

Informática Musical

Procesamiento de audio digital. PyAudio

1. Implementar un oscilador *chirp*(*frecIni*,*frecFin*,*dur*) que genere una señal sinusoidal de duración *dur*, que comience con una frecuencia *frecIni* y la incremente gradualmente (de manera lineal) hasta alcanzar *frecFin*. Este tipo de osciladores es conocido y utilizado en distintas aplicaciones (<https://en.wikipedia.org/wiki/Chirp>).

Para implementarlo se puede partir del oscilador básico visto en clase y modificarlo para variar el parámetro de frecuencia en el argumento de la función *sin*.

2. Implementar un sintetizador para hacer síntesis aditiva que tome como argumentos una tabla de frecuencias de la forma $[(frec_1, vol_1), \dots, (frec_n, vol_n)]$ (pares frecuencia-amplitud) y genere el sonido resultante de sumar osciladores sinusoidales con dicha tabla.
3. Implementar una versión elemental de un sintetizador sinusoidal basado en *tabla de ondas* (*wavetable*). Para ello se generará y almacenará una muestra con solo ciclo del oscilador en un array numpy. Después el oscilador debe recorrer cíclicamente dicho array para producir la señal requerida.

A continuación mejorar la funcionalidad para que, utilizando la misma muestra, pueda producir la señal con la frecuencia requerida.

4. (Entregable) Utilizar el filtro IIR visto en clase para implementar un filtro paso banda (BP) con frecuencia de corte y ancho de banda configurables. Utilizar dos filtros LP y HP en secuencia, calculando los valores de α según se ha explicado en clase.