International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I8P124

Volume 74 Issue 8, 359-380, August 2026

Original Article

EMG-Based HMI to Control a Robotic Car

Manoj Kumar Mukul[1], Nitish Kumar[2], Mukesh Kumar Ojha[3], Yogesh Pratap Singh[4], Rajkishore Prasad[5]

[1,2,4]*Electronic Science, University of Delhi, Delhi, India.*
[3]*Electronics and Communication Engineering, GNOIT, Greater Noida, India.*
[5]*Principal, B.N. College, Patna University, Patna, India.*

[1]*Corresponding Author: manojkumarmukul@south.du.ac.in*

Received: 12 February 2026 Revised: 05 July 2026 Accepted: 15 July 2026 Published: 29 August 2026

Abstract - *A Human-Machine Interface (HMI) is a user interface that allows people to control a machine, carry out a task, and receive feedback from the machine to adjust their control. This paper examines the use of Electromyogram (EMG) signals for machine control via wireless command transmission. The EMG signal includes various muscle responses. The challenge in processing the EMG signal comes from the transient muscle response, which has an unknown arrival time, duration, and shape. The EMG-based Human-Machine interface (HMI) has attracted much attention because it can convert muscle activity into usable control signals. This paper describes the design and implementation of a single-chip platform for an EMG-controlled system to achieve a small, effective and responsive control interface. The development of on-chip real-time processing for EMG-based HMI systems removes the need for an external computer or server. A complete embedded system has been created to control an external car using EMG signals. Different commands are also generated to manage the car's movements. This paper focuses on building a real-time embedded system to control the external car by analyzing hand muscle signals to create commands based on the envelope of both muscle signals. The proposed method for processing EMG signals in a real-time, low-cost system is part of the HMI system, and we discuss its future potential. With the new system, performance accuracy is over 90% in real time.*

Keywords - *HMI, EMG, Embedded system, Real-Time signal processing.*

1. Introduction

A person with a disability needs an assistive device to perform daily activities. [1] There are various types of disabilities in society. Technologists help individuals live comfortably by creating assistive systems. A man-machine interface is an essential part of an assistive system. [1, 2] It allows users to interact with external devices using different biopotentials from the human body, based on the system's ease and feasibility. Researchers are working on a neurorehabilitation system that uses biosignals collected from the human body. [3] The 21st century is often called the era of robotics and control systems. Recent technological advancements have enabled the development of new systems for various practical applications. One can integrate various types of control systems into our daily lives. They improve automation and the interaction between humans and machines. One notable type is the electromyogram (EMG) signal-based Human-Machine Interface (HMI). These interfaces have attracted significant attention because they can convert muscle activity into usable control signals. A Human-Machine Interface (HMI) is a technology that enables communication between a user and a machine via a computer or interface device.The idea of a man-machine interface is an important area of research where users interact with machines through interface devices in real time. A Brain-Computer Interface (BCI) is a related technology. In a BCI, users control external devices via a computer by analyzing incoming brain signals to understand the brain's state [4].

The human body is a combination of electrical, chemical and biological. There are many electrochemical processes happening in the human body to sustain life. The human body is controlled by the Central Nervous System (CNS). It is made up of nerve cells and glial cells that are located between neurons. Nerve cells react to stimuli and send information over long distances. These CNS networks connect to muscle tissue, allowing for body movement. The length and shape of individual muscle fibers in a whole muscle vary. The characteristics of the action potential in muscle fibers also differ at various fiber sites. This means the EMG signal depends on electrode placement. There is a need to acquire different physiological signals from the human body to interact with external devices. These include Electroencephalography (EEG), Magnetoencephalography (MEG), and Electromyography (EMG). The communication with external devices is indirect via the brain, giving rise to a brain-machine interface. Since 1999, researchers have sought to control the movement of electric wheelchairs and robots using the movement imagery paradigm. A Human-Machine Interface (HMI) is a user interface that lets people control a machine, perform a task, and receive feedback to adjust their control [5]. There are several techniques by which people can operate machines. A push button can start the machine in an industrial process, and it is widely used in HMIs. Similarly, biosignals can also control the machine. Bio-signal-based HMIs use signals from the human body to control machines or devices.

Researchers have frequently used Electromyography (EMG) and surface Electromyography (sEMG) signals in gesture recognition, prosthetic hand control, rehabilitation engineering, and human-machine interaction systems [6, 7]. Machine learning, deep learning, reinforcement learning, and edge AI technologies have recently focused on improving classification accuracy, real-time performance, robustness, and intelligent control. Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan performed one of the first comprehensive studies of Brain-Computer Interfaces (BCIs) [4]. The authors discussed translating EEG and EMG signals into commands for communicating with and controlling external devices. The research work covered the following areas: signal acquisition, adaptive signal processing, feature extraction and real-time control for assistive applications. The results of the research showed that neural signals can control external devices effectively without muscular movement, which demonstrates the feasibility of a BCI system for people with disabilities. The work laid theoretical and technological foundations for contemporary studies on neuroprosthetics and human-machine interfaces. Ángel Leonardo Valdivieso Caraguay, Juan Pablo Vásconez, Lorena Isabel Barona López, and Marco E. Benalcázar proposed Deep Q-Networks (DQN) and Double Deep Q-Networks (DDQN) based on reinforcement learning for an EMG gesture recognition system [6]. The authors collected EMG signals from forearm muscles and deep reinforcement learning models for gesture classification. The DDQN architecture was proposed to solve anomalies and computation problems frequently found in the standard DQN approaches. The research findings indicated that DDQN obtained better gesture classification accuracy, more stability, and faster convergence than DQN. The study provided evidence of the potential of reinforcement learning methods for improving intelligent prosthetic hand systems and adaptive gesture recognition applications. Pablo Arnillas Gómez, Laura Montesino San Martín, and Eduardo Castellanos Ramos, “Dataset for classification of electromyographic typing gestures for neurotechnological human-machine interfaces. "The study focused on collecting multi-channel EMG data while people typed and creating a publicly available benchmark dataset. The dataset included labeled gesture data and steps for preparing it for machine learning applications. The results of the research showed that the dataset contains reliable, high-quality EMG recordings suitable for training and testing gesture classification algorithms. As the authors point out, this dataset will contribute to the reproducibility and standardization of EMG-based HMI research. Iris Kyranou, Katarzyna Szymaniak, and Kianoush Nazarpour created a dataset of EMG for gesture recognition with arm translation [8]. The study demonstrated the impact of dynamic arm movements and arm postures on EMG signal patterns and classification performance. The authors collected sEMG data during several repetitions of gestures in different arm translation scenarios. The research findings showed that arm translation significantly affected the accuracy of gesture recognition and the robustness of traditional classifiers. The study concluded that prosthetic and wearable systems in the real world require adaptive, position-invariant algorithms. Francisco J. Castillo-García, Eduardo López-Larraz, and Nicolás Birbaumer acquired the sEMG data from healthy subjects performing hand movement tasks [9]. The main aim of this study was to record forearm muscle activity associated with different hand gestures and movements. The dataset has been generated to support research in machine learning, rehabilitation engineering, and prosthetic control applications. Experimental results show that the sEMG signals were stable, reliable, and useful for gesture recognition analysis. The authors concluded that the dataset can be used to evaluate EMG classification algorithms and rehabilitation systems. Mingxing Li, Michalis Xiloyannis, Dario Farina. Neuromuscular constraints in a real-time EMG-based controller for a prosthetic hand [10]. The system integrates EMG signal acquisition with biomechanical and neuromuscular modelling to enable more natural movements of prosthetic hands. The authors used a real-time control strategy to improve finger coordination and smoothness. The results of the research indicated that the proposed controller generated stable prosthetic operation in real time with biologically realistic movement patterns. The study indicated that neuromuscular constraints can greatly improve usability, coordination, and natural behaviour in prosthetic hands. Alessandro Scano, Dario Farina, and Christian Cipriani have presented an online. EMG-based decoding system for the control of a prosthetic hand that is able to simultaneously recognize hand postures and individual finger forces [11]. The study has utilized online EMG acquisition and multi-output decoding algorithms to estimate hand gestures and finger force levels in real time. The research results indicate that the proposed framework accurately decodes posture and force information simultaneously. The authors demonstrated that EMG can be decoded in real time to improve dexterity and fine motor control in advanced prosthetic hand systems. Md. Al Mehedi Hasan, Mohammad Hasan Imam, and S. M. Shovan developed an edge AI-based dynamic EMG hand gesture recognition system for human-robot interaction applications [12]. In this work, the authors combine EMG signal processing, edge computing, and deep learning models for low-latency gesture recognition. The authors used lightweight AI models for embedded and wearable devices. The experimental results have shown that the proposed Edge AI framework achieved high gesture classification accuracy with rapid, real-time performance and minimal computational delay. The study concludes that EMG systems with edge AI are suitable for wearable robotics, smart prostheses and human-robot interaction. In summary, the studies reviewed highlight recent advances in EMG-based gesture recognition, prosthetic control, and neurotechnological systems. Recent studies show that deep learning, reinforcement learning, adaptive modelling, edge AI and neuromuscular constraints enhance classification accuracy, robustness, real-time operation and practical usability significantly [13, 14]. The past studies also indicate the need for publicly available benchmark datasets for reproducible research and algorithm evaluation. These advancements lead to smart prosthetic systems, rehabilitation technologies, wearable devices, and next-generation human-machine interaction platforms. Based on previous studies, the authors of this paper aim to investigate the real-time performance of the state-of-the-art in EMG signal analysis.

This study reported an EEG- and EMG-based human-machine interface for navigating mobility-assistive wheelchairs [2]. Most research on EMG signals focuses on classifying movement actions by proposing pre-processing methods and using available classifiers [23-28]. This work demonstrates accuracy over 90% in offline settings. The hybrid system combines EMG and EEG signals to control an electric wheelchair and reports a detection accuracy above 95% [17]. Some real-time prosthetic devices have been developed, but they are expensive. Recently, a method for double-channel EMG-based hand gesture recognition was proposed using an SVM classifier [29]. Hand gesture recognition using wavelet packet transformation and a hybrid CNN trained on surface EMG and accelerometer signals has been reported in the literature [30]. A design for digital filters for peak detection of surface EMG was also proposed [31]. A smart method for EMG envelope extraction and effective denoising for human-machine interfaces was reported in. Various physiological artifacts can affect EMG signals after their collection. To eliminate these artifacts, different preprocessing methods were applied. Analyzing EMG data offline requires varying amounts of computational time and expensive hardware and software for each preprocessing method. The main challenges with BCI technology include electrode placement for EEG data acquisition and analyzing large amounts of data to infer brain states [32, 33]. The authors suggest an alternative where signals from hand muscles control external devices.

Based on the literature survey, it is observed that an EMG-based man-machine interface is a practical option for disabled individuals who cannot walk or move, especially when compared to EEG-based control of an electric wheelchair. In this paper, an EMG-based man-machine interface is presented, along with a low-cost solution developed using off-the-shelf hardware [35, 36]. The authors propose an EMG-controlled system on a single-chip platform for a compact, efficient, and responsive control interface. Capturing biopotential signals is a key challenge in building a user interface. It can be expensive to build a whole neurorehabilitation device on top of amplifiers that boost weak signals [35, 36]. In this paper, a low-cost device is studied and presented, and the potential of a human-computer interaction system for neurorehabilitation is discussed. It is about a real-time system for controlling a car using EMG signals. The authors propose an HMI (Human-Machine Interface) based on EMG (Electromyogram) signals [35, 36]. Electromyography is a useful tool for the study of human movement, the understanding of neuromuscular mechanisms, and the diagnosis of neuromuscular disorders. This paper focuses on using EMG signals for machine control through wireless command transmission. The EMG signal contains various muscle responses. Processing the EMG signal is challenging because the transient muscle response has an unknown arrival time, duration, and shape. The authors tested an EMG-based HMI system on volunteers, including both male and female subjects, and reported the results as accuracy [35, 36]. The authors have developed a real-time EMG data-acquisition system using the BioAMP EXP pill, which is an inexpensive amplifier to acquire the bio-signal [37]. The bio-signal amplifier from g.tec Austria is very expensive and difficult to buy. One can build the system by combining different boards, such as the Bio-AMP EXP pill, AD8232 [38], and Arduino Uno [39], since the equipment is expensive. There are several characteristics of electromyographic signals that one should understand in order to properly acquire and process them [19].

These properties depend on the muscle, the electrode location, and the physiological state. An EMG signal is recorded by a surface-mounted AgCl electrode with gel. The pre-processing algorithm designed for the Arduino Uno detects the EMG signal envelope. The instrumentation amplifiers, BioAMP EXP pill, and AD8232 EMG module record the EMG signal, achieving a better common-mode rejection ratio and amplifier gain. The filtering, rectification, and smoothing algorithm is developed and implemented on the Arduino hardware. By analyzing the envelope of the dual-channel EMG signal, a logical command is generated to control an external robotic car. This command is sent via Bluetooth (ESP32) from a wireless communication system to control the remote-controlled car. A summary of the general characteristics of surface EMG (sEMG) signals is presented in Table 1.

Table 1: Characteristics of the EMG signal

Component	Requirement
Electrode type	Surface (non-invasive) or needle (invasive)
Amplifier gain	100x to 10,000x
Filters	Bandpass (10-500 Hz), Notch (50/60 Hz)
ADC resolution	Minimum 12-bit; preferably 16-bit or higher
Sampling rate	At least 1 kHz (typically 5-6 kHz)

Whether the proposed algorithm can be applied in real time and whether the available devices support real-time preprocessing are additional challenging aspects. Keeping the above facts in mind, the Arduino UNO-based signal preprocessing methods are being developed for EMG signal analysis, including filtering and envelope smoothing of the EMG pulse generated by hand muscle contraction. This paper also used the digital IIR filter to analyze the incoming EMG signal to retrieve the command to control the external car [31]. The incoming EMG signal is a random signal corrupted by various noise sources [25]. To remove the artifacts, advanced signal preprocessing approaches were implemented for offline analysis on a computer or server. The objective of this paper is to develop a low-cost bio-signal acquisition system and feasible preprocessing methods that operate in real time with minimal computational time and cost. A band-pass filter is being developed in MATLAB, and its coefficients are stored on the Arduino UNO [25]. The incoming EMG signal is passed through the filter coefficients to clean it. A further smoothing filter is applied to dual-channel EMG signals coming from both hands, and the resulting signal is then passed through a developed algorithm for logical command or decision.

Furthermore, this paper focuses on on-chip real-time processing of EMG signals to develop EMG-based HMI systems, eliminating the need for an external computer or server. As researchers, one can face many challenges in this work. The primary constraint for this work is to utilize a low-computational algorithm for processing the EMG signal, which also requires minimal power and memory consumption. IIR filters are used for pre-processing. The frequency and phase response of the filters are studied in MATLAB. Here, the Electromyogram (EMG) signals are analyzed and visualized for offline characterization using the MATLAB DSP (Digital Signal Processing) Toolbox [25]. The toolbox contains essential preprocessing and filtering functions to efficiently remove noise and physiological artifacts common in EMG signals. Clean EMG signals can be obtained by using signal-processing techniques, particularly bandpass and notch filtering, to remove unwanted signals. Moreover, the toolbox provides specialized features for signal quantification and feature extraction, enhancing the accuracy of muscle activity interpretation. The desired IIR filter for the desired frequency band is developed in MATLAB, and its coefficients are used in real time to process the incoming EMG signals, which are stored in an Arduino Uno. This approach provides a reliable and efficient structure for the measurement of EMG signals in clinical and biomedical research. The classification techniques to identify hand gestures can be evaluated for their effectiveness. The hardware implementation of the classification algorithm is a more complex issue that needs to operate in real time on a computer or other advanced device. The authors are trained to use the Arduino Uno. Based on the envelopes of EMG signals of both hands, logical commands can be generated for the operation of the external robotic car [30]. Also, the work proposed in this paper is based on the recognition of hand gesture movements in dual-channel EMG signals without the use of any classifier [30]. Also, a logic based on the envelopes of both EMG channels is proposed to control in real time the movement of a robotic car using a dual-channel EMG signal. Based on the envelopes of dual-channel EMG signals, commands such as Backwards (BB), Forward (FF), Stop (S), and Right Rotation (RR) are generated by examining the envelopes of dual-channel EMG signals in real time [40, 41].

This paper is divided into sections that show the full development of a real-time controller for the external car. It covers EMG data acquisition and system integration and concludes with a discussion of future work. Section II describes offline datasets, and Section III describes the EMG data-acquisition system developed. Section IV describes the methodology applied to the offline analysis of available datasets. Section V presents the results and discussions over offline datasets. Section VI explains the proposed method for real-time EMG data analysis. Finally, Section IV details how the system integrates to manage the incoming EMG signal and determine the command to control the external car in a lab setting.

2. Offline Datasets

The research uses two sEMG datasets that include 17 hand gesture classes. These datasets were collected from 76 healthy participants with different acquisition systems. The combined dataset is a valuable resource for hand gesture recognition, testing machine learning, and developing deep learning models for human-computer interaction applications.

Dataset-I: The sEMG dataset (ckwc76xr2z-2) is a multi-channel surface electromyography dataset recorded from 40 healthy participants (20 males and 20 females) [42]. Participants performed 10 different hand gestures. Muscle activity was recorded by a BIOPAC MP36 medical-grade EMG system with 4-channel surface bipolar electrodes on forearm muscles. The system recorded the electrical signals at a sampling rate of 500 Hz, and each gesture lasted approximately 3 seconds. Participants completed 5 complete repetitive cycles of all 10 gestures, resulting in 2,000 total gesture trials. The dataset includes these 10 hand gestures: Rest/Neutral State/Wrist Extension/Wrist Flexion/Ulnar Deviation/Radial Deviation/Grip (Fist Clenching)/Finger Abduction/Finger Adduction/Supination/Pronation

Dataset II: The UCI EMG Data for Gestures is a raw electromyography dataset for gesture recognition, collected from 36 human subjects using a MYO Thalmic bracelet, a consumer-grade wearable device [43]. The device contains 8 equally spaced non-invasive sEMG sensors placed around the middle of the forearm with a sampling frequency of 200 Hz. Each gesture recording lasted 3 seconds, followed by a 3-second pause to allow the muscles to return to their rest position. Each subject performed 2 complete series of 6-7 gestures, resulting in approximately 504 total gesture trials. The dataset includes these 7 hand gesture classes: Unmarked- No gesture performed / Hand at Rest / Fist Clenched / Wrist Flexion / Wrist Extension / Radial Deviation / Ulnar Deviation / Extended Palm. The data is stored in CSV-like format (tab-separated text files) organized in subject-specific folders with timestamped files. Each row contains 9 columns: time (milliseconds), channels 1-8 (EMG signal values), and class (gesture label 0-7). Each file is approximately 4.7 MB.

Keeping the dataset's objective in mind, the authors developed an EMG data-acquisition system to generate EMG signals from the left and right wrists, individually and simultaneously, creating a human-machine interface that controls a robotic car without using brain signals. The objective of the above database is to develop EMG signal-processing methods and to evaluate the effectiveness of the classification algorithm. After analyzing the above dataset using the proposed methods, the authors considered evaluating the performance of the existing algorithm in real-time control of the robotic car without a lengthy machine-learning process by collecting ample data for each task. Based on the envelopes of channels 1 and 2, a threshold-based technique is proposed to select the command for controlling the robotic car's movement. The most commonly used approaches are the following:

Options I: Raw → DC remove → Notch → Band-pass → Segment + Hudgins features → Classify

Option II: Raw → DC remove → Notch → Band-pass → Rectify → Envelope → Segment + Envelope features → Classify

Option I is the most important technique, and it is an old one for analyzing the EMG signal. Option II is also a more popular, modern approach for extracting features from the recoded EMG signal to control the prosthetic hand. The gap analysis between offline and online gesture detection in the existing state of the art has not yet been discussed in the context of EMG data analysis. This paper contributes in many ways, including a gap analysis, control of a robotic car using hand gestures to generate real-time commands, low-cost hardware implementations for EMG data acquisition, embedded edge computing on a microcontroller to make decisions, and ESP32-based wireless pairing.

3. Developed EMG Data Acquisition

The dataset is available online from various labs worldwide for the development of gesture recognition using machine learning algorithms. EMG bands equipped with EMG-sensing sensors are costly. The experimental setup for acquiring Electromyographic (EMG) signals in this work utilises the BioAmp EXG Pill and AD8232, a compact analogue front-end designed for biopotential measurements, interfaced with an Arduino Uno development board running at 3.3V or 5V. The setup is described in Figure 1. This setup acquires real-time responses to muscle activity and serves as the foundation for implementing signal processing methods and control algorithms, as explained in subsequent sections. The BioAmp EXG pill and the AD8232 are available at a low cost. This study also reflects the feasibility of developing a real-time system using these available low-cost amplifiers for biopotential measurement. Discrete hardware, which is depicted in Table 2, is integrated for EMG signal acquisition and processing. The analog-to-digital converter of the Arduino UNO microcontroller is utilized to provide a stable sampling platform for signal acquisition. It is well known that the first stage in acquiring biosignals is a differential amplifier, which eliminates common-mode noise and amplifies the weak EMG signal, making it suitable for further processing to identify the hand gesture. The complete list of components integrated to develop the EMG data acquisition system for hand gesture recognition is presented in Table 2. Each module and its function are described in detail in the following subsections.

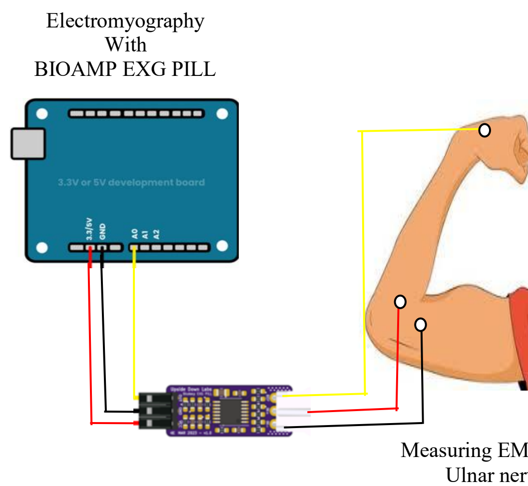


Fig. 1 General EMG signal acquisition setup

Table 2. List of hardware

S.No.	Name of Module	Specification
1	BioAmp EXG Pill	Powerful Analogue-Front-End (AFE) biopotential signal-acquisition board
2	AD8232 EMG module	Fully integrated single-lead ECG and EMG amplifier
3	Arduino UNO	ADC and DSP perform for EMG
4	ESP32 Dev Module	Bluetooth Transmitter and Receiver
5	DC motor	RC car
6	Ln298	DC motor driver

3.1. Bio Amp EXP Pill Module

It is known that biopotential signals are in the microvolt range. Before recording, these signals must be amplified and stored. Most laboratories use a g.tec amplifier to record high-resolution biopotential signals. However, the amplifier is expensive, and many labs do not have the budget for it. The authors of this paper also aim to evaluate low-cost amplifiers available on the market to determine whether they can successfully capture biopotential signals. The Bio-Amp EXG Pill is a novel Analogue Front-End (AFE) module for the acquisition of biopotential signals, such as Electromyography (EMG), Electrocardiography (ECG), and Electroencephalography (EEG). It is compact, low-cost and efficient. It can be easily interfaced to any Microcontroller Unit (MCU) or Single-Board Computer (SBC) with an Analogue-To-Digital Converter (ADC). The module contains a general-purpose JFET quad operational amplifier IC (TL084), which is known for its low noise and high input impedance. This makes it excellent for amplifying biopotential signals. Figure 2 shows the pinout configuration of the BioAmp EXG Pill, and Table 3 summarises the technical specs. The recorded dataset has used a commercially available arm sensor for the multichannel EMG signal.

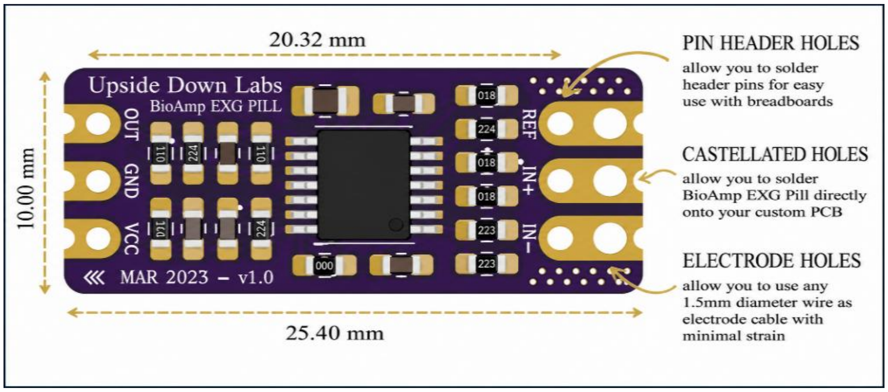


Fig. 2 BioAmp EXP Pill [37]

Table 3. Specification of BioAmp EXP pill

Parameter	Specification
Operating Voltage	5 V
Input Impedance	10^{12} ohm
Compatible Hardware	Any development board with an ADC (e.g., Arduino UNO & Nano, Adafruit QtPy, STM32 Blue Pill
Bio-Potentials	EMG, ECG, EOG, EEG (configurable band-pass; by default, configured for EEG & EOG)
No. of Channels	1
Electrodes	2 or 3 (By default configured for 3 electrodes)
Dimensions	25.4 × 10 mm

In this paper, the authors have developed their own data acquisition system to control the robotic car. This paper also contributes to single-channel EMG recordings from the left and right arms. The authors proposed integrating discrete components to develop a man-machine interface for controlling external hardware. The existing methods primarily collected multichannel EMG signals for different hand gestures. This paper considers only single channels from both hands to develop novel preprocessing methods.

3.1.1. AD8232

The AD8232 is a compact, integrated signal conditioning module designed primarily for acquiring biopotential signals, such as ECG and EMG. In this work, the AD8232 is also used to acquire EMG signals from the second channel, providing the initial amplification and filtering of the raw muscle signals before they are digitised by the microcontroller. This amplifier is well-suited for wearable or low-power applications due to its low noise, high input impedance, and integrated high-pass and low-pass filter. It is basic and low-cost, making the extraction of clean EMG signals from noisy environments ideal for real-time biomedical applications. The Arduino Uno is a general-purpose microcontroller board that uses an Atmel ATmega32 microcontroller.

Table 4. Depicts the specification of the AD8232

Parameter	Rating
Supply Voltage	3.6 V
Output Short-Circuit Current Duration	Indefinite
Maximum Voltage, Any Terminal	+Vs + 0.3 V
Minimum Voltage, Any Terminal	-0.3 V
Storage Temperature Range	-65°C to +125°C
Operating Temperature Range	-40°C to +85°C
Maximum Junction Temperature	140°C

The pin configuration of the AD8232 module is shown in Figure 3 and its key specifications are listed in Table 4 [10].

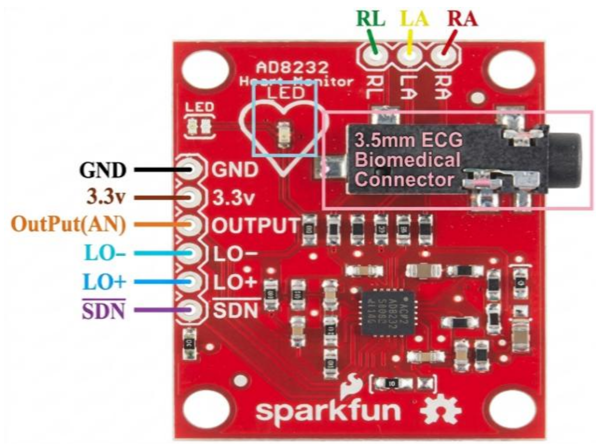


Fig. 3 Pin out of AD8232

3.1.2. Arduino UNO

The Arduino Uno is a general-purpose development board that uses an Atmel ATmega32 microcontroller. An Arduino board communicates with external devices, such as computers, sensors, and LEDs. By adding a communication interface, one can ensure that Arduino can receive and transfer data to external devices, thereby generating the required output. Its key specifications are listed in Table 5.

Table 5. Depicts the specifications of the Arduino Uno

Category	Specification
Microcontroller	ATmega328P
Operating Voltage	5V
Analogue Input Pins	6 (A0 to A5)
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB is used by the bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
USB Interface	Type-B USB connector
Communication Protocols	UART, SPI, I2C

4. Methodology for EMG Signal Processing

According to the literature survey, the most prominent methods for EMG signal analysis are Option I and Option II for offline gesture recognition. The effectiveness of the existing approach for real-time implementation is important to evaluate, given the lab's limited resources. Real-time implementation of the above approaches requires validation and modifications tailored to available resources. The proposed method, a modified version of approach II, has several stages, as shown in Figure 4, starting with electrode placement and ending with the control interface. A two-channel EMG recording setup is used with differential surface EMG (sEMG) electrodes, both referenced to the dorsal hand surface.

- EMG Data Acquisition: It consists of an amplifier, electrode placement, sampling the analog signal, and storing it in a digital format in a computer or microcontroller.
- Pre-processing: This stage filters out noise, rectifies the signal, and applies smoothing to get a clean incoming waveform of the EMG signal.
- Envelope Detection & Threshold Decision: The incoming muscle activation is detected through the envelope of the EMG pulse. A further thresholding technique is applied to extract muscle activation movements.
- Control Interface Unit: The wireless communication is established between Arduino Uno and the external robotic car using ESP32 modules.

Figure 16 shows the recorded EMG signal obtained using the sophisticated integrated data acquisition system discussed in Section II. The raw EMG data obtained in our setup are shown in this figure. It is then expertly processed through a number of pre-processing steps, such as filtering, rectification, and smoothing, to clearly visualize the envelope of EMG activity. To minimise noise and to successfully isolate the important frequency components of the EMG signal, a complex four-stage biquad Infinite-Impulse-Response (IIR) Butterworth filter is used in cascade. The IIR filter was chosen as the best alternative for real-time signal processing in embedded systems based on its good amplitude response and memory efficiency. “This accuracy means our results are reliable and meaningful.

In the literature, many preprocessing methods have been applied and reported, ranging from digital filtering to advanced signal processing techniques such as empirical mode decomposition, wavelet transforms, and wavelet packet decompositions. Advanced signal processing methods require powerful digital computers and specialized software to run the algorithms. This study considers a biquad IIR digital filter in cascade, so the total filter order is 8 to select the target frequency band. The filtered band-pass EMG signal passes through a rectification and smoothing filter to visualize the envelope of EMG signals related to hand gesture movement.

As illustrated in Figure 7, the electrodes are placed on both hands in order to acquire the signals from both hand gestures. These electrodes carry signals to two amplification modules: the BioAmp EXP pill and the AD8232. The raw EMG signal is amplified by an instrumentation amplifier located in these two boards to improve the signal-to-noise ratio for further processing. The signal is amplified and passed through a signal conditioning unit, then filtered by a bandpass filter set between 75 Hz and 150 Hz. The EMG signal appears broad in a frequency range of 0-450 Hz. The sEMG signals measured by the commonly used sensors are in the frequency range of 0-400 Hz, which depends on several factors, such as the electrode spacing and the thickness of the fatty tissue between the skin and the underlying muscle [7]. Then, after filtering, the signal is further rectified and smoothed by using an EMG smoothing filter (usually a low-pass filter) before a thresholding step to infer the hand gesture. The detected envelope describes the general shape of the EMG signal. It represents the muscle activation pulse over time.

This smoothed envelope is then fed to a threshold decision block. The threshold decision block compares the smoothed envelope to a preset threshold level to determine if the muscle has been sufficiently activated. The magnitude and phase response of a 4th-order cascade IIR EMG filter is shown in Figure 6. The filter removes high-frequency noise and low-frequency motion artifacts and captures the major frequency content of voluntary muscle contraction in the frequency band of 75 Hz to 150 Hz. It's a 4-stage Butterworth IIR bi-quad filter cascade. This will provide better frequency isolation and a sharper roll-off. This configuration enables all signal processing to be implemented on a single microcontroller chip, which is useful for the design of small, efficient biomedical devices.

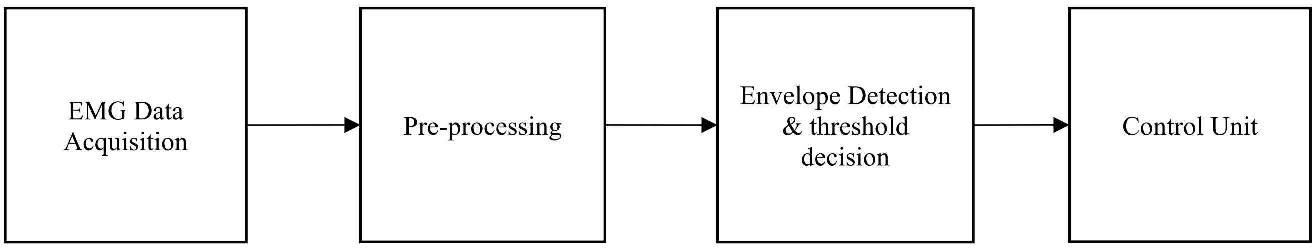


Fig. 4 Block diagram of the proposed methodology of real-time control of the robotic car

Figure 17 shows the filtered signal. The filtered EMG signals are then passed through an envelope detector. The two main steps in this process are the smoothing and the rectification. The signal swings above and below zero, with the energy being distributed between the two peaks. This is the point where the filtered EMG signal, which varies between positive and negative values, is transformed into an all-positive waveform. This is achieved by the absolute value of the filtered EMG. The EMG signal converts the signal energy to a positive value. The second step in the process involves using an envelope detector. This process smooths the high-frequency variations in the rectified signal, yielding a clean, continuous envelope. It is often performed using a moving average or low-pass filter. The inferred command is used to control the external robotic car with the smoothed dual-channel EMG signals. In this study, the robotic car is primarily considered, but it can be extended to other applications.

4.1. Offline Analysis Methodology

The real-time controller developed in this work relies on a threshold rule applied to the dual-channel EMG envelope rather than on a trained classifier. To establish how much discriminative information the surface EMG (sEMG) envelope actually carries and to justify the feature set used on the embedded device, an offline study was conducted on two publicly available sEMG databases. The first is the Lobov gesture database, which contains recordings from 36 participants performing seven basic hand states: rest, fist, wrist flexion, wrist extension, radial deviation, ulnar deviation and extended palm. The second is the WyoFlex forearm database, a four-channel recording sampled at 1 kHz that captures flexion and extension of both the right and the left forearm. The two databases were chosen because, together, they address the two questions that matter for the proposed interface: whether individual hand gestures can be separated and whether activity on the left and right arms can be distinguished, which underlies the forward, backward, and rotation mapping used in real time. The complete offline pipeline is shown in Figure 13. Both databases were processed using the same method to ensure similar results. Each channel was first de-trended to remove the DC offset. It was then filtered with a 60 Hz notch to reduce mains interference. Finally, it was band-pass filtered between 20 and 450 Hz to retain the main sEMG energy. The filtered signal was full-wave rectified and smoothed with a 50 ms moving-average window to obtain the envelope, which is the same quantity the embedded firmware extracts in real time. From each channel, Six Hudgins time-domain features: Mean Absolute Value (MAV), Root-Mean-Square (RMS), Waveform Length (WL), Zero Crossings (ZC), Slope-Sign Changes (SSC), and Variance (VAR) are calculated. This process resulted in a 24-dimensional descriptor for the four-channel WyoFlex recordings. All features using a z-score transform are standardized before classification. Four standard classifiers were trained and compared: Linear Discriminant Analysis (LDA), a Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and a Random Forest (RF) with 150 trees. For the Lobov database, two feature configurations, denoted Option 1 and Option 2, are reported, along with two validation protocols. The windowed protocol pools short analysis windows from the recording. In contrast, the more conservative trial-wise protocol keeps complete repetitions separate. This means that windows from the same contraction do not appear in both the training and test set. The difference between the two protocols shows how optimistic the windowed estimate is. For the WyoFlex database, five-fold cross-validation was used. It is also conducted a separate subject-grouped test to see if one could identify the active forearm (left or right) just from the envelope. The performance in terms of accuracy, precision, recall, and F1 score, backed by confusion matrices, is presented. It is examined how well flexion and extension classes could be separated using a two-sample t-test for each feature.

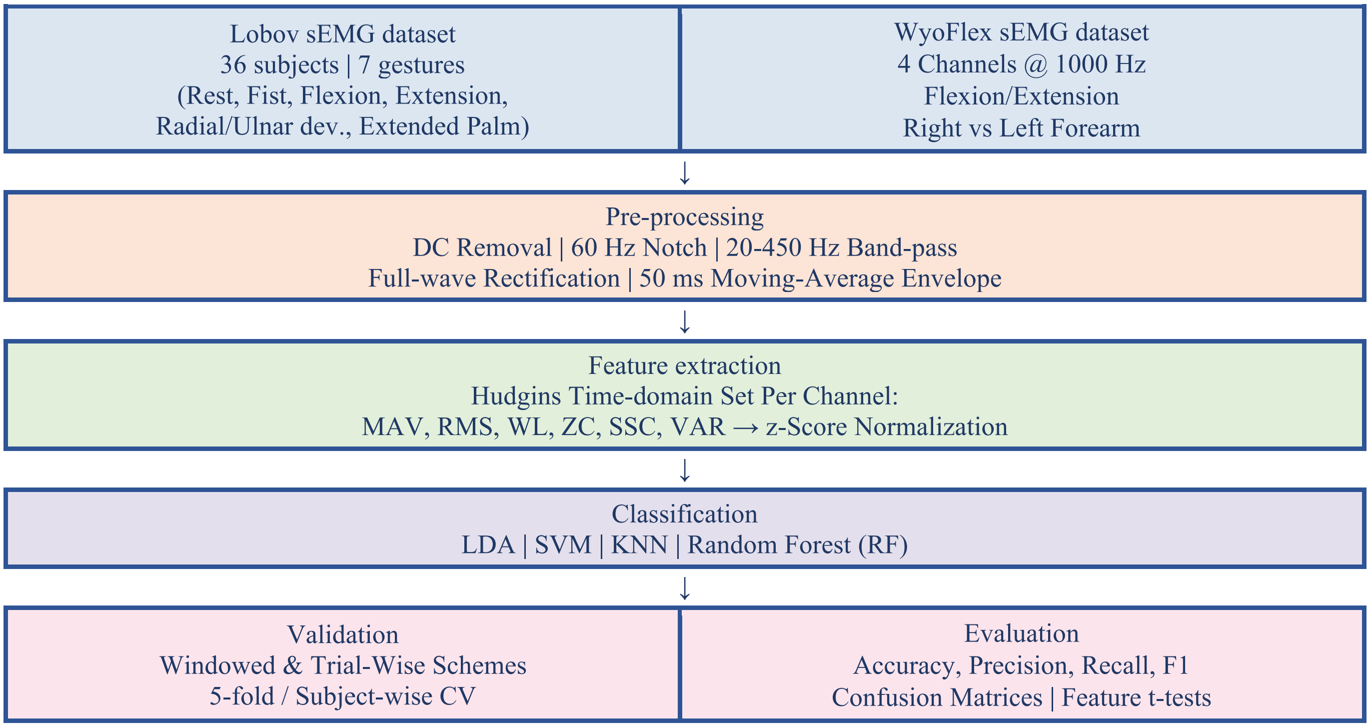


Fig. 5 Detailed offline analysis pipeline applied to the Lobov and WyoFlex sEMG databases

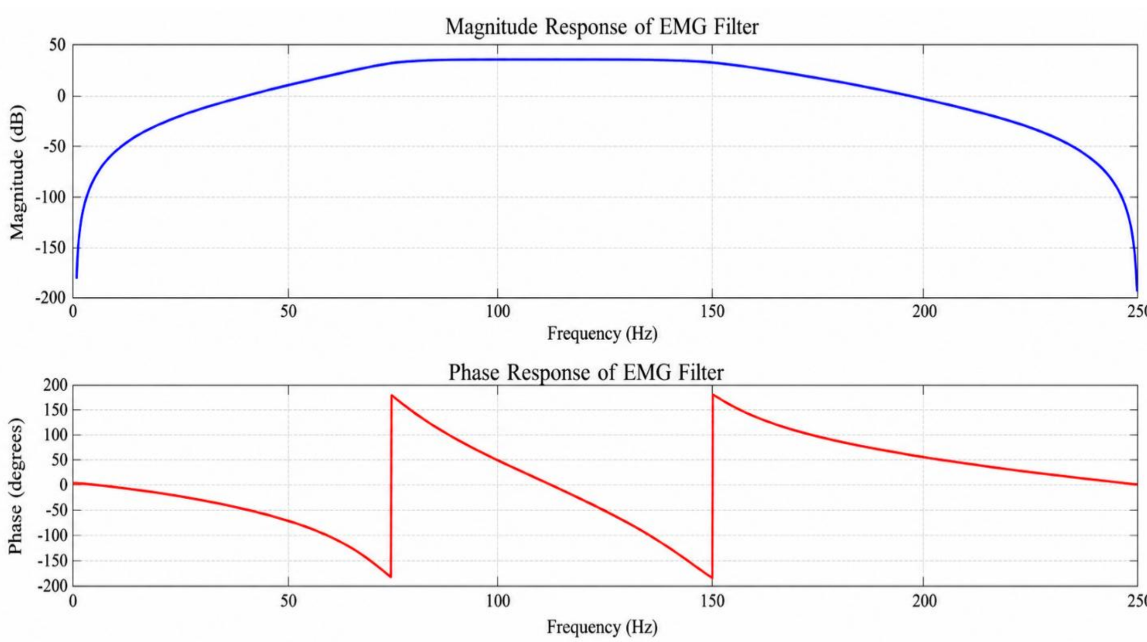


Fig. 6 Depicts the magnitude and phase response of the band-pass EMG IIR filter

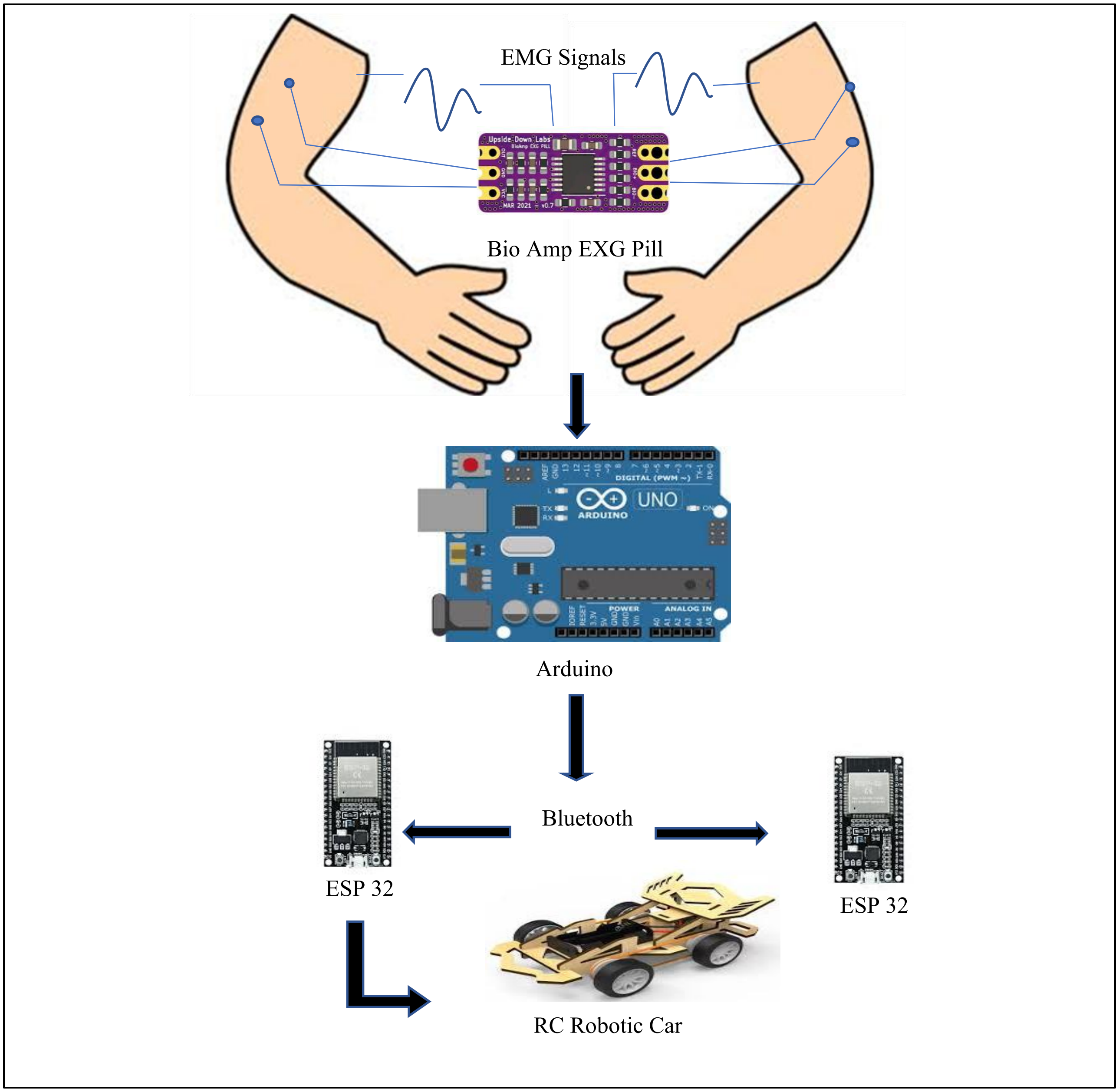


Fig. 7 Complete system integration

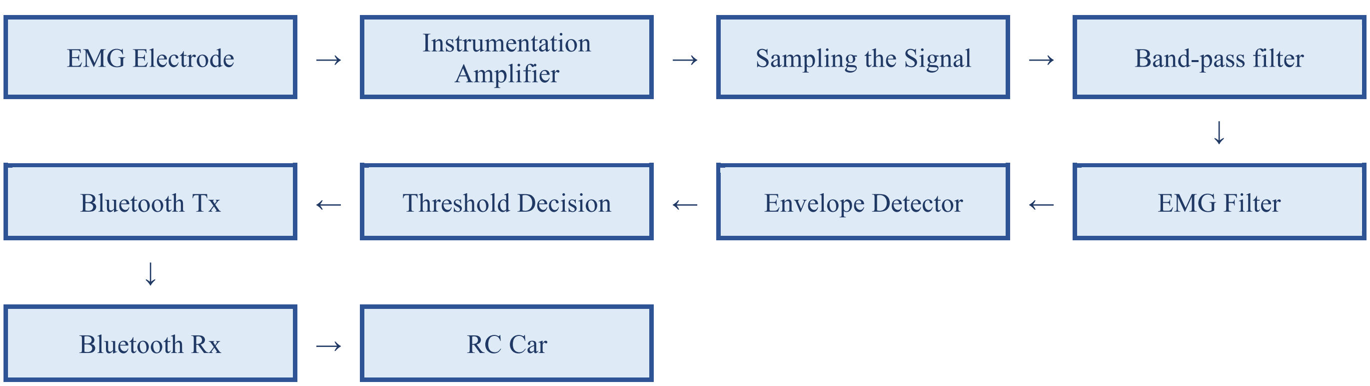


Fig. 8 Block diagram of an embedded system to make a complete real-time system

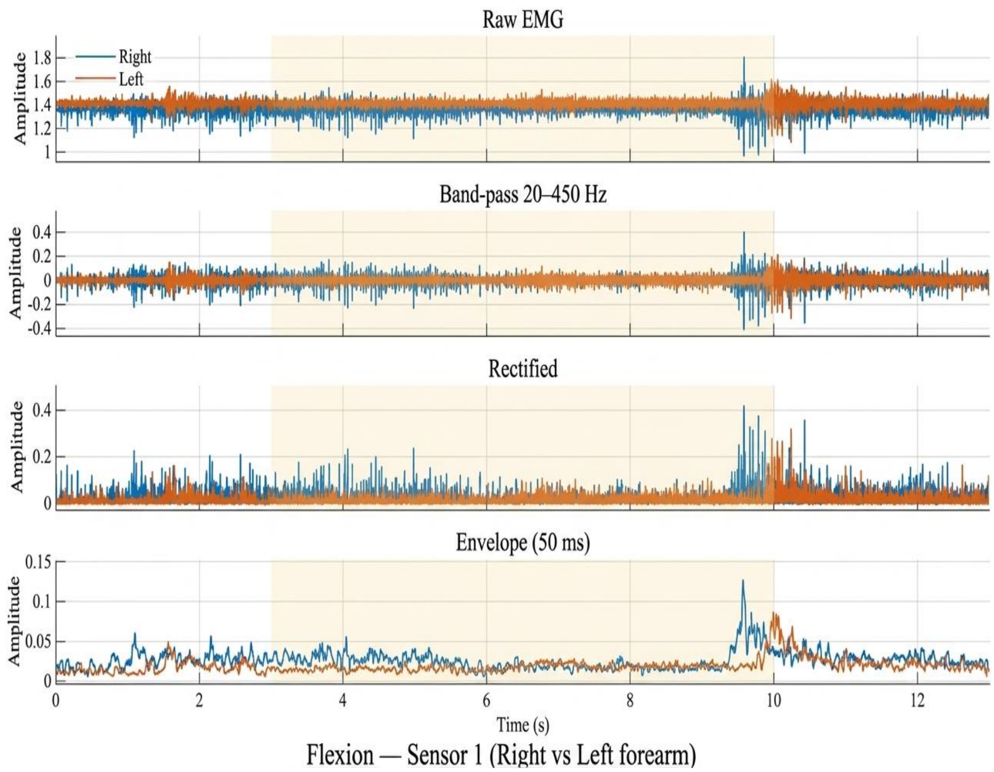


Fig. 9 Raw, band-pass filtered, rectified and 50 ms envelope stages for a representative flexion trial (right vs left forearm)

5. Results and Discussion

Figure 9 shows the four processing stages of a typical flexion trial from the two forearms. The raw trace is on a common baseline and is mainly affected by offset and broadband noise. After band-pass filtering, the burst linked to the contraction becomes visible. Following rectification and envelope smoothing, the activation looks like a single slow bump. Its timing and amplitude clearly indicate the active window, which is shaded. This confirms that the envelope contains the necessary information for detecting a contraction. This information is exactly what the embedded threshold logic uses in real time.

5.1. Lobov Gesture Database

Table 6 summarizes the classification performances on the Lobov database for both feature options and both validation protocols. Using the windowed protocol, all four classifiers achieved an accuracy above 87%, with the SVM reaching 94.75% and the random forest 94.66% for Option 2. A more rigid trial-wise protocol sees the same models drop down to ~81-84%, a much more realistic estimate of day-to-day use. For example, the KNN trial-wise accuracy increases from 72.11% to 77.25%, and this increase is statistically significant for all classifiers except the LDA, as shown in Table 6. Option 2 is always better than Option 1. The random forest gives the best trial-wise accuracy (84.43%) of the classifiers; the LDA is competitive and is the cheapest to deploy. The per-subject spread can be seen in Figure 10. The error bars (±1 SD) clearly show that the windowed estimates are both higher and tighter than the trial-wise estimates. The trial-wise accuracy for each gesture is shown in Figure 11. As shown, rest and fist are almost perfectly recognized (96-99%), while radial deviation is the weakest class (down to ~55-72%). The concentration of residual error depends on the classifier because the pattern of activation overlaps with the neighboring wrist movements. The confusion matrices in Figure 12 show that most of the residual error is concentrated between these neighboring wrists. The gestures are more concentrated in some classes. They are not equally distributed among all classes. For the proposed interface, the result is encouraging because the gestures actually used for control, namely a firm contraction of one forearm against the other, correspond to the most separable classes rather than the more confusable ones.

Table 6. Classification performance on the Lobov database (36 subjects, mean over subjects)

Config.	CV scheme	Classifier	Accuracy (% mean ± SD)	Precision (%)	Recall (%)	F1 (%)
Option 1	Windowed	LDA	91.37 ± 3.98	91.48	91.35	91.33
Option 1	Windowed	SVM	93.83 ± 3.3	93.86	93.81	93.82
Option 1	Windowed	KNN	87.96 ± 4.39	88.12	87.92	87.87

Option 1	Windowed	RF	93.19 ± 3.79	93.19	93.18	93.17
Option 1	Trial	LDA	83.16 ± 6.92	83.33	83.13	82.81
Option 1	Trial	SVM	79.03 ± 7.36	79.18	79	78.66
Option 1	Trial	KNN	72.11 ± 7.46	71.74	72.07	71.3
Option 1	Trial	RF	82.87 ± 7.57	82.83	82.85	82.55
Option 2	Windowed	LDA	91.46 ± 4.15	91.62	91.43	91.42
Option 2	Windowed	SVM	94.75 ± 2.94	94.79	94.74	94.75
Option 2	Windowed	KNN	91.1 ± 3.87	91.2	91.07	91.05
Option 2	Windowed	RF	94.66 ± 3.36	94.66	94.65	94.65
Option 2	Trial	LDA	83.33 ± 6.9	83.59	83.29	83.04
Option 2	Trial	SVM	80.97 ± 7.45	81.18	80.94	80.58
Option 2	Trial	KNN	77.25 ± 7.63	77.15	77.2	76.69
Option 2	Trial	RF	84.43 ± 7.37	84.39	84.4	84.1

Table 7. Per-activity accuracy on the Lobov database (Option 2, trial-wise cross-validation)

Gesture	LDA	SVM	KNN	RF
Rest	99.1	95.7	98.4	97.0
Fist	89.4	89.5	86.2	92.3
Flexion	75.4	73.4	70.6	77.9
Extension	84.0	86.1	81.0	87.5
Radial-dev	71.7	63.5	55.6	71.4
Ulnar-dev	82.2	79.7	73.9	82.4
Ext-palm	71.8	74.1	65.5	76.8

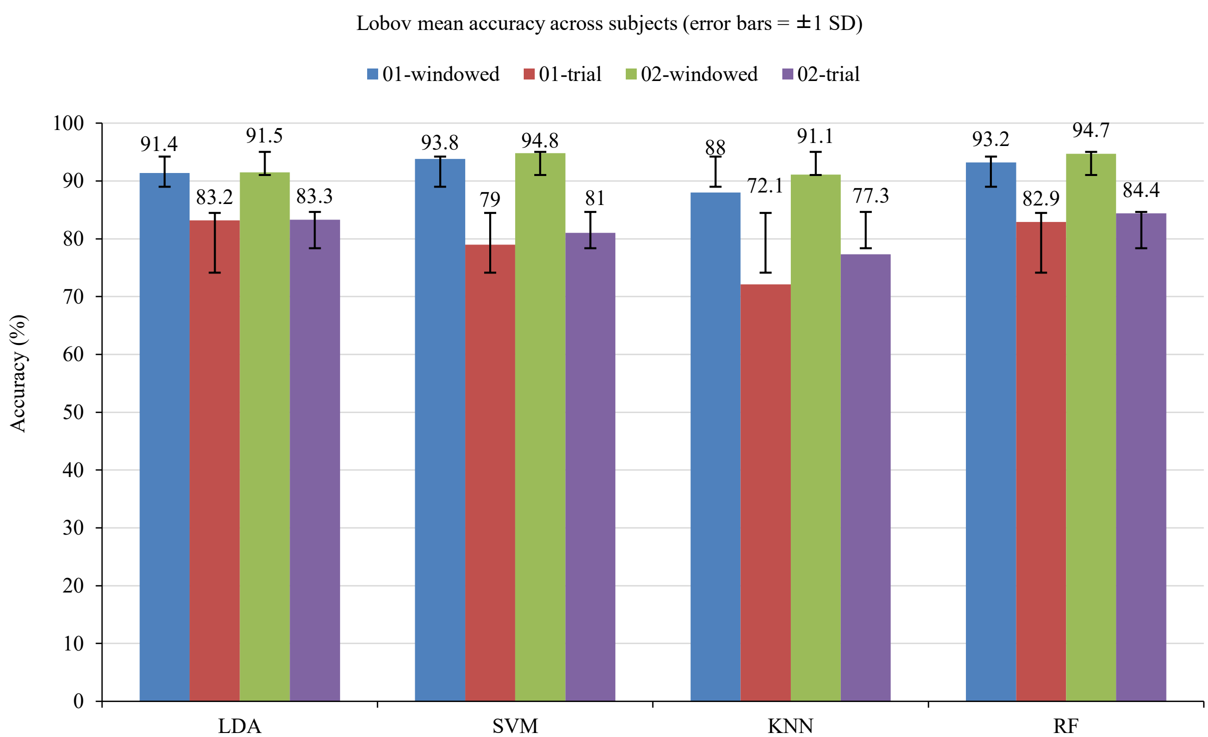


Fig. 10 Mean Lobov accuracy across subjects for each classifier and protocol (error bars ± 1 SD)

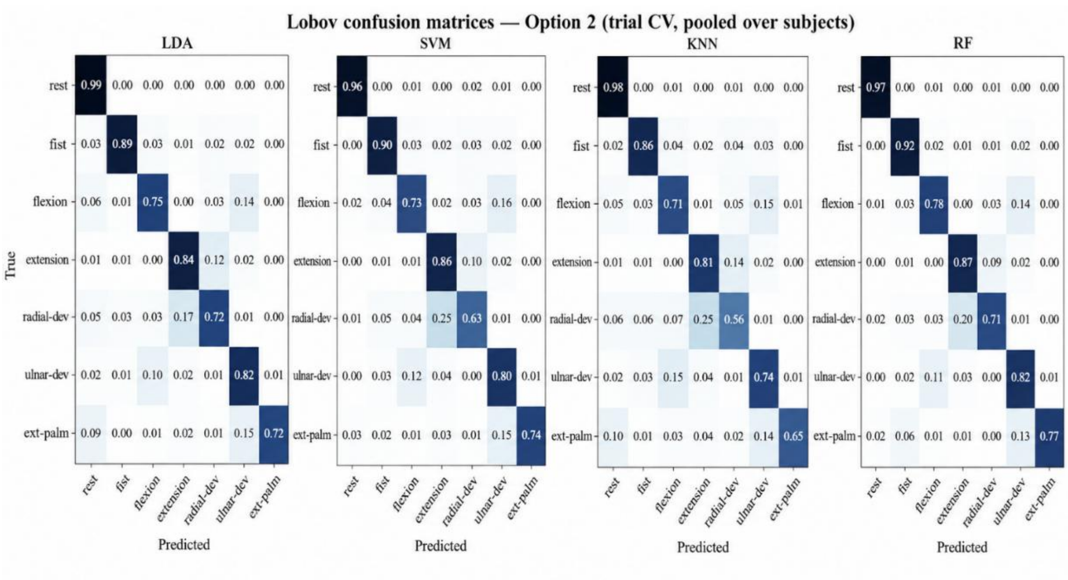


Fig. 11 Confusion matrices for the Lobov gestures (Option 2, trial-wise CV, pooled over subjects)

5.2. WyoFlex Forearm Database

Table 8. Statistical comparison of options and classifiers on the Lobov database (paired t-test, df = 35)

Comparison	t	df	p-value	Sig.	Mean diff. (pp)
Opt1_vs_Opt2 (windowed, LDA)	-0.706	35	0.4851	n.s.	0.09
Opt1_vs_Opt2 (windowed, SVM)	-5.967	35	<0.001	***	0.92
Opt1_vs_Opt2 (windowed, KNN)	-17.748	35	<0.001	***	3.14
Opt1_vs_Opt2 (windowed, RF)	-13.935	35	<0.001	***	1.47
Opt1_vs_Opt2 (trial, LDA)	-0.918	35	0.3649	n.s.	0.18
Opt1_vs_Opt2 (trial, SVM)	-9.43	35	<0.001	***	1.94
Opt1_vs_Opt2 (trial, KNN)	-20.055	35	<0.001	***	5.14
Opt1_vs_Opt2 (trial, RF)	-11.14	35	<0.001	***	1.56
LDA_vs_SVM (O2,trial)	7.235	35	<0.001	***	2.36
LDA_vs_KNN (O2,trial)	15.838	35	<0.001	***	6.08
LDA_vs_RF (O2,trial)	-2.473	35	0.0184	*	-1.09
SVM_vs_KNN (O2,trial)	12.261	35	<0.001	***	3.72
SVM_vs_RF (O2,trial)	-9.647	35	<0.001	***	-3.46
KNN_vs_RF (O2,trial)	-15.841	35	<0.001	***	-7.17

Table 9 reports the flexion and extension results for each forearm. Extension is the easier movement, recognised with F1 scores above 91% for every classifier and arm, while flexion is harder, with F1 between 70% and 92%. The random forest again performs best overall, reaching 69.76% validation accuracy on the left forearm with flexion and extension F1 scores of 92.5% and 95.3%, respectively. The overall accuracy across all classifiers is summarised in Figure 12, which shows the left and right forearms behaving very similarly, with no systematic advantage for either side.

Table 9. Flexion and extension performance on the WyoFlex database for the right and left forearms

Arm	Clf.	Val. Acc. (%)	Flex P (%)	Flex R (%)	Flex F1 (%)	Ext P (%)	Ext R (%)	Ext F1 (%)
Right	LDA	59.76	75.86	78.57	77.19	97.53	94.05	95.76
Right	SVM	66.43	91.67	78.57	84.62	100	92.86	96.3
Right	KNN	56.43	69.41	70.24	69.82	95.18	94.05	94.61
Right	RF	65.83	88.16	79.76	83.75	96.34	94.05	95.18
Left	LDA	60.12	89.71	72.62	80.26	94.87	88.1	91.36
Left	SVM	66.19	98.59	83.33	90.32	97.4	89.29	93.17
Left	KNN	56.07	79.75	75	77.3	91.86	94.05	92.94
Left	RF	69.76	97.37	88.1	92.5	94.19	96.43	95.29

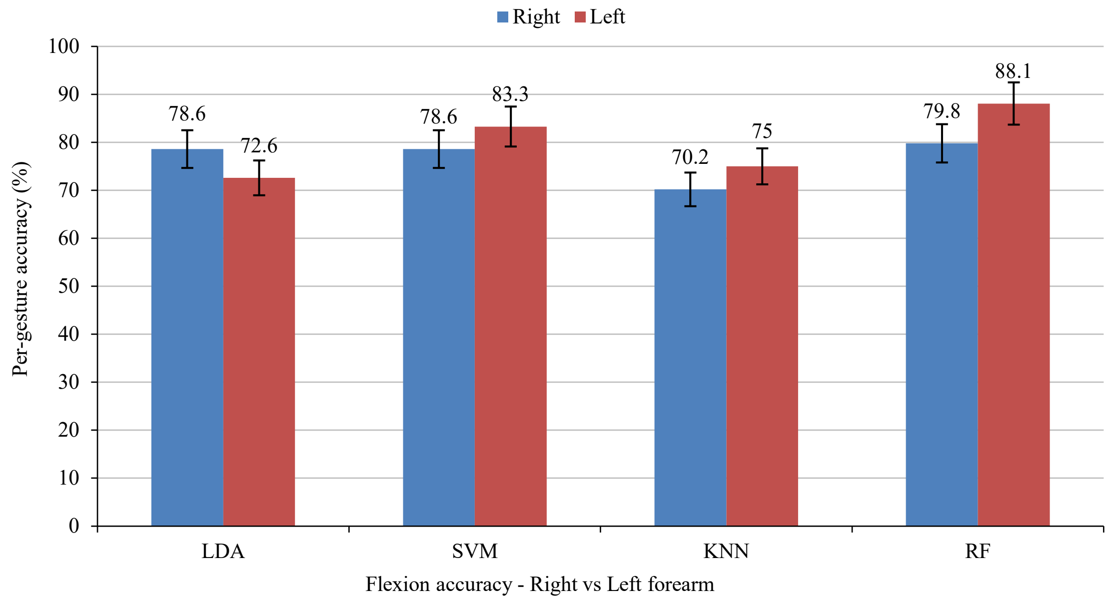


Fig. 12 Classification accuracy for Flexion hand gesture using Left and Right arms Data

A separate test asked whether the active forearm can be identified from the envelope features alone. Figure 13 shows the corresponding confusion matrices: identification remains close to the 50% chance level across several classifiers, indicating that the left and right channels are not trivially distinguishable from the feature statistics alone. This is the offline counterpart of the difficulty observed in real time when both hands are moved together to command a rotation, and it motivates the amplitude-and-timing threshold logic adopted on the device in preference to a learned left/right discriminator. The feature-significance analysis in Table 8 and Table 9, as well as Figure 13, shows which descriptors carry the flexion-versus-extension contrast. The amplitude features of the first channel, namely MAV, RMS, and WL, are by far the most discriminative, with |t| greater than 22 and p below 10^{-69}, followed by the same features on the third channel, whereas the zero-crossing feature is uninformative for every channel and could not be evaluated reliably.

This supports the decision to base the embedded envelope detector on amplitude-type quantities, effectively the MAV and RMS of the smoothed signal, and to ignore frequency-domain or crossing-based features that add computational cost without adding separability. Table 9 presents the overall validation accuracy for flexion and extension across the right and left arms for 36 subjects. Figure 12 illustrates the classification accuracy for hand gesture flexion between the left and right arms. The analyses address two key questions: whether one can distinguish between the left and right arms using hand gestures. Furthermore, what level of accuracy can be achieved for hand gestures when considering both the right and left arms. As demonstrated in Figure 12, the results suggest that hand gestures can effectively differentiate between the left and right arms. Consequently, the insights gained from this analysis could be applied to control external devices, such as robotic arms and robotic cars.

After analyzing both datasets, the authors understand that the single-hand data gives better classification accuracy for all different kinds of gestures, while the second database records both hand activities concurrently. This paper considers left-arm and right-arm data separately for all hand-gesture activities. It focuses mainly on extension and flexion studies to distinguish between the left and right arms using EMG data recorded from 36 subjects, with validation accuracies of 60%-70%. The authors proposed different approaches to EMG data acquisition, including left wrist flexion, right arm flexion, and both hand and wrist flexions to generate commands for robotic car movement; this approach fully incorporates method two, in which envelope detection is followed by thresholding to generate the four commands. The effectiveness of the proposed method is evaluated over five subjects. Managing the subject is very difficult, and many past studies have reported results across five subjects. The results are comparable to state-of-the-art. The author presents real-time results over 20 commands in this paper, which the author will discuss in the next section, "System Integration for Real-Time Validation." Each command, such as left-arm wrist movement 20 times, right-arm movement 20 times, and both-arm movements 20 times, was presented at the time of valuation. The correctly recognized commands, such as Forward Movement (FF), Backward Movement (BB), and Rotate Movement (RR), are derived based on a thresholding technique. The actual movement of the robotic car, as summarized in the table, is observed, and detection accuracy is presented in Table 8. There is a need to explore the other hand gestures for robotic car movement. This paper explores robotic car movement using a human-computer interface without machine learning, thereby avoiding the enormous data collection required to train a classifier before real-time implementation. This approach is the big advantage of the proposed methods; it is also a cost-effective implementation because there is no involvement of a personal computer to analyze the incoming EMG signals. This implementation is based on Arduino and does not involve a lengthy feature-extraction process in a real-time environment.

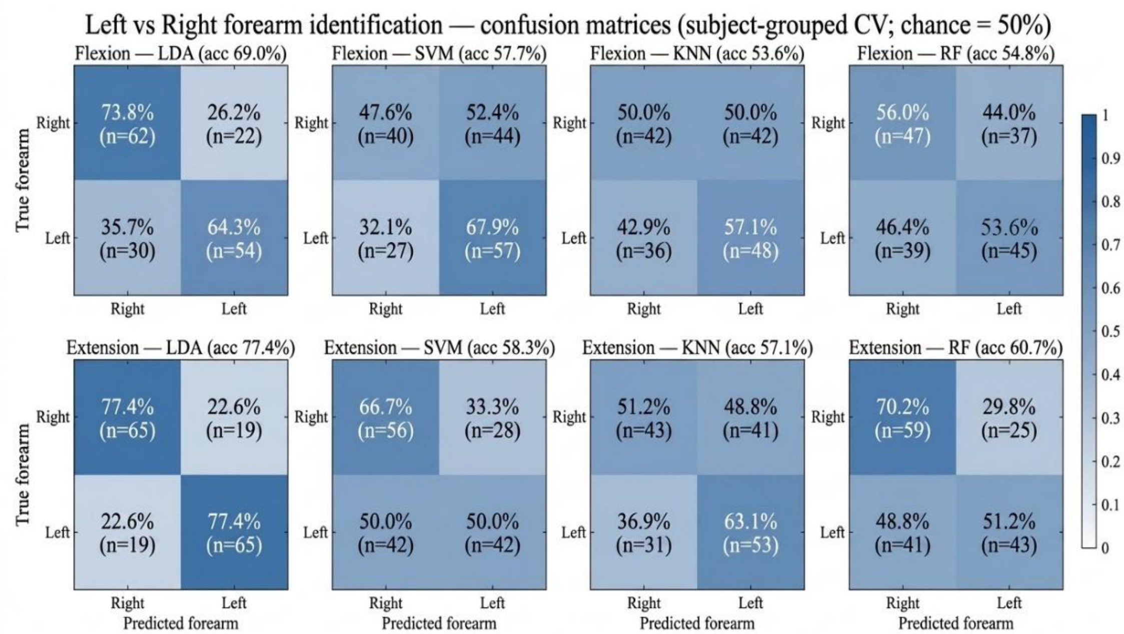


Fig. 13 Left vs right forearm identification confusion matrices (subject-grouped CV, chance = 50%)

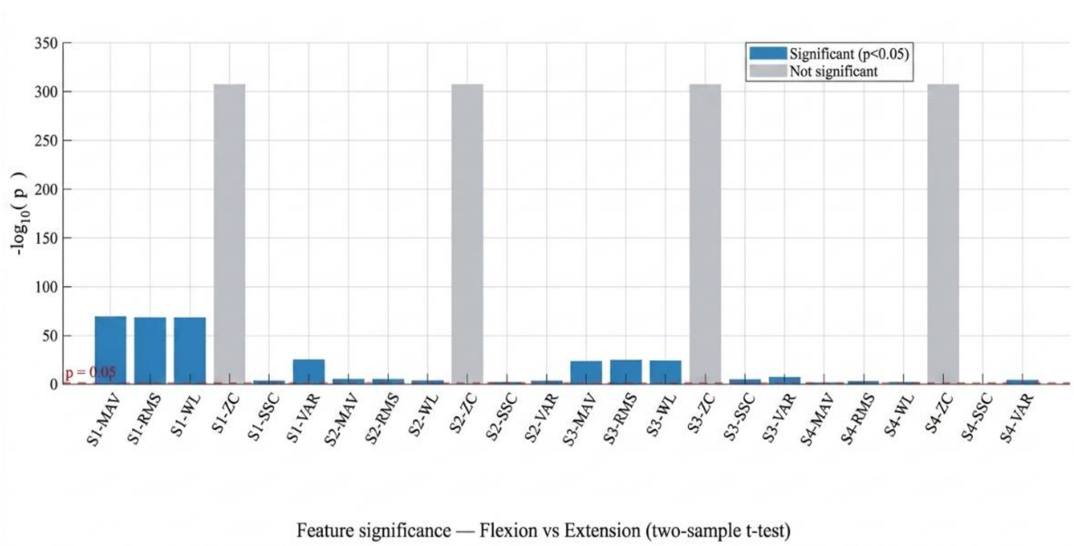


Fig. 14 Feature significance for flexion vs extension; bars mark statistically significant features ($p < 0.05$)

Table 10. Two-sample t-test of each feature for flexion vs extension (WyoFlex, S1-S4 = sensors 1-4)

Feature	t	p-value	Sig.	Feature	t	p-value	Sig.
S1-MAV	-22.928	1.5e-70	Yes	S3-MAV	11.231	4.8e-25	Yes
S1-RMS	-22.701	1.1e-69	Yes	S3-RMS	11.482	6.0e-26	Yes
S1-WL	-22.702	1.1e-69	Yes	S3-WL	11.369	1.5e-25	Yes
S1-ZC	n/a	n/a	No	S3-ZC	n/a	n/a	No
S1-SSC	-3.827	1.5e-04	Yes	S3-SSC	-4.656	4.7e-06	Yes
S1-VAR	-11.584	2.6e-26	Yes	S3-VAR	5.623	4.0e-08	Yes

S2-MAV	-4.838	2.0e-06	Yes	S4-MAV	2.386	1.8e-02	Yes
S2-RMS	-4.682	4.1e-06	Yes	S4-RMS	3.48	5.7e-04	Yes
S2-WL	-4.038	6.7e-05	Yes	S4-WL	2.868	4.4e-03	Yes
S2-ZC	n/a	n/a	No	S4-ZC	n/a	n/a	No
S2-SSC	-3.039	2.6e-03	Yes	S4-SSC	0.316	7.5e-01	No
S2-VAR	-3.677	2.7e-04	Yes	S4-VAR	4.181	3.7e-05	Yes

6. System Integration

The hardware setup is depicted in Figure 7. It shows how to operate a remote-controlled car with a small, effective EMG signal-acquisition and control system. The processed analogue signals are then fed into the Arduino UNO, which performs signal conditioning and decision-making based on the extracted EMG features. To record EMG signals from muscle activity, the system first applies surface electrodes to both arms. The BioAmp EXG Pill and AD8232, a precision analogue front-end with low noise levels intended for biomedical applications, then boost these bioelectric signals. Both amplifiers integrated and adjustable conditioning features condition the signals sent to an Arduino board, which uses its on-board ADC to sample the data (usually with a 10-bit resolution) and process it in real time. The robotic car, which is also constructed in the lab, has an ESP32 module that receives control commands from the Arduino via Bluetooth. The ESP32 has built-in Wi-Fi and Bluetooth, and it uses these signals to control the motors. In this work, two-channel EMG signals are recorded simultaneously to capture muscle activity from both hands. Figure 9 presents the block diagram of the proposed real-time embedded system for controlling the car. The incoming signals are processed by 4-stage biquad IIR digital filters implemented on an Arduino Uno. The filtered signal is then passed through an EMG filter that uses a 128-sample moving average to smooth the rectified EMG envelope and reduce high-frequency noise. Subsequently, the envelope of each channel EMG signal was extracted and analysed over time. As illustrated in Figure 17, the envelopes of the left- and right-hand gesture-related signals show distinct envelopes and sometimes overlap in time. Each time one hand performs a movement, a distinct envelope is formed, clearly differentiating between the two muscle activation events. To move the car, a mapping scheme is implemented: right-hand gestures control forward movement, left-hand gestures control backward movement, and both-hand gestures control rotation. A decision-making algorithm is based on the envelope's magnitude over time. The decision logic and control commands are summarized in Table 11. Additionally, the merged envelope for left- and right-hand gestures helps determine the threshold, as shown in Table 11.

The control logic is based on detecting muscle activity from both arms. When the right-hand gesture changes, it generates a strong EMG signal, interpreted as a control system that shows high accuracy in interpreting single-hand movements for forward and backwards motion, achieving over 80% control accuracy. However, the system exhibited unpredictable or unstable behaviour when both hands moved together to tell the robot to turn. While the robot consistently attempted to rotate, its movements were often erratic and difficult to control. Possible reasons include overlapping EMG signals, slight timing mismatches in muscle activation, and issues with tuning thresholds. The control accuracy for different gestures is summarized in Table 13. To develop a user interface device that works in real time, the fusion will be unique, and the flow of information between blocks should be smooth.

Table 11. Depicts the decision logic developed to control the external car based on the dual-channel envelop of EMG signals

Envelope 1 Condition (% power)	Envelope 2 Condition (% power)	Output	Meaning
> 90	< 20	FF	Forward (Channel 1 active only)
< 20	> 90	BB	Backwards (Channel 2 Active only)
< 20	< 20	S	Stop (No channel active)
> 90	> 90	RR	Rotate (Both Channels active)

The authors of this paper declare that a real-time prototype was developed in the laboratory using the schematic diagram of Figure 7. The sampling-to-threshold decision algorithm is implemented on the Arduino Uno development board, as shown in Figure 7. The command to move the car forward is depicted in Table 12. Similarly, when the left-hand gesture changes, it signals the car to move backwards. When both hands move simultaneously, the system interprets the gesture as a signal to rotate or take a circular turn. This setup demonstrates a real-time human-machine interface using bio-signals, enabling intuitive gesture-based control of robotic platforms. For wireless communication, the ESP32 module sends the Arduino's decision output to a remote-control unit via Bluetooth Classic. As shown in Figure 7, the processed analog signals are fed into the Arduino UNO, which conditions them and makes decisions based on the EMG features it extracts. The control unit of a car includes an ESP32 and an L298 motor driver, which interface with an RC car.

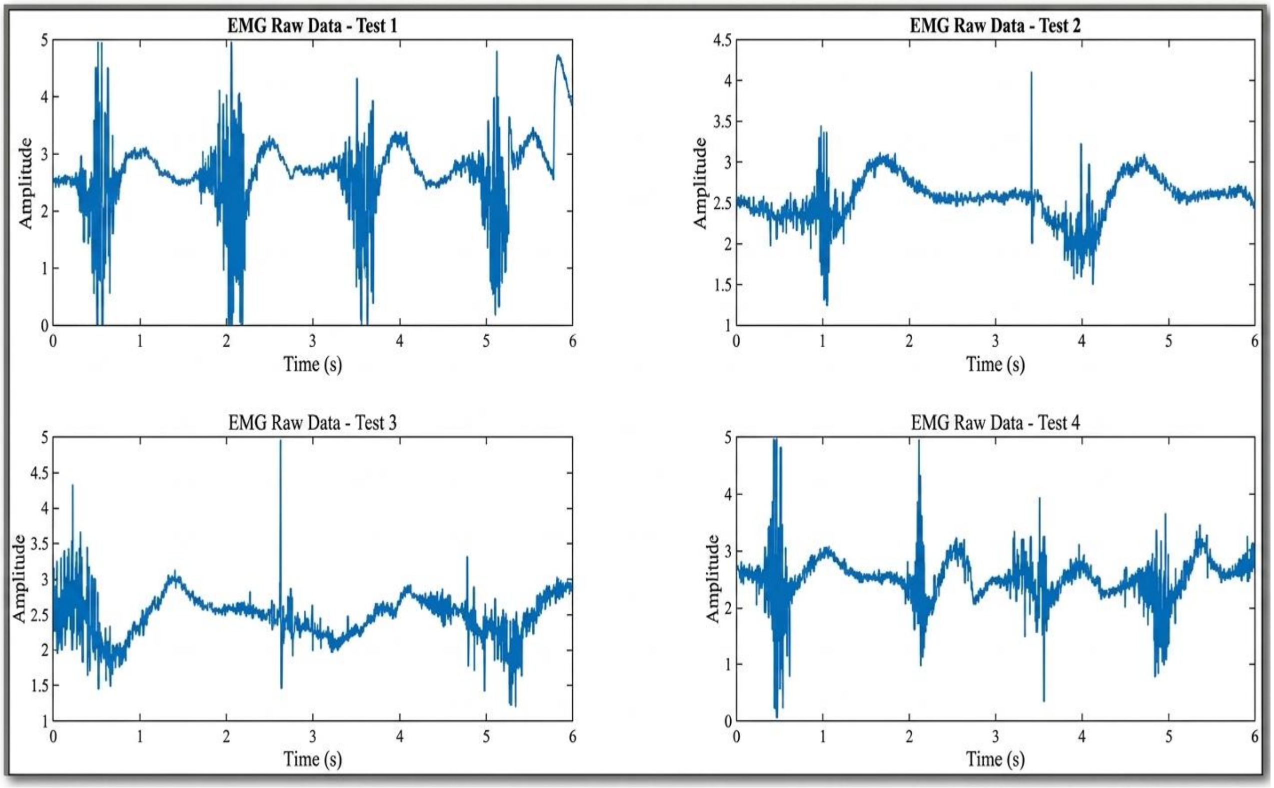


Fig. 15 Raw EMG signal recorded through the developed data acquisition system

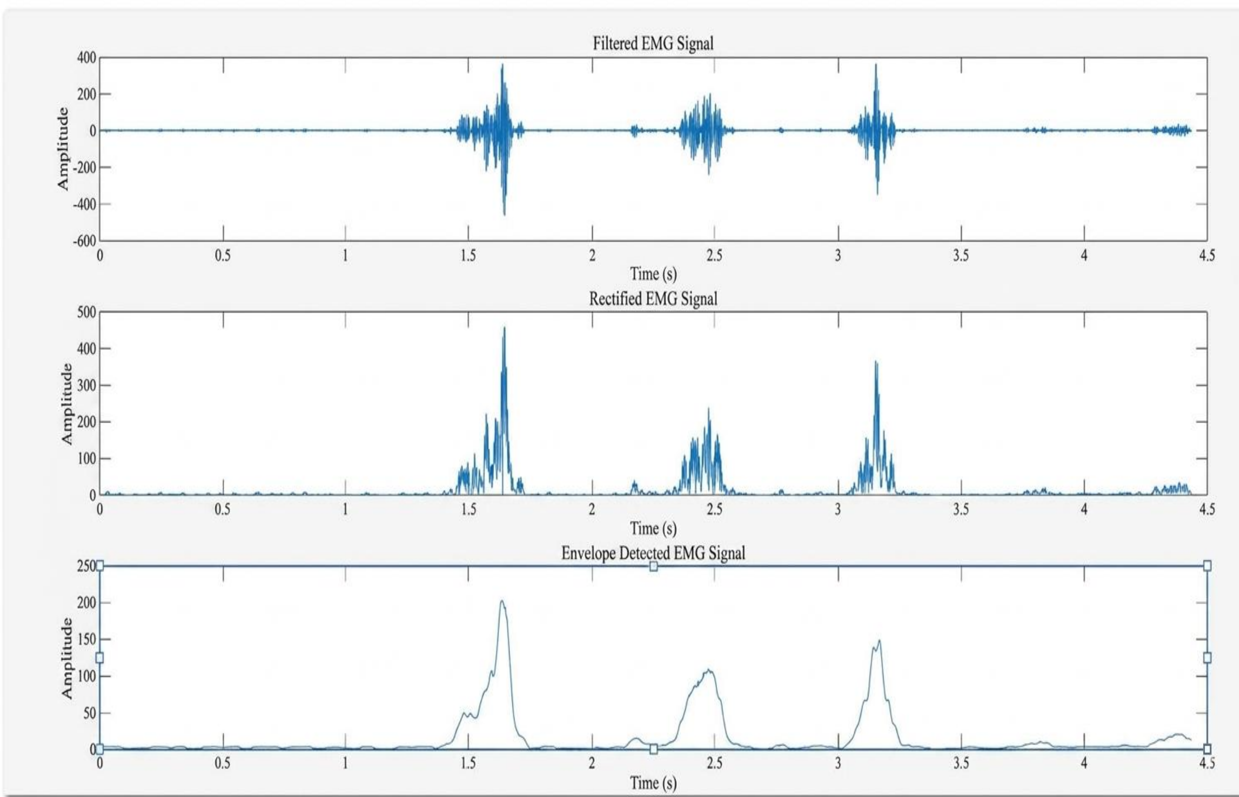


Fig. 16 Output of all stages, from preprocessing filtering to enveloping smoothing, starting from EMG acquisition filtered EMG Signals of channel 1 and channel 2

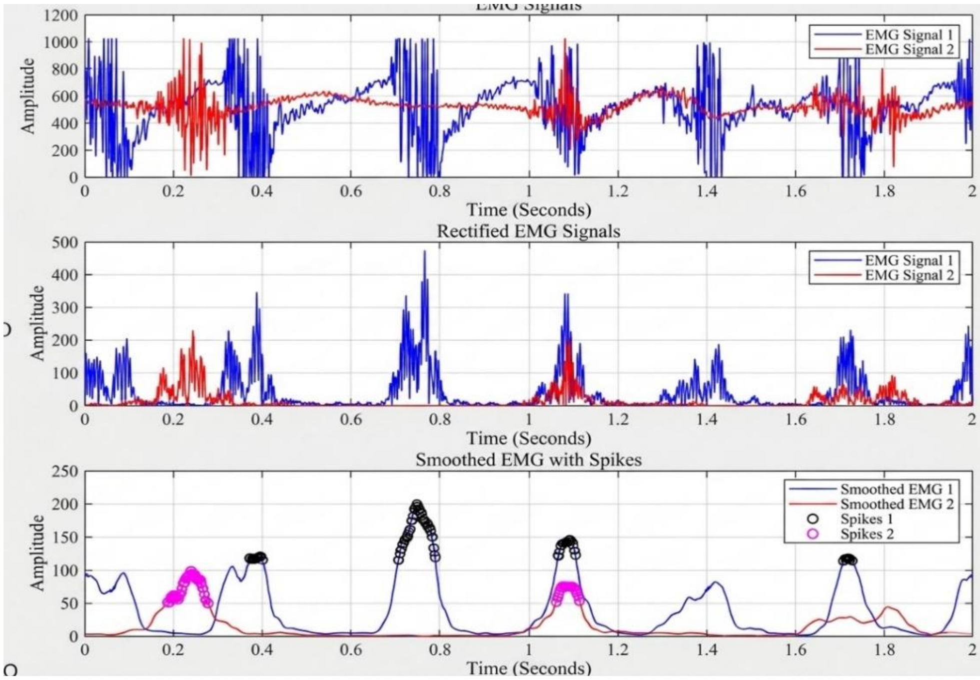
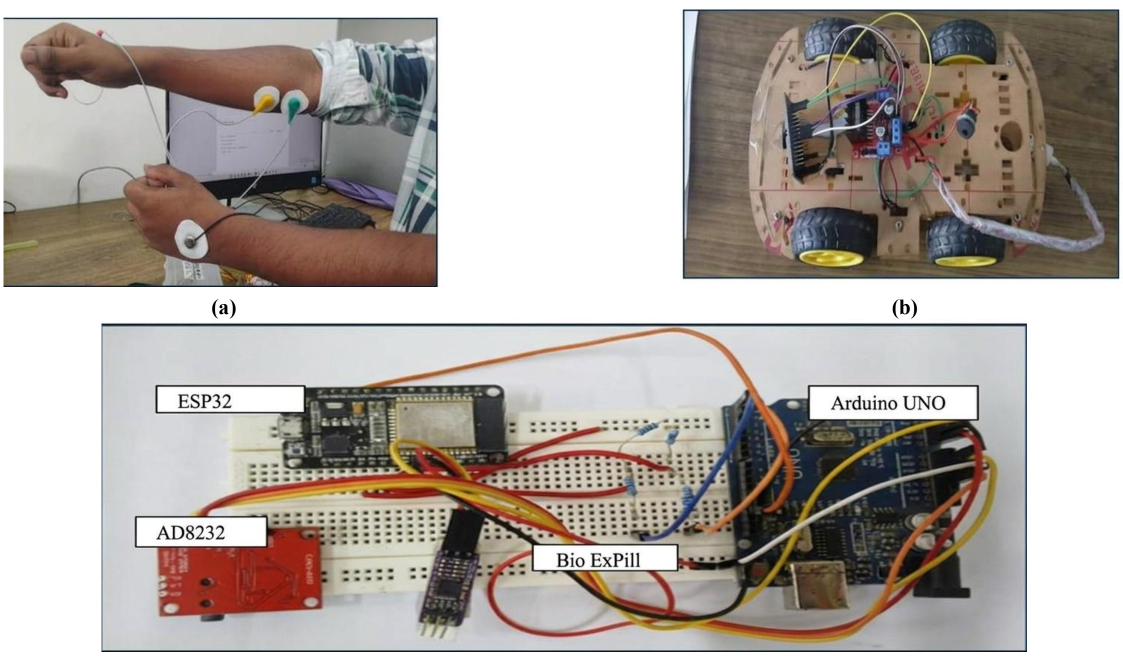


Fig. 17 Two-Channels EMG signal analysis and envelope comparisons



(a) (b) (c)

Fig. 18 Integration of System, (a) Electrode, (b) Robotic car, and (c) Tx and processing unit.

Based on these decisions, the ESP32 controls the motors, enabling gesture-based control of the car, as illustrated in Figure 18. In this system, the Arduino UNO processes the EMG signal and sends control commands to the ESP32 over UART at 115200 baud. The algorithm on the Arduino detects muscle activity through EMG envelope detection and sends commands such as "FF" (Forward), "RR" (Right), "BB" (Backward), or "S" (Stop) to the ESP32. The ESP32 modules function in Bluetooth classic master-slave mode. One ESP32 serves as the master, establishing a connection using the MAC address of the slave ESP32 connected to the robotic car. A wireless communication protocol connects the hand to the machine through Arduino for real-time car control. When the slave ESP32 receives the command, it interprets the signal and drives the car motors as needed. A right-hand movement triggers an "FF" command to move the car forward. A left-hand movement sends a "BB" command for backward motion. Moving both hands at the same time rotates the car.

Table 12. System performance

Gesture	Detected Action	Control Accuracy	Stability	Remarks
Right-hand movement	Forward ("FF")	>90%	Stable	High reliability across all users
Left-hand movement	Backward ("BB")	>80%	Stable	Consistent recognition and response
Both hands simultaneously	Rotation ("RR")	~80% (rotates always)	Unstable/Erratic	Needs improvement in signal timing & filtering

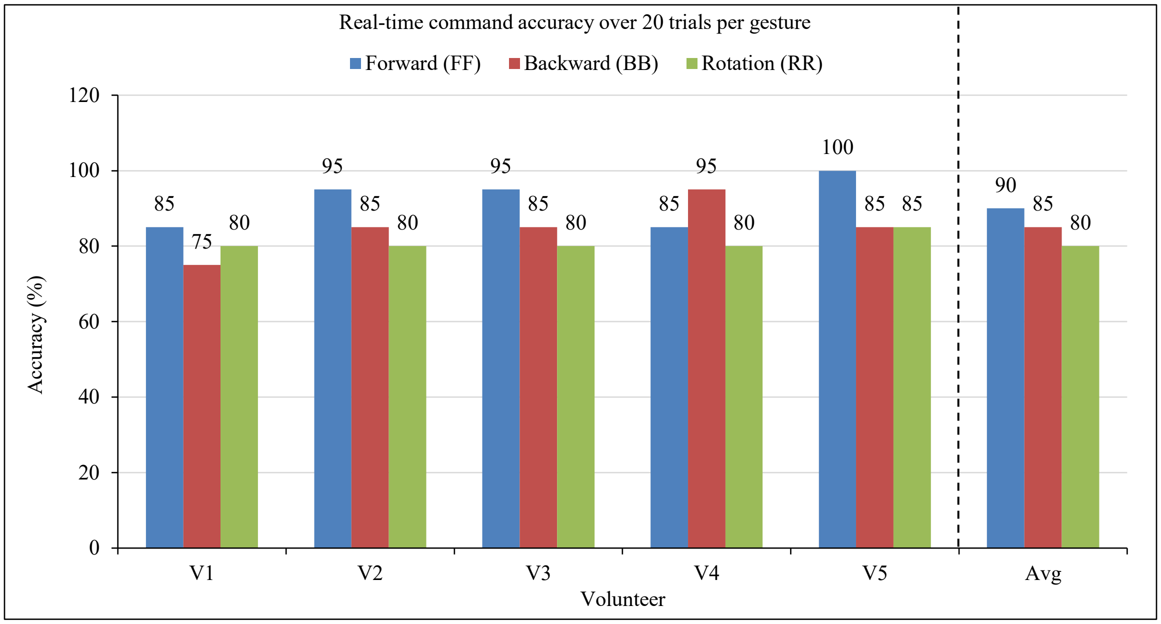


Fig. 19 Command accuracy (%) over five volunteers, 20 trials per gesture

This can be achieved using an "RR" command or a similar command. When idle, an "S" command keeps the car still. This setup creates a smooth, wireless, bio-signal-controlled robotic system that allows intuitive movement using only arm muscle activity. It enables gesture- or muscle-controlled car movement, showcasing the hardware's potential for assistive technology, rehabilitation robotics, and human-computer interaction. During testing with five volunteers (Table 13), EMG-based gestures, including left-arm wrist flexion, right-arm wrist flexion, and both-arm wrist flexion, are being implemented to validate the effectiveness of the proposed method. In this work, the authors have developed and tested a two-channel surface EMG (sEMG) acquisition and control system in the lab. An Arduino with a 10-bit ADC is used to sample two-channel EMG signals in real time. These signals are filtered, rectified, and processed directly on the Arduino to extract important features, specifically the signal envelope power for gesture classification. The command relies on the state of the hand gesture and is transmitted via Bluetooth Classic to an ESP32 module over distances of up to 5 meters, ensuring stable, reliable communication. The system is efficient and easy to use because all signal acquisition, processing, and transmission are done on a single, low-cost microcontroller. Control commands from hand gestures operate a robotic car with high accuracy for moving forward and backward. Although there is a need to improve rotation control with simultaneous dual-hand gestures, the overall system is a practical, low-latency solution for EMG-based human-machine interaction. The offline analysis reveals significant potential for controlling the robotic car using hand gestures. This issue is a significant limitation of this research paper. The authors will address similar situations in the next paper. This paper contributes in many ways, including a gap analysis, control of a robotic car using hand gestures to generate real-time commands, low-cost hardware implementations for EMG data acquisition, embedded edge computing on a microcontroller to make decisions, and ESP32-based wireless pairing.

Table 13. Accuracy for all commands over five volunteers (20 trials per command)

Volunteer ID	Forward Accuracy (%)	Backward Accuracy (%)	Rotation Accuracy (%)
1	85	75	80
2	95	85	80
3	95	85	80
4	85	95	80
5	100	85	85
Avg.	**90**	**85**	**80**

6.1 Real-Time Performance

The offline study indicates that the envelope retains most of the discriminative information and that amplitude features on a single dominant channel are sufficient to separate the gestures that matter, which is precisely the assumption built into the real-time controller. The real-time behaviour of the assembled system is reported in the following section (Table 12 and Figure 12), where five volunteers each issued twenty commands per gesture. Forward and backward commands, which depend on a single active forearm, were the most reliable, with mean accuracies of 90% and 85%, respectively, whereas the rotation command, which requires simultaneous activation of both forearms, settled at 80%. The lower rotation figure is consistent with the offline finding that left and right forearm identification is close to chance: when both channels are active at once, small timing and amplitude mismatches make the combined state harder to separate, exactly as the confusion analysis of Figure 19 predicts. Overall, the agreement between the offline benchmarks and the on-device results supports the practicality of a low-cost, threshold-based EMG interface for the forward, backward and rotation commands, while identifying the rotation logic as the main target for future improvements.

As referenced in [12], a few studies are reported on page 11, and in Table 4, it is found that robot control using single-hand EMG signals was studied. The applied methods are computationally intensive and require substantial data for training the classification model. The results reported for robotic arm control range from 84.2% to 93.3%. The methods proposed in this paper are being developed, along with various control strategies for the robotic car. For left- and right-hand wrist movement, the proposed methods achieve nearly identical classification accuracies of 90% and 85% in real-time validation for FF and BB movement, respectively. For rotation, it is slightly less than the reported results. The comparison is unfair because they used a single hand gesture, and when one examines Table 9 of the recorded dataset II for both arms, the accuracy remains low compared to the single hand gesture shown in Table 7.

7. System Limitations

It is important to discuss the limitations of the integrated system. A system is developed and integrated to perform certain tasks. Successful device-level implementation and promising results are not enough to show satisfactory work. This work has identified several limitations that highlight areas for improvement, particularly regarding the rotation command, which is triggered by simultaneous movement of both hands. The real-time command proved unreliable and very difficult to control precisely. Another limitation is that time variations and signal strengths between the two EMG envelopes often exhibit dramatic rotations, prompting us to consider more advanced signal synchronisation or decision logic. As discussed earlier, the implemented system uses Bluetooth for communication, which provides stable performance only up to 5 metres. Beyond this range, communication signal quality and reliability may vary, limiting real-world applications. The proposed system currently supports only three gesture-based decisions-forward, backward, and rotation. This study has opened many avenues for improving the man-machine interface. The setup, especially the ESP32 module with continuous Bluetooth communication and the robotic car motors, requires a high-power supply. A 4000mAh power bank is used for the ESP32, and a separate 12V Li-ion battery is required for the robot. This approach reduces the system's efficiency and is not suitable for small, portable use cases.

8. Conclusion and Future Works

This study shows how surface EMG signals can be used to create a portable HMI that allows wireless user-machine communication. This preliminary study reflects that a system can be built by integrating discrete components, and a real-time system can be developed to control an external machine via a user interface. While working on system integration, the fusion of different technologies should be unique and reflect recent technological trends. The distance between the user and the car is a major limitation for remote control. This limitation is one of the key points to further explore for usable real-time navigation in hazardous situations. Implementing more efficient, longer-range communication protocols such as Bluetooth Low Energy (BLE), Wi-Fi, or LoRa can significantly extend the operational range and reduce power consumption. This approach makes the system more suitable for mobile and outdoor applications [16]. The proposed signal processing methods already perform real-time filtering and envelope detection, and generate control commands based on thresholding. Major points learned through experimentation included generating commands to control the car in different directions at controlled speeds. This will lead to more stable command execution and improved hand gesture identification. This is another important area to explore. There is a need to expand the system to recognize more complex gestures. This will allow control over a wider range of devices and functions, creating a general-purpose gesture-based interface. Additionally, developing a mobile app for control or visualization would help users monitor EMG activity and adjust settings. Recently, the Internet of Things has become the future of communication. It is now used to send biomedical signals to remote locations. This technology can also be used to control the robot remotely for surveillance.

References

[1] Jordan Carver et al., “The Impact of Mobility Assistive Technology Devices on Participation for Individuals with Disabilities,” *Disability and Rehabilitation: Assistive Technology*, vol. 11, no. 6, pp. 468-477, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] D.V.D.S. Welihinda et al., "EEG and EMG-based Human-Machine Interface for Navigation of Mobility-Related Assistive Wheelchair (MRA-W)," *Heliyon*, vol. 10, no. 6, pp. 1-20, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] David Scherb, Sandro Wartzack, and Jörg Miehling, "Modelling the Interaction between Wearable Assistive Devices and Digital Human Models—A Systematic Review," *Frontiers in Bioengineering and Biotechnology*, vol 10, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] J.R. Wolpaw et al., "Brain-Computer Interface Technology: A Review of the First International Meeting," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 164-173, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[5] Jonathan Eby et al., "Electromyographic Typing Gesture Classification Dataset for Neurotechnological Human–Machine Interfaces," *Scientific Data*, vol. 12, no. 1, pp. 1-8, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Ángel Leonardo Valdivieso Caraguay et al., "Recognition of Hand Gestures based on EMG Signals with Deep and Double-Deep Q-Networks," *Sensors*, vol. 23, no. 8, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Iris Kyranou, Katarzyna Szymaniak, and Kianoush Nazarpour, "EMG Dataset for Gesture Recognition with Arm Translation," *Scientific Data*, vol. 12, pp. 1-11, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ulysse Côté Allard et al., "A Convolutional Neural Network for Robotic Arm Guidance using sEMG based Frequency-Features." *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), pp. 2464-2470, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Manuela Gomez-Correa et al., "Forearm sEMG Data from Young Healthy Humans during the Execution of Hand Movements," *Scientific Data*, vol. 30, pp. 1-16, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Nayan M. Kakoty et al., "Real-Time EMG based Prosthetic Hand Controller Realizing Neuromuscular Constraint," *International Journal of Intelligent Robotics and Applications*, vol. 6, no. 3, pp. 530-542, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Alycia Gailey, Panagiotis Artemiadis, and Marco Santello, "Proof of Concept of an Online EMG-based Decoding of Hand Postures and Individual Digit Forces for Prosthetic Hand Control," *Frontiers in Neurology*, vol. 8, pp. 1-15, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] EunSu Kim et al., "EMG-based Dynamic Hand Gesture Recognition using Edge AI for Human–Robot Interaction," *Electronics*, vol. 12, no. 7, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Qiyu Li, and Reza Langari, "EMG-based HCI using CNN-LSTM Neural Network for Dynamic Hand Gestures Recognition," *IFAC-Papers OnLine*, vol. 55, no. 37, pp. 426-431, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Javier O. Pinzón Arenas, Robinson Jiménez Moreno, and Ruben Darío Hernández Beleño, "Convolutional Neural Network with a DAG Architecture for Control of a Robotic Arm by Means of Hand Gestures," *Contemporary Engineering Sciences*, vol. 16, no. 12, pp. 547-557, 2018. [Google Scholar]

[15] Niosh Basnet et al., "Evaluating the Feasibility of EMG-based Human–Machine Interfaces for Driving," *The Journal of the Human Factors and Ergonomics Society*, vol. 68, no. 1, pp. 123-141, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[16] V. Rajesh, V. Sri Sravan, and N. Nanda Prakash, "EMG based Human Machine Integration for IoT based Instruments" *Proceedings of the International Conference on Internet of Everything and Quantum Information Processing. (IEQIP), Lecture Notes in Networks and Systems*, Springer, Cham, vol. 1029, pp. 27-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Kasper Leerskov et al., "Investigating the Feasibility of Combining EEG and EMG for Controlling a Hybrid Human Computer Interface in Patients with Spinal Cord Injury," *2020 IEEE 20th International Conference on Bioinformaticsand Bioengineering (BIBE)*, Cincinnati, OH, USA, pp. 403-410, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Dan Stashuk, "EMG Signal Decomposition: How can it be Accomplished and used?," *Journal of Electromyography and Kinesiology*, vol. 11, no. 3, pp. 151-173, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[19] Peter Konrad, "The ABC of EMG," *A Practical Introduction to Kinesiological Electromyography*, vol. 1, no. 2005, pp. 1-61, 2005. [Google Scholar]

[20] R. Merletti, and G.L. Cerone, "Tutorial. Surface EMG Detection, Conditioning and Pre-Processing: Best Practices," *Journal of Electromyography and Kinesiology*, vol. 54, pp. 1-21, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Jonathan R. Wolpaw et al., "Brain–Computer Interfaces for Communication and Control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767-791, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[22] Giho Jang et al., "EMG-based Continuous Control Scheme with Simple Classifier for Electric-Powered Wheelchair," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3695-3705, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rubana H. Chowdhury et al., "Surface Electromyography Signal Processing and Classification Techniques," *Sensors*, vol. 13, no. 9, pp. 12431-12466, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[24] Lasitha Piyathilaka et al., "Advances in EMG Signal Processing and Pattern Recognition: Techniques, Challenges, and Emerging Applications," *Electronics*, vol 15, no. 3, pp. 1-40, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[25] Saeid Sanei, and Jonathon.A Chambers, *EEG Signal Processing*, John Wiley and Sons, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[26] Anuj Ojha, "An Introduction to Electromyography Signal Processing and Machine Learning for Pattern Recognition: A Brief Overview," *Extensive Reviews*, vol. 3, no. 1, pp. 24-37, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Carlo J. De Luca et al., "Filtering the Surface EMG Signal: Movement Artifact and Baseline Noise Contamination," *Journal of Biomechanics*, vol. 43, no. 8, pp. 1573-1579, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[28] M.B.I. Reaz, M.S. Hussain, and F. Mohd-Yasin, "Techniques of EMG Signal Analysis: Detection, Processing, Classification and Application," *Biological Procedure Online*, vol. 8, no. 1, pp. 11-35, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[29] Mahmoud Tavakoli et al., "Robust Hand Gesture Recognition with a Double Channel Surface EMG Wearable Armband and SVM Classifier," *Biomedical Signal Processing and Control*, vol. 46, pp. 121-130, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[30] Le Wang et al., "Hand Gesture Recognition using Smooth Wavelet Packet Transformation and Hybrid CNN based on Surface EMG and Accelerometer Signal," *Biomedical Signal Processing and Control*, vol. 86, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[31] Zhengquan Xu, and Shaojun Xiao, "Digital Filter Design for Peak Detection of Surface EMG," *Journal of Electromyography and Kinesiology*, vol. 10, no. 4, pp. 275-281, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[32] Daniele Esposito et al., "A Smart Approach to EMG Envelope Extraction and Powerful Denoising for Human–Machine Interfaces," *Scientific Reports*, vol. 13, no. 1, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[33] Elisa Romero Avila, Sybele E. Williams, and Catherine Disselhorst-Klug, "Advances in EMG Measurement Techniques, Analysis Procedures, and the Impact of Muscle Mechanics on Future Requirements for the Methodology," *Journal of Biomechanics*, vol. 156, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[34] Erick Guzmán-Quezada et al., "Development of an Electromyography Signal Acquisition Prototype and Statistical Validation against a Commercial Device," *Sensors*, vol. 24, no. 21, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[35] Chiharu Ishii, and Ryoichi Konishi, "A Control of Electric Wheelchair using an EMG based on Degree of Muscular Activity," *2016 Euromicro Conference on Digital System Design (DSD)*, Limassol, Cyprus, pp. 567-574, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[36] Andre Ferreira et al., "Human-Machine Interfaces based on EMG and EEG Applied to Robotic System," *Journal of Neuro Engineering and Rehabilitation*, vol. 5, no. 10, pp. 1-15, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[37] Upside Down Labs, BioAmp EXG Pill, Upside Down Labs, 2022. [Online]. Available: https://store.upsidedownlabs.tech/product/bioamp-exg-pill/

[38] Analog Devices, AD8232: Single-Lead, Heart Rate Monitor Front End, Analog Devices, 2019. [Online]. Available: https://www.analog.com/en/products/ad8232.html

[39] Arduinodocs, UNO R3, Arduinodocs, 2026. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3/

[40] Arduinodocs, Nano ESP32, Arduinodocs, 2026. [Online]. Available: https://docs.arduino.cc/hardware/nano-esp32/

[41] K. Priyanka, and A. Mariyammal, "DC Motor Speed Control using PWM," *International Journal of Innovative Science and Research Technology*, vol. 3, no. 2, pp. 584-587, 2018. [Google Scholar] [Publisher Link]

[42] Mehmet Akif Ozdemir et al., "Dataset for Multi-Channel Surface Electromyography (sEMG) Signals of Hand Gestures," *Data in Brief*, vol. 41, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[43] N. Krilova et al., "EMG Data for Gestures," *UCI Machine Learning Repository*, 2018. [CrossRef] [Publisher Link]