

Instituto Tecnológico de Buenos Aires

22.05 ANÁLISIS DE SEÑALES Y SISTEMAS DIGITALES

Trabajo práctico 2: Análisis y Síntesis de Audio Digital

Grupo 1

MÁSPERO, Martina	57120
MESTANZA, Joaquín Matías	58288
NOWIK, Ariel Santiago	58309
PARRA, Rocío	57669
REGUEIRA, Marcelo Daniel	58300

Profesores

JACOBY, Daniel Andrés
IRIBARREN, Rodrigo Iñaki
BELAUSTEGUI GOITIA, Carlos

Presentado: 03/05/2019

Índice

1. FFT	3
1.1. Enfoque	3
1.2. El problema	3
1.3. Primera observación de interés	3
1.4. Algoritmo conceptual	3
1.5. Optimizaciones de memoria	4
1.5.1. Optimización 1	4
1.5.2. Optimización 2	4
1.6. Optimizaciones de tiempo	4
1.6.1. Cálculo $W[n][k] = e^{-i2\pi k/n}$	4
1.6.2. Optimización Butterfly	4
1.7. Conclusión	4
2. Programa principal	5
3. Síntesis aditiva de sonidos	6
3.1. Tipos de envolventes	6
3.1.1. ADSR	6
3.1.2. Envolvente en base a muestras	6
3.2. Procedimiento	6
3.3. Conclusiones	8
4. Síntesis mediante modulación en frecuencia	9
4.1. Introducción teórica	9
4.2. Frecuencias laterales reflejadas	10
4.3. Espectro armónico e inarmónico	10
4.4. Espectro dinámico	13
4.5. Síntesis para instrumentos de viento	13
4.5.1. Síntesis de clarinete	14
4.5.2. Síntesis de campana	15
4.5.3. Síntesis de trombón	16
4.5.4. Síntesis de la trompeta	17
4.6. Conclusiones	17
5. Síntesis de sonidos mediante modelos físicos	18
5.1. Bloque A Elemental	18
5.2. Karplus Strong 1	18
5.2.1. Análisis teórico	18
5.2.2. Estudio de la distribución polos y ceros, respuesta en frecuencia	19
5.2.3. Cálculo de F_R	19
5.2.4. Análisis mediante señales	19
5.3. Karplus Strong 2	21
5.3.1. Análisis teórico elemental	21
5.3.2. Respuesta en frecuencia	22
5.4. Sintetización de instrumentos	22

5.4.1.	¿Como tener libertad con la frecuencia fundamental?	22
5.4.2.	¿Comó filtrar la componente continua?	23
5.4.3.	¿Como atenuar el sonido de manera suave?	23
5.4.4.	Sintetización de sonido de guitarra	23
5.4.5.	Distorsión del sonido de guitarra	24
5.4.6.	Sintetización de instrumento de percusión	24
5.4.7.	Explicación caja de resonancia	25
6.	Síntesis basada en muestras	26
6.1.	Escalamiento Temporal (TSM)	26
6.1.1.	Overlap-Add (OLA)	26
7.	Espectrograma	30
7.1.	Función para realización de espectrograma	30
7.2.	Aplicación a escala de Sol Mayor (G3)	31
8.	Efectos de audio	32
8.1.	Reverbebrador	32
8.1.1.	Implementación de eco simple	32
8.1.2.	Implementación de reverberación plana	32
8.1.3.	Implementación de reverberación pasa bajos	33
8.1.4.	Implementación de reverberación completa	34
8.1.5.	Implementación de reverberación por convolución	35
8.2.	Efectos basados en delays variables	36
8.2.1.	Vibrato	37
8.2.2.	Flanger	38
8.3.	Robotización	38
9.	Rotary Loudspeaker Effect	39

1. FFT

1.1. Enfoque

Se decidió desarrollar un algoritmo FFT, ya que se contó con conocimiento del campo de ciencias de la computación para resolver problemas de programación dinámica. Se mostrará la derivación de un algoritmo menos eficiente pero más entendible, y luego se describirán sus posteriores optimizaciones. En el resultado final se llegó a un algoritmo con $\frac{n}{2} \log_2 n$ multiplicaciones, $6n$ memoria, en otras palabras, $\Theta(n \log(n))$ tiempo, $\Theta(n)$ memoria.

1.2. El problema

El cálculo de la DFT consiste en producir una la salida $X[k]$ de coeficientes mediante la entrada $x[n]$, ambos arreglos de igual longitud N . La fórmula que describe la DFT que utilizamos es:

$$DFT[X[n]] = X[k] = \sum_{n=0}^{N-1} x(n) e^{-i2\pi kn/N} \quad (1)$$

Se necesita resolver la DFT en tiempo subcuadrático, es decir en menos de $\Theta(n^2)$, que es la solución trivial.

1.3. Primera observación de interés

Si la longitud de la sucesión N es par se puede escribir $X[k]$ como:

$$DFT[X[n]] = X[k] = \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x(2n) e^{-i2\pi kn/(N/2)}}_{DFT[X[2n]=Y[n]} + e^{i2\pi k/N} \underbrace{\sum_{n=0}^{\frac{N}{2}-1} x(2n+1) e^{-i2\pi kn/(N/2)}}_{DFT[X[2n+1]=Z[n]} \quad (2)$$

$$DFT[\underbrace{X[n]}_{0 \leq n < N}] = DFT[\underbrace{Y[n]}_{0 \leq n < N/2}] + e^{i2\pi k/N} DFT[\underbrace{Z[n]}_{0 \leq n < N/2}] \quad (3)$$

Donde $X[2n] = Y[n]$, $X[2n+1]$, $Y[2n+1]$ son sub-sucesiones de $X[n]$ tomando los índices pares e impares respectivamente. Podemos decir que descompusimos el problema de la DFT como dos problemas de DFT con sucesiones de la mitad de tamaño, es decir, conseguimos una solución recursiva. Es importante observar que la única forma de que la solución sea viable es que N sea una potencia de dos, para poder garantizar que al dividir los subproblemas siempre N_i sea par.

1.4. Algoritmo conceptual

Si bien la formulación anterior da lugar a la implementación de un algoritmo recursivo “top-down” se procedió a desarrollar una fórmula “bottom-up” para resolver el problema iterativamente, y por lo tanto provocar que la ejecución tenga una constante menor.

Se definió el subproblema $DFT[i][j][k]$ como el k -ésimo coeficiente, del subarreglo de longitud 2^i que comienza en el casillero j . Se puede mostrar mediante la ecuación 3 en esta formulación es equivalente a

$$DFT[i][j][k] = \begin{cases} DFT[i-1][j][k \% 2^{i-1}] + W[2^i][k] DFT[i-1][j + N/2^i][k \% 2^{i-1}] & 1 \leq i \leq \log_2 N \\ & 0 \leq j < N/2^i \\ & 0 \leq k < 2^i \\ A[j] & i = 0 \\ & k = 0 \end{cases} \quad (4)$$

Donde $W[N][k] = e^{-i2\pi k/N}$. Con esta precisa formulación se puede implementar con 3 sentencias *for* un algoritmo de $O(n \log n)$ tiempo y $O(n^2 \log n)$ memoria.

1.5. Optimizaciones de memoria

Se presentan dos modificaciones superficiales y técnicas, no conceptuales, que no alteran la lógica esencial del algoritmo, pero que lo vuelven práctico.

1.5.1. Optimización 1

Se necesita reducir el orden de memoria de $\Theta(n^2)$, el cual es muy limitante. La primera optimización es mediante la observación de que para calcular $DFT[i]$ solo necesitamos tener previamente calculado $DFT[i-1]$. Por lo tanto podemos calcular en un arreglo el DFT siguiente y en la posterior iteración escribir de dicho arreglo al anterior, “yendo y volviendo”. De esta forma se reduce el uso de memoria a $\Theta(n^2)$. Este valor todavía no es conveniente, ya que la expectativa es que tanto el tiempo como la memoria del algoritmo sean subcuadráticos.

1.5.2. Optimización 2

La observación necesaria para reducir el uso de memoria es que, en cada iteración $jk < N$, por lo que en realidad es posible almacenar todos los valores en un arreglo de longitud N (es decir, “comprimiendo” la matriz). Como el número de filas y columnas de la matriz es variable para cada valor de i es necesario implementar una lógica que se adapte a este hecho ($mat[i][j] = arr[i * colsize + j]$). De esta forma se redujo la memoria del algoritmo a $\Theta(n)$.

1.6. Optimizaciones de tiempo

Se mostrarán dos optimizaciones de tiempo que si bien no mejoran la complejidad del algoritmo en tiempo (de hecho, no son conocidos algoritmos mejores que $\Theta(n \log n)$), reducen, en una constante, su demora.

1.6.1. Cálculo $W[n][k] = e^{-i2\pi k/n}$

Calcular cada vez que se necesitan dichos factores si bien no modificaría la complejidad del algoritmo, si impacta en el tiempo total. Para ello se precalcula en el inicio del algoritmo $W[k] = e^{-2i\pi k/N}$ con el N más grande. Como vale que:

$$W[n][k] = e^{-i2\pi K/n} = e^{-i2\pi K/N * N/n} = W[K * N/n] = W[K * N/2^i] = W[K * N/2^i \% N] \quad (5)$$

Por lo tanto calculando inicialmente $W[k]$ con $0 \leq k < N$ ya se tiene información para obtener en una sola operación la exponencial correspondiente cuando sea necesario.

1.6.2. Optimización Butterfly

Otra optimización importante que permitirá reducir a la mitad la cantidad de multiplicaciones complejas que realiza el algoritmo se denomina optimización “Butterfly”. Consiste en observar dos propiedades:

$$W[n][k] = -W[n][k + n/2] \quad (6)$$

$$DFT[.][.][k \% 2^{i-1}] = DFT[.][.][(k + n/2) \% 2^{i-1}] \quad (7)$$

La forma de utilizarla consiste en, computar en paralelo las siguientes expresiones:

$$DFT[i][j][k] = DFT[i-1][j][k \% 2^{i-1}] + W[2^i][k] DFT[i-1][j + N/2^i][k \% 2^{i-1}] \quad (8)$$

$$DFT[i][j][k + n/2] = DFT[i-1][j][k \% 2^{i-1}] - W[2^i][k] DFT[i-1][j + N/2^i][k \% 2^{i-1}] \quad (9)$$

Realizando el producto $W[2^i][k] DFT[i-1][j + N/2^i][k \% 2^{i-1}]$ una sola vez.

1.7. Conclusión

Se llegó a un algoritmo muy optimizado ($6n$ memoria, $\frac{n}{2} \log_2 n$ tiempo) con una complejidad equivalente a las de los más utilizados. El código combinado con todas las optimizaciones es difícil de leer porque consiste en la combinación de todos los detalles técnicos descriptos juntos. Si bien es posible optimizar aun más la memoria utilizando un método in-place que no utilice arreglos adicionales para almacenar datos parciales, como de todos modos es necesario utilizar un arreglo de N elementos para almacenar $W[k]$, dicha técnica no reduciría asintóticamente el uso de memoria adicional. El algoritmo final demoró, en una computadora moderna del 2015 con Windows 10 unos 127ms en calcular 100 FFT's de 4096 puntos. Fue verificado su correcto funcionamiento en comparación con el resultado de librerías de Python que calculaban FFT.

2. Programa principal

Se desarrolló un software para procesar archivos midis mediante las funciones de síntesis de instrumentos. Se utilizaron distintas librerías de python, entre ellas mido (para leer los archivos midi) y tkinter (para poder realizar una interfaz gráfica)

Los resultados fueron exitosos; se logró desarrollar una interfaz gráfica funcional que brindó la capacidad para seleccionar los instrumentos, el volumen de los canales, mostrar el espectrograma de la canción resultante.



Figura 1: Ejemplo funcionamiento GUI

Se adjunta el código fuente en la entrega de este práctico.

3. Síntesis aditiva de sonidos

La síntesis aditiva busca la construcción de sonidos a partir de una suma de senoides, donde cada una de ellas representa un armónico o parcial, que a su vez representan los modos de vibración en el instrumento que produce el sonido. Cada parcial tiene distinta frecuencia y distinta amplitud, que decae en el tiempo debido a la modulación de una envolvente.

De acuerdo con la teoría de Fourier, la señal sintetizada $y(t)$ se puede escribir como:

$$y(t) = \sum_{k=1}^K env_k(t) A_k \cos(2\pi k f_0 t + \phi_k) \quad (10)$$

Donde f_0 es la frecuencia fundamental de la señal y la frecuencia de la nota musical y A_k , $k f_0$ y ϕ_k son la amplitud, la frecuencia y la fase del k -ésimo armónico respectivamente. La función $env_k(t)$ representa la envolvente que es la causante de que la señal no sea infinita en el tiempo si no que se termine en algún momento determinado.

3.1. Tipos de envolventes

3.1.1. ADSR

Es la parametrización mas simple, es la que está propuesta en la consigna del trabajo práctico y es igual para cada parcial. Consiste en cuatro parámetros característicos: *attack*, *decay*, *sustain* y *release*. El *attack* es el tiempo que se tarda en ir desde 0 hasta la amplitud máxima. El *decay* es el tiempo que se tarda en ir desde la máxima amplitud hasta su nivel estable. El *sustain* es el nivel de amplitud que se tiene mientras que la tecla siga presionada, la duración de este período se define con la duración de la nota. El *release* es el tiempo que tarda la amplitud en llegar a cero desde el nivel de *sustain*.

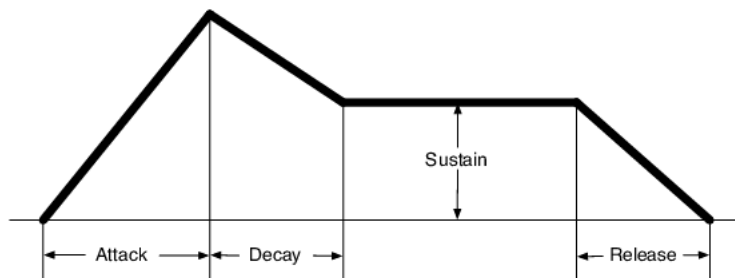


Figura 2: Envolvente ADSR

Al sintetizar una nota utilizando esta envolvente y escuchar el sonido resultante, se nota un sonido pobre y lejano del que se podría escuchar en la realidad. Esto se debe a las discontinuidades abruptas que aparecen al aplicar cada uno de los segmentos correspondientes a cada parámetro.

Para hacer que sonido se asemeje más a uno real, se probó utilizar una envolvente del tipo ADSR pero con forma exponencial en lugar de que sea lineal.

Los resultados obtenidos fueron mas realistas que los resultados anteriores, el sonido no es mas suave, pero de todas formas le falta riqueza armónica.

3.1.2. Envolvente en base a muestras

Para obtener un sonido más realista, se obtuvieron las envolventes mediante muestras de una nota para cada instrumento, además de utilizar una envolvente distinta para cada armónico.

3.2. Procedimiento

Para sintetizar un instrumento, se tomó una muestra de una nota del instrumento elegido. A la muestra obtenida se le aplicó la transformada rápida de Fourier con el fin de poder analizar el contenido armónico de la misma. De aquí se sacó la información de cuáles son los armónicos que estan presentes en la señal, la amplitud y la fase de los mismos.

La amplitud nos dice que tan fuerte va a sonar ese armónico y la fase es para conocer el desfase temporal que existe entre los diferentes componentes.

Para optimizar, se buscaron los picos con mayor amplitud, además de que éstos se encuentren separados con una distancia de aproximadamente la frecuencia fundamental, a tal forma de no tomar las componentes de ruido que aparecen en la muestra.

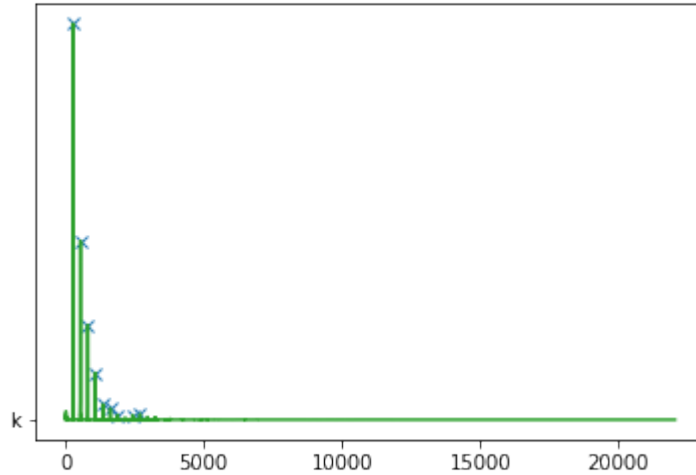


Figura 3: Transformada rápida de Fourier de la muestra con los picos utilizados para calcular la síntesis

Luego, para obtener la envolvente de cada armónico se utilizó la función 'spectrogram' de Python, de donde se obtiene la amplitud en función del tiempo para un cierto número de frecuencias, estas divididas en “bins”. En base a esta información, teniendo en cuenta en que bin caía la frecuencia del armónico se calculó la envolvente de amplitud que correspondía, encontrando los picos máximos y realizando una interpolación de estos.

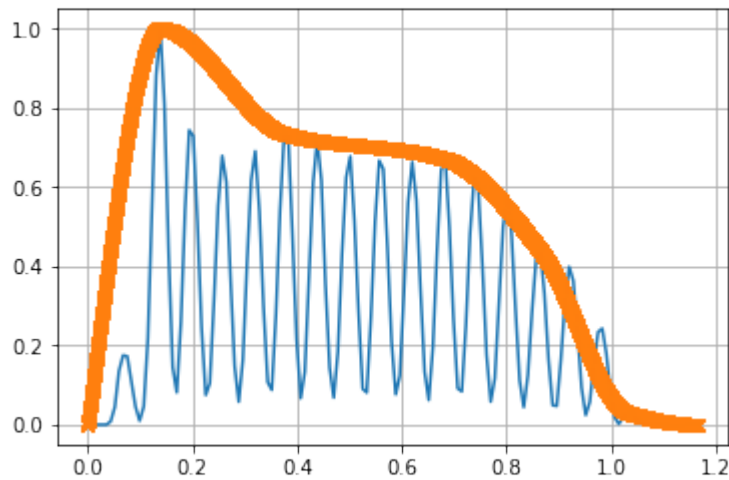


Figura 4: Envolvente obtenida para un armónico

Finalmente se construye la nota a sintetizar mediante la ecuación 10. A continuación se muestran la señal de entrada y la señal obtenida a la salida.

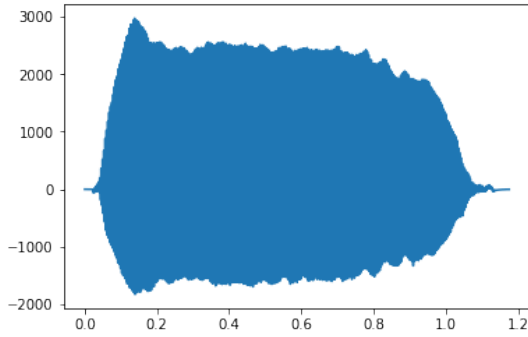


Figura 5: Señal de entrada

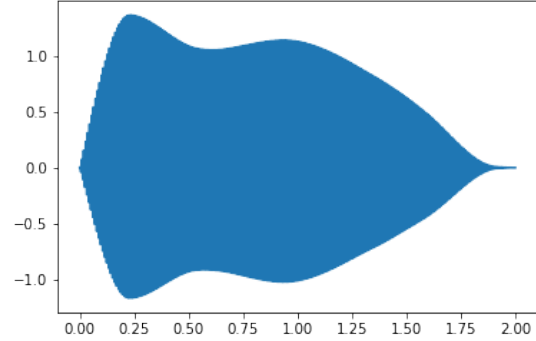


Figura 6: Señal de salida

La calidad del resultado obtenido depende de muchos parámetros. Las diferencias que se pueden observar entre la entrada y la salida se deben, por ejemplo, a que el número de armónicos tomados para el cálculo no coincide con el total, las envolventes calculadas para los parciales son una aproximación a la envolvente real, lo cual generará mayor discrepancia entre el sonido usado de muestra y el sonido obtenido mediante la síntesis.

3.3. Conclusiones

Si bien el sonido obtenido es bueno ya que el método permite un gran control sobre todos los parciales, se encontraron limitaciones en este tipo de síntesis. Por ejemplo, la gran cantidad de parámetros que se necesitan para poder conseguir un sonido más real. Otra limitación que se puede notar al sintetizar notas de distintas duraciones, es que el sonido se empieza a distorsionar cuando la duración de la nota es muy corta, y es más notorio cuando la duración llega a ser menor que el tiempo de ataque.

También, otra limitación es el rango de frecuencias que puede generar un instrumento, lo que provocan que si se sintetizara una frecuencia que no se encuentre dentro de su rango no se escuchará un sonido representativo del instrumento. Además, al utilizar una única muestra por instrumento, al alejarse más de un cierto rango de la frecuencia de la misma, el sonido ya se empieza a distorsionar.

Luego, se pudieron notar ciertas similitudes en las envolventes de los distintos grupos de instrumentos, como por ejemplo los de cuerda, como el piano y la guitarra (no está incluida en el trabajo pero se estudió su muestra), donde el tiempo de ataque es muy corto y luego la amplitud decae sin tener casi un tiempo donde haya un nivel de sustain y otro ejemplo son los instrumentos de viento, donde el nivel de sustain se mantiene casi toda la duración de la nota.

4. Síntesis mediante modulación en frecuencia

4.1. Introducción teórica

La modulación en frecuencia esta dada por:

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

Parámetros de la modulación en frecuencia:

- f_c = frecuencia de la señal portadora
- f_m = frecuencia de la señal modulante
- f_d = desviación de la frecuencia (respecto de la portadora)
- $I = \frac{f_d}{f_m}$ = índice de modulación

Por simple inspección se puede ver que si $I = 0$ la desviación de la frecuencia (f_d) debe ser 0 y por ende no hay modulación en frecuencia. Un caso más interesante es cuando I es mayor que 0, se puede observar que aparecen nuevas componentes espectrales lo que provoca que la energía de la señal sea redistribuida.

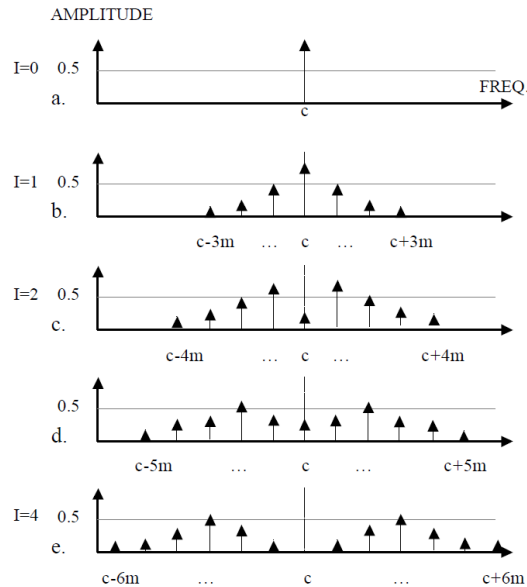


Figura 7: Nuevo contenido espectral con $I \neq 0$ ($c = f_c$ y $m = f_m$)

Las amplitudes de la portadora y las componentes laterales están dadas por las funciones de Bessel de primera especie y orden n-ésimo, donde el argumento de la función es el índice de modulación I ($J_n(I)$).

En general, $J_n(I)$ representa un coeficiente de escalamiento de amplitud:

$J_0(I) \leftarrow$ para la portadora

$J_1(I) \leftarrow$ para las primeras bandas laterales

$J_2(I) \leftarrow$ para las segundas bandas laterales

Y así sucesivamente. Mientras más grande sea el orden de la banda lateral se necesita más índice de modulación para que la misma tenga una amplitud significativa.

$$BW \approx 2(f_d + f_m)$$

Expresamos a $x(t)$ con funciones de Bessel:

$$\begin{aligned} x(t) = A(t) \cdot \{ & J_0(I) \text{sen}(2\pi f_c t) \\ & + J_1(I) [\text{sen}(2\pi(f_c + f_m)t) - \text{sin}(2\pi(f_c - f_m)t)] \\ & + J_2(I) [\text{sen}(2\pi(f_c + 2f_m)t) + \text{sin}(2\pi(f_c - 2f_m)t)] \\ & \dots \\ & + J_n(I) [\text{sen}(2\pi(f_c + nf_m)t) \pm \text{sin}(2\pi(f_c - nf_m)t)] \} \end{aligned}$$

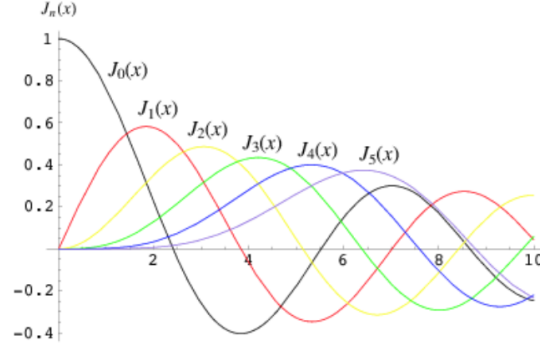


Figura 8: Funciones de Bessel de primer especie desde J_0 hasta J_5

Se puede ver que las bandas laterales bajas de orden impar poseen un signo negativo. Además se sabe que para un $I > 2.5$ las funciones de Bessel producen coeficientes de escalamiento negativos para algunas componentes. En general se suelen ignorar los signos negativos en el espectro ya que solo indican una inversión de la fase de la frecuencia correspondiente. En la próxima sección se verá por qué en realidad si son importantes esas inversiones de fase.

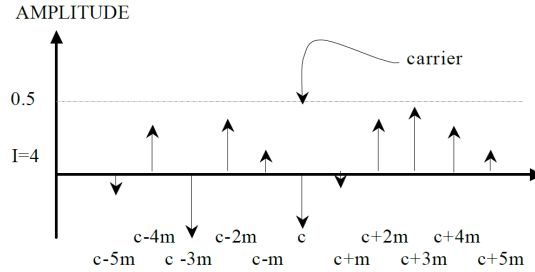


Figura 9: Espectro de ejemplo con $I=4$ e inversiones de fase

4.2. Frecuencias laterales reflejadas

La riqueza de la técnica de modulación en frecuencia yace en que el espectro ubicado en la parte negativa del dominio interfiere con la parte positiva dando lugar a una mezcla de componentes en ambas partes del espectro. La relación entre frecuencias f_c y f_m es la que da origen a un espectro armónico o uno inarmónico.

4.3. Espectro armónico e inarmónico

La relación entre frecuencias esta dada por

$$\frac{f_c}{f_m} = \frac{N_1}{N_2}$$

$$f_0 = \gcd(f_c, f_m)$$

La posición de las frecuencias laterales en serie armónica puede ser determinada por las siguientes relaciones:

$$k = N_1 \pm nN_2 \text{ para } n = 0, 1, 2, 3, 4 \dots$$

Donde k es el número de armónico y n es el orden de la frecuencia lateral. Excepto para $n = 0$, k toma dos valores por cada orden.

A continuación dejamos unas generalizaciones útiles:

- La portadora es siempre el N_1 -ésimo armónico en la serie
- Si $N_2 = 1$ el espectro contiene todos los armónicos y el fundamental es la frecuencia de la modulante
- Si N_2 es un número par el espectro tiene solo armónicos impares
- Si $N_2 = 3$ cada tercer armónico no se encuentra más en la serie

El número de armónicos que tienen amplitud significativa dependen del índice de modulación.

Para relaciones entre frecuencias bajas e índices chicos donde $N_1 \neq 1$ la fundamental puede no estar presente en el espectro.

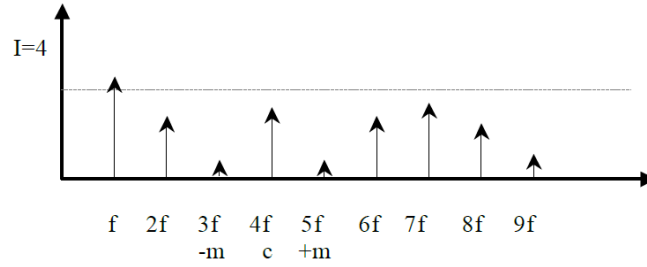


Figura 10: Espectro donde la relación $\frac{f_c}{f_m} = \frac{4}{1}$ (armónico) con $I=4$

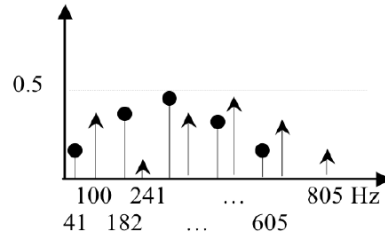


Figura 11: Espectro donde la relación $\frac{f_c}{f_m} = \frac{1}{\sqrt{2}}$ (inarmónico) con $I=4$

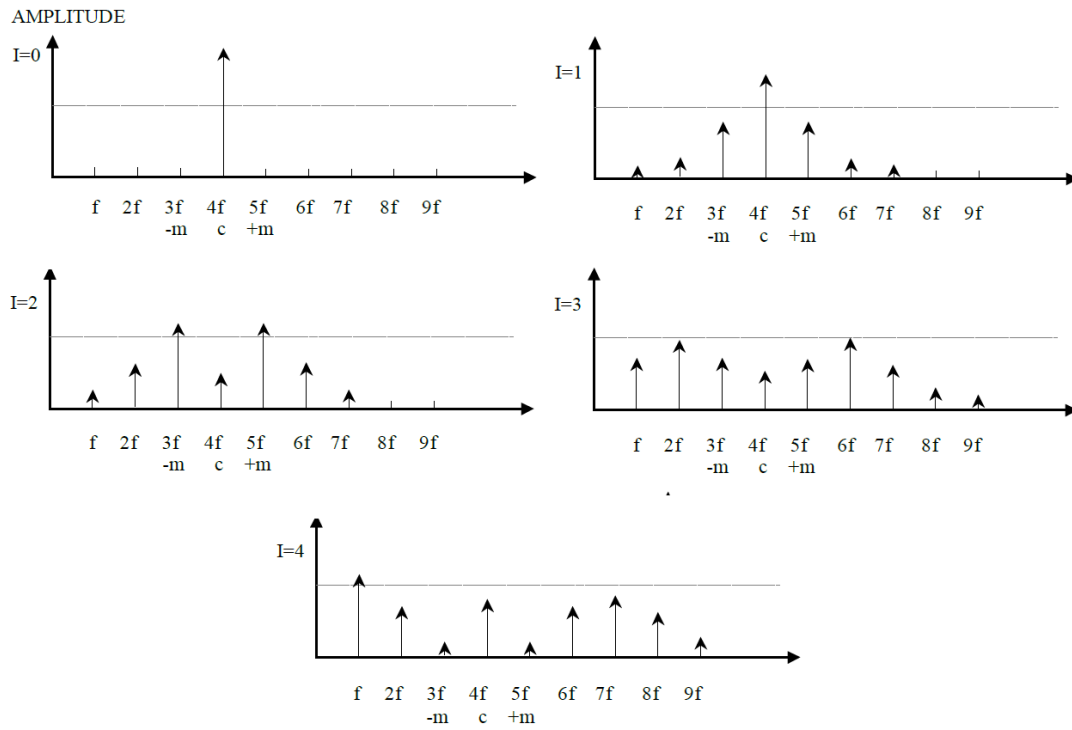


Figura 12: Espectros con distintos índices de modulación (cociente racional)

En resumen:

El espectro inarmónico se origina cuando el cociente $\frac{f_c}{f_m}$ da como resultado un número irracional.

El índice de modulación $\frac{f_d}{f_m}$ determina el número de componentes que van a tener amplitud significativa.

4.4. Espectro dinámico

La complejidad del espectro esta relacionada con el índice de modulación de forma tal que si el índice crece el bandwidth también crece. Entonces si pensamos en un índice de modulación que varía con el tiempo la evolución del bandwidth del espectro puede estar generalmente descripta por la forma de la función. No obstante, la evolución de cada componente es determinado por la forma de la funcion de Bessel correspondiente. Entonces, si el índice de modulación aumenta con el tiempo el bandwidth también lo hace, pero un componente del espectro va a crecer o decrecer en amplitud dependiendo de la pendiente de la función de Bessel en ese rango de indices.

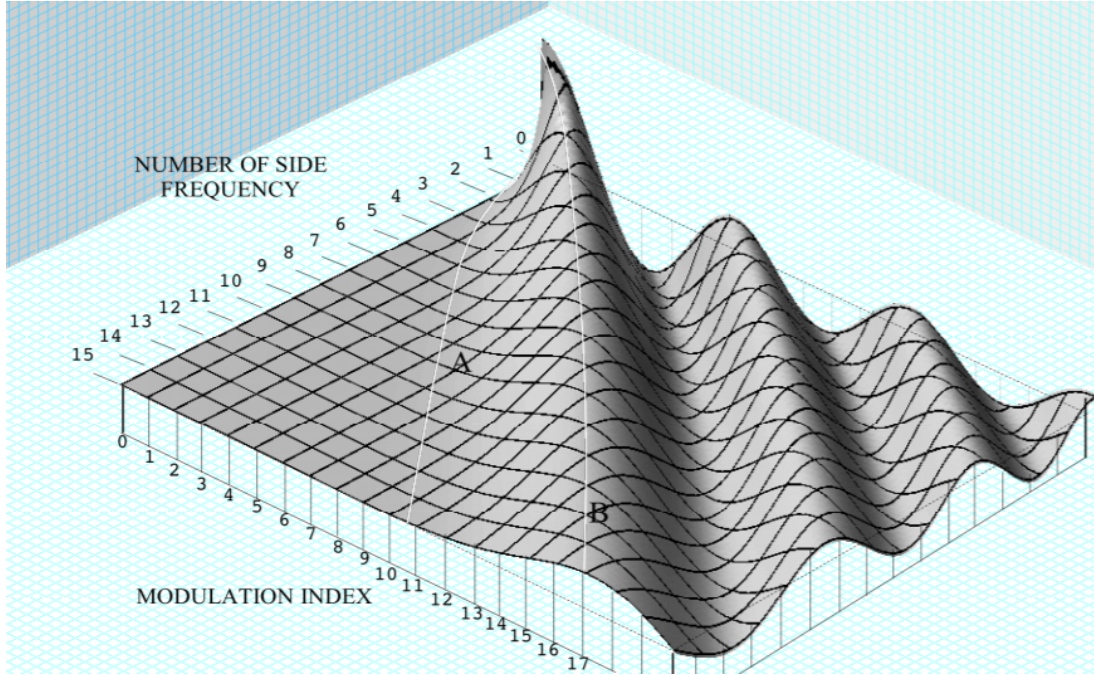


Figura 13: Funciones de Bessel desde J_0 hasta J_{15} e I desde 0 hasta 20. En este gráfico se puede ver rápidamente el bandwidth dado un índice I

4.5. Síntesis para instrumentos de viento

Para los instrumentos de viento:

$$\phi_m = \phi_c = -\frac{\pi}{2}$$

Resultando

$$x(t) = A(t) \cdot \text{sen}(2\pi f_c t + I(t) \cdot \text{sen}(2\pi f_m t))$$

4.5.1. Síntesis de clarinete

Para poder sintetizar un clarinete se establecieron los parámetros recomendados por la literatura:

$$\frac{f_c}{f_m} = \frac{2}{3}$$

Las envolventes sugeridas por la bibliografía poseen la siguiente forma:

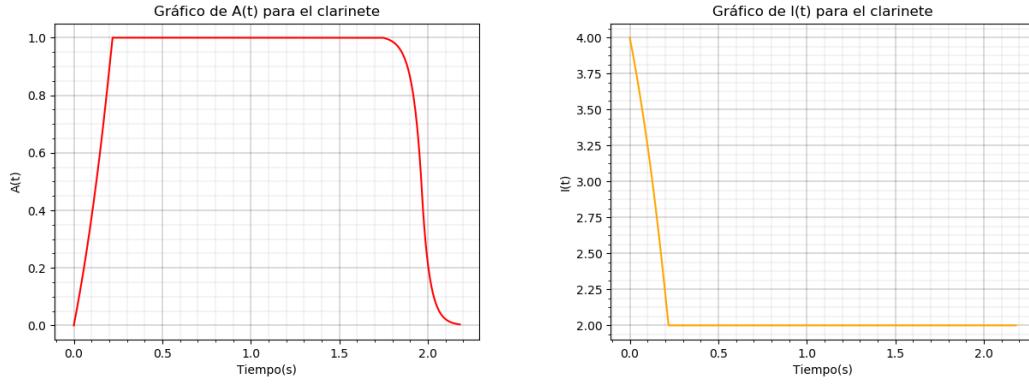


Figura 14: A(t) e I(t) respectivamente

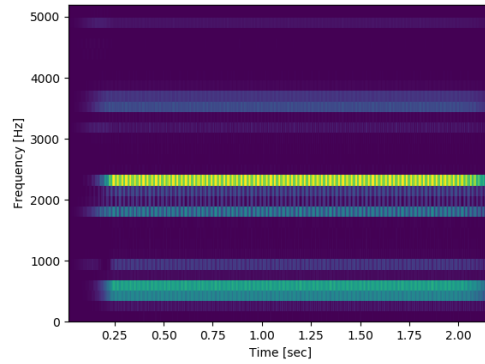


Figura 15: Espectrograma de la síntesis del clarinete a 440Hz (nota LA)

4.5.2. Síntesis de campana

Los parámetros recomendados por la literatura para la síntesis son:

$$\frac{f_c}{f_m} = \frac{2}{1}$$

Envolventes sugeridos:

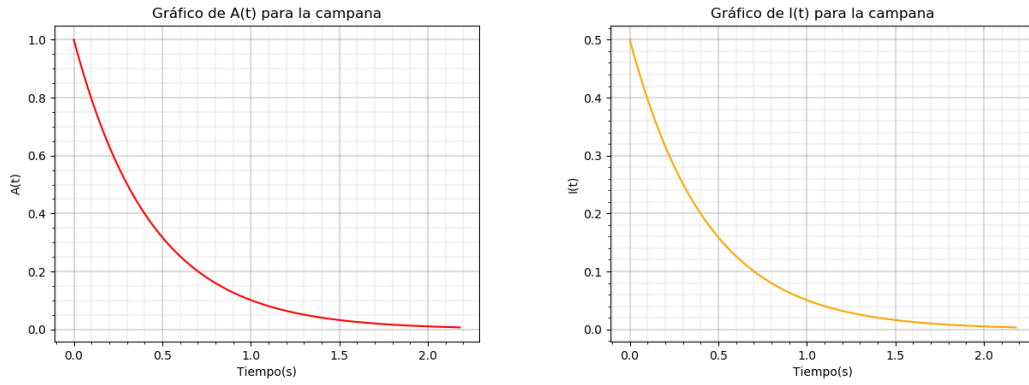


Figura 16: A(t) e I(t) respectivamente

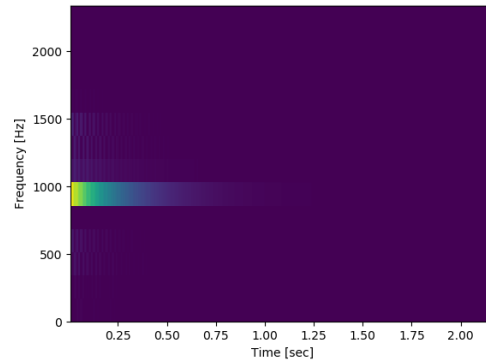


Figura 17: Espectrograma de la síntesis de la campana a 440Hz (nota LA)

4.5.3. Síntesis de trombón

Para los Brass Instruments se recomienda utilizar una relación:

$$\frac{f_c}{f_m} = \frac{1}{1}$$

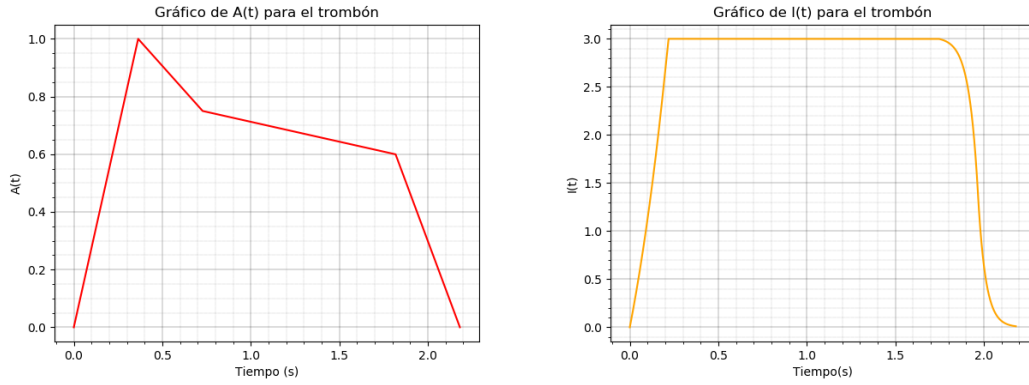


Figura 18: A(t) e I(t) respectivamente

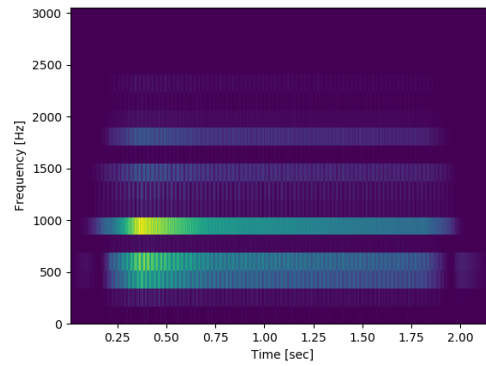


Figura 19: Espectrograma de la síntesis del trombón a 440Hz (nota LA)

4.5.4. Síntesis de la trompeta

En el caso de la trompeta se sugiere también utilizar una relación:

$$\frac{f_c}{f_m} = \frac{1}{1}$$

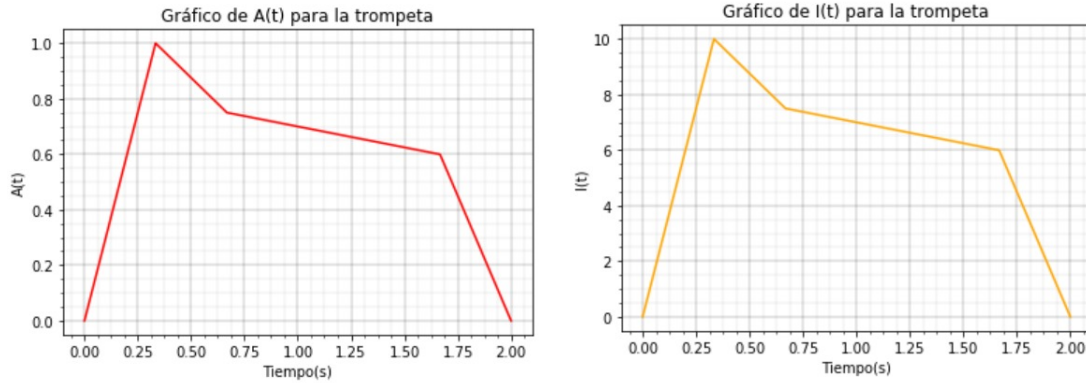


Figura 20: A(t) e I(t) respectivamente

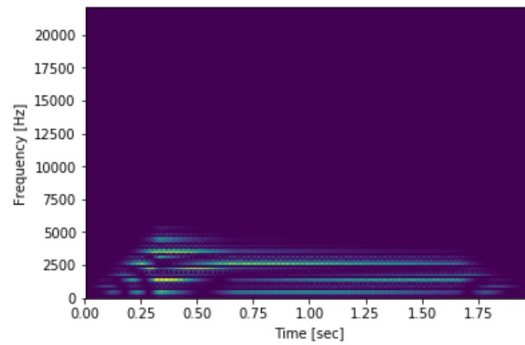


Figura 21: Espectrograma de la síntesis de la trompeta a 440Hz (nota LA)

4.6. Conclusiones

Para poder realizar una síntesis de un instrumento suele ser útil tener espectrogramas del mismo ya que nos dan información de cómo evoluciona el sonido con el tiempo y de esa forma se puede llegar a tener una idea de cómo puede llegar a ser A(t) e I(t). Si bien en la literatura hay envoltentes sugeridas, las mismas son sólo un molde inicial por el cual uno debería empezar a probar la síntesis pero para poder ejecutar una síntesis apropiada se debe cambiar el molde. De la misma forma la relación entre f_m y f_c no siempre tiene que ser la sugerida.

Referencias

- [1] The Synthesis of Complex Audio Spectra by Means of Frequency Modulation,
https://ccrma.stanford.edu/sites/default/files/user/jc/fm_synthesispaper-2.pdf
- [2] FM Synthesis for Musical Instruments: Bells and Clarinets,
http://home.eng.iastate.edu/~julied/classes/ee224/Labs/FMSynthesis_lab.pdf

5. Síntesis de sonidos mediante modelos físicos

Se analizarán dos variantes del modelo Karplus-Strong en paralelo, tanto de manera teórica como práctica.

5.1. Bloque A Elemental

Se resolverá, como cálculo auxiliar un bloque sencillo definido como:

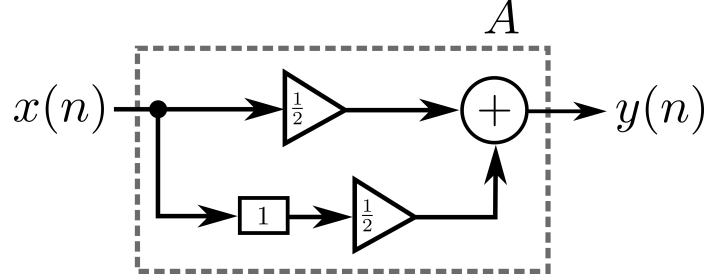


Figura 22: Bloque elemental

Este bloque solo promedia los dos ultimos valores de entrada. Su transferencia esta dada por

$$y(n) = \frac{1}{2}x(n) + \frac{1}{2}x(n-1) \quad (11)$$

$$Y(z) = \frac{1}{2}X(z) + \frac{1}{2}X(z)z^{-1} \implies A(z) = \frac{z+1}{z} \quad (12)$$

Se puede observar que el bloque A es pasa-bajos (cero en $Z = -1$); lo cual es, en principio, razonable, el bloque A suaviza la entrada.

5.2. Karplus Strong 1

5.2.1. Análisis teórico

Se resolverá un nuevo sistema, denominado S_1 , el cuál consiste en una adición de realimentación al sistema anterior.

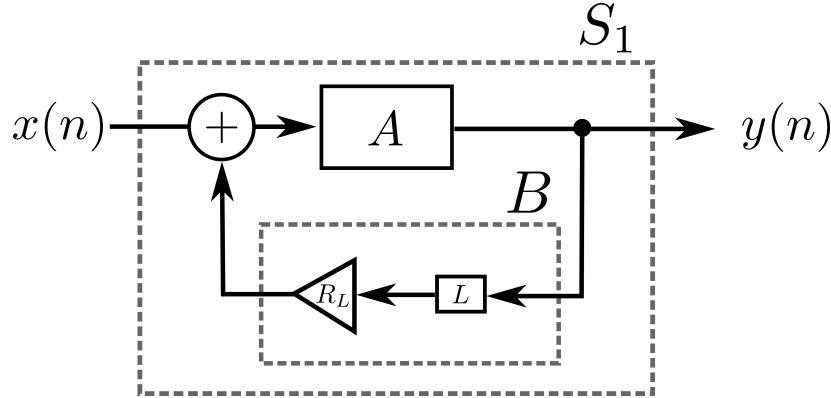


Figura 23: Bloque elemental

Mediante teoria de feedback a considerando que $B(z) = z^{-L}R_L$ se llega a que

$$S_1(z) = \frac{\frac{1}{2} + \frac{1}{2}z^{-1}}{1 - \frac{1}{2}R_Lz^{-L} - \frac{1}{2}R_Lz^{-L-1}} = \frac{\frac{1}{2}z^{L+1} + \frac{1}{2}z^L}{z^{L+1} - \frac{1}{2}R_Lz - \frac{1}{2}R_L} \quad (13)$$

$$y(n) = \frac{1}{2}x(n) + \frac{1}{2}x(n-1) + \frac{1}{2}R_Ly(n-L) + \frac{1}{2}R_Ly(n-L-1) \quad (14)$$

Es decir, una expresión con $L + 1$ polos y $L + 1$ ceros, en otras palabras, una ecuación diferencial de orden $L + 1$ con vector de longitud L como condición inicial. De la ecuación de diferencias se puede observar que la única forma de garantizar estrictamente la estabilidad del sistema será exigiendo $R_L < 1$. No obstante, en la práctica como la frecuencia de resonancia del sistema no será exacta; colocar $R_L = 1$ no provocará inestabilidad. Más aun; resulta provechoso aproximar R_L a 1 lo más posible para lograr estirar la duración de las oscilaciones.

5.2.2. Estudio de la distribución polos y ceros, respuesta en frecuencia

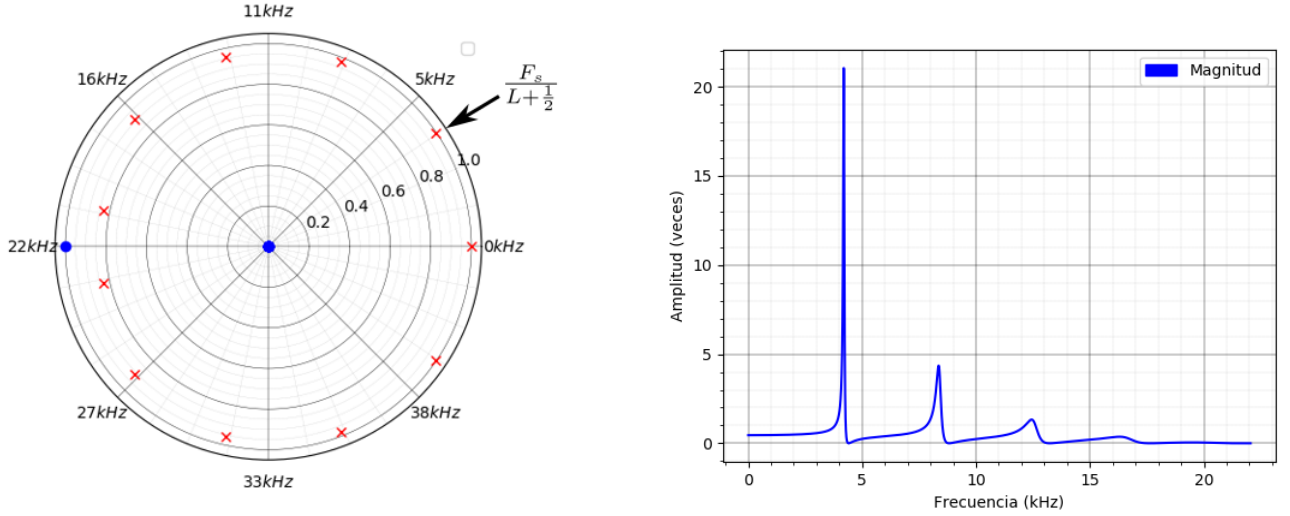


Figura 24: Polos y ceros (derecha), Rta en frecuencia (izquierda), S_1 con $R_L = 1$, $L = 10$, $f_s = 44.1kHz$

Del diagrama de polos y ceros y la respuesta en frecuencia podemos observar que hay una frecuencia de resonancia $F_R = F_s / (L + \frac{1}{2})$ que tiende a cumplir las hipótesis del criterio de Barkhausen, y por lo tanto provocar oscilaciones, lo cuál es el objetivo del bloque; conseguir una salida que perdure en el tiempo a partir de una entrada de longitud L muy corta.

5.2.3. Cálculo de F_R

Se mostrará porque el sistema resuena en $F_R = F_s / (L + \frac{1}{2})$. En la sección anterior se observó que la frecuencia de resonancia debía ser aquella que tendiera a cumplir las hipótesis del criterio de barkhausen; en otras palabras; que la ganancia del lazo se aproxime a 1 con 0 grados. El sistema esta compuesto por la superposición de un sistema con un retraso de lazo L y otro sistema con retraso de lazo $L + 1$. Por lo tanto; para que una señal no tienda a interferirse al recorrer el lazo debe tener periodo $\frac{L+L+1}{2}$; es decir una frecuencia $\frac{F_s}{L+1/2}$

5.2.4. Análisis mediante señales

Se procederá a estudiar como el sistema responde a diversas entradas, entre ellas, un impulso unitario, ruido gaussiano de longitud L , y ruido lineal de longitud L . Se decidió aumentar L en estos casos a 50 para conseguir frecuencias más cercanas a las audibles.

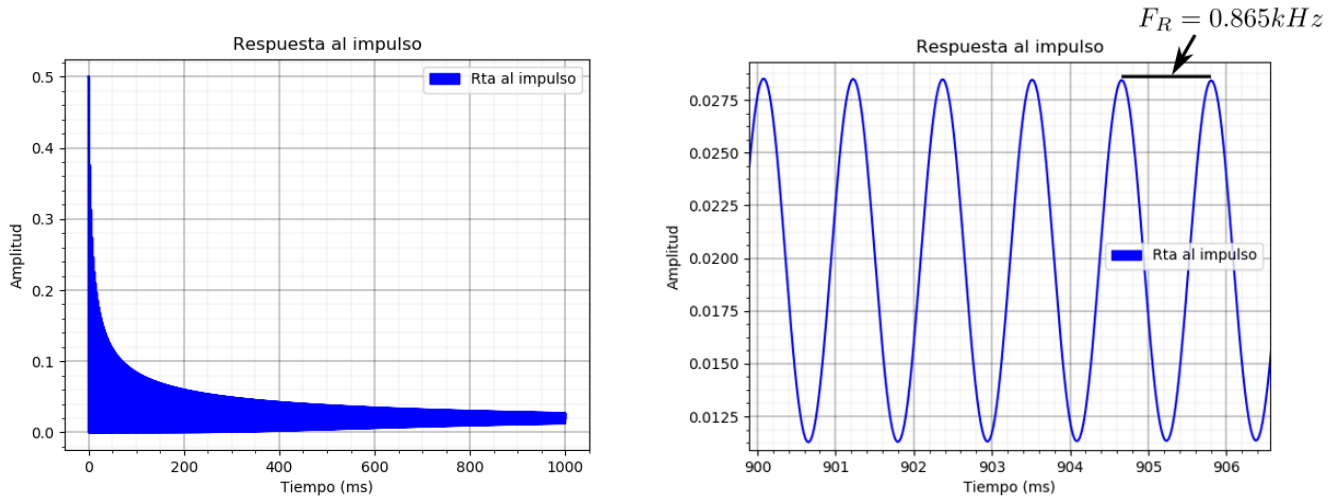


Figura 25: Respuesta al impulso con y sin zoom, $R_L = 1$, $L = 50$, $f_s = 44.1kHz$

Se puede observar que, de todas las frecuencias pertenecientes al impulso, la más amplificada vale $865Hz \approx \frac{f_s}{1/2+L}$. Por otro lado es importante observar que la salida se encuentra montada sobre una tensión continua, más precisamente, nunca es menor que 0. Esto se debe a que cuando la excitación es exclusivamente positiva, como la realimentación es positiva y no hay inversiones la salida siempre será positiva. Por ello es importante que la excitación inicial sea tanto positiva como negativa para lograr que la salida este centrada en 0.

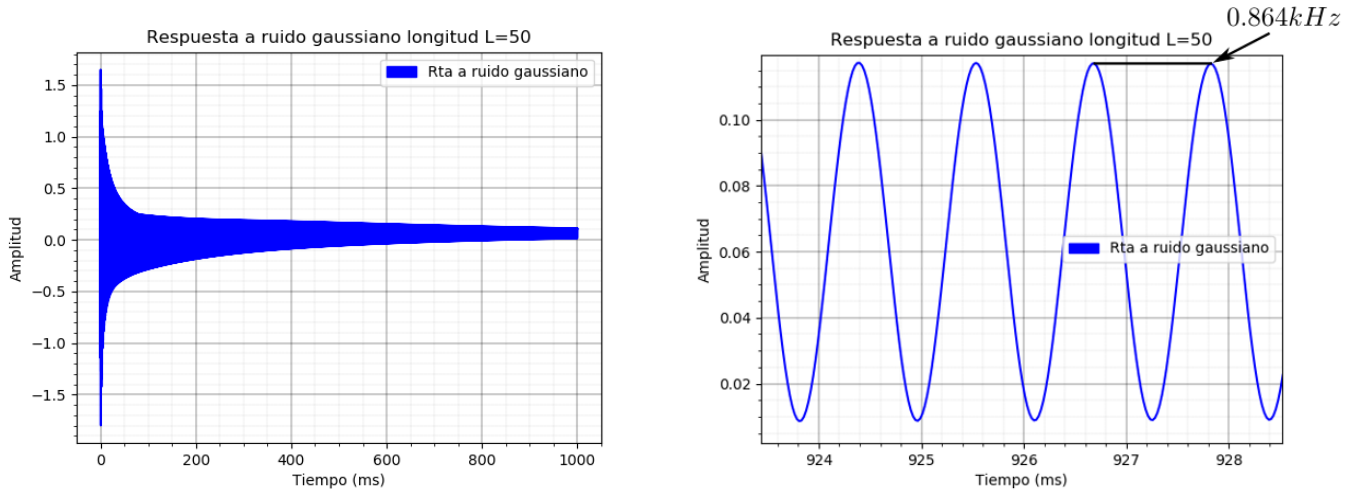


Figura 26: Respuesta a entrada Gaussiana con y sin zoom, $R_L = 1$, $L = 50$, $f_s = 44.1kHz$

Se puede ver que la salida fue a la misma frecuencia que en el caso anterior, al mismo tiempo que nuevamente, estuvo montada sobre una continua. No obstante la salida tuvo la posibilidad de tomar valores negativos.

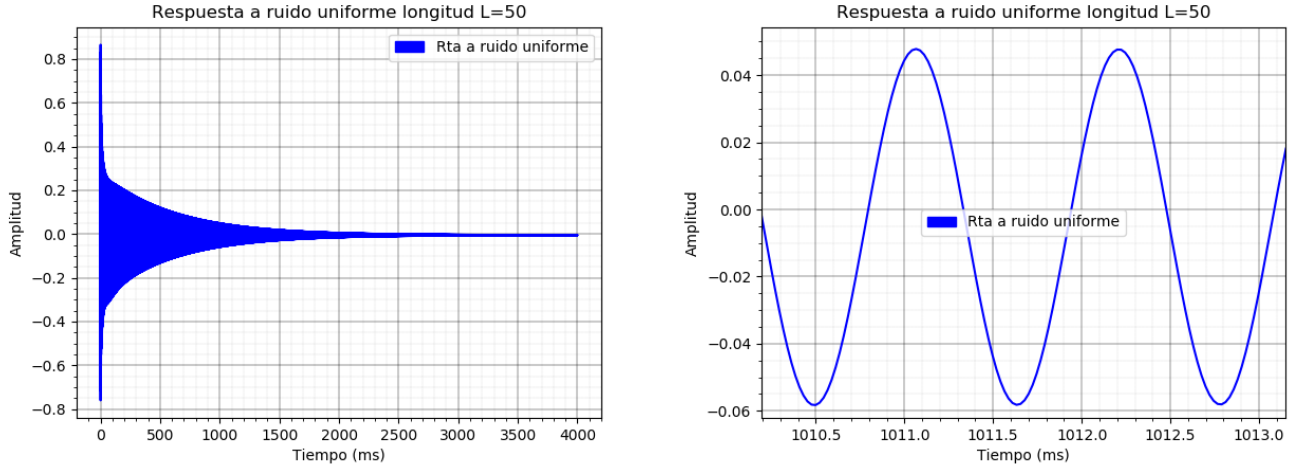


Figura 27: Respuesta a entrada de ruido uniforme con y sin zoom, $R_L = 1$, $L = 50$, $f_s = 44.1kHz$

Podemos observar que la salida fue similar a la de ruido gaussiano

5.3. Karplus Strong 2

5.3.1. Análisis teórico elemental

Se estudiará un nuevo sistema con una pequeña modificación la cual consiste en agregar un multiplicador en el bloque realimentador, cuyo valor es aleatorio, puede ser 1 o -1. La probabilidad de que sea 1 se llamará b

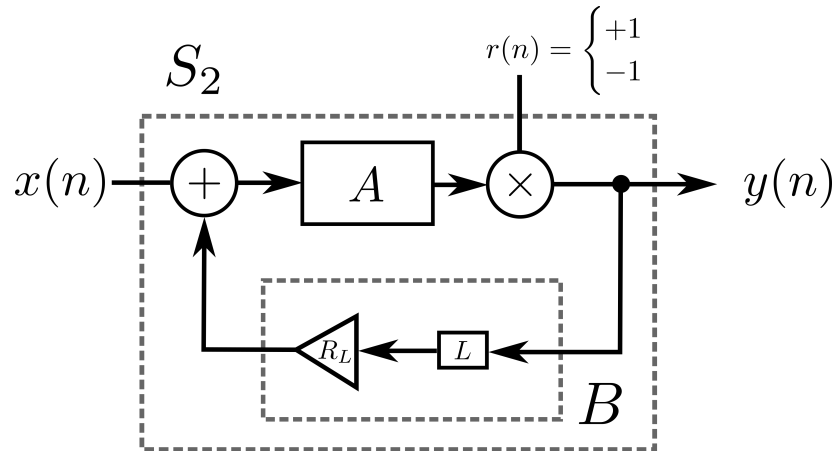


Figura 28: Diagrama del sistema S_2

Escrito en lenguaje matemático

$$A(z) = b\left(\frac{1}{2} + \frac{1}{2}z^{-1}\right) \quad (15)$$

$$B(z) = \frac{1}{2}z^{-L} \quad (16)$$

Usando teoría de feedback concluimos que

$$S_2(z) = \frac{\frac{1}{2}r + \frac{1}{2}rz^{-1}}{1 - \frac{1}{2}rR_Lz^{-L} - r\frac{1}{2}R_Lz^{-L-1}} \quad (17)$$

$$y(n) = \frac{1}{2}rx(n) + \frac{1}{2}rx(n-1) + \frac{1}{2}rR_Ly(n-L) + \frac{1}{2}rR_Ly(n-L-1) \quad (18)$$

Observamos de la ecuación de diferencias que el sistema es prácticamente equivalente al anterior con una diferencia; ahora la salida se invierte de manera aleatoria ciclo a ciclo. Esta aleatoriedad ayudará a simular la aleatoriedad de un sonido percutado.

5.3.2. Respuesta en frecuencia

Debido a la complejidad trigonométrica del cálculo exacto se optó por aproximar numéricamente el gráfico de $\phi(w) = \angle H(e^{jw})$ (respuesta en frecuencia del sistema). Por otro lado se decidió tampoco realizar el cálculo de la frecuencia de resonancia con exactitud, debido a su elevada complejidad. En realidad no es estrictamente necesario, en realidad en la práctica cuando b sea distinto de 1 o 0 ocurrirá notable interferencia en el lazo que destruya la resonancia; lo cual provocará que el sonido se extinga luego de breves instantes.

Ahora se mostrará la respuesta en frecuencia con $b = 1, b = \frac{1}{2}, b = 0$. Es importante observar que dado que la salida es estrictamente una variable aleatoria, la respuesta en frecuencia entonces estrictamente también es una variable aleatoria. Se decidió mostrar el valor medio de dicha variable aleatoria.

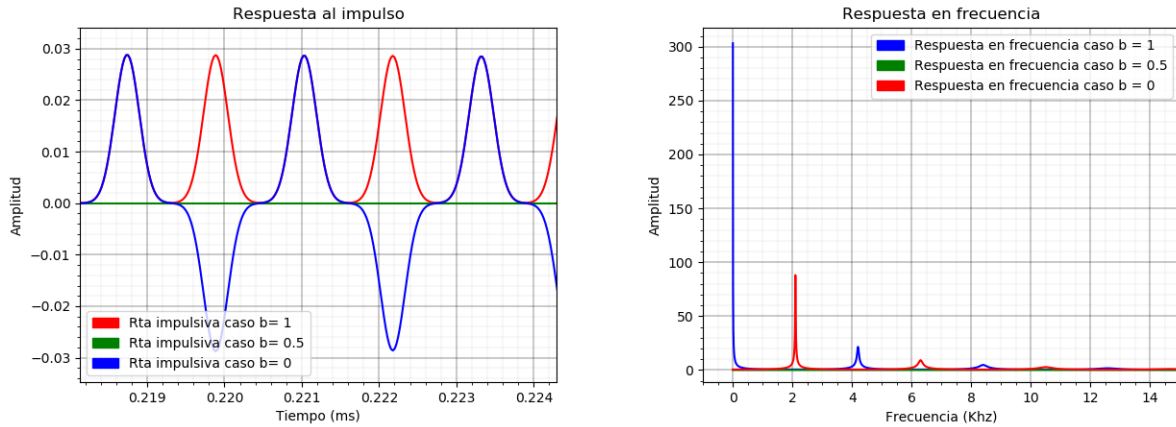


Figura 29: Respuesta al impulso y en frecuencia con $b = 1, 0.5, 0$ (izquierda a derecha)

Se observa que, como fue predicho, en el caso $b=0.5$ todas las frecuencias son destruidas luego de un pequeño transitorio. Además, se observó que el caso de $b=0$ mostró una inversión espectral que provocó que la frecuencia de resonancia se redujera en dicho caso lo cual implicó un sonido más grave, como el de un arpa. También se observó una gran amplificación en el caso $b = 1$ en la continua. Esto no es un gran problema ya que si la entrada tiene valor medio cercano a 0 y se filtra dicha componente a la salida es un problema evitable.

5.4. Sintetización de instrumentos

En base a las distintas pruebas realizadas anteriormente se procederá a realizar las funciones que permitan sintetizar el sonido de varios instrumentos que son:

- Arpa
- Guitarra
- Tambor (percusión)

5.4.1. ¿Como tener libertad con la frecuencia fundamental?

La primera necesidad fue tener la libertad de poder sintetizar un sonido con cualquier frecuencia que se necesite. La solución fue sencilla; consistió en modificar el bloque A para que nos permita una mayor libertad al elegir la frecuencia

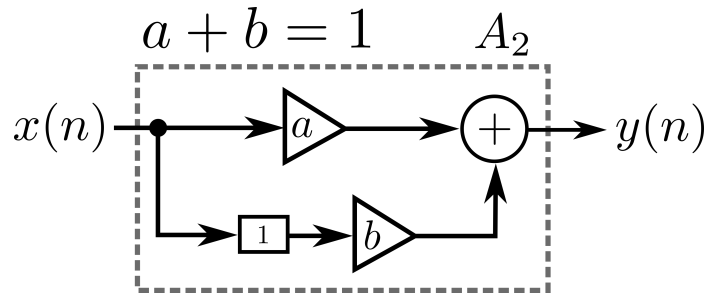


Figura 30: Modificación bloque A para poder personalizar la frecuencia

Con la nueva modificación la frecuencia de resonancia paso a estar dada por

$$F_R = F_s \left(a \frac{L}{2} + b \frac{L+1}{2} \right) \quad (19)$$

Eligiendo L adecuadamente se puede garantizar que existan a y b entre 0 y 1 que verifiquen la ecuación anterior

5.4.2. ¿Cómo filtrar la componente continua?

Se colocó un filtro pasabajo al final del filtro para eliminar la “molesta” componente continua que el sistema poseía. Dicha característica de la señal era aleatoria e impredecible por lo tanto fue muy importante colocar dicho filtro.

5.4.3. ¿Como atenuar el sonido de manera suave?

Cuando termina el sonido la amplitud no puede ser disminuida de manera inmediata. Se necesita entonces multiplicar la señal por una ventana para regular la duración y provocar una disminución gradual de la intensidad del sonido

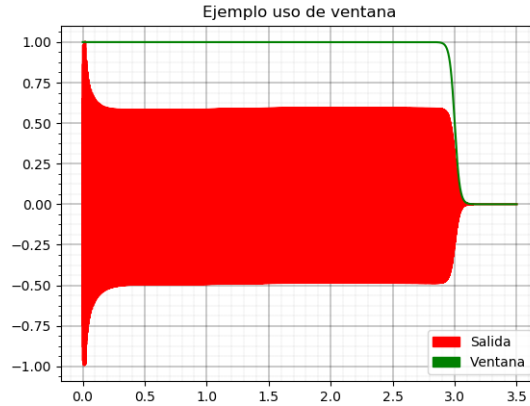


Figura 31: Ejemplo uso de ventana

Los resultados del uso de ventanas fueron satisfactorios, la atenuación del sonido se observó de forma gradual

5.4.4. Sintetización de sonido de guitarra

Se implemento el sonido de una guitarra utilizando el modelo Karplus Strong utilizando la modificación que permitió una elección continua de frecuencias. A continuación se mostrará el espectrograma del sonido de la guitarra producido.

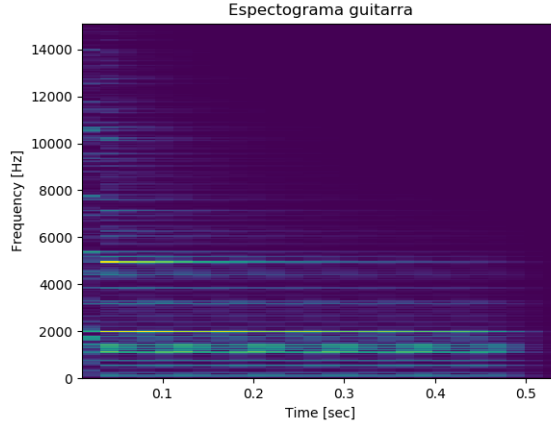


Figura 32: Espectrograma sonido de guitarra generado, nota: $A_0 = 440Hz$

El sonido escuchado fue como el de una guitarra, por lo tanto se consideró que los resultados fueron satisfactorios; los armónicos se aproximaron a los de una guitarra acústica.

5.4.5. Distorsión del sonido de guitarra

Se distorsionó el sonido de una guitarra utilizando una modificación al modelo. Se colocó un valor de $R_L > 1$ pero, utilizando una función normalizadora (entre -1 y 1) se conformó un control automático de ganancia que evitó la inestabilidad del sistema. Colocando diversos valores de R_L se armó distorsión de distintos tipos. Se mostrará un espectrograma de uno de los sonidos generados

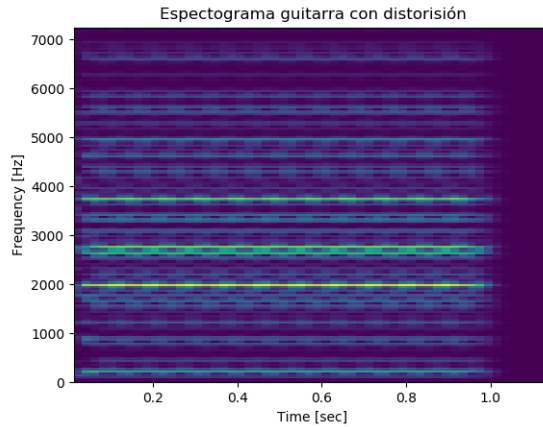


Figura 33: Espectrograma sonido de guitarra con distorsión, $R_L = 2$, nota: $A_0 = 440Hz$

Se observa que en general los armónicos generados producto de la distorsión son muchos más numerosos y además tienden a tener una frecuencia mayor que los de la guitarra acústica. El sonido generado fue percibido como el de un sonido eléctrico, que sonaba bien pero que no se asocia a ningún instrumento acústico.

5.4.6. Sintetización de instrumento de percusión

Utilizando la modificación del modelo de Karplus Strong la cual establecía una variable aleatoria se sintetizó el sonido de instrumentos de percusión. Se necesitó aplicar algunas modificaciones, en primer lugar se decidió colocar un valor de L elevado para elevar el tiempo de decaimiento. Por otro lado, se provocó que el estímulo al sistema cayera de manera abrupta, de otra manera no se podía observar el sonido percutido, sino simplemente ruido. A continuación se mostrará la salida en el tiempo y en frecuencia.

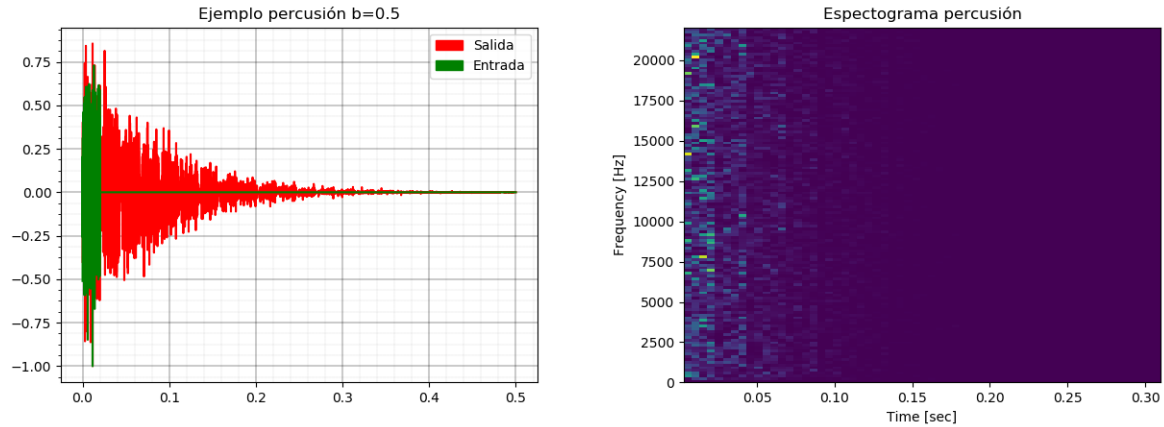


Figura 34: Respuesta en tiempo y en frecuencia de sonido percutado

5.4.7. Explicación caja de resonancia

Se explicará el funcionamiento de la caja de resonancia de una guitarra. No se implementará ya que la necesidad que existe de utilizar dicho elemento en una guitarra es física, en realidad, desde la perspectiva de señales, es posible sintetizar el sonido de una guitarra sin necesidad de simular la caja. Lo que se busca cuando se fabrica una guitarra es que el sonido generado por el estimo de las cuerdas se propague en una dirección con una intensidad tal, que permita una transmisión efectiva por el aire. Sin embargo, el aire es un medio de impedancia baja mientras que las cuerdas son una fuente de impedancia alta. Por lo tanto de solo existir las cuerdas el sonido si bien en el sentido más estricto sería producido, no lograría propagarse con intensidad suficiente. Por lo tanto surge la necesidad de agregar un elemento físico adicional, la caja de resonancia, que toma el sonido propagado por las cuerdas hacia “adentro” y lo vuelve a propagar hacia “afuera” con una intensidad mayor, amplificando todas las frecuencias por igual (de lo contrario ocurriría distorsión).

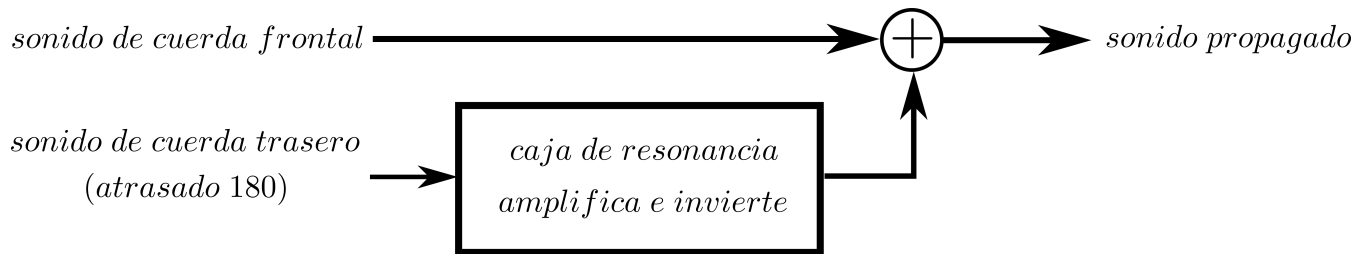


Figura 35: Caja de resonancia

Además impone una inversión de fase, idealmente en todas las frecuencias, ya que el sonido propagado por las cuerdas hacia “adentro” esta invertido en fase comparando con el emitido por las cuerdas hacia “afuera”.

Los diseños modernos de cajas de resonancia intentan maximizar tanto la inversión de fase como la amplificación de la mayor cantidad de frecuencias, en particular las más bajas que tienen más relevancia en el sonido.

6. Síntesis basada en muestras

La síntesis basada en muestras permite que a partir de una muestra con cierta duración, se pueda obtener una nota distinta a la original y con distinta duración achicando o alargando la muestra. Para ello es necesario poder modificar tanto la frecuencia fundamental como la duración independientemente.

La duración de la nota se puede cambiar a partir del resampleo. Al resamplear una señal, no solo se modifica la duración, sino que también la frecuencia fundamental de la misma.

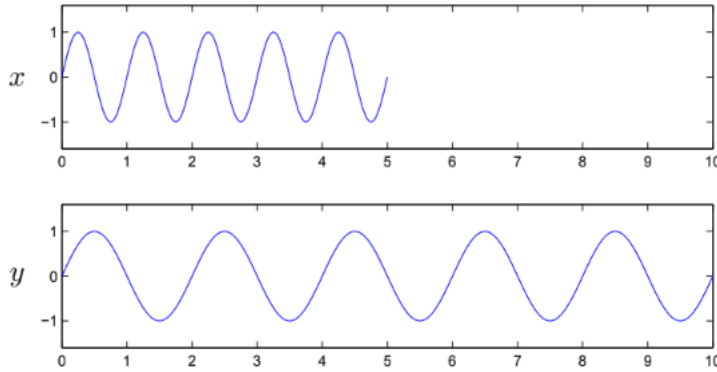


Figura 36: x está sampleada a una frecuencia de 22050 Hz y tiene una frecuencia de 1Hz. Resampleando la señal con una frecuencia de 44100 Hz se obtiene la señal y que tiene el doble de duración que x y la mitad de frecuencia (0.5Hz)

Por lo tanto, una vez cambiada la frecuencia mediante dicha técnica, se necesita un algoritmo que sea capaz de cambiar la duración del audio sin modificar nuevamente su frecuencia.

6.1. Escalamiento Temporal (TSM)

Existe una gran variedad de algoritmos que cumplen con la tarea de modificar la duración de un audio. Ya se mencionó el ejemplo del resampleo, a continuación se analizará otro algoritmo denominado Overlap-Add (OLA) que elimina en la mayor escala posible el problema de la alteración en la frecuencia.

6.1.1. Overlap-Add (OLA)

La idea de este algoritmo consiste en cortar la señal de entrada en pequeños segmentos y luego encadenarlos para obtener la longitud deseada.

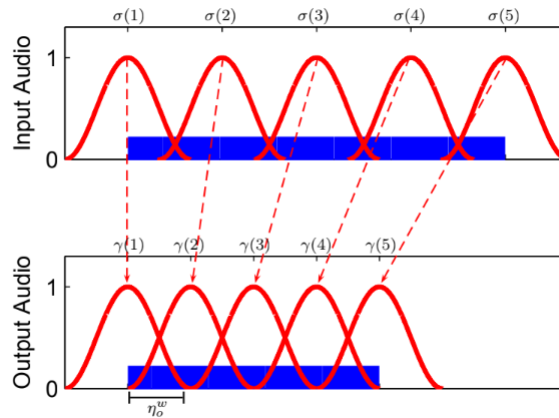


Figura 37: Algoritmo OLA

Los parámetros que utiliza el algoritmo son la muestra original, un factor de solapamiento, una función ventana y una función que 'mapea' los tiempos de la escala original a la nueva escala. Matemáticamente se puede escribir esta síntesis como:

$$y(n) = \frac{\sum_{k=1}^{len(\sigma)} \omega(n - \gamma(k))x(n - \gamma(k) + \sigma(k))}{\sum_{k=1}^{len(\sigma)} \omega(n - \gamma(k))} \quad (20)$$

Donde x es la señal de muestra, ω es la función ventana (en este caso utilizamos la ventana Hann, cuya longitud debe ser mayor que a lo sumo un período de la señal), σ es el vector que contiene las posiciones de las ventanas en la escala de entrada y γ es el vector con las posiciones de las ventanas en la escala de salida. Estos dos están vinculados mediante:

$$\sigma(n) = \tau^{-1}(\gamma(n)) \quad (21)$$

Donde τ es la función que convierte los puntos de la escala original a la deseada.

Entonces, como muestra la ecuación, se toma segmentos del original definidos por el tamaño y la amplitud de la ventana utilizada y luego se solapan estos segmentos de acuerdo lo indica la función τ , que es la que estira o acorta la escala de acuerdo con la duración que se desea para la salida.

La calidad del sonido que se obtiene a partir de este algoritmo es más aceptable para muestras que no tienen naturaleza armónica como es por ejemplo una señal de voz. Pero cuando se trata de estiramiento temporal para una señal con contenido armónico, como por ejemplo una nota musical, se introducen en la salida ciertos defectos como modulación y saltos de fase producidos por el solapamiento de las ventanas.

A continuación se muestran los resultados obtenidos para una señal con contenido armónico, donde se utilizó como señal de entrada una senoidal pura, y para una señal no armónica, utilizando como muestra una nota de voz.

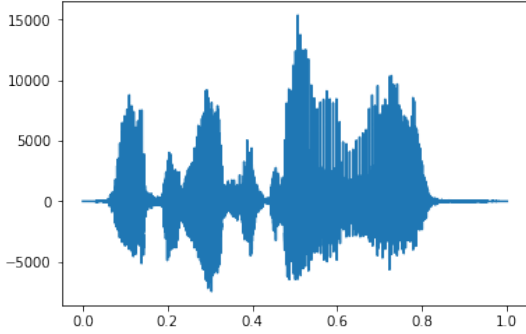


Figura 38: Señal de voz entrada

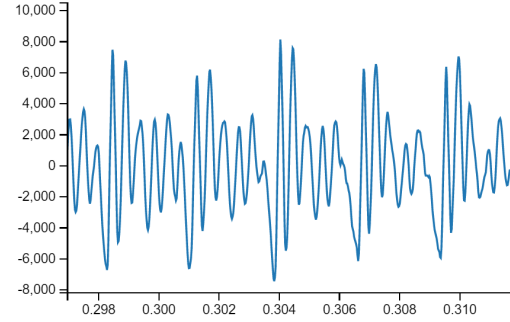


Figura 39: Señal de entrada con zoom

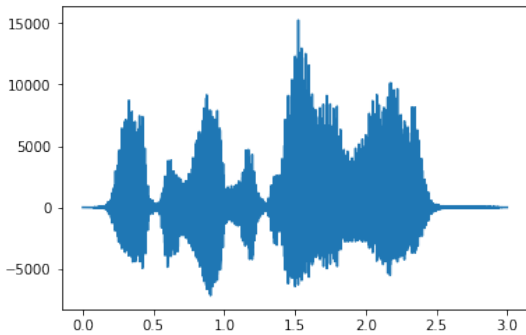


Figura 40: Señal a la salida con el triple de duración que la entrada

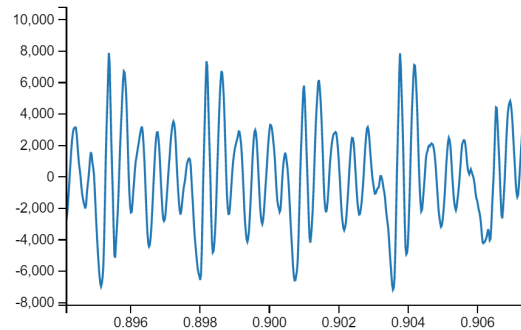


Figura 41: Señal de salida con zoom

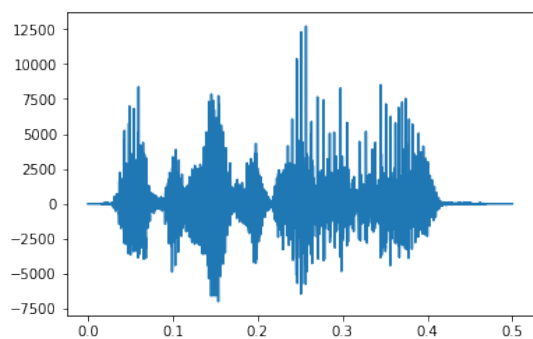


Figura 42: Señal a la salida con la mitad de duración que la entrada

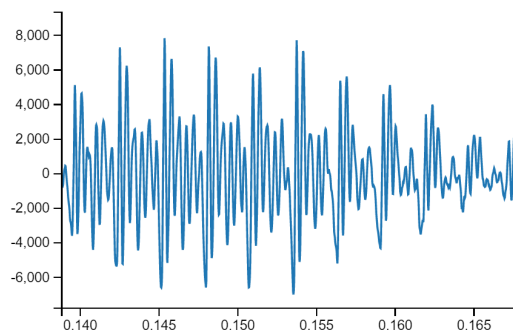


Figura 43: Señal de salida con zoom

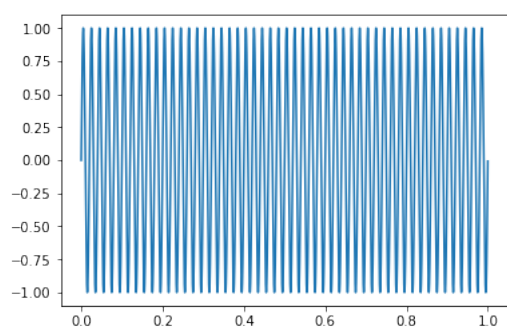


Figura 44: Señal senoidal de entrada (50Hz)

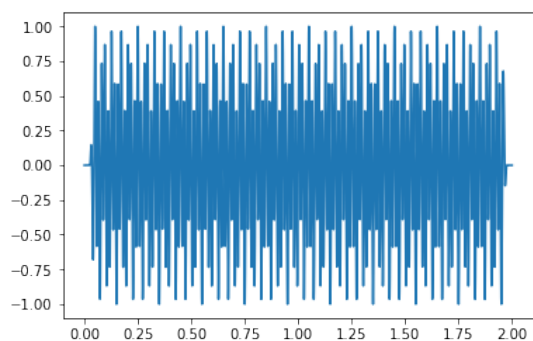


Figura 45: Señal a la salida con el doble de duración que la entrada

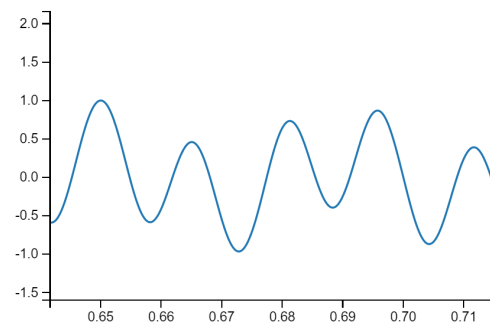


Figura 46: Señal de salida con zoom

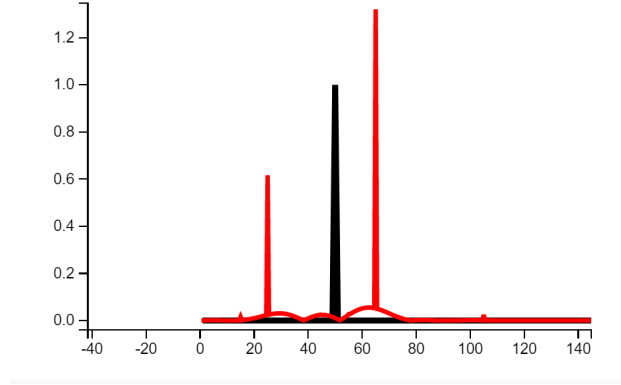


Figura 47: Espectro de las señales de entrada y salida. La negra es la entrada y la roja es la salida

Como se mencionó anteriormente, el algoritmo es mas fiel a la forma de la señal cuando esta no es de naturaleza armónica, pero introduce distorsiones cuando si lo es. En las fotos se puede observar la modulación presente en la salida.

Existen otros algoritmos más complejos que tienen como objetivo evitar este mal comportamiento de OLA en las señales con contenido armónico, uno de ellos es WSOLA.

Su funcionamiento es muy similar al de OLA, pero para que los segmentos sean lo más parecidos posible en la zona que se solapan y el resultado sea más satisfactorio, permite que los estos se puedan desplazar una cierta tolerancia temporal respecto de la posición original. Para encontrar este desplazamiento se busca la correlación máxima entre dos fragmentos adyacentes.

Matemáticamente se describe:

$$y(n) = \frac{\sum_{k=1}^{len(\sigma)} \omega(n - \gamma(k))x(n - \gamma(k) + \sigma(k) + \Delta_k)}{\sum_{k=1}^{len(\sigma)} \omega(n - \gamma(k))} \quad (22)$$

Es muy similar al OLA, salvo por los desplazamientos Δ_k .

Referencias

Time-scale Modification Algorithms for music audio signals by Jonathan Driedger

7. Espectrograma

7.1. Función para realización de espectrograma

Para la realización de los espectrogramas, se utilizó la función “spectrogram”, de la librería «scipy» de Python. La misma se basa en la implementación de transformadas de Fourier consecutivas sobre la señal a lo largo del tiempo. La sintaxis para su utilización es la siguiente:

```
scipy.signal.spectrogram(x, fs=1.0, window=('tukey', 0.25), nperseg=None, noverlap=None, nfft=None,
                        detrend='constant', return_onesided=True, scaling='density', axis=-1, mode='psd')
```

Donde los parámetros y sus efectos en detalle se describen a continuación.

- **x [array]**: arreglo con los valores que toma la señal en el tiempo (sobre los que se aplica la transformada de Fourier).
- **fs [float]**: frecuencia de sampleo de la señal $x(t)$. Por defecto normalizada: 1.0.
- **window [string ó tuple ó array]**: ventana temporal a utilizar. Influye tanto el tipo de ventana como el ancho de la misma.

Respecto al tipo de ventana, si se utiliza por ejemplo una ventana cuadrada, esta posee un corte abrupto. Si en dicha ventana no entra un número entero de períodos de la señal, se produce un corte abrupto.

Esto remite en la aparición de otras componentes de alta frecuencia que antes no había, lo que se conoce como «fuga espectral» (dado que la energía de los armónicos principales se “fuga” a los otros armónicos nuevos). Otras ventanas (como la Blackman-Harris) tienen un corte suave en los extremos (tienden a cero gradualmente), lo que minimiza la fuga espectral considerablemente.

El ancho de la ventana interviene en la resolución en tiempo y en frecuencia. Si la ventana es más ancha, se obtiene mayor resolución en frecuencia, dado que si la frecuencia de la señal sufre algún cambio en el tiempo (como en una señal FM), es posible captarlo con la ventana con mayor definición. Pero en el tiempo se pierde resolución dado que se sabe con menor precisión dónde ocurre exactamente el cambio de frecuencia. Con una ventana angosta, se gana resolución en el tiempo, pero en frecuencia se podría perder el cambio que antes se lograba captar en el tiempo con una ventana ancha, por lo que se vería una sola frecuencia en lugar de dos.

- **npersec [int]**: es el largo de cada segmento. Por defecto es «None», pero si la ventana se da en formato de «string» se considera 256, y si se da como «array» es el largo del mismo.
- **noverlap [int]**: es el número de puntos a solapar entre segmentos. Por defecto es «None», que es $npersec // 8$. Es decir, define la separación resultante entre ventanas.
- **nfft [int]**: es el largo de la FFT utilizada. Por defecto es «None», que determina el largo igual a “npersec”.
- **detrend [string ó function ó False]**: elimina la tendencia lineal a lo largo del eje temporal. Por defecto es “constant”. Si es de tipo string, se pasa directamente a la función «detrend» (también de scipy), y si es una función, se le pasa un segmento y lo devuelve ya con el desvío correspondiente. Si es «False», no se aplica ningún desvío. Para el caso de audio, no nos es de interés.
- **return_onesided [bool]**: si se asigna «True», se devuelve un espectro unilateral. Si es «False», el espectro será bilateral.
- **scaling ['density', 'spectrum']**: considerando a $x(t)$ en volts [V], se elige si procesar la densidad espectral de potencia [V^2/Hz] ó el espectro de potencia [V^2]. Por defecto se procesa la densidad espectral. Se simboliza como S_{xx} .
- **axes [int]**: es el eje a lo largo del cual se procesa el espectrograma. Por defecto es «-1» (es decir, el último utilizado).
- **mode [str]**: define que es lo que se espera que devuelva la función, entre [“psd”, “complex”, “magnitude”, “angle”, “phase”]. “psd” es la densidad espectral de potencia; con “complex” devuelve la STFT (Short-Time Fourier Transform) compleja; “magnitude” devuelve el valor absoluto de la STFT, y “angle” y “phase” el ángulo correspondiente complejo.

Los parámetros que devuelve son los siguientes:

- **f** [**ndarray**]: arreglo de dimensión “n” con las frecuencias de sampleo.
- **t** [**ndarray**]: arreglo de dimensión “n” con los segmentos de tiempo.
- S_{xx} [**ndarray**]: arreglo de dimensión “n” con el espectrograma de $x(t)$.

En referencia a las diferentes ventanas utilizables, además del ancho se debe considerar realizar el análisis con o sin overlap (solapamiento entre segmentos). En el caso más sencillo, no utilizar overlap podría conllevar pérdida de información: si se tiene un cambio de frecuencia de corta duración en comparación a la duración total del segmento, y se encuentra cerca del final de éste, al no utilizar segmentos solapados para la STFT podría perderse dicho evento. Para evitar dicha pérdida de información, se aplica la STFT a segmentos con un cierto grado de solapamiento. Se gana en no perder información, pero se requiere mayor tiempo de procesamiento, dado que al ser segmentos solapados, para cubrir la totalidad de la señal se necesita realizar más veces la transformación. El porcentaje de overlap se encuentra en general entre el 50 % y el 75 %, de acuerdo al paper «*On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform*» (1978, Fredric J. Harris, member IEEE).

7.2. Aplicación a escala de Sol Mayor (G3)

La tonalidad de *sol* mayor (en el sistema inglés abreviada como **G**) consiste en la escala mayor del sol y contiene las notas *sol*, *la*, *si*, *do*, *re*, *mi*, *fa* sostenido y *sol*. La notación **G3** corresponde a la pequeña octava, cuya frecuencia es aproximadamente 195.998Hz.

El espectrograma realizado se muestra a continuación.

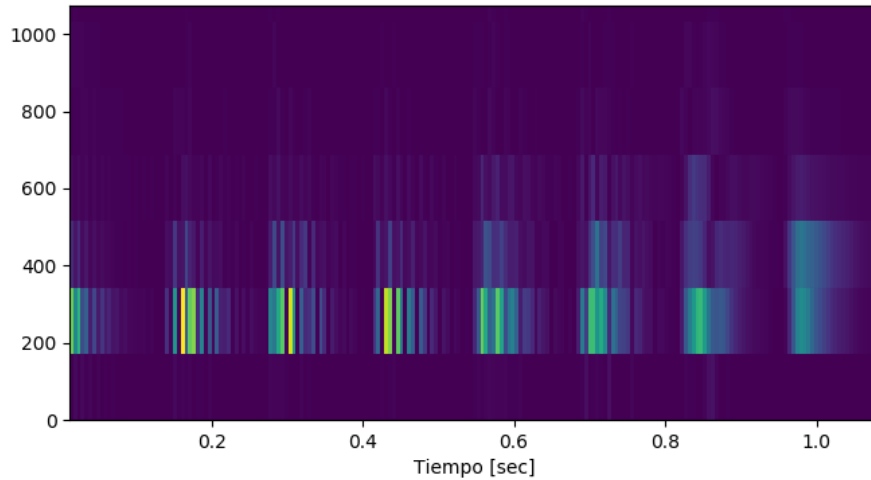


Figura 48: Espectrograma de Sol mayor

En el espectrograma obtenido sintetizando con un piano, resulta visible la frecuencia de la primera octava cercana a los 200Hz. Se utilizan 120ms por nota según lo sugerido, de manera tal que puede diferenciarse cada una por la escala creciente en frecuencia.

8. Efectos de audio

8.1. Reverbebrador

8.1.1. Implementación de eco simple

Se implementó un eco simple utilizando el sistema:

$$y(n) = x(n) + gx(n - M) \quad (23)$$

Se muestran a continuación los resultados con una señal de prueba.

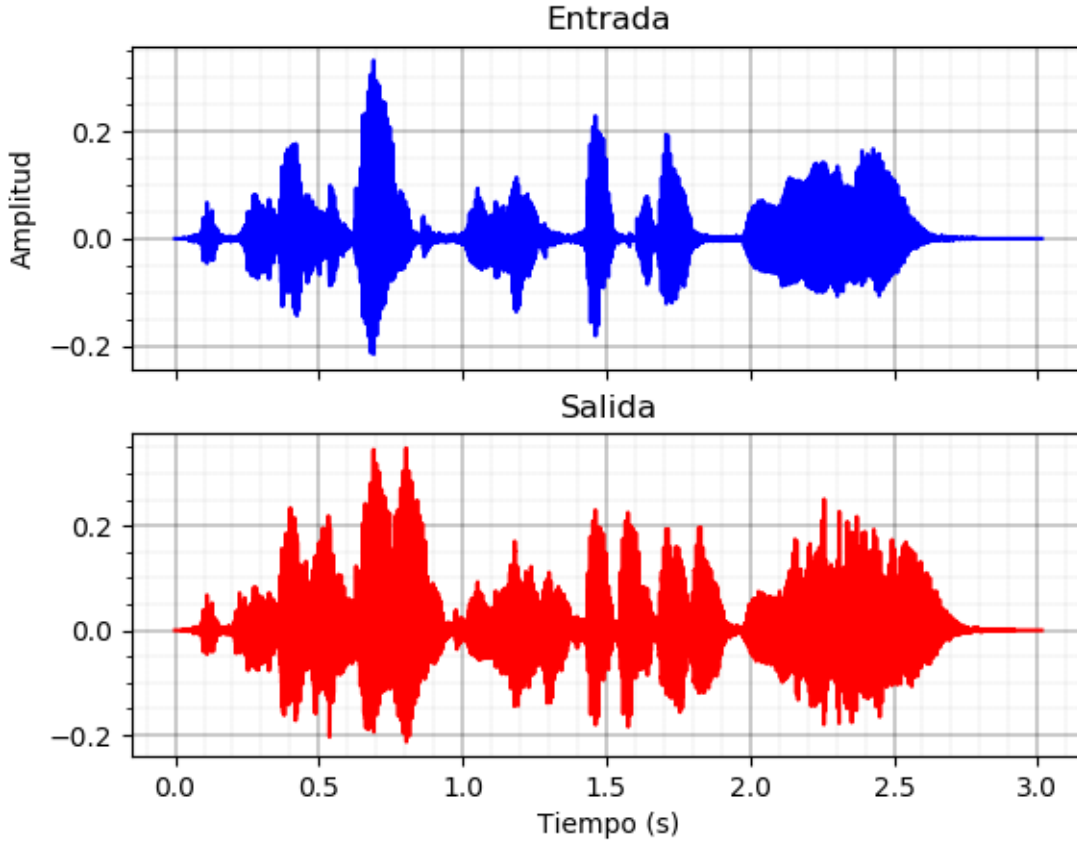


Figura 49: Resultados con $M = 5000$, $g = 0.999$

Se puede observar de los resultados intuitivamente como la señal de salida contiene repeticiones de la señal de entrada, y al escuchar el audio se pudo notar dicho efecto de eco. Fue necesario colocar un retraso muy grande ($M = 5000$) y una ganancia muy alta ($g = 0.999$) para que el efecto fuera notorio.

8.1.2. Implementación de reverberación plana

Se implementó una reverberación plana utilizando una ecuación de diferencias con feedback.

$$y(n) = x(n) + gy(n - M) \quad (24)$$

Se muestran a continuación los resultados con una señal de prueba.

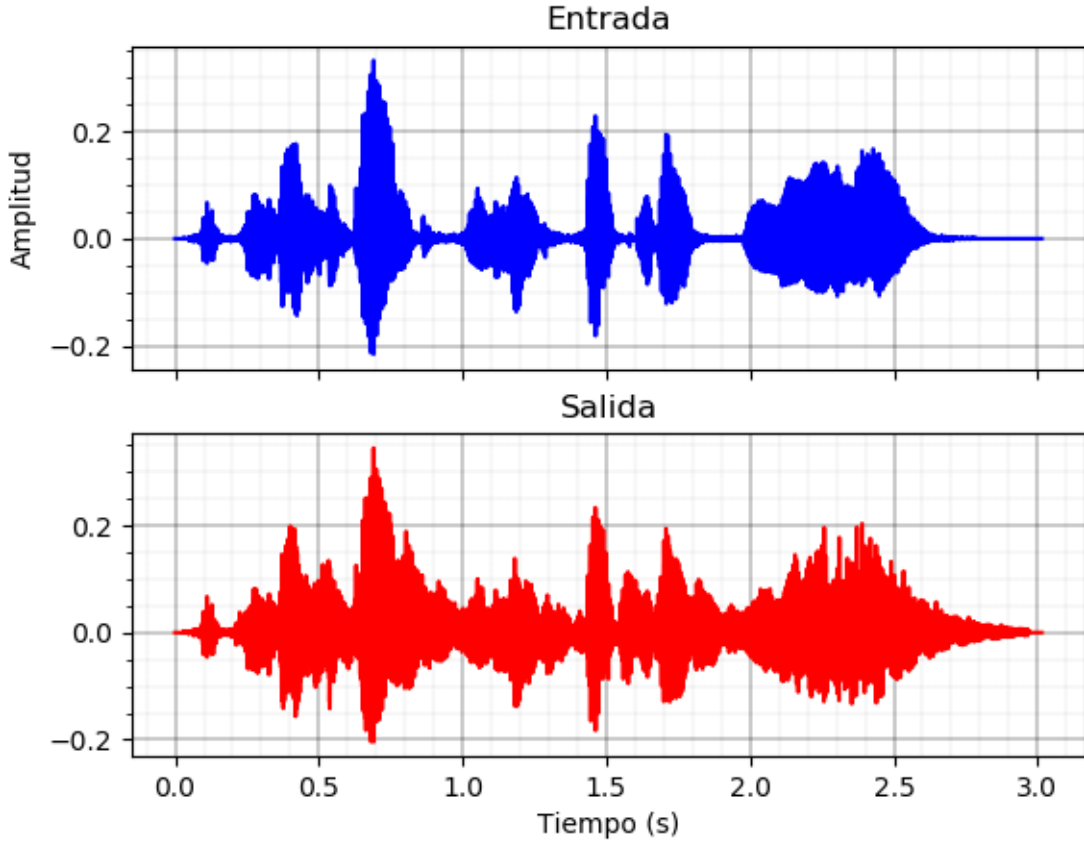


Figura 50: Resultados reverberación plana con $M = 500$, $g = 0.5$

Se necesitó disminuir fuertemente el valor de g para evitar que la salida saturara. Al tener realimentación (es decir, ser IIR) el sistema puede perder la estabilidad con facilidad.

8.1.3. Implementación de reverberación pasa bajos

Se le agrego un filtro pasabajo a la realimentación del sistema anterior. Se optó por un sencillo pasabajos similar al utilizado en el modelo Karplus Strong, de la forma $y(n) = \frac{x(n)+x(n-1)}{2}$. El sistema por lo tanto quedo descrito como:

$$y(n) = x(n) + \frac{1}{2}g(y(n-M) + y(n-M-1)) \quad (25)$$

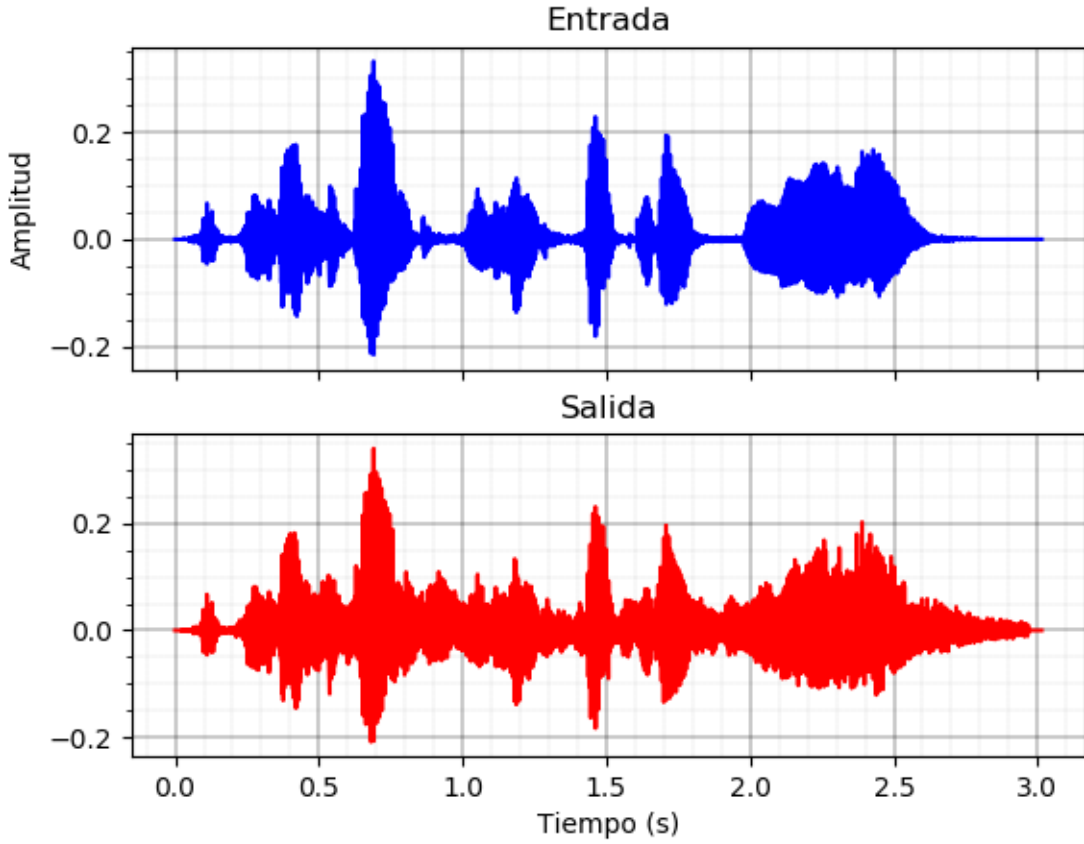


Figura 51: Resultados reverberación con pasabajos $M = 500$, $g = 0.5$

La señal fue similar al resultado del caso anterior con una sutil diferencia: el sonido se escuchó con un poco menos ruido. Esto se debe muy probablemente a que el filtro pasa bajos evitó la propagación de una frecuencia no deseada la cual no estaba presente en la señal original.

8.1.4. Implementación de reverberación completa

Se buscó estudiar un caso de reverberación completa, eligiendo el sistema denominado Schroeder. El mismo consiste de la conexión en paralelo de N reverberadores planos, continuados de M filtros tipo comb. Se utilizaron algunos criterios de los expresados en los apuntes de clase y también se “jugó” empíricamente para conseguir sonidos de reverberación interesantes.

A continuación se muestran los resultados obtenidos.

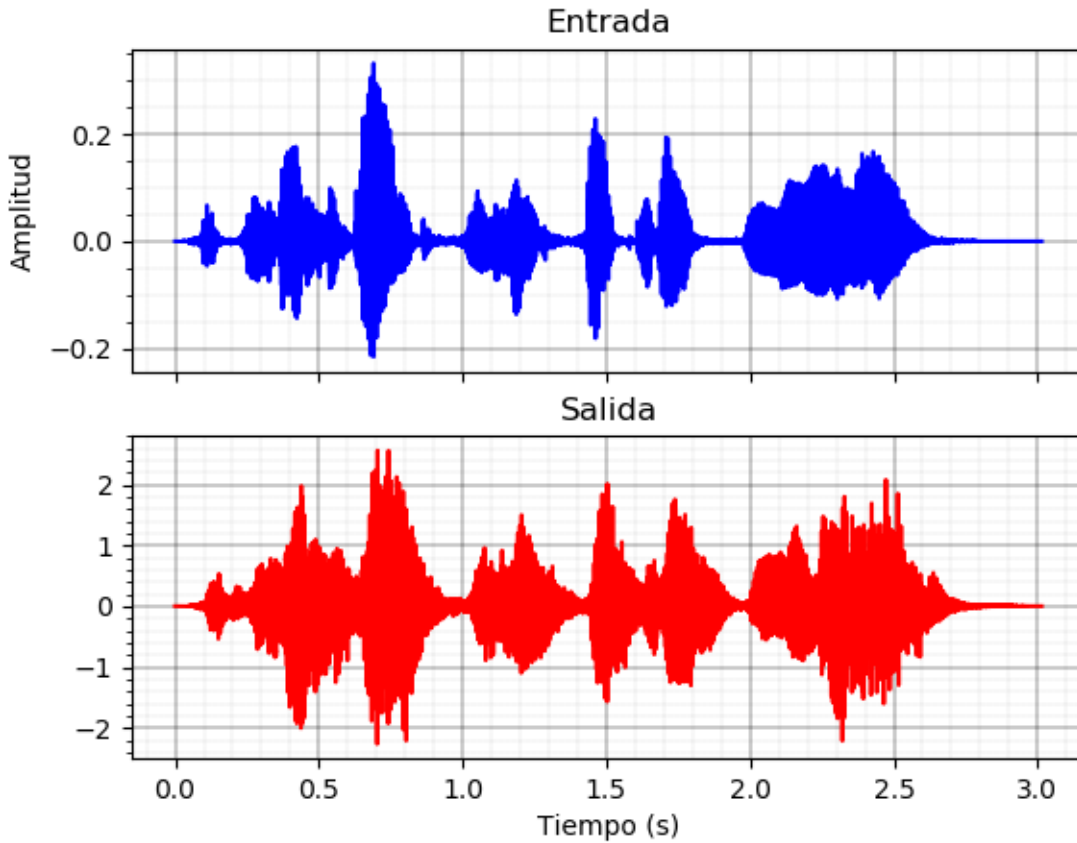


Figura 52: Resultados de un caso de reverberación completa, $N = 12$, $M = 2$, $a_{etapa1} = 0.999$ $a_{comb} = 0.5$

El sonido resultante fue muy interesante, siendo algo robotizado y obtenido tan solo ajustando parámetros del filtro. Esto da la pauta de que, con el reverberador completo se puede fácilmente ajustando sus parámetros conseguir tipos muy distintos de reverberación, sin necesidad de tener que convolucionar con una respuesta al impulso.

8.1.5. Implementación de reverberación por convolución

Se implementó una reverberación utilizando convolución con la respuesta al impulso característica de una fábrica. Se utilizó la ecuación en diferencias genérica siguiente:

$$y(n) = \sum_{i=k}^N h(k)x(n-k) \quad (26)$$

Debido a la complejidad algorítmica de la aplicación de la fórmula se debió limitar la longitud de la respuesta al impulso a solo 20000 muestras.

Se muestran a continuación los resultados con una señal de prueba.

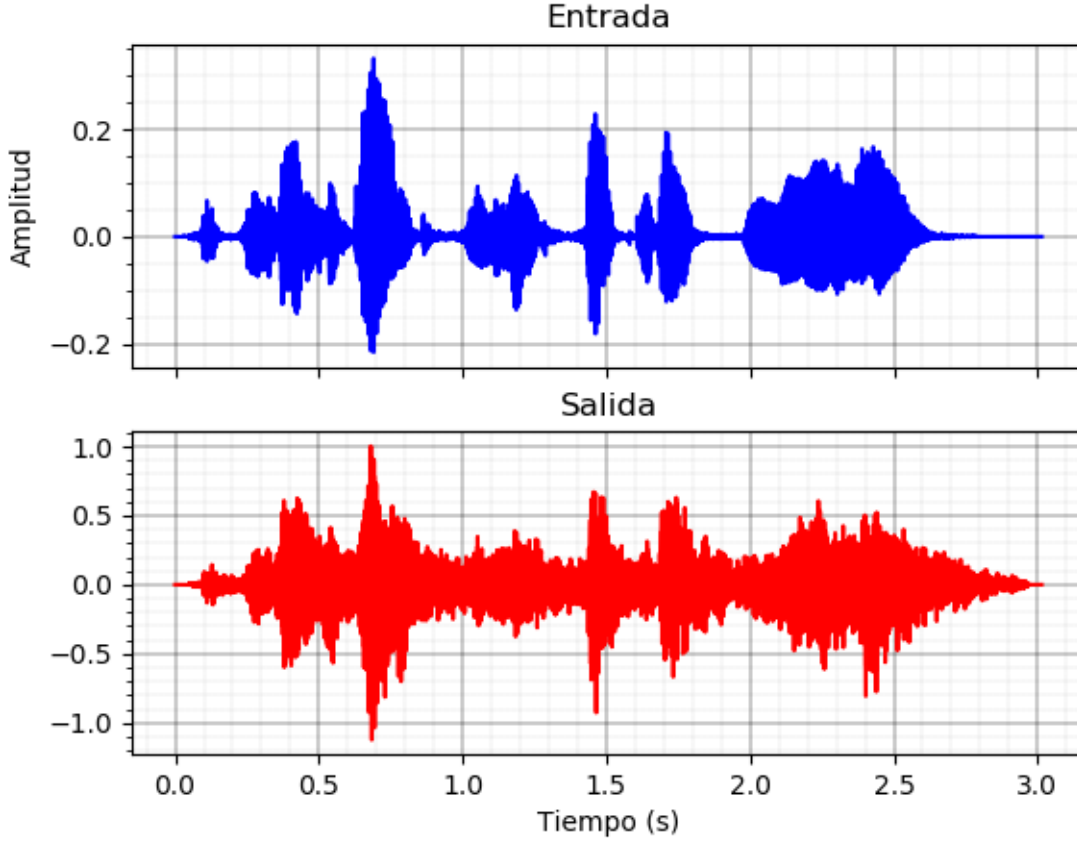


Figura 53: Resultados reverberación característica de una fábrica

Se observa como a diferencia de los casos anteriores la señal tiende mucho más a sostener sonido, esto se debe a que la respuesta al impulso es mucho más completa, y por lo tanto el sonido resultante persiste más tiempo. El sonido que se escuchó se correspondió con un eco muy realista, que podría ser el de una fábrica.

8.2. Efectos basados en delays variables

Para algunos efectos, es necesario que los delays presentes en la ecuación en diferencias que describe al sistema sean variables con el tiempo. Este es el caso tanto del flanger como del vibrato, entre otros.

Los efectos basados en delays variables pueden analizarse de manera general a partir de un filtro comb universal, cuyo diagrama de bloques se observa en la figura 54.

La variación del delay está dada por $M(n)$ (que debe ser $\geq 0 \forall n$ para que el sistema sea causal), mientras que los factores FF (feedforward), FB (feedback) y BL (blend) determinan qué tipo de efecto se producirá.

Las ecuaciones de este sistema son:

$$\begin{cases} x_h(n) = x(n) & +FB \cdot x_h(n - M(n)) \\ y(n) = BL \cdot x_h(n) & +FF \cdot x_h(n - M(n)) \end{cases} \quad (27)$$

Reemplazando la primera ecuación en la segunda, obtenemos que:

$$x_h(n - M(n)) = \left(\frac{1}{BL \cdot FB + FF} \right) \cdot y(n) - \left(\frac{BL}{BL \cdot FB + FF} \right) \cdot x(n)$$

Aplicando este resultado en la primera ecuación de 27, se llega a:

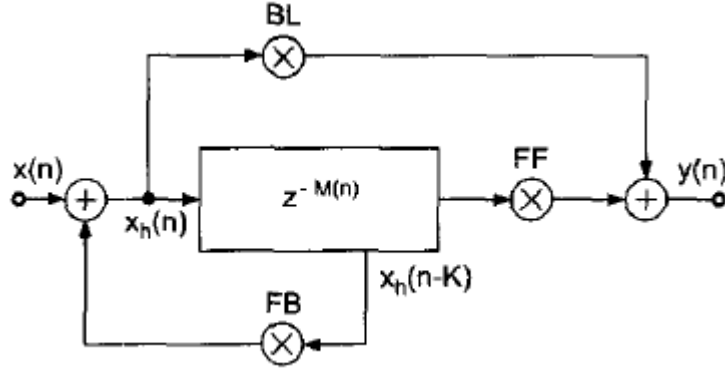


Figura 54: Diagrama de bloques del filtro comb universal

$$x_h(n) = \left(\frac{FF}{BL \cdot FB + FF} \right) \cdot x(n) + \left(\frac{FB}{BL \cdot FB + FF} \right) \cdot y(n) \quad (28)$$

Finalmente, reemplazando con esta expresión de $x_h(n)$ en la segunda ecuación del sistema 27, se obtiene que:

$$y(n) = BL \cdot x(n) + FF \cdot x(n - M(n)) + FB \cdot y(n - M(n)) \quad (29)$$

La transferencia en función del tiempo resulta entonces:

$$H(z, n) = \frac{BL \cdot z^{M(n)} + FF}{z^{M(n)} - FB} \quad (30)$$

Se ve pues que el filtro instantáneamente se comporta como un comb, si bien la cantidad y posición de los polos dependerán de la modulación $M(n)$ para cada n .

8.2.1. Vibrato

El efecto de vibrato consiste en introducir pequeñas variaciones periódicas en la frecuencia, resultando en una variación periódica de los tonos que se escuchan. Tomando como principio el efecto Doppler, podemos aplicar estos cambios en la frecuencia como cambios en el delay: de la misma manera que a medida que una ambulancia se aleja, se escucha más grave la sirena a causa de que aumenta el delay entre que se produce el sonido y lo escuchamos, si simulamos que el delay sube y baja periódicamente, la nota que se escucha subirá y bajará de la misma manera.

La ecuación que define este efecto es:

$$y(n) = x(n - M(n)) \quad (31)$$

Se ve, pues, que es un caso particular del comb universal, tomando $FF=1$, $FB=0$ y $BL=0$. La transferencia es entonces:

$$H(z, n) = z^{-M(n)} \quad (32)$$

Por lo tanto, el sistema tiene $M(n)$ polos en el origen y no tiene ceros de transmisión.

En la ecuación 31, $M(n)$ representa el delay variable con el tiempo, y se implementa con una senoidal que toma valores naturales entre 0 y un máximo K :

$$M(n) = \left\lfloor \frac{K}{2} \cdot (1 + \sin(2\pi f_0 \cdot nT)) \right\rfloor \quad (33)$$

Los parámetros que caracterizan al vibrato son pues:

$$\begin{cases} f_0 = \text{frecuencia de modulación} \\ K = \text{profundidad de modulación} \end{cases} \quad (34)$$

De acuerdo a lo pautado en la bibliografía sugerida, se tomó $f_0 = 5\text{Hz}$ y $K = 1\text{ms}$.

8.2.2. Flanger

El flanger implementado sigue el modelo sugerido por el libro, en el cual los parámetros son $FB = BL = FF = 0.7$, con $M(n)$ senoidal. La transferencia es pues:

$$H(z, n) = \frac{0.7 \cdot (z^{M(n)} + 1)}{z^{M(n)} - 0.7} \quad (35)$$

La definición de $M(n)$ que usaremos es idéntica a la del vibrato (ecuación 33), es decir, una senoidal que varía entre 0 y un máximo. Los valores óptimos se encontraron en $f_0 = 1\text{Hz}$ y $K=2\text{ms}$.

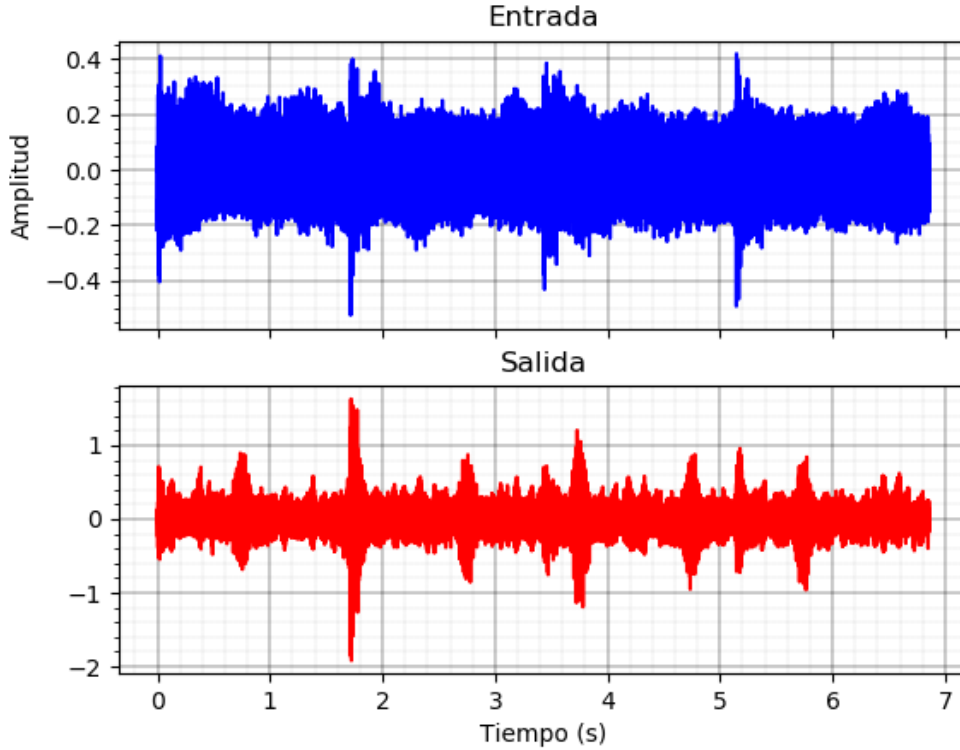


Figura 55: Salida obtenida con el flanger

Se observa en la figura 55 que la salida presenta picos a la misma frecuencia de modulación (una vez por segundo), pero por fuera de los mismos la forma de la función se conserva.

8.3. Robotización

Para este efecto, se siguieron los pasos pautados por la cátedra. Los anchos de ventana que produjeron resultados más satisfactorios fueron $N = 1024$ y $N = 512$, con $f_s = 44.1\text{kHz}$.

Se utilizó la ventana de Hanning con 50 % de overlap. Esta ventana utiliza como $w(n)$ la función de Hann, que para ancho de ventana $N \in \mathbb{N}$ se define como:

$$w(n) = \frac{1}{2} \cdot \left(1 - \cos \left(\frac{2\pi n}{N-1} \right) \right), \quad 0 \leq n < N \quad (36)$$

Con esta definición, no es cierto que $w(n - N/2) + w(n + N/2) = 1, \forall n, \forall N$, o siquiera constante, como se propone en la consigna. Esto sí ocurriría si se tomase ancho de ventana $N - 1$ manteniendo la misma definición de $w(n)$, pero se pierde en este caso simetría, y la función no se anula en los bordes. Se obtuvieron resultados más satisfactorios con la versión no COLA (constant overlap-add), e incluso empíricamente se verifica que para los anchos de ventana utilizados, si bien la suma no es 1 ni constante, es consistentemente mayor a 0.95 en todo tiempo.

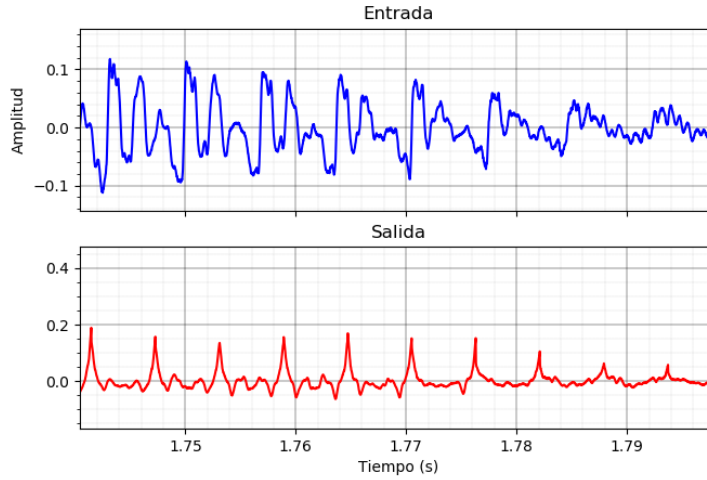


Figura 56: Salida obtenida con la robotización, con ancho de ventana $N = 512$ muestras

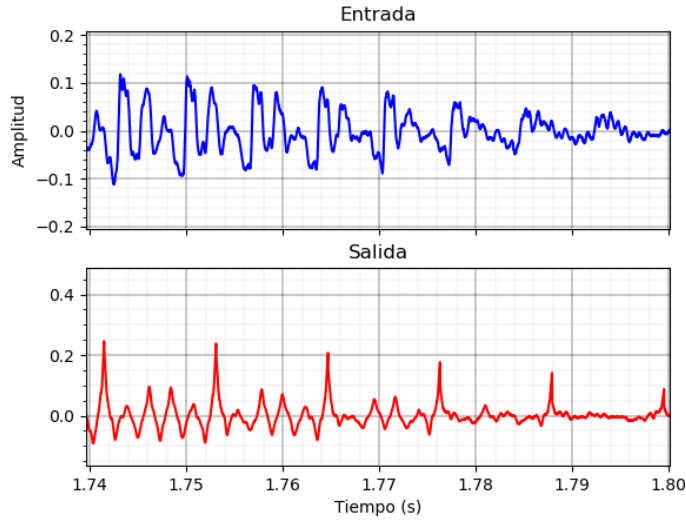


Figura 57: Salida obtenida con la robotización, con ancho de ventana $N = 1024$ muestras

En cuanto a los resultados, en las figuras 56 y 57 se observa la salida con los dos anchos de ventana con mejor performance. A grandes rasgos, se ve que cada ventana (identificable por simple inspección, siguiendo el patrón de los picos de la salida, que es el doble de rápido para la ventana más pequeña) posee aproximadamente la misma forma, y sólo se modula en amplitud, siguiendo la envolvente de la entrada. Esto es consistente con el objetivo del efecto de eliminar la información de los cambios de pitch.

9. Rotary Loudspeaker Effect

En este efecto de sonido lo que llega a las orejas del usuario es alterado por el efecto Doppler, la característica direccional de los speakers y efectos de fase debido a turbulencia aérea. El efecto doppler eleva y descende el tono dependiendo de la velocidad de rotación. La característica direccional de la bocina opuesta desenvuelve una variación de intensidad en los oídos del usuario. Tanto la variación del tono como la variación de intensidad son ambas modificadas por un speaker A y en dirección opuesta por el speaker B.

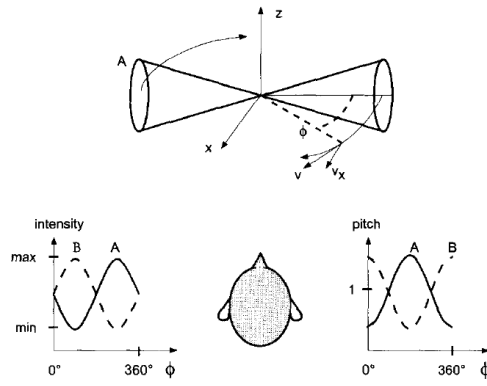


Figura 58: Esquema de Rotary Loudspeaker

La simulación del efecto doppler de dos bocinas opuestas es realizada mediante el uso de dos líneas de retardo moduladas con señales que están desfasadas 180 grados en configuración vibrato. Un sonido direccional característico similar al de speakers rotando puede ser logrado mediante modulación de amplitud de la señal de salida de las dos líneas de retardo. La modulación está sincronizada con el retardo de modulación de forma tal que la bocina trasera que se mueve tiene menor tono y amplitud decreciente. En el punto de retorno el tono es inalterado y la amplitud es mínima. Intuitivamente, mientras la señal del canal izquierdo sube, la del derecho baja y en consecuencia el oyente percibe un movimiento giratorio.

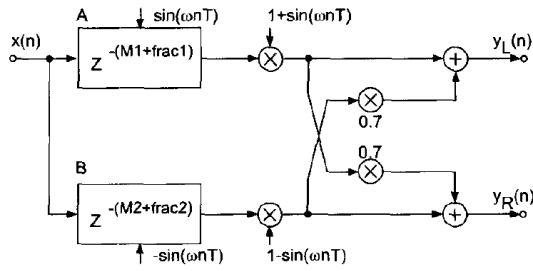


Figura 59: Simulación de Rotary Loudspeaker