Practica 1: Simulación de sistemas con Simulink. Cuantización de señales.

Procesamiento Digital de Señales, Universidad de Granada

Autor: Miguel Carracedo Rodríguez

Fecha: 29/03/2022

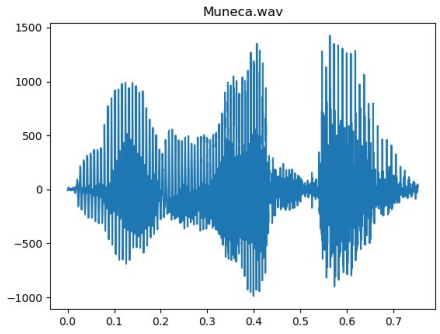
Introducción

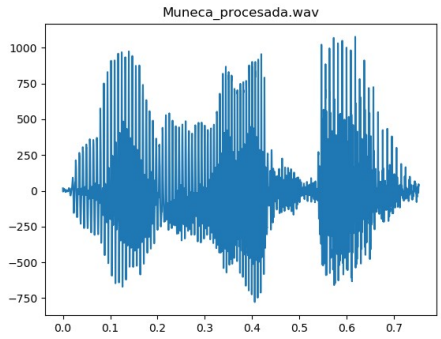
La idea general de esta primera práctica es recibir una señal de entrada la cual transformaremos aplicándole un algoritmo, de esta manera obtenemos una señal de salida distinta. Además en nuestro caso también servirá para comenzar a familiarizarse con la sintaxis de Python para aquellos alumnos que no tengan experiencia previa con este lenguaje.

Tareas a Realizar

Tarea 1: Implementar un filtro FIR

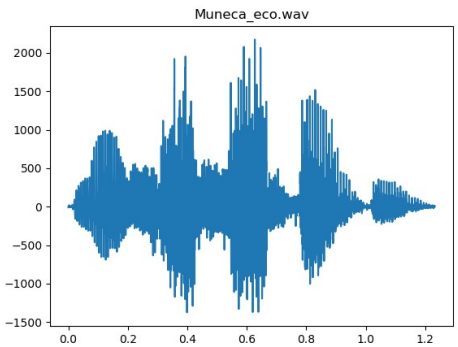
Este filtro nos servirá para calcular el promedio de las 3 últimas muestras entrantes. Una vez se ha aplicado el filtro la señal de voz que obtenemos como resultados “muñeca” posee mucho más ruido que la señal original y un mayor volumen. Esto se debe a que al aumento de la ganancia respecto a la señal original. A continuación muestro dos gráficas comparando la señal original (izquierda) y la señal una vez se le ha aplicado el filtro (derecha), podemos ver que a pesar de parecer muy similares la ganancia en la segunda señal es mayor que en ella primera generando ese ruido.





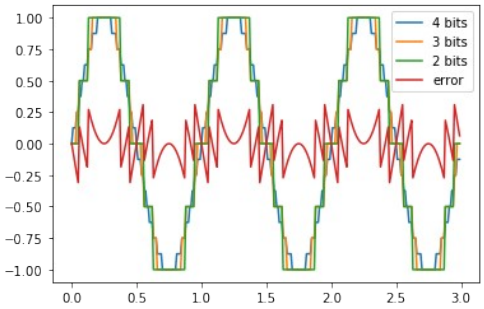
Tarea 2: Implementar un sistema de eco o delay

Para poder generar el efecto de eco voy a aplicar un convolve a la señal “retorciéndola” o mezclándola con ella misma y dándole el retraso que uno considere oportuno (en mi caso 250). Una vez hecho esto obtenemos una señal de voz que comienza igual de la del apartado anterior pero a la mitad de esta vuelve a reproducirse la señal por encima dando la sensación de un efecto de eco. En la grafica se puede ver como se repite el intervalo de la señal 3 veces.



Tarea 3: Cuantización uniforme

Ahora vamos a cuantizar la señal. Para ello sigo los pasos del guion y genero una gráfica con la longitud de onda de la señal cuantizada y el error. Aunque a la hora de ejecutar el programa se generen gráficas para cuantos de 4, 3 y 2 bits voy a mostrar una única grafica con todos debido a que no tendría espacio suficiente si pusiera tres gráficas.



En la gráfica generada para 4 bits el error oscila entre (**-0.1** , **0.1)** mientras que con 3bits y 2 bits oscila entre (**-0.12** , **0.12) (-0.25, 0.25)** respectivamente. Mientras que con 4 bits la calidad de la señal es más o menos aceptable, a medida que cambiamos el cuanto a 3 y 2 bits podemos comprobar como la continuidad de la señal va decayendo y el error va aumentando generando una señal con una calidad inaceptable. En conclusión, a menor número de bits peor será la calidad de la señal y mayor será el error.

Para poder ver esto de otra manera que no sea mediante una gráfica vamos a calcular la cuantización de la señal en decibelios (SNR) y comprobar como coinciden los valores si lo calculamos mediante el SNR experimental y el SNR teórico según se indica en el guion. Según vamos cambiando el intervalo de cuantización podemos observar como los valores obtenidos para cada medición son prácticamente iguales (los valores obtenidos se pueden ver por la terminal al ejecutar el programa)

Tarea 4: Cuantización uniforme señal de voz

Ahora realizaremos la cuantización con 8, 6 y 4 bits teniendo en cuenta que la señal se encuentra en el rango +-2048 (SNR 8 bits → 36.06 , SNR 6 bits → 23.7, SNR 4 bits → 12.3 decibelios). En cuanto a los resultados obtenidos y a las señales de voz generadas podemos comprobar como con 8 bits la señal está más “limpia” y con menor volumen que las señales anteriores, en concreto ya no llega a ser molesto el ruido generado al principio (“mu”) y al final (“ca”) de la señal. Según vamos disminuyendo el número de bits la calidad ira cayendo y el error aumentará de todas formas sigue siendo medianamente aceptable.

Tarea 5: Cuantización mediante Ley Mu

Ahora realizaremos la cuantización basándonos en la Ley Mu la cual debería de generarnos mejores resultados. A pesar de haber intentado seguir los pasos del guion y buscando por internet creo que este apartado no está bien hecho. De todas formas comento los resultados obtenidos y lo que creo que debería de pasar (los valores obtenidos se pueden ver por la terminal al ejecutar el programa).Los resultados obtenidos son prácticamente iguales y la señal de voz generada también. Lo que creo que debería de pasar es que los resultados obtenidos fueran ligeramente superiores y este pequeño cambio aclarase un poco más la señal pero no de una manera sustancial.

Opinión personal

En general la práctica me gusto como introducción al procesamiento digital de señales y Python. Una pega es que el guion de la práctica en algunas partes es un poco difícil de entender y tuve que estar bastante por internet investigando pero bueno supongo que eso también es parte de la práctica.