Augmented Reality Based System for Myoelectric Prosthesis Training

Syuzanna Matevosyan, *Department of Electrical and Computer Engineering, University of Patras, Greece*

***Abstract*— The loss of an upper limb has a significant effect on amputees' functional capabilities and quality of life. At present, advanced myoelectric prosthetics restore a certain degree of function after amputation. These devices commonly use surface electromyography (sEMG) signals of residual muscles to activate specific motion in prosthetic arm. While promising, mastering a myoelectric prosthesis remains a challenging task. Users require extensive training to be able to generate distinctive signals from residual limb. Traditional training approaches to master prosthesis are often characterized by monotony, limited feedback, and lack of engagement. This reduces motivation and ultimately contributes to high rates of prosthesis abandonment. AR represents a promising frontier in rehabilitation, offering more engaged system to improve motor learning and training. This paper introduces an AR-based platform to help individuals master myoelectric prosthesis control.**

***Index Terms*— Prosthetics, Augmented reality, Electromyography (EMG), Deep learning, Signal processing**

I. INTRODUCTION

THE myoelectric prostheses are powered devices that use electromyography (EMG) signals generated from the user's residual muscles [1]. These signals, detected by sensors placed on the skin (surface EMG) or implanted within the muscle (intramuscular EMG), represent the electrical activity associated with muscle contractions [2]. The user's intention to move, such as opening or closing a hand, produces specific muscle activation which gets interpreted by an onboard control system [3]. Compared to body-powered devices (which use cables and harnesses), myoelectric prostheses provide more natural, intuitive control and allow for more complex movements, better replicating the appearance and function of the human arm and hand [4].

Effective training is crucial for users to learn how to generate consistent and distinguishable EMG signals for prosthesis. This often involves practicing specific muscle activation patterns and associating them with desired movements [5]. Training programs for myoelectric prosthesis can be categorized in the following types:

Direct Control (DC) Training: Direct control (also called amplitude or threshold modulation control) is the most common method in commercial prostheses. It uses the amplitude of EMG signals from one or two muscle sites—typically antagonistic pairs such as flexors and extensors—to activate corresponding prosthesis functions [4]. For example, contracting the wrist flexor muscles opens the hand, while contracting the extensors closes it. Some devices allow proportional control, where the strength of muscle contraction determines the speed or force of movement. This control requires repeated practice from users to master the selective contraction of residual muscles [6].

Pattern Recognition (PR) Control Training: Pattern Recognition is an advanced control strategy using multichannel EMG data and machine learning. PR systems decode EMG patterns to recognize complex user intentions like grasp types or simultaneous movements [4]. Subsequently, this gives a greater degree of freedom and intuitive control. When using an advanced myoelectric prosthesis, a user needs to generate distinctive and repeatable muscle activation patterns [7]. For accurate control, signals should be recognized by the classifier embedded within the prosthetic device. For example, if the user wants to open the prosthetic hand, they might consistently contract their wrist extensors in a specific way. The classifier learns to associate this unique EMG signature with the "hand open" command. Therefore, an amputee requires training, often involving sessions with a physical or occupational therapist specializing in myoelectric control.

Regardless of the method used, controlling a prosthesis is a challenging task for a prospective user. The conventional training typically involves practicing specific movements, receiving real-time feedback on their muscle activation, and learning to isolate and control individual muscles [5]. Studies have shown that individuals who have such repetitive training often report feeling bored and frustrated, which directly impacts on the risk of prosthesis abandonment. The high abandonment rate, estimated to be between 25% and 50% for upper limb prostheses, underscores the urgency of addressing these shortcomings in conventional training protocols [8].

Extended Reality (XR), encompassing Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), are increasingly being leveraged in the training of myoelectric prosthesis users [9], [10], [11]. Such immersive technologies offer high engagement, real-time feedback, and realistic contexts vital for building effective motor skills and confidence. For instance, VR completely immerses users in interactive, simulated three-dimensional environments [12]. Patients use a virtual prosthetic limb to practice tasks like grasping and object manipulation. The main benefits of VR are summarized as follows:

Motivation and Engagement: The game-like, interactive nature of VR increases enthusiasm and adherence, especially during lengthy or repetitive rehabilitation phases.

Neurorehabilitation and Brain Plasticity: VR-based interventions leverage sensory-motor learning and principles of neuroplasticity. It was shown that providing visual feedback activates relevant brain networks (including mirror neurons) and supports motor recovery and learning [13]. Also, the [14] shows that early use of VR can stop phantom limb pain before it even starts.

Safe and Risk-Free Environment for Practice: VR provides safe environment for amputees to practice daily living tasks, such as grasping, manipulating objects, or navigating home-like scenarios. This reduces risks associated with damage and self-injury.

Despite these advantages, patients using VR commonly experience motion sickness or discomfort from prolonged sessions [15]. Also, this technology struggles to replicate the tactile richness and multisensory feedback of the real world [16]. AR systems superimpose virtual elements onto the real world, enabling users to interact with digital prosthetic models in a manner that is both immersive and contextually relevant [17]. Unlike VR, AR maintains a connection to the physical world. Hence AR reduces motion sickness and enhances the sense of embodiment. Also, the study [18] shows that AR improves the treatment of phantom limb pain.

The primary goal of this work is the development of an augmented reality based myoelectric prosthesis system to effectively train motor skills and reduce prosthetic abandonment rate.

II. Review of AR-Based Myoelectric Prosthesis Systems

A variety of promising AR systems have been introduced for the myoelectric prosthesis control. The diverse approached can be categorized into two groups from user viewpoint – third-person perspective or first-person perspective [11].

A. Third-Person Perspective

Third-person perspective (3PP) AR displays a mirrored user image on an external screen, superimposing a virtual prosthetic limb onto their body [19]. Users observe themselves externally and control the virtual limb using their muscle signals. Most of AR rehabilitation systems published between 2010 and 2016 used the third-person perspective, presumably also due to technical limitations in early head-mounted displays [19].

The ARMTrainer and the system developed by Lamounier et al. are 3PP AR-based training platforms for myoelectric prosthesis [19], [20]. These systems show a mirrored real-time video of the user, overlaid with a virtual arm on the residual limb. Virtual arm controlled using their own muscle activity detected via surface electromyography (sEMG). The result is a more intuitive connection between muscle control and virtual prosthesis movement, accompanied by engaging training elements such as game-based tasks. As a game platform the ARM Trainer uses Space ARMada game, where users control the virtual arm to interactively complete objectives. In Space ARMada, users take on the role of a space explorer defending against invading spaceships. The goal of the game is to shoot down these spaceships by controlling the virtual arm, which overlaid on the user's residual limb. The speed and spread of the bullets are controlled by the user's ability to modulate their muscle activity.

3PP systems have clear limitations: limited embodiment and realism. In these systems difference between onscreen actions and physical body, as well as suboptimal depth perception, hand-eye coordination hamper skill transfer to real prosthesis use.

B. First-Person Perspective

First-person perspective (1PP) AR presents the virtual prosthesis as viewed from the user’s own field of view, typically through head-mounted displays (HMDs) or smart glasses [21]. Such systems allow users to interact with both virtual and real objects as if the prosthesis were a natural extension of their body. Visualization of limb from first person perspective leads to improved technical skills and psychological adaptation to limb loss. Since most reviews in the past have focused on third-person perspective systems, this work extends previous review by focusing more on the first-person systems and presents a table (Table 1) summarizing the main characteristic of current applications.

This work clarifies the landscape of 1PP AR applications by providing components- and methods-based overview of the field. Fig. 1 briefly presents general components used in these systems.

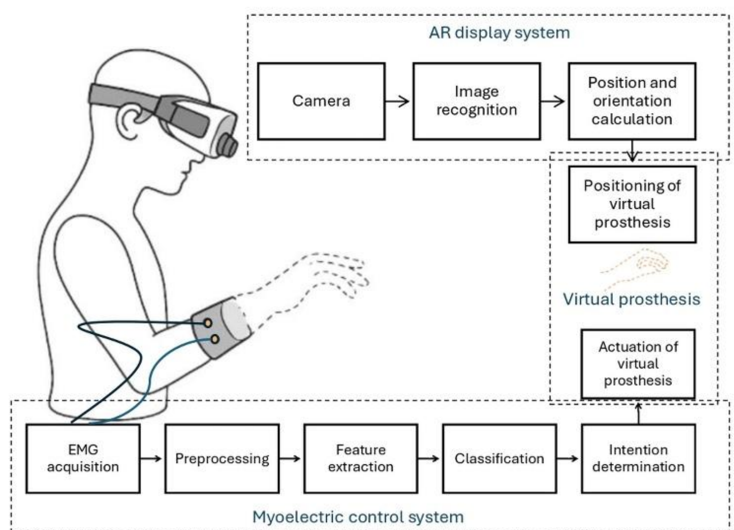

Figure 1. Schematic diagram of an Augmented Reality-based myoelectric prosthesis control system.

The main display systems used for such applications are Head-Mounted displays like Microsoft HoloLens, Microsoft HoloLens 2, Oculus Rift, and Moverio [11], [21], [22] – [26]. Through several studies they were used to project virtual

TABLE I
OVERVIEW OF THE 1PP AR-BASED MYOELECTRIC PROSTHESIS SYSTEMS

Study/ System	Purpose	AR HMD	Algorithm	Functionality	Tracking	Metric	No. of sub-jects
Boschmann et al. (2021) / ARlimb [11]	User training	Oculus Rift CV1 (110° diag., 100° × 98°)	LDA	Hand open, palmar grasp, lateral grasp, wrist pronation/supination	Custom AR markers	VCRT score	14
i-MYO [22]	Improve myoelectric	HoloLens 2 (42° × 29°)	Two-site thresholding	5 fingers flex/extend + thumb add/abd	Eye tracker	Grasp score	9
Sanchez-Rocamora et al. (2023) System [24]	User pre-training	-	HG-RNN1: HG-RNN2 HG-RNN3: 95.65%	Rest, hand open/close, victory sign, wrist flex/ext, tap	AprilTag	Time, accuracy	9
Nishino et al. (2017) Simulator [25]	Research	Epson Moverio (23° FOV)	ANN	Hand open/close	AR markers	Picking score, time	5
ProACT [26]	Research	Microsoft HoloLens 2	SVD-LDA	27 DoF	AR markers	Time, accuracy	8
MARTA System [23]	User training	Microsoft HoloLens	-	Hand open/close, pronation/supination	3 VIVE	Task score, time	1
HoloPHAM [21]	User training	Microsoft HoloLens	-	-	BOTS	PHAM score, time	-

objects and prosthetics onto the real-world environment. Microsoft HoloLens 2 eye-tracking capabilities make it advantageous for obtaining direction of the eyesight in real time. In study [22], developed i-MYO system uses gaze movement in combination with muscle activity to detect grasp-type intention. Although it achieved high effectiveness, the i-MYO system faced challenges with unintended grasp-type switches due to the limited field of view of the AR device.

Most of the system used only EMG sensors for the hand movement intention detection. Myo Armband and Ottobock sensors came across several studies [21], [23]. To classify those EMG signals into movements, most of the EMG based research studies used classifiers that employ pattern recognition algorithms [27]. Linear Discriminant Analysis (LDA) and, increasingly, deep learning methods like Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs) were used. LDA remains a prevalent choice due to its established reliability, computational efficiency, and proven success in accurately classifying various hand and forearm movements from EMG signals [28]. LDA is often used in combination with time-domain feature extraction techniques (such as mean absolute value, waveform length, slope sign changes, and zero crossings) to distinguish between discrete movement classes [29]. However, recent research trends demonstrate a shift toward deep learning approaches—such as ANNs and RNNs (including LSTM and GRU architectures)—which are especially effective at capturing the temporal and nonlinear characteristics of EMG signals [24], [25], [30]. RNNs have been successfully applied to EMG signal sequences to recognize more complex and varied hand gestures, showing better performance compared to traditional methods in some studies. [24] study proposes three different RNN architectures: HG-RNN1, HG-RNN2 and HG-RNN3, with HG-RNN3 being the most efficient demonstrated highest 95.65% accuracy.

To localize position and orientation of the virtual prosthetic limb various tracking approaches were used. Tracking systems are categorized into three groups named: Marker-Based, Markerless and Location Based. Marker-based tracking uses visual markers (e.g., AprilTags, ArUco) to display digital content [31]. These markers are recognized by the AR system, which then overlays digital information onto them. For example, when a marker is placed on the Myo bracelet, the system recognizes and positions the virtual prosthesis animation directly on the user's arm. It also continuously adjusts orientation as the user moves, thereby anchoring the virtual limb in the physical world. [24] showed that AprilTags offer higher recognition accuracy compared to Aruco markers. Commonly reported limitations of marker-based approach include depth perception (especially with monocular cameras), marker occlusion and limited camera field of view. Markerless AR relies on computer vision techniques to identify body parts or the environment without physical markers, allowing more flexible but sometimes less precise tracking [32]. While not as commonly used for high-precision limb placement, markerless approaches enable user-controlled movement and are valuable when discreetness or freedom from markers is preferred. Finally, Location-Based AR positions virtual models based on the user's geographical location (e.g., navigation cues in public spaces). These augmented models are used to provide local information based on the user's location such as walking directions or road signs.

Although AR-based prosthetic training shows promising results, most existing systems focus on single function and limited degree of freedom. Also, most solutions use limited features for machine learning training and don't provide explanation in their control strategy. These systems often show promising engagement and subjective outcomes in healthy subjects, however, there is an lack of validation on amputees to demonstrate improvement in real-world prosthesis function. Also, most published systems are either research-focused or require significant technical setup that is simply not feasible for most users. As discussed, due to the complexity of complicated or non-intuitive setup, even motivated patients may drop out, reducing the chance of meaningful rehabilitation.

This work addresses the limitations in the following ways. First, to improve control of virtual arm various features were evaluated and combined based on their importance. This multi-feature approach combined with tensor decomposition methods provides better understanding of muscle activation patterns and explains the development of control strategy. Also, unlike previous studies, that have limited functionality of the hand control, the suggested solution focuses on 17 different hand movements.

IV. Materials and Methods

The overall scheme includes several methodological components as illustrated in Fig 2. The system is composed of three main modules: myoelectric control system, communication in Python and display system. The modules are interconnected within each other and controlled in Python.

V. Myoelectric Control System

The myoelectric control system is responsible for acquiring, processing, training and classifying the user's movement intention from raw sEMG signals. This process transforms the biological signals generated by muscle contractions into discrete commands that control the virtual prosthetic hand. The architecture is designed for real-time performance and the acquisition is communicated with display system via Python.

A. Acquisition Setup

The acquisition of multi-channel EMG data was conducted using the Biometrics DataLITE wireless system. Eight LE230 wireless surface EMG sensors were placed on around elbow (Fig. 3). Before putting electrodes, the skin was carefully

Figure 3. Placement of sEMG sensors.

cleaned, and the placement was marked in the arm. The signals were sent to PC in real-time with a central DG2 Dongle interface connected via USB to a host PC. Within Biometrics Ltd. software the recording session was configured by registering each active sensor and assigning it to a unique data channel. The sampling rate for all EMG channels consistently set to 2000 Hz to ensure high-fidelity signal capture, which also matches the initial dataset (explained in the next paragraph)

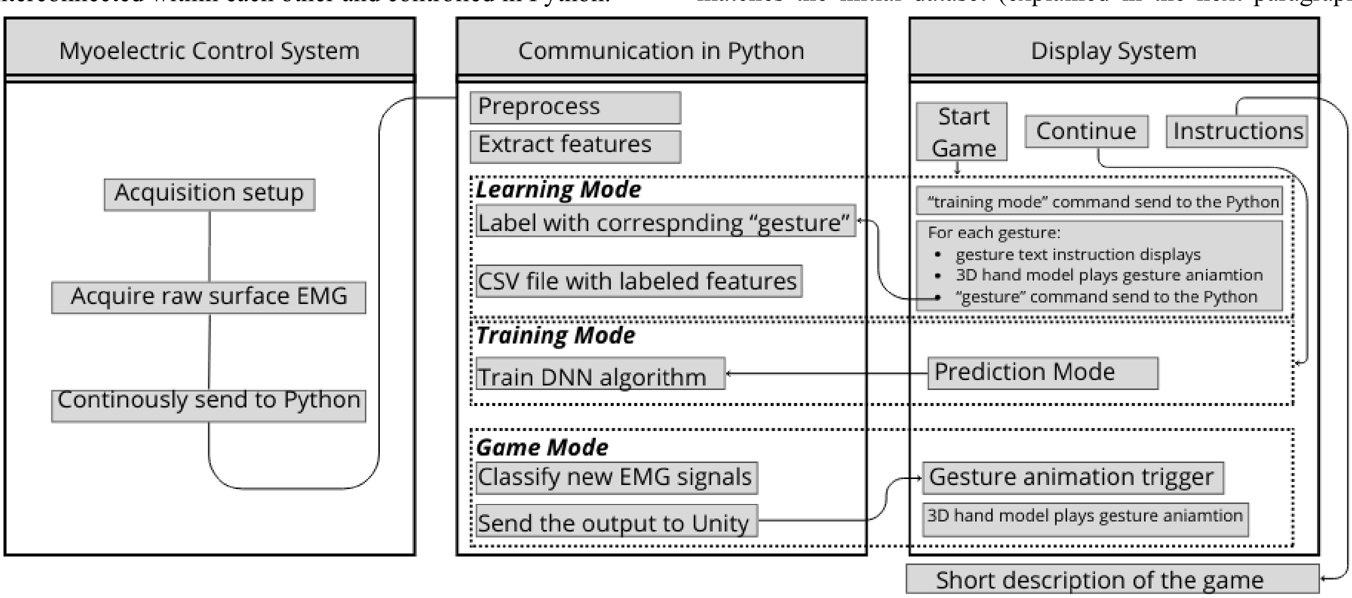

Figure 2. System architecture for EMG-controlled virtual hand.

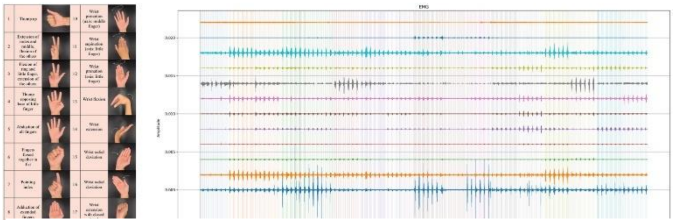

Figure 3. EMG signals of 17 movements across all 12 channels — plotted over time.

used for processing pipeline development. By deactivating the software's "File Save Mode," the system enabled to stream the calibrated, multi-channel data directly to the custom Python application for real-time processing and analysis.

The Python script handled EMG using a 64-bit Windows DLL (OnLineInterface64.dll) functions provided by Biometrics Ltd. This DLL facilitates low-level hardware communication for real-time EMG data capture and should be in the same folder as the python code. Since the Biometrics Ltd is written in C/C++, the Python script uses *ctypes* to directly call C functions, pass pointers, and handle memory structures. The acquisition process is a thread that continuously polls the EMG channels and saves in SAFEARRAY structure that shown below. Each acquisition block duration is defined in milliseconds and converted into a sample count using the device sampling rate. The retrieved raw data is structured into NumPy arrays and pushed into a thread-safe queue for signal processing pipeline.

B. The Ninapro DB7 Dataset

The signal processing pipeline is developed using Ninapro DB7 Dataset before transitioning to real-time EMG signals. This dataset served as the foundation for systematic evaluation of various methods to identify the best pipeline.

Ninapro Database 7 is a multimodal dataset containing EMG measurements from 20 intact subjects and 2 amputees, using 12 wireless EMG sensors with the 2000 Hz acquisition rate. Subjects performed 17 different upper-limb movements, each repeated 6 times, with the rest in between. As shown in fig. 3, the EMG signals of different movements produce different patterns with all 12 electrodes combined.

Preprocessing: The preprocessing pipeline is started by addressing the class imbalance problem. Since the rest is repeated after each movement, there is a significant class imbalance that may create bias toward the “rest” class. Down sampling is applied to ensure balance between movement classes. In the preprocessing pipeline the windowing parameters identified using PARAFAC tensor decomposition method. PARAFAC temporal components reveal the duration and shape of the signal patterns that are stable and meaningful for distinguishing gestures. Temporal loadings with different window size show that the most informative signal variation occurs in1000ms with 100ms sliding window, thus, it was used to extract features.

Feature Extraction: Diverse sets of features were extracted from each windowed segment of the signal, in order to analyze and choose the most discriminative features. After statistical methods such as ANOVA F-score, Random Forest and Mutual Information, the following 7 features were selected as top features, since they consistently ranked in the top (Supplementary Figure S1). The following features capture both time-domain and frequency-domain information: Waveform Length (WL), Root Mean Square (RMS), FFT Energy (FFTEnergy), Variance (VAR), Integrated EMG (IEMG), Mean Absolute Value (MAV) and Total Power (TP).

$$(1)\quad WL = \sum_{\{i=1\}}^{\{N-1\}} \left| x_{\{i+1\}} - x_i \right|$$

$$(2)\quad RMS = \sqrt{\left\{ \frac{1}{N} \sum_{i=1}^{N} x_i^2 \right\}}$$

$$(3)\quad E_{FFT} = \sum_{k=0}^{\left\lfloor \frac{N}{2} \right\rfloor - 1} |X[k]|^2$$

$$(4)\quad VAR(x) = \sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2$$

$$(5)\quad IEMG = \sum_{i=1}^{N} |x_i|$$

$$(6)\quad MAV = \frac{1}{N} \sum_{i=1}^{N} |x_i|$$

$$(7)\quad TP = \sum_{k=0}^{\frac{N}{2}-1} |X[k]|^2$$

To validate the findings, PCA is applied to reduce the feature space and visualize classes. Since 3 principal components (PCs) were required to preserve more than 90% of the variance in the original feature space, the 3D scatter plot was performed with its corresponding average scores. The 3D scatter plot enables better qualitative analysis of how different movement classes are grouped and separated (Fig. 4).

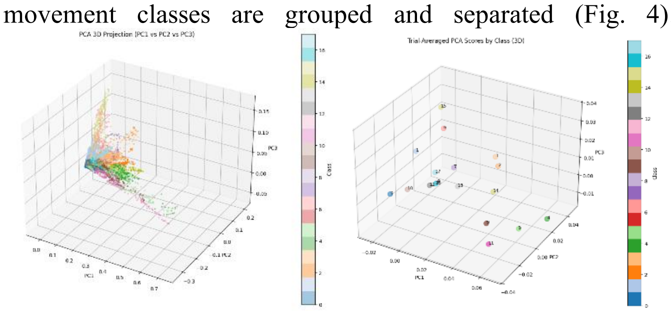

Figure 4. PCA 3D projection and trial-averaged PCA scores by 17 movement class

Classification: Various Machine Learning models were evaluated to select the best performed model. The classifiers tested include traditional ML models (SVM, LDA, RF, etc.) and a deep neural network (DNN). Results included both test accuracy and cross-validation performance. The entire pipeline — preprocessing, windowing, feature extraction, and model training — was repeated per subject. DNN consistently demonstrated the highest cross validation result, particularly for complex and overlapping gestures, owing to its ability to learn non-linear patterns and hierarchical representations from high-dimensional EMG feature vectors. The model architecture was further optimized through hyperparameter tuning, with layer sizes, dropout rates, and activation functions systematically adjusted to maximize validation accuracy and minimize overfitting. DNN were chosen for the further hand gesture recognition in real time EMG data.

C. Hand Gesture Recognition

Once raw EMG signals are acquired from eight channels using the Biometrics Ltd device and DLL interface, the data is continuously buffered and segmented into overlapping windows (typically 1000 samples per window, with a stride of 100 samples). For each window, 7 features are extracted per channel. During the training phase, these features are saved alongside their label's movement classes into a CSV file. Once sufficient labeled data is collected, a deep DNN is trained using this feature set. The model, constructed with several dense layers, batch normalization, and dropout for regularization, is optimized using categorical cross-entropy loss and Adam optimizer. After training, the system enters prediction mode, where incoming EMG features are normalized and classified in real time. The predicted gesture labels are smoothed over a prediction buffer of 10 predictions to improve temporal consistency and robustness. The most frequent gesture in the buffer, which is the final recognized gesture is transmitted to Unity via UDP, enabling interactive applications such as virtual hand control.

VI. Display System

The display system is developed entirely within the Unity 3D engine using C# scripting.

A. Unity Setup

The project was initiated using Unity Hub, which facilitates the management of different Unity Editor versions. Unity 2020.3 LTS (Long-Term Support) was selected for this project due to its stability and proven compatibility with the required AR and networking packages. Two main scenes were created: SampleScene and GameScene to sustain the smooth flow of the game.

SampleScene menu – SampleScene is the main menu interface. It serves as a welcome and navigation interface, which contain Canvas to host UI elements. Three buttons were added as shown in the Fig. 5. Script were created to handle UI navigations.

GameScene menu – GameScene is an interactive animation with 3D modeled virtual hand which uploads after person clicks "Start Game" or "Continue" buttons. In case of "Start Game"

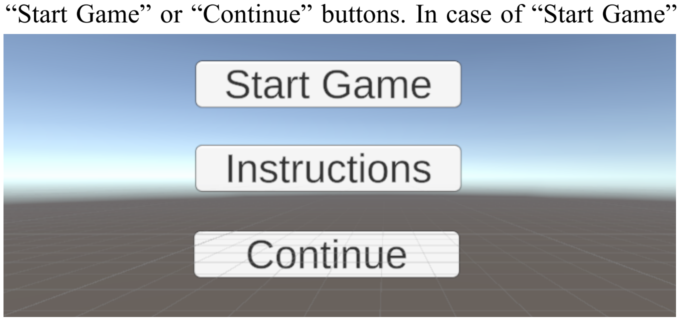

Figure 5. Interface of SampleScene - main menu

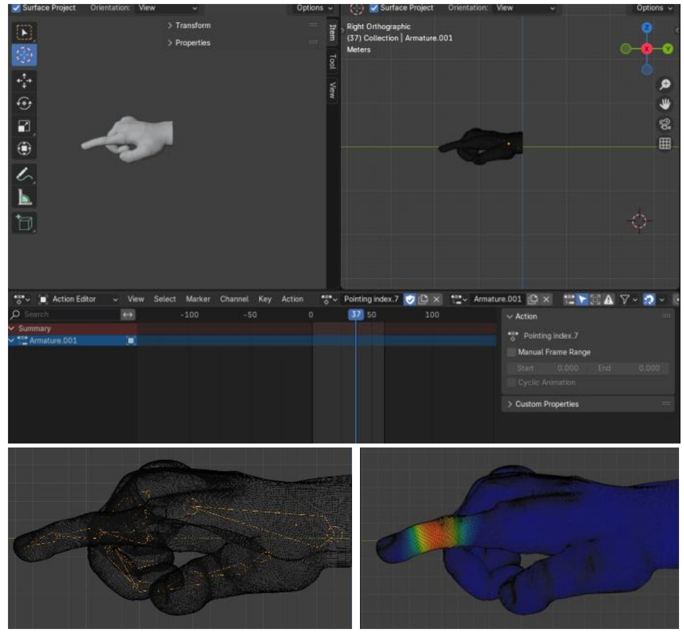

Figure 6. Animation of pointing up index finder movement in Blender

the 3D hand model functions as a guide for the user by making them to repeat the movements, while EMG signals are recorded and labelled. After the recording, the "Continue" mode is triggered, where the user can control the virtual hand, with the new upcoming EMG's signals decoding his intentions. The "Continue" mode also can be triggered by user pressing "Continue" button and skipping the training phase. This feature is added into the game, allowing players to train only one time and control the virtual arm every time when they will return to the game.

B. Hand Gesture Animation

A 3D hand model was created in Blender 2.9. The bones were accurately placed to each finger, the palm and the wrist. Once the skeleton is prepared, the hand mesh was parented to the armature using the "With Automatic Weights" option. This allows the mesh to move responsively with the underlying bones. The weight painting was used to refine vertex weight and ensure that each part of the mesh deforms naturally when underlying bones move. After rigging, the model was animated by entering pose mode and setting keyframes for different hand gestures (Fig. 7). 17 different animations were created, matching the animation of Ninapro DB7 hand gestures. This animated hand model served as the core visual element of the interactive motor skills game, providing clear movement cues that synchronize with EMG data in real time.

C. Game Mode

The game begins with a main menu offering two options: viewing instructions or starting the training session. Once the player initiates the game, they experience a countdown sequence followed by a systematic series of hand movement exercises. The training consists of 17 distinct movement types - each performed in sets of 6 repetitions with timed rest

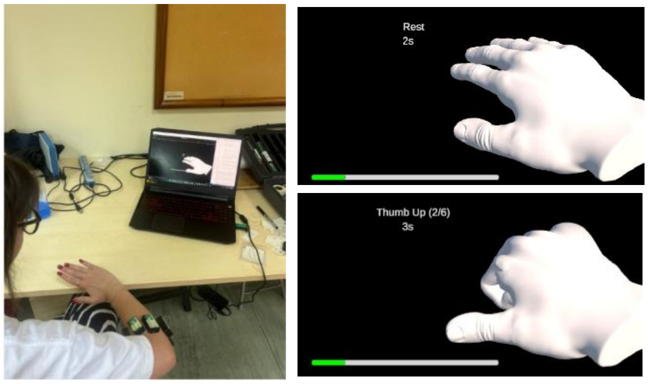

Figure 7. Real-time hand animation guides the user to perform specific gestures

periods between each movement. During each exercise, players see visual cues displaying instructions like "Close your hand (1/6)" while simultaneously watching an animated 3D hand model demonstrate the correct movement. The game maintains a consistent rhythm with 5-second movement phases followed by 3-second rest periods, creating a total of 17 animated movement sequences. Throughout the training, the system communicates with external Python code to collect and label corresponding EMG signals from the user's actual hand movements. After completing all movement sequences, players receive a congratulatory message and transition to a control phase where they can manipulate a virtual arm using their learned motor patterns.

The system's intelligent design includes a "Continue Game" feature that recognizes returning users and their previously saved EMG profile data. This means that once a user has completed the initial 17-movement calibration sequence, they can bypass the repetitive training phase on subsequent sessions. The system automatically loads their saved EMG signal patterns and neural mappings, allowing them to jump directly into the virtual arm control mode.

VII. Communication

To enable seamless interaction between signal acquisition, machine learning, and display systems, a multi-threaded socket communication framework was implemented. Two independent threads were continuously active within the Python environment: the EMG thread and the Unity thread. The EMG thread interfaced with the OnLineInterface64.dll—a 64-bit dynamic link library provided by Biometrics Ltd—to signals in real time. This thread continuously polled the connected wireless sensors, buffered the data, and stored it in a shared thread-safe queue for processing. Simultaneously, the Unity thread ran a dedicated UDP server that continuously listened for incoming commands sent from the Unity interface. These commands indicated the current movement instruction (e.g., “Hand Close (1/6)”), allowing the Python pipeline to begin labeling the corresponding EMG data for model training. Once the training phase was completed, the system switched to classification mode. A separate UDP client was launched in Unity using C# scripts, which continuously listened for gesture classification results sent from Python. The Python classifier analyzed the incoming EMG windows, predicted the user’s intended gesture using a pre-trained DNN model, and transmitted the output to Unity. Upon receiving the classified gesture ID, Unity triggered the corresponding 3D hand animation in real time, thus completing the interactive control loop. This architecture enabled synchronized training and real-time gesture control by maintaining continuous bi-directional communication between modules, while multithreading ensured non-blocking execution and high responsiveness.

VII. Results

The AR-based myoelectric prosthesis training system was evaluated through both offline testing using the Ninapro DB7 dataset and real-time experiments using EMG signals acquired via the Biometrics DataLITE system. The DNN model trained on subject-specific EMG features demonstrated high classification accuracy. On the Ninapro DB7 dataset, the cross-validated accuracy reached 71% ± 2.3%, with the most confusion observed between gestures with similar muscle activation patterns, such as “supination” and “pronation.”

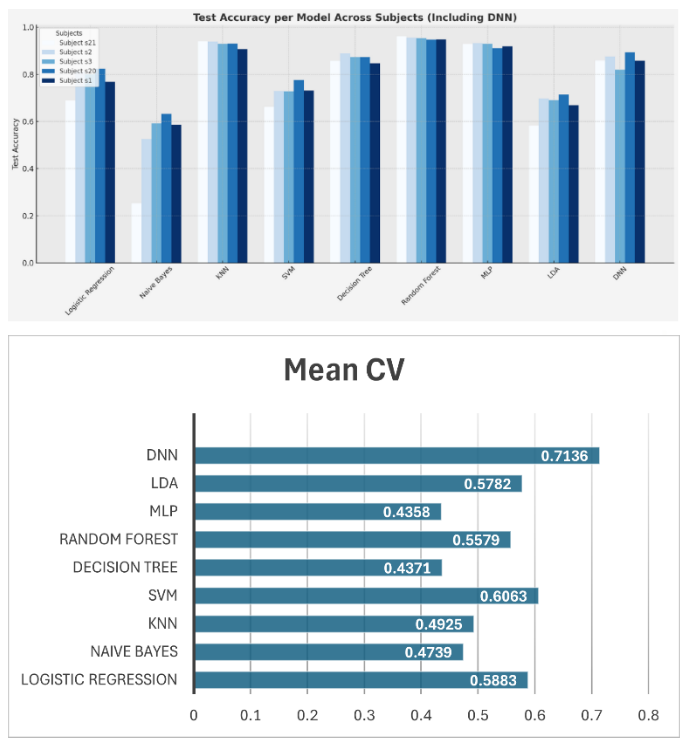

Figure 8. Test accuracy and mean cross-validation (CV) per model across subjects

When trained on real-time EMG signals from the Biometrics Ltd device, the model achieved an average accuracy of 70% across 8 tested movement classes. Performance dropped slightly compared to the offline dataset due to real-world noise, muscle fatigue, and sensor shift—but remained within a usable range for interactive control.

To improve accuracy, predictions were thresholder using a prediction buffer of 10 values. The most frequent value of the buffer was decided as a final output. This strategy reduced transient misclassifications and ensured consistent gesture triggering during control sessions. However, it makes the prediction speed slow, and further optimization needed to speed up this process.

During the training mode, users were guided through 17 animated hand gestures, each repeated six times. Visual cues and auditory feedback helped maintain rhythm and focus. Participant reported that the first-person view animation

Model	Dataset	Accuracy (%)	Notes
DNN (Ninapro DB7)	Offline	71 ± 2.3	17 classes
DNN (Real-time Biometrics)	Live EMG	70	17 classes

increased understanding of correct movement form and encouraged concentration. Key observations:

Engagement: Subjects rated the training experience highly engaging compared to conventional physiotherapy exercises.

Ease of Use: No technical background was required; participants followed clear instructions through the game interface.

Motivation: All subjects reported higher motivation to repeat training due to visual feedback and game-like progression.

Virtual Arm Control: After completing the training phase, subjects controlled the animated virtual arm using real-time EMG inputs.

VII. Discussions

This study demonstrates a functional framework for real-time hand gesture recognition using EMG signals and virtual hand control. While the system shows promising results, several areas require improvement to enhance usability, responsiveness, and application potential.

One key challenge is the responsiveness of the system, particularly during real-time prediction and feedback. Latency caused by feature collection windows and animation rendering slows down user interaction. Future work will focus on optimizing the animation speed and reducing the prediction collection threshold, enabling smoother transitions and more immediate system responses. This enhancement is critical for applications requiring high precision and low delay, such as prosthetics control or rehabilitation scenarios.

Another important consideration is the refinement of the gesture set. Certain movements, such as supination/pronation and axis-specific wrist rotations, introduced significant ambiguity and were prone to misclassification due to overlapping muscle activation patterns. These gestures will be revised or removed to reduce false detections and improve classification robustness.

For a more immersive and intuitive experience, next step is the integration of augmented reality (AR) devices. Although the unity environment were created for the AR integration, the next step requires real device for the AR testing. AR offers enhanced visual feedback during training and usage, enabling users to visualize hand movements and system predictions directly in their environment. This could be particularly impactful in rehabilitation, allowing patients and clinicians to interact in an engaging, spatial context.

VIII. Conclusion

This work presents an EMG-based real-time hand gesture recognition system integrated with a virtual hand for intuitive control and interaction. By extracting relevant features from EMG signals and utilizing a neural network for classification, the system successfully demonstrates the potential for responsive and natural gesture-based control. Despite some limitations in responsiveness and gesture ambiguity, the current framework lays a solid foundation for future improvements. With further optimization of system speed, refinement of gesture definitions, and integration of augmented reality technologies, this platform holds significant promise for applications in rehabilitation, prosthetics, and human-computer interaction.

References

1. M. P. Selvan et al., "Prosthetic Hand Using Emg," *Journal of Physics: Conference Series*, vol. 1770, no. 1, pp. 012018, 2021.
2. M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: detection, processing, classification and applications," *Biological Procedures Online*, vol. 8, no. 1, pp. 11-35, 2006.
3. M. K. Ulubas et al., "EMG Controlled Artificial Hand and Arm Design," *Wseas Transactions On Biology And Biomedicine*, vol. 19, pp. 41-46, 2022.
4. M. R. Mohebbian et al., "A Comprehensive Review of Myoelectric Prosthesis Control", 2021.
5. M. R. Dawson, J. P. Carey, and F. Fahimi, "Myoelectric training systems," *Expert Review of Medical Devices*, vol. 8, no. 5, pp. 581-589, 2011.
6. Y. Hara et al., "Development of an evaluation method for controlling skill of a myoelectric control hand," *Electronics and Communications in Japan*, vol. 93, no. 5, pp. 15-23, 2010.
7. L. Resnik et al., "Evaluation of EMG pattern recognition for upper limb prosthesis control: a case study in comparison with direct myoelectric control," *Journal of NeuroEngineering and Rehabilitation*, vol. 15, no. 1, 2018.
8. L. C. Smail et al., "Comfort and function remain key factors in upper limb prosthetic abandonment: findings of a scoping review," *Disability and Rehabilitation: Assistive Technology*, vol. 16, no. 8, pp. 821-830, 2020.
9. B. D. Winslow, M. Ruble, and Z. Huber, "Mobile, Game-Based Training for Myoelectric Prosthesis Control," *Frontiers in Bioengineering and Biotechnology*, vol. 6, 2018.
10. W. Li et al., "Current status and clinical perspectives of extended reality for myoelectric prostheses: review," *Frontiers in Bioengineering and Biotechnology*, vol. 11, 2024.
11. A. Boschmann et al., "Immersive augmented reality system for the training of pattern classification control with a myoelectric prosthesis," *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 1, 2021.
12. I. Wohlgenannt, A. Simons, and S. Stieglitz, "Virtual Reality," *Business & Information Systems Engineering*, vol. 62, no. 5, pp. 455-461, 2020.
13. R. S. Calabrò et al., "The role of virtual reality in improving motor performance as revealed by EEG: a randomized clinical trial," *Journal of NeuroEngineering and Rehabilitation*, vol. 14, no. 1, 2017.
14. C. Heinrich et al., "Feasibility and psychophysical effects of immersive virtual reality-based mirror therapy," *Journal of NeuroEngineering and Rehabilitation*, vol. 19, no. 1, 2022.
15. E. Chang, H. T. Kim, and B. Yoo, "Virtual Reality Sickness: A Review of Causes and Measurements," *International Journal of Human–Computer Interaction*, vol. 36, no. 17, pp. 1658-1682, 2020.
16. A. Maksatbekova, and M. Argan, "Virtual reality's dual edge: navigating mental health benefits and addiction risks across timeframes," *Current Psychology*, 2025.
17. Y. Ghazwani, and S. Smith, "Interaction in Augmented Reality," *Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations*, pp. 39-44, 2020.
18. M. Ortiz-Catalan et al., "Treatment of phantom limb pain (PLP) based on augmented reality and gaming controlled by myoelectric pattern recognition: a case study of a chronic PLP patient," *Frontiers in Neuroscience*, vol. 8, 2014.
19. F. Anderson, and W. F. Bischof, "Augmented reality improves myoelectric prosthesis training," *International Journal on Disability and Human Development*, vol. 13, no. 3, 2014.
20. E. A. Lamounier. Kenedy Lopes, "Using Augmented Reality Techniques to Simulate Myoelectric Upper Limb Prostheses," *Journal of Bioengineering and Biomedical Sciences*, 2013.
21. A. Sharma et al., "Augmented Reality Prosthesis Training Setup for Motor Skill Enhancement,", 2019.
22. C. Shi et al., "i-MYO: A multi-grasp prosthetic hand control system based on gaze movements, augmented reality, and myoelectric signals," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 20, no. 1, 2023.
23. S. Soliman, A. R. Moukarzel, I. Lee, M. C. Hodgson, C. L. Hunt, R. R. Kaliki, A. W. Shehata, and J. S. Hebert, "Effects of augmented reality training on pattern recognition control in myoelectric prostheses users: A case study," in *2023 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Singapore, 2023, pp. 1–6.
24. A. Sanchez-Rocamora, E. Martinez-Martin, and A. Costa, "A low-cost AR application to control arm prosthesis," *Virtual Reality*, vol. 27, no. 4, pp. 3469-3483, 2022.
25. W. Nishino, Y. Yamanoi, Y. Sakuma, and R. Kato, "Development of a Myoelectric Prosthesis Simulator using Augmented Reality," in *Proc. 2017 IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Banff, AB, Canada, Oct. 5-8, 2017, pp. 1046–1051
26. S. Guptasarma, and M. Kennedy, "ProACT: An Augmented Reality Testbed for Intelligent Prosthetic Arms," *arXiv*, 2024.
27. I. Batzianoulis et al., "EMG-Based Analysis of the Upper Limb Motion," *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, pp. 49-50, 2015.
28. J. Deng et al., "Discriminant Analysis Based EMG Pattern Recognition for Hand Function Rehabilitation," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 207-214, 2018.
29. M. A. A. Rahman, M. A. H. Akhand, and M. M. Islam,

"Classification of Ankle Joint Movements Based on Surface Electromyography Signals," *International Journal of Computer Applications*, vol. 147, no. 12, pp. 1–6, Aug. 2016.

30. D. Buongiorno et al., "A Survey on Deep Learning in Electromyographic Signal Analysis," *Lecture Notes in Computer Science*, pp. 751-761, 2019
31. S. Fleck, M. Hachet, and J. M. C. Bastien, "Marker-based augmented reality," *Proceedings of the 14th International Conference on Interaction Design and Children*, pp. 21-28, 2015.
32. T. Hellsten et al., "The Potential of Computer Vision-Based Marker-Less Human Motion Analysis for Rehabilitation," *Rehabilitation Process and Outcome*, vol. 10, 2021.

SUPPLEMENTARY

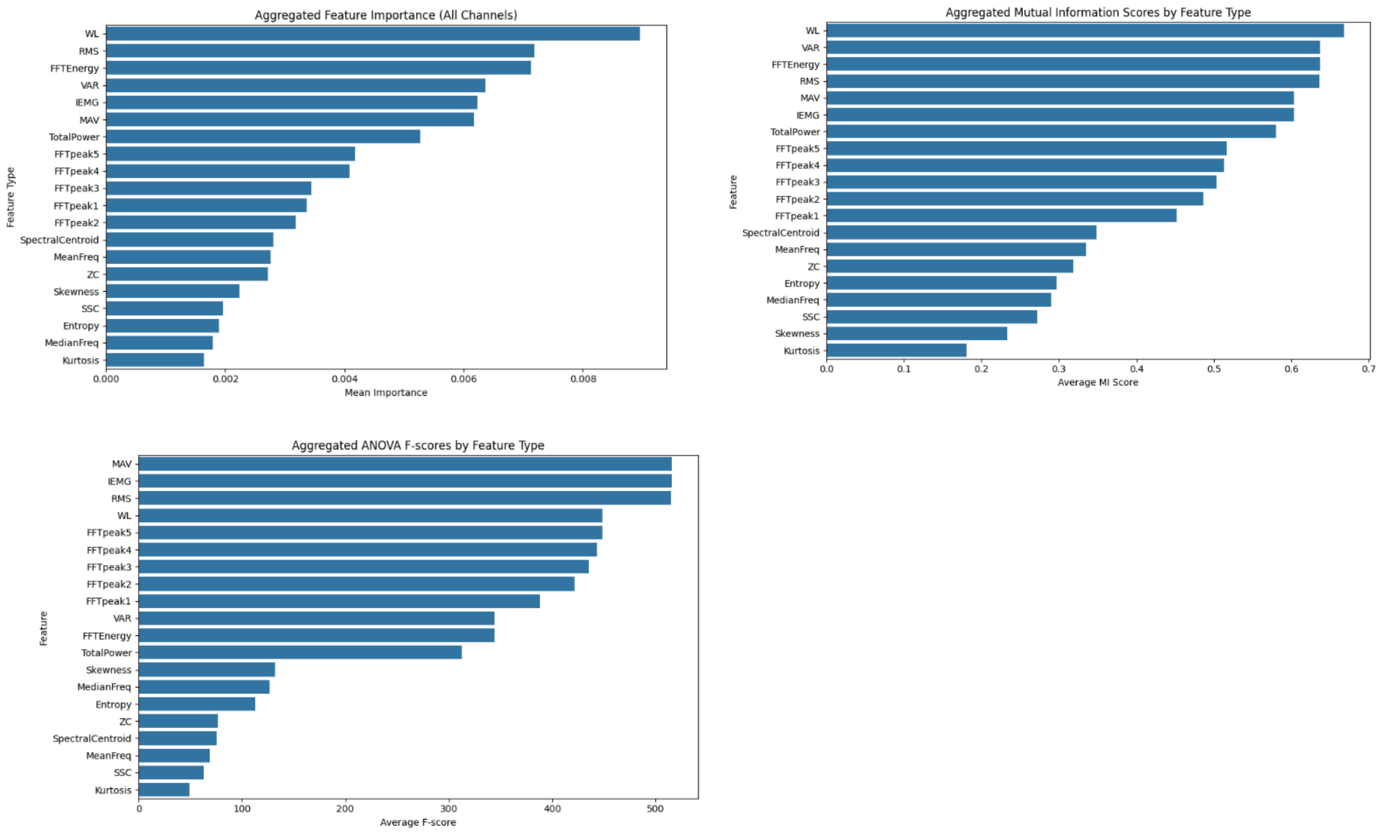

Figure 1 Statistical values of extracted features

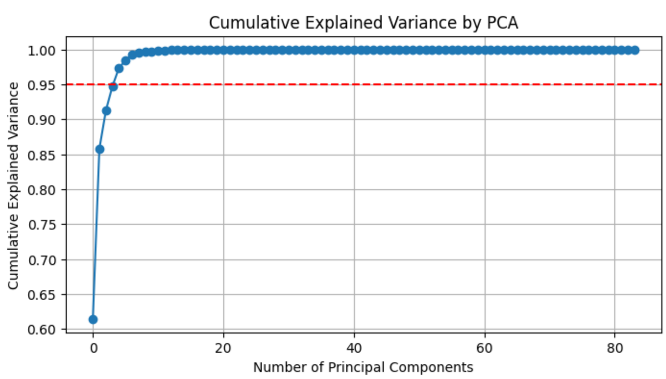

Figure 2 Cumulative explained varience by PCA

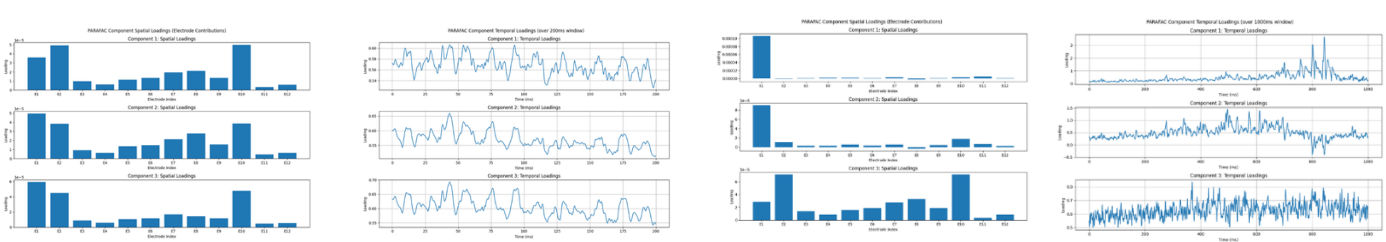

Figure 3 PARAFAC temporal and spatial loading for 200ms and 1000ms window size