

# Influencia de la Interacción Humana Vía Interfaz Cerebro Computadora (BCI) en el Control de un Robot

Román Pérez Marco Antonio<sup>1</sup>, Montañó Martínez Neyli-Han y Sepúlveda Cervantes Gabriel

Centro de Innovación y Desarrollo Tecnológico en Cómputo - Instituto Politécnico Nacional  
Av. Juan de Dios Bátiz s/n casi esq. Miguel Othón de Mendizábal edificio del CIDETEC. Colonia Nueva Industrial Vallejo, Delegación Gustavo A. Madero C.P. 07700, México

## Resumen

*En la actualidad debido a la necesidad cada vez mayor de calidad y rapidez en los sistemas de producción ha aumentado el uso de robots en la industria, por lo cual ha sido necesario desarrollar técnicas para controlar estos dispositivos.*

*Existen leyes de control que han sido propuestas, evaluadas y aplicadas para prácticamente cualquier actividad; sin embargo el hecho de que el usuario intervenga en la toma de decisiones de un robot no está del todo estudiado.*

*El propósito de un sistema de control es mantener la posición o movimiento del robot a lo largo de una trayectoria. En el presente artículo se pretende mostrar la influencia así como los resultados obtenidos al incluir la voluntad del usuario dentro de la ley de control.*

*El experimento consiste en programar a un robot para que realice una trayectoria definida, sin embargo el controlador se ve afectado por el grado de concentración del usuario, éste último obtenido mediante un dispositivo para lectura de ondas electroencefalográficas, el cual es una diadema que cuenta con 14 electrodos con los cuales se puede leer la actividad cerebral del usuario, de esta forma el desempeño del controlador depende también del usuario.*

**Palabras clave:** Interfaz Cerebro Computadora, Control, Robótica.

## 1. Introducción

Uno de los ámbitos de estudio más recientes es la investigación de interfaces cerebro computadora o BCI (Interfaz Cerebro Computadora por sus siglas en inglés). Las cuales son definidas como un medio de comunicación entre el cerebro y un dispositivo

externo, basándose en las ondas cerebrales emitidas por el mismo, las cuales son leídas empleando electrodos.

Una de las aplicaciones más estudiadas para este tipo de interfaces se encuentra en el área médica, debido a que pueden ser empleadas para devolverle la movilidad a una extremidad.

Es importante mencionar que el desempeño de una BCI depende en gran manera del grado de concentración del usuario, esto permite estudiar la forma en que el usuario puede intervenir en el desempeño de alguna actividad automatizada.

En nuestro caso de estudio pretendemos mostrar la forma en que la actividad cerebral del usuario afecta el desempeño de una actividad, en este caso el control de robots.

Como mencionamos anteriormente se han desarrollado diversas leyes de control para actividades específicas, sin embargo dichas leyes no contemplan al usuario, es por ello que en el presente trabajo se propone incluir al usuario como parte de la ley de control y mostrar la forma en que éste afecta los resultados.

## 2. Interfaz Cerebro Computadora

La principal clasificación, de este tipo de interfaces atiende a la forma en que los electrodos son colocados [1]. Y se denominan:

*Invasivas:* cuando los electrodos son implantados directamente en el cerebro del usuario.

*No Invasivas:* cuando los electrodos son colocados sobre el cuero cabelludo del usuario.

Las etapas por las que pasa una BCI son la etapa de adquisición de las señales, para la cual se trabaja con el dispositivo EPOC de la empresa Emotiv y la etapa de procesamiento de dichas

<sup>1</sup> E-mail: m.antonio.roman@gmail.com

señales, para la cual se trabaja en un PC utilizando las librerías de desarrollo del dispositivo.

La diadema EPOC cuenta con 14 electrodos de registro o activos y 2 electrodos de referencia, esto permite tomar la señal de cada uno de los electrodos independientemente de la señal de los demás. Además de que la comunicación con la PC es inalámbrica, lo que permite mayor movilidad al usuario. En la siguiente figura se pueden observar los elementos que componen a la diadema EPOC [2].



Fig. 1. Dispositivo EPOC de la empresa Emotiv.

El dispositivo EPOC cuenta con una interfaz gráfica, misma que se puede observar en la Fig. 2, con la cual podemos el utilizar la diadema para desplazar un cubo, sin embargo, para llevar a cabo algún desplazamiento con el cubo, es necesario que previamente se haya entrenado el desplazamiento, para que se pueda reconocer cual es el patrón que corresponde a determinada actividad.



Fig. 2. Interfaz del software Emotiv Control Panel.

El entrenamiento que se realiza con ayuda de esta interfaz, permite que las librerías de desarrollo puedan reconocer e interpretar dichas actividades para el desarrollo de otras aplicaciones.

### 3. Control del robot

Para el trabajo aquí desarrollado utilizamos el robot Falcon de la compañía Novint, el cual fue diseñado como un control para videojuegos pero puede ser controlado como un robot paralelo con tres grados de libertad; se comunica con la computadora vía puerto USB y cada grado de libertad se encuentra actuado por un motor de corriente directa y un encoder acoplado [3]. En la figura 3 podemos observar una imagen del robot.



Fig. 3. Robot Falcon

Con estas características es posible controlar el robot ya que podemos leer el valor de los encoders así como enviar fuerza a los motores desde la pc. Para la tarea que desempeña el robot utilizamos un control proporcional derivativo definido mediante la siguiente ecuación [4]:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (1)$$

Como podemos observar esta ley de control depende de dos constantes  $K_p$  y  $K_d$  las cuales son las ganancias proporcional y derivativa respectivamente. Estas ganancias afectan la forma en que se comporta el controlador al detectar un error de posición ( $e(t)$ ) o de velocidad ( $de(t)/dt$ ) del robot, de forma que mientras mayor sea la ganancia proporcional mayor será la fuerza con que responderá el robot para tratar de llegar a la posición deseada [5].

Con esta ley de control mandamos al robot a realizar círculos en su plano de operación  $xy$  usando como posiciones deseadas las siguientes ecuaciones:

$$x_d = r * \text{sen}(2 * \pi * f * t) \quad (2)$$

$$y_d = r * \text{cos}(2 * \pi * f * t) \quad (3)$$

Estas ecuaciones corresponden a las ecuaciones paramétricas del círculo donde  $r$  corresponde al radio,  $f$  a la frecuencia con que realiza el círculo y  $t$  es el parámetro del que dependen, en este caso tiempo, de acuerdo con las ecuaciones cada vez que el programa recalcula los valores de  $x_d$  y  $y_d$  el robot se moverá tratando de llegar a ellas, de manera que dibujará círculos.

#### 4. Integración del Control del Robot y la BCI

Como mencionamos anteriormente, las BCIs constituyen un nuevo medio de comunicación entre el usuario y una computadora, debido a esto podemos desarrollar aplicaciones que permitan interactuar con prácticamente cualquier dispositivo empleando únicamente su actividad cerebral, es decir, únicamente con pensar el usuario puede desarrollar actividades.

Para realizar lo anterior es necesario que el usuario tenga control sobre la actividad a realizar y esto se logra haciendo que el usuario envíe señales de control hacia el dispositivo a controlar al mismo tiempo que recibe una retroalimentación que en este caso es visual para cerrar el lazo de control.

Para utilizar la diadema EPOC es necesario llevar a cabo una serie de entrenamientos, generalmente de 30 a 60 minutos diarios dependiendo del usuario y el número de días depende de cuantas acciones desea controlar [6]. El entrenamiento consiste en lograr que el usuario relacione un pensamiento o serie de pensamientos a una acción que el software muestre.

La interfaz mostrada en la figura 2 permite al usuario entrenar hasta 13 acciones distintas tales como mover hacia: arriba, abajo, izquierda, derecha, enfrente, atrás, rotar, entre otras.

De esta manera el usuario debe entrenar cada acción que desea utilizar asociando a ella un pensamiento, por ejemplo si queremos mover un objeto a la derecha podemos pensar que movemos el brazo derecho, es importante mencionar que cuando se mueve una extremidad la actividad cerebral es muy similar a cuando simplemente pensamos en hacerlo.

El software de la diadema EPOC genera patrones dependiendo de las señales que detectan los electrodos al entrenar una acción, y genera un patrón distinto para cada acción que se entrene, de esta forma cuando el usuario piense en mover su brazo

derecho (siguiendo con el ejemplo anterior) el software compara la actividad leída por los electrodos con los patrones que tiene guardados y al detectar similitudes con alguno de ellos ejecuta la acción a la que esté relacionada.

Para la aplicación desarrollada en este trabajo utilizamos solo dos señales de control: mover hacia adelante y hacia atrás, las cuales afectan el valor de la ganancia  $Kp$  mostrada en la ecuación (1), de forma que cuando el usuario piensa en mover hacia adelante la ganancia aumenta de forma gradual, y cuando el usuario piensa en mover hacia atrás la ganancia disminuye.

Lo cual se logra integrando las librerías de desarrollo tanto del robot Falcon como del dispositivo EPOC en un solo proyecto de trabajo, utilizando el software Visual Studio 2008 y como lenguaje de programación Visual C++,

#### 5. Pruebas y Resultados Obtenidos

Las pruebas desarrolladas consistieron en que el usuario después de haber realizado al menos 5 sesiones de entrenamiento tratara de manipular la ganancia del controlador y con esto la forma en que se comportaba el robot al tratar de seguir la trayectoria establecida.

En la figura 4 se muestra la interfaz completa donde podemos ver al usuario utilizando la diadema EPOC, la computadora que controla el sistema y el robot Falcon que es controlado, la diadema se comunica a la computadora vía inalámbrica.

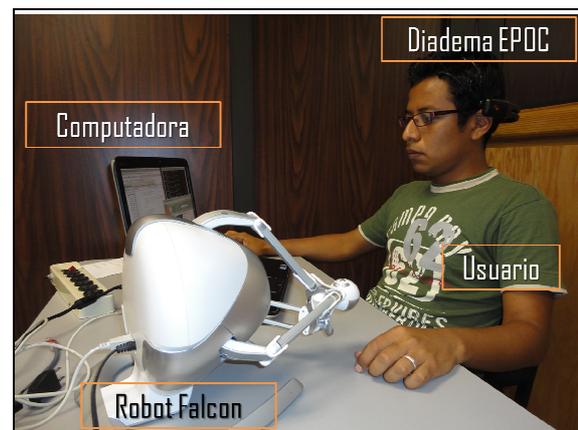
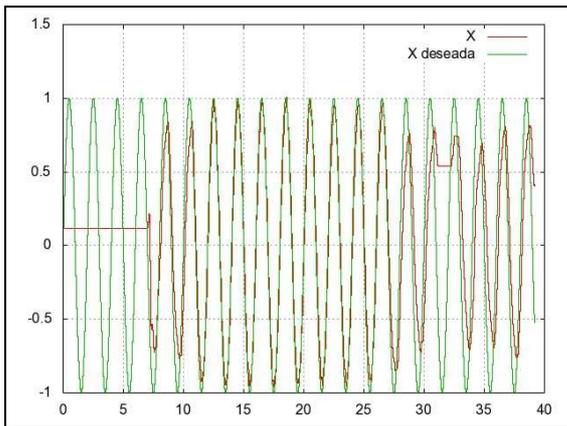


Fig. 4. Interfaz EPOC – Falcon

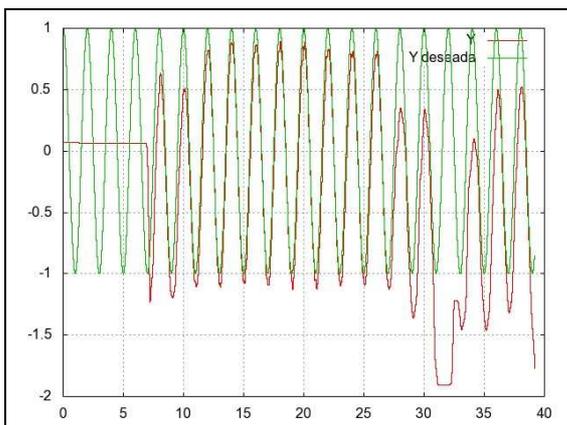
En la figura 5 podemos observar una gráfica que muestra la forma en que se movió el robot con respecto a la forma en que se le pedía que lo hiciera con respecto al eje x, de color verde observamos el valor de la X deseada y de color rojo el valor real de la coordenada en x del robot.



**Fig. 5. Error de posición en X**

En la parte inferior de la gráfica observamos los valores correspondientes al tiempo de la prueba en segundos y de lado izquierdo el desplazamiento del robot de acuerdo a la lectura de sus encoders.

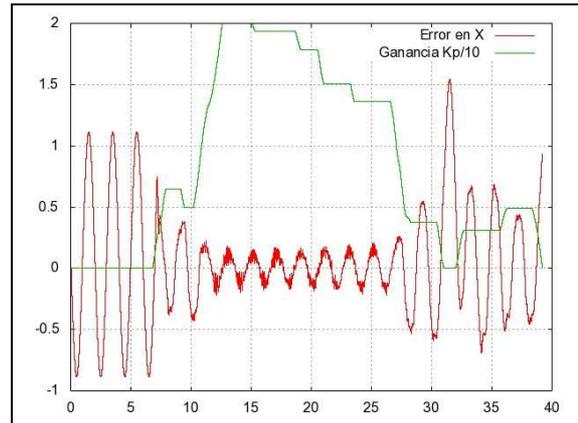
En la figura 6 podemos observar el movimiento del robot en el eje y con respecto a la posición deseada en ese eje. De la misma forma en color verde la posición deseada y de rojo la posición real del experimento



**Fig. 6. Error de posición en Y**

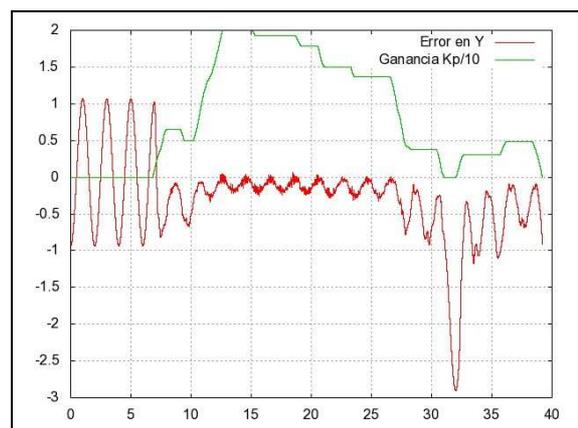
Como se mencionó anteriormente cuando el usuario piensa en mover el objeto hacia adelante

aumenta la ganancia Kp, esto lo podemos observar en la gráfica de la figura 7, donde se muestra como la ganancia (línea color verde) inicia en cero y cuando el usuario piensa en mover hacia adelante el objeto la ganancia empieza a aumentar a partir del segundo 6 aproximadamente, al mismo tiempo vemos como el error (línea roja) tiende a disminuir cuando la ganancia aumenta.



**Fig. 7. Error en X y ganancia del controlador**

En la figura 8 observamos el mismo comportamiento pero con respecto al eje y. Podemos observar como el error tiende a disminuir cuando el usuario afecta la ganancia y como a partir del segundo 30 aproximadamente el error aumenta debido a que la ganancia del controlador tiende a cero.



**Fig. 8. Error en Y y ganancia del controlador**

En la figura 9 podemos observar de color verde el círculo que deseamos que el robot dibuje y

de color rojo vemos las posiciones del robot al tratar de seguir la trayectoria deseada, los primeros círculos que realiza son muy alejados del deseado debido a que la ganancia es muy pequeña y a medida que el usuario aumenta la ganancia, el sistema tiende a mejorar sus resultados.

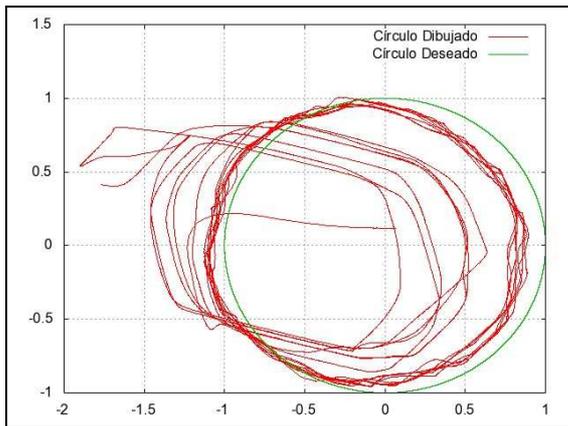


Fig. 9. Trayectoria real y deseada

Para obtener las gráficas aquí mostradas se creó un archivo de texto que guardara las posiciones reales del robot así como las posiciones deseadas, el valor de la ganancia y la variable tiempo para graficarlas. Una vez que se generó el archivo de texto fue posible realizar las gráficas utilizando el software libre GNUPlot.

## 6. Conclusiones

Como podemos observar en las gráficas cuando el usuario interviene y se convierte en un factor que influye en el control del robot, el sistema puede trabajar de mejor forma, esto nos da la posibilidad de desarrollar aplicaciones más complejas con este tipo de interfaces.

Es importante mencionar que para un mejor control de la aplicación fue necesario llevar a cabo más sesiones de entrenamiento. Desde la primera sesión es posible que el usuario controle una acción e incluso dos acciones, sin embargo el desempeño al

tratar de controlarlo es mucho menor que el de un usuario que ha realizado varias sesiones de entrenamiento.

En el trabajo realizado se observa que el estrés, así como la tensión en el usuario provoca que sus respuestas al tratar de controlar el robot sean poco precisas provocando que el desempeño del sistema disminuya.

Cabe mencionar que dentro de la literatura actual en cuanto a control de robots no existen sistemas en los cuales el usuario interactúe de forma directa con la ley de control, en muchos casos el usuario solo decide la acción a ejecutar y el sistema ejecuta rutinas pre programadas.

## Referencias

- [1] Minguez J. "Tecnología de Interfaz Cerebro – Computador". Departamento de Informática e Ingeniería de Sistemas. Universidad de Zaragoza, España.
- [2] Emotiv Software Development Kit User Manual for Release 1.0.0.4.
- [3] Novint Falcon™ User Manual.
- [4] Ogata K. "Modern Control Engineering", Ed. Pearson, España, 4ta. Edición, 2003.
- [5] Román M., Sepúlveda G., Montañó N., Gamboa J., "Entrenador de escritura para niños utilizando el robot Falcon Novint empleando un control PD", 10º Congreso Nacional de Mecatrónica, MECAMEX, pp. 3, Puebla, Puebla, 2010.
- [6] Ron R., García C., Díaz A., "Desarrollo de una interfaz cerebro computadora". Departamento de Tecnología Electrónica, Universidad de Málaga, España.