22.05 - Análisis de Señales y Sistemas Digitales

Instituto Tecnológico de Buenos Aires

Trabajo Práctico 2

Análisis y síntesis de audio digital

Grupo 1

AUTORES:

Pablo Martín Scheinfeld (59065),
Santiago Agustín Arribere (59169),
Facundo Farall (59345),
Gonzalo Joaquín Davidov (59117)
Malena Müller (57057),

PROFESORES:

Carlos F. Belaustegui Goitia,
Daniel Andres Jacoby
Rodrigo Iñaki Iribarren

CIUDAD AUTÓNOMA DE BUENOS AIRES

Mayo 2020

Contenido

1. Introducción

El objetivo del presente trabajo es el desarrollo de un programa que lea archivos MIDI y pueda realizar modificaciones a las pistas que el archivo contiene. Para ello, se aplicarán distintos tipos de síntesis para lograr obtener diversos resultados y ser capaces de generar una gran variedad de instrumentos.

2. Fast Fourier Transform

Dentro del directorio del presenta trabajo, se incluye una solución de Visual Studio conformada por dos proyectos: *ASSD-TP2* y *TestProject*. Las carpetas de ambos proyectos, junto con el archivo de la solución que los nuclea, pueden encontrarse en el *path*:
/path_to_tp2_directory/ASSD-TP2

El proyecto *ASSD-TP2* esta conformado fundamentalmente por dos clases: *NumCpp* y *FFTCalculator*, cuyas funciones son facilitar el uso de arreglos de señales e implementar el algoritmo Cooley-Tukey FFT para obtener la DFT de una señal dada, respectivamente. La primera fue inspirada en el paquete *NumPy* disponible para el lenguaje Python, y se buscó proveer funciones que favorecieran la fácil lectura del código para la utilización de la segunda clase (*FFTCalculator*), la más relevante de las dos.

El segundo proyecto (*TestProject*) esta vinculado al primero en el sentido de que hace uso de la herramienta *unittesting* para evaluar el funcionamiento de las clases implementadas en *ASSD-TP2*, en él se puede encontrar un archivo *TextProject.cpp* con una función *FFTTest1*, definida haciendo uso de la macro *TEST_METHOD*. Compilando la solución, la herramienta de *Text Explorer* de Visual Studio debería detectar esta función para poder correrla como tal. Si esto no ocurre, puede activarse la herramienta manualmente desde *Test->Test Explorer*, o con el comando *Ctrl + E, T*, tal y como se observa en la Figura 2.1.

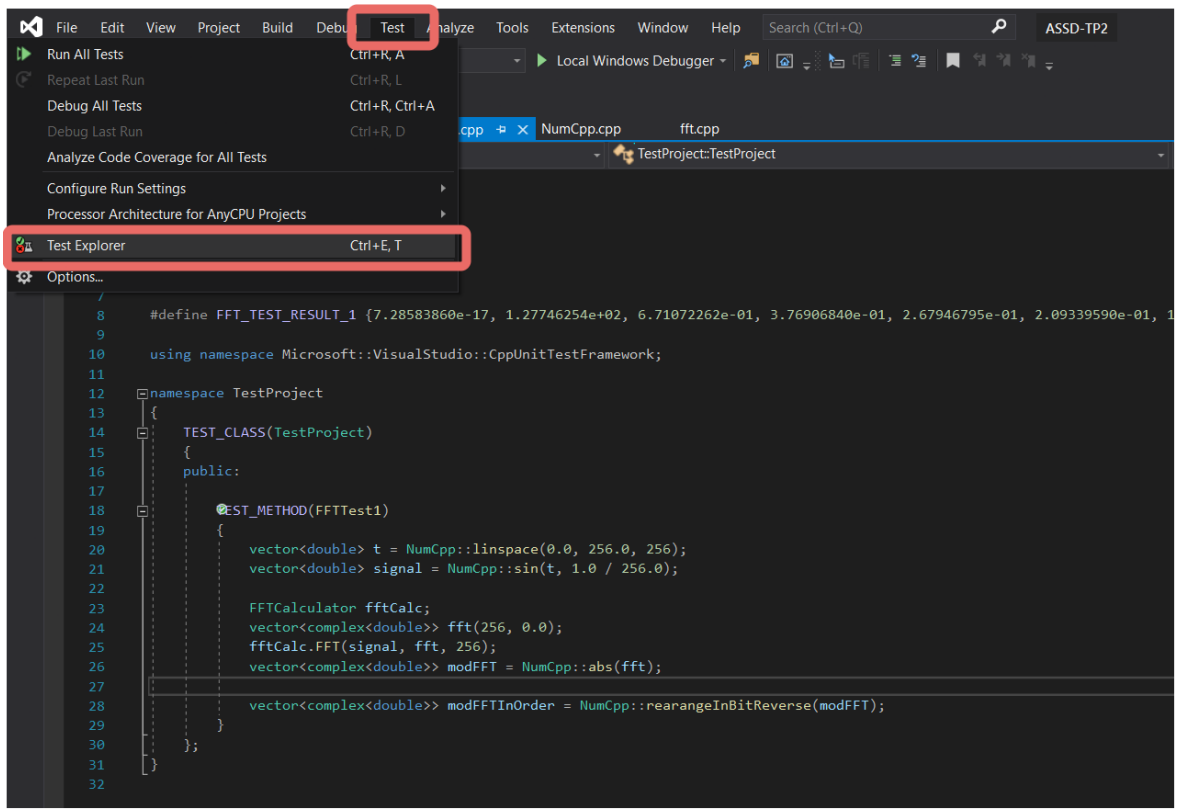

FIGURA 2.1: Ejecutar el *Text Explorer*.

Una vez descubierta la función de test, puede ejecutárselo de dos maneras: mediante el icono que surge junto a la definición de la función (Figura 2.2) o en la pestaña de *Text Explorer* (Figura 2.3).

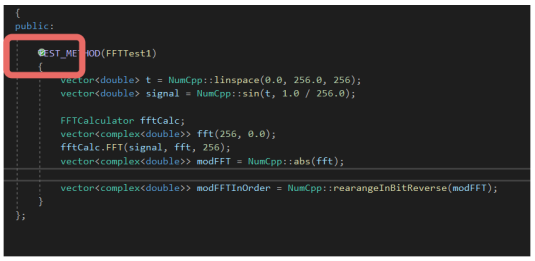

FIGURA 2.2: Ejecutar el *Text Explorer*.

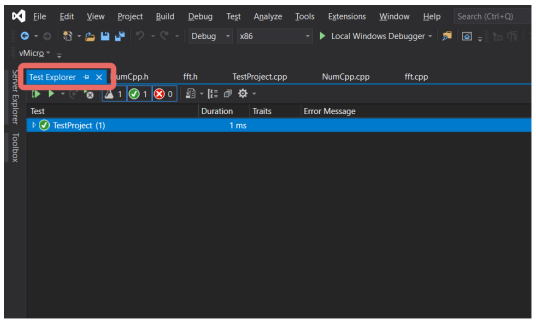

FIGURA 2.3: Ejecutar el *Pestaña Text Explorer, para ejecutar los tests.*.

La realización de esta implementación tiene valor didáctico en su función de ayudar a la comprensión de un algoritmo ampliamente utilizado en *signal processing*, y es de utilidad futura si debe emplearse en integrados sencillos sin soporte de lenguajes de más alto nivel. Sin embargo, no será la que se use en el resto del trabajo práctico, el cual sí hará uso de implementaciones en lenguajes de mayor nivel (en particular, Python), dado que se busca minimizar los tiempos desde la idea hasta la realización del proyecto, y provee una mejor plataforma para el desarrollo de GUIs.

Podría argumentarse que el uso de lenguajes de alto nivel está en contraposición con una aplicación de *real-time* como es esta, y esto sería técnicamente correcto. Sin embargo, si la implementación en Python se logra haciendo uso de librerías standard basadas en C, con una fina capa de alto nivel para proveer la interfaz, como lo es *NumPy* y *SciPy*, la ventaja en velocidad de lenguajes como C/C++ es solo apreciable cuando se complementa con directivas de optimización en la compilación, e incluso en esas circunstancias, no es una diferencia lo suficientemente grande como para contrarrestar el argumento previamente dado en favor de lenguajes de más alto nivel [1].

[1] Comparación detallada de tiempos de ejecución de C/C++ contra Python utilizando librerías standard.

3. Síntesis aditiva

La presente sección del trabajo tiene como objetivo la síntesis de distintos instrumentos mediante el método de síntesis aditiva. Este consiste en obtener un timbre deseado a partir de la suma de señales senoidales de distintas frecuencias y amplitudes. La determinación de estas señales, llamadas parciales, depende del instrumento a sintetizar y se corresponde con los distintos armónicos observados en el espectro de un tono analizado.

3.1. Envolventes de amplitud

Como se mencionó, los parciales son las distintas señales senoidales requeridas para generar un timbre. Con respecto a sus características, presentan variaciones en frecuencia y amplitud. Cada una se encuentra alrededor de uno de los respectivos armónicos del tono a reproducir. Por otra parte, sus amplitudes no son constantes sino que se exhiben distintos valores a través del tiempo. A dicha variación se la denomina envolvente acústico y existen diversas formas de especificarlo.

3.1.1. ADSR

La más común de las formas de especificar el envolvente acústico es la ADSR. Su nombre hace referencia a cada una de las etapas que la componen. Estas son: *attack*, o *ataque*; *decay* o *decaimiento*; *sustain* o *sostenimiento*; y *release* o *relajación*. En la figura 3.1 se puede observar el modelo ideal de este tipo de envolvente.

FIGURA 3.1: Etapas de la envolvente ADSR.

La primera etapa observable es la etapa de ataque, la cual responde a una pendiente positiva

cuyo inicio está dado por el accionar de una nota (por ejemplo la presión de una tecla en un piano). A su vez, esta etapa es la que determina la mayor amplitud y es aquí donde comienza la próxima etapa. Una vez alcanzada la máxima amplitud, comienza el decaimiento que representa un pronunciado descenso que termina con el comienzo de la etapa de sostenimiento. En la etapa de sustain, la amplitud se mantiene estable. Si bien en la figura 3.1 se observa un valor de amplitud constante en toda la fase, en la práctica se presenta una leve pendiente negativa - considerablemente menor en módulo a la del decay-. El sostenimiento de la amplitud se da hasta que finaliza el mecanismo de accionar la nota en el instrumento utilizado (por ejemplo, cuando se suelta la tecla en el piano). En tal momento, el sonido no finaliza instantáneamente, sino que se reduce progresivamente su amplitud hasta que finalmente alcanza el valor nulo. A este período se lo denomina relajación. Cabe descacar que las amplitudes y tiempos de las cuatro etapas dependen tanto del instrumento como de la duración de la nota reproducida.

3.1.2. AD

Aunque la envolvente del tipo ADSR se adapta en gran cantidad de casos a los timbres de los distintos instrumentos, existen otras formas de envolvente que funcionan mejor con algunos otros. Entre estas formas se encuentra la AD (*attack-decay*), la cual se puede tratar como un caso particular de la ADSR mencionada anteriormente. La particularidad de esta es que no presenta etapas de sustain ni de release. En la figura 3.2 se puede observar la forma que adquiere.

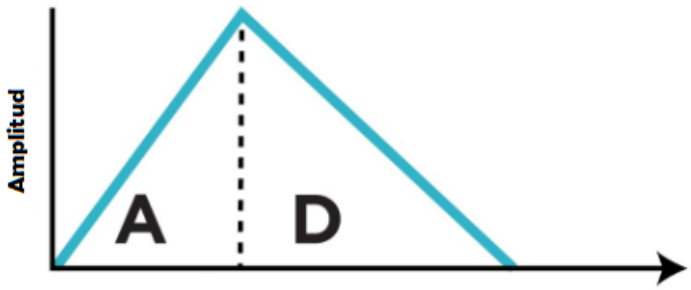

FIGURA 3.2: Etapas de la envolvente AD.

Los instrumentos que mejor se adaptan a esta envolvente son los de percusión, donde no es posible un sostenimiento del sonido. Una vez percutido el instrumento, el sonido crece hasta un pico (*attack*) y luego cae hasta su anulación (*decay*).

3.1.3. ASR

Otro caso particular de la ADSR es la ASR (*attack-sustain-release*), cuya particularidad principal es la no presencia de la etapa de decay. Así, el sonido crece hasta alcanzar un máximo, el cual se mantiene y luego decae hasta hacerse nulo. Distintos instrumentos presentan características de envolvente similares a este tipo, entre ellos ciertos instrumentos de cuerda, como la viola o el cello en ciertas circunstancias. En la figura 3.3 se exhibe su forma adoptada.

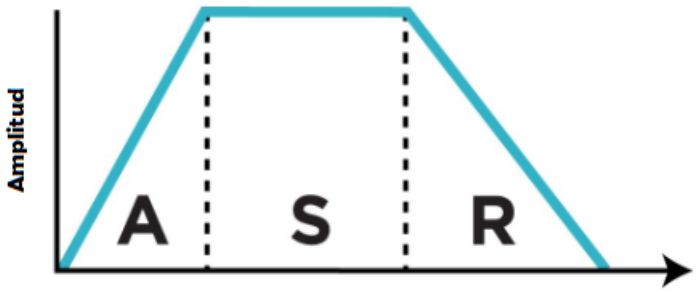

FIGURA 3.3: Etapas de la envolvente ASR.

3.2. Determinación de la envolvente de distintos instrumentos

A modo de poner en práctica los conceptos teóricos presentados, se procedió a observar las señales producidas por las notas de distintos instrumentos y relacionarlas con los tipos de envolventes de amplitud. Para realizar esto, se utilizó tanto un conjunto de datos de notas musicales aisladas, TinySOL, el cual cuenta con 2913 samples y 14 distintos instrumentos, como algunas muestras extrídas de The McGill University Master Samples. Cabe destacar que los gráficos mostrados a continuación corresponden a la señal completa y no a los parciales, los cuales podrían variar y se analizarán en la etapa de síntesis.

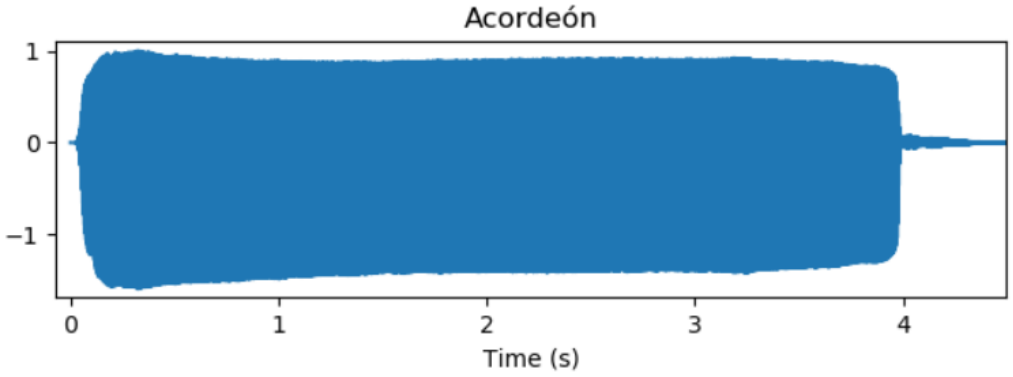

FIGURA 3.4: Envolvente obtenida del acordeón.

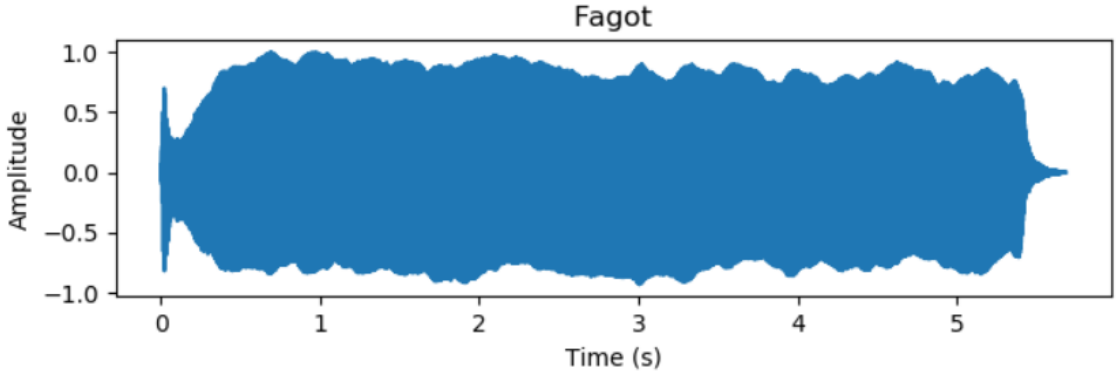

FIGURA 3.5: Envolvente obtenida del fagot o bassoon.

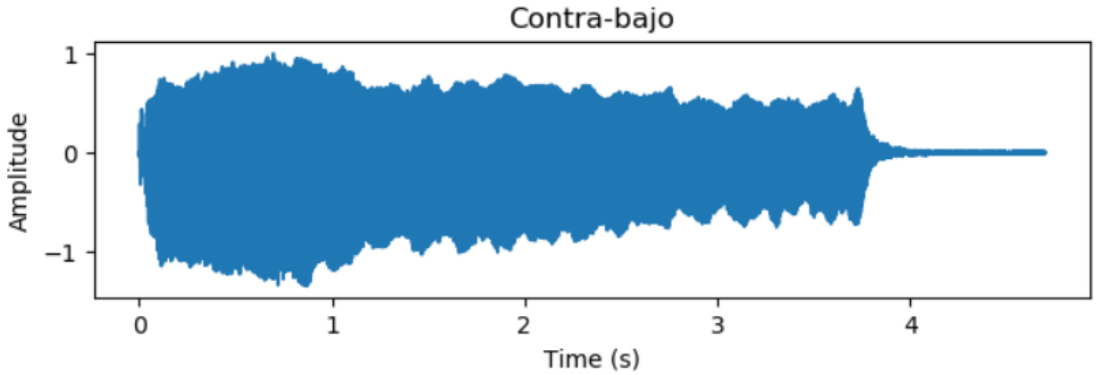

FIGURA 3.6: Envolvente obtenida del contra-bajo.

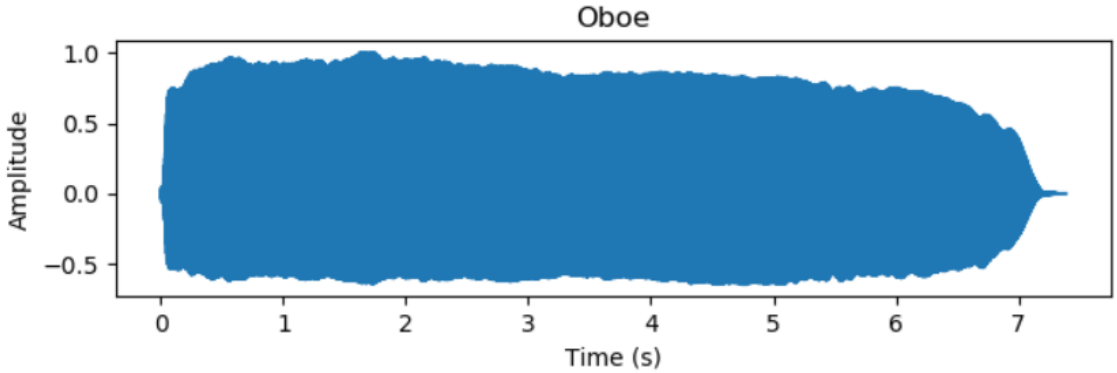

FIGURA 3.7: Envolvente obtenida del oboe.

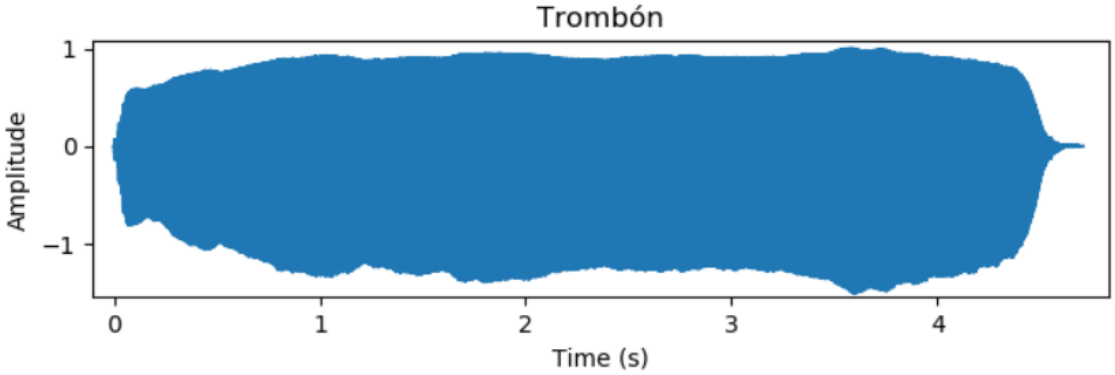

FIGURA 3.8: Envolvente obtenida del trombón.

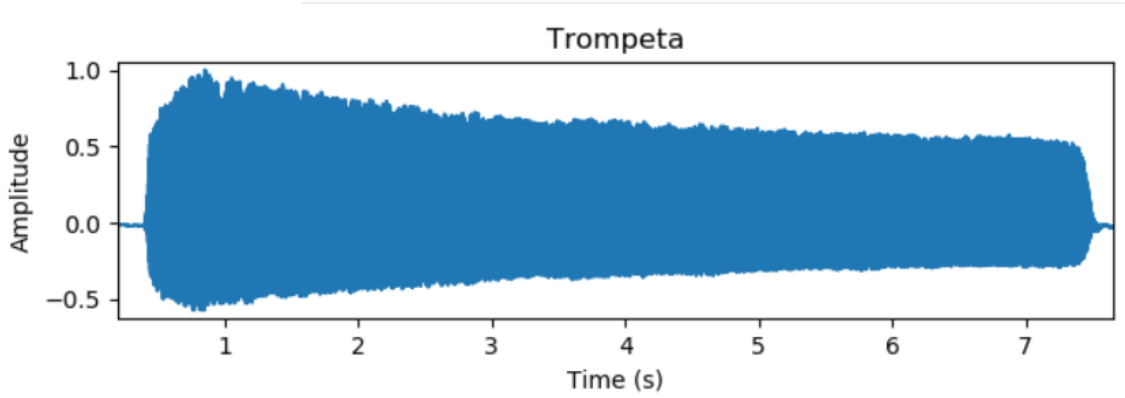

FIGURA 3.9: Envolvente obtenida de la trompeta.

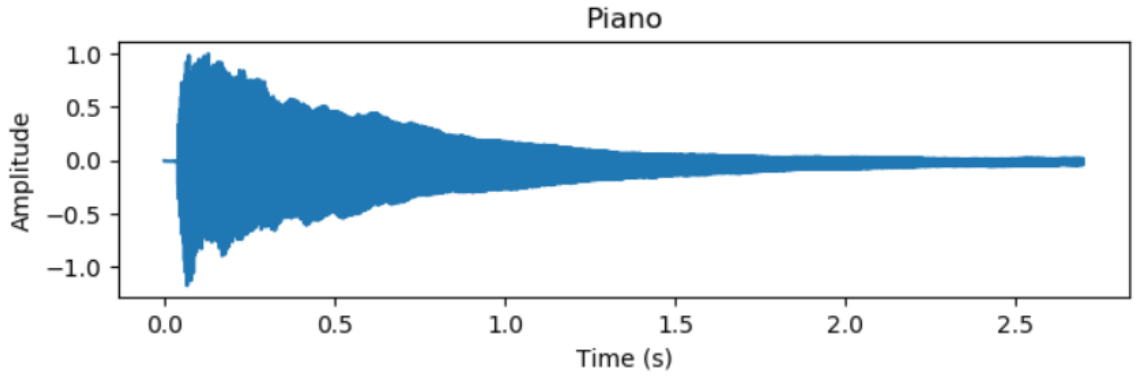

FIGURA 3.10: Envolvente obtenida del piano.

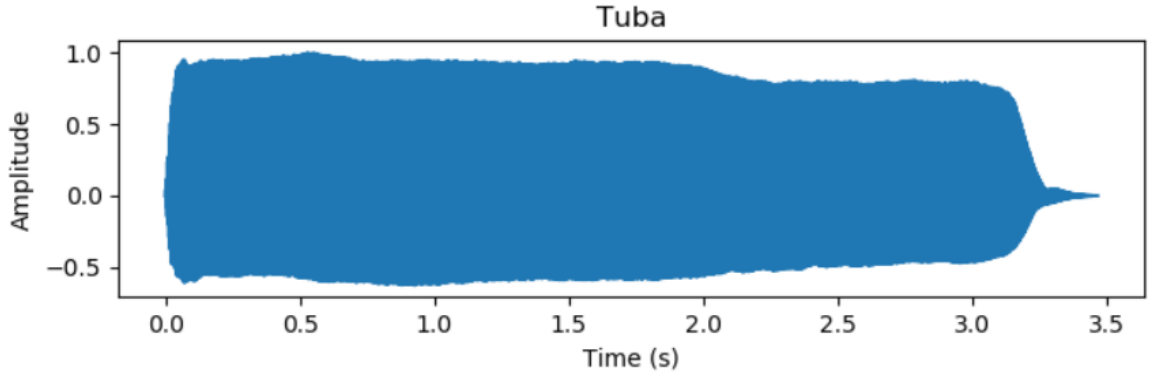

FIGURA 3.11: Envolvente obtenida de la tuba.

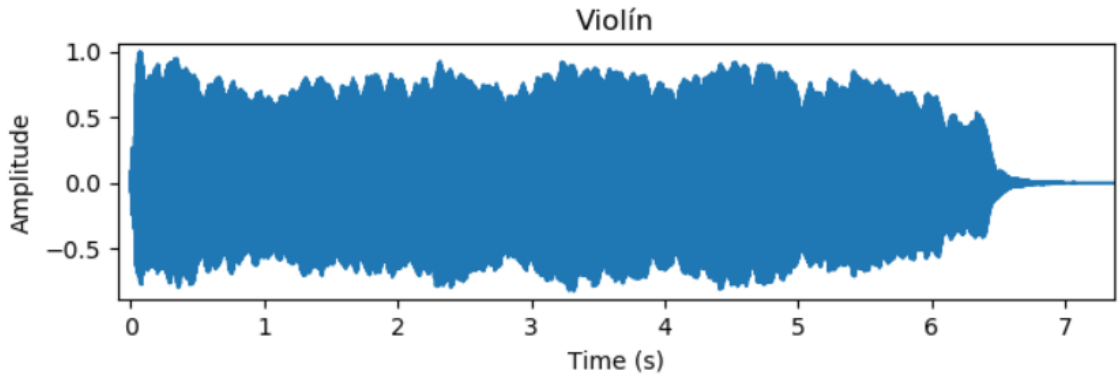

FIGURA 3.12: Envolvente obtenida del violín.

Tras obtener las distintas señales se comprueba que todas responden en mayor o menor medida a los tipos de envolventes introducidos en la sección anterior. Dependiendo de como se interpreten los gráficos pueden categorizarse en ADSR, ASR y AR. Las que presentan las cuatro etapas (attack, decay, sustain y release) según la interpretación tomada, son la del contra-bajo, la de la trompeta y la del violín (figuras 3.6, 3.9 y 3.12 respectivamente). Por su parte, el acordeón, el fagot, el oboe, el trombón y la tuba (figuras 3.4, 3.5, 3.7, 3.8 y 3.11 respectivamente) presentan señales mejor adaptadas a una ASR, ya que la etapa de decay no se evidencia o es despreciable. El pico inicial observado el fagot podría deberse a la suma de las distintas envolventes de sus parciales por lo cual se interpreta como parte del attack. Por último, en el piano (figura 3.10) solo se presentan las etapas de attack y release, por lo cual corresponde a la envolvente AR.

3.3. Síntesis

Tras haber obtenido el análisis temporal de las amplitudes de distintos instrumentos, se seleccionaron cuatros de ellos para efectuar su síntesis. Estos son el violín, el oboe, la trompeta y el acordeón.

Independientemente del instrumento se realizó el mismo proceso. En primer lugar se realizó un análisis espectral de las señales para observar su contenido armónico -utilizando notas C4 (261,63Hz) de las colecciones de samples mencionadas anteriormente-. Debido a que la señal varía en el tiempo, la mejor opción es graficar un espectrograma.

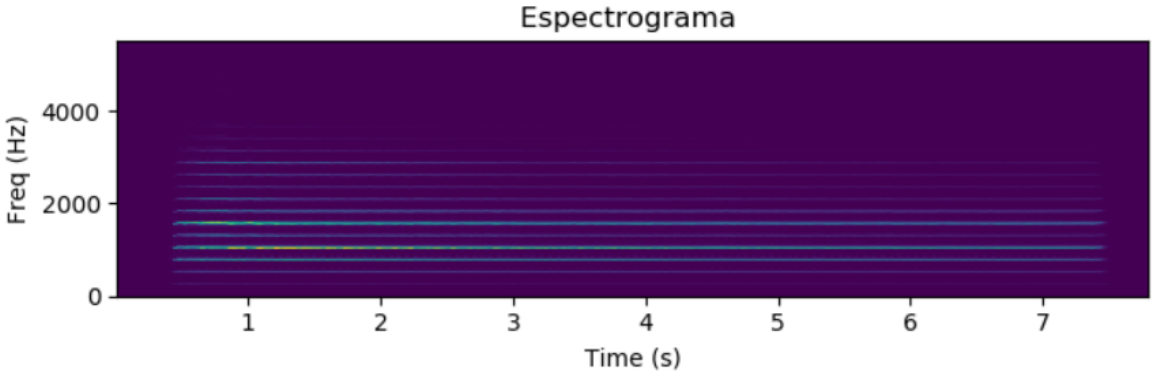

FIGURA 3.13: Espectrograma del sample C4 de la trompeta.

En la figura 3.13 se observa el espectrograma realizado a la señal obtenida de la trompeta. El dato de interés es que, aunque su magnitud varíe, el contenido espectral se mantiene posicionado en las mismas frecuencias a través del tiempo, como era de esperar. De esta manera, obteniendo el análisis espectral de un instante cualquiera, se pueden conseguir con mayor precisión la ubicación de estas frecuencias.

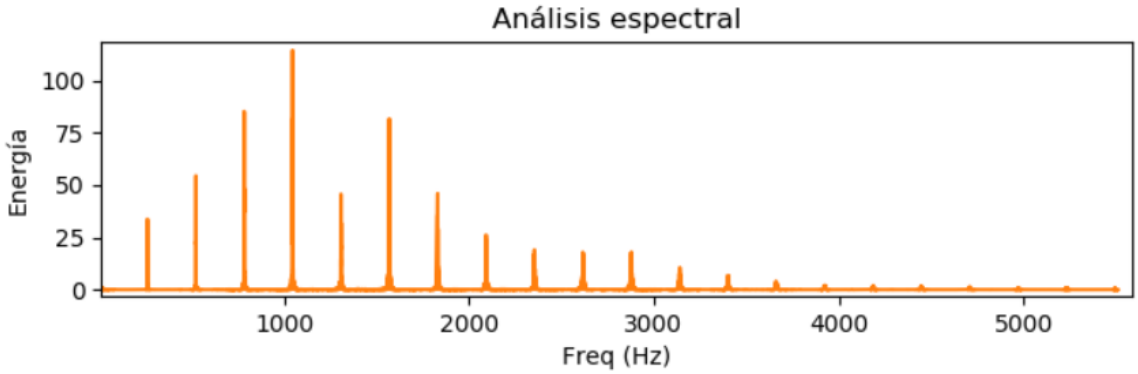

FIGURA 3.14: Análisis espectral del sample C4 de la trompeta.

A partir del espectro observado en la figura 3.14 se pueden distinguir las distintas frecuencias de los parciales de la señal resultante. Estas son cercanas a múltiplos de la frecuencia fundamental de la nota (261,63Hz), con una pequeña variación debido al corrimiento aleatorio.

Una vez obtenidas las frecuencias donde se ubican, se buscó obtener la envolvente de amplitud de cada uno de los parciales.

Como se mencionó, el análisis realizado para los distintos instrumentos es equivalente. En las figuras 3.15 y 3.16 se observan el espectrograma y el análisis espectral, respectivamente, del sample de C4 del oboe. Si bien en este caso el espectrograma muestra más variaciones, las conclusiones son similares y se obtienen las frecuencias de los parciales mediante el análisis del espectro.

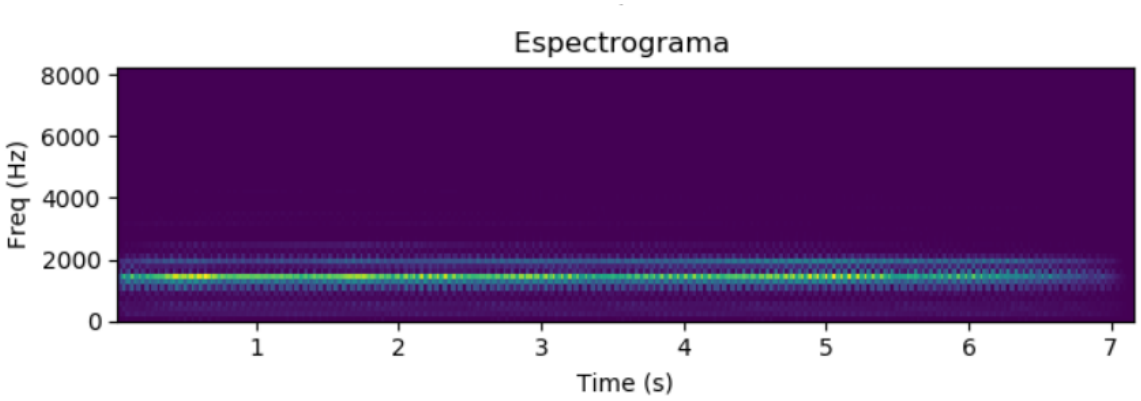

FIGURA 3.15: Espectrograma del sample C4 del oboe.

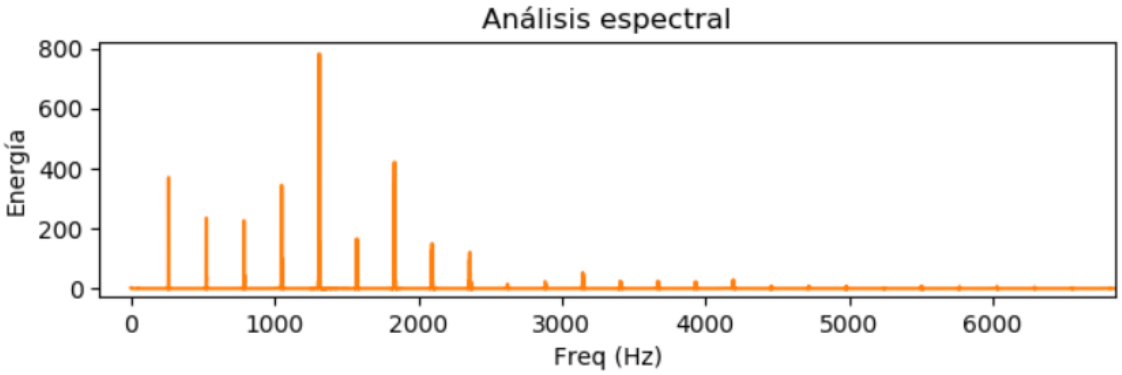

FIGURA 3.16: Análisis espectral del sample C4 del oboe.

Tras obtener las frecuencias de los parciales a sintetizar, la señal original fue filtrada alrededor de dichos valores, obteniendo así, sus variaciones en amplitud alrededor del tiempo. De aquí es de donde se obtienen los parámetros necesarios para sintetizar los instrumentos aplicando una envolvente de las mencionadas para cada uno de sus parciales.

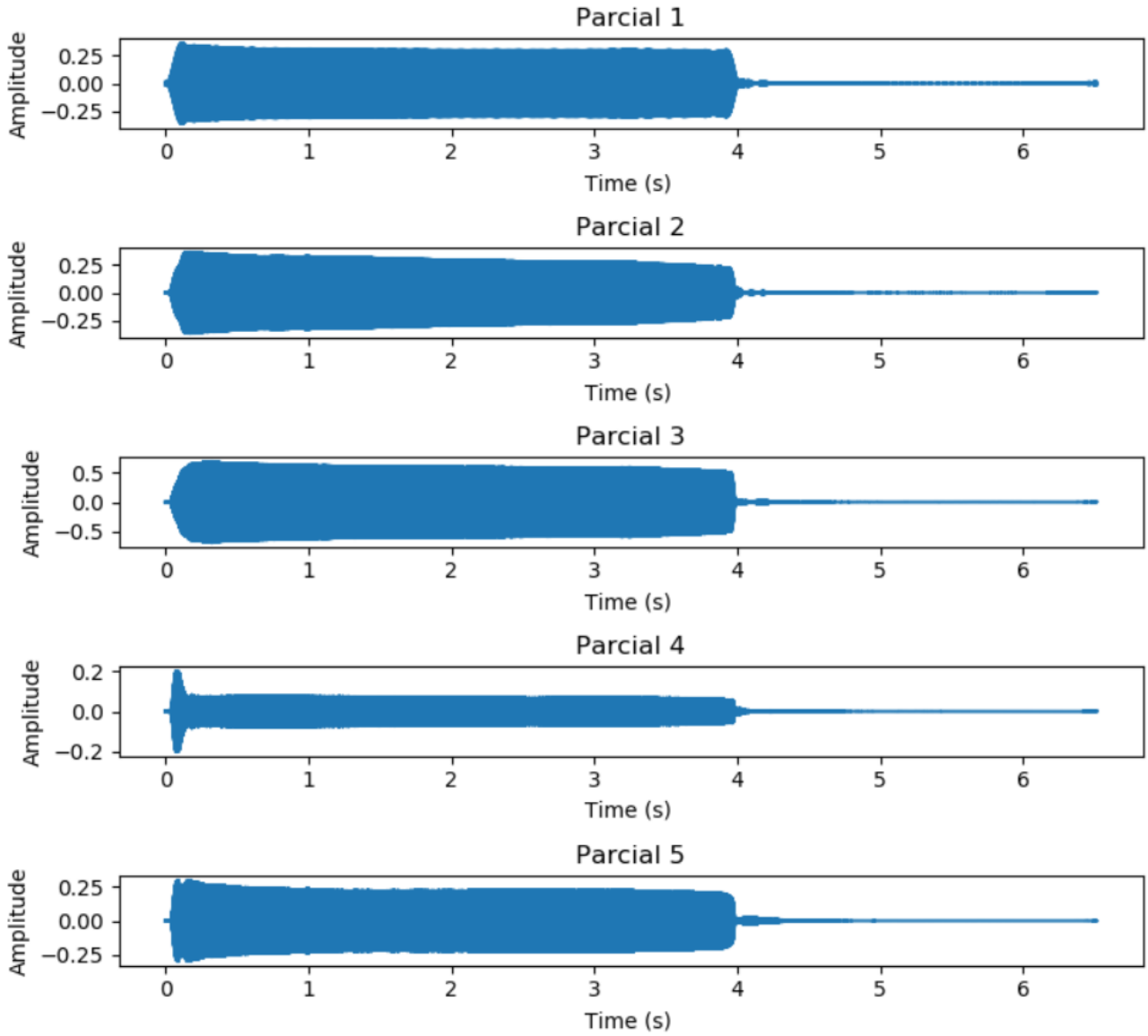

FIGURA 3.17: Señal de los parciales del acordeón.

En la figura 3.17 se muestran las señales de los parciales de la muestra del acordeón de la nota C4. Se puede apreciar que las variaciones en su amplitud respetan los tipos de envolventes mencionados en la sección previa.

A fin de sintetizar los instrumentos, las señales de cada uno de los parciales cuya amplitud no sea despreciable se aproximan a las envolventes registrando ciertos puntos de interés. Para aquellos que se puedan adaptar a una envolvente ADSR, se deben registrar los tiempos de inicio (alrededor de cero) y de la anulación del sonido junto a las duraciones de las etapas de attack, decay, sustain y release y junto a las amplitudes en los momentos de cambio de etapa. Si bien algunos parciales se asimilan más a una ASR que una ADSR, por lo que requerirían el registro de menos puntos, a modo de simplificar el cómputo e incrementar la precisión, siempre se registran la misma cantidad de puntos. En estos casos, se toman dos puntos de la etapa de attack y un punto del resto de las

etapas.

Tras efectuar la implementación de envolventes de alrededor de 10 parciales por cada instrumento, se procedió a sumar las señales a fin de obtener la señal de interés en el tono y el instrumento buscado. En primer lugar se realizó con una frecuencia de 261,63Hz para poder comparar el sonido resultante con el original del sample de la nota. Los resultados de realizar este proceso fueron similares a las señales de las muestras originales. A continuación se incluyen algunos ejemplos:

Muestra de acordeón Acordeón sintetizado
Muestra de trompeta Trompeta sintetizada

Sin embargo, no es preciso concluir que los parciales aportan la misma cantidad de energía independientemente de la frecuencia de la nota. Por lo contrario, los parciales tienden a tomar menores valores a medida que aumenta el pitch, debido a la ley de caída de los parciales. Para mostrar esto, a continuación se exponen espectrogramas de muestras de distinta frecuencia de la misma trompeta.

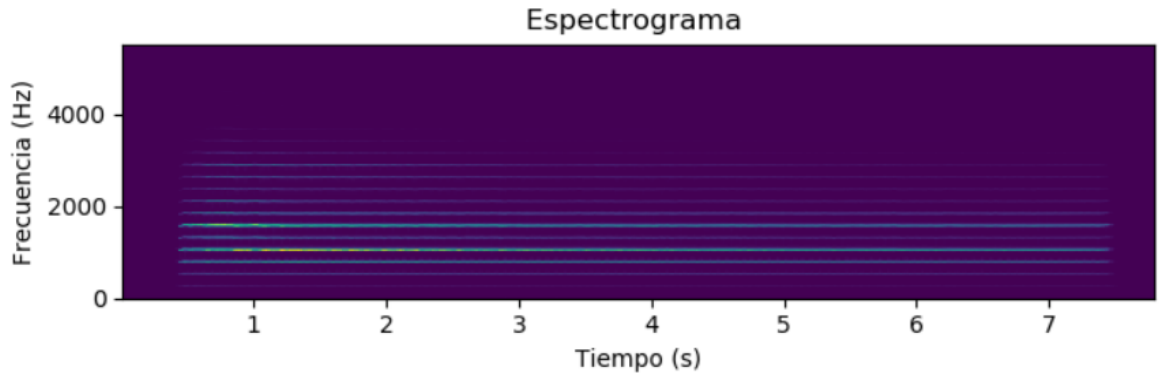

FIGURA 3.18: Espectrograma del sample C4 de la trompeta.

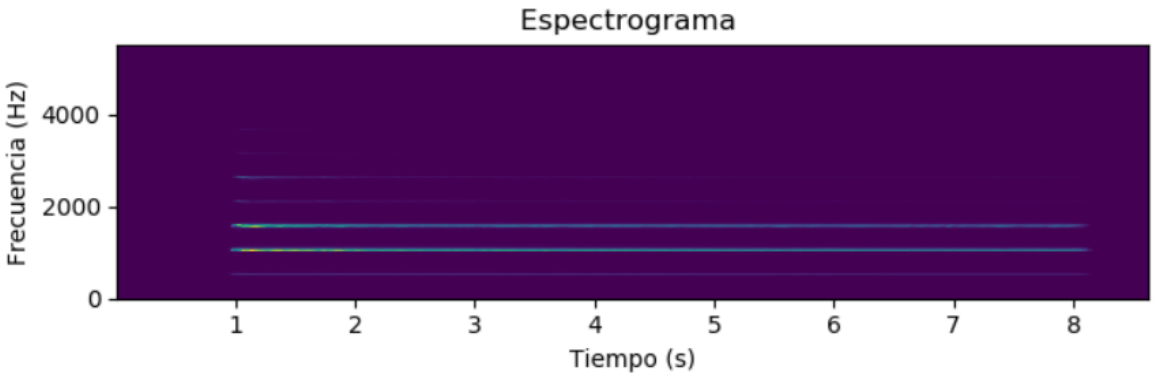

FIGURA 3.19: Espectrograma del sample C5 de la trompeta.

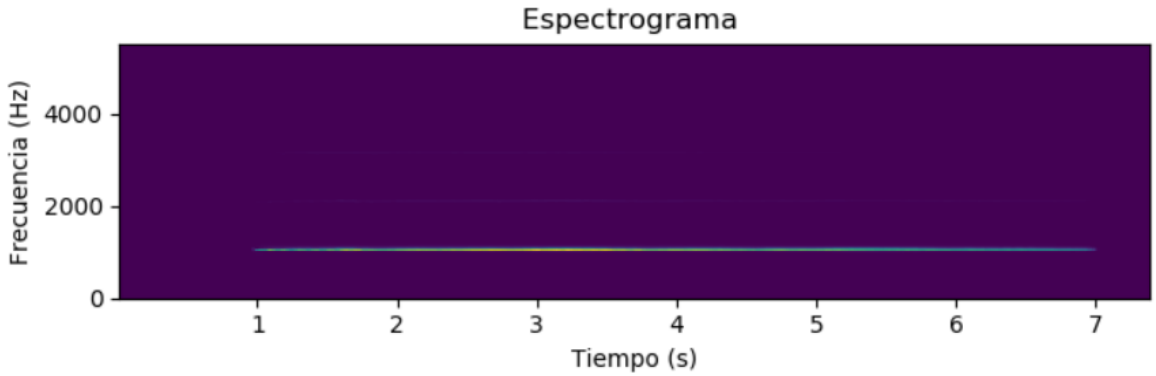

FIGURA 3.20: Espectrograma del sample C6 de la trompeta.

La figura 3.18 corresponde al sample de la nota C4 (261.63Hz) de la trompeta, la 3.19 lo hace al sample de la nota C5 (532,25Hz), mientras que la figura 3.20 corresponde al sample de la nota C6 (1046.5Hz) también de la trompeta. Es claramente apreciable que el contenido armónico producto de los parciales disminuye a medida que aumenta el pitch de la nota. Para contrarrestar este efecto y obtener una mejora en la síntesis de los instrumentos, se dividió el rango de frecuencias en dos alrededor de los 500Hz y para cada uno de estos se analizó un sample representativo. Así se lograron sintetizar notas de alta frecuencia con mayor precisión. Por ejemplo:

Muestra de oboe ($A\#5$) Oboe sintetizado ($A\#5$)
Muestra de trompeta (G5) Trompeta sintetizada (G5)

Hasta este punto, se trabajó con las frecuencias de los parciales ubicadas con un corrimiento del respectivo armónico dado por las muestras obtenidas. Si las señales contaran únicamente con contenido armónico, el audio resultante no sonaría real. Debido a que este corrimiento es aleatorio, se analizó si se obtendrían mejoras ubicando a los parciales en las frecuencias de los armónicos y luego realizando un corrimiento dentro de un rango determinado. Se realizó lo mencionado variando el corrimiento porcentual y se obtuvieron distintos resultados. Con un corrimiento aleatorio con máximo mayor al 10 % del valor frecuencial del armónico, las señales obtenidas distaron mucho de lo que se esperaría escuchar. Luego, se disminuyó este rango y sólo con valores muy cercanos a los armónicos (alrededor del 1 % de corrimiento), el sonido logrado fue similar. Sin embargo, en ninguno de los casos se logró alcanzar la similitud obtenida aplicando los corrimientos adquiridos de las muestras de audio de los instrumentos. Por tanto, se decidió seguir utilizando los corrimientos registrados a partir de las muestras de instrumentos aislados.

Para apreciar lo obtenido, a continuación se introduce un hipervínculo a un audio del resultado de la síntesis de 4 tracks de un fragmento de la melodía de la película Star Wars con los cuatro instrumentos realizados mediante síntesis aditiva (trompeta, violín, oboe y acordeón):

Melodía de Star Wars mediante síntesis aditiva (Enlace)

3.4. Conclusión

A modo de conclusión, se logró sintetizar mediante el método de síntesis aditiva cuatro instrumentos. Estos son violín, trompeta, acordeón y oboe.

El proceso realizado para todos ellos fue similar. En primer lugar se obtuvieron muestras de distintas notas de los instrumentos aislados. Se filtró cada una de estas muestras alrededor de cada uno de los parciales - sólo de aquellos cuyo contenido energético no fuera despreciable -, obteniendo así sus envolventes de amplitud. Estas fueron aproximadas a envolventes ADSR o ASR según el caso, que permitieran implementarse en el programa realizado. Así, mediante la suma de los parciales se obtienen las notas buscadas para cada uno de los instrumentos. Además, se tuvo en cuenta el corrimiento aleatorio en frecuencia de los parciales y la ley de caída de los parciales, analizando cual sería el mejor método a seguir para mejorar los resultados en cada caso.

Con respecto a la semejanza de los instrumentos sintetizados con los reales, a medida que la frecuencia es más distante de aquellas de las cuales se obtuvieron los parámetros, la similitud baja. Por otro lado, la aproximación de las envolventes de los parciales registrando poca cantidad de puntos, hizo perder cierto efecto natural de los instrumentos, como las vibraciones del sonido en el violín. Sin embargo, los resultados obtenidos fueron satisfactorios, logrando sonidos similares a los esperados.

4. Síntesis de Karplus Strong

Se utilizó esta síntesis para obtener tanto el sonido de una guitarra eléctrica como de platillos. Es importante destacar que en el caso de esta síntesis, no se utiliza el mismo algoritmo para ambos instrumentos. El algoritmo de Karplus-Strong es un algoritmo simple, que surge de un diagrama en bloques que será mostrado luego. Este algoritmo básico es empleado para la guitarra. Pero se le deben agregar unas modificaciones para obtener el sonido de un platillo.

4.1. Algoritmo básico de Karplus-Strong

A continuación se observa el diagrama en bloques del algoritmo básico de Karplus-Strong, en la figura 4.10.

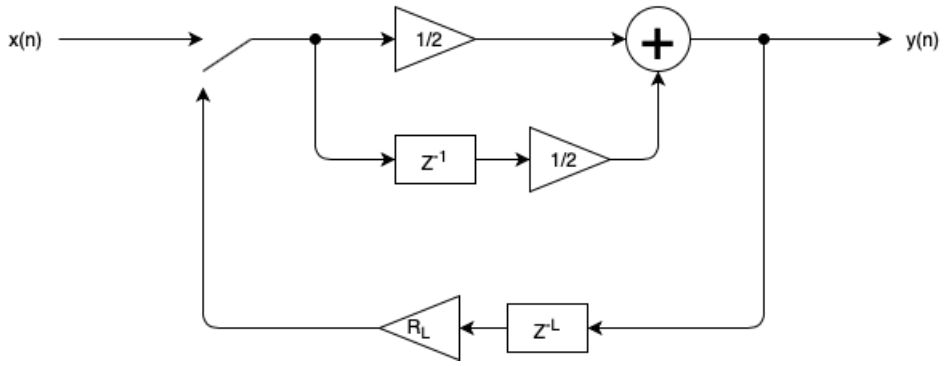

FIGURA 4.1: Diagrama en bloques de Karplus-Strong básico.

El diagrama anterior está dado por la siguiente expresión:

$$y(n) = \begin{cases} \frac{1}{2}\left[x(n) + x(n-1)\right] & n < L \\ \frac{R_L}{2}\left[y(n-L) + y(n-L-1)\right] & n \geq L \end{cases} \tag{4.1}$$

Es importante mencionar que para el sonido inicial de una nota el diagrama funciona con la llave conectada a x(n). Esto se debe a que se necesitan inicialmente L valores en y(n) con la finalidad de luego pasar la llave a su otra posición y poder emplear el retardo de L muestras sobre y(n). A continuación se detalla lo que representa cada parámetro del diagrama.

L: Es un retardo cuyo valor surge de la frecuencia fundamental de la nota que se desea escuchar. La frecuencia fundamental de la nota está dada por la expresión $f = \frac{f_s}{L+\frac{1}{2}}$; siendo f_s la frecuencia de muestreo. Por lo tanto, dada la frecuencia f de una nota en particular, se obtiene el retardo L como $L = \frac{f_s}{f} - \frac{1}{2}$. Para poder emplear este retardo de L muestras, y(n) se carga inicialmente con aquello proveniente de la x(n), que luego se explicará qué valores se eligen para x(n).

R_L : Este valor es una ganancia aplicada al retardo de las L muestras de y(n), permitiendo ajustar el tiempo de caída de la nota.

x(n): El sistema cuenta con esta señal de entrada únicamente al principio ya que se usa para llenar los primeros L valores de y(n) que son luego necesarios para el retardo de L muestras que contiene el sistema. La entrada x(n) definirá el sonido inicial de la nota (el "golpe"de sonido más fuerte) y también influirá en el sonido de cuando decae ya que todos los valores de y(n) terminan dependiendo de sus valores iniciales. Lo que se suele hacer es que x(n) contenga valores aleatorios para simular un tipo de ruido. Gracias a esto se logra que al reproducir dos veces la misma nota y con la misma duración, el sonido no sea exactamente el mismo, ya que en ambos casos esto depende de los valores de la x(n). Así es como este algoritmo permite generar un sonido más real y no un mismo sonido que suena siempre igual.

4.1.1. Función transferencia del sistema

H(Z) del diagrama con llave conectada a x(n)

Primero se comienza analizando la transferencia en el caso en el que la llave está colocada a la entrada x(n). Es decir, la realimentación de retardo L y ganancia R_L no son considerados para esta primera parte. De esta forma, la salida en el instante n es el promedio de la entrada en el instante n y en el instante n-1.

$$y_1(n) = \frac{1}{2}\left[x(n) + x(n-1)\right] \tag{4.2}$$

Aplicando transformada Z a la ecuación 4.2, se obtiene:

$$Y_1(Z) = \frac{1}{2}\left[X(Z) + X(Z)\cdot Z^{-1}\right] = X(Z)\left[\frac{1}{2} + \frac{1}{2}\cdot Z^{-1}\right] \tag{4.3}$$

Y por lo tanto se obtiene que la $H(Z)$ está dada por la expresi'on 4.4:

$$H_1(Z) = \frac{1}{2}\left[1 + Z^{-1}\right] = \frac{1}{2}\cdot\frac{Z+1}{Z} \tag{4.4}$$

H(Z) del sistema completo

A continuación se obtiene la transferencia para el esquema completo. Es importante remarcar que para esto se considera que en lugar de una llave a la entrada hay un sumador. Previamente se analizó cuál sería la H(Z) para el caso en el que se cargan los valores L iniciales de y(n). Sin embargo, dado que en realidad el sistema luego queda conectado con la llave al realimentador y es un sistema sin alimentación pero con condiciones iniciales, de esta forma NO habría manera de obtener la H(Z). Por lo tanto, se plantea un diagrama con un sumador en lugar de la llave y una x(n) que toma valores únicamente hasta la muestra n=L y luego se hace cero.

$$y(n) = \frac{1}{2}\left[x(n) + x(n-1)\right] + \frac{R}{2}\left[y(n-L) + y(n-L-1)\right] \tag{4.5}$$

Aplicando transformada Z:

$$Y(Z) = X(Z)\cdot\frac{1}{2}\left[1 + \cdot Z^{-1}\right] + Y(Z)\cdot\frac{R}{2}\cdot\left[Z^{-L} + Z^{-L-1}\right] \tag{4.6}$$

Y entonces la $H(Z)$ está dada por:

$$H(Z) = \frac{\frac{1}{2} \cdot (1 + Z^{-1})}{1 - \frac{R}{2}(Z^{-L} + Z^{-L-1})} \tag{4.7}$$

Respuesta en frecuencia

La respuesta en frecuencia del sistema correspondiente al Karplus-Strong básico está dada por:

$$H\left(e^{j\omega/\omega_s}\right) = \frac{\frac{1}{2}(1 + e^{-j/\omega_s})}{1 - \frac{R}{2}e^{-j\omega L/\omega_s} - \frac{R}{2}e^{-j\omega(L+1)/\omega_s}} \tag{4.8}$$

Aclaración importante: El modelo de Karplus-Strong no tiene una respuesta en frecuencia genérica para todas las notas. Cada nota tiene una frecuencia fundamental que la caracteriza, y el retardo L del modelo, DEPENDE de la frecuencia fundamental de la nota. Por lo tanto, se decidió mostrar a continuación la respuesta en frecuencia para tres notas distintas: f=220Hz, f=440Hz y f=880Hz. Se puede ver en las tres imagenes este cambio mencionado. Sin embargo, hay caracterśticas en común. Independientemente de la frecuencia fundamental de la nota que se desea escuhar, el sistema se comporta como un filtro comb (se observa en los picos) y a su vez pasabajos (se atenúan las altas frecuencias). Las imagenes obtenidas a continuación son la FFT del arreglo de muestras obtenido para tres notas distintas con el algoritmo implementado.

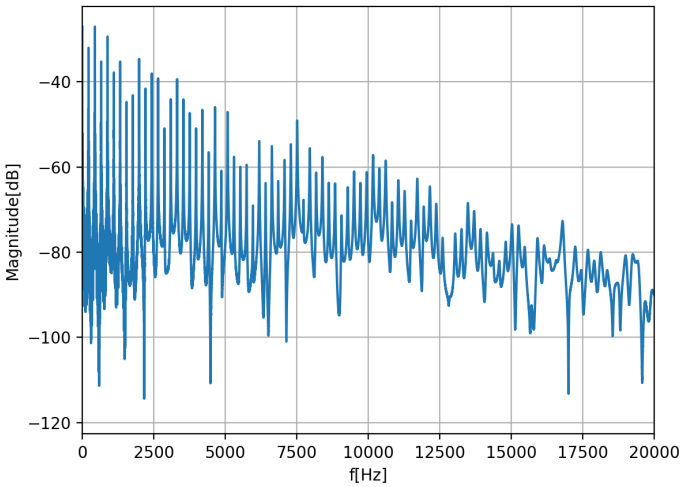

FIGURA 4.2: Respuesta en frecuencia para nota de f=220Hz, L=199.

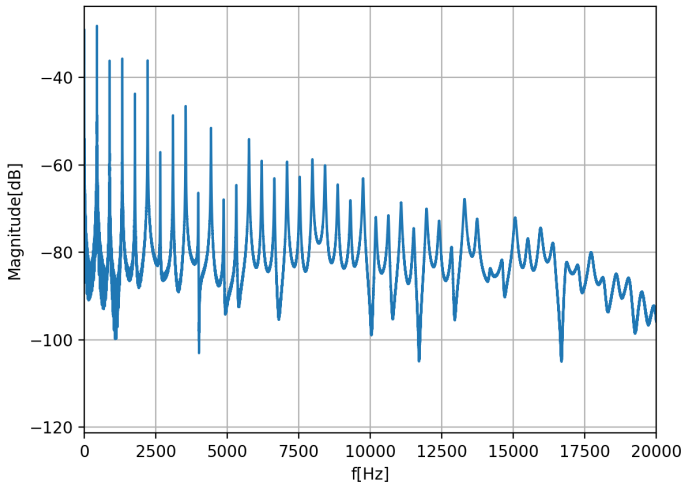

FIGURA 4.3: Respuesta en frecuencia para nota de f=440Hz, L=99.

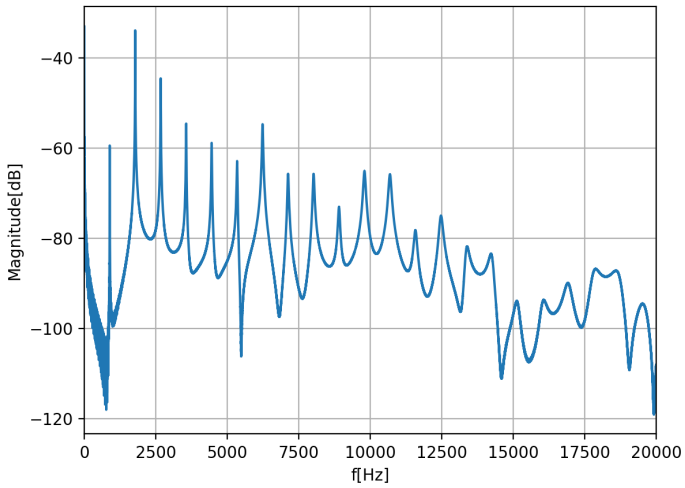

FIGURA 4.4: Respuesta en frecuencia para nota de f=880Hz, L=49.

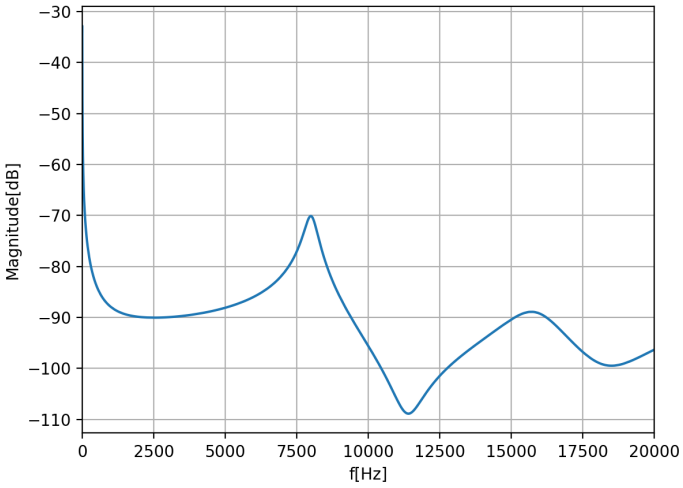

FIGURA 4.5: Respuesta en frecuencia para nota de f=7040Hz.

4.1.2. Diagramas de polos y ceros

Como se mencionó previamente, la H(z) va a depender de la frecuencia fundamental de la nota que se desea escuchar, ya que es lo que determina la L del retardo en el sistema. Por lo tanto, el diagrama de polos y ceros también cambia al cambiar el L. A continuación se muestran los diagramas de polos y ceros correspondientes a L=99 (figura 4.6) y a L=49 (FIGURA 4.7), respectivamente. En ellas se observa que para el caso de mayor L, aumenta la cantidad de polos en el diagrama, dado que aumenta el orden del denominador de la H(Z). A su vez, esto coincide con lo visto en los gráficos 4.3 y 4.4 donde en la de mayor L los picos del comb se ven mas próximos entre sí

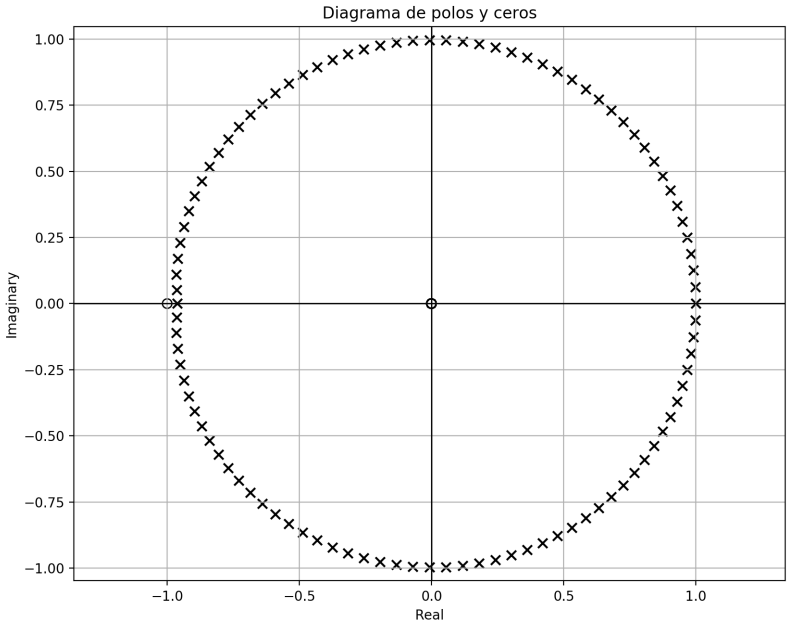

FIGURA 4.6: Diagrama de polos y ceros, para L=99

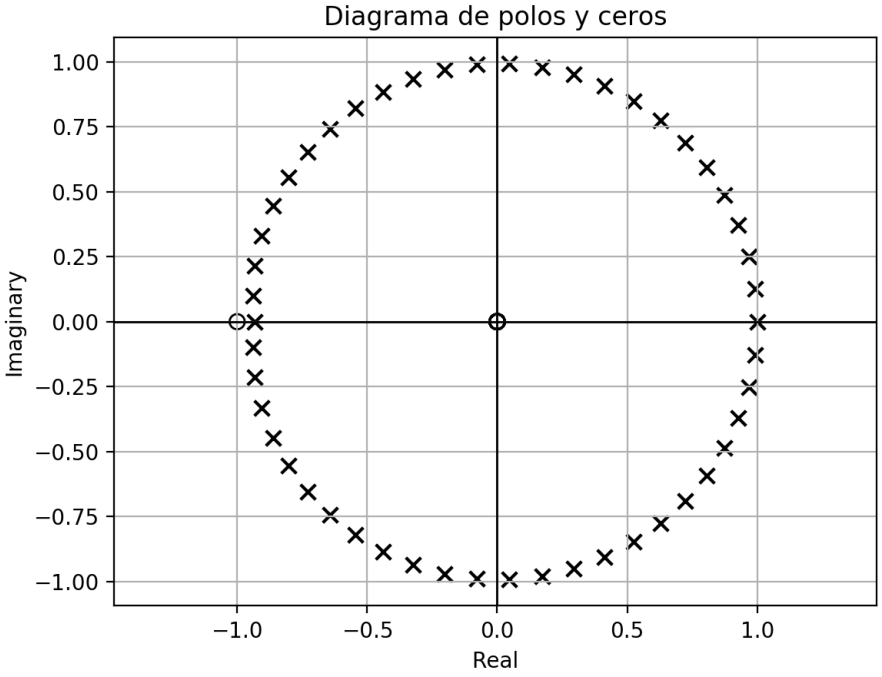

FIGURA 4.7: Diagrama de polos y ceros, para L=49

4.1.3. Relación entre R y la estabilidad del sistema

El valor de R corresponde al módulo de los polos del sistema. Por lo tanto, considerando que se trata de un sistema causal, y que para que el mismo sea estable la icrcunferencia de radio unitario debe pertencer a la región de convergencia, R debe ser menor a 1. Para la implementación de la síntesis se tomó $R = 0,999$. Además se intentó analizar la estabilidad con Jury-Marden. El problema para dicho análisis fue que la H(z) depende de L, y el L depende a su vez de la frecuencia fundamental de la nota, por lo que la H(z) sería distinta y de distinto grado según la nota que se quiere escuchar. Analizar los valores de R para los cuales el sistema sea estable empleando Jury-Marden no fue posible debido a eso. Se comenzó a dar valores a L (partiendo de L=1) y el mismo se fue aumentando para ver si se encontraba algún patrón, pero se encontró que siempre se cumplía la necesidad de que L sea menor a 1.

4.1.4. Comportamiento en el dominio del tiempo

El sistema es alimentado con x(n) unicamente al principio con la finalidad de obtener los L valores iniciales de y(n). Por lo tanto, se puede pensar al modelo como un sistema sin alimentación que presenta condiciones iniciales. Tratándose de un sistema estable, con el transcurso del tiempo la amplitud disminuye ya que la realimentación cumple con el criterio de Barkhausen. Cuanto mayor es la frecuencia fundamental de la nota, menores son los valores iniciales necesarios en y(n) ya que disminuye L. Esto hace que la amplitud caiga mas rapido al aumentar la frecuencia, haciendo que el sonido dure menos tiempo. A continuación se musetra como el sonido decae y se ve que para menores frecuencias tarda mas.

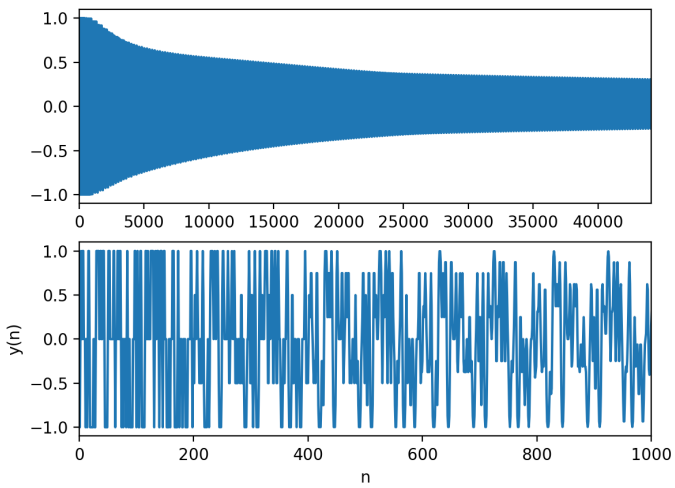

FIGURA 4.8: f=220Hz, L=199.

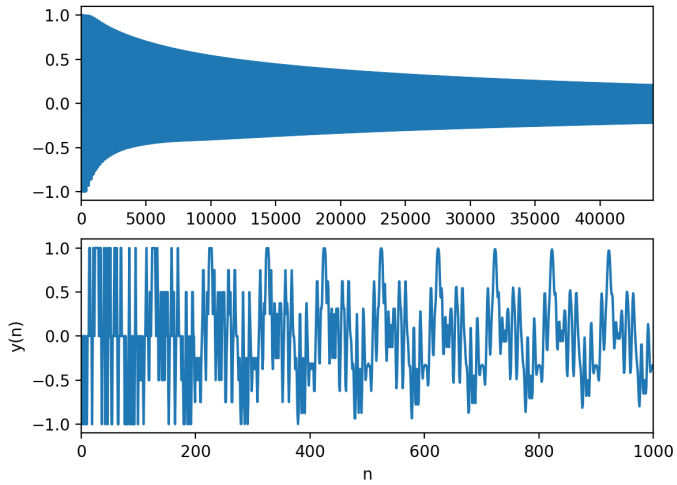

FIGURA 4.9: f=440Hz, L=99.

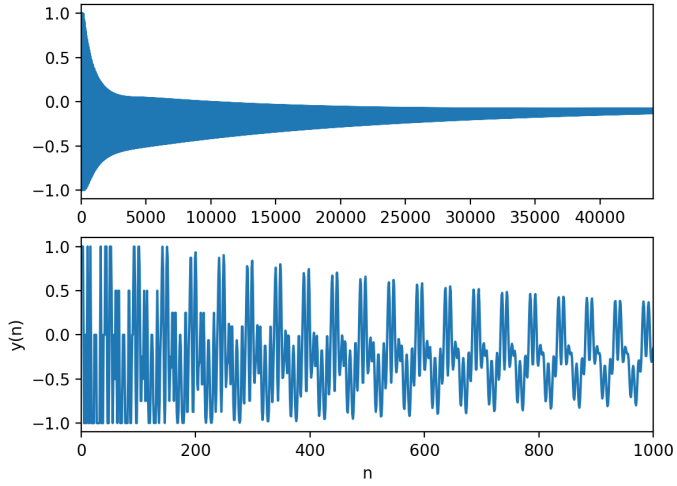

FIGURA 4.10: f=880Hz, L=49.

4.1.5. Influencia de la distribución de ruido

Se comparó la respuesta del sistema al cambiar la distribución de ruido de $x(n)$. Es importante remarcar que este ruido estará presente únicamente al comienzo del sonido, ya que x(n) toma valores de ruido solo al principio, para darle valores iniciales a los primeros L valores de $y(n)$ y luego la salida decae a partir de estos valores iniciales. Es decir, $x(n)$ toma valores de ruido para $n < L$ y luego se hace cero.

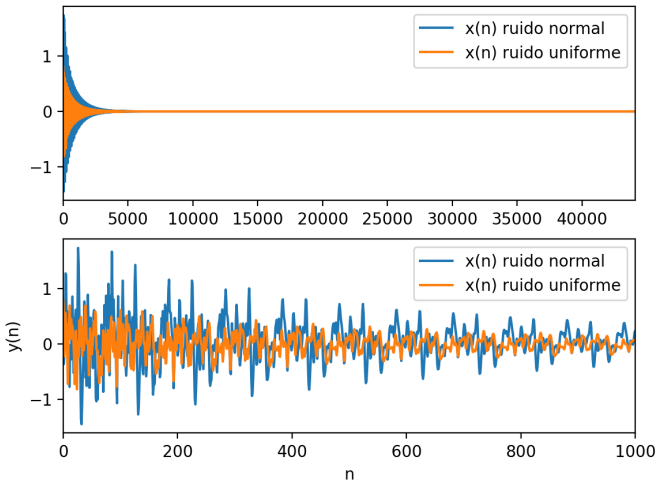

FIGURA 4.11: Comparación de y(n) dependiendo de la distribución de ruido de x(n).

En la imagen 4.11 se compara lo que sucede como respuesta a un ruido a la entrada con distribución uniforme entre -1 y 1 y como respuesta a un ruido con distribución normal. Esto se hizo para $L = 99$, que es el caso con el que se obtiene la nota LA cuya frecuencia fundamental es de $440Hz$. Dado que la entrada es ruido, se lo graficó distintas veces y se observó siempre el mismo comportamiento: Dado que en el caso de la distribución uniforme los valores de x(n) van a estar entre -1 y 1, la salida del sistema obtiene valores iniciales acotados en dicho rango, meintras que en el caso del ruido con distribución normal la salida puede obtener valores iniciales de mayor módulo. Esto hace que para el caso de distribución normal, el sonido comience más fuerte que para el caso con distribución uniforme y por lo tanto tarda un poco más en caer el sonido de la nota (tarda más en *apagarse*).

4.1.6. Problema que presenta el modelo

El modelo de Karplus-Strong mostrado en la figura 4.10, presenta un inconveniente debido a cómo está definida la frecuencia de la nota. Hemos visto que la misma define el la cantidad de muestras del retardo, ya que $f = \frac{f_s}{L+\frac{1}{2}}$. Al despejar L, se obtiene $L = \frac{f_s}{f} - \frac{1}{2}$. El inconveniente es que L debe ser un número entero, ya que es una cantidad de muestras de retardo, y esta última expresión no garantiza que esto se cumpla. Por lo tanto, el L empleado con la finalidad de obtener el sonido correspondiente a una frecuencia determinada, resultará ser el sonido resultante de otra frecuencia en los casos en los que no se cumpla que el L sea entero, ya que se lo redondea a un número entero. El error es mayor cuanto más aguda es la nota que se desea sintetizar. Es decir, cuanto mayor es su frecuencia. En la siguiente subsección (Primera modificación) se presenta una solución a este problema.

4.2. Modificaciones al algoritmo de Karplus-Strong

4.2.1. Primera modificación: para guitarra bien afinada

Esta modificación surge de la necesidad de obtener un sonido correspondiente al valor exacto de la frecuencia fundamental deseada Previamente, se mencionó que el sonido que se genera para la frecuencia fundamental de una nota no llega a representar correctamente esa frecuencia debido que para eso se necesita un retardo cuya cantidad de muestras está dada por L, la cual matemáticamente podría requerir no ser un número entero para obtener la frequencia esperada. Se había mencionado que el valor de la longitud de muestas del retardo, L, matemáticamente está dado por:

$$L = \frac{f_s}{f} - \frac{1}{2} \tag{4.9}$$

siendo f_s la frecuencia de muestreo y f la frecuencia fundamental de la nota que se desea escuchar. Pero en esta expresión el L no necesariamente toma un valor entero y para realizar esto es necesario que el número de muestras del retardo sea entero. Entonces llamamos L_{int} al valor entero que se usará en el programa, de forma que:

$$L = L_{int} + \beta \implies L_{int} + \beta = \frac{f_s}{f} - \frac{1}{2} \implies \beta = \frac{f_s}{f} - \frac{1}{2} - L_{int} \tag{4.10}$$

Esto nos indica que si hay forma de obtener un β que compense la diferencia entre el valor entero L_{int} y el L que realmente se necesita, se escucharía un sonido con la frecuencia fundamental correcta. Este β tiene que surgir de algo que no sea la longitud del retardo en la realimentación. Una forma para obtener este β es agregando un all-pass filter, ya que se necesita compensar el retardo de fase sin alterar la ganancia del filtro original del modelo de Karplus-Strong. El agregado de este filtro se muestra en la figura 4.12 a continuación:

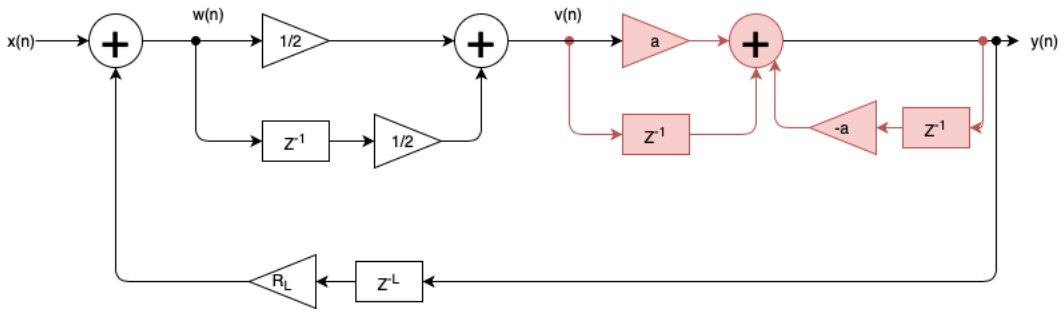

FIGURA 4.12: Diagrama en bloques de Karplus-Strong modificado, agregando un all pass filter para obtener f deseada.

A partir del diagrama de la figura anterior, se plantean las siguientes ecuaciones que representan

al sistema:

$$\begin{cases} \omega(n) = x(n) + R^L \cdot y(n-L) \\ v(n) = 0,5 \cdot \omega(n) + 0,5 \cdot \omega(n-1) \\ y(n) = a \cdot v(n) + v(n-1) - a \cdot y(n-1) \\ a = 1 - \frac{\beta}{1+\beta} \end{cases} \tag{4.11}$$

Así es como se obtiene la expresión de $y(n)$ que define al sistema:

$$\begin{aligned} y(n) = & \frac{a}{2}\Big(x(n) + R^L y(n-L) + x(n-1) + R^L y(n-L-1)\Big) \\ & + \frac{1}{2}\Big(x(n-1) + R^L y(n-L-1) + x(n-2) + R^L y(n-L-2)\Big) - a y(n-1) \end{aligned} \tag{4.12}$$

Guitarra: Para obtener el sonido de guitarra se implementó el algoritmo de Karplus-Strong con la primera modificación presentada. Esto se debe a que el algoritmo básico de Karplus-Strong tenía el problema de no tener una correcta afinación a altas frecuencias, debido al truncado del valor de L que se tomaba. Agregando el filtro pasa todo que se presenó en la primera modificación, se logró que, a frecuencias altas, las notas sonaran con la frecuencia correspondiente en lugar de con otra frecuencia. Se probó el sonido logrado con el algoritmo de Karplus-Strong con su primera modificación, correspondiente a las frecuencias de todas las notas de todas las octavas de una guitarra. Las mismas se muestran a continuación:

```
20  octava0 = 27.5, 29.135, 30.868
21  octava1 = 32.703, 34.648, 36.708, 38.891, 41.203, 43.654, 46.249, 48.999, 51.913, 55.000, 58.270, 61.735
22  octava2 = 65.406, 69.296, 73.416, 77.782, 82.407, 87.307, 92.499, 97.999, 103.826, 110.000, 116.541, 123.471
23  octava3 = 130.813, 138.591, 146.832, 155.563, 164.814, 174.614, 184.997, 195.998, 207.652, 220.000, 233.082, 244.942
24  octava4 = 261.626, 277.183, 293.665, 311.127, 329.628, 349.228, 369.994, 391.995, 415.305, 440.000, 466.164, 493.883
25  octava5 = 523.251, 554.365, 587.330, 622.254, 659.255, 698.456, 739.989, 783.991, 830.609, 880.000, 932.328, 987.767
26  octava6 = 1046.502, 1108.731, 1174.659, 1244.508, 1318.510, 1396.913, 1479.978, 1567.982, 1661.219, 1760.000, 1864.655, 1975.533
27  octava7 = 2093.005, 2217.461, 2349.318, 2489.016, 2637.020, 2793.826, 2959.955, 3135.963, 3322.438, 3520.000, 3729.310, 3951.066
28  octava8 = 4186.009, 4434.922, 4698.636, 4978.032, 5274.041, 5587.652, 5919.911, 6271.927, 6644.875, 7040.000
```

FIGURA 4.13: Barrido de frecuencias de notas de guitarra, en Python.

Al hacer esta pruebo, se notó que a bajas frecuencias el sonido resulta ser más metálico, pero en la segunda o tercera octava el sonido es más similar al de una guitarra acústica, si bien sigue pareciéndose al de una guitarra eléctrica. Para las frecuencias de las últimas tres octavas la duración de las notas es muy corta, ya que al aumentar la frecuencia, aumenta la velocidad con la que decae el sonido de la nota. Entre la segunda y la sexta octava, es donde mejor suena la guitarra, es decir, donde más se parece una guitarra real.

4.2.2. Segunda modificación: para instrumentos de percusión

El primer cambio que se le puede hacer al algoritmo básico de Karplus-Strong es agregar un multiplicador a la salida, el cual multiplica la salida del sistema por 1 con probabilidad b y -1 con probabilidad $1 - b$.

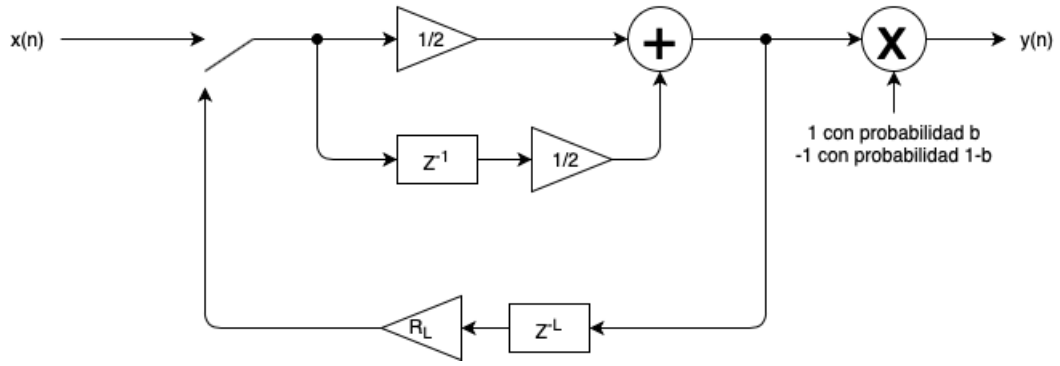

FIGURA 4.14: Diagrama en bloques de Karplus-Strong modificado, con signo aleatorio.

La salida del sistema, y(n), queda entonces dada por las siguientes expresiones:

$$y(n) = \begin{cases} y_1(n) & n < L \\ y_2(n) & n \geq L \end{cases} \tag{4.13}$$

Para $n < L$:

$$y_1(n) = \begin{cases} \frac{1}{2}\left[x(n) + x(n-1)\right] & probabilidad = b \\ -\frac{1}{2}\left[x(n) + x(n-1)\right] & probabilidad = 1 - b \end{cases} \tag{4.14}$$

Para $n \geq L$:

$$y_2(n) = \begin{cases} \frac{R_L}{2}\left[y(n-L) + y(n-L-1)\right] & probabilidad = b \\ -\frac{R_L}{2}\left[y(n-L) + y(n-L-1)\right] & probabilidad = 1 - b \end{cases} \tag{4.15}$$

Variaciones del valor de probabilidad b: Tomando $b = 1$ el algoritmo sigue siendo el básico de Karplus-Strong presentado inicialmente. Este es el caso correspondiente al de una guitarra. Si se toma $b = 0.5$, se obtiene una salida positiva o negativa, ambas con la misma probabilidad de ocurrencia, haciendo que el sonido a la salida sea más aleatorio. Así es como se obtiene el sonido de platillos de batería. Con $b = 0$, se leyó que el sonido correspondería al de un harpa, pero el mismo no fue implementado en el trabajo práctico. En este caso lo que ocurre es que todas las salidas del algoritmo básico son negadas, dado que se multiplica siempre por -1.

Desventaja y limitaciones del filtro: Con $b = 0,5$, el sonido es el de un instrumento de percusión. Se hizo un análisis del sonido para todas las frecuencias correspondientes a las distintas octavas y se notó que para las frecuencias más bajas el sonido parece ser con mas fuerza y al aumentar la frecuencia va disminuyendo la fuerza con la que alguien le pegaría a un platillo en la vida real. A altas frecuencias el sonido ya deja de parecer el de un platillo y no presenta ningún tipo de similitud ccon ningún instrumento de percusión, por lo que se decidió que el algoritmo es consistente con el instrumento hasta una frecuencia de 240Hz (tomando un rango de tolerancia).

Instrumentos de percusión: Con la finalidad de obtener el sonido de instrumentos de percusión, se implementó la segunda modificación propuesta al modelo básico de Karplus-Strong. Esta es la que cambia el signo a la salida, obteniendo un valor positivo con probabilidad b y un valor negativo con probabilidad $1 - b$. Tomando $b = 0,5$, ambos signos tienen la misma probabilidad de ocurrencia (tanto positivo como negativo), logrando que cada muestra de salida presente gran diferencia respecto a la anterior, y de forma aleatoria (lo cual a su vez hace que el sonido sea mas real ya que no es exactamente el mismo al correr el algoritmo dos veces seguidas). Se notó que el sonido corresponde al de los platillos de una batería, y cambia la fuerza del mismo al variar la frecuencia. Como se mencionó previamente, el sonido para el platillo suena bien hasta una frecuencia de aproximadamente 244Hz (terminando la tercera octava). Por lo tanto, para cumplir con que el sonido sea lindo y parecido al sonido real de los platillos, se decidió asignarle el sonido del algoritmo correspondiente a 244Hz para cualquier frecuencia mayor a esta. Este análisis fue hecho con la misma tabla de frecuencias que se usó para el análisis del sonido de la guitarra. Esta tabla es la de la imagen 4.13.

4.3. Referencias para Karplus-Strong

Los papers leidos se encuentran en la carpeta Resources/Karplus-Strong en la entrega del trabajo práctico.

5. Síntesis basada en muestras

5.1. Marco teórico

En la síntesis basada en muestras, se realizan los sonidos basados en muestras previamente tomadas de los distintos instrumentos a implementar. Ésto se logra mediante la realización de cambios en el tono de una nota (*Pitch shifting*) y modificando la duración de la misma (*Time stretching*).

Pitch Shifting

El *Pitch shifting* puede realizarse simplemente cambiando la cantidad de muestras tomadas por segundo, pero al realizar esto, también se estaría modificando el largo de la señal, y por lo tanto, la velocidad de reproducción. En la Figura 5.1 puede verse un ejemplo de una señal a la cual se le eliminan todos los valores que se encuentran en índices impares, consiguiendo así un sonido que se reproducirá 2 veces más rapido, por lo tanto cambiando la frecuencia y consecuentemente, el tono(*pitch*).

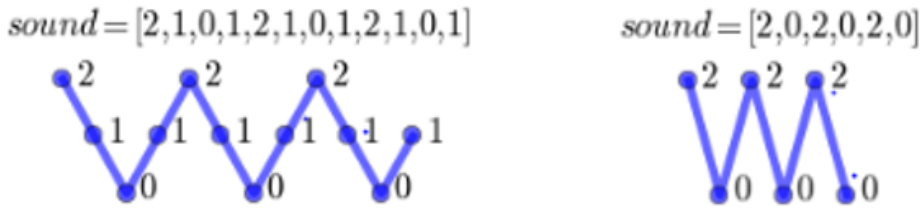

FIGURA 5.1

Realizando lo contrario, es decir, repitiendo cada valor de forma alternada armando un arreglo del doble de tamaño, se consigue el efecto opuesto al visto previamente, el sonido sera dos veces mas lento y nuevamente cambiará el tono.

Para lograr cambiar el tono sin modificar la velocidad de reproducción, se realiza primero un estiramiento del tiempo de la señal y luego se cambia la velocidad de reproducción de la señal estirada por el mismo factor.

Time stretching

Para realizar el escalamiento temporal de una señal existen diversos algoritmos, cada uno con sus ventajas y desventajas. Se decidió utilizar el algoritmo *"Phase Vocoder"* para el presente trabajo.

El *phase vocoder* trabaja en el dominio de la frecuencia, consiste en aplicarle la STFT (*Short-Time Fourier Transform*) a la señal que se desea escalar, luego, dependiendo si se desea alargar o comprimir la señal, se superponen las distintas ventanas obtenidas en mayor o menor medida. Se

calcula la diferencia de fase entre los dos bins de ventanas vecinas y se realinean en los bordes de las ventanas luego de ser superpuestas. Finalmente se normaliza la fase entre $-\pi$ y π y se aplica la ISTFT para recuperar la señal.

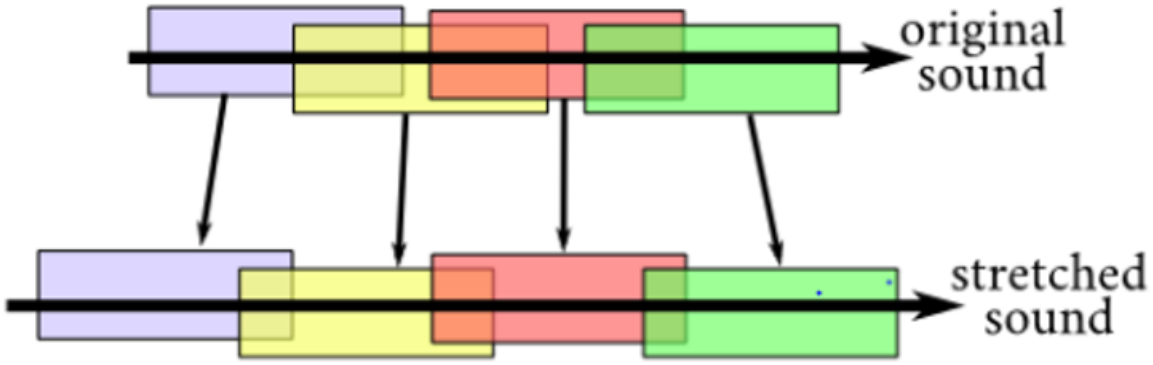

FIGURA 5.2

5.2. Puesta en práctica

Se logró realizar un sintetizador basado en muestras utilizando los conceptos explicados en el marco teórico. Como era de esperarse, implementar esta síntesis utilizando el *phase vocoder* hace que la síntesis sea lenta a comparación de otras síntesis pero se obtienen buenos resultados. Una gran ventaja de esta síntesis es la facilidad para sintetizar distintos instrumentos, ya que se realiza el mismo algoritmo para todos.

Se sintetizó un piano a partir de una muestra de 4 segundos de duración de cada nota descargada de la web, por lo cual al sintetizar un sonido con este instrumento no se realiza el *pitch shifting* sino que simplemente se estira o se comprimen las muestras para obtener el largo deseado.

Por otra parte, se incluyeron otros instrumentos con menor cantidad de muestras, por lo cual para todos ellos es necesario realizar el *pitch shifting* sobre las muestras guardadas para obtener notas de distintas frecuencias haciendo el proceso mas lento que con el piano. Estos instrumentos son: Violonchelo, viola, saxofón, mandolin, banyo, guitarra acústica, bajo eléctrico y fagot.

6. Efectos de audio

Los efectos de audio que se busco implementar son los referidos a basados en delay, entre los que se tiene los basados en delays constantes que son los de reverberación, y los basados en delays variables, Flanger.

6.1. Reverberación

Inicialmente se implemento el eco simple, el cuan conta en sumarle a la señal de entrada, la misma atenuada y retrasada en el tiempo.

$$y(n) = x(n) + gx(n - M)$$

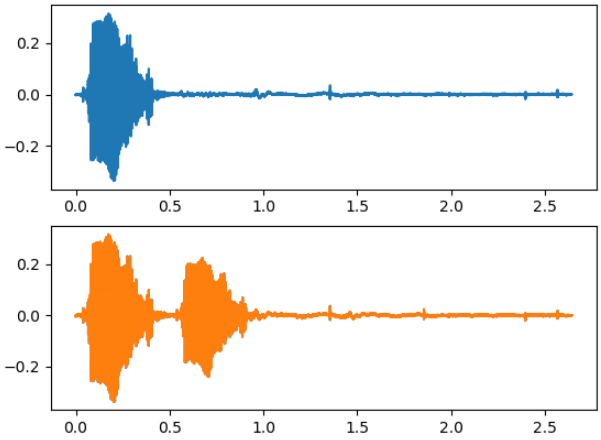

FIGURA 6.1: Respuesta de la implentacion del eco, con un delay de 100ms

Para que dicho efecto sea apreciable por el oído humano el delay ingresado debe ser mayor a 100ms, para entenderse de otra forma puede verse como la distancia del receptor a la fuente de sonido, la cual debe ser mayor a 34m. Luego el parámetro g puede entenderse como las pérdidas por la absorción del aire.

Este primer modelo toma en cuanta una única reflexión del sonido, si se tienen en cuenta mayor cantidad de reflexiones teniendo en cuenta que su amplitud disminuye exponencialmente se puede modelar un reverberador plano, mediante un filtro comba:

$$y(n) = x(n) + gy(n - M)$$

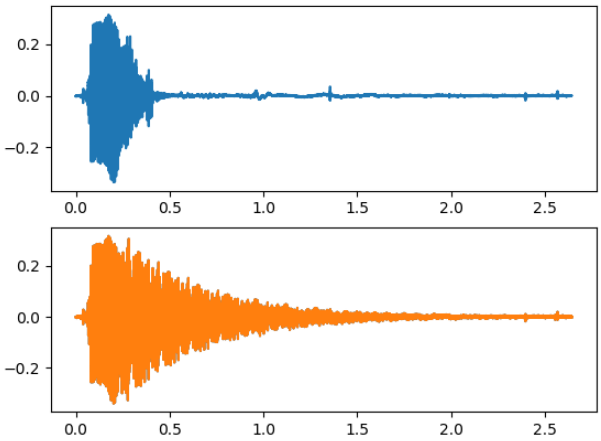

FIGURA 6.2: Respuesta asumiendo mayor cantidad de reflexiones

Ahora bien estos modelos asumen constantes las perdidas por la absorción del aire, una tercer aproximación se puede realizar cambiando dicho parámeto (g) por una función, la cual puede ser modelada como un filtro pasabajos, obteniendo de esta forma:

$$y(n) = x(n) - g \cdot (y(n - M) + y(n - M - 1)) \tag{6.1}$$

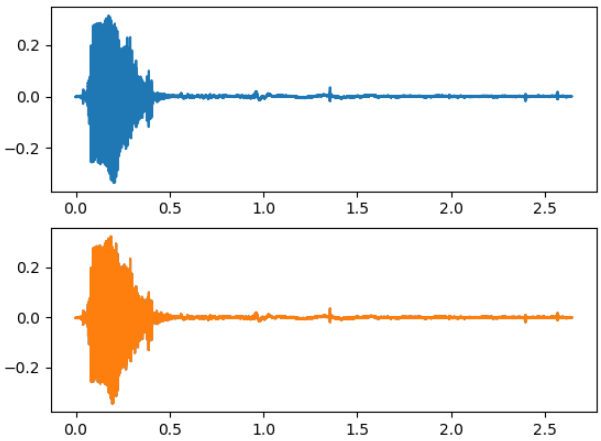

FIGURA 6.3: Respuesta con la absorción modelada mediante un pasa bajos

Finalmente a fin de conseguir implementar un efecto de sonido provocado por una sala se busco implemetnar el Reverberador de *Schroeder*, el cual consta de filtros comb conectados en paralelo, seguidos de filtros pasa todos en serie.

Para ello primero se implemento el filtro pasa todos según la ecuación:

$$y(n) = gx(n) + x(n - M) - gy(n - M)$$

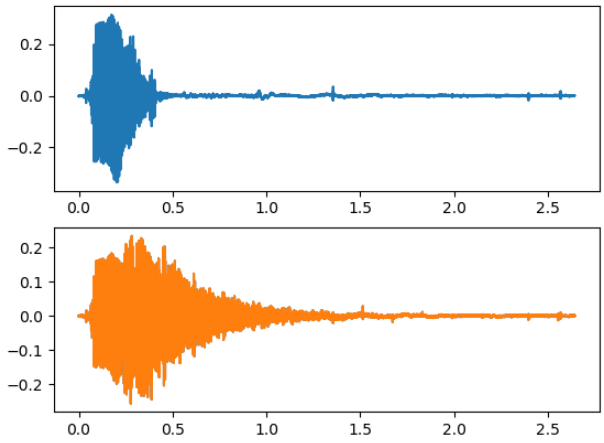

FIGURA 6.4: Salida de el filtro pasa-todos

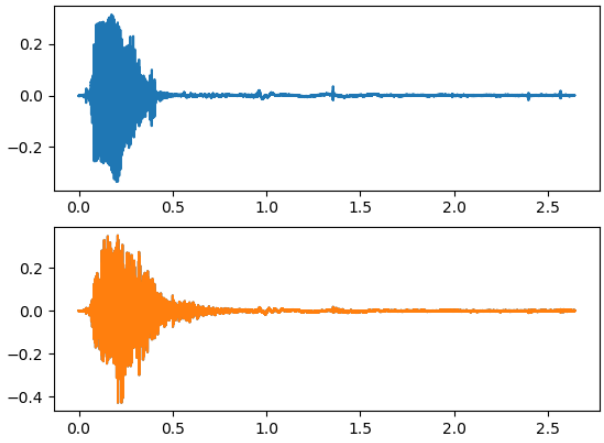

FIGURA 6.5: Salida del filtro reverberador completo

De esta forma se pueden modelar corrimientos de fase entre las componentes del espectro de la señal de entrada lo cual se condice con las características de propagacion del sonido.

Contando con los modelos basicos de los componentes del Reverberador de *Schroeder*, se implemento el mismo utilizando 4 filtros comb en paralelo y dos filtros pasa todos a la salida. Para determinar al ser un efecto con gran cantidad de etapas se limito la posibilidad de ajuste de los parámetros, buscando que el sonido a la salida sea lo mas controlado posible. Por lo tanto el unico parámetro de ajuste posible es el tiempo de reverberacíon, el cual esta definido como el tiempo en que la intensidad del sonido decae 60dB. Y este parámetro se lo asocia a los parámetros de los filtros según:

$$g_i = 10^{-\frac{M_i f_s}{t_{60}}}$$

Para los valores de delay se tomaron los valores expuestos en la bibliografia sugerida adecuados a

una frecuencia de muestreo de 44,1kHz.

6.2. Flanger

Para los efectos basados en delays variables se tomo como base el circuito propuesto en la figura 6.6.

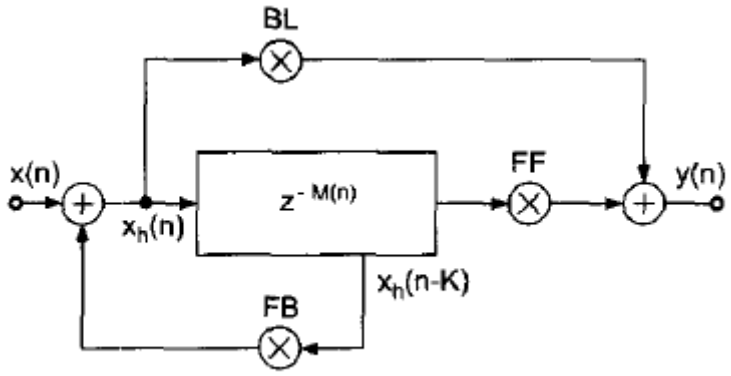

FIGURA 6.6: Diagrama efectos de delay variable

Del mismo se obtiene asi:

$$y(n) = BL \cdot x(n) + FF \cdot x\left(n - M(n)\right) + FB \cdot y\left(n - M(n)\right)$$

Siendo:

$$M(n) = M_0 \left(1 + sin\left(2\pi n \frac{f_0}{f_s}\right)\right)$$

Siendo M_0 el delay promedio, f_0 la velocidad de movimiento del flanger y f_s la frecuencia de sampleo de la muestra.

Tomando BL = 1, FB = 0 y tomando FF en el intervalo [0;1), se obtine el efecto Vibrato, para el cual se recomiendan bajas frecuencias (menores a 7Hz) y retardos promedios pequeños tambien, del orden de 5ms.

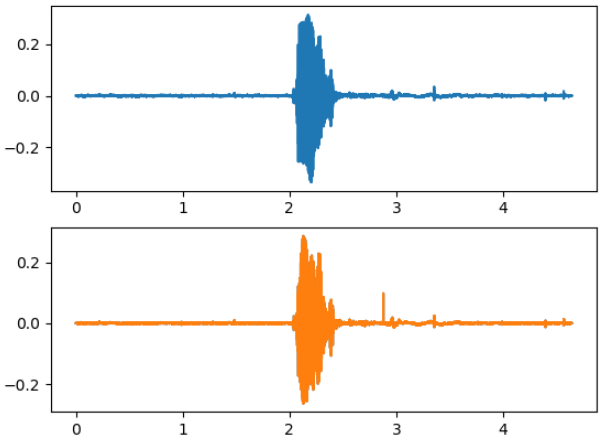

FIGURA 6.7: Audio con efecto vibrato

Y por ultimo tomando los tres parametros, BL, FF y FB, iguales y pertenecientes al intervalo [0,1) se obtiene el efecto conocido como flanger.

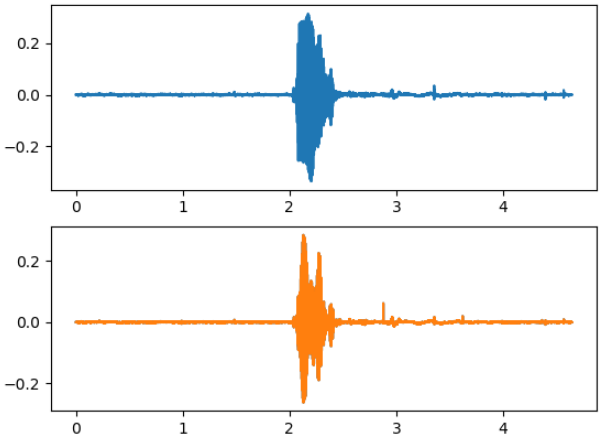

FIGURA 6.8: Muestra de audio con flanger

7. Graphical User Interface

A continuación se muestran algunas imagenes de la GUI, explicando las distintas partes que la componen. El programa permite sintetizar con los siguientes 15 instrumentos:

1. Violín
2. Trompeta
3. Oboe
4. Acordeón
5. Guitarra eléctrica
6. Platillos
7. Piano
8. violonchelo
9. Viola
10. Saxofón
11. Mandolín
12. Banyo
13. Guitarra acústica
14. Bajo eléctrico
15. Fagot

Primera parte: Selección de tracks y efectos en tiempo real. En la figura 7.1 se muestra la posibilidad que la GUI brinda para aplicar efectos de sonido en tiempo real, ya sea a un solo track, a varios tracks o a toda la canción, e incluso se le pueden ir aplicando distintos efectos a diferentes tracks. Una vez que se carga un archivo MIDI, se muestran en violeta los tracks que aún no tienen asitgnado un instrumento, y que por lo tanto no sonarán. Una vez que se asigna un instrumento a un track, el mismo se pone en color verde. En la columna de la izquierda se muestra la lista de los tracks. El símbolo del micrófono, al ser tocado, permite mutear el track para que el mismo no se escuche, y así poder elegir qué tracks escuchar y cuáles no. La barrita en cada track permita cambiarle el volumen al track. Con los botones de abajo a la izquierda, se sintetiza el sonido y se puede guardar el mismo, con las configuraciones hechas, en formato *.wav*.

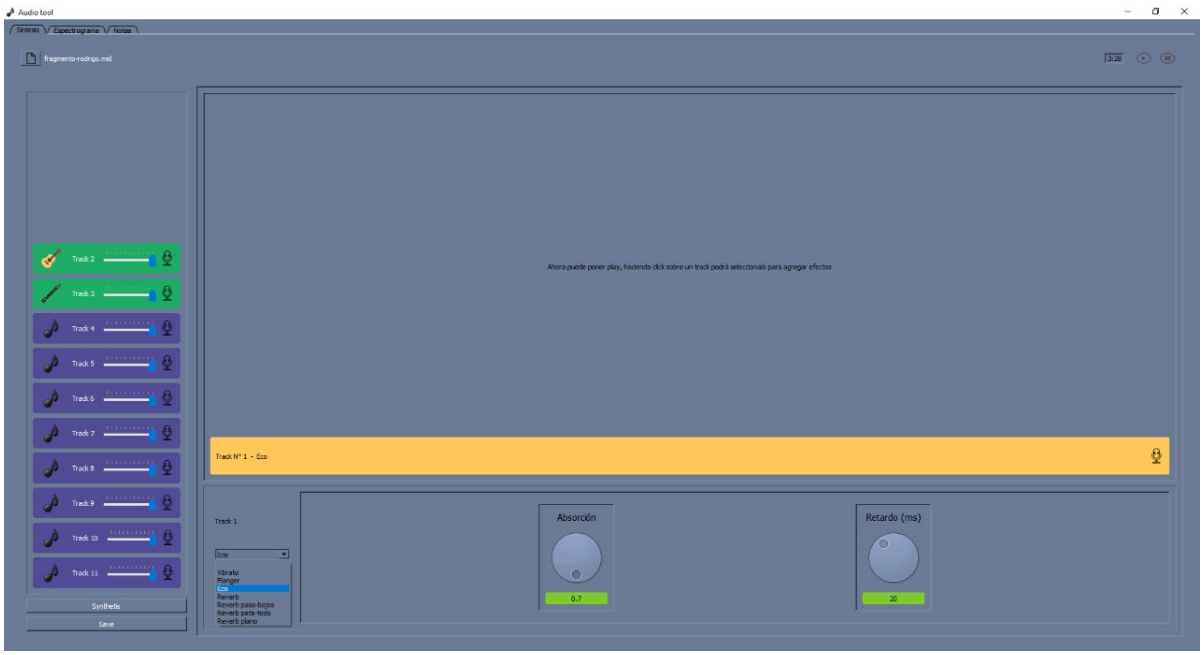

FIGURA 7.1: Aplicación de efectos en tiempo real.

Asignación de instrumentos: Para configurar el instrumento de cada track, se debe tocar el símbolo de nota musical en el track correspondiente y se abrirá la siguiente ventana, en la cual se selecciona el instrumento deseado. Ver la siguiente imagen 7.2.

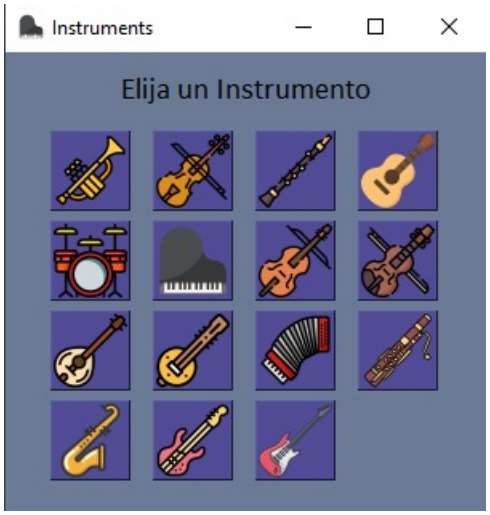

FIGURA 7.2: Asignación de instrumentos a cada track.

Segunda parte: Espectrograma en tiempo real. El programa permite realizar el espectrograma de la canción en tiempo real. A continuación se muestra una imagen de esta parte. El mismo puede ser guardado.

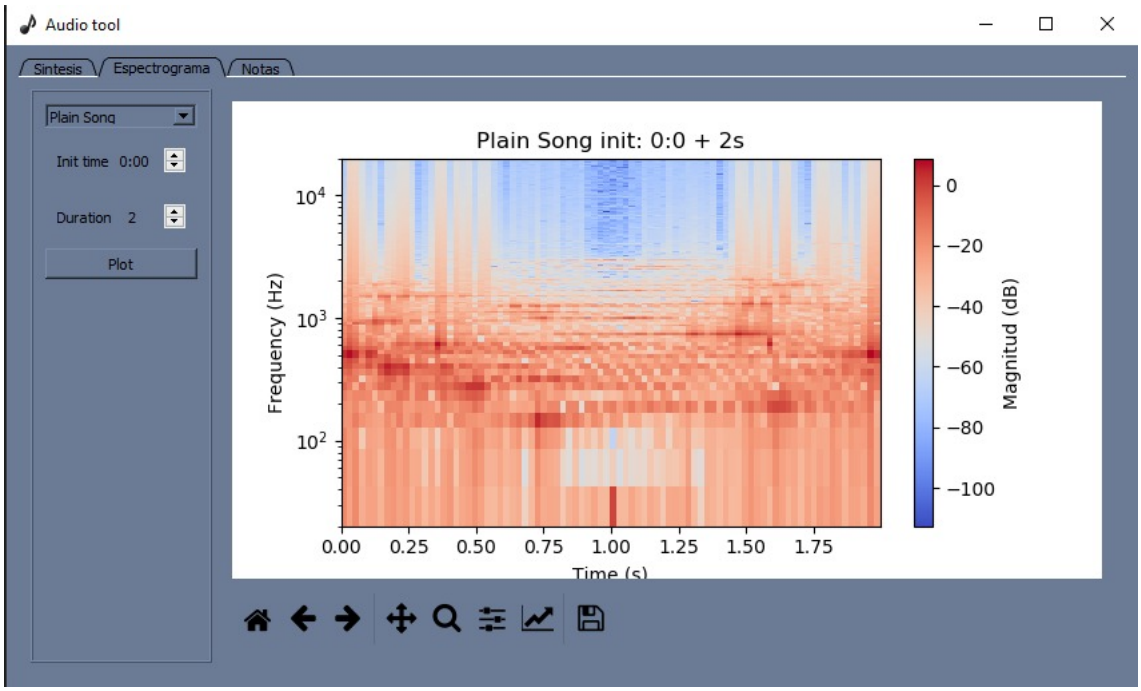

FIGURA 7.3: Espectrograma.

Tercera parte: Prueba de notas y acordes sueltos. El programa tiene una parte en la que se puede probar el sonido de distintas notas de los 15 instrumentos, o probar acordes. Se debe ingresar la frecuencia fundamental del sonido de una nota, el tiempo de inicio de su sonido, su duración y el instrumento deseado. También se puede cambiar el volumen. Al tocar *Agregar*, la nota es agregada al conjunto de notas que sonarán como acorde al tocar *Play*. La utilidad de esto es poder probar notas sueltas en caso de querer saber cómo suena una frecuencia en particular de un instrumento.

FIGURA 7.4: Prueba de notas y acordes sueltos.