

Comenzado el martes, 17 de marzo de 2020, 11:48

Estado Finalizado

Finalizado en martes, 17 de marzo de 2020, 15:00

Tiempo empleado 3 horas 12 minutos

Calificación 10,0 de 10,0 (100%)

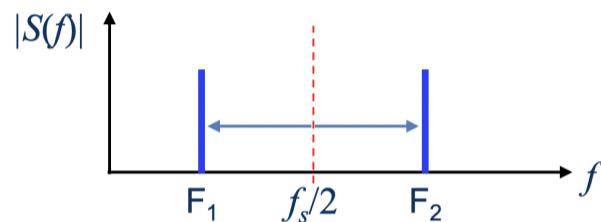
Información

Estudio del aliasing

Vamos a comprobar que la generación de ondas por encima de Nyquist ($f_s/2$) provoca el efecto de plegamiento de las frecuencias o *aliasing*. Generaremos ondas sinusoidales por debajo y por encima de Nyquist para comparar el resultado.

IDEA DEL EJERCICIO:

Para cada activación del instrumento generar una onda que sea la primera mitad de una frecuencia dada y la otra mitad de una frecuencia que esté por encima de Nyquist, pero que suene igual que la de la primera mitad. Para ello, ambas tendrán que estar a la misma distancia de la de Nyquist.



Estas dos frecuencias sonarán igual al estar a la misma distancia de la de Nyquist: F_1 con su valor y F_2 con el valor de $F_1 = f_s - F_2$ por *aliasing*.

Ejercicio 5.11:

- Descarga la estructura de este ejercicio: [PLANTILLA](#) (botón derecho → guardar como)
- Establece el valor de la frecuencia de muestreo (**sr**) para que sea 8 kHz.

En el instrumento:

- La amplitud PCM llegará desde la partitura en **p4**.
- La altura llegará en **p5** en octava.semitono que hay que convertir en Hz con **cpspch(·)**.
- Debes leer en cada ciclo el tiempo en segundos con **times**.
- **Si** el tiempo < tiempo_de_inicio + duración/2 **entonces** ; (recuerda que el tiempo de inicio se recibe en **p2**)
 - generarás la onda de frecuencia enviada desde la partitura (la F_1 de la figura) y
- **si no**,
 - generarás la que esté por encima de Nyquist que suene igual que F_1 (la F_2 de la figura) pero con la mitad de amplitud.
- Debes generar la onda sinusoidal con el operador **oscils**, usando la amplitud y frecuencia adecuadas y la fase inicial = 0.
- Si está bien hecho, ambas frecuencias sonarán igual y sólo se distinguirán por su diferente amplitud.
- Envía esta onda a la salida con **out**.

Partitura:

- Activa el instrumento 3 veces seguidas (sin pausas) para que genere 3 notas de 1 tiempo, según las especificaciones de la pregunta que sigue a esta descripción. Recuerda que cada par de frecuencias generada en cada activación deben de sonar iguales, salvo en su amplitud.
- Recuerda cuando compiles verificar en la consola que obtienes "overall samples out of range 0".
- Indica el valor de las frecuencias F_1 y F_2 para cada una de estas 3 notas en el cuestionario. Recuerda que puedes añadir los **print** que necesites en el instrumento para mostrar estos valores en la consola.

Pregunta 1

Finalizado

Puntúa 3,0
sobre 3,0

Todas las amplitudes serán de 15000 en PCM.

Las frecuencias a continuación se piden redondeadas a enteros.

1. Nota 1: Si_b₃ - frecuencia = Hz → Frecuencia por encima de Nyquist que suena igual:
2. Nota 2: SOL₅ - frecuencia = Hz → Frecuencia por encima de Nyquist que suena igual:
3. Nota 3: RE#₇ - frecuencia = Hz → Frecuencia por encima de Nyquist que suena igual:

Información

Gestión del panorama estéreo

Con Csound podemos coloar un sonido en cualquier posición del espacio 3D. En este ejercicio seremos más modestos y sólo vamos a trabajar en el panorama 2D estéreo, como lo vimos en el tema de psicoacústica.

La teoría nos dice que para mover un sonido en el panorama izquierda-derecha un criterio que mantiene la potencia del sonido es aplicar un factor de ganancia que depende del ángulo de azimut θ de la siguiente manera:

- en el extremo izquierdo del panorama $g_L = 1, g_R = 0$
- en el extremo derecho del panorama $g_L = 0, g_R = 1$
- en el centro del panorama $g_L^2 + g_R^2 = 1$ (para conservar la sonoridad)

La forma de cumplir todas estas condiciones es hacer que $g_L = \sin^2(\theta)$ y $g_R = \cos^2(\theta) = 1 - \sin^2(\theta)$.

Ejercicio 5.12:

Vamos a generar una señal que controle el panorama, haciéndolo "oscilar" un lado a otro, para posicionar así una señal. Para ello, sigue las siguientes instrucciones:

- Descarga la plantilla para este ejercicio: [PLANTILLA](#) (botón derecho → guardar como).
- La cabecera se da ya preparada:
 - Frecuencia de muestreo **sr** = 44.100 muestras/s
 - **ksmps** = 1, para que las señales se generen muestra a muestra.
 - **nchnls** = 2. Generaremos sonido estéreo.

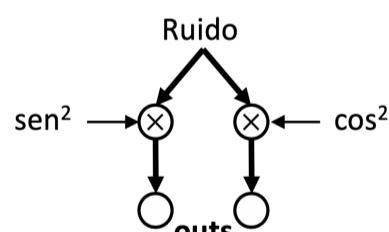
Para el **instrumento**:

- Llámalo **panorama**.
- Debe recibir los siguientes valores:
 - En **p4**: recibirá la amplitud del sonido en PCM.
 - En **p5**: recibirá la frecuencia de oscilación del ángulo panorámico.
- Una vez recibidos y asignados esos valores, debes crear un ruido rosa, que va a ser la señal que vamos a panoramizar (el ruido se sitúa mejor que un tono). Usa para ello el operador **pinkish**, al que indicarás la amplitud PCM del sonido.
- Para crear la señal oscilante que controla el panorama usarás la fórmula ya conocida:

$$\sin(2\pi f_p n / f_s)$$

donde f_p es la frecuencia de oscilación del panorama, n es el contador del bucle de control, que debes haber leído previamente con **timek** (almacena su valor en **kn**) y f_s es la frecuencia de muestreo del sistema, **sr**.

- Esa señal oscilante al cuadrado será directamente g_L , mientras que g_R será $= 1 - g_L$.
- Sólo falta sacar las ondas fuera, una por cada canal, con el operador para sonido estéreo **outs**. Para ello debes seguir el siguiente esquema:



- Al final hay código para escribir en consola los valores de las ganancias a los 0.5 s. Debes añadir las variables que has usado para las ganancias g_L y g_R .

En la partitura:

- Debes escribir 1 nota para el instrumento panorama con los parámetros de la pregunta siguiente.

Pregunta **2**

Finalizado

Puntúa 2,0
sobre 2,0

- Usa el tempo de 60 BPM desde el inicio (1 tiempo = 1 segundo)
- Duración de la activación = 5 tiempos
- Amplitud = 20000 PCM
- Frecuencia del panorámico = 0.26 Hz

Escribe aquí (redondeando con 3 decimales) los valores de las ganancias al cabo de 0.5 s, escritas en la consola:

$g_L = 0,531$

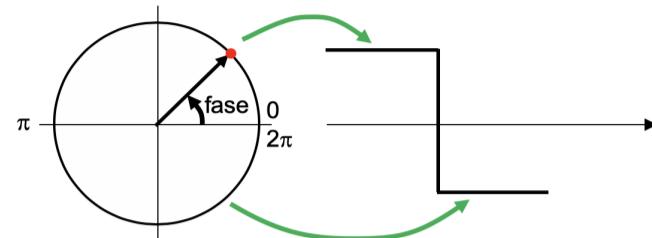
$g_R = 0,469$

Información

Ondas cuadradas

Podemos crear aproximaciones a ondas cuadradas combinando armónicos o hacerlo mediante operaciones no lineales, lo cual es más simple, aunque con riesgo de falta de control sobre su espectro. Vamos a probarlo de esta forma. La idea es:

- Crear una fase variable con $2\pi f n / f_s$
- Si la fase está entre 0 y π se pone la salida a +Amplitud y si está entre π y 2π se pone a -Amplitud.



- Luego añadiremos una envolvente ASR para evitar ruidos en el inicio y final.

Ejercicio 5.13:

- Para este ejercicio descarga la siguiente [PLANTILLA](#) (botón derecho → guardar como).
- Se da creado el instrumento **cuadrada**.

Instrumento:

- Recibirá la amplitud de la onda en **p4** en dBFS para convertir en PCM con **ampdbfs(·)**
- Recibirá la frecuencia ya en Hz en **p5**.
- Se creará la fase variable sobre una variable de tipo **k-**. Como en ejercicios anteriores, la expresión usa el contador, **n**, de número de muestras, que puedes leer con **kn timek**. Recuerda que la frecuencia de muestreo es **sr**.
- La condición para saber si la fase está en la primera mitad o en la segunda del círculo será:

SI fase % (2π) < π **ENTONCES**. ; (el % es la operación "resto de dividir entre")

- Si se cumple la condición, la señal valdrá +A y si no -A. Así la onda cuadrada estará creada.
- Para aplicarle una envolvente de tipo ASR debes usar el operador **linen**, que puede usarse como un modificador de señal, si se usa de la siguiente manera:

SEÑAL DE SALIDA **linen** **SEÑAL DE ENTRADA**, tiempo de ataque, duración, tiempo de relajación

- Los tiempos **t_A** y **t_R**, deben ser los de la pregunta que sigue.
- La señal modificada debes sacarla con **out**.

Partitura:

- Programa la partitura para que llame al instrumento "cuadrada" con una activación de 4 tiempos desde el inicio, según lo especificado a continuación.
- La plantilla dispone de un visor para ver las formas de onda que estás generando (botón visualizar Widgets del menú superior), para comprobar que es realmente una onda cuadrada.

Pregunta 3

Finalizado

Puntúa 2,0
sobre 2,0

- Tempo = 100 BPM
- Amplitud = -7 dBFS
- Frecuencia = 260 Hz
- Tiempo de ataque: **t_A** = 0.30 s
- Tiempo de relajación: **t_R** = 0.20 s

Escribe aquí (redondeando a entero) el valor máximo de amplitud PCM generado:

Amplitud máxima PCM = 14637

Escribe el valor de duración en tiempo real de la nota (impresa en azul en la consola entre **TT** y **M**):

Duración = 2,400

Información

Generación de ruidos mediante números aleatorios

Todos los ruidos estacionarios están basados en procesos aleatorios que podemos modelar en Csound. Los conceptos que vamos a utilizar para ello son los siguientes:

- **Ruido blanco:** es el más sencillo de todos. Es una sucesión de números aleatorios.
- **Ruido rosa:** el algoritmo es mucho más complejo, pero podemos aproximarla filtrando ruido blanco con un pasa-baja suave (su potencia cae -3 dB/octava).
- **Ruido rojo (o marrón):** se puede generar mediante el denominado paseo aleatorio (*random walk*) que consiste básicamente en decidir cada vez si aumentamos o disminuimos el valor de la muestra respecto a la anterior en una cantidad fija.

Ejercicio 5.14:

- Para este ejercicio descarga la siguiente [PLANTILLA](#) (botón derecho → guardar como).
- Se dan los parámetros de cabecera y en el instrumento (**ruidos**) la inicialización de los valores de tipo i- y el valor de la muestra, de tipo k-, kValor.

Instrumento:

- Recibirá en **p4** la amplitud A (en valor PCM) y en **p5** el tipo de ruido a generar.
- La creación e inicialización del valor de la muestra se hace mediante **init** para que sólo se haga una vez, al inicio de la activación del instrumento.
- Para generar los ruidos, debes implementar en Csound el siguiente pseudocódigo (ver aclaraciones abajo), que funcionará en función del tipo de ruido que se le pida al instrumento:

```

si se pide ruido blanco entonces
    señal de ruido generada como número aleatorio entre -A y +A
sino si se pide ruido rosa entonces
    señal de ruido generada como número aleatorio entre -A y +A
    filtrar señal de ruido con un filtro pasa baja con fc = 1 kHz
sino ; (es ruido rojo)
    generar un número aleatorio entre -1 y +1
    si este número > 0 entonces
        cambio = +A/100
    sino
        cambio = -A/100
    finsi
    si el valor absoluto de (valor+cambio) > A entonces
        cambio = -cambio
    finsi
    Valor = Valor + Cambio
    señal de ruido = Valor
finsi
Sacar fuera la señal de ruido

```

- La generación de los números aleatorios se realiza con el operador:
 - Valor de salida **random** valor_mínimo , valor_máximo
- El filtro pasa-baja, se implementará con el operador:
 - Señal filtrada **tone** Señal a filtrar , frecuencia de corte
- El valor absoluto se calcula con la función **abs()**.

Nota: La plantilla incluye un visor para ver las señales que estás generando (botón Widgets del menú superior).

Pregunta 4

Finalizado

Puntúa 1,0
sobre 1,0

Partitura:

- Programa la partitura siguiendo las siguientes especificaciones:
 - El tempo será 60 BPM desde el principio
 - Debes activar 3 veces consecutivas el instrumento, de 2 tiempos cada activación, con amplitudes 11250, 21250 y 30000 y tipos 1, 2 y 3, respectivamente.
- Escribe a continuación la amplitud máxima PCM conseguida para la segunda activación

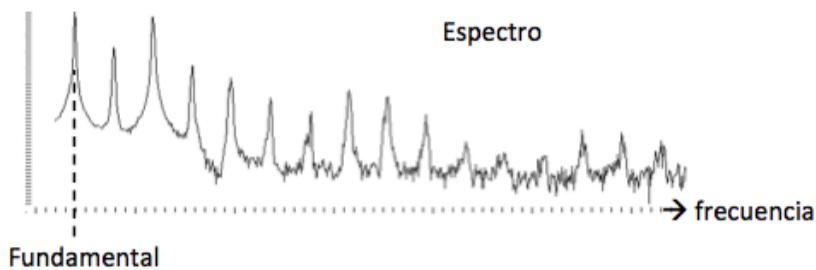
12375

(redondeo a entero)

Información

Frecuencia fundamental

La base del cálculo de los descriptores tímbricos armónicos es la detección de la frecuencia fundamental:



Hay muchas formas de calcularla. Csound dispone de varios operadores para hacerlo. El cálculo es inmediato, pero la dificultad está en las precauciones que hay que tomar para que sea un valor fiable.

Ejercicio 5.15:

Descarga la [PLANTILLA](#) de este nuevo proyecto (botón derecho → guardar como).

Orquesta

- El nombre al instrumento es **f0**
- Se ha definido la variable SFich que contendrá el nombre del fichero a analizar y que se enviará desde la partitura, como cuarto parámetro
- También se ha puesto, al final del instrumento, una orden para imprimir el valor de la fundamental en la consola.
- Añade la variable *iUmbral* que contendrá el valor -30 dBFS, por debajo del cual no se va a generar nada.
- La lectura del sonido también nos sirve.
- Borra el resto hasta el **out** (eso último lo dejas).
- El cálculo de la fundamental lo va a hacer el operador **pitchamdf**:
 - El primer parámetro es el sonido a analizar.
 - Segundo y tercero serán los límites de la banda en Hz en la que buscar la fundamental. La buscaremos entre 100 y 500 Hz.
 - Tiene 2 salidas: la evolución de la frecuencia en Hz y de la amplitud PCM.
- Como en otros casos, sólo nos interesará la F_0 si la amplitud está por encima del umbral:
 - Si la amplitud en dbFS < umbral entonces hacemos que la $F_0 = 0$ (como decir: no la detectamos).
- La detección puede ser un poco inestable. Para suavizarla ponemos un filtro pasa-baja especial para la evolución lenta de variables de tipo k-. Aplica **tonek** a la frecuencia detectada con una frecuencia de corte de 5 Hz.
 - sintaxis: kvariable **tonek** kvariable , frecuencia de corte
- La escritura de los valores de la F_0 se realizará con la orden (lo hará una vez por segundo):

```
printf 1, int(kFrec)
```

Partitura

Sólo cambia el nombre del instrumento en la orden **i** y la duración según lo que pida la pregunta siguiente del cuestionario.

Explicación de la medición

- Veremos en la consola la F_0 calculada cada segundo.
- Debes quedarte con el valor más repetido o, en caso de que fueran todos distintos, quédate con el último (por asegurarte de que se mide en la fase estable, cuanto más lejos del ataque mejor).

Pregunta 5

Finalizado

Puntúa 2,0
sobre 2,0

Descarga el fichero wav

0:00 / 0:00

y aplícale el programa del ejercicio: (usa duración = 6.0 s)

- Valor calculado para la F_0 = Hz

[◀ Entrega de la primera sesión P5 \(P\)](#)
[Ir a...](#)
[Entrega de la parte no presencial P5 \(NP\)](#)