

SÍNTESIS SUSTRACTIVA

3.1 RUIDO BLANCO y FILTROS

La síntesis sustractiva nace de la idea de poder crear un sonido nuevo restando amplitud a algunos componentes de sonido, más complejo de aquel que se va a obtener, por medio del uso de filtros. Un filtro es Un dispositivo que deja pasar ciertas frecuencias mejor que otras.

Comenzaremos viendo cómo obtener ruido blanco con C-sound. Usaremos la unidad generadora *rand* la cual requiere un solo argumento : la amplitud. Esto es así porque al obtener todas las frecuencias audibles juntas no requiere definición de una frecuencia particular. Además Rand va a generar una forma de onda aleatoria [que no contiene periodicidades] por lo cual no va a ser generada a partir de una función externa. Por esta razón el score no requiere sentencia f, más bien directamente comienza con las sentencias i.

VER ARCHIVO

[01 NOISE.csd](#)

Veremos como actúan sobre el ruido blanco los distintos filtros.

A un filtro se envía una señal de entrada, es decir el sonido que queremos modificar. Podremos definir el modo en que este filtro debe operar, es decir cuales frecuencias serán atenuadas. El resultado de esta operación, los datos que describen nuestro sonido modificado por el filtro, serán depositados como es costumbre en una variable de audio.

Filtro Pasa bajos:

Fc: Frecuencia de corte

a1 tone señal de entrada, fc

Filtro Pasa altos:

a1 atone señal de entrada, fc

3.2 FILTROS PASA BAJOS DE PRIMER ORDEN

El filtro pasa bajos es un filtro que atenúa las frecuencias superiores a la frecuencia de corte. Esta atenuación se hará según una determinada curva. Esta curva depende el orden del filtro: si el filtro es de primer orden la curva de atenuación bajará 6 dB por octava. Dado una fc de 300 hz., tendremos la siguiente curva de atenuación:

Amplitud Frecuencias en hz

0 dB	300
-6 dB	600
-12 dB	1200
-18 dB	2400

02 FILTRO TONE.csd

3.3 FILTROS PASA ALTOS DE PRIMER ORDEN

El opcode atone también es un filtro de primer orden, dada una f_c de 1000 hz tendremos

Amp	frec [hz]
0 dB	1000
-6 dB	500
-12 dB	250
-18 dB	125

03 FILTRO ATONE.csd

3.4 FILTROS DE ORDENES SUPERIORES

Un filtro de segundo orden atenúa 12 dB por octava; uno de tercer orden -18 dB/oct; etc.

Estos casos se realizan conectando varios filtros en serie, es decir la salida del primero es filtrada nuevamente por otro filtro igual y con la misma f_c .

04 FILTROS DE ORDENES SUPERIORES.csd

3.5 FILTROS PASA BANDA

Un filtro pasa banda permite atenuar tanto las frecuencias superiores como inferiores con respecto a una determinada banda de frecuencias. Esto se puede obtener en C-sound con la Unidad generadora *reson* [que es un filtro de 2º orden]. En él se debe establecer la frecuencia central de la banda pasante [también la representaremos como f_c] y el ancho de misma [bandwidth: bw] .

a1 reson señal de entrada, f_c , bw

En un filtro de este tipo, las frecuencias que se encuentra a una octava de los extremos del ancho de banda se atenuarán en 12 dB, las que están a dos octavas, 24 dB, etc.

También se pueden crear filtros pasabanda de cuarto orden, conectando dos *reson* en serie.

Cuando restringimos cada vez más el ancho de banda, se escuchará cada vez más nítida la f_c , hasta obtener un efecto similar a una senoide. En un filtro ideal a menor ancho de banda mayor será la atenuación de las frecuencias que están fuera de ella, pero en *reson*, como es un Filtro resonante, al aumentar la relación f_c/bw , se amplifica la señal central, por lo cual será necesario atenuar la amplitud de la señal de salida.

Esto se puede lograr de diversas maneras según el caso, o dividiendo la señal filtrada por algún coeficiente determinado (que habrá que obtener a prueba y error) o utilizando un tercer argumento opcional : el reescalamiento, donde se pueden poner dos números 1 o 2.

1: reescala la amplitud de la señal de salida de modo tal que la amplitud de pico del sonido filtrado sea la misma que la de la señal de entrada.

2: reescala la amplitud de la señal de salida de modo tal que el valor eficaz (RMS de la señal filtrada sea el mismo que el de la entrante

Por último se puede hacer un filtro dinámico modificando tanto el ancho de banda como la f_c , en cuyo caso se va a producir un glissando.

[05 FILTROS RESON.csd](#)