

Tema 5

Procesadores de efectos

Operaciones o filtros que corresponden a técnicas de procesamiento que se aplican sobre algunas propiedades para modificarlas, a veces, es inevitable al modificar una alterar otra.

Procesadores de altura

Tienen como objetivo la detección o modificación de la altura de los sonidos, por lo que la base de los métodos pasa por encontrar su frecuencia fundamental y, en su caso, alterarla y cambiar los valores de las frecuencias parciales de sus espectros.

Detección de la altura

Objetivo conocer el valor de la frecuencia fundamental de un sonido y seguir su evolución temporal. Una señal que contiene un sonido musical es analizado para detectar qué notas, con sus alturas y duraciones, han originado dicho sonido. Es una **conversión de audio a partitura**.

Dos categorías:

Busqueda del periodo fundamental del sonido en el dominio temporal (luego se convierte en la frecuencia fundamental como $f_0 = 1/T_0$),

busqueda de la f_0 directamente en el espectro de la señal.

Transposición

Consiste en cambiar la altura del sonido haciendo que sea más agudo o más grave, manteniendo las relaciones entre las frecuencias de los parciales

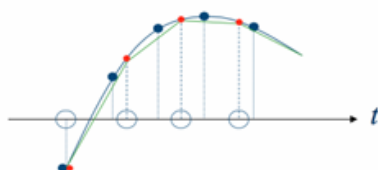
Ajustes microtonales: modificar levemente la frecuencia.

Intervalos tonales: trasponer ascendente o descendente una altura

Ejemplo. Si tenemos un sonido cuyo espectro consta de los cuatro primeros armónicos con frecuencias 50 (fundamental), 100, 150 y 200 Hz; su transposición en una sexta menor ascendente equivale (en la afinación justa) a multiplicar todas ellas por $8/5$, dando lugar a 80, 160, 240 y 320 Hz; donde la proporción de f_0 , $2f_0$, $3f_0$ y $4f_0$ entre las frecuencias se mantiene. □

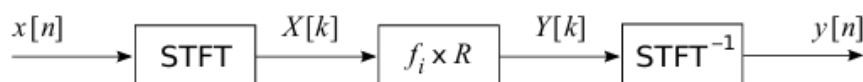
Para hacer esto en el dominio temporal, se reproduce la señal a velocidad distinta de la que fue grabada o muestreada. Si usamos menos muestras de las tomadas (ascendente) o más (descendente)

Ejemplo. Si se toman 4 de cada 5 muestras, las frecuencias parciales de la onda se verán elevadas en $5/4$ de las originales; es decir, habremos transpuesto una tercera ascendente (ver Fig. 5.2). Esta técnica produce un cambio de la frecuencia de muestreo efectiva. En el caso de transposiciones ascendentes hay que llevar cuidado porque la nueva $f'_s < f_s$ y, por lo tanto, también lo será la nueva Nyquist. En este ejemplo, $f'_s = \frac{4}{5}f_s$ y Nyquist será $f'_s/2 = \frac{2}{5}f_s$. Así, todas las frecuencias del espectro de la señal que se hallaran entre $2f_s/5 < f < f_s/2$ causarían *aliasing* y deberían ser filtradas previamente para evitarlo. □



grado del intervalo	relaciones pitagóricas	relaciones justas	ejemplo con notas
unísono	1 : 1	1 : 1	DO – DO
segunda menor	256 : 243	16 : 15	DO – RE \flat
segunda mayor	9 : 8	9 : 8	DO – RE
tercera menor	32 : 27	6 : 5	DO – MI \flat
tercera mayor	81 : 64	5 : 4	DO – MI
cuarta justa	4 : 3	4 : 3	DO – FA
cuarta aumentada	729 : 512	45 : 32	DO – FA \sharp
quinta disminuida	1024 : 729	64 : 45	DO – SOL \flat
quinta justa	3 : 2	3 : 2	DO – SOL
sexta menor	128 : 81	8 : 5	DO – LA \flat
sexta mayor	27 : 16	5 : 3	DO – LA
séptima menor	16 : 9	7 : 4	DO – SI \flat
séptima mayor	243 : 128	15 : 8	DO – SI
octava	2 : 1	2 : 1	DO _N – DO _{N+1}

Se puede trabajar en el dominio de la frecuencia, calcular el espectro de la señal y luego desplazar sus frecuencias, recuperar la señal modificada mediante la transformada inversa de Fourier



El desplazamiento de cada parcial linealmente proporcional, dependiendo de la relación de transposición deseada

Armonizadores

Producen una o varias señales transpuestas cuya relación armónica con la señal de entrada se fija de antemano y las señales resultantes se mezclan con la original

Ejemplo. Si de un sonido obtenemos dos copias transpuestas con las relaciones $R_{III} = 5/4$ y $R_V = 3/2$, respectivamente, y las sumamos, obtendremos un acorde I + III + V, que es una triada mayor, a partir de la señal monofónica original. □

Vibrato

modulación de altura a baja frecuencia

La técnica para implementar un vibrato sobre una señal cualquiera es insertar una línea de retardo variable controlada por un oscilador de baja frecuencia (LFO, low frequency oscillator). El LFO hará variar t_D , y oscila entre $t_D - \Delta D$ y $t_D + \Delta D$.

Ejemplo. Si queremos aplicar un vibrato de $f_m = 1$ Hz, variando el t_D central de 3 ms en $\Delta D = \pm 2$ ms, el LFO creará una señal moduladora sinusoidal entre -2 y +2 que se sumará a t_D para que el retardo oscile entre 1 y 5 ms, una vez por segundo. En un sistema digital con $f_s = 44.100$ Hz, esto supondrá que el número de muestras del retardo, $D[n]$, se irá modificando periódicamente para retrasar la señal entre $D = 44$ muestras (según $D = t_D/f_s$ cuando $t_D = 0,001$ s) y $D = 220$ muestras para 0,005 s. □

Procesadores de la dinámica

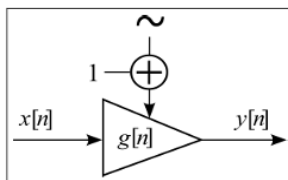
la ganancia es controlada automáticamente por medio de alguna señal externa o por el propio nivel de la señal de entrada

Se basa en un sistema de detección de nivel de amplitud, un algoritmo para obtener un factor de ganancia a partir del resultado del seguidor de envolvente y un multiplicador a la señal de entrada

Procesador	Actúa si...	Umbral	Relación de ganancia
Limitador	$X > \text{umbral}$	Θ_L grande	$R_L = n:1$ con n grande
Compresor	$X > \text{umbral}$	Θ_C grande	$R_C = n:1$ con n pequeña
Expansor	$X < \text{umbral}$	Θ_E pequeño	$R_E = 1:n$ con n pequeña
Puerta de ruido	$X < \text{umbral}$	Θ_P pequeño	$R_P = 1:n$ con n grande

Trémolo

Como el vibrato, pero ahora se modula la Amplitud de la señal



Un oscilador de baja frecuencia genera una onda cuya amplitud A_g será la ganancia máxima a aplicar

$g[n] = 1$, y oscilará entre $1-A_g$ y $1+A_g$ mientras $A_g < 1$

Ejemplo. Supongamos que tenemos una onda de amplitud constante -6 dBFS, que es la mitad del valor máximo que podemos digitalizar. Si aplicamos un trémolo debemos preocuparnos por que la nueva amplitud máxima no supere el límite de amplitud, por lo que no deberá suponer una amplificación mayor de $+6$ dB.

Compresores/Limitadores

La misión de un compresor es reducir la diferencia entre sus amplitudes mínima y máxima

La misión de los limitadores es evitar que se alcance el nivel de saturación de un equipo.

Impide que la amplitud se acerque peligrosamente a un nivel prohibido

Procesamiento de los compresores y limitadores

La reducción del nivel de la señal que se pasa del umbral se controla mediante una tasa de compresión (RC)

El algoritmo se ejecuta cuando su nivel supera el umbral:

$\Delta A_{IN} = X - \Theta_C$, siendo X el nivel en dBFS de la señal,

$$X = 20 \log_{10} \frac{x[n]}{A_{max}}$$

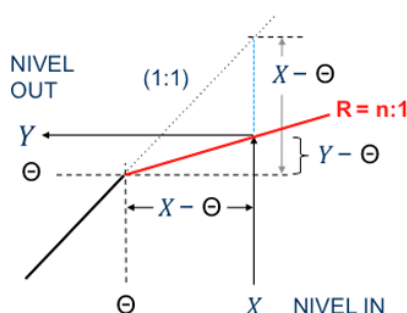
de manera que lo que sobra en:

$$\Delta A_{OUT} = Y - \Theta_C$$

es el resultado de aplicar la compresión en la entrada:

$$\Delta A_{OUT} = \frac{\Delta A_{IN}}{R_C} = \frac{X - \Theta_C}{R_C}$$

$$R_C = \frac{\Delta A_{IN}}{\Delta A_{OUT}} = \frac{X - \Theta_C}{Y - \Theta_C}$$



$$Y = \Theta_C + \frac{1}{R_C}(X - \Theta_C).$$

$$\hat{y} = \hat{x} \left(\frac{\hat{x}}{\theta_c} \right)^{\frac{1}{R_C} - 1} \quad R_C = \frac{\log_{10} \frac{\hat{x}}{\theta_c}}{\log_{10} \frac{\hat{y}}{\theta_c}} \quad g[n] = \frac{\hat{y}}{\hat{x}} \text{ será}$$

$$g[n] = \left(\frac{\hat{x}}{\theta_c} \right)^{\frac{1}{R_C} - 1}$$

Ejemplo. Supongamos que tenemos una señal a la que aplicamos un compresor con $R_C = 4:1$ y $\Theta_C = -8$ dBFS. Mientras el nivel de la señal esté por debajo de los -8 dB, el compresor no actuará y pasará sin ninguna modificación. Cualquier exceso sobre este nivel, se verá dividido por 4. Si en un momento dado, el nivel fuera -3 dB, ello supondría un exceso de $\Delta A_{IN} = X - \Theta_C = -3 - (-8) = +5$ dB sobre el umbral. Por tanto, la reducción será de $\Delta A_{OUT} = \Delta A_{IN}/R_C = 5/4 = +1,25$ dB, y el nivel de salida será $Y = -8 + 1,25 = -6,75$ dB. \square

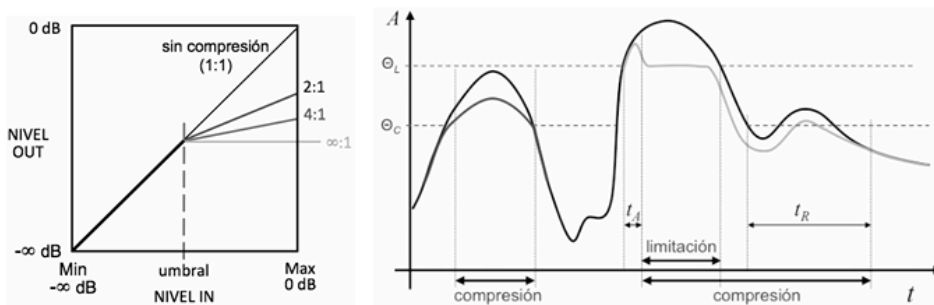


Figura 5.8: (izq.) Función de transferencia de un **compresor**/limitador (con diferentes tasas de compresión: $R_C = 2:1$, $4:1$ y $R_L = \infty:1$). (dch.) Su efecto sobre la envolvente de la señal con diferentes umbrales, con $\Theta_C = 2:1$ y $\Theta_L = \infty:1$, y tiempos de ataque y liberación en el segundo disparo.

Un limitador es exactamente igual que un compresor, pero con una tasa de compresión muy grande.

Aspectos prácticos de estos procesadores

Alteración de los ataques cortos: los compresores incorporan un tiempo de ataque, como el tiempo que tarda en empezar a actuar la reducción de ganancia después de que la señal haya superado el umbral

Encendidos y apagados frecuentes (efecto bombeo): tiempo de liberación, t_R por lo que la compresión sigue actuando un tiempo después de que la señal baje su nivel por debajo del umbral

Entrada brusca de la atenuación: Con rodilla, la compresión entra suavemente porque la tasa de compresión va creciendo suavemente desde un poco por debajo del umbral, hasta alcanzar R_C

Expansores y Puertas

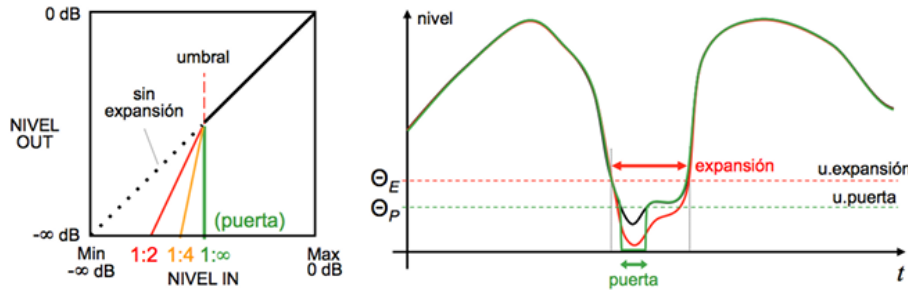
Es aumentar el rango dinámico de unas señales, es decir, la diferencia entre sus niveles máximo y mínimo. La forma de aumentar la diferencia entre esos niveles es reduciendo el volumen de las partes de menor amplitud.

Procesamiento de los expansores / Puertas

Mismo sistema que compresores y limitadores, con dos diferencias

El umbral suele ser mucho menor que el de compresión

Los umbrales de expansión suelen ser $E < -40$ dB



Ejemplo. R_E indica cuánto se atenúa la amplitud del sonido que entra por debajo del umbral. Una relación $R_E = 1:4$ para una señal que esté 3 dB por debajo del umbral producirá una señal que estará 12 dB por debajo de ese umbral en la salida. Si $\Theta_E = -40$ dBFS y el nivel de entrada fuera $X = -43$ dBFS, la salida (según la Ec. (5.3), pero usando los parámetros de expansión) tendría un nivel $Y = \Theta_E + \frac{1}{R_E}(X - \Theta_E) = -40 + 4(-43 + 40) = -40 - 12 = -52$ dBFS. \square

$$\theta_E = A_{\max} \times 10^{\frac{\Theta_E}{20}}$$

donde:

- Θ_E es el umbral en dBFS.
- A_{\max} es el valor máximo absoluto en la escala PCM (32768 para 16 bits).

Uso combinado

Ejemplo. Supongamos que $\Theta_C = -10$ dBFS y $R_C = 4:1$, para una amplitud de la entrada $X = 0$ dB, el exceso de nivel de la entrada será $X - \Theta_C = +10$ dB. La salida será $Y = -10 + 10/4 = -7,5$ dBFS. Es decir, al máximo nivel de la entrada, la salida no supera los $-10(1 - \frac{1}{4})$ dB, por tanto, el limitador nunca entrará en funcionamiento si $\Theta_L \geq -7,5$ dBFS. \square

Ejemplo. Si en un sistema de 16 bits, tenemos $\Theta_E = -66$ dBFS y $R_E = 1:2$. Una amplitud de entrada $X = -76$ dBFS está por debajo del umbral en $X - \Theta_E = -10$ dB. La salida (Ec. (5.3)) será $Y = -76 + 2 \times (-10) = -96$ dBFS. Es decir, cuando la entrada está $(\Theta_E + 96)/2 = 10$ dB por debajo del umbral, la salida será ya el silencio. Si $\Theta_P \leq -76$ dBFS la puerta de ruido nunca actuará. \square

Procesadores del tiempo

Alargan o encogen la duración de una señal sonora, y luego describiremos diferentes tipos de ecos

Estiramiento/Encogimiento: expansión o contracción de la escala de tiempos sin modificar las alturas ni sus espectros

Tanto en el dominio temporal como en el frecuencial. En el dominio temporal existen diversas técnicas, como la reproducción a velocidad variable o la granulación.

Reproducción a velocidad variable: la señal sea reproducida en un tiempo diferente a través de una modificación en el muestreo expresado como:

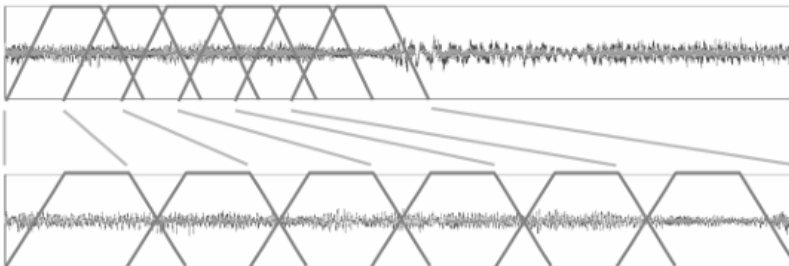
$$n T_{s,NEW} = n T_{s,IN} / v \quad v > 1 \text{ Encogimiento; } v < 1 \text{ estiramiento}$$

Hay que transformar f_{sNew} en f_{sOut} .

Si $v < 1$, $f_{sIn} > f_{sNew} < f_{sOut}$.

Si $v > 1$, $f_{sIn} < f_{sNew} > f_{sOut}$.

Granulación: partir la señal en pequeños trozos de unos pocos milisegundos, aplicandoles una ventana que se va desplazando con solapamiento y con una determinada envolvente



Retardos: uso de líneas de retardo para crear gran variedad de procesadores

$$t_D = D T_s = D / f_s.$$

Ejemplo. En un sistema que trabaje con una $f_s = 48 \text{ kHz}$, un retardo de $D = 24$ muestras, corresponde a un $t_D = 24/48000 = 0,0005 \text{ s}$ (0,5 ms) y, al contrario, un $t_D = 1 \text{ ms}$ se logra retrasando la señal en $D = t_D \times f_s = 48$ muestras. \square

Hay dos tipos, líneas de retardo donde el tiempo de retardo es fijo, y otras donde el tiempo de retardo es variable

Si $D = 1$, tendremos un filtro de pasa-baja (si restamos las muestras será pasa-alta)

Si $D = 2$, tendremos filtros elimina-banda o pasa-banda

Si $t_D = 1\text{ms}$, efectos de filtros de peine

Cortos: $t_D < 10 \text{ ms}$ (filtros de una única banda de paso o peines)

Medios $10 \text{ ms} < t_D < 50 \text{ ms}$. Resaltan la señal dando sonoridad

Largos: $t_D > 50 \text{ ms}$. Crean ecos percibidos como repeticiones del sonido original

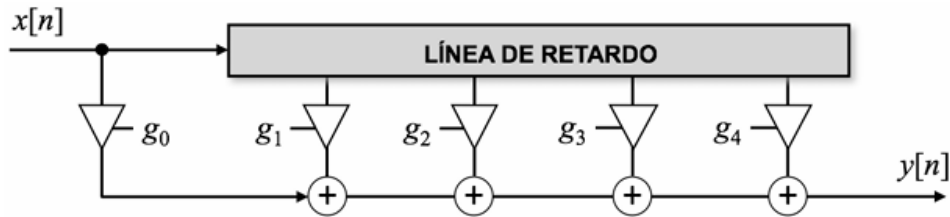
Ecos Simples

Para construir ecos, tenemos que tener un valor de D tal que $t_D > 50 \text{ ms}$ y su ecuación del filtro es:

$$y[n] = x[n] + g_R \cdot x[n - D].$$

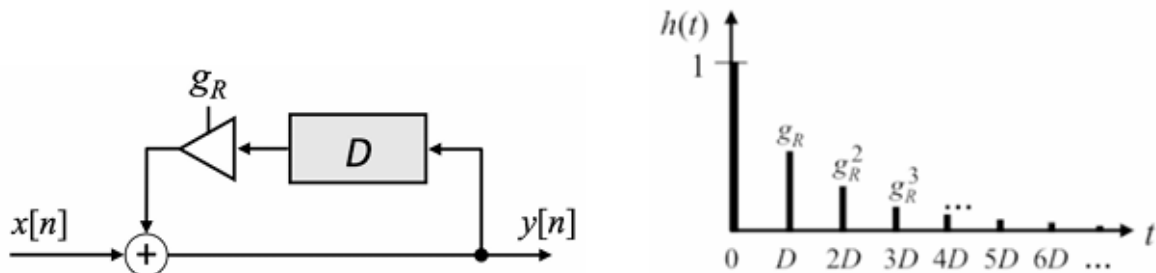
Ecos Múltiples Contables (multi-tap)

Si queremos tener más de 1 eco controlado, tendremos que aplicar un retardo múltiple.



Ecos Múltiples (filtro peine IIR para un número indefinido de repeticiones de ecos):

El valor de g_R es de gran importancia puesto que determina en parte el tiempo durante el cual se percibirá el eco. El tiempo de retardo determina la separación entre las copias



$$y[n] = x[n] + g_R \cdot y[n - D].$$

El tiempo durante el que se podrán oír los ecos se puede calcular de la siguiente forma:

$$t_{r60} = t_D \times \frac{60}{-20 \log_{10} g_R}$$

Para saber cuantas **repeticiones vamos a escuchar**, tenemos que dividir **tr60 / tD** en segundos

Ejemplo. Si $g_R = 0,6$, el denominador (que está en dB) valdrá 3. Si $t_D = 200$ ms obtenemos $t_{r60} = 2,7$ s. Si aumentamos la ganancia hasta $g_R = 0,8$ entonces $t_{r60} =$

6,2 s y si disminuimos $g_R = 0,4$ tendremos que $t_{r60} = 1,5$ s. Todo ello con el mismo valor de t_D , que influye directa y linealmente en t_{r60} . Si en la última configuración duplicamos t_D hasta 400 ms, entonces se duplicará también $t_{r60} = 3$ s. \square

1. Multiplicar ambos lados por $-20 \log_{10} g_R$ para eliminar el denominador:

$$t_{r60} \times (-20 \log_{10} g_R) = t_D \times 60$$

2. Dividir ambos lados por t_{r60} para despejar $-20 \log_{10} g_R$:

$$-20 \log_{10} g_R = \frac{t_D \times 60}{t_{r60}}$$

3. Dividir ambos lados por -20 para aislar el logaritmo:

$$\log_{10} g_R = -\frac{t_D \times 60}{20 t_{r60}}$$

4. Simplificar el coeficiente en el lado derecho:

$$\log_{10} g_R = -\frac{3 t_D}{t_{r60}}$$

5. Aplicar el antilogaritmo (la función inversa del logaritmo base 10) a ambos lados para despejar g_R :

$$g_R = 10^{-\frac{3 t_D}{t_{r60}}}$$

Ahí pone $3 \cdot t_D / t_{r60}$

Procesadores de la situación

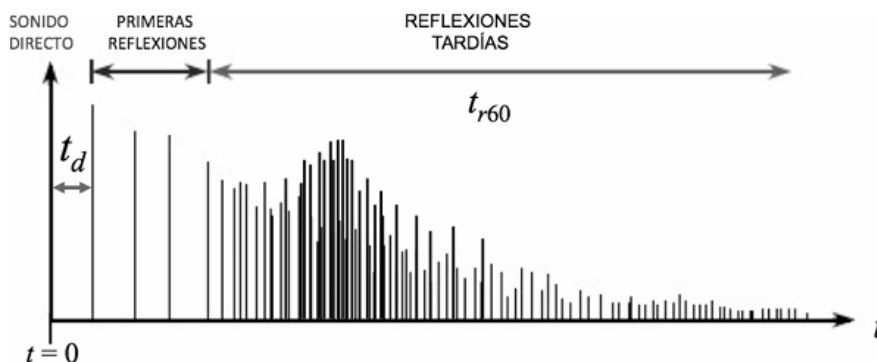
Las técnicas incluyen efectos como el control del panorama o de la distancia aparente para posicionar una fuente sonora en el espacio. Por otra parte, se usan también las llamadas reverberaciones.

Reverberaciones: resultado de múltiples reflexiones del sonido en un recinto acústico. Llegan al oyente en masa con retardos menores de 50 ms

Tiempo de retardo inicial (t_d): diferencia en tiempo entre que nos llega el sonido directo y su primera copia ($< 50\text{ms}$)

Primeras reflexiones: son las que se reciben a partir del primer reflejo en las superficies (entre 5 y 20)

Reflexiones tardías: se distribuyen más aleatoriamente y mucho más abundantemente
Tiempo de reverberación: tiempo durante el que se oyen las reverberaciones.



Existen unidades de reverberación especiales que tratan de crear sonoridades exóticas, como las reverberaciones puenteadas o las inversas. Estas cortan las reflexiones tardías antes de su duración natural.

Reverberadores analíticos: simulan múltiples reflexiones mediante la generación de copias del sonido usando filtros peine de tiempos medios y pasa-todas

Reverberadores por auralización: se replica el proceso sonoro de un espacio acústico mediante la convolución de la señal

Ecualizadores: proceso mediante el cual se modifica el espectro de un sonido, atenuando o aplicando determinadas bandas de frecuencias de la señal. No solo son capaces de atenuar sino también de amplificar (filtros activos)

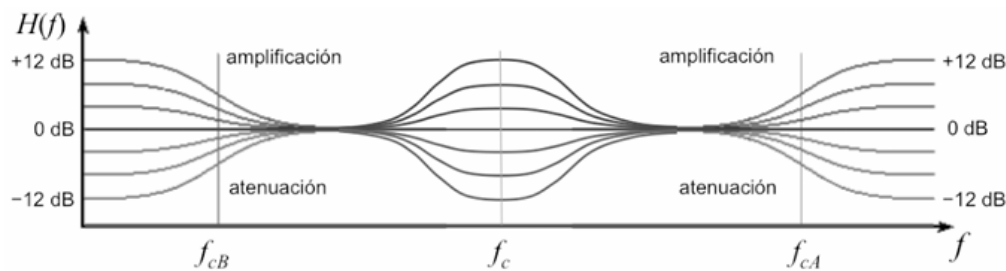
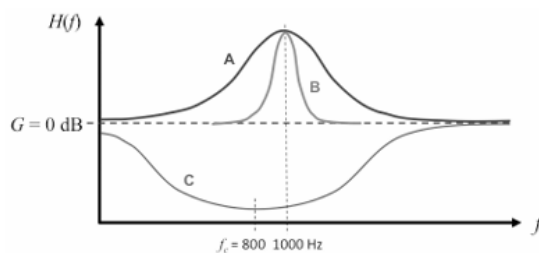


Figura 5.19: Respuesta en frecuencia de filtros pasa-baja, pasa-banda y pasa-alta en ecualizadores gráficos. f_{cB} y f_{cA} corresponden, respectivamente, a las frecuencias de corte de los filtros pasa-baja y pasa-alta. f_c es la frecuencia central del pasa-banda.

Ecualizadores gráficos: conjunto de filtros pasa-banda dispuestos en paralelo, cada uno de ellos centrado en una frecuencia f_c fija. El usuario controla la ganancia en cada una de las bandas de frecuencias. Interfaz intuitiva. Lo normal es que tengan filtros cuya frecuencia esté separada por una octava

Ecualizadores paramétricos: son más flexibles aunque también más difíciles de manejar. Constan de un filtro pasa banda en el que no solo se puede modificar la ganancia sino su frecuencia central y el ancho de banda

Ejemplo. La curva A corresponde a un comportamiento de amplificación con gran ancho de banda $Q = 1$, $f_c = 1$ kHz y $G > 0$ dB. El caso B corresponde a un ajuste con la misma ganancia y frecuencia pero con una $Q = 4$, por tanto menor ancho de banda, $W = f_c/4 = 250$ Hz. El caso C es totalmente diferente, pues atenúa ($G < 0$ dB) alrededor de 800 Hz y en un ancho de banda mucho mayor ($Q = 0,8$), igual a $800/0,8 = 1000$ Hz en 800 ± 500 Hz. □



Wah-wah: efecto popular entre guitarristas, consiste en un filtro pasa-banda de f_c variable y de banda estrecha. El valor de la f_c se suele controlar con un pedal

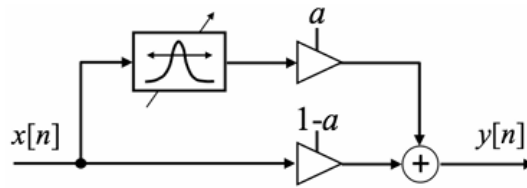


Figura 5.23: Esquema de un wah-wah digital, implementado como un pasa-banda de frecuencia central variable.

Flanger: se crea mezclando la señal con una copia de ella misma con un retardo de los que consideramos cortos ($1 < t_D < 15\text{ms}$) pero que cambia constante y cíclicamente

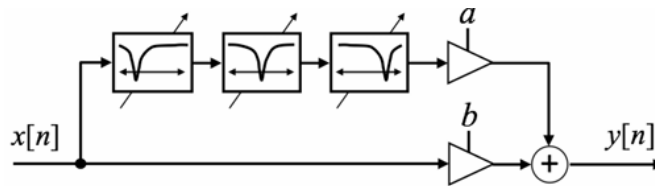


Figura 5.24: Esquema de un *phaser* digital, implementado como una cascada de filtros elimina-banda de frecuencia central variable, con coeficientes a y b que controlan los volúmenes relativos de efecto y señal seca, respectivamente.

Flanger digital usa una configuración de tipo peine IIR con una línea de retardo de tiempo variable $D[n]$. La ecuación del filtro es:

$$y[n] = x[n] + g_R y[n - D[n]] \quad (f_R = f_s/D = f_s/(f_s \times t_D) = 1/t_D)$$

Siendo $D[n] = D_0 + A_D \sin(2\pi f_m n / f_s)$

Ejemplo. Supongamos que el retraso D_0 fuera el número de muestras que haga retrasar la señal 2 ms. Con una señal muestreada a $f_s = 44.100$ Hz, este número es $D_0 = f_s \times t_D = 44.100 \times 0,002 = 88$ muestras. Si queremos que el LFO module este retraso estableciendo un límite inferior al t_D de 1 ms y que oscile dos veces por segundo, este oscilador debe crear una onda de amplitud $A_D = 44$ muestras con una frecuencia $f_m = 2$ Hz: $D = 88 + 44 \sin(4\pi n / f_s)$. \square

La diferencia entre el flanger y el phaser (pasa-toda variable) está en que en el phaser las muescas se distribuyen arbitrariamente a lo largo de las frecuencias y en el flanger los refuerzos están igualmente espaciados en el espectro

Chorus: pretende que un instrumento suene como si hubiese varios instrumentos iguales tocados al mismo tiempo. Se usa un retardo múltiple para obtener el efecto de un conjunto limitado de sonidos, y se van modificando ligeramente de forma continua mediante una señal aleatoria de baja frecuencia (**se usa un eco-multi tap**)

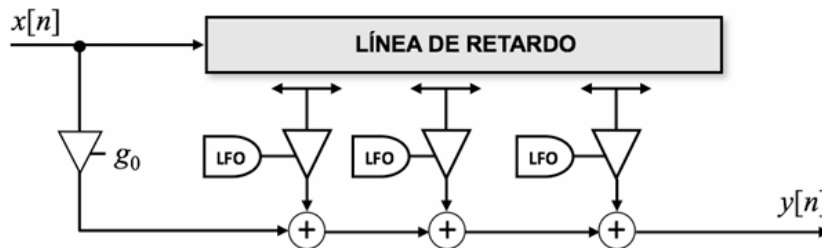


Figura 5.26: Implementación de un *chorus* digital. Las flechas bidireccionales representan puntos de acceso variables a la línea de retardo y los LFO son osciladores de baja frecuencia.

Tema 6

El estándar MIDI

- Los cables que unen los dispositivos;
- Los conectores de esos cables y los que tienen que incorporar los dispositivos;
- La manera de transmitir los mensajes por los cables;
- El tipo de mensajes que pueden intercambiar y su codificación.

La parte hardware de la especificación MIDI

MIDI tiene puertos y conexiones especiales para conectarlos a cables MIDI.

Tiene 3 tipos de puertos, IN, OUT y THRU

MIDI IN: el procesador interno del equipo recibe la información enviada desde otros aparatos

MIDI OUT: transmite a otros la información que el mismo genera

MIDI THRU: envía una copia de lo que recibe por el IN

Configuraciones y canales

Un sistema MIDI puede estar formado por varios dispositivos unidos por cables MIDI.

Debe haber un dispositivo maestro que conecte sincronice todos los demás dispositivos

“esclavos”. El Tiempo es marcado por un reloj, que transmite el maestro a los esclavos

Todo dispositivo MIDI tiene un reloj interno con el cual funcionará por defecto, pero esta configuración puede cambiarse para trabajar siendo temporizado mediante un reloj externo que le llegará por el puerto MIDI IN. Así pues, el maestro debe configurarse para que use su reloj interno y los esclavos para que usen el reloj externo. El esclavo recibirá un mensaje F8 cada 1/24 s



Figura 6.3: Conexiones de los cables entre puertos y configuración del reloj MIDI en dispositivos maestro y esclavo. La línea discontinua significa que esa conexión es opcional; sólo si se quiere enviar información en esa dirección.

Configuraciones típicas para conectar dispositivos:

En cadena: un dispositivo lo recibe en el IN, hace una copia en el THRU y lo envía al segundo, y así consecutivamente (la salida de la copia presenta retrasos acumulados, la señal se va deteriorando)

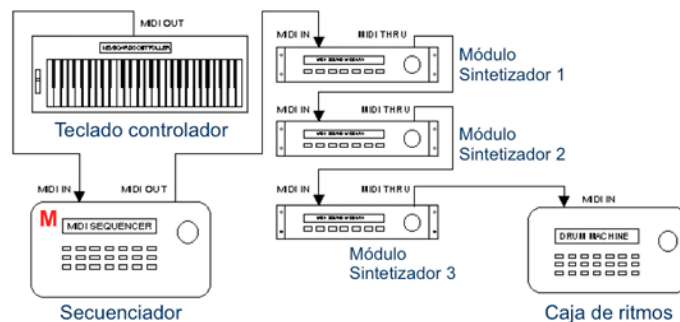


Figura 6.4: Ejemplo de configuración de un sistema MIDI con conexiones en cadena. Se ha marcado con una M el dispositivo que haría de maestro en este sistema.

En estrella: el maestro tiene más de una salida mediante un multiplexor y envía a cada esclavo un mensaje (mayor coste)

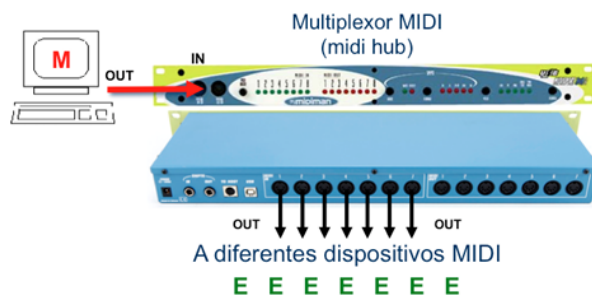


Figura 6.5: Ejemplo de configuración de un sistema MIDI con conexiones en estrella.

En bucle: los dispositivos envían cosas al maestro y este las distribuye,

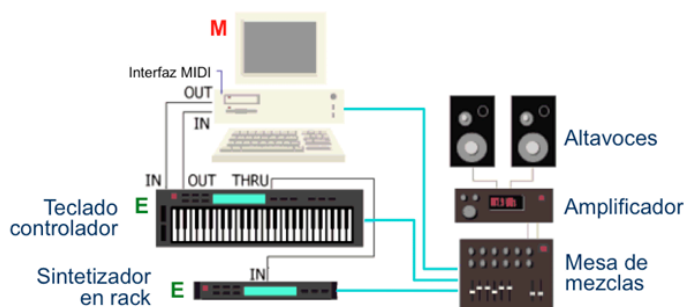


Figura 6.6: Ejemplo de configuración de un sistema MIDI con conexiones en bucle, con un ordenador como dispositivo maestro. Las líneas negras son conexiones MIDI mientras que las grises (o azul claro) son conexiones de audio.

Híbridos: combinaciones de las anteriores

Canales MIDI

Para poder enviar diferentes mensajes por los mismos cables, necesitamos los canales para que no haya interferencias entre los datos.

Por un cable se pueden transmitir hasta 16 canales (0 a 15). Esto es porque para representar los estados, en la estructura de los bytes, sólo se dispone de 4 bits para representar el canal ($2^4=16$)

Todos los dispositivos reciben la información, pero solo hacen caso a la que les llega por el canal con el que están sincronizados

Hablamos de sintetizador multitímbrico cuando un canal reproduce varias de las órdenes a la vez, y en alguna de ellas modifica el timbre

Cuando un único sintetizador reproduce mensajes que le llegan por distintos canales asignando un timbre distinto será un sintetizador multitímbrico que es capaz de recibir más de un canal simultáneamente para emitir diferentes timbres a la vez.

El número de canales diferentes que un dispositivo puede reproducir es la capacidad multitímbrica.

Aspectos generales de los mensajes MIDI

La estructura completa de un byte MIDI es:

0 E n n n n n n n 1

0 y 1 son el inicio y la parada del byte

E es el byte de estado (1 si es para bytes de estado y 0 para los de datos)

n son los bits que tiene de información útil del mensaje (entre 0 y 127)

Ejemplo. Supongamos que llegue un byte de estado que ponga a un sintetizador en el estado “*pon a sonar notas*”. A partir de ese momento los datos que vayan llegando serán interpretados como notas que empezarán a sonar. La llegada de otro byte de estado que indique “*detener el sonido de notas*” hará que los datos que le siguen sean interpretados como notas que tienen que dejar de sonar. Si el segundo estado hubiera sido “*modifica la afinación del instrumento*”, esos mismos datos hubieran sido interpretados de manera totalmente diferente por la máquina recibida. □

Hay tres tipos de estados:

Los que requieren 2 bytes de datos, los que requieren 1 byte de datos y los que no requieren datos.

Esta es la estructura de aptos tipos de bytes:

1 e e e c c c c 0 n n n n n n n

eee indican el tipo de estado (8 estados distintos)

cccc determinan el canal por el que se transmite

Ejemplo. Un mensaje E0_H en binario será 1110 0000 y codifica un mensaje de estado E enviado por el canal 1 (recuérdese que el valor 0 codifica el canal 1, etc.)

Un mensaje 83_H = 1000 0011 codifica el estado 8 enviado por el canal 4.

Un mensaje AF_H = 1010 1111 codifica el estado A enviado por el canal 16. □

Tipos de mensajes MIDI:

Mensajes de canal: tienen un número de canal

mensajes de voz: pulsaciones de teclas o cambio de timbre

mensajes de modo: indica cómo utilizar las voces y que canales usar

Mensajes de sistema: no afecta a ningún canal en concreto, y afecta a todos los equipos que lo reciben

mensajes comunes: canciones completas y afinación

de tiempo real: relativos a la sincronización de los dispositivos

exclusivos: de diseño libre del fabricante

Mensajes de canal

Actúan sobre dispositivos sintonizados en un canal determinado

Cuadro 6.1: Estructura de los mensajes de canal.

Estado	Hex.	Bin.	Byte datos 1	Byte datos 2
nota off	8x	1000xxxx	altura	velocidad
nota on	9x	1001xxxx	altura	velocidad (0 = off)
postpulsación polifónica	Ax	1010xxxx	altura	presión
cambio de controlador	Bx	1011xxxx	núm. controlador	valor
cambio de programa	Cx	1100xxxx	núm. programa	–
postpulsación de canal	Dx	1101xxxx	presión	–
variación de altura	Ex	1110xxxx	menos significativo	más significativo

Alturas en MIDI

Se pueden codificar alturas mediante valores entre 0 y 127

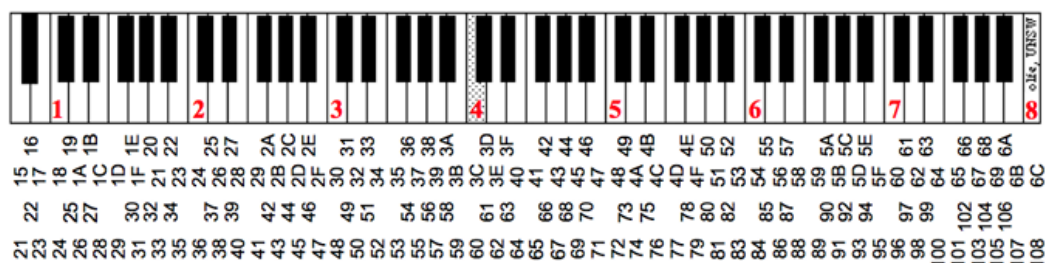
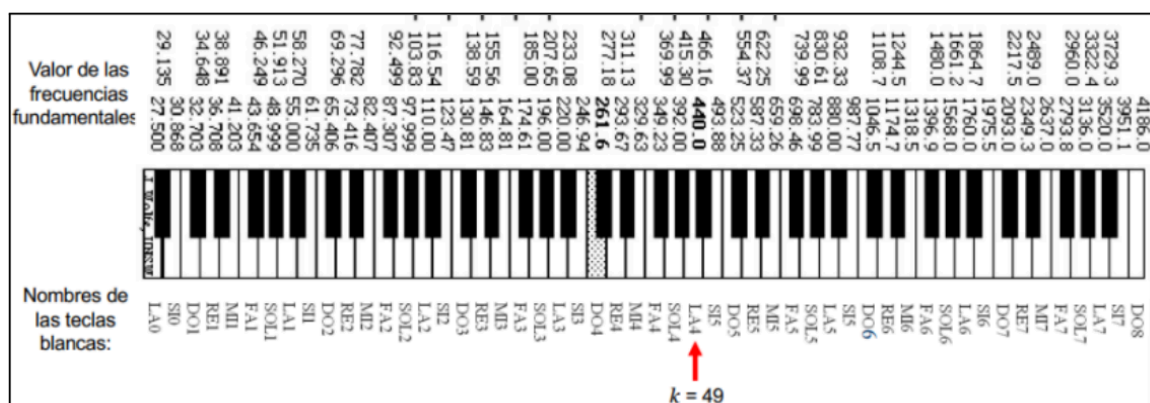


Figura 6.7: Tesitura del piano y valores numéricos en hexadecimal y decimal que asigna el estándar MIDI a dichas notas.



La altura en MIDI puede calcularse como:

$$h = \text{round} \left(69 + 12 \log_2 \left[\frac{f}{440} \right] \right)$$

y su frecuencia como:

$$f = 440 \cdot 2^{\frac{h-69}{12}}$$

Mensajes de voz

Se refieren a la interpretación de la música y muchos de ellos son parámetros que intervienen en la síntesis del sonido como: empezar notas, terminar notas, alterar alturas, modulación, seleccionar timbre

E6.12: Decodifica la siguiente secuencia de bytes MIDI:

C1 00 91 3C 7F E1 7F 7F 00 00 00 40 91 3C 00

C1 00 → selección de programa 0 (canal 2)

91 3C 7F → nota on DO₄

E1 7F 7F → alteración de la altura al máximo

00 00 → alteración de la altura al mínimo (Emisión del E1)

00 40 → alteración de la altura al 40 (Emisión del E1)

91 3C 00 → note off

Nota On → 9x aa vv

Indica que empiece a emitir una nota

Se indican la altura (a) y la velocidad (v) y la x es el canal (si es 1, el canal es 0).

La velocidad es el valor de volumen o intensidad del sonido

Si la velocidad es 0, significa que para la nota (nota off)

Ejemplo. Supongamos que el controlador emite la orden de activar un DO₄ por el canal 1 con velocidad 64. El código MIDI completo será: **10010000 00111100 01000000** (90_H 3C_H 40_H), donde el bit de estado de cada byte ha sido marcado en negrita.

Una secuencia de tres notas (DO₄ = 3C_H, DO₄[#] = 3D_H, RE₄ = 3E_H, por ejemplo) con esa misma velocidad por el mismo canal 1 puede generarse al menos con dos posibles secuencias de eventos MIDI (hexa.):

Con byte de estado para cada nota :	90	3C	40	90	3D	40	90	3E	40
Haciendo uso del estado actual :	90	3C	40		3D	40		3E	40

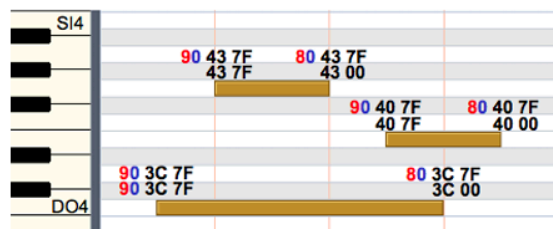
Nota off → 8x aa vv

Idéntico al nota on, pero para hacer callar una nota

Ejemplo. El mensaje que haría callar a la nota emitida en el ejemplo anterior sería: **10000000 00111100 00000000** (80_H 3C_H 40_H). Como el mensaje de hacer callar una nota es equivalente al de hacerla sonar con velocidad cero, las siguientes secuencias de eventos (hexa.) para hacer sonar y callar una nota son equivalentes:

Con <i>nota on</i> y <i>nota off</i> :	90	3C	40	80	3C	40
Sólo con <i>nota on</i> y haciendo uso del estado actual :	90	3C	40		3C	00

Ejemplo. Vamos a ver otro ejemplo en el que los inicios y finales de las notas se van intercalando, como puede suceder en cualquier situación práctica.



Las secuencias de bytes para generar la secuencia de notas de este diagrama de piano-roll son las siguientes: la primera secuencia se ha hecho utilizando todos los mensajes de cambios de estado, con notas on y off. La segunda usa sólo notas on.

Con *nota on* y *off* : 90 3C 7F 90 43 7F 80 43 7F 90 40 7F 80 3C 7F 80 40 7F
 Sólo con *nota on* : 90 3C 7F 43 7F 43 00 40 7F 3C 00 40 00

Cambio de programa → Cx pp

Cambiar el timbre del sintetizador o cambiar el programa

pp es el valor del programa nuevo

E6.10: Selecciona el programa 2 del banco número 12430 en un sintetizador sintonizado en el canal 1.

Hacemos el cambio de controlador. Dividimos 12430 entre 128. Como no te indican nada, el controlador que se coge es el 00, y para el 2º Byte, se coge el opuesto

Alteración de la altura → Ex rr cc

“Pitch bend” desafinar voluntariamente el sonido del sintetizador

rr es el resto, el dato menos significativo

cc es el coeficiente, el dato más significativo

Se calcula mediante la siguiente fórmula: $2^{\circ} \text{ Byte} \times 128 + 1^{\text{er}} \text{ Byte}$.

Ejemplo. Supongamos que queremos alterar la altura del sintetizador conectado al canal 16 subiendo la frecuencia a un poco más de la mitad del rango positivo. Este valor será +4100, que deberá codificarse mediante el número $8192 + 4100 = 12.292$. Para separar este número en los 2 Bytes que lo codifican, dividimos por 128 y nos quedamos con el cociente (96) y el resto (4) → $12.292 = 96 \times 128 + 4$, que en hexadecimal sería $60_H \times 80_H + 04_H$. La secuencia completa será:

byte de estado dato menos significativo dato más significativo
 EF 04 60

Postpulsación (“AfterTouch”)

Presión que se ejerce sobre el teclado mientras está bajada la tecla y que el controlador suele transmitir cada cierto tiempo

Polifónica (Ax nn vv): distingue la presión sobre cada tecla. n es el número de la altura y v la presión

De canal (Dx vv) que reconoce una sola presión máxima sobre todo el teclado. v la presión

Cambio de controlador (Bx ii vv)

Para variar el valor de un controlador.

i es el identificador del controlador y v es el nuevo valor que va a adoptar

Hay dos tipos de controladores:

Continuos: numerados desde el 0 al 63, con valores de 0 a 127

En estos se permite dar valores simples o de doble precisión, ya que los números del 0 al 31 están emparejados con los del 32 al 63

Byte del controlador 0-31 \times 128 + Byte del controlador 32-63.

Ejemplo. Para enviar un mensaje para situar el balance (control 8) del canal 10 en la posición máxima (127); sería 10111001 00001000 01111111 (B9_H 08_H 7F_H). Si se envía solo el control 40: B9_H 28_H 7F_H, también haría el mismo efecto. □

Ejemplo. Supongamos que queremos cambiar el volumen principal (control 7) del canal 1 (B0_H) con alta resolución (valores entre 0 y 16.383) para que nos quede con valor 10.000. Para codificar este valor con dos bytes de 7 bits, lo dividimos entre 128 y da 78 de cociente y resto 16. Por lo tanto, se codifica como $78 \times 128 + 16$ y se deberá enviar la siguiente cadena de mensajes: un control 7 (07_H) con valor 78 (4E_H) y un control 39 (27_H) con valor 16 (10_H). En definitiva:

Con los dos bytes de estado : B0 07 4E B0 27 10

Usando el estado actual : B0 07 4E 27 10

Interruptor: numerados desde 64 a 95, que solo pueden ser 0 o 127

Cuadro 6.2: Listado de controladores MIDI recogidos en el estándar 1.0.

(dec.)	(hex.)	Controladores continuos	MSB	(dec.)	(hex.)	LSB
0	00	Selección de banco	0-127	32	20	0-127
1	01	Rueda de modulación	0-127	33	21	0-127
2	02	Controlador por respiración	0-127	34	22	0-127
5	05	Tiempo de portamento	0-127	37	25	0-127
6	06	Entrada de datos	0-127	38	26	0-127
7	07	Volumen del canal	0-127	39	27	0-127
8	08	Balance	0-127	40	28	0-127
10	0A	Panorámico	0-127	42	3A	0-127
11	0B	Control de expresión	0-127	43	3B	0-127
12-31		Controlador continuo #N	0-127	44-63		0-127
Controladores interruptores						
64	40	Pedal sustain on/off	0=off, 127=on			
65	41	Portamento on/off	0=off, 127=on			
66	42	Sustenuto on/off	0=off, 127=on			
67	43	Soft pedal on/off	0=off, 127=on			
68-95		Sin definir	—			
96	60	Entrada de datos +1	—			
97	61	Entrada de datos -1	—			
98	62	NRPN - LSB	0-127			
99	63	NRPN - MSB	0-127			
100	64	RPN - LSB	0-127			
101	65	RPN - MSB	0-127			
102-120		Sin definir	—			

Cuando haces de doble precisión, el segundo valor tiene que ser el módulo del número entre 128

E6.16: ¿Cómo se pone el volumen a mitad de su nivel máximo en un sintetizador conectado al canal 10, usando simple y doble precisión?

E6.16: Simple: B9 07 40 o B9 27 40;

Doble: B9 07 40 27 00 (también B9 07 40 B9 27 00)

El 6 y 38 modifican último controlador que haya sido especificado
Y el 96 y 97 para hacer ajustes del controlador

Parámetros RPN y NRPN

RPN corresponde a registered parameter number (número de parámetro registrado)
mientras que NRPN viene de non registered parameter number (número de parámetro no registrado)

Los RPN se definen en la especificación del estándar MIDI y funcionan igual para todas las máquinas MIDI. Los NRPN son específicos de cada una y no es probable que funcionen igual para diferentes máquinas.

Cuadro 6.3: Listado de números de parámetros registrados (RPN)

Control		Función del parámetro	Valor
101	100		
0	0	Sensibilidad para la alteración de la altura	MSB = +/- semitonos LSB = +/- cent.
0	1	Afinación fina maestra	Resolution 100/8192 cent. 00 _H 00 _H = -100 cent. 40 _H 00 _H = A440 7F _H 7F _H = +100 cent.
0	2	Afinación gruesa maestra	Sólo se usa MSB Resolución 100 cent. 00 _H = -6400 cent. 40 _H = A440 7F _H = +6300 cent.
0	3	Tuning Program Change	Tuning Program Number
0	4	Tuning Bank Select	Tuning Bank Number
0	5	Rango profundidad de modulación	Definido en la especificación de GM2
127	127	RPN nulo	Sin efecto

Ejemplo. Para cambiar la sensibilidad de la alteración de la altura en un sintetizador conectado en el canal 7 hay que enviar los siguientes bytes de cambios de controlador:

Decimal usando el estado actual : 182 101 0 100 0 6 2 (38 0)
Secuencia equivalente hexadecimal : B6 65 00 64 00 06 02 (26 0)

Ejemplo. El General Standard de Roland (ver sección 6.6.2) define un parámetro denominado “frecuencia de vibrato”. Este parámetro se codifica en NRPN mediante el par (99 = 1, 98 = 8) con valores en un rango [0,127]. Si queremos asignar el valor máximo a este vibrato en el sintetizador conectado al canal 1, la secuencia será:

Decimal usando el estado actual : 176 99 1 98 8 6 127
Secuencia equivalente hexadecimal : B0 63 01 62 08 06 7F

Mensajes de Modo

Sirven para indicar cómo debe utilizar el instrumento sus voces internas y los canales MIDI. Se codifican como mensajes de cambio de controlador que utilizan los números de controlador entre 121 y 127

Cuadro 6.4: Mensajes de canal de modo.

Mensaje	Hexa.	Bin.	Byte dato 1	Byte dato 2
restaurar controladores			79	0
control local on / off			7A	7F = on, 0 = off
desactivar todas las notas			7B	0
modo omni off	Bx	1011xxxx	7C	0
modo omni on			7D	0
modo mono on (poly off)			7E	0-16 ^a
modo poly on (mono off)			7F	0

^a Si = 0 el número de canales lo determina el receptor; los otros valores especifican el número de canales para los que responderá, contando desde el canal actual.

Restaurar todos los controladores (Bx 79 00)

Devuelve todos los valores de los controladores a su estado óptimo inicial.

Control local/remoto (Bx 7A 7F/Bx 7A 00)

Permite que un sintetizador que disponga de un teclado propio pueda conectar o desconectar la conexión entre su teclado y el generador de audio. Si está desactivado, las órdenes MIDI generadas por el teclado no controlan la generación de sonido del sintetizador. Si está activado, las órdenes también afectan al sintetizador.

Silenciar todas las notas (Bx 7B 00)

Hacer callar en caso de emergencia las notas que hubieran podido quedar activadas por error

Selección de modo

Existen varios modos que se pueden combinar entre ellos:

	POLY	MONO
OMNI ON	1	2
OMNI OFF	3	4

OMNI ON polifónico. El dispositivo recibe mensajes por cualquier canal pero emite por uno solo. La activación del modo OMNI hace que el sintetizador interprete todas las órdenes, independientemente del canal por el que lleguen.

OMNI ON monofónico. El dispositivo recibe mensajes por cualquier canal, pero solo puede sonar una nota al mismo tiempo. Cuando recibe dos notas simultáneas, aplica algún criterio especificado anteriormente.

OMNI OFF polifónico. Es el más usado, porque cada instrumento puede tocar polifónicamente lo que llega por el canal en el que está sintonizado. La desactivación del modo omni implica que el dispositivo se sintoniza en uno de los canales MIDI.

OMNI OFF monofónico. Útil para que un sintetizador reciba varias melodías monofónicas, cada una por un canal distinto, asignándose cada uno a un timbre diferente

Mensajes del sistema

Mensajes para todos los dispositivos conectados en un sistema MIDI. Hay tres tipos: comunes, de tiempo real y exclusivos

Mensajes comunes:

Permiten ubicar la reproducción de una secuencia de mensajes en un punto determinado de una secuencia musical concreta de entre todas las almacenadas.

Puntero de posición de canción (F2 nn nn): indica a los dispositivos en que punto de la secuencia unirse

Selector de canción (F3 nn): envía un dato para seleccionar una canción entre las almacenadas

Petición de afinación (F5): afina los osciladores

Tramma temporal (F1): transmite o recibe códigos de tiempo

Ejemplo. La secuencia (en hexadecimal, haciendo uso del *running status*, lo que permite usar un único F1_H, en vez de uno antes de cada dato):

F1 00 11 22 33 44 55 60 70

codificaría en el estándar SMPTE el tiempo cronométrico 01:23:45:00. 1 hora, 23 minutos, 45 segundos y 00 *frames*. □

Mensajes en tiempo real

Usados por dispositivos maestros que almacenan y reproducen secuencias de mensajes

Reloj MIDI (F8): lo utilizan los dispositivos maestros para marcar el ritmo

Inicio, parada y continuación (FA/FC y FB): cuando se pulsa el botón de reproducir, cuando se pulsa el de parada y cuando se da a continuar.

Espera activa (FE): para verificar que los dispositivos MIDI están conectados entre sí

Reset del sistema (FF): deja el estado que lo recibe igual que cuando se enciende

Mensajes de sistemas exclusivos

Para que instrumentos intercambien información que solo concierne a ellos o para transmitir o recibir datos que operen sobre características especiales de un aparato determinado y que no puedan realizarse utilizando otros mensajes

Empieza con un F0 y acaba cuando hay un F7

El 1r byte de datos es para el identificador del fabricante, el segundo es para el identificador del dispositivo y el tercero para el identificador del modelo

Ejemplo. El mensaje de sistema exclusivo que en el sintetizador JV1080 (identificador de modelo 6A_H) de Roland (identificador de fabricante 41_H) permite seleccionar un tipo determinado de reverberación para los sonidos que emite es la siguiente serie de bytes (en hexadecimal):

F0	41	10	6A	12	01	00	28	06	51	F7
1	2	3	4	5	6				7	

1. byte de estado de inicio de Sysex
2. identificador de fabricante (Roland)
3. identificador de dispositivo (16)
4. identificador del modelo (JV-1080)
5. identificador de mensaje
6. datos
7. byte de estado de final de Sysex

Como este ejemplo, hay muchos otros mensajes del sistema exclusivo que han pasado a formar parte del estándar MIDI en las sucesivas ampliaciones que éste ha tenido. Otros ejemplos son la petición de volcado de configuraciones desde un dispositivo a un sistema de almacenamiento (normalmente un ordenador), la carga de muestras de audio digital en sintetizadores tipo *sampler*, la petición de identificación a dispositivos, etc.

Ejemplo. El mensaje que restituye los valores de los parámetros de la configuración estándar en un sintetizador acogido al sistema General MIDI (ver sección 6.6.1) es:

F0 7E 7F 09 01 F7

Algunos números de identificación se han utilizado para ampliar la norma MIDI, dando lugar a los mensajes de sistema exclusivo universales. Uno de ellos es, por ejemplo, el mensaje que permite indicar a un sintetizador cambiar el volumen global. Su estructura es la siguiente:

F0 7F 7F 04 01 dd dd F7

donde

1. F0_H: byte de estado de inicio de *sysex*
2. 7F_H: indicador universal de mensaje de tiempo real
3. 7F_H: identificador de dispositivo, con este valor lo recibirán todos
4. 04_H 01_H: identificadores de mensaje de cambio de volumen global
5. dd : valor del byte más significativo del volumen (entre 0 y 127)
6. dd : valor del byte menos significativo del volumen (entre 0 y 127)
7. F7_H: byte de estado de final de *sysex*.

Tarjeta de implementación

Permite definir cientos de mensajes

Hoja en un formato estandarizado que describe los mensajes que el dispositivo es capaz de transmitir y de recibir

MIDI Implementation Chart

[Portable Keyboard]

Date: 1993. 6. 15

Model: PSR-510

MIDI Implementation Chart

Version: 1.00

Function	Transmitted	Recognized	Remarks
Basic Channel	Default Changed	1~16 CH 1~16 CH	(*1) (*1)
Mode	Default Messages Altered	Mode 3 X ***** X	(*1) X X
Note Number : True voice		0~127 *****	0~127 0~127
Velocity Note on Note off		O 9nH, v=1~127 X 9nH, v=0	O 9nH, v=1~127 X 9nH, v=0 or 8nH
After key's Touch Ch's	X X	X O (*2)	
Pitch Bender	O	O	
Control Change	0, 32 1 6, 38 7 10 11 64 66 84 90 91 96 97 100, 101 120 121	O (*3) X O O O O O X O X O X O O O (*6) O X	O (*4) O O O O O O O O (*5) O O O O O (*7) O (*8)
Program Change : True #		O 0~127 *****	O 0~127 0~127 (*9)
System Exclusive		O (*10)	O (*10)
System : Song Position : Song Select Common : Tune	X X X	X X X	
System : Clock Real Time : Commands	O O (*12)	O (*11) O (*12)	
Aux : Local ON/OFF : All Notes Off Messages : Active Sense : Reset	X X O X	X O O X	

Mode 1: OMNI ON, POLY
Mode 3: OMNI OFF, POLY

Mode 2: OMNI ON, MONO
Mode 4: OMNI OFF, MONO

O: Yes
X: No

Ejemplo. Vamos a interpretar los contenidos de la tarjeta de implementación de la figura 6.10, intentado responder a algunas cuestiones sobre el funcionamiento de este dispositivo. Por ejemplo, ¿puede el sintetizador descrito en dicha tarjeta manejar mensajes de sistema exclusivos? Si miramos en la fila 'System Exclusive' vemos que hay un ☐ tanto en la columna de transmisión como de recepción, lo que indica que sí que puede tanto transmitirlos como recibirlos. El signo ***10** es una referencia a un comentario fuera de tabla (no mostrado) que indica que esto es configurable.

Otros ejemplos que podemos analizar son que este dispositivo no transmite *note off* sino que los implementa como *note on* con velocidad cero, o que no transmite el mensaje de sistema de silenciar todas las notas (All Notes Off) pero sí que lo reconoce. ☐

X NO
O SÍ

Canales MIDI
Modo por defecto y cambios
Rango de notas
¿Sensible a la velocidad?
¿Entiende la postpulsación?
Lista de controles que entiende
¿Entiende el cambio de programa?
¿Maneja mensajes sysex?
¿Entiende los mensajes comunes?
¿Entiende los mensajes de tiempo real?

Function ...		Transmitted	Recognized	Remarks
Basic Channel	Default Changed	1 1-16	1-16 1-16	Memorized
Mode	Default Messages Changed	Mode 1 x x	Mode 1 x x	Mono ignored
Note Number :	True Voice	0 0-127	0 36-96 0 36-96	
Velocity	Note ON Note OFF	0 0	0 x	
After Touch	Key's Ch's	x x	x x	
Pitch Bender		0	0	
Control Change	0,32 1 64	0 0 0	x 0 0	
Program Change :	True #	0 1-96 *****	0 1-96 0 1-96	
System Exclusive		0	0	
System :	Song pos	x	x	
	: Song sel	x	x	
Common :	Tune	x	x	
System :	Clock	x	x	
Realtime :	Commands	x	x	
Aux :	Local ON/OFF	x	x	
	: All Notes OFF	0	0	
Mes- :	Active Sense	x	x	
sages:	Reset	0	0	

Extensiones del estándar MIDI (mirar de los apuntes el 6.6)

Tema 7

La secuenciación es el proceso por el que se construye y almacena una secuencia de control.

Puede ser en tiempo real, cuando las órdenes de control son producidas en el momento que son enviadas, sin almacenamiento o gestión.

O Pueden ser en tiempo diferido, es decir, las órdenes de control van a ser usadas en otro momento teniendo que almacenarlas, visualizarlas, editarlas, etc... Es en este caso cuando necesitamos un sistema especializado en realizar estas tareas (el secuenciador) y las técnicas involucradas en ellas (la secuenciación).

La generalización del concepto de secuenciador de órdenes MIDI que también es capaz de trabajar con audio digitalizado, se conoce con el nombre de Digital Audio Workstation (DAW). Existen DAWs para todos los sistemas operativos y de todos los precios, por lo que su uso se está generalizando en todos los campos de trabajo relacionados con la creación sonora.

Los secuenciadores toman su nombre de los primeros generadores de breves secuencias de notas. Eran programadores de secuencias (breves) en tiempo real, tenían un control analógico de sintetizadores. No es MIDI.

Inspirados en esas máquinas analógicas surgen los sistemas de programación, almacenamiento y reproducción de secuencias MIDI. Hay dos tipos de secuenciadores:

Hardware:

Ventajas: robustos, fáciles de transportar, idóneos para el directo

Inconvenientes: difíciles de usar

Software:

Ventajas: riqueza de posibilidades, idóneos para el estudio

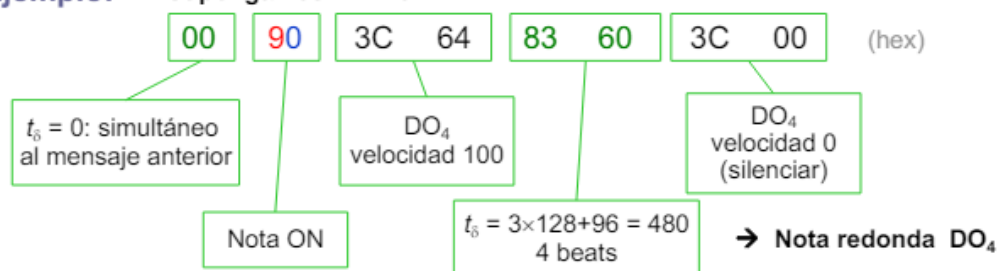
Inconvenientes: difíciles de transportar, dependencia del S.O. Dispositivos Mac se imponen en el uso profesional a la hora de secuenciar.

Bloque	Información	Longitud	Contenido
Cabecera	Firma	4	"MThd"
	Longitud	4	del resto del bloque = 6
	Formato	2	0 = 1 sola pista 1 = multi-pista (1 por instrumento) 2 = multi-secuencia
	Número de pistas	2	Si Tipo = 0 \Rightarrow siempre 1
	Resolución temporal	2	En pulsos por cada nota "negra"
<Número de pistas> veces:			
Pista (track)	Firma	4	"MTrk"
	Longitud	4	N
	Datos	N	Secuencia de mensajes MIDI: de canal + de sistema + meta-eventos + <i>delta-times</i> (separación temporal entre mensajes)

Tiempos delta (temporización del contenido)

- Hasta 127 ticks, sólo se usa 1 byte:
 $64 \text{ ticks} \rightarrow t_{\delta} = 40_H$ $127 \text{ ticks} \rightarrow t_{\delta} = 7F_H$
- A partir de 128 ticks, comienzan a usarse más bytes:
 $t_{\delta} = 81 \text{ } 00_H \rightarrow 128 \text{ ticks}$
 $t_{\delta} = C0 \text{ } 01_H \rightarrow \textcolor{red}{1}1000000 \text{ } \textcolor{red}{0}0000001 \rightarrow \begin{array}{cc} \text{1000000} & \text{0000001} \\ 40_H \times 80_H & + \quad 01 & = 2001_H \end{array} \rightarrow 8193 \text{ ticks}$
 $64 \times 128 + 01 = 8193$
- El máximo valor codificable es:
 $t_{\delta} = FF \text{ } FF \text{ } FF \text{ } 7F_H \rightarrow 268.435.455 \text{ ticks}$

Ejemplo: Supongamos $r = 120$



Para codificar el tiempo de delta

Haces la división entre 128 y lo que te da de cociente, se lo sumas a 80, y eso es lo que pones como primer byte. Luego el residuo lo pasas a hexadecimal y lo pones como segundo byte. Javier maricon

Cuadro 7.3: Los meta-eventos más utilizados en ficheros MIDI estándar.

Valor (hex.)	Bytes de datos	Descripción
00	02 ss ss	Número de secuencia (sólo para formato tipo 2)
01	nn texto	Texto genérico (nn = longitud en bytes del texto)
02	nn texto	Nota sobre copyright (nn = longitud en bytes del texto)
03	nn texto	Nombre de la pista (nn = longitud en bytes del texto)
04	nn texto	Instrumento de la pista (nn = longitud en bytes del texto)
05	nn texto	Letra de la canción (nn = longitud en bytes del texto) A diferencia de los anteriores, suele codificar sílabas simultáneas a las notas.
06	nn texto	Marca de posición (nn = longitud en bytes del texto)
07	nn texto	Marca de referencia (nn = longitud en bytes del texto)
2F	00	Marca de fin de pista
51	03 t1 t2 t3	Ajuste del tempo (en μ s/tiempo)
58	04 nn dd cc bb	Tipo de compás (nn=numerador, 2^{dd} =denominador, cc=mensajes de reloj por tiempo y bb=fusas por tiempo)
59	02 sf mm	Tonalidad (número de sostenidos si es positivo o bemoles si es negativo, y si es mayor mm = 0 o menor mm = 1)
7F	nn dd ...	Informaciones específicas (nn = longitud en bytes del texto)

inicio	Ident.(hex.)	Long.	Bytes datos	Descripción
FF	Del 01 al 07	nn	texto	Metaeventos de texto (nn = longitud en bytes del texto)
FF	51	03	t1 t2 t3	Ajuste del tempo (como la duración en microsegundos de 1 beat) Tempo codificado = $t1 \times 256^2 + t2 \times 256 + t3$ en μ s / beat Tempo en BPM = $6 \cdot 10^7 / \text{Tempo codificado}$
FF	58	04	nn dd cc bb	Métrica (tipo de compás) nn = numerador 2^{dd} = denominador + además tendremos: cc=mensajes de reloj por tiempo ($24 \rightarrow 18_H$), bb=fusas por tiempo (08)
FF	59	02	sf mm	Tonalidad (número de alteraciones y modo) Si sf > 0 \rightarrow número de sostenidos Si sf = 0 \rightarrow sin alteraciones (ni sostenidos ni bemoles) Si sf > 249 \rightarrow número de bemoles en complemento a 2 (255 = -1, 254 = -2, 253 = -3, etc.) Modo: mayor si mm = 0 ; menor si mm \neq 0

Referencia de tonalidades en función del número y tipo de alteraciones:

Diagram showing musical notation for 12 major and 12 minor scales, organized in two rows of six scales each. The scales are labeled with their names and the number of sharps or flats.

Top row (Major scales): SOL mayor, RE mayor, LA mayor, MI mayor, SI mayor, FA# mayor, DO# mayor.

Bottom row (Minor scales): MI menor, SI menor, FA# menor, DO# menor, SOL# menor, RE# menor, LA# menor.

Bottom row (Major scales): FA mayor, SIb mayor, MIb mayor, LAB mayor, REb mayor, SOLb mayor, DOb mayor.

Bottom row (Minor scales): RE menor, SOL menor, DO menor, FA menor, SIb menor, MIb menor, LAB menor.

Tema 8

Sintetizadores:

VCO: un oscilador analógico responde a una tensión de control continua, oscilando con una frecuencia que depende de ella.

DCO: un oscilador digital es un algoritmo informático que genera valores de muestras en función de unos parámetros de control.

- | |
|---|
| • Sintetizador sustractivo con capacidad polifónica = 5 |
| • 2 VCO por nota emitida (triangular, cuadrada, diente de sierra y pulso) |
| • LFO conectable al VCF |
| • VCF pasa-baja de 24 dB/octava con control de resonancia |
| • VCA con GE tipo ADSR |
| • Teclado de cinco octavas de DO a DO |

E8.1: ¿Este sintetizador es analógico o digital? ¿Cuántos osciladores tiene en total? y ¿cuántas teclas? Es analógico. 10 osciladores + 1 de baja frecuencia (LFO, son

osciladores, pero las señales que producen son de frecuencias $f < 20 \text{ Hz}$ = 11 en total.

Tiene 5 octavas * 12 + 1 = 61 teclas

E8.2: ¿Qué efectos sobre el sonido puede conseguir su oscilador de baja frecuencia?

El LFO conectado al VCF permitirá (al menos) modular la frecuencia de corte (f_c).

E8.3: Describe el filtro que tiene. El VCF (con la posibilidad de controlar sus parámetros de funcionamiento como la frecuencia de corte (con V_{fc}), ancho de banda (VQ), resonancia, etc. mediante valores de tensión continua) es un pasa-baja con alta pendiente que permite añadir ganancia positiva alrededor de la f_c .

E8.4: ¿Qué se puede controlar en el amplificador de este sintetizador? El VCA (sintetizador analógico, un amplificador controlado por tensión aplicará a la señal eléctrica que recibe, una ganancia que se controla mediante una tensión eléctrica) permite que su control de amplitud venga condicionado por un GE tipo ADSR.



E8.5:

Cutoff : control de la frecuencia de corte del filtro

Res : control de la cantidad de resonancia entorno a la f_c del filtro

Env : “cantidad de envolvente aplicada”

Kbd : keyboard tracking Slope : pendiente de la banda de transición del filtro en dB/octava (según número de polos del filtro)

Type: tipo filtro: pasa-alta, pasa-baja, pasa-banda o elimina-banda

ADSR: parámetros de la envolvente



E8.6:

K.F.: keyboard tracking (como Keyboard Following)

Type: tipo filtro: pasa-alta, pasa-baja, pasa-banda o elimina-banda (notch)

K.F. : Distancia entre la f_c del filtro y la frecuencia de cada nota

Reso : control de la cantidad de resonancia entorno a la f_c del filtro

Cut : frecuencia característica del filtro

E8.10: Habitualmente Csound genera sonidos con una resolución de 16 bits. Si en la cabecera tenemos $sr = 16000$ y $nchnls = 2$, a) ¿cuántos bytes ocupará un fichero generado si activamos un oscilador durante 50 segundos? b) Si la amplitud del oscilador = 0 dBFS, ¿qué valor de amplitud PCM presentará la señal digital?

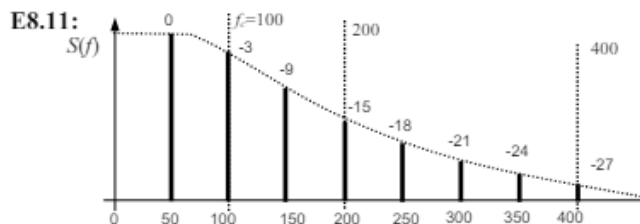
Tamaño del archivo (bytes) = Tasa de muestreo × Número de canales × Profundidad de bits por muestra × Duración en segundos = $16000 \text{ muestras/seg} * 2 \text{ canales} * 2 \text{ bytes/muestra} * 50 = 3200000 \text{ bytes}$

En un sistema digital de audio, 0 dBFS es el nivel máximo de amplitud que se puede representar sin distorsión. Con una resolución de 16 bits, el rango de valores que puede tomar una muestra es de -32768 a 32767 .

Dado que 0 dBFS representa el máximo posible sin distorsión, el valor de amplitud PCM de una señal a 0 dBFS será el máximo positivo para la profundidad de bits de 16, que es 32767.

Así que, si la amplitud del oscilador es 0 dBFS en un sistema de 16 bits, el valor de amplitud PCM que presentará la señal digital será 32767

E8.11: Sea un sintetizador sustractivo digital ($f_s = 22.050 \text{ Hz}$) que usa como forma de onda una aproximación con 8 parciales al tren de pulsos y que emite un SOL1, dibuja el espectro aproximado (con sus frecuencias y amplitudes en dB) resultante si usamos un DCF pasa-baja con $f_c = 100 \text{ Hz}$ de -12 dB/octava .



En los filtros pasa-baja, la frecuencia de corte se define típicamente como el punto donde la señal se atenúa por 3 dB.

Cada octava, cada vez que se duplica la frecuencia tiene que quitar 12. Por lo que si pintas cada 50, de 100 a 200 tiene que quitar 12 en dos pasos, cada vez quita 6.

Ahora para la siguiente octava, de esta a la siguiente, hasta que se duplique, de 200 a 400, tiene que quitar 12 en 4 pasos, 200 de diferencia / 50 cada vez que pintamos = 4. $-12 / 4 = -3$, cada paso quita -3.

E8.13: Implementa una orquesta de Csound con el sintetizador de E8.11 con un instrumento (nombre VIB), con un filtro tone para implementar el DCF y una envolvente ASR (con linen) con tiempos de ataque y relajación del 1% de la duración de la nota.

```

E8.13: sr = 22050
instr VIB
  idur = p3
  iamp = ampdbfs(p4)
  ifrec = cpspch(p5)
  inarm = p6
  if_c = p7
  asal buzz iamp, ifrec, inarm, 101
  asal tone asal, if_c
  asal linen asal, 0.01*idur, idur, 0.01*idur
  out asal
endin

```

E8.15: Sea esta tabla con sólo 8 muestras: a) ¿Qué forma de onda se sintetizará usándola en un sintetizador de tablas de ondas? b) ¿Qué valores del índice de fase son posibles para ella? c) ¿Cuál de esos valores corresponde a una fase = π ? d) ¿Qué valor tendrá la onda para una fase de $3\pi/2$ si $A = 100$ PCM?

triangular

0, 1, 2, 3, 4, 5, 6, 7

fase = π es la muestra central de la tabla, 4

fase = $\frac{3}{4}\pi$ es $\frac{3}{4}$ de 8 muestras, la muestra 6, si divides en grupos de 2 y coges el 1 del último. Valor -1, si la A es 100, $-1 * 100 = -100$ PCM

Key tracking. Los Sintetizadores pueden permitir que el filtro incluya este seguimiento de la nota. Este sistema permite que la f_c del filtro no sea un valor determinado, sino que dependa de la nota emitida.