

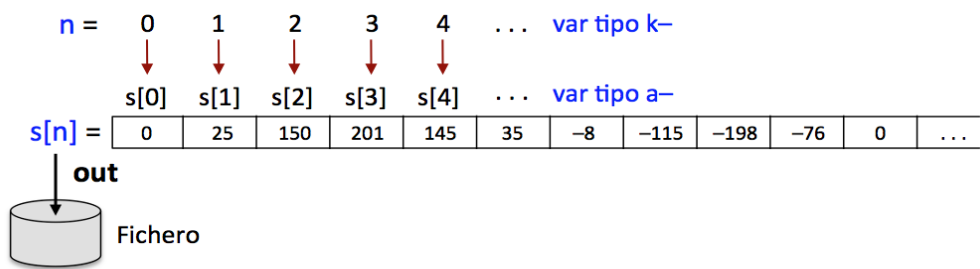
[Área personal](#) / [Mis cursos](#) / [SMC 21028](#) / [Práctica 5: audio digital: lectura, descripción y generación](#) / [Cuestionario de la segunda sesión P5 \(P\)](#)

Comenzado el	miércoles, 11 de marzo de 2020, 11:13
Estado	Finalizado
Finalizado en	miércoles, 11 de marzo de 2020, 12:53
Tiempo empleado	1 hora 39 minutos
Calificación	6,5 de 10,0 (65%)

Información

Creación de una onda sinusoidal

Vamos a generar una onda sinusoidal y a almacenarla en un fichero WAV. El proceso se resume en la siguiente imagen, donde n es el número de muestra y $s[n]$ la señal digital que vamos creando:



Como queremos crear una senoide, la operación que debemos usar para calcular cada muestra a partir de su número de muestra es:

$$s[n] = A \sin(2 \pi f n / f_s) ,$$

siendo la frecuencia de muestreo **sr** en Csound, f la frecuencia de la onda y A su amplitud, constantes.

¿Cómo obtener n ? El operador **timek** lee cuántas veces ha iterado el bucle de control y, si hacemos **ksmps** = 1, entonces el valor que nos devuelve **timek** = número de la muestra que se está calculando.

Ejercicio 5.6:

Para calcular las muestras y grabarlas en la salida, sigue las siguientes instrucciones:

- Descarga la [PLANTILLA](#) (botón derecho → guardar como)
- Se dan ya las inicializaciones y comentarios con las acciones necesarias:
 - Amplitud: viene en **p4** en dBFS y la pasamos a PCM,
 - Frecuencia: viene en **p5** como octava.semitono y la pasamos a Hz, y
 - π : el valor de pi lo creamos dándole su valor a una constante *ipi*.
- Hay que leer el número de muestra con **timek**.
- Luego se calculan una a una las muestras de la señal implementando la ecuación de arriba.
- La señal así creada se enviará a la salida con **out**.

La partitura:

- Contendrá una orden de control del tempo para establecerlo desde el inicio y una orden de activación del instrumento "**seno**" desde el tiempo 0. Para los detalles, sigue las instrucciones de la pregunta siguiente.
- Cuando compiles, como estás creando una onda nueva, es importante verificar en la consola que "**overall samples out of range 0**".
- Para ver que realmente tiene aspecto de onda sinusoidal, abre un visor que hemos preparado para ello pulsando el botón "Widgets" en la barra de botones de CsoundQt. Si no funcionara, ejecuta con "Render" para generar el fichero WAV y ábrelo con Audacity.

Pregunta **1**

Finalizado

Puntúa 1,0 sobre 2,0

Información

- Tempo = 90 BPM desde el inicio.
- Duración de la activación del instrumento = 1 negra ligada con corchea con puntillo (en 4/4).
- Amplitud = −3.6 dBFS
- Debe emitir la frecuencia de un MI₃ (usa la notación octava.semitono de Csound).

Escribe aquí los parametros que utilices en tu partitura tras el nombre del instrumento. Utiliza únicamente un espacio en blanco para separar instrucciones y parámetros entre sí.

Por ejemplo, si tu orden de partitura es: **i "seno" 0 1.0 -0.5 8.00** escribe en la respuesta: 0 1.0 -0.5 8.00

- Parámetros de la orden de control del tempo: t
- Parámetros de la orden de control del instrumento: **i "seno"**

Sinusoide de frecuencia variable

Una vez sabemos calcular una onda sinusoidal, no es difícil hacer que su frecuencia vaya cambiando. Basta con hacer que la frecuencia en la función $\text{sen}(2\pi f n / f_s)$ sea variable, en vez de constante.

Ejercicio 5.7:

Para este ejercicio se reusará el instrumento del ejercicio anterior, haciendo algunos cambios en el instrumento y en la partitura.

- Guarda el proyecto anterior como `ej5.7.csd`, y cambia el fichero de salida en las opciones a `ej5.7.wav`,

Instrumento:

- Cambia el nombre de instrumento (ahora **senovar**).
- Quita el conversor de altura a Hz (**cpspch**) porque ahora desde la partitura enviarás la frecuencia inicial, ya en hercios en **p5**, que se asignará a *iFrec*.
- Ahora, el valor de frecuencia *iFrec* será el inicial y el valor final se establecerá, según la indicación de la pregunta siguiente.
- Antes de calcular la fase, tienes que crear una frecuencia variable mediante un operador **line**:

```
kFrec line iFrec, p3, AQUÍ_LA_FRECUENCIA_FINAL
```

que creará una evolución lineal de la frecuencia a lo largo de la duración de la activación (en **p3**), desde el valor inicial enviado desde la partitura hasta el final indicado (que lo calcule Csound, no tú). Esta variable *kFrec* es la que pondrás ahora en el cálculo de la fase de la onda seno.

- En algún sitio después del cálculo de la muestra, inserta el siguiente código:

```
if kn==sr then
    printks "Frecuencia: %d Hz\n", 0, kFrec
endif
```

que escribirá en la consola el valor de la frecuencia al cabo de un segundo (por definición, cuando el número de muestras calculadas sea igual a la frecuencia de muestreo, **sr**).

En la partitura:

- Cambia la activación anterior para que active el instrumento "senovar" según las instrucciones de la siguiente pregunta.

Pregunta **2**

Finalizado

Puntúa 1,0 sobre 1,0

- Usa el tempo de 60 BPM desde el inicio (1 tiempo = 1 segundo)
- Duración de la activación = 6 tiempos
- Amplitud = −10 dBFS
- Frecuencia inicial = 230 Hz
- Frecuencia final (para el instrumento) = 1 octava por encima de la inicial

Escribe aquí (redondeando a entero) el valor de la frecuencia al cabo de 1 s, escrita en la consola:

Hz

Información

Generación de tonos DTMF

La marcación por tonos multi-frecuencial (dual-tone multi-frequency dialing) codifica cada una de las teclas del teléfono con una pareja de tonos sinusoidales de frecuencias no relacionadas armónicamente. Las frecuencias que se utilizan son las de la siguiente tabla:

	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

Así, cuando se teclea un cero suenan simultáneamente dos tonos sinusoidales: uno de 941 Hz y otro de 1336 Hz. Vamos a generar estos tonos con Csound.

Ejercicio 5.8:

- Descarga el código que hay en la siguiente [PLANTILLA](#)

El instrumento:

- Recibirá una amplitud en dBFS (en **p4**), que es convertida a PCM, y las frecuencias de los 2 tonos en Hz (en **p5** la frecuencia baja y en **p6** la alta).
- La especificación técnica de los DTMF dice que el tono alto debe sonar +2 dB más fuerte que el tono bajo. Hazlo.
- Debes construir 2 ondas sinusoidales por separado, cada una con su amplitud y su frecuencia, igual que antes hacías una.
- Luego las debes sumar en una onda única antes de enviarlas a la salida.

La partitura:

- Deberá contener una activación por cada uno de los dígitos que tienes que marcar, según la siguiente pregunta. Añade las activaciones que necesites.

Pregunta **3**

Finalizado

Puntúa 1,0 sobre 3,0

Programa la partitura para que marque el número 915701354

Responde a las siguientes preguntas:

1. ¿Cuál es la amplitud del tono alto en PCM? (ver consola, sin decimales):

13045
2. ¿Cuál es la frecuencia baja del primer dígito? :

0,300

Hz
3. ¿Cual es la frecuencia alta del último dígito? :

23402,4

Hz

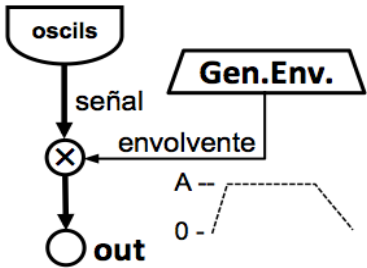
Información

Escala sin *clicks*

Cuando una onda es generada sin envolvente, produce problemas de ruidos en sus extremos. Estos están causados por saltos bruscos en la señal, sobre todo al terminar en valores diferentes de cero. Vamos a modificar un instrumento para añadir envolventes a los sonidos y resolver así ese problema.

Se aplicará una envolvente tipo ASR a cada nota de una escala, lo que asegura que el inicio y el final sean suaves. A diferencia de la aplicación de la envolvente al sonido cargado de fichero, ahora se aplica sobre la onda creada por un operador **oscils**. Usarás el operador **linen** para crearla.

La envolvente para cada nota se almacena en una variable (de tipo **k-**) que multiplica la onda generada antes de enviarla a la salida (figura).



Ejercicio 5.9:

- Descarga la [PLANTILLA](#) de este nuevo proyecto (botón derecho → guardar como).
- Esta plantilla ya es compilable. Hazlo y oirás una escala. Esta escala tiene ruidos en los inicios y finales de las notas. Eso es lo que vamos a solucionar aplicando envolventes a cada una.
- Añádele la envolvente al código del instrumento: creación con **linen** y que multiplique a la señal generada por **oscils**. Los parámetros de la envolvente los tienes en la pregunta siguiente.
- Compila de nuevo y comprueba que los *clicks* han desaparecido.

Pregunta 4

Finalizado

Puntúa 1,0 sobre 1,0

La envolvente tendrá los siguientes prámetros:

- Amplitud: $A_S = 0.8$,
- Tiempo de ataque: $t_A = 0.05$ s,
- Tiempo de relajación: $t_R = 0.12$ s
- La duración será el valor de **p3** (toda la duración de la nota).

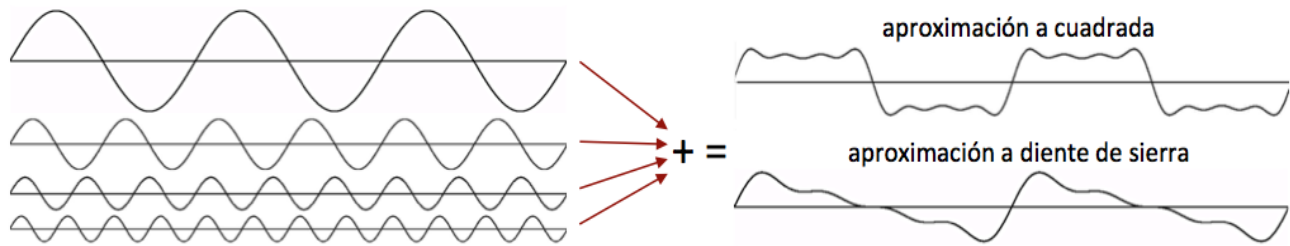
Indica a continuación la amplitud de las notas generadas (en "overall amps" en la consola): 23363,6

Información

Cálculo de ondas compuestas geométricas

Las ondas compuestas son aquellas cuyo espectro tiene más de un único parcial: todas las que no sean una senoide. Vamos a crear ondas compuestas "geométricas" sumando ondas sinusoidales de frecuencias armónicas.

Para generar una aproximación a una geométrica se usan los P primeros armónicos, cada uno de ellos con una amplitud que depende del número del armónico (ver material previo). Según los armónicos que se sumen y con qué amplitud se consiguen aproximaciones a diferentes ondas geométricas:



Ejercicio 5.10:

- Descárgate la [PLANTILLA](#) de este ejercicio (botón derecho → guardar como)
- Las opciones y variables de cabecera ya están escritas en el código.
- En el instrumento, las variables de tipo i- también están programadas:
 - En **p4** vendrá la amplitud en dBFS
 - En **p5** la frecuencia del primer armónico (la fundamental) en Hz.
 - Además tiene el valor de π en ipi.
- La lectura del número de muestra se hará como en casos anteriores, con **timek**.
- Cada uno de los armónicos será calculado con su propia función **sin()**, por lo que es más cómodo y compacto calcular antes la fase de la fundamental (en una variable $kFase$) y luego generar los armónicos (número de armónico $\times kFase$) multiplicados por sus respectivas amplitudes. Matemáticamente:

En vez de esto... (para 2 parciales)	hacer esto (para 2 parciales)
$A \times (A_1 \text{ sen}(2\pi f n/f_s) + A_2 \text{ sen}(2\pi 2f n/f_s))$	$\Phi = 2\pi f n / f_s$ $A \times (A_1 \text{ sen}(\Phi) + A_2 \text{ sen}(2\Phi))$

donde la amplitud A es la global de la onda (la que se le pasa desde la partitura) y las amplitudes A_p son las de cada armónico (1, 1/2, 1/3, 1/4, 1/5, 1/6, etc., para el caso de la diente de sierra y 1, 1/3, 1/5, 1/7, 1/9, etc., para la cuadrada).

- Prográmalo según las instrucciones de la pregunta siguiente.
- Compila, escúchalo y abre el visor de "Widgets" de CsoundQt, donde hemos preparado un visor para que puedas ver la forma de onda, para comprobar que se aproxima adecuadamente a la que se pide.

Pregunta 5

Finalizado

Puntúa 2,5 sobre 3,0

- Crea una aproximación con 5 armónicos a una onda diente de sierra.
- Cambia la partitura del proyecto para que tenga tempo 70 BPM y active el nuevo instrumento (inicio en cero y duración de redonda ligada a semicorchea (en 4/4)) con amplitud -5.0 dBFS y frecuencia 310 Hz.
- Escribe los valores de las frecuencias parciales que componen la señal, según aparezcan en la consola:

310 Hz 620 Hz 930 Hz 1240 Hz 1550 Hz

- Escribe la duración en tiempo real de la activación del instrumento (aparecerá en la consola en azul entre **TT** y **M**):

3,643 s

◀ Entrega de la primera sesión P5 (P)

Ir a...

Entrega de la segunda sesión P5 (P) ▶

