Mathematical Modeling of a Three Axis Robotic Arm

Aleyna ÖZDOĞAN   Umut ÖZEN
Supervisor: Prof. Dr. Başak KARPUZ

Abstract

This project mathematically models and simulates the spatial motion of a three-degree-of-freedom robotic arm using Python. Joint coordinates are calculated with link lengths (L_1, L_2) and angles (θ, φ_1, φ_2) through trigonometric expressions. The geometric and trigonometric conditions for the arm's endpoint to reach a target surface (e.g., a whiteboard) are analyzed, including valid joint angle ranges. Vector analysis, trigonometric transformations, and the law of cosines are applied, and the results are visualized and tested in Python. The project offers a theoretical and practical approach to solving inverse kinematics problems in robotics.

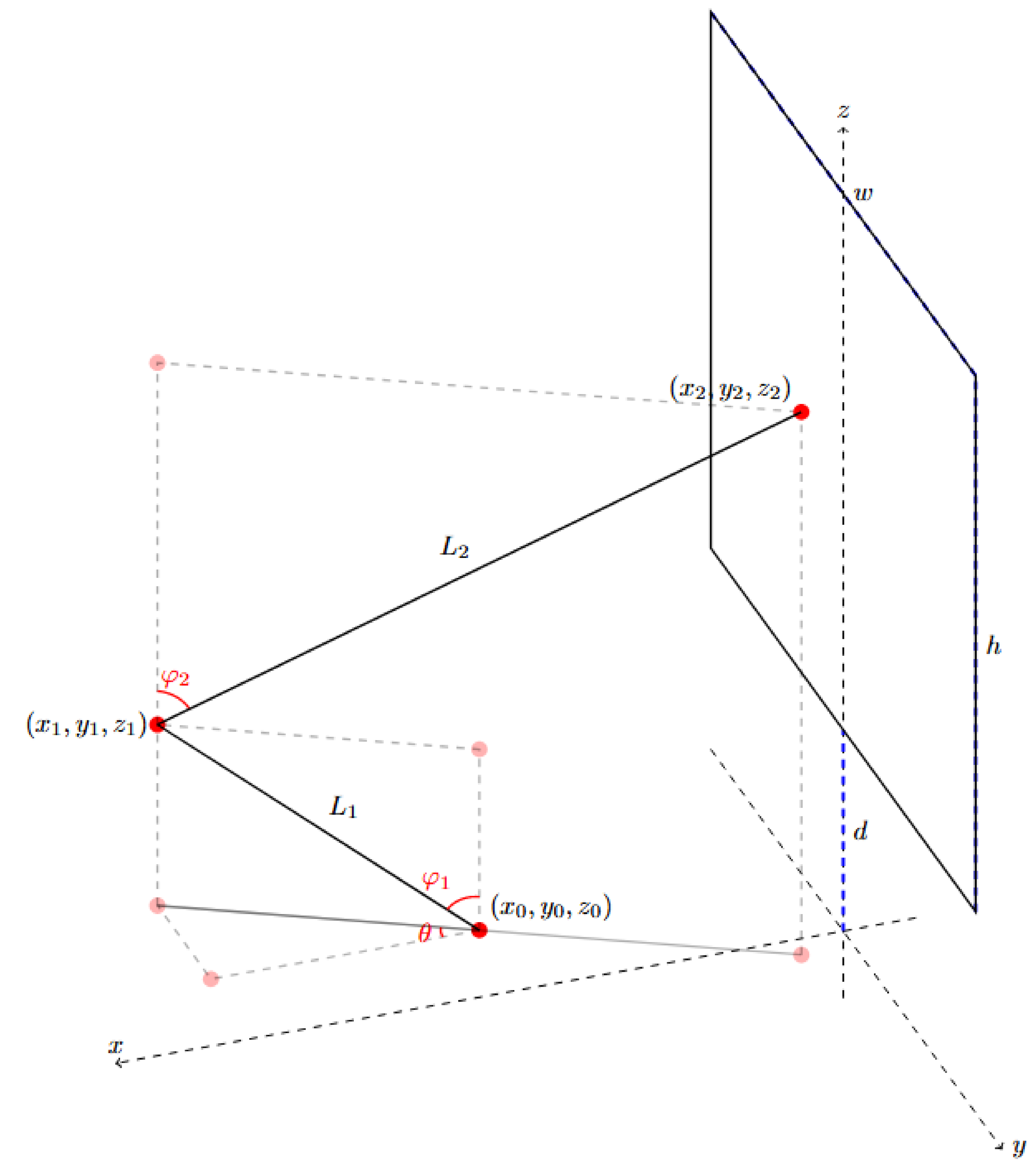
Applications of Three Axis Robotic Arm

- **Assembly lines:** 3D positioning and precise assembly of components.
- **Welding and painting applications:** Performing tasks at complex angles.
- **Surgical robots:** Medical operations requiring precise, multi-axis movements.
- **3D printing and CNC machining:** Executing complex toolpaths in manufacturing.
- **Research and education:** Used in advanced robotics control and kinematic studies.

Problem Statement

The precise control and spatial positioning of robotic arms with multiple degrees of freedom present significant challenges in robotics. Modeling the kinematics of a three-axis robotic arm mathematically is essential for understanding its movement, enabling accurate targeting and manipulation within a 3D workspace. This project aims to develop a comprehensive mathematical model and simulation framework to solve the forward and inverse kinematics problems for such robotic arms, ensuring efficient and reliable operation in real-world applications.

Representation of Robot Arm Angles and Coordinates



Let (x_0, y_0, z_0) denote the base point of the arm, and let L_1 and L_2 be the lengths of the links. Further, let φ_i denote the angle between the y-axis and the link L_i $(i = 1, 2)$, and let θ denote the angle between the z-axis and the xz-projection of L_1.

Joint Angle and Position Calculations

These equations define the forward kinematics of a 3-DOF robotic arm, expressing the end-effector position (x_2, y_2, z_2) relative to the base (x_0, y_0, z_0) as a function of link lengths L_1, L_2, joint angles φ_1, φ_2, and base rotation angle θ.

$$\begin{cases} x_2 = x_0 + \left[L_1 \sin(\varphi_1) - L_2 \sin(\varphi_2)\right] \cos(\theta) \\ y_2 = y_0 - \left[L_1 \sin(\varphi_1) - L_2 \sin(\varphi_2)\right] \sin(\theta) \\ z_2 = z_0 + L_1 \cos(\varphi_1) + L_2 \cos(\varphi_2). \end{cases}$$

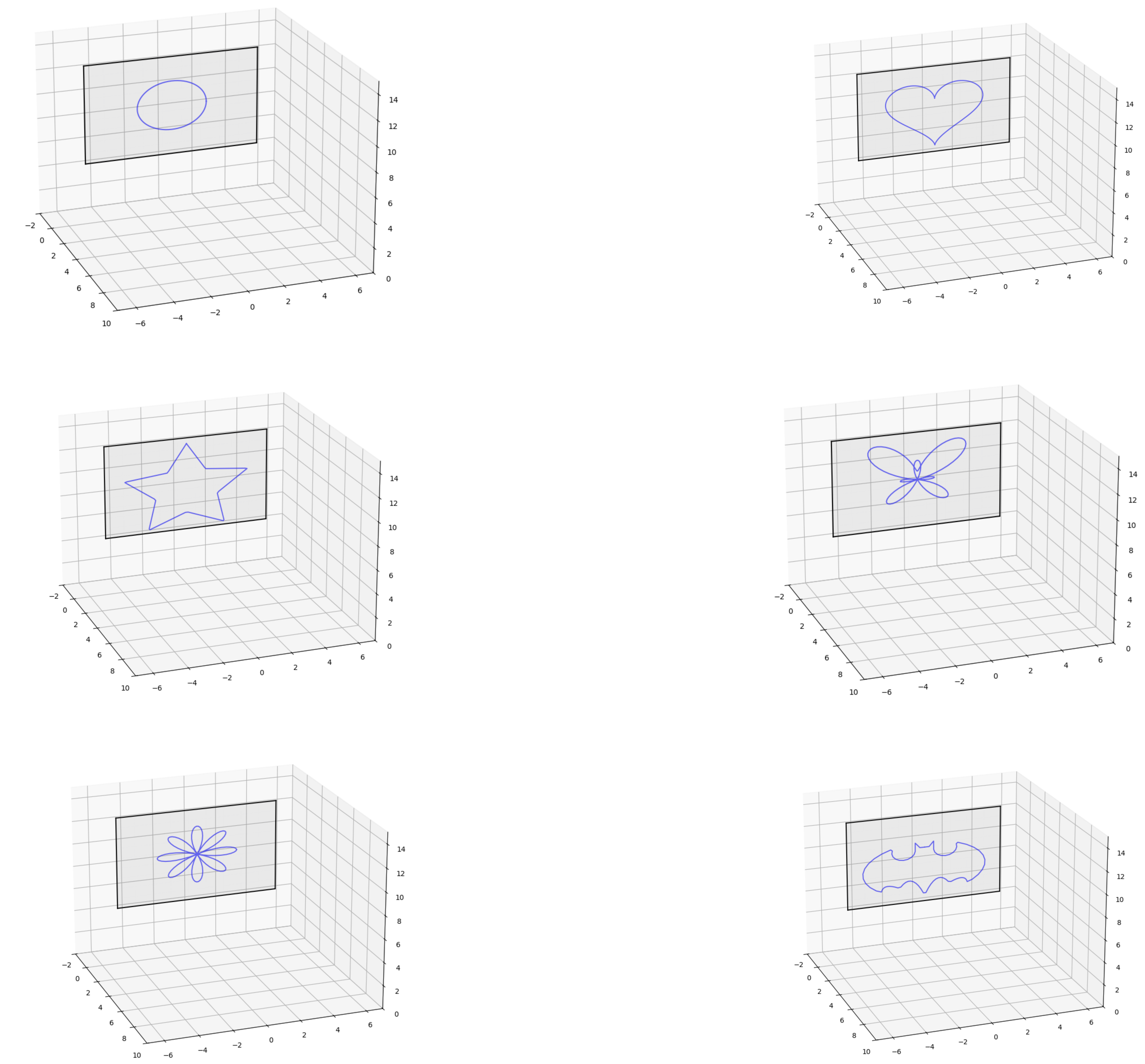
Determine the range of joint angles θ, φ_1, and φ_2 such that the end-effector position (x_2, y_2, z_2) lies on the whiteboard.

$$\theta = -\arctan\left(\frac{y_2 - y_0}{x_2 - x_0}\right)$$

$$\varphi_1 = \arctan\left(\frac{x_1 - x_0}{z_1 - z_0} \sec(\theta)\right)$$

$$\varphi_2 = \arctan\left(\frac{y_2 - y_1}{z_2 - z_1} \csc(\theta)\right)$$

Python-Based Simulation and Graphical Representation of a Two-Link Robot



Scan to See the Robot in Action

Watch this QR code to see a video demonstration of our two-link robotic arm simulation. The video illustrates the arm's motion and its interaction with the target workspace, providing a dynamic visualization of the mathematical model.

References

Tsai, L.-W. (1999). Robot analysis: The mechanics of serial and parallel manipulators. John Wiley Sons,
Stewart, J. (2016). Calculus: Early transcendentals (8th ed.). Cengage Learning.