Educational Purpose Robotic Arm

A Low-Cost, Modular Robotic Arm for Teaching Robotics and Control Systems

Syed Shafi Ahmed [001248523]   Supervisor : Dr. Jan Krabicka

PROJECT BACKGROUND & MOTIVATION

Robotic arms are widely used in industries for tasks such as **assembly, welding**, and **packaging**. However, their **high cost** and **complexity** limit their use in **education**. This project addresses this gap by developing a **low-cost**, **modular**, and **easy-to-use** robotic arm for **students** and **educators**. The system helps learners understand concepts in **robotics**, **embedded systems**, **control**, and **kinematics** through hands-on experimentation.

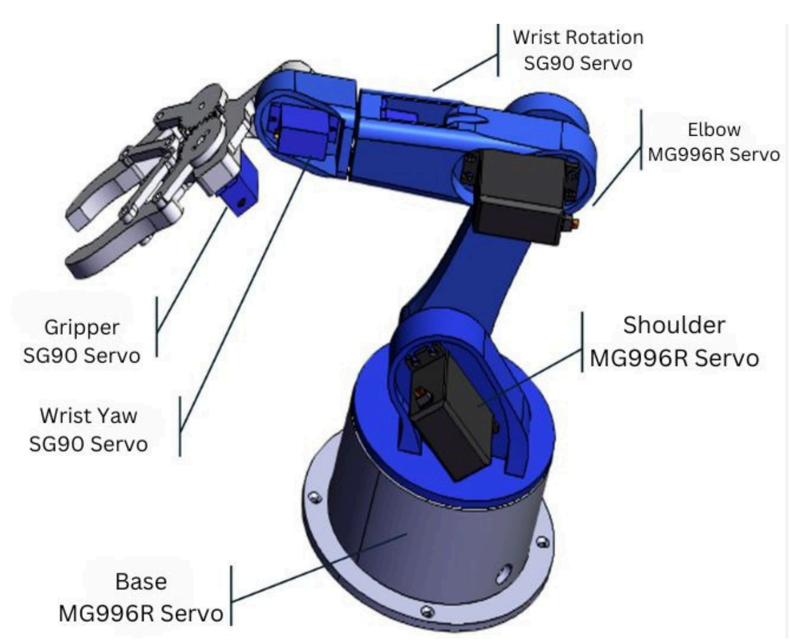
AIMS AND OBJECTIVE

- Design a modular robotic arm with **3D-printed** parts
- Use **Arduino** and **servo motors** for control
- Implement **inverse kinematics** for real-world positioning
- Enable **smooth movement** via l**inear interpolation**
- Ensure **low cost**, **easy assembly**, and **expandability**.

PROJECT DESIGN & METHODOLOGY

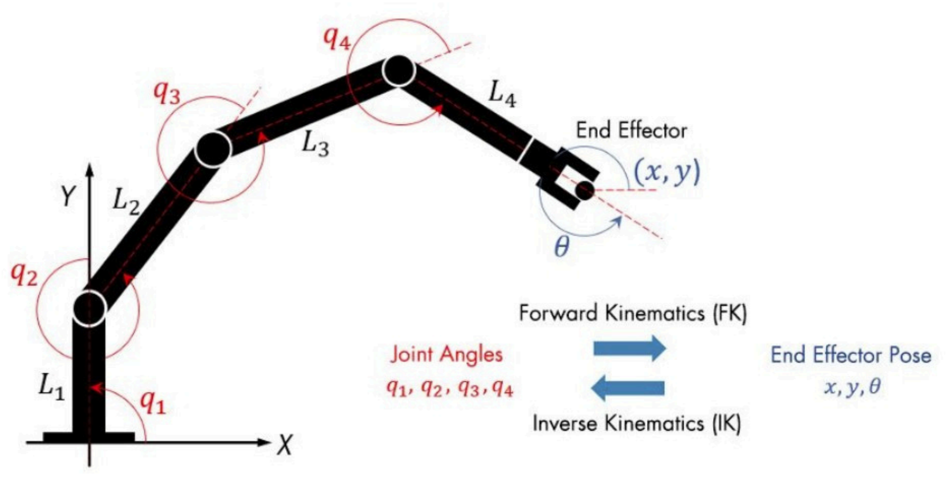
- **Programming Language:** Arduino C++
- **Hardware:** Arduino Uno R3
- **Components:** Servo Motor, 12V 2amps Power, CAD model
- **Key Algorithm:** Kinematics, Linear Interpolation
- **Software and UI:** Arduino IDE
- **Library:** Servo.h, Math.h

System Design

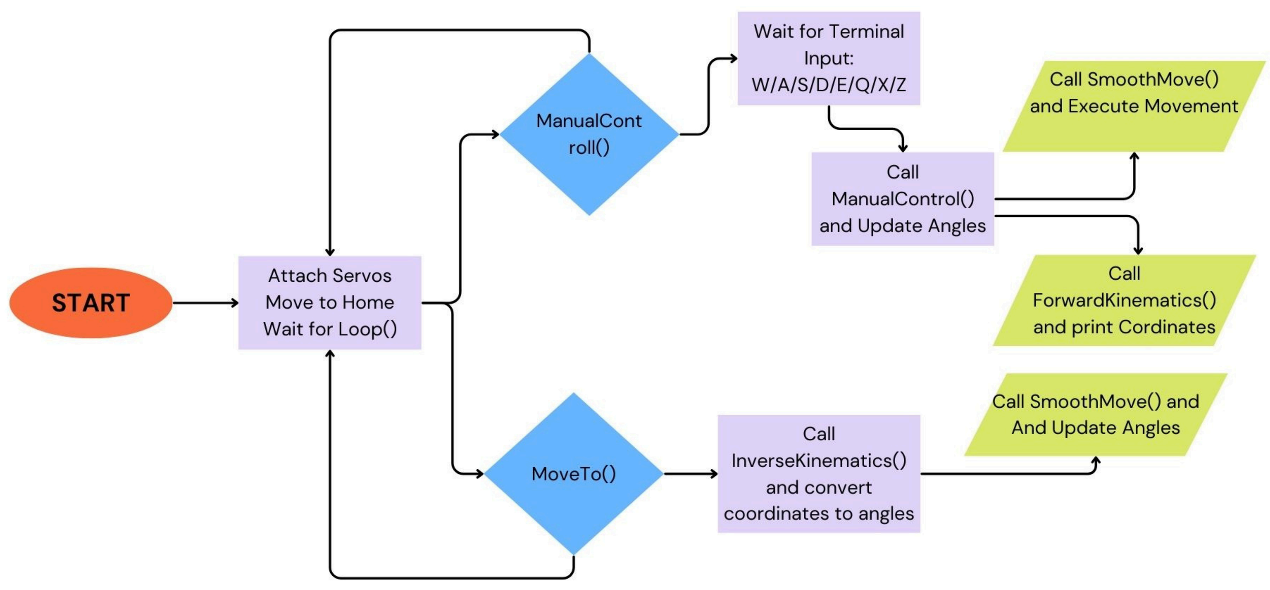


Motion Theory

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



Flow Chart

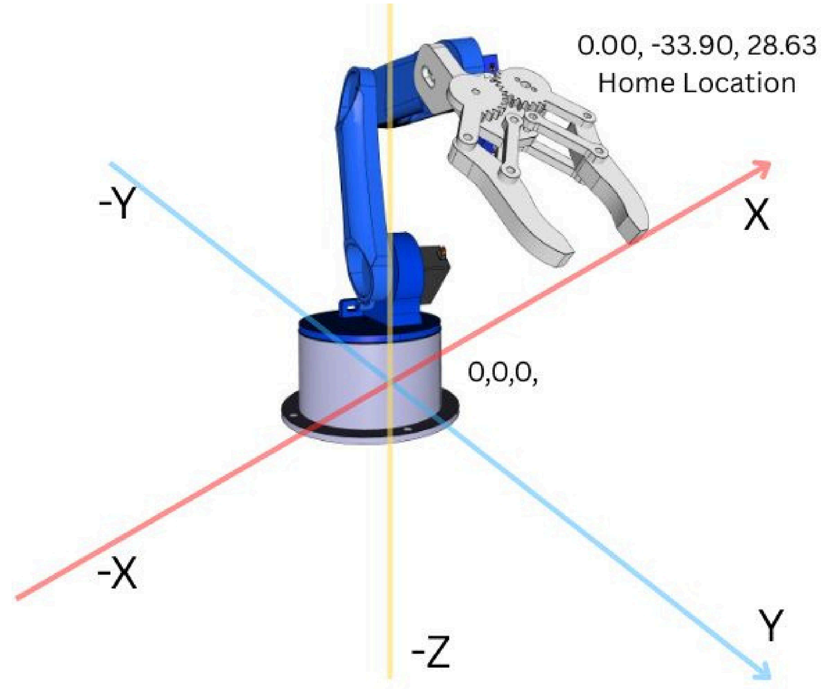


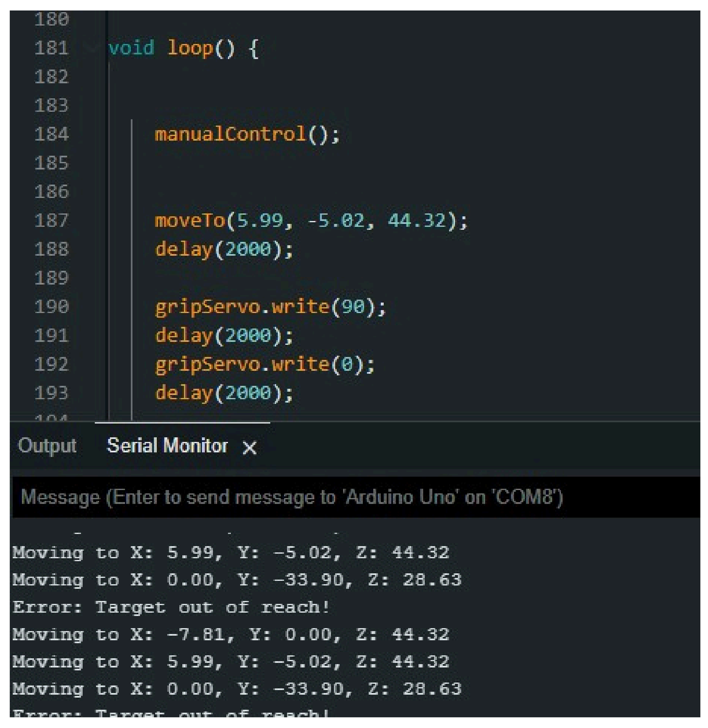
PROJECT IMPLEMENTATION

Kinematics allows to create **virtual coordinate system** and **advanced motion control**

```
void inverseKinematics(float X, float Y, float Z) {
    float r = sqrt(X * X + Y * Y);
    float z = Z;

    float cosTheta2 = (r * r + z * z - L1 * L1 - L2 * L2) / (2 * L1 * L2);
    if (cosTheta2 < -1 || cosTheta2 > 1) {
        Serial.println("Error: Target out of reach!");
        return;
    }

    float theta2 = acos(cosTheta2);
    float k1 = L1 + L2 * cos(theta2);
    float k2 = L2 * sin(theta2);
    float theta1 = atan2(z, r) - atan2(k2, k1);
    float thetaBase = atan2(Y, X);

    baseAngle = constrain(radToDeg(thetaBase), 0, 180);
    shoulderAngle = constrain(radToDeg(theta1), 0, 180);
    elbowAngle = constrain(radToDeg(theta2), 0, 180);

    smoothMove(baseServo, baseAngle, baseAngle);
    smoothMove(shoulderServo, shoulderAngle, shoulderAngle);
    smoothMove(elbowServo, elbowAngle, elbowAngle);
}
```

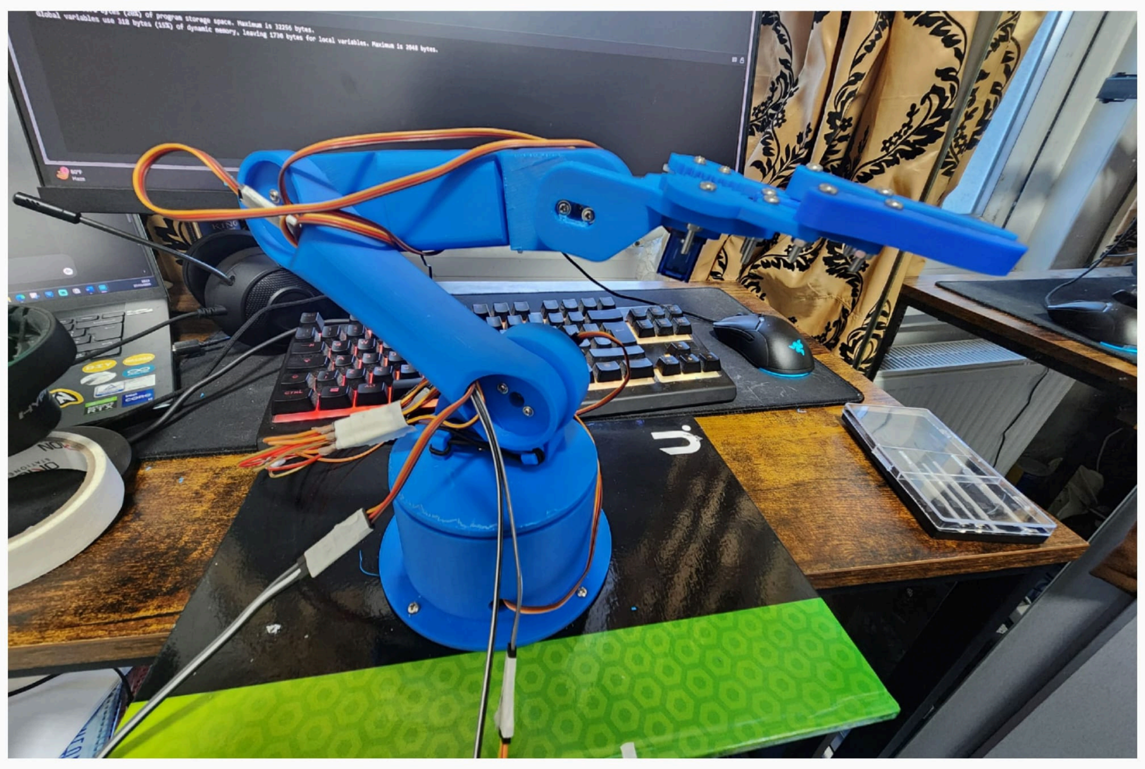
Uses **Arduino IDE** terminal for **manual control**. **Individual links** controlled by changing **angles** using **W/A/S/D/Q/E/Z/C** characters. **Advanced Control system** and **Automation** includes **Programming** using **Functions**.





OUTCOME

- Functional **6DOF robotic arm prototype** built and tested
- Achieved accurate **positioning** using **IK**
- **Movement** smoothed with **interpolation**
- **Total cost** remained within **£70–£100**
- Suitable for **hands-on learning** in **robotics courses**
- Easy to control using **IDE terminal** and **coding**.



CONCLUSION

This project demonstrates that with thoughtful **design** and **affordable components**, **robotics education** can be both **accessible** and **engaging**. The robotic arm bridges the gap between **theory** and **practical application**, offering **a hands-on learning experience**. Future improvements such as **wireless control**, **sensor feedback**, and **ROS integration** will enhance its functionality.

UNIVERSITY of GREENWICH