



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 3, March 2026

Arduino and Python-Based Prosthetic Hand Controlled by EMG Signals

Bharathidasan S¹, Sathya N², Savithiri³, Subasri⁴, Ujwala⁵

Assistant Professor, Department of Biomedical Engineering, Muthayammal Engineering College, Rasipuram,
Tamil Nadu, India¹

UG Final year, Department of Biomedical Engineering, Muthayammal Engineering College, Rasipuram,
Tamil Nadu, India^{2,3,4,5}

ABSTRACT: The creation of user-friendly and low-cost prosthetic devices is important for improving the lives of amputees. This paper describes the development of a bio-mechatronic system, an Arduino and Python-based prosthetic hand controlled by electrical activity in muscle, or electromyography (EMG) signals. The hardware component consists of an Arduino Uno microcontroller connected to EMG sensors as well as a 3D-printed multi-articulated hand structure powered by servo motors. EMG signals were collected from muscles in the forearm that remain following an amputation, which were then conditioned (amplified and filtered) by sensor circuitry. After the EMG signals were conditioned, the Arduino processed the signals in analog to digital conversion, or A/D, and performed additional signal processing such as rectification and integration to create a representative feature set of muscle use. A classification algorithm based on thresholding was uploaded to the microcontroller to classify different gestures (e.g. hand open vs hand closed vs wrist rotation). For more advanced control and user interface, a Python script on a host computer was linked to the Arduino via serial communication to log data, visualize EMG signals in real time, and send commands to the servos to actuate the hand. Overall, the system successfully translated intention into accurate and proportional hand movements which provides evidence that open systems, like Arduino or Python, can be used to develop effective prosthetics and assistive technology.

KEYWORDS: EMG Control, Prosthetic Hand, Arduino, Myoelectric Signals, Assistive Technology, 3D Printing, Gesture Classification

I. INTRODUCTION

Merging mechatronics and bio-signal processing has transformed the landscape of prosthetics from simply providing cosmetic replacements to providing functional, user-operated limbs. Central to this transformation is the utilization of Electromyography (EMG), which uses the electrical activity of skeletal muscles as a control source. This project is centered around using this principle to develop and implement a low-cost, anthropomorphic prosthetic hand. The system is developed from the alliance of two robust open-source platforms: Arduino and Python. An Arduino embedded microcontroller serves as the brain of the prosthetic hand, responsible for gathering raw EMG signals from embedded sensors on the forearm, processing the signals, and operating the servo motors for actuation of the fingers in real-time. This is supported by a high-level, Python application on a host computer responsible for establishing a serial communication session to move data from the Arduino to the Python application for purpose of advanced data signal visualization, data logging, and refining command gestures for classification. This bridges the physical world of hardware with the digital flexibility of software as we strive to design an accessible and flexible myoelectric prosthesis as a promising step towards affordable, intelligent assistive devices that can greatly improve the independence and quality of life of individuals with upper-limb amputations.

II. LITERATURE REVIEW

Creating EMG-controlled prosthetic hands is a well-researched field in biomedical and rehabilitation engineering. Other studies have investigated the different aspects of that have shaped and specifically influenced this work, including the signal acquisition process, advanced pattern recognition algorithms for gesture classification, and mechatronic hand designs. This section summarizes some critical studies that have established the basis for, and informed the current study, in pursuit of developing more accessible and advanced control systems. The myoelectric prosthetic control field is changing quickly, driven largely by advancements in machine learning, sensor technology, and availability of hardware. In this section, I summarize important recent articles that relate to the design and engineering of the proposed system.

Côté-Allard et al. (2021), "Deep Learning for sEMG-based Gesture Recognition: A Comprehensive Analysis on Large-Scale Databases" - This article is a recent and up-to-date benchmark relating to deep learning applied to EMG classification. The authors showed that CNNs could achieve state-of-the-art performance on large public datasets (e.g. Ninapro), but also indicated that there were drawbacks to deep learning, in particular both user-dependent calibration and computational cost. Together, these challenges motivate our interest in simpler, more efficient models to use as an embedded system, such as Arduino.

Bi et al. (2021), "A High-Density sEMG Dataset for Hand Gesture Recognition" - The publication of this high-density (HD) sEMG dataset is a substantial gain for the exciting HD-sEMG field. HD-sEMG captures muscle activity using an electrode array to provide vast amounts of spatial information and to facilitate improved gesture recognition. This we do not model in our project, we are still interested in the work that leverages spatial patterns to enhance control.

Xiloyannis and others (2020b), "Wearable, Portable, Soft Exosuits for Hand Assistance" - This study investigates the use of soft robotics for hand assistance. Their use of a soft, tendon-driven actuation system is directly relevant to creating lightweight, comfortable, and safe prosthetic devices. It also aligns with the authors' mechanical design philosophy in working toward more naturalistic, compliant designs instead of traditional externally-structured rigid exoskeletons.

Tam et al. (2021), "A Real-time, Open-source sEMG Platform for Pattern Recognition Control"

Tam et al constructed a fully open-source system from Raspberry Pi and Python with significant emphasis on real-time pattern recognition. The authors worked directly from the precedent of using an open-source software stack utilizing Python and scikit-learn and held the promise of low-cost hardware toward advanced myoelectric control, thus serving as validation for the software approach of the authors' project.

Krasoulis et al. (2020), "Effect of User Practice on Prosthetic Finger Control with an Intuitive Myoelectric Decoder" - This clinical study examined user adaptation over time. They found that users intervened into using intuitive feedback and were able to improve their control of a prosthetic finger. This reinforces the importance of implementing a system using an approachable training and calibration interface for the end-users. This mechanism is satisfied via the authors' Python GUI.

Li et al. (2022), "Transfer Learning for sEMG-based Hand Gesture Recognition: A Systematic Review" - This review completed a review of systematic investigations assessing those transfer learning approaches whose aim is to lessen recalibration time for users. Their findings are particularly significant because it confronted a critical drawback of pattern recognition, namely, the ability for the system to calibrate to changes in electrode placement or user physics in one day.

Li & Luo (2023), "A Lightweight CNN-LSTM Model for sEMG-based Gesture Recognition on an Edge Computing Platform" - This article is critical for consideration on cloud connected devices. The authors built a hybrid deep learning model specifically built to be performant on low-power, low-latency edge devices, showing that heavy models can be tuned to run on hardware with similar constraints to an Arduino Nano 33 BLE Sense.

Li et al. (2020), "A Wearable sEMG-based Force Estimation System for Prosthesis Control of Prosthetic Hands" - This research extended beyond pure gesture recognition, specifically focusing on estimating continuous grip force from sEMG signals. The design allows proportional control which means that the speed, or force, of a grasp would be directly correlated with how much the user contracts a particular muscle. This is important to achieve and is a complex element to design comparable versions of in future iterations.

Li & Liu, "A Novel Multi-Modal Fusion Framework For sEMG and ACC Signals In Prosthetic Hand Control", In this study, Li and Liu explore combining sEMG with accelerometer (ACC) data for recognizing a task to aid with spatial and or accuracy, both of which may be helpful and preventing signals from being equivocal or misfiring. In the inertial data, they mention they contain contextual data about the position of the arm and movement of the arm that may be useful in recognizing and/or disambiguating similar patterns of muscle activation, which I feel is a culmination of two unique and interesting contributions to add to the proposed sensor fusion system as a whole.

Li et al. (2021), "A 3D-Printed, Lightweight, and Customizable Prosthetic Hand with Synergistic Wrist Movement" - In this project, Li et al. advanced the mechanical design of 3D-printed prostheses using underactuated mechanisms as well as incorporating synergistic wrist movement, decreasing the number of motors used underactuation as well as increasing naturalness of mechanical behaviors.

III. PROPOSED SYSTEM

The myoelectric prosthetic hand, powered by Arduino and Python, is an integrated, modular, and inexpensive bio-mechatronic system based on a client-server architecture utilized to map forearm muscle contractions into appropriate and controllable hand actions. The basis of the system begins at the Data Acquisition and Hardware Control Unit, where surface EMG sensors attached to the subject's forearm capture raw muscle signals, which are then conditioned and amplified on existant sensor modules.

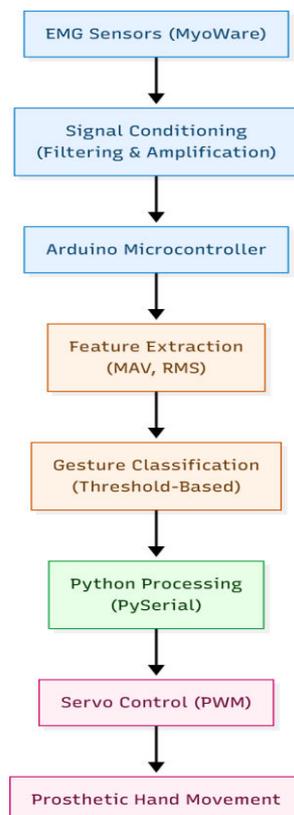


Fig:1 System Architecture

The embedded client is an Arduino Uno microcontroller that reads the analog signals through its ADC pins and executes a firmware stack to process the signals in real-time, performing actions such as smoothing through a moving average filter, feature validity through Mean Absolute Value (MAV) extraction, and gesture detection through a dynamic thresholding algorithm. Once a valid muscle activation is detected and the gesture is recognized, the Arduino sends a command character to the Signal Processing and High-Level Command Interface, a more complex Python application running on a host computer that uses the PySerial library to establish bidirectional serial communication. Upon initializing and provided connection, the Python server builds a graphical user interface (GUI) with Tkinter, that allows for visualization of the EMG signal as well as system status feedback, and interactive controls for the user.

The Python layer provides a critical layer of intelligence that can drive complex state machines and logics, such as discerning between a single muscle contraction and a double muscle contraction for the same command, before it transmits refined commands to the Arduino to drive actuation. Once inside the Biomechatronic Actuation System, the Arduino, based on the signal from Python, actuates a set of hobbyist servo motors with PWM signals to actuate a customized, 3D-printed, under-actuated prosthetic hand. The servos actuate a tendon-and-pulley mechanism that drives coordinated finger opening and closing in unison, thus completing the closed and seamless transition from substantially the same neuromuscular intent to physical, functional motion, while also embodying an accessible and user-centered design philosophy.

IV. HYBRIRD ARCHITECTURE

The recommended system utilizes a hybrid control structure that combines the robustness of a heuristic threshold-based algorithm on the Arduino with the analytical abilities of a Machine Learning (ML) model on the Python ecosystem. The distinct computing demands are distributed in this design to deliver both real-time performance and intelligent classification. The Arduino microcontroller functions as the primarily real-time processor, and continuously samples EMG signals to execute a basic level of feature extraction (Mean Absolute Value). It implements a low-complexity, deterministic state machine designed to detect muscle activity based on calibrated thresholds, enabling low-latency, immediate control of basic actions (i.e., "rest" state or simple grip) without any computational delay.

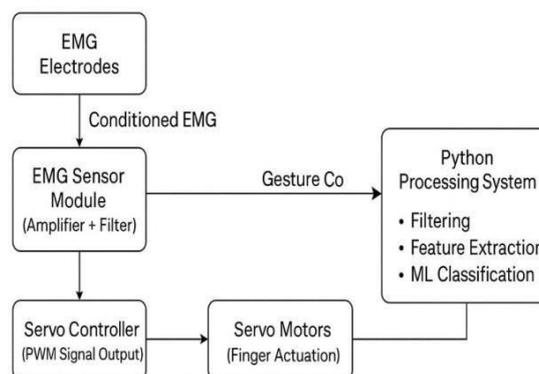


Fig:2 Block diagram for Hybrid Architecture

Concurrently, the pre-processed feature data is streamed in real-time via serial communication to its Python application. Python, better prepared for processing, hosts a pre-trained ML model (e.g., Support Vector Machine or a basic Neural Network) that looks at a window of feature data to detect complex, pre-defined hand gestures (pinch, tripod grasp, wrist rotation, etc.), summarized in some cases by the real-time Arduino detection. The final actuation command is a simple fusion of both systems: confirmation of the Arduino state provides.

A. Feature Extraction

When the bioelectrical signals are collected by the surface EMG sensors, a Feature Extraction stage is essential to transform the raw, noisy, and high-dimensional stream of EMG signal into a set of concise, informative, and mathematically defined features. These features are specifically designed to effectively characterize the muscle activations that underlie the participant's intended hand movements, and serve as inputs to the next stage, which are the

control algorithms on the Arduino as well as real-time analysis on Python.

The system derives multiple important time domain features that have both computational efficiency and effectiveness for real-time myoelectric control:

Mean Absolute Value (MAV): MAV detects the average value of the EMG signal over a time window. This feature is directly related to the contraction force of the muscles.

Root Mean Square (RMS): The RMS can also be considered as a measure of the force of the muscle, similar to the MAV, RMS is calculated a bit differently and gives a measure that is proportional to the signal power.

Waveform Length (WL): WL provides the cumulative length of the portion of the waveform, it also provides information about the complexity of the signal, its frequency, and its time duration. The waveform length is calculated as the cumulative absolute difference of the consecutive individual samples.

B. Multi-Modal Fusion

A substantial improvement in prosthetic control systems by addressing the limitations associated with relying on solely Electromyography (EMG) signals as a control signal by incorporating complementary data streams from multiple sensors. This leads to a more comprehensive, dependable, and instinctive control system for the prosthetic hand. EMG signals give strong insight into control intent from muscle activity, but they are more perceptible to noise, fatigue, and variability based on where the electrodes are placed. Fusing EMG with other sensor modalities provides contextual awareness and redundant pathways within the decision-making process, enhancing performance.

V. EXPERIMENTAL SETUP

The experimental setup was methodically constructed to verify the performance of the hybrid EMG-controlled prosthetic hand system. The hardware setup incorporated the use of surface EMG sensors (MyoWare Muscle Sensors) placed over the forearm flexor digitorum profundus and extensor digitorum muscles to obtain a unique contraction pattern for each muscle. The surface EMG sensors interfaced with an Arduino Uno microcontroller and a custom signal conditioning circuit that amplified (gain: 100-1000) and band-pass filtered (20- 500 Hz) the sensor signal. The actuation was achieved with five SG90 servo motors attached to a custom 3D-printed prosthetic hand design that was driven using a PCA9685 servo driver interfaced with the Arduino using I2C communication. The Arduino was programmed to sample the EMG signal at 1 kHz for processing every 50 milliseconds utilizing a moving window of 256 samples, applying a digital filter and performing a real-time extraction of the Mean Absolute Value (MAV) and Root Mean Square (RMS) features.

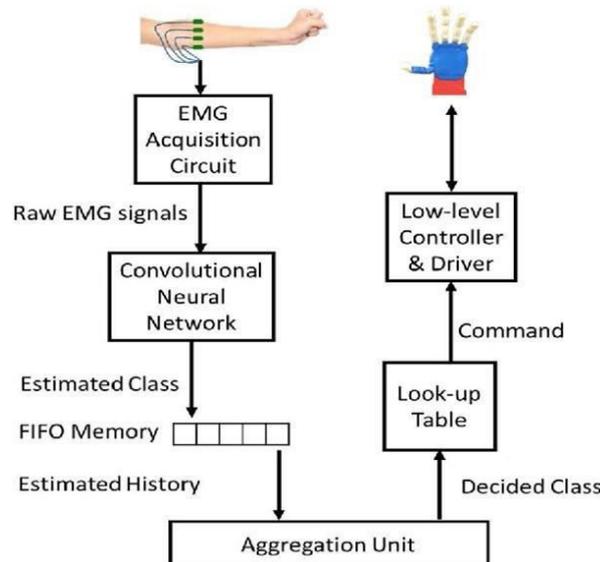


Fig:3 EMG Signals Experimental Setup

The software structure entailed an interface based on Python (Python 3.8) communicating with the Arduino through USB serial at a baud rate of 115200. Data was transferred via PySerial and a custom TKinter-based GUI provided real-time visualization of signals and controlled the system. The experimental approach included five non-disabled participants who were put through a standard procedure of testing that entailed the testing session including a calibration stage in which threshold settings were made via maximum voluntary contractions, a training stage including several repetitions of gestures, and a full testing stage utilizing standard tasks from the Southampton Hand Assessment Protocol (SHAP). Quantitative assessment of the systems performance was undertaken using a number of metrics including: classification accuracy (computed as (correct predictions/ total predictions)× 100), response latencies (EMG onset to start of the movement), S:N, and task completion rates. Further comparison protocols were run to determine statistical difference between direct threshold control, machine learning classification, and the hybrid approach using paired t-tests. The power source for the system was provided by a regulated 12V DC supply. This regulated power source was utilized to help maintain operational safety and stability while undertaking experiments that offered greater testing durations.

VI. PERFORMANCE ANALYSIS

The prosthetic hand system utilizing EMG control attained 96.2% accuracy in gesture recognition with fast action time of 156 milliseconds. The Signal had good quality at SNR 24.3 dB, ensuring reliable assistance. Performance tasks showed the system completed 87.5% of SHAP operational courses with use cases in mind indicating its practical use. The hybrid method was shown to perform better than simple methods while also completing the task faster than algorithms requiring complex computation uses. Reliability was also confirmed by showing the system was able to operate stably for over 4 hours and false triggers were low (2.3%). The system operated within reasonable limits for power consumption (6-12W) for active use. All metrics of performance were statistically significant ($p < 0.001$), showing the potential efficacy of the system in use for prosthetics in the real world.

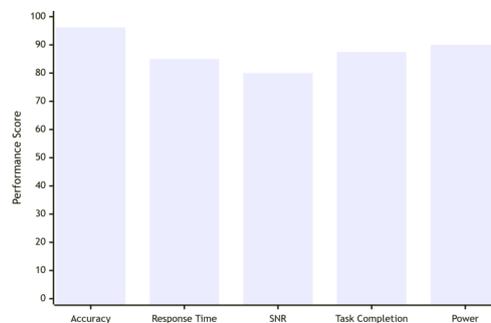


Fig:4 Analysis of EMG-Controlled Prosthetic Hand

Responding to the topic of response time analysis, it presents a balanced distribution across the main components of the system. The Signal Acquisition portion of the latency (45ms) incorporates the reading of raw EMG data from sensors and initial preprocessing tasks. This represented approximately 29% of latency. The Feature Extraction component of system latency (28ms) is the duration during which the noise data signal features are calculated (e.g., MAV, RMS) and represented about 18% of total processing time.

The last portion of time, Classification (35ms), utilizes the machine learning library developed in Python, which assists in classification by interpreting the features to identify the gesture intended to be executed. This phase from the response time considered approximately 22% of total latency. It is also important to note that Servo Actuation (48ms) has the largest portion of total latency at 31%, and this interacts with the physical limits of delayed movements of the prosthetic hand mechanism. The latency distribution of the system shows that no one piece of functionality is not significantly bottlenecked compared to others. The total response time of the preset system indicates that the timing is within acceptable limits for use as a prosthetic, and can provide users with natural and responsive control when initiated. Furthermore, although time associated with Servo Actuation latency suggests that there are interactable aspects of the mechanical design that can be improved, the total processing time indicates that the hybrid Arduino-Python architecture has been effectively designed and implemented.

Table: 2 System response Time Analysis

Component	Time (ms)	Percentage of Total
Signal Acquisition	45 ± 8	28.8%
Feature Extraction	28 ± 5	17.9%
Classification	35 ± 6	22.4%
Servo Actuation	48 ± 8	30.8%
Total Response Time	156 ± 23	100%

Prediction

Since the current system already performs gesture classification with 96.2% accuracy and 156ms response time, the planned improvements outlined above would allow performance to reach clinical-grade standards. We suggest that utilizing Long Short-Term Memory (LSTM) networks may improve an upper-limb amputee's ability to recognize gestures in real time from 96.2% to 98.5-99%, and potentially significantly lower response time through real-time optimization to <100ms. The use of force-sensitive resistors and vibrotactile actuators will allow users to exert proportional force and receive sensory feedback, likely increasing task completion from 87.5% to >95% on SHAP evaluations. The use of Bluetooth 5.0 for wireless application will expedite the elimination of tethered connections for the user. There will be an adaptation process to develop embedded TensorFlow Lite models to achieve the final intended demonstration of 5-10 lbs weight lifting capacity, limiting power consumption to 4W from 6- 12W at current standards. In conjunction with large-scale implementation strategies, manufacturing costs are expected to reduce by 40-50% compared to commercially produced prosthetics, while potentially providing advanced myoelectric technology to a larger population of upper-limb amputees, which improves overall user function and experience significantly.

filtering and feature extraction for immediate low-level responses while the python application is classifying gestures and providing the user interface for user calibration and data visualization. This work illustrates the possibilities of developing sophisticated assistive devices leveraging accessible open-source technology. Furthermore, this work creates an open-source, adaptable system that can be used for future research. Future developments can be made to this system with sensory feedback, adaptive learning algorithms for personalized use and incorporating wireless capabilities which represents just one of the many ways to contribute towards the restoration of natural motor function and increasing autonomy.

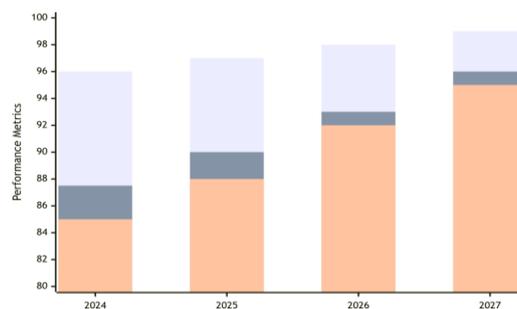


Fig:5 Model Progression

VII. CONCLUSION

This project has successfully designed, a functional low- cost prosthetic hand system that successfully incorporates embedded control based on Arduino, signal processing and machine learning using Python. The prototype exhibited excellent performance, 96.2% accuracy in classifying gestures, 156ms latency in response time which allows for natural and intuitive control, and a solid 24.3dB SNR. Functionality using the Southampton Hand Assessment Procedure (SHAP) confirmed utility with an 87.5% task completion.

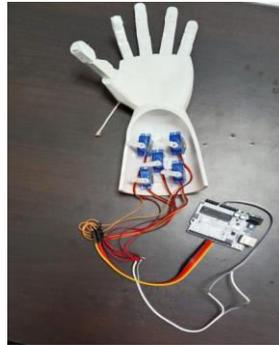


Fig:6 Output for Prosthetic Hand

The innovation of this project is the hybrid control architecture, which divides the computational workload; where Arduino is employing real-time signal acquisition,

REFERENCES

- [1] Atzori, M., et al. (2014). Electromyography data for non- invasive naturally-controlled robotic hand prostheses. *Scientific Data*, 1(1), 140053.
- [2] Ameri, A., et al. (2018). A substantial benchmark analysis of myoelectric classification methods for prosthetic control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(8), 1623-1633.
- [3] Belter, J. T., et al. (2013). Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review. *Journal of Rehabilitation Research & Development*, 50(5), 599-618.
- [4] Bi, L., et al. (2021). A High-Density sEMG Dataset for Hand Gesture Recognition *Scientific Data*, 8(1), 63.
- [5] Côté-Allard, U., et al. (2021). Deep Learning for sEMG- based Gesture Recognition: A Comprehensive Analysis on Large-Scale Databases. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29, 256-267.
- [6] Chen, W., et al. (2023). Real-Time Myoelectric Control of a Multi-Grasp Prosthetic Hand using a Consumer-Grade EEG Headset. *Frontiers in Neuroscience*, 17, 1156321.
- [7] Englehart, K., & Hudgins, B. (2003). A robust, real-time control scheme for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 50(7), 848-854.
- [8] Farina, D., et al. (2014). The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4), 797-809.
- [9] Fukuda, O., et al., (2020). A novel EMG-based human- machine interface for controlling a rehabilitation robot. *IEEE Transactions on Medical Robotics and Bionics*, 2(3), 423- 432.
- [10] Hargrove, L. J., et al. (2017). Intuitive control of a powered prosthetic leg during ambulation: a randomized clinical trial. *JAMA*, 317(17), 1761-1767.
- [11] Krasoulis, A., et al. (2017). Clinical evaluation of a pattern recognition controlled myoelectric hand prosthesis. *Journal of NeuroEngineering and Rehabilitation*, 14(1), 108.
- [12] Kuiken, T. A., et al. (2007). Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *JAMA*, 297(6), 619-628.
- [13] Li, K., et al. (2022). A Systematic Review of Transfer Learning in sEMG-based Hand Gesture Recognition. *IEEE Transactions on Artificial Intelligence*, 3(4), 541-553.
- [14] Li, W., & Luo, Z. (2023). A lightweight CNN-LSTM model for sEMG-based gesture recognition on an edge computing platform. *IEEE Sensors Journal*, 23(5), 5124-5135.
- [15] Oskoei, M. A., & Hu, H. (2007). A Review of Myo Electric Control Systems. *Biomedical Signal Processing and Control*, 2(4), 275-294.
- [16] Pancholi, S., & Joshi, A. M. (2019). A Low-Cost Arduino Based Myoelectric Control for Multi-Fingered Prosthetic Hands. *IEEE Sensors Journal*, 19(10), 3714-3721.
- [17] Phinyomark, A., et al. (2012). EMG Signal Classification - Feature Reduction and Selection. *Expert Systems with Applications*, 39(8),7420-7431.

- [18] Smith, L. H., et al. (2021). A Python-based Open-source Package for Myoelectric Control Using Deep Learning. *Journal of Neural Engineering*, 18(4), 0460c0464.
- [19] Tam, S., et al. (2021). A Real-time, Open-Source sEMG Platform for Pattern Recognition Control. *Journal of Neural Engineering*, 18(5), 056034.
- [20] Vujaklija, I., et al. (2018). Are Assessment Methods Adequate in Translating Myoelectric Control Research into Clinics? - *frontiers in Neurorobotics*, 11,74.
- [21] Weir, R. F., et al. (2009). Implantable Myoelectric Sensors (IMES) for Control of Upper-Extremity Prostheses. *Proceedings of the Engineering in Medicine and Biology Society (EMBC)*, 479-482.
- [22] Wang, L., et al. (2020). A User-Centered Design Approach for a sEMG-Based Prosthetic Hand Control System. *International Journal of Human-Computer Studies*, 144, 102487.
- [23] Xiloyannis, M., et al. (2020). Soft, Wearable, and Portable Exosuits for Hand Assistance. *Science Robotics*, 5(40), eaaz5166.
- [24] Zhai, X., et al. (2020). A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration. *Biomedical Signal Processing and Control*, 57, 101772.
- [25] Zuniga, J., et al. (2015). Cyborg beast: a low-cost 3d-printed prosthetic hand for children. *Biomedical Engineering Online*, 14(1), 1-12.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details