
Detección del ECG Fetal en el ECG Materno

Juan Casado Ballesteros

UAH - 22 Marzo, 2020



Universidad
de Alcalá

Filtros adaptativos

Para detectar el ECG fetal en el ECG materno utilizaremos filtros adaptativos. Es necesario utilizar este tipo de filtros ya que ambos ECG se manifiestan en un rango de frecuencias similar de modo que de utilizar filtros selectivos en la frecuencia (paso bajo, paso alto o de banda) no podríamos separarlos adecuadamente o acabaríamos perdiendo información.

La forma de utilizar el filtro adaptativo será la siguiente:

- Se tomará un ECG de la madre en la zona abdominal. Esta ECG contendrá tanto el ECG fetal como el ECG materno.
- Se tomará otro ECG de la madre en la zona torácica el cual no contendrá el ECG fetal.
- Se pasará un filtro adaptativo sobre el ECG de ambos utilizando el torácico como señal de referencia de modo que se separarán el ECG materno que está repetido en la señal de referencias en la de ambos del ECG fetal que solo está en la de ambos.

El filtro adaptativo modelará la señal de referencia extrayéndola de la señal combinada separando la componente materna de la fetal.

El algoritmo encargado de minimizar el error entre la señal de referencia y la combinada es LMS (Least Mean Square). Este algoritmo aplicado a los filtros adaptativos realiza dos acciones.

Filtrado: se obtiene una salida por combinación lineal de la entrada y el error de la salida anterior con la referencia.

Adaptación: se ajustan los parámetros de la combinación lineal de modo que el error para la siguiente entrada sea menor que el producido para la actual.

Funciones auxiliares

Para resolver algunas de las secciones de la práctica necesitaremos poder generar señales ECG sintéticas. Para ello utilizaremos la función ECG proporcionada por el profesor. Sobre esa función crearemos funciones auxiliares que nos ayudarán a crear nuevas señales ECG de forma más sencilla, también crearemos otras funciones para visualizar las señales ECG sean sintéticas o reales. Estas funciones serán de gran utilidad también para cuando utilicemos señales reales pues serán compatibles con ellas también. Adicionalmente nos permitirán adaptar los ECG que generemos a los ECG reales de modos que coincidan en escala.

Crear ECG sintético

```
function [multy_ecg_wave, ecg_waves] = createEcg( ...
    wave_resolution,...    % Points in a single ecg wave
    amplitude,...          % Maximun amplitude of the ecg wave
    samples,...            % Points in the multy_ecg_wave
    roundness,...
    offset...
)
if nargin < 4
    roundness = 21;
end
    ecg_waves = samples/wave_resolution;    % Number of beaps in the
multy_ecg_wave

    % Obtain a random offset from which to start taking ecg data
if nargin < 5
    offset = round(wave_resolution*rand(1));
end
    % Calculate the number of waves to be created to have enough data
num_ecg_waves = ceil((samples + offset)/ wave_resolution);

    % Create a single wave, smooth it, replicate it and trim it following the
configuration
    single_ecg_wave = amplitude*ecg(wave_resolution);
    full_ecg_wave = sgolayfilt(kron(ones(1, num_ecg_waves), single_ecg_wave), 0,
roundness);
    multy_ecg_wave = full_ecg_wave((1:samples) + offset);
end
```

Esta función nos permitirá obtener ECG con una resolución, cantidad de puntos y amplitud concretos. Adicionalmente podremos controlar el offset con el que se inicia el ECG que de no hacerlo será aleatorio y el factor con el que se redondea la curva. Si

tenemos pocas muestras el factor de redondeado de la curva deberá de ser menor que si estas son muchas para que el ECG pueda verse adecuadamente.

Esta función puede utilizarse para crear ECGs sintéticos que sean compatibles con ECGs reales en base a las siguientes restricciones:

- La amplitud del ECG fetal debe ser unas 15 veces menor que la del ECG materno.
- La cantidad de muestras en ambos ECGs deben ser las mismas.
- Podemos utilizar el siguiente código para calcular la resolución del ECG de modo que obtengamos este con unos bpms determinados.

```
resolucion_sintetico = ceil((frecuencia_real/bpms_deseados_sintetico)*60);
```

Cálculo de los bpms

Cuando sepamos la resolución del ECG, es decir, la cantidad de puntos que hay en una onda completa podremos utilizar la siguiente función para saber sus bpms. Esto solo lo podremos saber en los ECG sintéticos.

```
function [bpms] = calculateBpms (...  
    wave_resolution,...    % Points in a single ecg wave  
    sample_frequency...    % Frequency at which ecg is recorded  
)  
    bpms = (sample_frequency/wave_resolution)*60;  
end
```

Cuando el ECG no sea sintético utilizaremos el código creado en la práctica anterior para detectar los picos de la curva y calcular a partir de la distancia en tiempo entre ellos los bpms.

Visualizar el ECG

La visualización del ECG solo necesita el cálculo de un vector que escale las muestras de la curva del ECG (amplitud en cada punto) a la escala de tiempo definida.

El resto de acciones realizadas en la función como visualizar solo un fragmento de la curva o ponerle un título a la gráfica o añadir una celdilla en el fondo son solo para ayudar a la correcta interpretación del ECG.

Esta función será compatible con los ECG sintéticos y los reales.

```

function plotEcg(...
    multy_ecg_wave,...    % Ecg waves to be plotted
    sample_frequency,...  % Frequency at which ecg is recorded
    view_width,...        % Number of ecg waves to visualize
    amplitude,...         % Maximun amplitude of the ecg wave
    samples...            % Points in the multy_ecg_wave
)

% Create the time scale for the ecg
sample_period = 1/sample_frequency;
time_scale = sample_period:sample_period:(samples*sample_period);

% Plot the ecg
plot(time_scale,multy_ecg_wave);
axis([0 view_width -(amplitude+0.5) (amplitude+0.5)]);
grid;
xlabel('Time [sec]');
ylabel('Voltage [mV]');
end

```

Leer ECG reales

Para poder leer los ECG reales extraídos de PhysioBank necesitamos crear un pequeño código que los convierta al mismo formato con el que se manipulan los ECG sintéticos. Debemos leer información de archivo .mat que contiene las amplitudes del ECG y formatearla con los datos del archivo .info para el mismo ECG. Dentro de cada archivo existirán varios ECG lo cual deberemos tener en cuenta también de modo que acabemos proporcionando la información para uno concreto.

```

function [samples, sampling_frecuency, data]=loadATM(Name, index)

infoName = strcat(Name, '.info');
matName = strcat(Name, '.mat');
load(matName);
fid = fopen(infoName, 'rt');
fgetl(fid); fgetl(fid); fgetl(fid);
[freqint] = sscanf(fgetl(fid), 'Sampling frequency: %f Hz Sampling interval: %f
sec');
interval = freqint(2);
fgetl(fid);

for i = 1:size(val, 1)
    [row(i), signal(i), gain(i), base(i), units(i)]=...
        strread(fgetl(fid), '%d%s%f%f%s', 'delimiter', '\t');
end

```

```
fclose(fid);
val(val==-32768) = NaN;

for i = 1:size(val, 1)
    val(i, :) = (val(i, :) - base(i)) / gain(i);
end

sampling_frecuency = freqint(1);
samples = size(val,2);
data = val(index,1:size(val,2));
end
```

El archivo .info nos indica como debemos de tratar la información contenida en el archivo mat el cual nos crea una variable de Matlab llamada val al leerlo. Debemos escalar los valores contenidos en val para tenerlos en los valores de amplitud con los que trabajaremos.

También nos interesa conocer la cantidad de valores que tenemos y la frecuencia con la que fueron recogidos. Con todos estos datos para un único de los posibles múltiples ECG que contendrán los archivos que nos descarguemos de PhysioBank podremos tratarlos del mismo modo que si los hubiéramos creado nosotros mismos.

ECG materno sintético - ECG fetal sintético

Primero deberemos elegir los ajustes con los que deseamos trabajar. Estos ajustes definen la forma de los ECG que generemos, así como nos proporcionan toda la información necesaria para dibujarlos y manipularlos.

% MEASUREMENT DEVICE CONFIGURATION

```
samples = 30000;           % Points in the multy_ecg_wave
sample_frequency = 4000;   % Frequency at which ecg is recorded
view_width = 3;           % Seconds of ecg to visualize
noise_rate = 0.05; % Measured signals always come with some ammount of noise
```

% ECG CONFIGURATION

```
mother_wave_resolution = 2700; % Points in a single ecg wave
mother_amplitude = 3.5;        % Maximun amplitude of the ecg wave
child_wave_resolution = 1725;  % Points in a single ecg wave
child_amplitude = 0.25;        % Maximun amplitude of the ecg wave
```

La configuración se ha extraído del ejemplo proporcionado por el profesor de modo que producirá idénticos resultados.

A continuación crearemos de forma sintética los ECG materno y fetal utilizando las funciones explicadas anteriormente que en última instancia utilizan la función ecg proporcionada.

% MOTHER ECG

```
mother_ecg = createEcg(mother_wave_resolution, mother_amplitude, samples);
mother_bpms = calculateBpms(mother_wave_resolution, sample_frequency);
```

% CHILD ECG

```
child_ecg = createEcg(child_wave_resolution, child_amplitude, samples);
child_bpms = calculateBpms(child_wave_resolution, sample_frequency);
```

El ECG de la madre se correspondería con el ECG torácico en un caso real y el ECG del niño con el que desearíamos obtener tras aplicar un filtro adaptativo sobre el combinado de ambos.

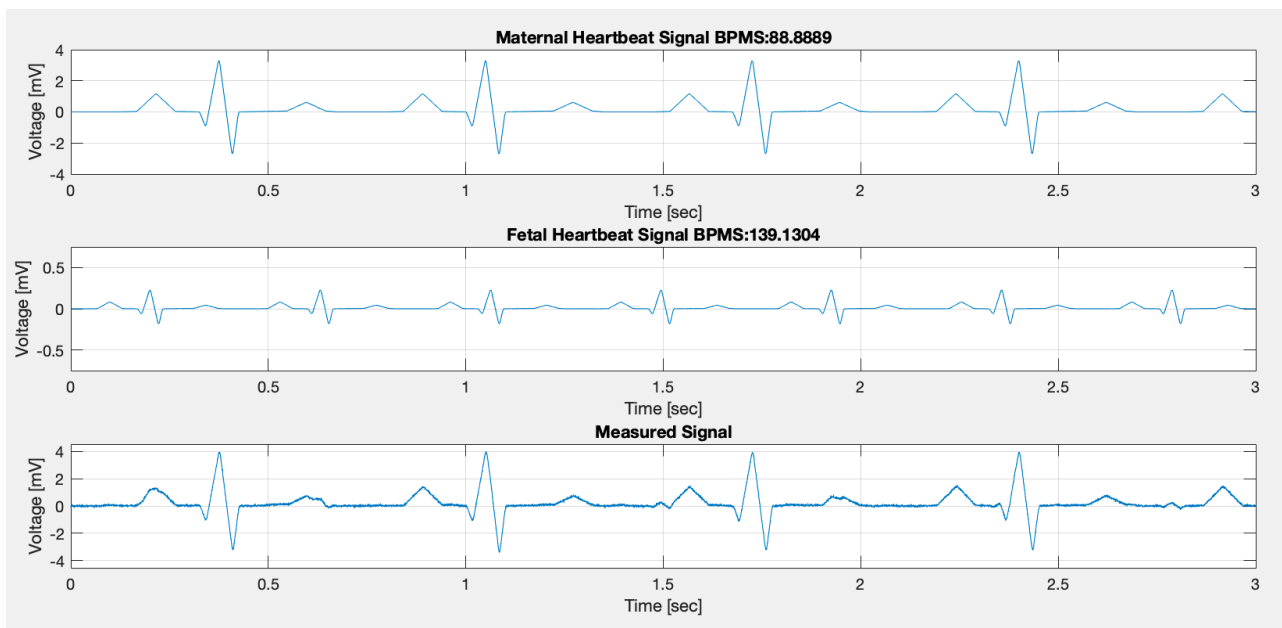
Crearemos ahora a partir de esos ECG el ECG combinado que sería el que mediríamos en el abdomen de la madre en un caso real, para hacerlo más realista le añadiremos ruido.

% COMBINED ECG OF MOTHER + CHILD + NOISE

(WOULD HAVE BEEN CAPTURED DURING PREGNANCY)

```
fir_coeficients = [0 1 -0.5 -0.8 1 -0.1 0.2 -0.3 0.6 0.1];
measured_ecg = filter(fir_coeficients, 1, mother_ecg) +
                child_ecg + noise_rate*randn(1,samples);
```

Mostramos en el mismo orden en el que fueron creados los ECG, primero el materno, luego el fetal y por último el combinado de ambos con el ruido añadido.



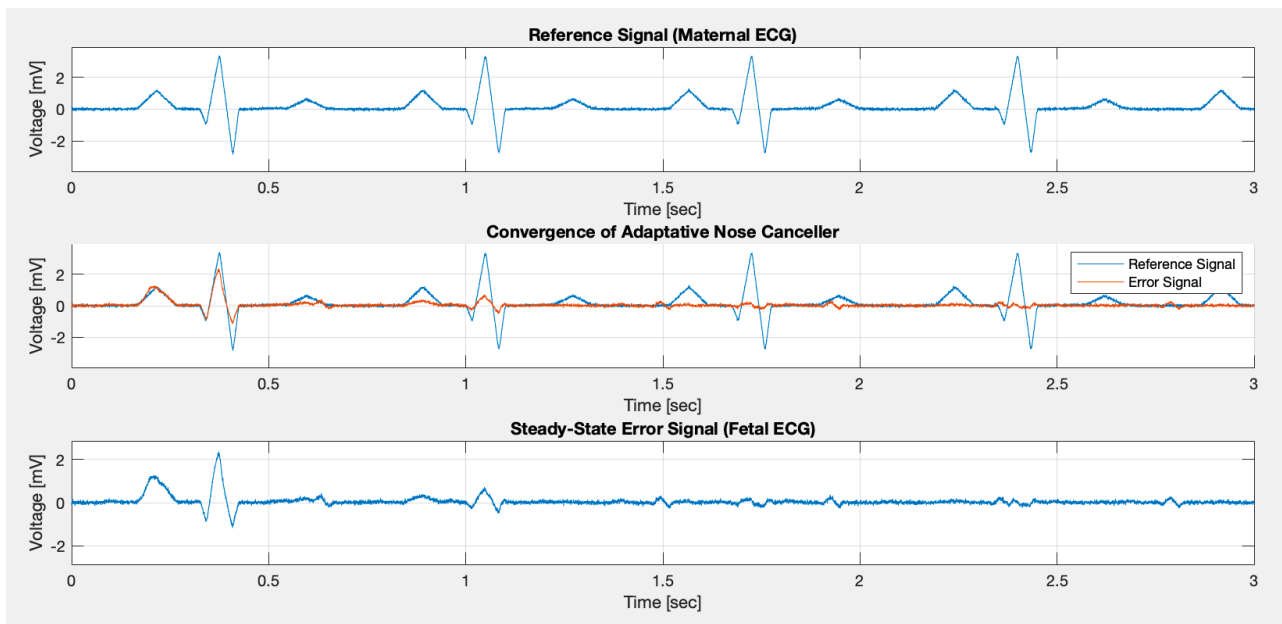
En un caso real el ECG de referencia (medido en el tórax de la madre) no sería exactamente el mismo que el que formaría parte del ECG combinado. Es por ello que hemos añadido ruido en el ECG combina y por lo que añadiremos ruido de nuevo al ECG de la madre antes de utilizarlo como señal de referencia en el filtro adaptativo.

```
% MOTHER ECG + NOISE (WOULD HAVE BEEN CAPTURED WITHOUT PREGNANCY)
reference_signal = mother_ecg + noise_rate*randn(1,samples);
```

A continuación aplicamos el filtro adaptativo sobre el ECG combinado proporcionando la señal de referencia con lo que obtendremos el ECG fetal así como lo que el filtro piensa que es el ECG materno.

```
% LMS adaptative filter to extract child ecg from the combined one
lms = dsp.LMSFilter(15, 'StepSize', 0.00007);
[estimated_mother_ecg, stimated_child_ecg] =
    lms(reference_signal', measured_ecg');
stimated_mother_ecg = stimated_mother_ecg';
stimated_child_ecg = stimated_child_ecg';
```

Tal y como ya se nos advierte en el guión de la práctica la función `lms` utiliza vectores columna así que transbordemos las entradas y las salidas para poder utilizarla. Mostramos a continuación los resultados obtenidos.



Podemos ver la señal del ECG materno con el ruido aplicado sobre ella, esta señal es la utilizada por el filtro como señal de referencia.

Posteriormente tenemos la salida del filtro, en rojo lo que se correspondería con el ECG fetal y en azul lo que se correspondería con lo que el filtro entiende como el ECG materno. Podemos ver como al principio ambas señales son idénticas y que poco a poco según LMS ajusta los parámetros del filtro adaptativo para reducir el error de la salida cada una comienza a diferenciarse de la otra separando ambas señales.

Finalmente mostramos el ECG fetal extraído por el filtro del ECG combinado.

Análisis de los resultados

Nos interesa ahora saber cómo de bien o mal está funcionando nuestro filtro de modo que procederemos a comparar los resultados obtenidos con lo que idealmente hubiéramos deseado obtener.

Calcularemos el error cuartico medio entre los ECG proporcionados por el filtro y los generados sintéticamente.

```
mother_error = mean((mother_ecg - stimated_mother_ecg).^2);
child_error = mean((child_ecg - stimated_child_ecg).^2);
disp(strcat('Maternal mean squared error: ', num2str(mother_error)));
disp(strcat('Fetal mean squared error: ', num2str(child_error)));
```

```
> Maternal mean squared error:0.038992
> Fetal mean squared error:0.026567
```

Los errores obtenidos son satisfactoriamente bajos.

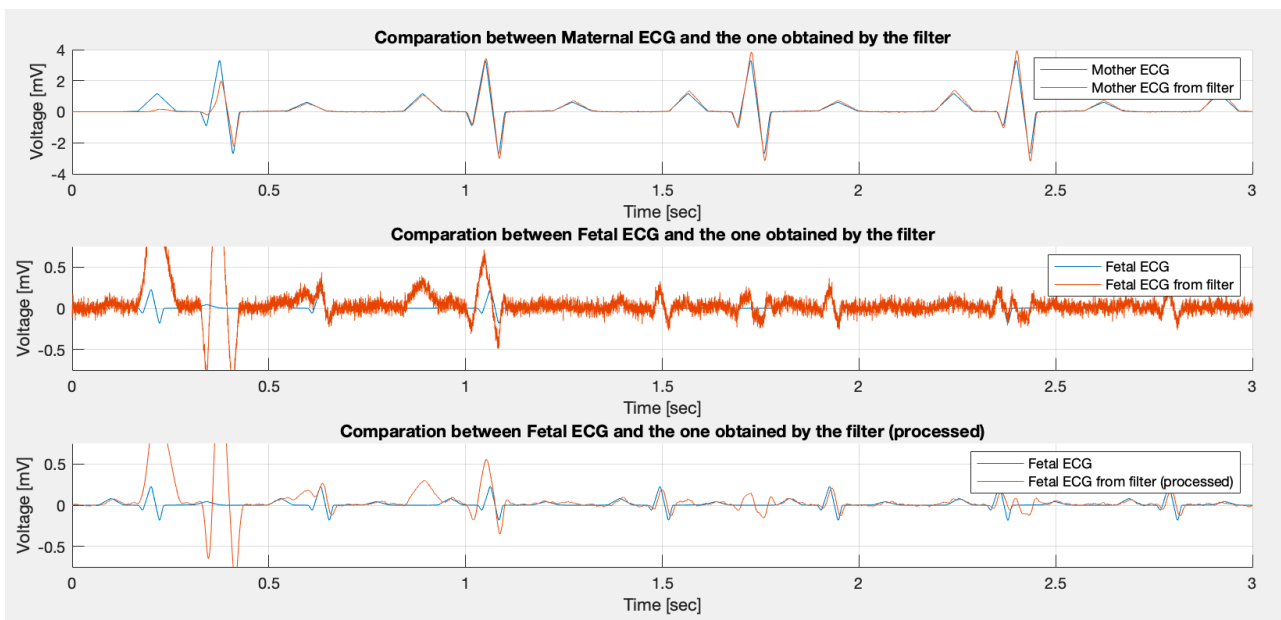
No obstante tras haber evaluado visualmente los resultados descubrimos que la señal del ECG fetal es muy ruidosa y que este no puede distinguirse de forma adecuada. Decidimos por tanto aplicar un filtro sobre él con la intención de minimizar el error cuartico medio por lograr que se asemeje más al ECG fetal original.

Probamos aplicando un filtro de media, un filtro pasa bajos y un filtro de mediana. Los mejores resultados los obtuvimos con el filtro de media debido que el ruido a pesar de ser de frecuencia alta no lográbamos eliminarlo con el pasa bajos sin distorsionar la señal. El filtro de mediana también la distorsionaba en el QRS. Por el contrario el filtro de media lograba resultado suaves obteniendo valores similares a los del ECG fetal real.

```
estimated_child_ecg_filtered =  
    filter(1/50*ones(50,1),1,estimated_child_ecg); % Mean filter  
%estimated_child_ecg_filtered =  
    medfilt1(estimated_child_ecg, 50); % Median filter  
%estimated_child_ecg_filtered =  
    lowpass(estimated_child_ecg, 0.01); % Lowpass filter  
  
child_error = mean((child_ecg - estimated_child_ecg_filtered).^2);  
disp(strcat('Fetal mean squared error: ', num2str(child_error)));  
  
> Fetal mean squared error:0.023488
```

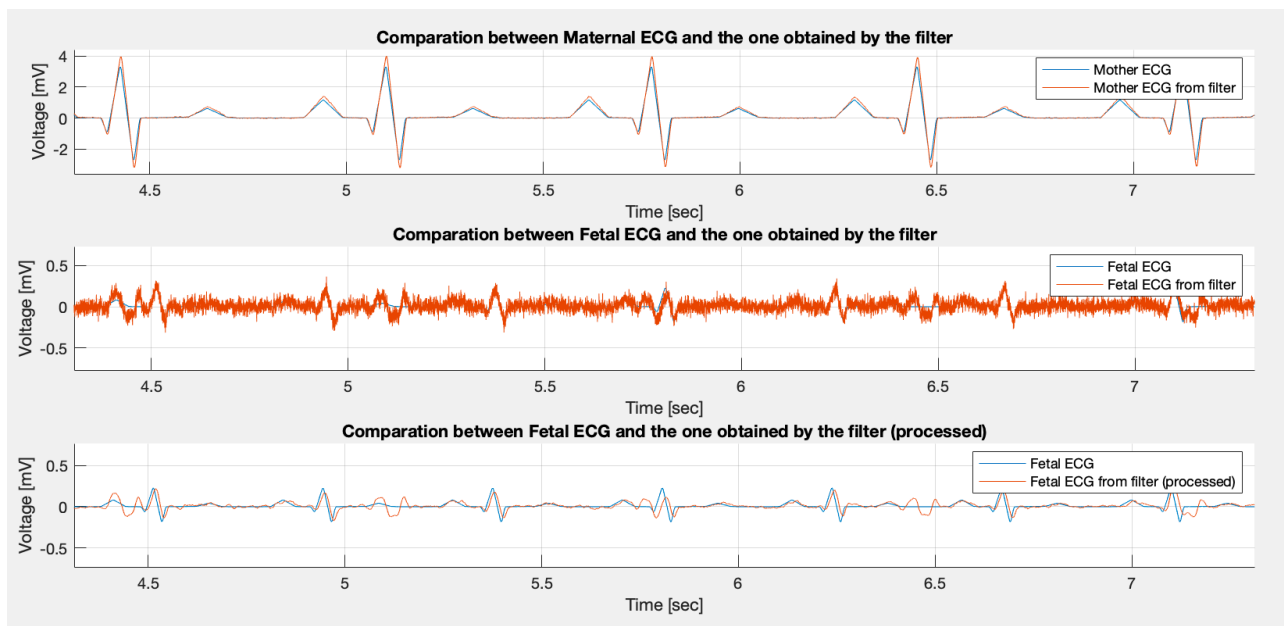
Como vemos el error se ha reducido considerablemente.

Analizamos ahora los resultados obtenidos de forma visual.



Podemos ver que las señales obtenidas son muy similares a las ideales, especialmente según avanzamos en el tiempo, esto se debe a que el filtro converge proporcionando mejores resultados según va filtrando la señal.

Visualizaremos por eso también el final del ECG pues los mayores errores los encontraremos siempre al principio.



Vemos que en efecto las señales obtenidas al final de ECG son mucho más limpias que las que estaba al principio debido a la convergencia de LMS. Podemos ver también como el ECG fetal filtrado es de mayor calidad que el ECG fetal sin filtrar.

Solo el QRS del ECG materno nos proporciona algunas distorsiones en el ECG fetal, vemos que este no es completamente filtrado y nos proporciona pequeñas ondulaciones a intervalos regulares que coinciden con la frecuencia del pulso cardiaco de la madre.

ECG materno real - ECG fetal sintético

Repetiremos ahora los mismos cálculos pero utilizando un ECG real en lugar de uno sintético. Se ha elegido un ECG de adulto que no necesariamente fuera un ECG materno para utilizar un ECG de una base de datos distinta al que se utilizará en la parte final de la práctica. La base de datos utilizada ha sido MIT-BIH Long-Term ECG Database.

La mayor diferencia entre esta parte y la anterior es que en este caso la configuración con la que realizar el ECG sintético (el fetal en este caso) no puede ser arbitraria si no que debe extraerse del ECG real que es el que no podemos modificar.

```
% MEASUREMENT DEVICE CONFIGURATION
view_width = 5;    % Seconds to visualize
noise_rate = 0.05; % Measured signals always come with some ammount of noise

% Mother ECG from: MIT-BIH Long-Term ECG Database
[samples, sample_frequency, mother_ecg]=loadATM('14046m', 1);
samples = min(samples, 29*sample_frequency);
mother_ecg = mother_ecg(1:samples)*10;
mother_amplitude = max(mother_ecg);
```

Sobre este ECG utilizaremos el código de la práctica anterior para calcular los bpm.

```
% CALCULATE MOTHER BPMS
[peaks, at_times] =
    findpeaks(mother_ecg,'MinPeakHeight', mother_amplitude*0.4);
at_times = at_times*1/sample_frequency;
previous_time = 0;
bpms = zeros(1,size(at_times,1));
for i = 1:size(at_times,1)
    time_interval = at_times(i) - previous_time;
    bpms(i) = time_interval*sample_frequency;
    previous_time = at_times(i);
end
mother_bpms = mean(bpms);
```

A continuación creamos el ECG fetal y combinamos ambos tal y como hicimos en la parte anterior. La única diferencia es que para elegir la resolución de una honda del ECG fetal debemos tener en cuenta la frecuencia de muestreo a la que el ECG real se obtuvo para que ambos estén en la misma escala. Esto nos proporciona una única variable dependiente con la que configurar el ECG fetal que en nuestro caso expresaremos como los bpm con los que deseamos obtener este ECG.

Cabe destacar que a este ECG combinado no es necesario añadirle el ruido del filtro lineal pues ya lo contiene por ser real, es decir, no es necesario simular que haya atravesado el

cuerpo pues realmente lo ha atravesado. No obstante le añadiremos el gaussiano para que no sea exactamente el mismo ECG que el de la señal de referencia.

% CHILD ECG

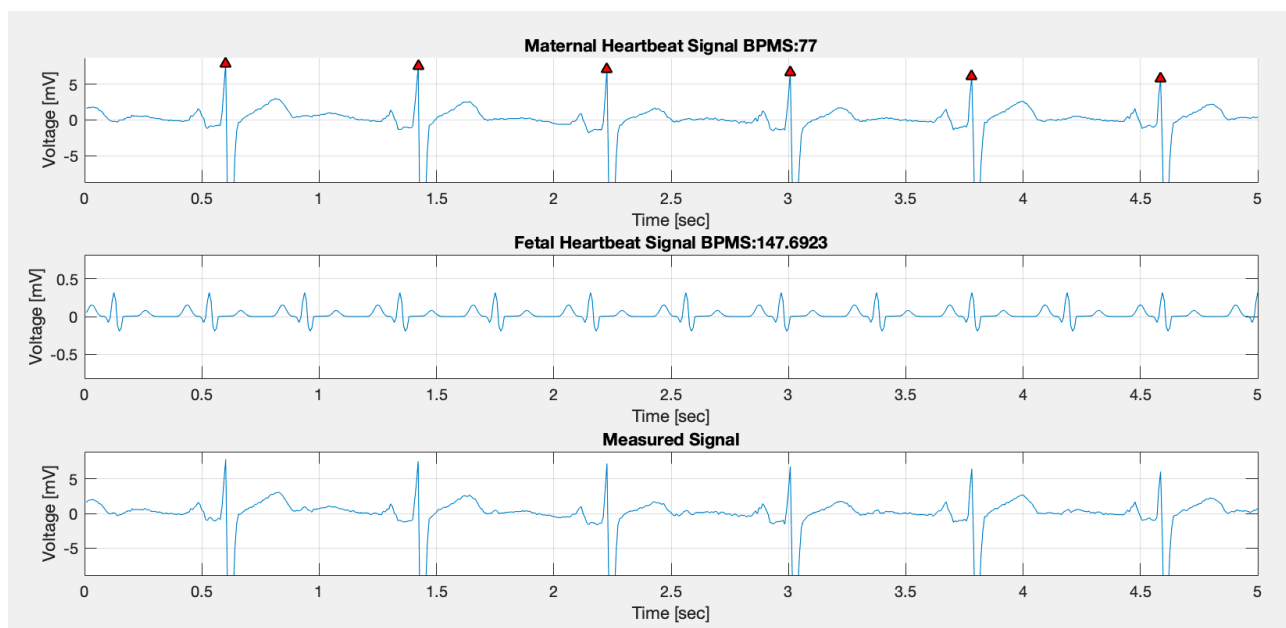
```
% Calculte the points of the child ecg wave based on the desired bpms for it
child_desired_bpms = 150;
child_wave_resolution = ceil((sample_frequency/child_desired_bpms)*60);
child_ecg = createEcg(child_wave_resolution, mother_amplitude/15, samples, 3);
child_amplitude = max(child_ecg);
child_bpms = calculateBpms(child_wave_resolution, sample_frequency);
```

% COMBINED ECG OF MOTHER + CHILD + NOISE

(WOULD HAVE BEEN CAPTURED DURING PREGNANCY)

```
measured_ecg = mother_ecg + child_ecg + noise_rate*randn(1,samples);
```

Visualizamos ahora el ECG real que hemos utilizado, el ECG fetal que hemos sintetizado y la combinación de ambos con algo de ruido añadido.



Tanto en este caso como en el anterior se puede ver que la señal combina es prácticamente idéntica a la del ECG de la madre pues el del feto tiene una amplitud mínima.

Obtenemos ahora la señal de referencia que será el ECG de la madre con algo de ruido añadido para simular que sean señales distintas la de referencia y la que hemos incorporado como la de la madre en la combinada.

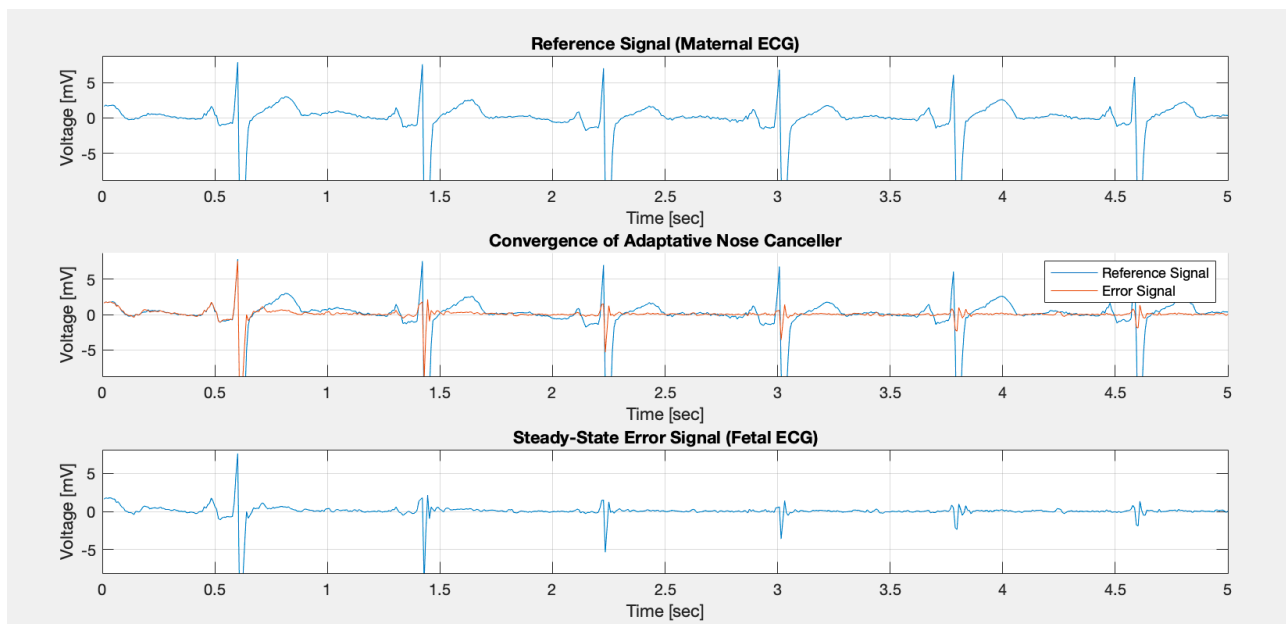
% MOTHER ECG + NOISE (WOULD HAVE BEEN CAPTURED WITHOUT PREGNANCY)

```
reference_signal = mother_ecg + noise_rate*randn(1,samples);
```

Aplicamos el filtro LMS sobre la señal combinada proporcionándole la de referencia.

```
% LMS adaptative filter to extract child ecg from the combined one
lms = dsp.LMSFilter(20, 'StepSize', 0.0007);
[estimated_mother_ecg, estimated_child_ecg] =
    lms(reference_signal', measured_ecg');
estimated_mother_ecg = estimated_mother_ecg';
estimated_child_ecg = estimated_child_ecg';
```

Mostramos a continuación los resultados proporcionados por el filtro.



Podemos ver que en este caso a diferencia de cuando los dos ECG eran sintéticos que la convergencia del filtro es mucho menor. En el caso anterior el ECG fetal comenzaba a dominar rápidamente de modo que era fácilmente distinguible incluso al inicio de la muestra. No obstante podemos ver cómo este no solo no es todavía distinguible si no que además lo que es predominante es el QRS de la madre.

Análisis de los resultados

Compararemos ahora los resultados del filtro con las señales ideales que deberíamos haber obtenido, es decir, el ECG real original utilizado con lo que el filtro nos proporciona como resultado y el ECG fetal sintetizado con el error que el filtro nos da filtro.

Antes de hacerlo calculamos los errores cuarticos medios entre estas señales. Podemos ver que los errores obtenidos son bajos de modo que a primera vista parece que el filtro se ha aplicado de forma adecuada.

```

mother_error = mean((mother_ecg - stimated_mother_ecg).^2);
child_error = mean((child_ecg - stimated_child_ecg).^2);
disp(strcat('Maternal mean squared error: ', num2str(mother_error)));
disp(strcat('Fetal mean squared error: ', num2str(child_error)));

```

> Maternal mean squared error:0.22823

> Fetal mean squared error:0.2309

Tras la inspección visual de los resultados obtenidos se puede ver que la señal del ECG fetal obtenido es bastante ruidosa. Debido a esto se deciden aplicar tres tipos de filtro, de media, de mediana y de frecuencia pasa bajos. Tras evaluar los resultados obtenidos con cada uno de ellos finalmente se decide que el filtro de media es el que logra mejores resultados.

Esta decisión se toma en base a que este filtro no distorsiona la señal lo suficiente como para que no llegue a los picos de amplitud que debería, porque no desplaza la señal tanto como los otros y porque en general logra que esta se asemeje más a la de referencia.

```

stimated_child_ecg_filtered =
    filter(1/3*ones(3,1),1,stimated_child_ecg); % Mean filter
%stimated_child_ecg_filtered =
    medfilt1(stimated_child_ecg, 2); % Median filter
%stimated_child_ecg_filtered =
    lowpass(stimated_child_ecg, 0.35); % Lowpass filter

child_error = mean((child_ecg - stimated_child_ecg_filtered).^2);
disp(strcat('Fetal mean squared error: ', num2str(child_error)));

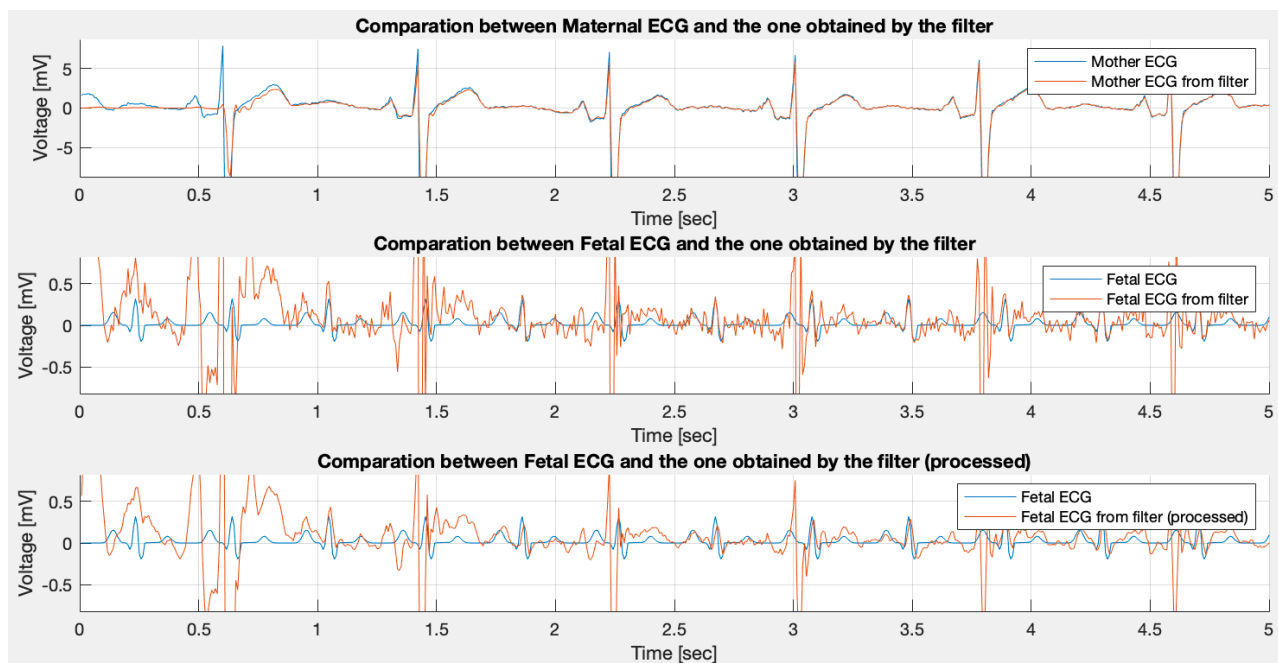
```

> Fetal mean squared error:0.12011

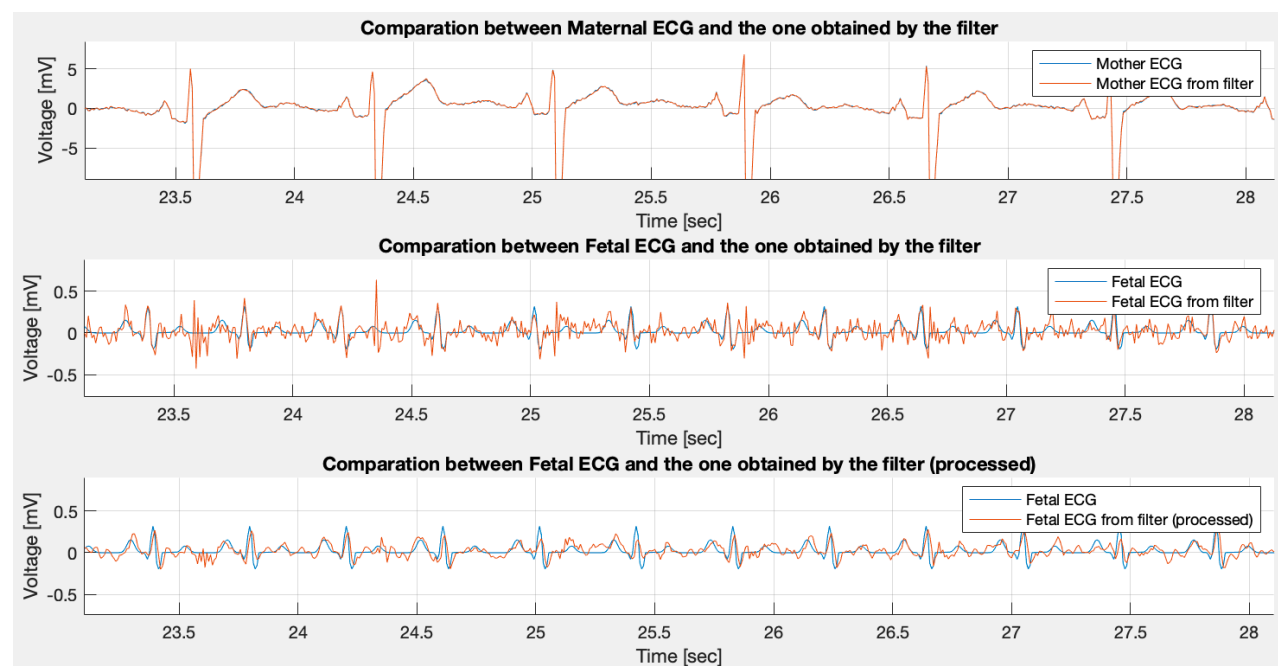
Como vemos el error se ha reducido en gran medida. Es necesario destacar que este se concentra sobre todo al inicio de la gráfica y decae según el filtro adaptativo logra cada vez mejores resultados.

Procedemos ahora a mostrar las gráficas que han guiado el análisis visual. Primero mostramos el inicio del ECG donde el filtro adaptativo todavía no ha podido adaptarse de forma adecuada a las señales de entrada.

Los errores en el ECG materno obtenido con respecto al esperado no pueden apreciarse debido a la escala, no obstante son similares a los que sí se pueden ver en el ECG fetal. En este se ve como hay amplitudes mayores en la zona del QRS fetal pero también hay perturbaciones donde sucede el QRS materno. En este punto ya se puede ver que la señal procesada con el filtro de mediana es más clara que la que carece de él.



Avanzamos ahora al final de la gráfica donde el filtro adaptativo proporciona sus mejores resultados.



Podemos ver que la señal obtenida se asemeja más a la ideal. No obstante se aprecia como el QRS de la madre introduce gran cantidad de ruido. No obstante este es eliminado casi al completo con el filtro de media.

ECG materno sintético - ECG fetal real

Se ha decidido para esta parte utilizar un ECG fetal sintetizado obtenido del repositorio PhysioNet. Estos ECG buscan ser fieles a lo que sería un ECG fetal capturado de forma directa y según indican presentan su misma morfología y características. Los ECG fetales reales se utilizarán solo en la última parte.

Partiremos de este ECG de modo que el que generaremos de forma sintética para simular el ECG de la madre se adaptará a él en número de muestras y frecuencia de muestreo.

Leemos el ECG fetal.

```
% MEASUREMENT DEVICE CONFIGURATION
view_width = 5;    % Seconds to visualize
noise_rate = 0.05; % Measured signals always come with some ammount of noise

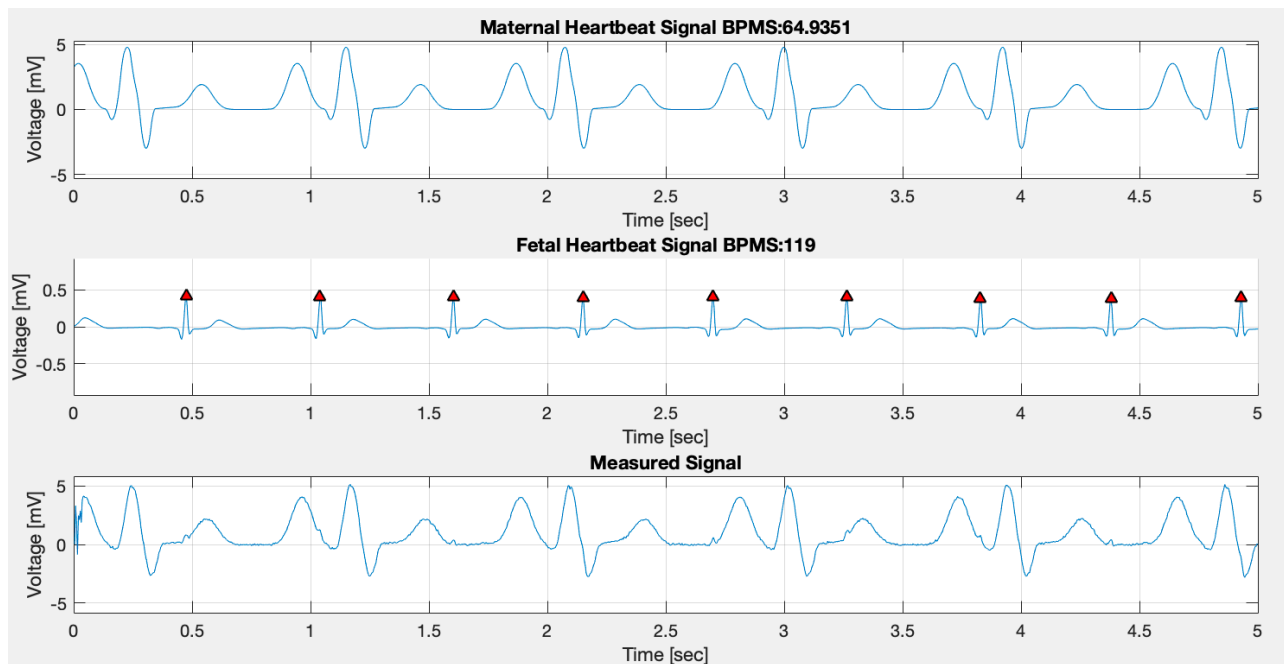
% Child ECG from: Fetal ECG Synthetic Database
[samples,sample_frequency,child_ecg]=loadATM('sub01_snr00dB_l1_c0_mecgm',1);
child_ecg = child_ecg*0.2;
child_amplitude = max(child_ecg);
```

Creamos el ECG materno de modo que sea compatible con el ECG fetal. Le añadimos ruido a la señal combinada de modo que se simule el paso por el cuerpo. Una vez combinado añadimos otro vector de ruido gaussiano para simular el ruido de una medida.

```
% MOTHER ECG
% Calculte the points of the child ecg wave based on the desired bpms for it
mother_desired_bpms = 65;
mother_wave_resolution = ceil((sample_frequency/mother_desired_bpms)*60);
mother_ecg = createEcg(mother_wave_resolution, child_amplitude*35, samples);
mother_bpms = calculateBpms(mother_wave_resolution, sample_frequency);
mother_amplitude = max(mother_ecg);

% COMBINED ECG OF MOTHER + CHILD + NOISE
(WOULD HAVE BEEN CAPTURED DURING PREGNANCY)
fir_coeficients = [0 1 -0.5 -0.8 1 -0.1 0.2 -0.3 0.6 0.1];
measured_ecg = filter(fir_coeficients, 1, mother_ecg) +
               child_ecg + noise_rate*randn(1,samples);
```

A continuación visualizamos los ECG que utilizaremos en el resto de la sección, el fetal, el de la madre y el combinado de ambos. Es justo con esta combinación de ECGs dónde mejor se aprecian las “marcas” del ECG fetal sobre el materno.



Ya que del ECG fetal desconocíamos los bpm's hemos utilizado el código de la práctica anterior tal y como hicimos en la sección anterior para calcularlos.

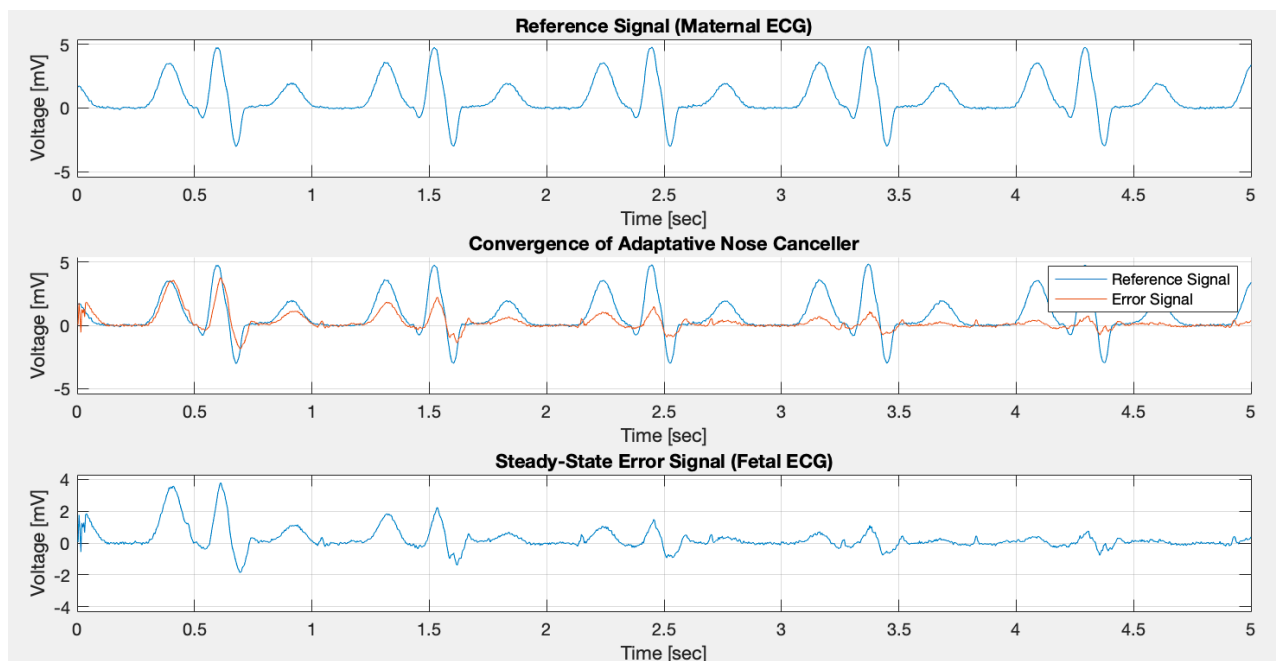
Aplicamos ruido ahora sobre el ECG de la madre para que no sea el mismo que hemos utilizado para crear el ECG combinado. Utilizaremos esta señal como referencia en el filtro adaptativo.

```
% MOTHER ECG + NOISE (WOULD HAVE BEEN CAPTURED WITHOUT PREGNANCY)
reference_signal = mother_ecg + noise_rate*randn(1,samples);
```

Utilizamos ahora el filtro adaptativo para separar la señal combinada en el ECG materno y el ECG fetal haciendo uso de la señal de referencia.

```
% LMS adaptative filter to extract child ecg from the combined one
lms = dsp.LMSFilter(20, 'StepSize', 0.0009);
[stimated_mother_ecg, stimated_child_ecg] =
    lms(reference_signal', measured_ecg');
stimated_mother_ecg = stimated_mother_ecg';
stimated_child_ecg = stimated_child_ecg';
```

Mostramos a continuación los resultados obtenidos. En esta primera visualización ya se puede ver que esta combinación de señales ha tardado más en converger en el filtro que otras. A pesar de que se ha tratado modificar los parámetros del filtro para impedirlo.



Puede verse como el QRS de la madre tiene gran influencia en el ECG obtenido como ECG fetal, al menos cuando el filtro todavía no ha terminado de converger.

Análisis de los resultados

Compararemos cada señal obtenida con las ideales que pudiéramos haber obtenido para comprobar la correcta aplicación del filtro.

Calculamos los errores cuarticos medios entre estas señales que como podemos ver son ligeramente superiores que en los otros casos para el ECG materno pero muy inferiores que en los otros casos para el ECG fetal.

```
mother_error = mean((mother_ecg - stimated_mother_ecg).^2);
child_error = mean((child_ecg - stimated_child_ecg).^2);
disp(strcat('Maternal mean squared error: ', num2str(mother_error)));
disp(strcat('Fetal mean squared error: ', num2str(child_error)));
```

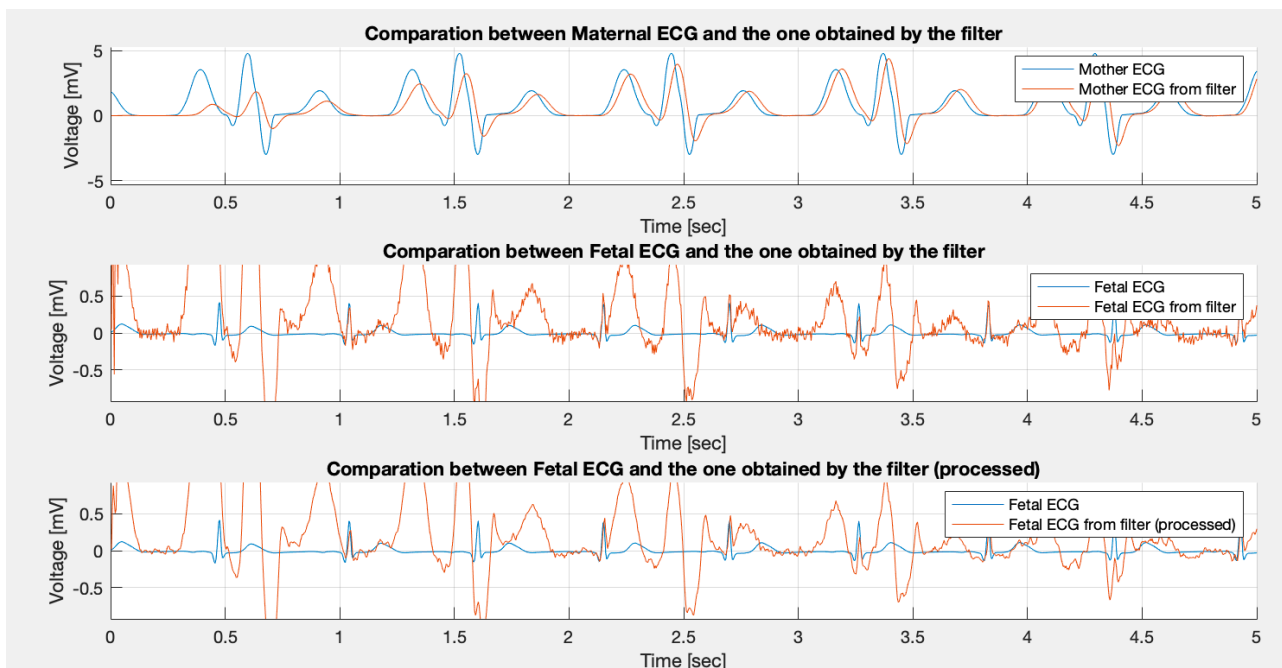
> Maternal mean squared error:0.9706

> Fetal mean squared error:0.049652

Al igual que en las secciones anteriores aplicamos un filtro de media sobre el ECG fetal con la intención de suavizarlo y mejorar su visualización. Para esta nueva señal logramos el siguiente error cuartico medio.

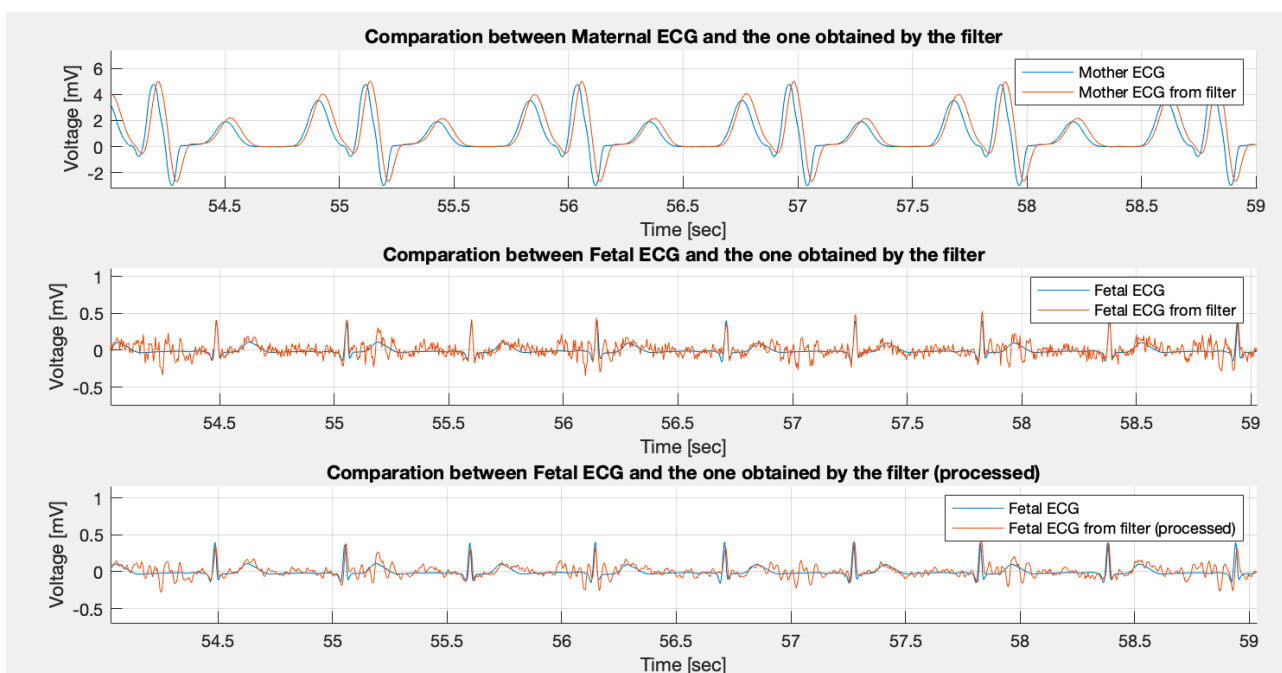
> Fetal mean squared error:0.04767

Mostramos a continuación las señales obtenidas, primero la del ECG materno obtenido comparada con el ECG materno esperado, en la segunda gráfica lo mismo para el fetal y finalmente el ECG fetal tras aplicar el filtro de media.



Cómo vemos la separación del ECG materno no es buena cuando el filtro todavía no ha convergido. A diferencia de en los otros casos parece haberse extraído desfasada. Lo cual encaja con los errores obtenidos.

Si avanzamos en el tiempo al final de la señal cuando el filtro adaptativo ha podido converger podemos ver que la calidad de los ECG extraídos aumenta considerablemente.



No obstante se sigue viendo que el ECG materno se extrae desfasado lo que hace que su QSR se traspase al ECG fetal. Probablemente esto pudiera haberse subsanado con una mejor configuración del filtro adaptativo que a pesar de haberlo intentado no se ha logrado encontrar.

ECG materno real - ECG fetal real

En este caso utilizaremos un ECG de adulto modificado mediante el comando resample como ECG fetal. El comando resample nos permitirá remuestrear el ECG de modo que podamos cambiar su frecuencia hasta obtener los bpm's esperados para un feto (entorno a 140bpm's). Adicionalmente el ECG real de adulto podrá ser utilizado como ECG materno.

El resto del análisis procederá según se ha venido realizando.

El ECG que se utilizará se ha tomado de Physionet, en concreto de la base de datos MIT-BIH Normal Sinus Rhythm Database.

Haciendo uso del comando resample comparamos el ECG de partida a la mitad y lo duplicamos para no modificar el número de muestras generando así el ECG fetal.

% MEASUREMENT DEVICE CONFIGURATION

```
view_width = 5; % Seconds to visualize
```

```
noise_rate = 0.05; % Measured signals always come with some ammount of noise
```

% Mother ECG from: MIT-BIH Long-Term ECG Database

```
[samples, sample_frequency, original_ecg]=loadATM('16272m', 1);
```

```
resampled_ecg = resample(original_ecg, sample_frequency, sample_frequency*2);
```

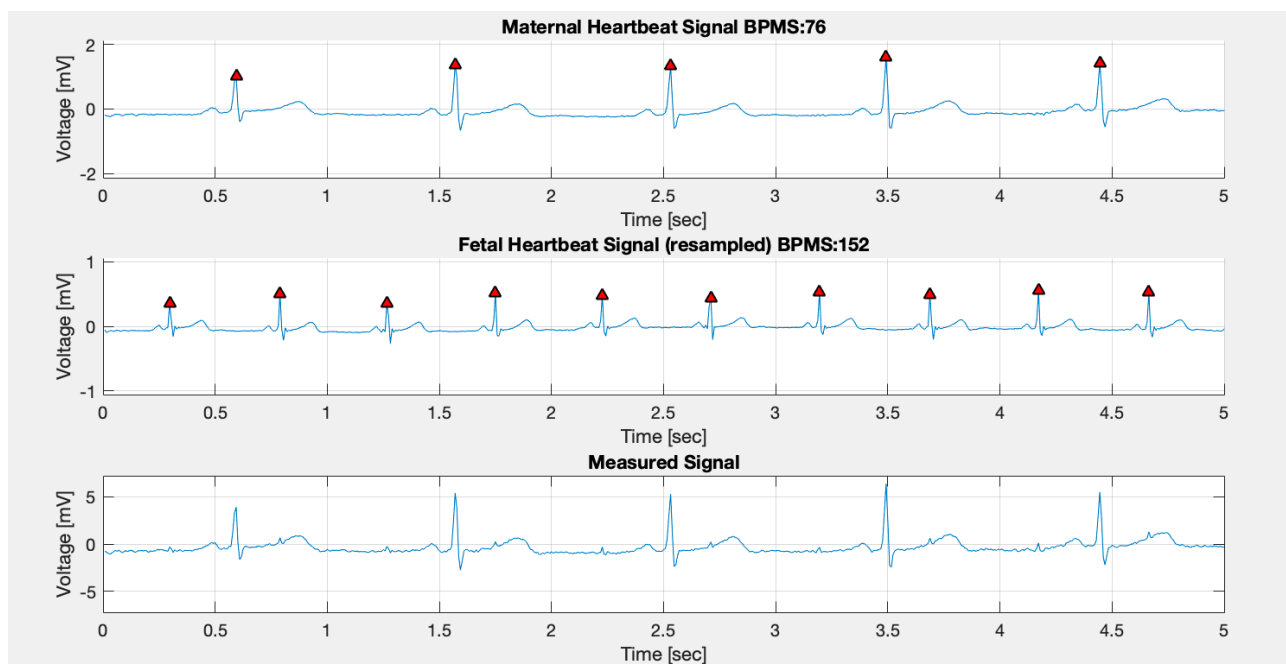
```
child_ecg = [resampled_ecg, resampled_ecg];
```

```
child_ecg = child_ecg*0.3;
```

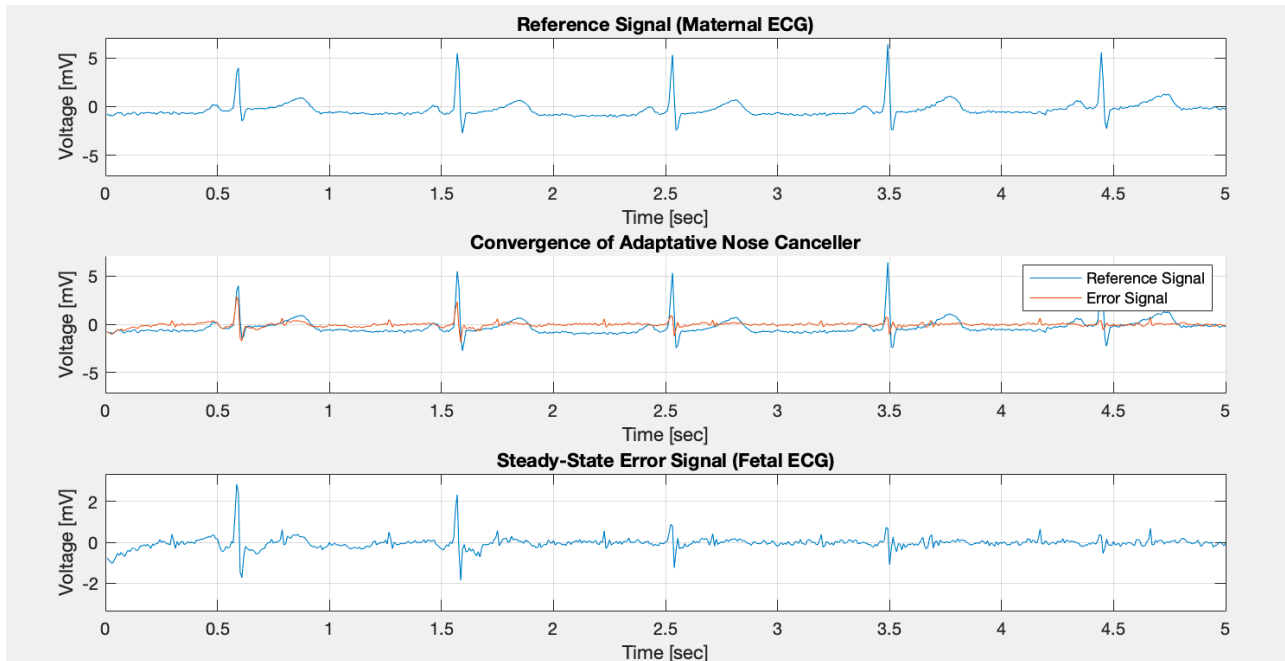
```
original_amplitude = max(original_ecg);
```

```
child_amplitude = max(child_ecg);
```

Mostramos ahora el ECG original como ECG materno y el que utilizaremos como ECG fetal calculando para ambos sus bpm's. Adicionalmente mostramos la combinación de ambos con ruido añadido del mismo modo que hemos estado haciendo hasta ahora.



Creamos a partir del ECG materno una versión ruidosa de este que poder proporcionar al filtro adaptativo como señal de referencia. Aplicamos ahora el filtro adaptativo para separar en dos la señal combinada siendo una de ellas el ECG materno y la otra el ECG fetal.



Se puede ver, como en todos los otros casos que según avanza el tiempo el error del filtro reduce su amplitud adaptándose a una próxima a la del ECG fetal de modo que podemos concluir que el filtro ha convergido.

La configuración del filtro adaptativo para este caso ha sido la más distinta al resto hasta ahora.

% LMS adaptative filter to extract child ecg from the combined one

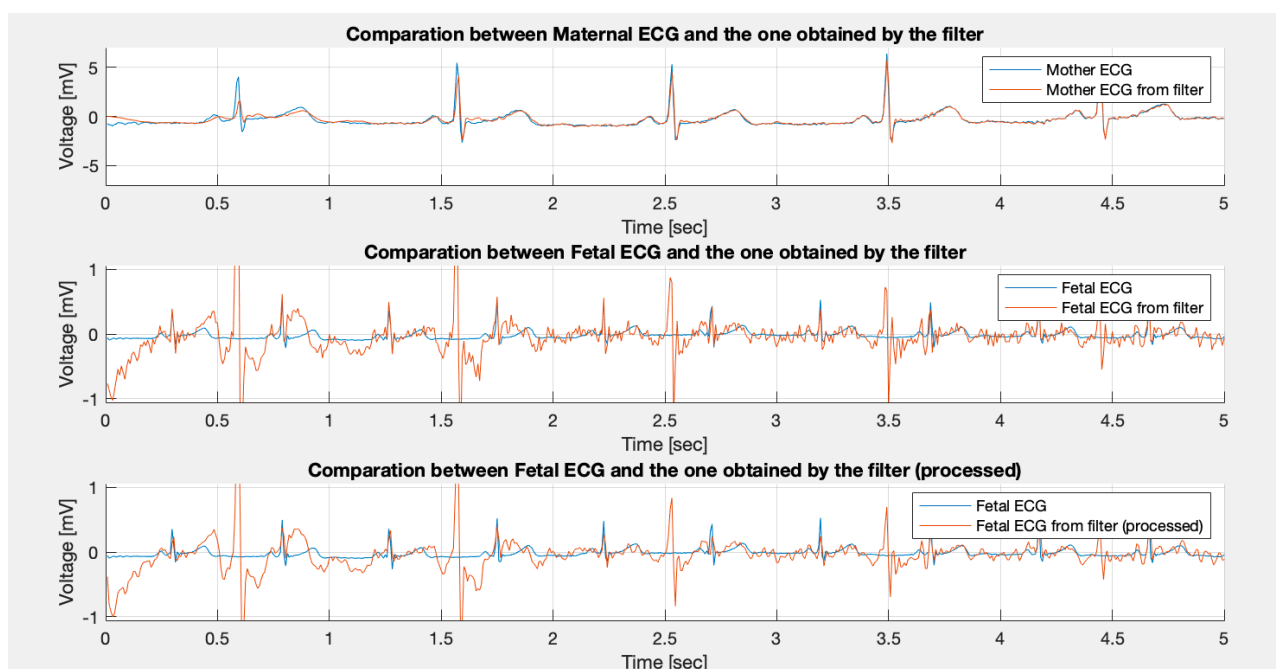
```
lms = dsp.LMSFilter(30, 'StepSize', 0.01);
[estimated_mother_ecg, estimated_child_ecg] =
    lms(reference_signal', measured_ecg');
estimated_mother_ecg = estimated_mother_ecg';
estimated_child_ecg = estimated_child_ecg';
```

Análisis de los resultados

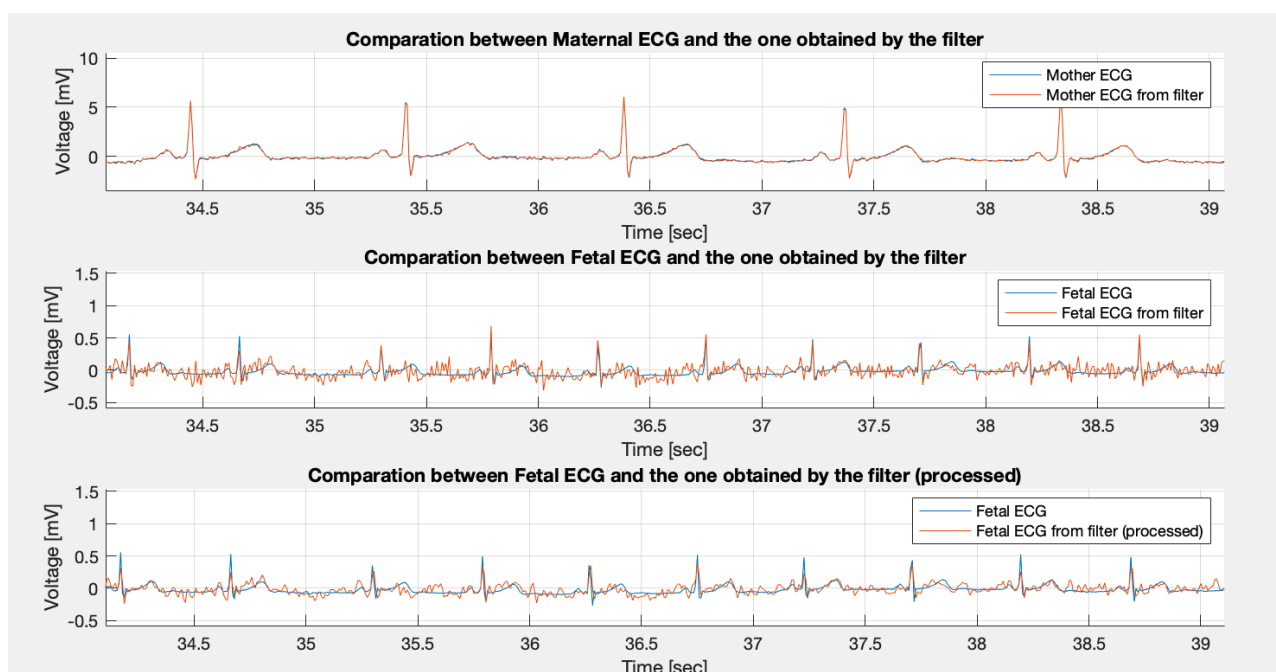
Comenzamos el análisis mostrando directamente los errores cuadráticos medios para el ECG materno con respecto a la señal esperada y para el ECG fetal con respecto a la señal esperada con y sin filtro. Una vez más tras probar los tres tipos de filtro (de media, mediana y pasa bajos) el de media ha proporcionado los mejores resultados.

>> Maternal mean squared error:0.0161
>> Fetal mean squared error:0.01822
>> Fetal mean squared error:0.013913 (with filter)

Vemos que para este caso los errores son más reducidos que en las otras secciones. También podemos ver solo por el error que con estas señales aplicar el filtro sobre el ECG fetal ha obtenido la menor de las mejoras.



Mostramos el estado inicial de la señal donde el filtro adaptativo todavía no ha convergido. La señal del ECG materno aparentemente se adapta bien aunque al hacer zoom se aprecian irregularidades.



Visualizando el final de la señal donde el filtro adaptativo ya ha convergido vemos que ECG fetal obtenido es muy similar al esperado. El QRS materno en este caso apenas ha traspasado al ECG fetal. No obstante El ruido de fondo es elevado y no se ha logrado eliminarlo, el filtro de media a pesar de ser lo más reducido posible afecta gravemente al QRS fetal. Otros filtros cómo el paso bajos lo afectan más todavía pues la amplitud del ruido tiene frecuencias inferiores al QRS y superiores a la honda T.