Utilization of EMG Signals for Mouse-less Cursor Control (April 2022)

Samira Kalamdani, Grant Acker, Amy Woo, David Brenton Svacha, University of Michigan – Ann Arbor

***Abstract* – People with constant computer mouse usage often develop Carpal Tunnel Syndrome (CTS), a neuropathic disorder which causes pain and numbness in the median nerve at the carpal ligament. To reduce instances of CTS in people who require constant computer mouse usage, there is a need to develop a mouse-less cursor control system. This system allows users to reduce reliance on currently used computer mouses. In this paper, we introduce a cursor control system that utilizes EMG signals obtained from muscles of both wrists and calves for right and left click, and scrolling up and down. This design would enhance user convenience by utilizing easily controlled muscle groups for cursor control. When the subject flexes the fingers on their right hand, a right click occurs. When the subject flexes the fingers on their left hand, a left click occurs. When the subject flexes their right calf, the cursor scrolls down. When the subject flexes their left calf, the cursor scrolls up. To validate the design, we conducted an experiment that timed how long it takes the subject to carry out a series of specified actions involving cursor control for multiple trials. Overall, we validated our device with a 93.33% success rate.**

Index Terms – *EMG signals, computer mouse control, Carpal Tunnel Syndrome, LabVIEW design, Python*

I. INTRODUCTION

Carpal Tunnel Syndrome (CTS) is a compression neuropathy of the median nerves that passes through the carpal tunnel located at the wrist, which causes discomfort and numbness throughout the hand. It has been highlighted that computer mouse usage is one of the main causes of CTS development; exposure to repetitive hand use and hand forces from constant computer mouse usage increases pressure to the carpal tunnel [1]. This is especially problematic as today 4 to 10 million people are affected by CTS each year in the United States alone [2]. To prevent CTS, reducing wrist usage and alleviating force put on the median nerve at the carpal nerve are necessary. Thus, there is a need for a cursor control device that does not involve poor wrist posture that increases pressure at the carpal tunnel.

Our mouse-less cursor control design specifically targets the ergonomic objective of reducing wrist usage during cursor control. It is designed to incorporate various muscle groups in our body with a user-friendly interface: for example, left click corresponding to left-hand movement. To create this cursor control system, there were three main technical objectives. The first objective was to design a system which isolates and conditions different EMG signals from four muscle groups (two hands and two calves). The second objective was to develop a LabVIEW VI for signal acquisition and analysis from the signal output received from the DAQ card. The last objective was to develop a python code which aligns the EMG signals with corresponding mouse actions.

II. EXPERIMENTAL SETUP

The system we made was divided into three major subsections. The first subsection included EMG signal acquisition and conditioning hardware, which includes a gain stage and a filtering stage in order to filter and amplify the EMG signals. The second subsection included a LabVIEW VI which handled EMG signal processing in preparation for the third subsection, which included a Python program that translated the LabVIEW output into real-time computer monitor commands. This entire process can be visualized in the flow diagram below in Fig. 1.

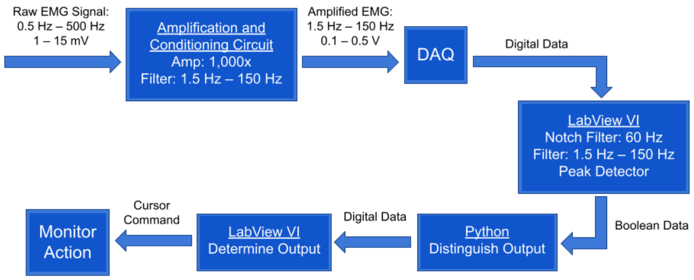

Fig. 1 – Process visualization of the EMG cursor control system

The EMG signal was acquired from four both wrists, and both calves (Fig. 2). Before attaching electrodes to each region, we prepped the subject's skin with alcohol wipes. A ground electrode was applied to the subject on the elbow and connected to the breadboard in the ground input to serve as a common ground for all of the electrodes. Each of these key regions had two electrodes applied to the area of interest, typically at the site where the muscle contracts the most, approximately 1 cm apart from each other. Each of these electrode pairs were fed into the AD620 differential amplifier stage, where the difference in signal can be acquired.

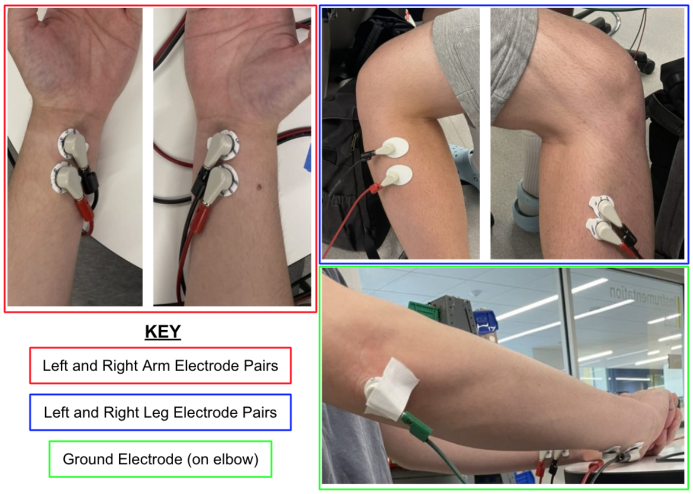

Fig. 2 – Compilation of electrode pair placement on our subject

A. Conditioning Circuit

The EMG hardware consists of the surface snap electrodes which attach to the subject in the key signal acquisition areas, followed by an AD620 differential amplifier gain stage, and a quad-input LM324AN operational amplifier. Each of these stages had appropriate resistors and capacitors whose element values were configured to have gains of 100x and 10x over the AD620 and LM324AN stages respectively for a combined gain of 1000x. This allowed for the 1-15 mV EMG signals to be amplified to about 0.1-0.5 V to make the signals more distinguished and to utilize more of the range of the DAQ board. Additionally, the LM324AN stage would filter out acquired EMG signals, whose frequency range is from 0.5 to 500 Hz, down to a frequency bandwidth with cutoff frequencies at 1.5 and 150 Hz. The AD620 stage, followed by our LM324AN stage, comprised each of the channels which conditioned our EMG signal, totaling to four channels. The configuration for these stages of the conditioning circuit are available in Fig. 3.

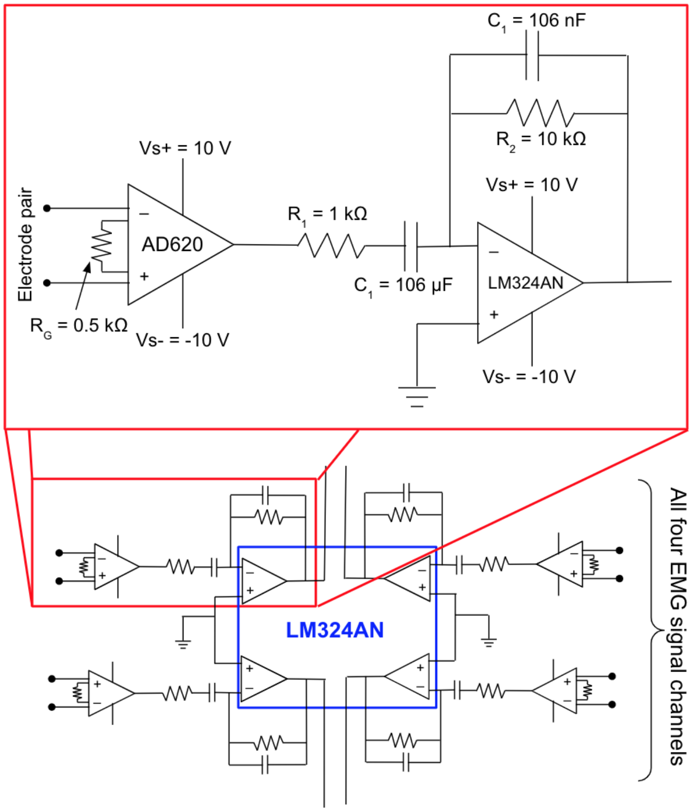

Fig. 3 – Circuit diagram of AD620 gain stage and LM324AN filter stage

Bode plots from all four of the filters in the LM324AN stage can be found in Fig. 4 and the actual low and high frequency cutoffs in the hardware can be found in Table I. The error (%) for all the circuit components is low, so our circuits performed as expected.

B. LabVIEW VI

The four signals were acquired via the DAQ and further filtering was applied to each signal for better visualization and analysis. In LabVIEW, a band notch filter with cutoff frequencies of 55 Hz and 65 Hz filtered out the 60 Hz power line noise out of each signal. Additionally, a bandpass filter with cutoff frequencies 50 Hz and 150 Hz filtered out any excess noise that remained after filtering in the signal conditioning circuits. The filtered signals were displayed in real time. Threshold Detector VI was configured to detect the occurrence of a specified peak voltage in each signal. The specified peak voltage corresponded to a muscle flexion event that was detected via the surface electrodes on the subject. The threshold level selection is described in the calibration section. The output of each Threshold Detector VI was a boolean value that indicates whether a peak occurred or not. This boolean output was the input for the Python module that allowed our system to perform an executable action.

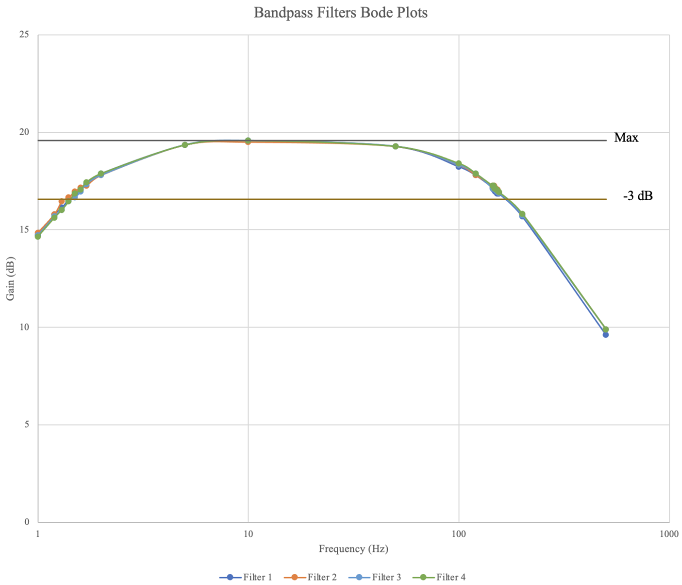

Fig. 4 – Bode plots for all four LM324AN filter stages used to condition our EMG signals. The designed low and high cut-off frequencies are 1.5 Hz and 150 Hz, respectively.

TABLE I
AMPLIFIER AND FILTER MEASUREMENTS AND ERRORS

Specifications	(Theoretical Value) Measured Value	Error (%)
AD620 gain (dB)		
Differential Amplifier #1	(40 dB) 40.34 dB	0.85
Differential Amplifier #2	(40 dB) 40.34 dB	0.85
Differential Amplifier #3	(40 dB) 40.34 dB	0.85
Differential Amplifier #4	(40 dB) 40.34 dB	0.85
LM324AN gain (dB)		
Filter #1	(20 dB) 19.57 dB	2.15
Filter #2	(20 dB) 19.50 dB	2.5
Filter #3	(20 dB) 19.57 dB	2.15
Filter #4	(20 dB) 19.57 dB	2.15
Low-pass filter cut-off		
Filter #1	(1.5 Hz) 1.43 Hz	4.67
Filter #2	(1.5 Hz) 1.34 Hz	10.67
Filter #3	(1.5 Hz) 1.46 Hz	2.67
Filter #4	(1.5 Hz) 1.44 Hz	4
High-pass filter cut-off		
Filter #1	(150 Hz) 164 Hz	9.33
Filter #2	(150 Hz) 172 Hz	14.67
Filter #3	(150 Hz) 163 Hz	8.67
Filter #4	(150 Hz) 165 Hz	10

C. Python

Our system was able to interact with the computer's operating system via the use of a Python .py file that we incorporated in our LabVIEW VI via the use of the built-in Python node. The Python file uses functions from the installed 'mouse' package to control the normal operations of the mouse. The file is composed of four functions, one for each action. Each function uses if/else statements to check the inputted boolean value and execute the action if the conditions are met. To provide further confirmation that the system was functioning as intended, the Python file also returned an integer '0' when a peak was detected and a non-zero integer when a peak was not detected. This indicator can be seen in Fig. 6 for each of the four designated actions.

In LabVIEW itself, the Python node is placed inside the while loop to ensure that the system will execute the correct action until the system is turned off. The nodes work by referencing a single function in the .py file, so four nodes were used, one for each signal.

The logic and progression of our LabVIEW VI is displayed in our functional block diagram below in Fig. 5 with the final set of arrows and boxes representing what occurs in our Python component.

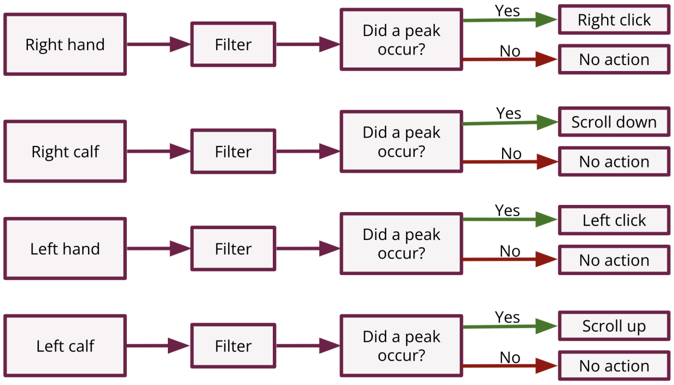

Fig. 5 – Functional Block Diagram for our LabVIEW VI

D. Calibration

An important step for successful use of our device is calibration. The setup of the software allows the user to calibrate the system based on their desired EMG signal strength. This is achieved through setting the threshold that the Threshold Detector VI detects. A subject that has more powerful muscle flexion can increase the threshold, and a subject that has less powerful muscle flexion can decrease the threshold. To calibrate the device, we had the subject perform each action individually: right-hand flex, left-hand flex, right calf flex, and left calf flex. We would observe the amplitude of the signal when each flex occurred. If the subject was satisfied with the amount of flexion necessary, we would choose a threshold value that would detect a peak at the observed amplitude. Table II shows the calibrated threshold level for each signal for our subject.

TABLE II
THRESHOLD DETECTION LEVEL SETTINGS

Signal	Threshold Level (V)
Right Hand	0.03
Left Hand	0.03
Right Calf	0.03
Left Calf	0.02

III. RESULTS

After going through five design iterations, we were successfully able to acquire and filter the four differentiable signals needed, as seen in Fig. 6. Our second objective was to utilize our LabVIEW nodes to read the signals and output a boolean indicator determining if an action has occurred. The LabVIEW VI successfully accomplished this, as evidenced by the on/off indicators shown in Fig. 6. Our final objective was to integrate the use of Python packages in our LabVIEW VI in order to create a VI that was capable of manipulating a computer's mouse control. The VI was successful and enabled a subject to control the right and left clicks of the computer, as well as influence the scroll wheel. For further verification, our LabVIEW front panel depicted a zero when an action was occurring, as seen in Fig. 6.

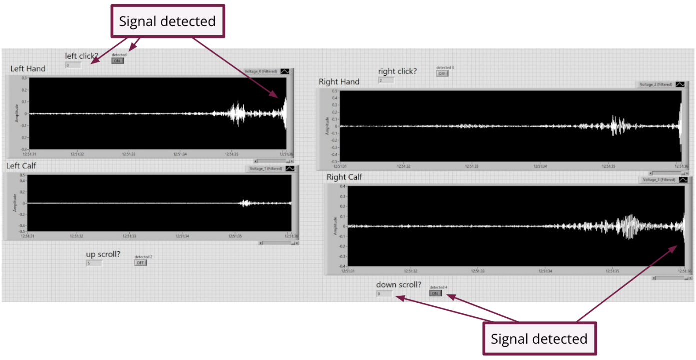

Fig. 6 – Screenshot of the front panel in LabVIEW. In this example, the right calf and left hand have EMG signals which surpass the Peak Detector VI's threshold value, resulting in a left click and down scroll command.

The intent for this project was to provide an alternative for computer cursor control systems which can cause neuropathic disorders such as CTS. The acceptance criterion was that the subject was able to perform a set number of tasks successfully in sequential order with a success rate of at least 90% with a low variability in the amount of time it took to complete across trials.

To test whether our cursor control system was a success, we used a testing protocol where a trial would be counted as being successful if all the sequential tasks were performed correctly. We used the click-score-based website called Cookie Clicker in our protocol [3]. The protocol we used was as follows: 1) scroll down on the Google results page for Cookie Clicker; 2) scroll up to the appropriate link for the Cookie Clicker website; 3) right click on link; 4) left click on link to open the page in a new tab: 5) achieve a score of at least 10 on Cookie Clicker. The subject performed 15 identical trials and the time and success/failure for each trial was recorded. The compilation of these 15 trials were recorded in Fig. 7.

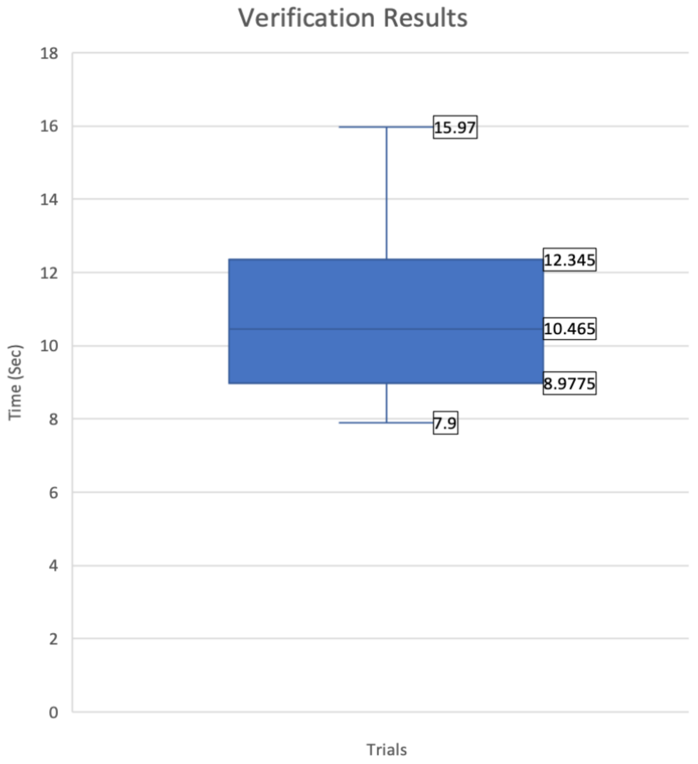

Fig. 7 – Amount of time taken for each trial. Success rate was 14/15 = 93.33%; median time was 10.465 seconds; mean time was 10.898 seconds; standard deviation was 2.378 seconds.

The cursor control system worked very well overall. We were able to perform the intended task with high accuracy, assuming that the system was properly calibrated. Our success rate was 93.33%, which exceeds our acceptance criteria. The mean time of trial completion was 10.898 seconds, with a standard deviation of 2.378 seconds. The low variability confirms the reproducibility of the results, and further verifies the accuracy of the system. As evident in Fig. 5, the signal was very clean with very little noise when the subject was not activating a muscle group of interest.

IV. DISCUSSION AND CONCLUSION

The main objectives of this project were: 1) to design a system which isolates and conditions different EMG signals, 2) to develop a LabVIEW VI for signal acquisition and analysis of the EMG signals, and 3) to develop a python code which corresponds the EMG signals with specific mouse actions. Together, these objectives create a system that will help reduce reliance on a traditional computer mouse, and, therefore, help reduce occurrences of CTS.

Surface electrodes placed on both wrists and both calves of a subject detected isolated EMG signals that corresponded to intentional user-controlled muscle flexions. The signals were sent into four identical signal conditioning circuits that correctly amplified and filtered the signals for analysis.. The conditioned signals were sent to LabVIEW via the DAQ, and the LabVIEW VI acquired, conditioned, and displayed the signals in real time. The LabVIEW VI also generated an output that was usable for a python module. Lastly, the python module accurately linked user-controlled muscle activity to specified computer mouse actions. The success rate of our device was 93.33% with a low variability in time across multiple trials. This verifies the accuracy and reproducibility of our device. Overall, we determined that this project successfully met the three technical objectives we outlined.

A source of error in our experiment is the movement of wires during a muscle flexion. When the subject would flex the fingers on their hands or flex their calves, the wires connecting the surface electrodes to the signal conditioning circuits would occasionally move as well. This would cause inaccurate results and prevent us from observing the actual EMG signal we wanted to detect. To mitigate this, we reduced the amount of slack on the wires by holding them down and taping them to the subject. This greatly reduced the amount of wire movement detected, but some residual movement may have still been detected by the software without us noticing.

V. APPLICATION

Our system can be improved when used in conjunction with an eye tracker system. The eye tracker would be vital in controlling the movement and positioning of the cursor, while our system would allow the cursor to interact with the display by providing inputs for mouse actions. This combination enables those with impaired fine motor skills to still use and operate a computer.

To improve our system itself, we could implement more accurate electrodes to ensure a better recorded signal. Using these electrodes, we could undergo more testing, with different muscle groups, at varied locations on the body to look to minimize the number of limbs our system needs to function. Requiring the use of only one limb is ideal and also makes attaching the electrodes easier. Similarly, we would work towards integrating the electrodes into a sleeve that keeps the electrodes in place without the need for adhesive. Further down the line, we'd want to make the system more portable and accessible by including the circuits and software inside the sleeve. That way, anyone could simply connect the system to their computer wirelessly and wouldn't have to worry about having to be hooked up or having the relevant software downloaded.

ACKNOWLEDGEMENTS

The authors would like to thank the University of Michigan – Ann Arbor's Biomedical Engineering Department for funding this project. Also, thank you to the BME 458 teaching team for support and advice throughout this project with a major shout-out to David Bosek, our favorite instructor's assistant.

REFERENCES

[1] Keir PJ, Bach JM, Rempel D. Effects of computer mouse design and task on carpal tunnel pressure. Ergonomics. 1999 Oct;42(10):1350-60. doi: 10.1080/001401399184992. PMID: 10582504.

[2] American College of Rheumatology. (2021, December). Carpal Tunnel Syndrome. American College of Rheumatology. Retrieved February 16, 2022, from https://www.rheumatology.org/I-Am-A/Patient-Caregiver/Diseases-Conditions/Carpal-Tunnel-Syndrome

[3] Cookie Clicker. (n.d.). Cookie clicker. Cookie Clicker. Retrieved April 13, 2022, from https://cookie-clicker.co/