

Práctica 5: Diseño de filtros digitales. Filtrado de acordes mediante un filtro de Butterworth

Las notas musicales son señales senoidales donde la frecuencia de la onda determina el sonido de la escala musical que queremos reproducir. La nota DO en la escala primaria se genera como una senoide con una frecuencia de 65,4064 Hz. En Matlab se genera como, $DO1 = \sin(2 * \pi * freq_{DO1} * t)$. Esta señal la podemos escuchar en la tarjeta de sonido del ordenador usando la función de Matlab `sound(señal, fm)`, donde *señal* es la señal que queremos escuchar y *fm*, la frecuencia de muestreo de la señal.

NOTA: La frecuencia de muestreo que usa por defecto Matlab para reproducir los sonidos es de $fm = 8192$ Hz, y es la frecuencia de muestreo que usaremos en todo el ejercicio; el periodo de muestreo T será por tanto su inversa.

1.- Representa gráficamente la señal correspondiente a la nota DO en un intervalo de tiempos entre $[0 \dots 6]$ s.

2.- Escucha la nota DO usando la función `sound` de Matlab.

Un acorde musical está compuesto de varias notas que suenan al mismo tiempo. Desde el punto de vista de señales lo podemos ver como la suma de distintas senoideas a distintas frecuencias, donde cada frecuencia se corresponde con una nota de la escala musical.

3.- Sabiendo las frecuencias de las siguientes notas musicales:

- DO (escala primaria): $DO1 = 65.4064$ Hz
- FA (escala secundaria): $FA2 = 174.6141$ Hz
- LA (escala terciaria): $LA3 = 440.0000$ Hz
- DO (escala cuaternaria): $DO4 = 523.2521$ Hz

genera el acorde (DO1,FA2,LA3,DO4), represéntalo gráficamente en un intervalo de tiempos entre $[0 \dots 5]$ s y escucha como suena el acorde.

NOTA: Recuerda que la función `sound` requiere que la señal que le pasamos, como variable *señal*, esté normalizada entre $[-1, \dots, 1]$. Consultar en la ayuda de Matlab los parámetros de dicha función.

4.- Si queremos escuchar sólo los sonidos de un determinado acorde que posean una frecuencia inferior a una frecuencia determinada debemos filtrar el acorde original mediante un filtro pasa-baja.

Para diseñar un filtro digital de Butterworth pasa-baja usando la transformación bilineal (ver la figura 1) tenemos que realizar los siguientes pasos:

I Convertir las especificaciones dadas en el dominio digital (Ω) a especificaciones analógicas (ω) usando las fórmulas vistas en Teoría:

$$\omega = \frac{2}{T} \tan \frac{\Omega}{2} \quad (1)$$

II Determinar el orden n del filtro a partir de las características especificadas,

III Diseñar el filtro en continuo.

IV Discretizar la función de transferencia en continuo.

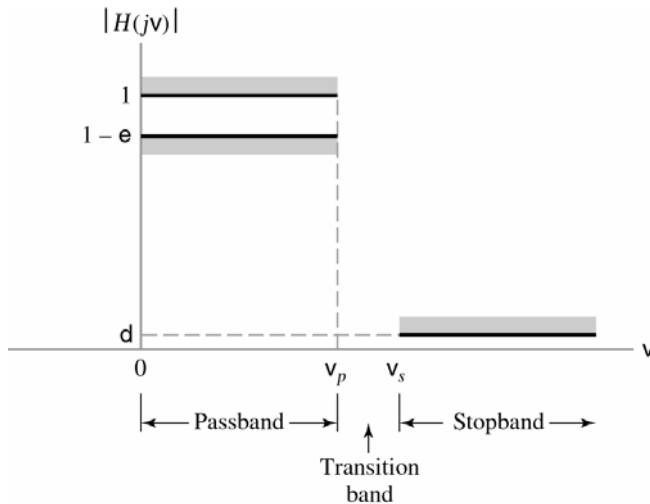


Figura 1: Filtro pasa-baja

$$1-e \leftrightarrow 1-\varepsilon \quad ; \quad d \leftrightarrow \delta \quad ; \quad V_p \leftrightarrow W_p \quad ; \quad V_s \leftrightarrow W_s$$

Diseñaremos un filtro de Butterworth, con las siguientes especificaciones digitales:

Frecuencias de paso y de rechazo : $W_p=0,1912$ rad/s $W_s=0,2669$ rad/s
 Tolerancias de las bandas de paso y de rechazo : $R_p=1$ dB $R_s=6$ dB

5.- Determinar el orden del filtro.

Utilizar la función de MATLAB `[n,Wn]=buttord(Wp,Ws,Rp,Rs,'s')`

6.- Diseño del filtro en continuo.

Utilizar la función de MATLAB `[num,den]=butter(n,Wn,'s')`

7.- Discretización del filtro continuo.

Usar la Transformación Bilinear de Tustin. Esta transformación está implementada en Matlab mediante la función `[numd, dendl]=bilinear(num,den,fm)`

8.- Comprueba que el filtro digital que has diseñado se comporta como un filtro pasa-baja. Para ello dibuja el Diagrama de Bode del filtro analógico y digital, para el digital. Matlab posee la función `dbode(numd, dendl, T)`.

9.- Filtrado de la señal.

Filtra el acorde original con el filtro digital que acabas de diseñar. Para filtrar la señal usa la función de Matlab `[sfilt]=filter(numd, dendl, signal)`, donde `signal` es la señal que queremos filtrar. La señal filtrada nos la devuelve la función en `sfilt`.

10.- Comprobación del filtrado.

Podemos comprobar que el filtrado se realizó correctamente de tres formas:

a.- Auditivamente: envía primero la señal filtrada mediante el filtro que has diseñado a la tarjeta de sonido, y a continuación envía la señal filtrada que se desearía obtener, la cual la generáis vosotros (sabiendo que la frecuencia de corte del filtro que habéis diseñado es aproximadamente de 300Hz).

b.- Gráficamente: representa gráficamente la señal filtrada mediante el filtro diseñado, y la señal filtrada real.

c.- Representa los espectros de frecuencias de todas las señales (**fft**).