

MANUAL DE USUARIO



CENTRO DE BIOINFORMÁTICA Y BIOLOGÍA COMPUTACIONAL DE COLOMBIA

Elaborado por:

Luis Eduardo Vélez Quintero
Esteban Correa.
Carlos Diego Ferrin.



Tabla de Contenido

1.	Introducción.	3
2.	Metodología.	3
3.	Protocolo de preparación del paciente.	4
4.	Módulo 1: Control de movimiento.	6
4.1.	Descripción.	6
4.2.	Instalación.	6
4.2.1.	Generación de proyecto en CMake.	6
4.3.	Configuración.	8
4.3.1.	Detección de eventos.	9
4.3.2.	Movimiento de miembro virtual con ViLimbs.	10
4.3.3.	Modificación de umbrales.	10
4.4.	Ejecución.	11
4.4.1.	Ejecución por consola.	11
4.4.2.	Ejecución por interfaz gráfica.	13
5.	Módulo 2: Realidad aumentada.	15
5.1.	Descripción.	15
5.2.	Instalación.	15
5.2.1.	Instalación OpenCV 2.4.11.	15
5.2.2.	Instalación de la biblioteca de marcadores Aruco.	16
5.2.3.	Instalación Ogre.	16
5.2.4.	Calibración Cámara.	16
5.2.5.	Compilación ViLimbs.	17
5.3.	Ejecución.	17
6.	Anexos.	19
6.1.	Configuración módulo 1: Control de movimiento.	19
6.1.1.	Emotiv Xavier Control Panel	19
6.1.2.	Myo Connect	20
6.2.	Configuración módulo 2: Realidad Aumentada.	22
6.2.1.	Problemas Frecuentes.	22
6.3.	Eventos y publicaciones.	23
6.3.1.	CHI 2015	23
6.3.2.	ANNA 2015	25

1. Introducción.

En este documento se compila información amplia y suficiente acerca de la aplicación ViLimbs, la cual constituye una de las aplicaciones operativas del Muro de Visualización en el marco del proyecto titulado “FORTALECIMIENTO DE CTel EN BIOTECNOLOGÍA PARA EL DEPARTAMENTO DE CALDAS APOYADO POR INFRAESTRUCTURA COMPUTACIONAL AVANZADA Y TRABAJO COLABORATIVO” de conformidad con las obligaciones contenidas en el Convenio Especial de Cooperación No. 08112013-0621 suscrito entre la Gobernación de Caldas y el Centro de Bioinformática y Biología Computacional de Colombia, Sigla CBBC.

ViLimbs es una versión virtualizada de la terapia de espejo, una técnica ampliamente utilizada en el tratamiento de las sensaciones fantasma en pacientes amputados. Con ayuda del Muro de Visualización, que en este caso actúa como espejo, se obtiene un reflejo del paciente amputado quien con la ayuda de la computación se vuelve a ver si misma con su miembro amputado recuperado pudiendo inclusive controlar los movimientos del mismo.

A continuación se exponen la metodología, el protocolo de preparación del paciente y los procesos de configuración e instalación a nivel software de los dos módulos que constituyen el aplicativo ViLimbs. A este documento lo acompaña también todo el código fuente, instaladores y modelos 3D necesarios para compilar y ejecutar el aplicativo.

2. Metodología.

Para el desarrollo de ViLimbs se contempló una arquitectura distribuida tipo servidor-cliente para alivianar la carga computacional de los dos elementos importantes que constituyen esta aplicación, a saber, el módulo de control de movimiento y el de realidad aumentada. Ver Figura 1. El módulo de control de movimiento es una aplicación encargada de decodificar las intenciones de movimiento del paciente a partir de señales electromiográficas y electroencefalográficas, valiéndose para ello del Myo device y del EPOC+ respectivamente. El módulo de realidad aumentada se encargar de visualizar, en el Muro de Visualización de BIOS, la imagen del paciente con un miembro virtual superpuesto. Para esto, se utiliza una cámara, un modelo de brazo 3D y un marcador. El marcador se ubica estratégicamente sobre el paciente a fin de que el brazo 3D se puede sobreponer adecuadamente sobre la imagen del muro una vez que la cámara junto con un algoritmo de visión por computador deduzcan la orientación en el espacio físico del marcador.



Figura 1. ViLimbs: módulos, herramientas y caso de uso.

3. Protocolo de preparación del paciente.

En la Figura 2 se puede observar el protocolo de preparación del paciente amputado. El paciente debe estar sentado antes de ponerle los elementos mostrados en el lado izquierdo de esta misma figura. El marcador, sujeto a una bracería, se ubica de tal forma que la misma quede sobre el bíceps del paciente.

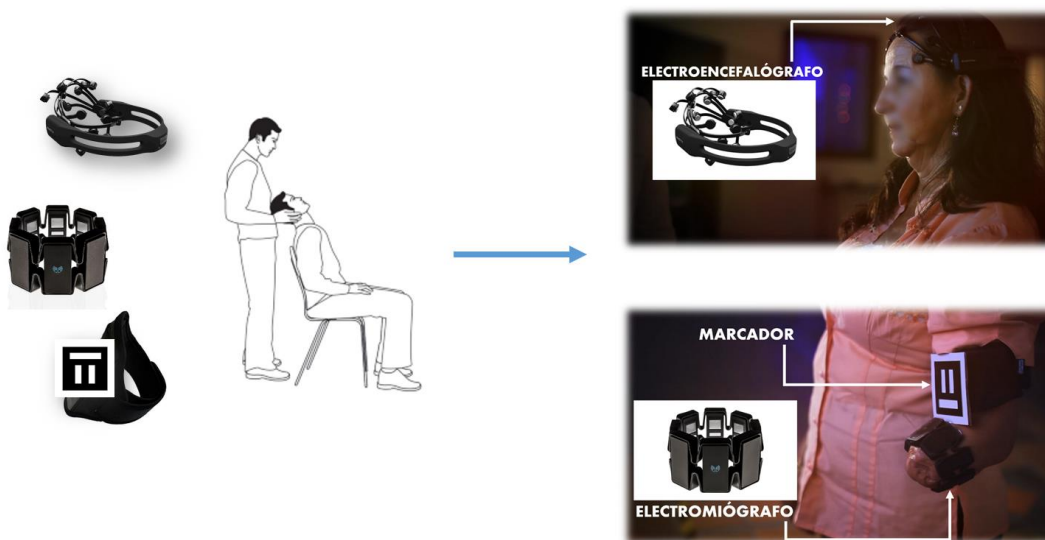


Figura 2. Protocolo de preparación del paciente para uso de ViLimbs.

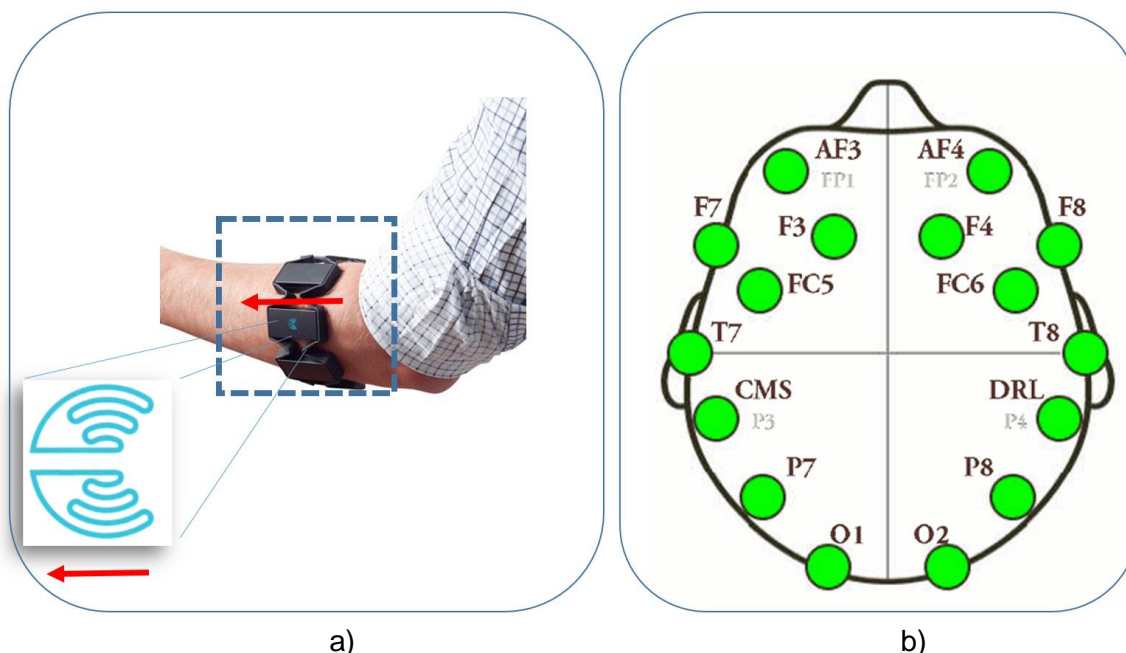


Figura 3. a) Orientación del MYO device. b) Ubicación de los electrodos del EPOC Plus y el color que deben presentar en la interfaz de preparación.

El Myo device se ubica cerca de la zona del muñón y debe estar orientado tal como se indica en la Figura 3a. La orientación de este dispositivo es muy importante dado que el giroscopio adquiere sus valores basado en la posición de sus ejes virtuales preestablecidos, por lo tanto, si el Myo no se pone de forma consistente en todas las personas, los valores arrojados por el giroscopio serán diferentes aun cuando se realicen los mismos movimientos.

En <https://www.myo.com/> se puede tener más información sobre el correcto uso de este dispositivo. Por su parte el EPOC+ debe colocarse sobre el cuero cabelludo del paciente siguiendo las recomendaciones dadas en <https://emotiv.com/> acerca de este dispositivo. La interfaz de control del EPOC+ permite visualizar el adecuado contacto de los electrodos sobre el cuero cabelludo generándose un color verde esto se produce. Ver Figura 3b.

Finalmente el paciente, en posición erecta, se ubica enfrente de la cámara la cual debe ubicarse adecuadamente para que el muro genere la sensación de estar parado enfrente en un espejo. En las siguientes secciones se detallará un poco más al respecto.

4. Módulo 1: Control de movimiento.

4.1.Descripción.

El primer módulo de ViLimbs es una aplicación enfocada al uso de interfaces hombre-máquina en el proceso de terapia para el miembro fantasma. El primer dispositivo utilizado es una interfaz controlada mioeléctricamente (MCI) denominada Myo, con ella se adquieren señales bioeléctricas provenientes de los movimientos musculares, en este caso específico, es usado para detectar las señales del antebrazo. El segundo dispositivo es una interfaz cerebro-computador (BCI) denominado Emotiv EPOC+ que procesa las señales bioeléctricas cerebrales para controlar alguna actividad externa.

Las dos interfaces son usadas para adquirir las señales, y mediante algoritmos computacionales y técnicas de procesamiento de señales se logra que el miembro virtual del segundo módulo de ViLimbs tenga los movimientos que la persona intenta realizar a partir de manifestaciones cerebrales y musculares.

En resumen, el primer módulo de ViLimbs, denominada “ViLimbs Controller” es la aplicación encargada de procesar las señales detectadas en los dispositivos y de enviar los comandos para el movimiento del miembro virtual construido. Está diseñada para funcionar en sistema operativo Microsoft Windows.

4.2.Instalación.

4.2.1. Generación de proyecto en CMake.

CMake es un programa que facilita la construcción de proyectos de desarrollo para diferentes IDE en diferentes plataformas. El proyecto “*ViLimbs_Controller*” fue desarrollado en C++ en Visual Studio 2010 para plataformas con sistema operativo Windows. Teniendo en cuenta que el proyecto tiene una versión de consola y otra versión con interfaz gráfica, el proceso que se describe a continuación aplica para cualquiera de los dos proyectos aunque se tomará de ejemplo la versión gráfica.

Para iniciar la generación del proyecto se debe especificar donde se ubica el código fuente de CMake denominado “*CMakeLists.txt*”, en el ejemplo de la Figura 4 el código fuente se encuentra en la carpeta “*../ViLimbs_Controller/ViLimbs_GUI*”. Luego se debe seleccionar la ruta donde se creará el proyecto del IDE especificado, se recomienda generarla en una carpeta diferente a donde se encuentra el código fuente con el fin de facilitar la eliminación de generación de proyectos sin afectar el código original; en el ejemplo la ruta se realiza en “*../ViLimbs_Controller/ViLimbs_GUI/BUILD2010/*” haciendo referencia a la versión de Visual Studio en la que se construyó el proyecto.

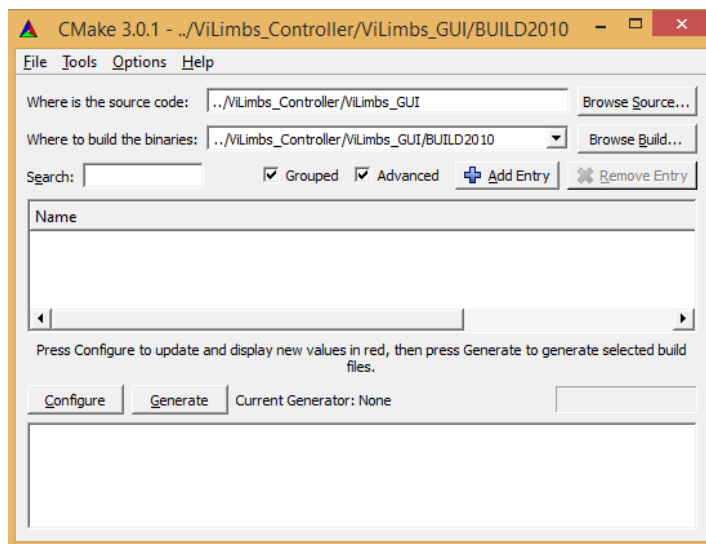


Figura 4. Configuración de CMake.

Se recomienda usar Visual Studio 2010 para generar el proyecto de CMake debido a que es la versión en la que se desarrolló la aplicación originalmente y tiene plena compatibilidad con las librerías utilizadas. Sin embargo, es posible configurar el proyecto para entornos de CodeBlocks, CodeLite, Eclipse, SublimeText, entre otros.

Al configurar el proyecto se solicitarán las rutas donde se encuentran las bibliotecas externas con sus respectivas funciones. Para el proyecto solo es necesaria una librería externa, la ruta de la carpeta se ilustra en la Figura 5, en esta carpeta se ubica el SDK con el código fuente necesario para la conexión y adquisición de datos provenientes del Emotiv.

Name	Value
Ungrouped Entries	
XAVIERSDK_DIR	..\ViLimbs_Controller\ViLimbs_GUI\3rdparty\Emotiv\XavierSDK-Premium
CMAKE	

Figura 5. Configuración librerías CMake.

Luego de configurar CMake se debe generar la solución del proyecto presionando el botón “**Generate**”, este crea los archivos necesarios para el IDE seleccionado. Cuando la configuración y la generación del proyecto se han realizado correctamente se muestra un mensaje como el de la Figura 6. Las advertencias que se muestran en rojo pueden ser omitidas ya que no afectan el desarrollo de la solución del proyecto.

```

Configure Generate Current Generator: Visual Studio 10 2010

CMake Warning (dev) at app/CMakeLists.txt:28 (link_
  This command specifies the relative path

  ../../3rdparty/Myo/lib

as a link directory.

Policy CMP0015 is not set: link_directories() tre
source dir. Run "cmake --help-policy CMP0015" fo
cmake_policy command to set the policy and suppre
This warning is for project developers. Use -Wno-d

Configuring done
Generating done

```

Figura 6. Fin de configuración de CMake.

El proyecto de Visual Studio 2010 incluye varios archivos de configuración del IDE, de CMake y un árbol de carpetas relacionado con el modo de compilación del proyecto; sin embargo, el archivo que da apertura a todo el proyecto configurado tiene extensión “.sln” como se resalta en la Figura 7. A partir de allí se puede continuar el proceso de desarrollo y compilación del proyecto según las necesidades existentes.

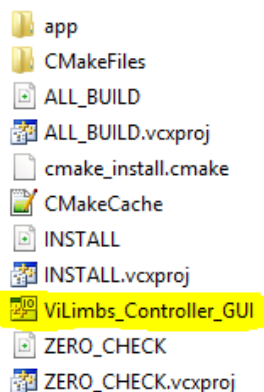


Figura 7. Solución generada para Visual Studio 2010.

4.3. Configuración.

La preparación inicial del Emotiv EPOC+ y del Myo se realiza teniendo en cuenta el siguiente procedimiento:

- Conectar los 16 electrodos a cada uno de los canales del Emotiv, verificando que se encuentren lo suficientemente húmedos para una correcta transmisión de señales.
- Conectar el receptor bluetooth al computador en el cual se ejecutarán las aplicaciones.
- Conectar el cable USB al Myo y al computador, esta es la única manera de encender el dispositivo si no está en estado de reposo.
- Conectar el receptor bluetooth al computador para habilitar la adquisición de información desde el Myo.

“ViLimbs Controller” se debe ejecutar la aplicación después de haber iniciado el “Emotiv Xavier Control Panel” y el “Myo Connect” (la configuración de cada aplicación se describe paso a paso en el primer anexo, sección 6.1.), estos dos programas son los que habilitan la adquisición de datos a partir de cada uno de los dispositivos, estos datos son procesados para enviar comandos que muevan el miembro virtual en la aplicación de realidad aumentada de ViLimbs explicada en la sección 5. A continuación se describen algunos de los aspectos que comprenden el funcionamiento del módulo de control de movimiento de ViLimbs.

4.3.1. Detección de eventos.

La aplicación obtiene los eventos provenientes del Emotiv y el Myo, luego los asocia a uno de los comandos de control del miembro virtual. Los eventos detectados del Emotiv son dos:

- **Emotiv Neutral:** Es el estado por defecto en el Emotiv, enviado cuando la persona no genera señales EEG fuertes que puedan ser discriminadas en eventos.
- **Emotiv Push:** Se genera a partir de un entrenamiento de señales EEG con el Emotiv Xavier Control Panel, el evento se activa con la aparición de la señal entrenada. El entrenamiento de esta señal se describe detalladamente en el anexo 6.1.1.

Del dispositivo Myo no es coherente entrenar los eventos que trae el software por defecto debido a que son diseñados para personas con sus brazos completos, en este caso, como se trabaja con personas con una deficiencia física, se realiza el entrenamiento de otros eventos que permitan ser clasificados con mayor facilidad aún en personas que les falta alguna parte de sus extremidades. Teniendo en cuenta lo anterior, los eventos que se detectan con el Myo son descritos a continuación:

- **Myo Neutral:** Es el evento por defecto en el Myo cuando el brazo está en estado de reposo, donde no existe ninguna señal EMG lo suficientemente fuerte.
- **Myo Push:** Este evento es activado cuando la zona muscular donde se ha colocado el dispositivo es tensionado y presenta gran potencia en la señal EMG, esto no implica una modificación en la posición del brazo.
- **Myo Close Arm:** Este evento es detectado usando el giroscopio, la señal se activa cuando la persona flexiona el codo. La posición del codo en la cual se activa la señal puede ser modificada fácilmente configurando los umbrales de detección, explicado en la sección 4.3.3.
- **Myo Extend Arm:** Este evento es clasificado usando los datos obtenidos del giroscopio, se activa cuando la persona extiende el brazo hasta el umbral definido.

Los eventos anteriores pueden ser ejecutados de forma simultánea, es decir, con el Myo se puede tener el brazo extendido o cerrado mientras se está tensionado. De esta forma se envían dos comandos.

4.3.2. Movimiento de miembro virtual con ViLimbs.

El módulo dos de ViLimbs es una aplicación de realidad aumentada dirigida a funcionar en sistema operativo Linux, por lo tanto es incompatible con “ViLimbs Controller” concebida para ejecutarse sobre Windows.

Con el fin de dar solución a controlar el brazo procesado en Linux desde la aplicación de Windows, se implementó un protocolo de comunicaciones con el cual se pueden enviar los comandos al módulo de realidad aumentada de ViLimbs desde cualquier computador en la red LAN, usando comunicación por sockets.

Los comandos permitidos por ViLimbs para activar los movimientos en el miembro virtual a partir de los datos obtenidos por el socket son los mostrados en la Tabla 1.

Tabla 1. Comandos admitidos por ViLimbs.

COMANDO	DESCRIPCIÓN
RH	Posición por defecto del miembro virtual. Cada vez que se ejecuta este comando se llega a la posición inicial en los dedos, muñeca y codo.
EH	La muñeca del brazo se mueve apuntando hacia afuera.
FH	La muñeca del brazo se mueve apuntando hacia adentro.
CH	Los dedos de la mano se cierran completamente.
FA	El codo se dobla para formar un ángulo de 90° entre el antebrazo y el bíceps.
EA	El codo se extiende para quedar paralelo al eje del cuerpo de la persona.

4.3.3. Modificación de umbrales.

Hay dos umbrales básicos que se requieren modificar para adaptar la clasificación de las señales según las condiciones fisiológicas de cada persona. Ambos están relacionados a la detección de eventos con el Myo.

El primer umbral delimita la **potencia** de la señal requerido para que se envíe el evento MyoNeutral o MyoPush; y el segundo umbral define el valor de **posición** que activa los eventos MyoCloseArm y MyoExtendArm basado en los datos recopilados del giroscopio. En la Tabla 2 se explican detalladamente los umbrales y los eventos asociados a cada uno de ellos.

Tabla 2. Configuración de umbrales en el Myo.

UMBRAL	RANGO	EVENTOS	DESCRIPCIÓN
Potencia	[0, 4000] Depende de la potencia de la señal EMG obtenida en los 8 canales del Myo.	Myo-Neutral	Emitido cuando la potencia es menor al umbral especificado.
		Myo-Push	Emitido cuando la potencia es mayor al umbral especificado.
Giroscopio	[-1, 1] El valor -1 se da cuando el brazo está extendido al lado del cuerpo, el valor 1 se da cuando el antebrazo y el bíceps están unidos.	Myo-Extend Arm	Emitido cuando el valor del giroscopio es menor que el especificado.
		Myo-Close Arm	Emitido cuando el valor del giroscopio es mayor al umbral especificado.

4.4. Ejecución.

Al instalar la aplicación “ViLimbs Controller” se crea el acceso directo a la versión gráfica del programa. Sin embargo, dentro de los archivos que se crean en el directorio de instalación hay un ejecutable llamado “ViLimbs_Console.exe” como se muestra en la Figura 8, el cual es una versión en línea de comandos de ViLimbs Controller.

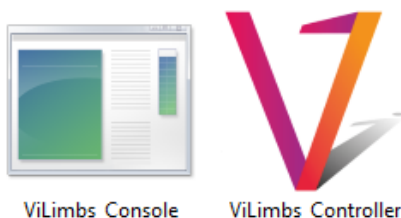


Figura 8. Aplicación de consola y gráfica de ViLimbs.

A continuación se describen aspectos importantes a tener en cuenta para la ejecución de la aplicación en cada modo.

4.4.1. Ejecución por consola.

Al ejecutarse se despliega una consola de Windows donde se pregunta qué dispositivos se desean usar, es posible ejecutar la aplicación solo con el Myo, solo con el Emotiv o ninguno (aunque quedaría inútil la aplicación).

```
Do you want to connect to Myo? (No: 'N' key - Y: Any key): n
Do you want to connect to EPOC? (No: 'N' key - Y: Any key): y
```

Figura 9. Configuración de dispositivos a usar en ViLimbs versión consola.

La configuración de qué comando se va a enviar con cada evento detectado se realiza desde un archivo de configuración llamado “Myo_EPOC_Commands.txt”, este archivo está localizado en la misma carpeta de instalación de la aplicación. Si al iniciar el programa no existe el archivo se creará automáticamente con unos datos de plantilla con la estructura mostrada a continuación:

```
RH --->Myo NEUTRAL
CH --->Myo PUSH
RH --->Myo Close arm
EA --->Myo Extend arm
RH --->EPOC NEUTRAL
FH --->EPOC PUSH
```

En este archivo se asocia un comando a un posible evento detectado, lo único que se debe modificar son las dos primeras letras de cada fila basado en los comandos descritos en la sección 4.3.2, las cuales configuran la posición del miembro virtual.

La aplicación por consola recibe parámetros para configurar características de funcionamiento, como son los umbrales para el giroscopio y la potencia de la señal, y la dirección IP del equipo Linux donde se ejecuta el módulo de realidad aumentada de ViLimbs. Todas las posibles configuraciones por consola se describen a continuación.

Si la aplicación se ejecuta sin parámetros de entrada, se ejecutará de manera local sin enviar los comandos vía WinSock al equipo que corre el procesamiento gráfico de ViLimbs. Generalmente es utilizado para efectos de prueba de los umbrales.

```
| >>ViLimbs_Console.exe
```

Si solo se pasa un parámetro al ejecutar la aplicación, como se muestra a continuación, se conectará a la dirección IP especificada. Se debe tener en cuenta que la dirección dada debe ser visible desde el computador, esto puede comprobarse con el comando ping. Los umbrales quedan configurados por defecto como 1000 para la potencia y 0.2 para el giroscopio.

```
| >>ViLimbs_Console.exe [DireccionIP]
```

Si se ejecuta la aplicación con dos parámetros, se modifican los umbrales de la forma descrita a continuación, tener en cuenta los límites de cada umbral para un correcto funcionamiento. En este modo de ejecución no se envían comandos a ViLimbs y puede ser usado para pruebas de funcionamiento de los umbrales.

```
| >>ViLimbs_Console.exe [UmbralPotencia] [UmbralGiroscopio]
```

La última opción es ejecutar la aplicación con tres parámetros, los umbrales y la dirección IP destino. Esta es la opción más completa porque personaliza los umbrales según la persona que lo usa y realiza la comunicación por sockets para utilizar el brazo de realidad aumentada de ViLimbs.

```
| >>ViLimbs_Console.exe [Umb.Potencia] [Umb.Giroscopio] [DireccionIP]
```

4.4.2. Ejecución por interfaz gráfica.

La versión gráfica de “ViLimbs Controller”, mostrada en la Figura 10, es una forma más agradable de utilizar la aplicación y propicia para usuarios menos experimentados. En ella se pueden configurar fácilmente los dispositivos a usar, la conexión con las aplicaciones externas, la modificación de umbrales y comandos a enviar.

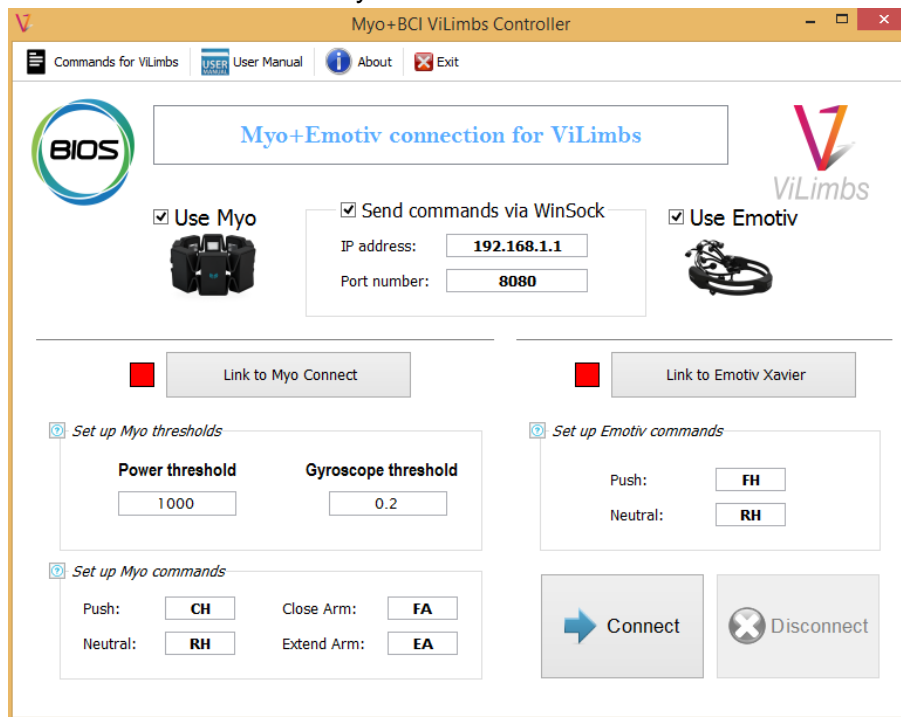


Figura 10. Módulo de control de movimiento para ViLimbs, versión gráfica.

La primera sección de la interfaz está compuesta por dos partes (Figura 11), la selección de los dispositivos que se usarán y si se desea usar el protocolo de comunicación por sockets para enviar los comandos a ViLimbs. Dependiendo de los dispositivos que se utilicen se habilita su respectiva configuración y si hay algún inconveniente en la comunicación con el destino se mostrará un error al intentar conectar el servicio.

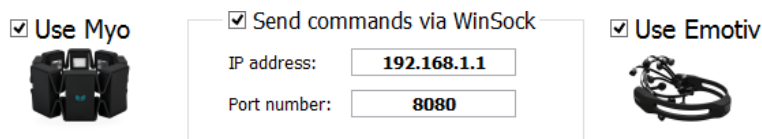


Figura 11. Configuración de dispositivos y comunicación.

Dependiendo de la interfaz a utilizar, se debe realizar el enlace con cada una de las aplicaciones externas, para el Myo es “Myo Connect” y para el Emotiv es el “Emotiv Xavier Control Panel” los cuales deben estar abiertos y configurados antes de presionar los botones de enlace mostrados en la Figura 12, si existe algún problema de configuración en el proceso se muestra la ventana con el error respectivo.



Figura 12. Conectar a las aplicaciones externas para el control de dispositivos.

En caso de utilizar el Myo, se deben configurar los umbrales descritos en la sección 4.3.3, esto se realiza utilizando los cuadros de texto mostrados en la Figura 13. Tener en cuenta los rangos de valores permitidos para cada variable.

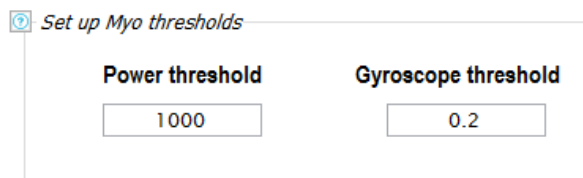


Figura 13. Modificación de umbrales para la detección de eventos en el Myo.

Cada evento detectado desde el Myo o el Emotiv tiene asociado un comando que mueve el brazo virtual de ViLimbs en el módulo de realidad aumentada. Estos comandos se configuran usando las casillas mostradas en la Figura 14, para ver los posibles comandos de configuración se puede presionar el botón en la esquina superior izquierda de cada recuadro o la primera opción de la barra de herramientas.



Figura 14. Configuración de comandos por evento.

Finalmente, para iniciar o detener el servicio establecido en la aplicación se usan los botones mostrados en la Figura 15. Solo se puede conectar la aplicación cuando se ha seleccionado alguno de los dispositivos y se ha realizado el enlace con el programa externo respectivo. Si existe algún problema en el inicio del servicio relacionado con el protocolo de comunicación o enlace con las aplicaciones externas se informará con la ventana respectiva.



Figura 15. Conexión y desconexión de servicio.

5. Módulo 2: Realidad aumentada.

5.1.Descripción.

El módulo de realidad aumentada es una aplicación que se ejecuta localmente en Linux Ubuntu. Este módulo utiliza una cámara Logitech C930 junto con la información de la calibración de la cámara, la cual ha sido obtenida previamente mediante un proceso de calibración de cámara, para obtener la orientación de un marcado ubicado en el bíceps del paciente. Cuando esta orientación se detecta, se procede a renderizar el brazo 3D sobre la imagen 2D obtenida por la cámara, creándose la ilusión de un paciente amputado que ha recuperado su extremidad virtualmente. Este módulo se apoya en las bibliotecas de OpenCV, OGRE y Aruco para llevar a cabo el proceso de superposición de miembro virtual sobre el paciente amputado.

A continuación se describe el proceso de configuración, compilación e instalación de este módulo desde su código fuente.

5.2.Instalación.

Para la correcta instalación de ViLimbs son necesarios los siguientes requerimientos de software:

- OpenCV 2.4.11
- Aruco 1.3.0
- Ogre 1.9.0

A continuación se describe el proceso de instalación y configuración de cada uno de estos paquetes.

5.2.1. Instalación OpenCV 2.4.11.

La instalación de esta librería, tiene dos formas: la primera a través del manejador de paquetes (apt) y la segunda, a través de la compilación directa del código fuente, la cual es la recomendada por el autor. Esta sección explicará el segundo método de instalación.

Primero se deben instalar los paquetes por defecto requeridos para la compilación del código fuente, esto es *gcc*, *python* y los codecs *ffmpeg* (se supone que el lector posee permisos de administrador de la máquina ya que serán necesarios para completar el proceso:

```
$ sudo apt-get install build-essential  
$ sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev  
libavformat-dev libswscale-dev cmake ccmake-gui  
$ sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev  
libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev
```

Se debe descargar el código fuente de OpenCV en versión 2.4.11 o 2.4.9 desde la siguiente página web: <http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/>

Se descomprime el archivo donde se instalará la biblioteca y se configura con los siguientes comandos:

```
$ cd opencv_folder/  
$ mkdir build  
$ cmake ../
```

En este momento se debe verificar que cmake encuentra todas las dependencias como *python*, *libpng*, etc. Si todo va bien se configura presionando dos veces la tecla “c” y finalmente se genera un archivo “*makefile*” con la tecla “g”. Una vez generado el archivo se procede a compilarlo e instalarlo mediante las siguientes instrucciones, el proceso durará entre 10 y 30 minutos dependiendo la configuración del equipo base.

```
$ sudo make  
$ sudo make install
```

5.2.2. Instalación de la biblioteca de marcadores Aruco.

ViLimbs hace uso extensivo de los marcadores fiduciaros, los cuales requieren de una librería extra conocida como Aruco, de libre distribución LGPL y que se puede descargar de la dirección web <http://sourceforge.net/projects/aruco/files/>

Una vez descargada, se descomprime el archivo y se compila el proyecto de la siguiente forma:

```
$ mkdir build  
$ cd build  
$ cmake ..  
$ make  
$ sudo make install
```

Se puede optar por probar uno de sus ejemplos siguiendo los pasos del archivo de instrucciones “*Readme*”, descargando el paquete opcional de datos desde la página <http://sourceforge.net/projects/aruco/files/testdata/>

5.2.3. Instalación Ogre.

Ogre es el motor gráfico usado por la solución para el despliegue de los elementos virtuales, su instalación es muy sencilla y es soportada por el manejador de paquetes apt, bastando solo un comando para su instalación:

```
$ sudo apt-get install libogre-1.9-dev
```

5.2.4. Calibración Cámara.

Una vez se tiene todo listo, es necesario calibrar la cámara a usar. Mediante este proceso se obtienen los parámetros intrínsecos de la cámara con el fin de corregir las aberraciones de la imagen que son agregadas por los distintos lentes que la componen. En este caso OpenCV ofrece una forma de hacerlo fácilmente. Sólo es necesario mostrar imágenes de un panel de un tablero de ajedrez de dimensiones conocidas (adjunto en la carpeta de ViLimbs en un archivo llamado *pattern.jpg*) y la toma de varias fotos desde distintas poses del tablero, dando como resultado un archivo *.yml* que será usado por el aplicativo. Es importante mencionar que cada cámara requiere **obligatoriamente** de una calibración

debido a que un lente nunca es igual a otro, en la Figura 16 se muestra un ejemplo de calibración usando el tablero de ajedrez. A continuación se ilustran los comandos necesarios para calibrar la cámara, haciendo uso de uno de los aplicativos de OpenCV que se encuentra dentro de los ejemplos compilados.

```
| $ ./calibration -w 8 -h 6 -s 0.025 -o camera.yml -op
```

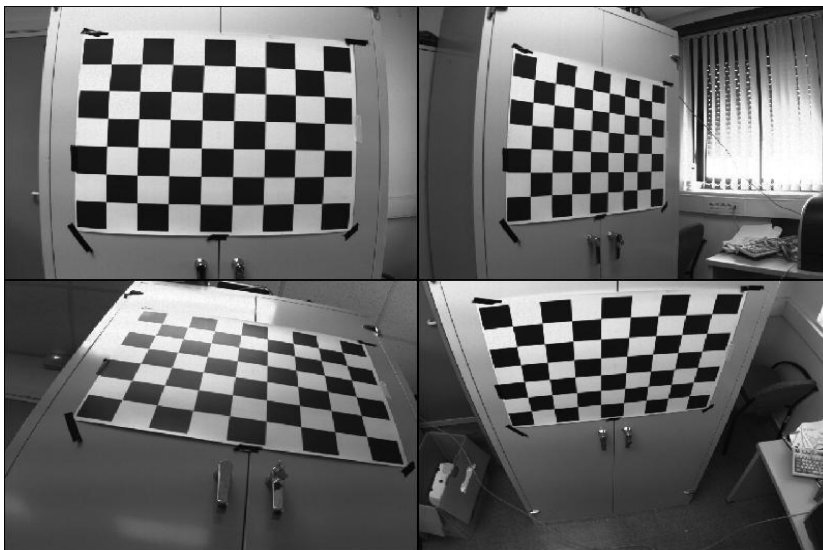


Figura 16. Ejemplo de las distintas poses del tablero de ajedrez (Screenshot tomado del proyecto Aruco versión web).

5.2.5. Compilación ViLimbs.

ViLimbs al igual que OpenCV y aruco, hace uso de cmake para facilitar las tareas de compilación al usuario. Por lo tanto, solo es necesario usar el archivo de código fuente (previamente descomprimido en una carpeta) y seguir el mismo proceso de compilación mencionado a continuación:

```
| $ mkdir build
| $ cd build
| $ cmake ..
| $ make
```

5.3. Ejecución

Una vez calibrada nuestra cámara podremos ejecutar el proyecto de la siguiente forma:

```
| c930e|live <camera.yml> <markersize> <asset filename>
```

c930e|live: Especifica una cámara de entrada. Usar 'live' para capturar streaming.

<camera.yml>: Archivo de configuración de la cámara.

<markersize>: Tamaño del marcador en metros.

<asset filename>: Cadena de texto con el nombre de archivo.

Un ejemplo de ejecución en la consola de Ubuntu se muestra a continuación:

```
$ ./vilimbsAR live ../../highCalib.yml 0.076 brazohombre27.zip
$ ./vilimbsAR c930e ../../c930e_transpose_fhd.yml 0.076 brazohombre27.zip
```

Donde el primer argumento indica el uso del streaming de la cámara, el segundo la ruta donde se encuentra el archivo de calibración de dicha cámara, el tercero es la medida en metros del marcado impreso y el último es el nombre del archivo 3D (formato .zip). Luego de una correcta configuración se obtiene la pantalla en el motor gráfico como se muestra en la Figura 17.

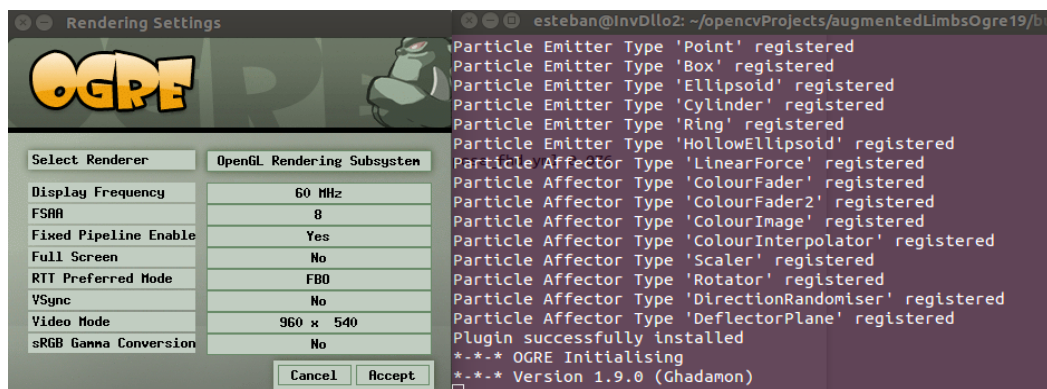


Figura 17. Selección de parámetros de Ogre.

Una vez escogidos los parámetros de resolución y recursos según la máquina de ejecución, se ejecuta ViLimbs como se ilustra en la Figura 18.

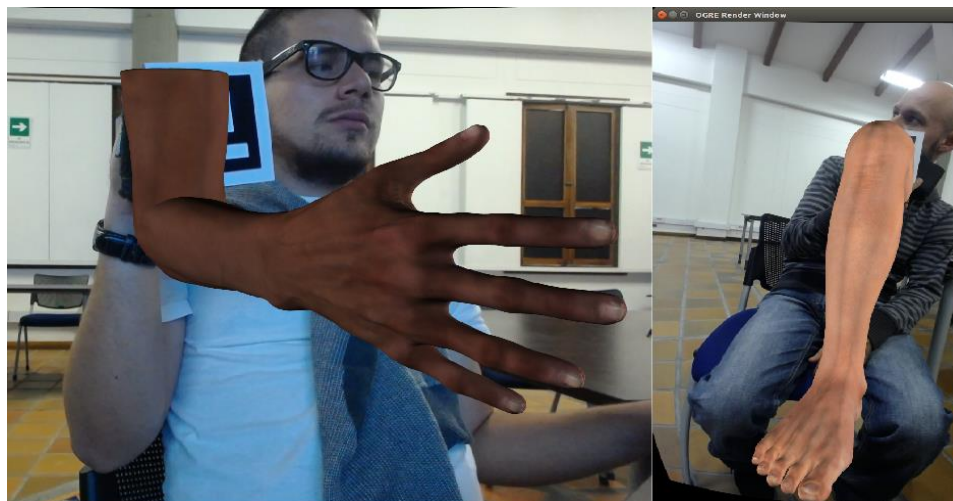


Figura 18. Ejemplo de ejecución de ViLimbs.

6. Anexos.

6.1. Configuración módulo 1: Control de movimiento.

6.1.1. Emotiv Xavier Control Panel

Verificar contacto de electrodos del Emotiv.

Luego de instalar y ejecutar el Control Panel del Emotiv, se configura la cuenta y el perfil adecuadamente para que se despliegue la ventana mostrada en la Figura 19. En ella se ilustran los 16 canales del Emotiv con el estado de conexión de cada uno de ellos. Aquellos canales que no están haciendo contacto adecuadamente deben lubricarse o moverse para que estén más cerca del cuero cabelludo para facilitar la adquisición de señales.

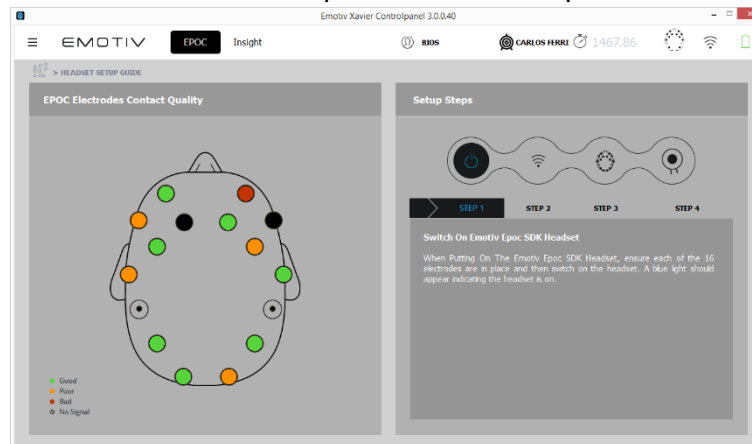


Figura 19. Verificación de contacto de electrodos en el Emotiv.

Entrenamiento de comandos mentales.

El cuerpo humano no es homogéneo en todas las personas porque cada una tiene diferentes manifestaciones físicas cuando se presentan a un mismo estímulo, el cerebro también funciona de esta forma. Por esta razón, se debe entrenar el Emotiv para identificar la respuesta cerebral característica de la persona que lo va a utilizar y así clasificar correctamente las señales como eventos. En la Figura 20 se ilustra el menú donde se puede entrenar el dispositivo.

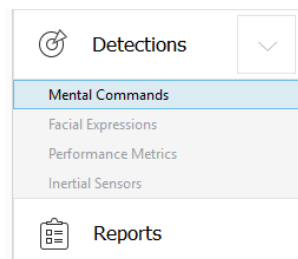


Figura 20. Menú para configurar comandos mentales.

La ventana de configuración de comandos es la existente en la Figura 21, en la parte izquierda se presenta un cubo de interacción para probar los eventos entrenados y en la

derecha se ubica un recuadro para configurar un evento específico a las señales entrenadas. El principal evento a utilizar para que el Control por BCI funcione apropiadamente es el “Push”.

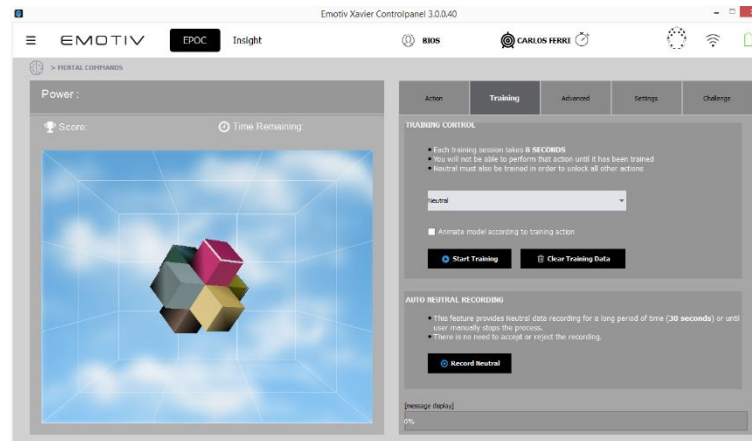


Figura 21. Ventana de configuración de comandos mentales.

El recuadro mostrado en la Figura 22 corresponde al entrenamiento de los eventos, la acción “Neutral” se debe entrenar cuando la persona esté parada, con bajas pulsaciones y en tranquilidad, la acción “Push” se debe entrenar con la persona parada y concentrada en un pensamiento específico. También es posible entrenarlo generando una tensión muscular desde el cuello para que el sensor detecte señales EMG en reemplazo a EEG, teniendo en cuenta que no es el uso adecuado del dispositivo pero es una solución para la etapa de entrenamiento. El entrenamiento de cada acción dura 8 segundos.

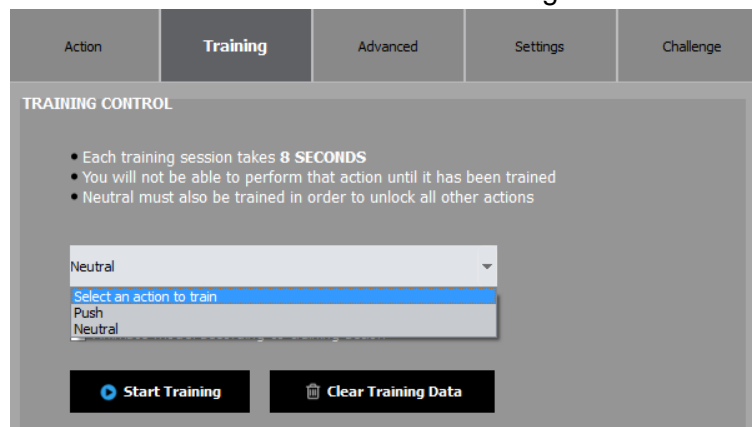


Figura 22. Entrenamiento de comandos.

6.1.2. Myo Connect

El Myo Connect se ejecuta en el centro de notificaciones como se muestra en la Figura 23, todas las opciones para configurar el dispositivo se encuentran en el menú desplegable. La opción más importante es el Armband Manager, donde se puede configurar la conexión y el estado de la batería.

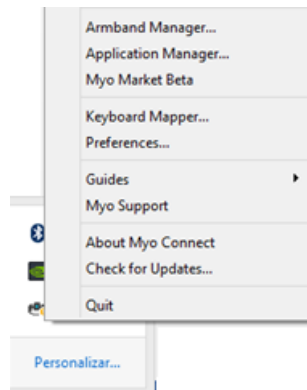


Figura 23. Ejecución de Myo Connect.

Configuración del Myo.

Al conectar el Myo con el cable USB, se enciende el dispositivo pero toca realizar el enlace de la conexión bluetooth. Como se muestra en la Figura 24, se debe conectar el receptor bluetooth y establecer la comunicación; también es posible probar la comunicación con un Ping y apagar el dispositivo en caso de que no se desee utilizar más.

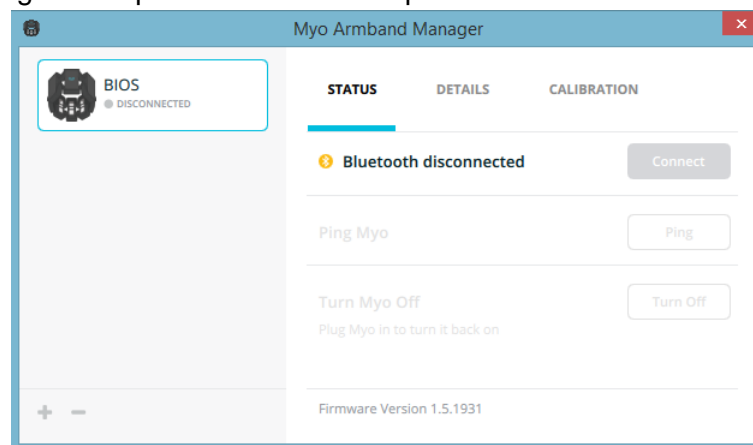


Figura 24. Configuración básica Myo.

Configuración de eventos.

En caso de que se deseen utilizar los eventos por defecto del Myo se debe configurar un nuevo perfil para entrenar el dispositivo, esto se realiza en la opción resaltada en azul en la Figura 25. Al crear el perfil se asocian los eventos del Myo al movimiento muscular del usuario.

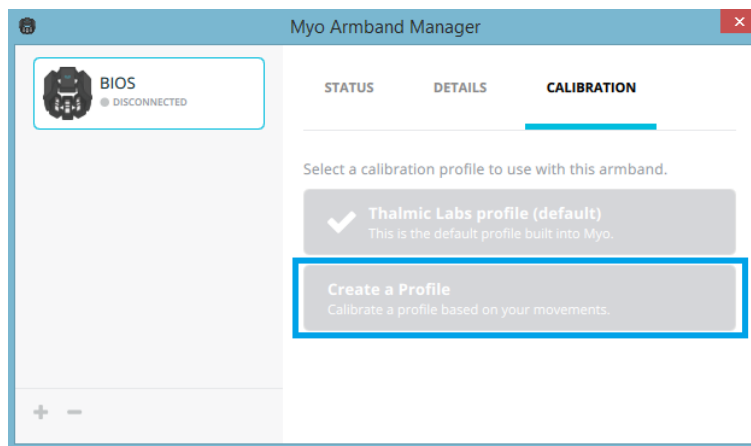


Figura 25. Creación de perfil del Myo.

El Myo Connect tiene cinco eventos simples que se pueden entrenar por defecto para realizar las acciones de control sobre las aplicaciones deseadas. Los posibles eventos entrenables del Myo son:

- WaveIn Doblar la muñeca hacia adentro.
- WaveOut Extender la muñeca hacia afuera.
- Fist Contraer fuertemente todos los dedos hacia el interior.
- Spread Fingers Extender todos los dedos lo máximo posible.
- Double Tap Tocar dos veces el dedo índice con el pulgar.

Para la aplicación de ViLimbs puede resultar inconveniente realizar la detección de estos cinco eventos debido a que la persona que le falta la extremidad no tiene los músculos que generan estas acciones. Por consiguiente, es necesario realizar un entrenamiento de eventos personalizado para el paciente y que no dependa del movimiento muscular del antebrazo. Este entrenamiento es posible desde la aplicación realizada.

6.2. Configuración módulo 2: Realidad Aumentada.

6.2.1. Problemas Frecuentes

Uno de los problemas reportados en OpenCV 2.4.11 es el enlazado de la librería *V4L2*, el cual no funciona adecuadamente en tiempo de compilación. Si al compilar, el lector se encuentra con este tipo de error o similar:

```
/usr/bin/ld: CMakeFiles/vilimbs.dir/main.cpp.o: undefined reference to
symbol 'v4l2_ioctl'
//usr/lib/x86_64-linux-gnu/libv4l2.so.0: error adding symbols: DSO
missing from command line
```

Se debe agregar en la última línea del archivo `~/src_vilimbs/build/CMakeFiles/vilimbs.dir/links.txt`, la palabra `-lv4l2`. Luego se retoma el proceso de compilación descrito.

6.3.Eventos y publicaciones

6.3.1. CHI 2015

ViLimbs

Improving Phantom Limb Treatment Through Multisensory Feedback

Esteban Correa-Agudelo, Andrés M. Hernández, Carlos Ferrín, and Juan D. Gómez

Center for Bioinformatics and Computational Biology, Manizales, Colombia





AUGMENTED-REALITY BASED APPROACH FOR IMPROVING MIRROR THERAPY

The system consists of three main modules:

- **Signal Processing** that collects and treats both neural and muscle activity.
- **Vision Module** that uses camera tracking and augmented reality to create the virtual limb.
- **Multisensory Feedback**, which uses image processing and/or clinical devices for measuring proprioceptive signals.

ABSTRACT

Each year, **anti-personnel mines** contribute to a vast number of amputated victims due to the current conflict in Colombia [1]. The recovery of such victims, includes psycho-motor therapies to reduce neuromuscular alternaths, such as **phantom limb pain (PLP)** [2]. Evidence suggests that is possible for PLP to lead patients to commit suicide due to the chronic pain[3].

ViLimbs is a **Virtual-Reality-based (VR)** system aimed at enhancing classical mirror therapy for amputees using an immersive video-wall that renders the patient himself with a superimposed virtual limb. The patient-virtual-limb interaction is achieved by means of **brain computer interfaces (BCI)** and **myoelectric signals** read from the remaining part of the limb. Thus, voluntary movements get naturally transferred from patients to virtual limbs. Last but not least, patients sense of ownership over an alien virtual limb is enhanced through **multisensory feedback (cardio-visual)**[4], which tends to lessen rehabilitation times.

DISCUSSION AND FUTURE WORK

This ongoing work is an initial step for getting conclusive results in the middle-term. Note that, unlike common therapies, this approach can be likely regarded as a **noninvasive method**. Currently, We are detecting by now as much as six movement intents with a high rate of accuracy in real time.

In terms of computer vision, **marker-less tracking** solutions need to be explored since the current marker-based approach has proved to be extremely limiting, for it does not allow freely stump motion. Moreover, **wireless clinical devices** are highly desired if more comfortable interaction is to be attempted.

Back to the clinical evidence, it is worth noticing that multisensory approaches tackled in this work are still lacking for prove. Last but not least, further work on **motor imagery signal processing** needs to be done provided that, at the current state of the work, our system relies 80% on myoelectric reading.

REFERENCES

- [1]. UNICEF: Sembrando Minas Cosechando Muertes: Colombia y las minas antipersonales. (E. de C. y U. C. Ministerio de Comunicaciones, Ed.).
- [2]. Flor, H. Phantom limb pain: characteristics, causes and treatment. *Lancet Neurol.* (2002), 182-189.
- [3]. Ortiz-Catalán, M. et al. Biopatrec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. *Source Code for Biology and Medicine* (2013), 11.
- [4]. Rognini, G. et al. Visuo-tactile integration and body ownership during self-generated action. *European Journal of Neuroscience* (2013), 1120-1129.

CONTACT

Esteban Correa-Agudelo: esteban.correa@bios.co
 Juan D. Gómez: juan.gomez@bios.co

ACKNOWLEDGEMENT

This work has been fully funded by the Caldas Government through the Colombian REGALÍAS project (No. 08112013-0621). We would also like to thank Clínica La Presentación and Neurocentro for supporting us with clinical advice and medical equipment.





Figura 26. Póster de presentación del proyecto en el evento internacional CHI 2015.

ViLimbs: Improving Phantom Limb Treatment Through Multisensory Feedback

Esteban Correa-Agudelo
Center for Bioinformatics and
Computational Biology,
Manizales, Colombia
esteban.correa@bios.co
Andrés M. Hernández
Center for Bioinformatics and
Computational Biology,
Manizales, Colombia
andres.hernandez@bios.co
Carlos Ferrin
Center for Bioinformatics and
Computational Biology,
Manizales, Colombia
carlos.ferrin@bios.co
Juan D. Gomez
Center for Bioinformatics and
Computational Biology,
Manizales, Colombia
juan.gomez@bios.co

Abstract

Each year, anti-personnel mines contribute to a vast number of amputated victims due to the current conflict in Colombia. The recovery of such victims, includes psycho-motor therapies to reduce neuromuscular aftermaths, such as phantom limb pain (PLP). Therefore, improving rehabilitation strategies can potentially have a large positive impact in the recovery outcomes for long-term treatments. ViLimbs is a Virtual-Reality-based (VR) system aimed at enhancing classical mirror therapy for amputees using an immersive video-wall that renders the patient himself with a superimposed virtual limb. The patient-virtual-limb interaction is achieved by means of brain computer interfaces (BCI) and myoelectric signals read from the remaining part of the limb. Thus, voluntary movements get naturally transferred from patients to virtual limbs. Last but not least, patients sense of ownership over an alien virtual limb is enhanced through multisensory feedback (cardio-visual), which tends to lessen rehabilitation times.

Author Keywords

Phantom limb pain; myoelectric signals; augmented reality; computer vision; neurorehabilitation; multisensory feedback;

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
CHI'15 Extended Abstracts, Apr 18-23, 2015, Seoul, Republic of Korea
ACM 978-1-4503-3146-3/15/04.
<http://dx.doi.org/10.1145/2702613.2732874>

Figura 27. Artículo científico presentado para el CHI 2015.

6.3.2. ANNA 2015



ViLimbs Augmenting Mirror Therapy for Phantom Limb Pain Treatment

Hacia la Reducción de Dolor de Miembro Fantasma a través de una Terapia de Espejo Aumentada y Bio-señales
+ Esteban Correa Agudelo/Carlos Ferrin/ Andrés M. Hernández/ Paula Vélez/Juan D. Gómez
+ CENTER FOR BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

ANNA 2015
I Congreso Internacional de Avances en
Neurología y Neurociencias Aplicadas

ViLimbs Bio-signal-based interaction
with virtual limbs

Introduction
ViLimbs is a Virtual-Reality-based (VR) system aimed at enhancing classical mirror therapy for amputees using an immersive video-wall that renders the patient himself with a superimposed virtual limb. The patient-virtual-limb interaction is achieved by means of brain computer interfaces (BCI) and myoelectric signals read from the remaining part of the limb. Thus, voluntary movements get naturally transferred from patients to virtual limbs. Last but not least, patients sense of ownership over an alien virtual limb is enhanced through multisensory feedback (cardio-visual), which tends to lessen rehabilitation times.

Methods
The system consists of three main modules:

- *Signal Processing that collects and treats both, neural and muscle activity,
- *Vision Module that uses camera tracking and augmented reality to create the virtual limb.
- *Multisensory Feedback, which uses image processing and/or clinical devices for measuring proprioceptive signals.

Discussion
This ongoing work is an initial step for getting conclusive results in the middle-term. Nevertheless, some points can be discussed for this first setup:

- *By now, as much as six movement intents are detected with real time accuracy in real time.
- *In terms of **computer vision**, marker-less tracking solutions need to be explored since the current marker-based approach has proved to be extremely limiting, for it does not allow freely stump motion.
- *Back to the **clinical evidence**, it is worth noticing that multisensory approaches tackled in this work are still lacking for prove.
- *Further work on **motor imagery and signal processing** needs to be done provided that, at the current state of the work, our system relies 80% on myoelectric reading.

Acknowledgements
This work has been fully funded by the Caldas Government through the Colombian REGALIAS project (No.08112013-0621). We would also like to thank Clínica La Presentación and Neurocentro for supporting us with clinical advice and medical equipment.

Bibliography
[1] Programa Presidencial para la Acción Integral Contra Minas Antipersonales. Informe de Víctimas de Minas Antipersonales. <http://www.accionintegral.gov.co/Paginas/Victimas.aspx>.
[2] Sembrando Minas cosechando Muertes: Colombia y las minas antipersonales. (E. de C. y U. C. Ministerio de Comunicaciones, Ed.). <http://www.unicef.org/co/Minas/pub.htm>.
[3] Flor, H. Phantom limb pain: characteristics, causes and treatment. *Lancet Neurol.* (2002), 162-165.

Contact
+ PhD. Juan D. Gomez: juan.gomez@bios.co
+ MSc. Esteban Correa Agudelo: esteban.correa@bios.co

Figura 28. Póster de presentación del proyecto en el I Congreso Internacional de Avances en Neurología y Neurociencias Aplicadas.