Csound: PALABRAS o VARIABLES RESERVADAS

Ayuda y manuales

Un **opcode** en Csound es el equivalente a una función en C. Toma una serie de parámetros de entrada y devuelve una salida.
Dependiendo del tipo de variable de salida, la función se llamará una vez cada nota, o tantas veces como indique la frecuencia de control o de muestreo.
Veremos esto con más detalle en los distintos **opcodes** que explicaremos.

Como siempre tener el manual cerca es de ayuda:
`http://csound.sourceforge.net/#Documentation`

Pero también hay muchos recursos útiles en línea:
`http://www.adp-gmbh.ch/csound/opcodes/index.html`

Estructura del programa

<Csoundsynthesizer>
<Csoptions> </Csoptions>
<Csinstruments> </Csinstruments>
<CsScore> </CsScore>
</Csoundsynthesizer>

Variables iniciales de CsInstruments

```
;sample rate
sr = 44100
;control rate
ksmps = 1
;number of channels
nchnls = 1
```

sr significa *sample rate*, frecuencia de muestreo y nos indica cuántas muestras

por segundo se generarán. Normalmente se usa el valor de 44100 que corresponde con la frecuencia de muestreo del CD y que proporciona una buena respuesta en frecuencia*
ksmps significa número de muestras por bloque de control, y da idea de la velocidad de actualización del cálculo de las variables con prefijo k-. [Normalmente y con los ordenadores actuales el valor es 1, que significa que las variables k- se actualizan con cada muestra.]
nchnls significa *number of channels,* número de canales del sonido resultante: 1 = mono, 2 = estéreo, etc.

```
instr 1

endin
; endin indica fin de la definición del instrumento

;la “e” indica end, fin de partitura
e
```

Definición de Instrumentos

Generadores de Sonido

Por ahora utilizaremos como generador de sonido un opcode que produce una señal sinusoidal.

Generación de señal sinusoidal, OPCODE oscils

Una de las fuentes necesarias para la síntesis sonora es el oscilador senoidal, que proporciona la forma de onda necesaria para la síntesis aditiva o el vibrato entre otras posibilidades. Este opcode en particular fue desarrollado por Itsvan Varga en 2002, es muy eficiente. y no requiere la definición previa de una tabla de datos.

Sintaxis:

```
asound oscils iamplitude, ifrequency, iphase, [, iflg]
```

Entrada:

iamplitude: amplitud maxima de la onda

icps: frecuencia en Hz. (No debe superar **sr**/2)oscils

iphase: fase inicial de 0 a 1.

Salida:

asound contendrá los valores de una onda senoidal con amplitud maxima `iamplitude`, frecuencia `ifrequency` y fase arcsin(`iphase`)

```
<CsoundSynthesizer>

<CsOptions>
</CsOptions>

<CsInstruments>
sr = 44100
ksmps = 10
nchnls = 1

; Instrument #1 - a fast sine oscillator.
instr 1
  iamp = 10000
  icps = 440
  iphs = 0
  a1 oscils iamp, icps, iphs
  out a1
endin
</CsInstruments>

<CsScore>
i 1 0 2 ; Play Instrument #1 for 2 seconds.
e
</CsScore>
</CsoundSynthesizer>
```

Generación de ruido, OPCODE rand

El opcode **rand** nos permitirá contar con una fuente de números aleatorios distribuidos uniformemente en el rango -iamplitud..+iamplitud
ej.

anoise **rand** iamplitud

El filtrado digital puede realizarse de múltiples maneras y el lenguaje Csound ofrece varias de ellas. En concreto usaremos los filtros Butterworth implementados.
Su uso es muy sencillo, sólo tendremos que especificar las frecuencias donde trabajan y el ancho de banda en el caso del filtro pasa-banda.

Salida de Audio

Varía dependiendo de si producimos una señal mono o una estéreo

Salida Mono, OPCODE out

```
out asound
```

Entrada: asound
Salida: asound se envía a la salida del instrumento (puede ser un archivo o directamente al hardware de audio)

El opcode **aout** envía la señal de audio asound a la salida del instrumento, que puede estar definida como un archivo o como la salida de audio hardware del computador.
Obsérvese que la salida no aparece explictamente en el código, y debe ser especificada en tiempo de compilación con un flag.

Salida estéreo, OPCODE outs

```
out asoundleft, asoundright
```

Entrada: asoundleft, asoundright, que contienen el audio del canal izquierdo y derecho, respectivamente.
Salida: asoundleft se envía a la salida 1 o left del instrumento y asoundright a la salida 2 o right. Si se está compilando hacia un archivo el resultado estará en el formato escogido (WAV, AIFF, etc) con los dos canales izquierdo y derecho entrelazados (interleaved), es decir, una muestra Left, una muestra Right, y así sucesivamente. El editor de audio sabe cómo separar los datos.
En el caso de salida en tiempo real, asoundleft irá a la salida 1 del hardware (normalmente marcada como LEFT) y asoundright a la salida 2.

Modificación de la señal de sonido

Retardos de la señal, OPCODES delay y delay1

Muchas veces es necesario mezclar una señal de audio con si misma tras un cierto retraso (delay). Esto se puede hacer con el opcode **delay** cuyo uso es:

adelayed **delay** asound, idelaytime

¿Qué significa la “i” delante de idelaytime? indica que el tiempo de retraso debe ser fijado al principio de la nota y no se puede cambiar durante la

ejecución.

El opcode **delay1** retrasa la señal de audio una sola muestra (o sea 1/**sr =** 1 / sample rate segundos):

adelayed **delay1** asound

Envolvente de la señal, OPCODE adsr

Una de las características que distinguen a un sonido de otro es la envolvente de amplitud. En pocas palabras, es la curva que "envuelve" a la onda durante el tiempo que dura el sonido.

El opcode **adsr** proporciona una manera sencilla de generar una función o curva envolvente que luego multiplicaremos por la onda a la que queremos aplicar la curva.

Sintaxis:

```
ar   adsr      iatt, idec, isust_level, irel [,idel]
```

Entrada:
iatt: tiempo de ataque (desde el valor cero al valor uno)
idec: tiempo de decaimiento (desde 1 al nivel de sostenimiento)
isust_level: nivel de sostenimiento de la nota)
irel: tiempo para pasar del nivel de sostenimiento a 0)

Salida:
ar: la curva envolvente que va de 0 a 1 en iatt segundos, baja de 1 a isustlevel en idec segundos y pasa de 1 a 0 en irel segundos.

Ejº:

```
sr     = 44100
ksmps  = 10
nchnls = 1

instr 1
  iduracion = p3 ; duracion de la nota
  iamplitud = p4 ; amplitud maxima de la nota
  ifrecuencia = p5 ; frecuencia de la nota

  aenv adsr iduracion * 0.1, iduracion * 0.6, 0.8, duracion * 0.3

  asound oscils iamplitud, ifrecuencia, 0

  out aenv * asound
endin
```

En el ejemplo, el instrumento 1 recibe una nota con duración = p3 y amplitud = p4 procedente de la partitura. Inicialmente pasa p3 y p4 a dos variables con nombres más reconocibles. Las variables empiezan con “i” porque sólo actualizarán su valor una vez por cada nota. Ahora el opcode **adsr** generará en la variable aenv, la curva envolvente usando para ello el valor idur, que contiene la duración de la nota. De esta manera la envolvente se ajusta con precisión a la duración de cada nota.
Finalmente se multiplica aenv por asound que es la variable que contiene el sonido al que se le quiere aplicar la envolvente.

Generación de rectas

Crear un segmento de recta, OPCODE line

El opcode **line** se usa cuando hay que pasar linealmente de un valor a otro durante un tiempo determinado.

Sintaxis:
krecta **line** ivalora, idur, ivalorb
arecta **line** ivalora, idur, ivalorb

Entrada:
ivalora: valor inicial
idur: duración del proceso
ivalorb: valor final

Salida:
krecta ó **arecta** toma el valor ivalora para t=0 y toma el valor ivalorb para t=idur pasando linealmente por los puntos entre ivalora e ivalorb.

Sencillamente, el valor que devuelve **line** pasa de ivalora a ivalorb en idur segundos.

Normalmente idur dependerá de alguna forma (pero no siempre) de la duración de la nota (p3).

Crear varios segmentos de recta consecutivos, OPCODE linseg

```
kr      linseg     ia, idur1, ib[, idur2, ic[...]]
ar      linseg     ia, idur1, ib[, idur2, ic[...]]
```

La extensión natural del opcode **line** es el opcode **linseg**, que permite realizar múltiples segmentos de líneas.

Un uso evidente es el de generar envolventes.

Filtrado de la señal, opcodes butterlp, butterbp y butterhp

Los opcodes **butterlp, butterbp, y butterhp** son implementaciones de filtros Butterworth pasa-baja, pasa-banda y pasa-alta.

Se utilizan cada vez que se requiera filtrar una señal y dejar sólo las frecuencias más bajas que una dada (filtro pasa-baja), dejar un rango de frecuencias en torno a una dada (filtro pasa-banda) o dejar sólo las frecuencias superiores a una dada (filtro pasa-alta).

• Filtro pasobanda, Opcode **butterbp**

Sintaxis:
`afiltered` **`butterbp`** `asig, kfreq , kband`

Entrada:
asig: señal de entrada para ser filtrada
kfreq: frecuencia central
kband: ancho de banda en torno a la frecuencia central (bandwidth).

Salida:
afiltered: Es la fuente asig, pero dejando pasar las frecuencias en un ancho de banda kbandwidth en torno a la frecuencia kfreq.

Los parámetros kfreq y kband son de tipo k y no de tipo i. Esto significa que podemos cambiarlos durante la duración de una nota usando por ejemplo el opcode **line.**

El siguiente ejemplo está tomado de:
http://www.adp-gmbh.ch/csound/opcodes/butterbp.html

a440 is a signal that oscilates with 440 Hz. **kfrq220_660** is the center frequency for *butterbp* and moves from 220 Hz up to 660 Hz. The filtered signal (**afiltered**) is sent to one channel while a440 is sent to the other channel. The filtered signal reaches the same amplitude as a440 after a second (when kfrq220_660 is 440 Hz) and is otherwise attenuated.

```
sr     = 44100
kr     = 4410
nchnls = 2

instr 1

  ilen = p3
  a440          oscili   10000, 440, 1
```

```
  kfrq220_660   line     220, ilen, 660
  afiltered     butterbp    a440, kfrq220_660, 110
  outs          a440, afiltered

endin
```

El resultado sonoro con una nota de 2 segundos (i1 0 2) es el de la figura:

Se ve como la frecuencia de 440 Hz del canal izquierdo no sufre atenuación mientras que el canal derecho deja pasar mejor la señal de 440 Hz cuando el filtro está centrado justamente en esa frecuencia, es decir en el segundo 1.0s de la nota.

•Filtro pasobajo, Opcode **butterlp**
Sintaxis:
`afiltered` **`butterlp`** `asig, kfreq`

Entrada:
asig: señal de entrada para ser filtrada
kfreq: frecuencia de corte

Salida:
afiltered: Es la fuente asig, pero dejando pasar las frecuencias inferiores a la frecuencia de corte kfreq.

•Filtro pasoalto, Opcode **butterhp**
Sintaxis:
`afiltered` **`butterhp`** `asig, kfreq`

Entrada:
asig: señal de entrada para ser filtrada
kfreq: frecuencia de corte

Salida:
afiltered: Es la fuente asig, pero dejando pasar las frecuencias superiores a la frecuencia de corte kfreq.

Ver ejemplo de uso en el opcode **diskin**

Tareas a realizar:

1. Generar una envolvente mediante ADSR o LINESEG y aplicarla al ejemplo del apartado 5 de la práctica anterior.
2. Generar un ruido blanco. Aplicarle un filtro pasobanda con parámetros fijos. Modificar el código para que tanto la frecuencia central como el ancho de banda sean variables
3. Cargar un fichero wav con el opcode DISKIN y reproducirlo a varias velocidades tanto hacia adelante como hacia atrás.
4. Aplicar el filtro pasobanda de parámetros variables al resultado del diskin.