9

LE1:G2

Práctica 7: Filtro adaptativo utilizando el algoritmo LMS

Laboratorio de PDS

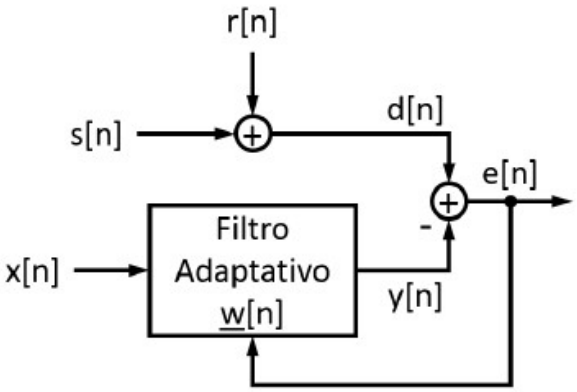
Álvaro Prado Moreno (201800742) y Javier Álvarez Martínez (201707599)
29-4-2021

Introducción

En esta práctica se va a construir un filtro adaptativo utilizando el algoritmo de Last Mean Squares que permita cancelar el ruido de ambiente de una señal. Para observar cómo de eficiente es este algoritmo, una vez se ha realizado el filtrado se compararán la señal resultante de audio y el ruido estimado con los valores reales tanto en el dominio espectral como en el dominio temporal.

Implementación del algoritmo LMS

En este apartado se va realizar una implementación software de la lógica representada en el sistema de bloques que se muestra a continuación. Por medio de iteraciones consecutivas se van a ir adaptando los valores de los coeficientes del filtro $w[n]$ hasta conseguir unos que nos permitan obtener una buena aproximación del ruido $r[n]$ a través del ruido correlado $x[n]$. De esta manera seremos capaces de eliminar esa componente ruidosa de la señal de audio $d[n]$.



s[n]: señal sin ruido de fondo

r[n]: ruido de fondo capturado

d[n]: señal de audio con ruido de fondo

x[n]: ruido correlado con r[n]

y[n]: ruido estimado

e[n]: error del sistema adaptativo

A continuación, se muestra el código del algoritmo LMS que nos permite minimizar la potencia media de la señal de error e[n] del sistema.

```
%% Implementación del algoritmo LMS

%Calculamos todos los coeficientes de nuestro filtro

%El set de coeficientes adaptados corresponde al set anterior
con
%modificaciones

%W[k+1]= w[k]+2*x[k]*e[k]*mu

%Inicializacion de los coeficientes del filtro
w= zeros(1, M+1);

%Inicializamos memoria del filtro
memoria= zeros(1, length(w));
```

```
%Inicializamos el error

e=zeros(1, length(x));

%Matriz que guarda los coeficientes en cada interacion

Matriz_coef=zeros(length(x), length(w));

%Inicializo la y

y=zeros(1, length(x));

for i=1:length(x)

    %Introducimos en la memoria una muestra nueva de x
    memoria=[x(i), memoria(1:end-1)];

    %y[i]=x[k]conv wi[k]

    y(i)=sum(w.*memoria);

    %Calculamos el error: e(i)= d(i)-y(i)
    e(i)=d(i)-y(i);

    %Nuevo set de coeficientes
    w= w + memoria.*(2*e(i)*mu);

    Matriz_coef(i, 1:end)=w;
end

%Acabado el proceso tengo el set de coeficientes bueno
```

En la parte teórica habéis puesto un "−". Está bien en el código, pero tenedlo cuidado.

Justificación del valor del parámetro μ

Falta justificar el valor de μ mediante el cálculo del autovalor máximo de R_x.

El algoritmo implementado se apoya en la definición matemática del gradiente de la superficie del error. El gradiente nos indica la pendiente de la recta tangente a la superficie del error en un punto, es decir, hacia donde el error tiende a ser mayor.

Utilizamos esa información para avanzar en el sentido contrario e intentar minimizar el error. Para ello, restamos a los coeficientes la aproximación del valor del gradiente de la superficie del error ponderada con μ.

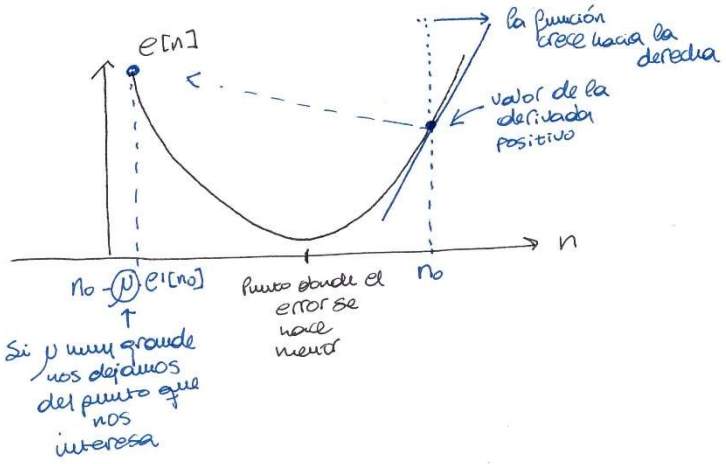
Esta no es la justificación

$$w[n+1] = w[n] - \mu\widehat{\nabla\varepsilon[n]} = w[n] + 2 * \mu * x'[n] * e[n]$$

cuidado

Si el valor de μ no estuviera bien escogido, el valor de los coeficientes finales obtenidos no sería bueno. Esto se debe a que no nos estaríamos acercando cada vez más al punto donde el error se hace 0 si no que estaríamos divergiendo de ese punto.

A continuación, se adjunta una gráfica en 2 dimensiones sustituyendo el concepto del gradiente por la derivada para tratar de aclarar lo explicado:

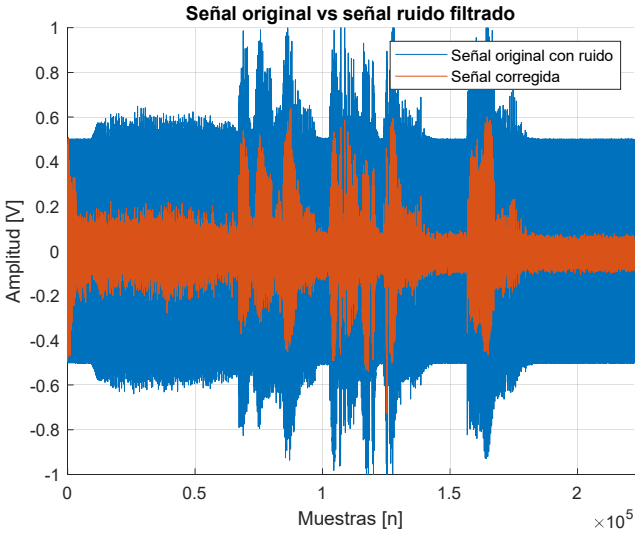


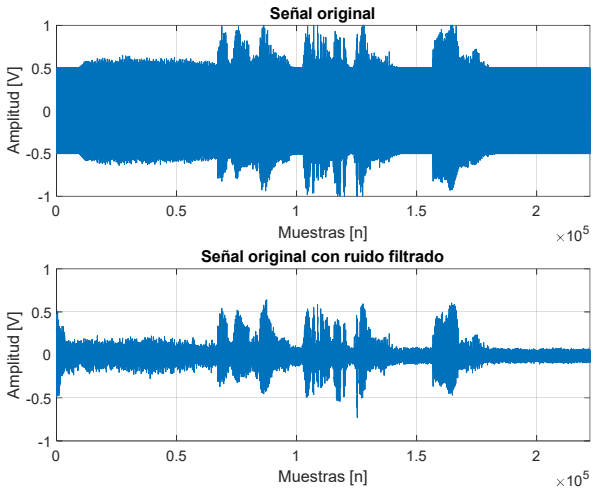
Análisis de resultados

En esta sección vamos a analizar los resultados que obtenemos de aplicar el algoritmo LMS a las señales que se nos ha facilitado.

Análisis temporal de las señales d(t) y e(t)

A continuación, vamos a analizar las señales en el dominio del tiempo $d(t)$ que se corresponde con la señal de audio original del ruido y la señal e(t) que se corresponde a la diferencia entre la señal $d(t)$ y la señal de ruido estimado $y(t)$.



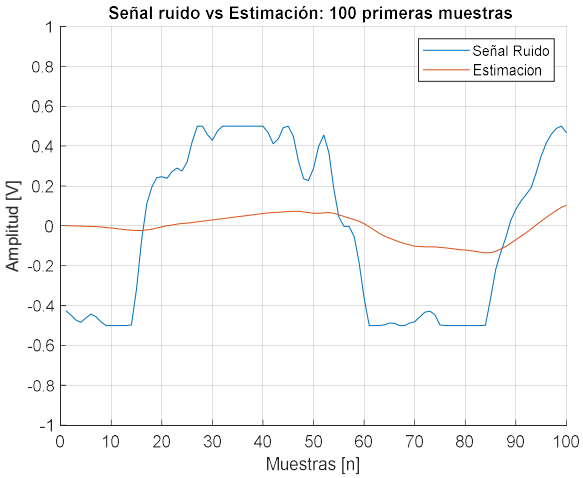


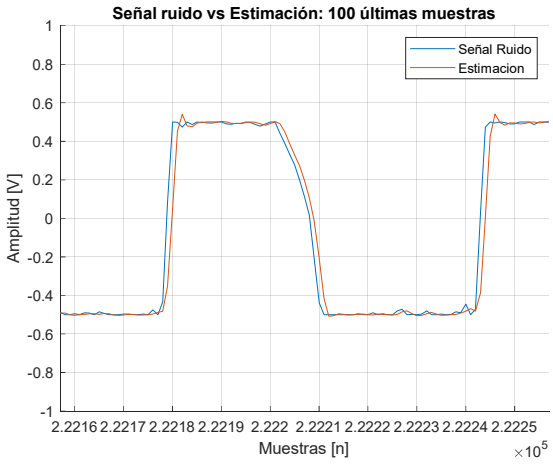
Como se puede observar la aplicación del algoritmo LMS es correcta ya que elimina todas las interferencias externas que se pueden observar en la señal original dejando únicamente la señal con el ruido filtrado.

La señal error, vemos que mantiene la misma forma que la señal original, pero con una menor amplitud y esto se debe a que lo que se elimina es ese aporte del ruido en amplitud.

Análisis temporal de las señales x(t) e y(t)

En este apartado vamos a analizar las señales x(t) que se corresponden al ruido e y(t) que se corresponden con su estimación.

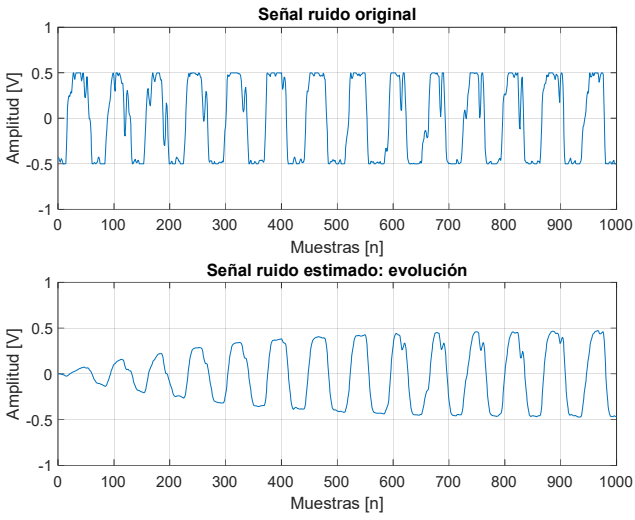




Como se puede observar, en las primeras 100 muestras la estimación del ruido no es la correcta porque aún no han entrado en el filtro el número apropiado de muestras de la señal.

Sin embargo, como se muestra en la imagen de la derecha, en las últimas 100 iteraciones del algoritmo, la aproximación del ruido se aproxima casi en su totalidad con el ruido real.

En la siguiente figura se trata de ilustrar la evolución del ruido estimado $y(t)$. En la imagen se aprecia que tan solo con 1000 iteraciones la aproximación ya es bastante exacta.

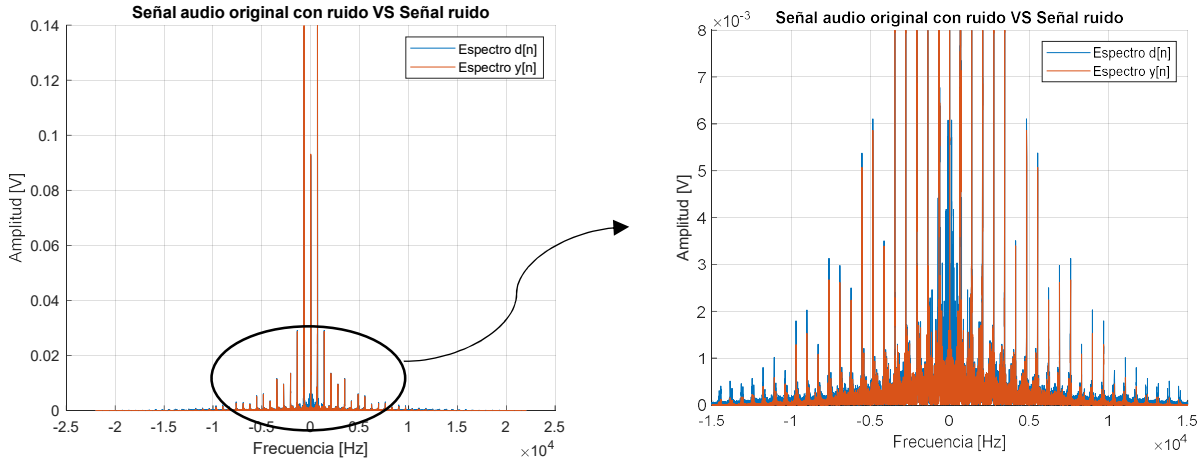


Análisis en frecuencia de las señales d[n], x[n], y[n] y e[n]

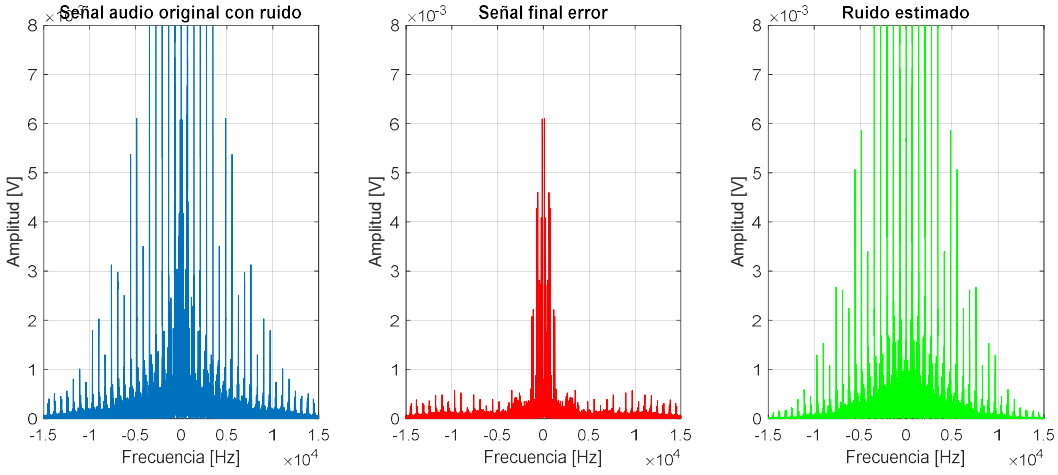
A continuación, se muestra en el dominio de la frecuencia las señales de audio original, la señal de ruido, la señal con el ruido estimado y la señal de audio con el ruido filtrado respectivamente.

En las dos figuras que se adjuntan a continuación, se muestra la comparación entre el espectro de la señal ruido estimado $y[n]$ y la señal de audio original con el ruido $d[n]$. Lo que se puede apreciar es que para las frecuencias que la señal de audio no comparte con el ruido aparece el espectro del ruido no solapado con nadie (la parte azul). Mientras que en las frecuencias que sí coincide con el audio se suman las amplitudes y aproximadamente igual que el ruido estimado (a diferencia de la pequeña aportación del audio).

Estas gráficas se ven mejor en dB. Aquí es complicado ver las diferencias



A continuación, se adjuntan los espectros de la señal error, la señal de ruido estimada y la señal de audio original con su ruido. En el subplot se puede apreciar como sumado la señal error al espectro del ruido estimado se obtiene la señal original.

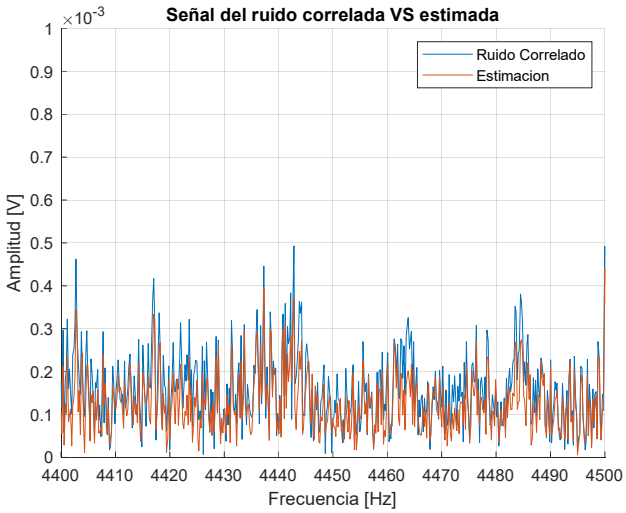


→ mejor decir "aspecto"

Se aprecia una cosa interesante del filtro adaptativo, este es capaz de aportar un valor añadido. No es como un LPF, BPF, HPF que me atenúa ciertas regiones de la banda de frecuencias. El filtro adaptativo es capaz de eliminar la aportación del ruido en una frecuencia sin eliminar por ello el audio en esa frecuencia.

En la siguiente figura se ha representado el espectro del ruido correlado y el del ruido estimado al someter al primero al filtrado adaptativo. Se puede observar como gana amplitud.

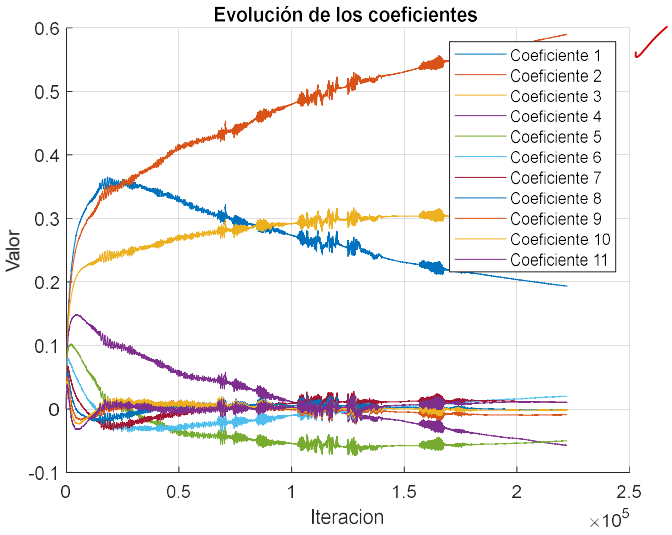
Nota: Se ha realizado zoom para apreciar las diferencias en el espectro.

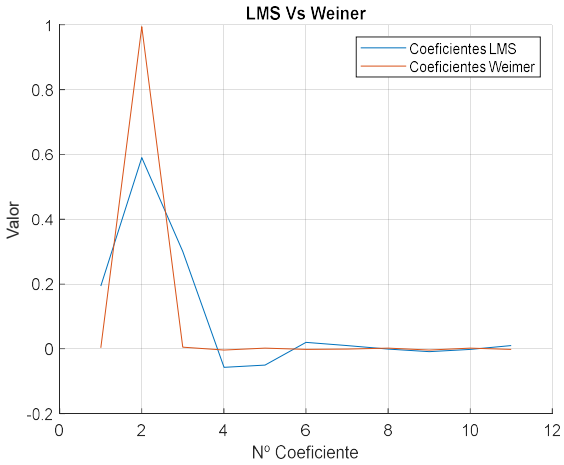


Representación de los coeficientes del filtro vs representación coeficientes filtro Wiener

A continuación, vamos a mostrar los coeficientes de nuestro filtro a lo largo de las iteraciones y vamos a realizar una comparativa con los coeficientes del filtro de Wiener. Este filtro nos da los coeficientes para los cuales el gradiente de la superficie de error se hace 0, es decir, los coeficientes del filtro que nos permiten minimizar nuestra superficie de error.

Se muestran la gráfica con nuestros coeficientes del filtro a lo largo de todas las iteraciones, la comparativa entre los coeficientes del filtro LMS y los coeficientes del filtro de Wiener.





Podemos observar como las mayores variaciones en los coeficientes se producen en las primeras iteraciones, pasadas 17000 iteraciones los coeficientes se estabilizan y el crecimiento hacia los coeficientes óptimos es positivo. Además, podemos observar como el valor de los coeficientes parece que 'tiembla', esto podemos entender que es debido a instantes donde existe únicamente señal de audio y como el valor de los coeficientes tiende a minimizar la señal de error cuando existe solo la señal de audio el valor oscila porque el error no se puede hacer 0 ya que la señal y[n] es una estimación del error y no de nuestra señal de audio.

Como se observa en la segunda gráfica los coeficientes de nuestro filtro y el filtro de Wiener son muy parecidos, esto es correcto ya que como sabemos los coeficientes del filtro de Wiener son exactamente los óptimos y no son aproximados, por eso vemos una ligera diferencia en el valor de estos coeficientes.

Dependiendo del valor de mu cuanto más se acerque a $1/\frac{1}{\lambda max}$,valor máximo que puede tomar, más rápido convergerá el algoritmo. En este caso a pesar de todas las iteraciones vemos que los coeficientes del filtro adaptativo no llegan a alcanzar el valor óptimo del filtro Wiener.

Conclusión

En esta práctica hemos aprendido a realizar un sistema de cancelación de ruido ambiente mediante la realización del algoritmo de Last Mean Squares. Hemos podido aprender como inicialmente la estimación del ruido no es la apropiada, pero, según van pasando más coeficientes por el filtro podemos estimar