

Entrenador mioeléctrico para dispositivos protéticos de mano

Juan Sebastián Cabrera Hidalgo. Ingeniero de Desarrollo Hardware en Investigación y Tecnología S.A. Magister en Ingeniería de Confiabilidad y Riesgo de la Universidad de Las Palmas de Gran Canaria. Ingeniero Físico de la Universidad del Cauca. Candidato a Magister en Ingeniería, área: Electrónica y Telecomunicaciones de la Universidad del Cauca.

Héctor Fabio Jaramillo Ordoñez Docente Universitario en la Universidad del Cauca. Magister en ingeniería, área: Telemática de la Universidad del Cauca. Ingeniero en Electrónica y Telecomunicaciones de la Universidad del Cauca.

Luis Eduardo Rodríguez Cheu. Docente Universitario en la Escuela Colombiana de Ingeniería. Doctor de la Universidad Politécnica de Cataluña. Magíster en Ingeniería Electrónica de la Universidad de los Andes. Ingeniero Eléctrico de la Escuela Colombiana de Ingeniería.

Resumen

El presente trabajo describe la construcción de un entrenador mioeléctrico a partir de un conjunto normalizado de interfaces, es decir, desarrollar el entrenador implementando las interfaces previamente establecidas en una arquitectura global hardware-software para dispositivos prostéticos de mano.

Se tiene como objetivos: facilitar la adaptación de una prótesis mioeléctrica para amputados de mano a través de un entrenamiento previo al manejo de la prótesis y que a su vez, dicho entrenador sea un desarrollo modular que permita la reutilización e integración de distintos resultados de investigaciones relacionados con dispositivos de este tipo.

Por sugerencias de profesionales del área de fisioterapia del Centro Integral de Rehabilitación de Colombia, CIREC, el entrenador se conforma de una mano virtual de

apariciencia humana y un video juego los cuales son controlables a partir de las señales mioeléctricas generadas por el paciente.

El sistema se ha probado en cinco personas sin amputación, los cuales han realizado un entrenamiento exitoso consiguiendo controlar el video juego y la apertura y cierre de la mano.

Palabras Claves

Entrenador de prótesis, Adecuación de prótesis, Reuso, Encapsulamiento, Interfaz normalizada, Arquitectura de prótesis.

1 Introducción

Las dificultades que enfrenta una persona amputada son desoladoras, desde su propia autoestima al sentirse "menos-válido", hasta la consecución de un trabajo desde el cual pueda sostenerse a sí mismo y a su familia; en Colombia debido a los conflictos armados muchas personas, tanto civiles y militares, son víctimas de amputaciones que reducen su autonomía en la sociedad, dificultando aún más las condiciones de vida en nuestro país.

En el mundo, muchos centros de investigación y empresas privadas realizan estudios y desarrollos para brindar opciones de una mejor vida a las personas amputadas a través del uso de prótesis robóticas. Entre los dispositivos prostéticos al que mayor atención se le ha prestado es la prótesis de mano, ya que nuestras manos son nuestra principal herramienta de acción creativa; sin embargo, las soluciones que existen suelen

ser muy costosas para nuestra sociedad, y gran parte de este costo se debe a la falta de reutilización de los avances alcanzados por otros en el área; es así como cada solución a pesar de tener una arquitectura de base similar su implementación es propietaria y suele mantener un alto acoplamiento entre los módulos de la arquitectura utilizada.

Por lo tanto, dada la baja reutilización y el alto acoplamiento en los desarrollos relacionados con dispositivos prostéticos de mano, en (Cabrera, 2009) se propone un conjunto normalizado de interfaces para los componentes que integran a los dispositivos prostéticos de mano con el fin de permitir un desarrollo sostenible y comprensible del mismo, de tal forma que si se hace necesario cambiar total o parcialmente la estrategia de control en alguno de los componentes se realice teniendo en cuenta las interfaces previamente establecidas, sin causar inconvenientes al funcionamiento de los demás componentes.

Lo anterior facilitaría la integración y depuración de los componentes hardware/software que conforman el entrenador de prótesis de mano y permitiría llevar de una manera metódica un sistema de control de versiones, el cual registre los cambios realizados con miras a la puesta a punto, mantenimiento y evolución del sistema.

Para el desarrollo del presente proyecto se tomó como referencia el modelo para la construcción de soluciones expuesto en (Serrano, 2005). De acuerdo al modelo se realizaron las siguientes actividades: Especificaciones, Diseño, Implementación y Pruebas del Entrenador Mioeléctrico. Dichas actividades serán descritas a continuación.

2 Especificaciones del Entrenador Mioeléctrico

De acuerdo con (Alonso, Sánchez, Espino, Liptak, & De la Rosa, 2002) “una prótesis mioeléctrica es un sistema accionado por servomotores que se gobierna a partir de señales EMG recogidas en el muñón del paciente, para lo cual es preciso un reconocimiento en tiempo real de las características de dichas señales”. En esta definición se introduce el concepto de tiempo real, el cual sugiere que la respuesta del dispositivo protésico no solo debe ser correcta, sino que debe generarse en un tiempo especificado.

En (Farrell & Weir, 2007) se sugiere que un retardo óptimo para la respuesta del dispositivo protésico se encuentra entre 100ms para agarres rápidos y 125ms para los lentos. Por lo tanto, para garantizar el cumplimiento de este requisito temporal se tuvieron en cuenta los conceptos de tiempo real los cuales fueron aplicados al dispositivo protésico a través del uso de un RTOS, de sus siglas en inglés, Sistema Operativo de Tiempo Real.

Por otro lado, se especificó que el entrenador debiese contar con una mano de apariencia humana con el fin disminuir el nivel de impacto al paciente. En entrenadores como los de fabricados por (Otto Bock HealthCare GmbH , 2010) se presentan manos de apariencia ósea o robotizada, las cuales, de acuerdo con profesionales de fisioterapia del CIREC podrían indisponen psicológicamente al paciente.

De acuerdo con (Dupont & Morin, 1994) el uso del entrenador virtual para prótesis de mano se asemeja al control de un video juego, el cual lo convierte en un sistema apropiado para niños amputados ya que reduce la sensación de rechazo inicial que supone

la colocación de un elemento extraño, en este caso la prótesis, sobre el brazo de los pequeños. Este estudio reafirmó la sugerencia realizada por el CIREC de la importancia de agregar un video juego al sistema de entrenamiento. De igual manera, el video juego pretende estimular e incrementar la habilidad en la generación de la señal ME, previamente al control del dispositivo protésico real.

Para el control ya sea del video juego o de la mano virtual se tomó como referencia al dispositivo protésico PROTECI el cual se describe en (Rodríguez, 2006). La estrategia de control se basa en la detección de la duración de la actividad muscular a través de la señal ME. De acuerdo a dicha duración se reconoce la intención de movimiento a ejecutar, por ejemplo, abrir o cerrar la mano. Con esta información, el módulo controlador del dispositivo fija las trayectorias para la ejecución del movimiento y a través de un conjunto de sensores se realimenta tanto de la posición de cada articulación como de la fuerza que le imprime el dispositivo protésico al objeto.

Considerando la estrategia de control expuesta anteriormente y el conjunto normalizado de interfaces para la construcción de dispositivos protésicos de mano sugerido en (Cabrera, 2009), ver Figura 3.1, a continuación se definirán las responsabilidades de los módulos que integran al Entrenador Mioeléctrico.

3 Diseño del Entrenador Mioeléctrico

El conjunto normalizado de interfaces presentado en la Figura 3.1 se integra de siete módulos, los cuales, a su vez se conforman de un amplio grupo de responsabilidades que responden a múltiples estrategias de control para dispositivos protésicos de mano.

Los módulos que conforman el conjunto normalizado de interfaces son:

- Módulo Gestor Analógico (MGA).
- Módulo Gestor Digital (MGD).
- Módulo Caracterizador de Señal (MCS).
- Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)
- Módulo Controlador del Dispositivo (MCD)
- Módulo Gestor Sensorial (MGS)
- Módulo Gestor de Realimentación (MGR)

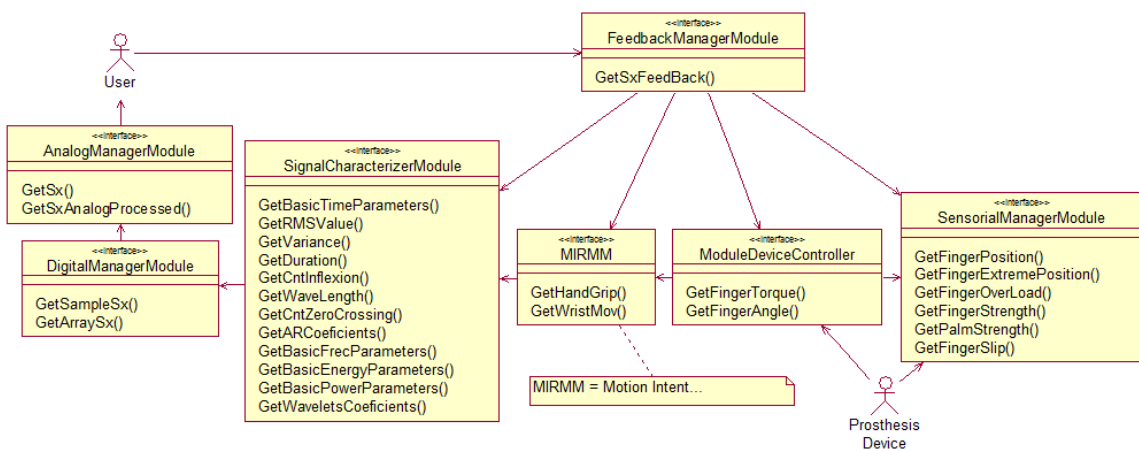


Figura 3.1 Diagrama UML del conjunto normalizado de interfaces para los componentes de un dispositivo próstético de mano.

Ahora bien, partiendo del modelo normalizado expuesto en la Figura 3.1 se selecciona por cada uno de los módulos las responsabilidades que aportan directamente a la implementación de la estrategia establecida para el presente Entrenador Mioeléctrico.

Inicialmente se consideran los módulos gestor analógico (MGA) y gestor digital (MGD), de los cual se implementarán respectivamente las responsabilidades de adquirir la señal analógica pre-procesada y adquirir un arreglo de muestras digitalizadas de dicha señal.

Del módulo caracterizador de la señal (MCS) se implementará la responsabilidad *GetDuration*, en español obtener duración, con el fin de detectar el tiempo de la actividad muscular; a partir de este valor el módulo de reconocimiento de intención de movimiento (MGRIM) se encarga de identificar el tipo de movimiento a ejecutar. Conociendo el movimiento deseado por el paciente, el módulo de control del dispositivo (MCD) se encarga de escoger el conjunto de trayectorias relacionadas con dicho movimiento y ejecutarlas ya sea fijando los ángulos o el valor de torque correspondiente a cada uno de los motores que controlan a las articulaciones del dispositivo protésico.

El módulo gestor sensorial (MGS) tendrá la responsabilidad de conocer en cada instante la posición de cada una de las articulaciones del dispositivo protésico virtual, por lo tanto, tendrá a disposición la función *GetFingerPosition* o en español obtener la posición de cada dedo. Dicha función será accedida por el Módulo Controlador del Dispositivo (MCD) con el fin de realizar las correcciones pertinentes. Por lo tanto, para lograr este objetivo, se requiere que el dispositivo protésico virtual informe de la posición angular en que se encuentra cada una de las articulaciones que conforman al dispositivo protésico.

Por último, el módulo de realimentación (MGR) debe generar una señal sonora con el fin de realimentar al usuario acerca de la terminación de la ejecución del movimiento por parte del dispositivo protésico. Esta función se implementará a través de la función *GetSoundLevel* o en español obtener nivel de sonido.

A partir de la anterior descripción y del conjunto normalizado de interfaces se definió la arquitectura presentada en la Figura 3.2

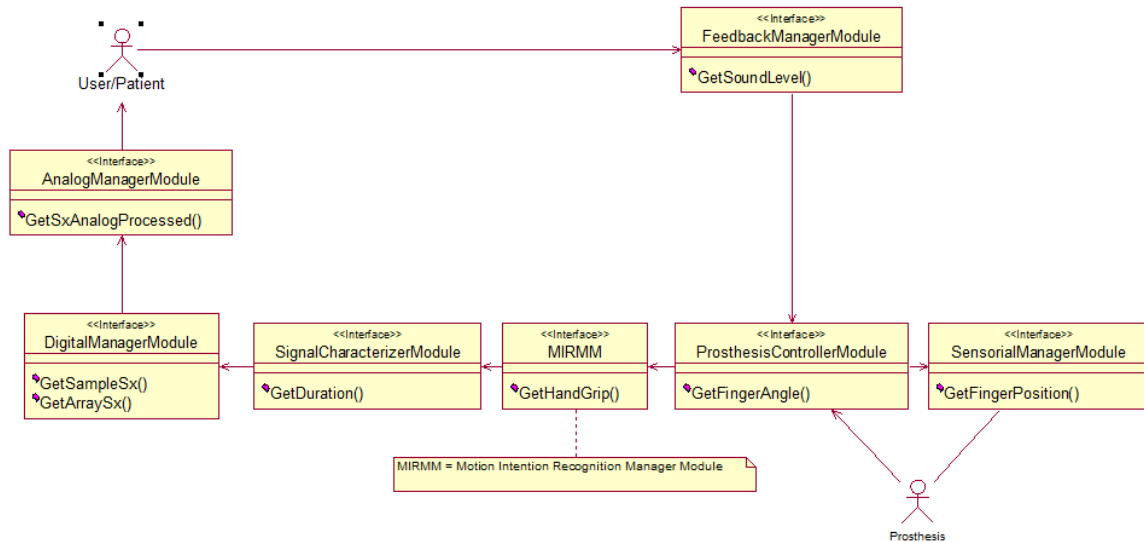


Figura 3.2 Arquitectura del Entrenador Virtual

4 Implementación del Entrenador Mieléctrico

De acuerdo a la arquitectura presentada en la Figura 3.2 se describirán las características más relevantes de cada uno de los módulos que la conforma. Sin embargo, para contextualizar la implementación de la arquitectura propuesta se prefiere inicialmente detallar las herramientas hardware-software utilizadas en el desarrollo del Entrenador Mieléctrico.

Dado que el Entrenador es un paso previo a la utilización de una prótesis real se decidió realizar la implementación de la arquitectura directamente en un microcontrolador; de esta manera, el presente desarrollo servirá como herramienta inicial para la creación de un sistema prostético real.

Se elige utilizar el microcontrolador MCF51CN128 de *Freescale* por su bajo costo y sus prestaciones para la aritmética de punto fijo en formato Q16, el cual, de acuerdo con (Choi, 2010) es un formato usado en el procesamiento digital de señales sobre sistemas embebidos. Dicho microcontrolador de 32 bits cuenta a su vez con módulo de conversión analógica digital de 12 bits utilizado para digitalizar la señal ME y un módulo de comunicación serie asíncrona por el que se transmite las señales de control al dispositivo prostético virtual.

Como lenguaje de programación se elige el lenguaje C, gracias a que es un lenguaje utilizado en la implementación de sistemas operativos, es apreciado por la eficiencia del código que produce y es altamente difundido entre los programadores de sistemas embebidos.

Con el fin de garantizar el tiempo de respuesta requerido y facilitar la administración de las tareas que integran al dispositivo prostético se decide utilizar un sistema operativo de tiempo real. Para la selección del sistema operativo a utilizar se tuvo en cuenta criterios tales como, lenguaje de implementación, portabilidad entre procesadores de diferentes marcas, soporte y costo, entre otros. En la Tabla 4.1 se presenta una comparación entre algunos de los sistemas operativos reconocidos en el mercado.

A partir de la Tabla 4.1 se elige el sistema operativo FreeRTOS dado que cumple con los criterios de selección relevantes para el desarrollo del presente proyecto, tales como: bajo consumo de memoria, portabilidad entre microcontroladores y bajo costo.

ITEM \ RTOS	MQX	uC/OS	FreeRTOS
Expulsivo	SI	SI	SI
Cooperativo	NO	NO	SI
ROM Mnimo	12Kbytes	6Kbytes	2Kbytes
RAM Mnima	2,5Kbytes	-	-
Tareas, ISR	SI	SI	SI
Prioridades	SI	SI	SI
Semaforos	SI	SI	SI
Mutex	SI	SI	SI
Colas	SI	SI	SI
Listas	NO	NO	SI
Lenguaje	C	C	C
Portabilidad	Solamente Coldfire	Diversas plataformas	Diversas plataformas
Soporte	US\$ 12.000 / ao	Incluido en licencia	Comunidad libre
Cdigo abierto	SI	SI	SI
Mltiples protocolos	USB TCP/IP	USB TCP/IP	USB TCP/IP
Costo	US\$ 0	US\$ 6.000	US\$ 0

Tabla 4.1 Comparacin entre algunos de los sistemas operativos reconocidos en el mercado

Respecto a la implementacin de la mano virtual o del video juego se decide utilizar el *Framework XNA Games* desarrollado por *Microsoft* para la creacin de juegos de la consola *XBOX 360*. Es importante resaltar que XNA es una herramienta gratuita de libre distribucin incluso para fines comerciales, el cual se programa a travs del lenguaje C#.

En las Figuras 4.1 y 4.2 se presentan respectivamente la mano virtual y el video juego denominado catapulta mioelctrica.

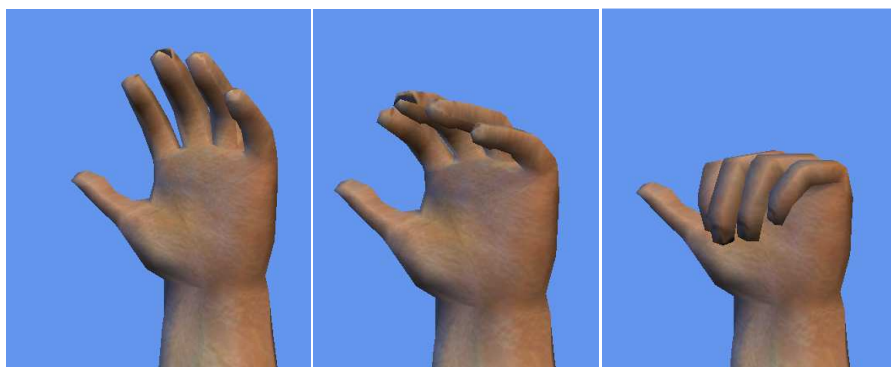


Figura 4.1 Ilustracin de movimiento de apertura y cierre de mano



Figura 4.2 Video juego - Catapulta Mioeléctrica

A continuación se describirá las responsabilidades implementadas en cada uno de los módulos que conforman la arquitectura establecida para el Entrenador Mioeléctrico.

4.1 Módulo Gestor Analógico (MGA)

Se conforma principalmente de un circuito de amplificación diferencial basado en el integrado AD620 fabricado por Analog Device. El diseño del circuito se encuentra en la nota de aplicación (Analog Device, 1999).

De acuerdo con el conjunto normalizado de interfaces propuesto, el módulo MGA debe implementar las responsabilidades *GetSX* y *GetSxAnalogProcessed*. Por lo tanto, para la implementación de la interfaz *GetSX* se utilizan electrodos superficiales desechables, tales como los que se presenta en la Figura 4.3.



Figura 4.3 Electrodo Superficiales

Respecto a la interfaz *GetSxAnalogProcessed* se implementó el circuito presentado en la Figura 4.4, el cual recibe como parámetro de entrada a la señal ME adquirida directamente de los electrodos y retorna la señal ME procesada de manera analógica.

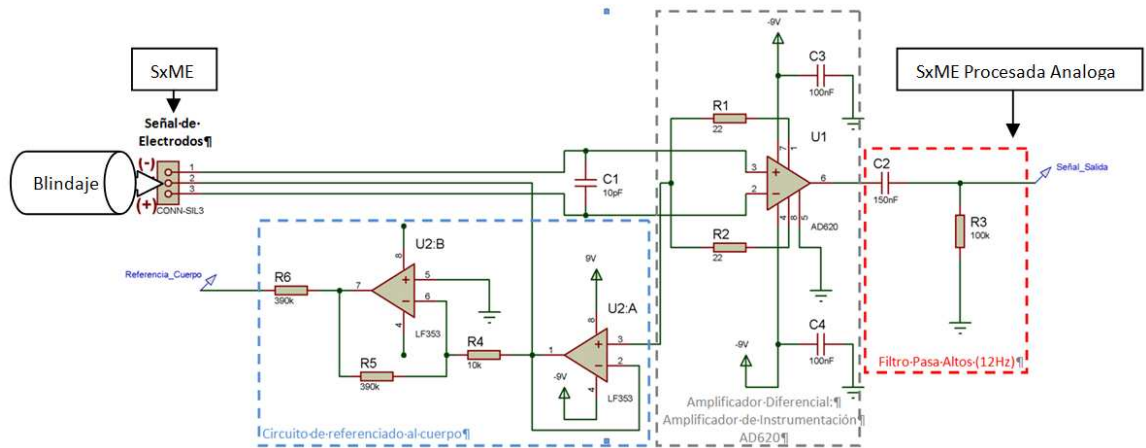


Figura 4.4 Circuito de adquisición y procesamiento analógico de la señal ME

Es importante resaltar que el circuito presentado en la Figura 4.4 es uno de los posibles diseños a implementar por la interfaz *GetSxAnalogProcessed*. Otra posible implementación es la suministrada en las notas de aplicación de la hoja técnica del amplificador de instrumentación INA2128 (Brown, 2009).

4.2 Módulo Gestor Digital (MGD)

Este módulo ejecuta una tarea periódica con periodo de 1ms la cual se encarga de adquirir una muestra de la señal ME procesada analógicamente a través del módulo de conversión analógica digital y posteriormente someterla a un proceso de filtrado. Dadas las exigencias temporales que impone el procesamiento digital de señales, a esta tarea se le asigna la mayor prioridad entre las tareas que controla el sistema operativo.

Para este proyecto se implementó un filtro de promedio móvil en donde la muestra filtrada es almacenada en un arreglo de 125 muestras con el fin de tener disponible los últimos 125ms de información.

Para conocer el resultado de este proceso se realiza a través de las responsabilidades *GetSampleSx* y *GetArraySx* definidas previamente en la arquitectura. Cada que se adquiere una muestra de la señal se cede un semáforo a través de la función *xSemaphoreGive* con el fin de Informarle al Módulo Caracterizador de la Señal (MCS) que se dispone de una nueva muestra de la señal.

4.3 Módulo Caracterizador de la Señal (MCS)

Este módulo ejecuta una tarea que se encarga de detectar el lapso en que las muestras superan un umbral determinado por el 50% de la máxima contracción voluntaria de los músculos en cuestión.

Esta tarea espera a que el Módulo Gestor Digital (MGD) le envíe una señal informado que cuenta con una nueva muestra de la señal ME para ser procesada. De igual manera, como el MDG le envía una señal al módulo MCS, este envía una señal con el fin de informarle al Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM) que se cuenta con un nuevo dato de activación muscular.

Para acceder a la información procesada por este módulo se realiza a través de la interfaz *GetDuration* la cual retorna el tiempo en milisegundo que duró dicha activación.

4.4 Módulo Gestor de Reconocimiento de Intención de Movimiento (MGRIM)

Este módulo ejecuta una tarea que se encarga de determinar la intención de movimiento del paciente a partir de las características generadas por el MCS.

Esta tarea espera a que el Módulo Caracterizador de la Señal le envíe una señal informando que se cuenta con una nueva característica para ser procesada. En este punto se relacionan las características de la señal ME a una intención de movimiento en particular.

Para el presente proyecto se determina que si la duración de la actividad muscular se encuentra entre 200ms y 400ms la intención del paciente es cerrar la mano y si la actividad muscular está entre 400ms y 600ms la intención es abrirla. Estos valores podrían ser ajustados de acuerdo al criterio de usabilidad de cada paciente.

Otra posible relación entre el tiempo de actividad muscular y la intención de movimiento es detectar pulsos de corta duración que superen el 50% de la Máxima Contracción Voluntaria MCV. Por ejemplo, si el usuario genera dos pulsos ME continuos separados entre 100 y 200ms, la intención de movimiento es cerrar la mano y ya sea con la detección de uno o tres pulsos la intención de movimiento podría ser abrirla.

En el instante en que el MGRIM detecta una nueva intención de movimiento este envía una señal al Módulo Controlador del Dispositivo (MCD) con el fin de informarle que se cuenta con una nueva orden para ser ejecutada por el dispositivo protésico.

La información de la intención de movimiento se retorna a través de la interfaz *GetHandGrip* la cual es definida por el conjunto normalizado de interfaces propuesto en este proyecto.

4.5 Módulo Controlador del Dispositivo (MCD)

Este módulo corre una tarea que se encarga de controlar a través de la rotación de las articulaciones los movimientos ejecutados por el dispositivo protésico.

Al recibir una señal por parte del MGRIM informando de una nueva intención de movimiento, el MCD procede a ejecutarlo. Para ello, inicialmente se selecciona la trayectoria a ejecutar y se conoce la posición actual de cada articulación a través del Módulo Gestor Sensorial (MGS) y su interfaz *GetFingerPosition*.

A partir de esta información se procesa un algoritmo de control el cual genera una serie de órdenes que ubican a las articulaciones en la posición predefinida por las trayectorias. Dichas órdenes se retornan a través de la interfaces *GetFingerAngle* o *GetFingerTorque* propuestas en el conjunto normalizado de interfaces.

Es importante mencionar que para el presente proyecto se fijarán los movimientos directamente a través de ángulos, los cuales son referenciados con la convención de “mano extendida, cero grados de rotación”.

4.6 Módulo Gestor Sensorial (MGS)

Este módulo ejecuta una tarea encargada de mantener la información actualizada de la posición angular de cada una de las articulaciones. En este caso, el dispositivo protésico

virtual será quien suministre dicha información la cual será accedida por el MGS a través del puerto de comunicación asíncrona y retornada a través de la interfaz *GetFingerPosition*.

4.7 Módulo Gestor de Realimentación (MGR)

Este módulo ejecuta una tarea encargada de realimentar al usuario determinada información relacionada con el estado del dispositivo protésico. Particularmente en este proyecto se define que la información a realimentar es el momento en que el dispositivo protésico termina la ejecución de un movimiento definido por el usuario. Dicha información es retornada a través de la interfaz *GetSoundLevel*.

5 Pruebas del entrenador Mioeléctrico

El Entrenador Mioeléctrico se ha probado en cinco personas no amputadas y sin conocimiento previo del control del sistema. En cada una de las personas se siguió el siguiente protocolo:

1. Se conecta dos electrodos superficiales separados por 1cm en los músculos flexores del antebrazo. Se conecta un tercer electrodo al codo del paciente, el cual se utiliza como referencia de tierra para el sistema electrónico.
2. Se le solicita a la persona que realice un empuñamiento con el fin de observar la señal mioeléctrica de manera gráfica.
3. Se instruye al paciente en el control del video juego y se solicita que intente controlarlo.

4. Se instruye al paciente en el control de la mano virtual y se solicita que intente controlarla.

De acuerdo al protocolo anteriormente expuesto se registraron los tiempos que se tarda el paciente en cada uno de los pasos, ver Tabla 5.1.

Persona	1	2	3	4	5
Paso del protocolo	(min)	(min)	(min)	(min)	(min)
1. Tiempo de adaptación del Entrenador.	3 min.	4 min.	3 min	5 min	7 min
2. Tiempo para la generación de la Señal ME.	2 min	5min	4min	5min	2min
3. Tiempo para el control del video Juego.	10 min	12min	10min	15min	10min
4. Tiempo para el control de la mano virtual.	1min	2min	3min	2min	3min

Tabla 5.1 Registro temporal de los pasos del protocolo de prueba.

En la Tabla 5.1 se evidenció que en el control del video juego se tardó un mayor tiempo respecto a los demás pasos del protocolo, sin embargo, es importante resaltar que las cinco personas lograron terminar los cuatro pasos de manera exitosa en un tiempo menor a 30 minutos.

Al final de la realización del protocolo se le preguntó a cada persona acerca de la experiencia con el Entrenador; se les preguntó que tan difícil les había parecido controlar tanto del video juego como de la mano virtual, si el juego era entretenido o no, qué impresión les causó el aspecto de la mano virtual y por último cuál sería su sugerencia para mejorar al Entrenado Mioeléctrico.

En general las respuestas coincidieron en lo mismo; básicamente las cinco personas afirmaron que la generación gráfica de la señal mioeléctrica y el control de la mano virtual habían sido los pasos más sencillos respecto al control del video juego, el cual tuvo un mayor grado de dificultad debido a que no solo hay que genera la señal ME sino también porque hay que comprender la dinámica del mismo. Sin embargo, afirmaron que gracias la motivación que les generó el video juego y al grado de destreza adquirido

en él se les facilitó en gran medida el posterior control de los movimientos del dispositivo protésico o mano virtual.

Respecto a al aspecto de la mano virtual en general la impresión fue buena; ya que les pareció de un aspecto agradable, incluso entretenida, generado cierta motivación para su control. Mencionan que fue muy satisfactorio ver que la mano respondiendo a las órdenes generadas y que en dos de las cinco personas afirmaron que un momento dado sintieron la mano virtual como si realmente fuera la de ellos.

Las sugerencias para el mejoramiento del Entrenador Mioeléctrico fueron: adicionar más video juegos incluso de mayor dificultad, adicionar otros posibles agarres a la mano virtual e incluir el movimiento de la muñeca.

Por último, a manera de ilustración en la Figura 5.1 se muestra la conexión del entrenador virtual a una persona y se identifican las partes que conforman al Entrenador Mioeléctrico.

Esta gráfica es temporal dado que hasta el momento no cuento con una buena imagen.

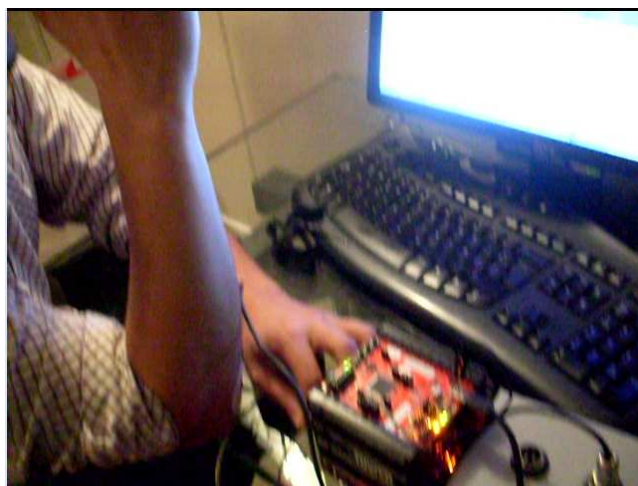


Figura 5.1 Entrenador Mioeléctrico conectado a una persona

6 Conclusiones

La simplicidad, portabilidad y bajo costo del sistema posibilitan la utilización del Entrenador Mioeléctrico directamente en el domicilio del paciente; de esta manera se facilita y acorta el período de aprendizaje para el control la prótesis mioeléctrica real. Esta característica es de gran importancia debido a que no todos los pacientes cuentan con los recursos necesarios para adquirir prontamente una prótesis mioeléctrica, por lo tanto, convierte al Entrenador en una buena herramienta para mantener las conexiones neuronales activas y continuar generando señal mioeléctrica adecuadamente.

El haber implementado la arquitectura de control del Entrenador Mioeléctrico directamente en un microcontrolador nos permite acercarnos al comportamiento que se obtendría si se dispusiese del dispositivo prostético real y de esta manera realizar una evaluación más asertiva de las posibles y variadas estrategias que se implementan en dispositivos de esta índole. Así, al contar con el dispositivo prostético real, tanto para los desarrolladores como para los usuarios, la migración del dispositivo virtual al real contendrá menos traumatismo y se habrá logrado el objetivo del simulador el cual es entrenar a los pacientes para el control de la prótesis mioeléctrica real.

Generar una arquitectura del sistema en la cual se definan detalladamente sus responsabilidades, permitió abstraer en un solo plano de una manera detallada los componentes que integran una prótesis mioeléctrica de mano. Luego de una previa socialización del modelo de análisis con los integrantes del Grupo de Investigación se evidenció el ágil entendimiento de los subsistemas que integran un desarrollo de esta

índole. Tener en un mismo plano gráfico interrelacionadas la mayoría de soluciones en el área de prótesis mioeléctrica de mano, permitió a los integrantes del proyecto realizar aportes en miras al mejoramiento de la solución propuesta.

7 Bibliografía

Alonso, A., Sánchez, R., Espino, P., Liptak, L., & De la Rosa, R. (2002). *Myoelectric prostheses trainer for hand and arm amputees* (Vol. 3). Barcelona: Medicina Mapfre.

Analog Device. (1999). *Low cost, low power: Instrumentation amplifier*. Recuperado el Noviembre de 2008, de <http://users.ece.utexas.edu/~valvano/Datasheets/AD620.pdf>

Brown, B. (2009). *INA2128*. Recuperado el 12 de 10 de 2009, de <http://www.datasheetcatalog.com/>

Cabrera, J. S. (2009). *Conjunto normalizado de interfaces para los componentes de un simulador virtual de prótesis de mano robótica*. Tesis de Maestría, Universidad del Cauca, Popayán.

Choi, H. (2010). *Connexions, Fixed Point Arithmetic*. Recuperado el 17 de Enero de 2010, de <http://cnx.org/content/m11054/latest/>

Dupont, A., & Morin, E. (1994). A myoelectric control evaluation and trainer system. *EEE Transactions on Rehabilitation Engineering* , 100-107.

Farrell, T. R., & Weir, R. F. (2007). The Optimal Controller Delay for Myoelectric Prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* , 15 (1), 111-118.

Otto Bock HealthCare GmbH . (14 de 01 de 2010). *Ottobock quality for life*. Recuperado el 2010, de <http://www.ottobock.com.mx/>

Rodriguez, L. E. (2006). *PROTECI*. Barcelona: Universidad Politécnica de Cataluña.

Serrano, C. E. (2005). *Modelo integral para el profesional en ingeniería*. Popayán: Universidad del Cauca.