Applied Bionics and Biomechanics
Vol. 7, No. 3, September 2010, 199–207

Taylor & Francis
Taylor & Francis Group

Interface based on electrooculography for velocity control of a robot arm

Eduardo Iáñez[a*], José M. Azorín[a], Eduardo Fernández[b] and Andrés Úbeda[a]

[a] *Virtual Reality and Robotics Lab, Miguel Hernández University, Elche, Spain;* [b] *Bioengineering Institute, Miguel Hernández University, Elche, Spain*

(*Received 31 January 2010; final version received 18 June 2010*)

This paper describes a technique based on electrooculography to control a robot arm. This technique detects the movement of the eyes, measuring the difference of potential between the cornea and the retina by placing electrodes around the ocular area. The processing algorithm developed to obtain the position of the eye at the blink of the user is explained. The output of the processing algorithm offers, apart from the direction, four different values (zero to three) to control the velocity of the robot arm according to how much the user is looking in one direction. This allows controlling two degrees of freedom of a robot arm with the eyes movement. The blink has been used to mark some targets in tests. In this paper, the experimental results obtained with a real robot arm are shown.

Keywords: electrooculography (EOG); robot; disabled people; human interface

1. Introduction

Nowadays numerous aid systems exist for persons with disabilities, which are intended to achieve better mobility of the affected person or facilitate the control of devices. For example, to improve mobility there are automated wheel chairs, robotic prostheses for the extremities (Jeong et al. 2000) or exoskeletons (Sankai 2006). In addition, to control devices there are techniques to detect the eye position using a camera (Hutchinson et al. 1989) or brain–machine interfaces (BCI: brain computer interface), which allow controlling things from the cursor of a computer to mobile devices (Millán et al. 2004, 2005; Iturrate et al. 2009).The main goal of these systems is to enhance the quality of life of disabled people, increasing their independence and granting greater social inclusion. In this paper, the use of a robot arm is proposed to help disabled and/or elder people.

Some of these interfaces use the movements of the eyes in order to control a device, such as videooculography (VOG) or infra-red oculography (IROG) (Hutchinson et al. 1989; Oyekoya and Stentiford 2006; Úbeda et al. 2009). These techniques can be used to detect the eye position using a camera.

In this paper, the electrooculography technique (EOG) is used to detect eye motion (Iáñez et al. 2009). In this technique a camera is not used to detect the eye position. The EOG detects the movement of the eyes by measuring through electrodes the difference of potential between the cornea and the retina (Nicolau et al. 1995).This technique has some important advantages related to the techniques that use a camera. With EOG, the eye motion can be detected independently of light conditions. In addition, EOG does not depend on head orientation; on the contrary, in VOG or IROG the performance depends on the head position with regard to the camera. Furthermore, in EOG the eye motion can be detected even if the eye is partially closed; in VOG and IROG the camera must have direct vision of the open eye.

The EOG interface has been used in several works to interact with devices (Úbeda et al. 2010), or for guidance of a wheelchair (Barea et al. 2003). It has also been used to control mobile robots (Ho and Sasaki 2001) and use blinking to control its movement (Duguleana and Mogan 2010). These technologies could eventually allow people with severe disabilities to control robots in order to help them in their daily life activities.

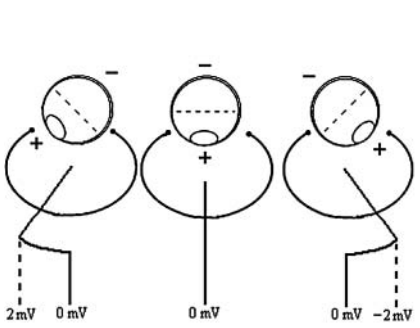
This paper aims to develop a processing algorithm that allows obtaining the position of the eye with the EOG technique. This algorithm allows obtaining four different values (from zero to three) for each direction (up, down, right and left) and the combination of them to achieve diagonals. Using these values, the control of two degrees of freedom of a robot arm with different velocities has been done. The blink of the user is also detected in order to mark some targets during the tests when the robot is over them.

This paper is organised as follows. Section 2 explains the theory of electrooculography. In Section 3, the hardware and processing algorithm are described. Experimental results obtained from the training with a graphical interface and a real robot arm are shown in Section 4. Finally, the main results are summarised in Section 5.

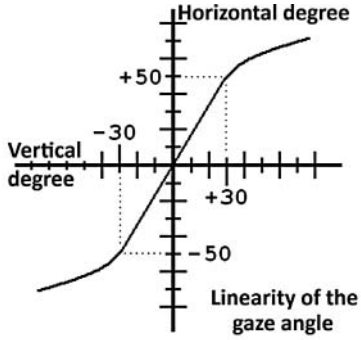
*Corresponding author. Email: eianez@umh.es

ISSN: 1176-2322 print / 1754-2103 online

DOI: 10.1080/11762322.2010.503107
http://www.informaworld.com



(a) Ocular Dipole.



(b) Linearity of the horizontal and vertical gaze angles.

Figure 1. Electrooculography.

2. Electrooculography

EOG is one of the most commonly used methods of registering eye movements (Sivert and Jacob 2000; Usakli et al. in press). It is based on the fact that the eye acts as an electrical dipole between the positive potential of the cornea and the negative potential of the retina (Nicolau et al. 1995). Thus, in normal conditions, the retina has a bioelectrical negative potential related to the cornea. For this reason, rotations of the ocular globe cause changes in the direction of the vector corresponding to this electric dipole (Figure 1a). The recording of these changes requires placing some small electrodes on the skin around eyes. The EOG value varies from 50 to 3500 μV with a frequency range of about DC-100 Hz between the cornea and the Bruch membrane located at the rear of the eye (Barea et al. 2002). Its behaviour is practically linear for gaze angles of $\pm 50^{\circ}$ horizontal and $\pm 30^{\circ}$ vertical (Chen and Newman 2004) (Figure 1b).

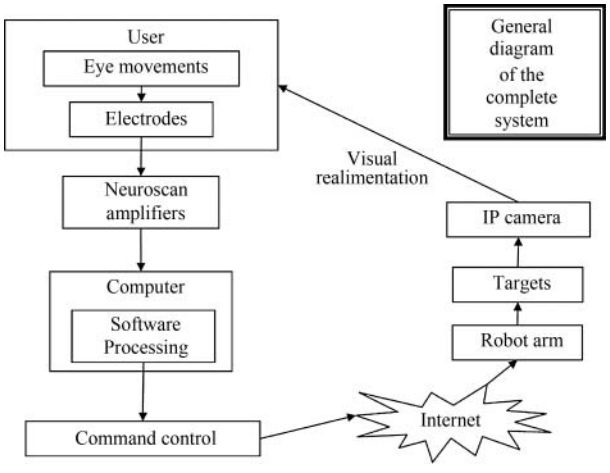


Figure 2. General diagram of the complete system.

3. Human–robot interface

This section describes the hardware and software architecture used to measure the EOG signals encoding the horizontal and vertical positions of the eyes. The hardware configuration and the processing algorithm developed are described. Finally, the performance of the velocity control is presented.

In Figure 2, a general diagram of the architecture is shown, and a real image of the local environment with the user, the acquisition device and the computer can be seen in Figure 3.

3.1. Hardware

EEG signals have been registered through the commercial device NeuroScan® (Figure 3). This device has two bipolar channels for EOG measurement. A sample frequency of 300 Hz has been used. To obtain the data on the computer and be able to process it in real time, the application programming interface (API) of Matlab offered with the device has been used. The processing algorithm has also been developed in Matlab.

Before placing the electrodes the skin is cleaned with an abrasive gel then a conductor gel, to improve the quality of the signal and reduce the impedance between the skin and the electrodes, is used to place the electrodes, as shown in Figure 4:

- In order to detect horizontal movement, two electrodes are placed on the right and the left of the eyes, respectively, Horizontal_Right (HR) and Horizontal_Left (HL).
- The vertical movement is detected by placing two electrodes on the top and bottom parts of the eye, respectively, Vertical_Up (VU) and Vertical_Low (VL).
- The reference electrode (REF) is placed approximately on the forehead (FPz position of the International System 10/20; American EEG Society 1991).

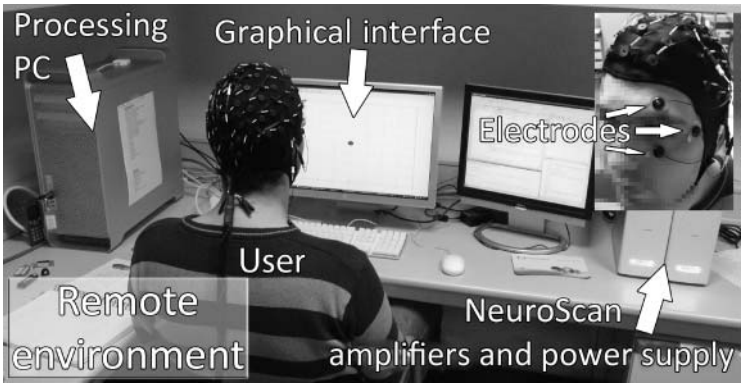


Figure 3. Real image of the local environment with the user, the acquisition hardware and the computer.

3.2. Thresholds

As indicated in Section 1, an application for controlling a robot arm with different velocities has been developed. Therefore, before explaining the processing algorithm it is necessary to explain the thresholds selected for this purpose. Four different thresholds are used:

- The first threshold is used to differentiate a rest state (when the user is looking to the centre) or the noise from a movement; therefore, the output value is zero.
- Every value of the signal between the first and second thresholds generates an output value of one.
- Every value of the signal between the second and third thresholds generates an output value of two.
- And every value of the signal between the third and last thresholds generates an output value of three. If the value of the signal is greater than this threshold, a blink is detected and the blink flag is activated.

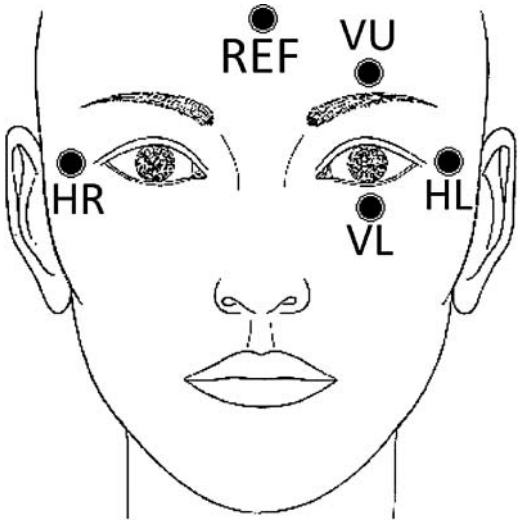


Figure 4. Position of the electrodes on the face around the eyes. Reference electrode (REF), HR (horizontal right) and HL (horizontal left) for horizontal movement, VU (vertical up) and VL (vertical low) for vertical movement.

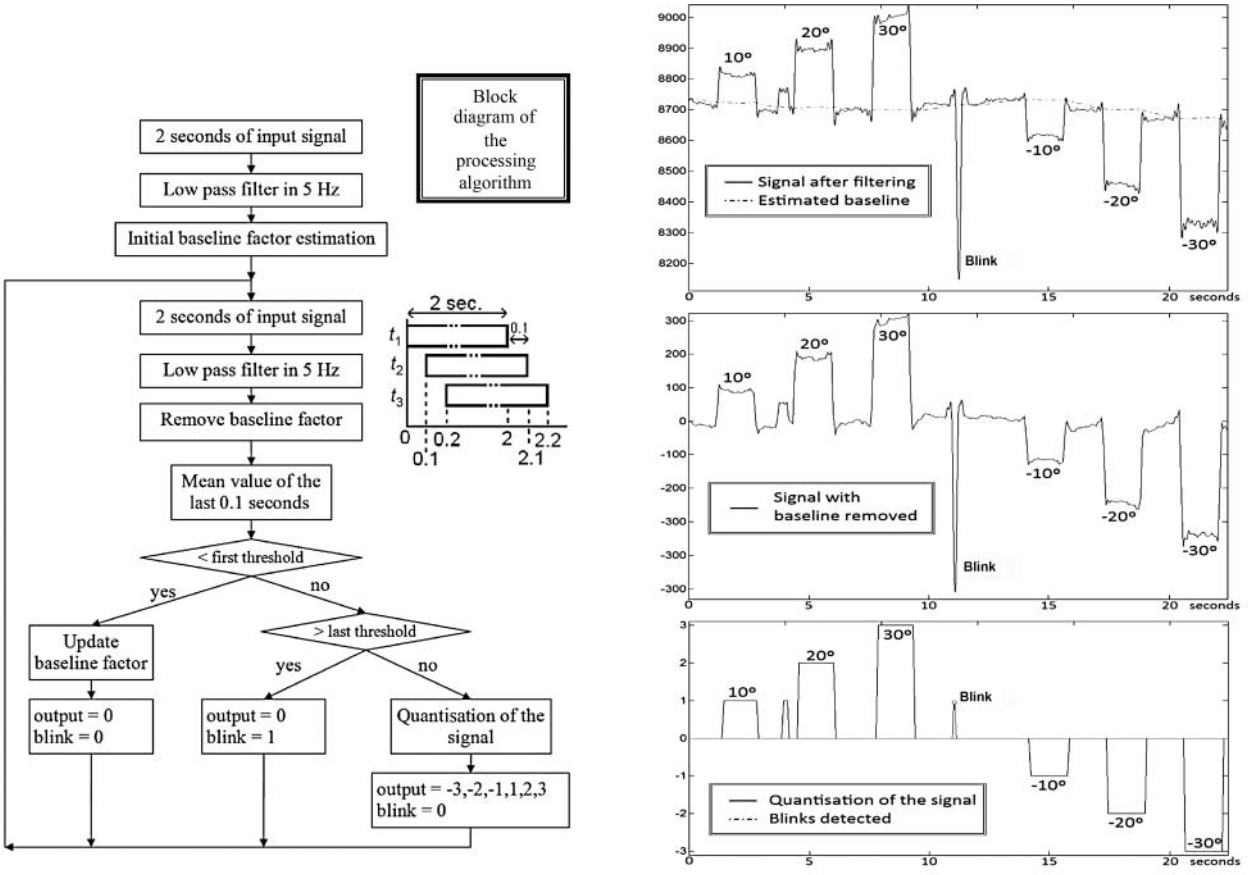
The thresholds explained are for one channel and for one direction. The opposite direction has negative thresholds. Each channel has a total of eight thresholds. An example of these thresholds is:

- horizontal thresholds (mV): [−650, −300, −150, −60, 50, 105, 230, 650];
- vertical thresholds (mV): [−650, −120, −60, −30, 30, 60, 120, 650].

Moreover, thresholds can vary depending on the person and the channel (horizontal and vertical). For example, as the vertical movement of the eye has a smaller range the vertical thresholds are smaller than the horizontal thresholds. As explained in Section 4.2, a function that allows automatic calculation of thresholds has been developed.

3.3. Processing algorithm

In previous works a system with an imposed high pass filter of 0.5 Hz was used (Iáñez et al. 2008). Because of this filter, part of the information of the EOG signal is lost, as it is only able to detect abrupt changes of the gaze. In this paper, the signal has been registered from DC, without a high pass filter, allowing the detection of the degree of the gaze. Now, if a user looks in a direction and keeps the gaze, the signal remains stable until the user returns the gaze to the centre. Even so, it is an inconvenience when processing the signal because the signal has a DC level, the baseline, which is not constant over time. This DC level changes over time due to factors such as the situation of the electrodes, their impedance, or movements of the user. On that basis, a processing algorithm that is able to estimate the baseline in real time in order to subtract it and gain the original signal, not losing the information about the gaze angle of the user, has been developed.



(a) Diagram of the EOG-processing algorithm to obtain the robot arm command from the eye movement.

(b) Evolution of the signal in different steps of the processing algorithm.

Figure 5. Processing algorithm.

In Figure 5 a diagram of the processing algorithm and an evolution of the signal in different stages of this processing are shown. The algorithm is explained below.

In the beginning of the test, it is indicated to the user to look at the centre of the screen. These initial instants are used to calculate the baseline factor. Therefore, the first two seconds of the signal are registered, a low pass filter of 5 Hz is applied and the result is averaged. This value is the baseline factor that is used and updated in the rest of the algorithm.

Following this, the program starts a loop taking decisions about the gaze direction and the position of the eye each 0.1 seconds. Even so, an overlap of 1.9 seconds has been taken, giving a signal of 2 seconds to work with. This has been done because filtering only 0.1 seconds (30 samples at a frequency sample of 300 Hz) provides only a few samples and the results obtained for the filtering are not correct.

The next steps have been applied to each 2 second fragment of signal, and each 0.1 second:

- The 2-second signal is filtered with a low pass filter of 5 Hz. An example of this signal is shown at top of Figure 5(b).
- The last 30 samples (0.1 seconds) are selected.
- The baseline factor is subtracted from these samples. The signal is now centred at zero. The baseline factor is shown at the top of Figure 5(b) and the signal after subtracting the baseline factor is shown in the middle.
- The result is averaged and this value is checked with the thresholds explained before.
  - ⋆ If the value does not exceed the first threshold, the user is looking to the centre in a rest position of the eyes, the output is zero.
    - In this case the baseline factor is updated, because as it has been indicated before this is not constant in time.
    - The loop starts again with the next fragment of signal.

- ⋆ If the value does not exceed the first threshold, it is checked if the value exceeds the last threshold. In this case the user has blinked.
  - The output is zero and the blink flag is activated.
  - The loop starts again with the next fragment of signal.
- ⋆ If the value is between the first and the last threshold, a quantisation of the signal is done as a function of the remainder thresholds. An example of a quantised signal is shown at the bottom of Figure 5(b) with a mark when a blink is detected.
  - The output is one of the next values: -3, -2, -1, 1, 2 or 3, as a function of the gaze angle and the position of the eye.
  - The loop starts again with the next fragment of signal.

The processing is the same for both the horizontal and vertical channels, using the specific thresholds for each one. The outputs are values between -3 and 3 and a flag to indicate the blink.

In addition, an average of the last three instants of the output has been done to avoid possible punctual errors.

3.3.1. Performance of the velocity control

Then, using the output of the processing algorithm, the velocity control is done. In the initial position, this is increased as a function of the output value of the processing algorithm. For example, if the absolute value of the output in the horizontal channel is one, an increment of one in the horizontal position in the direction of the sign is done. If the output is three, the increment will be three. The increments are in pixels for the user graphical interface and in millimetres for the robot arm. The graphical interface used for the experiments is 360×300 pixels and the workspace used by the robot arm is 360×300 mm. The same size has been chosen to make the training with the graphical interface similar to that of the robot arm for the user.

The same procedure is carried out in the vertical channel, and the combination of both forms a trajectory in the function of the gaze direction of the user, with velocity as a function of the gaze angle. If the user blinks the increment is zero and a mark is done.

4. Experimental results

In this section, the robot arm used in tests and the software to control it are explained. Thereafter, different modes of a graphical interface designed for training are shown. Subsequently, the test protocol is described. Finally, the results obtained by training with the graphical interface and the robot arm are described.

4.1. Robot arm

The robot arm FANUC LR Mate 200iB has been used for the final tests. This robot has six degrees of freedom and can load up to 5 kg in the end effector. The robot arm is shown in Figure 6. The software used to control the robot arm has been programmed in C++ and it allows us, through the libraries of the robot, to make a direct connection with it and send commands with the necessary positions for drawing the trajectories. The software has been integrated into Matlab for use in the tests.

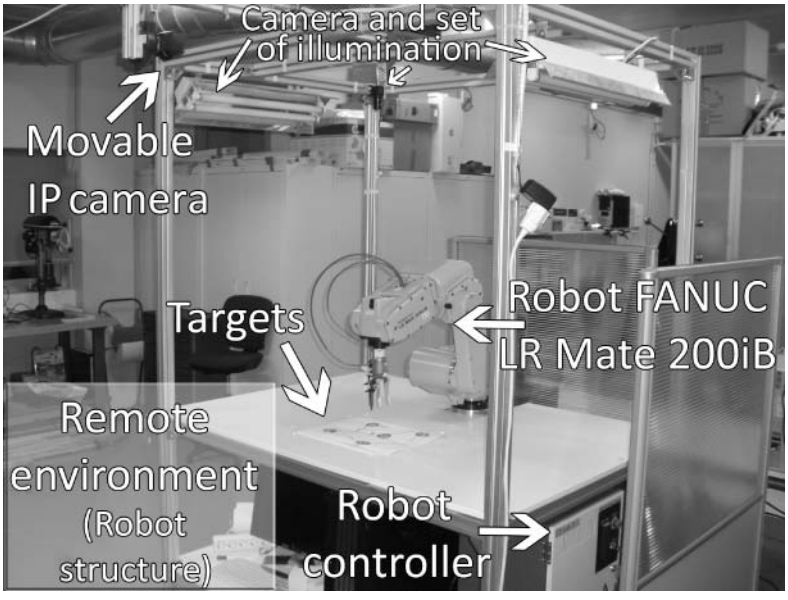


Figure 6. Remote environment. Structure with the robot arm FANUC LR Mate 200iB, the IP camera and targets.

The robot arm is installed in a structure (used in the laboratory for different situations) with a set of illuminators, fixed cameras, and the robot controller (Figure 2). Because the robot is not in the same building where the user is performing the tests, a movable IP camera has been installed allowing us to see it at all times.

Because the detected movements of the eyes can only be up/down, left/right and combinations of them, only two degrees of freedom of the robot have been controlled by the eye movements. To make the movements of the robot arm, Cartesian coordinates have been used, moving the end effector of the robot arm in the XY plane and keeping the plane Z and the orientation constant. The range of the robot arm has been limited to a section in the XY plane for safety reasons. It matches the size of the target paper.

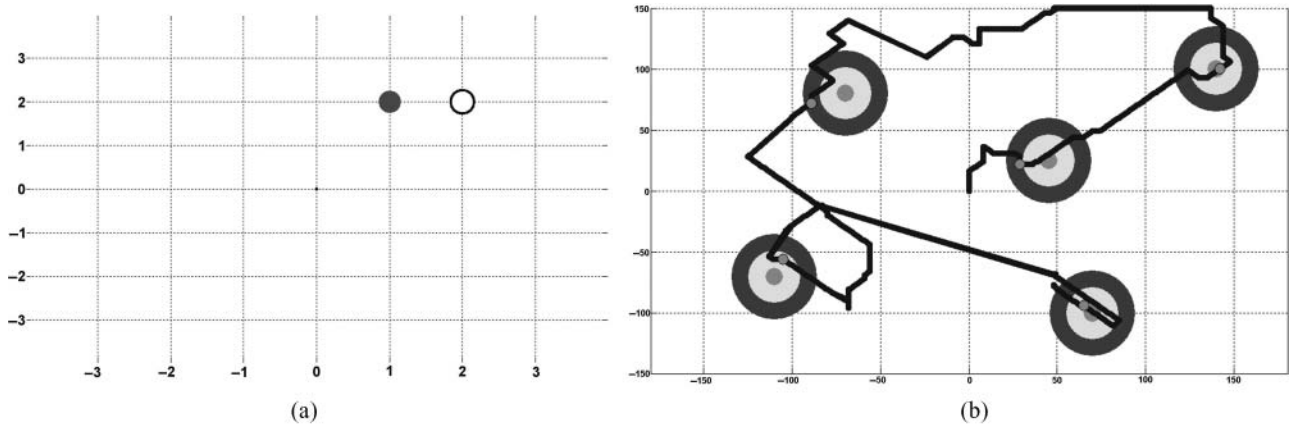
Since the processing algorithm explained in Section 3.3 takes decisions each 0.1 seconds, the control program will send a new command to the robot arm each 0.1 seconds in order to get a new position. Because the robot takes time in performing a movement and because the commands are sent very quickly (every 0.1 seconds), at specific moments, the robot arm cannot follow all the instructions, generating a delay in the trajectory. On this basis, the time has been included in each instruction of movement. Every time the robot reads a new instruction of movement, it checks the time rejecting the command if this time is earlier than a specific value. Although this happens only in concrete moments, it should be taken into account to avoid an accumulative delay in the application.

4.2. Graphical interface

Several graphical interfaces have been developed in Matlab to allow the user to train before using the robot arm (Figure 7).

These interfaces allow connection with the NeuroScan device in order to register the eye movement data in real time. Every 0.1 seconds the interface processes the data obtaining the direction and position where the user is looking. Four kinds of interfaces have been developed and are explained as follows:

- The first interface allows us to calculate the thresholds automatically (Figure 7a). The user must follow a black circle that appears on the screen. In certain instances a blink is requested to the user. While processing the signal, registered thresholds are automatically calculated for use in the rest of the tests.
- In the second interface, the user can freely test the system. A grid appears with values from -3 to 3 in the horizontal and vertical channels. Because the outputs of the processing algorithm are integer values from -3 to 3, only integer values are allowed in this interface. A red dot is shown on the screen in the position where the user is looking to. If the user blinks, the dot turns green.
- The third interface is similar to the second one, but in this case certain random targets are shown on the screen with a black circle. The objective of the user is matching the red dot with the black circle in all moments (Figure 7a). The distance between the red dot and the black circle target is measured during the test to obtain how well the user can control the system. The results of this test are shown in Section 4.4.
- The last interface has been developed to train in the way similar to the robot arm. In this case some targets appear on the screen and the user makes a trajectory moving a dot on the screen. The dot moves in the gaze direction and with a velocity as a function of the degree of the gaze. When the user blinks, a mark is made. The user must go to the targets and mark them. The size of this



(a) Graphical interface to calculate thresholds automatically/ Graphical interface to train the different positions of the eye and the blink.

(b) Graphical interface to train with the velocity before use the robot arm.

Figure 7. Graphical interfaces in Matlab.

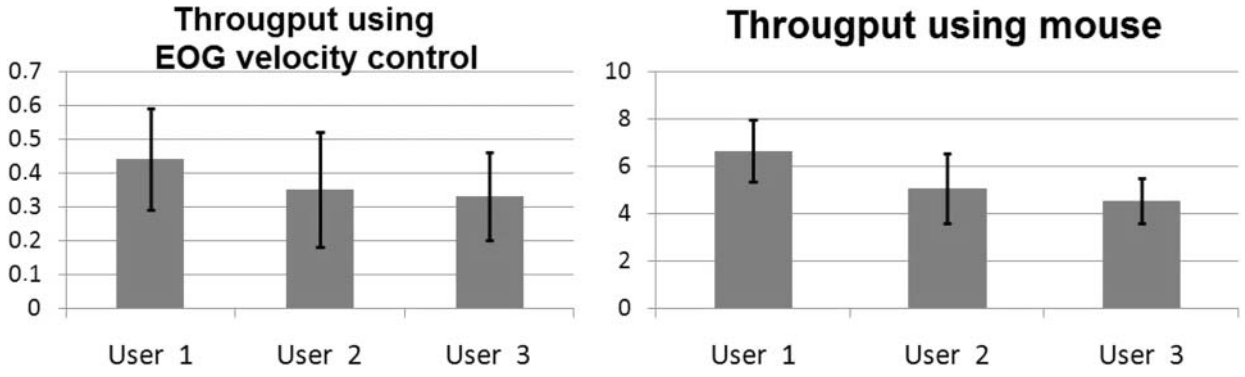


Figure 8. Throughput.

interface is 360 × 300 points. This size has been chosen because the workspace of the robot arm is the same, but in millimetres. An example of this interface with a trajectory is shown in Figure 7(b).

The last interface has also been used to send appropriate command controls to the robot arm when it is activated. However, in this case the user uses the image from the IP camera as visual feedback of the robot arm.

4.3. Tests protocol

If a new user wants to use the system he or she must follow the following phases:

- First, the person must use the application for automatic threshold calculation in order to establish the thresholds that allow detecting the direction of the eyes, the degree of the gaze, and blinks.
- The second phase is the training of the user and three graphical interfaces are used:
  - First, the second interface is used to test thresholds freely.
  - Next, the user uses the third interface where the distances to some random targets are measured.
  - To finish the training, the user trains with velocity control in the fourth interface trying to cross the targets and mark them, performing the movements with a dot on the screen.
- Finally, the user controls the robot arm in order to mark some targets.

4.4. Results

Different experiments have been performed in order to test the EOG-based control for the robot arm.

Tables 1 and 2 show the results using the third interface for the three users. In each test 10 random targets are shown to the user and the Euclidean distance and the success percentage of these targets are measured. In Table 1, the mean and standard deviation of the Euclidean distance have been measured independently for both the horizontal and vertical channels. The global column indicates the Euclidean distance taking into account both channels simultaneously. The third interface takes into account only the integer values from −3 to 3 for both channels, which are the possible outputs for the processing algorithm. In Table 2, the mean and standard deviation of the success percentage have been measured independently for both the horizontal and vertical channels. The global column indicates the success percentage taking into account both channels simultaneously. The results of the vertical channel are higher than the horizontal channel. The Euclidean distance is larger and the success percentage decreases. This is because of the lower range of the vertical channel, causing a certain amount of instability when the user has the eye in a specific position.

The fourth interface has been used to train the user. A comparison between the throughput using the mouse and the EOG velocity control has been done. (Zhang and MacKenzie 2006). In this paper, the throughput is calculated taking into account the distance to the target, time of arrival and error produced when marked. First, the user

Table 1. Results of the training using the third graphical interface in Matlab. The mean ± standard deviation of the global, horizontal and vertical Euclidean distance of the three users.

	Global	Horizontal	Vertical
User 1	0.23 ± 0.18	0.26 ± 0.22	0.35 ± 0.29
User 2	0.21 ± 0.22	0.09 ± 0.17	0.32 ± 0.31
User 3	0.28 ± 0.20	0.21 ± 0.23	0.35 ± 0.35

Table 2. Results of the training using the third graphical interface in Matlab. Mean ± standard deviation of the global, horizontal and vertical success percentage of the three users.

	Global	Horizontal	Vertical
User 1	88.6 ± 13.1%	95.6 ± 3.0%	90.3 ± 12.8%
User 2	85.0 ± 17.5%	98.1 ± 3.7%	85.6 ± 16.7%
User 3	83.0 ± 15.1%	95.0 ± 3.9%	85.8 ± 15.3%

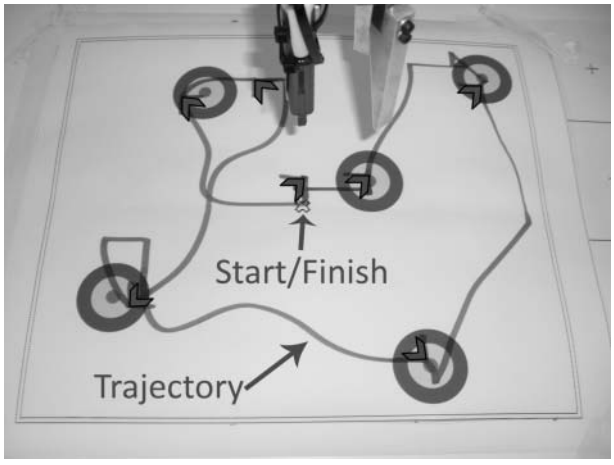


Figure 9. Zoom of a trajectory made while marking the targets and the end effector of the robot arm.

performs the test using the mouse. The user must move the pointer of the mouse from the centre of the screen to a random target marking nearest of the centre of the target. The throughput and the standard deviation calculated are shown in Figure 8 (right). Thereafter, the user performs the same test using the EOG velocity control. The throughput using EOG is shown in Figure 8 (left). Throughputs are very different because, first, the EOG velocity control has a limited velocity. The user can only move 30 pixels with the maximum velocity while the mouse does not have this limit. If this limit is eliminated, the results should be more similar. Second, the users are people who usually use mice in their daily life.

Finally, after completion of the training, the user can use the robot arm to perform a trajectory in order to cross some targets and mark them blinking. Figure 9 shows a zoom of the piece of paper with the targets. The user is looking at this image when he or she is controlling the robot arm. The trajectory is shown and when the user blinks, an 'arrow' is painted, as shown in the figure. The results have been successful. The users were able to draw a trajectory and mark some targets while controlling a robot arm with the eyes.

5. Conclusions

In this paper, EOG-based control of a robot arm has been presented. Successful results have been obtained controlling the movement of the robot arm with different velocities as a function of the degree and the direction of the position of the eye.

The users learn how to control gaze direction and the position of the eye with little training. The users have managed to create a trajectory controlling the robot arm and marking certain targets.

This technology could eventually allow people with severe disabilities to control robots only with the movements of the eyes, which can help them in activities of daily life.

Our future work is focused on detecting with more accuracy the degree of the movement of the eyes to perform an analogue control of the velocity. In this way, the robot arm could be moved faster according to deviation in the degree of eyes from the centre position. In addition, the protocol control of the robot arm will be extended in order to perform more complex tasks.

Acknowledgements

This research has been supported by grants DPI2008-06875-C03-03 (Ministerio de Ciencia e Innovación) and SAF2008-03694 from the Spanish Government.

References

American EEG Society. 1991. American Electroencephalographic Society guidelines for standard electrode position nomenclature. J Clin Neurophys. 8(2):200–202.

Barea R, Boquete L, Bergasa LM, Lpez E, Mazo M. 2003. Electrooculographic guidance of a wheelchair using eye movements codification. I J Robot Res. 22(7–8):641–652.

Barea R, Boquete L, Mazo M, López E. 2002. Wheelchair guidance strategies using EOG. J Intell Robot Syst. 34(3):279–299.

Chen Y, Newman WS. 2004. A human–robot interface based on electrooculography. In Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, New Orleans, LA, USA, pp. 243–248. IEEE.

Duguleana M, Mogan G. 2010. Using eye blinking for EOG-based robot control. Emerging Trends in Technical Innovation, IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg. 314:343–350.

Ho CK, Sasaki M. 2001. Mobile robot control by neural networks EOG gesture recognition. In the 8th International Conference on Neural Information Processing, vols. 1–3, pp. 169–175. Shanghai: Fudan University Press.

Hutchinson TE, White KP, Martin WN, Reichert KC, Frey LA. 1989. Human–computer interaction using eye-gaze input. IEEE Trans Syst, Man Cybern. 19(2):1527–1534.

Iáñez E, Azorín JM, Fernández E, Morales R. 2008. Electrooculography-based human interface for robot controlling. In Proceedings of the 13th Annual Conference of the International Functional Electrical Stimulation Society (IFESS), vol. 53, pp. 305–307. New York: Walter de Gruyter.

Iáñez E, Azorín JM, Furió MC, Fernández E, Sabater JM. 2009. EOG and EEG control of robot arms. In IEEE International Conference on Robotics and Automation (ICRA 2009). Workshop on Interfacing the Human and the Robot (IHR). Kobe, Japan, May 12–17.

Iturrate I, Antelis JM, Kübler A, Minguez J. 2009. A noninvasive brain-actuated wheelchair based on aP300 neurophysiological protocol and automated navigation. Trans Rob. 25(3):614–627.

Jeong Y, Lee D, Kim K, Park J. 2000. A wearable robotic arm with high force-reflection capability. In IEEE International Workshop on Robot and Human Interactive Communication, Osaka, Japan.

Millán JR, Ferrez PW, Buttfield A. 2005. Non-invasive brain–machine interfaces – Final report. IDIAP Research Institute, ESA.

Millán JR, Renkensb F, Mouriñoc J, Gerstnerb W. 2004. Brain-actuated interaction. Artif Intell. 159:241–259.

Nicolau MC, Burcet J, Rial RV. 1995. Manual de técnicas en electrofisiología clínica. Universidad de las Islas Baleares, Palma de Mallorca, España, pp. 215.

Oyekoya OK, Stentiford FWM. 2006. Eye tracking: a new interface for visual exploration. BT Technol J. 24(3):57–66.

Sankai Y. 2006. Leading edge of cybernics: robot suit HAL. In SICE-ICASE International Joint Conference, Korea, pp. 1–2.

Sibert LE, Jacob RJ. 2000. Evaluation of eye gaze interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands). CHI'00. ACM, pp. 281–288.

Úbeda A, Azorín JM, Iáñez E, Sabater JM, 2009. Eye-tracking interface based on artificial vision for robot. In Proceedings of The 13th IASTED International Conference on Artificial Intelligence and Soft Computing. September 7–9, Palma de Mallorca, Spain.

Úbeda A, Iáñez E, Pérez C, Azorín JM. 2010. Haptic and ocular human–robot interface. In Proceedings of the IEEE International Conference on Robotics and Automation, Workshop on Multimodal Human–Robot Interfaces, pp. 23–27.

Usakli AB, Gurkan S, Aloise F, Vecchiato G, Babiloni F. In press. On the use of electrooculogram for efficient human computer interfaces. Comput Intell Neurosci.

Zhang X, MacKenzie IS 2007. Evaluating eye tracking with ISO 9241 – Part 9. In Proceedings of HCI International. Springer, Heidelberg, pp. 779–788.