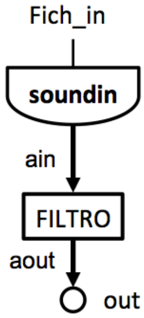
Práctica 6: Filtros en audio digital

En esta práctica aplicaremos conceptos vistos en teoría para programar filtros digitales implementados en el tiempo, procesando directamente las señales, sin pasar al dominio frecuencial. Se propondrán ejercicios para diseñar y construir filtros pasa-baja, pasa-alta, pasa-banda y peine, en versiones FIR e IIR.



Como regla general, en cada instrumento se creará una variable de tipo audio de nombre *ain* para la señal de entrada ($x[n]$), leída con el operador **soundin**, y una señal de nombre *aout* ($y[n]$) que será enviada a la salida con el operador **out**. De esta manera, a un filtro como el de la izquierda, le correspondería el esquema de la derecha y esta codificación en Csound:


```
ain     soundin   "F_in.wav"
aout    FILTRO    ain
        out       aout
```

FILTRO no tiene por qué ser un operador, sino que puede tener que realizarse mediante una serie de operaciones más o menos numerosas, dependiendo de la complejidad y tipo de filtro.

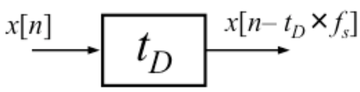
UNIDADES DE RETARDO EN CSOUND

Para diseñar los filtros de esta práctica utilizaremos retardos de 1 o más muestras. En Csound estos retardos se implementan mediante los operadores **delay1** y **delay**, respectivamente.

delay1 : Retrasa una señal en una muestra (en un tiempo equivalente a 1 periodo de muestreo). Si la de entrada es $x[n]$, la de salida sería $x[n–1]$:


```
aRetrasada  delay1  aSenal
```

delay : Retrasa una señal un tiempo t_D especificado <u>en segundos</u>. t_D segundos son D muestras separadas T_s segundos cada una: $t_D = D\,T_s \rightarrow D = t_D/T_s = t_D f_s$ (ver figura).




```
aRetrasada  delay   aSenal, itretardo
```

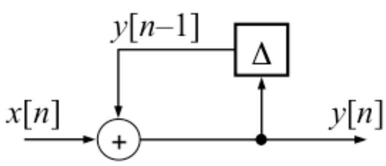
delay1 *ain* es equivalente a **delay** *ain*, $1/sr$ puesto que $1/sr = 1/f_s = T_s$, es decir, es un periodo de muestreo.

USO DE LA REALIMENTACIÓN EN CSOUND

Los Filtros IIR usan realimentación: las salidas calculadas anteriormente vuelven "hacia atrás" para usarse como entradas la vez siguiente. Csound es capaz de usar realimentaciones de una manera muy sencilla pero que puede resultar poco intuitiva respecto al uso habitual en los lenguajes de programación.

Hay que recordar que el código de un instrumento es como si estuviera metido dentro de un bucle no visible (que itera *kr* veces por segundo). En Csound, las variables que están <u>a la derecha</u> de los operadores, tienen los valores del ciclo anterior respecto a los valores <u>a la izquierda</u>. Así, una expresión *aout* = *aout*, correspondería a $y[n] = y[n-1]$ (si **ksmps** = 1), por lo que de esta sencilla manera podemos implementar la recursividad de los filtros IIR.

Por ejemplo, el esquema de esta figura corresponde con la ecuación $y[n] = x[n] + y[n-1]$. Supongamos que $x[n]$ viene de leer un fichero. Entonces, la implementación en Csound de este esquema sería (la tercera línea):




```
aout   init     0
ain    soundin  iFichero
aout   =        ain + aout   ;   (y_n = x_n + y_n-1)
```

Es cierto que *ain* también está en el lado derecho, pero la segunda sentencia ya le da el nuevo valor (que es $x[n]$) a *ain* al leerlo del fichero y, por lo tanto, el anterior ($x[n–1]$) ya ha quedado sustituido por este.

Al hacer esto tenemos que *aout* aparece a la derecha de un operador sin haber aparecido alguna vez antes a la izquierda, por lo que no estará creada. Para evitarlo, se pone la orden **init** al principio, lo cual crea la variable de audio *aout* e inicializa todas sus muestras a ceros (por el 0 a la derecha).

Filtros pasa-baja (FPB)

En los filtros pasa-baja, debes recordar que la suma de los valores absolutos de los coeficientes de los filtros que diseñes debe ser la unidad. De esta manera, te aseguras que la amplitud (de pico) de la onda de salida sea igual a la de entrada, no amplificas ni atenúas la señal.

Pasa-baja FIR

En este tipo de filtros, la primera frecuencia anulada es $\boxed{f_x = f_s \,/\, (M+1)}$, siendo M el número de muestras retrasadas usado (el *orden* del filtro). Cuantas más uses, mayor será la pendiente de caída de $H(f)$, pero serán necesarios más cálculos y aparecerán lóbulos laterales apreciables junto al principal.

El caso más sencillo es

$$y[n] = a_0 \cdot x[n] + a_1 \cdot x[n-1] \ ,$$

con $a_0 = a_1 = 1/2$, pero su efecto es tan sutil que apenas se aprecia (la frecuencia que anula es $f_x = f_s \,/\, 2$ = Nyquist, ver figura). Para conseguir filtros más restrictivos hay que ir añadiendo retrasos y sumando las muestras retrasadas.

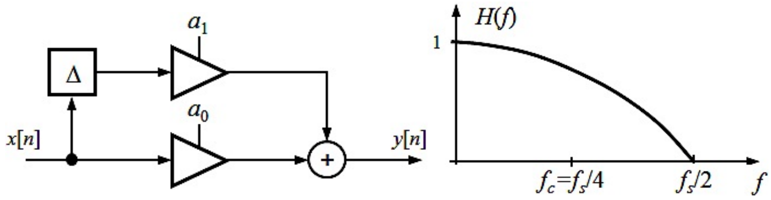


Diagrama de bloques de este FPB FIR y su respuesta en frecuencia.

Pasa-baja IIR

La ecuación del FPB de tipo IIR más sencillo es:

$$y[n] = (1-a) \cdot x[n] + a \cdot y[n-1] \ \ .$$

Al aumentar el coeficiente del lazo de realimentación, a, la f_c disminuye, aunque debe ser siempre menor que 1 o el filtro será inestable. El coeficiente $1-a$ para $x[n]$ garantiza que la suma de los dos sea la unidad.

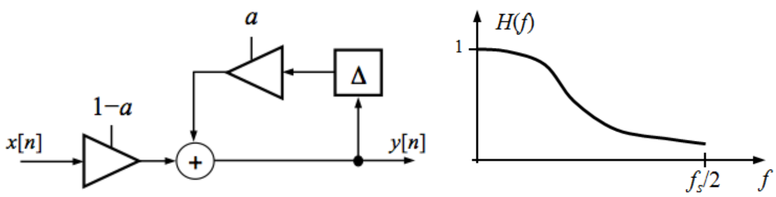


Diagrama de bloques de este FPB IIR y su respuesta en frecuencia.

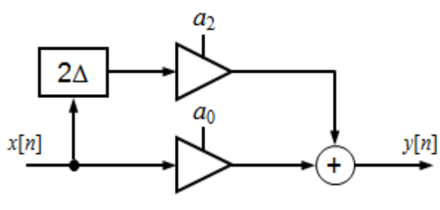
Filtros pasa-alta (FPA)

La estructura de los filtros pasa-alta es idéntica a los pasa-baja (tanto FIR como IIR), pero las muestras con retardos impares (n, $n-2$, $n-4$,… son pares; $n-1$, $n-3$, … son impares) se restan en vez de sumarse (se suman con el coeficiente negativo). Por ejemplo, un pasa-baja $y[n]$ = 1/2 $x[n]$ + 1/2 $x[n-1]$ se convierte en pasa-alta si hacemos $y[n]$ = 1/2 $x[n]$ – 1/2 $x[n-1]$.

De manera equivalente a lo que pasa en el pasa-baja FIR, cuanto mayor sea el número de muestras retardadas (el orden), mayor será la f_c, con la misma consecuencia de aparición de lóbulos laterales que en los FPB. La condición de normalización sigue vigente: la suma de sus valores absolutos debe valer uno.

Filtros pasa- y elimina-banda

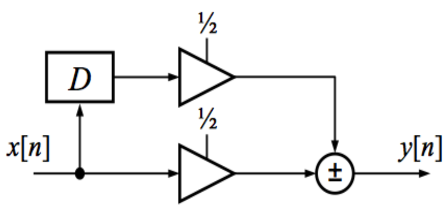
Si en el esquema del filtro pasa-baja FIR anterior cambiamos el retardo de una muestra por un retardo de orden 2, correspondiente a la ecuación $y[n] = a_0 \cdot x[n] + a_2 \cdot x[n-2]$ (ver figura), entonces cambia su comportamiento. Si $a_0 = a_2 = 1/2$, este nuevo filtro se comportará como un elimina-banda y si $a_2 = -a_0 = -1/2$, lo hará como un pasa-banda. En ambos casos, la frecuencia de corte será $f_c = f_s/4$.



Filtros peine FIR

Los filtros peine FIR crean muescas (ceros) en la $H(f)$, en frecuencias relacionadas armónicamente en el espectro de la señal de entrada. Su ecuación, en el caso de los FIR, es:

$$y[n] = \tfrac{1}{2}x[n] \pm \tfrac{1}{2}x[n-D] \ \ .$$

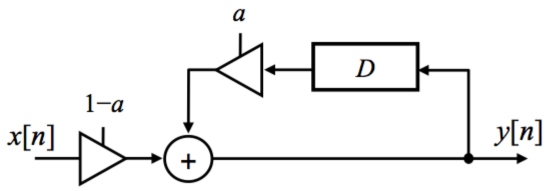


Cuando *se usa la resta* se producirán ceros (f_x) en 0, en $f_s \,/\, D$ y en sus múltiplos enteros hasta Nyquist ($\leq f_s/2$), quedando $H(f) = 1$ (frecuencia inalterada) en las frecuencias intermedias entre cada dos ceros: $f/2$, $3f/2$, $5f/2$, ... Si *se usa la suma*, las posiciones de los ceros y de $H(f) = 1$ se intercambian, quedando los ceros en $f_x = f_s / 2D$.

La relación entre el número de muestras de retardo, D, que usa el filtro y el tiempo de retardo especificado es $D = t_D \times f_s$ (redondeando a entero). Si este tiempo es tal que los retrasos son $t_D > 0{,}1$ ms entonces se empiezan a percibir al oído las múltiples cancelaciones de frecuencias.

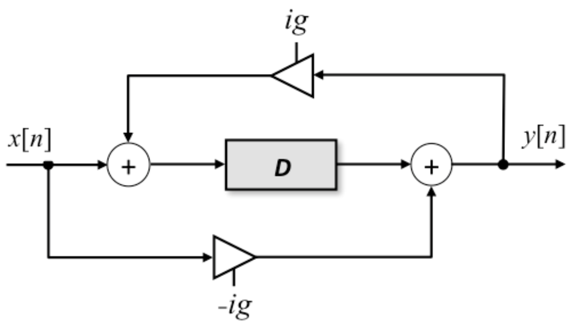
Filtros peine IIR

Estos filtros crean picos (polos) en la $H(f)$. La ecuación de un filtro peine IIR es igual que la del FIR anterior salvo por el hecho de lo que se suma a la entrada es la señal de salida retrasada realimentada. La ecuación ahora es $y[n] = (1-a)\, x[n] + a\, y[n-D]$.



Filtros pasa-toda

Estos filtros no alteran de manera apreciable las amplitudes de los parciales de la señal, pero desplazan sus fases. En sonidos estacionarios no causan ningún efecto, pero "colorean" levemente el sonido en los sonidos con rápidos transitorios. Se implementan como dos peines juntos: uno FIR y otro IIR, ambos con el mismo retardo y con ganancias iguales, pero de signo contrario. Su esquema es el mostrado en la figura. Los parámetros son el tiempo de retardo t_D y la ganancia ig.



ANÁLISIS DE LOS RESULTADOS

Los resultados del filtrado se pueden *oír* (cómo cambia nuestra percepción del sonido) o *ver* (qué ha pasado con las señales o con sus espectros). Algunos ejercicios están diseñados para oír los resultados y otros para verlos.

Para oír el efecto de los filtros se proporcionaran sonidos muy ricos en parciales, para percibir el efecto de la pérdida de determinadas bandas de frecuencia por la actuación de los filtros. En esos casos, puedes escuchar comparativamente los ficheros original y filtrado abriendo ambos con **Audacity** y oyéndolos por separado con el botón **[Solo]** en una pista y en la otra.

Ver el efecto significa ver cómo es la respuesta en frecuencia del filtro. Para ello se proporcionarán ficheros con "señales" diseñadas <u>para ver</u> los efectos causados en ellas y comprender qué está pasando. El fichero `f_var.wav` contiene un barrido lineal de 1 s entre frecuencias de 0 Hz y 22.050 Hz. Por lo tanto, al avanzar el tiempo *se ve* el comportamiento del filtro para la frecuencia que corresponde a ese instante. A cada tiempo t del fichero, le corresponde una frecuencia $t \times f_s / 2$.

Por ejemplo: en $t = 0.5$ s (duración/2), como el barrido es lineal, la señal tendrá $f = 0{,}5 \times 22.050 = 11.025$ Hz (ver figura). Por tanto, observar la envolvente de esta señal filtrada es una forma sencilla de ver la respuesta en frecuencia $H(f)$ del filtro sin tener que calcular su espectro.

<u>Nota:</u> como usaremos en esta práctica siempre $f_s = 44100$ Hz, entonces Nyquist será 22050 Hz.