

```

%-----
clc
clear
close all
%-----
% PRESENTACIÓN
disp('RECONOCIMIENTO DE GESTOS DE MANO UTILIZANDO SEÑALES EMG')
disp('Bienvenido al sistema de reconocimiento de gestos de la mano con
Myo Armband,')
disp('el sistema permite el reconocimiento de cinco gestos de
la mano.')
disp('- FIST')
disp('- WAVE IN')
disp('- WAVE OUT')
disp('- OPEN')
disp('- PINCH')
disp('')
disp('RECOMENDACIONES')
disp('* Colocarse el brazalete a 6 [cm] aproximadamente desde CODO al MYO
ARMBAND.')
disp('* Si es necesario, se debe realizar un ajuste del brazalete')
disp('INICIO')
disp('Para empezar, se requiere de una configuración rápida de
los gestos,')
disp('para ello se debe seleccionar el método de clasificación de
su preferencia')
disp('POR FAVOR, ELIJA UN MÉTODO DE CLASIFICACIÓN:')
n1 = input(' [1] DISTANCIA EUCLIDIANA (CENTROIDE) \n [2] REDES NEURONALES
ARTIFICIALES (MLP) \n:');
%-----
% IMAGENES DE LOS GESTOS
imagen1=imread('NO_GESTURE.jpg');
imagen2=imread('FIST.jpg');
imagen3=imread('OPEN.jpg');
imagen4=imread('WAVE_IN.jpg');
imagen5=imread('WAVE_OUT.jpg');
imagen6=imread('PINCH.jpg');
%-----
% CONSTANTES NECESARIAS
s=0;
ss=0;
abcd=0;
ventana2=2; % 400 muestras
it=0;
g=1;
Q=1;
i=1;
k=0;
fin=0;
as=1;
terminar=0;

FI=0;
PI=0;
WI=0;
WO=0;
OP=0;

```

```
FI1=0;  
FI2=0;  
FI3=0;  
FI4=0;  
FI5=0;  
FI6=0;  
FI7=0;  
FI8=0;
```

```
WI1=0;  
WI2=0;  
WI3=0;  
WI4=0;  
WI5=0;  
WI6=0;  
WI7=0;  
WI8=0;
```

```
WO1=0;  
WO2=0;  
WO3=0;  
WO4=0;  
WO5=0;  
WO6=0;  
WO7=0;  
WO8=0;
```

```
OP1=0;  
OP2=0;  
OP3=0;  
OP4=0;  
OP5=0;  
OP6=0;  
OP7=0;  
OP8=0;
```

```
PI1=0;  
PI2=0;  
PI3=0;  
PI4=0;  
PI5=0;  
PI6=0;  
PI7=0;  
PI8=0;
```

```
u1=0.5;  
u2=0.5;  
u3=0.5;  
u4=0.5;  
u5=0.5;  
u6=0.5;  
u7=0.5;  
u8=0.5;
```

```
%-----  
--
```

```

% VERIFICACIÓN DE VALORES INGRESADOS
control_1 = ~isempty(n1);
assert(control_1, 'No ha ingresado ningún valor');
%-----
% CONTROL DE SELECCIÓN
switch n1
    case 1 % Sistema utilizando el método del centroide
        %-----
        % CONSTANTES PARA LA ADQUISIÓN DE SEÑALES (MATLAB - MYO ARMBAND)
        countMyos=1;
        m = MyoMex(countMyos); % propiedades de la instancia MyoMex
        m1 = m.myoData(1); % objetos para cada uno de los Myos
        m1.timeEMG;
        m1.emg;
        m1.stopStreaming();
        m.myoData.clearLogs();
        m1.startStreaming();
        emg=zeros(40,8);
        order=1;
        runMYO=1;
        imshow(imagen1); % instrucción para mostrar los gestos en una
figura
%-----
        %INICIO DE BUCLE INFINITO
        while runMYO==1
%-----
            timeEMG = m1.timeEMG_log;
            if ~isempty(timeEMG) % validación de datos adquiridos
                T_emg=timeEMG(:,1)>=(timeEMG(end,1)-ventana2);
                emg=m1.emg_log(T_emg,WM_X(order));
            end
            assignin('base','SIGNALS',emg(:,:)) % asignación de la matriz
de señales EMG a la variable SIGNALS
                                % para el reconocimiento de gestos.
%-----
            % SEPARACION DE CANALES
            [c1,c2,c3,c4,c5,c6,c7,c8] = separacion(SIGNALS,i); % función
para separar cada señal EMG de la matriz SIGNAL
                                % para su
procesamiento de manera individual.
%-----
            % VERIFICACIÓN DE GESTO REALIZADO
            [it] =
umbral(c1,c2,c3,c4,c5,c6,c7,c8,i,it,Q,u1,u2,u3,u4,u5,u6,u7,u8); % función
para verificar si
% un usuario ha realizado algún gesto
%-----
            % GESTO
            if (it==1 && ss==0) % condicional para determinar el nivel
de ruido de cada canal para aplicar en la función umbral

[u1,u2,u3,u4,u5,u6,u7,u8]=ruido_adaptativo(c1,c2,c3,c4,c5,c6,c7,c8,i); %
función para determinar el nivel de ruido

% de cada canal, para una detección de los

```

```

% gestos con mayor precisión en la función umbral
    ss=2;
end
if (it==4 && s==0) % condicional para determinar el gesto
realizado
%-----
    % VENTANA DE ANÁLISIS DE SEÑAL
    tic

[gp1, gp2, gp3, gp4, gp5, gp6, gp7, gp8]=ventana_analisis(c1,c2,c3,c4,c5,c6,c7,c
8,i); % función para extraer la ventana que

% contiene la señal EMG para analizar
%-----
    % RECTIFICACIÓN

[gt1, gt2, gt3, gt4, gt5, gt6, gt7, gt8]=rectificacion(gp1, gp2, gp3, gp4, gp5, gp6, g
p7, gp8); % función para realizar

% rectificación de cada señal EMG
%-----
    % ENVOLVENTE TEMPORAL

[ba1, ba2, ba3, ba4, ba5, ba6, ba7, ba8]=envolvente(gt1, gt2, gt3, gt4, gt5, gt6, gt7,
gt8); % función para obtener la envolvente

% superior de las señales EMG
%-----
    % SUAVIZADO DE CURVAS

[Am1, Am2, Am3, Am4, Am5, Am6, Am7, Am8]=suavizado_curvas(ba1, ba2, ba3, ba4, ba5, ba
6, ba7, ba8); % función para realizar filtrado

% de las envolventes de señales EMG
%-----
    % AREA BAJO LA CURVA
    [t1, t2, t3, t4, t5, t6, t7, t8] =
area(Am1, Am2, Am3, Am4, Am5, Am6, Am7, Am8); % función para obtener el area bajo
la envolvente

% filtrada de las señales EMG
%-----
    % MÉTODO DEL CENTROIDE
    [LONGITUD] = centroide(t1, t2, t3, t4, t5, t6, t7, t8, ...
% función para determinar la distancia euclidiana en 8 dimensiones

FI1, FI2, FI3, FI4, FI5, FI6, FI7, FI8, ...

WI1, WI2, WI3, WI4, WI5, WI6, WI7, WI8, ...

WO1, WO2, WO3, WO4, WO5, WO6, WO7, WO8, ...

OP1, OP2, OP3, OP4, OP5, OP6, OP7, OP8, ...
    PI1, PI2, PI3, PI4, PI5, PI6, PI7, PI8);
%-----
    % CÁLCULO DE CENTROIDES DE CADA GESTO

```

```

        k=k+1;
        if k<=15 % condicional para cálculo de centroides.
Funciona unicamente en la configuración rápida
            [FI1,FI2,FI3,FI4,FI5,FI6,FI7,FI8,...
             WI1,WI2,WI3,WI4,WI5,WI6,WI7,WI8,...
             WO1,WO2,WO3,WO4,WO5,WO6,WO7,WO8,...
             OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,...

PI1,PI2,PI3,PI4,PI5,PI6,PI7,PI8,Q,inputs]=calculo_centroide_o_inputs(t1,t
2,t3,t4,t5,t6,t7,t8,k,g); % función para cácular los centroides

% en la fase de configuración rápida
end
if k==15
    title('EMPECEMOS')
    pause(2)
    title('          ')
    pause(2);
end
g=g+1;

%-----
% DETERMINACION DE GESTO
if k>=16
    toc
    [FI,WI,WO,OP,PI,abcd] =
determinacion_gesto(LONGITUD); % función para mostrar el gesto que ha
realizado el usuario
end
s=1;
end

%-----
% VERIFICACION DEL GESTO SOSTENIDO
[D1,D2,D3,D4,D5,D6,D7,D8,terminar,as] =
sostenido(c1,c2,c3,c4,c5,c6,c7,c8,i,s,...

abcd,fin,Q,FI,WI,WO,OP,PI); % función para verificar si un usuario sigue
realizando un gesto

% e imprimir el gesto que mantenido.

% Si se mantiene el gesto WAVE OUT 5 segundos,

% termina el programa
fin=fin+1;

%-----
% TERMINACIÓN DE PROGRAMA
if terminar ==1
    break
end

%-----
% LIMPIAR VARIABLES -> % programa para limpiar variables
necesarias
if as==1
    it=0;
    i=0;
    s=0;
    abcd=0;

```

```

        terminar=0;
        fin=0;
    end
%-----
    %PAUSAS -> Pausas necesarias para cada iteración
    i=i+1;
    if k>=15
        pause(0.14)
    end
    if k<13
        pause(0.12)
    end
    if k==13 || k==14
        pause(0.08)
    end
end
m1.stopStreaming(); %Stop streaming Myo_1 all data
m.myoData.clearLogs();
m.delete; %Delete variable for Myo_1
clear m
#####
#####
case 2
%-----
    %CONSTANTES PARA LA ADQUISIÓN DE SEÑALES (MATLAB - MYO ARMBAND)
    imshow(imagen1);
    countMyos=1;
    m = MyoMex(countMyos);
    m1 = m.myoData(1);
    m1.timeEMG;
    m1.emg;
    m1.stopStreaming();
    m.myoData.clearLogs();
    m1.startStreaming();
    emg=zeros(40,8);
    order=1;
    runMYO=1;
    wavein= [1;0;0;0;0;0];
    waveout= [0;1;0;0;0;0];
    fist= [0;0;1;0;0;0];
    open= [0;0;0;1;0;0];
    pinch= [0;0;0;0;1;0];

    targets=[repmat(wavein,1,3),repmat(waveout,1,3),repmat(fist,1,3),repmat(o
pen,1,3),repmat(pinch,1,3)];
%-----
    %INICIO DE BUCLE INFINITO
    while runMYO==1
%-----
        timeEMG = m1.timeEMG_log;
        if ~isempty(timeEMG) % validación de datos adquiridos
            T_emg=timeEMG(:,1)>=(timeEMG(end,1)-ventana2);
            emg=m1.emg_log(T_emg,WM_X(order));
        end
        assignin('base','SIGNALS',emg(:,:)) % asignación de la matriz
% de señales EMG a la variable SIGNALS
% para el reconocimiento de gestos.

```

```

%-----
% SEPARACION DE CANALES
[c1,c2,c3,c4,c5,c6,c7,c8] = separacion(SIGNALS,i); % función
% para separar cada señal EMG de la matriz SIGNAL
% para su procesamiento de manera individual.
%-----
% VERIFICACIÓN DE GESTO REALIZADO
[it] =
umbral(c1,c2,c3,c4,c5,c6,c7,c8,i,it,Q,u1,u2,u3,u4,u5,u6,u7,u8); % función
% para verificar si

% un usuario ha realizado algun gesto
%-----
% GESTO
    if (it==1 && ss==0) % condicional para determinar el nivel %
de ruido de cada canal para aplicar en la función umbral

[u1,u2,u3,u4,u5,u6,u7,u8]=ruido_adaptativo(c1,c2,c3,c4,c5,c6,c7,c8,i); %
% función para determinar el nivel de ruido

% de cada canal, para una detección de los

% gestos con mayor precisión en la función umbral
    ss=2;
end
    if (it==4 && s==0) % condicional para determinar el gesto %
realizado
%-----
tic
% VENTANA DE ANÁLISIS DE SEÑAL

[gp1,gp2,gp3,gp4,gp5,gp6,gp7,gp8]=ventana_analisis(c1,c2,c3,c4,c5,c6,c7,c
8,i); % función para extraer la ventana que

% contiene la señal EMG para analizar
%-----
% RECTIFICACIÓN

[gt1,gt2,gt3,gt4,gt5,gt6,gt7,gt8]=rectificacion(gp1,gp2,gp3,gp4,gp5,gp6,g
p7,gp8); % función para realizar

% rectificación de cada señal EMG
%-----
% ENVOLVENTE TEMPORAL

[ba1,ba2,ba3,ba4,ba5,ba6,ba7,ba8]=envolvente(gt1,gt2,gt3,gt4,gt5,gt6,gt7,
gt8); % función para obtener la envolvente

% superior de las señales EMG
%-----
% SUAVIZADO DE CURVAS

[Am1,Am2,Am3,Am4,Am5,Am6,Am7,Am8]=suavizado_curvas(ba1,ba2,ba3,ba4,ba5,ba
6,ba7,ba8); % función para realizar filtrado

% de las envolventes de señales EMG
%-----

```

```

        % AREA BAJO LA CURVA
        [t1,t2,t3,t4,t5,t6,t7,t8] =
area (Am1,Am2,Am3,Am4,Am5,Am6,Am7,Am8); % función para obtener el area bajo
% la envolvente

% filtrada de las señales EMG
%-----
        % MÉTODO DEL CENTROIDE
        [LONGITUD] = centroide(t1,t2,t3,t4,t5,t6,t7,t8,...
% función para determinar la distancia euclidiana en 8 dimensiones

FI1,FI2,FI3,FI4,FI5,FI6,FI7,FI8,...

WI1,WI2,WI3,WI4,WI5,WI6,WI7,WI8,...

WO1,WO2,WO3,WO4,WO5,WO6,WO7,WO8,...

OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,...
        PI1,PI2,PI3,PI4,PI5,PI6,PI7,PI8);
%-----
        % CÁLCULO DE CENTROIDES DE CADA GESTO
        k=k+1;
        if k<=15 % condicional para cálculo de centroides.
%Funciona unicamente en la configuración rápida
        [FI1,FI2,FI3,FI4,FI5,FI6,FI7,FI8,...
        WI1,WI2,WI3,WI4,WI5,WI6,WI7,WI8,...
        WO1,WO2,WO3,WO4,WO5,WO6,WO7,WO8,...
        OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,...

PI1,PI2,PI3,PI4,PI5,PI6,PI7,PI8,Q,inputs]=calculo_centroide_o_inputs(t1,t
2,t3,t4,t5,t6,t7,t8,k,g); % función para calcular los centroides

% en la fase de configuración rápida
        end
        g=g+1;
%-----
        % CREACIÓN DE LA RED NEURONAL
        if k==15
            Q=0;
[redes_neuronales]=redes_neuronales(inputs',targets);
            %Test the Network
            imshow(imagen1);
            title('ESPERE...')
            pause(3)
            title('LISTO')
            pause(1)
            title('EMPECEMOS')
            pause(1)
            title(' ')
        end
        if k >=16
            gesto_realizado =[t1;t2;t3;t4;t5;t6;t7;t8];
%-----
        % DETERMINACION DE GESTO
            y_5=redes_neuronales(gesto_realizado);
            [M,LONGITUD]=max(y_5);
            toc

```



```

                                [FI,WI,WO,OP,PI,abcd] =
determinacion_gesto(LONGITUD); % función para mostrar el gesto realizado

                                end
                                s=1;
%-----
                                end
%-----
                                % VERIFICACION DEL GESTO SOSTENIDO
                                [D1,D2,D3,D4,D5,D6,D7,D8,terminar,as] =
sostenido(c1,c2,c3,c4,c5,c6,c7,c8,i,s,...

abcd,fin,Q,FI,WI,WO,OP,PI); % función para verificar si un usuario sigue
% realizando un gesto
% e imprimir el gesto que mantenido.

% Si se mantiene el gesto WAVE OUT 5 segundos,

% termina el programa
                                fin=fin+1;
%-----
                                % TERMINACIÓN DE PROGRAMA -> programa para terminar el
%programa
                                if terminar ==1
                                    break
                                end
%-----
                                % LIMPIAR VARIABLES -> programa para limpiar variables
%necesarias
                                if as==1
                                    it=0;
                                    i=0;
                                    s=0;
                                    abcd=0;
                                    terminar=0;
                                    fin=0;
                                end
%-----
                                % PAUSAS -> Pausas necesarias para cada iteración
                                i=i+1;
                                if k>=15
                                    pause(0.14)
                                end
                                if k<13
                                    pause(0.12)
                                end
                                if k==13 || k==14
                                    pause(0.08)
                                end
                                end
                                m1.stopStreaming(); %Stop streaming Myo_1 all data
                                m.myoData.clearLogs();
                                m.delete; %Delete variable for Myo_1
                                clear m
                                otherwise
                                    disp('OPCIÓN INCORRECTA')
                                end
end

```