Parte 6: Síntesis por modulación AM y FM

Hasta ahora hemos utilizado osciladores con frecuencias fundamentales ***fijas***, o dependientes de una nota MIDI, pero estáticas en términos generales. Del mismo modo, las intensidades para cada nota han sido estáticas, al menos durante la fase de sustain de una envolvente ADSR, o completamente estáticas en otros casos, limitándose a encenderse y apagarse. Disponiendo de distintos tipos de osciladores, podemos enriquecer mucho más el timbre y la expresividad de nuestros sintetizadores al utilizar ***modulación***. En lugar de utilizar el oscilador para producir frecuencias fundamentales directamente audibles, podemos utilizarlos para modificar en el tiempo (modular) distintos parámetros básicos de otro oscilador, principalmente su frecuencia (FM o ***frecuencia modulada***) o su amplitud (AM o ***amplitud modulada***).

La terminología tradicional denomina ***portadora*** a la forma de onda primaria, cuya fundamental en principio será audible, y ***moduladora*** a la forma de onda que modifica su frecuencia o amplitud. Hacemos esta distinción de "frecuencias audibles" porque es habitual utilizar, para la función moduladora, osciladores de baja frecuencia (LFO) cuyas fundamentales se encuentran en el rango de 0 a 20 Hz (o 50 Hz dependiendo el caso).

Existen distintas aproximaciones para generar ambos tipos de modulación. Por lo general realizamos AM a una señal multiplicándola directamente por la modulante, mientras que haremos FM sumando a su frecuencia de oscilación la señal modulante, aunque esto no es siempre estricto. Es habitual, sin embargo, utilizar una combinación de multiplicaciones con sumas y otras operaciones simples para adaptar los rangos de valores. Por ejemplo, algunas veces buscamos que las moduladoras solo provean valores positivos, para evitar como resultado frecuencias negativas o desplazamientos de continua (por multiplicación de dos números negativos).

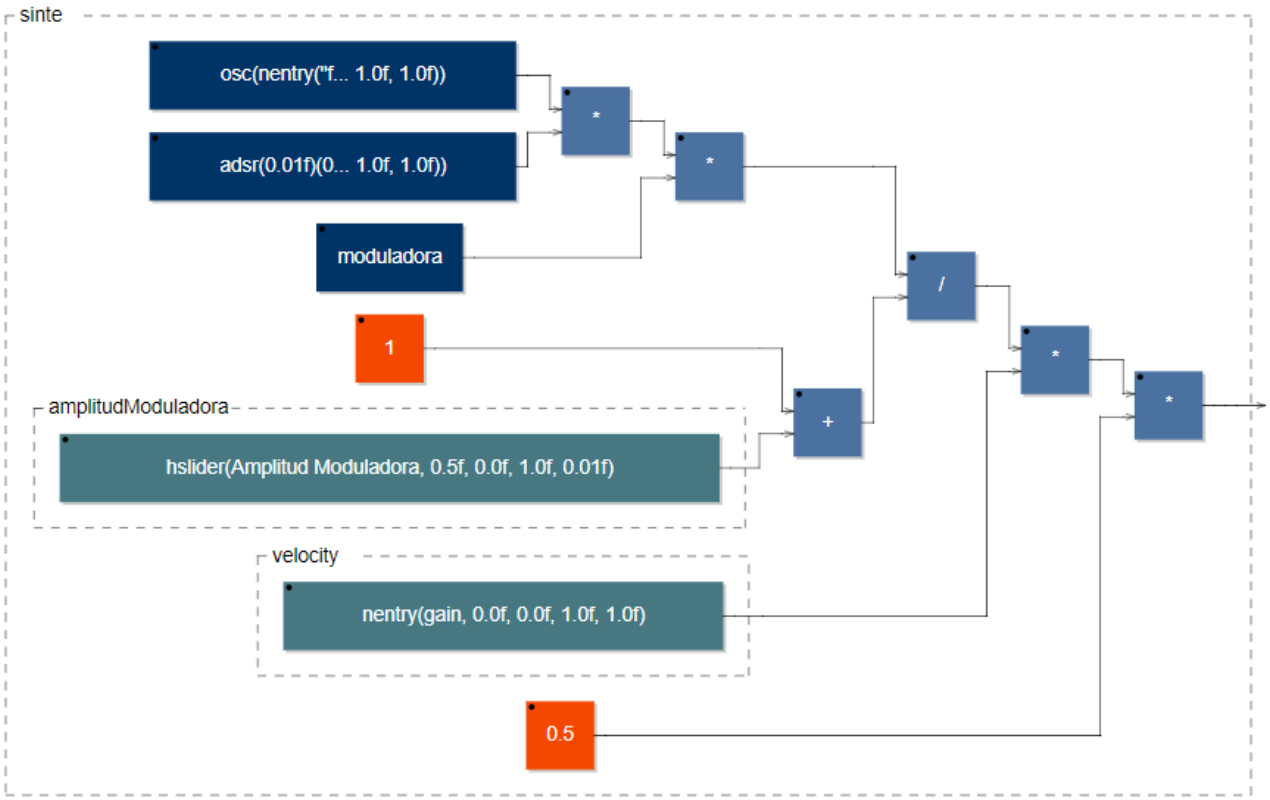
Contenidos | Parte 6

Ejemplo 1: Modulación de amplitud (trémolo)

Comenzaremos con algo muy simple: modular la amplitud de una señal en base a un oscilador. Para ello generaremos una señal ***portadora***, que puede tener cualquier forma de onda, y la multiplicaremos por una ***moduladora***. De esta manera, al oscilar la moduladora modificará muestra por muestra la amplitud de la portadora. Por supuesto que podemos elegir todo tipo de variantes y combinaciones, por ejemplo utilizando distintas formas de onda para la moduladora. El siguiente ejemplo parte de una onda cuadrada, pero pueden utilizarse otras formas de onda y escuchar el resultado. Nada nos impediría tampoco "modular a la moduladora" y otras combinaciones peculiares, aunque es un proceso más común al modular frecuencias (FM).

Excepto el oscilador sinusoidal `os.osc`, que funciona sin inconvenientes, disponemos de algunas variantes de osciladores específicamente diseñadas para trabajar con bajas frecuencias, incluso menores a 1 Hz, y que llevan el prefijo `lf_`. En el ejemplo se encuentran comentadas estas alternativas.

Por otro lado, al utilizar formas de onda con cambios abruptos como la cuadrada o la diente de sierra, es posible que el sonido produzca saltos no deseados. Por esta razón introduciremos el uso de la función `si.smooth(nivelSuavizado)`, que es un ***interpolador*** de valores. Para quienes conozcan PureData, se trata de algo muy similar al ***line***. Podemos también utilizar esta función para suavizar cambios en controles de volumen y otros de ser necesario. La función recibe un solo parámetro numérico entre 0.0 y 1.0, correspondiendo al "grado" de suavizado. De esta manera suavizaremos ligeramente las ***transientes*** de la moduladora para evitar saltos de sonido no deseados. Se debe tener en cuenta que este proceso modifica efectivamente la forma de onda, reduciendo su contenido en altas frecuencias.



```
import("stdfaust.lib");

// controles para MIDI
gateNotas = nentry("gate", 0, 0, 1, 1);
frecPortadora = nentry("freq", 0, 0, 1, 1);
velocity = nentry("gain", 0, 0, 1, 1);

// controles de la moduladora
suavizador = si.smooth(hslider("Suavizado moduladora", 0.9, 0.8, 1.0, 0.001));
frecModuladora = hslider("Frecuencia Moduladora", 9, 1, 50, 1);
amplitudModuladora = hslider("Amplitud Moduladora", 0.5, 0, 1, 0.01);

// Probar:  os.osc | os.lf_triangle | os.lf_squarewave | os.lf_saw
// sumamos 1 para tener numeros positivos entre 0 y 2
moduladora = (1 + (os.osc(frecModuladora) * amplitudModuladora)) : suavizador;

// Nuestro sinte con ADSR y modulacion
// aqui compensamos la amplificacion que provoca la moduladora al ser entre 0 y 2
sinte = os.osc(frecPortadora) * en.adsr(0.01, 0.1, 0.7, 0.5, gateNotas)
        * moduladora / (1 + amplitudModuladora) * velocity * 0.5;

volumen = hslider("Volumen", 0.5, 0, 1, 0.01);
process = sinte * volumen <: _, _;
```

Podemos hacer algunos experimentos, por ejemplo, añadirle una envolvente de frecuencia a la propia moduladora para quebrar un poco la monotonía de la modulación:

```
moduladora = (1 + (os.osc(frecModuladora * en.adsr(0.1, 0.1, 1, 0.5, gateNotas)) * am
plitudModuladora)) : suavizador;
```

Además del habitual efecto de trémolo, se recomienda también explorar con frecuencias de modulación más allá de los 20 Hz: encontraremos otro tipo de sonidos nuevos al entrar en el rango de frecuencias audibles para la fundamental de la moduladora.

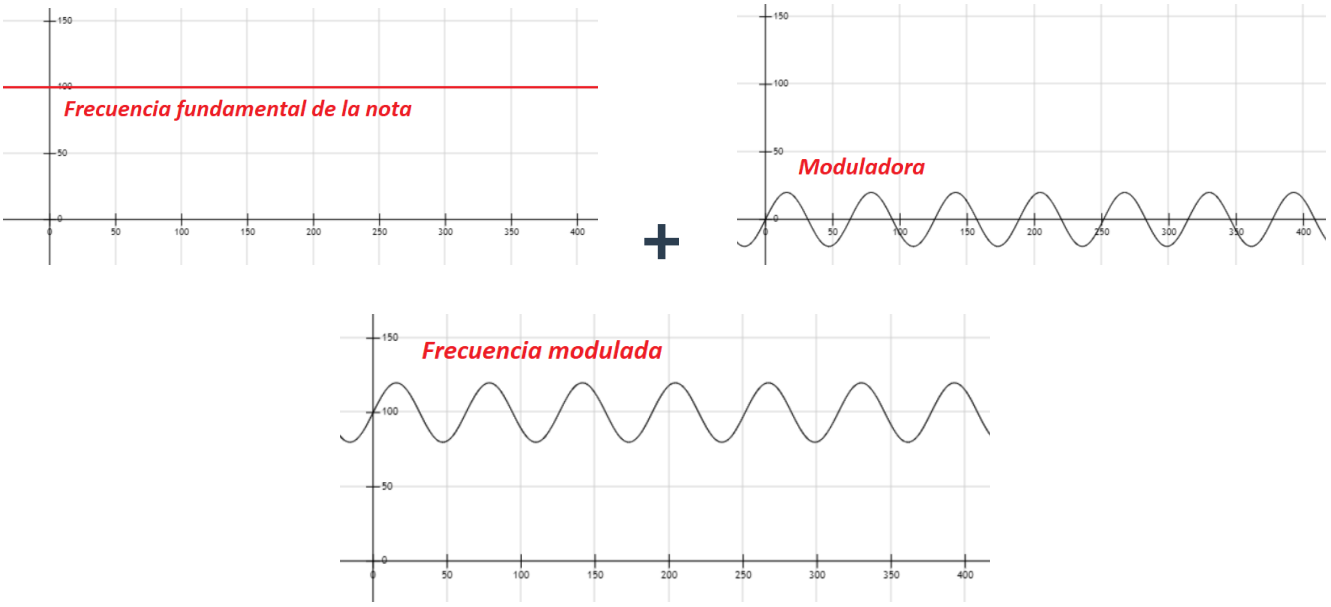
Ejemplo 2: Modulación de frecuencia (vibrato)

Antes de entrar directamente en el terreno de la síntesis por frecuencia modulada, nos apoyaremos en el ejemplo anterior para modificar, esta vez, la fundamental de la portadora, simulando un efecto de “vibrato”. Para ello debemos plantearnos algunas cuestiones.

En primer lugar, debemos decidir si nuestro vibrato se realizará ***alrededor*** de la fundamental, ***hacia abajo***, o ***hacia arriba***. En nuestro ejemplo lo haremos alrededor, es decir, un valor fijo en Hz oscilando hacia arriba y hacia abajo con centro en la fundamental de la portadora. El modo más sencillo de realizar esto es ***sumando*** la moduladora a la portadora. Sin embargo, deberemos realizar un escalado a la moduladora, ya que a priori tendremos una señal de -1 a 1 de amplitud, y poco ganaríamos cambiando en sólo +/- 1 Hz a la portadora. En algunas ocasiones, para obtener una modulación requeriremos ***sumar*** señales, y en otras ***multiplicarlas***, dependiendo de cada caso particular.

Por otra parte, debido a que la altura musical sigue una escala logarítmica, no es conveniente realizar un escalado fijo (digamos multiplicando por 10 o por 100 u otro número), ya que este valor tendrá un resultado muy diferente según la nota utilizada. Por ejemplo, si le restamos 50 Hz a una fundamental de 100 Hz, estaremos efectivamente descendiendo una octava completa; en cambio, si la fundamental fuera de 1000 Hz, aquellos mismos 50 Hz representan menos que un semitono. Una estrategia cruda pero efectiva es utilizar la propia fundamental de la portadora como factor de escalado, de manera que la moduladora ***adapte*** automáticamente su escala conforme a la altura musical.

Los siguientes gráficos ilustran el proceso que buscamos. Suponiendo una fundamental de 100 Hz para la portadora, generaremos una moduladora que oscile entre números positivos y negativos alrededor de 0, pero escalados según una ***profundidad*** de vibrato deseada, en este caso de -20 a 20. Luego, sumamos ambas señales y obtendremos un resultado que oscila entre 80 y 120 Hz.



En el ejemplo a continuación generaremos un pequeño sintetizador de diente de sierra, en el que controlaremos la frecuencia de la moduladora y la profundidad de modulación. Es conveniente que la moduladora no pueda tener frecuencia 0 Hz, ya que esto la bloquearía en un valor fijo

provocando una desafinación estática. Para controlar la profundidad de modulación crearemos un multiplicador basado en la frecuencia fundamental de la portadora. De esta manera, profundidad 0 representará no tener modulación en absoluto, mientras que 1 provocará una oscilación muy extrema entre 0 Hz y 2*fundamental. Los valores habituales de profundidad suelen ser debajo de 0.1. Para facilitar la selección de valores precisos, incorporaremos la etiqueta `[scale:exp]` dentro del slider, lo que hará que su movimiento siga una escala exponencial, teniendo mayor definición en valores pequeños.

```
import("stdfaust.lib");

// controles para MIDI
gateNotas = nentry("gate", 0, 0, 1, 1);
frecPortadora = nentry("freq", 0, 0, 1, 1);
velocity = nentry("gain", 0, 0, 1, 1);

profundidad = hslider("Profundidad vibrato [scale:exp]", 0.01, 0, 1, 0.001);
frecVibrato = hslider("Frecuencia vibrato", 4, 0.1, 20, 0.1);

// una onda cuadrada daría algo similar a un tremolo de notas
// Probar:  os.osc | os.lf_triangle | os.lf_squarewave | os.lf_saw
moduladora  = os.osc(frecVibrato);

multiplicador = profundidad * frecPortadora;
volumen = hslider("Volumen", 0.5, 0, 1, 0.01);

sinte = os.sawtooth(frecPortadora + (moduladora*multiplicador))
        * en.adsr(0.01, 0.1, 0.7, 0.5, gateNotas)
        * velocity;

process = sinte * volumen : fi.lowpass(1, 8000) <: _, _;
```

Según la profundidad y el estilo musical buscado, es posible que convenga utilizar vibratos más complejos, por ejemplo, siguiendo intervalos musicales en lugar de valores fijos en Hz; o bien que oscilen por debajo de la fundamental.

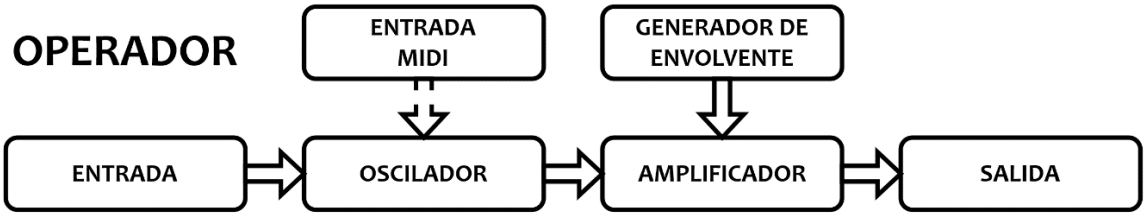
Algo que podemos sumarle a nuestro ejemplo para darle una mejor expresividad, como en el caso de la AM, es una envolvente en la propia moduladora. De esta manera el vibrato aparecerá gradualmente, o con retraso respecto del ataque de la nota. En estos casos es importante mantener el valor de ***sustain*** en 1, ya que no se trata del nivel dinámico o de volumen, sino del valor por el que la envolvente escalará a la moduladora, siendo 1 el valor “normal” o “completo”. El parámetro crítico aquí será el de ***ataque***, se recomienda explorar con distintos valores. Para añadir esta envolvente, modificar la siguiente línea del ejemplo anterior:

```
multiplicador = profundidad * frecPortadora * en.adsr(0.7, 0, 1, 0.5, gateNotas);
```

Ejemplo 3: Síntesis por modulación de frecuencia (FM)

La síntesis de FM es una de las técnicas más poderosas y versátiles para la generación de todo tipo de timbres sonoros. Puede crear sonidos tanto armónicos como inarmónicos utilizando unos pocos elementos, volviéndola muy eficiente en comparación con otros sistemas capaces de producir sonidos similares. Una gran variedad de sintetizadores comerciales utiliza síntesis por FM (quizás con el Yamaha DX7 como emblema clásico), siendo utilizados por innumerable cantidad de artistas durante las últimas décadas. El dominio de esta técnica es un arte realmente complejo, y no lograremos abarcar todas sus posibilidades en este curso, pero podemos dar los primeros pasos con un par de conceptos muy simples.

Habitualmente se describe un sintetizador por FM en términos de ***combinaciones de operadores***. Llamamos ***operador*** a una agrupación de elementos simples que venimos trabajando hasta ahora, por lo general podemos pensarlo de esta manera:



No es más que un bloque de los mismos elementos que hemos visto en este curso, pero sistematizados (la entrada MIDI no necesariamente está siempre presente). La síntesis FM se basa en la combinación de distintos operadores, trabajando en serie y/o en paralelo, modificándose mutuamente e incorporando filtros y efectos en la cadena. Algo similar hemos hecho al generar el vibrato, donde un oscilador controla el funcionamiento de otro. Alterando los parámetros de formas de onda, las frecuencias, envolventes, la forma de combinarlos y otros, podemos generar todo tipo de timbres sonoros.

No es sencillo anticipar el resultado de una combinación particular, pero en términos generales podemos decir que para sintetizar sonidos ***armónicos***, la señal de modulación debe tener una relación armónica con la señal portadora original, es decir, sus frecuencias fundamentales deben guardar una relación proporcional entera. A medida que aumenta la cantidad de modulación de frecuencia, el sonido se vuelve progresivamente más complejo. Utilizando moduladores con frecuencias que no sean múltiplos enteros de la portadora (es decir, una relación ***inarmónica***), se pueden crear espectros de percusión, campanas y otros. Podemos utilizar todo tipo de osciladores, incluyendo generadores de ruido.

En el siguiente ejemplo crearemos un sencillo sintetizador FM con dos operadores, muy similar a lo que realizamos con el vibrato, pero buscaremos crear parámetros que nos permitan generar sonidos interesantes a partir de la modulación. Es común encontrarnos que, según las relaciones de frecuencias fundamentales, la síntesis por FM altera significativamente la altura percibida respecto de la fundamental original, por eso dispondremos de un sencillo control ***octavador*** para cambiar fácilmente la afinación. Su función será proveernos un multiplicador para la frecuencia fundamental, que convierta por ejemplo “una octava arriba” en *2, o “dos octavas abajo” en *0.25. Esto lo haremos utilizando la función `pow(base, exponente)`, que no es otra cosa que una elevación exponencial para convertir números enteros en multiplicadores de base 2, incluyendo fracciones por debajo de 1.

Cabe aclarar que, por lo general, cuando hablamos de sintetizadores por frecuencia modulada comerciales y semiprofesionales, suelen construirse modulando la ***fase*** de la señal (***modulación de fase***). Es un proceso muy similar pero que conlleva algunas ventajas respecto de la modulación de frecuencia directa. Aquí nos limitaremos a realizar sólo esta última, en vistas de comprender la idea fundamental y de mantener sencillos los ejemplos.

Entre las nuevas características que tendrá nuestro ejemplo, incorporaremos un control de ***múltiplo*** (respecto de la portadora en el operador 1) para la frecuencia de la moduladora (operador 2). Esto nos dará control total pero también una referencia de la ***armonicidad*** según se relacione de manera entera con la portadora (¡esto incluye múltiplos fraccionales!). La amplitud del operador 2 controlará el grado de modulación, dejando intacto al operador 1 al llegar a 0. Las envolventes de ambos operadores las realizaremos esta vez utilizando `adsre`, que utiliza los mismos parámetros que `adsr`, pero genera curvas exponenciales en lugar de líneas rectas como envolvente. Por último, añadiremos un filtro pasa altos para bloquear corrimientos de DC, un pasa bajos para atenuar el brillo excesivo, y un reverberador artificial provisto por las librerías de Faust que utiliza el algoritmo ***Freeverb***, bastante popular. Se recomienda explorar los cambios tímbricos resultantes según los parámetros de modulación.

```
import("stdfaust.lib");
// Entrada MIDI
gateNotas = nentry("gate", 0, 0, 1, 1);
octava = pow(2, hslider("[3] Octava", 0, -4, 4, 1)); // multiplicador de octava
velocity = nentry("gain", 0, 0, 1, 1);

// Operador 1 (portadora)
fundamental = nentry("freq", 0, 0, 1, 1) * octava; // añadimos el octavador
envolventeOp1 = en.adsre(0.02, 0.1, 1, 2, gateNotas);
operador1 = os.osc(fundamental + operador2) // sumamos el resultado del operador2!
            * envolventeOp1 * velocity;

// Operador 2 (modula al operador 1)
multOperador2 = hslider("[0] Multiplo Operador 2 [scale:exp]", 3, 0.00, 16, 0.001);
amplitudOperador2 = hslider("[1] Amplitud Operador 2", 0.6, 0, 1, 0.001) * 10000; // escalado!
ataqueOperador2 = hslider("[2] Ataque Operador 2 (s)", 0.3, 0, 2, 0.001);
envolventeOp2 = en.adsre(ataqueOperador2, 0.1, 1, 0.2, gateNotas);
operador2 = os.osc(fundamental*multOperador2 /* + Operador 3 si hubiera...! */ )
            * envolventeOp2 * amplitudOperador2;
// PROBAR: qué pasa si en lugar de un ataque fijo o por slider, le ponemos "1 - velocity"

volumen = vslider("[3] Volumen", 0.25, 0, 1, 0.01);

process = hgroup("Sintesis FM",
            vgroup("Operadores", operador1)
            : fi.highpass(1, 20)    // este filtro combate corrimiento de DC (centro)
            : *(volumen) : fi.lowpass(1, 10000) <:  // filtro de brillo
            vgroup("Reverb", dm.freeverb_demo));    // Reverb
```