

PRIMERA PRÁCTICA SONIDO

1. Objetivo general:

El objetivo de esta práctica es que el alumno se familiarice con los conceptos de amplitud y frecuencia y los llegue a dominar, así como el efecto que tiene la variación de estos parámetros sobre el sonido resultante.

Rango de frecuencias audibles.

Respuesta en frecuencia del oído.

Sonidos graves, medios y agudos.

Secuencias de sonido de distintas frecuencias, amplitudes y duraciones (melodía).

Combinación de sonidos simultáneos con frecuencias distintas entre sí (armonía).

PARTE 1: RANGO DE FRECUENCIAS AUDIBLES Y RESPUESTA EN FRECUENCIA DEL OÍDO.

Objetivos:

Con esta práctica se trata de que el alumno sea consciente de la respuesta auditiva de su propio oído y de la relación que existe entre volumen, amplitud y frecuencia. Al mismo tiempo debe concretar sus ideas de frecuencias bajas, medias y altas a rangos determinados por la frecuencia.

Explicación:

Las curvas isofónicas (de igual sonoridad) muestran la dependencia que existe entre amplitud y frecuencia. Es decir, percibimos distinto volumen no sólo debido a diferencia en amplitud, sino también a diferencias en frecuencia. Esta curva representa la respuesta en frecuencia del oído. Como parámetro estándar se el volumen percibido de cualquier sonido con el volumen al que se percibe un tono de 1 Khz de frecuencia. Si la curva nos indica que necesitamos muchos más decibelios que el tono patrón para que se escuche al mismo volumen, esto indica que a nuestro oído le cuesta detectar los sonidos con dicha frecuencia.

Tarea:

Copia el siguiente fichero con un nombre acabado en “.csd” o descarga el fichero audiodB.csd de la página moodle de la asignatura.

```
<CsoundSynthesizer>
```

```
<CsOptions >
```

```
;salida a un archivo WAV de nombre audiciondB.wav
```

```
-o audiciondB.wav
```

```
</CsOptions >
```

```
<CsInstruments>
```

```
;sample rate
```

```
sr = 44100
```

```
;control rate
```

```
ksmps = 1
```

```
;number of channels
```

```

nchnls = 1
;definición del instrumento 1
instr 1
;asignamos a la variable iamplitude el valor de p4
iamplitude = p4
iampdB=ampdb(iamplitude)
;asignamos a la variable ifrequency el valor de p5
ifrequency = p5
;asignamos a la variable iphase el valor 0
iphase = 0
;asignamos a la variable asound la salida del opcode oscili
asound oscils iampdB, ifrequency, iphase
;enviamos asound a la salida de audio
out asound
;fin de la definición del instrumento 1
endin
</CsInstruments>

<CsScore>
;el valor de p1 indica que la nota se creará con el instrumento 1
;el valor de p2 indica que la nota empezará para tiempo=0
;el valor de p3 indica que la nota durará 5 segundos
;el valor de p4 indica la amplitud en dB
;el valor de p5 indica la frecuencia de la nota
;p1 p2 p3 p4 p5
i1 0 1 50 32
i1 1 1 50 64
i1 2 1 50 128
i1 3 1 50 256
i1 4 1 60 512
i1 5 1 50 1024
i1 6 1 50 2048
i1 7 1 50 4096
i1 8 1 50 8192
i1 9 1 50 16384
e
; e indica fin de la partitura
</CsScore>

</CsoundSynthesizer>

```

El parámetro p1 indica el instrumento a utilizar, el parámetro p2, indica el instante de comienzo en segundos, el parámetro p3 indica la duración, p4 la amplitud en decibelios (dB) y p5 la frecuencia de la nota.

En clase: Generar una curva isofónica a partir del fichero ejemplo. Para ello hay que ejecutar el programa cambiando las amplitudes hasta que el alumno escuche todas las notas con el mismo volumen.

En casa:

- 1. modificar el fichero para incluir al menos el doble de frecuencias a medir y hacerlo en cuatro situaciones, según la intensidad del tono de 1024 Hz valga: 20, 40, 60 y 80 dB.***
- 2. Ejecutar dicho programa (con las nuevas frecuencias) y ajustar los valores de la amplitud para que todas las notas se escuchen al mismo volumen. Es decir, hay que generar cuatro curvas diferentes.***
- 3. Construir una gráfica con octave que represente la amplitud de cada una de las cuatro curvas obtenidas en dB frente a la frecuencia.***
- 4. Indicar cuales son las frecuencias bajas, medias y altas, según tu percepción. Indicar que frecuencias escuchas a un nivel muy bajo tanto por el lado de las frecuencias altas como por el lado de las frecuencias bajas.***

PARTE 2: DURACIÓN Y RITMO

Objetivo:

El alumno debe comprender el efecto que la duración de un sonido tiene en su percepción del mismo.

Explicación:

Las notas se clasifican por su duración. En primer lugar se debe fijar una duración de una “nota de referencia”. Por ejemplo, la duración de la nota de referencia es 1 segundo. Esto genera que las posibles notas más rápidas duren: 0.5 s., 0.25 s., 0.125. s, 0.0625. s., la duración de las notas más lentas sería de 2 y 4 segundos. Si cambiamos el tiempo de la nota de referencia a 0.75 s., por ejemplo, deberá cambiar también el de sus múltiplos y submúltiplos. Tengamos en cuenta que el caso mas rápido presentado en este ejemplo permitiría ejecutar 16 notas por segundo.

Esta clasificación se puede aplicar también a sonidos generados artificialmente. No obstante, en este caso, el autor tiene libertad total para decidir la duración de los mismos.

Tarea:

Ahora vamos a utilizar una escala lineal de amplitudes, por lo que debemos tener en cuenta las siguientes líneas:

```
instr 1
;asignamos a la variable iamplitude el valor de p4
iamplitude = p4

;asignamos a la variable asound la salida del opcode oscili
asound oscils iamplitude, ifrequency, iphase
```

Partiendo del fichero anterior, (o del fichero duracion.csd) modificar la “partitura” del fichero de la siguiente manera.

```
<CsScore>
;el valor de p1 indica que la nota se creará con el instrumento 1
;el valor de p2 indica que la nota empezará para tiempo=0
;el valor de p3 indica que la nota durará 5 segundos
;el valor de p4 es la amplitud (en escala lineal)
;el valor de p5 es la frecuencia de la nota
;p1 p2 p3 p4 p5
i1 0 0.5 7000 493
i1 1 0.5 7000 392
i1 2 1 7000 261
e
; e indica fin de la partitura
</CsScore>

</CsoundSynthesizer>
```

Construir varios ficheros, tal que las duraciones y frecuencias de cada una de las notas sean variables siguiendo algún patrón determinado por el alumno.

PARTE 2: ESCALA MUSICAL

Objetivo:

A través de esta práctica el alumno debe comprender que la separación entre dos sonidos correspondientes al mismo tono no es constante sino que aumenta con la frecuencia. Por otro lado, el conocimiento de las frecuencias fundamentales de la escala cromática y el conocimiento de las frecuencias utilizadas en las cuatro octavas más comunes le permitirá tener criterios a la hora de poder generar sonidos más complejos .

Explicación:

En la imagen precedente podemos ver una escala de 12 notas. Para generar la siguiente escala mas aguda es necesario multiplicar la frecuencia de cada nota por 2, y para generar una octava más grave, dividir la frecuencia de la nota por 2.

Tarea:



Si	493.88
Sib (La#)	466.16
La	440.00
Lab (Sol#)	415.30
Sol	392.00
Solb (Fa#)	369.99
Fa	349.23
Mi	329.63
Mib (Re#)	311.13
Re	293.66
Reb (Do#)	277.18
Do	261.63

Utilizando el fichero “escala.csd” generar cuatro escalas, la anterior a la dada en la imagen, y las tres posteriores. Situar cada una de las escalas en las curvas isofónicas obtenidas y decir en cuál de las tres regiones en que se puede dividir la respuesta se encuentra cada una de ellas.

PARTE 4: Composición de sonidos complejos

(armonía)

Objetivo:

El objetivo de esta práctica es que el alumno comprenda qué sonidos puede combinar entre sí para producir sonidos agradables o poco agradables al oído. Se explorarán tres casos: Frecuencias cercanas entre sí, frecuencias sin ninguna relación de multiplicidad entre ellas y frecuencias múltiplos entre sí (armónicas).

Explicación:

Tareas:

Problema de tonos con frecuencias próximas:

Utilizando el fichero *armo1.csd* comprobar si dos notas con frecuencias muy cercanas se escuchan bien o se puede reconocer algún tipo de “efecto” no deseable en las mismas. Hacerlo para frecuencias en el entorno de 200, en el entorno de 800 y en el entorno de 3400 Hz). ¿Ocurre el mismo fenómeno siempre?. Con los valores de las frecuencias utilizadas componer una señal de 600 puntos en octave y representarla gráficamente. ¿Se observa alguna pauta que permita explicar el efecto?.

La partitura debe modificarse de la siguiente manera.

```
<CsScore>
;el valor de p1 indica que la nota se creará con el instrumento 1
;el valor de p2 indica que la nota empezará para tiempo=0
;el valor de p3 indica que la nota durará 5 segundos
;el valor de p4 es la amplitud de la nota en escala lineal
;el valor de p5 es frecuencia de la nota
;p1 p2 p3 p4 p5
i1 0 1 7000 500
i1 0 1 7000 512
e
; e indica fin de la partitura
</CsScore>

</CsoundSynthesizer>
```

Problema de tonos con frecuencias que no guarden relación de multiplicidad entre sí:

Utilizando el fichero *armo2.csd* comprobar con el siguiente programa que ocurre si las frecuencias no guardan ninguna relación de multiplicidad entre sí (al dividir la frecuencia mayor por la menor el número resultante no es entero). Hacerlo para varios conjuntos de frecuencia diferentes (¿Que conclusión sacas?).

La partitura debe modificarse como sigue:

```
<CsScore>
;el valor de p1 indica que la nota se creará con el instrumento 1
;el valor de p2 indica que la nota empezará para tiempo=0
;el valor de p3 indica que la nota durará 5 segundos
;el valor de p4 es 10000
;el valor de p5 es 440
;p1 p2 p3 p4 p5
i1 0 1 7000 500
```

```
i1 0 1 7000 610
```

```
e
```

Problema de tonos con frecuencias que no guarden relación de multiplicidad entre sí:

Utilizando el fichero armo3.csd modificar la partitura para que cree un conjunto de tonos que se van a ejecutar progresivamente pero que acaban todos a la vez. La característica más importante es que la frecuencia de todos los tonos añadidos debe ser múltiplo de la primera de ellas. ¿Mejora el sonido al añadir armónicos?

```
<CsScore>
```

```
;el valor de p1 indica el instrumento con el que se creará la nota
```

```
;el valor de p2 indica el instante de comienzo
```

```
;el valor de p3 indica que la duración de la nota
```

```
;el valor de p4 es la amplitud en escala lineal
```

```
;el valor de p5 es la frecuencia de la nota.
```

```
;p1 p2 p3 p4 p5
```

```
i1 0 5 7000 440
```

```
i1 1 4 5000 880
```

```
i1 2 3 2000 1320
```

```
i1 3 2 1000 1760
```

```
i1 4 1 500 2200
```

```
e
```

```
; e indica fin de la partitura
```

```
</CsScore>
```

```
</CsoundSynthesizer>
```

*Obsérvese que la amplitud de los armónicos de mayor frecuencia es bastante menor que la de los armónicos de baja frecuencia. **Crear un sonido nuevo modificando el número de armónicos y sus amplitudes.** La primera de las frecuencias tal que todas las demás son múltiplos de ella se denomina fundamental y también puede ser cambiada por el alumno.*

PARTE 5: Combinación de tonos complejos con distintas amplitudes

Objetivo:

A partir del concepto de armónico de la práctica anterior el alumno debe comprender que la amplitud de cada armónico tiene una importancia primordial sobre el sonido generado y que de hecho, la variación de las amplitudes de los armónicos modifica el timbre del sonido.

Tarea:

A partir de los ficheros anteriores (ejemplo fichero aditiva.csd) generar un fichero nuevo que genere una melodía. El fichero .csd se puede generar a partir de los anteriores con la siguiente modificación:

```
;primer parcial o fundamental
```

apartial1 oscils iamplitude, ifrequency, iphase
;segundo parcial, frecuencia doble, amplitud 0.8 del parcial 1
*apartial2 oscils iamplitude * 0.8, ifrequency * 2 , iphase*
;añadimos pequeñas desafinaciones
*apartial3 oscils iamplitude * 0.5, ifrequency * 4 , iphase*
*apartial4 oscils iamplitude * 0.7, ifrequency * 8 , iphase*
*apartial5 oscils iamplitude * 0.4, ifrequency * 16 , iphase*
asound = apartial1 + apartial2 + apartial3 + apartial4 + apartial5

*De esta manera, cada vez que ejecutemos una nota, se ejecutará un sonido con una frecuencia fundamental y 4 armónicos. Cambiando el valor de ifrequency, cambiaremos la tonalidad de cada nota. **Implementar la melodía creada para el apartado 2 con un programa creado a partir de esta modificación. El alumno tiene libertad total para modificar las amplitudes (factor que multiplica a iamplitude) y las frecuencias (factor que multiplica a ifrequency) de cada nota.***