

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.05 ANÁLISIS DE SEÑALES Y SISTEMAS DIGITALES

TRABAJO PRÁCTICO DE LABORATORIO N°2

Análisis y Síntesis de Audio Digital

Grupo 2:

Matías LARROQUE
Leg. 56597

Tomás Agustín GONZÁLEZ ORLANDO
Leg. 57090

Lucero Guadalupe FERNANDEZ
Leg. 57485

Manuel MOLLÓN
Leg. 58023

Ezequiel VIJANDE
Leg. 58057

Profesor:

Daniel JACOBY
Carlos BELAUSTEGUI GOITIA
Rodrigo Iñaki IRIBARREN

Entregado: 3 de Mayo de 2019

1 FFT

Se implementó una FFT que responde a la DFT desnormalizada en C++ teniendo en cuenta que la longitud de los vectores de prueba serán en potencias de 2.

La interfaz indicada por consigna para la implementación de dichos algoritmos es:

1. void fft(vector<complex<float>>& in, vector<complex<float>>& out);
2. void ifft(vector<complex<float>>& in, vector<complex<float>>& out);

Se contempló el caso en que el vector de input sea el mismo que el de output.

Para la implementación de dichos algoritmos se utilizó referencias externas, aclaradas en el código en cada sector en que se las necesitó.

Se utilizó la librería numpy de python para confirmar el correcto funcionamiento de la implementación.

Se escribió en el archivo “inputs_prueba.txt” los inputs de prueba y la respuesta a cada input. Para parsear el archivo y corroborar su correcto funcionamiento, se indica la manera en que este fue escrito:

- La primera línea antes de un nuevo input de prueba tendrá una letra ‘P’ mayúscula
- La segunda línea tendrá el input de prueba un vector “in” escrito bajo el formato “[a, b, c,..., n]” donde a,b,c... son números complejos escritos en la forma “(re, im)”, donde re es la parte real del número complejo e im es la parte imaginaria del mismo número complejo.
- La tercera línea tendrá el vector de output “out” luego de haberle aplicado la fft a in. “out” estará escrito de la misma manera que “in” en el ítem anterior.
- La cuarta línea tendrá el vector de output luego de pasar por la ifft. Se corroboró que este fuera el mismo que “in” con alguna cota de error dada por cálculos computacionales.

Los inputs de prueba fueron generados aleatoriamente.

2 Síntesis Aditiva

2.1 Introducción

En esta sección se trata la síntesis aditiva y su implementación en código. Comenzando con una breve introducción teórica de la síntesis aditiva, luego se explican las formas implementadas de síntesis aditiva y los resultados obtenidos. Para finalizar se presenta el rango de funcionamiento de los instrumentos sintetizados y las conclusiones obtenidas a lo largo de la investigación. Tras todo lo aprendido sobre las síntesis aditiva, se crearon presets para cuatro instrumentos para incluir en el programa principal.

2.2 Marco Teórico

La síntesis aditiva es un tipo de síntesis el cual genera un timbre específico al sumar sinusoidales de diferentes frecuencia. El timbre de un instrumento es lo que diferencia un instrumento del otro. Ya que una guitarra tocando un concert A (440Hz) no suena igual a una flauta tocando la misma nota. Lo que define el timbre de un instrumento son los armónicos presentes ademas de la frecuencia fundamental y la amplitud de estos armónicos. Estos armónicos son denominados partials o overtones. Cada partial es una senoidal pura de una frecuencia, a la cual su amplitud varia con el tiempo. De manera que se puede expresar la síntesis aditiva mas simple como:

$$y(t) = \sum_{k=1}^N A_k(t) \cdot \sin(2\pi k \cdot f_o \cdot t)$$

Donde A_k es la envolvente para el k-esimo partial y es una función que varia en el tiempo.

2.3 Parametrización ADSR

La parametrización mas simple de una envolvente para la síntesis aditiva es la parametrización ADSR donde esta es igual para cada partial. Bajo esta suposición, los instrumentos tienen cuatro parámetros característicos, reflejados claramente en la siguiente imagen:

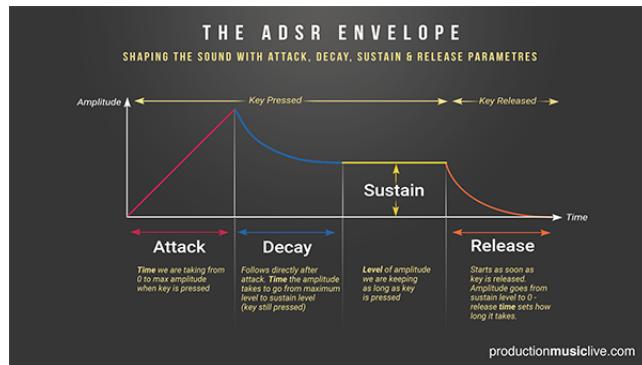


Figura 1: Envolvente ADSR

Los parámetros son Attack, Decay, Sustain y Release. El parámetro attack corresponde al tiempo que tarda un instrumento en llegar a su amplitud maxima. Los instrumentos de cuerda suelen tener tiempo de attack muy cortos

ya que al soltar la cuerda el sonido llega a su maxima de manera casi instantánea. Otros instrumentos como los de viento pueden tardar mas en llegar a su máximo potencial. El parámetro Decay corresponde al tiempo que tarda un instrumento en estabilizarse en su nivel de Sustain tras llegar a su maxima amplitud. Un instrumento puede no tener decay, en caso de que el sustain sea igual a la amplitud maxima. El parámetro sustain indica el porcentaje de la amplitud maxima que tiene un instrumento una vez estabilizado, es un parámetro en amplitud, no en tiempo. Ya que el sustain no tiene un tiempo fijo, sino que es definido por el largo que se toca la nota. El ultimo parámetro es el Release, que indica el tiempo que tarda en desvanecerse la nota una vez que se dejo de tocar.

Modelando esta función matemáticamente se obtiene la siguiente función:

$$f(t) = \begin{cases} \frac{t}{t_a} & 0 \leq t \leq t_a \\ \frac{s-1}{t_d} \cdot (t - t_a) & t_a \leq t \leq t_a + t_d \\ s & t_a + t_d \leq t \leq t_t - t_r \\ \frac{s}{t_r} \cdot (t_t - t) & t_t - t_r \leq t \leq t_t \end{cases}$$

Donde:

$$\begin{cases} t_a & \text{tiempo - de - attack} \\ t_d & \text{tiempo - de - decay} \\ s & \text{nivel - de - sustain} \\ t_r & \text{tiempo - de - release} \\ t_t & \text{tiempo - total - de - nota} \end{cases}$$

2.3.1 Obtención de parámetros

A continuación se obtuvieron los partials y parámetros de la envolvente ADSR de cuatro instrumentos con la ayuda de un software (Sonic Visualizer). A partir de la transformada de Fourier se obtuvieron el numero de armónicos predominantes en los instrumentos y las amplitudes de estos. La transformada de Fourier no refleja con exactitud los parámetros ya que estas amplitudes varían con el tiempo. Pero es una buena aproximación si se quiere implementar una envolvente ADSR común a todos los partials, variando solo la amplitud de las senoidales para cada armónico. A continuación se muestra lo encontrado a partir de las muestras.

- Guitarra

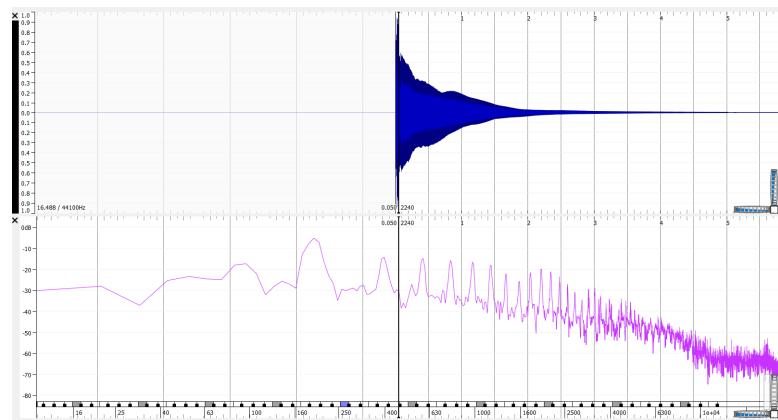


Figura 2: Envolvente y Transformada de Fourier

| t_a | t_d | s | t_r |
|---------|----------|-----|-------------------|
| $0.07s$ | $0.038s$ | 0.6 | $t_t - t_a - t_d$ |

Cuadro 1: Tabla de parámetros

- Violín

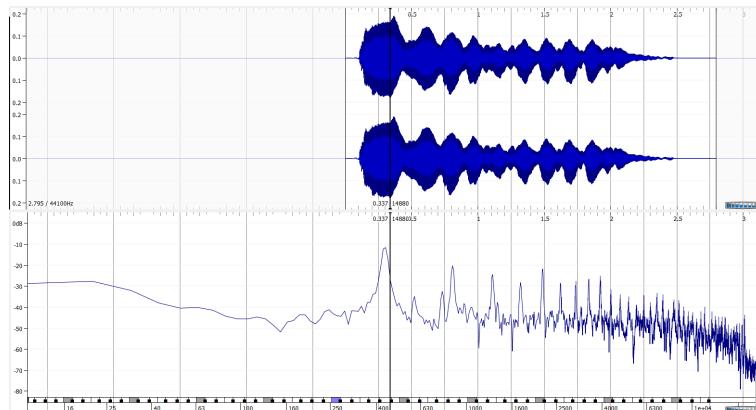


Figura 3: Envolvente y Transformada de Fourier

| t_a | t_d | s | t_r |
|--------|--------|-----|-------|
| 0.364s | 0.105s | 0.8 | 0.4s |

Cuadro 2: Tabla de parámetros

- Saxo

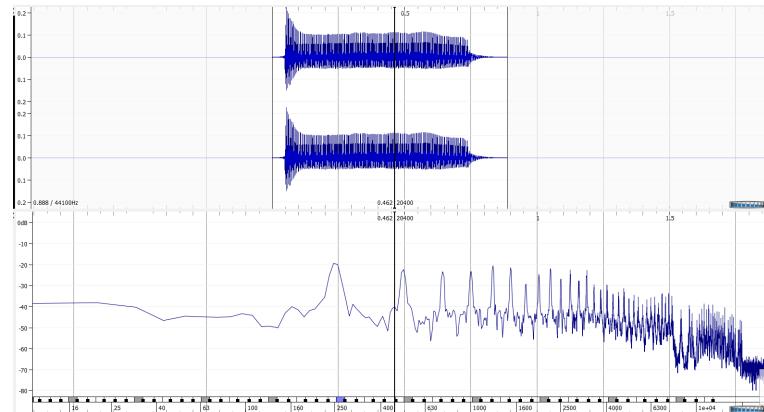


Figura 4: Envolvente y Transformada de Fourier

| t_a | t_d | s | t_r |
|--------|--------|-----|-------|
| 0.008s | 0.055s | 0.5 | 0.2s |

Cuadro 3: Tabla de parámetros

- Trompeta

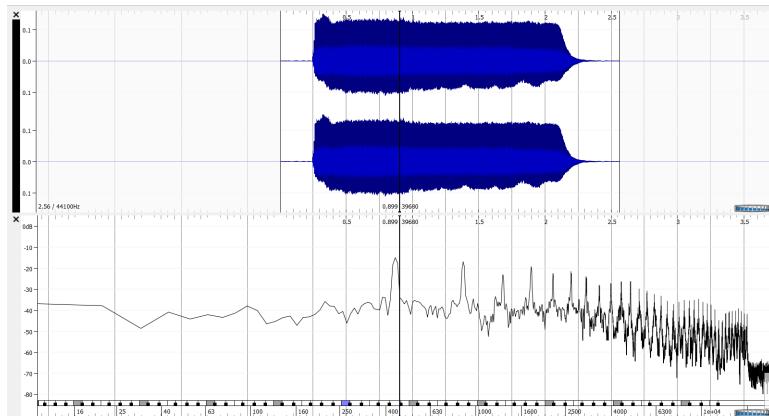


Figura 5: Envolvente y Transformada de Fourier

| t_a | t_d | s | t_r |
|-------|-------|-----|--------|
| 0.05s | 0s | 1 | 0.162s |

Cuadro 4: Tabla de parámetros

2.3.2 Resultados

Al implementar la síntesis aditiva con los parámetros anteriormente encontrados, se obtuvieron los siguientes resultados:

- Guitarra

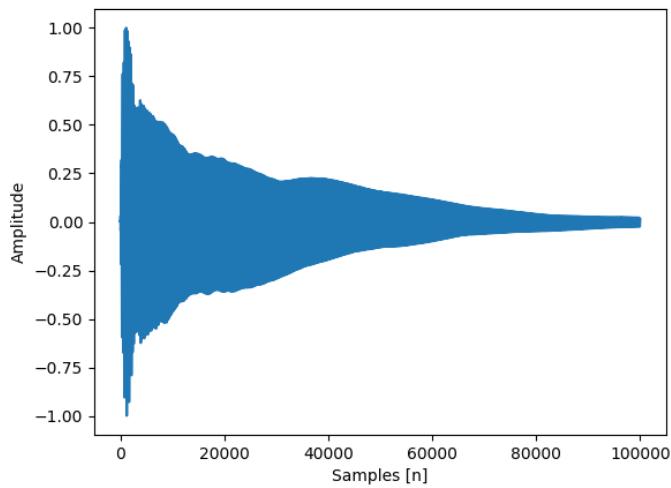


Figura 6: Señal Original

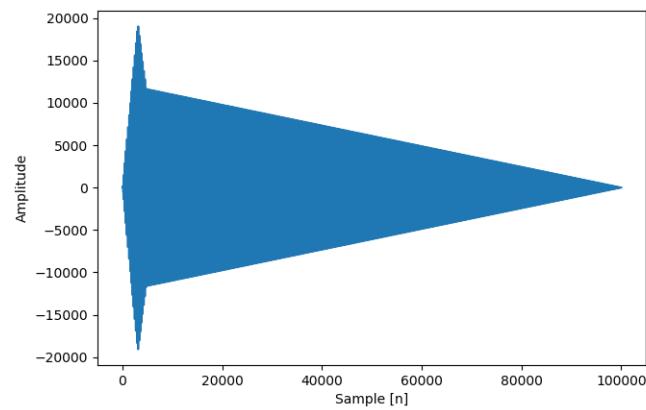


Figura 7: Señal Sintetizada

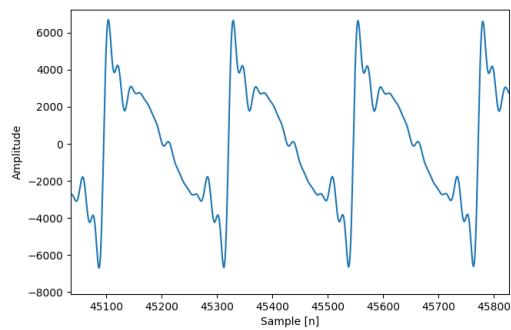


Figura 8: Detalle de Señal Sintetizada

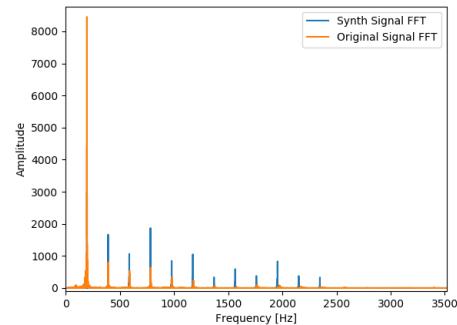


Figura 9: Transformadas de Fourier

- Violín

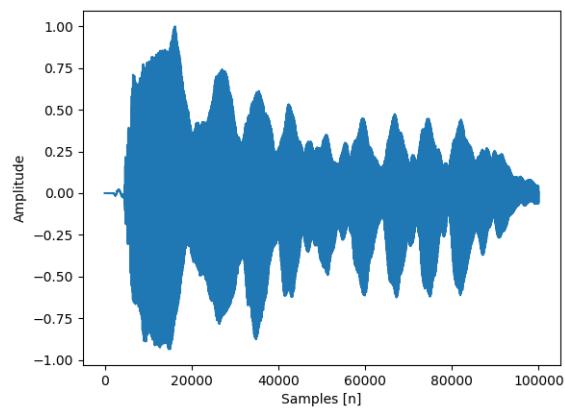


Figura 10: Señal Original

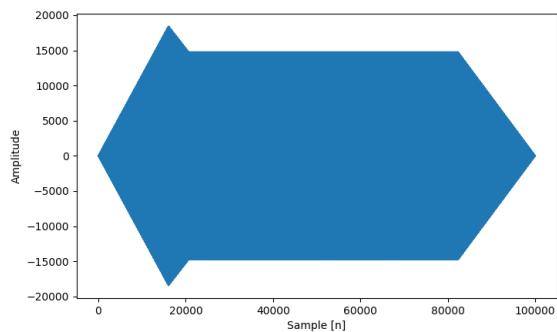


Figura 11: Señal Sintetizada

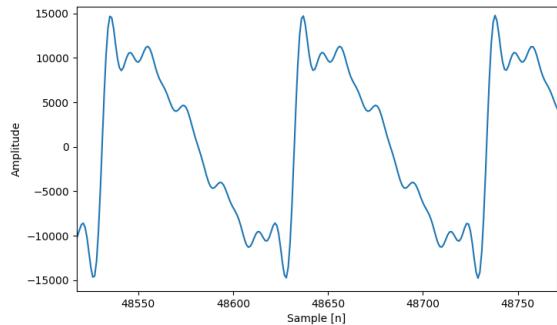


Figura 12: Detalle de Señal Sintetizada

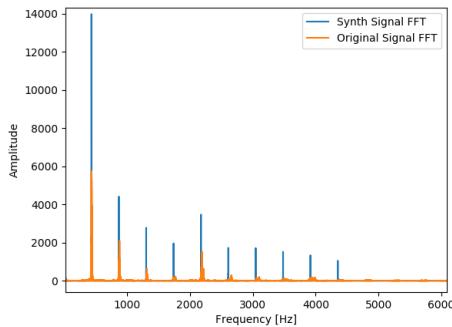


Figura 13: Transformadas de Fourier

- Saxo

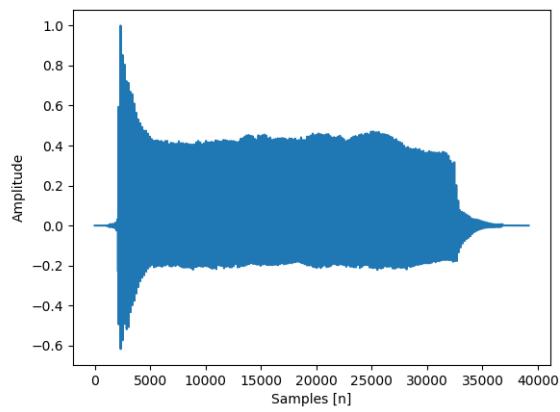


Figura 14: Señal Original

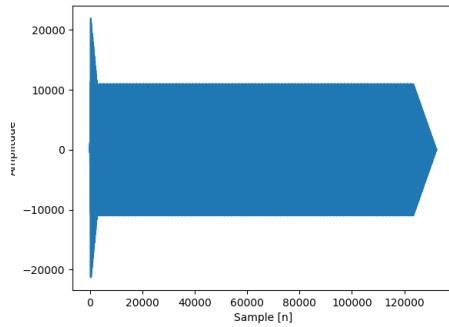


Figura 15: Señal Sintetizada

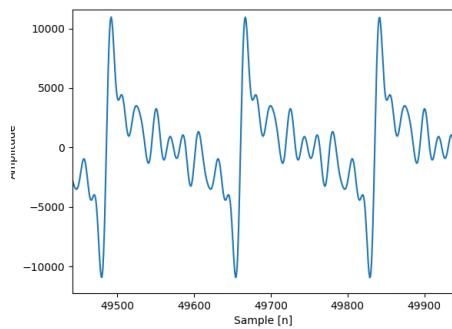


Figura 16: Detalle de Señal Sintetizada

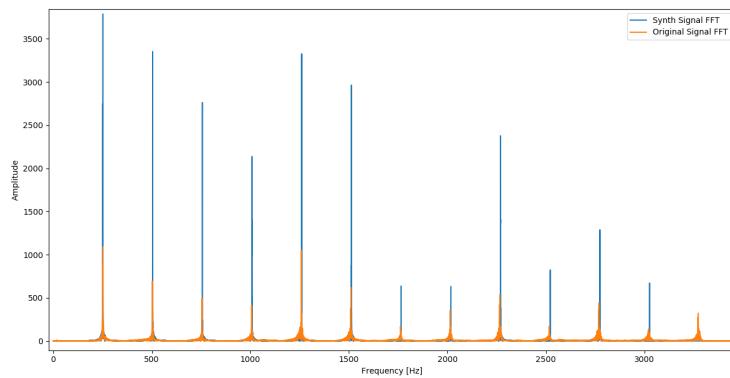


Figura 17: Transformadas de Fourier

- Trompeta

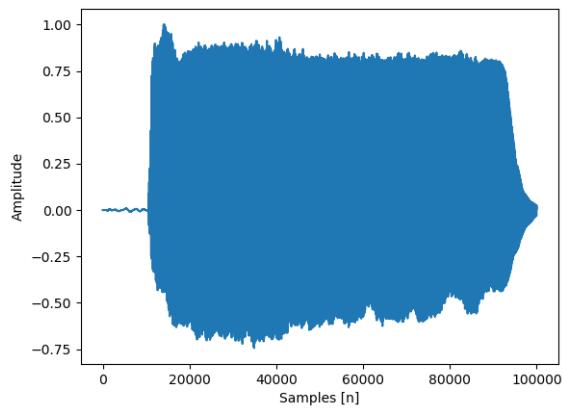


Figura 18: Señal Original

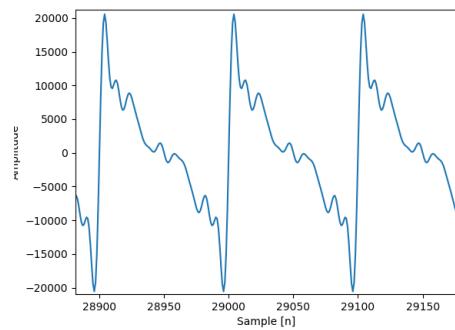


Figura 20: Detalle de Señal Sintetizada

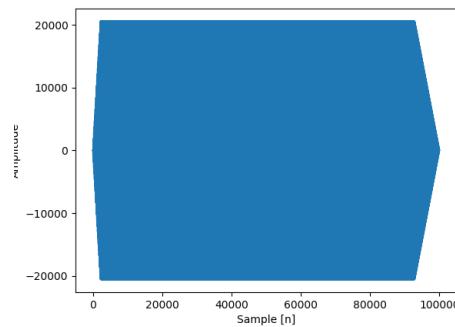


Figura 19: Señal Sintetizada

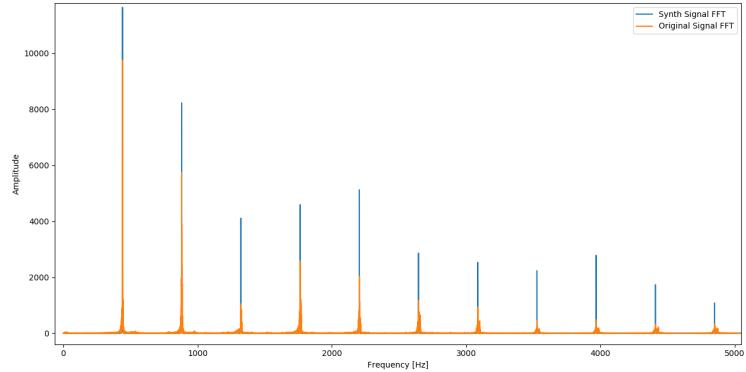


Figura 21: Transformadas de Fourier

2.4 Investigación de Mejoras

Para mejorar esta síntesis aditiva, se probó tener en cuenta la desviación aleatoria en frecuencia de cada parcial, el cual se ve en los instrumentos reales, los cuales no cumplen los armónicos perfectamente. Sino que puede variar la frecuencia de ese verdadero fundamental. Para programar esta mejora se midió el desvío de cada parcial para cada instrumento y se hizo el promedio de estos desvíos para cada instrumento. A continuación se sintetizaron nuevamente las mismas notas que la sección anterior, con un desvío agregado a cada parcial, aleatorio generado por la computadora. El resultado fue el siguiente:

- Guitarra

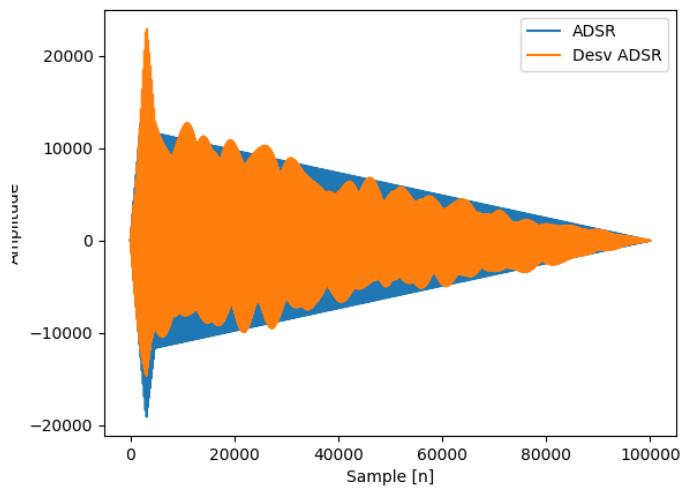


Figura 22: Comparación de sintetizadores

- Violín

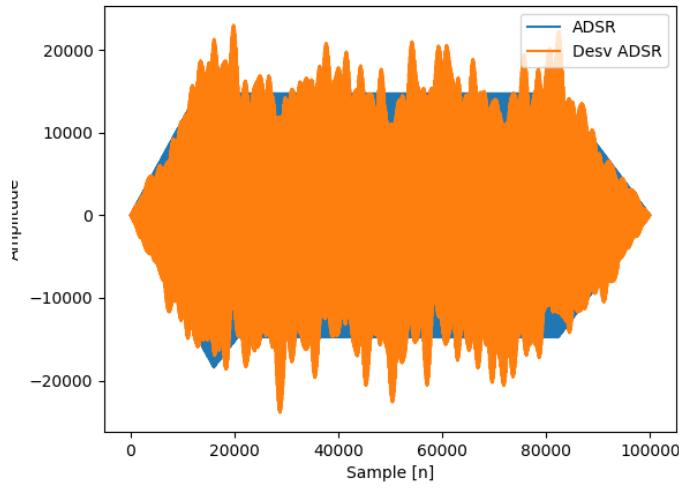


Figura 23: Comparación de sintetizadores

- Saxo

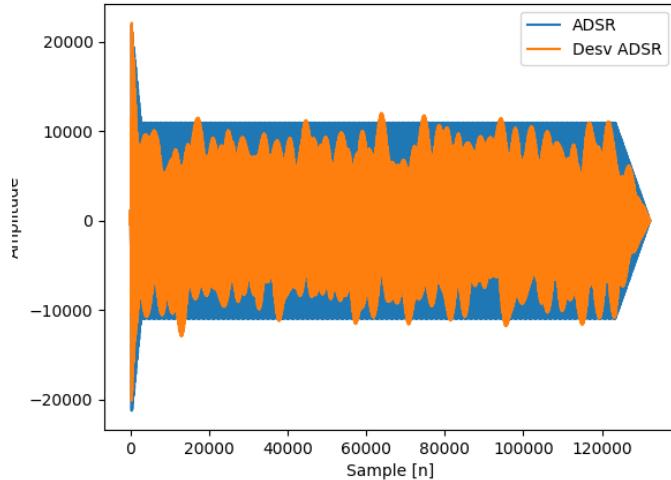


Figura 24: Comparación de sintetizadores

- Trompeta

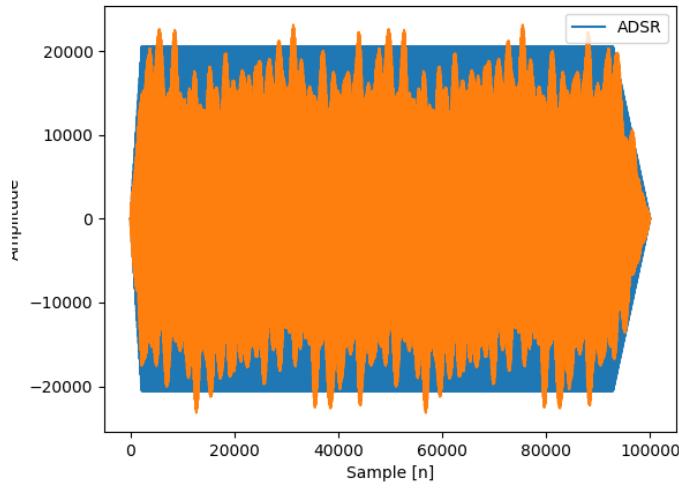


Figura 25: Comparación de sintetizadores

Como se puede apreciar, este desvío genera que la envolvente no sea tan lineal como en el caso anterior, lo cual hace que visualmente se aproxime mejor a la señal original. De cualquier manera, desde la percepción auditiva no se encontrar cambios significativos a la hora de imitar instrumentos reales. Por lo que para ahorrar tiempo de procesamiento, esta característica no se agrega a los presets para el programa principal.

2.5 Proposición de parametrización

Al escuchar los audios generados por la parametrización ADSR, se puede notar como estos suenan sintéticos y no se acercan mucho a la realidad a la hora de imitar instrumentos reales. Para mejorar este problema se puede empezar comparando la envolvente ADSR con envolventes reales obtenidas de muestras.

A partir de la comparación de estas envolventes, se pudo proponer una envolvente mas real, la cual llevaría a mejores resultados. Esta envolvente tiene también parámetros de attack, decay y release. Pero estos se dividen en dos etapas, una etapa exponencial y una etapa logarítmica. Ademas el sustain contiene un droop rate. De manera que la función se puede graficar de la siguiente manera:

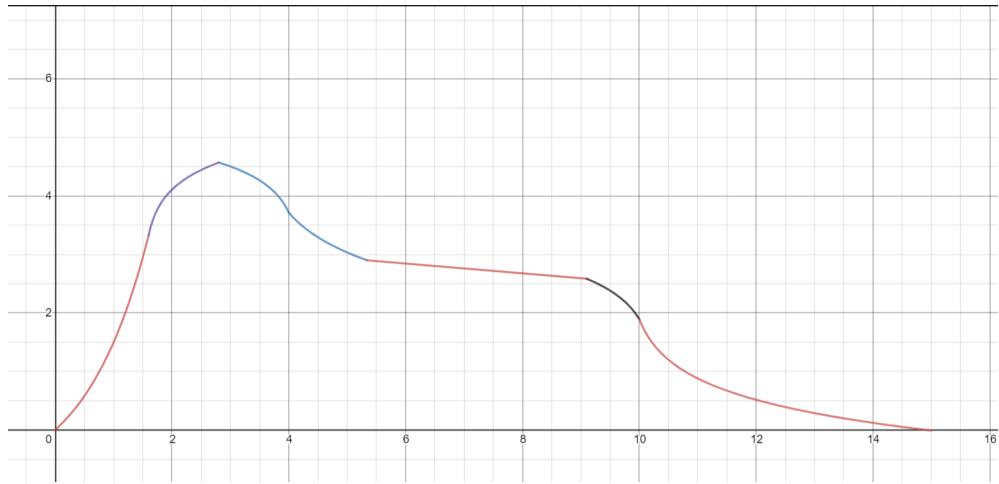


Figura 26: Envolvente ADSR exponencial/logarítmica

Y la expresión matemática es la siguiente:

- 1 $b^x - 1 \quad \{0 < x \leq a\}$
- 2 $\frac{(b^a - 1)}{\ln(a + d) + f} \cdot (\ln(x + d) + f) \quad \{a < x \leq c\}$
- 3 $\frac{(b^a - 1)}{\ln(a + d) + f} \cdot (\ln(c + d) + f) \cdot \frac{\ln(-x + g) + h}{\ln(-c + g) + h} \quad \{c < x \leq i\}$
- 4 $(b^a - 1) \cdot \frac{\ln(c + d) + f}{\ln(a + d) + f} \cdot \frac{\ln(-i + g) + h}{\ln(-c + g) + h} \cdot \frac{-\ln(x + n) + p}{-\ln(i + n) + p} \quad \{i < x \leq j\}$
- 5 $(b^a - 1) \cdot \frac{\ln(c + d) + f}{\ln(a + d) + f} \cdot \frac{\ln(-i + g) + h}{\ln(-c + g) + h} \cdot \frac{-\ln(j + n) + p}{-\ln(i + n) + p} \cdot \left(\frac{k \cdot x + l}{k \cdot j + l} \right) \quad \{j < x \leq m\}$
- 6 $(b^a - 1) \cdot \frac{\ln(c + d) + f}{\ln(a + d) + f} \cdot \frac{\ln(-i + g) + h}{\ln(-c + g) + h} \cdot \frac{-\ln(j + n) + p}{-\ln(i + n) + p} \cdot \left(\frac{k \cdot m + l}{k \cdot j + l} \right) \cdot \frac{\ln(-x + r) + s}{\ln(-m + r) + s} \quad \{m < x \leq o\}$
- 7 $(b^a - 1) \cdot \frac{\ln(c + d) + f}{\ln(a + d) + f} \cdot \frac{\ln(-i + g) + h}{\ln(-c + g) + h} \cdot \frac{-\ln(j + n) + p}{-\ln(i + n) + p} \cdot \left(\frac{k \cdot m + l}{k \cdot j + l} \right) \cdot \frac{\ln(-o + r) + s}{\ln(-m + r) + s} \cdot \frac{-\ln(x + u) + t}{-\ln(o + u) + t} \quad \{o < x \leq q\}$

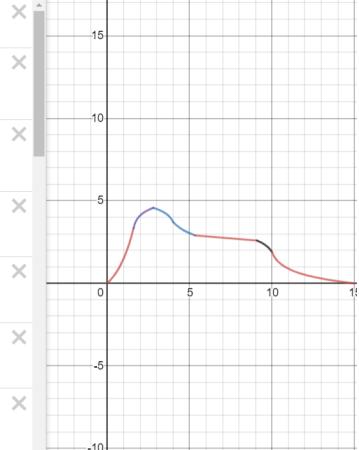


Figura 27: Expresiones matemáticas

Se encontró que se asimila mas a las muestras observadas, y da mas flexibilidad a la hora del diseño de una envolvente. La gran notable desventaja es la cantidad de parámetros a tener en cuenta a la hora de diseñar la envolvente. Se encontró que la manera mas eficiente es gráficamente ir variando los parámetros hasta encontrar la envolvente deseada con un graficador como lo es Desmos.

Una envolvente de esta manera puede tener parte exponencial cero en el decay y parte logarítmica cero en attack, formado lo siguiente:

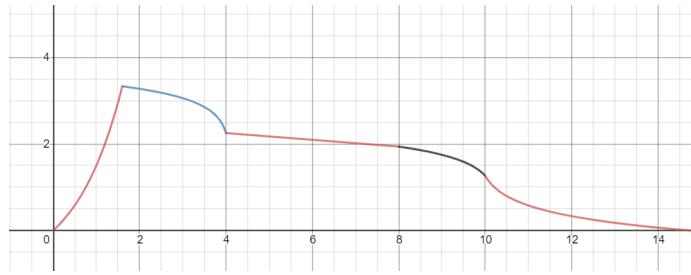


Figura 28: Ejemplo 1 Envolvente

O se puede tener un release solo exponencial para conseguir el siguiente efecto:

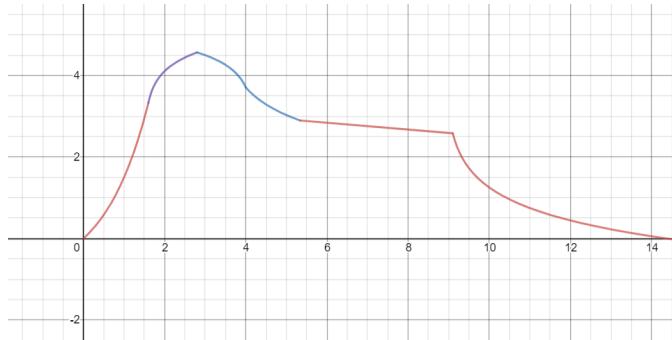


Figura 29: Ejemplo 2 Envolvente

Igualmente, se encontró que la manera de obtener el sonido mas realista es guardando las envolventes de muestras con el uso de espectroscopia e utilizando el algoritmo OLA para extender o contraer estas envolventes a la necesidad de la longitud de la nota.

2.6 Síntesis Aditiva a base de muestras

Como se explico anteriormente, se obtuvo un sonido mas realista al obtener la envolventes de muestras con el uso de un espectrograma. A continuación se presentan las envolventes del armónico fundamental de los instrumentos sintetizados y como resulta la señal final en comparación con la original.

- Guitarra

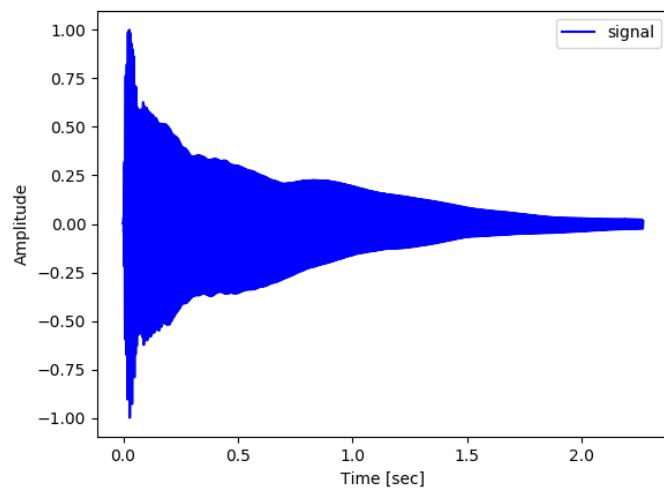


Figura 30: Señal Original

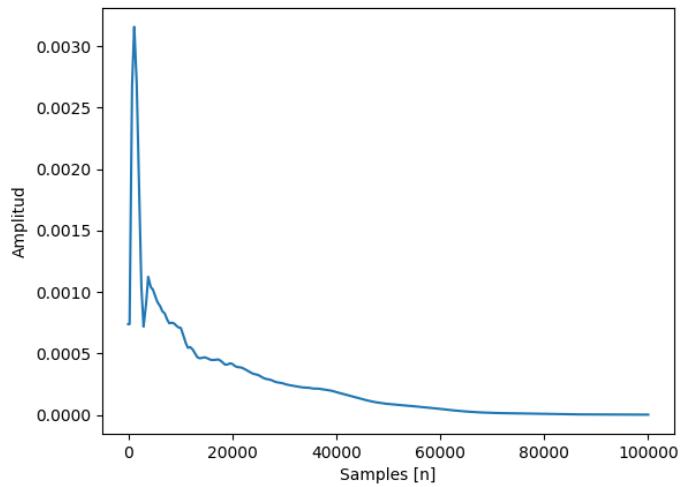


Figura 31: Envolvente de frecuencia fundamental

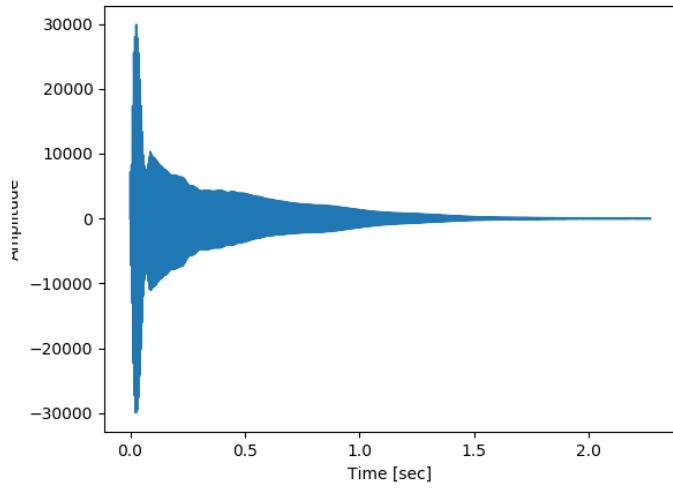


Figura 32: Señal Sintetizada

- Violín

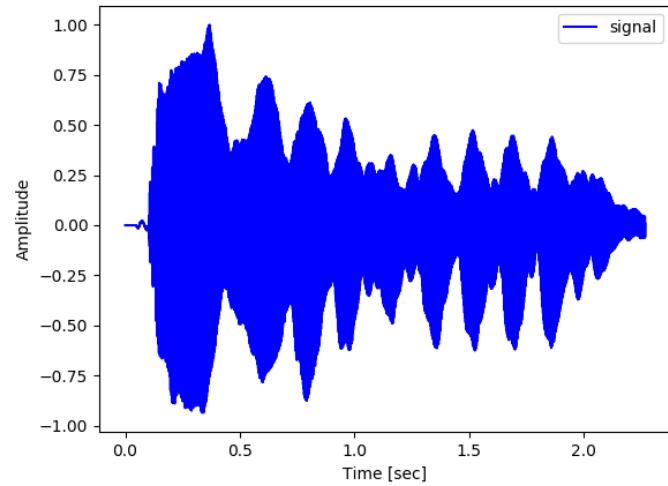


Figura 33: Señal Original

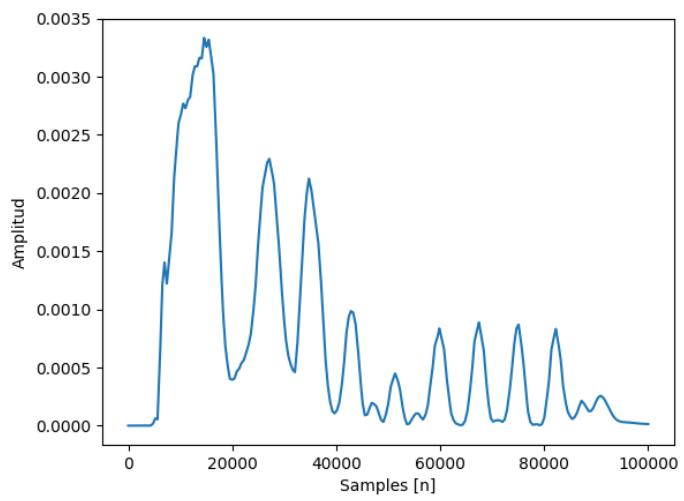


Figura 34: Envolvente de frecuencia fundamental

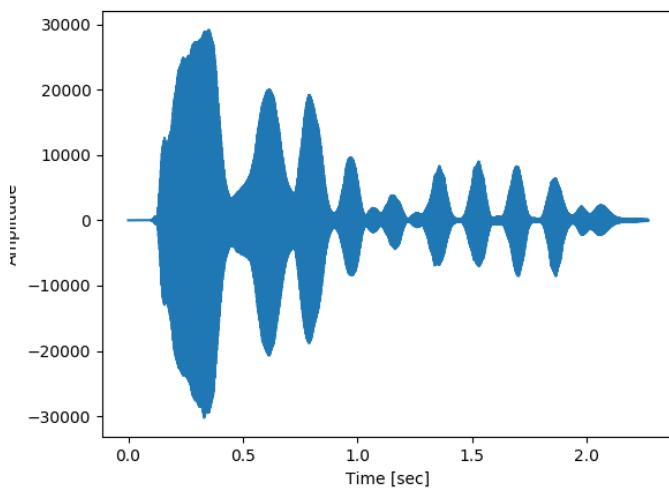


Figura 35: Señal Sintetizada

- Saxo

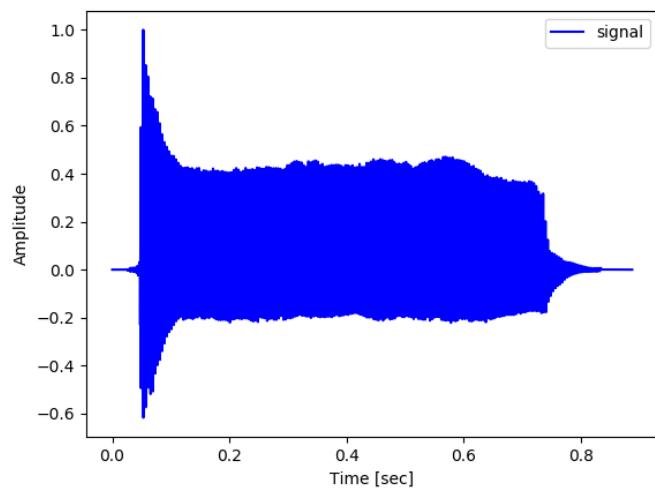


Figura 36: Señal Original

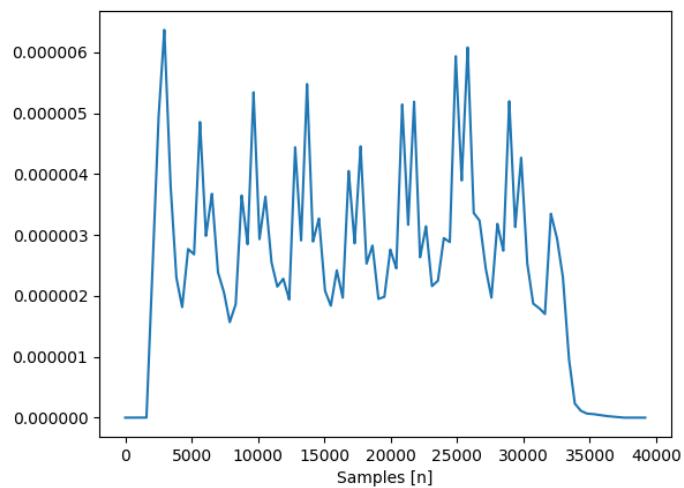


Figura 37: Envolvente de frecuencia fundamental

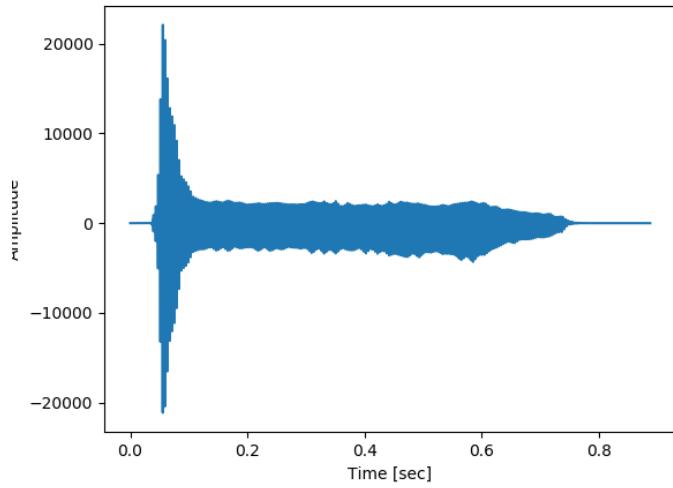


Figura 38: Señal Sintetizada

- Trompeta

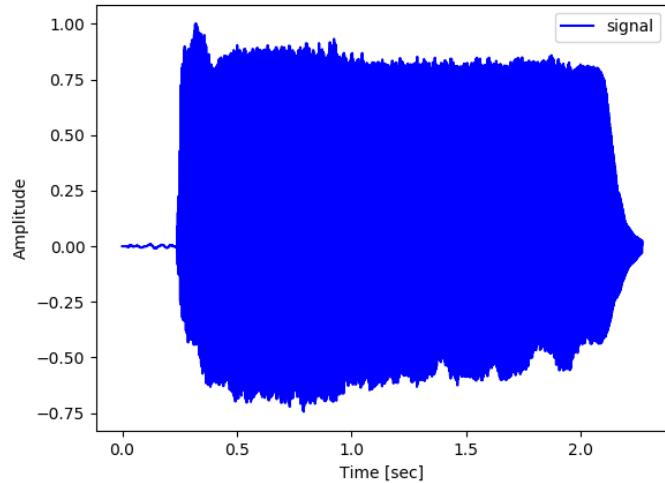


Figura 39: Señal Original

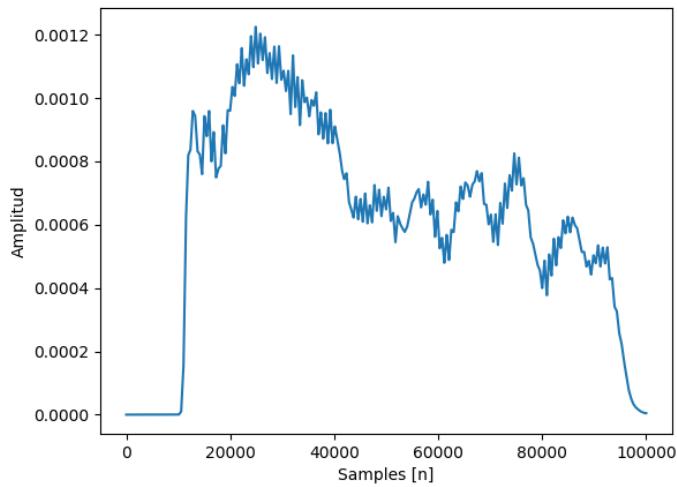


Figura 40: Envolvente de frecuencia fundamental

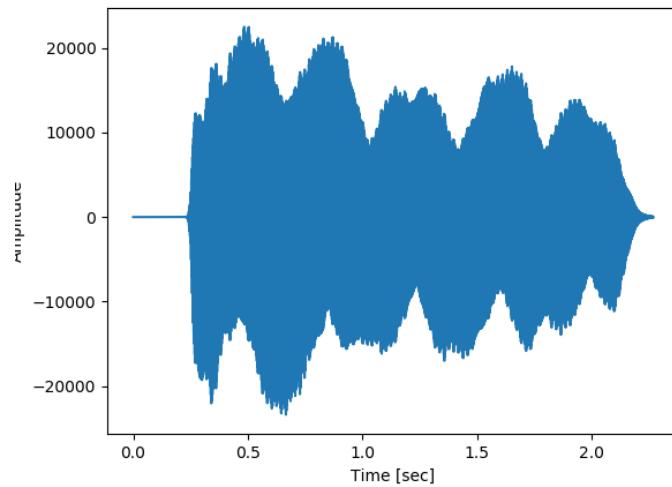


Figura 41: Señal Sintetizada

2.7 Rango de Funcionamiento

La primer limitación del funcionamiento es el rango natural de cada instrumento el cual se sintetiza. A continuación se muestra una tabla de estos mismos:

| Instrumento | Rango de frecuencia |
|-------------|----------------------------|
| Guitarra | E2 (82 Hz) a F6 (1397 Hz) |
| Violín | G3 (196 Hz) a B7 (3951 Hz) |
| Saxo | B2 (123 Hz) a F5 (698 Hz) |
| Trompeta | E3 (165 Hz) to B5 (988 Hz) |

Cuadro 5: Parámetro de funcionamiento acorde rango de frecuencias

La otra limitación de funcionamiento de debe a la longitud de la nota que se quiere sintetizar, ya que una nota no puede tardar menos de lo que duro el tiempo de attack mas el tiempo de decay mas el tiempo de release. Ademas al usar el algoritmo OLA para modificar la envolvente, la compresión o alargamiento tiene un limite antes de que se introduzca distorsión. Se observo experimentalmente que el rango aceptable de duración es aproximadamente entre $0.1\text{seg} - 5\text{seg}$.

2.8 Conclusiones

En conclusión, se pudo notar como la síntesis aditiva es un método rápido y sencillo para sintetizar instrumentos, pero este no es el indicado a la hora de imitar instrumentos reales. Cuando se obtuvieron envolventes de muestras se mejoró sustancialmente la calidad de sonido. Una posible mejora para mejorar la calidad de sonido es obtener las envolventes de un espectrograma superior al de la librería suministrada por Python. Ademas, se podría detectar los tiempos de attack, decay y release de las envolventes extraídas de muestras para así aplicar el algoritmo OLA a solo los tramos correspondientes al sustain.

3 Síntesis FM

3.1 Introducción

Los orígenes de la síntesis mediante el método FM se remontan a 1973 con el paper del padre de dicho método, John Chowning, titulado “The Synthesis of Complex Audio Spectra by Means of Frequency Modulation”.

En una entrevista con la Audio Engeneering Society (ver referencias), John Chowning cuenta que el descubrimiento del método surgió cuando estaba experimentando con dos osciladores, modulando en frecuencia de manera infrasónica para producir una suerte de vibrato en una señal. A medida que cambiaba la amplitud de dicho vibrato (vibrato depth) y la frecuencia a la cual se estaba haciendo oscilar a los osciladores al punto en que ya no podía seguir la frecuencia instantánea de la señal, pudo percatarse de que las frecuencias que aparecían como resultado caían dentro del rango audible y eran ricas en timbre.

Fue así como comenzó a analizar las relaciones entre frecuencias y amplitudes que permitió el desarrollo de la teoría presentada a continuación, que dio el origen a la síntesis FM o síntesis de modulación en frecuencia.

A modo de establecer un marco teórico a partir del cual se establecerán los conceptos y bases utilizados en el desarrollo del sintetizador, se procederá a explicar las relaciones matemáticas y los razonamientos en los que se basa dicho método.

3.1.1 Características del sonido

Un tono musical puede ser visto como una señal del tiempo que cae dentro de la banda audible, con frecuencias de 20Hz a 20kHz. Por lo general, cuando se habla de tonos musicales se considera una señal periódica en el tiempo, por lo que las frecuencias en el espectro de la misma serán discretas y múltiplos de una frecuencia fundamental f_0 .

Como un sonido musical en realidad no es una señal periódica a lo largo del tiempo, el espectro del mismo tendrá variaciones en función del tiempo y contendrá inharmonías, o frecuencias NO múltiplos de f_0 . Cuanto menos ideal sea el tono musical escuchado, más contenido inharmonico tendrá. Pero es importante destacar que el no idealismo no implica una peor calidad de sonido, sino todo lo contrario: Son aquellas no idealidades las que le dan el grado subjetivo a la música y caracterizarán a una nota como agradable o desagradable al oído.

Una nota musical no es lo mismo que un tono musical. Una nota musical hablará de la frecuencia fundamental de aquel sonido ideal mencionado anteriormente (el tono), una cierta duración, y estará descripta por una cierta notación musical.

Un tono musical puede ser entonces descripto por 4 características primordiales, a saber:

1. Tono: frecuencia fundamental del tono, es decir, aquella frecuencia cuya componente prevalece por sobre las demás, suele ser la primera a partir del cero, aunque no necesariamente.
2. Duración: Intervalo de tiempo en el que la señal no es nula.
3. Intensidad: Amplitud de la señal que describe al sonido en cierto instante de tiempo.
4. Timbre: Lo que le da “color” al sonido. Permite identificar a una nota musical siendo tocada por un instrumento y por otro. Consiste principalmente en tres cosas:

La evolución de las componentes en frecuencia a lo largo del tiempo.

Las frecuencias características de la señal.

La envolvente dinámica, o la variación del sonido en el tiempo.

Es en el timbre en el que nos centraremos en este informe y es por la habilidad de conseguir un timbre característico y no demasiado irreal con fórmulas matemáticas simples y bien establecidas lo que le dio su reconocimiento al método de síntesis FM.

3.1.2 Análisis matemático

La fórmula general para una señal modulada en frecuencia, otorgada por consigna, es:

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

Para simplificar cuentas y cálculos que tenga que realizar la computadora a la hora de sintetizar, elegimos $\phi_m = \phi_c = -\frac{\pi}{2}$, por lo que la expresión anterior queda reducida a:

$$x(t) = A(t) \cdot \sin(2\pi \cdot f_c \cdot t + I(t) \cdot \sin(2\pi \cdot f_m \cdot t))$$

f_c será entonces la frecuencia portadora y f_m la frecuencia moduladora. $A(t)$ e $I(t)$ definirán entonces las envolventes de las señales portadora y moduladora respectivamente.

Usando la expansión en funciones trigonométricas de $x(t)$, la misma puede ser reescrita como:

$$x(t) \approx J_0 \cdot A \cdot \sin(\alpha \cdot t) + A \cdot \sum_{k=1}^n J_k(I) \cdot [\sin((\alpha + k\beta) \cdot t) + (-1)^k \sin((\alpha - k\beta) \cdot t)]$$

Donde $J_k(I)$ son las funciones de Bessel de primera especie.

Los $J_k(I)$ tendrán entonces la forma:

$$J_k(I) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \cdot \gamma(m+k+1)} \cdot \left(\frac{I}{2}\right)^{2m+k}$$

Donde $\gamma(z)$ es la función gamma.

y los $J_k(I)$ responden a la ecuación diferencial $x^2 \cdot \frac{d^2y}{dx^2} + x \cdot \frac{dy}{dx} + (x^2 - \alpha^2) \cdot y = 0$.

De esta manera, es posible determinar el comportamiento de los $J_k(I)$ numéricamente.

De la expansión trigonométrica de $x(t)$ podemos, para un A y un I fijos, estudiar las componentes en frecuencia de la señal en parte estudiando el comportamiento de las funciones de Bessel de primera especie:

Primero, es importante hacer notar que el valor de I necesario para que un determinado valor de k produzca una amplitud lo suficientemente grande en las senoidales representativas para que la frecuencia que representan sea tenida en cuenta es creciente a medida que el valor de k aumenta. Esto es fácilmente observable de los primeros gráficos de $J_k(I)$:

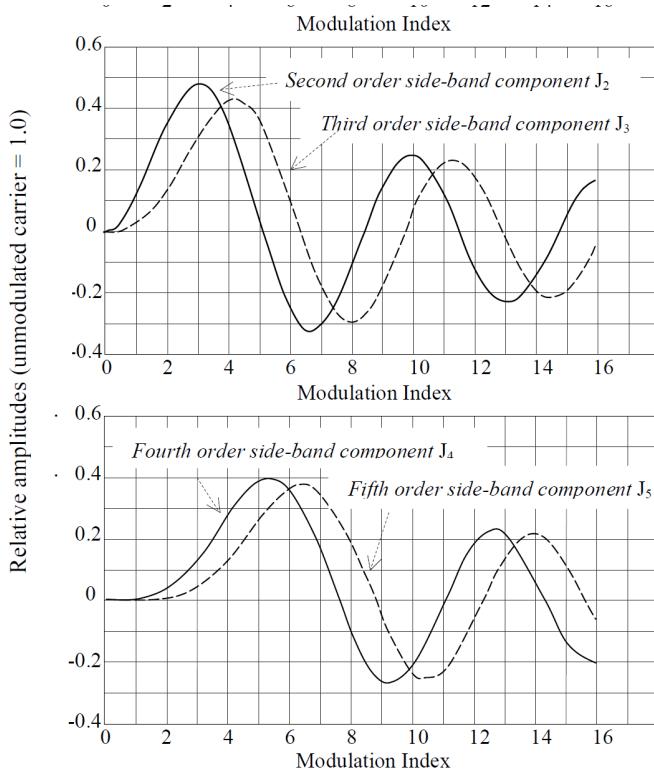


Figure 42: Gráfico de algunas funciones de Bessel de primera especie

A partir de lo anterior se deduce que a medida que se aumenta el índice de modulación I , más frecuencias entrarán en juego, cada frecuencia nueva que aparece lo hace de a pares y lateralmente a las frecuencias que ya existían previamente con un valor de I menor. Esto implicará que el ancho de banda de la señal en frecuencia aumentará a medida que I crece.

Este ancho de banda será aproximadamente igual a:

$$BW \approx 4\pi \cdot f_m(I + 1)$$

De aquí hacemos notar que, dado que el espectro estará centrado en la frecuencia portadora f_c (único factor sobreviviente cuando $I=0$), cuando $f_c < 2 \cdot f_m \cdot (I + 1)$, las frecuencias negativas entrarán en juego, reflejándose sobre las positivas y así se superpondrán componentes entre sí, alterando el espectro. En realidad, puede ser que se superpongan componentes como que no, y esto dependerá de la relación entre la frecuencia moduladora y la portadora. De esta manera, podrán también aparecer componentes inharmónicas al aumentar el I .

Nos preguntamos entonces cuándo aparecerán componentes inharmónicas y cuándo no, además de preguntarnos como determinar entonces la frecuencia fundamental de la señal en el tiempo y se pueden apreciar suficientes componentes harmónicas como para poder simular sonidos naturales.

La respuesta a estas preguntas resulta simple y la da el paper de Chowning, aunque se deduce fácilmente de la expansión trigonométrica de $x(t)$:

Si $f_c = N_1 \cdot f_o$ y $f_m = N_2 \cdot f_0$, entonces la frecuencia fundamental para el tono musical será f_0 y esta recién comenzará a prevalecer a partir de valores mayores a uno para el índice de modulación.

Sim en cambio, no hay una relación racional entre las frecuencias moduladora y portadora, entonces comenzarán a surgir frecuencias inharmónicas en el espectro, pudiéndose así lograr efectos subjetivamente placenteros al oído, como a su vez “arruinar” el sonido que se tenía originalmente. Como principalmente los instrumentos de percusión tienen componentes inharmónicas en su espectro, eligiéndose relaciones no racionales se podrá sintetizar sonidos percusivos, y eligiéndose relaciones racionales se podrá sintetizar sonidos de instrumentos para los cuales las componentes en frecuencia principales son harmónicas de la fundamental.

3.2 Síntesis de un clarinete

Se obtuvieron muestras de distintas notas musicales de un clarinete de la página de Philharmonia Orchestra <https://www.philharmoniaorchestra.com/>

Se procedió luego a intentar encontrar las funciones $A(t)$ e $I(t)$ para que la función del tiempo sintetizada se parezca lo más posible a las muestras en el tiempo de los clarinetes grabados.

Se muestra el gráfico en el tiempo de , nota que fue sampleada con una frecuencia de sampleo de 44100 Hz, estándar para CDs.

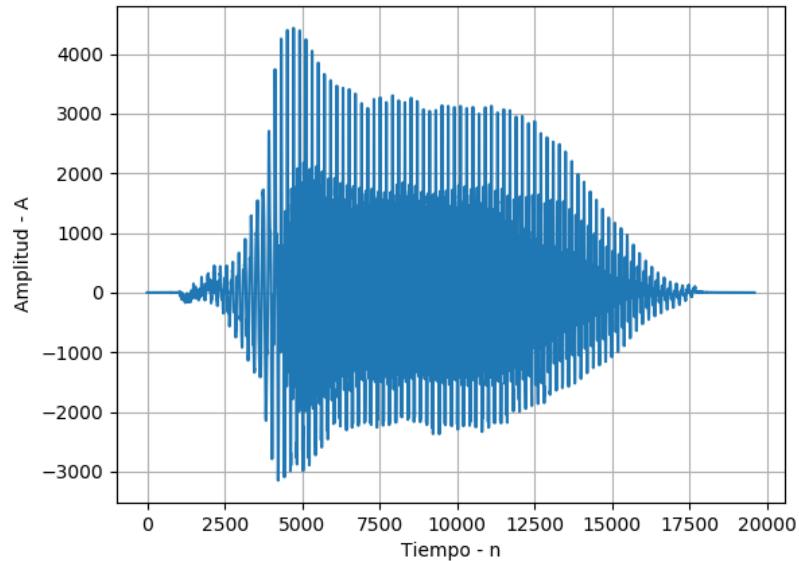


Figure 43: Nota de un clarinete, donde cada n corresponde a un tiempo $t = \frac{n}{f_s}$ segundos

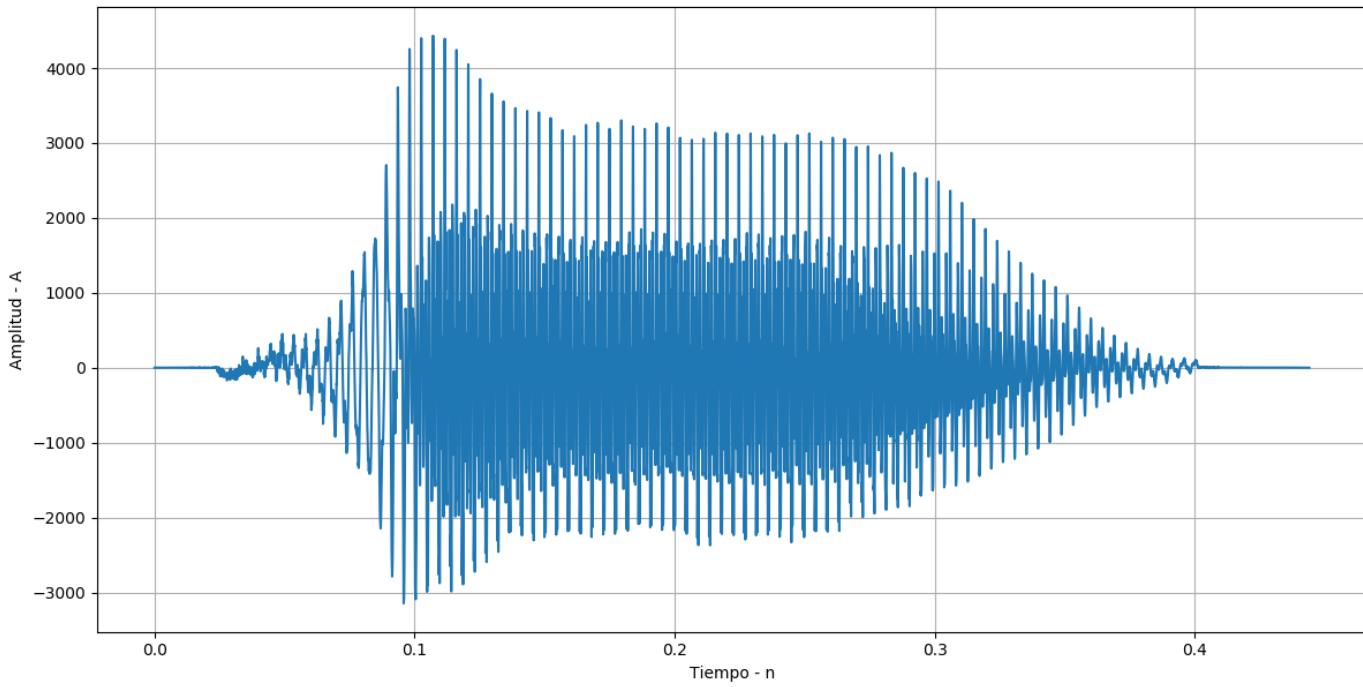


Figure 44: Nota de un clarinete en función del tiempo

Observamos de aquí principalmente 3 frecuencias que definen a este gráfico. Las hallaremos experimentalmente y veremos cómo el método de síntesis FM encarta correctamente con estas ocurrencias empíricas.

De este gráfico observamos que hay cuatro etapas fácilmente distinguibles para la nota, que denominaremos la etapa de Attack, Decay, Sustain y Release siguiendo la notación que nos impone el método de síntesis aditiva explicado en otros puntos.

Dado que el tiempo de sustain (llamamos tiempo de sustain al tiempo en que la nota se mantiene en el nivel de sustain fijado por el parámetro ADSR) es el tiempo preponderante en la nota y es el tiempo en que la nota se encuentra establecida, pudiéndose así considerarla en este intervalo como una función periódica del tiempo con frecuencia fundamental f_0 , usaremos a este intervalo como el intervalo representativo en tiempo de la función. Es en él en donde procederemos a hallar las frecuencias de importancia.

En un intervalo de menor longitud de la nota, procedemos entonces a hallar la frecuencia fundamental y la frecuencia cuya amplitud del orden de la de la fundamental:

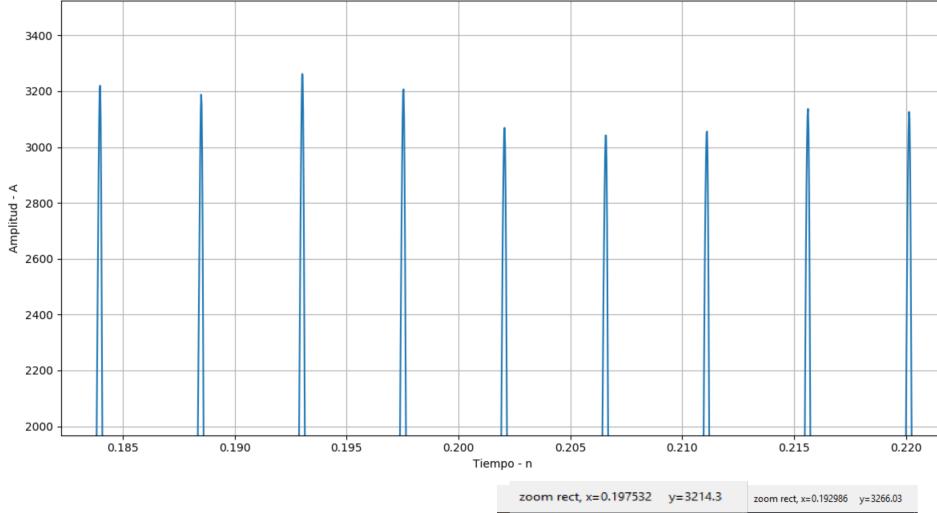


Figure 45: Obtención gráfica de las frecuencias de importancia de la nota

Dado $t_0=0.197532s$ y $t_1 = 0.192986s$ tiempos en los que se encuentra una repetición en amplitud, obtenemos el período fundamental como $\Delta t=t_1 - t_0$ y su consecuente frecuencia fundamental $f_0 = \frac{1}{\Delta t} = 219.9Hz$, lo cual coincide con el marco teórico de las notas musicales, bajo el cual la frecuencia de un A_3 resulta ser $f_{A_3}=220Hz \approx f_0$.

Luego, observamos la segunda frecuencia preponderante en la nota, lo cual nos arroja una frecuencia $f'_o = 882Hz$. Siguiendo la teoría del paper de Chowning explicada en la introducción, observamos que eligiendo $f_c=f_{A_3}$ y $f_m = 4 \cdot f_{A_3}$, obtendremos una frecuencia fundamental para nota de $f_0 = f_{A_3}$ y la segunda frecuencia preponderante $f_m = 882Hz \approx f'_o$. De aquí quedan entonces determinadas las constantes de consigna $n = 1$ y $m = 4$.

Además, observaremos entonces una desaparición de los armónicos pares (debido a $N2 = 4$), característico de los clarinetes, como veremos en mayor detalle en la sección de análisis de frecuencia.

Observamos luego una nota de igual duración pero distinta frecuencia, en particular un D4 de 0.25 segundos, forte, normal:

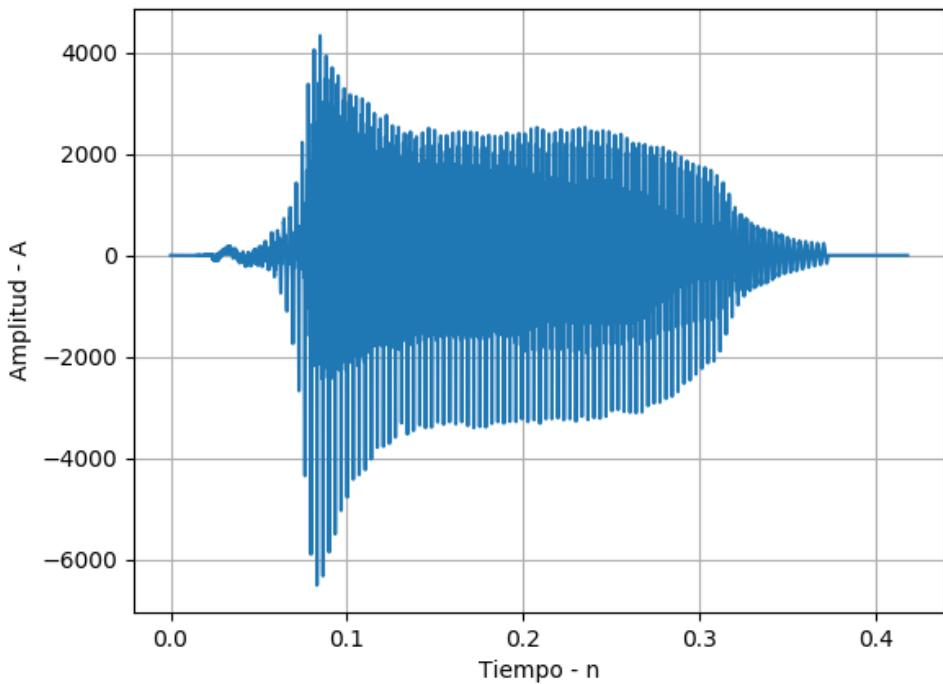


Figure 46: Nota de distinta frecuencia fundamental pero igual duración

Observamos también una nota de distinta duración pero igual frecuencia, en particular un A3 de 0.5 segundos, forte, normal.:

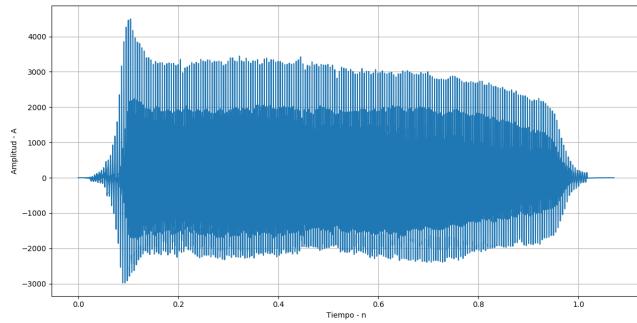


Figure 47: Nota de igual frecuencia fundamental pero distinta duración

Finalmente, observamos una nota de distinta duración y distinta frecuencia, en particular un E5 de 0.5 segundos, forte, normal:

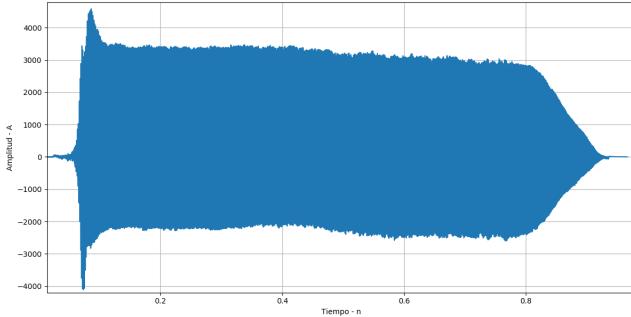


Figure 48: Nota de distinta frecuencia fundamental y distinta duración

De aquí extrapolamos que independientemente de las frecuencias fundamentales de las notas y de su respectiva duración, los tiempos de ataque, decay y release no sólo durarán aproximadamente lo mismo sino que estarán regidas por la misma forma matemática. Esto nos permite hacer un análisis aproximado de cómo deberá simularse la envolvente de cada nota en sus distintos estadíos por separado, y luego unir aquellas etapas para llegar al resultado final.

El tiempo de sustain, por otro lado, se expandirá o comprimirá según la duración de la nota pero tendrá la misma forma de envolvente, que se aproxima como constante.

Los próximos gráficos que se utilizarán para los análisis de las distintas etapas serán para la nota anteriormente mostrada, A_3 , elegida arbitrariamente.

3.2.1 Determinación de las envolventes

Dado el gráfico de la nota en función del tiempo, se pueden identificar dos zonas de densidades muy diferentes (distintas frecuencias). La zonas de mayor densidad (o mayor frecuencia) se corresponderán gráficamente con la frecuencia modulante f_m , que es 4 veces superior a la f_c , que a su vez estará asociada a las zonas de menor densidad.

Es de aquí que surge la idea de que la envolvente $I(t)$ fijará la amplitud de la zona de baja densidad, ya que está fijando la envolvente del coseno de frecuencia central f_c , mientras que la envolvente $I(t)$, en cambio, alterará la envolvente de la zona de alta densidad, la de frecuencia f_m .

3.2.2 Análisis del Attack

Se analiza la envolvente del attack de la nota de manera general:

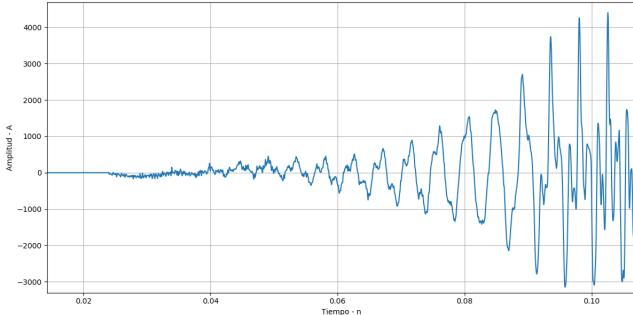


Figure 49: Attack de un clarinete

Se observa para la envolvente un crecimiento exponencial de la forma $A_1(t)=e^{\frac{t}{\tau_1}} - 1$ donde τ_1 es constante.

Dado que la amplitud máxima de la nota estará fijada por el último tiempo t del attack y que genéricamente $A(t) \in [-A_{máx}; A_{máx}]$, donde $A_{máx}$ está fijado por la cantidad de bits utilizados para representar la amplitud en el .wav (en nuestro caso 16 bits), se asociará a $A_{máx}$ con la amplitud máxima de la nota real y luego se realizará el apropiado escalamiento para adaptarlo al sistema digital.

Procedemos calcular entonces el valor de τ_1 empíricamente, aproximando a la duración del tiempo attack Δt_{att} como 4000 samples a 44100Hz de sampleo, es decir, $\Delta t_{att}=0.091$ segundos.

De este tiempo, despejamos $\tau_1 = \frac{\Delta t_{att}}{\ln(1+A_{máx})}$, lo que arroja un resultado en tiempo de $\tau_1 = 0.0108s$, al haber aproximado $A_{máx} = 4200$ para el .wav

Se muestra el gráfico de la exponencial aproximadora superpuesta con el attack real.

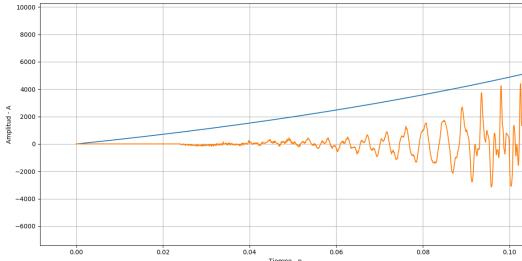


Figure 50: Attack y envolvente aproximada superpuestos. $A(t)$ en azul.

Luego, para la zona de mayor densidad o frecuencia, la envolvente $I(t)$ estará fijada por un crecimiento exponencial durante el attack de la forma $I(t)=I_0 \cdot (e^{\frac{t}{\tau_1}} - 1)$. Dicha fórmula es extraída de la forma exponencial de los gráficos del paper de Chowning.

Se determinó I_0 experimentalmente mediante prueba y error, obteniéndose resultados más realistas para aquellos valores en los que $I_0 < 0$.

3.2.3 Análisis del Decay

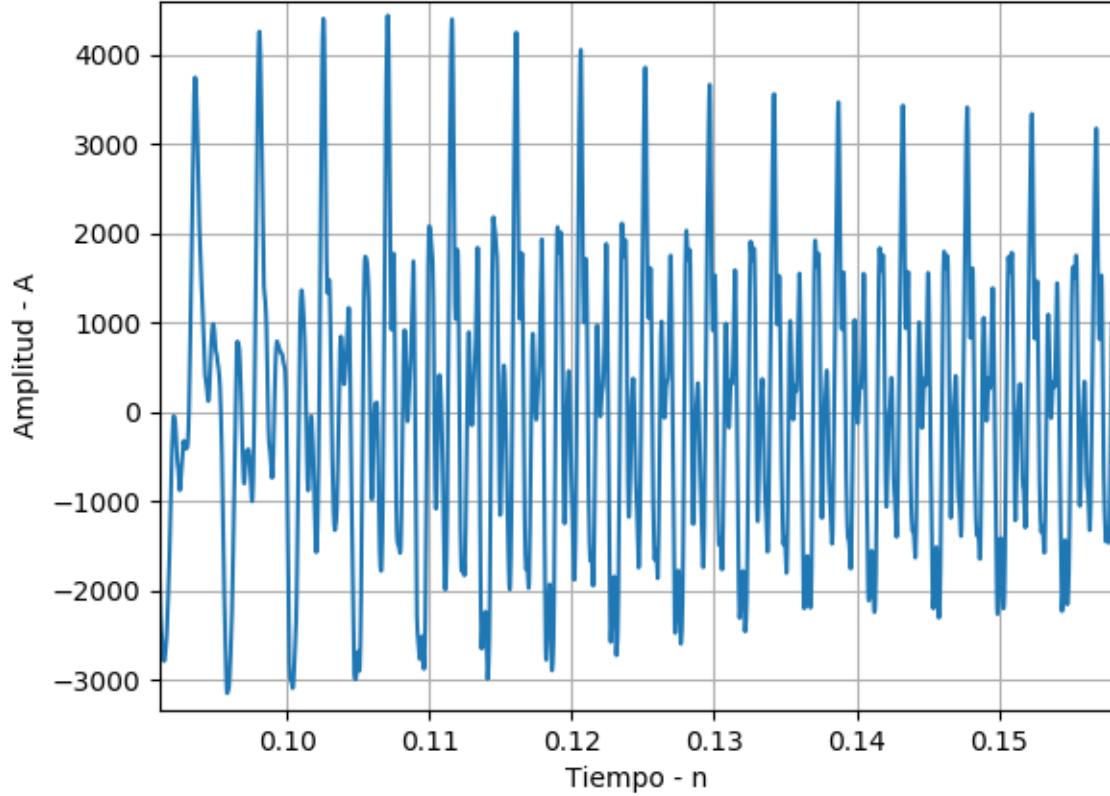


Figure 51: Decay de un clarinete

Se observa para la envolvente un decaimiento exponencial de la forma $A_2(t) = A_{máx} \cdot e^{-\frac{t}{\tau_2}}$ donde tanto $A_{máx}$ coincide con el valor fijado por el attack.

Determinamos $\Delta t_{dec} = 0.074$ segundos, donde Δt_{dec} es la duración de la etapa de decay. Teniendo en cuenta que la etapa de decay decae exponencialmente hacia la constante de sustain, se despeja $\tau_2 = -\frac{\Delta t_{dec}}{\ln(\frac{A_{sus}}{A_{máx}})}$ como la constante de tiempo de la etapa de decay. En la última expresión, A_{sus} resulta ser la amplitud de la etapa de sustain, que se aproxima a $A_{sus} = 3000$.

3.2.4 Análisis del Sustain

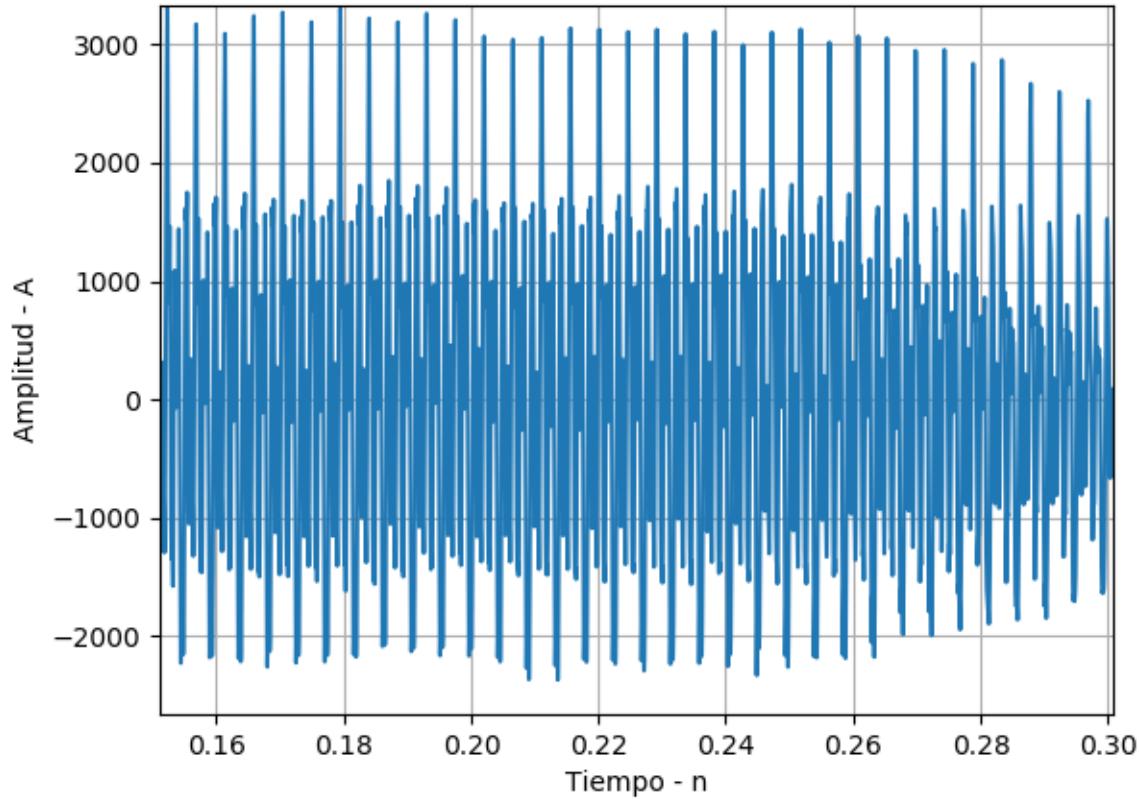


Figure 52: Sustain de un clarinete

Se observa una envolvente constante, de la forma $A_3(t) = 0.71 \cdot A_{máx} = A_{sus}$

El tiempo en que la nota se mantiene en el nivel de sustain estará dado por $\Delta t_{sus} = \Delta t_{nota} - \Delta t_{rel} - \Delta t_{dec} - \Delta t_{att}$, donde Δt_{nota} la duración total de la nota, Δt_{rel} la duración del release, Δt_{dec} la duración del decay, y Δt_{att} la duración del Attack.

Los valores de $I(t)$ resultan ser constantes para la etapa de sustain.

3.2.5 Análisis del Release

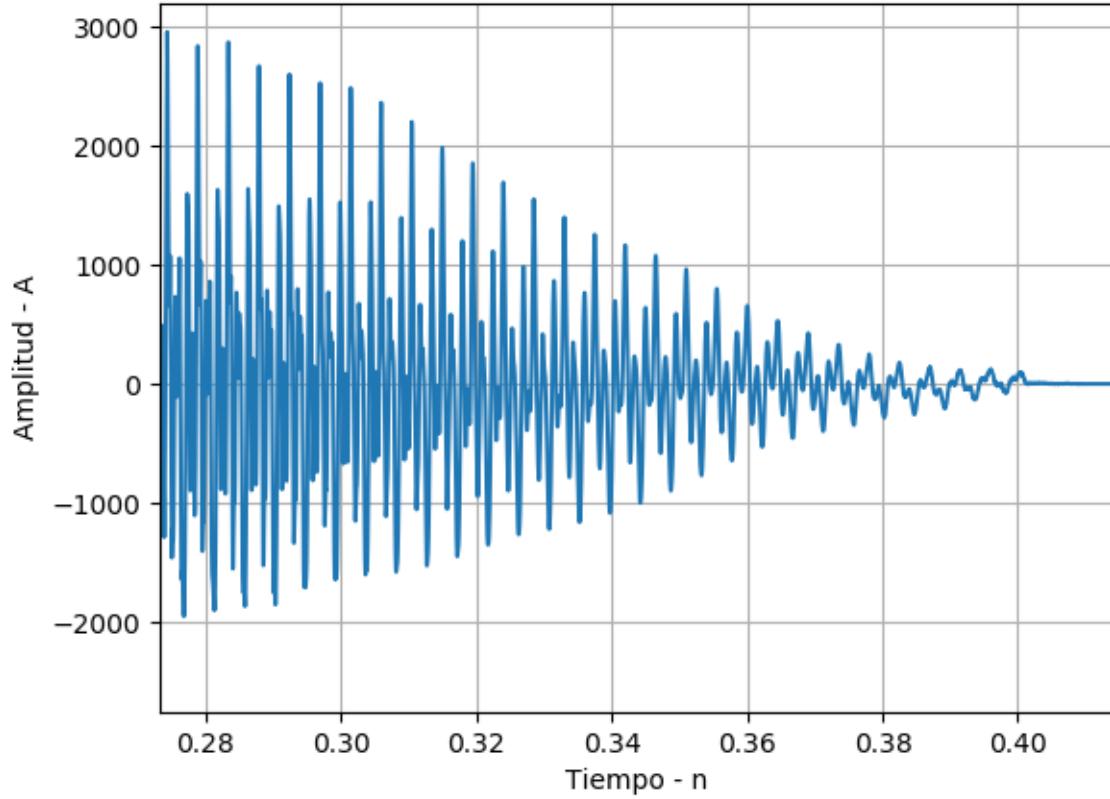


Figure 53: Release de un clarinete

Al igual que con la etapa de decay, se observa un decaimiento exponencial de la forma $A_4(t) = A_{sus} \cdot e^{-\frac{t}{\tau_4}}$, donde, fijando al A_f la amplitud final de la nota como $A_f = \frac{A_{sus}}{32}$, la constante de tiempo queda determinada como $\tau_4 = -\frac{\Delta t_{rel}}{\ln(\frac{A_{sus}}{A_f})}$. Volviendo a citar el paper de Chowning, se eligió valores de $I(t)$ constantes e iguales a los del sustain durante el release.

3.2.6 Resultados intermedios

Con los razonamientos utilizados hasta aquí, se llega al siguiente gráfico en función del tiempo para la señal sintetizada.

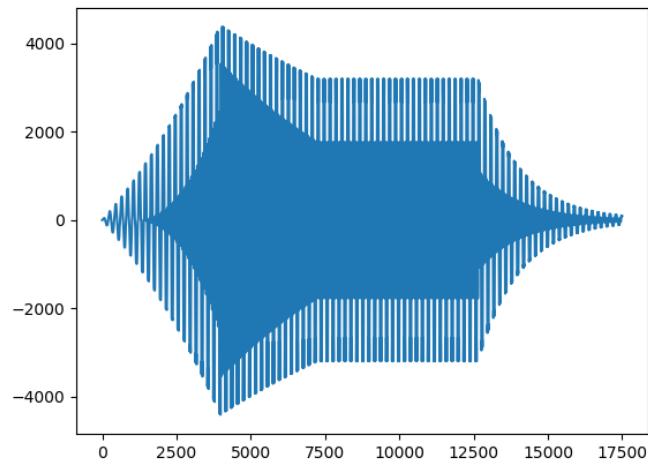


Figure 54: Nota sintetizada con el método, versión no final

Se muestra también el gráfico de la nota original para contrastar.

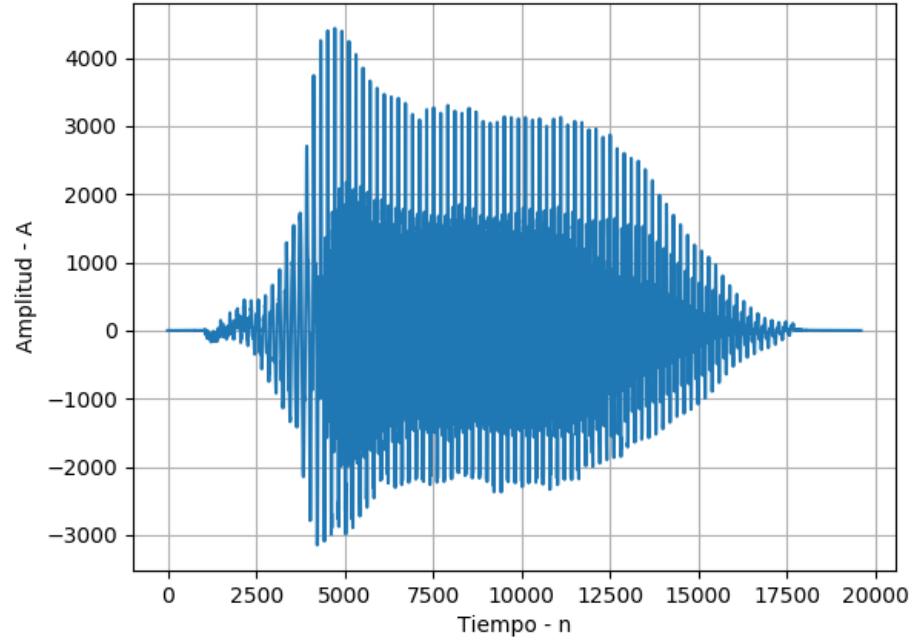


Figure 55: Nota original

Se logran apreciar las diferentes amplitudes para las zonas densas y no densas, que resultan ser del orden de las

originales o dentro del mismo rango.

Las envolventes, si bien aproximada por fórmulas matemáticas básicas, parecen seguir a la señal original correctamente.

Se analizan cortos intervalos de tiempo en la etapa de sustain para la señal original y la

3.2.7 Agregado de frecuencias inharmónicas para efectos más realistas

En el gráfico de la señal, podemos observar la existencia de zonas de densidad intermedia de mediana amplitud. Con el objetivo de hacer aparecer estas nuevas frecuencias, sumamos a la frecuencia modulante una frecuencia baja de $0.3Hz$ de manera tal que $f_m=4 \cdot f_c + 0.3Hz$. El paper de Chowning sugiere una frecuencia de $0.5Hz$ pero observamos que los $0.3Hz$ coincidían de mejor forma con la señal sintetizada por las envolventes definidas.

A continuación se muestran las dos señales sintetizadas y se comparan las densidades agregadas con la original para poder corroborarlo.

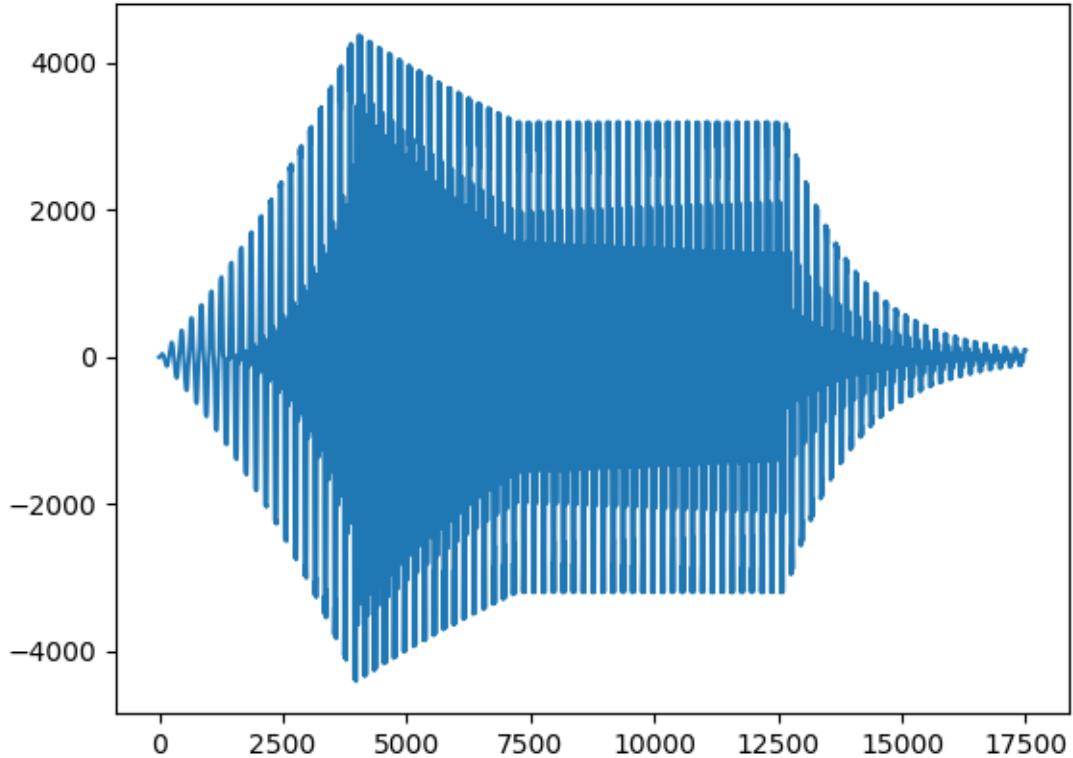


Figure 56: Frecuencia agregada de $0.3Hz$

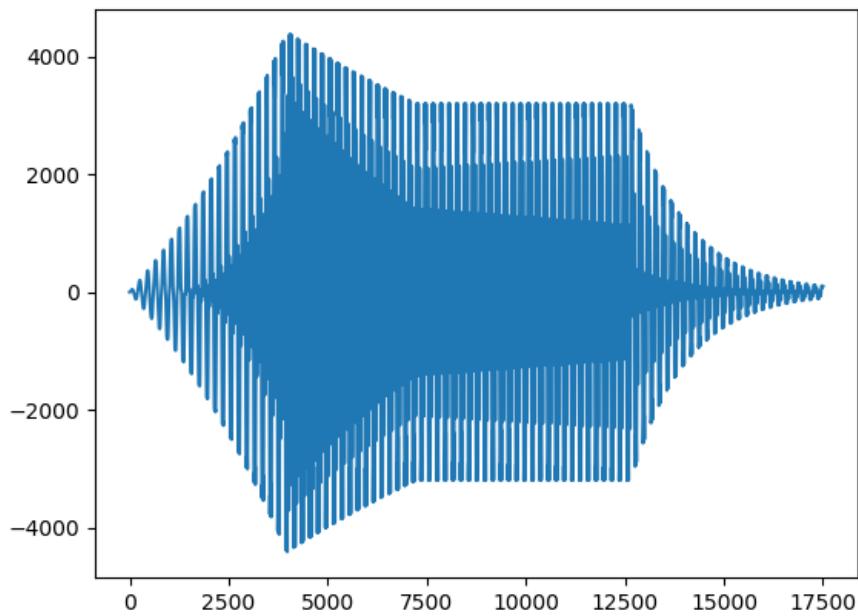


Figure 57: Frecuencia agregada de 0.5Hz

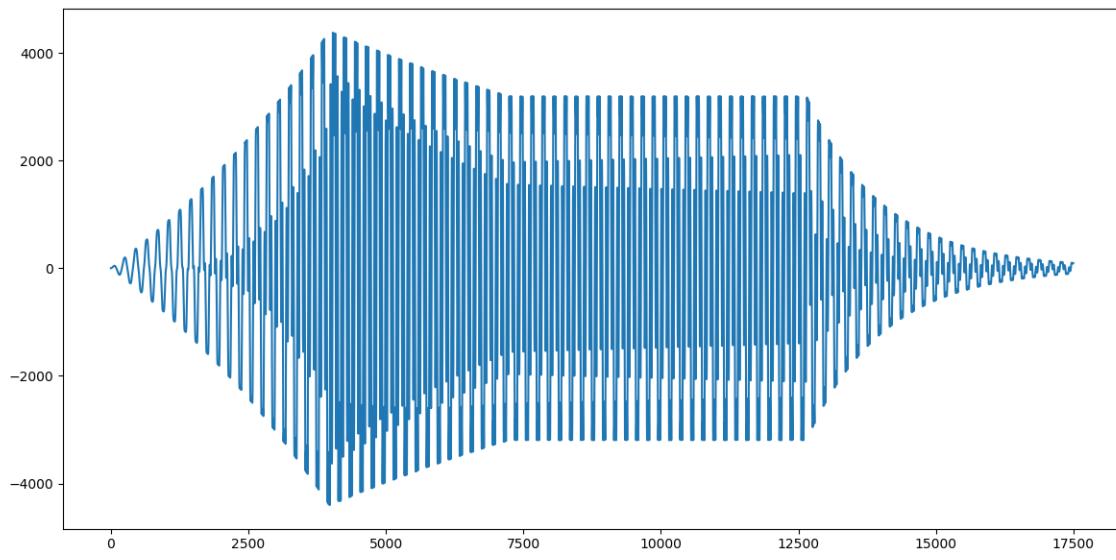


Figure 58: Nota sintetizada, versión final

3.2.8 Resultados finales

Para elegir los parámetros sintetizadores, se probó con notas sueltas y de distinta duración hasta llegar al resultado final. La constante I_{sus} de sustain resultó valer $-\frac{4}{3} \cdot A_{sus}$, teniendo entonces un valor negativo.

Una vez elegidos los parámetros de envolvente y frecuencia mencionados en las subsecciones anteriores, se sintetizó el duo para clarinete y fagot de Beethoven con resultados exitosos.

Se notó que la velocidad de síntesis de este método con respecto a otros, como el de síntesis a partir de muestras, es demás superior, en detrimento de una peor calidad de sonido.

3.2.9 Análisis en frecuencia

Se procede a realizar un espectrograma de la nota sintetizada y se la compara con la nota original que se intentó imitar.

Los gráficos mostrados permiten visualizar frecuencias no mayores a los 2.5kHz porque el espectrograma de ambas notas muestra que no hay componentes significativas en un rango superior.

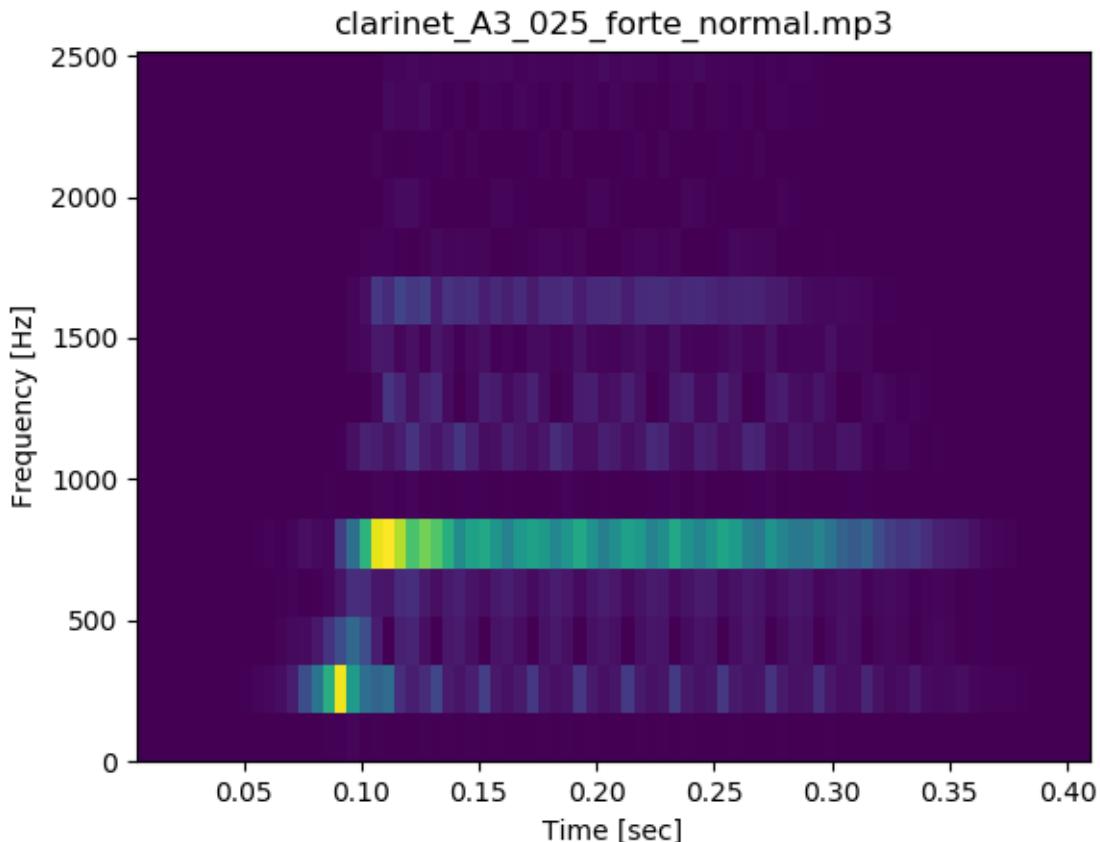


Figure 59: Nota original

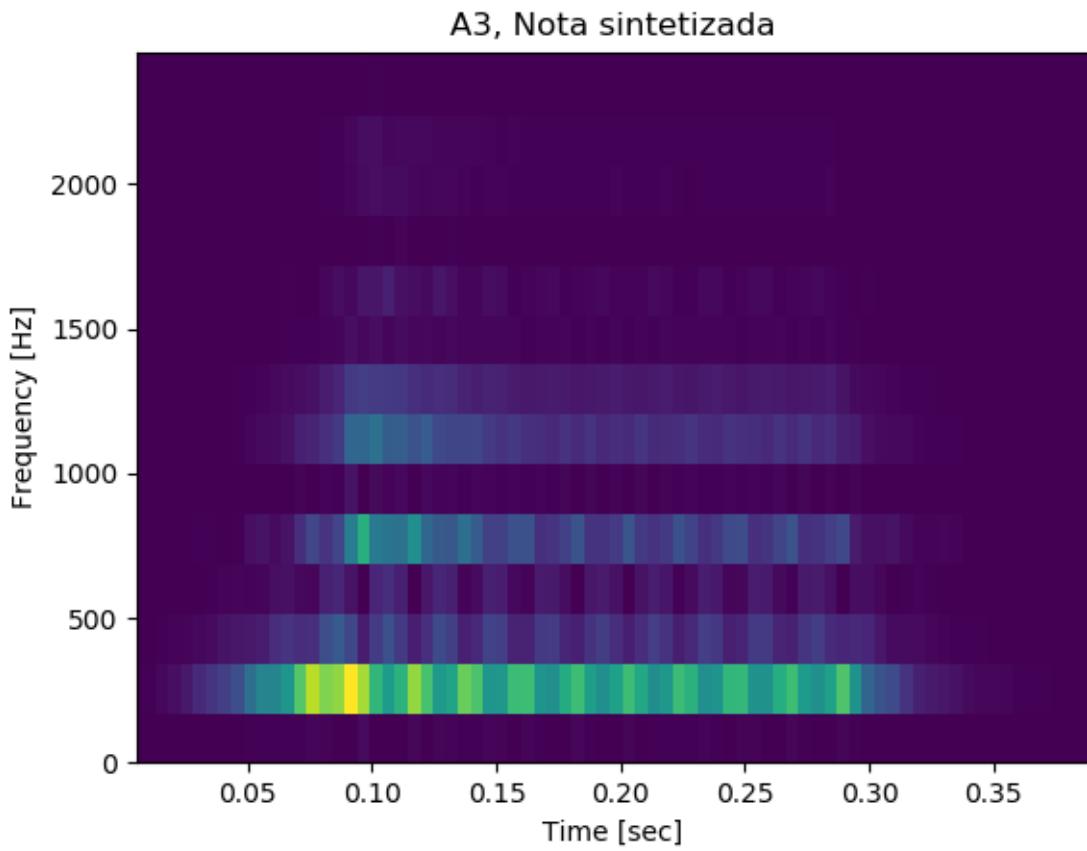


Figure 60: Nota sintetizada

Se observa que las líneas espectrales en las que hay amplitud coinciden correctamente, incluso en aquellas en las que las frecuencias tienen amplitudes cercanas a las nulas como en la franja de los 1kHz.

Sin embargo, se observa que si bien la nota que se muestreó afirma ser un A3, la frecuencia más importante/significativa del gráfico resulta ser la de 880 Hz, la que se eligió como frecuencia modulante a la hora de sintetizar.

Como se tuvo la intención de hacerlo, la nota sintetizada muestra su franja más significativa de frecuencias centrada en los 220Hz. Un análisis más intenso en frecuencia y en teoría musical podría explicar por qué la muestra evoluciona hacia unos 880 Hz a medida que avanza en el tiempo, pero en este artículo nos limitamos a observar que las franjas de frecuencias coinciden y resultan satisfactoriamente similares.

3.2.10 Conclusión

Si bien el método de síntesis FM no resulta completo en el sentido de que la nota sintetizada a partir de él resulta ser imperfecta, sí se logran buenos resultados a partir de la misma y al combinarse con otras técnicas (como la de ADSR) de síntesis permite obtener resultados simples matemáticamente y ágiles en cuanto a cálculo, con un mayor color de tono al que se obtiene de otros métodos de síntesis más antiguos.

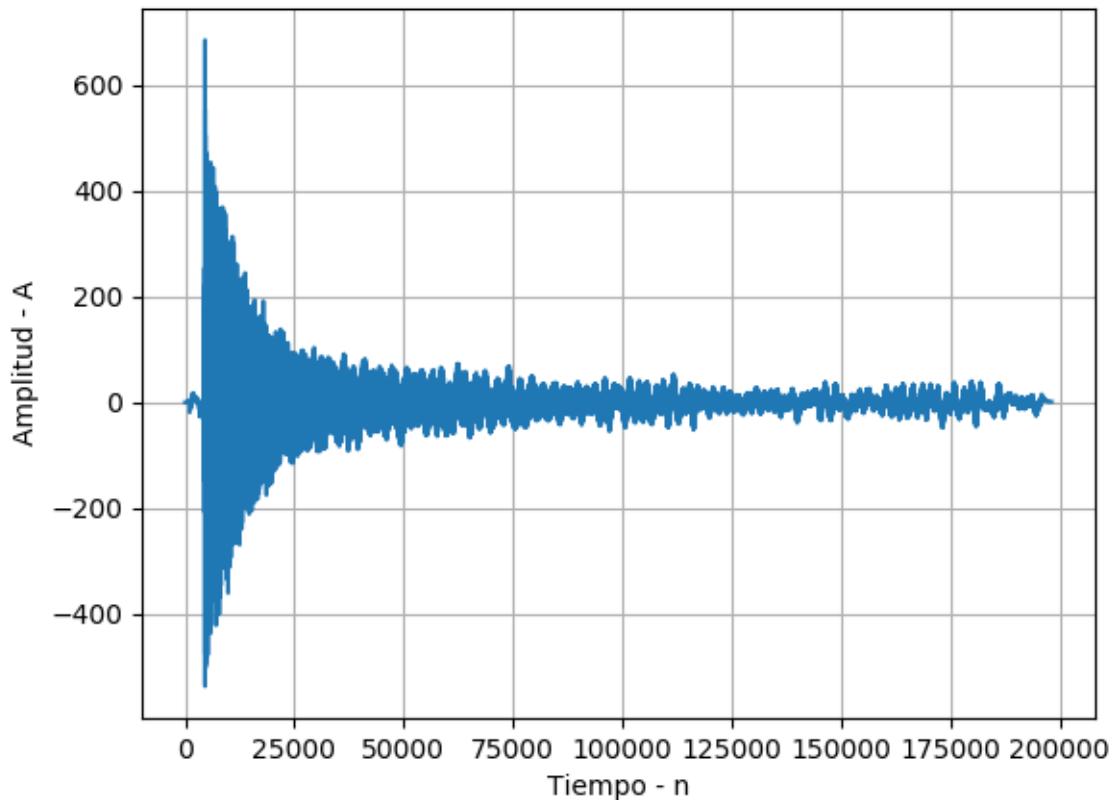


Figure 61: Triángulo

Por todo lo expuesto en el párrafo anterior, es fácil comprender el éxito de este método de síntesis en la música y los videojuegos de los años 80.

3.3 Síntesis de una campana

Según el paper de Chowning, las campanas tendrán envolvente exponencial decreciente y también lo tendrá el índice de modulación.

Esto se confirma al observar el sonido de un triángulo (sonido de percusión parecido al de una campana) grabado por la Philharmonia Orchestra.

Como se mencionó anteriormente, dado que el triángulo es un instrumento de percusión, presentará componentes inharmonicas en frecuencia, lo cual es fácilmente observable en el gráfico en los picos erráticos de la función del tiempo recién mostrada.

Sin embargo, las campanas tienen un comportamiento más melódico que otros instrumentos de percusión como la batería, y es por esto que el índice de modulación no deberá ser demasiado grande, ya que estoy traería demasiadas

inharmonías al sonido.

Luego de varios intentos y simulaciones, se decidió fijar $f_c = 440Hz$ y $f_m = 1.4 \cdot f_c$ como las frecuencias en las que el sonido de la campana resulta más real y agradable al oído. Para frecuencias menores también se logran resultados interesantes, pero a partir de cierto valor de frecuencia a medida que se va subiendo en el espectro se comenzará a conseguir un sonido más robótico o menos natural.

Para las fórmulas de las envolventes, se siguió las sugerencias de Chowning, haciendo a $I(t)$ proporcional a $A(t)$, y fijando

$$A(t) = A_{máx} \cdot e^{-\frac{t}{\tau}}, \text{ por lo que } I(t) = I_0 \cdot e^{-\frac{t}{\tau}}, \text{ con } I_0=0.5 \text{ y } A_{máx} \text{ la intensidad inicial del sonido.}$$

La constante de tiempo se fijó de manera tal que

A continuación se muestra un gráfico de la señal sintetizada en tiempo:

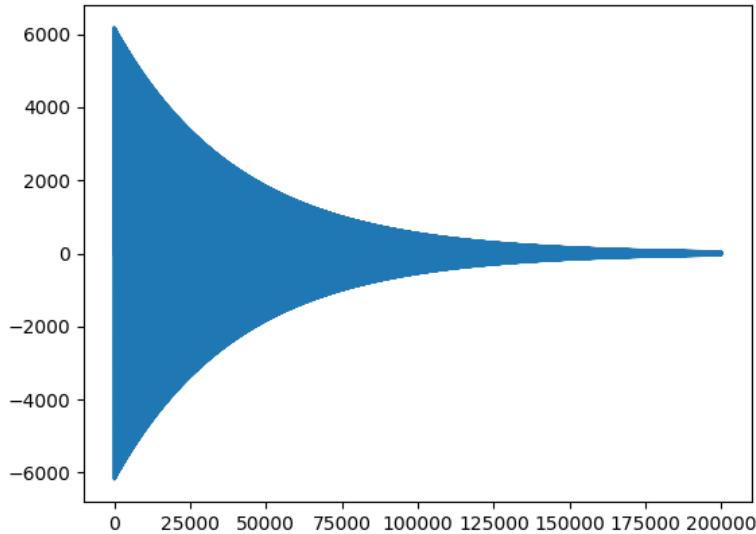


Figure 62: Señal sintetizada

Finalmente, se muestra un acercamiento en tiempo para mayor comprensión:

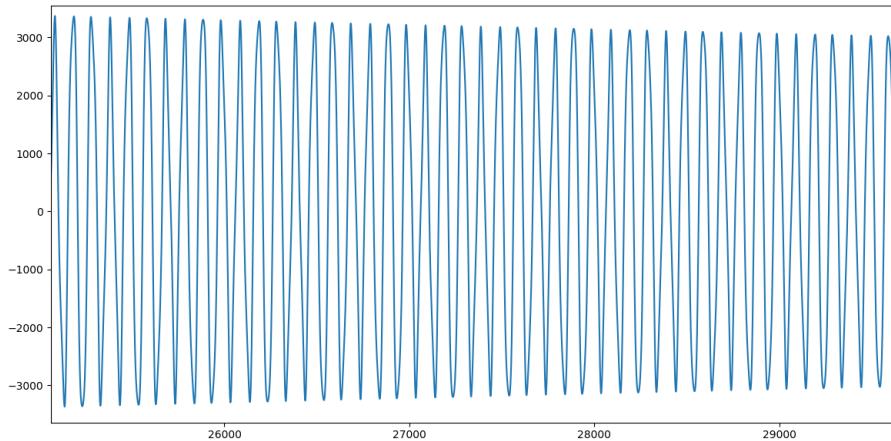


Figure 63: Acercamiento en tiempo de la señal sintetizada

3.4 Síntesis de una trompeta

Notamos que para la síntesis de la trompeta, Chowning describe el mismo tipo de envolvente que fue utilizada para la síntesis del clarinete.

Debido a que el paper indica que instrumento como la trompeta contienen todas las componentes armónicas (pares e impares), se utiliza $N1 = 1$ y $N2 = 2$.

Ya habiendo explicado el razonamiento a utilizar, procederemos a hacer el análisis gráfico de las notas de una trompeta. En particular, A3 forte normal.

Con los mismos razonamientos que para el clarinete, se obtuvieron los tiempos de Attack, Decay, Sustain y Release:

- Tiempo de attack de 0.16 segundos.
- Tiempo de Decay de 0.034 segundos.
- Tiempo de Release de 0.226 segundos.

Se observan las siguientes características notables para la nota:

1. La nota tiene zonas de mayor y de menor densidad, como era de notar para el clarinete.
2. La nota es notablemente asimétrica en amplitud: Las amplitudes positivas alcanzan el triple del máximo de lo que alcanzan las negativas.
3. La zona de mayor densidad de la nota se encuentra mayormente del lado negativo que del positivo.
4. Las zonas de mayor densidad tienen baja amplitud con respecto a las de baja densidad, siendo esta diferencia de amplitud mucho más grande que con el clarinete.

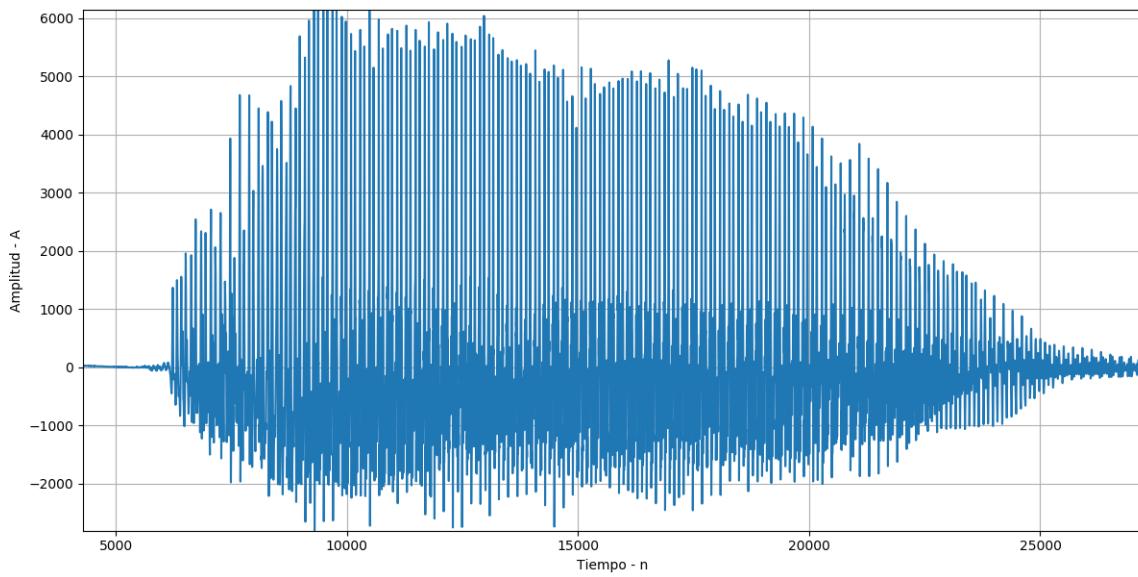


Figure 64: Nota de una trompeta

5. El attack de la nota resulta aproximadamente del mismo orden que el clarinete, pero el decay es más rápido y el release más lento.
6. Al igual que con el clarinete, aparecen ciertas frecuencias bajas montadas sobre la señal, esto estando reflejado en aquellas zonas de densidad intermedia.

Las observaciones anteriores llevan a las siguientes conclusiones, que fueron aplicadas a la hora de realizarse la síntesis:

- Debido a la notable diferencia en amplitud entre zonas de distinta densidad/frecuencia, se eligió un índice de modulación bajo, que haga que las zonas de alta frecuencia tengan menor amplitud. El factor Io que se menciona en la síntesis del clarinete resultó ser igual para todas las etapas ADSR para asegurar “continuidad” en la envolvente de cada zona de densidad alta. Luego de experimentar con las distintas posibilidades, se eligió un $Io=-1$. Se muestra el cambio en la amplitud de las zonas de densidad con distintos valores de Io .

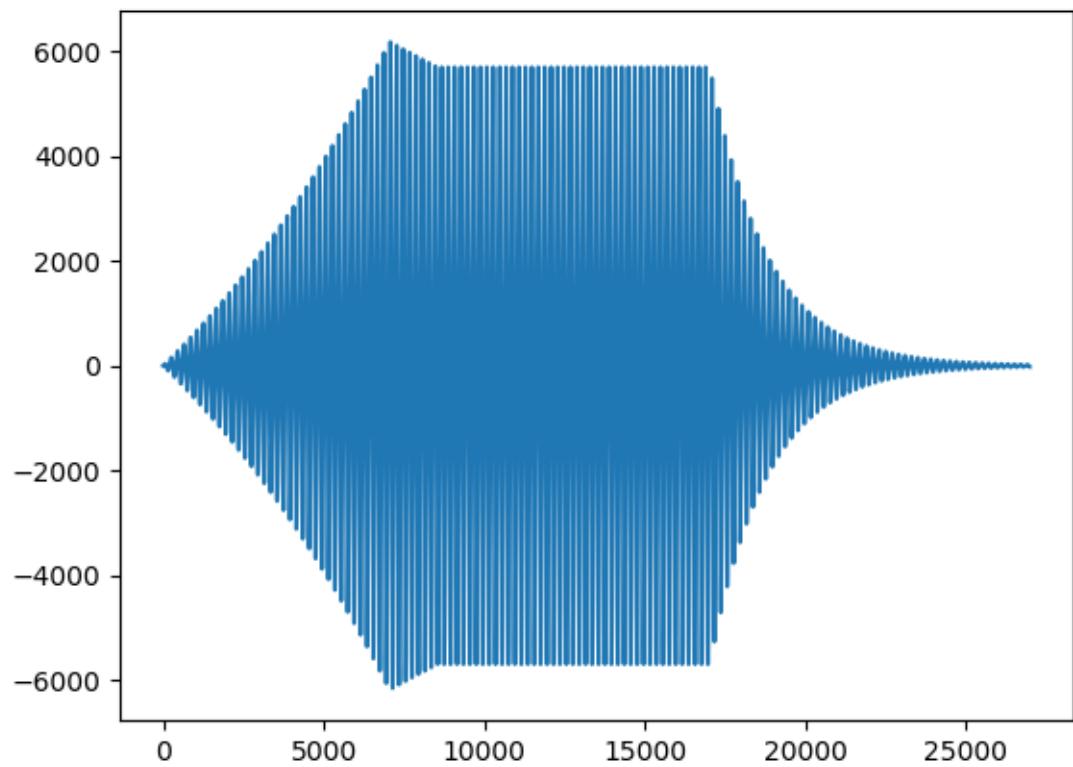


Figure 65: $\text{Io} = -0.5$

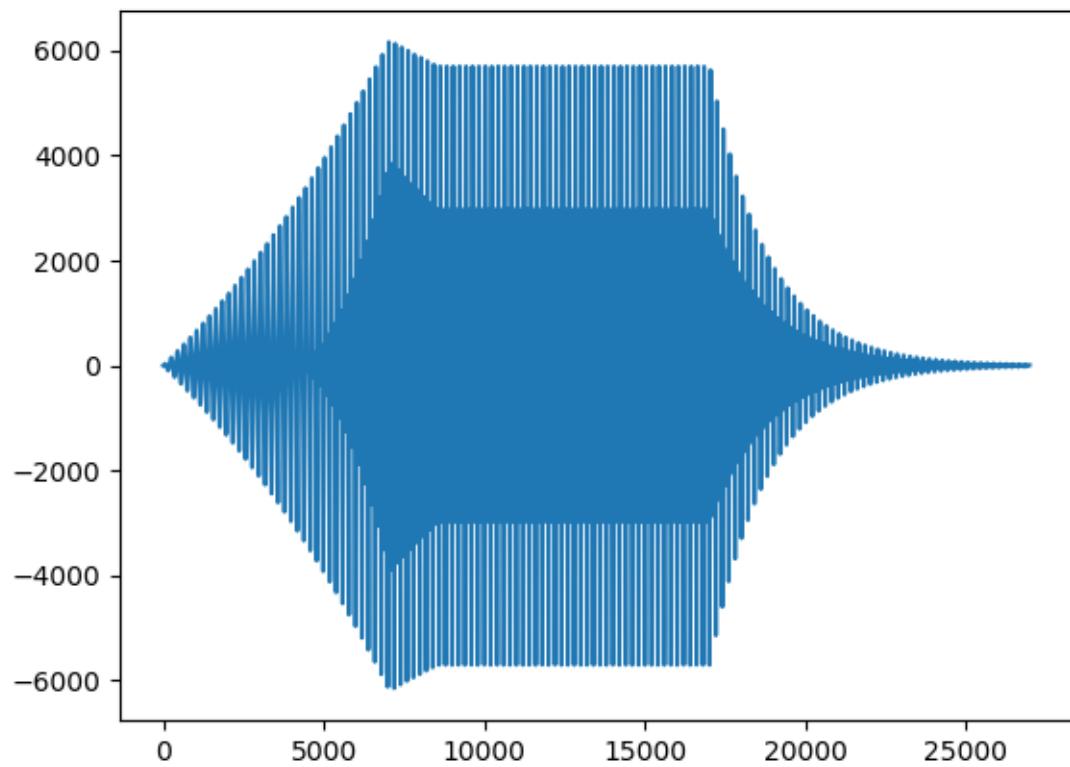


Figure 66: $\text{Io} = 2$

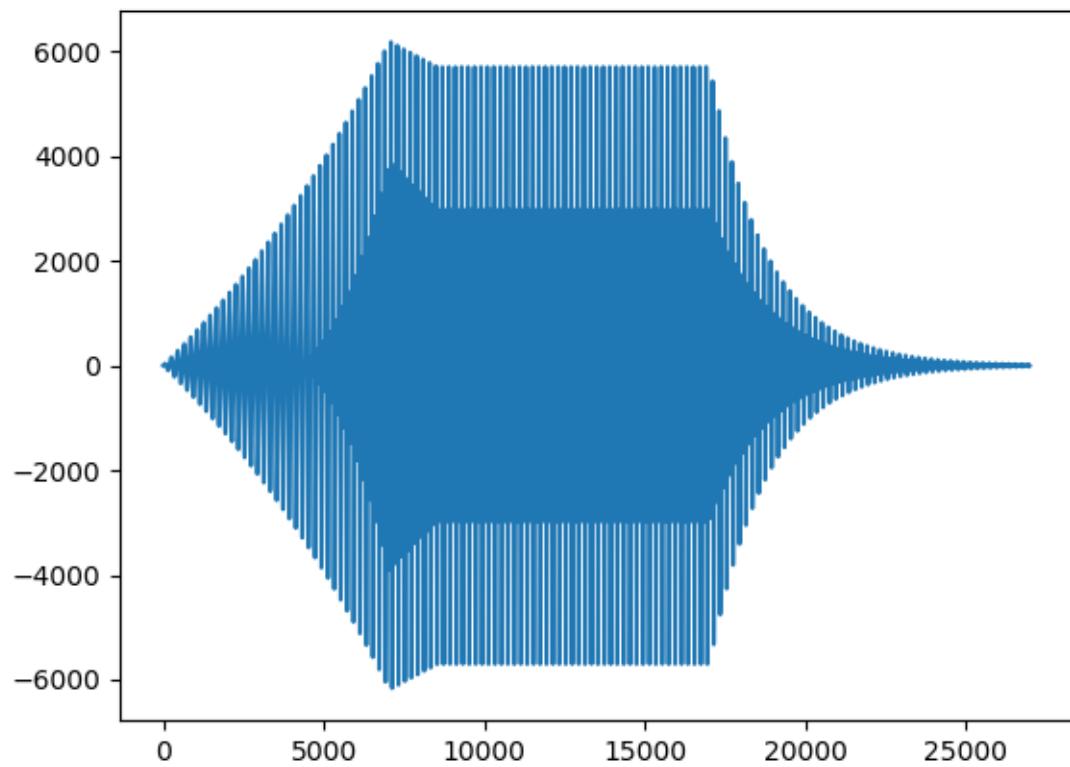


Figure 67: $\text{Io}=-2$

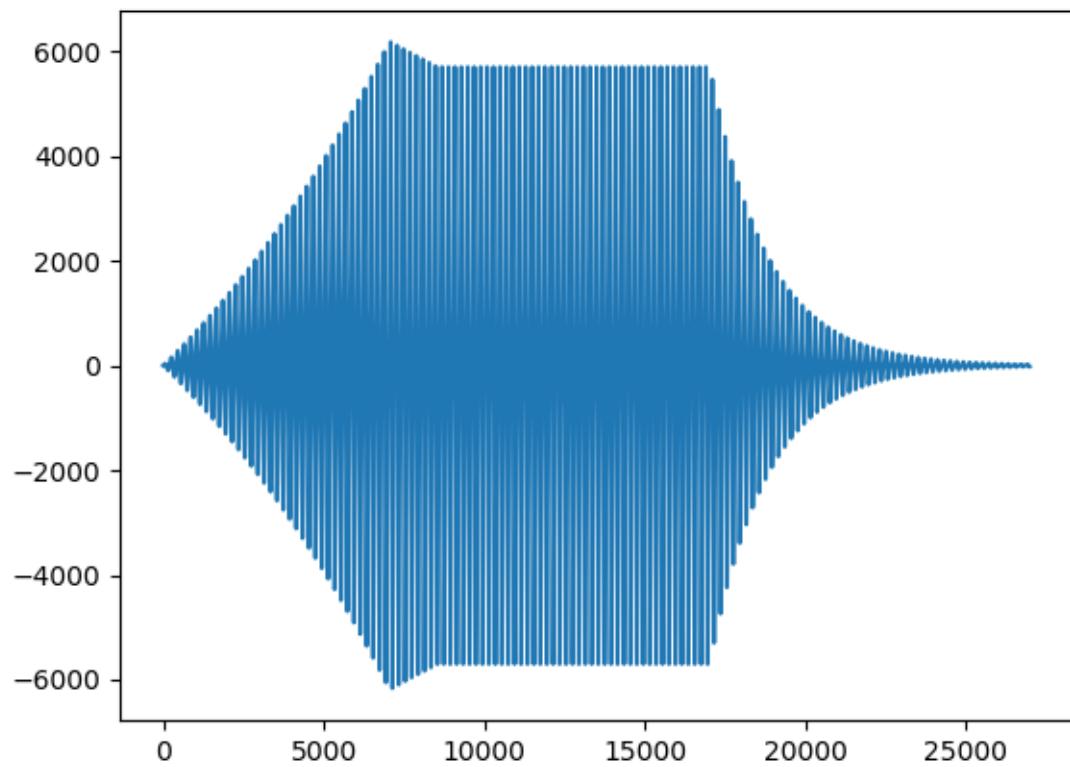


Figure 68: $\text{Io} = -1$. Se ve que las zonas de alta densidad tienen baja amplitud con respecto a las de baja densidad

- Como las amplitudes mínimas son tres veces más chica en valor absoluto que las amplitudes máximas para la nota, se eligió escalar las amplitudes negativas en ese factor a la hora de sintetizar la nota. A continuación se muestra el resultado de dicha operación.

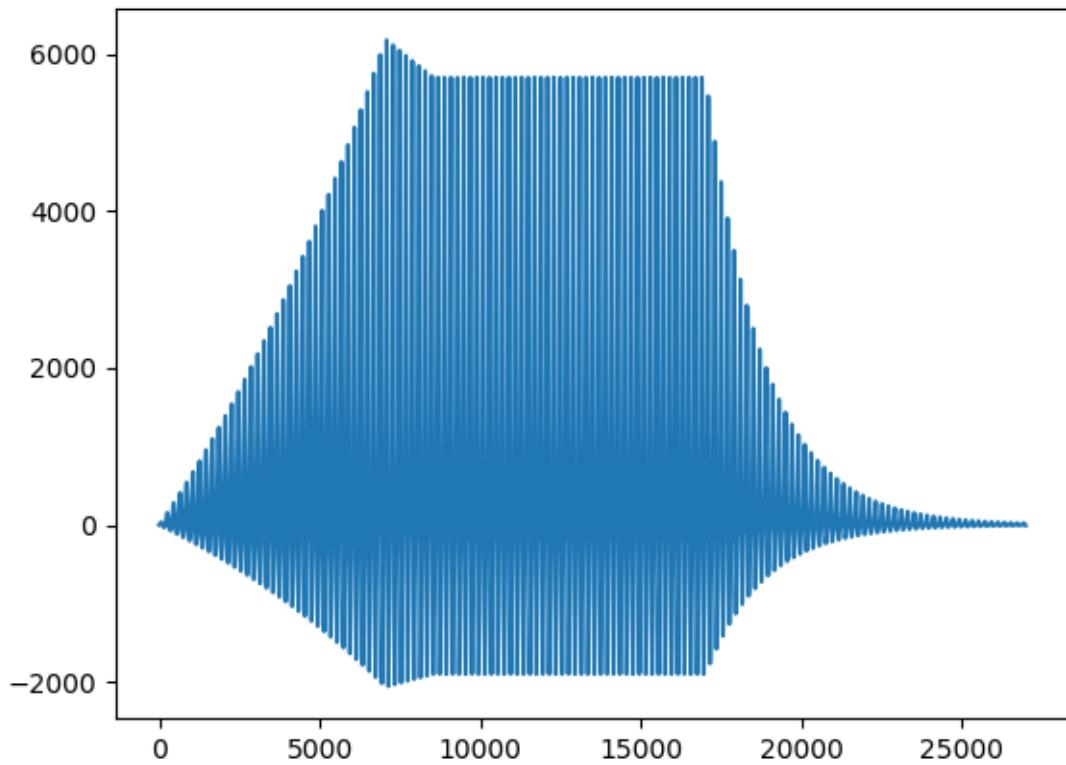


Figure 69: Nota sintetizada con amplitudes negativas escaladas

- Habiendo notado que la zona de mayor densidad se ubica mayormente del lado negativo de las amplitudes, y habiendo notado también la presencia de frecuencias agregadas como en el clarinete, que se muestran en zonas de densidad intermedia, se busca una relación entre los dos factores. Se descubre que cuando se suma una frecuencia baja (del orden de los 0.3Hz) a la frecuencia modulante, la zona de mayor densidad baja en amplitud (se hace más negativa) en la etapa de sustain. De lo contrario, si se resta una frecuencia baja, la zona de mayor densidad se hace más positiva en amplitud. Se muestran gráficos representativos de esta situación

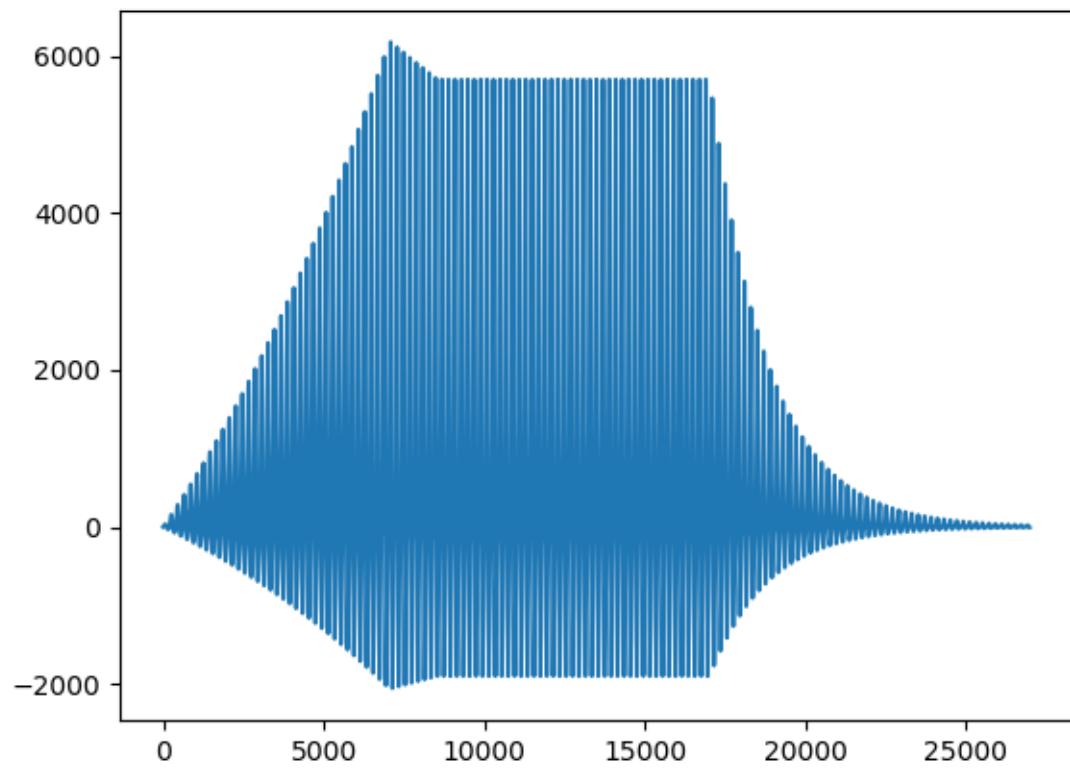


Table 6: Nota sintetizada con parámetros ADSR (versión no final)

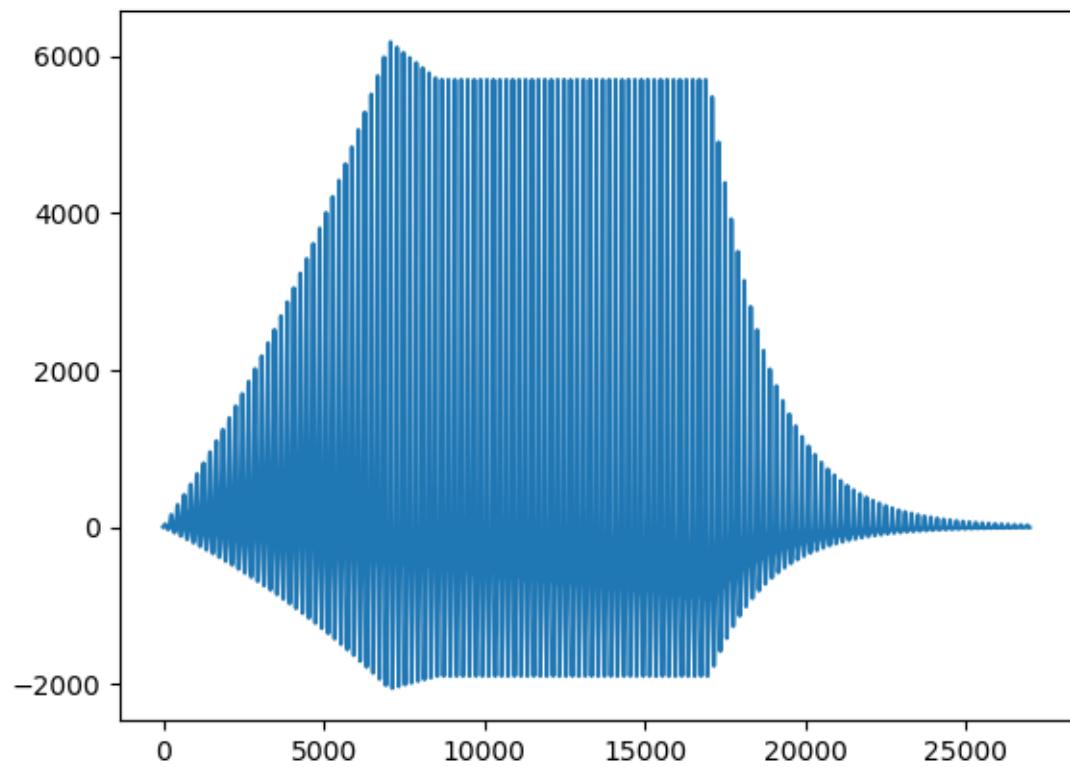


Figure 70: Nota mostrada anteriormente con $f'_m = fm + 0.2Hz$

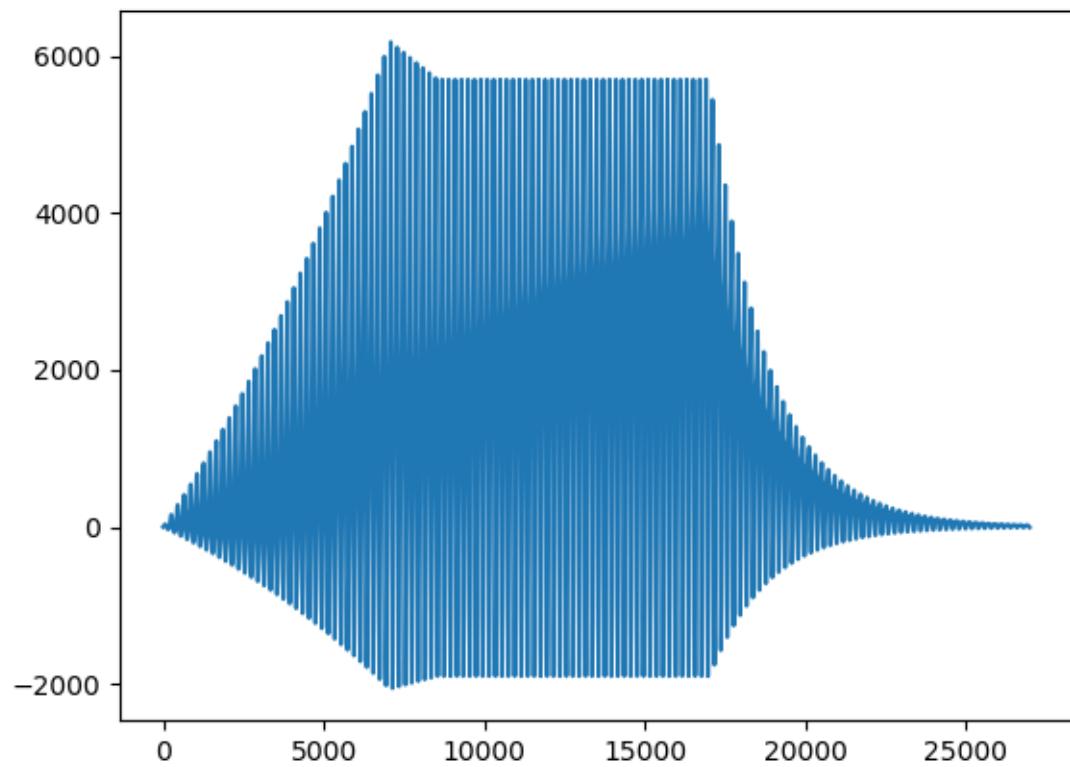


Figure 71: Nota mostrada anteriormente con $f'_m = fm - 0.3Hz$

Finalmente se llega a la versión final de la nota:

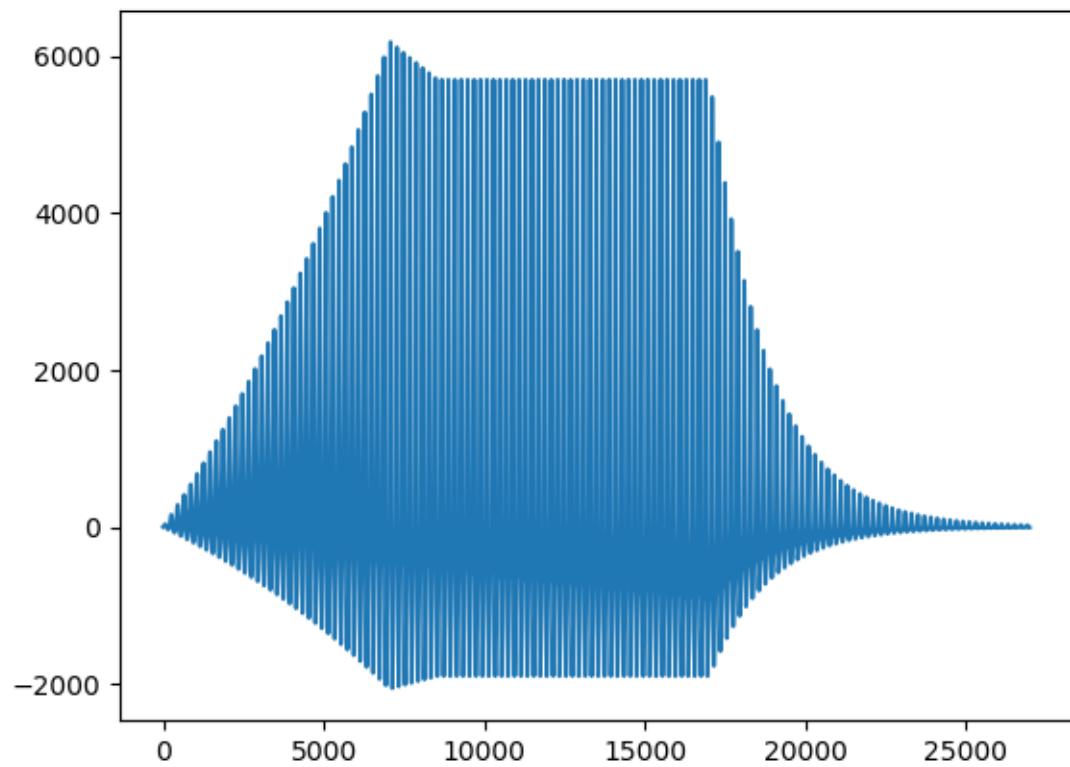


Figure 72: Nota sintetizada, versión final

4 Síntesis mediante modelo físico (Karplus-Strong)

4.1 Modelo Básico

4.1.1 Función Transferencia

El algoritmo de Karplus-Strong se utiliza para modelar instrumentos a partir de la simulación del modelo físico del mismo.

La siguiente figura ilustra el modelo conceptual del algoritmo.

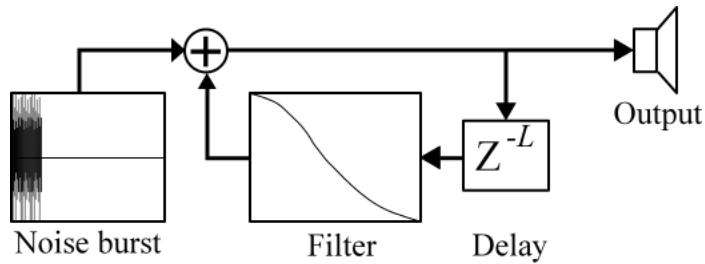


Figure 73: Modelo conceptual.

La línea de retardo simula la longitud de la cuerda, y su largo determina la frecuencia fundamental de vibración, según $L = \frac{f_s}{p}$, con L el largo de la cuerda, es decir de la línea de retardo, f_s la frecuencia de sampleo ($44100\ Hz$) y p el tono de la nota, redondeado al entero más cercano.

La excitación de la cuerda puede contener energía a cualquier frecuencia, por lo que esto se simula con ruido blanco: valores aleatorios entre $[-1, 1]$. Luego de la excitación inicial, la cuerda comienza a vibrar, y ocurre el desplazamiento de la onda en el tiempo. El algoritmo simula lo anterior con una realimentación, que sólo amplifica la frecuencia fundamental y sus armónicos. El filtro es un pasabajos, que, al estar en el lazo de realimentación, tiene el efecto de atenuar gradualmente los armónicos de altas frecuencias, dejando pasar aquellos de bajas frecuencias, lo que se condice con el sonido de una cuerda de guitarra.

En síntesis, una excitación de ruido es pasado por la unidad de delay; entonces, ajustando la longitud del delay y su ganancia, se modifica el tono y su resonancia, respectivamente. Entonces, al aumentar la ganancia, se controla por cuánto tiempo resuena el sistema. Si esta ganancia estuviera al 100, tendríamos un oscilador, por ésto, la misma no debe superar la unidad.

La siguiente figura muestra una implementación del modelo básico de Karplus-Strong.

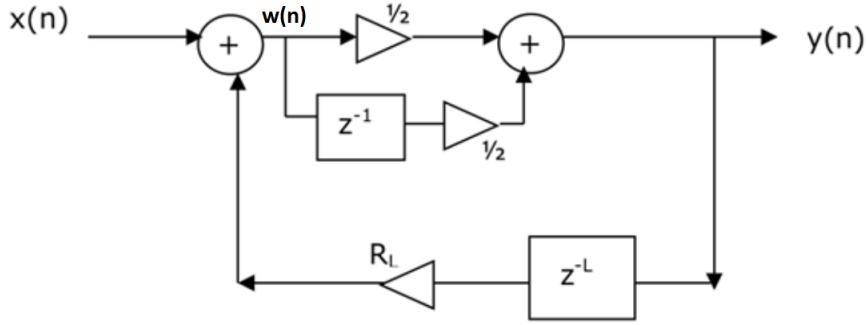


Figure 74: Implementación del modelo básico de Karplus-Strong.

Del anterior diagrama podemos deducir la ecuación en diferencia a partir de los siguientes nodos:

$$\begin{cases} y(n) = \frac{1}{2}w(n) + \frac{1}{2}w(n-1) \\ w(n) = x(n) + R_L y(n-L) \end{cases} \quad (1)$$

Despejando de 1, se llega a:

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

Donde $x(n)$ será ruido, lo que equivale a excitar la cuerda. Podemos entonces, la ecuación en diferencia quedará simplificada a, lo que simplificará su implementación:

$$y(n) = x(n) + \frac{R_L}{2}(y(n-L) + y(n-L-1)) \quad (3)$$

Por otro lado, si aplicamos la transformada \mathcal{Z} , resulta en la siguiente expresión:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{R_L}{2}z^{-L}(1 + z^{-1})} \quad (4)$$

Y luego, reemplazando $z = e^{j\omega/f_s}$, se obtiene la respuesta en frecuencia:

$$H(e^{j\omega/f_s}) = \frac{1}{1 - \frac{R_L}{2}e^{-j\omega L/f_s}(1 + e^{-j\omega/f_s})} \quad (5)$$

El tono resultante de este algoritmo posee un período de $N + 0.5$ muestras, y está dado por $p = \frac{f_s}{N+0.5}$.

4.1.2 Diagrama de polos y ceros

Partiendo de 4, podemos plantear la estabilidad para obtener el rango de valores de R_L para el que el sistema es estable. Intuitivamente se puede ver que si $|R_L| < 1$ el sistema es estable. Ésto se comprobó gráficamente mediante software. A continuación, se muestra el diagrama de polos y ceros del sistema. En este caso, $R_L = 0.996$, y el sistema

es estable, como se puede observar, pues los polos se encuentran dentro de la circunferencia unitaria. Algo que no se observa en el gráfico es la cantidad de ceros en el origen que tiene; el sistema cuenta con $N + 1$ ceros en el origen.

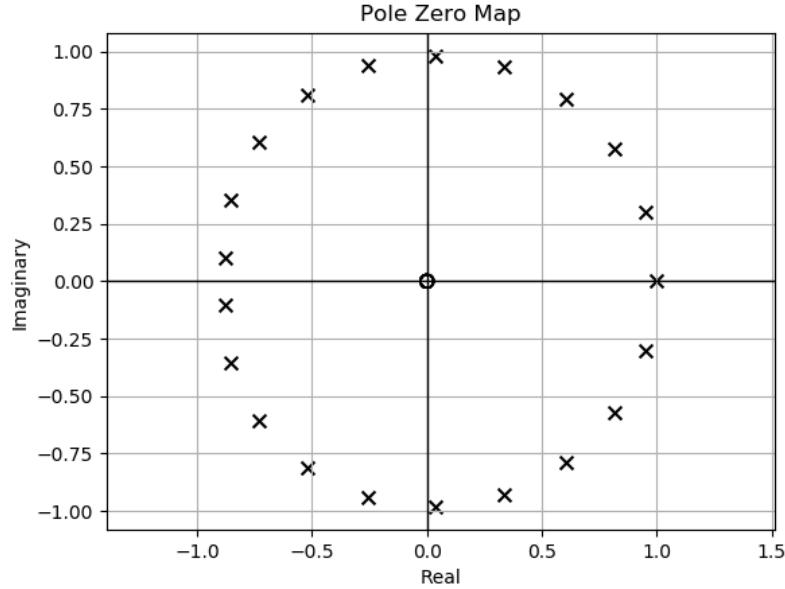


Figure 75: Diagrama de polos y ceros para el modelo básico, con $R_L = 0.996$.

4.1.3 Ruido

El paper en el que se basa este análisis, mencionado en la sección '*Referencias*', menciona que el ruido utilizado como excitación de la cuerda debe tener la siguiente forma:

$$X_t = \begin{cases} A & \text{probabilidad} = \frac{1}{2} \\ -A & \text{probabilidad} = \frac{1}{2} \end{cases} \quad -p \leq t \leq 0$$

Sin embargo, se implementó tanto con ruido uniforme como con normal. Se decidió utilizar ruido uniforme pues es el que más se asemeja al mencionado en el paper, ya que cada X_t tendrá igual probabilidad de ocurrencia, en contraposición al normal, con su respectiva campana gaussiana.

A continuación, dos muestras, una con ruido uniforme y la siguiente con normal. Si bien, en cuanto a sonido sus diferencias son imperceptibles, se puede ver que la muestra con ruido blanco tiene más ármonicos, aunque de menor amplitud. Por otro lado, el ruido normal tiene amplitud impredecible, al tener una desviación, pero van a estar mayormente centrados en la media, al contrario con el uniforme.

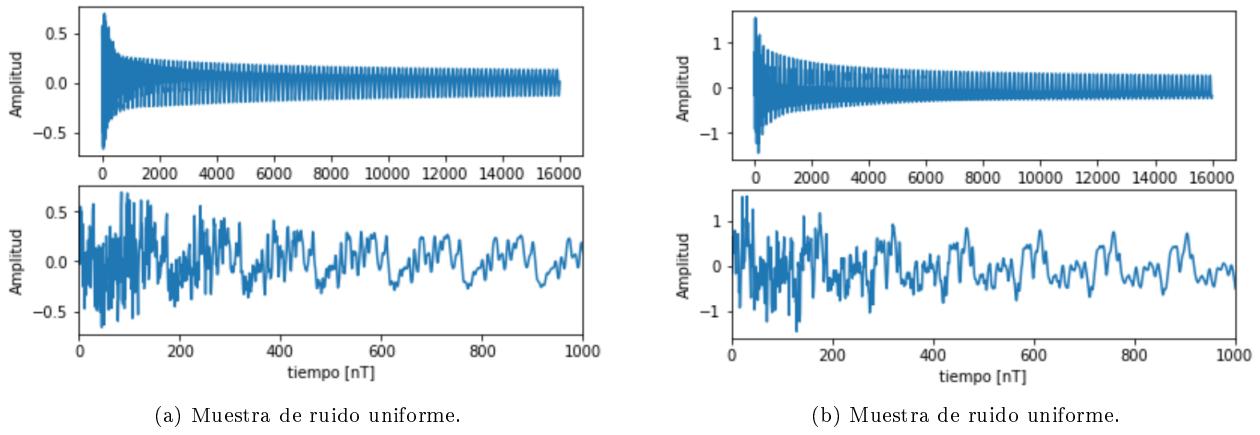


Figure 76: Muestras de Ruido.

4.1.4 Caja de Resonancia

El cuerpo de la guitarra actúa como un resonador. Su objetivo es matchear impedancias. Las cuerdas son altas impedancias y el aire es un medio de baja impedancia, por lo que se necesita un 'transformador' –la caja– para obtener la máxima transferencia de energía; o un amplificador, en el caso de una guitarra eléctrica.

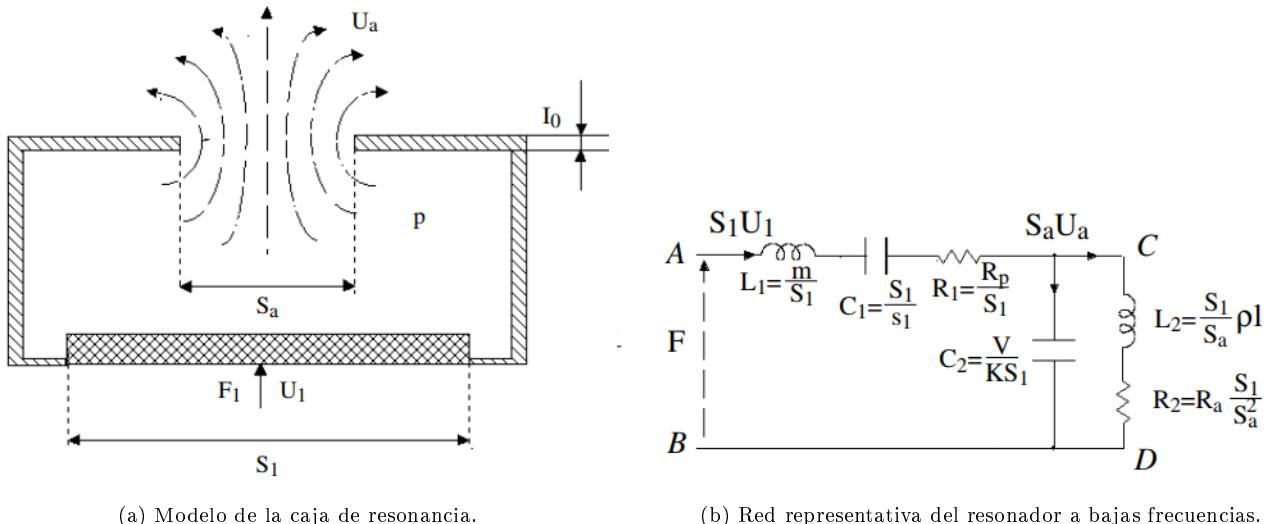


Figure 77: Modelo caja de resonancia.

Donde S_1 en la primera figura representa la superficie activa del resonador, a lo que se le aplica una fuerza externa $F(t)$, que excita el resonador. Esto puede representarse también, con el modelo de la derecha, ya que la impedancia mecánica del resonador $Z(t)$ puede calcularse de la siguiente manera:

$$Z(t) = \frac{F(t)}{U_1(t)} \quad (6)$$

La onda reflejada del puente de la guitarra se relaciona con esta función. Introduciendo los polos y ceros complejos se obtiene:

$$Z(s) = \frac{F(s)}{U_1(s)} = m \frac{(s^t + s\alpha + \omega_1^2)(s^2 + s\alpha_2 + \omega_2^2)}{s(s^2 + s\alpha_h + \omega_h^2)} \quad (7)$$

donde ω_1 y ω_2 son las frecuencias naturales del resonador, ω_h la frecuencia resonante del resonador de Helmholtz, y $\alpha_1, \alpha_2, \alpha_h$ son los parámetros de amortiguación, relacionados con el factor de calidad de la frecuencia característica de la siguiente manera:

$$\alpha_j = \frac{2\pi f_j}{Q_j} \quad Q = \frac{f_j}{\Delta f_j} \quad (8)$$

Conociendo entonces estos parámetros podría implementarse un filtro que simule la caja de resonancia de la guitarra; en síntesis, lo que se busca es un matchear impedancias de la cuerda y el ambiente (aire), para transmitir la onda, en vez de que se refleje.

4.1.5 Afinación

Como se comentó en secciones anteriores, el tono de la nota está dado por:

$$f = \frac{f_s}{N + 0.5} \quad (9)$$

El hecho de que N sea un número entero limita el rango de notas posibles a implementar, hace que los tonos posibles estén cuantizados; principalmente para altas frecuencias, ya que para grandes valores de N (tonos bajos), la diferencia entre el tono a N y $N + 1$ es pequeña. Por otro lado, para tonos altos, N y $N + 1$, se relacionan con tonos completamente diferentes, lo que dificulta la afinación o tuning. En un diagrama, se puede visualizar así:

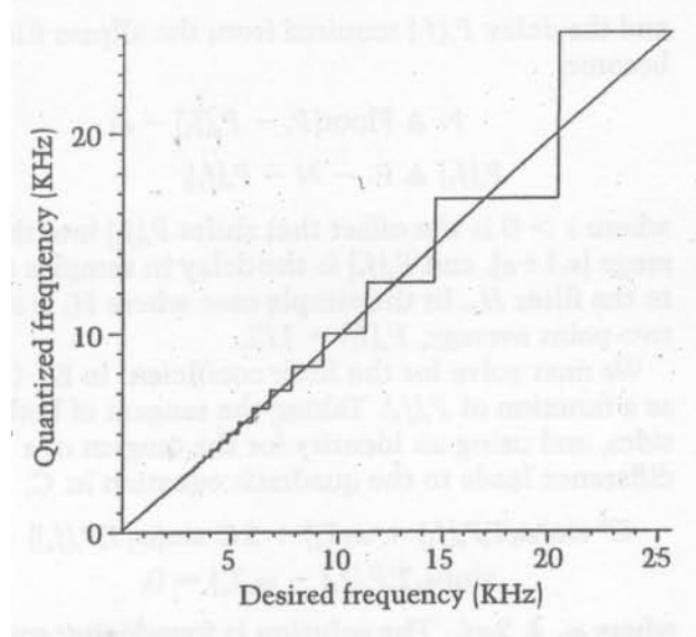


Figure 78: Tono deseado vs. tono resultante para $f_s = 50kHz$.

Una posible solución visualizada por Jaffe & Smith se refiere un filtro en el lazo de realimentación que introduzca un delay sin alterar la ganancia del lazo. El filtro es un pasatodo de la siguiente forma:

$$H_c(z) = \frac{C + z^{-1}}{1 + Cz^{-1}} \quad (10)$$

En primer lugar se pide que en valor absoluto C sea menor a la unidad. Se define $P_c(z)$ como la fase del ya mencionado filtro, y $P_1 = \frac{f_s}{f_1}$, con f_1 la frecuencia fundamental. En resumen, para un error deseado ε , $P_c(f_1)$ se aproxima a:

$$P_c(f_1) = N - \text{floor}(N - \varepsilon) \quad (11)$$

Y finalmente:

$$C = \frac{\sin\left(\frac{\omega_1 T_s - \omega_1 T_s P_c(f_1)}{2}\right)}{\sin\left(\frac{\omega_1 T_s + \omega_1 T_s P_c(f_1)}{2}\right)} \quad (12)$$

4.2 Modelo Modificado

4.2.1 Función transferencia

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

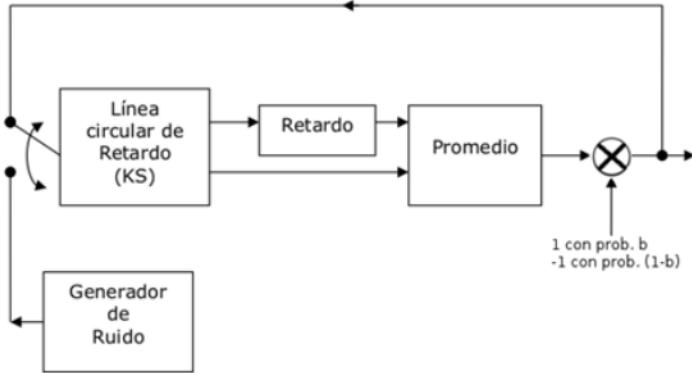


Figure 79: Diagrama en bloques de Karplus-Strong con la variante para la síntesis básica de instrumentos de percusión.

Partiendo del modelo básico, se puede generalizar de la siguiente manera:

$$y(n) = \begin{cases} \frac{1}{2} (y(n-L) + y(n-L-1)) & \text{probabilidad} = b \\ -\frac{1}{2} (y(n-L) + y(n-L-1)) & \text{probabilidad} = 1-b \end{cases} \quad (13)$$

Resultando en:

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

donde si $b = 1$, es el caso de la sección anterior, para la guitarra.

Además se puede obtener la respuesta en frecuencia:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{b}{2} z^{-L} (1 + z^{-1})} \quad (15)$$

De lo que resulta,

$$H(e^{j\omega/f_s}) = \frac{1}{1 - \frac{b}{2} [e^{-j\omega L/f_s} + e^{-j\omega(L+1)/f_s}]} \quad (16)$$

En cuanto a la fase, utilizando la ecuación de Euler para números complejos expresados en forma exponencial ($e^{jx} = \cos x + j \sin x$), se tiene:

$$H(e^{j\omega/f_s}) = \frac{1}{1 - \frac{b}{2} [\cos(\omega L/f_s) \cos(\omega(L+1)/f_s) - j \sin(\omega L/f_s) \sin(\omega(L+1)/f_s)]} \quad (17)$$

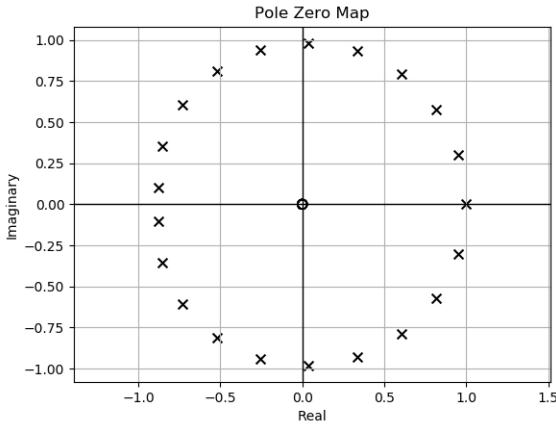
De dicha ecuación se obtiene entonces:

$$\angle H(e^{j\omega/f_s}) = \tan \left(\frac{\frac{b}{2} \sin(\omega L/f_s) \sin(\omega(L+1)/f_s)}{1 - \frac{b}{2} \cos(\omega L/f_s) \cos(\omega(L+1)/f_s)} \right) \quad (18)$$

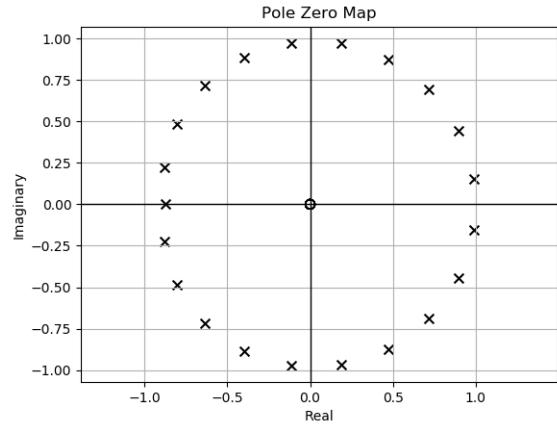
Es válido destacar que esta función no es lineal, ya que depende trigonométricamente de la frecuencia.

4.2.2 Diagrama de polos y ceros

De manera análoga para el caso de $b = 1$, para la guitarra, se graficó el diagrama de polos y ceros tanto para 14 con b valiendo -1 y 1.



(a) Diagrama de polos y ceros para el caso $b=1$.



(b) Diagrama de polos y ceros para $b=-1$.

Figure 80: Algunos diagramas de polos y ceros posibles.

Como se puede ver, el diagrama de polos y ceros depende de la probabilidad, ya que eso determina la constante b , según la ecuación 14.

4.3 Modelo Extendido

Esta sección está basada en el paper de Sullivan[2]. Si se quiere implementar una guitarra con distorsión, el sistema debe tener retardo de grupo constante, es decir su fase debe ser lineal. La necesidad de basarse en este paper y no modificar el modelo básico de Karplus-Strong es que si recordamos, la fase de este último estaba dada por:

$$\angle H(e^{j\omega/f_s}) = \tan \left(\frac{\frac{b}{2} \sin(\omega L/f_s) \sin(\omega(L+1)/f_s)}{1 - \frac{b}{2} \cos(\omega L/f_s) \cos(\omega(L+1)/f_s)} \right)$$

Esta última expresión no es lineal, lo que resultará en una distorsión indeseable.

El modelo propuesto por Sullivan[2] se basa en el siguiente diagrama de bloques:

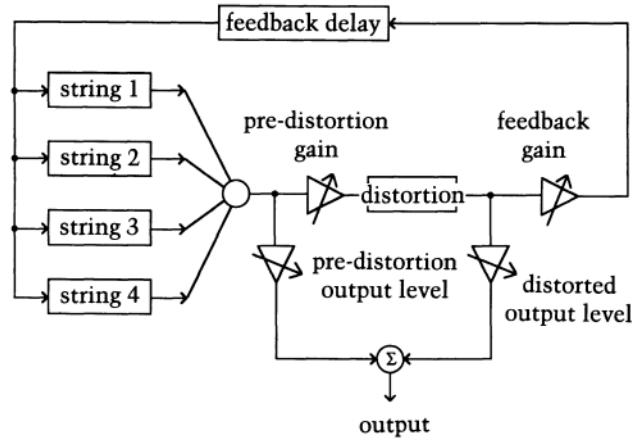


Figure 81: Diagrama en bloques de Sullivan para guitarra eléctrica con distorsión y feedback.

4.3.1 Generación de la cuerda

El bloque *string* está compuesto por:

- Filtro pasabajos
- Filtro para bloquear DC

En primer lugar, el filtro pasabajos es necesario para controlar la respuesta de frecuencias altas y bajas, la rapidez a la que decaen la fundamental y sus armónicos. Lo que permite un mayor control del timbre del instrumento. Pensando en que se quiere distorsión, se busca un filtro cuya fase sea lineal. Uno filtro FIR descrito por la siguiente ecuación cumple esta característica:

$$y(n) = a_0x(n) + a_1x(n-1) + a_0x(n-2) \quad (19)$$

donde $x(n)$ es la entrada, $y(n)$ la salida, y los a_i factores multiplicativos. Una manera sencilla de calcularlos es elegir tal que $a_1 \geq 2a_0 \geq 0$.

Un problema de este filtro es que tiene ganancia en continua, por lo que puede ocasionar que la señal nunca se atenúe. Para esto, se necesita un filtro que bloquee la misma.

Esto nos lleva a un filtro pasa altos descrito por la siguiente ecuación:

$$y(n) = \hat{a}_0x(n) + \hat{a}_1x(n-1) + b_1y(n-1) \quad (20)$$

Eligiendo:

$$\begin{cases} \hat{a}_0 = \frac{1}{1+w_{co}/2} \\ \hat{a}_1 = -\hat{a}_0 \\ b_1 = \hat{a}_0(1 - w_{co}/2) \end{cases}$$

donde w_{co} es la frecuencia de corte muy por debajo de la frecuencia fundamental de la nota a tocar (se puede elegir unas diez veces menor).

Pasando varias veces la entrada por el pasabajos ya mencionado se puede calibrar cuan suave se quiere la nota, es decir, si por ejemplo, se tocó con la púa, lo que resultaría en un sonido más 'metálico' o con el dedo (varias pasadas por el filtro).

La salida del sistema se pasó nuevamente por el pasabajos en cuestión ya que resultaba en sonidos más agradables.

4.3.2 Distorsión

Resta ahora la distorsión. Una función que da muy buenos resultados es la que vendría a ser un soft-clipping de la entrada, dada por:

$$f(x) = \begin{cases} 2/3 & x \geq 1 \\ x - x^3/3 & -1 < x < 1 \\ -2/3 & x \leq -1 \end{cases}$$

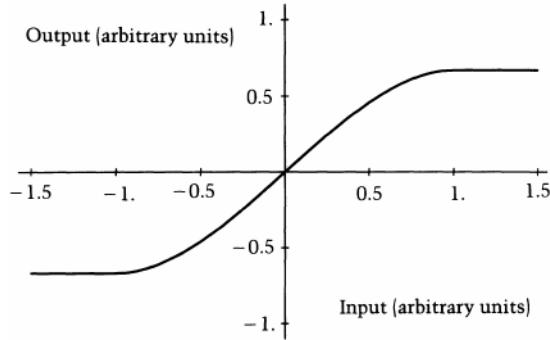


Figure 82: Función elegida para la distorsión.

4.3.3 Feedback

El feedback suele utilizarse para que la nota se sostenga indefinidamente una vez que la distorsión alcanza un nivel estable. Puede utilizarse un filtro en el feedback loop, como se menciona en [3], que estará dado por:

$$y(n) = x(n) - Qx(n-1) + Qy(n-1) + (1-Q)y(n-N) \quad (21)$$

4.4 Conclusiones

Se vieron varios métodos para sintetizar tanto una guitarra acústica, como una eléctrica, pasando por instrumentos de percusión.

Los resultados con Karplus-Strong tanto para cuerdas (guitarra acústica) y percusión resultaron muy acertados a la realidad. Logró implementarse un factor de estiramiento para simular el efecto de atenuación de la nota en el tiempo.

Por otro lado, también, se consiguió un sonido bastante fiel en cuanto a una guitarra distorsionada, sin embargo no se logró implementar en código, correctamente la realimentación, el feedback per se.

En principio, quedan muchas posibilidades de desarrollo a futuro, variedades de filtros por implementar, algunos, son: un filtro pasabajos relacionado a la dinámica –intensidad de cada nota–, un filtro que simule la resistencia de cada cuerda, y otro que simule la caja de resonancia de la guitarra.

References

- [1] Karplus, K., and A. Strong. 1983 “Digital Synthesis of Plucked-String and Drum Timbres”. Computer Music Journal 7 (2): 43-55. Cambridge, Massachusetts: The MIT Press.
- [2] Sullivan R. C. 1990. “Extending the Karplus-Strong Algorithm to Synthesize Electric Guitar Timbres with Distortion and Feedback”. Computer Music Journal, Vol. 14, No. 3 (Autumn, 1990), pp. 26-37. Princeton, New Jersey: The MIT Press.
- [3] Jaffe, D. A., and J. O. Smith. 1983. “Extensions of the Karplus-Strong Plucked-String Algorithm”. Computer Music Journal 7 (2): 76-87. Cambridge, Massachusetts: The MIT Press.

5 Síntesis basada en muestras

5.1 Introducción

La síntesis basada en muestras consiste en la realización de un audio a partir de un numero reducido de archivos de audios que ya se tenían almacenadas. Los sonidos que puedan llegar a requerirse que no se tengan almacenados (ya sea porque no coincide la duración, pitch, volumen, etc) se generan a partir de la modificación de las muestras que se tienen almacenadas. La ventaja de este tipo de síntesis es que permite sintetizar canciones que suenan muy similares a lo que serían si se hubieran realizado con instrumentos físicos.

A lo largo de esta sección se presenta cuales son las principales dificultades de la síntesis basada en muestras y los algoritmos que existen para lidiar con estas dificultades. Asimismo, se compara las diferencias, ventajas y desventajas de cada uno de estos algoritmos, y a partir de esto se discute cual fue el razonamiento que llevo al sintetizador implementado.

5.2 Time Scaling Algorithms

El mayor problema a enfrentar en este tipo de síntesis es como generar los sonidos que se necesitan para los cuales no se tiene una muestra de audio. Por ejemplo, si se tiene un audio almacenado de como suena la tecla correspondiente al *do4* en un piano con una duración de un segundo. Es posible obtener la nota correspondiente a un semitono mas alto *do4#* de la misma duración que la muestra almacenada?

Para ajustar la 'altura' (pitch), que tan grave o aguda se escucha la nota, se puede remuestrear el audio de una nota que se tenga a una frecuencia de muestreo distinta y re-interpretar la señal obtenida con la misma frecuencia de muestreo original. Esto se puede describir matemática como el siguiente sistema:

$$y(nT) = x(n.m.T) \quad (22)$$

Se pueden distinguir dos casos posibles, cuando $m < 0$ se interpolan muestras para obtener todas las muestras del vector $y(nT)$ lo que resulta en un audio de mayor duración en el tiempo debido al incremento de muestras ademas de un pitch mas 'bajo' en un factor m que el que se tenía previamente. El otro caso es para $m > 1$, en este caso se toman menos muestras de las qque se tenian originalmente en el input por o que disminuye la duracion del audio, a su vez sube el pitch del audio en un factor m .

El problema con remuestrear el audio es que ademas de cambiar el contenido armónico de la señal también cambia su extensión en el tiempo. Para lograr de independizar el escalamiento en frecuencia del escalamiento en tiempo de una señal existen los denominados algoritmos de TSM (TimeScaleModification). Si se puede lograr cambiar la extensión temporal de una señal de audio sin distorsionar su contenido armónico (conservar su pitch) entonces simplemente se puede remuestrear una muestra almacenada para obtener el pitch deseado y luego aplicar un algoritmo TSM para corregir la extensión temporal a la cual se desea.

Hay un par de nociones básicas que se suelen utilizar en todos los algoritmos TSM. En general estos algoritmos consisten en dividir la señal de entrada en bloques centrados en una determinada posición (comúnmente se los denomina

'bloques de análisis), aplicar algún tipo de operación matemática sobre estos bloques y copiarlos a otro bloque en la salida(denominados 'bloques de síntesis). Se define la función de mappeo(conocida generalmente como 'time stretch function') a:

$$\tau(nT) = f(nT) \quad (23)$$

$n \in [0, N - 1]$, donde N es el numero de muestras de la señal de entrada

En principio la función de mappeo τ no tiene porque ser una función lineal,solo se asume que la función es monótona creciente y que $\tau(0) = 0$.

La función de mappeo definida en (23) sirve como parámetro para todos los algoritmos de TSM que se desarrollaran a continuación.El rol de esta función es la de asignar que posición le corresponde a la salida a un valor de la entrada tomado en el tiempo nT .

Existen una gran variedad de algoritmos TSM,generalmente se distinguen entre los que trabajan en el dominio de la frecuencia y los que trabajan en el dominio del tiempo. A continuación se presentaran y discutirán los tres algoritmos implementados en nuestro sintetizador. Los primeros dos (OLA y WSOLA) corresponden a algoritmos que trabajan en el dominio del tiempo, el ultimo y tercer algoritmo(Phase Vocoder) trabaja en el dominio de la frecuencia.

5.2.1 Overlap and Add(OLA)

OLA corresponde al algoritmo TSM mas simple y posiblemente el mas intuitivo.Existen una gran variedad de algoritmos TSM que se basan en una versión modificada del método OLA¹.

En definitiva consiste en dividir la señal que se desea modificar en segmentos(denominados grains o slots) de una longitud determinada mediante la aplicación de ventanas,y luego copiar estos segmentos a una ubicación determinada en el output dada por una función de mappeo(la misma que se definió en (23)). Gráficamente el esquema seria el siguiente:

¹Algunos ejemplos notables son SOLA, PSOLA y WSOLA(el cual se analiza en este informe)

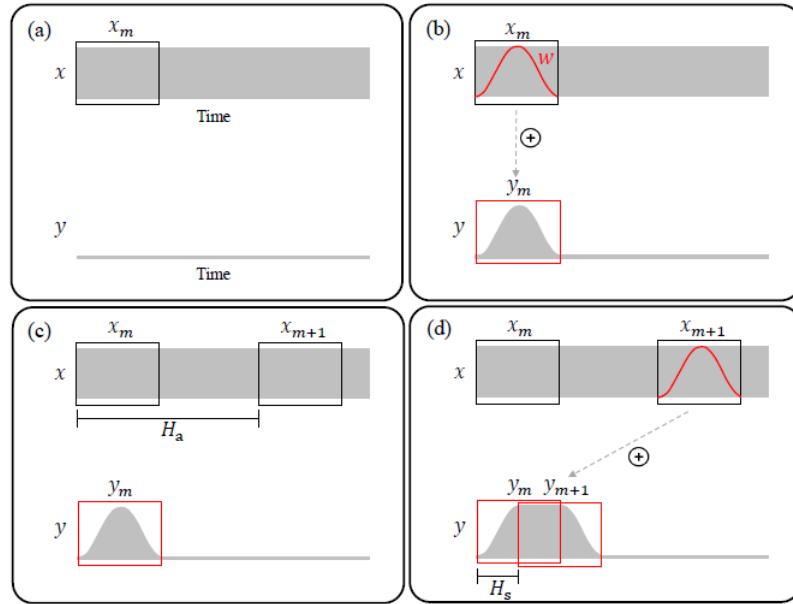


Figura 83: Representación grafica del metodo OLA.a)Muestra el segmento m-esimo de la entrada que se copiara a la salida.b)Se aplica la ventana a dicho segmento y se lo copia a la salida.c)Toma el proximo segemento a copiar que se encuentra a una distancia H_a del anterior.d)Le aplica la ventana al segmento m+1-esimo y lo copia a su ubicacion correspondiente a la salida, que esta a una distancia H_s del bloque de sintesis anterior.

5.2.2 Parámetros utilizados

El algoritmo necesita recibir tres parámetros (ademas del vector con la señal de entrada),los cuales son:

- La ventana que se desea aplicarle a cada uno de los segmentos del input
- La función de mapeo definida en (23)
- El factor de overlap.

Los primeros dos parámetros ya se presentaron brevemente. El factor de overlap es un numero que toma valores entre 0 y 1 (no inclusive). Este factor decide la separación H_s entre dos bloques de la entrada consecutivos como se muestra en la figura (83). La separación H_s entre bloques de síntesis esta dada por la relación:

$$H_s = (1 - OF).largo(Ventana)$$

Donde OF es el overlap factor. Como se puede ver de la relación anterior, cuando OF toma el valor extremo $OF = 0$ entonces los bloques de síntesis están separados por la misma distancia que el largo de la ventana, lo que quiere decir que no hay superposición entre bloques de síntesis. En el caso extremo en el que $OF = 1$ entonces $H_s = 0$ por lo que no habría separación entre bloques de síntesis sino que habría una superposición del 100 % y estarían todos centrados en la misma posición a la salida.

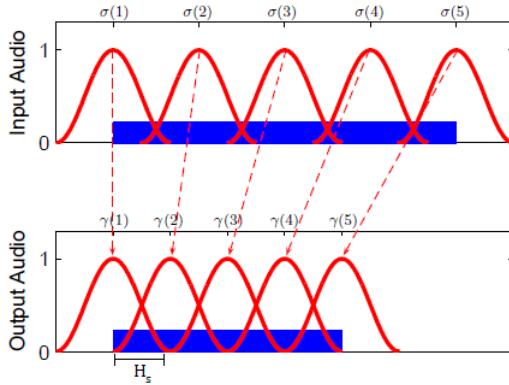


Figura 84: Posición de los bloques de análisis y síntesis dados por σ y γ

5.2.3 Algoritmo

Una vez definidos los parámetros el método consiste en formar un vector con las posiciones en las cuales están centrados cada bloque de síntesis a la salida. Esto es trivial ya que se conoce la separación entre bloques de síntesis y que $\tau(0) = 0$. Se denotara dicho vector con la letra γ . Entonces se tiene que:

- $\gamma(1) = 0$, El primer bloque de síntesis esta centrado en 0, dado que $\tau(0) = 0$
- $\gamma(k) = \gamma(k - 1) + H_s$
- $largo(\gamma) = \left\lceil \frac{largo(output)}{H_s} \right\rceil$

Una vez que se tienen los valores de este vector se puede utilizar conjuntamente con la función de mapeo τ para obtener las posiciones en las que están centrados cada bloque de análisis necesario para obtener los bloques de síntesis dados por γ . Se denotara el vector con las posiciones en las que están centrados los bloques de análisis con la letra σ . Para este vector se tiene:

- $\sigma(1) = 0$, Este caso es igual que para γ debido a que $\tau(0) = 0$
- $\sigma(k) = \tau^{-1}(\gamma(k))$, donde τ^{-1} es la función inversa de la time stretch function.
- $largo(\sigma) = largo(\gamma)$

El rol de estos vectores queda mucho mas claro con una representación gráfica, por lo que aquí se presenta un esquema de como es la relación entre estos dos vectores.

Los rectángulos azules en el input y en el ouput simbolizan las muestras de audio. En el caso anterior se muestra que el rectángulo azul en el output es de menor extensión que el del input, lo que significa que se esta usando el algoritmo OLA para comprimir en el tiempo el input. Sin embargo, la figura (84) se presenta para mostrar la función de los vectores σ y γ , la cual es la misma tanto para comprimir como para extender en el tiempo una señal.

Finalmente el audio a la salida esta dado por la siguiente relación matemática:

$$output = \frac{\sum_{k=1}^{largo(\sigma)} \omega(n - \gamma(k)) input(n - \gamma(k) + \sigma(k))}{A(n)} \quad (24)$$

ω hace referencia a la función ventana que se recibe como parámetro. La función $A(n)$ que se encuentra dividiendo abajo se utiliza para normalizar la salida ya que la aplicación de ventanas puede causar distorsiones de amplitud no deseadas en la señal de salida. Para obtener la función $A(n)$ que normaliza el valor de $y(n)$ se parte de la base de que en un caso ideal se desea que la suma de las ventanas a la salida sea la función unitaria:

$$\sum_{k=1}^{largo(\gamma)} \omega(n - \gamma(k)) = 1$$

En este caso $A(n) = 1$, por lo que dividir por $A(n)$ no introduce ningún cambio a la salida. Sin embargo cuando la suma de la superposición de ventanas a la salida es distinto de la función unitaria es necesario normalizar la salida para compensar distorsiones de amplitud, y la función de normalización esta dada por:

$$\sum_{k=1}^{largo(\gamma)} \omega(n - \gamma(k)) = A(n)$$

5.2.4 Ventajas y desventajas

La principal ventaja de OLA es que es el algoritmo más simple de todos los implementados tanto conceptualmente como computacionalmente, por lo que es posible realizar síntesis de notas musicales de manera muy rápida si se utiliza OLA. Sin embargo, el audio obtenido es de una calidad muy baja. Para síntesis de instrumentos con contenido armónico importante (por ejemplo el violín) la distorsión introducida tapa casi por completo el sonido original del instrumento, ademas para extensiones o compresiones de tiempo grandes (por lo menos duplicar la longitud o acortarla a la mitad) la distorsión para cualquier instrumento es apreciable.

Esto se debe a que OLA simplemente copia un bloque de análisis de la entrada y lo pega a la salida, esta mecanismo simple no tiene en cuenta factores importantes para la preservación del contenido armónico, como la continuidad de la fase.

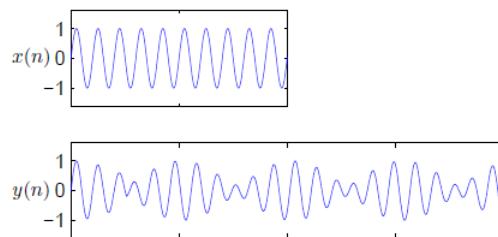


Figura 85: Distorsión armónica introducida por OLA (extendiendo la duración de la entrada el doble)

A pesar de la pésima calidad de OLA para preservar el contenido armónico de un audio, el algoritmo puede ser sumamente útil para modificar las duraciones de muestras tomadas de instrumentos percusivos(también referidos como 'no armonicos'). Esto se debe a que en los instrumentos percusivos toma mucha mayor importancia el ritmo y las características temporales del sonido que su contenido armónico (la introducción de armónicos a causa de discontinuidades no es tan perceptible como para los instrumentos con armónicos bien definidos).

5.2.5 Waveform Synchronous Overlap and Add(WSOLA)

Este algoritmo pertenece a la familia de algoritmos que son una derivación de OLA. En principio el algoritmo sigue exactamente el mismo procedimiento que OLA. La única diferencia entre los dos es que para WSOLA se agrega un parámetro mas definido como 'tolerancia'.

Se menciono en la subsección anterior, como el problema principal de OLA es que introduce discontinuidades en la fase a la salida que generan nuevos armónicos y distorsión notablemente el sonido resultante. Como solución a este problema, el parámetro adicional de WSOLA permite tomar cada bloque de análisis en un rango posible de ubicaciones a la entrada dado por:

$$\sigma'(k) = \sigma(k) \pm \Delta_k \quad (25)$$

σ tiene el mismo significado aquí que el que se le dio en la subsección anterior. El Δ_k es un numero que pertenece al rango de valores comprendido entre $[-\Delta_{max}, \Delta_{max}]$ donde Δ_{max} es el nuevo parámetro llamado 'tolerancia' que se menciono previamente.

Lo que se esta realizando es permitir centrar el bloque de análisis en un rango de posiciones cercanas a la dada por $\sigma(k)$ con el fin de preservar lo mejor posible las características armónicas de la señal original. El único problema restante es como conseguir el valor de Δ_k óptimo que minimice las discontinuidades a la salida. Para ello se define como progresión natural de un bloque de análisis a aquel bloque que colocado a una distancia H_s del bloque de síntesis anterior no presente ninguna discontinuidad de fase. Esto se cumple siempre para el bloque de análisis cuyo centro esta dado por:

$$\sigma_{natural}(k) = \sigma'(k - 1) + H_s$$

Por lo que Δ_k debe ser tal que el bloque de análisis k-esimo tomado sea lo mas similar posible a la progresión natural del bloque de análisis anterior.

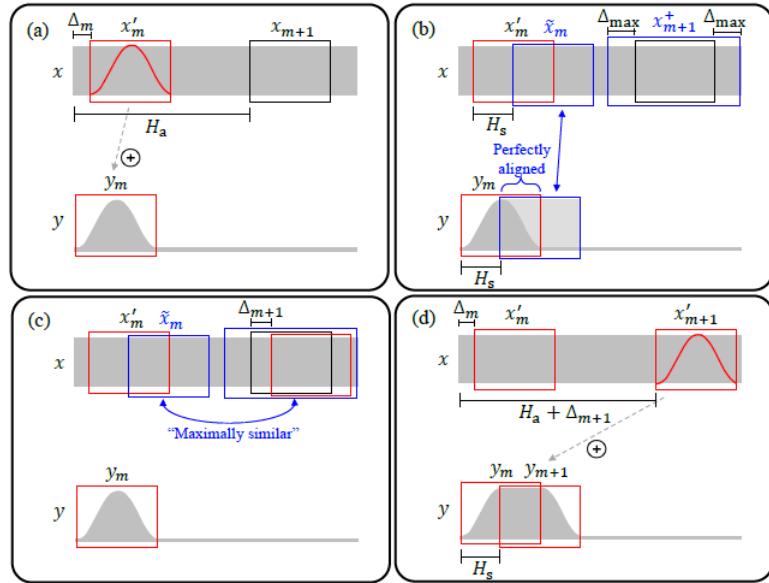


Figura 86: Gráfica del metodo WSOLA.a)Se muestra el bloque de analisis m-esimo ya elegido y desplazado un Δ_m de su centro original dado por σ .b)El bloque indicado por \tilde{x}_m es la 'progresion natural' de x'_m ,debe elegirse un x'_{m+1} que se encuentre dentro del rectángulo azul de la derecha.c)Se encunetra el bloque de analisis mas similar a \tilde{x}_m desplazandose un Δ_{m+1} a la derecha de la posicion dada por $\sigma(m + 1)$.d)Se toma dicho bloque de analisis que es maximamente similar a la progresion natural y se lo copia a una distancia de H_s respecto del bloque de sintesis anterior.

5.2.6 Calculo del Δ_k optimo

Previamente se menciono que el Δ_k optimo es aquel que maximiza la similitud entre el bloque de análisis actual y la progresión natural del ultimo bloque de análisis elegido. Para encontrar dicho bloque de análisis, realizamos la correlación entre el bloque correspondiente a la progresión natural y el rectángulo azul de la derecha presentado en la figura(86(b)).Donde la correlación entre dos bloques esta definida como:

$$\text{correlacion}\{x, y\}(n) = \sum_{k=0}^{\text{largo(ventana)}} x(k+n)y(k) \quad (26)$$

La correlación es máxima donde la suma de los productos es máxima,lo cual ocurre cuando los dos bloques son similares, entonces el Δ_k optimo se obtiene como:

$$\Delta_{k_{optimo}} = n / \text{correlacion}\{x, y\}(n) = \max(\text{correlacion}\{x, y\}(n))$$

5.2.7 Ventajas y desventajas

WSOLA soluciona considerablemente bien el problema de la distorsión armónica que se encuentra en OLA. El algoritmo sigue siendo relativamente rápido, aunque menos que OLA ya que se debe realizar una correlación entre dos bloques por cada bloque de síntesis en la salida. Sin embargo, el problema principal de WSOLA es que selecciona los bloques de análisis con tal de minimizar las discontinuidades de fase de la frecuencia fundamental del audio. Para instrumentos compuestos por varias frecuencias o efectos sutiles de modulación de amplitud (como el violín o la flauta) WSOLA solo preserva la continuidad de fase para una frecuencia por lo que se distorsiona el timbre del instrumento.

5.3 Phase Vocoder

El último algoritmo implementado fue el phase vocoder. Aunque el funcionamiento del phase vocoder es similar a OLA y WSOLA en que descompone a la señal de entrada en bloques de análisis por medio de la aplicación de ventanas y luego coloca la suma de los bloques de síntesis a la salida, este algoritmo es muy distinto a los anteriores conceptualmente.

En primer lugar, el phase vocoder es uno de los algoritmos de TSM que trabaja en el dominio de la frecuencia. Como se vio previamente, el principal problema de OLA y WSOLA es su desempeño para instrumentos armónicos y con timbres complejos debido a la distorsión armónica que generan. El phase vocoder busca minimizar eso buscando que se mantenga la continuidad de fase a la salida para todos los armónicos que componen a la señal.

El algoritmo consiste en aplicarle la FFT (Fast Fourier Transform) a cada bloque de análisis que se toma del input. El resultado que se obtiene es una STFT (Short Time Fourier Transform) de la entrada, donde cada intervalo de tiempo es un bloque de análisis. Luego, se ajustan las fases de los espectros obtenidos de forma tal que al superponer la suma de los espectros a la salida no haya discontinuidades en la fase. Finalmente se aplica la IFFT a cada uno de los espectros modificados y se obtienen los bloques de síntesis los cuales son multiplicados nuevamente por una ventana antes de ser superpuestos a la salida.

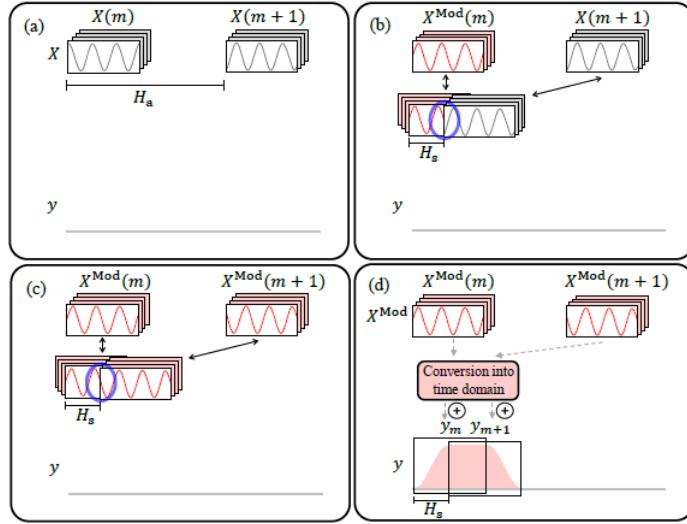


Figura 87: Esquema del funcionamiento del Phase Vocoder.a)Se tienen dos arreglos uno es X_m el cual se obtiene de aplicarle la fft al bloque de análisis x_my el otro es X_{m+1} el cual se obtiene de aplicarle la fft al bloque de análisis que le sigue a x_m .b)Al superponer los espectros en las ubicaciones de los bloques de síntesis se presentan discontinuidades.c)Se modifican los espectros de forma tal que ya no haya discontinuidades al superponerse a la salida.d)Se antitransforman los espectros mediante la IFFT y se obtienen los bloques de síntesis correspondientes a la salida.

5.3.1 Parámetros

Al igual que los dos algoritmos previos recibe como parámetros la función de estiramiento presentada en la ecuación(23), y la ventana que se desea utilizar.Sin embargo, en nuestra implementación no recibe el factor de overlap sino que partimos de la base de que $H_a = H_s = \text{HopSize}$, donde el HopSize es un parámetro que elige el usuario y es la separación en muestras tanto de los bloques de análisis como de los de síntesis. Esto cambia ligeramente la definición de los vectores γ y σ introducidos previamente en que estos dos ya no tendrán la misma longitud y sus valores estarán dados por:

- $\sigma(1) = \gamma(1) = 0$
- $\sigma(k) = \sigma(k - 1) + \text{HopSize} = \gamma(k) = \gamma(k - 1) + \text{HopSize}$
- $\text{largo}(\sigma) = \left\lceil \frac{\text{largo}(input)}{\text{HopSize}} \right\rceil$
- $\text{largo}(\gamma) = \left\lceil \frac{\text{largo}(output)}{\text{HopSize}} \right\rceil$

5.3.2 Algoritmo y propagación de fase

El método consiste en aplicarle la fft a cada bloque de análisis dado por σ y a partir de estos espectro conseguir un numero de espectros igual a la cantidad de bloques de síntesis,modificar los mismos para corregir las discontinuidades

de fase y finalmente sumar y superponer la anti-transformada de los espectros a la salida.

El primer paso es obtener una cantidad de espectros igual a la cantidad de bloques de síntesis a partir de los espectros de los bloques de análisis. Esto se puede hacer simplemente mediante una simple interpolación de los espectros que se tienen según:

$$X_{interp}(k) = f(\tau^{-1}(k)) \quad (27)$$

$$f(t) = \frac{t - \sigma(n)}{\sigma(n+1) - \sigma(n)} X(n+1) + \frac{\sigma(n+1) - t}{\sigma(n+1) - \sigma(n)} X(n), \sigma(n) \leq t \leq \sigma(n+1) \quad (28)$$

Las magnitudes del espectro final son iguales a las interpoladas. Sin embargo, es necesario ajustar la fase del siguiente paso es corregir los espectros para que no haya discontinuidades de fase. El objetivo es asegurar que la fase de cada bin de frecuencia sea continua y lineal a través de los bins de tiempo (en este caso son los bloques de síntesis).

La fase debe ser lineal ya que asumiendo que el espectro en un bin de tiempo es una suma de senoidales se tiene:

$$X_{nH_s}(k) = A_k e^{j(\omega_k n H_s + \phi_k)} = A_k e^{j(\frac{2\pi k}{N} n H_s + \phi_k)}$$

De la expresión anterior se tiene que la diferencia de fase entre dos bins de tiempo consecutivos es:

$$\phi_{n+1} - \phi_n = \Delta\phi = \frac{2\pi}{N} H_s \quad (29)$$

Donde en nuestro caso H_s es *HopSize* y N es igual al largo de la ventana de análisis. Con la ecuación (29) en teoría se podría obtener la fase en todos los bins de tiempo con solo conocer la fase inicial para el bin de frecuencia dado. Sin embargo, en la realidad las señales con las que se trabaja no están formadas por armónicos de periodo fundamental $\frac{1}{N}$, por lo que también es necesario estimar la frecuencia exacta para cada bin de frecuencia.

La frecuencia exacta de un bin puede expresarse como:

$$\omega_k = \frac{2\pi}{N} k + \Delta\omega_a = \frac{2\pi}{N} k + \frac{\overbrace{\phi_{n+1} - \phi_n - \frac{2\pi k H_s}{N}}^{\epsilon[-\pi, \pi]}}{H_s} \quad (30)$$

La frecuencia exacta puede determinarse a partir del cambio de fase que se observa entre dos bins consecutivos de tiempo y el cambio de fase esperado si la frecuencia exacta fuera un múltiplo de $\frac{1}{N}$. Para obtener las fases corregidas se define:

$$g_m = \left\lfloor \frac{\tau^{-1}(\gamma(m))}{HopSize} \right\rfloor$$

g_m es el mayor indice del vector σ para el cual el valor devuelto es menor a $\tau^{-1}(\gamma(m))$ y a su vez $g_m + 1$ devuelve el menor indice de σ tal que el valor devuelto es mayor a $\tau^{-1}(\gamma(m))$.

A partir de g_m se definen los espectros:

$$X_1(m) = X(g_m)$$

$$X_2(m) = X(g_m + 1)$$

De los espectros anteriores se puede definir la fase corregida del espectro de síntesis como:

$$\angle X_s(n) = \angle X_s(n - 1) + (\angle X_2(m) - \angle X_1(m))$$

$$\angle X_s(1) = \angle X(1)$$

Con ese ya se tienen las fases y los módulos del espectro corregido por lo que el espectro corregido esta dado por:

$$X_s(n) = |X| \cdot e^{j\angle X_s(n)} \quad (31)$$

Finalmente se antitransforma cada espectro para obtener los bloques de síntesis a la salida y se suman a su posición correspondiente a la salida.

5.3.3 Ventajas y desventajas

La ventaja principal de este método es que garantiza continuidad de fase para todas las frecuencias de la señal de entrada, esto significa que es el mejor de los 3 algoritmos en cuanto a preservación de contenido armónico y timbre de instrumentos. La desventaja es que es un algoritmo muy lento ya que requiere el calculo de la FFT para cada bloque de análisis, la interpolacion de los espectros, la corrección de las fases, la generación del espectro de síntesis y finalmente la IFFT de dicho espectro. Asimismo el phase vocoder no tiene buen desempeño con señales percusivas ya que garantiza la continuidad de fase a lo largo del tiempo (coherencia horizontal de la fase) mientras que para este tipo de señales la continuidad de frecuencias dentro de un mismo bin de tiempo toma mayor relevancia.

5.4 Desempeño y comparación de cada algoritmo

A continuación se presenta una tabla en la que se muestra el desempeño en tiempo de cada algoritmo para distintos instrumentos:

| Instrumento | OLA(seg) | WSOLA(seg) | Phase Vocoder(seg) |
|-----------------------|----------|------------|--------------------|
| Violín(Makar) | 9 | 10 | 69 |
| Bombo(Piratas) | 19 | 22 | 233 |
| Corn Anglais(Piratas) | 26 | 28 | 218 |
| Trumpet(Mario) | 92 | 96 | 772 |

Figura 88: Tabla con los tiempos de ejecución para cada algoritmo

Cada uno de los audios sintetizados mostrados en la tabla se encuentran dentro de la carpeta titulada AudiosEj6. Se puede notar al escuchar los audios que el phase vocoder provee definitivamente el audio de mejor calidad para todos los instrumentos salvo para los bombos.

5.5 Implementación del sintetizador

A continuación se presenta una tabla con los instrumentos que se implementaron así como el algoritmo utilizado para cada uno de ellos:

| Instrumento | Algoritmo empleado |
|--------------|--------------------|
| Bombo | OLA |
| Guitarra | Phase Vocoder |
| Violín | Phase Vocoder |
| Corno inglés | Phase Vocoder |
| Trompeta | Phase Vocoder |

Cuadro 7: Instrumentos y su algoritmo correspondiente

Se guardaron 3 notas por octava para cada instrumento. Para cada una de esas notas, en lo posible se guardaron tres duraciones posibles(0.25seg, 1seg y 1.25 seg) y para cada una de esas duraciones se guardaron dos intensidades distintas(forte y piano). Con lo que aproximadamente se tienen alrededor de 18 muestras por octava por instrumento. La excepción a esto es el bombo dado que el mismo no tiene notas, por lo que solo se almacenaron 9 muestras de 5 duraciones distintas.

5.6 Conclusiones

Se puede concluir que hay una relación de compromiso notable entre tiempo de ejecución de la síntesis y calidad del audio obtenido. Utilizando los algoritmos OLA y WSOLA se obtienen tiempos de síntesis rápidos, pero la calidad del audio está bastante distorsionada salvo para el caso de la percusión. Por otro lado, al sintetizar con el phase vocoder se obtienen tiempos de síntesis considerablemente largos pero a cambio de un audio de una calidad notablemente superior a OLA y WSOLA, nuevamente sin tener en cuenta la síntesis del bombo.

6 Espectrograma

6.1 Introducción

Un espectrograma es una gráfica tridimensional en la que se busca representar como varia el contenido armónico de una señal a lo largo del tiempo. Normalmente se toma el eje horizontal de la gráfica como el tiempo, y el eje vertical como las frecuencias de la señal. La intensidad de un armónico para un tiempo dado suele representarse mediante el uso de un color, generalmente colores cerca del rojo indican mayor intensidad en dicho armónico y colores cerca del azul baja intensidad.

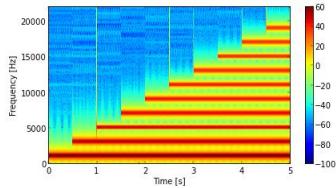


Figura 89: Ejemplo del formato de un espectrograma

Existen diversas aplicaciones en el que el espectrograma de una señal puede ser de utilidad. En este informe se enfoca en el uso del mismo para identificar características en una señal de audio. Las señales de audio son no estacionarias, lo que quiere decir que su contenido armónico no permanece igual a lo largo del tiempo, es por esto que es conveniente tener el espectrograma del audio para tener una idea no solo del contenido armónico de la señal pero sino también de como el mismo evoluciona a lo largo del tiempo.

El objetivo de esta sección es la de explicar los parámetros relevantes que deben tenerse en cuenta para realizar el espectrograma de una señal y como cada uno afecta el resultado obtenido. Con este objetivo se presentara y analizara la herramienta matemática conocida como 'Short Time Fourier Transform'(STFT) ya que para obtener el espectrograma de una señal es necesario realizar la STFT de la misma. Asimismo, se presentara la implementación de un espectrograma para una señal de audio y se comentara sobre cuales son los parámetros apropiados a utilizar y porque.

6.2 Desarrollo matemático

6.2.1 Definición

El espectrograma de una señal puede obtenerse directamente de la STFT de una señal según:

$$\text{Espectrograma}(t, \omega) = |\text{STFT}\{x(t)\}|^2(t, \omega) \quad (32)$$

La definición previa vale tanto para el caso en el que x (la señal de interés) es una señal en tiempo continuo como para si lo fuera en tiempo discreto, solo falta sustituir $t = nT_{step}$ y $\tau = mT_{step}$.

En cuanto a la STFT de una señal, la misma se define como:

Tiempo continuo:

$$STFT\{x(t)\}(t, \omega) = \int_{-\infty}^{+\infty} x(t + \tau)w(\tau)e^{-j\omega\tau}d\tau$$

Tiempo discreto:

$$STFT\{x(nT)\}(n, \omega) = \sum_{m \in \mathbb{Z}} x(nT + mT)w(mT)e^{-j\omega m}$$

Donde w es una función que sirve de ventana, esto quiere decir que tiene valores no nulos en un intervalo de longitud L y es nula fuera de ese intervalo. Teniendo en cuenta los fines prácticos de este informe de aquí en mas se trabajara con la definición en tiempo discreto y asumiendo $T = 1$ sin perdida de generalidad.

Puede interpretarse la STFT como la transformada de producto de dos señales:

$$\begin{aligned} STFT\{x(n)\}(n, \omega) &= \mathcal{F}_d\{x(n+m)w(m)\} = \mathcal{F}_d\{x(n+m)\} * \mathcal{F}_d\{w(m)\} \\ &= \frac{1}{2\pi} \int_{-2\pi}^{2\pi} X(\theta)W(\theta - \omega)e^{j\theta n}d\theta \end{aligned}$$

Esta ultima forma análoga de ver la señal muestra que la STFT es equivalente a la IDTFT del espectro de la señal ponderado por una función W centrada en ω .

En la practica no es posible calcular la STFT para un continuo de frecuencias si no que se debe calcular para N muestras por lo tanto la expresión cambia a:

$$STFT\{x(n)\}(n, k) = \sum_{m \in \mathbb{Z}} x(n+m)w(m)e^{-jk\frac{2\pi}{N}m}, k \in [0, N-1]$$

De forma similar, la STFT se calcula para ciertos intervalos de tiempo separados por una distancia $R \in \mathbb{N}$:

$$STFT\{x(n)\}(r, k) = X(r, k) = \sum_{m=0}^{L-1} x(rR + m)w(m)e^{-jk\frac{2\pi}{N}m}, k \in [0, N-1], r \in \mathbb{Z}$$

De la expresión anterior se puede llegar a la siguiente expresión final:

$$X(r, k) = X_r(k) = \sum_{m=0}^{L-1} x_r(m)e^{-jk\frac{2\pi}{N}m} \quad (33)$$

$$\text{donde : } x_r(m) = x(rR + m)w(m)$$

De 33 se puede ver que la STFT es simplemente una secuencia de DFTs de x con una ventana aplicada cada R muestras, visualmente seria lo siguiente:

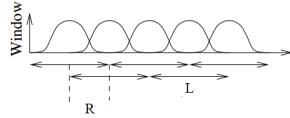


Figura 90: Esquema de como están dispuestas las ventanas sobre x

La STFT correspondería a una secuencia de DTFTs de x donde hay una DFT por cada ventana aplicada sobre x . R es la distancia entre ventanas y L el largo de las mismas. Como puede verse en la imagen anterior, si $R < L$ entonces habrá superposición de ventanas mientras que si $R > L$ no habrá superposición de ventanas. Si se desea recuperar la señal a partir de la STFT de la misma entonces se debe cumplir que:

$$N \geq L \geq R$$

Si N fuera menor a L entonces no sería posible recuperar la señal a partir de la IDFT del espectro, en cuanto a el lado derecho de la desigualdad. Se explico previamente que el caso en el que $R > L$ hay muestras que no caen en ninguna ventana por lo que se pierde información de la señal que no es posible recuperar, el caso en el que $R = L$ es el caso límite en el que no hay overlap pero todas las muestras están incluidas dentro de alguna ventana.

6.2.2 Resolución en el tiempo vs resolución en frecuencia

Existe una relación de compromiso entre la resolución en frecuencia y la resolución en tiempo de la STFT. Debido al principio de incertidumbre no es posible medir simultáneamente y con precisión arbitraria el contenido armónico de la señal así como el intervalo de tiempo en el que se encuentra.

Esta relación esta dada por:

$$\Delta t \cdot \Delta \omega \geq 2$$

6.3 Parámetros

6.3.1 Ventana

La ventana sirve dos propósitos fundamentales. El primero que es el mas evidente, es el de limitar la duración de la señal de interés para poder calcular su transformada en un intervalo acotado. El segundo es el de reducir la 'fuga espectral', esto se logra eligiendo ventanas cuyos extremos decaen rápido y suavemente a cero con el fin de evitar posibles discontinuidades en los extremos.

El largo y la forma de la ventana elegida son de suma importancia. El largo de la misma definirá cual sera la resolución en frecuencia y en tiempo del espectrograma. Como se analizo previamente, al aumentar el largo de la ventana se tiene mayor resolución en frecuencia pero menor resolución en tiempo y al disminuir el mismo se tiene mayor resolución en tiempo pero menor resolución en frecuencia.

La forma de la ventana tendrá impacto en el grado de fuga espectral que tomara lugar en el espectrograma. La fuga espectral es consecuencia de que la transformada de una señal se calcula en un intervalo finito que puede no coincidir con un múltiplo del periodo fundamental de la misma, lo que genera discontinuidades en los extremos de la extensión periódica de dicha señal y por ende agrega contenido armónico que no es propio de la misma. Las formas de ventana que disminuyen a cero tanto en su valor como en sus derivadas cerca de los extremos reducen la fuga espectral ya que disminuyen la relevancia del contenido armónico aportado por los extremos del intervalo.

6.3.2 Largo del segmento

El largo del segmento es igual largo de la ventana. Este parámetro define el largo de la ventana que a su vez define la resolución temporal y en frecuencia del espectrograma.

6.3.3 Overlap entre segmentos

El overlap entre segmentos depende de la forma de la ventana utilizada. Normalmente las ventanas tienden a cero o valores muy pequeños cerca de sus extremos por lo que si no se superponen las ventanas se puede perder información útil de la señal. Un buen criterio para decidir la superposición es pedir que la suma de la superposición de todas las ventanas sea igual a la función unitaria:

$$Sum_{windows} = \sum_{k=0}^R window(n - kH_w) = 1 \quad (34)$$

Donde la cantidad de overlap(en porcentaje) entre ventanas esta dado por :

$$Overlap(\%) = \frac{H_w}{largo(window)} \cdot 100 \%$$

Generalmente se utiliza un factor de overlap del 50 %.

6.3.4 Largo de la FFT

Como fue explicado en el desarrollo matemático, el mínimo de muestras necesarios de la fft para reconstruir la señal es igual a la duración de la señal($N = L$). Si se incrementa N sobre el valor de L ,es equivalente a agregar ceros al vector de la fft(zero-padding) no se esta agregando información nueva a la señal por lo que la resolución de la misma no mejora. Sin embargo, al tener un N mayor se obtiene una gráfica que parece ser mas continua y que puede permitir distinguir mejor rasgos del espectro.

| | |
|-----------|---------|
| Window | Hanning |
| N_overlap | 400 |
| N_fft | 4000 |
| N_Window | 800 |

Figura 92: Parámetros utilizados

6.4 Aplicación del espectrograma

Se utilizo el programa de síntesis implementado para generar la escala de sol mayor G3 mediante la síntesis aditiva, seleccionando como instrumento el violín. El resultado obtenido fue el siguiente:

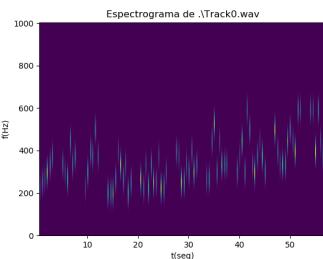


Figura 91: Espectrograma de la escala sol mayor en violín

El overlap y la ventana fueron elegidos de manera tal que ya suma de las ventanas sea la función unitaria(hanning y overlap del 50 %). En cuanto al largo de la ventana se eligio 800 muestras ya que la frecuencia mínima de la escala es de 196Hz, como el audio se guardo a una frecuencia de muestreo de 44100Hz, un periodo de un armónico de 196Hz necesita al menos $\frac{1}{196\text{Hz}} * 44100\text{Hz} = 225\text{muestras}$ para ser detectado, se tomo un numero mayor de muestras para mejorar la resolución en frecuencia. Aunque el nfft no aumente la resolución en frecuencia del espectrograma, la interpolacion que genera permite que se logren distinguir mejor los rasgos caracteristicos del espectrograma. Es por esto que se tomo un nfft bastante mayor al largo de la ventana lo que corresponde a zero-padding.

6.5 Conclusiones

Como conclusión, al realizar un espectrograma es de suma importancia ajustar los parámetros del mismo dependiendo de cual es el tipo de señal y cuales son las características que se desean ver. Es importante considerar si se le desea dar prioridad a la resolución en el tiempo o a la resolución en frecuencia dado que no es posible tener exactitud arbitraria en ambas por el principio de incertidumbre. A su vez la elección de la ventana es de suma importancia, mientras mayor su tamaño se tiene una mejor resolución en frecuencia a expensas de una peor resolución en el tiempo y vice versa. En cuanto a la forma de la ventana, esta influye en la fuga espectral, mientras mas rápido y sube sea

la transición hacia cero en los extremos de la ventana entonces menor sera la fuga espectral generada.Sin embargo, hay una relación de compromiso entre la resolución en frecuencia dada por la forma y amplitud de la ventana en el lóbulo central y la forma y amplitud de la ventana en los lóbulos extremos que define la fuga espectral.

7 Efectos de Audio

7.1 Introducción

En esta parte del trabajo, se propuso la realización de varios efectos de audio en tiempo real mediante el procesamiento digital de señales. Este procesamiento se basa en el hecho de que se puede estructurar un programa de manera que resulte un sistema para una entrada digital (de audio, en este caso) modificando ciertas características de las mismas, ya sea mediante el punto de vista del espacio de la frecuencia o del espacio temporal. Además, para poder generar dicho sistema que provoque una modificación en la entrada que sea “agradable” al usuario, hay ciertas bases de investigación, en el campo de la física, la matemática y la acústica en las cuales debese apoyar para mejores resultados.

A continuación se presentarán los efectos que se intentaron realizar, pero dicha presentación será breve a modo de tan solo acompañar los resultados obtenidos en el programa computacional de esta parte del trabajo.

7.2 Reverberadores

Un efecto reverberador busca imitar las reflexiones que se producen en una cámara reverberante (que puede utilizarse en un estudio de grabación, por ejemplo). El programa computacional realizado podrá simular de mejor manera esta situación a medida que se tengan en cuenta mayor cantidad de factores influyentes en la reverberación, como pueden ser los índices de reflexión de los materiales con los que están conformados las paredes de las habitaciones reverberantes, entre otras cosas. A continuación se presentará el eco simple, reverberador plano, pasabajos, pasa todo, por convolución y el reverberador completo.

7.2.1 Eco Simple

El eco simple se generá a partir del hecho de que la salida no solo está definida por la entrada, si no que también se le agrega lo que fue la entrada un instante de tiempo anterior, y esto se repite para cualquier instante de tiempo. Entonces la salida puede definirse mediante la siguiente expresión:

$$y(n) = x(n) + g.x(n - M)$$

Siendo M el equivalente al tiempo de retardo. Para poder apreciar un echo, es recomendable que el factor “ g ” sea de 0.5 y el tiempo de retardo sea mayor a 50ms.

7.2.2 Reverberador Plano

Si se analiza el eco simple mediante la transformada Z, el transferencia del sistema queda definida de la siguiente manera:

$$H(Z) = \frac{Y(Z)}{X(Z)} = 1 + g.Z^{-1}$$

Sin embargo, si se agregaran un número considerable de “líneas de retardo” ($y(n) = x(n) + g.x(n - M) + g^2.x(n - 2M) + \dots$) entonces el sistema se podría describir de la siguiente forma:

$$H(Z) \simeq \sum_{n=0}^{\infty} (g.Z)^{-nM} = \frac{1}{1+gZ^{-M}} \Rightarrow y(n) = g.y(n - M) + x(n)$$

En esto, entonces, se basa en un reverberador plano, el cual imita mejor los efectos de una cámara reverberante ya que en cada reflexión se añade un nuevo retardo, siendo la cantidad de veces que se dan estas reflexiones un número considerablemente “grande”.

7.2.3 Reverberador Pasa Bajos

El reverberador pasabajos, tiene en cuenta que en cada reflexión se produce un decaimiento del retardo ya que existe físicamente pérdida de energía (coeficiente de reflexión menor a 1). Entonces, se simula este hecho agregando en la línea de retardo un pasabajos (por ejemplo, con función transferencia $G(Z)$) quedando la transferencia total del sistema como $H(Z) = \frac{1}{1+(gZ^{-M}) \cdot G(Z)}$.

7.2.4 Reverberador Pasa Todo

El reverberador pasatodo es interesante ya que logra efectos cambiando la fase de la entrada pero sin generar un decaimiento en ninguna componente armónica de la misma. Una transferencia de este tipo se puede lograr mediante un sistema como el siguiente:

$$y(n) = x(n - M) + g \cdot y(n - M) - g \cdot x(n) \Rightarrow H(Z) = \frac{z^M - g}{1 - g \cdot Z^{-M}}$$

Siendo esta última transferencia la de un ecualizador de fase.

7.2.5 Reverberador Completo

Para lograr un reverberador completo es necesario aplicar varios reverberadores simples de manera que convivan los efectos de los mismos y se logre simular una habitación reverberante más cercana a la realidad. Existen diversos tipos de arquitecturas posibles, y en el programa computacional implementado en este trabajo se utilizó el esquema del “Reverberador de Moore” que consta de una serie de filtros “pasatodo” sumado a filtros “comb” en paralelo. Es de esperar la presencia de los filtros pasatodos por su efecto en la fase, pero además se simula un decaimiento en la salida mediante la implementación de los filtros comb.

7.2.6 Reverberación por Convolución

En este tipo de efecto, en lugar de simular computacionalmente un espacio reverberante, se toma la respuesta impulsiva de uno existente (y real), se digitaliza y se utiliza para obtener la salida del sistema digital mediante la aplicación de la convolución para sistemas LTI (lineal y tiempo invariante). Esto tiene la ventaja de tratar directamente con un efecto existente en la realidad, y en efecto se obtienen resultados favorables en la salida (el efecto es relativamente “cercano” al que se produciría en la realidad).

En cuanto a la respuesta impulsiva del sistema, que es digitalizada, se puede obtener mediante la grabación de la respuesta de una cámara reverberante ante un disparo de pistola, por ejemplo, realizándose la aproximación de que un impulso se puede representar mediante el mismo disparo.

7.3 Otras Distorsiones

Mediante la aplicación de sistemas que distorsionan, se pueden obtener efectos deseables para algunas situaciones específicas. A continuación se presentarán los efectos de robotización, flanger y vibrato.

7.3.1 Robotización

El efecto de robotización puede ser deseable para audios de voz, y se logra provocando que la salida de audio tenga siempre el mismo tono. Para esto, se utiliza la transformada de fourier fragmentada (FFT) para que todas las componentes armónicas posean fase nula, para luego antitransformar y lograr finalmente lo deseado.

En la implementación del programa, la fft se aplicó con un overlap de 50% con ventana de “Hanning”. Esto produjo efectos satisfactorios, que pueden variar si se varía el ancho de la ventana utilizada.

7.3.2 Vibrato

La voz humana, generada por la vibración de cuerdas vocales, posee un efecto de vibrato (de hecho, los cantantes profesionales tienen la capacidad de regular el efecto según su canto) debido a la oscilación de las ondas mecánicas. Con el programa de este trabajo se buscó simular este efecto mediante delays variables modulados por una onda senoidal de baja frecuencia. Es decir:

$H(Z) = Z^{-M}$ con " M " variable según la frecuencia de la senoidal (típicamente entre 4Hz y 14Hz, con un retardo entre 5ms y 10ms).

Con este procesamiento digital se logra un efecto de vibrato generalmente más evidente que el que existe naturalmente en las cuerdas vocales, y se suele utilizar para música como el rock.

7.3.3 Flanger

El efecto flanger, al igual que el de vibrato se logra mediante la modulación de un retardo, pero con una frecuencia aún más baja (aproximadamente de 1Hz) y además este retardo se añade a la salida en conjunto con la misma entrada. Es decir que se respeta el mismo esquema que eco simple, pero al variar lentamente un retardo relativamente bajo (aproximadamente 15ms) se logra un efecto audible que, nuevamente, es deseable en ciertos ambientes musicales.

7.4 Conclusión

Al implementar el programa de procesamiento digital para generación de efectos de audio, se pudo notar la importancia tanto del aspecto artístico y perceptivo de los efectos, como los razonamientos matemáticos para dar con un sistema que logre una respuesta satisfactoria. En el análisis digital, el modelado mediante la teoría de Fourier y los aportes realizados por diversos matemáticos y físicos fueron cruciales para obtener calidad y poder realizar lo que se ha propuesto. Además, cabe remarcar que los efectos que se han implementado tienen el beneficio de ser aplicables en tiempo real, para lo cual fue necesario aplicar un criterio de programación que permita la implementación de forma eficiente en cuanto a la velocidad de generación de un efecto, teniendo en cuenta además de no sobredimensionar la memoria que se requiere para el programa.