

Development of an interactive tool for the visualization and training of muscular synergies

Juan Antonio Gandolfo*
Engineering Department
Universidad Católica del Uruguay
Montevideo, Uruguay
juan.gandolfo@correo.ucu.edu.uy

Melissa Imbert*
Engineering Department
Universidad Católica del Uruguay
Montevideo, Uruguay
melissa.imbertz@ucu.edu.uy

Pablo Iturralde, PhD
Engineering Department
Universidad Católica del Uruguay
Montevideo, Uruguay
pablo.iturralde@ucu.edu.uy

Muscle coordination patterns are crucial for movement control. Events like strokes can disrupt these patterns. This disruption has been correlated with movement dexterity, which implies that the recruitment of muscle coordination patterns may be an effective target for rehabilitation therapies to restore dexterous movement. In this project, a specific tool was developed to evaluate this potential therapeutic effect. Specifically, a human-computer interface was developed, in which a subject's muscle activity, obtained through surface electromyography, controls a two-dimensional cursor on a screen but only if said activity includes the recruitment of specific muscle coordination patterns. Through visualization of the cursor's movement on the screen, the tool allows for the subject to receive real-time feedback about their muscle coordination patterns, establishing a "human-in-the-loop" feedback control system. The developed system consists of four interconnected modules through standard communication protocols. Surface electromyography signals are obtained using sensors from Delsys, Inc., which deliver data through eight channels at 2kHz. These samples are collected by a module responsible for serving them to an upstream processing module. This second module is responsible for detecting muscle activation through envelope detection and determination of cursor displacement through a movement speed controller. Finally, two application modules were designed. The first one utilizes the processed data to generate a two-dimensional game with clear objectives for the user and the second one displays a real-time history of muscle activations providing information about the patient's performance. Future work will validate the usage of this tool for rehabilitation purposes.

Keywords—Muscle Synergies; Surface Electromyography; Human-Computer Interface; Human-in-the-loop Control; Rehabilitation.

I. INTRODUCTION

There is robust evidence that the recruitment of muscles to produce various movements is organized into patterns of muscle coordination, known as muscle synergies (MS) [1]. This enables simpler control of the body's muscles than would be possible if each muscle were activated separately. From an algebraic point of view, this means that muscle activity can be thought of as lying in a manifold of lower dimension than the number of considered muscles, thereby reducing the complexity of movement and functioning as a redundancy mechanism.

In patients who have suffered a stroke or cerebral palsy with chronic movement sequelae, it has been observed that MS are often modified or lost. Strokes are a leading cause of disability. In Uruguay, it is estimated that about 6,000 strokes occur annually, and most leave sequels in patients [2]. This issue creates the need for effective recovery therapies in the

affected. Currently, no commercial tool or therapy targets recovery at the underlying source of the impairment -the neural control of movement-, although some recent research has explored this possibility [3]. Instead, the standard-of-care approach is to assist subjects in replicating movement kinematics, which can help patients with the recovery of mobility and induce healthy-looking movements, but do not necessarily stimulate the recovery of healthy neural control.

This project focused on developing a visualization and a muscle synergy-based control tool, enabling patients to receive feedback about their muscle coordination patterns both visually and through an interactive task. Through this tool, future research can be conducted to evaluate its effectiveness as a rehabilitation therapy.

Cursor-control tools based on coordination patterns from the nervous system have been successfully used with non-human primates (NHP) to elicit novel activity patterns through practice and learning [4] [5]. In those studies, researchers were working directly with neuron action potentials from the central nervous system, which can only be acquired invasively.

Here we merge this approach with the usage of surface electromyographic signals, which have previously been used to enact control of a cursor or device (see e.g. [6]). These signals are acquired over the skin of a subject and reflect the electrical activity of motor units, which are driven by motoneurons from the peripheral nervous system. As such, we still work with output from the nervous system but in a noninvasive manner.

The rest of the paper is organized as follows. In section II, we explain the development of the tool, including first a description of the hardware followed by a description of the architecture of the tool and its technical details. Finally, we present the obtained results in section III and a discussion regarding future work in section IV.

II. METHODOLOGY

The selected system for the sensing was a Trigno Wireless Biofeedback System from Delsys, as the system ensures many of the capabilities desired, like multiple coordinated surface sensors and robust noise filtering. This is a system that transmits data over a serial port connected via USB to the computer. The manufacturer provides a SDK with an API on any programming languages for clients to develop their own software.

Along with the system six sensors were bought. Three different types of sensors were selected, a pair of each type. The models are Trigno Avanti Sensor, with one electromyography (EMG) channel, Trigno Mini sensor, also with one channel, and Trigno Duo Sensor, with two channels;

This Project was financed by the National Research and Innovation Agency of Uruguay (ANII) project FMV_3_2022_1_172858.

* These authors contributed equally.

allowing a total of eight channels for the tool development.

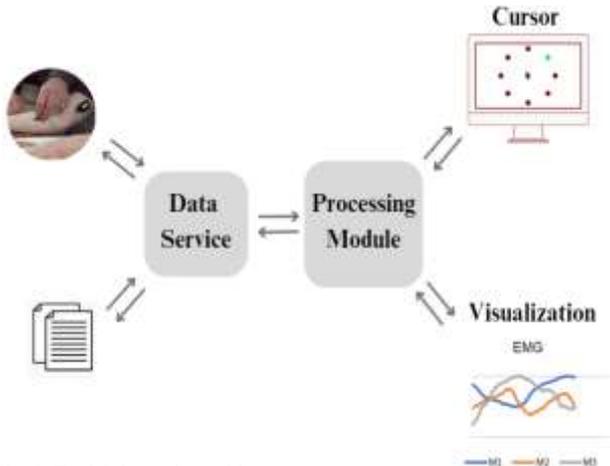


Fig. 1. Tool Schematic Architecture.

The latter two models differ from the first in that, due to their design, they enable more precise location of the sensing electrodes, allowing for measurement of narrower muscles.

In what follows we will explain the design of the tool, starting with the architecture and delving into the implementation details.

A. Architecture

The general architecture follows a modular approach, where the modules in Fig. 1 were defined. It was implemented using Python 3.8 given the versatility to implement both the data handling and the graphic user interfaces and leveraging its high-level programming features.

B. Data Service

The primary purpose of this module was to enable the rest of the tool to be as independent as possible from the sensors used. It allows real-time access to EMG data to the rest of the tool. The data acquisition can be done from the sensors over a serial communication or a .csv file at 2kHz. The last option allows the user to perform simulations.

Additionally, a Control Panel to configure the sensors and manage the flow of the tool is displayed in this module. The framework chosen was Pyside2 [7].

C. Processing Module

This module was implemented to run in the background, acting as an intermediary between the Data Service and the interactive interfaces. It is responsible for processing the data and detecting the synergies. The first step is to rectify the signal, for which we implemented a full-wave rectifier. Then, the signal is passed through a low pass filter, implemented using the SciPy library [8]. The decision was to use a second-order Butterworth with a cutoff frequency of 40Hz, as the signal information typically is found in the 20Hz and lower frequencies [9].

The energy of the signal does not depend solely on the muscle size, but also on the positioning of the electrodes relative to the fibers of the muscle, among other factors. To maintain the relative weight of each muscle sensed the same, normalization was performed. Both the baseline noise and the peaks of the signal were estimated for each muscle in a

calibration stage, and the resulting values were used to normalize the signals to the 0-1 range.

Matrix factorization was used to extract muscle synergies. The nature of the muscle activations is non-negative and so should be the synergy activations. Considering this restriction, many matrix factorization algorithms were investigated. Taking advantage of the experience of Torres-Oviedo [1], the non-negative Matrix Factorization (NNMF) algorithm was used. Many Python libraries implement this algorithm, but the preferred one was Sklearn [9] for its broad documentation.

$$M_{m \times n} \cong W_{m \times k} \times H_{k \times n} \quad (1)$$

Equation (1) describes linear dimensionality reduction by factorizing the matrix M as the sum of k outer vector products of appropriate dimensionality, where k is an integer that defines the dimensionality of the model. In this application, M contains the muscle activations, determined by the number of data points (m) and the number of muscles (n), and k corresponds to the number of synergies. The resultant matrix H represents the base synergies, with their activations given by W.

To identify the synergy base that best explains the muscle activations, factorization was performed by varying the number of synergies k up to the number of muscles and calculating the Variance Accounted For (VAF) of the model as shown in equation 2. This information insight allows the user to select the optimal synergy base to work with.

$$VAF = 1 - \frac{\|M - W \cdot H\|_F^2}{\|M\|_F^2} \quad (2)$$

D. Cursor

This module implements a visual interface that updates the position of a cursor in two dimensions. This implementation was done using the Pygame library [10], as it has a small learning curve and is as efficient as needed.

In this game, a cursor is controlled by activating specific MS, with visual goals to reach. The movement of the cursor was controlled with a human in the loop strategy, where the controlled variable is the displacement of the cursor. The strategy is hence a speed control, as the user is trying to control the position by varying its speed.

To calculate the displacement of the cursor, two projections are implemented by the Processing Module. The first one is a transformation from muscle activations to synergy activations. This is calculated by projecting each sensed data point to the synergy base calculated in the calibration stage. This synergy activation data point is then projected to the cursor direction matrix. This is a matrix calculated to enable the user to select in which direction the synergy is going to pull.

To ensure all the user's muscle activations are reflected in the cursor movement, the displacement is integrated until the cursor module is ready to receive new data. When the module requests the displacement, the integration variable is sent and immediately reset to ensure no data carries over to the next request.

E. Visualization

This module is a visual interface that graphically represents muscular activity. It shows the real-time level of muscle and synergy activation. Subsampling at 100Hz was implemented before transmitting the data from the Processing Module, and a refresh rate of 60Hz was selected considered enough for the human eye. The framework utilized was Bokeh [11] which allows for efficient data management with many visualization options.

F. Communications

The modules communicate with each other via the Transmission Control Protocol (TCP), following a Server-Client architecture. The Data Service acts solely as a server and it only accepts one connection, which is the Processing Module. The Processing Module acts both as a client to the Data Service and as a server to the graphic user interfaces (GUIs). Both GUIs connect over the same socket but are handled in different ports to ensure no collisions. The client and server functionalities are run in different threads to ensure that both can run simultaneously and independently.

Finally, both GUIs are implemented as clients, requesting the necessary information to the Processing Module as soon as possible. To implement that, two frameworks were used. Socket [12] for connecting the modules over TCP/IP and Msgpack [13] for serializing the data. The second framework was chosen, given its capabilities to efficiently pack floating points as well as dictionaries.

III. RESULTS

Fig. 2 shows a capture of a session with eight muscles and four synergies. Each muscle activation is shown as a bar plot in the top right quadrant, with a four-second history in the top left quadrant. Similarly, the activation of the selected synergies is shown in the bottom right quadrant, with a four-second history in the bottom left quadrant. In the middle, a legend identifies the traces of the historical graphs by color and provides the names of the bar plots. It also allows the user to select which muscles or synergies to display.

Fig. 3 shows the composition of the synergy basis employed in the experiment mentioned above. The basis comprises four synergies, each represented by a bar chart. In each chart, the bars correspond to the relative contribution

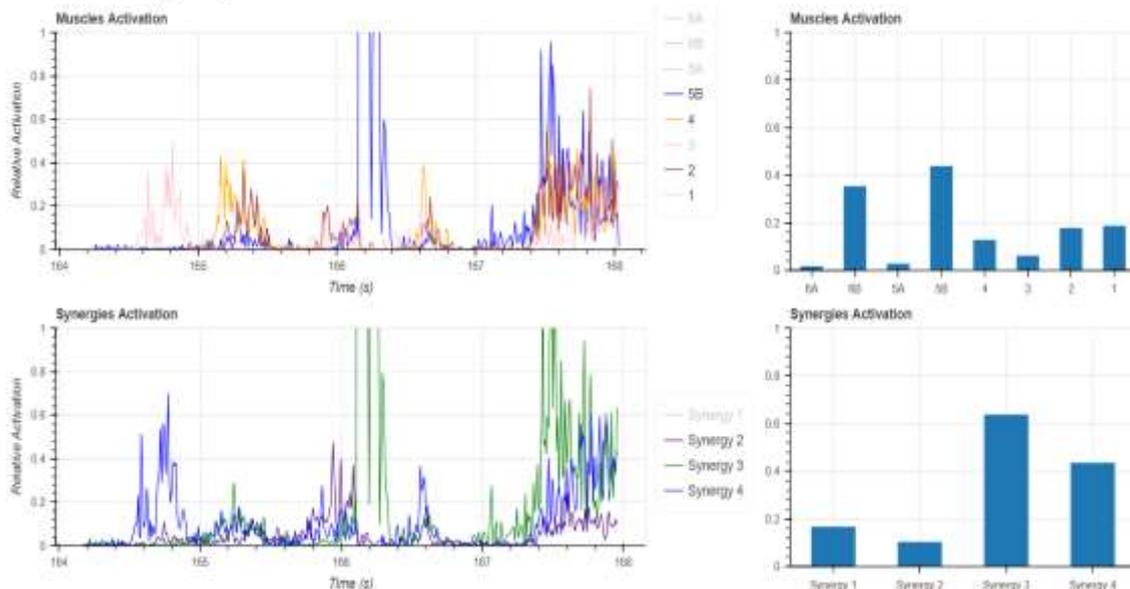
(weight) of muscle activation within the respective synergy. The activations plotted, which represent the relative activation, range between 0 and 1 as the data is normalized. For the historical graphs, the x-axis represents the actual timestamp of the data received by the visualization module. This functionality was validated by several individuals, qualitatively ensuring that real-time visualization was achieved in a smooth and user-friendly way.

Fig. 4 shows a capture of the cursor game screen. The game consists of eight square objects arranged in a circle: seven red squares represent enemies, while the green square represents the target for the user to reach. The cursor to be controlled by the user is displayed as a white square. Each time the cursor reaches the target, the position of the target is randomly reassigned.

Control of the cursor was validated by several individuals by activating the muscles constituting each selected synergy and confirming their correct projection. Different projection mappings were created using up to eight sensors and varying muscle synergies. Initially, simple synergies composed of one or two muscles were used for intuitive learning and clearer muscle-cursor relationships. As users progressed, more complex projections involving multiple muscles increased the challenge, demonstrating gradual adaptation to cursor control through visual feedback from the tool.

The muscle synergy estimation algorithm was validated using synthetic data generated from a predefined synergy basis. The temporal activation of the synthetic synergies was constructed using phase-shifted, half-wave rectified sinusoidal functions, ensuring coordinated synergy activation as in natural movements. Variations in activation amplitudes and harmonics up to 20 Hz were introduced to capture the dynamic nature of muscle activations and enhance the complexity of the electromyographic signals (EMG). The muscle activation matrix was derived from equation (1), with noise generated from a normal distribution added to simulate physiological measurement noise.

Algorithm performance was quantified using the cosine similarity metric. Obtained values were consistently above 0.7, with some exceeding 0.9 for specific vectors. This process validated the algorithm's capability for precise matrix factorization, modeling muscle activation through the



decomposition into muscle synergies and their activation. It is essential to note that “synergies” here refer to factors obtained through matrix decomposition.

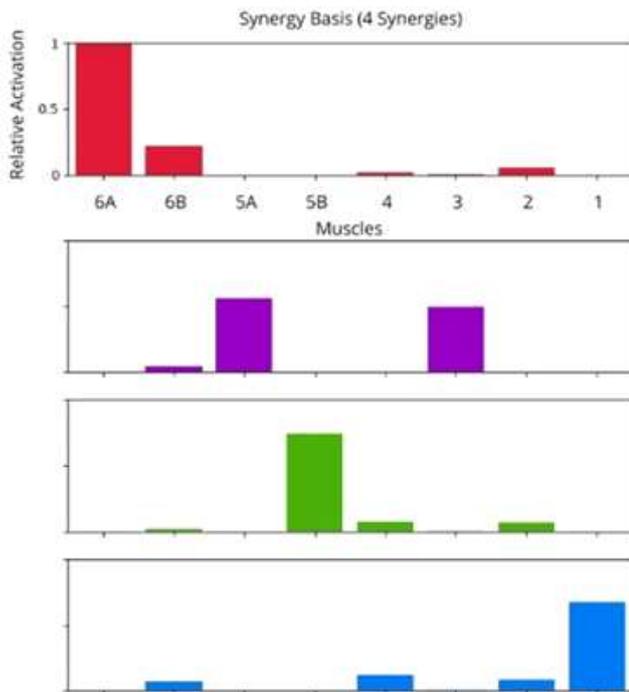


Fig. 3. Synergy basis used in the experiment shown in Fig. 2.

IV. CONCLUSION

Even though the validation and characterization phase is ongoing, it is already possible to state that the tool enables the user to control a cursor by activating muscle synergies that can be automatically detected or selected by the user. It allows the user to understand how each synergy is conformed, and which synergies are being recruited at any given time through the visualization module. In the final phase of the project, the tool is being validated by allowing healthy people to try and control the cursor. The objective is to find both the tool’s intrinsic limitations and the limitations that may arise from the users’ ability to learn novel muscle activity to cursor movement maps. Learning limitations will be tested by giving the users arbitrary mappings from synergy activity to cursor control and measuring the learning curve. Following this validation, we will conduct a clinical evaluation of its therapeutic potential to help patients re-acquire natural

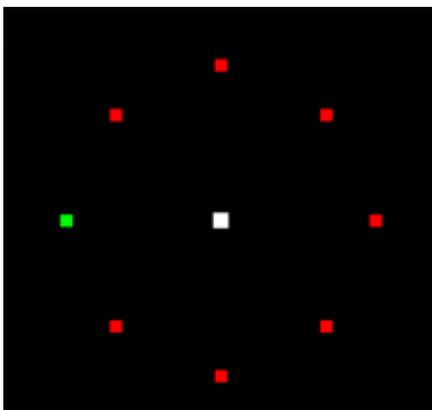


Fig. 4. Interactive game. The white square is the player, the green one is the objective, and the red ones are enemies to be avoided.

muscle coordination patterns. This novel therapy will provide patients with feedback regarding their muscle coordination instead of the more traditional feedback based on the movement achieved. We expect that this feedback, which is richer and more tightly linked to the neural control of movement, may facilitate recovery in some patients who require motor therapy.

V. REFERENCES

- [1] G. Torres-Oviedo, J. M. Macpherson and L. H. Ting, "Muscle Synergy Organization Is Robust Across a Variety of Postural," *J Neurophysiol*, Atlanta, Georgia, 2006.
- [2] MSP, "Ministerio de salud pública," 29 Octubre 2021. [Online]. Available: <https://www.gub.uy/ministerio-salud-publica/comunicacion/noticias/29-octubre-dia-mundial-prevencion-del-acv>.
- [3] N. J. P. Seo, A. . M. Barry, M. P. Ghassemi, K. M. M. Triandafilou and M. E. P. Stoykov, "Use of an EMG-Controlled Game as a Therapeutic Tool to Retrain Hand Muscle Activation Patterns Following Stroke: A Pilot Study," *Journal of Neurologic Physical Therapy*, vol. 46, no. 3, pp. 198-205, 2022.
- [4] E. R. Oby, M. D. Golub, J. A. Hennig, A. D. Degenhart, E. C. Tyler-Kabara, B. M. Yu, S. M. Chase and A. P. Batista, "New neural activity patterns emerge with," *PNAS*, New York, 2018.
- [5] P. T. Sadtler, K. M. Quick, M. D. Golub, S. M. Chase, Stephen, E. C. Tyler-Kabara, B. M. Yu and A. P. Batista, "Neural constraints on learning," *Nature*, 2015.
- [6] S. M. O'Meara, M. C. Shyr, K. R. Lyons and S. S. Joshy, "Comparing Two Different Cursor Control Methods which Use Single-Site Surface Electromyography," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2019, pp. 1163-1166.
- [7] QT for Python Team, "Pyside2 Documentation," 14 Enero 2022. [Online]. Available: <https://pypi.org/project/PySide2/>.
- [8] The SciPy Community, "Scipy Documentation," 24 June 2024. [Online]. Available: <https://scipy.org/>.
- [9] scikit-learn developers, "scikit-learn Documentation," 2024. [Online]. Available: <https://scikit-learn.org/stable/modules/decomposition.html>.
- [10] "Pygame Documentation," 2024. [Online]. Available: <https://www.pygame.org/docs/>.
- [11] "Bokeh Documentation," 2024. [Online]. Available: <https://docs.bokeh.org/en/latest/>.
- [12] Python Software Foundation, "Socket Documentation," 13 July 2024. [Online]. Available: <https://docs.python.org/3/library/socket.html>.
- [13] S. Furuhashi, "Msgpack Dcoumentation," 2021. [Online]. Available: <https://msgpack.org/>.