22.05 - Análisis de Señales y Sistemas Digitales

Instituto Tecnológico de Buenos Aires

Trabajo Práctico Número 2

Síntesis de Instrumentos Musicales

Grupo 1

AUTORES:

Federico Tondi (59341)

Franco Moriconi (58495)

Alan Vekselman (59378)

Carola Pedrosa (59059)

Franco Scapolla (58465)

PROFESORES:

Daniel Jacoby

CIUDAD AUTÓNOMA DE BUENOS AIRES

Mayo 2021

Contenido

1. Sintesis aditiva

1.1. Marco teórico

La síntesis aditiva es una técnica de generación de sonidos de forma artificial. Consiste en, a partir de una suma de senoidales, replicar la estructura de armónicos de diferentes instrumentos u otras fuentes de sonidos reales. En el fondo, esto no es más que una aplicación de series de fourier que permiten reconstruir cualquier señal periódica a partir de una suma de senos y cosenos con diferentes frecuencias y amplitudes. Incluso a pesar de que el sonido producido por un instrumento no es periódico, una serie de fourier con ciertas consideraciones podrá simularlo exitosamente.

Exiten múltiples parametrizaciones para trabajar con síntesis aditiva. En particular, en el marco de este trabajo práctico se trabajará primero con ADSR (Attack, Decay, Sustain, Release). Esta técnica de parametrización se basa en la idea de que cada sonido puede descomponerse en las cuatro etapas mencionadas:

1. *Attack:* Comprende el comienzo del sonido hasta que llega a su amplitud máxima.
2. *Decay:* Comprende desde que comienza a caer desde la amplitud máxima hasta que llega a la etapa de *sustain*.
3. *Sustain:* Etapa donde la amplitud se mantiene aproximadamente constante, con una caida determinada.
4. *Release:* Es la caída de la amplitud del parcial hasta cero, tras, por ejemplo, soltar la tecla de un piano.

En un gráfico de amplitud en función del tiempo, esto se ve representado gráficamente cómo:

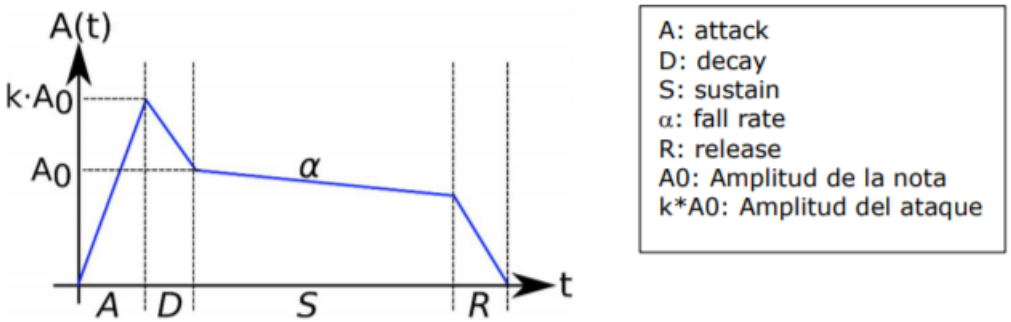

FIGURA 1.1: ADSR genérico representado gráficamente

Entonces, podemos afirmar que el ADSR simula con mayor precisión el comportamiento de un sonido en el tiempo, incorporando a la simulación la variación en amplitud del mismo en el tiempo, ya que no es realista suponer que un sonido tiene amplitud constante desde su nacimiento hasta su desaparición.

1.2. Simulación de instrumentos

En esta sección, se seleccionarán cuatro instrumentos diferentes y se obtendrán sus parámetros experimentalmente, con software de análisis espectral. Luego, con estos parámetros, se producirán muestras artificialmente de alguno de ellos y se utilizaran estas notas para tocar un tema musical, analizando luego los resultados y estudiando posibles variaciones o correcciones a la simulación de las notas y su impacto en la calidad del audio resultante.

1.2.1. Simulación de piano

Para complementar las demás secciones de este informe, donde se incorporaron guitarras, arpones y tambores, se decidió incorporar un piano simulado con síntesis aditiva. El método de obtención fue a través del espectrómetro del programa de edición de sonido *Audacity*, en el cual se analizó muestras reales. Se tomó como primer caso la nota "La.ª 440Hz, y su espectro resultante fue:

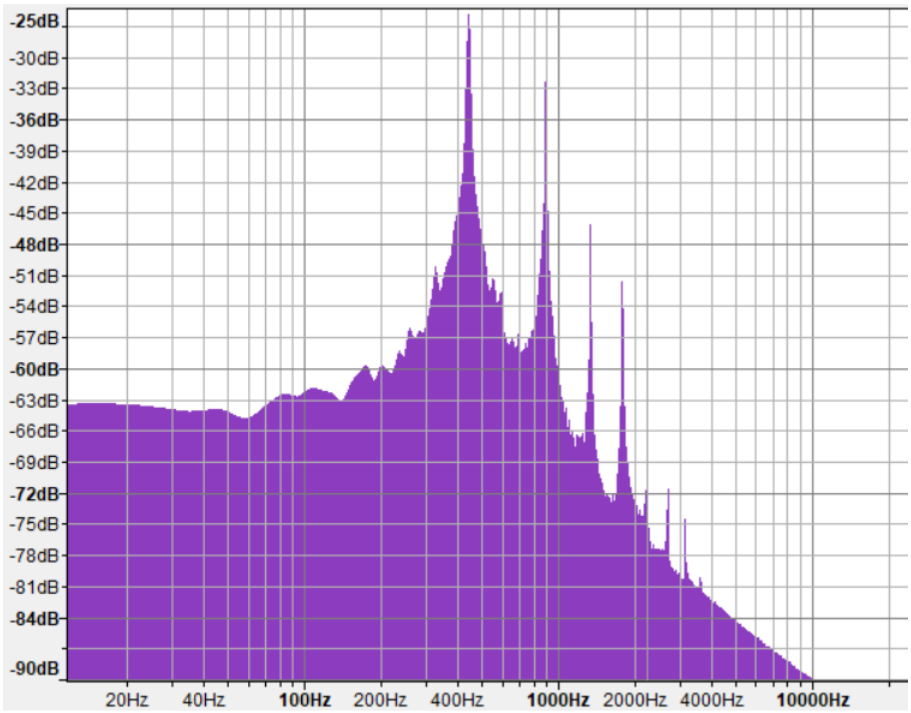

FIGURA 1.2: Espectro en frecuencia de una nota de un piano

Luego, se reconstruyó este espectro a partir de senoidales cuyas frecuencias corresponden a la del fundamental de la nota analizada y sus armónicos. El espectro resultante se ve a continuación:

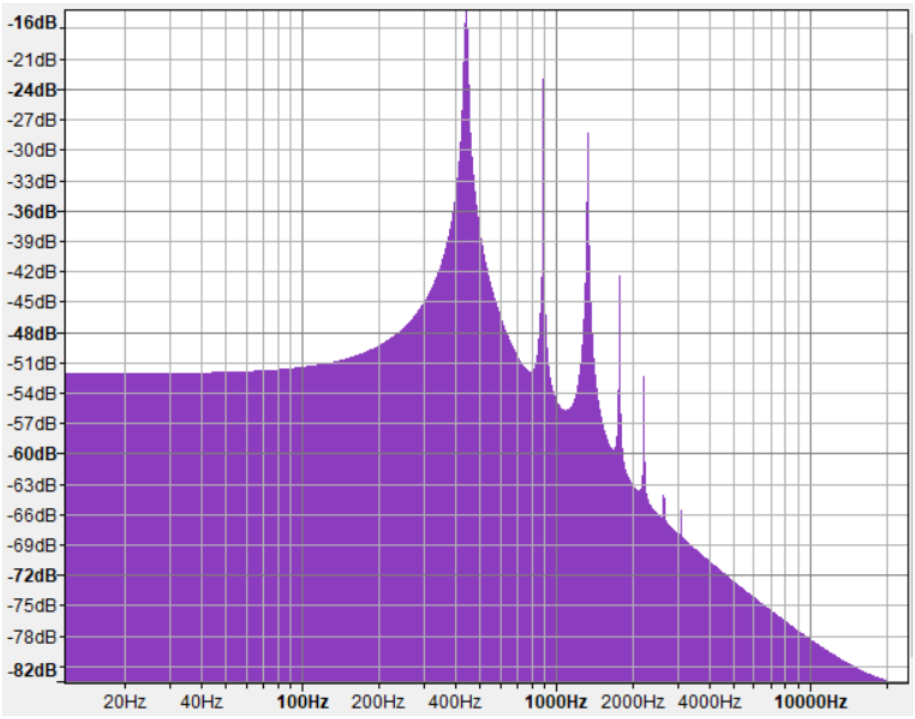

FIGURA 1.3: Espectro en frecuencia simulado de una nota de un piano

Es claro que el espectro, como lo presenta Audacity, es mucho más suave que el real ya que son senoidales puras sumadas.

La nota en el tiempo, como se ve en la figura 1.4, nos da un indicio de como construir su ADSR. El ataque deberá ser rápido, prácticamente instantáneo. El decay es algo más lento, y luego el sustain compone la mayor parte de la nota. El release es casi imperceptible, pues la nota en cuestión suena hasta que se extingue naturalmente. Es importante destacar que esta señal tiene una duración total de 3 a 4 segundos, siendo en su gran mayoría el sustain. En la simulación, se sumó una serie de senoidales con las amplitudes obtenidas anteriormente y se multiplicó el resultado por la señal envolvente, generando así la señal vista en la figura 1.5.

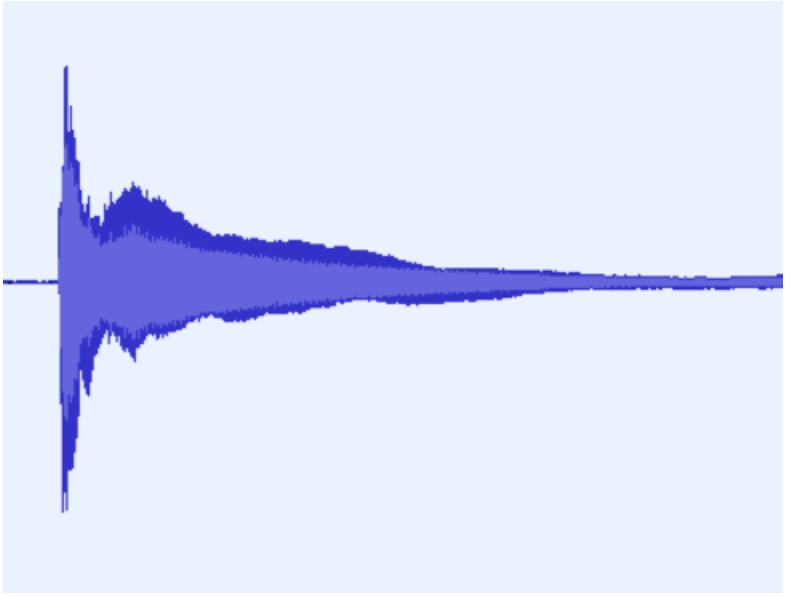

FIGURA 1.4: Gráfico del sonido de un piano en el tiempo

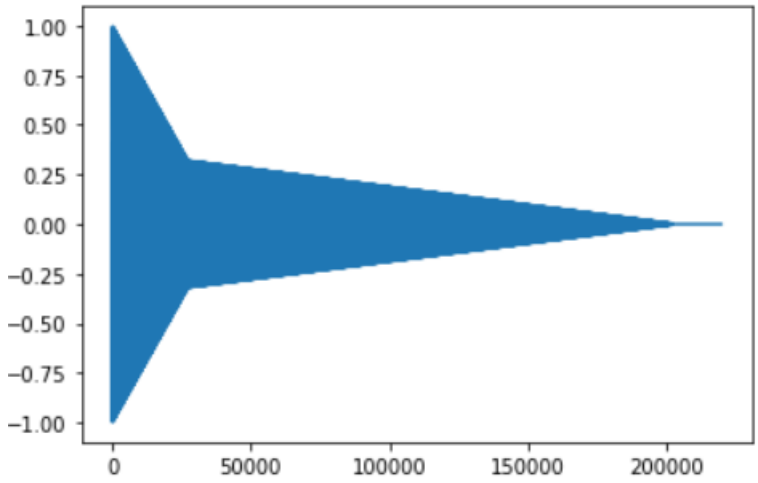

FIGURA 1.5: Suma de fundamental y armónicos de la nota estudiada con la envolvente ya aplicada

Esta señal se mejoró agregando una ligera desviación aleatoria en las frecuencias de los armónicos, lo cual produce un sonido algo más natural. La diferencia está en la "vibración"que se genera al variar ligeramente la amplitud de la señal resultante, en lugar de un movimiento lineal y artificial como se vió en la figura 1.5. La onda generada resultante, claramente mucho más similar a una real, se logró con una variación aleatoria de $\pm 5Hz$ para los armónicos del piano sintetizado. Más que esto producía una señal deformada en demasía, dejando de parecerse a la real.

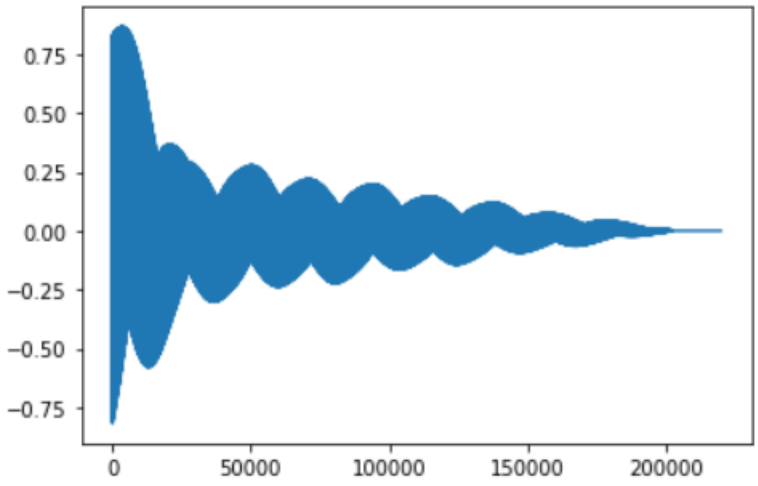

FIGURA 1.6: Señal producida con ligera variación aleatoria en los armónicos

Adicionalmente, se observó experimentalmente que diferentes frecuencias duraban más (Si eran más graves) o menos (si eran más agudas) que la muestra original, con lo cual su ADSR cambiaría. Por ejemplo, una nota aguda duraba muchísimo menos que una nota "La.ª 440Hz. Para esto, se aplicó un coeficiente de normalización en el código, de forma empírica, que corrigió de forma aceptable la duración de los parámetros acorde a la frecuencia fundamental Esto podría fácilmente extenderse a cada parcial de forma individual, modificando su envolvente para acortar su duración como se sugiere en la consigna. No obstante, no se implementó en la versión final como consecuencia del impacto del rendimiento del programa al repetir el proceso para cada parcial en lugar de una sola vez para cada nota.

A continuación, se presenta una tabla con parámetros ADSR que se hallaron empíricamente a través de muestras para cuatro instrumentos diferentes:

Instrumento	A	D	S	α	R
Piano	0.001	0.2	0.25	0.125	0.2
Trompeta	0.218	0	1	0	0.133
Flauta	0.08	0	1	0	0.08
Bajo	0.02	0.1	0.25	0.25	0.01

TABLA 1.1: Tabla ADSR múltiples instrumentos

El que se implementó en el programa final fue, únicamente, el piano, que tuvo resultados muy positivos ante los diferentes samples. Como se mencionó antes, para cada instrumento se obtuvo los parámetros experimentalmente. Resulta curioso que los instrumentos de viento se sostienen infinitamente siempre y cuando se soplen en ellos. Como su amplitud máxima es este nivel al que se sostiene, no tienen decay y su fall rate es nulo. En el caso del bajo, el ataque y decay se producen tras soltar la cuerda. Luego, el sonido se extinguirá naturalmente hasta llegar a cero. El bajista puede, sin embargo, detener el sonido de forma abrupta si detiene la vibración apretando la cuerda, con lo cual podría considerarse al release como este acto de detener la nota y será de

forma prácticamente instantánea.

2. Síntesis Basada en Muestras

Tal como su nombre lo indica, la síntesis basada en muestras se basa en la utilización de muestras previamente grabadas para obtener sonidos de distinto tono y duración. El primer sintetizador por muestras se vio en el Fairlight Computer Music Instrument, lanzado en 1979, con un precio de 1200 libras, en cuyo interior descansaba un sistema que incorporaba un DSP y que permitía al usuario, a través de un teclado, visualizar distintos tipos de sonidos y modificarlos para realizar la síntesis por muestras. Naturalmente, con el avance de la tecnología, la reducción en el costo de la memoria, y el desarrollo de nuevos algoritmos de procesamiento de señales, los sintetizadores basados en muestras pasaron a ser más asequibles y mucho más precisos.

Para ilustrar el problema principal detrás de la síntesis basada en muestras, considérese la siguiente situación de la vida real: se graba un comercial, y por temas legales se deben agregar algunas aclaraciones al final del mismo. Debido a que el tiempo al aire es altamente costoso, se puede en un principio suponer que una forma de incluir esta nueva aclaración sin cambiar el tiempo total de aire es grabar el mensaje a velocidad normal, y luego reproducirlo a velocidad más rápida. Por ejemplo, cuatro veces más rápido. Una forma de realizar este incremento de velocidad, por ejemplo, sería tomar la señal original, a la cual se la puede pensar en el dominio digital como un arreglo de valores o muestras de la señal analógica, y quedarse sólo con los valores en los índices múltiplos de cuatro. De esta forma se obtiene un arreglo que se reproducirá cuatro veces más rápido. A continuación se presenta un gráfico del contenido espectral a distintas frecuencias de una grabación de un discurso a velocidad normal, y otro de la misma grabación pero reproducida cuatro veces más rápido a través del método explicado arriba.

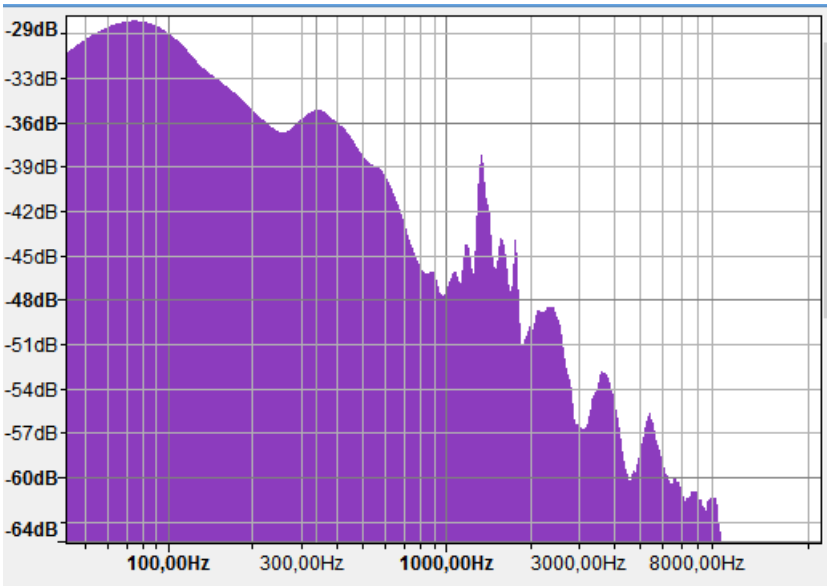

FIGURA 2.1: Contenido espectral de un discurso reproducido a velocidad normal

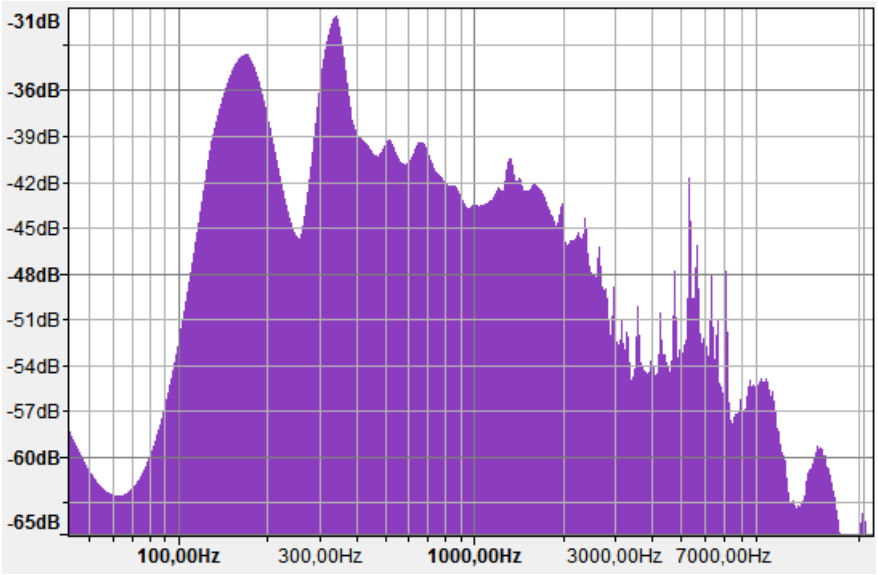

FIGURA 2.2: Contenido espectral de un discurso a velocidad cuatro veces mayor

Se observa que mientras en el primer caso la mayor parte del contenido espectral se encuentra entre los 80 y 300 Hz, en la muestra reproducida a mayor velocidad el contenido espectral se trasladó hacia frecuencias más altas, lo cual ocasiona que al reproducir el mismo el tono sea más agudo que el original. Es decir, este método no permite hacer un ajuste de la duración de una grabación sin alterar su tono. Esto era de esperarse pues existe una relación entre el dominio temporal y el de la frecuencia, y al realizar el método anterior lo que se está haciendo es resampleando la muestra, cambiando por lo tanto la frecuencia de muestreo y afectando al contenido espectral. El objetivo de este trabajo, por lo tanto, es estudiar e implementar algún algoritmo que permita sintetizar tracks de audio a partir de muestras, cambiando su duración o tono de forma independiente.

2.1. Phase Vocoder

Hay distintos algoritmos que permiten modificar la duración y el tono de una muestra de audio de forma independiente. Algunos de ellos, como el Time-Domain Pitch Synchronous Overlap-Add (PSOLA por sus siglas en inglés) trabajan directamente en el dominio del tiempo, mientras otros, como el Phase Vocoder, lo hacen en el dominio de la frecuencia. Para este trabajo se decidió implementar este último. A continuación se presenta una breve descripción del mismo.

Lo que se busca en un principio es realizar el estiramiento o compresión en tiempo de la señal (TMS por Time Scale Modification) sin modificar su contenido espectral o tono o pitch. Una vez logrado esto, el cambio de tono se logra a través de primero un estiramiento sin modificar el tono, y luego un resampleo que modifica el tono al valor deseado y devuelve la señal a su longitud original. La idea central detrás del Phase Vocoder es la Transformada de Fourier en Tiempo Corto o STFT (Short Time Fourier Transform). Un esquema de los distintos procesos que componen al algoritmo se presentan en la figura 2.3.

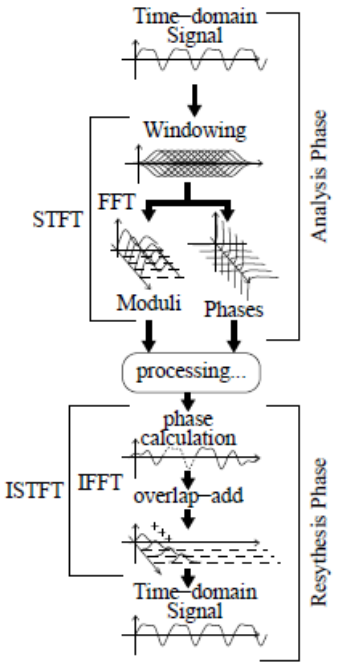

FIGURA 2.3: Esquema del algoritmo de Phase Vocoder

Se identifican tres etapas: una primera de análisis, en donde se transforma la señal en el dominio del tiempo al dominio de la frecuencia usando la STFT; una de procesamiento donde se pueden introducir distintos efectos; una de síntesis, donde se utiliza la Inverse Short Time Fourier Transform (ISTFT) para pasar los distintos bloques del dominio de la frecuencia al del tiempo, y luego la técnica de overlap and add para reconstruir la señal en el dominio del tiempo.

En el caso particular del TSM, la idea es que, al realizar la STFT durante el análisis, cada una de las ventanas sobre la que se toma la FFT capture el contenido espectral local asociado a esa ventana. Luego al realizar la transformación inversa durante el proceso de síntesis, se cambia el espaciado entre ventanas con respecto al análisis, de forma que las muestras quedan más estiradas o comprimidas en el tiempo pero conserven la tonalidad local de cada parte de forma que el tono de la pieza no se modifica. En general, tanto en análisis como en síntesis las ventanas se superponen. Si el largo de la ventana sobre la que se toma la FFT es de N muestras, se define el Hop Size como la cantidad de puntos o muestras que se corren hacia la derecha antes de tomar la nueva ventana para computar la nueva FFT; así, si Hop Size es N/4, luego hay una superposición del 75 %. El grado de superposición influirá en la resolución temporal (coherencia horizontal) y en la resolución espectral (coherencia vertical). De esta forma, si se tiene que H_a es el Hop Size en la fase de análisis y H_s es el Hop Size en la fase de síntesis, se define el factor de escalamiento temporal como $= \frac{H_s}{H_a}$. Naturalmente en este procedimiento surge el problema de que, en un principio, nada garantiza que al reacomodar las ventanas durante el proceso de síntesis no aparezcan nuevas discontinuidades de fase en los bordes de las mismas, que se trasladen en cambias indeseados en el tono de la señal reconstruida, como se ilustra en la figura 2.4 y 2.5. Se busca, entonces, realizar una correción de fase antes de aplicar la ISTFT.

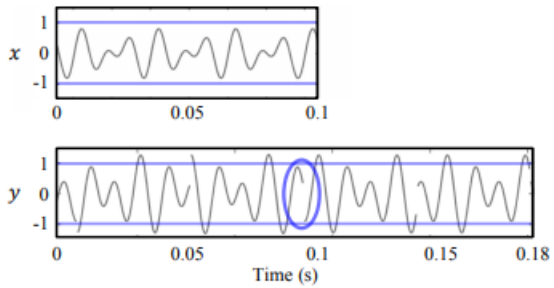

FIGURA 2.4: Interferencia de ondas en la señal reconstruida debido a discontinuidades en la fase

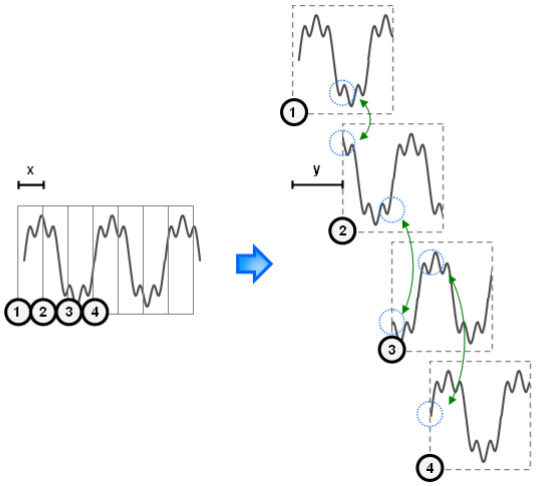

FIGURA 2.5: Interferencia de ondas en la señal reconstruida debido a discontinuidades en la fase

2.2. Implementación del algoritmo

2.2.1. STFT

A continuación se busca hacer un análisis algo más detallado del algoritmo en sí y de cómo realizar la corrección de fase en el dominio de la frecuencia antes de transformar para lograr un resultado final sin cambios de tono. Sea $x(t)$ la señal original en el dominio del tiempo, y sea $w(t-\tau)$ la ventana que determina el bloque sobre el que se aplica la FFT. De esta forma, se tendrá que la STFT de x en el tiempo τ estará dada por:

$$X(w,\tau) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-j\omega t} \tag{2.1}$$

En particular, si ahora se considera la señal muestreada $x(nT)$, se tendrá que la STFT será:

$$X(m,k) = \sum_{r=\frac{-N}{2}}^{\frac{N}{2}-1} x_m(r)w(r)e^{-j\frac{2\pi}{N}kr} \tag{2.2}$$

Donde m es el índice que indica qué ventana temporal se está transformando, y k entre 0 y $N-1$ es el índice del bin de frecuencia. Es decir, como se mencionó antes, para cada bloque en el tiempo dado por el índice m, se obtiene una $DFT(FFT)$ de N puntos. Si H_a es la cantidad de muestras hacia la derecha que se corre antes de calcular la DFT del nuevo bloque (es decir, la cantidad de muestras que se corren al aumentar en uno a m) y se conoce la frecuencia de muestreo f_s de x, se puede establecer una relación entre m y el tiempo físico según:

$$T(m) = \frac{mH_a}{f_s} \tag{2.3}$$

Donde $T(m)$ está en segundos.

De forma similar, la frecuencia física asociada al bin k estará dada por:

$$F(k) = \frac{kf_s}{N} \tag{2.4}$$

Donde $F(k)$ está en Hertz.

Por lo tanto, el resultado de aplicar la STFT sobre la señal muestreada dará un conjunto de números $X(m,k)$, llamados bins de frecuencia-tiempo, donde $X(m,k)$ denota el coeficiente de Fourier k-ésimo de la ventana de análisis m. Al ser una magnitud compleja, se tendrá

$$X(m,k) = |X(m,k)|e^{j2\pi\varphi(m,k)} \tag{2.5}$$

En particular, las magnitudes $|X(m,k)|$ puede usarse para construir el espectograma de la señal. El mismo será un gráfico donde el eje x estará dado por los índices m de las ventanas de análisis, y el eje y por las frecuencias correspondientes a los k bines, indicando con colores el valor de $|X(m,k)|$ en cada punto.

2.2.2. Correción de fase

Una vez aplicada la STFT durante la etapa de análisis, se obtendrán los coeficientes $X(m,k)$ mencionados en la sección anterior. Para poder hacer la modificación en tiempo sin afectar el tono, debe aplicarse una corrección de fase de forma que al cambiar el intercalado de las ventanas se evite en la mayor medida posible las discontinuidades en la fase.

Puede interpretarse a $X(m,k)$ como una componente senoidal de amplitud $|X(m,k)|$ y fase $\varphi(m,k)$ que contribuye a la m ésima ventana de análisis de x. Sin embargo, se sabe que en la DFT las frecuencias son discretas y corresponden a los valores $k\frac{f_s}{N}$, de donde en un principio la resolución de la DFT no es suficiente para asignar una frecuencia precisa a esta componente senoidal. La idea, entonces, es explotar la información de fase disponible para mejorar la estimación

de la frecuencia de esta senoidal.

Considérese $\varphi_1 = \varphi(m,k)$ y $\varphi_2 = \varphi(m+1,k)$, como las fases de la componente senoidal asociadas al bin k en tiempos $t_1 = T(m) = \frac{mH_a}{f_s}$ y $t_2 = T(m+1) = \frac{(m+1)H_a}{f_s}$. Por lo dicho anteriormente, se tiene en un principio una aproximación de la frecuencia de esta senoidal dada por $f = k\frac{f_s}{N}$. Se desea hallar una estimación de la frecuencia real $IF(f)$ de esta componente. Para facilitar el análisis, se incluye la figura 2.6. La señal roja corresponda a una senoidal con frecuencia f y fase φ_1 en la ventana m, mientras que la verde corresponde a una senoidal de frecuencia f y fase φ_2 en la ventana $m+1$. La señal gris es aquella cuya frecuencia real se quiere estimar.

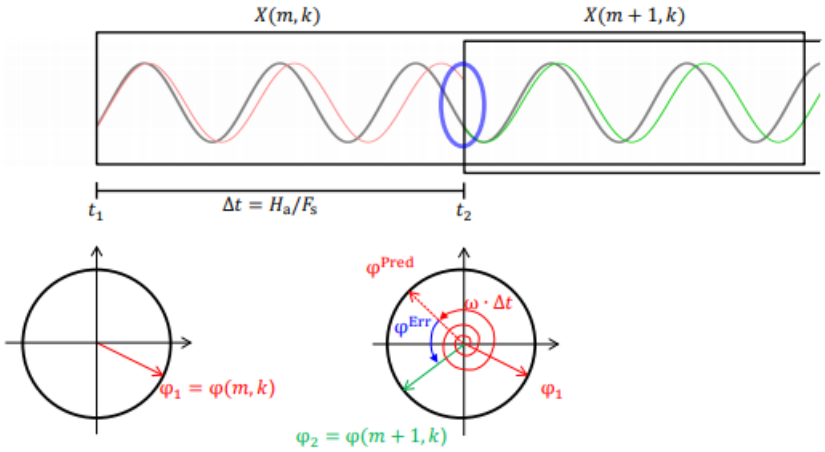

FIGURA 2.6: Estimación de la frecuencia real

Se hace notar que si bien la señal gris y la roja tienen la misma fase en t_1, la diferencia comienza a crecer debido al hecho de que sus frecuencias no son iguales (recordar que la de la señal roja era la correspondiente al bin k, es decir, su frecuencia es $\frac{kf_s}{N}$). Lo que sí puede hacerse, ya que se conoce su frecuencia, es computar la fase que tendría la señal roja en t_2, dada por:

$$\varphi^{Pred} = \varphi_1 + w\Delta t \tag{2.6}$$

(siendo $w = 2\pi f$)

Como también se conoce la fase en t_2, puede calcularse el error de fase entre la fase en t_2 y la predicha con una frecuencia w, obteniendo:

$$\varphi^{Err} = Arg(\varphi_2 - \varphi^{Pred}) \tag{2.7}$$

Donde Arg denota el argumento principal del valor, es decir, el error de fase se obtiene como un número entre $-\pi$ y π.

De esta forma, puede estimarse el valor de fase de la señal gris de forma similar a como se hizo con la roja, asumiendo que en t_2 la diferencia de fase entre ambas será menor a medio período, lo cual se cumplirá siempre que se elija bien el solapamiento entre ventanas (en general se toma

$H_a = \frac{N}{4}$, que equivale a un solapamiento del 75 %). La estimación de fase queda entonces:

$$\varphi_{gris} = w\Delta t + \varphi^{Err} \tag{2.8}$$

Este valor corresponde al número de oscilaciones de la señal gris en $t_2 - t_1 = \Delta t$ segundos, de donde la frecuencia de la señal puede estimarse como

$$IF(w) = \frac{w\Delta t + \varphi^{Err}}{\Delta t} = w + \frac{\varphi^{Err}}{\Delta t} \tag{2.9}$$

Se ve aquí entonces que $\frac{\varphi^{Err}}{\Delta t}$ corresponde a un offset entre la frecuencia real de la senoidal y la frecuencia de bin. Por lo tanto, puede estimarse la frecuencia real de cada componente de la STFT y así mejorar su resolución en frecuencia. Para $X(m, k)$ se redefinirá la frecuencia del bin como

$$F^{IF}(m, k) = w + \frac{Arg(\varphi_2 - (\varphi_1 + w\Delta t))}{\Delta t} \tag{2.10}$$

Se logró implementar un algoritmo del tipo Phase Vocoder en Python, y se pudo sintetizar un piano, un banjo, y un bassoon a partir de muestras obtenidas de la página web de la Universidad de Iowa. La ventaja de este método es que el algoritmo es el mismo independiente del instrumento a sintetizar, de donde si se quisiesen agregar instrumentos sólo se deberían conseguir las muestras de dichos instrumentos. Se comparó el algoritmo realizado con las implementaciones de Phase Vocoder de las librerías librosa (la que se utilizó para realizar la STFT) y de Python Rubberband. La función de librosa no realiza la correción de fase, de donde los resultados que se obtiene son, en general, de mala calidad. La función de Python Rubberband sí realiza la correción de fase, y los resultados que se obtienen con la misma son similares a los de la implementación propia realizada. Como desventaja, se hace notar que el tiempo de computación para sintetiza un track de un MIDI es en general mayor que con el otro tipo de síntesis. Además, para ciertos casos, se aprecia la introducción de un efecto 'metálico' en el sonido del track sintetizado, tanto en la implementación propia como en la de Python Rubberband. La misma se debe a la naturaleza del algoritmo, en las aproximaciones de la estimación de la frecuencia real, y existen diversas mejoras al algoritmo para solucionarla como son el Peak o Loose Phase Locking. Por lo tanto, si bien los resultados son en general buenos, se hace notar que el algoritmo podría mejorarse aún más.

3. Síntesis por modelos físicos

El método de síntesis por modelos físicos es un algoritmo basado en la síntesis de Karplus-Strong, que tiene en cuenta los parámetros físicos de los instrumentos para plantear la ecuación en diferencias. La idea principal se centra alrededor de la síntesis de *plucked strings*.

El circuito equivalente del modelo de Karplus-Strong es el siguiente:

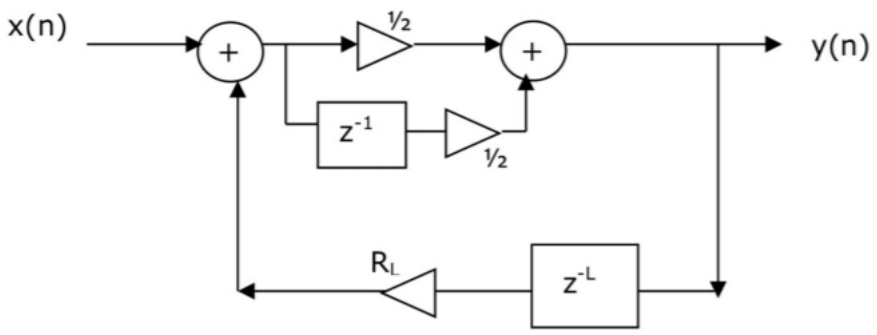

FIGURA 3.1: Modelo de Karplus-Strong

A lo largo de este inciso se desarrollará el análisis de este modelo, como así también diversas variaciones que se pueden implementar para mejorar la calidad del sonido, corregir ciertos defectos, simular de forma más acertada a los instrumentos reales y extender el algoritmo a otros instrumentos, como un harpa o un instrumento de percusión.

3.1. Función transferencia

3.1.1. Modelo original

Planteando el sistema de la Fig. 3.1, se puede encontrar su ecuación en diferencias, que será:

$$\frac{x(n) + R_L \cdot y(n-L)}{2} + \frac{x(n-1) + R_L \cdot y(n-L-1)}{2} = y(n) \tag{3.1}$$

Considerando a ambas señales como causales, se puede escribir en transformada Z como:

$$\frac{X(z)}{2} + \frac{R_L}{2} \cdot z^{-L} \cdot Y(z) + \frac{z^{-1}}{2} \cdot X(z) + \frac{R_L}{2} \cdot z^{-L-1} \cdot Y(z) = Y(z) \tag{3.2}$$

Ésta se puede reescribir como:

$$X(z) \cdot (1 + z^{-1}) = Y(z) \cdot R_L \cdot \left(\frac{2}{R_L} - z^{-L} - z^{-L-1} \right) \tag{3.3}$$

Finalmente, su función transferencia será:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + z^{-1}}{R_L \cdot \left(\frac{2}{R_L} - z^{-L} - z^{-L-1} \right)} = \frac{z^L \cdot (z+1)}{R_L \cdot \left(\frac{2}{R_L} \cdot z^{L+1} - z - 1 \right)} \tag{3.4}$$

Como se puede ver, el sistema tendrá sólo un cero útil en z = -1 ($f = \frac{f_s}{2}$). En cuanto a polos, tendrá L + 1, y su módulo estará directamente relacioando con el valor de R_L.

Para distintas combinaciones de R_L y L, su respuesta en frecuencia será la siguiente:

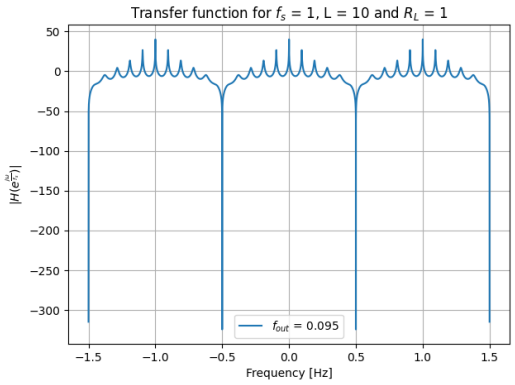

FIGURA 3.2: $R_L = 1$, $L = 10$, $f_s = 1Hz$

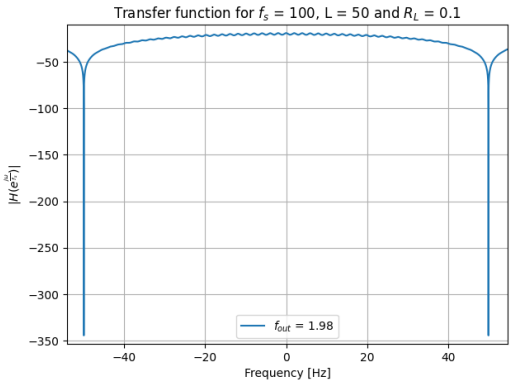

FIGURA 3.3: $R_L = 0.1$, $L = 50$, $f_s = 100Hz$

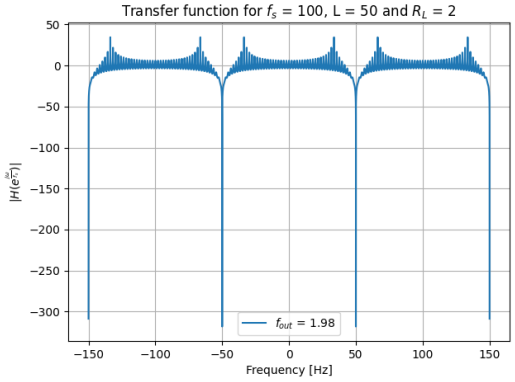

FIGURA 3.4: $R_L = 2$, $L = 50$, $f_s = 100Hz$

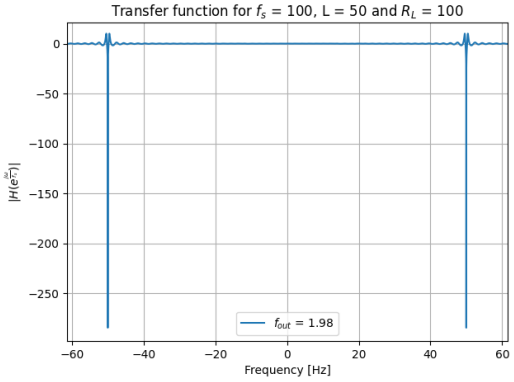

FIGURA 3.5: $R_L = 100$, $L = 10$, $f_s = 1Hz$

A simple vista, se puede observar una clara tendencia en cuanto al orden de magnitud de los resultados. Se observa que el valor de R_L será un factor importante en el resultado final de la respuesta en frecuencia. En la Fig. 3.3 y la Fig. 3.5, se puede ver que valores extremos de R_L generarán un comportamiento no deseado, ya que se buscará tener una especie de peine, como en la Fig. 3.2.

En esencia, podría decirse que valores grandes de R_L atenuarán las frecuencias medias y amplificarán las frecuencias altas, mientras que valores pequeños de R_L tendrán el efecto opuesto. Valores de R_L cercanos a 1, finalmente, tendrán el efecto deseado.

Para ver las salidas también en distintos casos, se analizaron tres aspectos principales:

- Su comportamiento exclusivamente temporal
- Su comportamiento exclusivamente espectral (FFT)

- Su comportamiento mixto en tiempo y frecuencia (espectrograma)

Los resultados obtenidos fueron los siguientes:

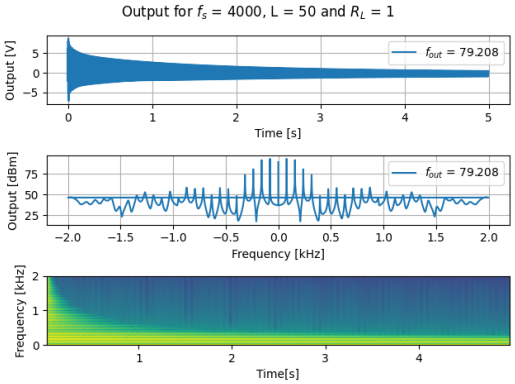

FIGURA 3.6: $R_L = 1$, $L = 20$, $f_s = 4kHz$

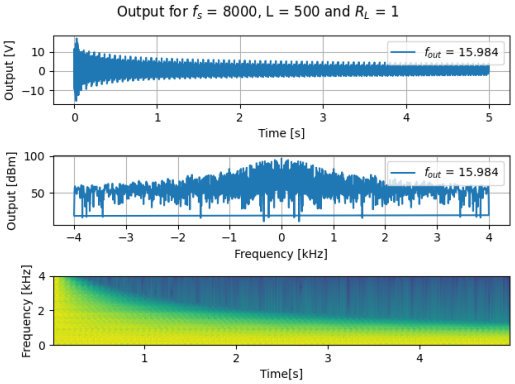

FIGURA 3.7: $R_L = 1$, $L = 500$, $f_s = 8kHz$

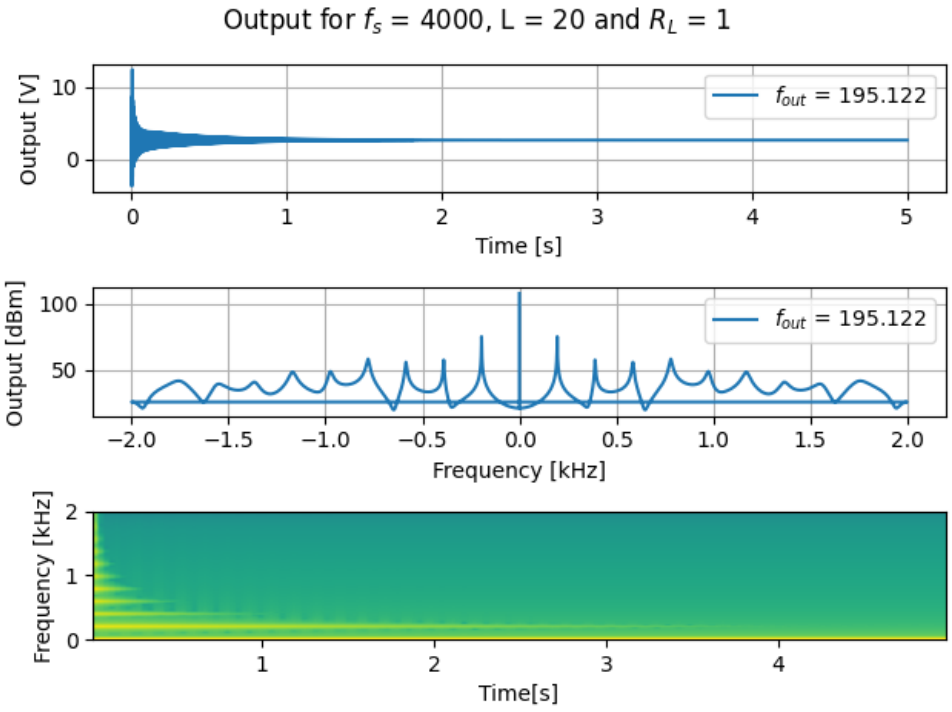

FIGURA 3.8: $R_L = 1$, $L = 50$, $f_s = 4kHz$

Sobre todo en la Fig. 3.6 y la Fig. 3.8, se puede ver que la salida presenta componentes armónicos periódicos, es decir, que tiene picos en múltiplos enteros de una frecuencia fundamental. Esto da la pauta de que el método de Karplus-Strong puede ser útil para la generación de instrumentos, como se está buscando.

3.1.2. Modelo modificado

El modelo modificado será igual al anterior, pero con la diferencia de que ahora R_L no es un número, sino que tomará los valores 1 y -1 con probabilidad b y $1 - b$, respectivamente. Por lo tanto, la nueva H(z) será:

$$H(z) = \begin{cases} \frac{z^L \cdot (z+1)}{(2 \cdot z^{L+1} - z - 1)}, & \text{con probabilidad b} \\ \frac{z^L \cdot (z+1)}{(2 \cdot z^{L+1} + z + 1)}, & \text{con probabilidad 1 - b} \end{cases}$$

Al cambiar el valor de b y comenzar a alejarlo de los extremos (0 y 1), las salidas tendrán un ataque más corto y morirán más rápidamente. Un valor de b = 0.5 tiene un sonido similar al de un instrumento de percusión.

3.2. Polos y ceros

3.2.1. Modelo original

Por simple inspección, se puede ver que su función transferencia tendrá L ceros en el origen y un cero en z = -1, es decir, en la frecuencia de Nyquist.

Con respecto a los polos, éstos surgirán de resolver la ecuación:

$$2 \cdot z^{L+1} = z + 1 \tag{3.5}$$

Tomando $R_L = 1$.

Esto se puede reescribir como:

$$2 \cdot z^{L+\frac{1}{2}} = z^{\frac{1}{2}} + z^{-\frac{1}{2}} \tag{3.6}$$

Reemplazando $z = a \cdot e^{i\omega}$, se tendrá:

$$2 \cdot a^{L+\frac{1}{2}} \cdot e^{i\omega\left(L+\frac{1}{2}\right)} = a^{\frac{1}{2}} \cdot e^{\frac{i\omega}{2}} + a^{-\frac{1}{2}} \cdot e^{-\frac{i\omega}{2}} \tag{3.7}$$

Si suponemos $a \approx 1$, entonces esto se puede reescribir como:

$$2 \cdot e^{i\omega\left(L+\frac{1}{2}\right)} = 2 \cdot cos\left(\frac{\omega}{2}\right) \tag{3.8}$$

De donde se tendrá:

$$\omega = \frac{2n\pi}{L + \frac{1}{2}} \tag{3.9}$$

Si desnormalizamos, finalmente se llegará a:

$$\omega = \frac{\omega_s \cdot n}{L + \frac{1}{2}} \tag{3.10}$$

Donde n es natural tal que $0 \leq n \leq L$.

Este resultado se graficó para distintos valores de L, obteniéndose los siguientes resultados:

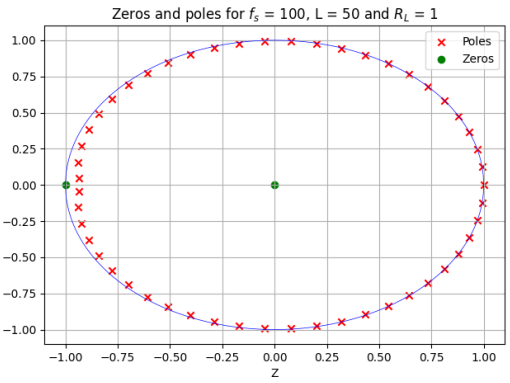

FIGURA 3.9: $R_L = 1$, $L = 50$, $f_s = 100$

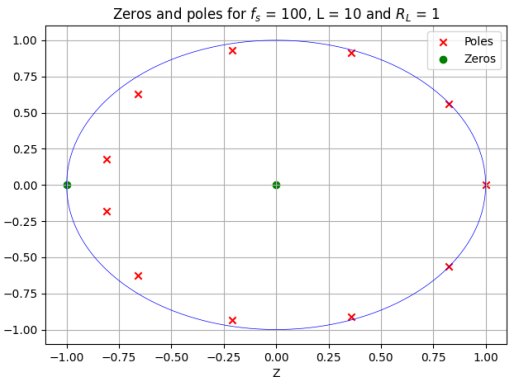

FIGURA 3.10: $R_L = 1$, $L = 10$, $f_s = 100$

Como se puede ver, los polos y ceros siempre estarán incluidos dentro del círculo de radio 1, con lo cual el sistema será estable. Sin embargo, al modificar R_L, el módulo de los polos se modificará, ya que este valor absoluto será del orden de $R_L^{\frac{1}{L+1}}$. Graficando para valores de R_L Ie 1, se obtendrán resultados similares a los siguiente:

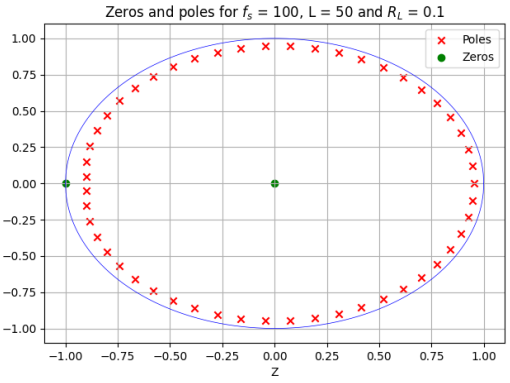

FIGURA 3.11: $R_L = 0.1$, $L = 50$, $f_s = 100$

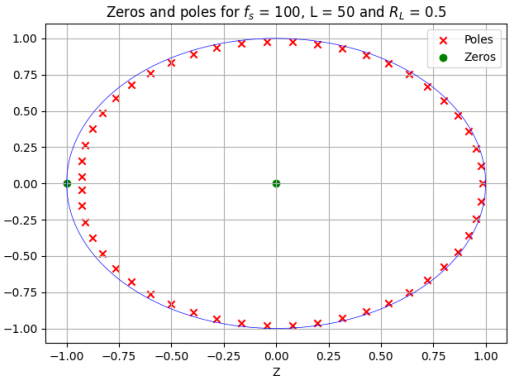

FIGURA 3.12: $R_L = 0.5$, $L = 50$, $f_s = 100$

Como se puede ver en la Fig. 3.11 y la Fig. 3.12, el sistema seguirá siendo estable. Sin embargo, si ahora se utilizan valores de R_L mayores a 1:

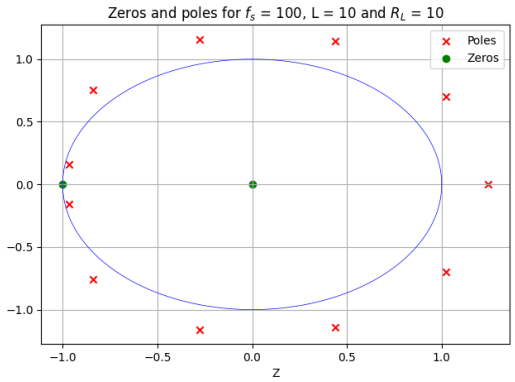

FIGURA 3.13: $R_L = 10$, $L = 10$, $f_s = 100$

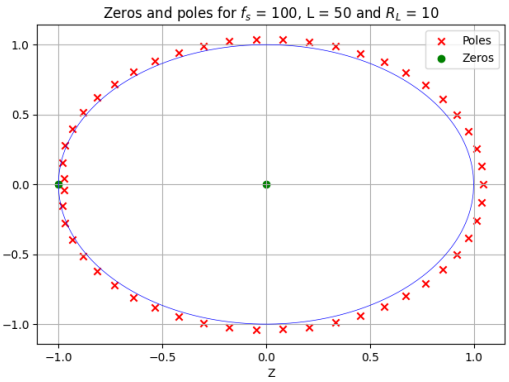

FIGURA 3.14: $R_L = 10$, $L = 50$, $f_s = 100$

Se observa claramente en la Fig. 3.13 y la Fig. 3.14 que con valores de R_L mayores a 1, el sistema dejará de ser estable, ya que tendrá polos fuera del círculo unitario. Esto se debe, como se mencionó antes, a que el orden de magnitud del valor absoluto de los polos será similar a $R_L^{\frac{1}{L+1}}$. Esto quiere decir que, si bien valores de R_L mayores a 1 harán que el sistema sea inestable, este efecto se puede contrarrestar tomando valores grandes de L, que harán que los polos tiendan a caer sobre la circunferencia unitaria.

3.2.2. Modelo modificado

En cuanto a los ceros, el modelo modificado los tendrá en el mismo lugar. En cuanto a los polos, el procedimiento es el mismo, sólo que, cuando R_L = -1, se llegará a algo de la forma:

$$2 \cdot e^{i\omega\left(L+\frac{1}{2}\right)} = 2i \cdot sin\left(\frac{\omega}{2}\right) \tag{3.11}$$

De donde se tendrá que:

$$\omega = \frac{\pi \cdot (2n+1)}{L+\frac{1}{2}} \tag{3.12}$$

Y desnormalizando:

$$\omega = \frac{\omega_s \cdot \left(n+\frac{1}{2}\right)}{L+\frac{1}{2}} \tag{3.13}$$

Finalmente:

$$\omega_P = \begin{cases} \frac{\omega_s \cdot n}{L+\frac{1}{2}}, & \text{con probabilidad b} \\ \frac{\omega_s \cdot \left(n+\frac{1}{2}\right)}{L+\frac{1}{2}}, & \text{con probabilidad 1 - b} \end{cases}$$

Como se puede ver, la parte donde R_L = -1 lo que hace efectivamente es reducir la frecuencia resultante a la mitad y eliminar los armónicos pares. Esto quiere decir que donde antes habría una frecuencia resultante de f, $2f$, $3f$..., ahora habrá $\frac{f}{2}$, $\frac{3f}{2}$, $\frac{5f}{2}$...

3.3. Límite de R_L

A partir de la Fig. 3.13 y la Fig. 3.14, se puede ver que el sistema será estable siempre que $|R_L| \leq 1$, ya que, de no ser así, la transferencia tendría polos por fuera del círculo unitario. El motivo matemático se explicó en la Sec. 3.2.1.

En cuanto a su interpretación física, este comportamiento se puede explicar teniendo en cuenta la ecuación en diferencias. Si R_L tuviera módulo mayor a 1, entonces las salidas se estarían constantemente amplificando y realimentando, lo cual causaría que el sistema tienda a crecer infinitamente incluso para entradas acotadas.

3.4. Caja de guitarra

Siguiendo el procedimiento de este trabajo, se puede incluir un filtro más en el lazo, con el fin de generar el efecto de la caja de la guitarra. Se puede ver que el filtro planteado es el siguiente:

$$H_L(z) = \frac{0.8995 \cdot z + 0.1087}{z + 0.0136} \tag{3.14}$$

O, de forma más genérica:

$$H_L(z) = \frac{a \cdot z + b}{c \cdot z + d} \tag{3.15}$$

Poniendo este filtro en el loop, ahora la nueva función transferencia será:

$$H(z) = \frac{z^L \cdot (z+1)}{R_L \cdot \left(\frac{2}{R_L} \cdot z^{L+1} - (z+1) \cdot H_L(z)\right)} \tag{3.16}$$

Multiplicando y diviendo por $c \cdot z + d$, y aplicando distributiva, se puede llegar a:

$$H(z) = \frac{z^L \cdot (c \cdot z^2 + (c+d) \cdot z + d)}{R_L \cdot \left(\frac{2}{R_L} \cdot c \cdot z^{L+2} + \frac{2}{R_L} \cdot d \cdot z^{L+1} - a \cdot z^2 - (a+b) \cdot z + b\right)} \tag{3.17}$$

La transferencia de este filtro es un pasabajos, que tiene como finalidad emular la atenuación que generan las paredes de la guitarra. Su respuesta en frecuencia se puede ver en el siguiente gráfico:

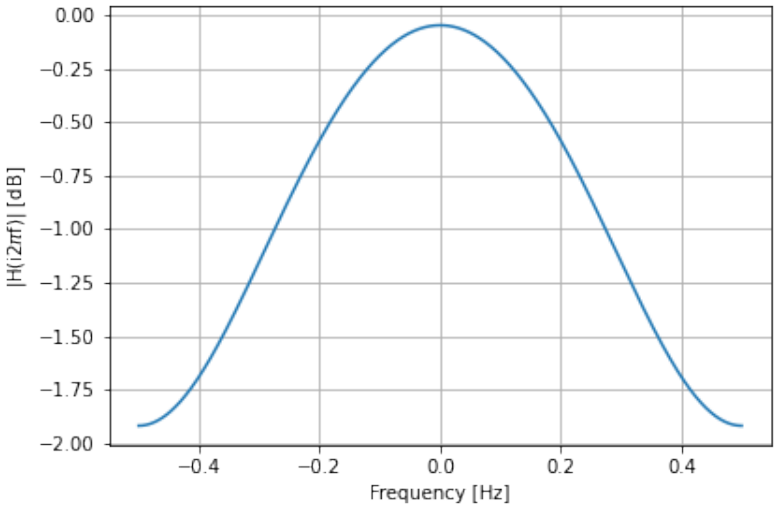

FIGURA 3.15: Función transferencia de filtro emulador de caja de guitarra

Como se puede observar, la transferencia es la de un pasabajos de primer orden, con pendiente suave. Lo que esto hará es atenuar a las altas frecuencias, de forma tal que su sonido durará menos tiempo. Para frecuencias muy altas, este efecto no es del todo deseado, porque de por sí ya el algoritmo de Karplus-Strong actúa como pasabajos. En la siguiente sección se puede ver cómo solucionar este problema.

3.5. Ajuste de redondeo

Para mitigar el error introducido por el redondeo de la frecuencia, ya que ésta será $\frac{f_s}{L+\frac{1}{2}}$, una posible solución consiste en introducir un all-pass filter en el loop. Análogo al caso anterior, esta vez los coeficientes serán:

- a = d = C
- b = c = 1

De esta manera, la transferencia será:

$$H(z) = \frac{z^L \cdot (z^2 + (1+C) \cdot z + C)}{R_L \cdot \left(\frac{2}{R_L} \cdot z^{L+2} + \frac{2}{R_L} \cdot C \cdot z^{L+1} - C \cdot z^2 - (1+C) \cdot z + 1\right)} \tag{3.18}$$

Su respuesta en frecuencia, como se mencionó, es la de un all-pass filter, es decir, constante. Esto quiere decir que no tendrá efecto sobre la amplitud de los armónicos, sino en su fase. Esto introducirá polos y ceros que, a la hora de evaluar el valor absoluto de la transferencia total, pesarán también la distancia de estos puntos a la circunferencia unitaria, mejorando los errores por redondeo de $\frac{f_s}{L+\frac{1}{2}}$.

Sin embargo, como se mencionó antes, el único problema para las altas frecuencias no es el error por redondeo, sino también la excesiva atenuación. Una forma de solucionar este problema

es cambiando el promedio por un promedio ponderado, con un factor S que dará un estiramiento o *stretch*. La nueva ecuación en diferencias será:

$$\frac{x(n) + R_L \cdot y(n-L) \cdot \left(2 - \frac{1}{S}\right)}{2} + \frac{x(n-1) + R_L \cdot y(n-L-1) \cdot \frac{1}{S}}{2} = y(n) \tag{3.19}$$

Y su transferencia será:

$$H(z) = \frac{z^L \cdot (z+1)}{R_L \cdot \left(\frac{2}{R_L} \cdot z^{L+1} - \left(2 - \frac{1}{S}\right) \cdot z - \frac{1}{S}\right)} \tag{3.20}$$

Su respuesta en frecuencia tendrá la siguiente forma:

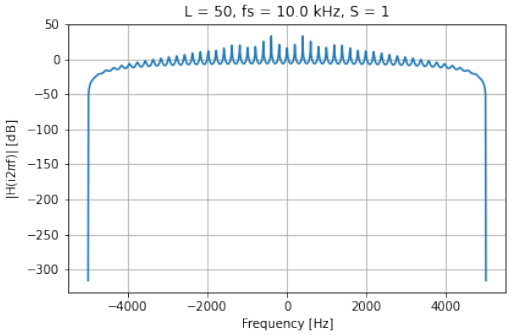

FIGURA 3.16: $S = 1$

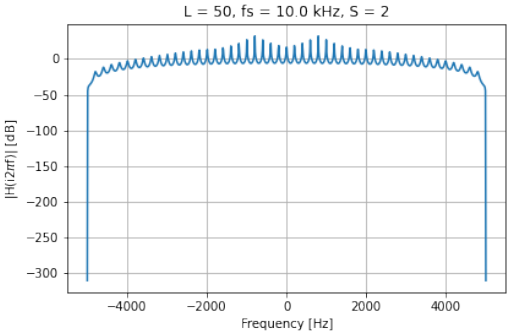

FIGURA 3.17: $S = 2$

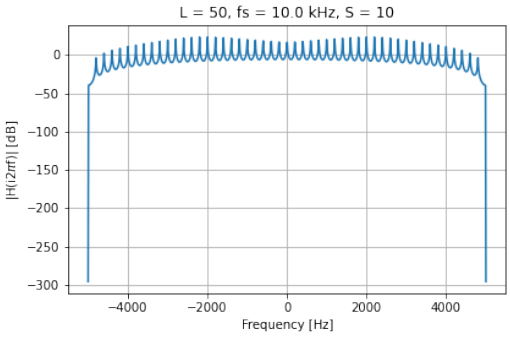

FIGURA 3.18: $S = 10$

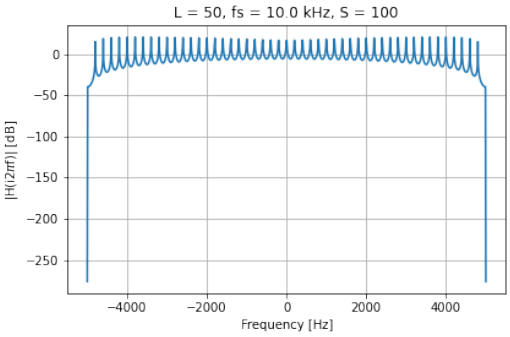

FIGURA 3.19: $S = 100$

En las figuras anteriores se puede ver claramente el efecto que tiene S sobre la respuesta en frecuencia. A medida que S aumenta, también aumenta la ganancia del sistema en altas frecuencias. Esto hará que la atenuación introducida por el modelo de Karplus-Strong original en altas frecuencias se vea reducida de forma proporcional a este factor, resultando en un decaimiento más lento de las frecuencias altas. A continuación se pueden ver distintas salidas de este modelo:

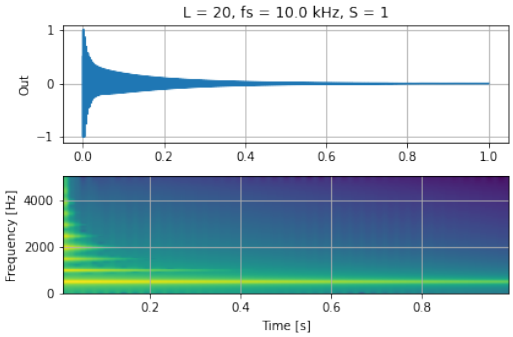

FIGURA 3.20: $S = 1$

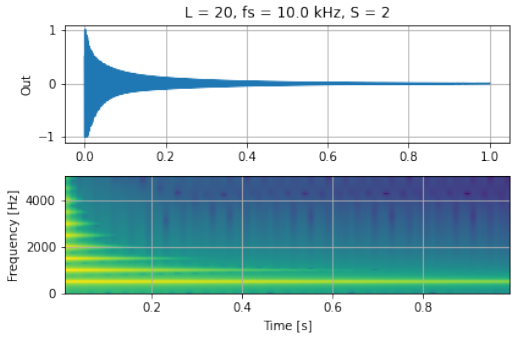

FIGURA 3.21: $S = 2$

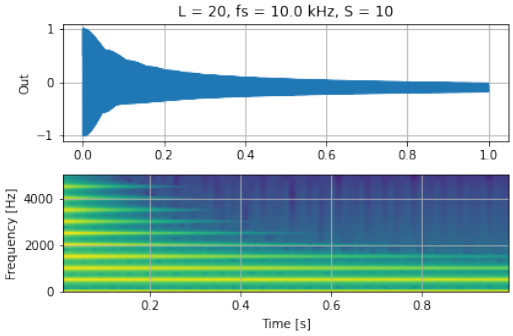

FIGURA 3.22: $S = 10$

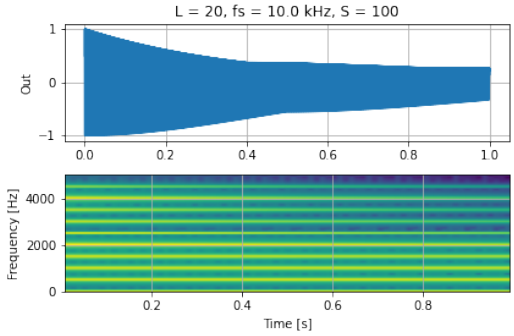

FIGURA 3.23: $S = 100$

Se puede observar, nuevamente, que un valor mayor de S efectivamente genera un decaimiento más lento. Además, en los espectrogramas se observa cómo los armónicos de más alta frecuencia tienen mayor duración a medida que S aumenta.

Por último, este modelo dará como resultado que la frecuencia a la salida será:

$$f_k = \frac{f_s}{L + \frac{1}{2S}} \tag{3.21}$$

Como se puede ver, este factor S no sólo ayuda al estiramiento de las altas frecuencias, sino también a mejorar el error por redondeo.

3.6. Fase en función de b

De la expresión 3.4, se puede escribir:

$$H(z) = \frac{e^{i\omega L} \cdot (e^{i\omega} + 1)}{R_L \cdot \left(\frac{2}{R_L} \cdot e^{i\omega(L+1)} - e^{i\omega} - 1\right)} \tag{3.22}$$

Luego, se puede expresar $\Phi(\omega)$ como la fase del numerador menos la del denominador. De esta manera, se tendrá:

$$\Phi_N(\omega) = \omega \cdot L + arctan\left(\frac{sin(\omega)}{1 + cos(\omega)}\right) \tag{3.23}$$

$$\Phi_D(\omega) = arctan\left(\frac{\frac{2}{R_L}\cdot sin(\omega\cdot(L+1)) - sin(\omega)}{\frac{2}{R_L}\cdot cos(\omega\cdot(L+1)) - cos(\omega) - 1}\right) \tag{3.24}$$

Por lo tanto:

$$\Phi(\omega) = \begin{cases} \Phi_N(\omega) - arctan\left(\frac{2\cdot sin(\omega\cdot(L+1)) - sin(\omega)}{2\cdot cos(\omega\cdot(L+1)) - cos(\omega) - 1}\right), & \text{con probabilidad b} \\ \Phi_N(\omega) - arctan\left(\frac{2\cdot sin(\omega\cdot(L+1)) + sin(\omega)}{2\cdot cos(\omega\cdot(L+1)) + cos(\omega) + 1}\right), & \text{con probabilidad 1 - b} \end{cases}$$

3.7. Frecuencia fundamental en función de b

Por último, se puede obtener una expresión para b en función de β, donde:

$$\beta = \frac{f_s}{f_k} - (L + \frac{1}{2}) \tag{3.25}$$

Siendo f_k la frecuencia fundamental. Analizando en los casos límites, se tendrá:
Con b = 1, $f_k = \frac{f_s}{L+\frac{1}{2}}$, entonces:

$$b = 1 \rightarrow \beta = 0$$

Luego, con b = 0, $f_k = \frac{f_s}{2\cdot\left(L+\frac{1}{2}\right)}$, como se vio previamente. Por lo tanto:

$$b = 0 \rightarrow \beta = \left(L + \tfrac{1}{2}\right)$$

Por lo tanto, se puede extrapolar de ambos puntos y se obtendrá:

$$b = 1 - \frac{\beta}{L + \frac{1}{2}} \tag{3.26}$$

O, lo que es lo mismo:

$$f_k = \frac{f_s}{(2-b)\cdot\left(L + \frac{1}{2}\right)} \tag{3.27}$$

4. Efectos

4.1. Eco simple

Un efecto utilizado frecuentemente es la aplicación de un eco simple, que puede verse como un *delay* en la señal. Este retardo en la señal recibida se debe a los rebotes que sufre la señal en diferentes partes del espacio en el que se propaga el sonido. Para implementar este retraso en la señal, se utiliza un filtro *comb*, que suma a la señal de entrada la salida en un instante anterior multiplicada por cierta ganancia. Para visualizar este filtro se puede observar la Fig. 4.1.

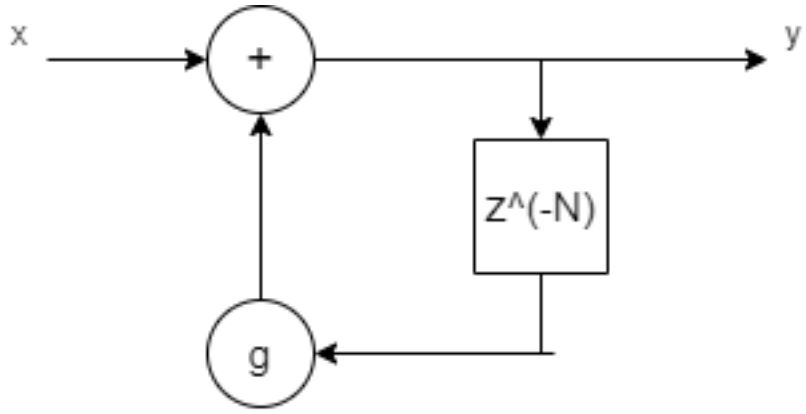

FIGURA 4.1: Comb filter

Puede verse que la ecuación en diferencia del filtro queda como sigue.

$$y(n) = x(n) + g \cdot y(n - N) \tag{4.1}$$

De la ecuación en diferencias se aprecia claramente que el valor de N es el retardo de la señal. La función transferencia se puede hallar fácilmente aplicando transformada Z a la ecuación anterior.

$$H(z) = \frac{1}{1 + g \cdot z^{-N}} \tag{4.2}$$

Este filtro lo que produce son deltas a una distancia uniforme τ entre sí, cuyas amplitudes van disminuyendo conforme pasa más tiempo. La generación de estas deltas son las que harán escuchar la repetición de una señal y así produciendo el efecto de eco. Es de importancia aclarar que el factor "g" debe ser en módulo menor a 1 para que los impulsos se vayan atenuando y no se amplifiquen las señales conforme pase el tiempo.

Los parámetros importantes de este efecto son el *loop time* (τ) y el *Reverb time*. El primero determina la distancia entre los deltas de la respuesta impulsiva, y el segundo representa el tiempo en que la repetición de la señal se atenúa 60dB.

4.2. Filtro comb pasabajos

Un filtro usado por efectos más complejos como el *Freeverb* es el *lowpass-feedback comb filter*. Este filtro, actúa de pasabajos y a la vez de resonador, dado que la salida del filtro sufre una

atenuación en las altas frecuencias y a la vez se le suma un retardo. La función transferencia que se obtiene es la que se muestra a continuación.

$$H(z) = \frac{z^{-N}}{1 - f\frac{1-d}{1-d\cdot z^{-1}}z^{-N}} \tag{4.3}$$

Al igual que en el caso anterior, "N" representa el *delay* deseado en la señal. El parámetro "d" se lo conoce como *damping factor* y puede interpretarse como un proceso que transforma la vibración en otro tipo de fenómeno físico como el calor, para que la vibración que produce el sonido se apague más rápido. De esta forma, controlando el valor que tome "d" se puede controlar la intensidad con la que una nota puede estar presente y llegar a producir el efecto. Por otro lado, "f" representa el tamaño de la habitación en la que se propaga el sonido, por lo que este factor está muy relacionado con el tiempo que perdura dicha señal. Un dato no menor, es que para que el sistema sea estable, "f" debe ser menor a uno.

4.3. Flanger

A diferencia de los filtros anteriores, en este caso el retraso es variante en el tiempo. Además, el retraso se genera sobre la señal de entrada y no sobre la salida. La forma del filtro puede verse en la Fig. 4.2.

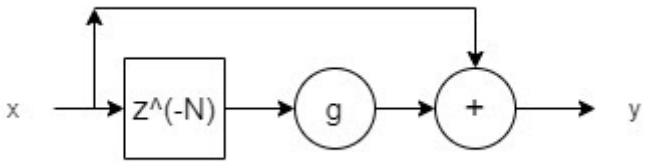

FIGURA 4.2: Filtro Flanger

Se puede sacar la ecuación en diferencias como:

$$y(n) = x(n) + g \cdot x(n - M(n)) \tag{4.4}$$

M(n) es el desfasaje variante en el tiempo. En el caso particular de este informe se tomó una señal senoidal para hacer de variación del retraso. Como el *delay* varía con el tiempo para la implementación se decidió trabajar directamente en el dominio del tiempo. Llevándolo al dominio de la frecuencia, la expresión queda:

$$H(z) = 1 + z^{-M(n)} \tag{4.5}$$

Este tipo de efectos se usa mucho en el ámbito de la música y suele aplicarse para agregarle un sonido más metálico al instrumento.

5. Espectrograma

Se simuló a través de síntesis aditiva la escala de Sol Mayor G3, con una duración de 120ms por nota. A continuación, se muestra el espectrograma resultante.

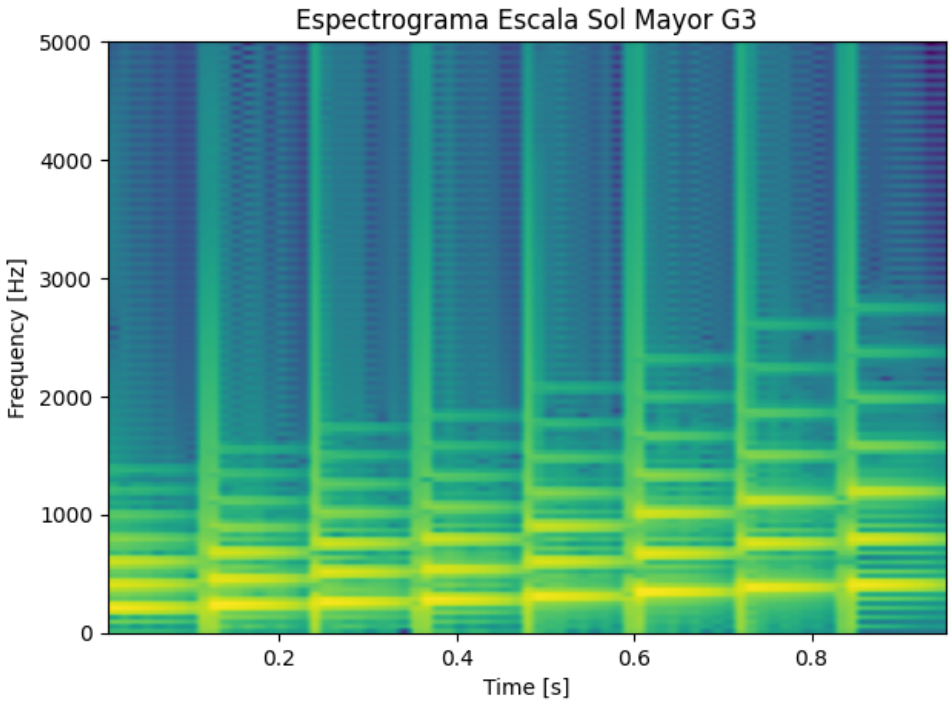

FIGURA 5.1: Espectrograma de la escala a estudiar

La ventana en cuestión a utilizar fue Bartlett, con una overlapping del 50 %. Bartlett se utilizó porque tiene un lóbulo principal más angosto y estamos tratando con deltas en frecuencia, en lugar de un espectro continuo, con lo cual el lóbulo angosto funciona para la señal elegida. Se probaron múltiples ventanas, pero se decidió por Bartlett ya que fue la que mayor nitidez presentó para las frecuencias de los armónicos, mientras que con otras ventanas se veían más difusas.

Si el overlapping fuese cero, según la ventana se puede estar perdiendo información. En el caso de una triangular, como la ventana de Bartlett, atenúa todo excepto la parte central de ese chunk. En particular, los bordes estarán muy atenuados hasta llegar a cero. Para no perder esta información, se superponen las ventanas de modo tal que el mínimo resultante no sea cero, si no el punto de cruce entre las ventanas. Si hacemos esto con triangulares, el mínimo va a ser el porcentaje de overlapping seleccionado (por ejemplo, 10 % sería 0,1, 25 % sería 0,25, etc).

Se puede ver una cantidad limitada de armónicos, como consecuencia a la decisión de diseño para la síntesis aditiva de tomar las amplitudes de los armónicos manualmente, cortando a partir de aquellos cuya amplitud comienza a ser despreciable.