INTRODUCCION TO ROBOTICS.

Group B2

Member:

- Cristhyan Ojeda
- Mariano Ovelar
- Juan Farina

Professor: Shuhao Liang.

T.A. Marnel Altius

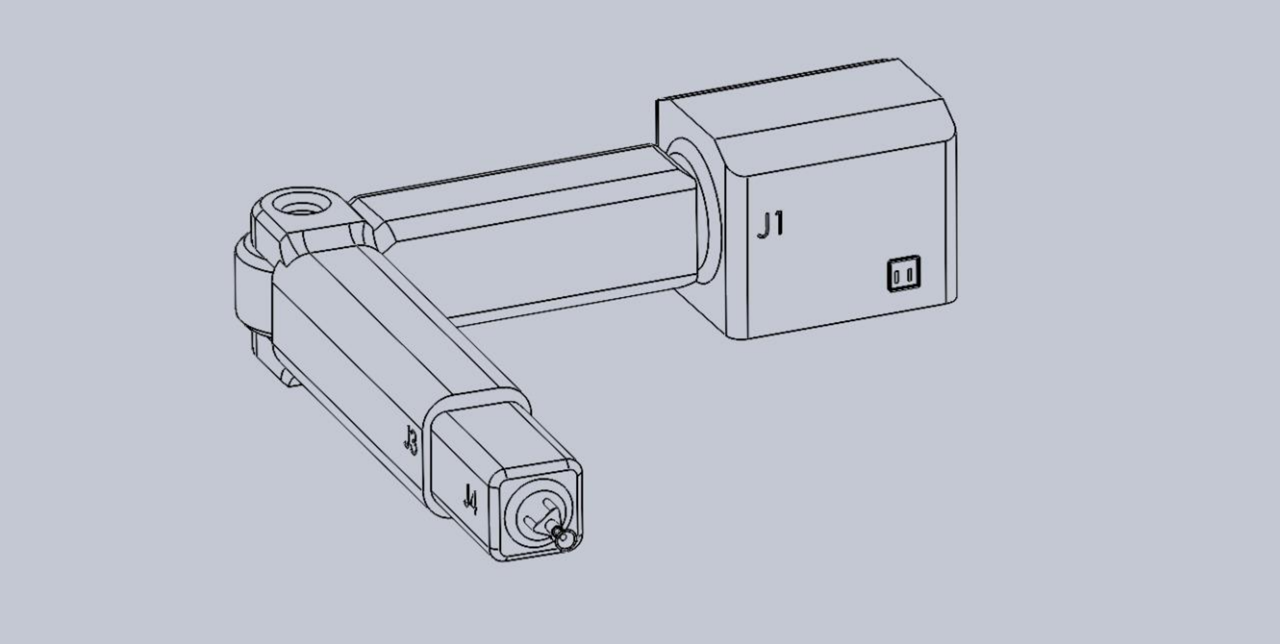
Why We Designs this robot?

The **BATT-BOT** was designed to be an affordable solution for industrial automation, especially for small and medium-sized businesses looking to streamline their processes without incurring the high costs of traditional industrial robots.

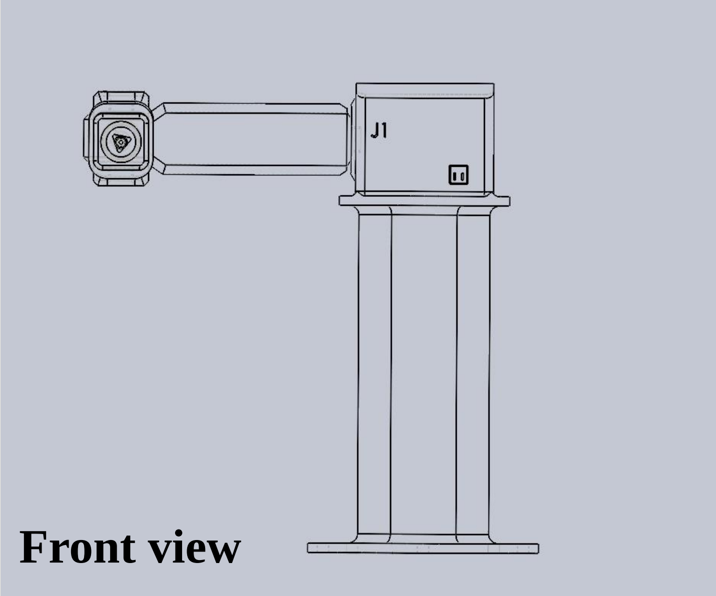
Reasons for Low Cost:

1. **Simplified Design:**
2. **Use of Economical Components:**
3. **Ease of Maintenance:**
4. **Energy Efficiency:**

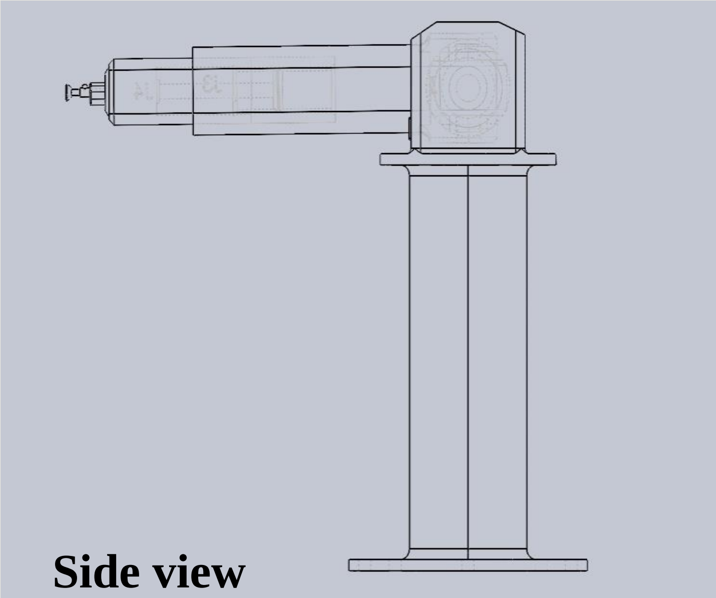
Benefit for Industry



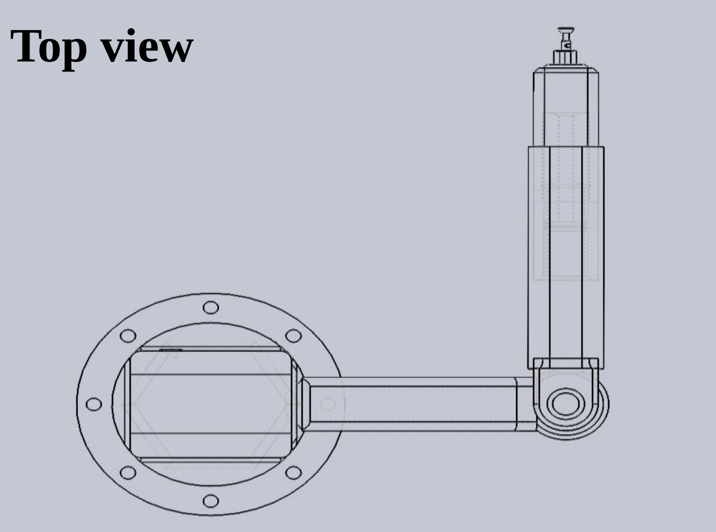
Schematic designs



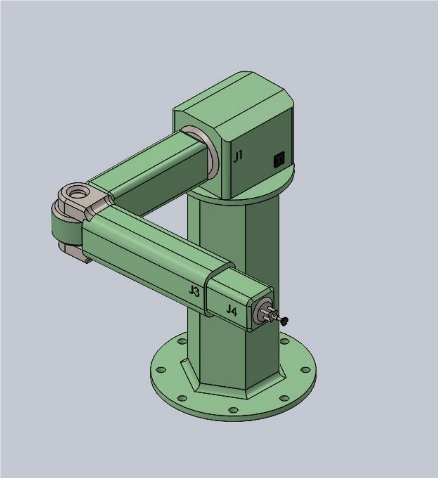
Front view



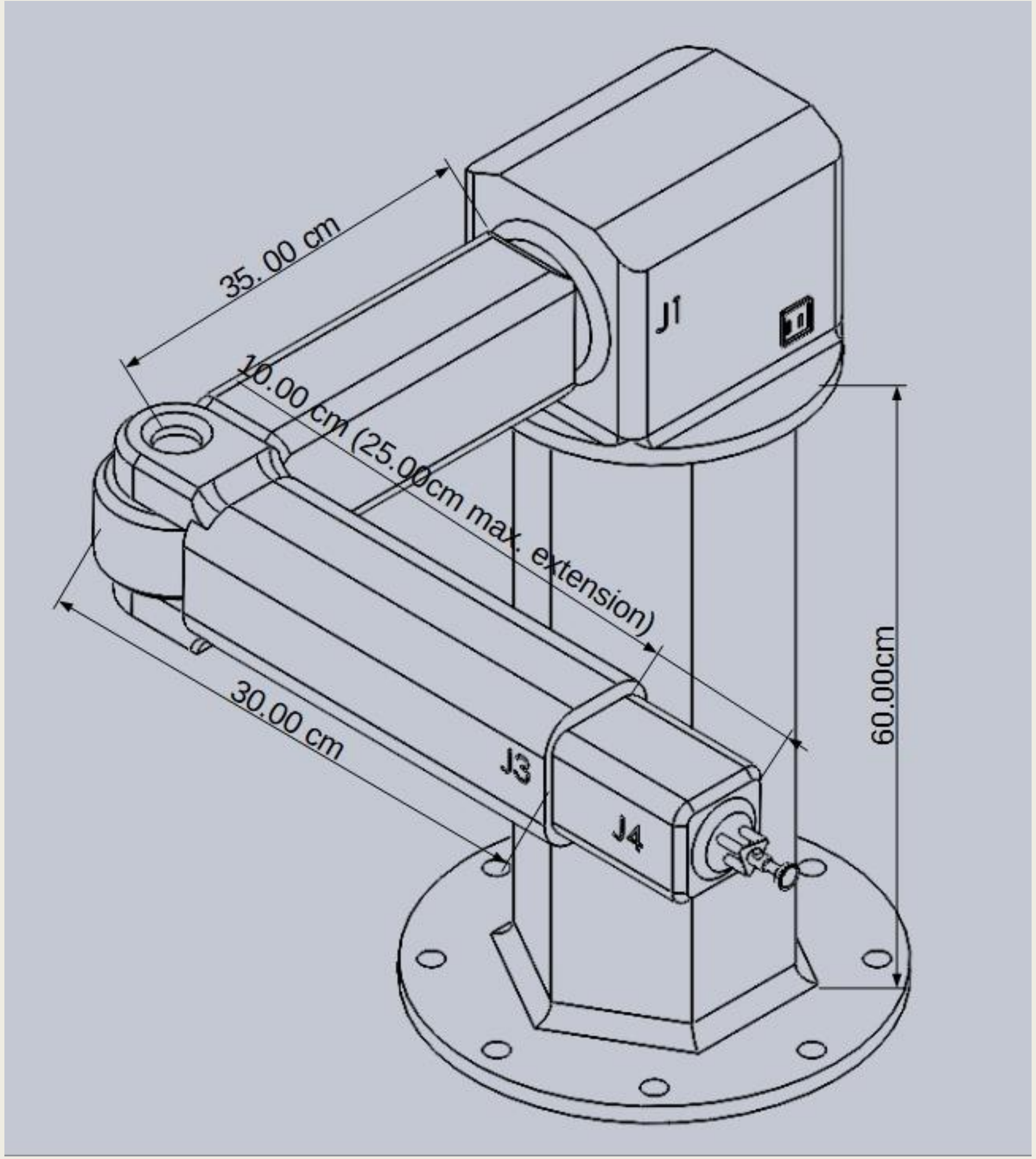
Side view



Top view



Isometric view



Isometric and Dimensions view

Mathematical Model of the Robot

The mathematical model of a robotic arm consists of two main components:

1. **Kinematics**
2. **Dynamics**

1. Kinematics

1.1. Denavit-Hartenberg (DH) Parameters

Using the DH convention,

Link	θ (rad)	d (mm)	a (mm)	α (rad)	Joint Type
1	0	0	35	0	Rotational
2	0	0	40	0	Rotational
3	0	0	0	0	Prismatic
4	0	0	15	0	Rotational

1.2. Homogeneous Transformation Matrix

The transformation matrix for each link is given by:

$$T_i^i + 1 = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 1: $\boldsymbol{T_0^1}$

For the first link:

- $\theta_1 = 0, d_1 = 0, a_1 = 35, \alpha_1 = 0$

$$T_0^1 = \begin{bmatrix} 1 & 0 & 0 & 35 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 2: $\boldsymbol{T_1^2}$

For the second link:

- $\theta_2 = 0, d_2 = 0, a_2 = 40, \alpha_2 = 0$

$$T_1^2 = \begin{bmatrix} 1 & 0 & 0 & 40 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 3: $\boldsymbol{T_2^3}$

For the third link

- $\theta_3 = 0, d_3 = d_3$ (prismático), $a_3 = 0, \alpha_3 = 0$

$$T_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step 4: T_3^4

For the fourth link:

- $\theta_4 = 0, d_4 = 0, a_4 = 15, \alpha_4 = 0$

$$T_3^4 = \begin{bmatrix} 1 & 0 & 0 & 15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Final Transformation

The total transformation T_0^4 is obtained by multiplying the matrices:

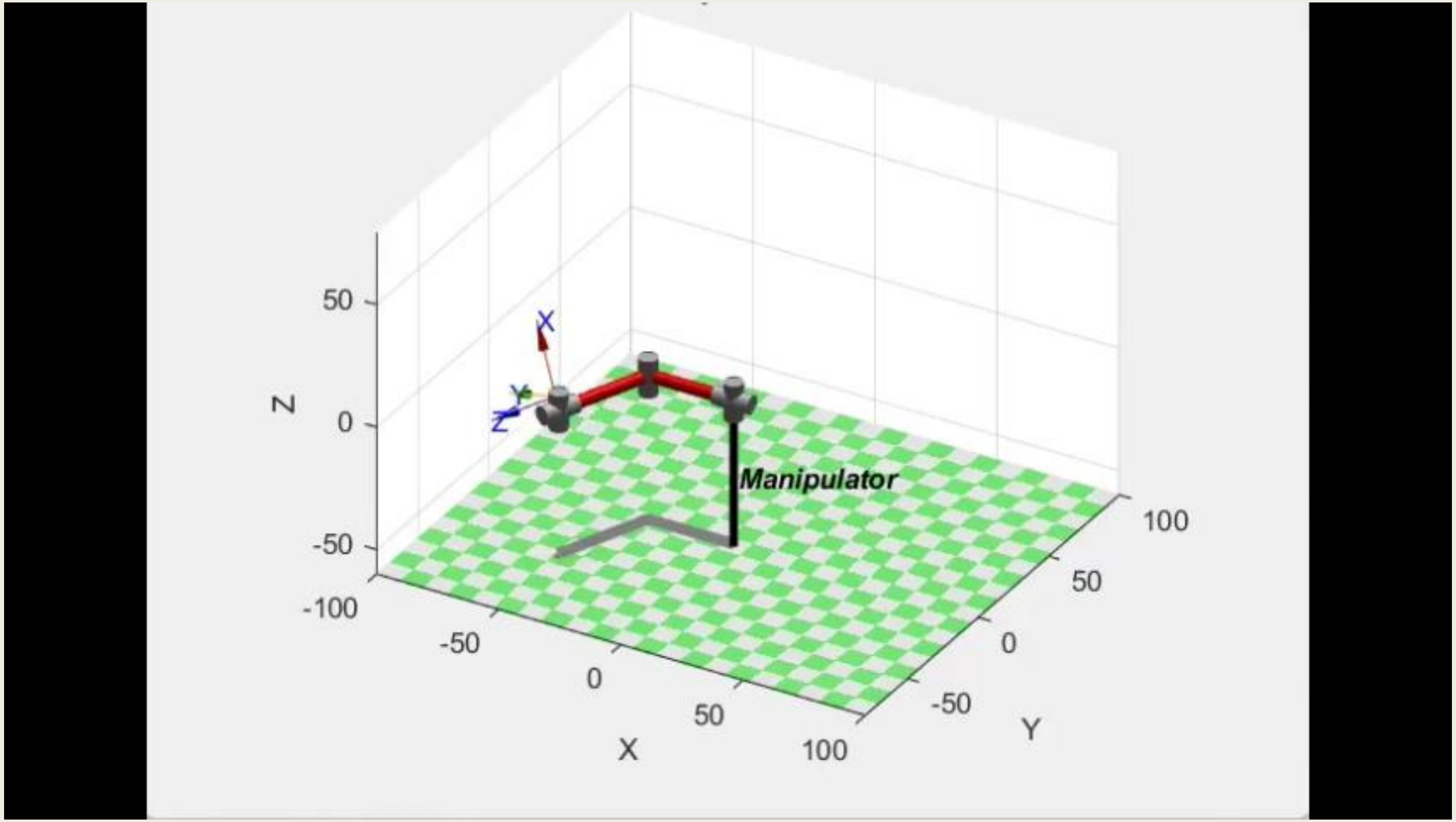
$$T_0^4 = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4$$

The total homogeneous transformation matrix, considering $d3$=0

$$T_0^4 = \begin{bmatrix} 1 & 0 & 0 & 90 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This means that the final position is **at x = 90 mm, y = 0, z = 0** in space, relative to the base coordinate system.

Simulation of BATT-BOT



ectlunes2.m | fisbote1.m

```
clc;
clear;

% Define link lengths
link1 = 0;   %no
link2 = 35;  %si
link3 = 40;  %si
link4 = 0;   %no
link5 = 0;   %si
link6 = 0;  %no
link7 = 5;  %si

% Define robot links using standard DH parameters: [θ, d, a, α, joint type]
L(1) = Link([0, 0, link1, -pi/2, 0]);
L(2) = Link([0, 0, link2, 0, 0]);
L(3) = Link([0, 0, link3, 0, 0]);
L(4) = Link([0, 0, link4, -pi/2, 0]);
L(5) = Link([0, 0, link5, 0, 1]);       % prismatic
L(6) = Link([0, 0, link6, 0, 0]);
L(7) = Link([0, 0, link7, 0, 0]);

% Define the robot model
robot = SerialLink(L, 'name', 'Manipulator');

% Define base rotation angles
thetaX = pi/2;  % base rotation about X-axis
thetaY = 0;
thetaZ = -pi/2;  % base rotation about Z-axis

% Define the combined transformation (rotate about Z then X)
Rx = trotx(thetaX); % Rotation about X-axis
Ry = troty(thetaY);
Rz = trotz(thetaZ); % Rotation about Z-axis
T_base = Rz * Rx * Ry;  % Combined transformation matrix
```

```

    % Interpolate each joint position over time
    q_interp = zeros(num_steps, size(motions, 2)); % Preallocate for efficiency
    for j = 1:size(motions, 2)
        q_interp(:, j) = linspace(q_start(j), q_end(j), num_steps); % Interpolate each joint
    end

    motions_interpolated = [motions_interpolated; q_interp]; % Append interpolated motions
end

% Visualize the interpolated motion
figure;
robot.plot(q_initial, 'workspace', workspace, 'linkcolor', [1, 0, 0], ...
         'basecolor', [0, 0, 0], 'jointcolor', [0.75, 0.75, 0.75]);
title('Robot manipulator simulation');
hold on;

for i = 1:size(motions_interpolated, 1)
    q = motions_interpolated(i, :); % Get interpolated joint configuration
    robot.plot(q, plot_options{:}); % Visualize the configuration
    pause(0.01); % Adjust for smoother animation
end
```

2. Dynamics

The dynamics of the robot describe how forces and torques are generated in response to desired motion, considering the robot's physical properties. These dynamics are typically represented using the **Euler-Lagrange equation:**

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)$$

Where:

- τ: Vector of joint torques/forces.
- $\mathbf{M}(\mathbf{q})$: Inertia matrix.
- $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$: Coriolis and centrifugal forces.
- $\mathbf{G}(\mathbf{q})$: Gravitational forces.
- $\mathbf{q}$: Vector of joint variables (angles for rotational joints, displacements for prismatic joints).
- $\dot{\mathbf{q}}$: Joint velocities.
- $\ddot{\mathbf{q}}$: Joint accelerations.

2.1. Inertia Matrix M(q)

The inertia matrix represents how the masses of the links affect the robot's movement:

$$M(q) = \begin{bmatrix} I_1 + I_2 + I_3 + I_4 & 0 & 0 & 0 \\ 0 & I_2 + I_3 + I_4 & 0 & 0 \\ 0 & 0 & I_3 + I_4 & 0 \\ 0 & 0 & 0 & I_4 \end{bmatrix}$$

Where:

- $I_i = \frac{1}{3} m_i L_i^2$: Moment of inertia of link i.
- m_i: Mass of link i.
- L_i: Length of link i.

2.2. Coriolis Matrix C ($q, \dot{q}$)

The Coriolis matrix contains terms that account for interactions between joint velocities:

$$C(q,\dot{q}) = \begin{bmatrix} 0 & -c_{12}\dot{q}_2 & -c_{13}\dot{q}_3 & -c_{14}\dot{q}_4 \\ c_{12}\dot{q}_1 & 0 & -c_{23}\dot{q}_3 & -c_{24}\dot{q}_4 \\ c_{13}\dot{q}_1 & c_{23}\dot{q}_2 & 0 & -c_{34}\dot{q}_4 \\ c_{14}\dot{q}_1 & c_{24}\dot{q}_2 & c_{34}\dot{q}_3 & 0 \end{bmatrix}$$

The coefficients $\mathbf{c_{ij}}$ depend on the robot's configuration and physical properties.

2.3. Gravitational Terms G(q)

The gravitational vector represents the torques needed to counteract gravity:

$$G(q) = \begin{bmatrix} m_1 g \frac{L1}{2} \\ m_2 g \frac{L2}{2} \\ m_3 g \frac{L3}{2} + m_{\text{load}} g L_3 \\ m_4 g \frac{L4}{2} + m_{\text{load}} g L_4 \end{bmatrix}$$

```
>> Roboticslinks4R1
Gravitational Terms G(q):
    2.5751
    2.3544
    1.3243
    0.7358

>> RoboticslinksR1
Inertia Matrix M(q):
    0.1329         0         0         0
         0    0.0717         0         0
         0         0    0.0077         0
         0         0         0    0.0017

Coriolis Matrix C(q, q_dot):
[      0, -q2_dot, -q3_dot, -q4_dot]
[q1_dot,        0, -q3_dot, -q4_dot]
[q1_dot,  q2_dot,        0, -q4_dot]
[q1_dot,  q2_dot,  q3_dot,        0]
```

These dynamic equations form the foundation for calculating the forces and torques needed to move the robot accurately and efficiently

Summary of the Mathematical Model

1. **Forward Kinematics:**

$$T_0^4 = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4$$

- The end-effector's position is expressed as (x, y, z) in terms of the joint angles.

1. **Dynamic Equation:**

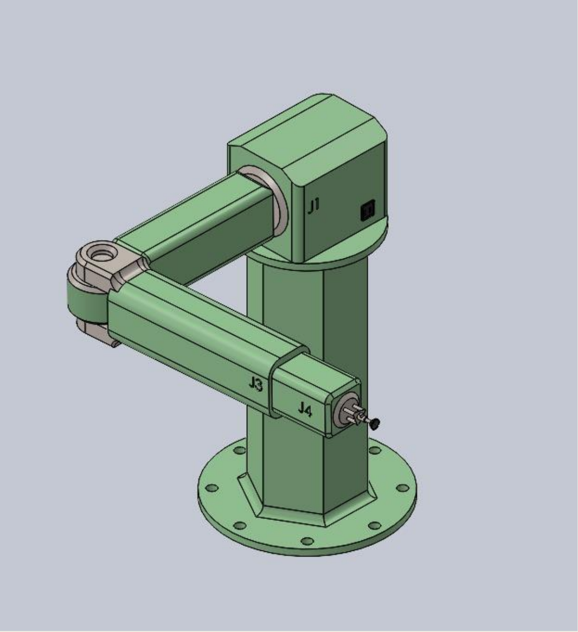
$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)$$

Where:

- M(q) Inertia matriz.
- C (q, $\dot{q}$) Coriolis terms.
- G(q): Gravitational terms

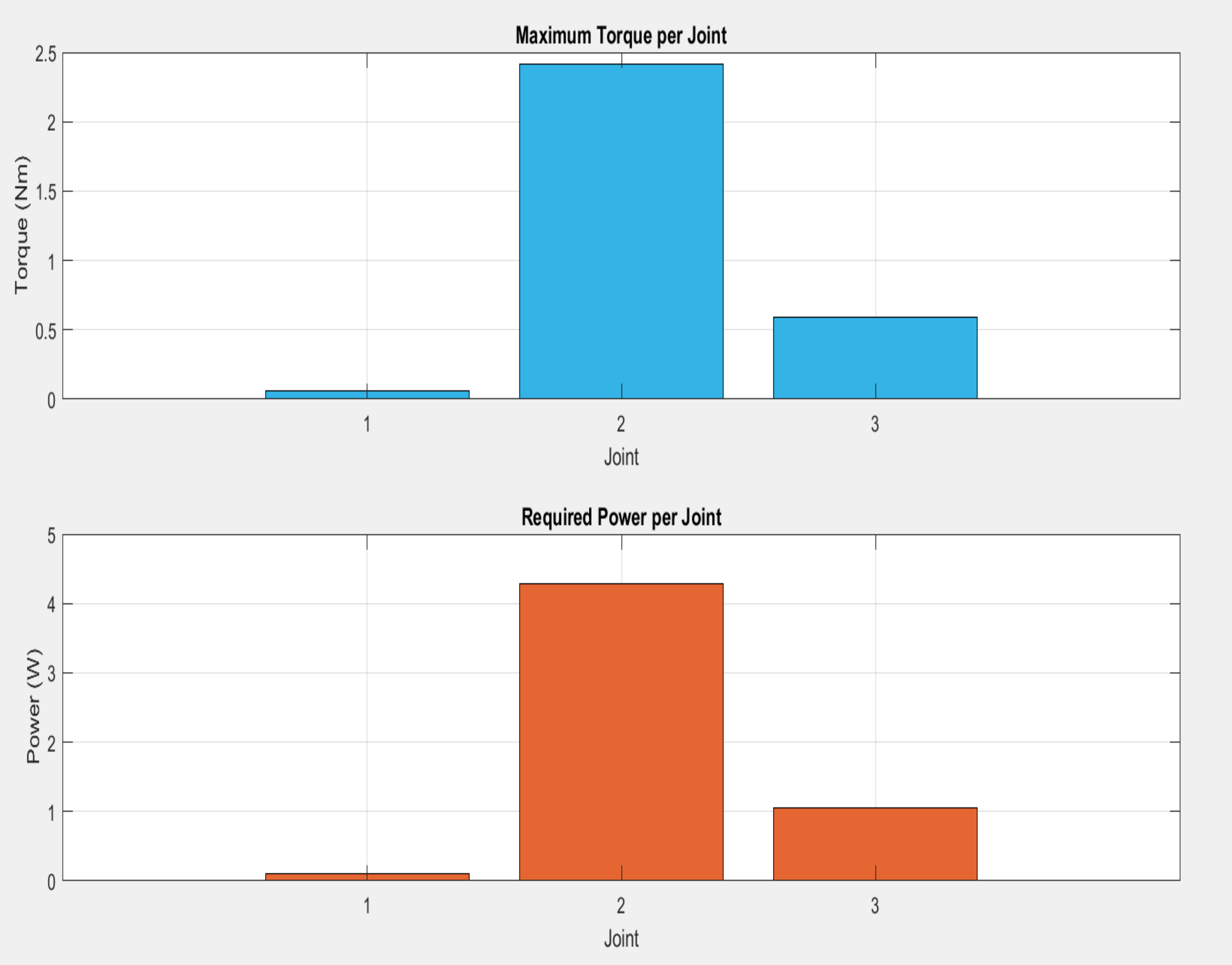
Robot Features

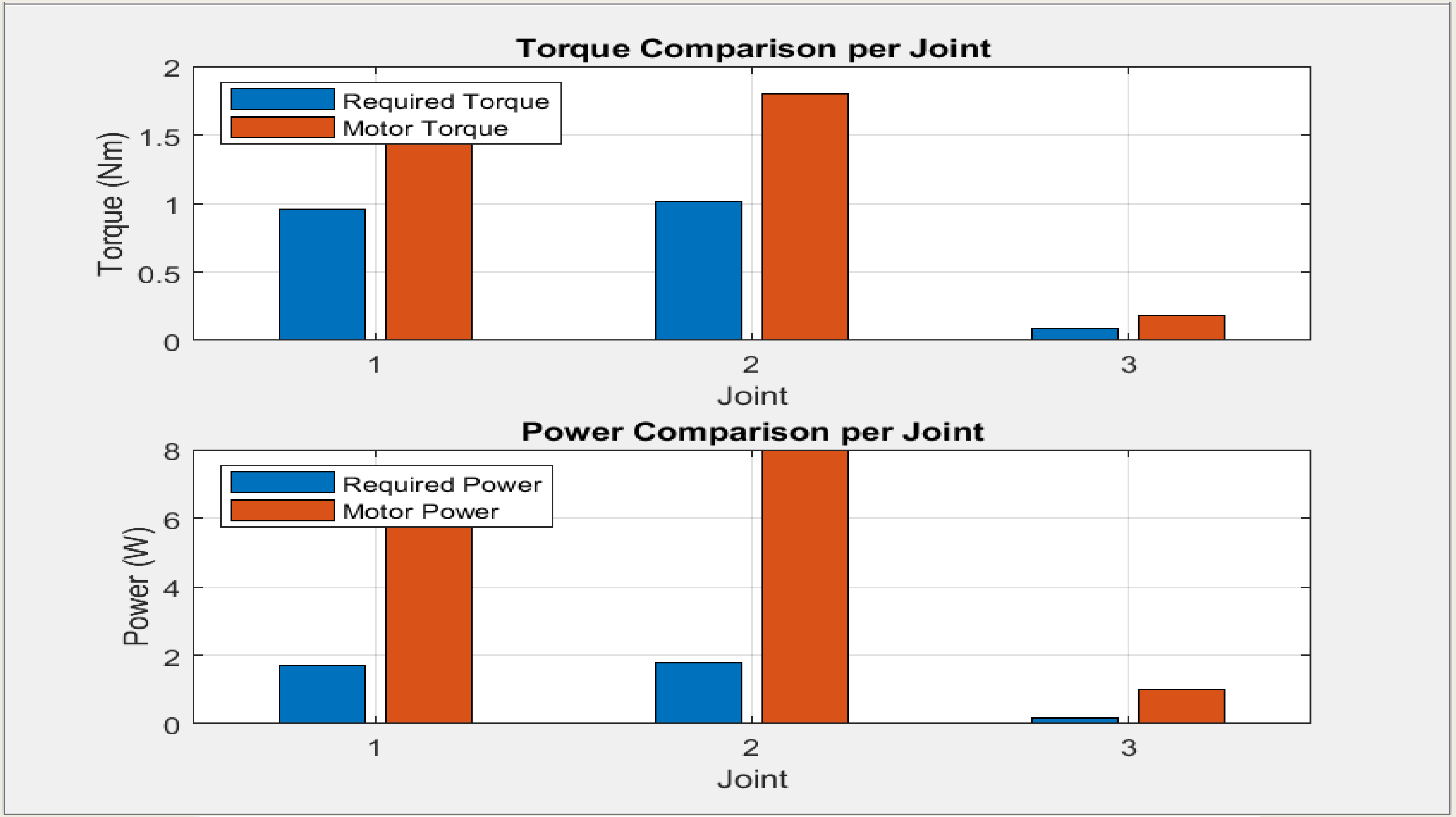
Parameter	Value
Length of Link 1	L_1=35cm (0.35 m)
Length of Link 2	L_2=40cm (0.40 m)
Length of Link 3	L_3=15cm (0.15 m)
Mass of Robot Body	m=1.5kg (link 1)
	m=1.2kg (link 2)
	m=0.8kg (link 3)
Mass of the Load	m_{load}=0.5kg
Maximum Angular Acceleration	α=1rad/s
Maximum Angular Velocity	ω=1.77rad/s
Gravitational Force	g=9.81m/s2
Maximum Torque	Calculated for each joint
Maximum Required Power	Calculated for each joint



Calculate Torque and Power for each joint

Specifications of motors for each joint:
Joint 1:
  Maximum Torque: 0.06 Nm
  Required Power: 0.11 W
Joint 2:
  Maximum Torque: 2.42 Nm
  Required Power: 4.28 W
Joint 3:
  Maximum Torque: 0.59 Nm
  Required Power: 1.05 W





Motor Specifications and Comparison:

Joint	Required_Torque_Nm	Motor_Torque_Nm	Required_Power_W	Motor_Power_W
1	0.96	1.6	1.7	6
2	1.01	1.8	1.78	8
3	0.09	0.18	0.17	1

Motor Specifications

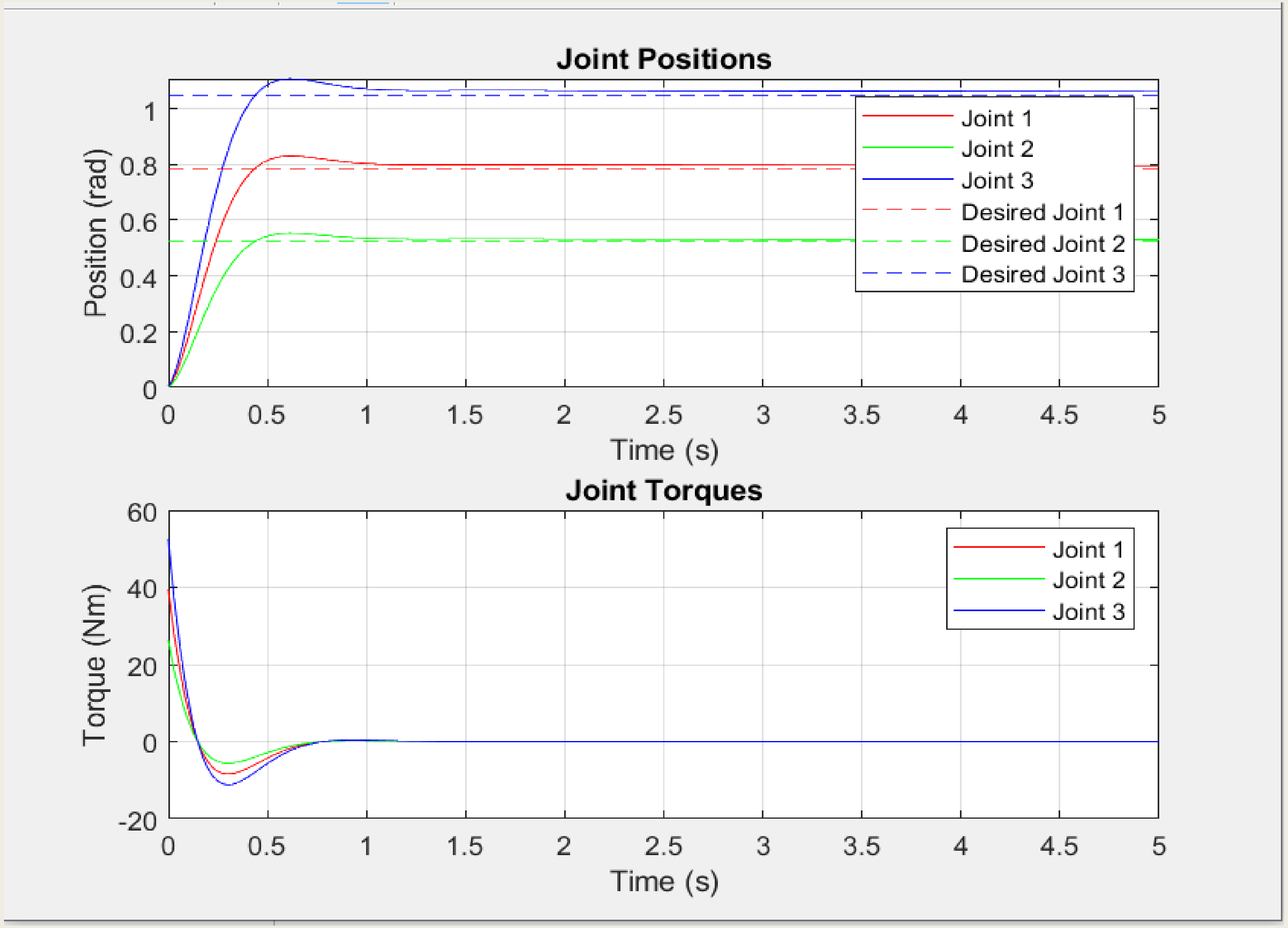
JOINT 1	JOINT 2	JOINT 3
Required torque: τ_1 = 0.96 Nm Required power: P_1 = 1.70 W Recommendation: **Motor:** Pololu 37D DC Gearmotor 19:1 **Continuous torque:** 1.6 Nm (sufficient for safety factor). **No-load speed:** 320 rpm (33.5 rad/s, easily adjustable with PWM control). **Power:** 6 W	Required torque: τ_2=1.01 Nm Required power: P_2=1.78 W Recommendation: **Motor:** NEMA 17 Stepper Motor (with 5:1 gearbox) **Continuous torque:** 1.8 Nm (sufficient with margin). **Speed:** Supports 1.77 rad/s when using a suitable controller. **Power:** 8 W (more than enough).	Torque required: τ_3 = 0.09 Power required: P_3 = 0.17 W Recommendation: **Motor:** MG90S Servo Motor **Continuous torque:** 0.18 Nm (twice the required torque). **Speed:** 0.1 s/60° (equivalent to 10 rad/s). **Power:** 1 W (well above the requirement).

JOINT 1

JOINT 2

JOINT 3

PID controller



The high initial torques compensate for the large initial error, while the system response is smoothed by the derivative action.

These graphs validate the PID controller as an effective solution for position control in robotic manipulators.

Model of BATT-BOT SERIES (S M L)

1. BATT-BOT S (Small) Application: Ideal for ultra-compact assembly tasks, focusing on small and lightweight components. Specifications: **Link Dimensions:** L1: 25 cm (0.25 m) L2: 30 cm (0.30 m) L3: 10 cm (0.10 m) **Horizontal Reach:** 65 cm (0.65 m). **Maximum Payload:** 0.3 kg. **Accuracy (Repeatability):** ±0.05 mm. **Maximum Angular Speed (ω):** 1.5 rad/s. **Total Robot Weight:** 4 kg. **Power Required:** <80 W. **Applications:** Assembly of small batteries, connectors, and lightweight electronic components.	2. BATT-BOT M (Medium) Application: Designed for standard production lines, handling assembly tasks with extended reach and medium payloads. Specifications: **Link Dimensions:** L1: 35 cm (0.35 m) L2: 40 cm (0.40 m) L3: 15 cm (0.15 m) **Horizontal Reach:** 90 cm (0.90 m). **Maximum Payload:** 0.5 kg. **Accuracy (Repeatability):** ±0.03 mm. **Maximum Angular Speed (ω):** 1.77 rad/s. **Total Robot Weight:** 6.5 kg. **Power Required:** <100 W. **Applications:** Standard battery assembly, handling cells and small electronic device modules.	3. BATT-BOT L (Large) Application: Optimized for assembly on larger production lines, with greater reach and payload capacity. Specifications: Link Dimensions: L1: 50 cm (0.50 m) L2: 60 cm (0.60 m) L3: 20 cm (0.20 m) **Horizontal Reach:** 130 cm (1.30 m). **Maximum Payload:** 1.0 kg. **Accuracy (Repeatability):** ±0.02 mm. **Maximum Angular Speed (ω):** 2.0 rad/s. **Total Robot Weight:** 10 kg. **Power Required:** <150 W. **Applications:** Assembly of large modules, assembly on more demanding lines, handling of heavier cells.

Illustrative photo

Comparison in Table:

Model	Horizontal Reach (m)	Maximum load (kg)	Precisions (mm)	Total Weigh (kg)	Potency (W)
BATT-BOT S	0.65	0.3	±0.05	4	<80
BATT-BOT M	0.90	0.5	±0.03	6.5	<100
BATT-BOT L	1.30	1.0	±0.02	10	<150

Advantages of Having Three Models:

1. Flexibility: They adapt to different production needs.

2. Scalability: They allow companies to grow without changing technology.

3. Cost-Efficiency: Each model is optimized for its specific use, saving unnecessary costs.

Gracias por su atención.