

Informática Musical

Procesamiento de audio digital. PyAudio

Los ejercicios marcados con **(Entregable)** pueden presentarse como práctica de laboratorio evaluable. Debe presentarse el programa Python correspondiente, debidamente comentado y listo para ejecutar.

1. Implementar un sintetizador FM con “ n ” frecuencias moduladoras similar al que se ha visto en clase, pero que permita controlar en tiempo real (mediante teclado) las frecuencias y los coeficientes β de las mimas, así como los de la portadora.
2. Extender el sintetizador FM del ejercicio anterior para permitir otros tipos de onda: cuadrada, diente de sierra, etc.
3. **(Entregable)** El *theremin* es un instrumento electrónico basado en osciladores, que consta de dos antenas metálicas que detectan la posición relativa de las manos del intérprete (*thereminista*). Esa posición determina la frecuencia y la amplitud del tono (la frecuencia se controla con una mano y la amplitud (volumen) con la otra). Puede verse un vídeo en <https://www.youtube.com/watch?v=K6KbEnGnymk>

En este ejercicio se implementará un *theremin* con alguno de los modelos de síntesis visto en clase (por ejemplo el sintetizador FM del ejercicio anterior). Las antenas se simularán con el ratón: la posición horizontal determina la frecuencia (en un rango prefijado, como [100,10000]) y la vertical, la amplitud. Para el obtener la posición del ratón en una ventana puede utilizarse la librería **pygame**. El siguiente código permite obtener la posición del ratón:

```
# creacion de una ventana de pygame
import pygame
from pygame.locals import *

WIDTH = 64
HEIGHT = 480

screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Theremin")
...
# obtencion de la posicion del raton
for event in pygame.event.get():
    if event.type == pygame.MOUSEMOTION:
        mouseX, mouseY = event.pos

pygame.quit()
```

4. En este ejercicio implementaremos un *sintetizador armónico*, i.e., un sintetizador que produce acordes (combinaciones de notas simultáneas) en vez de notas aisladas como explicaremos más adelante. Nuestro sintetizador utilizará la *escala diatónica* (las teclas blancas del piano: *c-d-e-f-g-a-b-c*) y podrá utilizar dos tipos de afinación, justa y temperada, según la siguiente tabla:

	A	B	C	D	E	F	G	a
Afinación justa	440	495	550	586.67	660	733.33	825	880
Relaciones	1	9/8	5/4	4/3	3/2	5/3	15/8	2
Afinación temperada	440	493.88	554.36	587.33	659.26	739.99	830.61	880
Relaciones	1	$2^{2/12}$	$2^{3/12}$	$2^{5/12}$	$2^{7/12}$	$2^{8/12}$	$2^{10/12}$	$2^{12/12} = 2$

Recordemos que con esta tabla podemos obtener todas las octavas: podemos subir una nota una octava multiplicando su frecuencia por 2, y bajarla una octava dividiendo entre dos.

Nuestro sintetizador producirá las *tríadas (acordes) elementales de la escala diatónica* que se forman a partir de una nota cualquiera, añadiendo otras dos por encima *en saltos de 2*. Por ejemplo, a partir de la *nota C* obtenemos el *acorde C* con las notas *C-E-G*, y a partir de la *nota A* obtenemos el *acorde A* con las notas *A-C-E*. Como es habitual, utilizaremos las teclas *zxcvbnm* para una octava (baja) y *qwertyu* para la superior (alta). Cuando se pulse *z* sonará el acorde *C*.

Como generador de sonido podemos utilizar el modelo de síntesis FM combinado con las envolventes ADSR (ambos vistos en clase).

Discusión: ¿Qué afinación suena mejor, la justa o la temperada?

5. **(Entregable)** Implementar un piano **polifónico** con síntesis por tabla de onda con el algoritmo de Karplus-Strong visto en clase. La nota producida por el algoritmo visto en clase está definida en función del tamaño de la tabla de onda. Es fácil ver qué relación existe entre ese tamaño y la nota producida.

Una forma de conseguir la polifonía es manejar un array de canales (streams) a donde se enviarán las notas obtenidas por pulsación de teclado. El tamaño de dicho array determina el número máximo de notas simultáneas que pueden sonar (la polifonía de nuestro piano).

6. **(Entregable)** Implementar un sampler que permita cargar *muestras anotadas* con intervalo de *sustain* de modo que pueda alargar las notas el tiempo deseado haciendo loop sobre dicho intervalo. Este sintetizador comenzará a reproducir una nota con un mensaje *noteon* (con una frecuencia dada) y utilizará el sample para producir esa nota, que se mantiene en el tiempo utilizando reproduciendo en loop la región *sustain* y termina (sale de dicha región) con el mensaje *noteoff*.