

Práctica 3:

Cambio de la frecuencia de muestreo en tiempo discreto

Curso 2005/2006

1. Introducción

En esta práctica se presenta la realización práctica del diezmado y la interpolación, herramientas necesarias para decrementar y incrementar respectivamente la frecuencia de muestreo de una señal discreta con procesado en tiempo discreto.

La señal sobre la que se realiza el cambio de frecuencia de muestreo es una señal de audio que incluye un trozo de voz seguido de una parte con instrumentos de percusión. La aplicación sobre esta señal se incluye para escuchar los efectos derivados del solapamiento o *aliasing* en una señal de audio.

La señal de audio con la que se modifica la tasa de muestreo se guarda en el fichero *chieftains_10s.wav*. En MATLAB para manejo de señales de audio existen las siguientes funciones que es necesario analizar en esta práctica: `wavread()`, `wavplay()`, `wavwrite()`, `sound()` y `soundsc()`.

Además, en esta práctica se proporcionan 2 funciones de MATLAB: una para dibujar la transformada de Fourier de una señal discreta `espectro()`; y otra para calcular los coeficientes de un filtro FIR de fase lineal que calcula los coeficientes de los filtros paso bajo que se necesitan `filtros()`.

2. Diezmado

Para la reducción de la tasa de muestreo por un número entero M de una señal discreta se emplea una etapa de diezmado. Básicamente, esta etapa se basa en un diezmador que es un bloque discreto que escoge una muestra de cada M de la señal discreta. Sin embargo, este procedimiento puede producir solapamiento o *aliasing* en la señal resultante. La solución es filtrar la señal antes de diezmar, en el caso ideal el filtro es un filtro paso bajo ideal a frecuencia $\frac{\pi}{M}$ donde M es el factor de diezmado. Como conclusión cabe decir que es mejor filtrar la señal que permitir que se produzca

solapamiento aunque esta afirmación se prueba de forma práctica a lo largo de este apartado.

Abra el fichero de audio proporcionado y realice el diezmado con factor M desde 2 hasta 6 sin filtrar previamente la señal. Compruebe en frecuencia el solapamiento que se produce dibujando el espectro de la señal. En este sentido, debido a que la señal de audio es altamente no estacionaria no se recomienda dibujar el espectro de más de 1 segundo de duración de la señal. Escuche las señales resultantes comprobando la distorsión que aparece. Analice el factor de diezmado que produce una distorsión apreciable para la señal de voz en fonemas sonoros y sordos, y en señal de audio para golpes de percusión de alta y baja frecuencia.

Otra importante característica del bloque de diezmado es que es un sistema variante en el tiempo. Compruebe que si la señal de entrada se retrasa por un número entero n_0 que no es múltiplo del factor de diezmado M la señal de salida no se corresponde con $x[n - n_0]$.

El siguiente paso es aplicar el filtrado antes del diezmado, de forma que se evite el solapamiento a cambio de filtrar la señal de entrada. Realice el diezmado con factor M de 2 hasta 6 filtrando previamente la señal. Compruebe que la distorsión audible de la señal de salida es menos molesta que para la señal con solapamiento.

3. Interpolación

La interpolación se utiliza para incrementar la tasa de muestreo de una señal en tiempo discreto por un factor L . El primer bloque de una etapa de interpolación consiste en un relleno con $L - 1$ ceros de la señal por cada muestra de entrada. Dibuje el espectro de salida tras aplicar este bloque con un factor de interpolación desde $L = 2$ hasta $L = 6$. Escuche la señal de salida para comprobar el efecto de las copias espectrales en frecuencia.

Para eliminar estas copias espectrales se aplica un filtro paso bajo, que en teoría es ideal. Filtre la señal de salida de la interpolación y dibuje la respuesta del filtro con la función `freqz()`. Compruebe el resultado en frecuencia tras el filtrado, así como la calidad perceptual de la señal de salida.

Se puede observar que el número de ceros del filtro necesario es muy elevado, de forma que la complejidad del filtro también lo es. Una solución para sistemas con baja capacidad de procesamiento es aplicar un interpolador más sencillo. Un interpolador muy sencillo es el interpolador lineal para las muestras que se ponen a cero inicialmente. Inicializar el filtro interpolador a un interpolador lineal de la forma:

$$[B] = [1:L \ L-1:-1:1]/L;$$

Compruebe la respuesta en frecuencia del filtro y el resultado en la señal interpolada. Escuche la señal tras la interpolación lineal analizando la distorsión producida por la característica del filtro.

4. Cambio de la frecuencia de muestreo por un número racional

Los bloques de interpolación y diezmado se puede utilizar en cascada con el propósito de cambiar la frecuencia de muestreo por un factor entero L/M .

Utilizando bloques de interpolación y diezmado con sus filtros paso bajo correspondientes cambie la frecuencia de muestreo del fichero de audio inicial de $44,1KHz$ a: $48KHz$, $32KHz$, $16KHz$ y $8KHz$. Compruebe que se puede eliminar el filtro paso bajo menos restrictivo de los bloques en cascada interpolación y diezmado; así como la calidad de la señal de salida.

Realizar el mismo proceso con las funciones de MATLAB apropiadas para este cometido: `interp()`, `decimate()` y `resample()`.

5. Resumen

Tras realizar la práctica, deberá haber aprendido:

- Qué funciones dispone MATLAB para trabajar con las señales de audio.
- Entender el funcionamiento en tiempo y frecuencia de los bloques de interpolación y diezmado.
- Conocer cómo realizar cambios de la velocidad de muestro de forma eficiente en MATLAB.