AI Drawing Arm

System Flow and Mathematical Documentation

Felipe Perassi
Lorenzo Minervino

August 13, 2025

Overview

AI Drawing Arm is a robotic system that interprets natural language prompts and produces symbolic drawings through a dual-arm mechanism controlled by inverse kinematics. This document presents the mathematical model and system flow of the project.

1 Kinematic Model

1.1 Forward Kinematics

Given joint angles θ_1 and θ_2, and segment lengths L_0, L_1, and L_2, the position of the end effector (x, y) is computed using geometric relationships.

Each arm (left and right) contributes to the final end-effector position:

$$(x, y) = \frac{(x_{\text{left}} + x_{\text{right}})}{2}, \quad \frac{(y_{\text{left}} + y_{\text{right}})}{2}$$

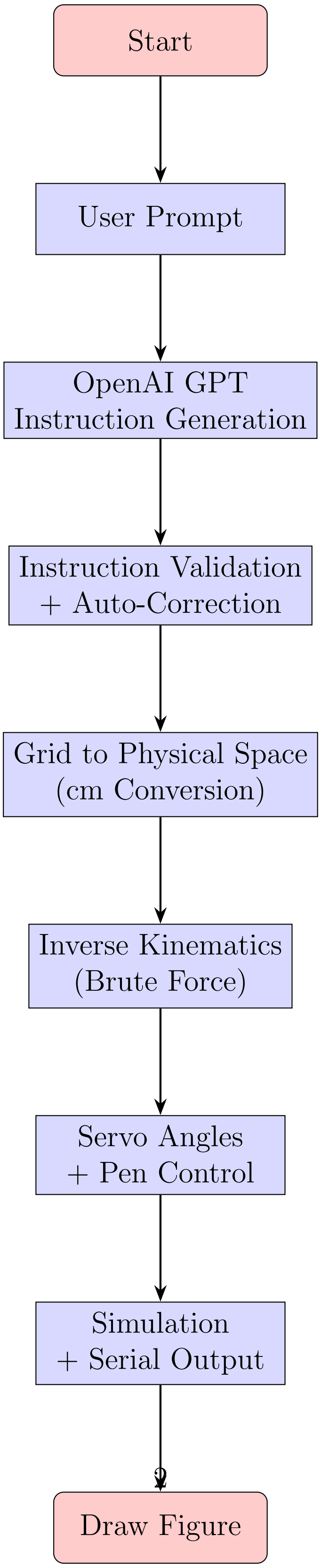
1.2 Inverse Kinematics (Brute Force)

For a target point (x^*, y^*), the goal is to find joint angles minimizing the error between actual and target position:

$$(\theta_1^*, \theta_2^*) = \arg\min_{\theta_1, \theta_2} \left[(x - x^*)^2 + (y - y^*)^2 \right]$$

This is solved via brute-force grid search over discrete angle steps.

2 System Flow Diagram


Start
User Prompt
OpenAI GPT Instruction Generation
Instruction Validation + Auto-Correction
Grid to Physical Space (cm Conversion)
Inverse Kinematics (Brute Force)
Servo Angles + Pen Control
Simulation + Serial Output
Draw Figure

3 Execution Pipeline

1. User provides a prompt like `"draw a cat"`.
2. GPT generates a list of absolute-position drawing commands:

   ```
   0 2 3
   1 4 3
   1 4 6
   0 7 5
   1 2 1
   ```

3. Commands are validated (e.g., grid bounds, format, duplicates).
4. The list is centered and scaled to match physical coordinates.
5. Each point is converted to servo angles via brute-force IK.
6. Angles and pen control signals are visualized and/or sent to the robot via serial.
7. The robot draws the final figure on a 14×10 grid (7×5 cm).

4 System Highlights

- Modular Python architecture: AI logic, validation, math, hardware separated.
- Fully autonomous: user prompt → physical drawing.
- Real-time simulation with `matplotlib`.
- Compatible with Arduino or ESP32 via serial communication.