

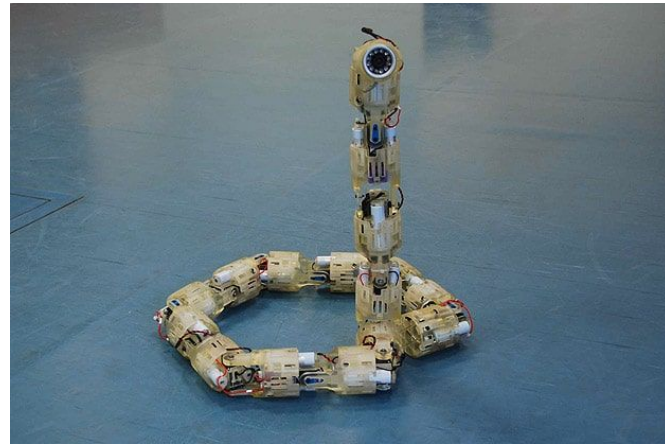
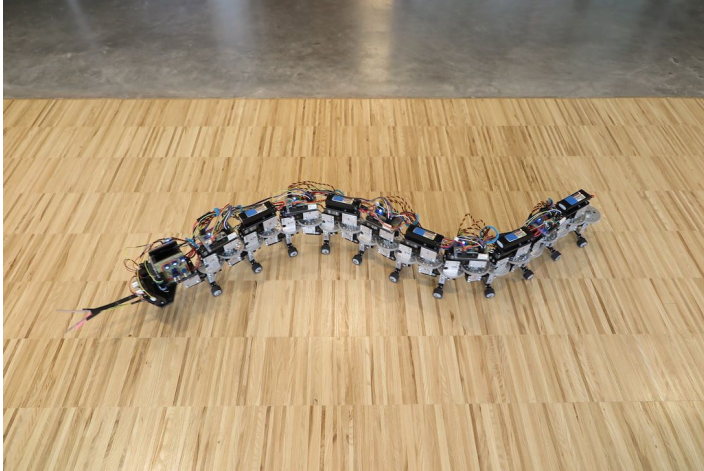
# Snakebot

Anirudh Gudi(Ms Robotics)  
Harshini Murugadoss(Ms Robotics)  
Prerana Pemmaraju(Ms Robotics)

ME225 - Dr. Sheng  
Department of Mechanical Engineering  
University of California, Riverside

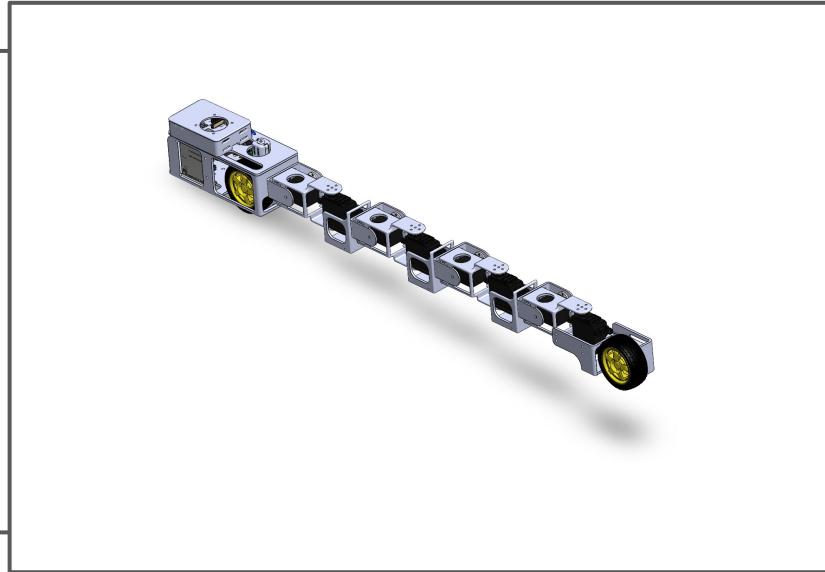
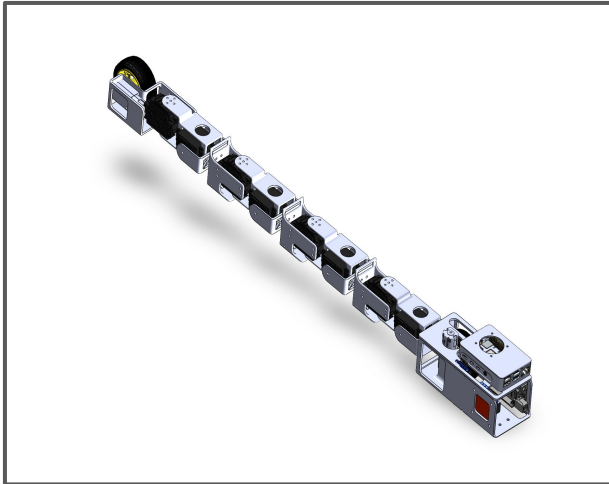
# Background

- Lightweight and compact robots are ideal for accessing dangerous or hard-to-reach environments.
- Snake-like robots can navigate through narrow gaps and rough terrain by mimicking biological snakes.
- Useful in disaster zones like **minefields** and **earthquake-affected areas** for inspection and rescue tasks.



# Design

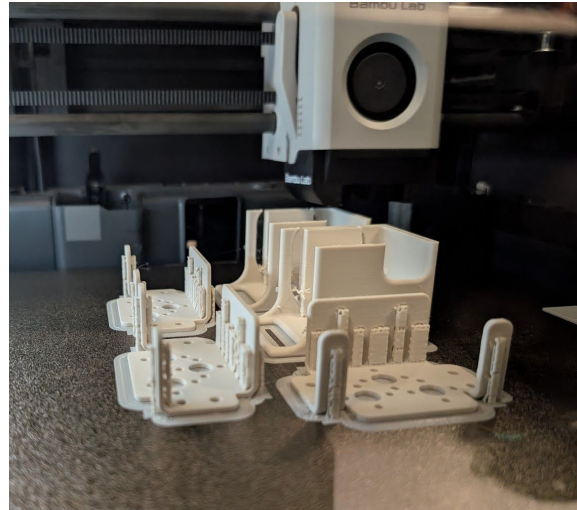
- The robot is inspired by the movement of real snakes, allowing it to maneuver through tight and uneven spaces.
- It uses **8 metal gear servo motors** for body articulation and serpentine motion.
- **2 DC motors** are used to drive the robot forward and backward on flat surfaces.



# Mechanical Design Specifications

## Chassis:

- Material: 3mm thick PLA sheet
- Shape : custom
- Process: 3D Printing



# Mechanical Design Specifications

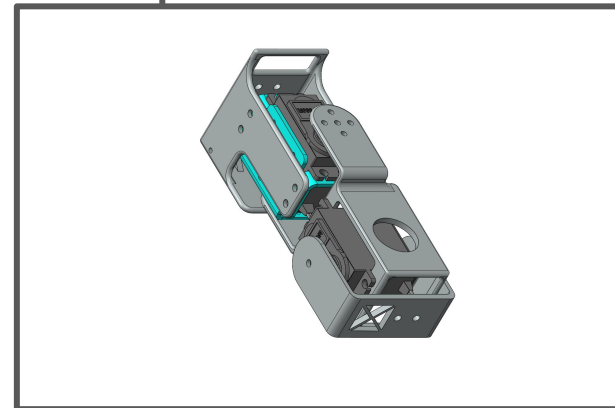
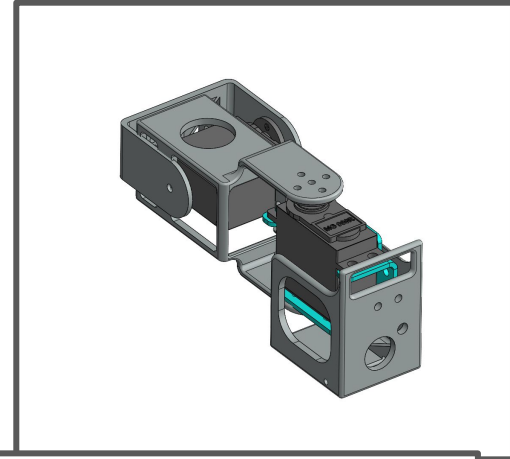
## Links -

Each link except the first and last links has 2 servo enabling the yaw and pitch motions  
Yaw servo moves the previous link around the y - axis leading to a serpentine motion .

The pitch servo lifts the previous link around x-axis enabling the obstacle maneuvering

The first and last links hold the dc motors for the forward locomotion.

The first link includes all the major components including the raspberry pi, pca9685, l298n, battery



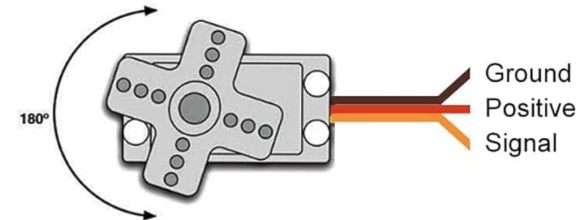
# Mechanical Design Specifications

## Motors:

- Type: High-torque servo motors, ideally with feedback for precise control.
- Quantity: 2 per link, 8 in total.
- Placement: Two motors in each link.



Product Model	MG995
Maximum Angle	180 degrees
Operating Voltage	5.0 - 6.0V
Operating Speed	0.11 seconds / 60 degrees
Temperature Range	0 °C -- 55 °C
The dead-band setting	5 microseconds



# Calculations

Link1 - 70 gms

Link 2 .. 4 - 60 gms

Link 5 - 150 gms

Distances -

Link 1 : 10

Link 2 , 3 , 4 : 6

Link 5 : 10

Force =  $m \times a$

$$M = 70 + 60 + 60 + 60 + 150 = 400$$

$$F = 400 * 9.8 = 3920 = 3.920 \text{ N}$$

Total  $d \sim 0.4 \text{ m}$

$$\text{Total Torque required} = F \times d = 3.92 \times 0.4 = 1.568 \text{ Nm} \sim 1.6 \text{ Nm}$$

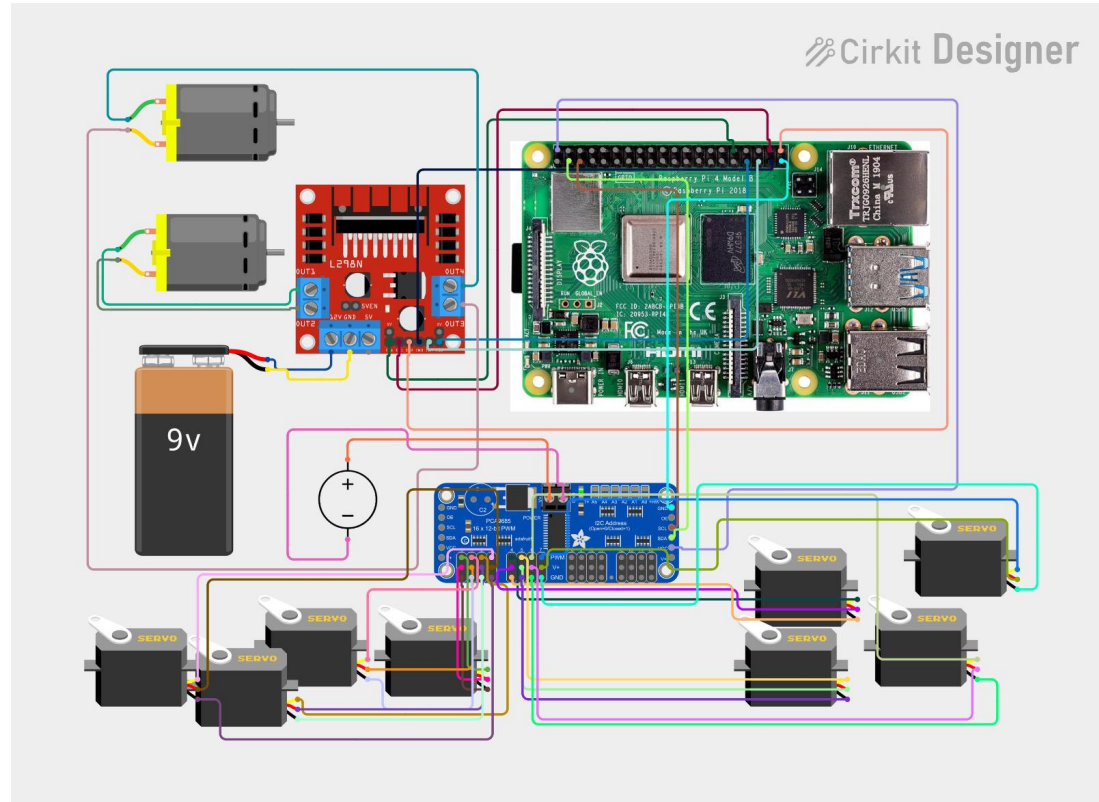
$$\text{Power} = T \times w \quad \{ 100 \text{ rpm motor} \}$$

$$W = (2 \times \pi \times N) / 60$$

$$W = (2 \times 3.14 \times 100) / 60 = 10.467 \text{ rad / s}$$

$$\text{Power} = 1.6 \times 10.467 = 16.747 \text{ W}$$

# Circuit Diagram -



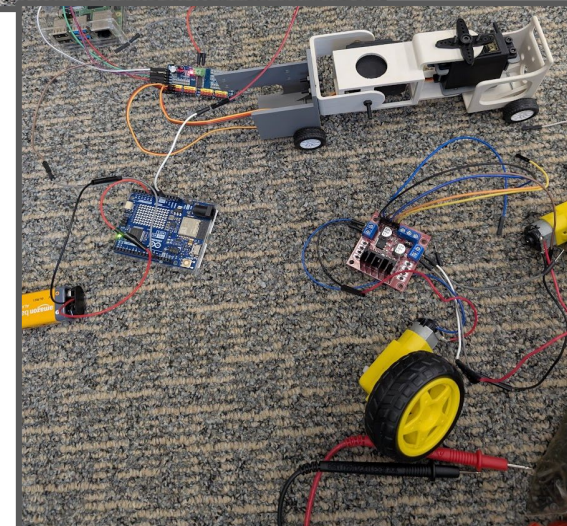
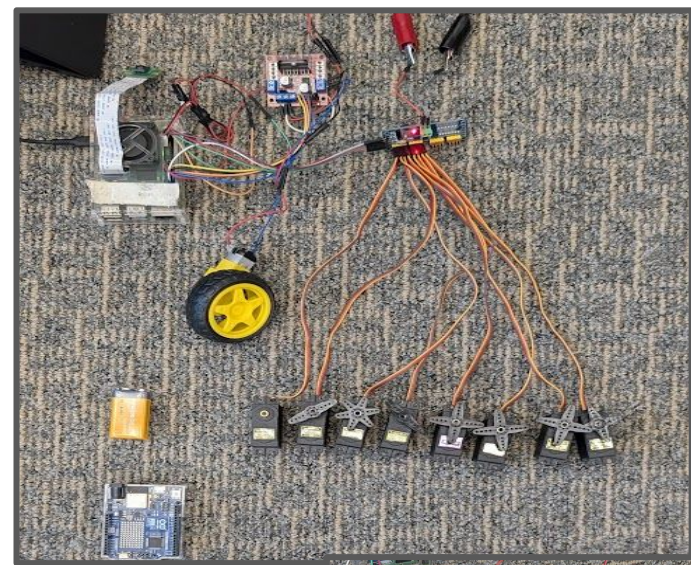
# Electrical Design

- **All servo motors** were directly connected to the **PCA9685** driver using 3-pin jumper wires (**5V**, **GND**, and **Signal**).
- The **PCA9685** was connected to the **Raspberry Pi 4** via the I2C interface (**SDA**, **SCL**).
- **DC motors** were connected to the **L298N motor driver**, which was also controlled by GPIO pins from the Raspberry Pi.
- Power and ground connections were made using **jumper wires**, with power supplied from a regulated source to both the PCA9685 and L298N.



# Servo & DC motor Testing

- Vin and Gnd pins used to power the servos
- Sample code written to home all servos to 90 degree (home position)
- Afterwards the joints were attached to the servos at the home position
- The legs range of motion was tested and verified.
- Powered all the servos together to check current draw
- Completed the hardware ready to be tested with the leg movement algorithm
- DC motors are tested to understand their rpm.
- Tuned to adjust their speeds and directions.

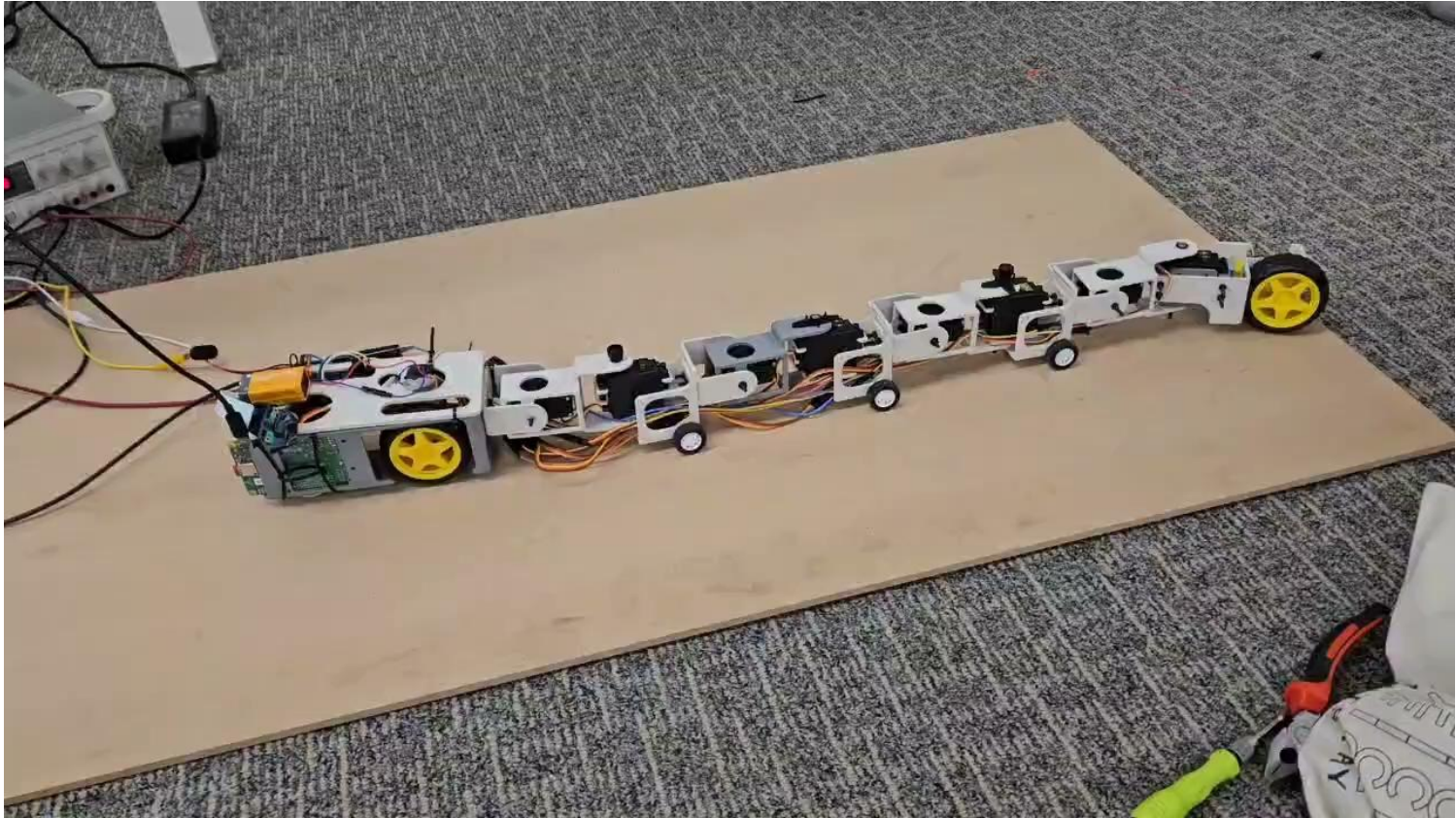


# Mechanical Goal

The snake robot performs various maneuvers using coordinated servo and motor control:

- **Serpentine Motion:** Snake-like lateral undulation using 8 servo motors for realistic crawling movement.
- **Forward & Backward Motion:** Achieved using 2 DC motors for linear movement on flat surfaces.
- **Turning Maneuver:** By adjusting the timing and angle of servo movements, the robot can turn smoothly while crawling.

# Demonstration





Thank You