Problem Statement

Friday, 28 November 2025 8:08 PM

Stage 1:

5 DoF Robotic Arm controlled using input via laptop (Serial Protocol) or using control using physical input devices such as potentiometer or joysticks.

Stage 2:

Upgrading the Arm made in Stage 1 to mimic the motion recorded using a similar (maybe smaller) structure and mimic the motion by the Arm.

Diagram for illustration:

This image is taken from an online GitHub repository. For structure we can use either cardboard built structure/ 3D Printing parts/ Laser cut Acrylic sheets. Other options can also be used.

Also the picture is of 6 DoF Robotic Arm, we wish to make only 5 DoF, removing the the Pick & Drop feature as of now.

Starting with Stage 1

- Step 1: We plan to use serial input via laptop to Arduino or any other microcontroller we use and control each motor individually.
- Step 2: Using a rotary encoder or potentiometer as the physical input for each motor and mapping these inputs directly to the corresponding motor outputs.
    - Another step or feature which can be added in this later is using Joysticks to control all the motor more easily as it would be easier for user to control the movement. (This does not comply with the Stage 2 objective but we haven't decided the complete track/objective of the project, therefore need guidance in this part.)

Points to Consider:

- Using rotary encoder requires double the pins than potentiometer for sensing the rotation and its direction, contrary to potentiometer by which we could determine the position by the output value of resistor.
    - For 5 potentiometers - 5 Analog Pins
    - For 5 rotary encoder - 10 Digital Pins
    - Using motors requires additional pins in addition to these

Stage 2:

- Instead of using serial commands from a laptop or separate potentiometers/rotary encoders as inputs, we aim to replicate the motion recorded from a similar reference structure.
- Step 1: Mapping and optimizing the inputs and outputs and using real time direct mapping to mimic motion.
    - Directly attaching the potentiometers in the structure and optimizing the values accordingly. (This step simply involves installing the sensory input item in slots and mapping them accordingly, instead of using separate input controls/ joystick)
- Step 2: Adding UI using push buttons and a display (optional), making a system in which we can record different positions of arm and save them as states and using FSM model to mimic the motion continuously.

Points to Consider:

- Potentiometer has poor accuracy in comparison to rotary encoders, this can be a problem when trying to mimic the motion.
    - Potential Solution: Using 2 potentiometers of different resistance and taking their average values to determine its position.
- Encoder Motor provides closed loop system and a possibility of applying P/PI/PID control system, whereas Servo Motor is an open loop system.

Reference Link to ideal objective: https://www.youtube.com/shorts/UqKgUrqy8Rk
In this video the robot mimics the motion once captured instead of pre-programmed motion.

Reference Link to Stage 1: https://www.youtube.com/watch?v=dzyKqRVN2kc
This video showcases a readymade market available kit assembly.

Technical Content Involved:

The project incorporates essential electronics and embedded systems concepts such as microcontroller-based firmware development, I^2C communication for display and sensors, PWM control for servo motors, encoder feedback with PID control for higher accuracy, efficient memory management for recording motion data, and an FSM-based architecture to manage system states. Together, these technical elements enable real-time control, smooth motion, accurate playback, and a robust embedded control system for a 5-DoF robotic arm.

Items Required:

Components:

- Microcontroller Board (Arduino Mega or anything which fulfills required no. of pins based on choices of other items.)
- Motor * 5
- Sensory input
    - 5 req. acc to 1 per DoF, but for Stage 2 if we plan to use potentiometer it may be 2 per DoF
- Push Button * 2 or 3
- Breadboards and other basic wiring items

Options for motor:

- Servo (Open Loop System)
- Stepper Motor (Open Loop System) (Will require additional stepper motor controller IC's)
- DC Motor with encoder (Closed Loop System with feedback)

Options for sensory input:

- Potentiometer
- Rotary Encoder

Additional Items which can be added:

- Display (I2C OLED display or LCD Display)
- Joy Stick (if we do Stage 1 Step 2 in detail as mentioned above)

Component	Qty	Servo Option	Encoder Motor Option
Motors	5	5 × ₹220 = **₹1,100**	5 × ₹500 = **₹2,500**
Display	1	₹250	₹250
Buttons	3	₹40	₹40
Joystick (optional)	1	₹150	₹150
Total (No joystick)	—	**₹1,390**	**₹2,790**
Total (With joystick)	—	**₹1,540**	**₹2,940**

Feature / Board	ESP32 Dev Board	Arduino UNO	Arduino Mega 2560
Price (Robu.in)	**₹350**	**₹500**	**₹2,999**
Digital GPIO Pins	~26 usable	14	54
PWM Pins	All GPIO support PWM (16 channels)	6	15
Analog (ADC) Pins	18	6	16

PS: Arduino Mega is an overkill over here. Therefore taking Rs. 500 as an average for microcontroller price.

In above calculations power supply, breadboards and wires needed are not taken into account. That's accounting for extra Rs. 500 for the above mentioned and other miscellaneous items required.

Total estimated cost: Rs. (1540 + 500 + 500) = 2540 ~ Rs. 2.5K (using Servo Motor)
Total estimated cost: Rs. (2940 + 500 +500) = 3940 ~ Rs. 4K (using Encoder Motor)