Señales y Sistemas

Práctica 5

Señales y sistemas en tiempo discreto

ÍNDICE

1. Sistemas discretos LTI.
2. Efectos de audio.
   2.1. Eco infinito.
   2.2. Eco finito.

En esta práctica analizaremos algunos sistemas sencillos y su interconexión en cascada. Filtraremos señales con sistemas discretos lineales e invariantes en el tiempo (LTI) descritos en ecuaciones en diferencias y por su respuesta al impulso. Para terminar, implementaremos la simulación de varios efectos de audio sobre una señal de voz.

1. SISTEMAS DISCRETOS LTI

Aquí estudiaremos la interconexión en serie de dos sistemas lineales e invariantes en el tiempo (Figura 1). También introduciremos la función **`filter`** para calcular la respuesta de un sistema LTI cuando es conocida su ecuación en diferencias.

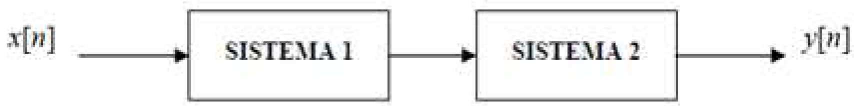

Figura 1: Asociación en cascada o serie de dos sistemas LTI.

La función **`y=filter(b,a,x)`** filtra la señal en el vector X con el filtro descrito por el vector de coeficientes **b** y el vector de coeficientes **a**, donde **`a(i)`** y **`b(i)`** provienen de una ecuación en diferencias de la forma:

```
a(1)·y[n] + a(2)·y[n-1] + ... + a(N+1)· y[n-N] =
b(1)·x[n] + b(2)·x[n-1] + ... + b(M+1)· x[n-M]
```

Si **`a(1)`** es igual a cero, la función `filter` devuelve un error.

El vector de salida **y** será de igual longitud que el vector de entrada **x**.

a) Utilizando el comando `filter`, calcula y representa la respuesta $y[n]$ de un sistema LTI tipo **IIR** descrito por la siguiente ecuación en diferencias,

$$y[n] - \frac{1}{5}y[n-1] - \frac{1}{25}y[n-2] = x[n] + \frac{1}{3}x[n-1],$$

a la excitación:

$$x[n] = 3\cos(0{,}05\pi \cdot n) \cdot \prod\left(\frac{n-10}{50}\right)$$

Considera el intervalo de tiempo $0 \leq n \leq 100$.

b) Calcula, usando también la función `filter`, y representa la respuesta al impulso $h_{IIR}[n]$ del sistema anterior.

c) Supón que después se conecta en cascada un segundo filtro de tipo **FIR** (Figura 1) cuya respuesta al impulso es:

$$h_{FIR}[n] = \delta[n] + 0{,}8\delta[n-10] + 0{,}64\delta[n-20],$$

Calcula y representa en Matlab cuál será la respuesta final de la asociación sobre la señal $x[n]$ utilizando la función `filter` para filtrar la salida obtenida en el apartado a).

d) Calcula la respuesta al impulso equivalente de la asociación en cascada, usando la operación de convolución (comando **conv**). Representa el resultado en el intervalo $0 \leq n \leq 100$. En la representación ten en cuenta que la duración del vector que se obtiene con la convolución de $h_{IIR}[n]$ y $h_{FIR}[n]$ es más largo. En concreto, recuerda que se cumplirá,

```
length(conv(hIIR,hFIR)) == length(hIIR)+length(hFIR)-1
```

e) Busca otro modo de calcular la respuesta al impulso equivalente sin usar `conv`.
f) ¿Qué ocurre si cambiamos el orden de los sistemas en cascada? ¿Cambia la respuesta final? ¿Por qué?

2. EFECTOS DE AUDIO

A diferencia de las prácticas anteriores de carácter básico, en este apartado se desarrolla una aplicación más avanzada de tratamiento digital de la señal. Ahora se utilizará la teoría de sistemas LTI discretos para la implementación de efectos típicos de audio. En principio se dará una explicación intuitiva de ellos para, posteriormente, implementarlos.

2.1 Eco infinito

El sistema digital de la figura 2 muestra el modo más sencillo de implementar un eco infinito. La señal final debe ser la suma de la señal original más su versión retardada y atenuada.

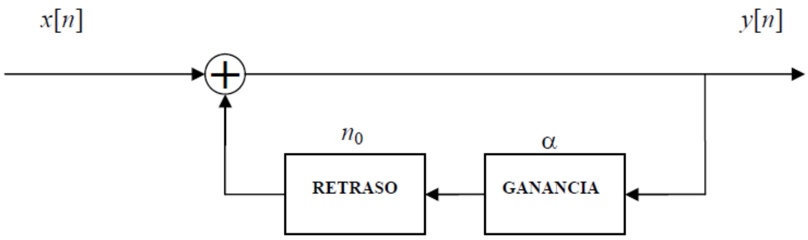

Figura 2: Esquema de implementación discreta de un eco infinito.

La ecuación en diferencias que define el sistema es entonces:

$$y[n] = x[n] + \alpha \cdot y[n - n0] \ ,$$

donde α< 1 representa la atenuación sufrida por el eco respecto a la señal original y $n0 > 0$ es el número de muestras que equivale al retardo temporal del eco. Ambos efectos se deben a la diferencia entre los caminos recorridos por la señal directa y su eco.

Si $t0$ es el tiempo de retardo en segundos del eco, se puede obtener un valor aproximado de $n0$ haciendo:

```
n0=fix(t0*fs)
```

donde `fs` es la frecuencia de muestreo.

A la hora de implementar este eco en Matlab hay que introducir tres parámetros: la señal a filtrar, la frecuencia de muestreo f_S en Hz, el tiempo de retardo del primer eco $t0$ en segundos y el factor de ganancia α (que implicará una atenuación por ser menor que 1). Crea la siguiente función y pruébala sobre muestras de voz o sobre cualquier otra señal digital de audio. Si no dispones de ninguna, puedes usar el fichero *voz.wav* en el que se ha grabado la palabra "casa", para lo cual deberá emplearse la función **`wavread`**:

```
[x,fs]=wavread('nombre_fichero.wav');
```

A continuación se facilita la función que simula el efecto del eco infinito sobre la señal:

```
function y=ecoinfinito(x,fs,alfa,t0)
%ECOINF: Implementación de un eco infinito
%Uso: Y=ecoinf(X,FS,ALFA,T0)
% X vector de muestras originales
% FS frecuencia de muestreo en Hz
% ALFA factor de ganancia (será menor que 1)
% T0 retardo en segundos
n0=fix(t0*fs);
x=x(:).'; %asegura que vector x sea fila
x=[x, zeros(1,4*n0)]; %se alarga x para 4 ecos
b=[1];                     % vector de coeficientes b
a=[1 zeros(1,n0-1) -alfa]; % vector de coeficientes a
y=filter(b,a,x);
```

Para escuchar la señal original **x** y el resultado **y** puedes emplear la orden **sound** de Matlab. Realiza una primera prueba con `t0=0.5` y `alfa=0.8`. Después modifica estos dos parámetros y observa cómo afecta su valor al resultado.

2.2 Eco finito

De una forma intuitiva, el eco es el resultado de sumar a una determinada señal una versión retardada y atenuada de sí misma que volverá a recorrer de nuevo el mismo camino y sufrirá de nuevo el mismo retardo y atenuación, etc, en un ir y venir sin fin. Es decir:

$$y[n] = x[n] + \alpha \cdot x[n - n0] + \alpha^2 \cdot x[n - 2\,n0] + \alpha^3 \cdot x[n - 3\,n0] + \cdots$$

Consideraremos ahora en el procesado sólo <u>dos</u> términos del eco, esto es,

$$y[n] = x[n] + \alpha \cdot x[n - n0] + \alpha^2 \cdot x[n - 2\,n0].$$

Al truncar la ecuación en diferencias estamos describiendo otro tipo de eco, el <u>eco finito</u>. En esta aproximación a un número finito de ecos se está suponiendo que a partir de cierto eco en adelante éstos se atenúan tanto que no afectan significativamente a la señal.

Elabora una función, `y=ecofinito(x,fs,alfa,t0)`, que implemente el sistema correspondiente a este eco truncado.

Compara el resultado obtenido aplicando la función de eco infinito y la de eco finito sobre una señal de audio.