

## PRÁCTICA 4: CANCELADOR DE ECO ACÚSTICO

Se realizará un programa que realice la cancelación de un eco que contamina a una señal de voz local. El eco es versión de una señal remota modificada (filtrada) por un *canal desconocido*. Las señales corresponden a los siguientes ficheros de audio con formato *wav*:

- *local.wav*: señal local a transmitir (supuestamente no conocida).
- *remota.wav*: señal remota original de la que deriva el eco.
- *signal.wav*: señal local mezclada con el eco.

Las señales se leen/escriben con los comandos *read / write* del módulo *scipy.io.wavfile* (*audioread / audiowrite* en MatLab). Estas señales se encuentran muestreadas a 8 KHz en enteros de 16 bits. En caso de implementación en Python, las muestras pueden convertirse en floats (atributo `.astype(np.float)` de los arrays de señal) para su procesamiento para evitar problemas de formato numérico. La señal de salida float puede convertirse a entera mediante el atributo `.astype(np.int16)` (comando *int16* en MatLab).

### Cuestiones:

1. Calcular la SNR obtenida sin cancelación del eco.

La SNR entre una señal  $x(n)$  y una distorsión aditiva de la misma  $d(n)$  se computa como:

$$SNR = 10 \log_{10} \left( \frac{\sum_n x^2(n)}{\sum_n d^2(n)} \right)$$

En nuestro caso, la señal  $x(n)$  sería la señal *local* que se quiere transmitir y la distorsión es  $d(n) = y(n) - x(n)$ , donde  $y(n)$  es la señal que se transmite finalmente (en este caso sin cancelación,  $y(n)$  corresponde a *signal*).

2. Implementar el cancelador con el algoritmo NLMS inicializando el filtro con un vector de ceros. Buscar el parámetro de convergencia ( $u = 0,0005, 0,001, 0,002, 0,004, 0,008, \dots, 0,128$ ) y el orden del filtro ( $p = 2, 3, \dots, 7$ ) que proporcionen una mejor SNR (en este caso  $y(n)$  es la señal con eco cancelado, que coincide con la señal error). Dibujar la evolución temporal de los coeficientes del filtro (para la configuración de mejor SNR) y comentar el resultado. Escribir la señal local estimada (con eco cancelado) del mejor caso en un fichero formato *wav* (*signal\_canceled.wav*), y escuchar y comentar el resultado.
3. Repetir el apartado anterior suponiendo que un detector de *double talk* (DTD) detecta voz local a partir de la muestra número 2150, de forma que el filtro sólo se adapta hasta esta muestra (para instantes posteriores se mantiene fijo el último cancelador obtenido). Comentar los resultados obtenidos y compararlos con los del apartado anterior.

Dibujar la respuesta en frecuencia del cancelador FIR fijado para los instantes posteriores al 2150 (para el caso con mejor SNR). Comentar qué tipo de filtro es, y qué nos dice esta respuesta en frecuencia sobre la respuesta en frecuencia del *canal desconocido* que transforma la señal remota en el eco que contamina a la señal local.