Theo Jansen Based Walking Mechanism

Team MSD Group+ Mechaneers

June 5, 2025

1 The Jansen Linkage

The Jansen mechanism has 12 rods connected in a specific configuration. We care mostly about the 9 critical angles: θ_i, θ_1 through θ_8. When we rotate the crank (angle θ_i), the leg moves in a walking pattern.

2 Why We Chose the Jansen Linkage

When exploring mechanical leg linkages, we considered several options, including the Klann linkage, the Strandbeest (Jansen mechanism), and other walking systems. After analyzing them based on motion quality, mechanical complexity, and learning potential, we decided to proceed with the Jansen linkage for the following reasons:

2.1 1. Smooth, Natural Gait

The Jansen linkage is designed to replicate the fluid and efficient walking motion seen in nature. Its foot trajectory minimizes vertical oscillations, which means:

- The "foot" stays relatively flat on the ground during stance phase.
- The lift-off and swing phases are smooth, without jerky transitions.
- There is minimal slippage or bouncing, ideal for walking on flat surfaces.

In contrast, mechanisms like the Klann linkage tend to produce more angular or robotic motion, which can be less stable or natural-looking.

2.2 Complexity Suited for Learning

While more complex than simpler linkages, the Jansen mechanism provides a rich opportunity to learn about:

- Inverse kinematics and solving nonlinear systems.
- Mechanism synthesis and constraint satisfaction.

- 3D CAD design and joint simulation in software like Fusion 360.

By taking on a more intricate system, we challenged ourselves to understand deeper engineering principles while still being able to simulate and build the result.

2.3 Modular and Scalable

The leg design is modular and can be replicated easily. This made it feasible to:

- Simulate a full four-legged walker by phase-shifting crank angles.
- Add motors or actuators in the future to drive the system.
- Expand into larger multi-legged walking robots using the same leg unit.

While the Klann and similar linkages are excellent for simplified walking robots, the Jansen linkage offers a unique balance of smooth motion, aesthetic appeal, and mathematical richness. For our goal — a realistic, mechanically interesting, and educational walking mechanism — the Jansen system was the optimal choice.

3 Forward Kinematics of the Jansen Linkage

Forward kinematics allows us to calculate the positions of all joints in the mechanism when the input crank angle and dependent joint angles are known. In the Jansen mechanism, the entire leg configuration can be determined once we know:

- The crank angle θ_i
- The resulting joint angles θ_1 through θ_8
- The fixed frame dimensions (offsets a and b)
- The link lengths $l_i, l_1, \ldots, l_9$

Using trigonometry, we can recursively compute the (x, y) coordinates of all key joints.

3.1 Joint Position Calculations

Let's define:

- Joint 0: fixed crank origin, located at $(0, 0)$
- Joint 1: second fixed point, located at $(-b, -a)$

Given this reference frame, we compute the positions of the other joints step-by-step:

Joint A (end of crank):

$$x_A = l_i \cos(\theta_i), \quad y_A = l_i \sin(\theta_i)$$

Joint B:

$$x_B = x_A + l_1 \cos(\theta_1), \quad y_B = y_A + l_1 \sin(\theta_1)$$

Joint C:

$$x_C = x_B + l_3 \cos(\theta_3), \quad y_C = y_B + l_3 \sin(\theta_3)$$

Joint D:

$$x_D = -b + l_7 \cos(\theta_7), \quad y_D = -a + l_7 \sin(\theta_7)$$

Joint E:

$$x_E = x_D + l_8 \cos(\theta_8), \quad y_E = y_D + l_8 \sin(\theta_8)$$

Joint F (foot): Joint F is connected to Joint D by a link of length l_9 at an offset angle c (computed using the cosine rule):

$$x_F = x_D + l_9 \cos(\theta_8 + c), \quad y_F = y_D + l_9 \sin(\theta_8 + c)$$

Where the angle c between links l_8 and l_9 is computed using the law of cosines:

$$c = \cos^{-1}\left(\frac{l_8^2 + l_9^2 - l_{10}^2}{2 l_8 l_9}\right)$$

with l_{10} being the length of the link connecting joints E and F.

3.2 Full Position Vector

Combining all joint positions, we define a forward kinematics function:

$$\mathrm{FK}(\theta_i, \theta_1, \ldots, \theta_8) = \begin{cases} (x_A, y_A) \\ (x_B, y_B) \\ (x_C, y_C) \\ (x_D, y_D) \\ (x_E, y_E) \\ (x_F, y_F) \end{cases}$$

3.3 Implementation Notes

This model is used to:

- Animate the leg in simulation once the angles are known.
- Trace the foot (joint F) trajectory over time.
- Compare the effect of small changes in geometry.

4 System of Equations

We create equations based on the lengths of the rods and the angles between them. Using basic trigonometry, the vertical and horizontal position of each joint must match, regardless of which path we use to get there.

Equations:

$$l_i \sin(\theta_i) + l_1 \sin(\theta_1) = a + l_2 \sin(\theta_2) \tag{1}$$
$$l_i \cos(\theta_i) + l_1 \cos(\theta_1) = b + l_2 \cos(\theta_2) \tag{2}$$
$$l_i \sin(\theta_i) + l_1 \sin(\theta_1) + l_3 \sin(\theta_3) = a + l_4 \sin(\theta_4) \tag{3}$$
$$l_i \cos(\theta_i) + l_1 \cos(\theta_1) + l_3 \cos(\theta_3) = b + l_4 \cos(\theta_4) \tag{4}$$
$$l_i \sin(\theta_i) + l_6 \sin(\theta_6) = a + l_7 \sin(\theta_7) \tag{5}$$
$$l_i \cos(\theta_i) + l_6 \cos(\theta_6) = b + l_7 \cos(\theta_7) \tag{6}$$
$$l_i \sin(\theta_i) + l_1 \sin(\theta_1) + l_3 \sin(\theta_3) + l_5 \sin(\theta_5) = a + l_7 \sin(\theta_7) + l_8 \sin(\theta_8) \tag{7}$$
$$l_i \cos(\theta_i) + l_1 \cos(\theta_1) + l_3 \cos(\theta_3) + l_5 \cos(\theta_5) = b + l_7 \cos(\theta_7) + l_8 \cos(\theta_8) \tag{8}$$

5 Inverse Kinematics: How We Make the Jansen Leg Walk

Inverse kinematics (IK) is the method we use to figure out what each joint in a mechanical leg should do so that the leg moves the way we want. Think of it like this: if someone moves your foot, your knee and hip must adjust to stay connected. That's what inverse kinematics figures out.

5.1 Forward vs Inverse Kinematics

First, let's understand the difference between forward and inverse kinematics:

- **Forward Kinematics** answers: "If I rotate these joints, where does the foot go?"
- **Inverse Kinematics** asks: "If I want the foot to go here, what angles should the joints be?"

For robots or mechanical linkages like Jansen's leg, IK is much harder, especially when there are many joints involved — in our case, eight.

5.2 The Problem in the Jansen Linkage

In the Jansen linkage, we rotate a single crank at joint θ_i to simulate one leg walking. But once we rotate that crank, the whole system of connected rods and joints must adjust so that every piece remains connected (no rods stretch or compress). This means:

- Each joint's position depends on several angles.

- We must solve a system of equations to find those angles.

So, given a value for θ_i (the crank angle), we want to find values for the other angles: θ_1 to θ_8.

5.3 How the Equations Are Built

Each joint in the system must meet at a certain point — this creates geometric constraints. If we use trigonometry to write down the x and y positions of each joint, we can equate the different "paths" to the same joint. For example:

$$l_i \sin(\theta_i) + l_1 \sin(\theta_1) = a + l_2 \sin(\theta_2)$$

$$l_i \cos(\theta_i) + l_1 \cos(\theta_1) = b + l_2 \cos(\theta_2)$$

This pair of equations says: "If you go from joint 0 to A to B, and from joint 1 to B, the vertical and horizontal distances must match."

We do this for multiple joints until we have 8 equations like these. We then define a vector of unknown angles:

$$\mathbf{x} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_8 \end{bmatrix}$$

and a function vector:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_8(\mathbf{x}) \end{bmatrix}$$

where each $f_i(\mathbf{x})$ is one of the geometric constraint equations. Our goal is to find $\mathbf{x}$ such that:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

This is a classic inverse kinematics setup — solving nonlinear equations that describe a mechanical linkage.

5.4 Using the Newton-Raphson Method

To solve $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, we use the multidimensional Newton-Raphson method. It works like this:

1. Start with a guess $\mathbf{x_0}$ for the angles.

2. Use the Jacobian matrix $J(\mathbf{x})$, which contains partial derivatives of each f_i with respect to each angle.

3. Update your guess using:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - J^{-1}(\mathbf{x_n}) \cdot \mathbf{f}(\mathbf{x_n})$$

4. Repeat until $\mathbf{f}(\mathbf{x})$ is very close to 0.

This process gives us the eight angles that satisfy all constraints for a given crank angle θ_i.

5.5 Jacobian Matrix Details

The Jacobian J is a matrix where each entry is a derivative:

$$J_{ij} = \frac{\partial f_i}{\partial \theta_j}$$

So for 8 equations and 8 unknowns, J is an 8x8 matrix. We compute this at every iteration based on the current guess of angles, which allows the Newton-Raphson method to adapt.

5.6 Stepping Through Motion

Once we solve for θ_1 through θ_8 for one crank angle, we slightly increase θ_i (e.g., by 0.1 rad) and repeat the Newton-Raphson method using the previous solution as the next initial guess. Since the crank moves smoothly, the rest of the leg moves smoothly, too.

5.7 Visualizing the Result

For each step, we calculate the position of all joints using:

$$x = x_0 + l\cos(\theta), \quad y = y_0 + l\sin(\theta)$$

and draw the leg. By repeating this for many steps, we trace a smooth path of the foot (point F), which shows a walking motion.

5.8 Finding Joint Positions

After solving for the angles, we use trigonometry to find where each joint is:

$$
\begin{aligned}
x_A &= l_i \cos(\theta_i), & y_A &= l_i \sin(\theta_i) \\
x_B &= x_A + l_1 \cos(\theta_1), & y_B &= y_A + l_1 \sin(\theta_1) \\
x_C &= x_B + l_3 \cos(\theta_3), & y_C &= y_B + l_3 \sin(\theta_3) \\
x_D &= -b + l_7 \cos(\theta_7), & y_D &= -a + l_7 \sin(\theta_7) \\
x_E &= x_D + l_8 \cos(\theta_8), & y_E &= y_D + l_8 \sin(\theta_8) \\
x_F &= x_D + l_9 \cos(\theta_8 + c), & y_F &= y_D + l_9 \sin(\theta_8 + c)
\end{aligned}
$$

Where c is the fixed angle between two rods and is calculated with the cosine rule.

$$
c = \arccos\left(\frac{l_8^2 + l_9^2 - l_{10}^2}{2 l_8 l_9}\right)
$$

5.9 Results

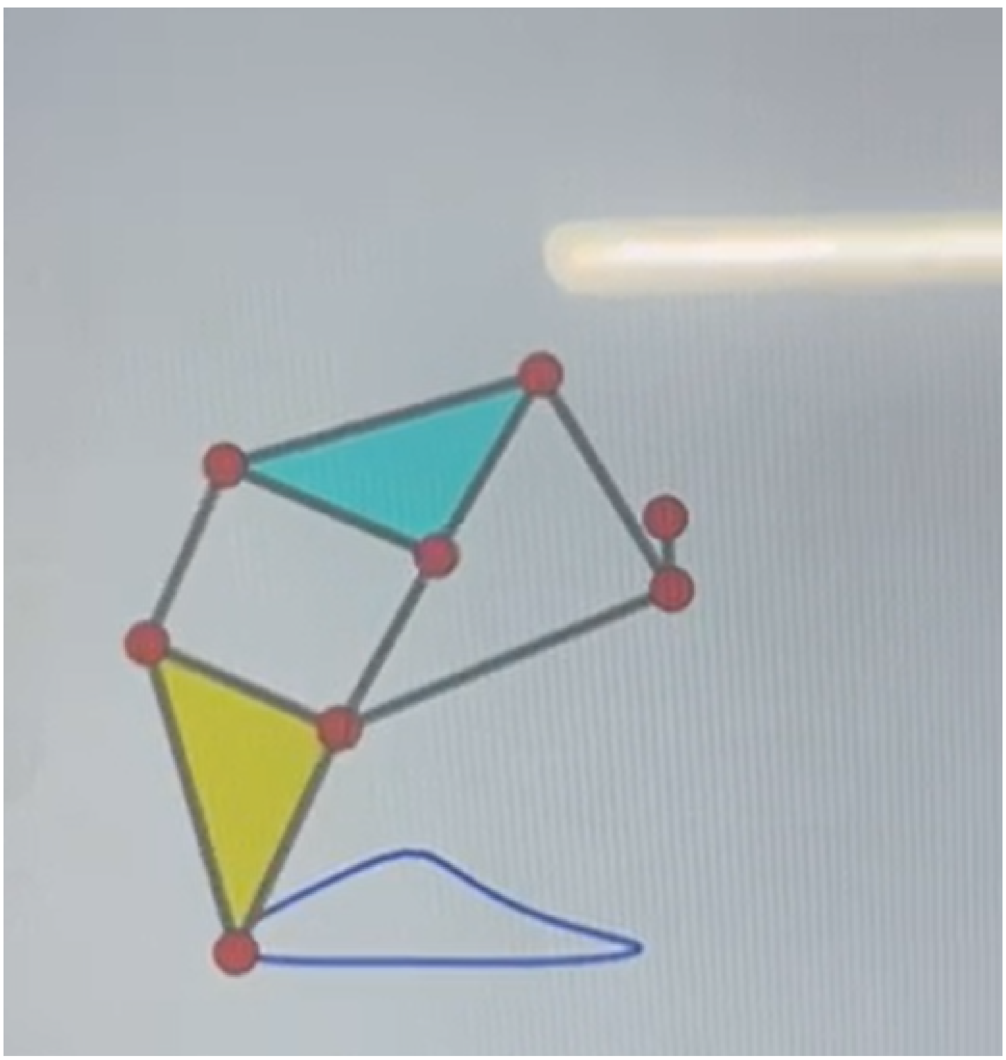

Figure 1: Single leg movement and trajectory

Figure 2: 4 leg movement and trajectory

6 CAD Modeling in Fusion 360

To complement our mathematical simulation of the Jansen linkage, we built a 3D model using Autodesk Fusion 360. This allowed us to visualize how each component moves in real life and prepare for physical fabrication.

6.1 Step 1: Setting Up the Project

1. Open Fusion 360 and create a new project.
2. Create a new component named `Jansen Leg`.
3. Set your units to **millimeters**.

6.2 Step 2: Sketching the Linkages

Sketch Each Link as a Separate Component:
For each link (e.g., l_1, l_2, ... l_9), follow these steps:

1. Create a new component: `Create > New Component`.
2. Start a new sketch on the top plane: `Create Sketch`.

3. Draw a simple rectangle or line segment to represent the link.
4. Use the `Dimension Tool (D)` to set its exact length as per Table 1.
5. Add construction circles at both ends to represent pivot holes (diameter: 4–6 mm, depending on your shaft size).
6. Use `Finish Sketch`, then `Extrude` the profile (typical thickness: 5–8 mm).
7. Name the component appropriately (e.g., `Link_l1`).

Table 1: Linkage Lengths Used in the Fusion 360 Model

Link	**Length (mm)**
a	38.0
b	41.5
c	39.3
d	40.1
e	55.8
f	39.4
g	36.7
h	65.7
i	49.0
j	50.0
k	61.9
l	7.8
m	15.0

6.3 Step 3: Assembling the Mechanism

1. Create a new component: `Main Assembly`.
2. Insert all previously created link components into the assembly.
3. Begin placing joints using `As-Built Joint` or `Joint`.
4. Use `Revolute Joints` between holes to simulate pin connections.
5. Define fixed components:
   - Fix joint 0 (crank base).
   - Fix joint 1 for better stability.
6. Ensure each linkage is connected to the correct pair of joints.

6.4 Step 4: Driving the Crank Motion

1. Right-click on the crank link (l_i).
2. Select `Drive Joint`.
3. Choose the `revolute joint` that rotates the crank.
4. Set the angle step and animation speed (e.g., 5° per step).

6.5 Step 5: Adding Constraints and Refinement

- Add **rigid groups** for link clusters that move together.
- Use **contact sets** to ensure links don't interfere.
- Test the full range of motion to make sure no parts intersect.

6.6 Step 6: 3D Printing Preparation

If you want to 3D print the leg:

- Add fillets to link ends for strength.
- Add tolerances (0.3–0.5 mm) around pin holes.
- Export each component as an STEP.

6.7 Observations from CAD Model

After assembling and simulating the Jansen linkage, we observed:

- The leg follows a smooth walking path when the crank rotates uniformly.
- The trajectory of the foot (joint F) matches our MATLAB simulation.
- Some linkages are under more stress during retraction — useful for real-world material selection.

7 Results

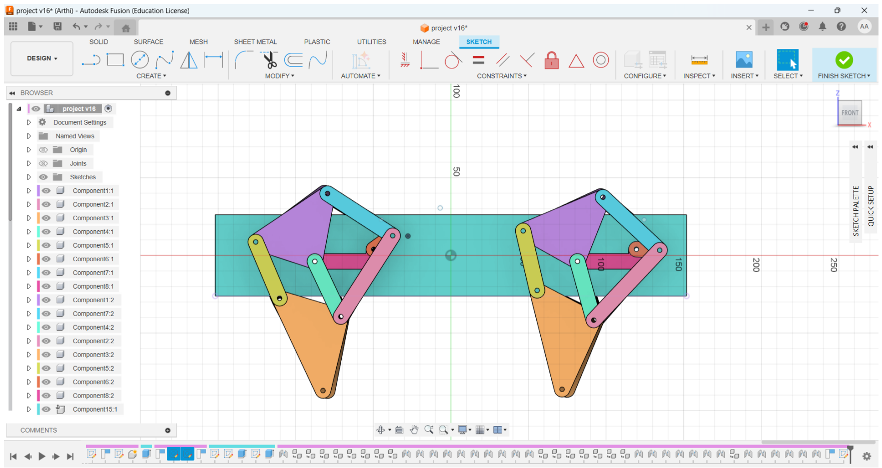

Figure 3: Side View

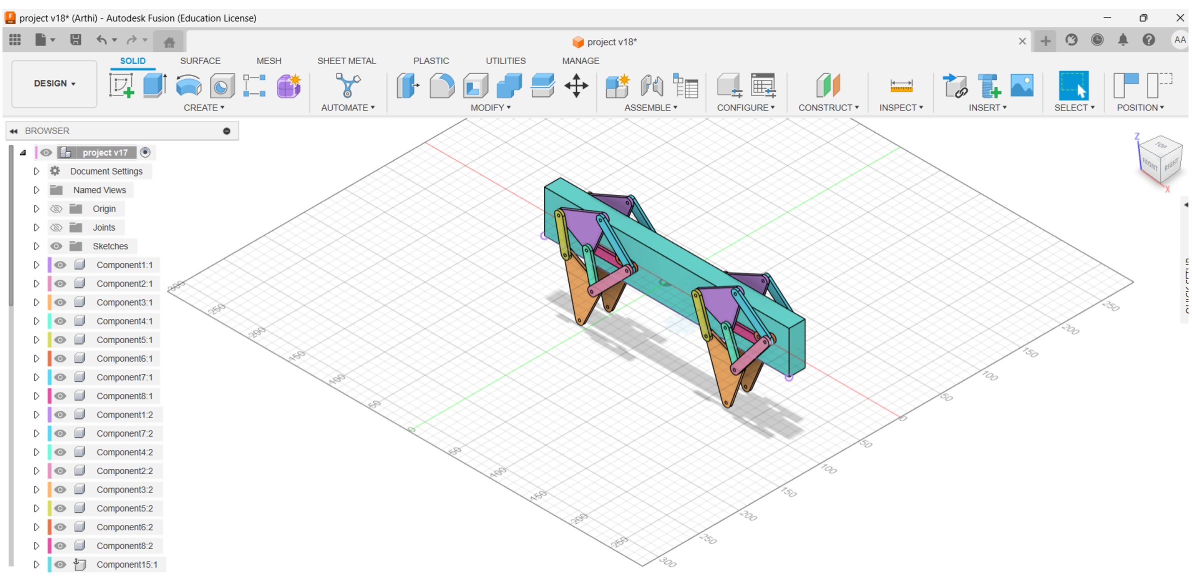

Figure 4: Top View

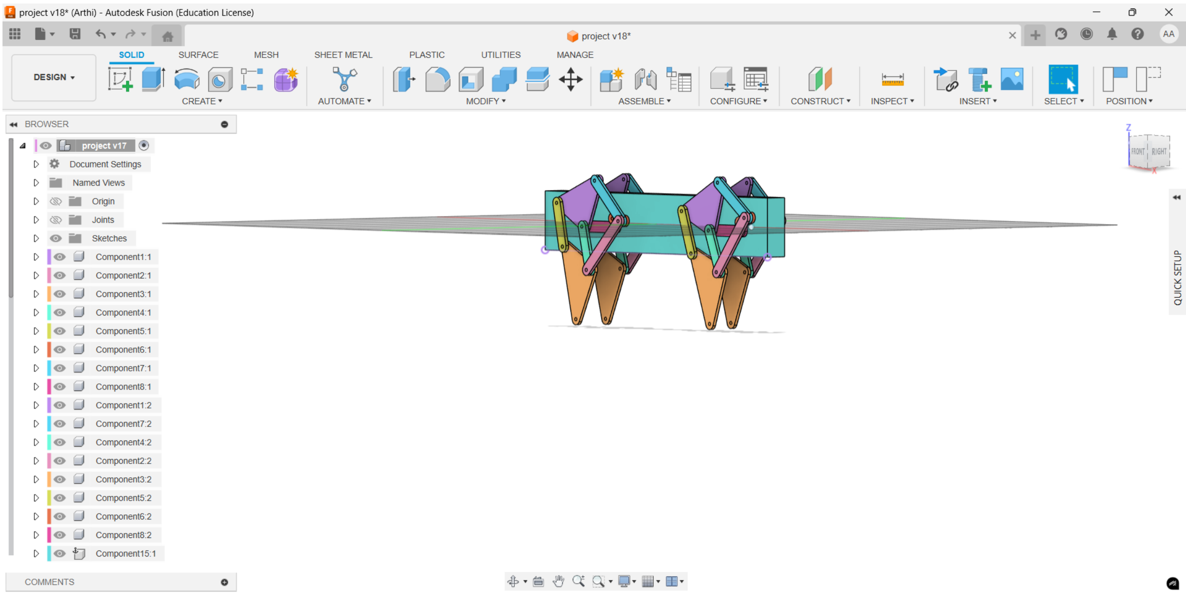

Figure 5: Back Side View

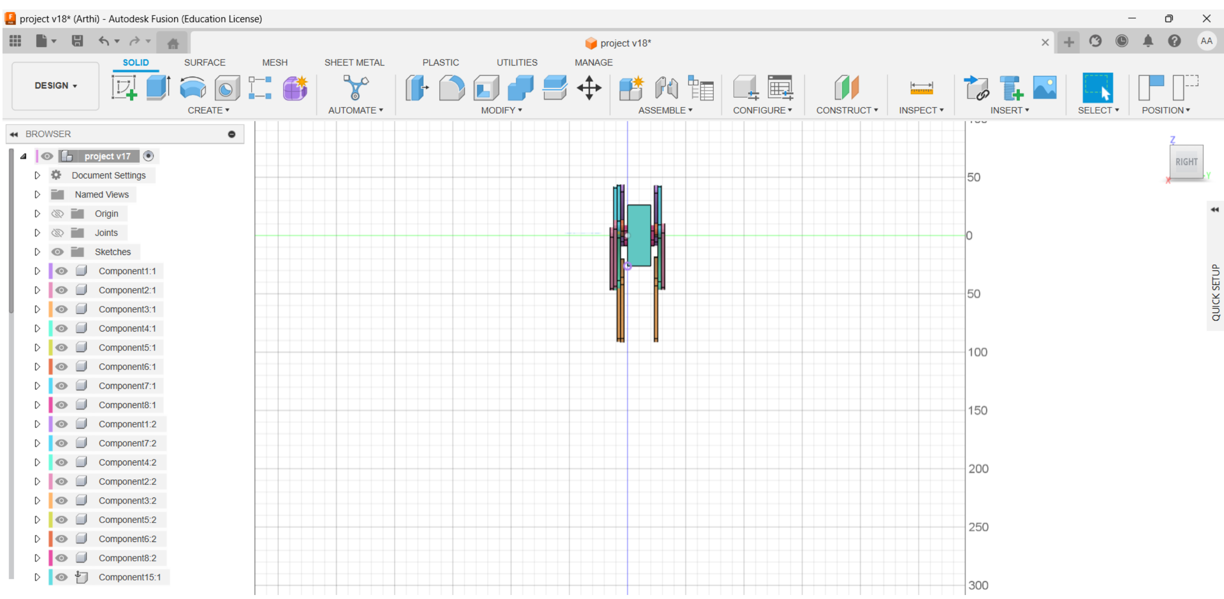

Figure 6: Front View

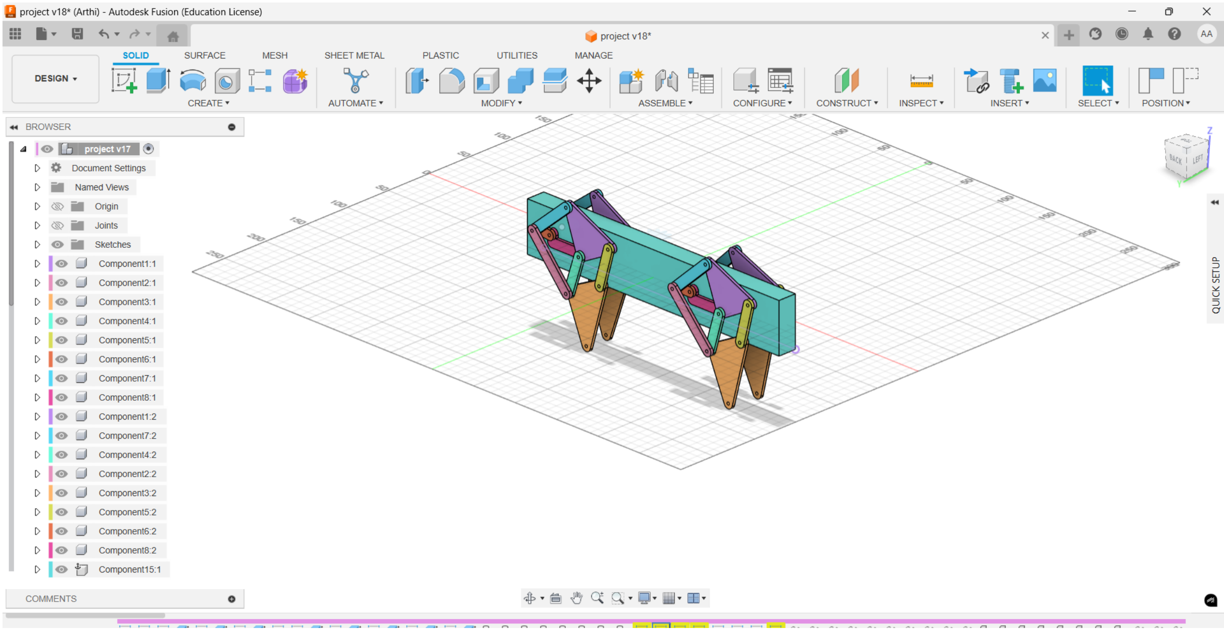

Figure 7: Front Side View

The above images for chassis are representational. The real chassis, designed considering motor and gear sizes, is shown below. The motors are supposed to fit in the middle holes and the holes on the left and right side are for spur gears which will be interlocking each other and hence will support the stability and free movement of the legs. The stl extension files are provided in the zip folder of the report.

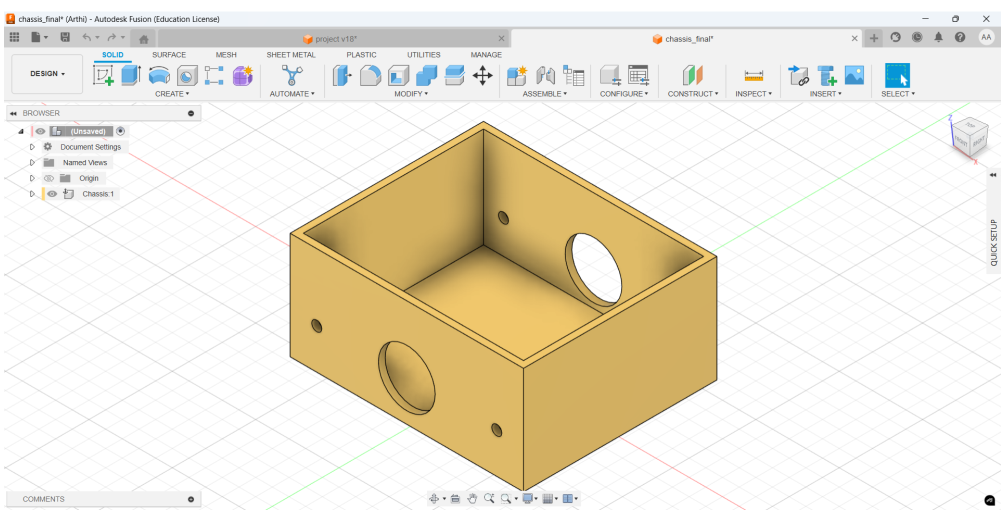

Figure 8: Final Chassis Design

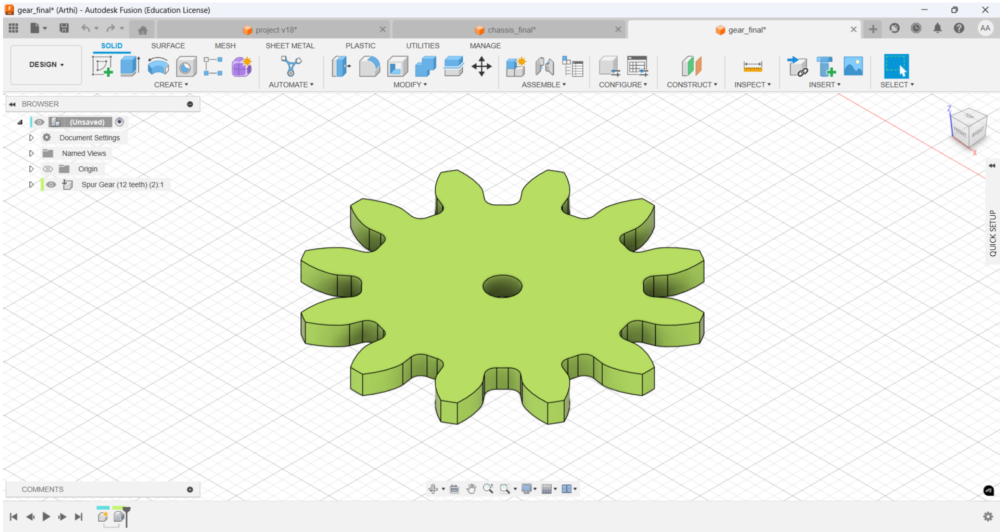

Figure 9: Spur Gear Design

The spur gear attached to the chassis and motor is as attached below:

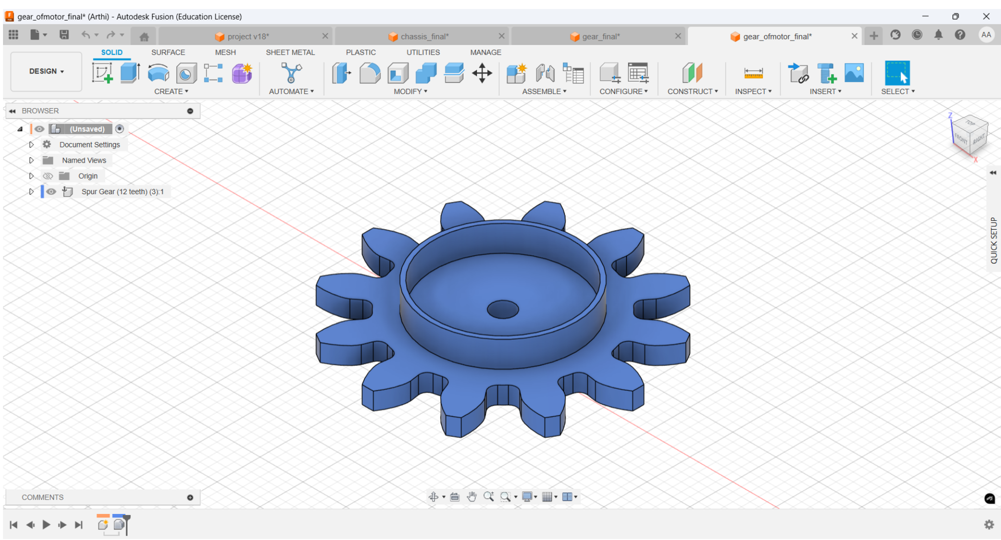

Figure 10: Spur Gear Of The Motor Design

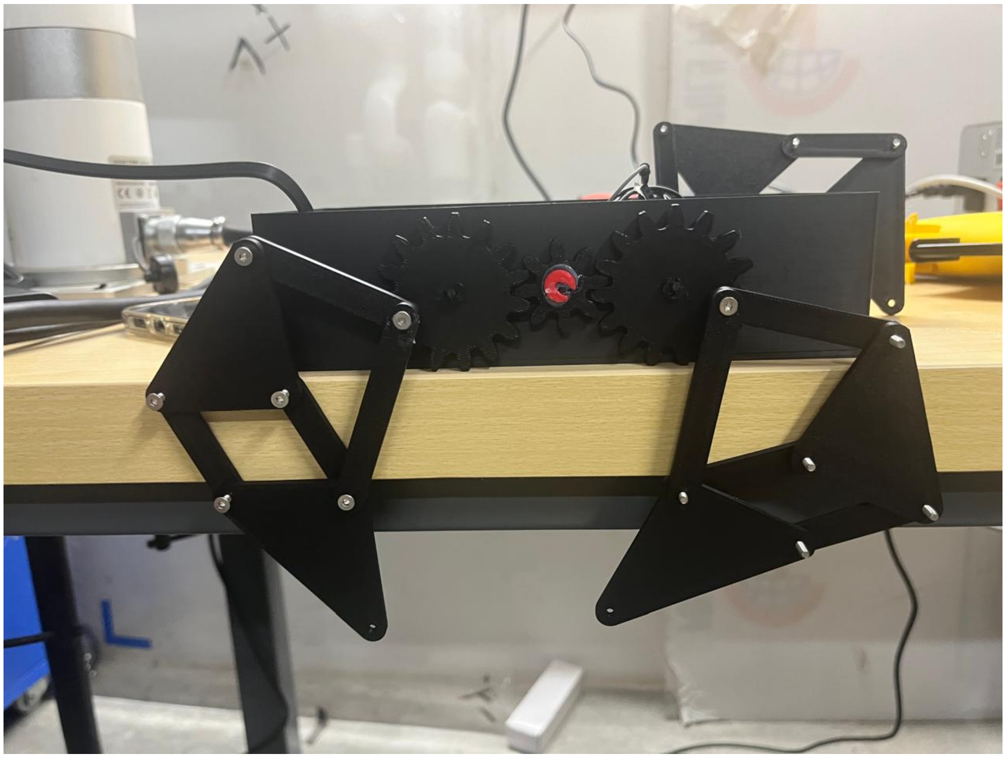

Figure 11: Hardware Image