34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

The simulation of Click and double-click through EMG signals*

Carlos G. Pinheiro, Jr and Adriano O. Andrade

Abstract—**Patients with severe motor impairments, victims of stroke, amyotrophic lateral sclerosis and spinal cord injury are prevented from oral and gesture communication, demanding alternative channels and methods of communication, possibly using a computer. In order to obtain the complete emulation of a standard mouse, the single-click and double-click actions are desirable functionalities. In this study, the implementation of such actions is executed by the analysis of the electromyographic signal recorded from the Frontalis muscle. Muscle activity is discriminated from noise and this information is used to feed a state-machine that in turn decides which action is intended. The method uses an adaptive threshold, which offers freedom for the selection of the parameters of the system. The rate of successfully detected commands found was up to 100% for the single-click and 92% for the double-click. Even though good results were found for double-clicks, the experiment indicate muscle fatigue in the short term. The time response found was below 300 ms suggesting real–time implementation is feasible. Also, other devices can be operated with this approach, if it is accepted as a two symbols system generator.**

I. INTRODUCTION

Several clinical conditions are characterized by severe motor impairment. Spinal cord injury (SCI) and amyotrophic lateral sclerosis (ALS) are the most known, but stroke and others also lead to motor impairment. For patients in extreme conditions, the employment of biosignals to control human.–machine interface (HMI) are of major importance, allowing access to communication, leisure and work opportunities that impact their quality of life. Brain activity, expressed by electroencephalographic signals is used by brain.–computer interface (BCI) devices [1], but usually presents low information transfer rate. Gaze–based devices provide cursor control by means of image analysis or electro–oculographic signals [2]. Even tongue movement has been explored to provide control information [3], [4]. Muscle activity has also been used to control wheelchairs [5] and cursors [6].

Cursor control using electromyographic signals is also possible. In [7] the EMG signal is used to control a cursor, but no mouse click was provided. In this study, it is proposed a method for using EMG signal recorded by a single channel as a source for ON/OFF information to compose the two most important mouse actions (single and double–click) and this approach could be easily extended to a number of applications, such as the control of a virtual keyboard. The devised system employs an extended finite state–machine (EFSM), that presents as advantages the small code and little information necessary for its implementation, being therefore appropriate for real–time applications. The EFSM designed in this study demands fewer parameters when compared to more traditional approaches, such as neural networks, which also demand training. The results obtained show that the approach fits to the problem perfectly, providing a simple and elegant solution.

II. METHODS

The designed system architecture is depicted in Fig. 1. Initially, the raw EMG signal is divided into windows of fixed length. From each window the variance is estimated and later compared to a predefined threshold. If the variance is larger than the threshold, a digital pulse replaces the window. In the next step, the pulse is treated as an event for the state machine, which interprets the command intended by the user.

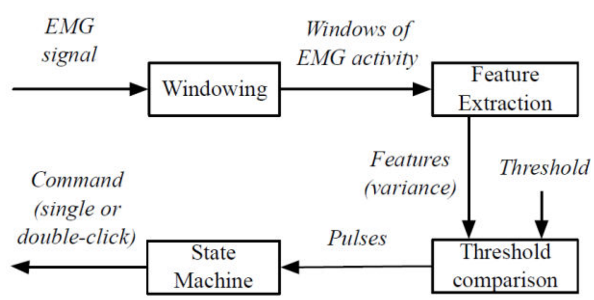

Figure 1. Architecture of the devised system.

A. Protocol for data acquisition

For the evaluation of the implemented digital signal processing tools, EMG signals were collected from a single subject (33 years old; able–body male). The signals were detected, on the *Frontalis* muscle. This muscle is preserved, for example, in high-level spinal cord injury cases. Also, as it is not used during speaking, it allows the user to communicate orally and to operate a device at the same time. The signal was recorded using standard Ag/AgCl sensors with diameter of 1.5 cm and inter–electrode distance of approximately 1 cm. A reference electrode was placed on the shoulder of the subject. The signals were digitized at 600 Hz. The low sampling rate was a limitation found in the device employed for signal acquisition. However, the EMG signal presented enough information to separate muscle activity from noise.

For data collection the subject was asked to quickly raise the eyebrows and execute the four series of actions: (i) one movement every 0.5 second;(ii) one movement every

*Research supported by FAPEMIG, CAPES and CNPq.

C. G. Pinheiro Jr is with Universidade Federal de Goiás (e-mail:cgalvao@eeec.ufg.br).

A. O. Andrade is with Universidade Federal de Uberlândia (e-mail: aoandrade@feelt.ufu.br).

second; (iii) one movement every 2 seconds and (iv) two close movements every second.

The time interval between series of actions was no longer than a minute. Timing was controlled by auditory stimulus generated by customized software. Each series lasted 20 seconds. The dataset available contained EMG signals from 280 actions of one movement and 76 actions of two close movements.

B. Feature extraction and signal windowing

The signal processing consisted in segment the signal in windows of fixed length without superposition. The variance (σ^2) estimated in each window is used to feed the EFSM. The visual inspection of the EMG activity indicated that some movements did not last longer than 100 ms. Therefore, the window size should be smaller than this interval. The values adopted are 20 ms and 50 ms, as studies [8], [9] indicating that it is possible to extract relevant information from EMG signals using those window lengths.

C. Threshold estimate

In order to discriminate noise from EMG activity, we used a single-threshold approach. We used two distinct threshold estimate methods. The first was calculated from values extracted from a period of time without muscle activity (silent period). The threshold (*Th*) is obtained from the variance of N windows:

$$Th=\gamma*\max(\sigma^2_i) \qquad i=1,...,N \tag{1}$$

The value of γ in this study was defined in the interval [2, 70] of integer numbers.

The second method was originally used for heart–beat detection [10]. The dynamic threshold method uses two buffers (with *NBuffer* elements each). One buffer (B_1) is for features of windows considered as noise and the other buffer (B_2) for features of windows considered as EMG activity. When both buffers are full, the threshold is calculated through Equation (2).

$$Th = th_{factor}*MN+(1-th_{factor})*MAC \tag{2}$$

Where MN is the average of B_1 and *MAC* is the average of B_2. The th_{factor} determines the weight of each factor to the final adaptive threshold. Both buffers B_1 and B_2 are updated whenever a new window is available and evaluated. As suggested in [10] the buffer length was set to 8 and th_{factor} to 0.325.

D. Algorithm for command interpretation

Once the EMG signal is detected, it is transformed into a series of pulses, which will be treated as the events of the EFSM used to interprete the command intended by the user. Due to the stochastic nature of the signal, which should be corrected, this series may present erroneous transitions (introducing gaps and fast bursts). The EFSM (Fig. 3) was therefore, designed to execute two activities: (i) to interprete the commands issued by the user and (ii) to act as a post–processor for the EMG detection.

Two variables are used: *td* and *ti*. The variable *td* is the duration of the detected EMG activity. The variable *ti* is the interval between detected muscle activities (Fig. 2).

Furthermore, the EFSM employs three constants for decision–making.

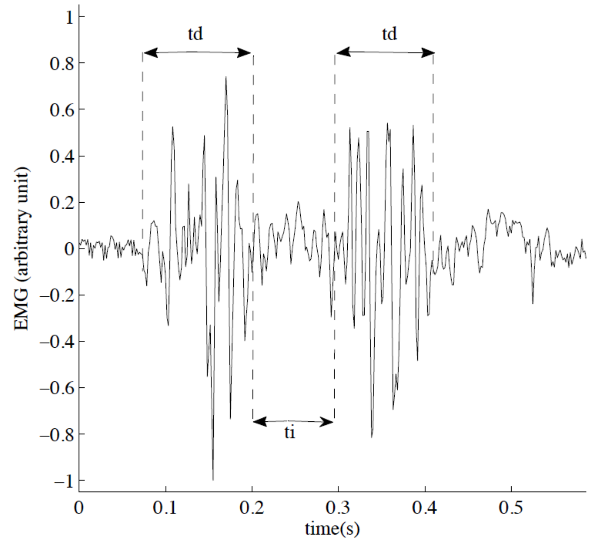

Figure 2. EMG signal and representation of *td* and *ti*

The first is the interval within the same contraction (*ISC*), which is the maximum allowed time interval between two pulses belonging to the same muscle contraction. The second is the noise duration (*ND*), which is the time interval that characterizes noise. If the detected EMG activity does not last longer than *ND*, then the burst is considered to be noise. The last constant is the interval between bursts (*IBB*) and it is analogous to the interval between double–clicks that can be customized in any operating system.

The initial state of the EFSM is *W0*. When *event=1*, it must be verified if *td>ND*. If the condition is satisfied, the burst detected is an EMG activity and the state becomes W3, else, the state becomes *W1*. The *W1–W2* loop was designed to identify eventual false gaps during muscle contractions and also to eliminate short bursts yielded by artefacts. If the state remains at *W1*, then *td<ND* and the signal detected as EMG might actually be noise. If event = 0 when the state is *W1*, then the value of *td* is stored and the state becomes *W2*. When at *W2*, if *event=1* and *ti≤ISC*, then a false gap is detected and the value of *td* is updated with the value of *td* previously stored and added to *ti*, else, if *ti>ISC*, then the previous burst was noise and the test *td>ND* is carried out. When *td>ND* the first click is recognised and the state becomes *W3*. The state *W3* is used to guarantee that the first click is over and it is done by verifying if *ti>ISC*. After this, it is verified if the command is a single–click (Command1) or double.–click (Command2). If *ti>IBB* then the command is interpreted as a single.–click and the state becomes *W0*, else, the state becomes W4. When state is W4 and event = 1, the incoming burst must be evaluated either as noise or muscle activity (the *W5–W6* loop has the same function as the *W1–W2* loop described earlier). When *td>ND*, the double–click is identified, and the state becomes *W7*

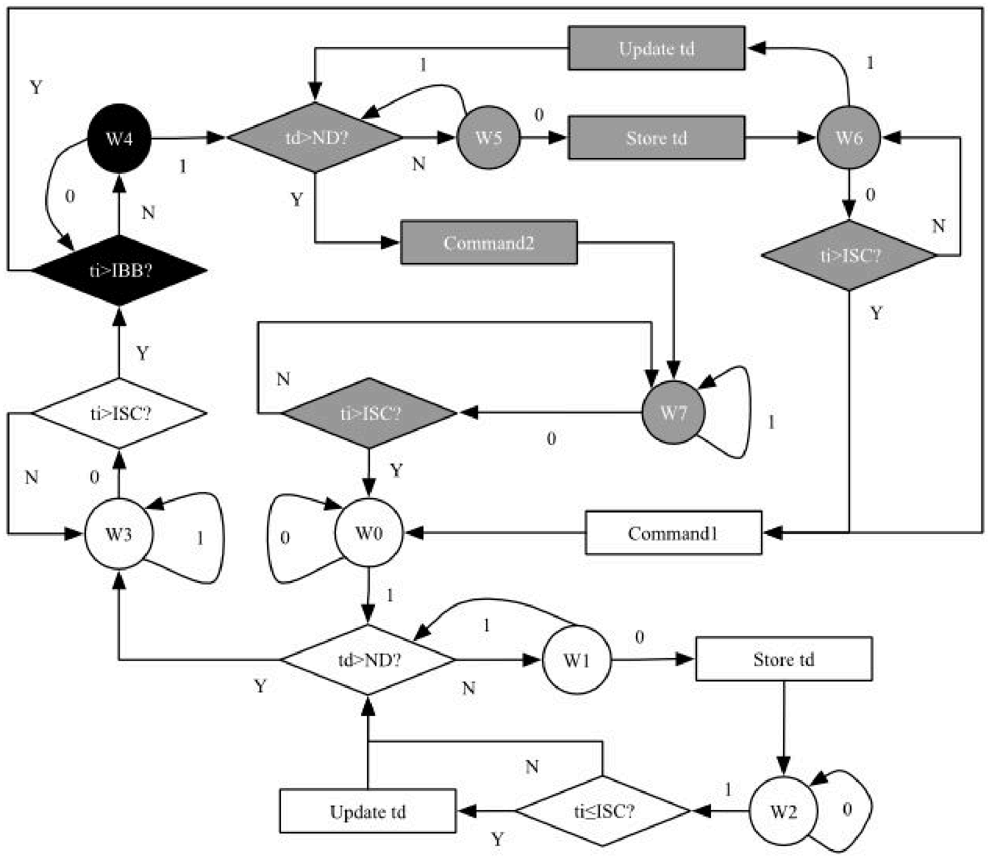

Figure 3. Extended finite state machine for the detection of the command intended by the user. The event considered to state transition is the output of the comparison between the variance of the window and the threshold. The variables td and ti measures the duration of the EMG signal and the interval between detected muscle activities. The constants *ISC* and *ND* are used in the post–processing of pulses. The *IBB* represents the time interval between clicks in the double.–click command. (White elements: detection of first click; Gray elements: detection of the second click; Black elements: detection of both clicks).

returning to W0 after ti>ISC. The other possibility of exiting the W5–W6 loop is the existence of a period of silence superior to ISC, what means that the last burst was noise and the command was actually a single–click, with the state becoming *W0*. With a different approach for click emulation, a simpler EFSM was designed to detect single–clicks only.

In this case, as it is not necessary to discriminate between commands, as soon an EMG activity is detected *(td>ND*), the single–click command is characterized and state becomes *W3*. After a period of silence superior of ISC, the state returns to *W0*.

It is expected for a real–time system that the response to be roughly 300 ms [11]. The latency for detecting *Command1* is about *ND+IBB* and for *Command2*, this delay (*ND+IBB*) must be added to *ND* (time to confirm the second click). However, the EFSM leaves state *W0* only after the EMG is detected and therefore, other variables impact on the time response: threshold and window size. The threshold affects the detection of the end of the EMG signal, and the later it is detected, the larger will be the time response. The effect of the window size is explained in [11]. When using the EFSM for single–click only it is expected that the response time is approximately *ND*.

The value of *IBB* was set to 200 ms, after visual inspection of the EMG signal. Some studies consider that erroneous transitions after the EMG detection present short duration, such as 30 ms in [12]. Therefore the value of *ND* was set to 20 ms, which is the smallest window used in this study.

III. RESULTS

The system performance is related to three measures: *Hit*, *FA* (False Alarm) and response time. The *Hit* is the number of correctly detected commands by the system. The FA is the number of wrongly detected commands. The instant the EMG activity begins was determined by visual inspection and the values stored into an array. This information was used to calculate the response time, which is the interval between the instant the subject started the contraction and the instant the command was identified. To verify the system performance, two datasets were used: one for the design and the second for testing. The results for both sets are summarized in Table I for single and double–click detection. The Table II presents results for single–click detection only. As expected, the response time was considerably lower, as IBB delay is removed.

TABLE I. RESULTS FOUND USING DESIGN AND TESTING DATASETS. THE BEST RESULTS WITH WINDOWS OF 20 ms USED ISC = 80 ms, ND = 20 ms AND γ = 24. WITH WINDOWS OF 50 ms, THE BEST PERFORMANCE WERE ACHIEVED WITH ISC = 50 ms, γ =40 AND ND = 20 ms. FA – FALSE ALARM.

Designing Set				
Command	***Window (ms)***	***Hit(%)***	***FA(%)***	***RT(ms)***
Single-click	20	100.0	5.5	253.9 ± 64.9
	50	100.0	7.3	247.1 ± 84.9
Double-click	20	89.5	0.0	268.8 ± 35.0
	50	78.9	0.0	274.6 ± 66.0
Testing Set				
Command	***Window (ms)***	***Hit(%)***	***FA(%)***	***RT(ms)***
Single-click	20	100.0	3.6	252.8 ± 63.9
	50	100.0	3.6	246.4 ± 87.2
Double-click	20	94.7	0.0	269.8 ± 33.9
	50	89.5	0.0	272.5 ± 64.9

TABLE II. FOR SINGLE-CLIK DETECTION ONLY, Best RESULTS OBTAINED WITH WINDOWS OF 20 ms USED ISC = 80 ms, ND=20 ms AND γ=24. WITH WINDOWS OF 50 ms, THE BEST PERFORMANCE WERE ACHIEVED WITH ISC = 50 ms, γ =40 AND ND = 20 ms. FA .– FALSE ALARM. RT .– RESPONSE TIME

Window(ms)	Hit(%)	FA(%)	RT(ms)
20	100.0	0.0	22.25±10.52
50	100.0	0.0	44.34±21.76

The results reported in Tables I and II used the static threshold. As the signal conditions changes throughout one session, using the adaptive threshold method [10] was proven useful. For γ=2 with static threshold, the *Hit* was 62% and 21% for *Command1 (single-click)* and *Command2 (double-click)* respectivelly. The *FA* was 21% and 98% for *Command1* and *Command2* respectivelly. Also for γ=2, the system using adaptive threshold presented performance of 98% and 86.8% for *Hit* and of 1.8% and 0% for *FA*.

IV. DISCUSSION

The employment of an EFSM proved to fit perfectly to the problem addressed in this study. Added to the intrinsic advantages of an EFSM regarding the requirements of code and information necessary to the implementation, the state–machine designed offered the built–in post–processing capability. Offering the user the possibility of executing two different commands is clearly an advantage but a few remarks should be made. The FA found in Table I is the result of double–clicks issued by the user but recognized as single–clicks. A command misunderstood can be more annoying than a command missed. On the other hand, a double.–click interpreted as a single–click does not pose a serious problem, as no action will be executed by the operating system. The separation of the original dataset into two datasets, for design and test, showed consistent results, with the adoption of 50 ms windows presenting lower performance in the classification task. Surprisingly, the training set presented slightly best results than the designing set, which could be explained by better electrode positioning or even muscle condition.

Furthermore, the response time average found for detecting both single and double.–clicks was below 300 ms, but with some commands being detected after this time interval. On the other hand, detecting only single–clicks is a safer choice regarding response time, being always below 100 ms for every command performed.

V.CONCLUSION

The results indicated the feasibility of using EMG signals to emulate some of the functions found in a standard mouse, considering three key aspects: the choice of a muscle available to most of people with severe motor impairments, the possibility of real–time implementation and intuitive operation. Furthermore, our solution provided a simple and elegant solution, by using a state–machine capable of converting EMG activity into user commands.

ACKNOWLEDGMENT

The authors would like to thank Dr. Pierre Pino from the University of Metz-France, for the collaboration in designing the experimental protocol of this study. The authors would like also to thank the financial support from FAPEMIG, CAPES and CNPq.

REFERENCES

[1] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G. and Vaughan, T. M.,.“Brain.–computer interfaces for communication and control.”, Clinical Neurophysiology, vol. 113, pp. 767.–791, 2002

[2] Jacob, R. J. K.,.“Eye movement-based human.–computer interaction techniques: Toward non-command interfaces.”, Advances in Human-Computer Interaction, vol. 4, pp. 151-190, 1993

[3] Huo, X., Wang, J. and Ghovanloo, M., .“A wireless tongue-computer interface using stereo differential magnetic field measurement.”, in Proc. 29th Annual International conference of the IEEE EMBS, 2007, pp. 5723.–5726

[4] Huo, X., Wang, J. and Ghovanloo, M., .“A magneto-inductive sensor based wireless tongue-computer interface.”, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 16, 497.–504, 2008

[5] Tsui, C. S. L., Jia, P., Gan, J. Q., Hu, O. and Yuan, K., .“EMG.–based hands.–free wheelchair control with EOG attention shift detection.”, in Proc. International Conference on Robotics and Biomimetics, 2008, pp. 1266.–1271

[6] Chin, C. and Barreto, A., .“Neural control of the computer cursor based on spectral analysis of the electromyogram.” in Proc. 2nd International IEEE EMBS Conference on Neural Engineering, 2005, pp. 446.–449.

[7] Huang, C. -N. and Chen, C. -H. and Chung, H. -Y., .“Application of facial electromyography in computer mouse access for people with disabilities.”, Disability and Rehabilitation, vol. 28, pp. 231.–237, 2006.

[8] Xu, L. and Adler, A., .“An improved method for muscle activation detection during gait.”, in Canadian Conference on Electrical and Computer Engineering, vol. 1, 2004, pp. 357.–360.

[9] Kumaran, R. S., Narayanan, K. and Gowdy, J. N., .“Myoelectric signals for multimodal speech recognition.” in Proc. INTERPEECH (Eurospeech), 2005, pp. 1189.–1192.

[10] Patrick S. Hamilton (2002), Open source ECG Analysis Software. [Online] Available: http://www.eplimited.com/confirmation.htm.

[11] Englehart, K. and Hudgins, B., .“A robust, real-time control scheme for multifunction myoelectric control.” in IEEE transactions on Bomedical Engineering, vol. 50, 2003, pp. 848.–854.

[12] Bonato, P., D.’Alessio, T. and Knaflitz, M., .“A statistical method for the measurement of muscle activation intervals from surface myoelectric signal during gait.”, IEEE Transactions on Biomedical Engineering, vol. 45, 1998, pp. 287.–299.