

Análisis de Señales y Sistemas Digitales

Trabajo Práctico de Laboratorio N2

“Análisis y Síntesis de Audio digital”

Para este trabajo práctico deberán implementar distintos métodos de síntesis de instrumentos musicales y efectos de audio.

Existe material extenso, disponible en la web, en papers y libros. Se recomienda que investiguen. Vean las referencias al final del documento.

0 FFT (Precalentamiento)

Este inciso está separado de los demás puntos del T.P. El objetivo es que entiendan la FFT.

1. Implementar una FFT mediante el algoritmo de Cooley-Tukey, en el lenguaje C. Debe poder operar con una cantidad de puntos desde 1 hasta, al menos 4096, sujeto a las restricciones del algoritmo (potencias de 2). Son libres de implementar el código como prefieran. Externamente, deberá presentar la siguiente interfaz:

```
void fft(complex float *in, complex float *out, size_t n);
```

Entrada

in: Puntero al buffer con datos de entrada.

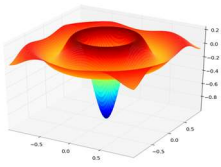
out: Puntero al buffer con datos de salida.

n: Cantidad de puntos. En caso de no ser una potencia de dos, el comportamiento es indeterminado.

El buffer de salida puede ser el mismo que el buffer de entrada. La función debe operar correctamente aún en este caso.

2. Validar el funcionamiento. Comparar con una implementación de referencia.

ovadd



1 Programa Principal

Deberán desarrollar un programa que sintetice una pista a partir de un archivo MIDI (.mid).

Los archivos MIDI contienen una cantidad de “tracks”. Cada “track” se puede asignar a un instrumento, o “programa”, en la jerga MIDI. Los “tracks” contienen los instantes en que se comienza una nota (por ejemplo, cuando se presiona la tecla en un piano), la intensidad de cada nota (“velocidad”, en la terminología MIDI) y el momento en que termina la nota (cuando se levanta el dedo de la tecla o cuerda).

Su programa debe ser capaz de:

- Cargar un archivo en formato MIDI
- Asignar cualquier instrumento a cualquier track.
- Mezclar los distintos *tracks* para generar una pista.
- Agregar y configurar diferentes efectos, para cada track y para la mezcla total.
- Guardar la pista generada en algún formato de audio (wav, mp3, etc).
- Mostrar el espectrograma del audio generado (ver el punto correspondiente del TP).

- Se debe poder reproducir cualquier nota del instrumento, o acordes, con cualquier duración

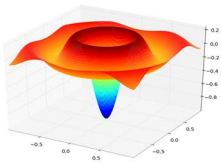
Una vez completado el sistema, ensayarlo con un fragmento considerable del segundo movimiento (Adagio) del Concierto de Aranjuez, de Joaquín Rodrigo. En la web se pueden encontrar archivos midi con la obra. Incluir el .wav generado como parte de la entrega.

Observaciones:

- La duración de una nota no es lo mismo que la duración de un sonido. Muchas veces, el sonido se extiende durante más tiempo que lo que dura la nota. Por el contrario, si se toca *stacatto*, el sonido durará menos. Si tienen esto en cuenta, la calidad de su síntesis mejorará considerablemente.
- Existen librerías para leer archivos MIDI. Se sugiere utilizarlas.
- Lo que se describió anteriormente es la funcionalidad esencial que deberá tener su programa. Pueden agregar características y funciones.

Se deben implementar en el programa las siguientes prestaciones:

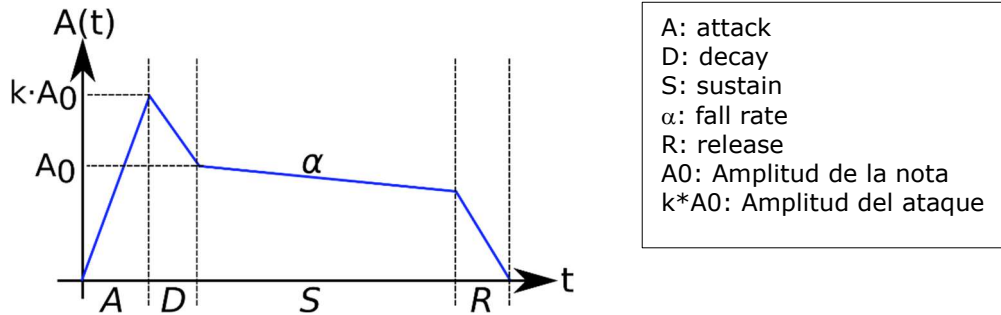
- Síntesis Aditiva o FM (una de las dos y sintetizar UN instrumento)
- Síntesis por modelado físico de los instrumentos
- Síntesis por Muestras
- Delay y Reverberación



2 Síntesis aditiva de sonidos

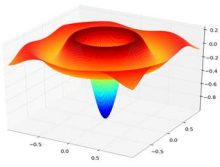
La síntesis aditiva consiste en combinar un número de señales senoidales de distintas frecuencias (llamadas “parciales” o *partials*), con distinta amplitud. De esta manera se intenta reproducir la estructura de armónicos de un instrumento musical. La ventaja de la síntesis aditiva es la baja complejidad de cómputo y moderados requerimientos en memoria.

Si la amplitud de cada parcial es constante, la calidad del sonido resultante no será buena. En la realidad, cada nota tiene una envolvente de amplitud. La manera más básica de especificar la envolvente se denomina ADSR.



Se pide lo siguiente:

1. Obtener las amplitudes de los parciales para al menos 4 instrumentos distintos. Sumar y modular la resultante con una envolvente ADSR. Encontrar los parámetros óptimos.
2. La parametrización ADSR es una posibilidad para especificar la envolvente. Proponga una mejor parametrización y/o forma de envolvente.
3. En general, no es realista suponer que todos los parciales comparten la misma envolvente. Sintetizar al menos 4 instrumentos, con envolventes independientes para cada parcial. La cantidad de parámetros a determinar puede llegar a ser muy grande. Se recomienda que desarrollen un programa aparte para extraer esta información a partir de muestras. No es necesario restringirse a envolventes ADSR. Utilicen la parametrización que consideren mejor.
4. Investigar e implementar, en caso que mejoren la calidad de síntesis, las siguientes mejoras:
 1. Ley de caída de los parciales según la altura (*pitch*) de la nota.
 2. Desviación aleatoria en frecuencia de cada parcial. Esto puede transformar el espectro en inarmónico. Estudiar empíricamente el rango apropiado de desviación para “mejorar” el realismo del sonido generado.
5. Sacar conclusiones de los resultados.



3 Síntesis de sonidos mediante frecuencia modulada

Una señal de F.M. está definida mediante la siguiente expresión:

$$x(t) = A(t) \cos \psi(t)$$

donde A es la amplitud, variable en el tiempo, y Ψ es la fase, también variable en el tiempo.

La derivada de la fase es la frecuencia instantánea:

$$f_i = \frac{1}{2\pi} \frac{d}{dt} \psi(t)$$

La fase se varía mediante la siguiente ley:

$$\psi(t) = 2\pi f_c t + I(t) \cos(2\pi f_m t + \phi_m) + \phi_c$$

Reemplazando en $x(t)$:

$$x(t) = A(t) \cos (2\pi f_c t + I(t) \cos(2\pi f_m t + \phi_m) + \phi_c)$$

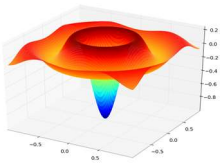
La síntesis F.M. es especialmente adecuada para simular instrumentos de viento. Se utiliza

$$\phi_m = \phi_c = -\pi/2$$

Lograr un buen sintetizador consiste en elegir adecuadamente $A(t)$ e $I(t)$.

Se pide lo siguiente:

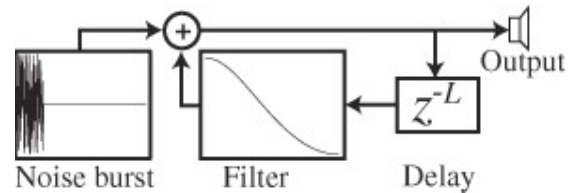
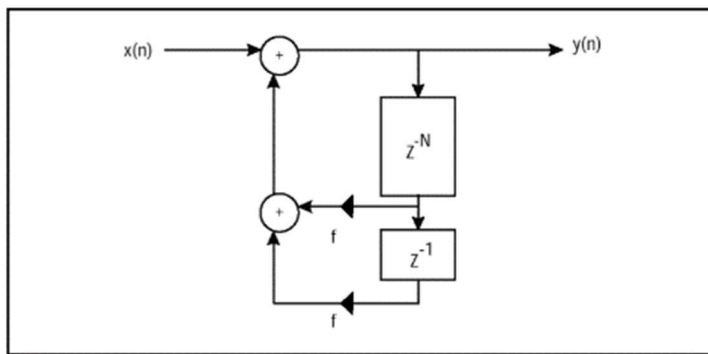
1. Sintetizar un clarinete. Para ello deberán crear las funciones $A(t)$ e $I(t)$. En este instrumento la relación entre f_c y f_m es n/m y ϕ_0 resulta del máximo común divisor entre ellas. Investigar la literatura para determinar A , I , n , m . Procurar obtener el sonido más realista.
2. Buscar otros instrumentos de viento. Ajustar n y m para emular esos instrumentos. Se puede variar $A(t)$ e $I(t)$.
3. Sacar conclusiones.



4 Síntesis de sonidos mediante modelos físicos¹

El modelo de Karplus-Strong es utilizado para sintetizar instrumentos de cuerda percutida. Se trata de un sistema lineal excitado con una secuencia aleatoria de longitud finita. El sistema consiste en una línea de retardo de L muestras, realimentada mediante un filtro. La siguiente figura ilustra el modelo conceptual de Karplus-Strong.

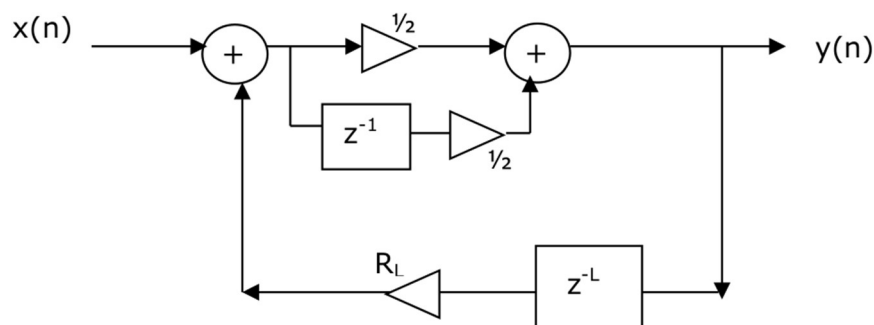
Figure 1. Karplus Strong String-Synthesizer Model



La línea de retardo simula la longitud de la cuerda. A medida que los distintos armónicos recorren la línea, se atenúan y su amplitud va decayendo. El filtro suele ser un pasa-bajos. De esta manera

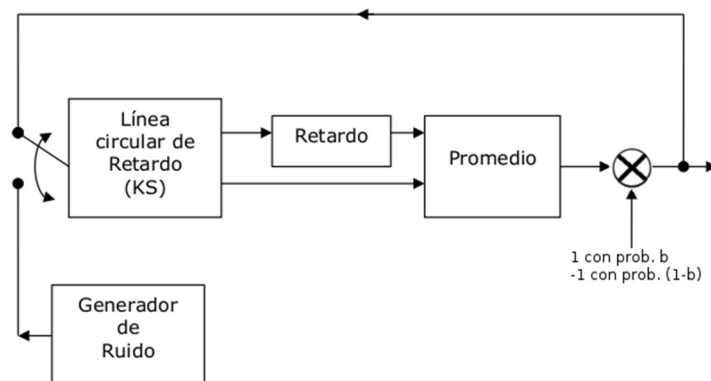
se atenúan más rápidamente las frecuencias más altas. La longitud de la secuencia de entrada debe ser de L muestras.

La siguiente figura muestra una posible implementación del modelo de Karplus-Strong:

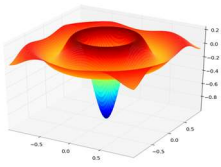


En este caso, la salida $y(n)$ se toma luego del pasa-bajo. La ganancia R_L permite ajustar el tiempo de caída de la nota. La frecuencia de la nota es $f_s/(L+1/2)$.

Mediante una modificación al modelo, es posible sintetizar instrumentos de percusión. El cambio consiste en realimentar la línea de retardo con ganancia positiva o negativa, de manera aleatoria.



¹ Ver *Perceptual Audio Signal Processing* (P.A.S.P.), de Julius O. Smith III.



Si $b=1$, el sistema se comporta como un filtro pasabajo y el sonido es como un instrumento de cuerda. Si $b=0.5$, el sonido es diferente a una cuerda. Se pierde su “pitch” y se asemeja más a un sonido percutado. Si $b=0$, la señal es negada cada $p+0,5$ muestras. El sonido es de un arpa en los registros bajos.

Se pide, tanto para el modelo original como para el modificado:

1. Encontrar la $H(z)$ y la respuesta en frecuencia del modelo analíticamente. Obsérvese que el modelo modificado no es tiempo invariante y por lo tanto, estrictamente hablando no tiene $H(z)$.
2. Implementar el modelo. Observar salidas relevantes. Investigar la influencia de la distribución de ruido (normal vs uniforme).
3. Encontrar y graficar la posición de los polos y los ceros.

Para modelo original:

4. ¿Cual es el rango de valores que puede tomar R para que el sistema sea estable?
5. Investigar e implementar un filtro que emule la caja de resonancia de la guitarra.
6. El modelo planteado anteriormente tiene un problema. Dado que la frecuencia de la nota está dada por la longitud de la línea de retardo más $\frac{1}{2}$, y esta longitud es un número entero, no se puede obtener cualquier frecuencia. El error resultante es más grande cuanto más aguda la frecuencia. Investiguen e implementen una solución para mitigar este problema.

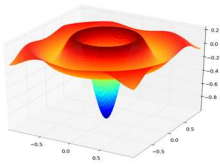
Para el modelo modificado

7. Hallar la fase $\Phi(\omega)$ del filtro en función de b .
8. Hallar una expresión de b en función de β que permita obtener el valor exacto de la frecuencia fundamental deseada:

$$\beta = \frac{f_s}{f_K} - (L + \frac{1}{2})$$

9. Hallar la desventaja introducida por este filtro e indicar las limitaciones que genera. (Evaluar en que octavas conviene usarlo).

Nota: Con las modificaciones y ajustes adecuados, la síntesis de instrumentos de cuerda puede hacerse de muy buena calidad. Investiguen e implementen lo que consideren adecuado.



5 Síntesis basada en muestras

La síntesis basada en muestras busca reproducir un sonido más realista, a partir de muestras grabadas de instrumentos. Como no es factible tener grabaciones de todas las notas con todas las duraciones, se deben generar las notas a partir de una cantidad reducida de muestras, que se pueden estirar y comprimir en tiempo y frecuencia.

La siguiente figura ilustra el “estiramiento” temporal de una grabación.

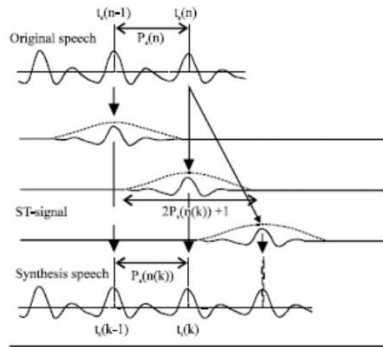
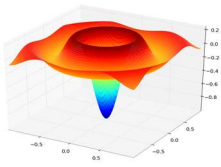


Figure 5: Esquema de funcionamiento de PSOLA.

Se pide:

Obtener varias muestras de uno o varios instrumentos e implemente un sintetizador basado en muestras. Investigue la literatura sobre el tema.



6 Efectos de Audio

Reverberación

Se deberá implementar al como mínimo 2 efectos para aplicar a la salida de la síntesis:

- Eco simple.
- Reverberador plano.
- Reverberador pasa-bajos.
- Reverberador completo. Ver referencias al final del documento.
- Reverberador por convolución. Consiste en convolucionar la señal con la respuesta al impulso de un ambiente reverberante. Buscar en Internet o grabar distintas respuestas al impulso.

Para cada caso:

1. Permitir al usuario ajustar los coeficientes o, mejor, proveer ajustes amigables que traduzcan en coeficientes para los filtros.
2. Proveer *presets*.

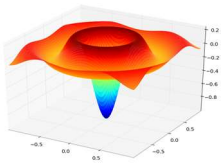
Flanger

Un flanger es un sistema lineal variable en el tiempo. Suma una señal a la misma señal retrasada. El retardo varía lentamente con el tiempo, de manera cíclica. Existen flangers feedforward y feedback. Ver bibliografía citada.

1. Encontrar la respuesta en frecuencia (en función del tiempo) del flanger.
2. Implementar un flanger:
 - Vibrato
 - Chorus
 - Flanger
 - Phaser
 - Wah-Wah

Otros efectos

Se valorará la inclusión de otros efectos de audio (Distorsión/otros).



7 Espectrograma

Un espectrograma es una imagen tiempo frecuencia de una señal. El espectrograma que implementen debe permitir cambiar los parámetros, como el tamaño de bloque o la ventana.

1. ¿Cual es el overlap requerido en un espectrograma? Demostrar.
2. Obtenga el espectrograma de una escala de sol mayor (G3), generada con la síntesis aditiva. Emplee aproximadamente 120ms por nota. ¿Cuales son los parámetros apropiados?

8 Referencias

De todo un poco:

- <https://ccrma.stanford.edu/~jos/GlobalJOSIndex.html>
- <https://www.aalto.fi/en/departament-of-signal-processing-and-acoustics/virtual-analog-synthesis-and-audio-effects>

Ventanas y espectrogramas, entre otros:

- <https://ccrma.stanford.edu/~jos/sasp/>

Karplus-Strong:

- https://ccrma.stanford.edu/~jos/pasp/Karplus_Strong_Algorithm.html

Reverberación:

- <http://cnx.org/content/m15491/latest/>
- https://ccrma.stanford.edu/~jos/pasp/Artificial_Reverberation.html#13681

Flanger:

- <https://ccrma.stanford.edu/~jos/pasp/Flanging.html>

Lectura recomendada:

DAFx Digital Audio Effects de Udo Zölzer

Implementación de efectos en tiempo real

Para hacerlo recomendamos usar C11 o C++11 o Python y librerías cross-platform como PortAudio, pyaudio

La implementación y su interfaz corren por cuenta de los alumnos

Se deberá entregar el testbench utilizado para validar su funcionamiento.