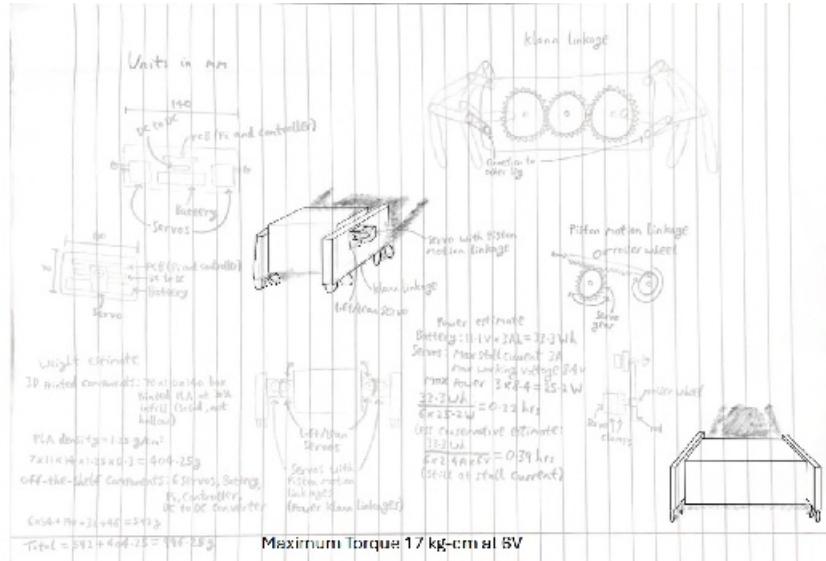


Step by step process

Initial Sketches:

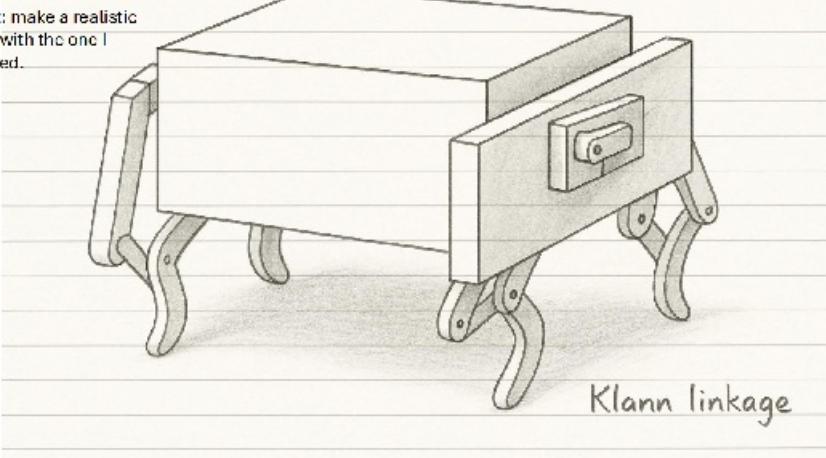
We came up with 3 initial sketches, landing on the crawler idea where we could use one servos to actuate each Klann linkage while using the others for the arm, leaving potential for crab walking later.

Concept 1 - The Crawler

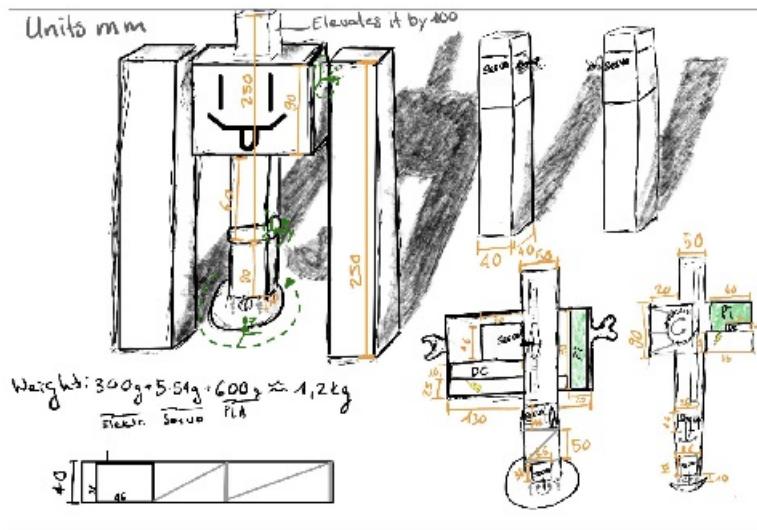


Units in mm

Prompt: make a realistic sketch with the one I uploaded.



Concept 2 - TarsGPT



Battery:

$$11,1V \times 3Ah = 33,3\text{ Wh}$$

Sensors:

max stall current 3A

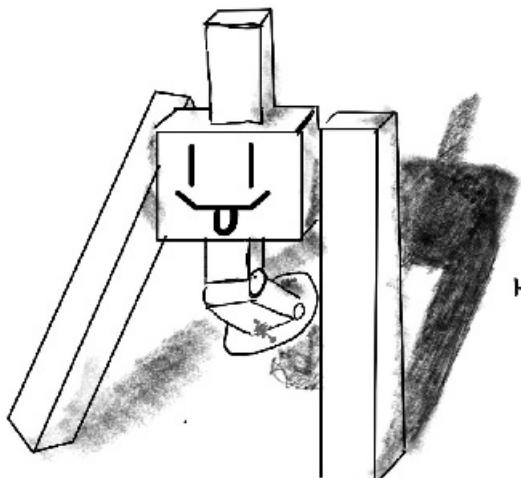
max working Voltage 8,4V

max power $3A \times 8,4V = 25,2W$

$$\frac{33,3\text{ Wh}}{25,2W} = 0,264h$$

less conservative estimate
 $\frac{33,3\text{ Wh}}{5 \times 2,4A \times 6V} = 0,4675h$

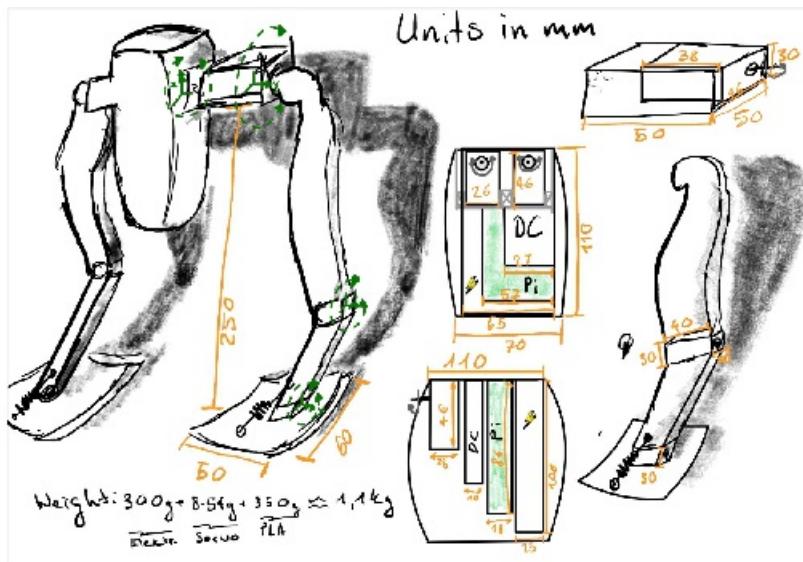
Maximum Torque 17 kg·cm at 6V



Prompt: make a realistic version photo of my sketch which is photographed with a Sony Alpha 7R IV, 85mm f/1.2 lens, silver and graphite tones with technical highlights

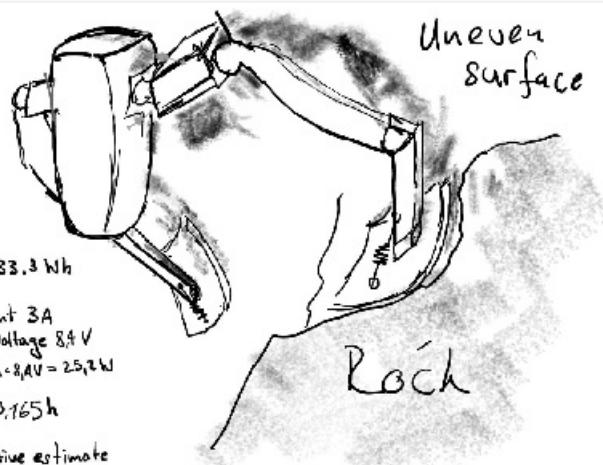


Concept 3 - Rock Climber



$$\text{Weight: } 300\text{ g} + 8.54\text{ g} + 35\text{ g} \approx 1.1\text{ kg}$$

Benzene Succin PLA



$$\begin{aligned}
 & \text{Battery:} \\
 & 11.1V \times 3Ah = 33.3 \text{ Wh} \\
 & \text{Sensors:} \\
 & \text{max stall current } 3A \\
 & \text{max working Voltage } 8.4V \\
 & \text{max power } 3A \times 8.4V = 25.2W \\
 & \frac{33.3 \text{ wh}}{8.4 \times 25.2W} = 0.165h \\
 & \text{less conservative estimate} \\
 & \frac{33.3 \text{ wh}}{8 \times 2Ah \times 6V} = 0.289h \\
 & \text{Maximum Torque } 17
 \end{aligned}$$

Prompt: make a realistic version photo of my sketch which is photographed with a Sony Alpha 7R IV, 85mm f/1.2 lens, silver and graphite tones with technical highlights



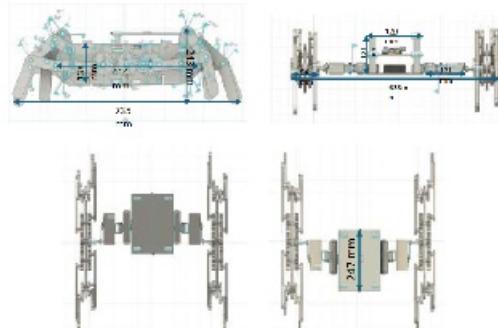
Initial CAD:

Now that we'd decided on our design, we started our early CAD model. We decided to construct our main body as a 'server-style' design with stacked plates for easy access to the electronics. We also decided to keep our Klann linkages exposed for the same reason.

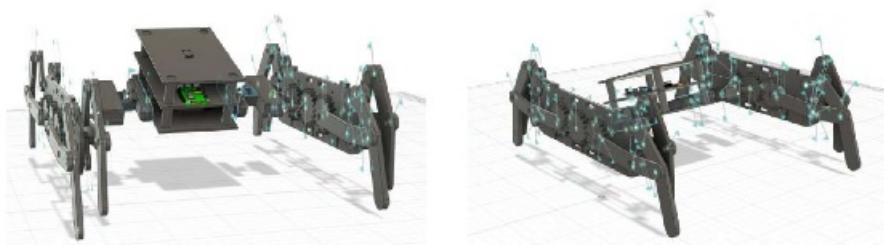
Crawler concept

Specifications:

- Servos – 6
- Estimated weight – 996.25 g
- Estimated Speed – 15 cm/sec

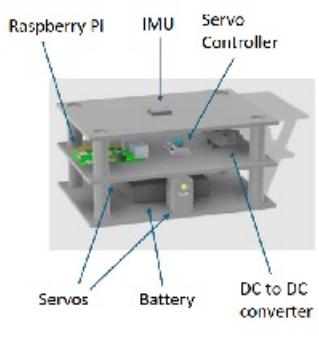


Other poses

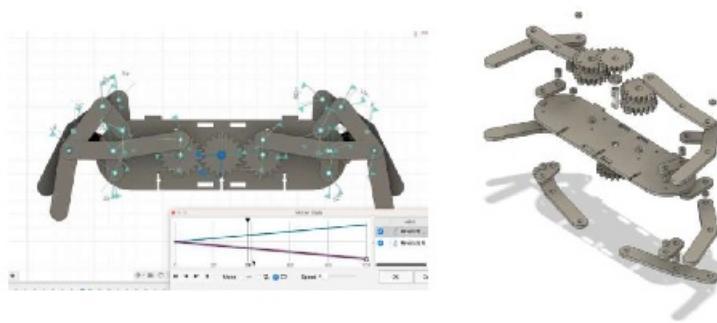


Main Body

Photorealistic renderings

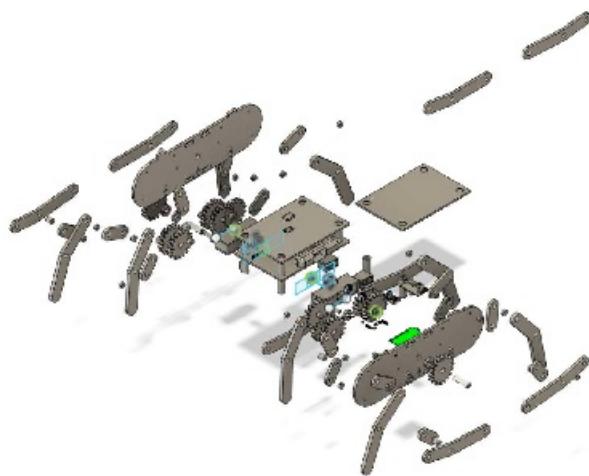


Klann Linkage

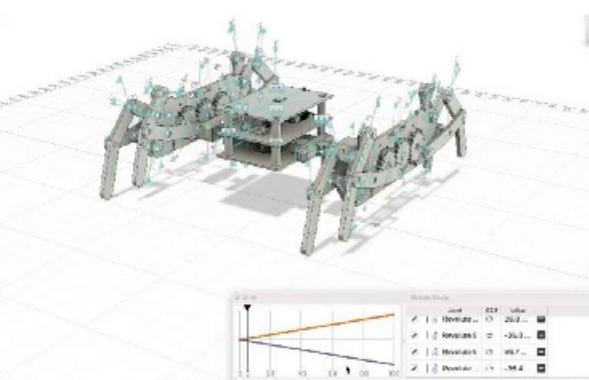


https://drive.google.com/file/d/1w7H_04qPfLwQJmKgDWeSdOViewFucUshing

Exploded view



Animation



https://drive.google.com/file/d/1N8KUJ3BvmpR_BuiWzokTop/view?usp=sharing



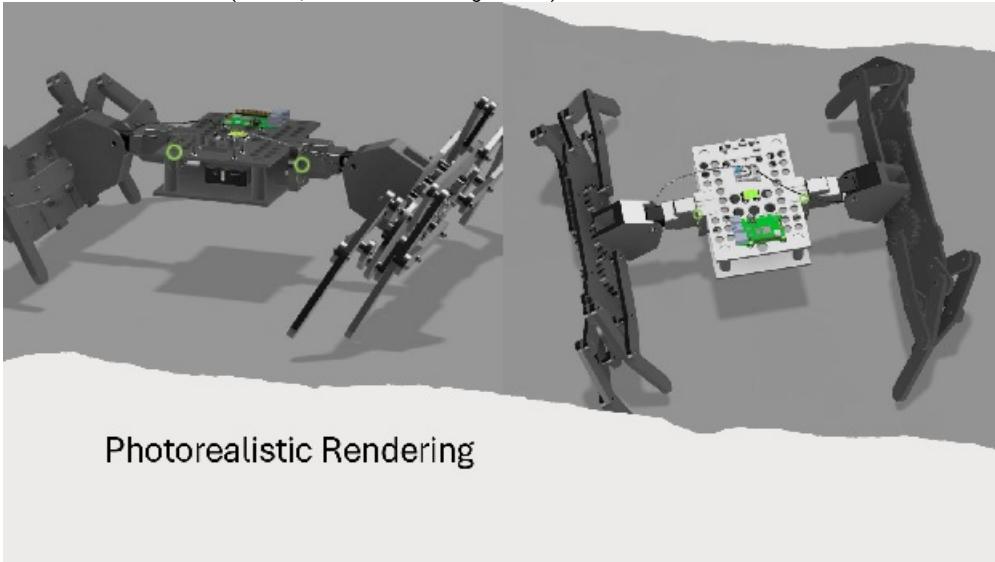
Photorealistic rendering



Context rendering

Detailed CAD:

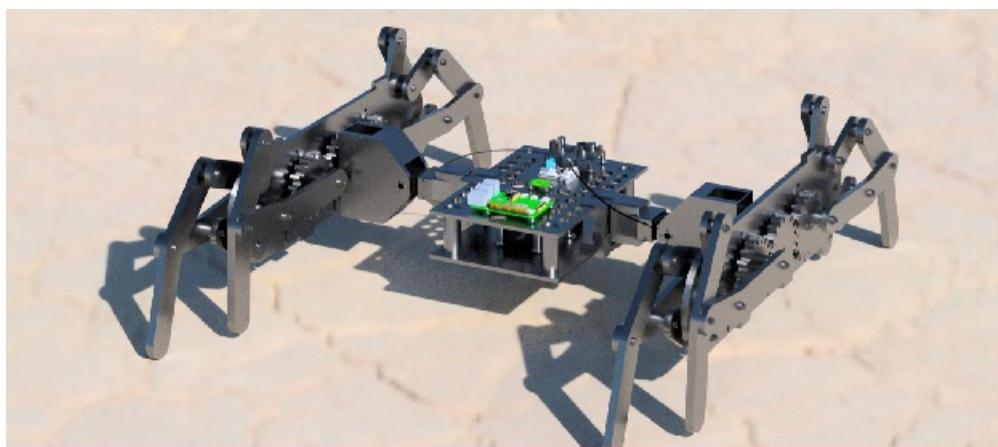
Next we finalized our CAD (for now, we would still change it later). The main additions here were smaller details like screws, wires and detailed electronic components.



Photorealistic Rendering



Photorealistic Rendering



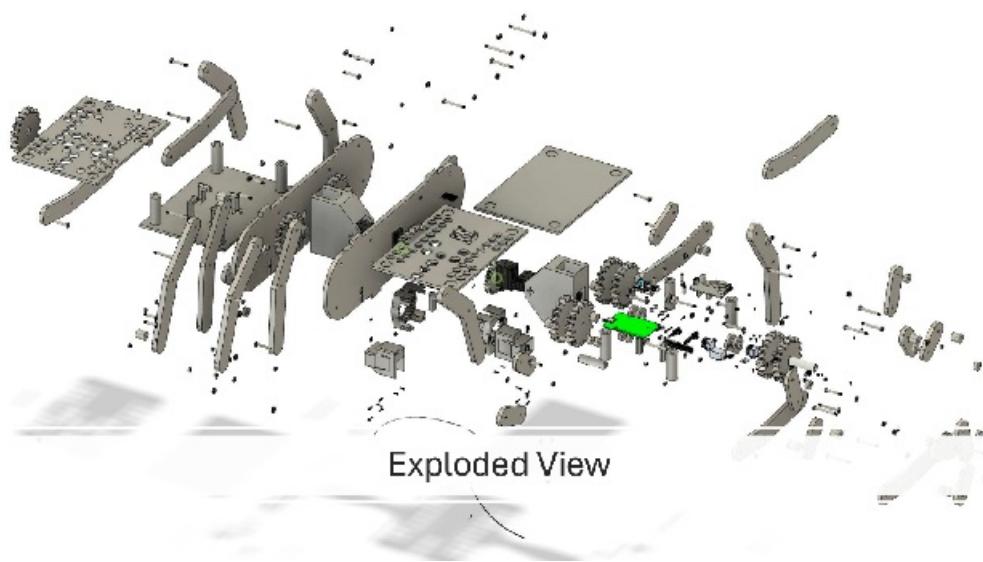
Context Rendering

Context Rendering



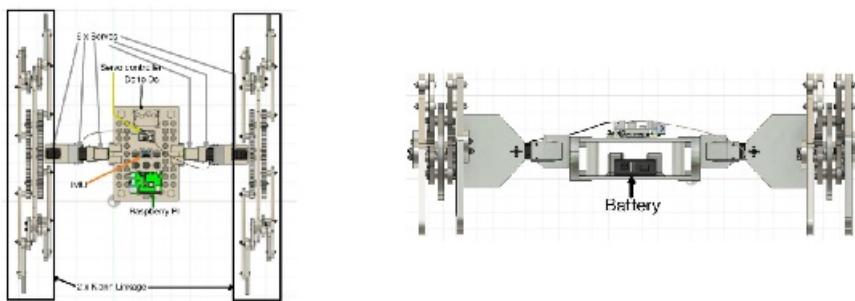


Context Rendering

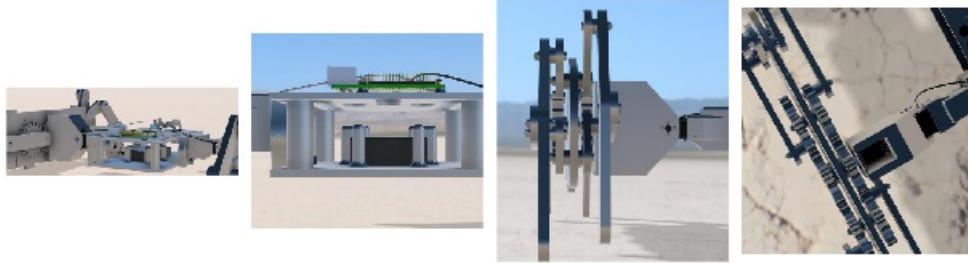


Exploded View

Key Components

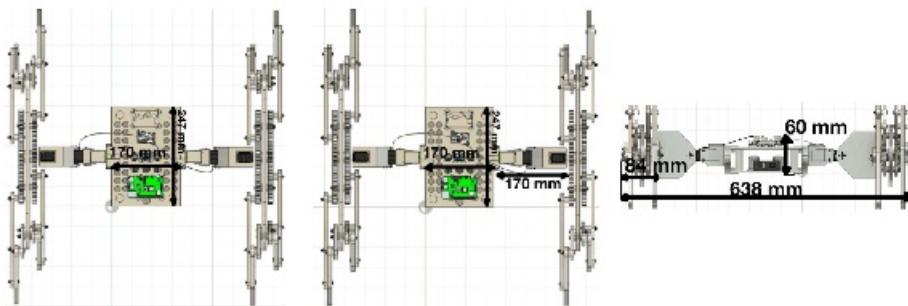


Detailed Close-Ups



Key specs and dimensions

- Servos – 6
 - Estimated weight – 996.25 g
 - Estimated Speed – 15 cm/sec



Bill of Materials



Working Leg:

Next we started construction on the first leg. We got the basic servo motor circuit off the test code on a laptop while 3D printing the components. It wasn't perfect, we redesigned it later to be full of holes to save filament and would go on to print our own pins to replace the screws which would either seize up or unscrew completely.



Motor attachment

- Initially we tried screwing an attachment between the gear shaft and the servo but there wasn't enough space
- Redesigned the gear shell to be two locking pieces with an end which slots directly into the servo spline

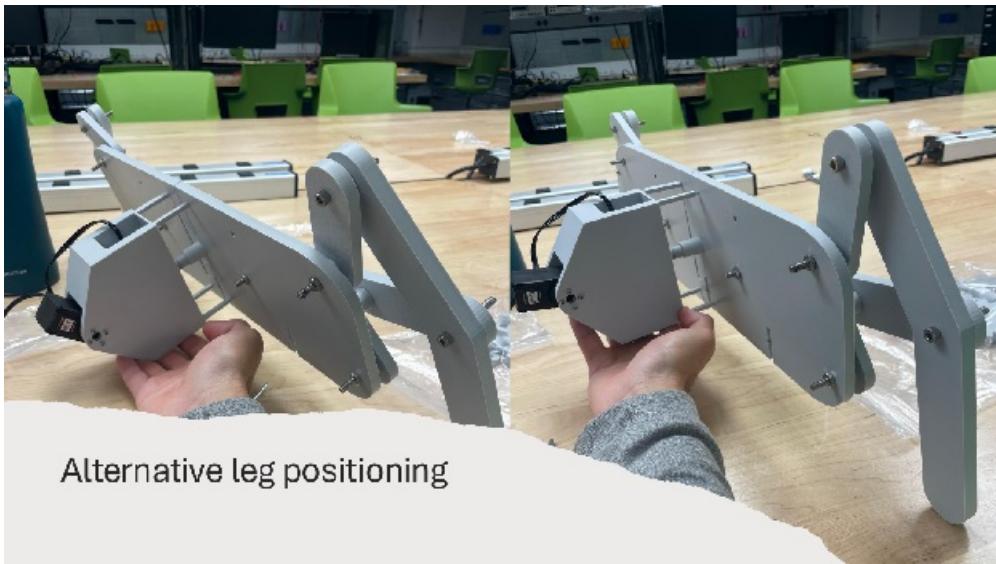


Leg in motion

Video

<https://drive.google.com/file/d/1mxgwMdxHgfCVJ4kEvcBoYHfmsbf0KtQY/view?usp=sharing>

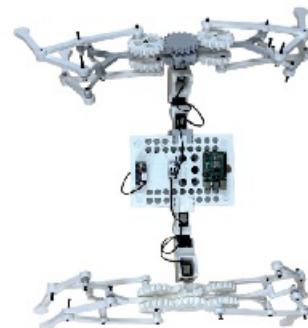




Alternative leg positioning

Full Assembly:

While we redesigned the legs, we also printed the rest of the full assembly. This was far more simple than the complicated leg linkages. Aside from some issues with screwing in the servos into their respective brackets, this was relatively painless.

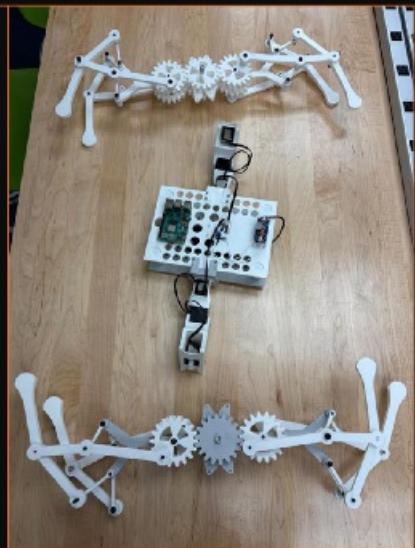


Glamour photo

Multiple poses



Modularity

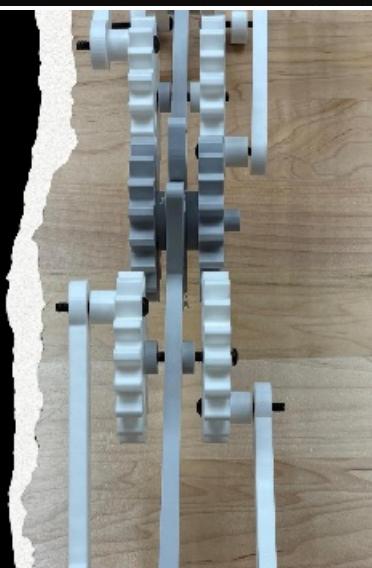


Modularity - parts



Assembly issues

- We identified screws aren't ideal
- Also found that standoffs are unstable

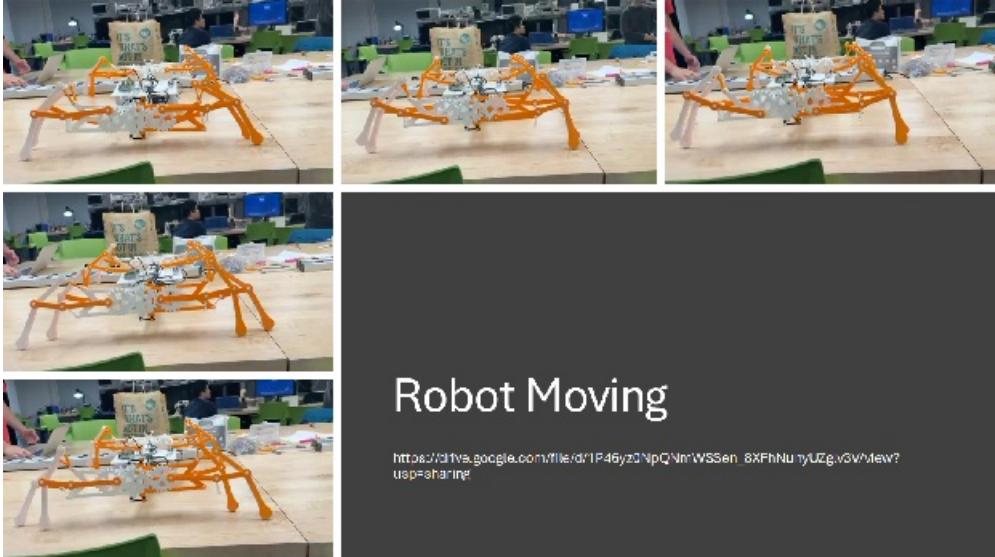


```
takuyab@Takuyas-MacBook-Pro-578 PyLX-16A-master % python3 servo_adjustment_test.py
servo 1 is at 90.96 degrees
servo 1 is now at 90.96 degrees
servo 2 is at 93.12 degrees
servo 2 is now at 92.16 degrees
servo 3 is at 91.68 degrees
servo 3 is now at 91.44 degrees
servo 4 is at 92.16 degrees
servo 4 is now at 91.44 degrees
servo 5 is at 92.64 degrees
servo 5 is now at 91.2 degrees
servo 6 is at 92.4 degrees
servo 6 is now at 91.68 degrees
takuyab@Takuyas-MacBook-Pro-578 PyLX-16A-master %
```

Boottest

Baby Steps:

Finally, it was time to test the whole robot walking. This is exactly what it looks like, aside from one of the pin caps coming off from the legs, it worked for an initial gait.



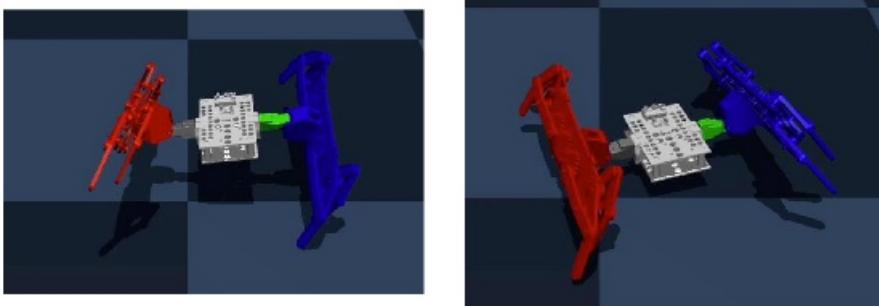
Robot Moving

https://drive.google.com/file/d/1P46yz0NpQNmWSSen_8XPhNuNyUZg.v3v/view?usp=sharing

Simulation:

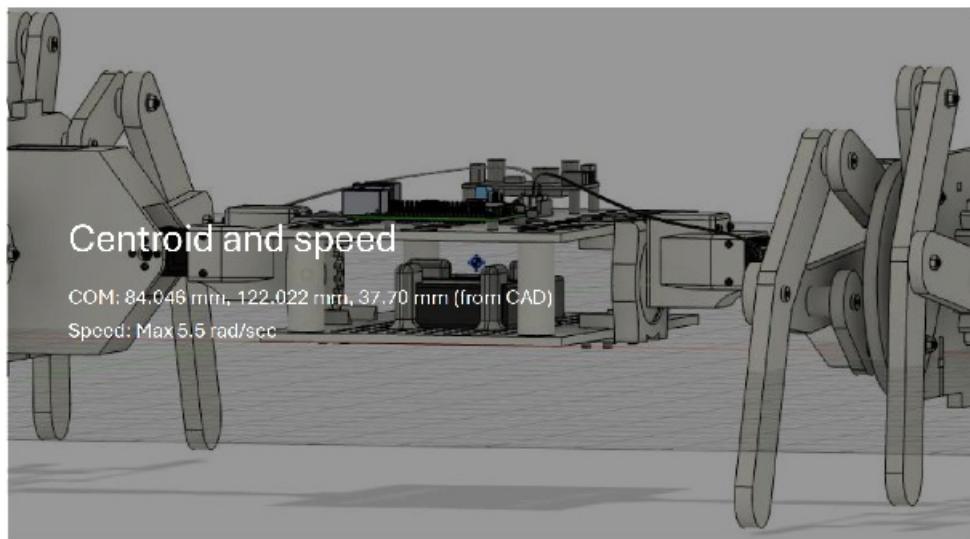
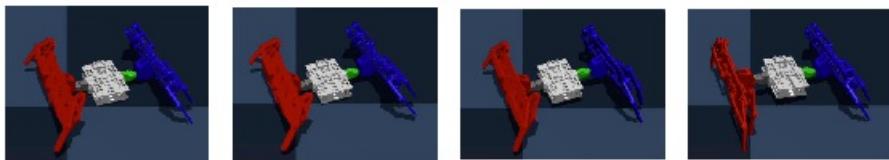
Next we exported the CAD assembly into the MuJoCo simulation, putting a rudimentary sin-based gait to make it shuffle sideways without using the Klann linkages.

Screenshots

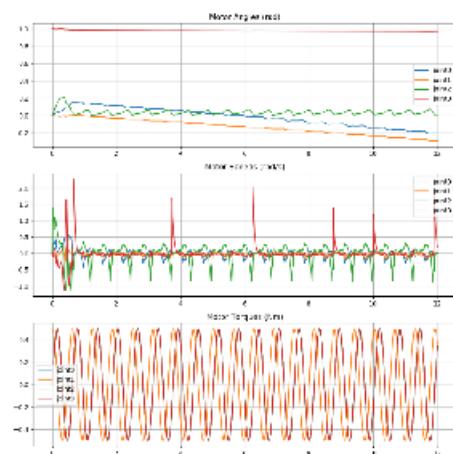


Video and frames

https://drive.google.com/file/d/10nFBPiPwV_FAH7TwGypf2ife66V8HaQ/view?usp=sharing



Motor Plots





```

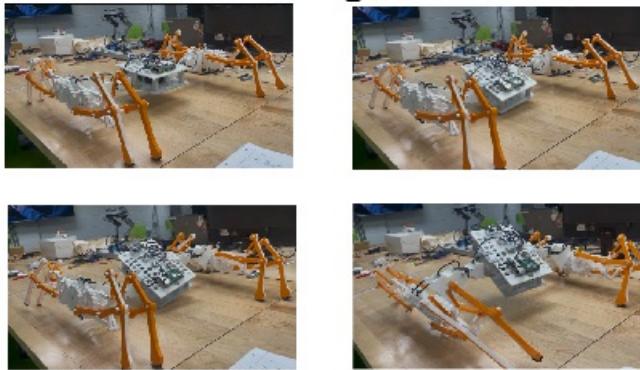
<?xml version="1.0" encoding="UTF-8"?>
<robot name="crabwalk">
  <link name="base_link">
    <visual>
      <geometry>
        <box size="0.1 0.1 0.1" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="1.0" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <link name="left_chassis_link">
    <parent link="base_link" />
    <visual>
      <geometry>
        <box size="0.05 0.05 0.05" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="0.5" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <link name="right_chassis_link">
    <parent link="base_link" />
    <visual>
      <geometry>
        <box size="0.05 0.05 0.05" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="0.5" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <link name="left_leg_link">
    <parent link="left_chassis_link" />
    <visual>
      <geometry>
        <box size="0.02 0.02 0.02" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="0.2" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <link name="right_leg_link">
    <parent link="right_chassis_link" />
    <visual>
      <geometry>
        <box size="0.02 0.02 0.02" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="0.2" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <link name="left_foot_link">
    <parent link="left_leg_link" />
    <visual>
      <geometry>
        <box size="0.01 0.01 0.01" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="0.1" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <link name="right_foot_link">
    <parent link="right_leg_link" />
    <visual>
      <geometry>
        <box size="0.01 0.01 0.01" />
      </geometry>
      <material>
        <color r="0.7 0.7 0.7" g="0.7 0.7 0.7" b="0.7 0.7 0.7" />
      </material>
    </visual>
    <inertial>
      <origin>
        <xyz>0 0 0</xyz>
        <rpy>0 0 0</rpy>
      </origin>
      <mass value="0.1" />
      <inertia ixx="0.001" ixy="0.001" ixz="0.001" iyy="0.001" iyz="0.001" izz="0.001" />
    </inertial>
  </link>
  <joint name="left_chassis_joint" type="fixed">
    <parent link="base_link" />
    <child link="left_chassis_link" />
    <origin>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </origin>
    <axis>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </axis>
  </joint>
  <joint name="right_chassis_joint" type="fixed">
    <parent link="base_link" />
    <child link="right_chassis_link" />
    <origin>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </origin>
    <axis>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </axis>
  </joint>
  <joint name="left_leg_joint" type="revolute">
    <parent link="left_chassis_link" />
    <child link="left_leg_link" />
    <origin>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </origin>
    <axis>
      <xyz>0 0 1</xyz>
      <rpy>0 0 0</rpy>
    </axis>
  </joint>
  <joint name="right_leg_joint" type="revolute">
    <parent link="right_chassis_link" />
    <child link="right_leg_link" />
    <origin>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </origin>
    <axis>
      <xyz>0 0 1</xyz>
      <rpy>0 0 0</rpy>
    </axis>
  </joint>
  <joint name="left_foot_joint" type="prismatic">
    <parent link="left_leg_link" />
    <child link="left_foot_link" />
    <origin>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </origin>
    <axis>
      <xyz>0 0 1</xyz>
      <rpy>0 0 0</rpy>
    </axis>
  </joint>
  <joint name="right_foot_joint" type="prismatic">
    <parent link="right_leg_link" />
    <child link="right_foot_link" />
    <origin>
      <xyz>0 0 0</xyz>
      <rpy>0 0 0</rpy>
    </origin>
    <axis>
      <xyz>0 0 1</xyz>
      <rpy>0 0 0</rpy>
    </axis>
  </joint>
</robot>

```

Machine Learning:

Next was using machine learning and reinforcement learning to teach the robot to crabwalk. First, multiple simpler methods were tried in simulation. Random search, hill climber and random parallel mutation hill climber were all tried to optimize the parameters on a sin-based gait. Next we moved on to using reinforcement learning in simulation, initially trying REINFORCE before settling on PPO. Once we had a PPO-based gait in MuJoCo we attempted sim-to-real. We did get movement but as can be seen in the video, it was not exactly an effective translation!

Reinforcement Learning

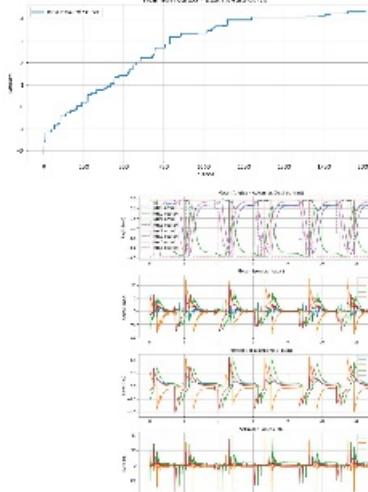
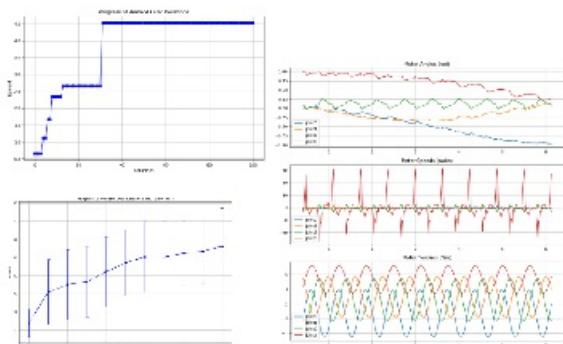


https://drive.google.com/file/d/1aYf1yVu_bxjPpcCKrTVdksLZSF3tXNW/view?usp=sharing

Reinforcement Learning (PPO)

Learning curves

Hill climber/random search



Other environments/damage scenarios

- https://drive.google.com/file/d/1QCdOclAsBaW93KewpgJwEKeqPBw_e6pY/view?usp=sharing
- <https://drive.google.com/file/d/1nJksYuwPT1t6NrBhaE89B1abymilDP-/view?usp=sharing>

Side by side

<https://drive.google.com/file/d/11bgbTCCPymhbP3AIMRv99k7sLTgPEgB8/view?usp=sharing>