SCARA DRAWBOT

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

ADVANCED MECHATRONICS

FINAL PROJECT

DATE: 05/08/2025

ETHAN NIV	N13878111
THEO DUFFAUT	N18406718
GHANASHYAMA PRABHU	N15244467

NYU

CONTENTS

NYU

INTRODUCTION

Human-robot interaction has evolved significantly with advancements in real-time control, machine vision, and embedded systems. This project focuses on the development of a SCARA (Selective Compliance Articulated Robot Arm) robotic system capable of replicating drawings based on real-time remote user input. The goal is to demonstrate how robotic arms can mimic human motion precisely using vision-based tracking and inverse kinematics.

The system employs a Raspberry Pi camera module to track an ArUco marker attached to a stylus or pen held by the user on a designated drawing surface. Using ROS (Robot Operating System) and computer vision algorithms, the Raspberry Pi processes image frames to determine the marker’s real-time position and orientation. This pose is interpreted as a target trajectory, which is then translated into joint commands for the SCARA robot via inverse kinematics.

The calculated coordinates are communicated to an Arduino-based controller that drives the SCARA arms servo motors, allowing the robots end-effector to follow the user’s pen motion. The system enables a real-time drawing experience where the robot mirrors human input with high fidelity, thereby showcasing the integration of computer vision, embedded control, and robotic manipulation in an interactive application.

This project not only serves as a proof of concept for human-guided robotic drawing but also offers a framework for more advanced applications in remote teaching, teleoperation, and collaborative robotics.

PROBLEM STATEMENT

In many remote collaboration and educational scenarios, there is a need for intuitive, real-time interaction between human users and robotic systems. Traditional robotic drawing systems lack adaptability to uneven surfaces, suffer from limited user input mechanisms, and often require pre-programmed trajectories, reducing flexibility and interactivity. The challenge lies in developing a robotic system that can dynamically follow a human-drawn path in real-time with minimal setup, while providing precise motion control.

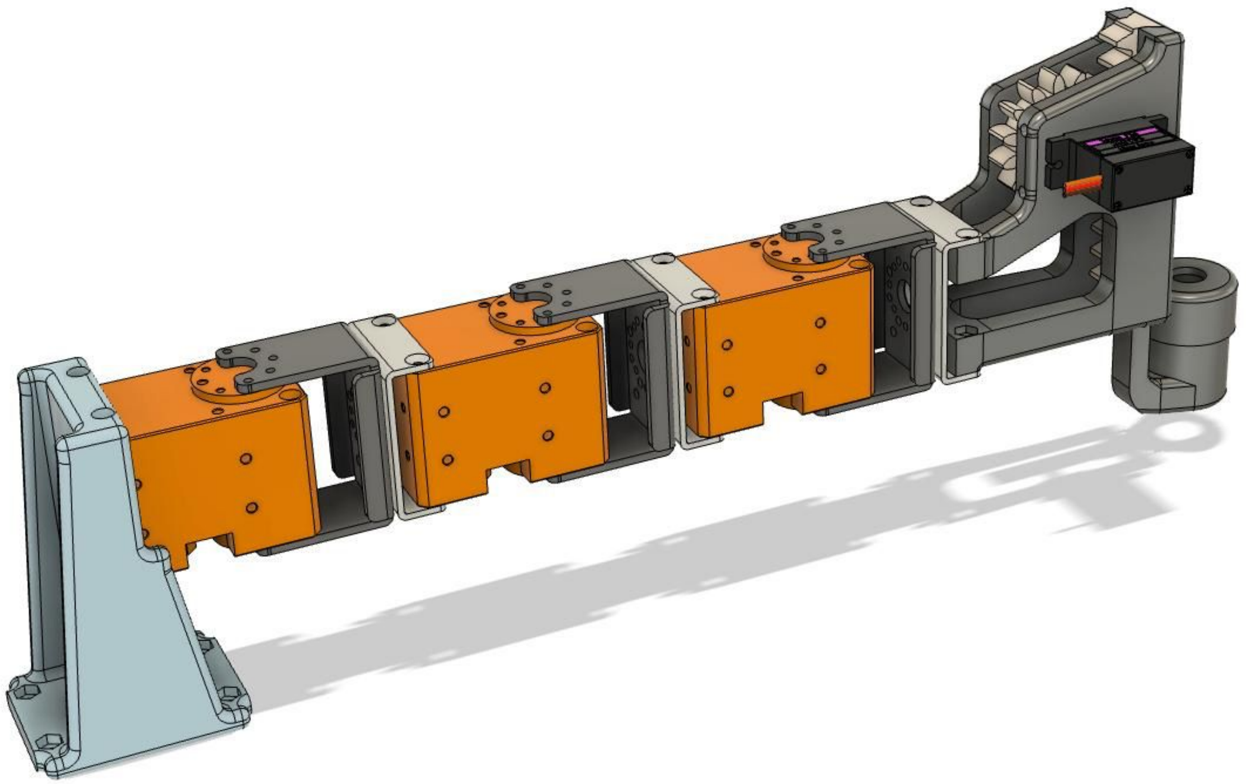
NYU

SOLUTION

The proposed system leverages a Raspberry Pi camera to track the position and orientation of a pen equipped with an ArUco marker. Through computer vision algorithms, the camera captures the marker's real-time spatial data, which is processed on the Raspberry Pi. This positional information is then used to compute the inverse kinematics of a SCARA robotic arm, translating the user's pen movements into corresponding joint angles. These calculated angles are communicated to an Arduino-based motor controller, enabling the robotic end-effector to closely follow the pen's trajectory. The system achieves seamless and intuitive human-robot interaction by combining real-time vision tracking with precise motion control, eliminating the need for pre-programmed paths and allowing adaptability to various surfaces and user-defined drawing spaces.

MECHANICAL DESIGN

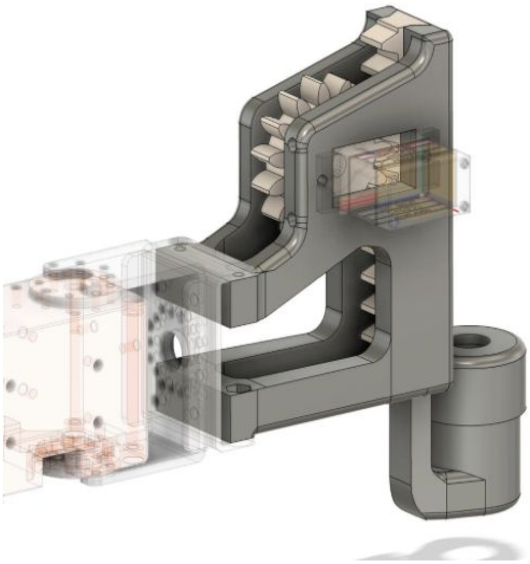
The Mechanical design of the SCRA Drawbot system emphasizes modularity, adaptability, and precision, enabling seamless interaction between human input and robotic response. The robotic arm is based on the open manipulator SCARA configuration, selected for its horizontal reach and mechanical simplicity, making it suitable for 2D planar drawing tasks.

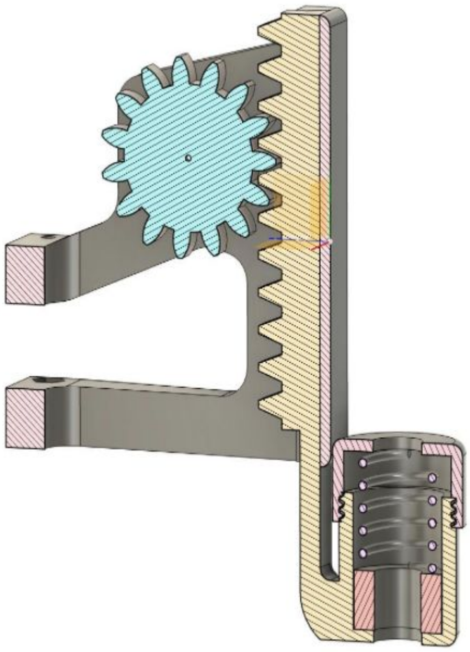


SCARA Manipulator

Custom End-effector

The core structure of the robot utilizes the open manipulator SCARA model, equipped with Dynamixel XL430 servo motors to control joint motion. A custom-designed end -effector is integrated at the arm's terminus, housing a pen mechanism that features a spring-loaded vertical compliance system. This allows the pen tip to automatically adjust, ensuring consistent contact with the drawing surface. The compliance system not only provides smooth writing motion on flat surface like an ipad but also enables adaptability on slightly uneven or textured surfaces.





Custom End-effector

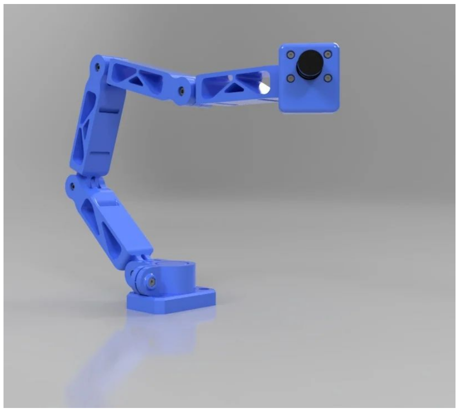
Pen Holder with ArUco marker

A custom pen holder was designed using lightweight materials and 3D printing, incorporating surfaces for ArUco marker. The marker acts as a positional reference for Raspberry Pi camera and defines the trajectory that the robot must follow. The design ensures stable alignment of the pen and marker while allowing the user to draw freely with a predefined workspace. The marker’s visibility from multiple angles was optimized to maintain robust tracking performance even during fast or non-linear pen movements.

Pen Holder

Camera Mount and Flex Arm

The off-the-shelf camera mount is flexible, articulated arm that allows for dynamic height and angle adjustment. This flexibility ensures that the camera can be positioned to cover a wide variety of workspace sizes and orientations. The camera holder's design allows users to reposition or recalibrate the system as needed without mechanical disassembly.

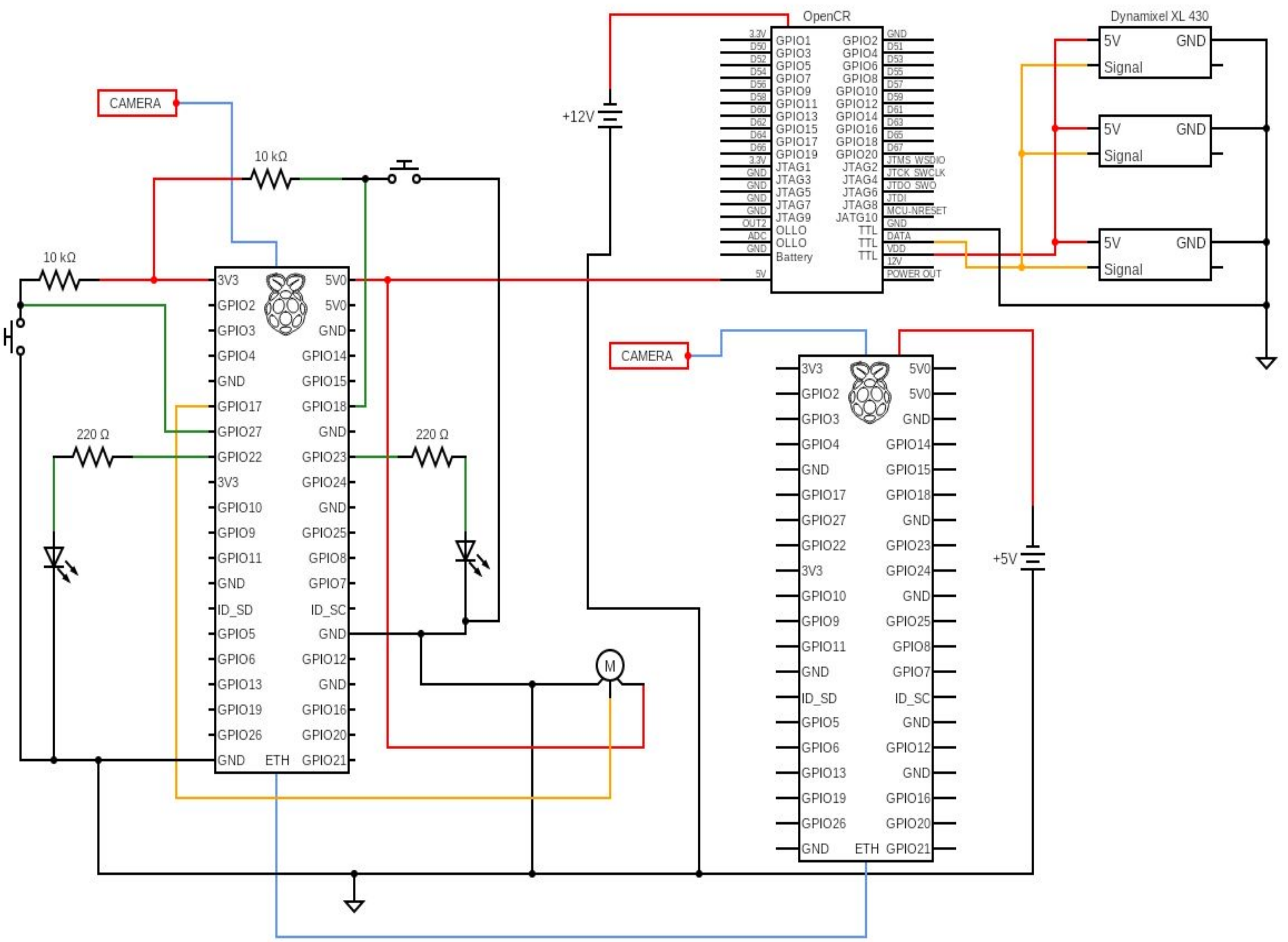


Camera Mount

Mounting Frame and DrawSurface

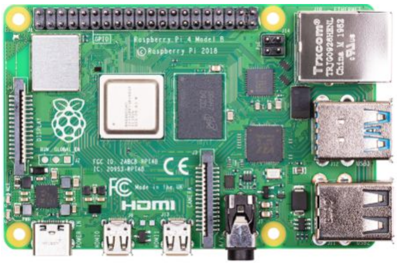
The entire system is supported on rigid base frame that stabilizes the SCARA arm and isolates vibrations. The drawing surface(ipad) is fixed on the wooden platform, ensuring it is stable while drawing.

ELECTRICAL DESIGN



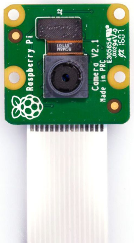
Circuit Diagram

1. <u>Raspberry Pi 3 :</u> The Raspberry Pi 3 model B is compact and versatile single-board. It features a 1.2GHz quad-core ARM cortex-A53 processor, 1GB RAM, built-in Wi-Fi, Bluetooth 4.1 and four USB ports, making it suitable for a wide range of computing and various applications. The board also includes a 40-pin GPIO header, which allows interfacing with various sensors, motors, and other hardware modules making ideal for robotics projects.



Raspberry Pi Module

2. <u>Pi Camera:</u> It is a high-quality image and video capture device designed specifically for the Raspberry Pi. It connects via the Camera serial interface, ensuring high data throughput and low latency. The camera supports various functionalities such as exposure control, white balance, and image effects, making it suitable for computer vision, surveillance, and motion detection projects. When combined with OpenCV and Python, the camera module becomes a powerful tool for real-time image processing applications, enabling the development of facial recognition systems, object tracking algorithms, and machine learning-based vision solutions.



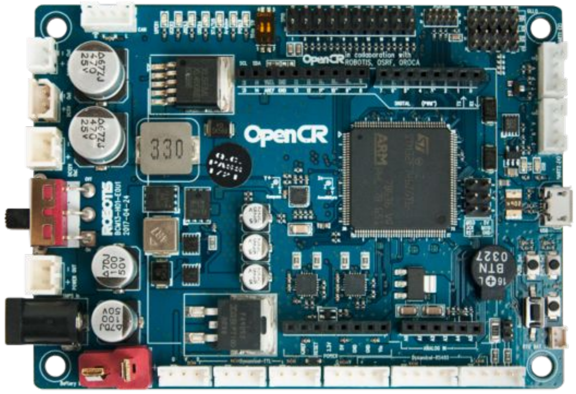
Raspberry Pi Camera

NYU

3. <u>Arduino Uno</u>: The Arduino Uno serves as the actuation controller for the robotic arm. It receives the desired end-effector positions from the raspberry Pi via serial or USB communication. The Uno interprets these positions and generates appropriate PWM signals or serial control commands to drive the motors accordingly.

Arduino Uno

4. <u>OpenCR Motor Controller Shield</u>: It is an advanced motor controller based board designed for seamless integration with dynamixel smart servos. It connects directly to the arduino or can be used as a standalone board for ROS-compatible robots. It is responsible for driving the dynamixel XL430 servos used in the SCARA arm, handling the motor feedback and providing real-time performance and communication robustness.



Open CR Motor Controller

3. Dynamixel XL430: High-performance smart actuators capable of precise position control, daisy-chained communication, and internal feedback. They drive the primary joints of the SCARA arm. Controlled using the OpenCR

Arduino Uno

4. Mg90s servo: A lightweight, analog servo motor used to control the pen-lifting mechanism for engaging/disengaging the drawing surface. It provides sufficient torque and speed for intermittent up-down motion.

MG90S servo

NYU

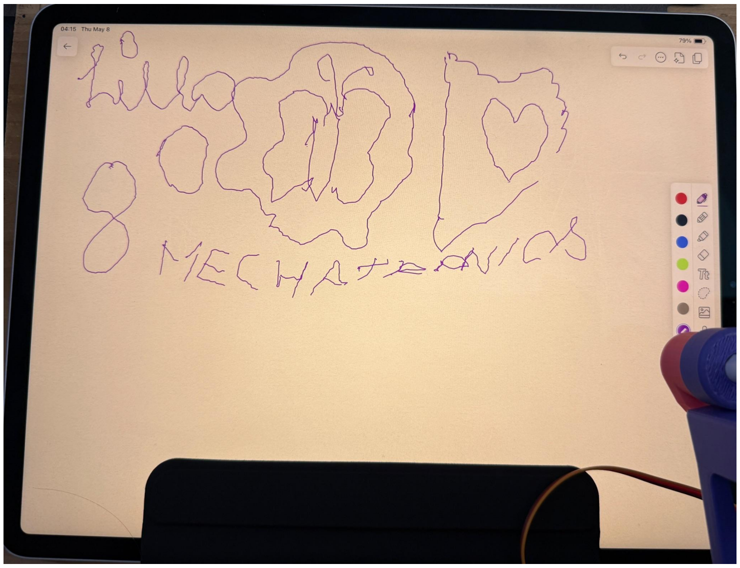
SOFTWARE DESIGN

ROS Noetic on Debian Raspberry Pi:

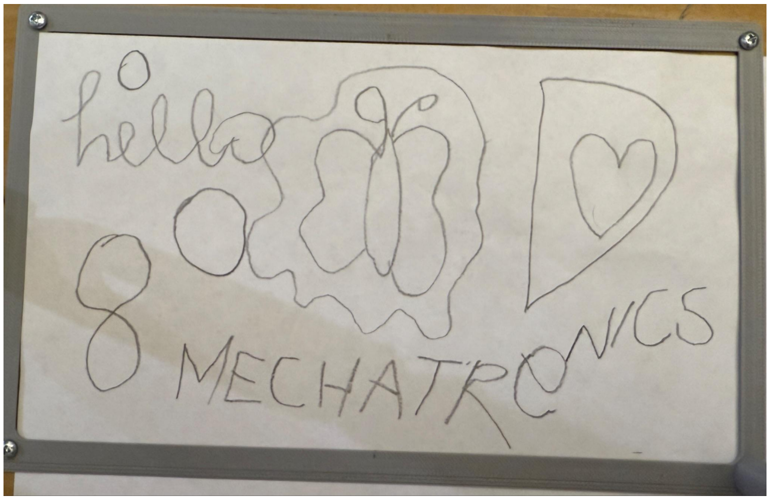
- Open Manipulator package
  ROS node responsible for managing and controlling the SCARA arm. It initializes the connection to the physical robot (via USB serial). The controller exposes a variety of ROS services that allow other ROS nodes to send commands to the arm. These commands include moving the arm to specified joint configurations or Cartesian poses (position and orientation, position only or orientation only), executing pre-defined drawing trajectories (like lines, circles, heart, etc), and controlling the end-effector/tool. It also publishes the robot's state, such as joint positions and kinematics information, via ROS topics, enabling other parts of a robotic system to monitor and react to the arm's status. Internally, it uses a timer-driven control loop to process commands and update the robot's state (100Hz control rate).
- External I/O Control Package
  Another node controlling a smaller servo (to control the pen up/down movements) as well as a input button and an LED for external user interface and feedback. Sets up services so that the end-effector can be easily actuated from any other ROS nodes.
- ArUco Marker Tracking Package
  Marker tracking node that subscribes to the images published by the Camera node and uses OpenCV to detect and estimate the pose of an ArUco marker.
- Pose Filtering
  Moving window average: Defining a window size L, and then averaging each field of the position and of the orientation independently across the past L measurements. X is chosen to be large enough to properly filter out noise, but at the same time small enough to precisely track changing poses.

RESULTS

The SCARA Drawbot system successfully demonstrated real-time replication of user-drawn trajectories. The integration of the Raspberry Pi camera and ArUco marker tracking provided an accurate detection of the user's pen position and orientation across the drawing surface. The inverse kinematics computations translated this tracked motion into precise joint commands for the robotic arm, resulting in smooth and responsive end-effector movements. The overall latency between user input and robotic response remained minimal, ensuring a natural and interactive drawing experience. These results validate the feasibility of using vision-guided robotic systems for human-in-the-loop applications such as remote sketching, collaborative design, and telepresence tasks.



Robot Workspace



User's Workspace

APPENDIX

Mathematical Formula:

The SCARA (Selective Compliance Articulated Robot Arm) configuration used in this project features three degrees of freedom (DOF), enabling planar motion along the X and Y axes as well as vertical movement along the Z axis. This setup is ideal for tasks such as drawing or pick-and-place operations on a horizontal surface, especially when minor variations in surface height must be accommodated.

The first two joints of the SCARA arm are rotational (revolute) and control the position of the end-effector in the horizontal XY plane. These joints are connected by rigid links of known lengths. The third degree of freedom is provided by a prismatic joint that allows the end-effector to move up and down in the Z-direction. This vertical motion is crucial for ensuring appropriate contact with the drawing surface and accommodating uneven surfaces or dynamic height adjustments.

The inverse kinematics (IK) solution calculates the necessary joint angles and displacement (θ_1, θ_2, d_3) to position the end-effector at a desired point (x, y, z) in space. Using trigonometric relationships derived from the robot's geometry, the IK equations first compute the angle of the elbow joint (θ_2) using the Law of Cosines. Then, the base joint angle (θ_1) is calculated based on the desired target coordinates and the geometry of the arm. Finally, the vertical displacement (d_3) is determined by comparing the desired Z-axis height with the home position of the end-effector.

This approach allows the robotic arm to track a moving marker in real-time by continuously recalculating the required joint positions, enabling precise and smooth motion that mimics the user’s input. The system's ability to solve inverse kinematics in real-time is essential for maintaining responsiveness and drawing accuracy in dynamic applications.

NYU

Inverse Kinematic equations:

Joint 1 (θ_1): Base rotation (horizontal plane)

Joint 2 (θ_2): Elbow rotation (horizontal plane)

Joint 3 (d_3): Linear movement (usually vertical, for the end-effector)

Link 1: Length = L_1 (between base and first joint)

Link 2: Length = L_2 (between first joint and end-effector base)

Joint 3: Controls the vertical position (z) of the end-effector (typically prismatic)

Step 1: Compute Theta2(θ2):

$$\cos(\theta_2) = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}$$

$$\sin(\theta_2) = \pm\sqrt{1 - \cos^2(\theta_2)}$$

$$\theta_2 = \mathrm{atan2}(\sin(\theta_2), \cos(\theta_2))$$

Step 2: Compute Theta1(θ1):

$$\theta_1 = \mathrm{atan2}(y, x) - \mathrm{atan2}(L_2\sin(\theta_2), L_1 + L_2\cos(\theta_2))$$

Step 3: Compute d3:

$$d_3 = z_{\text{base}} - z_{\text{target}}$$

θ_1 and θ_2 determine the position in the XY plane.

d_3 determines the vertical reach of the pen.