
Herramientas para captura de señales bioeléctricas

Juan Casado Ballesteros
Panagiotis Georgios Pennas

UAH - 24 February 2020



Universidad
de Alcalá

BITalino

BITalino es una plataforma de hardware y software abierta que ofrece herramientas para capturar señales bioeléctricas. Sus dispositivos hardware se les pueden comprar a ellos o construirlos uno mismo en base a los esquemas públicos donde se especifican los componentes utilizados y cómo conectarlos.

Respecto del software que proporcionan cabe destacar la aplicación de escritorio y la aplicación Android con las cuales es posible conectarse a la placa BITalino, capturar las medidas que esta realice y exportarlas en un formato común para ambas. Existen también una gran cantidad de APIs disponibles para múltiples plataformas de modo que comenzar a utilizar BITalino en cualquiera de ellas se convierte en una tarea bastante sencilla.

En nuestro caso utilizaremos la placa BITalino para realizar ECGs conectándonos los electrodos del mismo modo que hicimos en la primera práctica. Dichas mediciones serán capturadas en el canal 3 desde el punto de vista del software de BITalino. Extraeremos las mediciones utilizando la aplicación de escritorio OpenSignals, las aplicaciones móviles Bitadroid y OpenSignals Mobile y la toolbox de Matlab BITalino Toolbox. Todas estas aplicaciones se conectan a la placa BITalino utilizando Bluetooth; de modo que deberemos configurarlas con la mac del dispositivo, la contraseña y el canal del que queremos leer los datos.

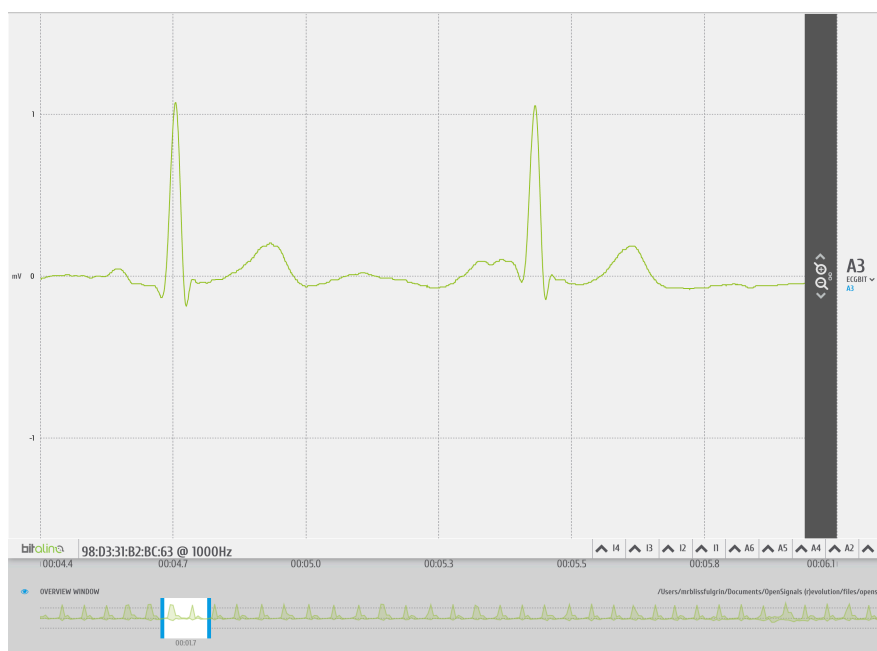
Mediciones realizadas

Expondremos a continuación cómo hemos realizado las mediciones con cada una de las aplicaciones y APIs que hemos mencionado anteriormente. Comentaremos también las diferencias observadas entre cada aplicación respecto a la experiencia de uso y la funcionalidad que cada una de ellas proporciona.

Open Signals

Conectarnos con Open Signals a la placa BITalino fue sencillo. Primero emparejamos la placa y el ordenador mediante Bluetooth y posteriormente solo fue necesario indicar la mac del dispositivo en la aplicación.

La aplicación nos permite visualizar múltiples canales de forma simultánea así como realizar mediciones utilizando varios de ellos a la vez. En la captura mostramos una medición que realizamos en la que hemos ampliado el canal 3 y mostramos una honda completa de pico a pico del complejo QRS.

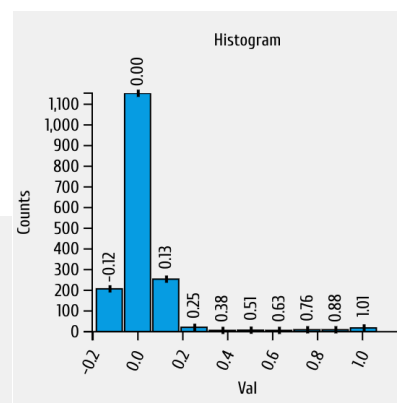


La aplicación open Signals nos permite también crear estadísticos de la medición que estamos visualizando. No obstante estos estadísticos se realizan sobre la amplitud de la onda lo cual creemos que nos es de gran utilidad debido a que esta depende en gran medida de cómo realizamos las mediciones y del amplificador utilizado.

En la primera foto debemos solo fijarnos en el canal 3, el resto de canales no están conectados a nada luego los estadísticos no tienen valor.

Como podemos ver en el histograma y en la media de las amplitudes esta es cercana a 0, es decir los picos positivos y negativos se compensan en su mayor parte.

	Min	Max	Mean	Median	STD	↑↓	RMS	RSS
	98:D3:31:B2:BC:63							
1	71.00	76.00	74.70	76.00	1.75	5.00	74.72	3071.61
2	70.00	78.00	71.85	71.00	1.65	8.00	71.87	2954.64
3	-0.18	1.07	0.02	-0.02	0.16	1.26	0.16	6.69
4	300.00	318.00	309.71	312.00	3.38	18.00	309.73	12732.94
5	12.00	12.00	12.00	12.00	0.00	0.00	12.00	493.32
6	35.00	36.00	35.90	36.00	0.31	1.00	35.90	1475.72



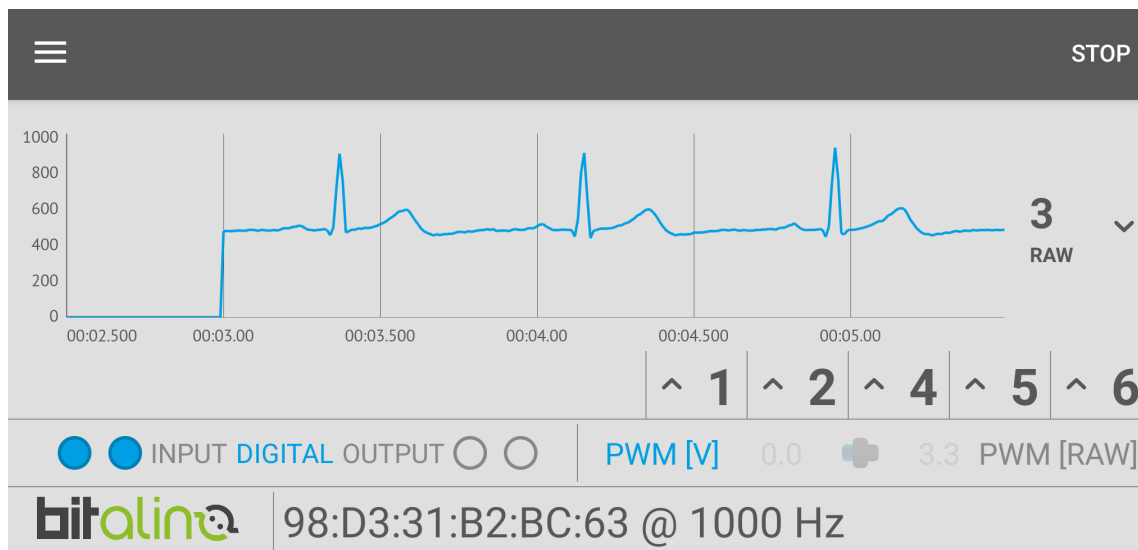
Cabe destacar la calidad de las mediciones realizadas, la cual puede apreciarse especialmente en el complejo QRS. Cuando realizamos las mediciones en la primera práctica las ondas Q y S (negativas al inicio y final del complejo) apenas se percibían. No obstante en las medidas realizadas con la placa BITalino pueden verse sin problemas.

Open Signals Mobile

Utilizaremos la aplicación Android de Open Signals para realizar otra medición del ECG. Al igual que con la aplicación anterior primero deberemos emparejar la placa BITalino al dispositivo por Bluetooth y posteriormente seleccionarlo en la aplicación.

Por desgracia la aplicación para Android no ofrece tanta funcionalidad como la de escritorio, es más difícil guardar y exportar los archivos en los que se almacenan los datos de las mediciones y no tenemos la posibilidad de calcular estadísticos.

Adicionalmente la captura de datos se interrumpía al poco de iniciarla. No obstante al final fuimos capaces de realizarla, visualizarla en la aplicación y exportar los datos de la medición al ordenador.

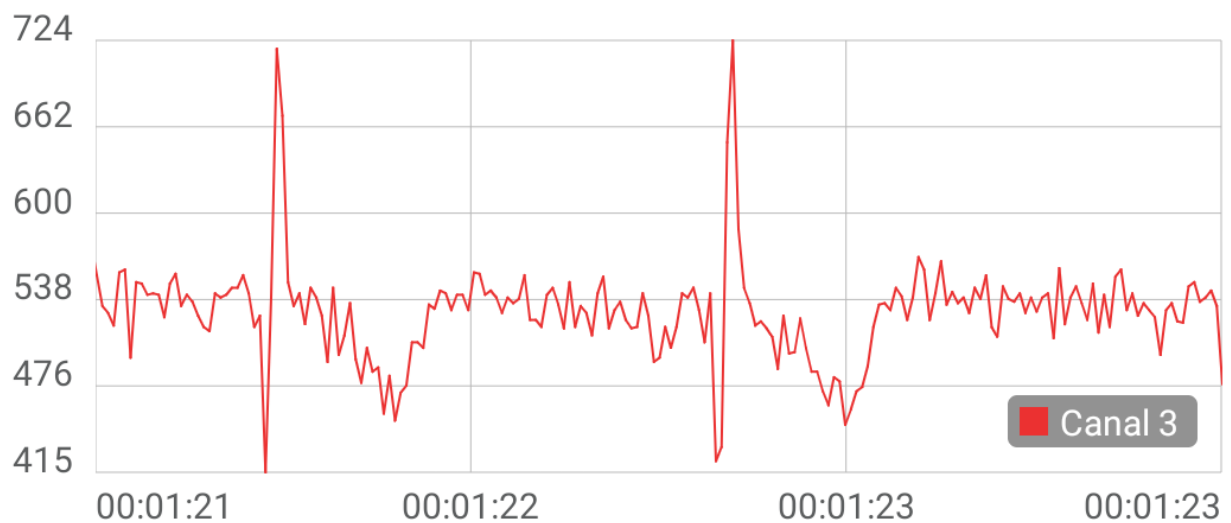


BitAlAndroid

Esta aplicación nos ha parecido más fácil y cómoda de utilizar que la anterior. Con ella hemos realizado otra captura de un ECG y exportado los datos al ordenador. A pesar de no poder realizar acciones adicionales desde la aplicación a parte de capturar y visualizar los datos sí podemos realizar pequeñas configuraciones a la hora de realizar la captura como modificar la frecuencia a la que se toman datos nuevos o el número de bits de precisión de cada medida.

Una desventaja de esta aplicación frente a la anterior es que una vez capturado el ECG no puede visualizarse dentro de la aplicación. Solo mientras se captura.

Mostramos a continuación un ECG capturado desde BitAlAndroid.

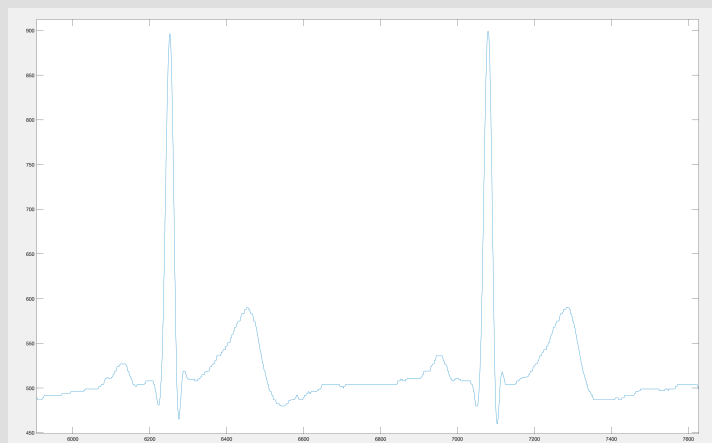


98:D3:31:B2:BC:6F

Matlab

También podemos utilizar la API para Matlab de BITalino. Solo tenemos que instalar la toolbox correspondiente la cual nos ofrecerá una interfaz sencilla desde la que poder conectarnos al dispositivo y capturar los datos que esté midiendo.

```
%% Capturar datos
% Crear el objeto Bitalino proporcionado por la Toolbox
b = Bitalino;
% Comenzar a captura datos
b.startBackground;
% Esperar 20 segundos
disp('Record stating')
pause(20);
disp('Record ended')
% Recuperar los datos del objeto
data = b.read;
% Finalizar la captura de datos
b.stopBackground
% Eliminar el objeto
delete(b)
```



Procesar un ECG

A continuación tomaremos uno de los ECG capturados utilizando la placa BITalino y una de las múltiples aplicaciones con las que hemos capturado datos. En concreto utilizaremos los datos capturados con la aplicación Open Signals para escritorio.

Para poder cargar los datos tan solo tendremos que abrir uno de los archivos .txt capturados y eliminar la cabecera en formato JSON que se encuentra al inicio del archivo. Hemos descubierto que todas las aplicaciones que capturan datos de BITalino utilizan el mismo formato de cabecera en el que describen la cantidad de canales de los que se tienen medidas, el dispositivo con el que fueron capturadas y la aplicación que las guardó.

Tras haber eliminado dicha cabecera podremos leer y mostrar los datos de la siguiente forma.

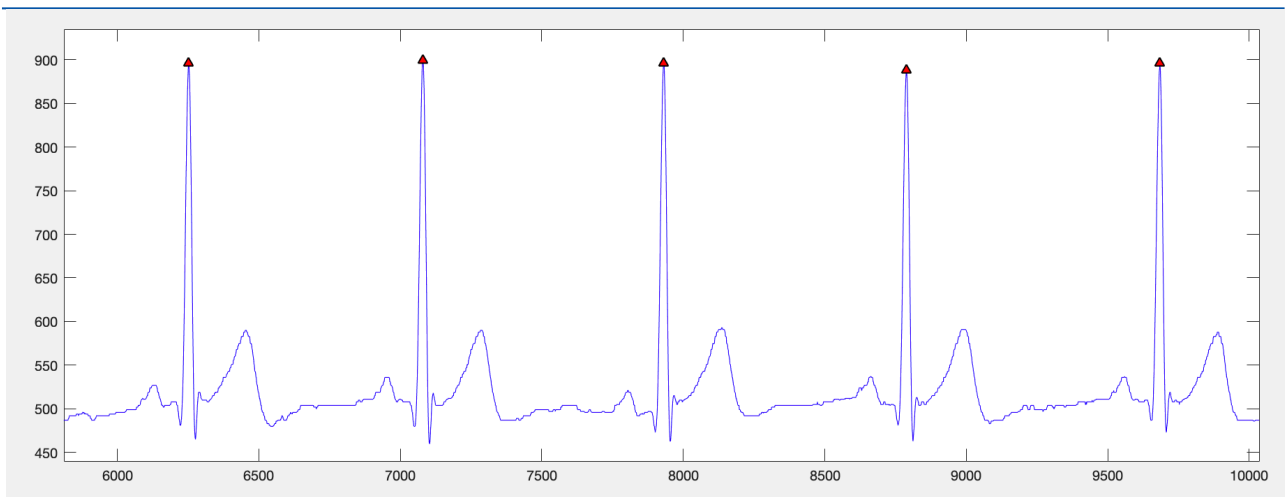
```
%%  
% Leer datos de un fichero  
load('./measure2/formated_data.txt')  
% Nos quedamos con la columnas en la que están los datos del ECG  
ECG_raw = formated_data(:,8);  
% Mostramos los datos en una gráfica  
plot(ECG_raw);
```

Extraer los máximos de la honda R

Para realizar esta tarea utilizaremos la función **findpeaks**. Esta función puede configurarse de múltiples formas como indicar la distancia mínima que debe haber entre pico y pico, la anchura máxima y mínima de los picos así como la amplitud máxima y mínima de estos. En nuestro caso utilizaremos la función de modo que nos proporcione todos los picos por encima de cierto valor de amplitud el cual decidimos observando los datos capturados.

```
%%  
% Buscar los picos  
[peaks, x_values] = findpeaks(ECG_raw,'MinPeakHeight', 800);  
% Mostrar los picos sobre el ECG  
plot(ECG_raw,'b-')  
hold on  
plot(x_values,peaks,'k^','MarkerFaceColor','r')  
hold off
```

De esta forma detectamos todos y solo los picos de la onda R. Mostramos una captura en la que se pueden ver 5 picos correctamente detectados.



Calculo del periodo de latido y bpm instantáneo

A partir de los instantes de tiempo en los que se alcanza cada pico de la onda R calcularemos el periodo de la todo como tiempo en segundos entre dichos puntos y los bpm como la frecuencia en minutos de aparición de los puntos.

```
%%
% Cálculo de periodos y de bpms

previous_time = 0;
time_intervals = zeros(1,size(x_values,1));
bpms = zeros(1,size(x_values,1));
for i = 1:size(x_values,1)
    time_interval = (x_values(i) - previous_time)/1000;
    time_intervals(i) = time_interval;
    bpms(i) = 1/(time_interval/60);
    previous_time = x_values(i);
end

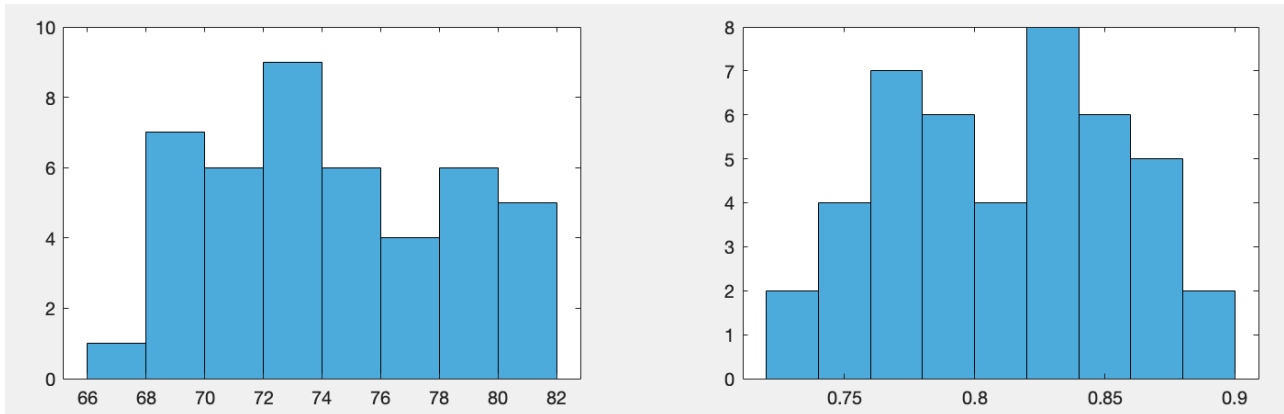
bpm = mean(bpms); time_interval = mean(time_intervals);
bpm_std = std(bpms); time_interval_std = std(time_intervals);
disp(strcat('BPM: ', num2str(bpm), ' | DESVIACIÓN: ', num2str(bpm_std)))
disp(strcat('TIME: ', num2str(time_interval),...
' | DESVIACIÓN: ', num2str(time_interval_std)))

>> BPM:74.2065 | DESVIACIÓN:4.0965
>> TIME:0.81095 | DESVIACIÓN:0.044543
```

Podemos ver que los bpm del ECG están dentro del rango esperado que para personas en repose es entre 70 y 100 bpm. Cabe destacar también que la desviación es reducida, es decir, todos los bpm instantáneos son similares.

Histograma de periodo de latido y bpm instantáneo

A continuación mostraremos dos histogramas correspondientes con los bpm y el periodo de latidos. El histograma mostrará la cantidad de veces que se ha producido cada valor. Para establecer la cantidad de rangos sobre los que calcularemos el número de ocurrencias utilizaremos el método Sturges que tratará de proporcionarnos una cantidad de rangos adecuada a nuestros datos.



Podemos ver a la izquierda el histograma para los bpm y a la derecha el de los periodos. Observamos que el rango de valores es muy reducido y que la forma del histograma parece tener dos pequeñas agrupaciones a ambos lados del centro del rango.

```
%%  
% PLOT BPMS HISTOGRAM  
  
subplot(1,2,1); histogram(bpms, 'BinMethod', 'sturges')  
subplot(1,2,2); histogram(time_intervals, 'BinMethod', 'sturges')
```

Ritmograma

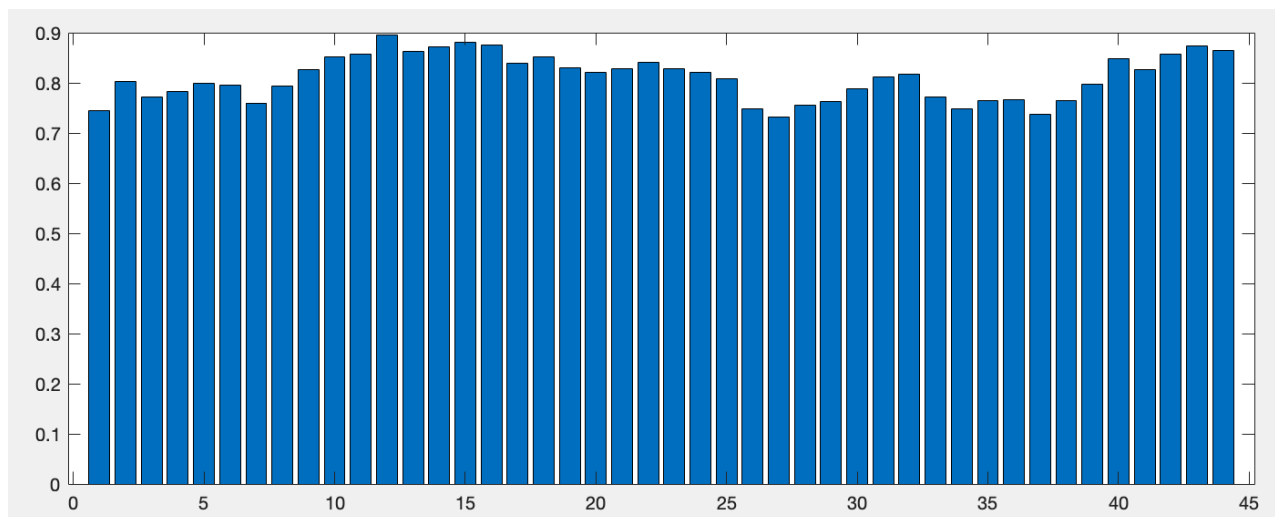
Otra técnica utilizada para ver cómo de estable son los periodos de los latidos en un ECG es crear un ritmograma. Mientras que el histograma nos muestra estos datos a partir de la frecuencia, el ritmograma se centra más en la evolución temporal de los datos. En ambas gráficas no obstante se sigue pudiendo observar el rango de valores, la media y la desviación de estos.

En el ritmograma no nos interesa la concentración de valores en una zona de la gráfica si no que nos interesa que los cambios de valor sean suaves. Podemos ver que el ritmograma obtenido no tiene grandes picos de variación si no que la variación que se produce es continua.


```
%%  
% RITMOGRAMA
```

```
bar(time_intervals)
```

Creamos el ritmograma a partir de una gráfica de barras que nos muestra la evolución de los periodos entre picos de la onda R a lo largo del tiempo.



Gráfica de Poincaré

La gráfica de Poincaré nos aporta información similar a la que el histograma y el ritmograma nos vienen dando. Esta gráfica nos muestra un mapa de puntos que representan dos series temporales con los mismos valores desplazadas un intervalo de tiempo.

En ella podemos observar si los valores desplazados se acoplan a la recta de regresión que es formada por los valores sin desplazar. En nuestro caso esto se produce de modo que podemos deducir que todos los valores de la serie temporal son similares, sin diferencias notables a lo largo de ella. Es decir, que aunque en dos instantes de tiempo los bpm instantáneos sean distintos esta distinción no es relevante.

Adicionalmente sobre la gráfica de Poincaré se representan los límites de desviación sobre la regresión para una ponderación de la unidad para la desviación típica en el teorema de Chebyshev. Dicho teorema nos garantiza que para la ponderación elegida al menos la mitad de los valores de la nube de puntos estarán entre las rectas de color gris. Como podemos observar el porcentaje de puntos es mayor por lo que se podemos confirmar la conclusión expresada en el párrafo anterior.

Debido a que nuestra muestra es reducida la forma ovalada esperada en la distribución de los puntos en la gráfica no es demasiado apreciable.

```

%%
% POINCARE

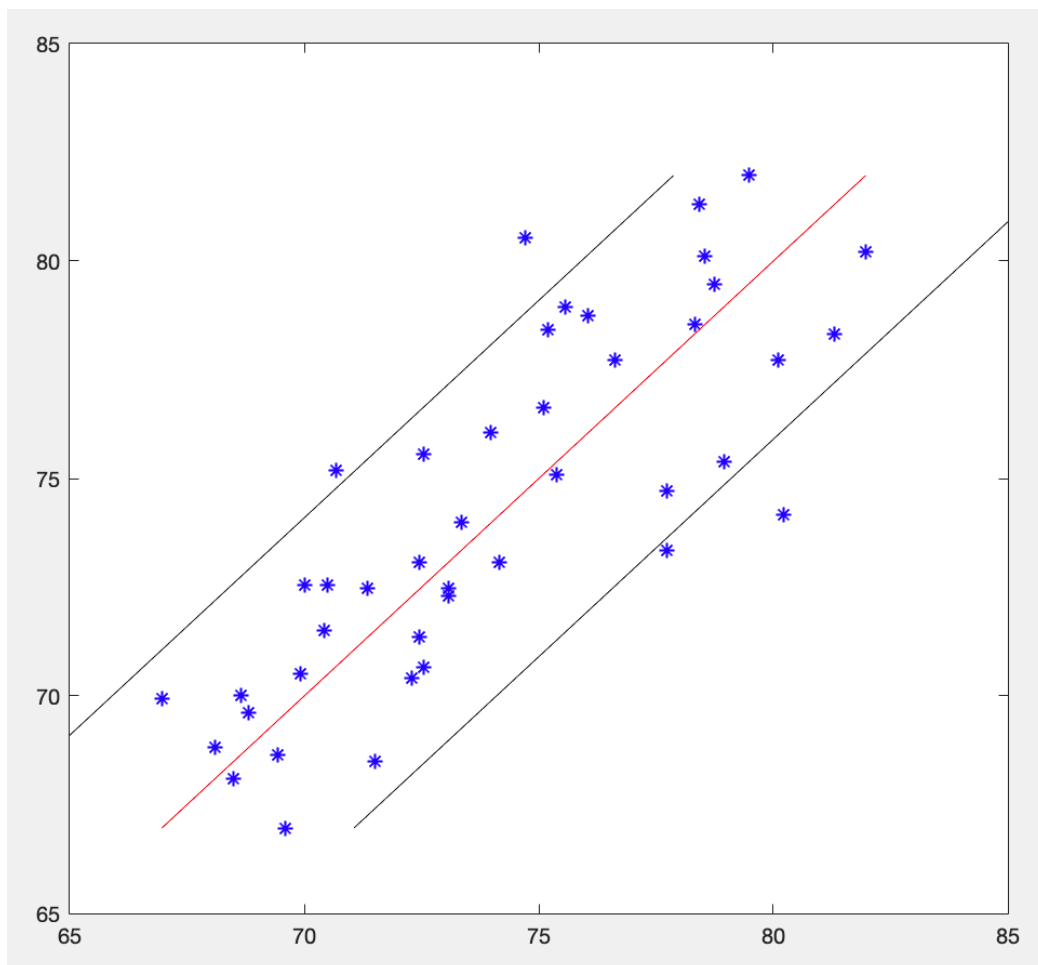
% La primera serie es n0, n1, n2 ...
% La segunda serie es n1, n2, n3 ...

X = bpms;
X(end)=[];
Y=bpms;
Y(1)=[];

% Mostrar la gráfica
plot(Y,X,'*b');
hold on
plot(bpms,bpms,'-r');
plot(bpms-bpm_std,bpms,'-k');
plot(bpms+bpm_std,bpms,'-k');
hold off

```

En azul las series temporales desplazadas, en rojo la regresión esperada y en gris el intervalo de Chebyshev ponderando la desviación típica con la unidad.



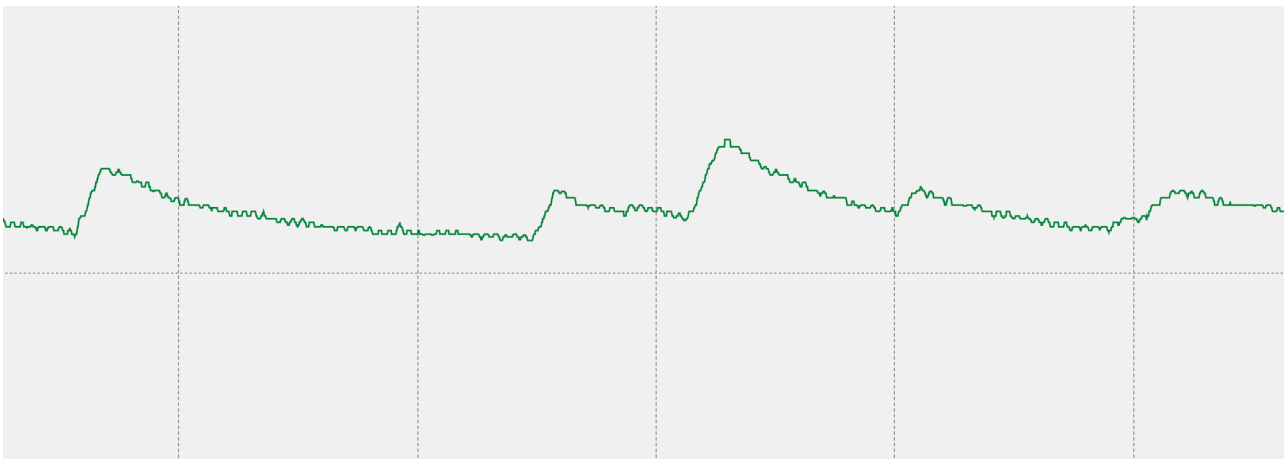
EOG

Hemos realizado un EOG utilizando la placa BITalino desde el programa OpenSignals para escritorio. Para hacer esta medición hemos colocado los electrodos en las sienes tal y como se nos indicó y el electrodo negativo en la cintura.

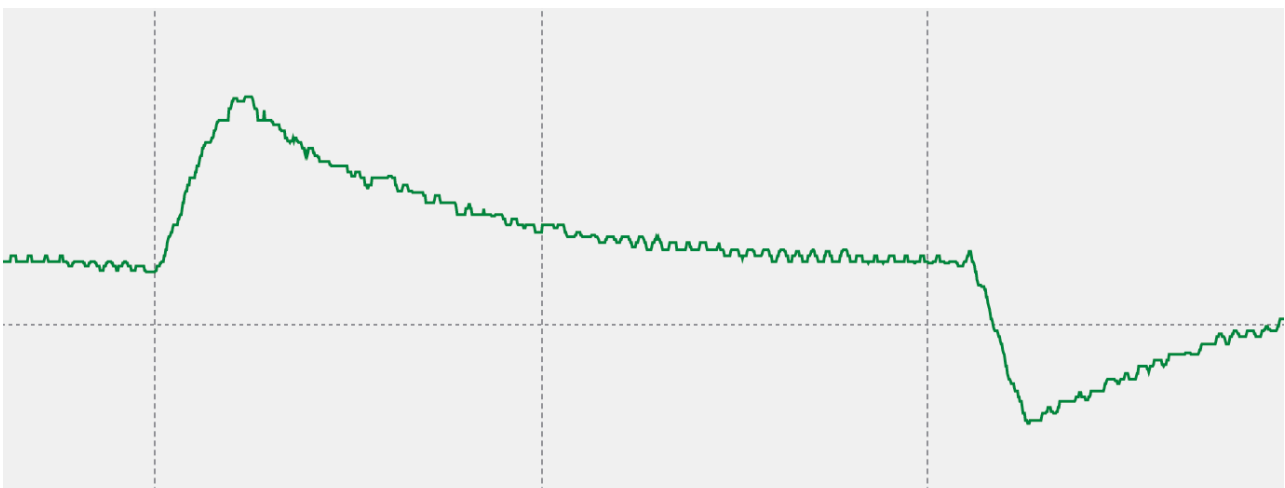
Hemos observado el siguiente comportamiento. Si movemos lo ojos hacia el lado derecho vemos una amplitud en un signo y si los movemos hacia el lado izquierdo vemos una amplitud similar pero de signo contrario. La amplitud de la escala es proporcional al ángulo con el que miramos hacia los lados.

Adicionalmente hemos podido ver que hay algo que hace que si mantenemos la mirada en un ángulo concreto hace que la amplitud medida caiga con el tiempo. Esto se debe a la presencia de un filtro formado por dos condensadores en serie con los electrodos cuyo efecto sobre la medición es que las señales que se mantienen en el tiempo y son distintas de 0 acaben convergiendo a ese valor.

En la primera captura podemos ver que estamos incrementalmente mirando hacia el mismo lado. En pequeños intervalos estamos aumentando el ángulo con el que miramos pero siempre hacia el mismo lado.



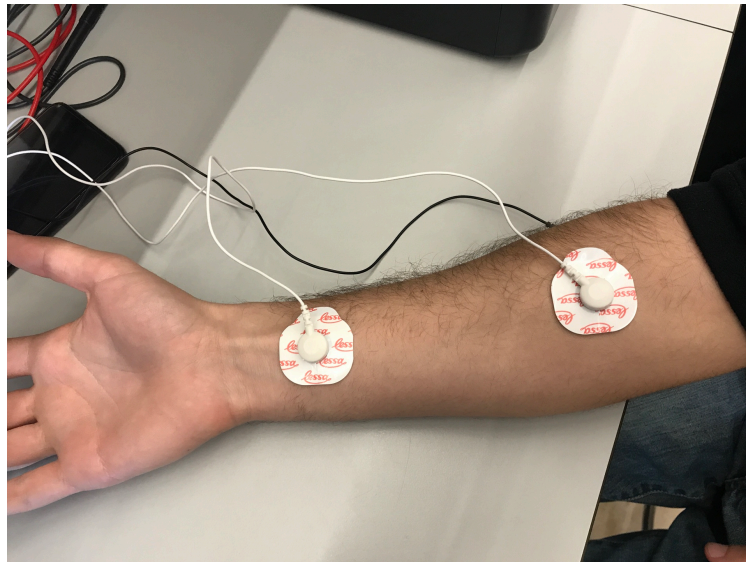
En este otro caso estamos mirando primero hacia un lado y luego pasamos a mirar hacia el otro, se puede ver que al inicio la amplitud es positiva y luego pasa a ser negativa.



EMG

Utilizando el mismo programa hemos capturado la actividad eléctrica de los músculos del antebrazo. Hemos comprobado que dichos músculos se activan con el movimiento de abrir y cerrar la mano.

Hemos elegido hacerlo sobre estos músculos pues hay poca piel sobre ellos, son accesibles para colocar los electrodos y también porque hemos encontrado fotos que nos indicaban como colocar los electrodos.



Esta es la posición en la que hemos puesto los electrodos.

La señal capturada se compone de tres partes. Cuando no realizábamos ningún movimiento solo capturábamos ruido. Por el contrario al abrir la mano se aprecia una onda en un sentido (positivo) y al cerrar en el sentido contrario.

En la captura incluimos dos movimientos de apertura+cierre para comprobar que el patrón que se repite al hacer los movimientos es similar.

