**Prosthetic Arm Controlling with EMG Muscle Sensor**

**1. Title**

**Smart, Affordable Prosthetic Arm: A 3D-Printed Muscle-Controlled Assistive System for Human and Robotic Use**

**2. Introduction**

Prosthetic limbs are critical tools for individuals who have suffered limb loss, enabling them to regain essential functionality and independence. However, advanced prosthetic technologies often remain inaccessible due to high costs and limited availability in developing regions. This project proposes a cost-effective, 3D-printed prosthetic arm controlled via electromyography (EMG) muscle sensors. It is designed to be functional, affordable, and accessible for both human and robotic applications.

**3. Uniqueness of the Project**

* Fully **3D-printed** design with a weight of 400g and modular assembly.
* Integrated with a **high-quality EMG sensor** that interprets muscle movement into real-time servo control.
* Uses **open-source components** and code, promoting educational and research use.
* Designed to serve **both amputees and robotic platforms**, which is rare in low-cost prosthetics.
* Utilises standard MG996R and MG90S servo motors for mechanical actuation, with tendons formed from fishing line and paracord.

**4. Research Gap**

* Most low-cost prosthetic solutions lack intuitive EMG control.
* Current open-source robotic hand models are not typically adapted for real-world prosthetics.
* Limited research on EMG-controlled arms that are **3D-printed**, **modular**, and **open-source**, particularly with affordable hardware suited for developing countries.

**5. Market Value and Usefulness**

* **For amputees**: Provides basic hand functionality such as grasping, pinching, and releasing.
* **In robotics**: Serves as a human-like actuator controlled via EMG or other signals for gesture replication.
* **In education**: Enables students to explore concepts in biomedical engineering, mechatronics, and robotics.
* **Market potential**: High in countries with limited access to advanced prosthetics. Potential for NGOs, assistive tech startups, and robotics labs.

**6. Integration for Humans and Robots**

* **Human use**: Surface EMG electrodes are placed on the forearm muscles. Signals are read and translated into joint movement using an Arduino.
* **Robotic use**: The EMG input can be replaced with digital or AI-based signals, allowing it to act as a programmable robotic manipulator.

**7. Manufacturing Process**

| **Step** | **Description** |
| --- | --- |
| 3D Design | Used open-source design from grossrc |
| 3D Printing | PLA filament, 10–30% infill, ~400g |
| Mechanical Assembly | 5 servo motors, tendon routing, glue |
| Electrical Setup | Arduino Uno, PCA9685 driver, EMG sensor |
| Coding and Signal Processing | Arduino IDE, EMG input filtering |
| Testing | Manual and EMG signal test via servo tester |

**8. Components and Specifications**

| **Component** | **Specification** | **Quantity** | **Source** |
| --- | --- | --- | --- |
| MG996R Servo Motor | Torque ~1.2–1.5 Nm, 180° rotation | 4 | RoboticsBD |
| MG90S Micro Servo | 180° rotation, lightweight finger control | 1 | RoboticsBD |
| Arduino Uno R3 | ATmega328P Microcontroller | 1 | RoboticsBD |
| PCA9685 16-Channel Servo Driver | 12-bit PWM resolution | 1 | RoboticsBD |
| EMG Muscle Sensor Kit | Surface electrodes, signal conditioner | 1 | RoboticsBD |
| Breadboard, Wires, USB Cable, Glue Stick,etc | Standard electronics | Various | RoboticsBD |
| 3D Printed Hand | Humanoid design from grossrc (Thingiverse) | 1 | Local print |

**9. Pseudocode for Control Logic**

Start System

Initialize Arduino, EMG Sensor, PCA9685, Servo Pins

LOOP:

Read analog EMG signal from muscle sensor

IF signal > threshold:

Normalize signal to angle range (0–180)

Map angle to corresponding servo for finger movement

Send angle as PWM output to each servo via PCA9685

ELSE:

Set all servos to rest position

End Loop

**9. Analytical and Logical Control Design for EMG-Controlled Prosthetic Arm**

**9.1 System Logic Overview**

The control loop transforms human muscle signals into finger movement using signal processing, linear mapping, and servo actuation.

**Signal Flow Architecture:**

1. EMG Sensor (Analogue Signal)

↓

2. ADC Conversion (10-bit: 0–1023)

↓

3. Noise Filtering (EMA Filter)

↓

4. Signal Thresholding (rest vs active)

↓

5. Normalisation to intensity (0–1)

↓

6. Mapping intensity to servo angle (0°–180°)

↓

7. Convert angle to PWM (pulse width)

↓

8. Servo actuation via PCA9685

**9.2 Mathematical Breakdown**

A white sheet with black text and black text

AI-generated content may be incorrect.**A math equations and formulas

AI-generated content may be incorrect.**

**9.3 Final Control Logic – Pseudocode**

START

// --- INITIALISATION ---

Set EMG\_input\_pin = A0

Set smoothing\_alpha = 0.2

Set θ\_min = 0°, θ\_max = 180°

Set PWM\_min = 1000 μs, PWM\_max = 2000 μs

Set threshold\_rest = 400

Set threshold\_active = 500

Set signal\_max = 900

Initialize PCA9685 (I2C)

Attach servos to channels: [thumb, index, middle, ring, pinky]

Set all servos to neutral (90°)

// --- MAIN LOOP ---

Loop Forever:

// Step 1: Read EMG analog value

raw\_signal = analogRead(EMG\_input\_pin)

// Step 2: Apply Exponential Moving Average Filter

filtered\_signal = (smoothing\_alpha × raw\_signal) + ((1 - smoothing\_alpha) × filtered\_signal\_previous)

filtered\_signal\_previous = filtered\_signal

// Step 3: Check if signal is above rest threshold

If filtered\_signal < threshold\_rest:

Set all servos to rest (θ = 90°)

continue

// Step 4: Calculate contraction intensity

intensity = (filtered\_signal - threshold\_active) / (signal\_max - threshold\_active)

intensity = Clamp between [0, 1]

// Step 5: Convert intensity to angle

θ = θ\_min + intensity × (θ\_max - θ\_min)

// Step 6: Convert angle to pulse width

P = PWM\_min + (θ / 180) × (PWM\_max - PWM\_min)

PWM\_tick = (P / 20000) × 4096

// Step 7: Send PWM tick to all servos

For each servo in [thumb, index, middle, ring, pinky]:

Set PWM\_tick on corresponding PCA9685 channel

// Step 8: Small delay to avoid jitter

delay(10 ms)

END LOOP

END

**9.4 Engineering Notes and Optimisations**

**Dynamic Signal Ranges:**

* Muscle strength varies per user. Calibration can be performed by taking average resting EMG for 5 seconds and setting that as threshold\_rest.

**Safety Enhancements:**

* Implement watchdog timers: If signal drops or noise spikes > 1 sec, reset system.
* Limit servos between 10° and 170° to avoid binding.

**Multichannel EMG (Advanced Scope):**

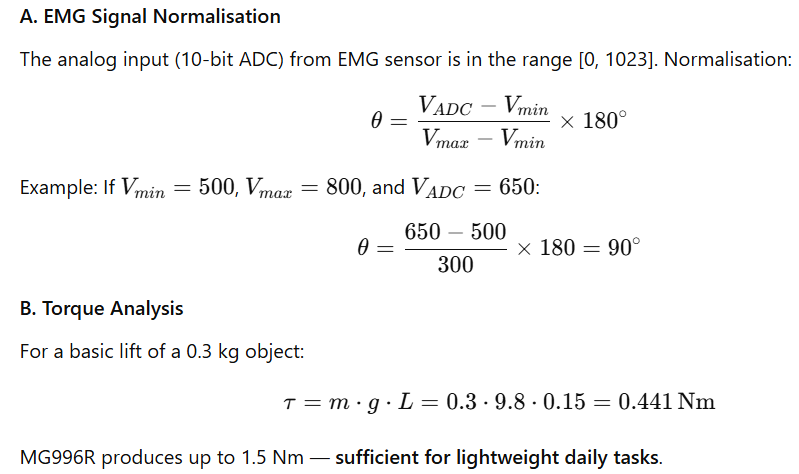
* For advanced versions, use multiple EMG sensors to differentiate thumb and finger contractions.

**9.5 Numerical Example**

A math problem with numbers and lines

AI-generated content may be incorrect.

**10. Mathematical Analysis**

****

**11. Budget Summary**

| **Item Category** | **Cost (BDT)** |
| --- | --- |
| Electronics (purchased) | 6,862 |
| 3D Printing (400g arm) | 4,000 |
| Miscellaneous (tools, glue) | 2000 |
| **Total Cost** | **12,862** |

( This price can be changed anytime due to high pricing of shipment from CHINA)

**12. Conclusion**

This project demonstrates a cost-effective, functional prosthetic arm that uses EMG muscle signals for natural control. Its affordability, modularity, and open-source nature position it as a valuable innovation in both healthcare and robotics. The system can be enhanced with gesture recognition, AI algorithms, or remote robotic integration, offering future development opportunities.