**MANUAL DE USUARIO**

El siguiente archivo es un manual de usuario detallado para el sistema de reconocimiento de gestos.

**ÍNDICE**

**DESCRIPCIÓN DEL SISTEMA**

El sistema permite:

* la adquisición y almacenamiento de señales EMG (*training*),
* Visualización de señales en tiempo real (*plot EMG*),
* el reconocimiento en tiempo de real de gestos (*recognition*) basado en las señales adquiridas en el entrenamiento,
* testeo del sistema al variar parámetros (*testing*).

ETAPAS

*Training*

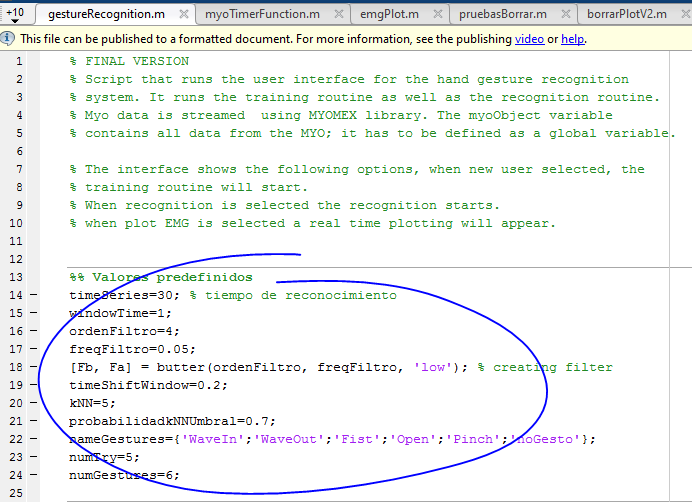
Esta etapa se una rutina de entrenamiento necesaria para cada nuevo usuario. Para ejecutarla desde la interfaz, se debe presionar New User. Se requiere al usuario que ingrese 2 valores, el tiempo a grabar de cada gesto, y el número de repeticiones para cada gesto. Las señales conseguidas son almacenadas por usuario y por gesto como archivos .mat en la carpeta usersData. Los valores por defecto son 2 segundos y 5 repeticiones por gesto. Las funciones involucradas son: training, recordGesture.

*Recognition*

Esta etapa consiste en la clasificación de una señal desconocida. La clasificación se basa en la construcción de una base de datos por usuario. Para el reconocimiento se realiza la lectura y clasificación de gestos en tiempo real por un tiempo definido. Varios parámetros son necesarios para ello: (timeSeries,database,windowTime,Fb,Fa,numTry,numGestures,timeShiftWindow,kNN,probabilidadkNNUmbral,nameGestures).

Desde la interfaz de usuario se debe seleccionar Recognition, en este caso al usuario solo se le pide ingresar el nombre de usuario. Con el nombre de usuario, el sistema busca las señales EMG de cada gesto para construir la base de datos.

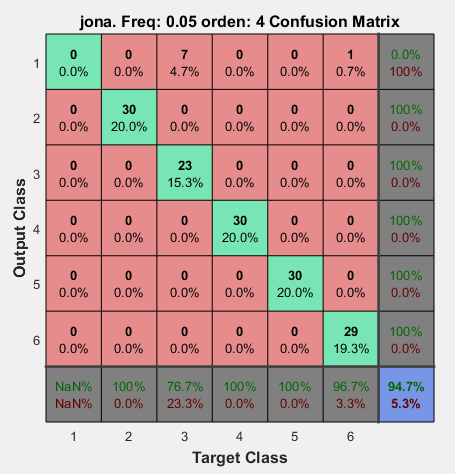
El usuario no necesita settear todos los parámetros, estos pueden ser modificados fácilmente pues se encuentran en la parte superior del script "gestureRecognition".



El gesto resultante de la clasificación puede ser accedido como variable global: "gesto".

*Testing*

Para medir la tasa de clasificación del sistema se pidió a 10 usuarios que realizaran 30 repeticiones de cada gesto, cada repetición en 5 segundos. Se realiza la clasificación de cada uno de estos gestos (comprobacionTest). Se obtienen matrices de confusión por cada usuario y también un histograma con todos los tiempos de procesamiento. Desde la interfaz de usuario se puede acceder únicamente a la rutina de adquisición de las señales. Para realizar la clasificación se debe ejecutar comprobacionTest.



**REQUISITOS PREVIOS DEL SISTEMA**

* Matlab (desarrollado en R2015a),
* librería Windows SDK instalada e incluida como variable de entorno,
* Myo Armband conectado a través de MYO Connect.

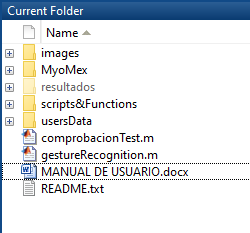
Myo Connect y Windows SDK disponibles en:

<https://developer.thalmic.com/downloads>

* Librería Myo Mex disponible en:

<https://github.com/mark-toma/MyoMex>

**DESCRIPCIÓN DE CARPETAS**



En la raíz de la carpeta se encuentran dos scripts:

* *gestureRecognition* es la interfaz de usuario. Esta interfaz únicamente accede a varias funciones.
* *comprobaciónTotal* es el script con el que se evalúa las pruebas realizadas. Está basado en señales para testeo (testing). Permite modificar los parámetros del clasificador.

Las carpetas contenidas son:

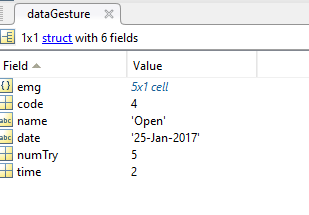
* *images*, contiene las imágenes para ser mostradas en la rutina de entrenamiento. Estás imágenes son cargadas en base al nombre y extensión del archivo (para modificaciones, sobrescribir los archivos).
* *MyoMex* es la librería utilizada para la lectura de datos del MYO ARMBAND.
* *resultados* en esta carpeta se grabaron las matrices de confusión total y por usuarios.
* *scripts&Functions* es una carpeta que incluye todas las funciones y scripts .m.
* *usersData* es la carpeta en la que se almacenan las señales EMG como archivos .mat.

**DESCRIPCIÓN DE FUNCIONES**

Todas las funciones están incluidas en la carpeta scripts&Functions, y tienen una descripción. Estas son las funciones utilizadas:

* *connectMyo*: esta función se utiliza para conectar Myo. No requiere valores de entrada.
* *databaseConstruction*: esta función se utiliza para contruir una celda Mx8 que incluye las señales EMG filtradas por canal. Se requiere ingresar el nombre de usuario con el que se buscarán los archivos del entrenamiento, y los valores del filtro.
* *emgPlot*: con esta función se plotea en tiempo real la señal EMG. Debe estar conectado MYO.
* *myoTimerFunction*: esta función es llamada en cada ejecución del timer. Únicamente lee los valores de emg\_log. Los valores son devueltos como variables globales.
* *recognitionPruebasPaper*: este script fue creado para las pruebas del sistema. Aplica el algoritmo de clasificación para todos los usuarios sobre una señal almacenada en testing. Es una variante de recognitionScript.
* *recognitionScript*: es una función que realiza la adquisición de datos y la clasificación en tiempo real. El gesto resultante puede ser accesado como variable global.
* *recordGesture*: Esta función es parte de la función training, esta función se encarga de almacenar una sola repetición de un solo gesto. La señal adquirida se graba como archivo .mat en la carpeta usersData. Es necesario ingresar el nombre del usuario, el tiempo de la repetición, y el nombre del gesto.
* *terminateMyo:* función para desconectar el Myo. No requiere entradas.
* *training*: esta función se encarga de toda la rutina de entrenamiento. Para ello, llama varias veces a la función record gesture. Es necesario indicar el número de repeticiones por gesto y el nombre del ususario.

**ESTRUCTURA DE DATOS**



* Todas las señales EMG almacenadas en la carpeta usersData están grabadas como archivos .mat por nombre de usuario y por gesto con formato de estructura y llevan el nombre *dataGesture*.
* En el campo dataGesture.emg están almacenadas, como celdas, las señales EMG de cada repetición.
* Los demás campos de dataGesture son (nombre de usuario, número de repeticiones, código del gesto, fecha de la grabación).
* También se almacenan archivos de la base de datos de cada usuario.
* Por último, cada señal de la secuencia de reconocimiento es almacenada. Tiene el mismo formato que *dataGesture*.
* El archivo resultadosTodos.mat contiene 3 variables. Estás variables son los resultados de las pruebas de testeo. Las variables valoresRealesVector y resultadosVector están en el formato para plotear la matriz de confusión total. tClassificationVector es un vector con todos los tiempos de procesamiento.

**CONSIDERACIONES ADICIONALES**

* Todos los nombres de variables y archivos utilizan la convención lower Camel Case.
* Las variables que llevan el sufijo \_*Vector* son variables para grabar resultados que se ejecutan en un lazo *(for, while)*.
* Todas las lecturas de las señales EMG están cronometradas con un timer. El timer requiere ejecutar una función (*myoTimerFunction*), esta función se llama en cada ejecución del timer. Para acceder a los datos de esta función se utilizan variables globales.
* Para el funcionamiento en tiempo real se utiliza lazos parfor para el cálculo del DTW. Debido a esto, es necesario esperar unos minutos la primera vez que se ejecuta el programa.