

Laboratorio de Procesado Digital de Señal - 3º GITT

Práctica 7: filtrado adaptativo utilizando el algoritmo LMS

En esta práctica se va a construir un filtro adaptativo, mediante el algoritmo LMS (*Least Mean Squares*), que va a permitir cancelar el ruido ambiente de una señal de audio.

Para el desarrollo de esta práctica, el alumno necesita unos altavoces o auriculares.

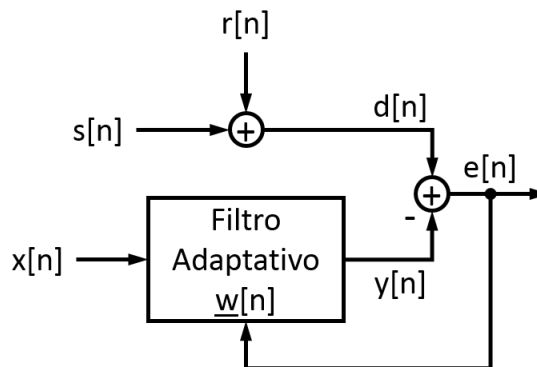
Programa en Matlab los siguientes apartados, comentando el código.

Al finalizar la práctica, suba a la plataforma web de la asignatura (Moodle) un archivo PDF, en el que se responda a los apartados de la práctica y en el que se presenten las figuras que se piden, y un archivo .m, con el código de Matlab con el que se obtienen los resultados.

Diagrama de bloques y datos

Como bien sabe el alumno, los sistemas adaptativos se puede emplear, entre otras cosas, para identificación de sistemas, predicción, cancelación de ruido, cancelación de eco y ecualización adaptativa. En esta práctica se va a construir un sistema adaptativo para cancelar el ruido de fondo de una grabación de audio.

El diagrama de bloques del sistema adaptativo a construir es el siguiente:



Donde $s[n]$ es la señal de audio sin ruido de fondo; $r[n]$ es el ruido de fondo que se ha capturado al grabar la señal de audio; $d[n] = s[n] + r[n]$ es la grabación de audio con el ruido de fondo, el cual se quiere cancelar; $x[n] = \tilde{r}[n]$ es el ruido, correlado con $r[n]$, al cual se debe adaptar el sistema para estimar $r[n]$; $\underline{w}[n]$ son los $M + 1$ coeficientes del filtro adaptativo para el instante n ; $y[n] = \hat{r}[n]$ es el ruido estimado; y $e[n] = d[n] - \hat{r}[n] = \hat{s}[n]$ es el error del sistema adaptativo y, para el caso de aplicación de esta práctica, la estimación de la señal de audio sin ruido de fondo.

Al alumno se le facilita:

- un archivo con la grabación de audio, a la cual queremos quitarle el ruido de fondo, que corresponde con la señal $d[n]$;
- un archivo con ruido de fondo, distinto al ruido de fondo de la grabación de audio, pero correlado con éste, que corresponde con la señal $x[n]$;
- un archivo con los parámetros M y μ del filtro adaptativo.

A partir de los datos facilitados al alumno, realice los siguientes apartados.

- a) Lea el archivo con la grabación de audio que le ha facilitado el profesor. La señal obtenida se denominará $d[n]$. Reproduzca esta señal de audio.
- b) Lea el archivo con el ruido de fondo que le ha facilitado el profesor. La señal obtenida se denominará $x[n]$. Reproduzca esta señal de audio.

Implementación del algoritmo LMS

En este apartado el alumno construirá el sistema adaptativo descrito en el bloque anterior y que cancele el ruido de fondo de la grabación. Para ello, va a implementar el algoritmo LMS visto en clase, como método para minimizar la potencia media de la señal error $e[n]$ del sistema. Puede consultar una introducción de este algoritmo al final de esta práctica y la documentación subida a Moodle.

Realice los siguientes apartados:

- a) Implemente el algoritmo LMS para el sistema adaptativo propuesto, con los parámetros del filtro M y μ facilitados por el profesor.
- b) Incluya el código del algoritmo en este apartado del informe de la práctica, comentando las líneas más significativas.
- c) Justifique si el valor de μ facilitado es coherente para el algoritmo LMS.

Análisis de resultados

En esta última parte de la práctica se van a analizar los resultados de aplicar el algoritmo LMS a las señales facilitadas por el profesor.

Realice los siguientes apartados, a partir de los resultados de los bloques anteriores:

- a) Analice, en el dominio del tiempo, las señales $d(t)$ y $e(t)$, correspondientes a la señal de audio original con ruido y a la señal de audio con el ruido filtrado, respectivamente. Exponga y justifique gráficamente las conclusiones extraídas de dicho análisis.
- b) Analice, en el dominio del tiempo, las señales $x(t)$ e $y(t)$, correspondientes al ruido y a su estimación, respectivamente. Exponga y justifique gráficamente las conclusiones.
- c) Analice, en el dominio de la frecuencia, las señales $d[n]$, $x[n]$, $y[n]$ y $e[n]$. Exponga y justifique gráficamente las conclusiones extraídas de dicho análisis.
- d) Represente el valor de todos los coeficientes del filtro en cada iteración del algoritmo.
- e) Los coeficientes ¿convergen hacia algún valor? Indique cuáles.

Introducción al algoritmo LMS

El algoritmo LMS es un tipo de algoritmo adaptativo que se basa en la aproximación del gradiente de la superficie de error a través de la expresión:

$$\nabla \xi(\underline{w}) \approx -2 \cdot e[n] \cdot \underline{x}'[n] \equiv \widehat{\nabla \xi}[n]$$

Donde el vector $\underline{x}'[n]$ corresponde con las últimas $M + 1$ muestras de entrada al filtro; y M es el orden de éste.

Con esto, los pasos para la implementación del algoritmo LMS son:

- 1.- Actualizar el vector $\underline{x}'[n]$ con la nueva muestra recibida en cada instante.
- 2.- Calcular la muestra de salida correspondiente con:

$$y[n] = \sum_{k=1}^{M+1} w_k[n] \cdot x'[n - k] = \underline{w}^T[n] \cdot \underline{x}'[n]$$

- 3.- Calcular la muestra correspondiente de la señal de error con:

$$e[n] = d[n] - y[n]$$

- 4.- Actualizar los coeficientes del filtro (pesos) siguiendo la aproximación del gradiente con:

$$\underline{w}[n + 1] = \underline{w}[n] + 2 \cdot \mu \cdot e[n] \cdot \underline{x}'[n]$$

- 5.- Continuar hasta la última muestra de $x[n]$.

Funciones de Matlab

Consulte la ayuda de Matlab para conocer el significado de cada uno de los argumentos.

Miscelánea	
Borra todas las variables del Workspace	<code>clear</code>
Cierra todas las figuras abiertas	<code>close all</code>
Limpia la ventana de comandos	<code>clc</code>
Carga en el Workspace las variables almacenadas en el archivo <code>nombre_archivo.mat</code>	<code>load 'nombre_archivo.mat'</code>
Devuelve la longitud del vector x	<code>y = length(x);</code>
Devuelve una matriz de ceros de tamaño a filas y b columnas	<code>y = zeros(a, b);</code>
Muestra el mensaje de texto por pantalla	<code>disp('texto');</code>
Convierte en string (cadena de caracteres) el número x	<code>y = num2str(x);</code>
Representación	
Abre una nueva ventana para representar gráficamente	<code>figure;</code>
Representación en ejes x-y uniendo los puntos	<code>plot(ejex, ejey, '-.');</code>
Congela la figura activa para poder superponer más representaciones	<code>hold on;</code>
Activa la rejilla de la representación	<code>grid on;</code>
Pone título a la representación	<code>title('Texto');</code>
Pone etiqueta en eje x . La función ylabel hace lo mismo en el eje y	<code>xlabel('Texto');</code>
Representa el eje x entre los valores ini y fin . La función ylim hace lo mismo en el eje y	<code>xlim([ini fin]);</code>
Muestra una leyenda en la figura, donde se muestran los textos indicados	<code>legend('Texto1', 'Texto2', ...);</code>

Audio		
Lee el archivo de audio file		<code>[y, fs] = audioread('file');</code>
Reproduce la señal de audio y a una frecuencia de muestreo fs		<code>sound(y, fs);</code>
Vectores		
Genera el vector de valores v entre ini y fin equiespaciados delta		<code>v = ini : delta : fin;</code>
Genera el vector v de N valores equiespaciados entre ini y fin (inclusive)		<code>v = linspace(ini, fin, N);</code>
Genera el vector v con los elementos del vector x desde el a hasta el b (inclusive)		<code>v = x(a:b);</code>
Concatena los vectores fila x e y		<code>v = [x , y];</code>
Concatena los vectores columna x e y		<code>v = [x ; y];</code>
Matemáticas		
Calcula el valor absoluto del número x , o el módulo de éste si es un número complejo		<code>y = abs(x);</code>
Calcula la transformada de Fourier del vector x		<code>X = fft(x, length(x)) / length(x);</code>
Trasposición del espectro en frecuencias X		<code>X = fftshift(abs(X));</code>
Calcula el sumatorio de todos los valores del vector x		<code>y = sum(x);</code>
Multiplica, elemento a elemento, los vectores/matrices x e y , los cuales deben tener el mismo tamaño		<code>z = x .* y;</code>
Calcula la potencia de la señal x		<code>y = var(x);</code>