Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

A simple highly efficient non invasive EMG-based HMI

N.Vitiello, U.Olcese, C.M.Oddo, J.Carpaneto, S.Micera, M.C.Carrozza, P.Dario

ARTS Lab
Scuola Superiore Sant'Anna
Pisa, Italy

n.vitiello@sssup.it

Abstract - **Muscle activity recorded non-invasively is sufficient to control a mobile robot if it is used in combination with an algorithm for its asynchronous analysis. In this paper, we show that several subjects successfully can control the movements of a robot in a structured environment made up of six rooms by contracting two different muscles using a simple algorithm. After a small training period, subjects were able to control the robot with performances comparable to those achieved manually controlling the robot.**

Index Terms – EMG signals, finite state controller, real-time control

I. Introduction

Several works showed how biopotentials can be successfully used to control a great number of mechatronic devices. EEG signals were used to control a mobile robot in an indoor environment [1]. EMG signals were used to decode human intentions in order to control prosthetic devices [2-4] and exoskeletons for teleoperation, muscle powering and functional rehabilitation in [5-8]. EMG signals can be analyzed in different ways, ranging from threshold analysis [2] to pattern classification and fuzzy logic [9-10].

In order to elaborate EMG signals, we propose to use a statistical pattern recognition system which is based on three easily obtainable features: square mean value, standard deviation and kurtosis index.

Here we show that such a system can be robust enough to control efficiently a mobile robot in an indoor environment with several rooms, corridor and doorways, shown in Figure 1. This environment was developed by using wooden plates, so that it could offer a good resistance to the impact of the robot.

The EMG analysis and elaboration was based on the extraction of features from the signals. It has been implemented by using a *ad hoc* C/C++ routine and subdivided in two steps: calculation of centroid of each pattern during a calibration phase, and classification of the features vector by minimizing the euclidean distance between the centroid and the vector itself.

II. Methods

The first step in our work has been the development of a control algorithm to be implemented on the robot's microcontroller. A finite state algorithm, which was successfully used to operate a mobile robot [1] and applied to EMG-based control [2] has been developed.

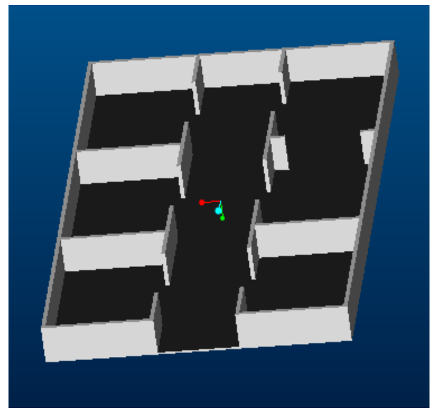

Fig. 1 CAD model of the experiment indoor environment (250x150 cm).

Figure 2 shows the essential of the finite state automaton. We decided to implement a four state controller, which would only need two commands, i.e. the active of relaxed state two muscles, extracted in real time from the EMG signals.

The four states (or behaviors) implemented in our controller were "STOP", "FORWARD", "LEFT TURN" and "RIGHT TURN". There are then four different inputs the subject can provide the robot by combining muscle contractions appropriately.

In fact, when the robot is in the "STOP" state, the 0 input means it has not to change its behavior. If command 1 is given, the robot starts turning left and continues while input 1 is provided. Only command 0 makes the robot stop. This way the effect of an undesired pulse during a turning state is minimized. Analogously, from "STOP" state, input 2 makes the robot "TURN RIGTH". Input 3 makes the robot move forward. In order not to fatigue subjects, contraction must not be kept during forward motion. Any input apart from 3 maintains the robot in the "FORWARD" state. So, a 3 single input means "forward" and a subsequent 3 single input means "stop".

Our robot was equipped with contact sensors, mounted in its frontal part, in order to avoid dangerous collisions. As soon as an obstacle was detected, the robot stopped moving and went backward for one second at slow speed.

A. *Mobile Robot*

A Japan Robotech LTD© RDS-X01 mobile robot was used during our experiments. It was equipped with two independent DC motors, which were used at 50% PWM Duty Cycle, so that

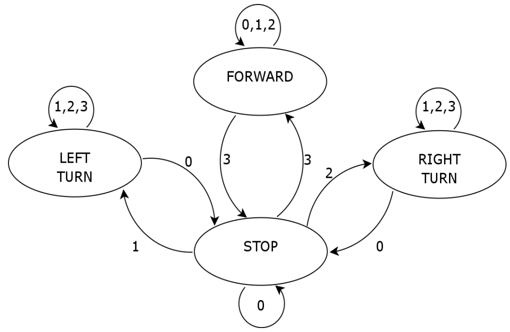

Fig. 2 Finite state automaton used for the control of the robot. The collision detection routine is not shown for the sake of simplicity.

the maximum speed of the robot was approximately 20 cm/s, quite fast but still easy to be controlled. Two contact sensors were mounted in front of the robot for low level control.

In order to implement communication between the PC, on which a C/C++ routine ran, and the robot we connected one analogical input port of the robot to one analogical output port on the NI 6025 board. The analogical input to the robot was then interpreted as a two bit digital one simply using thresholds, as reported in table 1.

The program implemented in the robot's microcontroller consisted of a loop in which our finite state controller was inserted. The first step of the loop was checking if the contact sensors had sent any signal. If so, wheels were moved backward for one second, then the "stop" state was set. Then, according to the state in which the robot was and the input voltage value, state was changed and wheels moved. If the robot had to move left, left wheel was moved backward for one second, while right wheel was moved forward for the same period. Then the program checked for new inputs. An analogous behaviour was implemented for right turns. In case the robot was ordered to move forward, both wheels started moving forward, and every 0.5 seconds the robot checked for new inputs after a 0.5 seconds "no listen" state; this last state was used only during the forward state in order to give the subject enough time to relax the muscles after cocontraction, preventing unwanted stops of the robot. This way, if the next input was 3, the robot would stop; otherwise it would keep on moving. This allows single contractions to be detected. Without this 0.5 seconds pause a single contraction would have immediately stopped the robot.

B. *EMG Signals and system architecture*

To acquire the EMG signals we used the commercial device NORAXON TeleMyo™ 2004T. Five long-term monitoring Ag/AgCl surface-EMG electrodes (Kendall, USA) are used, two on each biceps and one as reference, on the elbow. Detection sites were located according to standard anatomical text [11-12]. The signals were then transmitted to a PC by a NI PCI-6025E ADC device, digitalized at 1024 Hz and passed to the C/C++ routine developed for signal classification and robot control.

We chose to put electrodes on both arms' biceps, so that we could avoid interference as much as possible. Using closer

TABLE I
INPUT ENCODING

Calculated *class*	RELAX	LEFT BICEP	RIGTH BICEP	COCONTRACTION
Automaton input	0	1	2	3
Robot input thresholds	0-0.875	0.876-1.875	1.876-2.875	2.876-5
DAQ analogic voltage output	0	1	2	3
Remote controller voltage output	0	1.2	2.4	3.6

muscles, such as an extensor and its related flexor, could have made the robot easier to be operated, but cross-talking could have emerged. At the same time, this choice is very interesting because in most of the case also quadriplegics can use these muscles voluntarily (if the spinal injury is not too high).

C. *Classifier*

The classifier, implemented in the C/C++ routine, is a typical NN (Nearest Neighbour) statistical algorithm [13]. For two EMG sampled signals $X_1(nT)$ and $X_2(nT)$, where T is the sampling interval and n indicates the n[th] sample, the algorithm is structured in the following two steps.

1) *Calibration*: In this phase the centroid of each class is calculated. For two muscles there are four classes: RELAX, none muscle is contracted, LEFT BICEP, only left bicep is contracted, RIGHT BICEP, only right bicep is contracted, COCONTRACTION, both biceps are contracted. The centroid of the class *i* is a six component vector C_i defined as:

$$C_i = \frac{1}{M}\sum_{m=1}^{M} F_m \qquad (1)$$

where M is the number of non overlapped temporal windows on which C_i is calculated. F_m is a six component vector of features of the m[th] time window. For a single N sample temporal window, F is defined as:

$$F = \left[|\mu|_1,\ \rho_1,\ k_1,\ |\mu|_2,\ \rho_2,\ k_2\right]^T \qquad (2)$$

where $|\mu|_i$ is the mean absolute value (the offset has previously been subtracted from the signal), ρ_i is the variance and k_i is the kurtosis index, referred to signal i.

In this phase, in order to evaluate centroids, subjects were asked to perform the actions corresponding to each of the four classes: three times for each of them. A single action was 2.048 s long. C/C++ routine divided this interval into 8 non overlapped temporal windows, each of them made of 256 samples. This strategy was adopted in order to increase the variability of the EMG signals for the different classes. All calculated features were normalized in the range $[0,1]$; this is exemplified for the mean value as follows:

$$|\mu_i|_{norm} = |\mu_i| / |\mu_i|_{\max} \qquad i = 1,2 \qquad (3)$$

TABLE II
EXAMPLE OF A TYPICAL CALIBRATION MATRIX

Class	RELAX	LEFT BICEP	RIGHT BICEP	COCONTRACTION
$\lvert\mu\rvert_1$	0.0784	0.3733	1.0000	0.9821
ρ_1	0.5376	0.6318	1.0000	0.9674
k_1	0.7818	0.9039	0.9520	1.0000
$\lvert\mu\rvert_2$	0.0930	0.7292	0.2121	0.8774
ρ_2	0.3178	1.0000	0.1634	0.6819
k_2	0.6686	1.0000	0.6720	0.9278

Subsequently centroids were calculated as in (1).

2) *Classification*: After the calibration, the system can classify new signals. C/C++ routine enters a *while loop*: every loop is used to examine a new 256 samples non overlapped temporal window. The feature vector F is then calculated as in (2). The decision criterion, according to which the routine classifies EMG signals, is to find the minimum euclidean distance between F and each of the four centroids previously calculated.

Once $class$ is found the C/C++ routine puts on the analogical output of the NI-DAQ board a corresponding value as shown in Table I. The found $class$ determines the automaton input.

D. Human-Machine Interface Protocol

After positioning electrodes on subjects, they were asked to perform some contractions in order to give our software enough data to classify EMG signals. An interactive interface printed on screen the actions subjects should do (e.g., “Left biceps contraction”).

Subjects were then told how to control the robot following the finite state control method and the input encoding shown in Table I. Subjects were left some time to learn how to control the robot by making it wander around the environment. Training time was recorded. When a subject thought to be ready, training was stopped and the task could begin.

The task consisted in moving the robot through the environment, making it visit every room in a predefined order and then moving it back to its starting position. Task time, number of command given and number of errors (contractions which did not cause the corresponding action, i.e. they were not properly evaluated by the classifier) were recorded during the task. Subjects had to declare what they wanted the robot to do before performing any action. If the robot did not follow the given command, an error was registered.

After the task was performed, subjects were asked to make the robot move along the same path once more, but controlling it with a remote controller, in order to compare times and accuracy. This controller had two buttons corresponding to the two muscles. By bypassing our software it was possible to evaluate how it affected the overall performance.

Five healthy male subjects were involved in the trials, 19 to 24 years old. Informed consent was obtained from all subjects.

TABLE III
EMG TASK RESULTS

	Training Time (s)	Task Time (s)	Number of commands	Percentile error
Mean	770	271.6	94.8	3.41%
Std	338.4	62.6	26.5	1.84%

TABLE IV
REMOTE CONTROLLER TASK RESULTS

	Training Time (s)	Task Time (s)	Number of commands
Mean	28	127	56.8
Std	6.7	8.4	14.6

III. RESULTS AND DISCUSSION

We observed that subjects could control the robot in a proper way with a very short training; in that period, obviously, errors occurred more frequently respect to the subsequent phase, given that the subject had to learn how to obtain contractions as similar as possible to the ones made during the calibration phase. In a few cases, we also observed that, if during the calibration contractions were made with the maximum strength, some problems could occur due to fatigue. For this reason, subjects were recommended not to use their maximum strength during contraction, but to act so that contractions could be as much replicable as possible.

An important parameter is the time needed by the subject to perform the task. We compared it with the time needed to perform the task with the remote controller. The latter was comparably faster than the former, but it is important to underline that the number of commands provided to the robot in order to achieve a complete task was far smaller than the number of needed muscle contractions. This was due to the greater accuracy the remote controller could provide, as discussed further on. Another parameter we considered, as shown in table III, is the number of wrong actions in respect to the declared ones. The operator had to declare the desired movement, and wrong ones were registered. This parameter is useful to evaluate both calibration and training. If both were done appropriately, we would expect few errors, otherwise calibration had to be repeated or training had to be continued.

A final important result to underline is that subjects were always able to rotate the robot an angle smaller than 10°, and to move it forward about 10 cm, as a minimum. This means that the lag time is about 0.5 s, considering a speed of about 20 cm/s. Obviously, the lag time had to be greater than 0.25 s, which was the width of a single temporal window, but we observed that the delay introduced by the algorithm, and the one introduced by the subject, are reasonably small in respect to the signal observation time. The remote controller did not introduce any delay. In fact, the delay was only due to the acquisition system. The absence of any delay meant more accuracy when using the remote controller and, consequently, task could be completed faster and with fewer commands.

A. *Classification results*

Table II shows an example of the values we obtained for the calibration matrix, calculated as explained in section II.C. All six values which compose a C_i vector contribute in making the four centroids different.

B. *Task results*

In Table III it is possible to see task and average results for all the subjects. The parameters shown are training time, task time, number of commands, number of errors and percentile error. Table IV instead shows task time and number of steps when subjects used the remote controller.

Training time was always between 5 and 10 minutes. This means that, if calibration is carried on accurately, our system is
very easy to use and intuitive as well. It is a very important result, for similar systems [1] have much longer training times. In fact subjects only had to learn how to perform contractions in order to get the corresponding commands, as using the mobile robot was very easy.

Task time was, on average, four and a half minutes. When subjects used the remote controller, it reduced to about two minutes. These values have to be considered together with the number of steps. On average, they were about 95 when using muscle contractions and 55 when using the remote controller. Adding the presence of the lag time, this means that the two different times are comparable.

The average percentile error was about 3.5%. This means that, when the subject is properly trained and classification was accurate, this EMG Human Machine Interface is highly efficient, for almost all commands given are correctly understood by our C/C++ routine. We did not recorded errors in the task performed using the remote controller. This was due to the ease of using this device.

IV. Conclusions

The system we developed and presented in this manuscript has shown good results in terms of the extraction of voluntary motor commands. By elaborating EMG signals and classifying them with a Nearest Neighbour algorithm, we have succeeded in allowing some previously untrained subjects to move a mobile robot. Subjects were able to use the robot within minutes from their first approach with that, thanks to the intuitive driving method and the efficiency of the classifier we developed. The tests we carried out to prove our system show that subjects were able to drive the robot without making a significative number of mistakes, i.e. without having their commands understood erroneously by the C/C++ routine.

Further improvements could regard both the classifier and the hardware we used. A more efficient method for analyzing EMG signals could make use of neural networks or fuzzy systems. Moreover, overlapping temporal windows could be used, so that lag time could be reduced. Wireless communication between the robot and the PC could then be implemented, in order to let the robot move in a broader environment and make subjects feel more comfortable.

Our system could be used in many applications, substituting manual control or adding extra capabilities. For example such an EMG Human Machine Interface would be useful for activities performed in hostile environments, such as extra vehicular activities in space programs.

Other applications could include prosthetics, master-slave systems and devices to aid personal motion like wheelchairs.

Acknowledgments

The authors would like to thank Ettore Cavallaro for his useful suggestions and all the students of Scuola Superiore Sant'Anna for their enthusiastic participation to our experiments.

References

[1] J. del R. Millàn, F. Renkens, J. Mouriño and W. Gerstner, "Noninvasive brain-actuated control of a mobile robot by human EEG", *IEEE Trans. Biomedical Engineering*, vol. 51, no. 6, June 2004.

[2] M. C. Carrozza, G. Cappiello, G. Stellin, F. Zaccone, F. Vecchi, S. Micera and P. Dario, "On the development of a novel adaptive prosthetic hand with compliant joints: experimental platform and EMG control", *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005),* pp.1271 - 1276, 2005

[3] D. Nishikawa, W. Yu, H. Yokoi and Y. Kazaku, "EMG prosthetic hand controller using real-time learning method", *Proc. IEEE Int. Conf. Syst Man Cybern* Vol. 1, pp. I-153-I-158, 1999.

[4] Yonghong H, Englehart KB, Hudgins B, Chan ADC, "A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses", *IEEE Trans on Biomedical Engineering,* vol. 52(11*)*, pp. 1801-1811, 2005.

[5] Kiguchi K, Iwami K, Yasuda M, Watanabe K, Fukuda T, "An Exoskeletal Robot for Human Shoulder Joint Motion Assist", *IEEE/ASME Trans Mechatronics* Vol. 8, n° 1, pp. 125 -135, 2003.

[6] Rosen J, Brand M, Fuchs MB, and Arcan M, "A Myosignal-Based Powered Exoskeleton System", *IEEE Trans Systems, Man, Cybernetics - Part A: Systems and Humans*, vol. 31, n° 3, pp. 210-222, 2001.

[7] O. Fukuda, T. Tsuji, M. Kaneko and A. Otsuka, "A human-assisting manipulator teleoperated by EMG signals and arm motions", *IEEE Trans. Robotics and Automation,* vol. 19, no. 2, April 2003

[8] K. A. Farry, I. D. Walker and R. G. Baraniuk, "Myoelectric teleoperation of a complex robotic hand", *IEEE Trans. Robotics and Automation,* vol. 12, no. 5, October 1996

[9] W. Putnam and R. B. Knapp, "Real-time computer control using pattern recognition of the electromyogram", *Proc. 15th Annual Int. Conf. Engineering Medicine and Biology Society*, pp.1236-1237, Oct. 1993.

[10]A. B. Ajibove and R. F.ff. Weir, "A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control", *IEEE Trans. Neural Systems and Rehabilitation Engineering* Vol. 13, I. 3, pp:280 - 291 Sept. 2005.

[11]J. Basmajian, A. Blumenstien, Eletrode Placement in EMG Biofeedback, William & Wilkins, 1980.

[12]J. F. Cram and G. S. Kasman, Introduction to Surface Electromyography, Aspen Publishing, 1998.

[13]Carrozza MC, Cappiello G, Cavallaro E, Micera S, Vecchi F, Dario P., "Design and control of an underactuated cybernetic artificial hand", *10th International Symposium on Robotics and Applications - World Automation Congress (ISORA-WAC 2004)*, Sevilla, 2004