

Comenzado el miércoles, 1 de abril de 2020, 11:08

Estado Finalizado

Finalizado en miércoles, 1 de abril de 2020, 13:47

Tiempo 2 horas 38 minutos

empleado

Puntos 9,0/9,0

Calificación **10,0** de 10,0 (**100%**)

Información

Procesadores del sonido

En esta práctica vamos a diseñar y programar en Csound procesadores de los diferentes parámetros psicoacústicos vistos en teoría. El material previo publicado es, de nuevo, muy importante para realizar estos ejercicios.

Información

Procesadores del tiempo

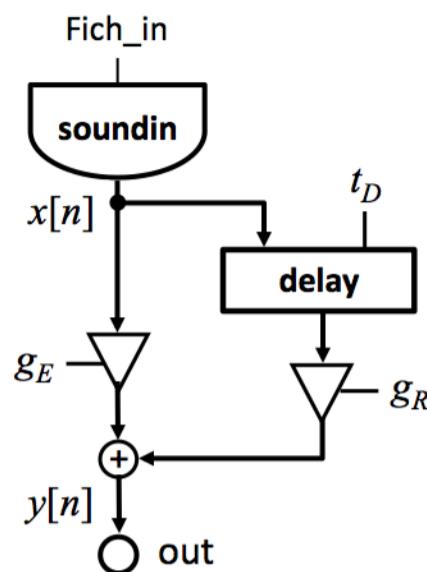
Información

Eco simple

Se implementa con un filtro peine FIR con un número de muestras de retardo D tal que $t_D = D / f_s > 50$ ms, con la siguiente ecuación:

$$y[n] = g_E x[n] + g_R x[n - D]$$

El esquema de implementación en Csound es el que aparece a continuación:



Ejercicio 7.6:

- Implementa el instrumento **eco1** usando esta [PLANTILLA](#) (botón derecho → guardar como)
- El instrumento recibirá desde la partitura:
 - El nombre del fichero en **p4**,
 - Las ganancias g_E y g_R , en **p5** y **p6**, y
 - El tiempo de retardo t_D en segundos en **p7**.

Para la partitura:

- Se usa tempo 60 BPM constante desde el principio. Así los tiempos serán segundos.
- Activa tres notas consecutivas (con el carácter + en el parámetro de inicio) que procesen el fichero

0:00 / 0:00

(la activación debe durar tanto como la señal, 9.0 s + el tiempo de retardo, t_D) con los siguientes parámetros (observa que en la última nota el volumen del eco es mayor que el de la señal):

Pregunta 1

Finalizado

Puntúa 2,0
sobre 2,0

| | g_E | g_R | t_D (s) |
|-----------------------------|-------|-------|-----------|
| Sin eco | 1.0 | 0.0 | 0.001 |
| Eco simple corto | 1.0 | 0.4 | 0.151 |
| Eco simple más largo | 0.5 | 1.5 | 0.364 |

Indica las duraciones acumuladas, que Csound imprime en la consola de salida, en azul, entre **tt** y **m**:

- Sin eco:
- Eco simple corto:
- Eco simple más largo:

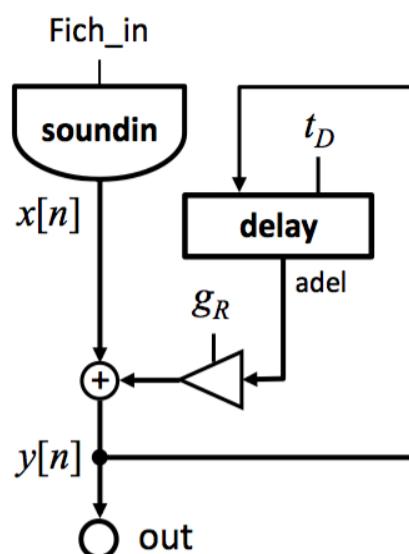
Información

Eco múltiple incontable

Se implementan mediante filtros peine IIR, a los que corresponde esta ecuación, con la ganancia de la señal de entrada = 1 para no alterar su volumen:

$$y[n] = x[n] + g_R y[n - D]$$

cuya implementación en Csound sería como en la siguiente figura:



Ejercicio 7.7

- Implementa el peine IIR de la figura en el instrumento **ecomul**.
- Descarga la siguiente [PLANTILLA](#) (botón derecho → guardar como)

Instrumento:

- Recibirá desde la partitura:
 - El nombre del fichero a procesar (**p4**),
 - La ganancia de la realimentación, g_R (**p5**), y
 - El tiempo de retardo, t_D , en milisegundos (**p6**).
- El funcionamiento del instrumento es el mismo que el peine IIR en la práctica de filtros.

La partitura:

- Debe activar el instrumento para procesar completamente el sonido de los 2 ficheros siguientes (descárgalos):

1.

0:00 / 0:00

(5.0 s de duración)

2.

0:00 / 0:00

(10.5 s)

Nota: para responder a la pregunta siguiente, ten en cuenta que Csound puede hacer las operaciones por ti y escribir el resultado en la consola mediante un **print**. (el logaritmo decimal es la función **log10()** en Csound)

Pregunta 2

Finalizado

Puntúa 2,0
sobre 2,0

Parámetros de las activaciones:

- El sonido 1. debe procesarse con ganancia de realimentación $g_R = 0.41$ y tiempo de retardo $t_D = 1460$ ms.
- El sonido 2. debe procesarse con $g_R = 0.65$ y 390 ms de retardo.
- La duración de cada activación del instrumento debe permitir oír el efecto completo, por lo que debe ser igual a la duración del fichero + el tiempo de extinción, t_{R60} , que corresponda en cada caso: $t_{R60} = t_d \times \frac{60}{-20 \log_{10}(g_R)}$
- Escribe a continuación el cálculo de las duraciones de ambas activaciones:

[5.0 +] (usa 2 decimales; aquí con coma, en la partitura con punto)

[10.5 +] (lo mismo)

Información

Procesadores del espacio

Información

Reverberador por convolución (auralización)

Csound proporciona el operador **pconvolve** para realizar el procesamiento por convolución visto en teoría:

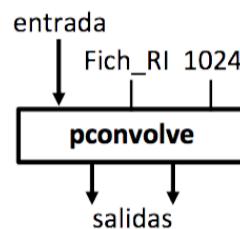
$$\text{señal } x[n] * \text{respuesta_al_impulso } h[n] = \text{señal procesada } y[n]$$

Ejercicio 7.8:

- Para implementar el instrumento **rever**, descarga esta [PLANTILLA](#) (botón derecho → guardar como)

El instrumento:

- Recibirá desde la partitura:
 - El nombre del fichero con el sonido a procesar en **p4**
 - El nombre del fichero que contiene la respuesta al impulso (RI) de un recinto acústico en **p5**
 - También se incluye la longitud de la ventana para trocear la RI (*iLVent* = 1024 muestras) que necesita **pconvolve** como tercer parámetro.
- El fichero que procesaremos está en mono, pero la RI que usaremos es estéreo, por lo que **pconvolve** tendrá 2 salidas correspondientes a los dos canales estéreo (figura).



- A las salidas reverberadas habrá que sumarles la entrada, pues es el sonido directo, que también debe oírse.

En la partitura:

- Tempo 60 BPM para que los tiempos estén en segundos.
- El sonido a procesar está en:

0:00 / 0:00

(duración = 8.5 s)

- La RI de una sala pequeña (por lo que será corta) está en [ht_sala.wav](#):

0:00 / 0:00

- Activa el instrumento durante la duración del fichero a procesar + la duración del fichero de la RI (su t_{r60}).
 - Para saber la duración exacta de la $h[n]$, abre su fichero con **Audacity** para verla (y si quieres oírla) y saber cuánto dura. Si pulsas el cursor irá al final y podrás ver su duración en las ventanas de tiempo, abajo.
- Cuando funcione bien, añade una segunda activación en la partitura, que empiece cuando termine la anterior (con inicio +)
- Debes pasar el mismo fichero a procesar, pero como RI, debes pasar ahora la de una gran sala de conciertos ([ht_teatro.wav](#)):

0:00 / 0:00

La duración a añadir sobre la duración del fichero de audio es ahora de +1.9 s.

- Compila y escucha el resultado. Conviene compararlo con el sonido sin procesar.

Información

Procesadores del timbre

Información

Flanger

Un *flanger* es como un eco incontable (que se hace mediante un peine IIR), pero con 2 diferencias importantes:

1. El tiempo de retardo es de tipo corto (< 10 ms) en vez de décimas de segundo, y
2. El tiempo de retardo es variable.

Ejercicio 7.9:

- Para implementar el nuevo proyecto, haz una copia del proyecto del eco incontable y llámale ej7.9.csd.

En las opciones de compilación:

- Cambia el fichero donde se graba la salida para que sea ej7.9.wav.

En el instrumento:

- Cambia su nombre a **FLANGER**.
- Añade los siguientes parámetros, que se enviarán desde la partitura:
 - **p7** : la amplitud (máxima desviación) A_D (en ms) de la variación del t_D .
 - **p8** : la frecuencia de la modulación (en Hz).
 - Cambia **p6**, porque se convertía de ms a s y ahora ya no es necesario, porque el tiempo de retardo para el retardo virtual **vdelay** se dará en ms (tal y como llega de la partitura).
- Debes crear una señal de modulación de baja frecuencia con un operador **lfo** con la amplitud y frecuencia de la modulación. Será de tipo sinusoidal (tercer parámetro de **lfo** = 0).
- Cambia el operador **delay** por un **vdelay**, porque ahora el tiempo de retardo será variable = t_D constante + señal de modulación generada por el LFO.
- El tercer parámetro de **vdelay** es el retraso máximo que se puede producir, que será $t_D + A_D$.

En la partitura:

- Usa la siguiente plantilla para sustituir la del proyecto anterior y haz la nueva nota sustituyendo las que había anteriormente:

```

;      p1      p2      p3          p4          p5      p6      p7      p8
;  instrum  ini   dur       fichero      gR    tD  max_desv fm
;                                ms      ms        Hz
;-----
```

Pregunta 3

Finalizado

Puntúa 3,0
sobre 3,0

- Activa el instrumento durante la duración de este fichero:

0:00 / 0:00

(duración = 10.5 s) con $g_R = 0.67$, $t_D = 2.2$ ms con una desviación máxima de 1.7 ms y una frecuencia de modulación de 0.32 Hz.

Responde:

- ¿Cuántas muestras de retardo usará el filtro? (antes de la modulación)

$$D = t_D \times f_s = 97 \quad (\text{entero})$$

- Dado el tiempo de retardo y la desviación máxima, ¿entre qué tiempos de retardo en ms oscilará el filtro? (1 decimal)

$$t_D \text{ mínimo} = 0,5 \quad \text{ms}, \quad t_D \text{ máximo} = 3,9 \quad \text{ms}$$

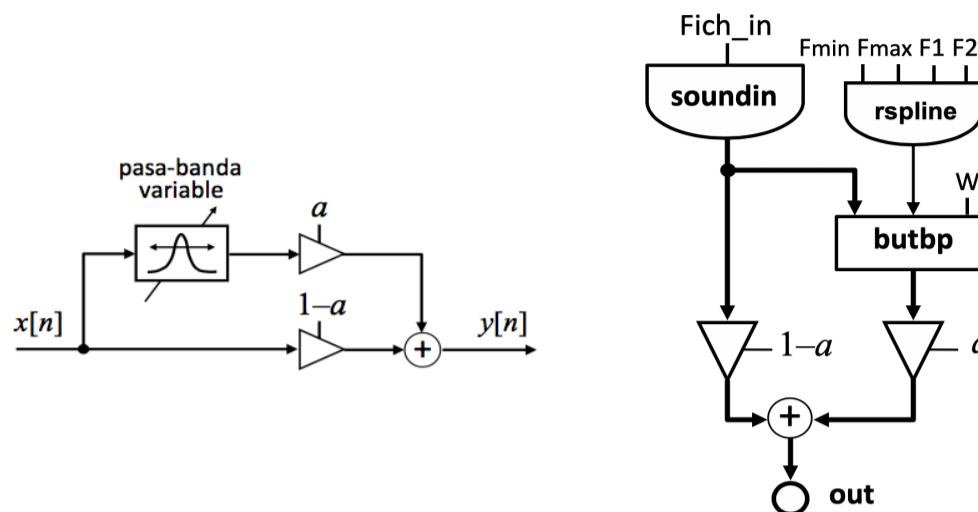
Información

Wah-wah

El wah-wah básicamente es un filtro pasa-banda de frecuencia central variable (ver figura de abajo a la izquierda). En realidad la frecuencia se suele modificar con un pedal. Aquí lo haremos con una curva generada aleatoriamente.

Ejercicio 7.10:

- Implementa el instrumento **WahWah** según la especificación de la figura de la derecha.



- Usa la siguiente: [PLANTILLA](#) (botón derecho → guardar como). En ella están ya indicados todos los parámetros que hay que recibir desde la partitura, que se explican abajo.
- El pasa-banda se implementa con el operador **butbp**, que requiere la frecuencia central f_c y el ancho de banda W (ambos en Hz).
- La frecuencia central del filtro será controlada por el operador **rspline**, que generará una curva aleatoria que se moverá entre los valores F_{\min} y F_{\max} con factores de variación entre F_1 y F_2 .
- Los parámetros de funcionamiento se reciben desde la partitura en este orden:
 - p4**: el nombre del fichero a procesar,
 - p5**: La frecuencia central mínima del filtro (la F_{\min}),
 - p6**: La frecuencia central máxima del filtro (la F_{\max}),
 - p7**: El ancho de banda del filtro, y
 - p8**: El coeficiente a (cantidad de efecto en la suma).

Nota: Hay preparado un **prints** para ayudarte a contestar a la siguiente pregunta.

Pregunta 4

Finalizado

Puntúa 2,0
sobre 2,0

Utiliza los siguientes valores para los parámetros de trabajo:

- Para el generador de la curva de frecuencia (**rspline**):
 - Frecuencia central mínima, $F_{\min} = 100$ Hz
 - Frecuencia de la modulación, $F_{\max} = 1500$ Hz
 - Factores de variación: $F_1 = 0.5$ y $F_2 = 5$
- Para el filtro pasa-banda variable (**butlp**):
 - La frecuencia central del filtro la proporciona el operador anterior, y
 - Ancho de banda constante = 100 Hz.
- El valor del coeficiente (volumen del efecto) será $a = 0.70$.
- Aplícalo al sonido completo (8.0 s) que hay en

0:00 / 0:00

Responde a las siguientes preguntas sobre el procesador. Recuerda que $Q = f_c / W$.

- ¿Cuál es la Q mínima del filtro (cuando su $f_c = F_{\min}$)?

1,0

(adimensional, con 1 cifra decimal)

- ¿Cuál es la Q máxima del filtro (cuando su $f_c = F_{\max}$)?

15,0

(adimensional, con 1 cifra decimal)

[◀ Entrega de la primera sesión P7 \(P\)](#)
[Ir a...](#)
[Entrega de la segunda sesión P7.2 \(P\) ▶](#)