmetry, it looks the same no matter what. However, if you rotate it around the line perpendicular to the line of symmetry, it will look different as you rotate it.

One property of the COM is that you can balance the object on it. For example, if you place two boxes of equal mass on a seesaw equal distance from the pivot/fulcrum, then the seesaw will balance. If you think of the pair of boxes as being a single object, then the COM is on the pivot/fulcrum.

To measure the COM, stand a hardcover book such that its spine is facing up. If you can't get the book to stand or don't have a hardcover, put your four(and your thumb if you choose, although it works better without it) parallel to each other, as shown below.



Then balance the RC car on top such that your fingers are perpendicular to the length of the RC car, as shown below.

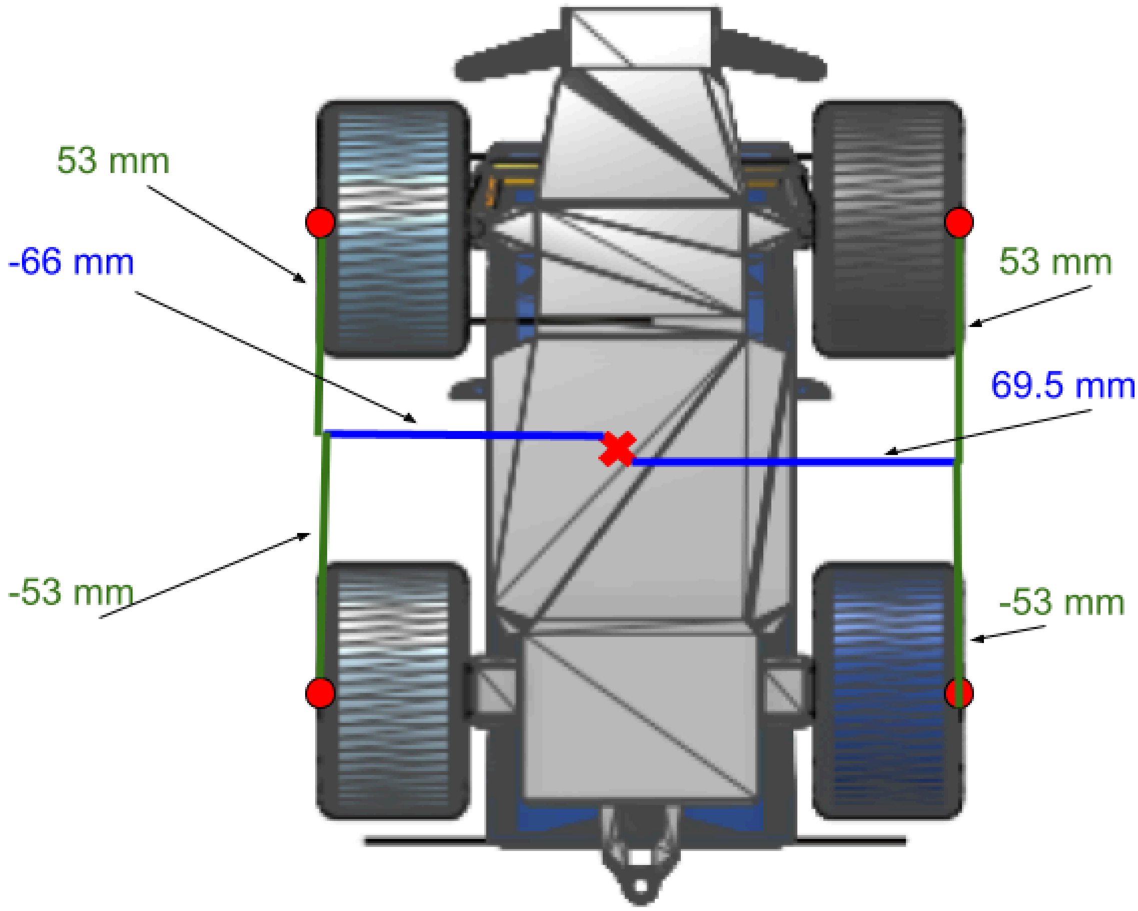


It doesn’t have to exactly balance, but it should be as close to balancing as possible. Mark the point with a pencil on the RC car.

Do the same thing, but with your fingers parallel to the length of the car.

Draw a line from the first pencil mark to the other end of the RC car, perpendicular to the length of the car(preferably on the bottom of the RC car, if possible, although a piece of cardboard stuck on top of the car works too). Then draw a second line from the second pencil mark to the other end of the RC car, parallel to the length of the RC car. It is better to use a ruler. Put an “X” on the intersection of these two lines. This “X” is around where the COM is.

Measure the distances from the “X” to the centers of the outsides of the wheels, both parallel and perpendicular to the length of the RC car. If the wheel’s center is on the left side or below the X, when the RC car is viewed from the top with the front facing away from you, make that distance negative. An example is shown below.



Download of IDLE

Note: If you have another Python IDE, you may use that. However, you must be able to import PyTorch and Numpy. Don't worry if you don't have access to GPUs or TPUs in your IDE; the code will check for access to these, and if not available, it will run on your computer's CPU, although the training will be slower.

Go to this link and download IDLE and follow the steps: Download Python | Python.org. If possible, download version 3.12. This will allow you to run code that will tell you which orientations the legs should start at, and how to move the RC car to make it walk. Don't worry if you can't code; the code will be provided for you to copy and paste.

Setting Up the Virtual Environment on Windows

A virtual environment is a way of letting you run the code without having to deal with a bunch of messy details.

If you are on Windows, open up Command Prompt. You should see something like this:

```
C:\Users\rohan>
```

If you see something like the picture below, type “**cd ..** ” and **ENTER** until you reach the image above.

```
C:\Users\rohan\Downloads>
```

```
C:\Users\rohan\Downloads>cd..
```

```
C:\Users\rohan>
```

Type “**python -m venv myenv**” and **ENTER**, as shown below.

```
C:\Users\rohan>python -m venv myenv
```

Then, on the next line, type “**myenv\Scripts\activate**” and **ENTER**.

```
C:\Users\rohan>python -m venv myenv

C:\Users\rohan>myenv\Scripts\activate

(myenv) C:\Users\rohan>
```

On this last new line, type “**pip install torch**” and **ENTER**. If you have torch installed already(unlikely if you have never used IDLE before), you will see the image below. Otherwise, you should see the installation of torch.

```
(myenv) C:\Users\rohan>pip install torch
Requirement already satisfied: torch in c:\users\rohan\myenv\lib\site-packages (2.3.1)
Requirement already satisfied: filelock in c:\users\rohan\myenv\lib\site-packages (from torch) (3.15.4)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\rohan\myenv\lib\site-packages (from torch) (4.12.2)
Requirement already satisfied: sympy in c:\users\rohan\myenv\lib\site-packages (from torch) (1.12.1)
Requirement already satisfied: networkx in c:\users\rohan\myenv\lib\site-packages (from torch) (3.3)
Requirement already satisfied: jinja2 in c:\users\rohan\myenv\lib\site-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in c:\users\rohan\myenv\lib\site-packages (from torch) (2024.6.1)
Requirement already satisfied: mkl<=2021.4.0,>=2021.1.1 in c:\users\rohan\myenv\lib\site-packages (from torch) (2021.4.0
)
Requirement already satisfied: intel-openmp==2021.* in c:\users\rohan\myenv\lib\site-packages (from mkl<=2021.4.0,>=2021
.1.1->torch) (2021.4.0)
Requirement already satisfied: tbb==2021.* in c:\users\rohan\myenv\lib\site-packages (from mkl<=2021.4.0,>=2021.1.1->tor
ch) (2021.13.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\rohan\myenv\lib\site-packages (from jinja2->torch) (2.1.5)
Requirement already satisfied: mpmath<1.4.0,>=1.1.0 in c:\users\rohan\myenv\lib\site-packages (from sympy->torch) (1.3.0
)

[notice] A new release of pip is available: 24.0 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(myenv) C:\Users\rohan>
```

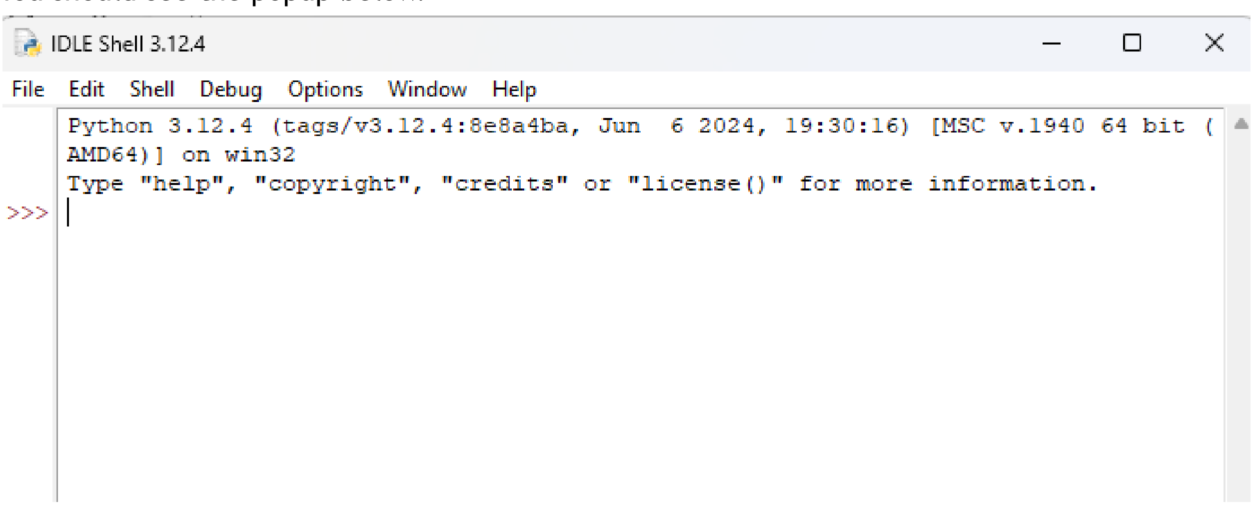
On this last new line, type "**`pip install numpy`**" and **ENTER**.

```
(myenv) C:\Users\rohan>pip install numpy
Requirement already satisfied: numpy in c:\users\rohan\myenv\lib\site-packages (2.0.0)

[notice] A new release of pip is available: 24.0 -> 24.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(myenv) C:\Users\rohan>
```

Then type "**`python -m idlelib.idle`**" and **ENTER**.

```
(myenv) C:\Users\rohan>python -m idlelib.idle
```

You should see the popup below.



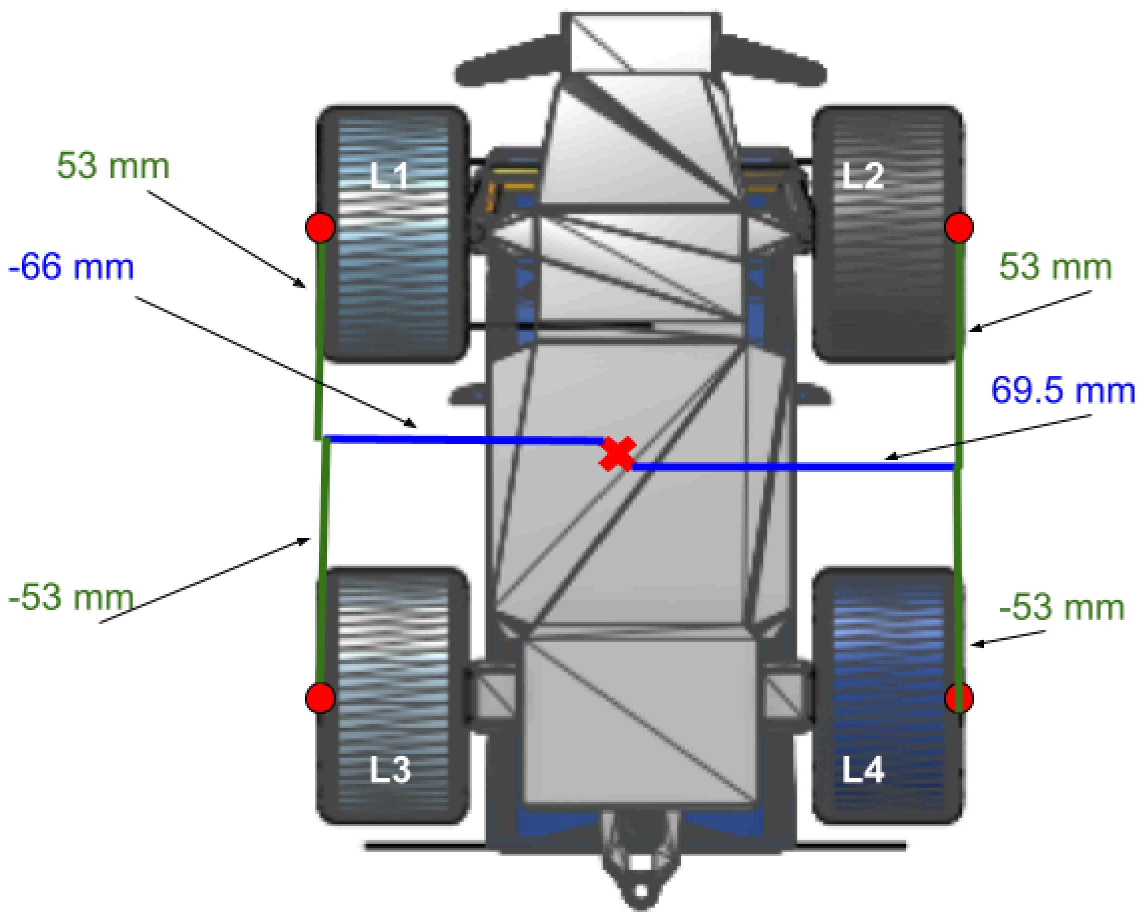
Click File → New at the top to create a new file.

Code

Go to the link below.
https://github.com/Thelta48/Walking-RC-Cars

If your front and back wheels can move independently from each other, open the file `TwoWheels.py`. If they move together(meaning that if you rotate the front wheels, the back wheels rotate), open the file `FourWheels.py`. Copy and paste the code into the new empty file in IDLE.

In the **Measurement of COM** section, you measured distances from the COM to the axles of the wheels. Let's designate the wheels as **L1**, **L2**, **L3**, and **L4**.



On the top of the IDLE file, click Options → Show Line Numbers.
You should see numbers on the left side of the code.

```
1 import numpy as np
2
3 def DistanceSweptOutByLegsUntilAngle(theta):#If this was a circular wheel, this would be the length of the arc
4     Ans = 0.0
5     gamma1 = -1.51491088379#angle between edge of flat tip and midline
6     gamma2 = -1.02925556679#angle between midline and where circle meets rectangle
```

Go to line 315 in the `TwoWheels.py` file, 313 in the `FourWheels.py` file. You should see the lines of code below.

```
x1 = 7.5
y1 = 8
x2 = 7.5
y2 = -8
x3 = -6
y3 = -8
x4 = -6
y4 = 8.5
```

Replace the numbers with the distances you measured. For example, in the example given, it would be.

```
x1 = -66
y1 = 53
x2 = 69.5
y2 = 53
x3 = -66
y3 = -53
x4 = 69.5
y4 = -53
```

Click **Ctrl + S** or **Command + S**, depending if you're on Windows or Mac, to save the file. Pick any file name you choose. Then go to the top of the file and click Run → Run Module. You should see the main IDLE page and some text and numbers appearing. Wait until you see something similar to the code below.

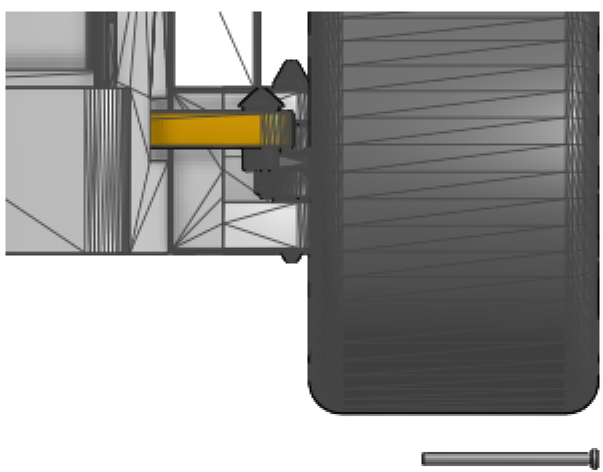
```
[-45.07826607 -10.40920169 -13.17122522  -6.50829386]
[np.float64(1.0), np.float64(1.0), np.float64(-1.0), np.float64(1.0), np.float64(1.0), np.float64(-1.0), np.float64(-1.0), np.float64(1.0), np.float64(-1.0)]
```


```
[-45.07826607 -10.40920169 -13.17122522  -6.50829386]
[np.float64(1.0), np.float64(1.0), np.float64(-1.0), np.float64(1.0),
np.float64(1.0), np.float64(-1.0), np.float64(-1.0), np.float64(1.0),
np.float64(-1.0)]
```

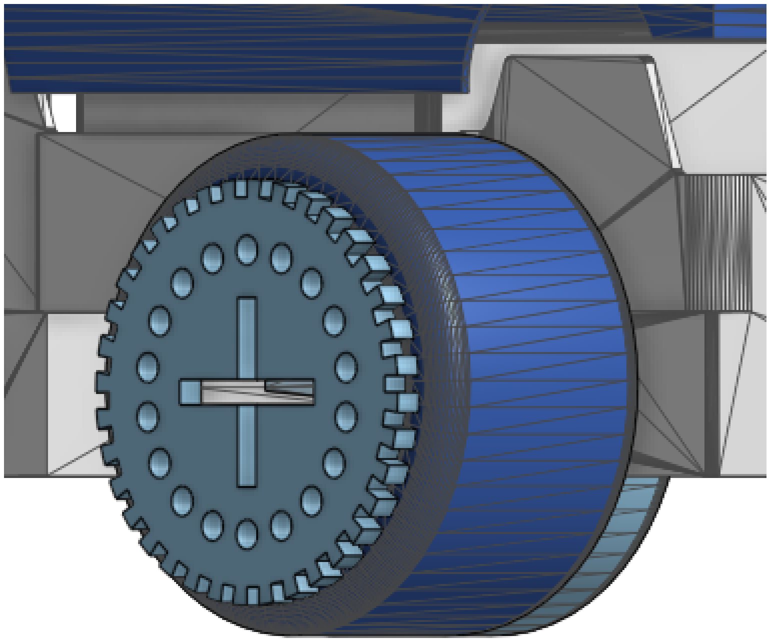

Copy down the first four numbers on a sheet of paper. Then copy the 1's and -1's onto the paper in the order that you see them, from reading left to right.

Leg Placement

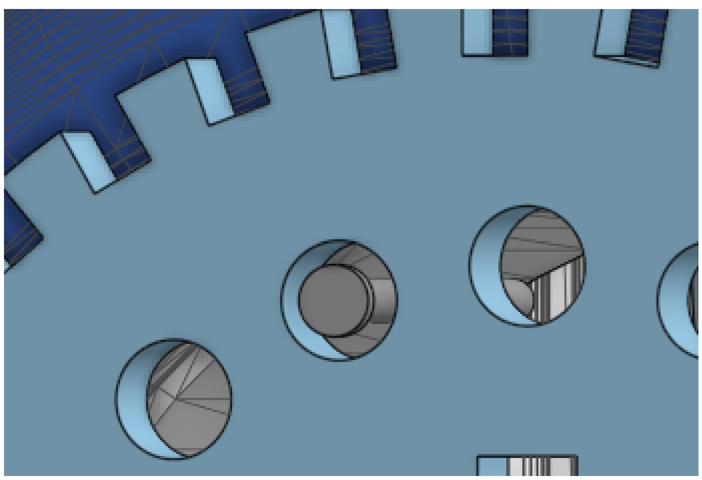
Get 8 screws which are long enough to reach from the outer face of the wheel to the spokes inside.

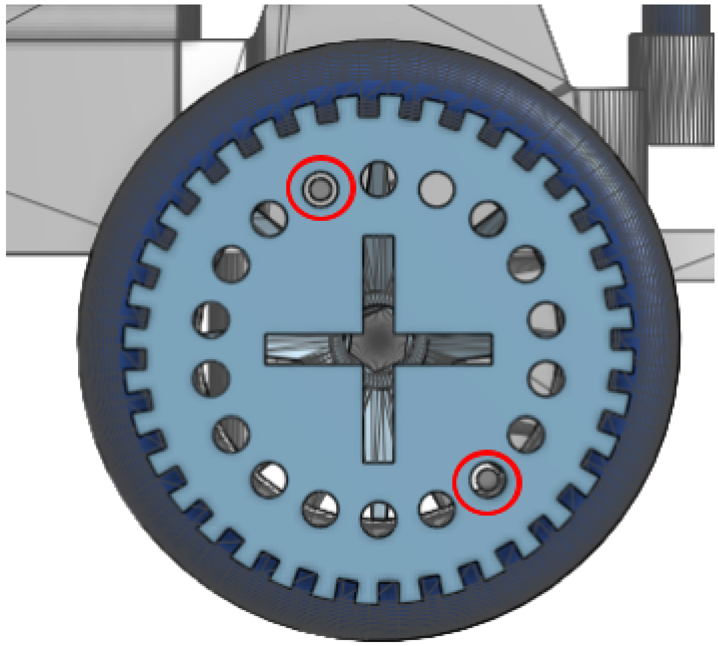


Overlay the circular disk part below on top of the outer face of the wheel.

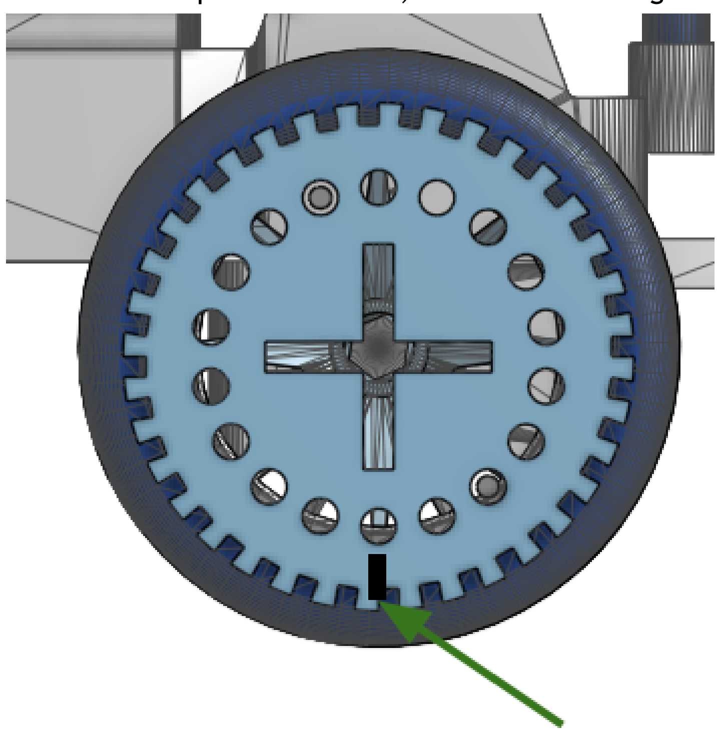


Insert two screws through two of the holes in the disk such that they bump against the spokes, preventing the disk from rotating.

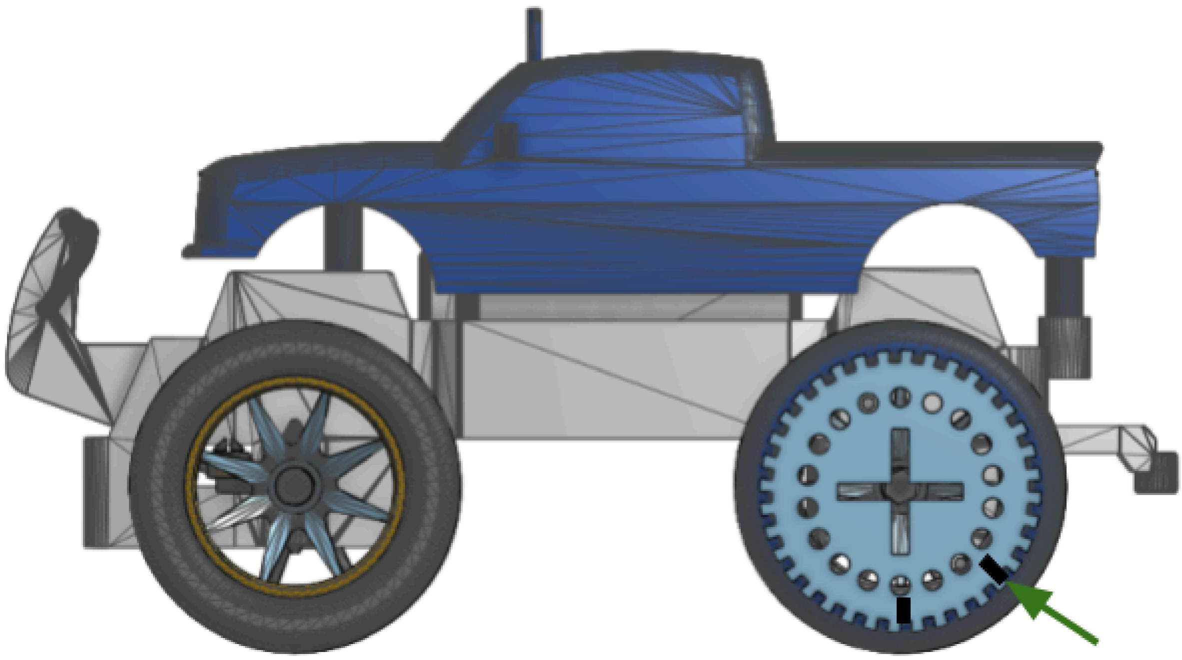




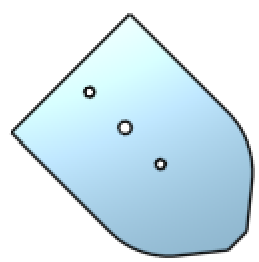
Make a pencil mark on the bottom part of the disk, where the 270 degree angle would be.



Take the disk away from the wheel. Look at the first four numbers you copied down on that sheet of paper. Make each number positive. If the wheel is L1, refer to the first number. If the wheel is L2, refer to the second number. And so on. This number is the angle at which you will place your leg. Use a protractor to make a mark below the 0 degree angle, on the side closest to one end of the RC car. For example, if your first angle is 45, and the wheel is L4 closest to the back of the RC car, the mark will be at



Then use screws to fasten the leg such that the vertical midline aligns with that mark, and that the curved part of the leg is below the 0 degree angle. The leg will be bigger than the disk and will hide it.



Put the disk back onto the wheel, aligning the screws to bump against the spokes. Use tape or rubber bands to fasten the disks to the wheels. If using rubber bands, it may be useful to use the gear teeth.

Do this for all four wheel-leg pairs.

Now consider the sequence of 1's and -1's you copied onto the sheet of paper. Pair your remote controller with the RC car. Once it is paired, for every 1, move the car forwards until the limiters stop it. For every -1, move it backwards until the limiters stop it. Do this in order for the entire sequence, and repeat the sequence many times. You should see the RC car move forwards, perhaps slowly, despite the limiters. It should move similar to the video below.
https://youtu.be/NqyTWU_zi5I

[1]Boston Dynamics unveils new Atlas robot for commercial use (nbcnews.com)
[2]See the humanoid work robot OpenAI is bringing to life with artificial intelligence - CBS News