



Universitat d'Alacant
Universidad de Alicante

Escola Politècnica Superior
Escuela Politécnica Superior



ESCUELA POLITÉCNICA SUPERIOR

**Ingeniería técnica en telecomunicaciones especialidad en
imagen y sonido**

Proyecto fin de carrera

CONTROL EN TIEMPO REAL DE LA SÍNTESIS DEL
SONIDO MEDIANTE UN INSTRUMENTO
ELECTROACÚSTICO

Pedro Hernández Rodríguez

15 de septiembre de 2013



Universitat d'Alacant
Universidad de Alicante

Escola Politècnica Superior
Escuela Politécnica Superior



ESCUELA POLITÉCNICA SUPERIOR

Ingeniería técnica en telecomunicaciones, especialidad en imagen y sonido

CONTROL EN TIEMPO REAL DE LA SÍNTESIS DEL SONIDO MEDIANTE UN INSTRUMENTO ELECTROACÚSTICO

- Departamento: Lenguajes y Sistemas Informáticos
- Tutor del proyecto: Jose Manuel Iñesta Quereda
- Autor del proyecto: Pedro Hernández Rodríguez

Alicante, 15 de septiembre de 2013

Fdo: Pedro Hernández Rodríguez

Agradecimientos

Quiero agradecer en primer lugar a mi familia, que me ha motivado y ha servido siempre de ejemplo e inspiración a la hora de emprender proyectos de cualquier ámbito y naturaleza; académicos, musicales, de desarrollo personal y creativo; me ha alentado a superar los obstáculos que se interponen en el proceso de realización de éstos, siempre ha reconocido los logros obtenidos en caso de éxito , así como recomendado aprender de los errores cometidos en el caso contrario.

También agradezco y destaco la actitud y profesionalidad de mi tutor, en primer lugar por la aceptación de mi propuesta de proyecto, ofreciéndose como tutor del mismo, y por toda la ayuda, orientación, motivación y conocimientos que me ha proporcionado, sin los cuales habría sido imposible llevar a cabo la realización del presente proyecto.

Por último agradezco el gran apoyo de mis amigos, profesores y compañeros de carrera más cercanos, personas que han convivido conmigo durante años y me han respaldado cuando ha sido necesario.

Gracias a todos ellos por haber hecho de mí, una persona capaz.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2013 Pedro Hernández Rodríguez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Índice general

1. Introducción y contexto del proyecto	1
1.1. Introducción	1
1.2. Motivación	1
1.3. Objetivos	2
2. Conocimientos teóricos previos	5
2.1. Conceptos básicos de la teoría musical	5
2.1.1. Ritmo musical	5
2.1.2. Melodía	6
2.1.3. Armonía	8
2.2. Nociones de acústica y generación del sonido	10
2.2.1. Introducción	10
2.2.2. Oscilaciones y vibraciones armónicas	10
2.2.3. Sistemas vibrantes de dos y tres dimensiones	12
2.2.4. Instrumentos musicales de cuerda frotada	18
2.2.5. Acústica y fisiología de la audición	22
2.3. Captación y transmisión eléctrica del sonido: Transductores.	25
2.3.1. Micrófonos dinámicos	25
2.3.2. Micrófonos electrostáticos	26
2.3.3. Micrófonos piezoeléctricos	27
2.3.4. Transmisión eléctrica del sonido: Señal de audio	29
2.4. Digitalización del sonido	30
2.4.1. Muestreo (S)	31
2.4.2. Cuantificación (Q)	32
2.4.3. Tarjetas de sonido	34
2.5. Análisis de una señal de audio digital	35
2.5.1. Análisis en el dominio temporal	36
2.5.2. Análisis en el dominio frecuencial	37
2.6. Caracterización de una señal de violín eléctrico	39
2.6.1. Registro	39
2.6.2. Amplitud	40
2.6.3. Contenido espectral, Timbre	41
2.7. Técnicas de seguimiento de tono: Pitchtracking	42
2.7.1. Introducción	42

2.7.2.	Técnicas PDA en el dominio del tiempo	42
2.7.3.	Técnicas PDA en el dominio de la frecuencia	43
2.8.	Síntesis digital del sonido - Técnicas de síntesis	46
2.8.1.	Introducción	46
2.8.2.	Síntesis lineal	47
2.8.3.	Síntesis no lineal	51
2.8.4.	Síntesis computacional	55
2.9.	MIDI	56
2.9.1.	Introducción	57
2.9.2.	El estándar MIDI	58
2.9.3.	General MIDI	63
2.9.4.	Controladores	64
3.	Sistemas actuales relacionados	67
3.1.	Sistemas Software	67
3.1.1.	RTPITCH Versión 1.3	67
3.1.2.	Dalton Harts Harmonizer	68
3.2.	Sistemas Hardware (MIDI pickups)	69
3.2.1.	MIDI Shadow	70
3.2.2.	Roland GK3	70
4.	Tecnologías utilizadas	73
4.1.	Matlab	73
4.2.	CSound	75
4.2.1.	Archivos .csd	75
4.2.2.	Inicio y fin del algoritmo	76
4.2.3.	Sección de opciones	76
4.2.4.	Sección de instrumentos	76
4.2.5.	Sección de partitura	77
4.2.6.	Tipos de variables en CSound	77
4.3.	CsoundQt	78
4.4.	ASIO4ALL	79
4.5.	LoopMIDI	79
5.	Diseño e implementación del algoritmo.	81
5.1.	Estructura del sistema	81
5.2.	Experimentación en matlab	82
5.3.	Implementación en CSound	86
5.3.1.	Estructura del algoritmo	86
5.3.2.	Fases del proceso	87
5.3.3.	Interfaz	91
5.3.4.	Restricciones del sistema	92
5.3.5.	Adecuación de parámetros	93

6. Construcción de un violín electroacústico	97
6.1. Introducción	97
6.2. Cálculos y medidas básicas	98
6.3. Detalle del proceso de construcción	100
7. Test del sistema	107
7.1. Errores de detección	107
7.2. Latencia	111
7.3. Salida MIDI	112
8. Conclusiones y posibles mejoras	115
Bibliografía	117
Código fuente	119
Opcodes CSound utilizados ordenados alfabéticamente	125
Tabla de intervalos musicales	139
Funciones Bessel	141
Alturas MIDI	143
Lista de archivos en el CD adjunto:	145
GNU Free Documentation License	147
1. APPLICABILITY AND DEFINITIONS	147
2. VERBATIM COPYING	149
3. COPYING IN QUANTITY	150
4. MODIFICATIONS	151
5. COMBINING DOCUMENTS	153
6. COLLECTIONS OF DOCUMENTS	153
7. AGGREGATION WITH INDEPENDENT WORKS	154
8. TRANSLATION	154
9. TERMINATION	155
10. FUTURE REVISIONS OF THIS LICENSE	155
11. RELICENSING	156
ADDENDUM: How to use this License for your documents	156

Indice de figuras

2.1. Figuras rítmicas: notas y silencios	6
2.2. Escala de Do mayor y frecuencias fundamentales asociadas	7
2.3. Distancia tonal y relación de frecuencias (Do Mayor)	8
2.4. Detalle en el espacio del tiempo, de la superposición de ondas coherentes armónicamente.	9
2.5. Modo fundamental de vibración de una cuerda sujeta por los dos extremos	13
2.6. Vientres y nodos en los modos 2 y 4 de vibración de una cuerda sujeta por los dos extremos	14
2.7. Principales modos de vibración de una barra sujeta por un extremo	15
2.8. Principales modos circulares de una membrana sujeta	16
2.9. Segundo modo diametral de una membrana sujeta	16
2.10. Primeros modos de vibración, tubos acústicos	18
2.11. Contrabajo, violonchelo, viola y violín	19
2.12. Arco utilizado en los instrumentos occidentales de cuerda frotada	19
2.13. Movimiento de Helmholtz: "stick and slip"	20
2.14. Forma de onda "diente de sierra" provocada por el movimiento de Helmholtz	20
2.15. Elementos transmisores de la vibración en un violín	21
2.16. Partes del oído humano	23
2.17. Curvas de igual sonoridad	24
2.18. Curvas de ponderación A,B y C	25
2.19. Esquema micrófono dinámico	26
2.20. Esquema micrófono electrostático	27
2.21. Esquema micrófono piezoeléctrico.	28
2.22. Piezoeléctrico instalado en el violín.	28
2.23. Cable de audio tipo RCA y conector de 1/4 de pulgada (mini-jack)	29
2.24. Gráfico: Señal de audio balanceada	30
2.25. Cables de audio balanceado	30
2.26. Muestreo de una señal analógica	32
2.27. función de cuantificación $Q(x)$	33
2.28. Primera tarjeta de sonido alojada en un PC	35
2.29. Representación temporal de una señal periódica	36
2.30. Representación frecuencial de un sol4 reproducido por un violín	38
2.31. Espectrograma o sonograma	39
2.32. Frecuencias de las cuerdas de un violín	40
2.33. Espectro de la cuerda más grave de un violín (Sol_3)	41
2.34. Autocorrelación	43

2.35. HPS	44
2.36. Estructura método HPS	44
2.37. Generación del espectro ideal	45
2.38. Maximum Likelihood	46
2.39. Diferentes tablas de onda	47
2.40. Espectro de ruido blanco	48
2.41. Espectro filtrado	49
2.42. Esquema: síntesis aditiva	50
2.43. Espectro de una señal generada por síntesis aditiva	50
2.44. Vocoder	51
2.45. Representación temporal de las señales Moduladora, Portadora y Modulada AM	52
2.46. Espectro de una señal modulada en frecuencia	54
2.47. Ejemplo de distorsión de una señal por Waveshaping: comportamiento no lineal	55
2.48. Interfaz de un sintetizador por modelado físico.	56
2.49. Conectores tipo DIN	58
2.50. Estructura de un byte MIDI	59
2.51. Tabla: Mensajes de canal (voz)	60
2.52. Tabla: Mensajes de canal (modo)	60
2.53. Tabla: Mensajes de sistema	61
2.54. Tarjeta de implementación MIDI	62
2.55. Logos: General MIDI, GS Standard y XG MIDI	63
2.56. Distintos controladores MIDI	64
3.1. Interfaz RTPITCH 1.3	68
3.2. Interfaz Harmonizer	69
3.3. Shadow , Conversor MIDI	70
3.4. Esquema de instalación en guitarra , Roland Gk3	71
4.1. Interfaz Matlab	74
4.2. Editor de figuras Matlab	74
4.3. Estructura básica de un documento .csd	76
4.4. Cabecera de la sección de instrumentos: Configuración	77
4.5. Interfaz y widget CsoundQt	78
4.6. Interfaz ASIO4ALL	79
4.7. Interfaz LoopMIDI	80
5.1. Esquema del sistema	81
5.2. Cruces por cero	82
5.3. Tabla comparativa zerocrossing	83
5.4. Detección (F0) DEP	83
5.5. Tabla comparativa MAX DEP	84
5.6. Densidad espectral de potencia MI5 (2 armónicos añadidos)	84
5.7. Detección (F0) HPS MI5	85
5.8. Tabla comparativa HPS	85
5.9. Esquema de la estructura del algoritmo	86
5.10. Interfaz gráfico <i>FO-AMP</i>	91

6.1. Portada del tratado práctico de luthería	98
6.2. Diseño y medidas del prototipo	99
6.3. Muestras de madera de pino empleadas	100
6.4. Bloques 1 2 y 3	100
6.5. Bloques cortados y lijados	101
6.6. Clavos de unión	101
6.7. Clavijas instaladas	102
6.8. Vista trasera	102
6.9. Batedor de goma dura	103
6.10. Sistema de fijación para un correcto secado y pegado	103
6.11. Detalle del orificio	104
6.12. Alma atravesando el bloque1	104
6.13. Detalle placa y micrófono de contacto	105
6.14. instalación sistema de captación	105
6.15. pintado	106
7.1. Representación respuesta Ptrack a una escala	108
7.2. Representación respuesta Ptrack a un arpegio	109
7.3. Diferentes melodías con diferentes Técnicas (<i>ptrack, pitchamdf</i>)	110
7.4. Medición de latencia-condiciones de trabajo	111
7.5. Partitura pianoroll de la señal de prueba (arpegio SolM)	112
7.6. Partitura en notación musical de la señal de prueba previamente adecuada	113

Capítulo 1

Introducción y contexto del proyecto

1.1. Introducción

En las últimas décadas se ha avanzado mucho en el campo de la tecnología musical: Nuevas técnicas de grabación y síntesis del sonido, avances en tratamiento y edición sonora, etc. Han proporcionado herramientas muy útiles tanto para ingenieros acústicos, como para músicos profesionales o aficionados. Esto ha incrementado las posibilidades del usuario convencional, el cual, hoy en día puede adquirir procesadores y dispositivos informático-musicales por un precio relativamente razonable y asequible, de esta forma la producción musical tanto de grabación como de síntesis del sonido ya no está restringido al ámbito de los estudios profesionales de grabación ligados a su vez a las grandes marcas de la industria musical, que hasta hace pocos años eran la única vía posible de producción, grabación y distribución fonográfica.

Estas mejoras han sido posibles gracias al rapidísimo desarrollo de los procesadores informáticos y del software y hardware informático-musical. La creación del formato MIDI como lenguaje de comunicación entre dichos dispositivos así como el desarrollo de lenguajes de programación enfocados a la síntesis sonora son un ejemplo de las más importantes innovaciones que la tecnología actual nos brinda, cada vez con más eficiencia.

1.2. Motivación

Personalmente, el hecho de ser músico profesional y estar paralelamente estudiando ingeniería técnica en Telecomunicaciones, ha sido clave para alimentar y experimentar inquietudes creativo-musicales respaldadas por una sólida base científica. Muchos conceptos y procesos físicos, acústicos y electroacústicos asentados y asimilados previamente en mis estudios musicales, se han estudiado con mucha más profundidad y detalle en la carrera de ingeniería técnica, la cual me ha dotado de herramientas científicas y técnicas para el diseño y análisis de sistemas y dispositivos informático-musicales.

Concretamente en lo que atañe a este proyecto de fin de carrera, ampliar las posibilidades tímbricas de mi instrumento, el violín, ha sido siempre una de mis grandes inquietudes, desde hace bastantes años, antes de empezar a estudiar la carrera, he experimentado con dispositivos tales como pedaleras de efectos, violines electrónicos, así como con diferentes técnicas de interpretación, con el fin de variar el sonido del violín para poder tocar con bandas musicales de diferentes estilos y experimentar con nuevas texturas sonoras, al fin y al cabo, dichos dispositivos y técnicas, distorsionan o modifican la señal “fuente” proveniente del instrumento, es decir, están estrictamente ligados a las características de la señal del violín, limitando así la posibilidad de una mayor variación tímbrica.

Al contrario, con la técnica empleada en este proyecto, la señal del violín no se modifica ni se distorsiona para generar sonidos, sino que se analiza y se extrae de ella la información básica de altura tonal y amplitud, siendo otros instrumentos virtuales programados en c-sound los que sintetizan el sonido, de naturaleza radicalmente diferente y con una posibilidad enorme de variación.

Además, la necesidad de conectar y controlar diferentes dispositivos musicales, y que la interacción se produzca a través de un instrumento con el cual el usuario intérprete esté familiarizado, es notable y creciente; este proyecto de final de carrera es una aplicación informática que contribuye también a satisfacer dicha necesidad.

1.3. Objetivos

En la aplicación o programa se digitaliza el sonido emitido por un instrumento electro-acústico para que sirva como controlador de un sistema de síntesis o de gestión de secuencias de control de la misma. De esta forma se consigue la posibilidad de expansión tímbrica ilimitada, controlada en tiempo real por un instrumento musical electro-acústico, (cuya construcción también ha sido efectuada artesanalmente como complemento a la aplicación y será comentada y analizada en posteriores apartados), aumentando así la versatilidad del intérprete de un instrumento tradicional, así como la capacidad experimental y creativa del músico.

Este algoritmo analiza la señal de entrada de audio del dispositivo, a la que se conectará un instrumento musical electro-acústico, obtiene y hace un seguimiento en tiempo real de los parámetros necesarios (frecuencia, duración y amplitud) para controlar sintetizadores software. En posteriores capítulos se compararán y analizarán diferentes técnicas para realizar esta labor.

El programa incluye varias técnicas de síntesis: Aditiva, FM, Sampler... que generan a su vez el sonido partiendo de los parámetros frecuencia fundamental (F0) y amplitud detectadas en el análisis, ofreciendo al usuario la posibilidad de selección de un instrumento sonoro programado y la alteración de parámetros asociados a dicho instrumento.

A la señal generada por los mencionados instrumentos sintéticos se les puede añadir algún efecto sonoro básico independiente como la reverberación, el eco (delay), y modificar sus parámetros básicos como: reverb time, mix, timedelay, feedback... etc.

También proporciona la opción de convertir esta señal analizada en una secuencia de comandos de protocolo MIDI con la mínima latencia, con el fin de permitir tanto el control de sintetizadores internos software, hardware... así como la construcción de secuencias MIDI para elaboración de partituras digitales en diferentes formatos (secuenciación en tiempo real).

Capítulo 2

Conocimientos teóricos previos

En este capítulo se hablará resumidamente de multitud de conceptos musicales y de la ingeniería, cuyo conocimiento es indispensable para poder comprender y analizar el proceso de desarrollo del proyecto.

2.1. Conceptos básicos de la teoría musical

Llamamos música, al resultado de reproducir diversos eventos sonoros, con propiedades tonales y rítmicas, secuenciados y combinados inteligentemente, que producen una sensación determinada en el oyente, los productores de dichos sonidos son los llamados instrumentos musicales.

Esta actividad considerada arte y forma de expresión universal, es tan antigua como el ser humano, y posee multitud de variantes, estilos, y tendencias, dependiendo de la época, zona geográfica etc. En este resumen, se hablará sobre todo de la música occidental, sus características básicas y sus instrumentos principales.

Cabe destacar tres conceptos básicos, dos de los cuales son indiscutibles a la hora de discriminar entre música o secuencias sonoras no musicales, en orden creciente de complejidad: Ritmo, melodía y armonía.

2.1.1. Ritmo musical

Hay múltiples definiciones del concepto ritmo musical, prácticamente todas coinciden en que el sentido del ritmo es una de las capacidades de los seres humanos, que los distingue del resto de seres vivos.

Haciendo una descripción personal, me aventuro a decir que el ritmo pertenece a la música tanto como la música al ritmo, considerando el concepto ritmo musical como determinadas secuencias de sonidos sin propiedad tonal, o armonicidad necesaria, que se suceden unos a otros

inteligentemente dispuestos en el espacio del tiempo.

Para muchos teóricos expertos, el ritmo es el único elemento indispensable de la música.

La duración de estos sonidos y silencios es la que precisamente ordena la secuencia. Los siguientes nombres y símbolos son, según la escritura actual occidental, la conversión o traducción al papel, de las mencionadas duraciones.

Antes de definir las duraciones rítmicas es necesario conocer el concepto de tiempo.

Tempo: velocidad con la que se reproduce o ejecuta una pieza musical, a lo largo de la historia la indicación del tempo se ha efectuado de varias formas, antes de la invención del metrónomo, se relacionaba con los estados de actividad humanos, lento, andante, allegro, presto etc. En occidente actualmente se indica el número de pulsos, tiempos musicales o “beats” por minuto, a más tempo, más pulsos se reproducen en un minuto, es decir más velocidad posee la pieza interpretada.

Las figuras rítmicas del siguiente esquema, no tienen una duración determinada en segundos, sino que dependen exclusivamente del tempo que tome la figura principal asociada al pulso o “beat” (generalmente la negra) y tienen una relación de proporción entre ellas.



Figura 2.1: Figuras rítmicas: notas y silencios

En los esquemas anteriores se intuyen las proporciones de valor entre unas figuras rítmicas y otras, por ejemplo: una redonda equivale a dos blancas, y sucesivamente.

2.1.2. Melodía

Una melodía es también, una secuencia monofónica en este caso de eventos sonoros con características tonales o de armonicidad además de las rítmicas estudiadas anteriormente, es decir sonidos que al escucharlos revelan una cierta frecuencia fundamental que les caracteriza tonalmente, estos poseen altura tonal, cabe mencionar que una parte muy importante del desarrollo del presente proyecto se basa en la detección de esta frecuencia fundamental, que nosotros llamaremos F0 y que define la altura tonal.

Para que una secuencia de eventos sonoros con altura tonal sea considerada melodía musical, al igual que en el ritmo, debe tener una estructura temporal organizada inteligentemente, hay multitud de posibilidades melódicas, combinaciones características y por supuesto muchas teorías, reglas y métodos que estudian el proceso de creación de melodías o su análisis musical.

Las alturas tonales están directamente relacionadas con la frecuencia fundamental, que a continuación definiremos.

Cualquier sonido con carácter tonal, que produce un instrumento, puede ser descompuesto como superposición de una vibración de frecuencia fundamental, que es la de valor más bajo y armónicos superiores, múltiplos de la mencionada F_0 , vibraciones con valores más elevados en frecuencia.

Estas frecuencias fundamentales asociadas a alturas tonales se pueden representar con la terminología expresada en el siguiente gráfico.

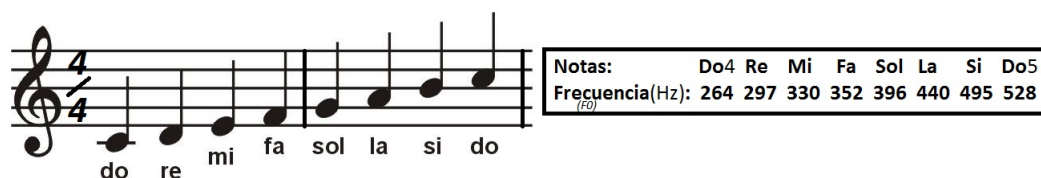


Figura 2.2: Escala de Do mayor y frecuencias fundamentales asociadas

Como se puede apreciar, las figuras rítmicas de las que hablamos en el apartado anterior, están dispuestas sobre una división espacial horizontal producida por cinco líneas, llamada pentagrama, a mayor altura tonal, la figura asciende de posición, siendo las posibles ubicaciones justo encima de la línea o en un espacio entre líneas.

Al principio del pentagrama aparecen varios símbolos, la clave de sol, que hace referencia al intervalo de alturas tonales en el cual se van a representar las figuras rítmicas del pentagrama, dicha clave determina que la representación gráfica de la nota sol 4 (395Hz) se dispondrá sobre la segunda línea desde abajo del pentagrama.

También aparece una indicación rítmica de compás, que revela el número de pulsos o “beats” por compás, lo cual caracteriza rítmicamente a la melodía o fragmento musical de una forma más global.

■ Intervalos

Las distancias entre varias alturas tonales se denominan intervalos, se clasifican por la diferencia de tonos y semitonos entre ambas notas, o también por la relación proporcional entre

sus frecuencias fundamentales, los diferentes intervalos están definidos y organizados de la siguiente forma ¹:

Nombre del intervalo	Valor en tonos y semitonos	Relación de frecuencias
Unísono ² / Primera justa	0	1
Segunda mayor	1 tono	9/8
Tercera mayor	2 tonos	5/4
Cuarta justa	2 1/2 tonos	4/3
Quinta justa	3 1/2 tonos	3/2
Sexta mayor	4 1/2 tonos	5/3
Séptima mayor	5 1/2 tonos	15/8
Octava justa	6 tonos	2

Figura 2.3: Distancia tonal y relación de frecuencias (Do Mayor)

2.1.3. Armonía

Previamente definiremos acorde como conjunto de notas musicales que se reproducen simultáneamente en el tiempo y guardan algún tipo de relación en cuanto a frecuencia fundamental, los acordes más comunes son los formados por 3 notas, llamados tríadas. Existe una gran variedad de acordes; mayores menores, aumentados y disminuidos, etc.

Este apartado de la teoría musical es muy extenso, pero no vamos a profundizar demasiado en él, puesto que no es del todo relevante para el propósito de comprensión del proyecto.

¹”En el apartado de anexos se incluye la tabla completa de intervalos”

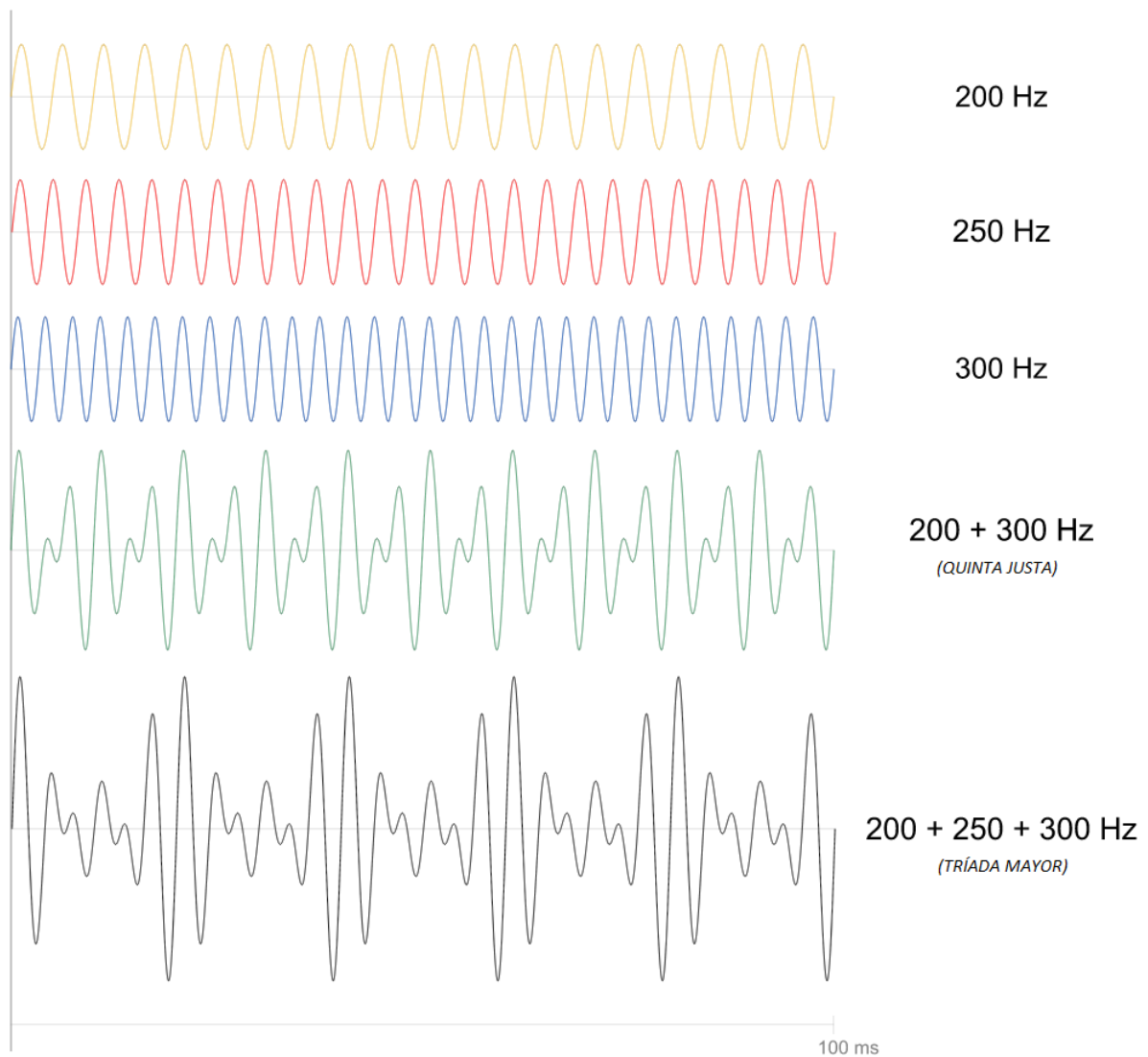


Figura 2.4: Detalle en el espacio del tiempo, de la superposición de ondas coherentes armónicamente.

La armonía es la disciplina que estudia la sucesión de los mencionados acordes, llamadas cadencias de acordes. Como en la melodía, para su análisis hay multitud de teorías, metodología y estudios sobre la armonía, clásica y moderna. Dado el carácter monofónico del sistema que estamos estudiando, no hay necesidad de ahondar en detalles de este campo de la teoría musical.

2.2. Nociones de acústica y generación del sonido

En este apartado, se explicarán brevemente algunos conceptos de la acústica así como el proceso de generación del sonido, instrumentos musicales y sus aspectos acústicos básicos.

2.2.1. Introducción

La acústica es la ciencia que estudia el sonido, su producción, transmisión y efectos, se trata de una materia interdisciplinar, pues requiere conceptos y técnicas de la mecánica, termodinámica etc. Además, los problemas y cuestiones que trata la acústica están extendidos en muchos campos del conocimiento humano, algunos ejemplos de esta diversidad disciplinar pueden ser la música, la medicina y la arquitectura.

Cabe destacar que consideramos también acústicas, a las vibraciones con frecuencias fuera de los valores audibles para el ser humano, llamadas ultrasonidos si están por encima del rango audible (mayores que 18000 Hz aprox.) o infrasonidos si están por debajo (menores de 30 Hz aprox.), pero en este proyecto prácticamente sólo se van a abordar cuestiones referentes al rango o intervalo audible ya que se trata de una aplicación enfocada a la innovación musical.

2.2.2. Oscilaciones y vibraciones armónicas

La generación del sonido requiere de un fenómeno vibratorio; En este apartado se describen varios tipos de vibraciones y se plantean analogías mecánicas (masa, muelle, rozamiento...) para poder estudiar matemáticamente su comportamiento. Un movimiento se puede catalogar como oscilatorio si cumple con las siguientes características: su movimiento está acotado entre los límites

$$X = +A$$

y

$$X = -A$$

siendo A la amplitud o elongación máxima del movimiento, y debe tratarse de un movimiento periódico siendo T el periodo de la vibración, es decir el tiempo que transcurre hasta que la posición de la masa vuelve a ser la misma. A su vez definimos la frecuencia lineal de la vibración f como el número de oscilaciones que suceden en un segundo, así se establece que el periodo y la frecuencia lineal tienen una relación inversamente proporcional

$$f = 1/T$$

■ Oscilador armónico simple

La analogía de este movimiento consiste en un sistema masa muelle que puede vibrar libremente, es un movimiento que no sucede naturalmente, pero su estudio matemático ayuda a simplificar en entendimiento de los conceptos vibratorios.

la ley de Hooke establece que el alargamiento de un material elástico es proporcional a la fuerza que se le aplica

$$F = -kx$$

Siendo k la constante elástica del material deformado, que depende de sus propiedades de composición, estructura etc. y x la posición de la masa. Conociendo la fuerza aplicada, la masa del muelle y midiendo la elongación producida, si despejamos " k " de la ecuación anterior podemos determinar experimentalmente la constante elástica del resorte.

Como también: $F = ma$ se puede determinar la ecuación del movimiento desarrollando:

$$F = m \cdot a \rightarrow -k \cdot x = m \cdot \frac{d^2x}{dt^2} \rightarrow \frac{d^2x}{dt^2} + \omega_0^2 \cdot x = 0$$

Cuya solución empleando la función coseno resulta la ecuación del movimiento armónico simple:

$$X(t) = A \cos(\omega_0 t + \varphi_0) \quad (2.1)$$

Donde: φ_0 es la fase inicial de la vibración, depende de la posición en la que comienza a vibrar la masa, en el caso de que el movimiento se inicie en $+A$, la fase inicial será igual a cero. y ω_0 es la frecuencia angular del movimiento que tiene la siguiente relación con la masa y la constante elástica $\omega_0 = \sqrt{\frac{k}{m}}$ por lo tanto el periodo de la oscilación es también:

$$T = \frac{2\pi}{\omega_0} = 2\pi \sqrt{\frac{m}{k}}$$

■ Oscilador armónico amortiguado

Se trata de un caso más real en el cual se contempla la fuerza de rozamiento, que a su vez es proporcional a la velocidad de la masa

$$F_{roz} = -R \cdot v$$

Donde R es la constante de rozamiento característica del material de la superficie en contacto con la masa.

En este caso la amplitud A depende del tiempo, esto es debido a que existe un amortiguamiento provocado precisamente por la mencionada fuerza de rozamiento.

La solución de la ecuación del movimiento armónico amortiguado es la siguiente:

$$X(t) = A(t) \cos(\omega t + \varphi_0) = A_0 e^{-\lambda t} \cos(\omega t + \varphi_0) \quad (2.2)$$

Se observa que: $A(t)$ que es la amplitud dependiente del tiempo se sustituye por $A_0 e^{-\lambda t}$ siendo λ el coeficiente o parámetro de amortiguamiento.

La frecuencia angular no posee subíndice por que también se ve alterada ligeramente por el amortiguamiento.

Tanto la frecuencia angular amortiguada como el coeficiente de rozamiento obtienen con las siguientes ecuaciones.

$$\omega = \sqrt{\frac{k}{m} - \frac{R^2}{4m^2}} = \sqrt{\omega_0^2 - \lambda^2}$$

$$\lambda = \frac{R}{2m}$$

Para la comprensión del proyecto es necesaria la definición de un parámetro asociado al movimiento armónico amortiguado, se trata del **Tiempo de relajación** τ . que describiremos como el tiempo que transcurre desde una amplitud máxima hasta que el valor de A es e veces menor. Está inversamente relacionado con el coeficiente de rozamiento. $\tau = \frac{1}{\lambda}$

■ Oscilador armónico Forzado

En este caso se le aplica una fuerza exterior con un cierto periodo al ya mencionado movimiento amortiguado. pudiendo distinguir dos fases de movimiento, fase transitoria (desde el principio del movimiento hasta que se establecen constantes los parámetros A y ω), y la fase estacionaria (a partir de ese preciso instante en adelante) cuyas soluciones a las ecuaciones de movimiento son:

$$X(t) = A_0 e^{-\lambda t} \cos(\omega t + \varphi_0) \quad (2.3)$$

para la parte transitoria, que es equivalente al movimiento amortiguado, y:

$$X(t) = \frac{f_0}{\sqrt{(\omega_0^2 - \omega_F^2)^2}} \cos(\omega_F t - \arctg(\frac{2\lambda \cdot \omega_F}{\omega_0^2 - \omega_F^2})) \quad (2.4)$$

En su fase estacionaria. Donde ω_F es la velocidad angular del sistema que provoca la fuerza externa aplicada.

2.2.3. Sistemas vibrantes de dos y tres dimensiones

Este apartado resumirá los tipos de sistemas vibrantes reales y proporcionará sus ecuaciones características.

■ Vibraciones en cuerdas sujetas por dos extremos

Una cuerda sujeta por dos extremos y con una cierta tensión es un sistema vibrante real, de hecho se trata de la fuente generadora de sonido en los instrumentos musicales de cuerda como el violín, por tanto, este apartado tiene mayor relevancia en cuanto a los conceptos previos necesarios para la comprensión del presente proyecto.

Al excitar golpeando, pulsando o bien frotando una cuerda se produce una onda transversal unidimensional (eje x en el gráfico) y estacionaria (dadas las condiciones de contorno), que se mueve con una velocidad de propagación $v = \sqrt{\frac{T}{\mu}}$ siendo T la tensión en la cuerda y μ su densidad lineal, provocando así el movimiento en la masa de un trozo infinitesimal de la cuerda en dirección perpendicular a la propagación de la onda (eje y en el gráfico).

La posición de los diferentes trozos infinitesimales de la cuerda se determina con la siguiente ecuación:

$$y(x) = A \operatorname{sen}\left(\frac{\omega}{v} \cdot x + \varphi_0\right) \quad (2.5)$$

El valor frecuencial de la oscilación producida en una cuerda es inversamente proporcional a la longitud de la cuerda y directamente proporcional a la tensión a la que está sometida, por consiguiente también posee relación directa con la velocidad de propagación transversal.

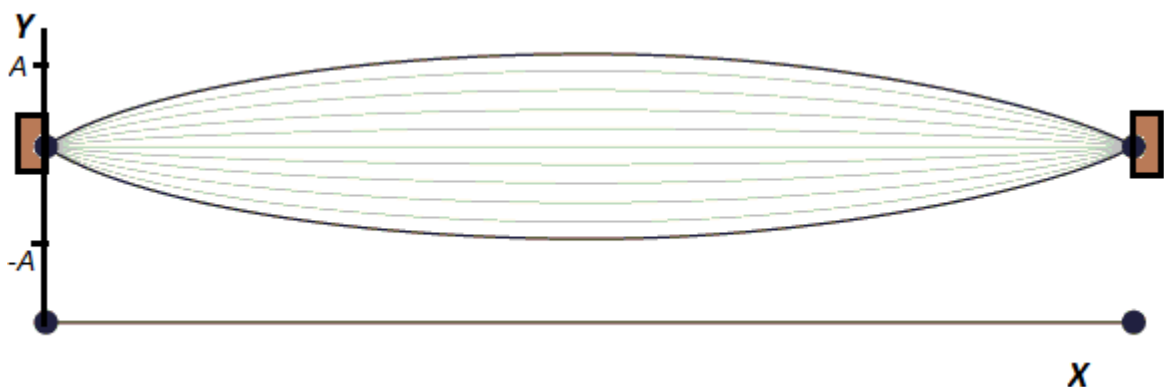


Figura 2.5: Modo fundamental de vibración de una cuerda sujeta por los dos extremos

Dependiendo de la zona de la cuerda donde se produzca la excitación o perturbación, y de la técnica empleada, pueden producirse diferentes modos de vibración, por la aparición de nodos donde la porción infinitesimal de la cuerda, no se mueve, sin contar claro está con los nodos existentes en los extremos.

En el siguiente gráfico se observa la disposición de nodos y vientres para los modos $n=2$ y $n=4$.

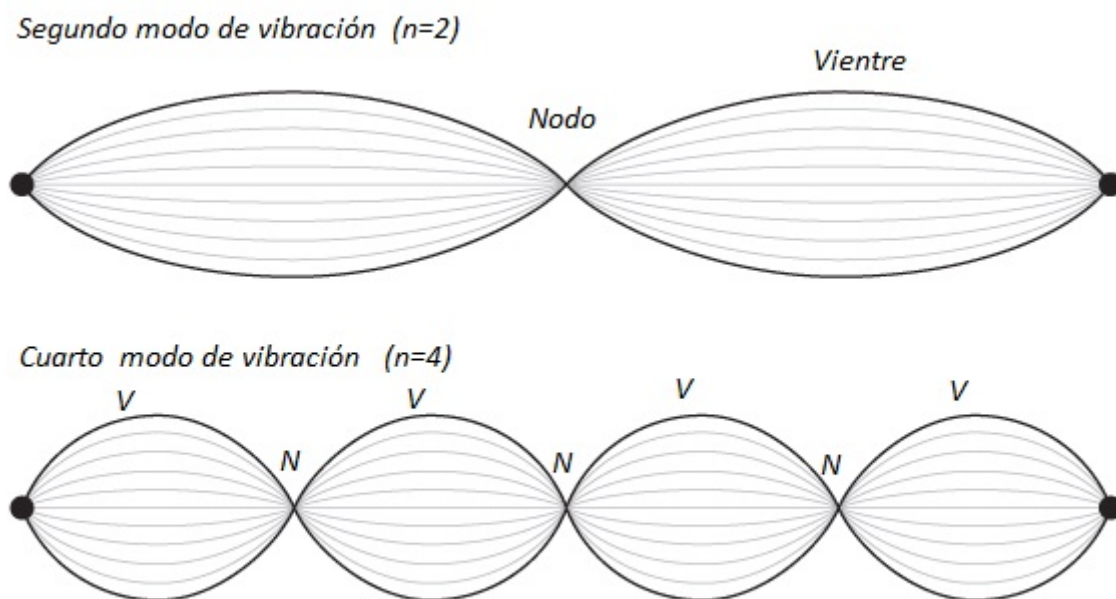


Figura 2.6: Vientres y nodos en los modos 2 y 4 de vibración de una cuerda sujeta por los dos extremos

Si llamamos L a la longitud total de la cuerda, se observa que la posición de el primer nodo sin contar extremos, se produce en $L/2$ para $n = 2$, y en $L/4$ para $n = 4$, y así sucesivamente. En violines, guitarras e instrumentos similares, estos modos de vibración se pueden forzar impidiendo ligeramente con el dedo el desplazamiento oscilatorio de la cuerda justamente en el lugar del primer nodo, dependiendo del modo que se desee excitar. De esta forma se generan los armónicos principales de la cuerda relacionados con los primeros modos de vibración. Las frecuencias lineales resultantes de los modos n -ésimos se calculan con la siguiente ecuación:

$$f_n(x) = A_n \text{sen}\left(\frac{n\pi}{L} \cdot x\right) \quad (2.6)$$

(Para valores enteros de $n = 1,2,3\dots$)

Cabe destacar que los instrumentos de cuerda suelen producir generalmente vibraciones de modo fundamental, como se puede deducir de la fórmula anterior; La variación de la longitud efectiva de la cuerda vibrante provoca un cambio inversamente proporcional en la frecuencia, es decir que si colocamos un dedo acortando la cuerda, las frecuencia resultantes de sus modos aumentarán proporcionalmente. Es así como se consiguen las frecuencias de las notas musicales en la mayoría de los instrumentos de cuerda, salvo en el piano por ejemplo, que posee tantas cuerdas como notas puede producir.

■ Vibraciones en barras

Este apartado carece de relevancia para el propósito de este proyecto, por lo tanto se proporcionarán muy resumidamente las fórmulas del comportamiento vibratorio en barras.

Consideramos una barra de longitud L de un material relativamente elástico sujeta por un extremo, y también la constante específica, del material **Módulo de Young** E que interviene directamente en la velocidad de propagación de la onda a través del material y es análogo a la constante elástica en los resortes.

La posición de las diferentes secciones infinitesimales de la barra cuando vibra se determina con la siguiente ecuación:

$$y(x) = A \operatorname{sen}\left(\frac{\omega}{v} \cdot x\right) \quad (2.7)$$

Y, al igual que en el apartado anterior las frecuencias asociadas a los distintos modos de vibración se pueden obtener con la ecuación:

$$f_n = \frac{2n - 1}{4L} \cdot v \quad (2.8)$$

Siendo: $v = \sqrt{\frac{E}{\rho}}$ la velocidad de propagación y ρ la densidad volumétrica del material.

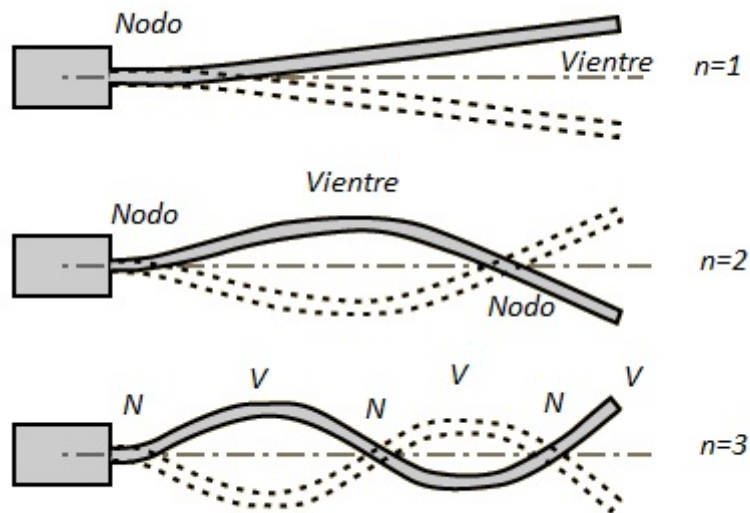


Figura 2.7: Principales modos de vibración de una barra sujeta por un extremo

la posición de los nodos y los vientres en el caso de la barra sujeta por los dos extremos, es la misma que la mostrada anteriormente en las cuerdas, también en este apartado son válidas las expresiones del capítulo anterior para la obtención de las frecuencias de los distintos modos de vibración.

■ Vibraciones en Membranas y placas

El comportamiento de las membranas completamente sujetas y tensas es análogo al de las cuerdas sujetas por los dos extremos, con la diferencia de que la velocidad de desplazamiento de la perturbación en este caso es superficial, es decir, sucede en 2 dimensiones, mientras que las porciones de masa infinitesimales de la membrana oscilan en dirección normal al plano.

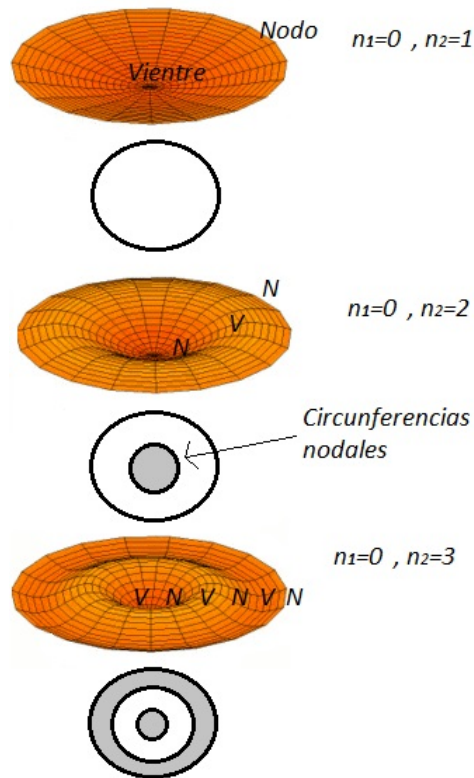


Figura 2.8: Principales modos circulares de una membrana sujeta

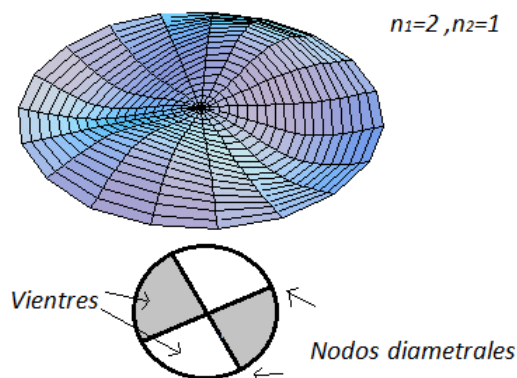


Figura 2.9: Segundo modo diametral de una membrana sujeta

En este caso n_1 corresponde al número de nodos diametrales existentes y n_2 al número de nodos circulares.

De la misma forma que en el estudio de las cuerdas, en placas vibrantes también podemos calcular las diferentes frecuencias fundamentales de los modos de vibración con la ecuación:

$$W_{n_1, n_2} = v \cdot \sqrt{\left(\frac{n_1 \pi}{L_x}\right)^2 + \left(\frac{n_2 \pi}{L_y}\right)^2} \quad (2.9)$$

Siendo v como es habitual, la velocidad de propagación de la onda $v = \sqrt{\frac{T}{\sigma}}$, T La fuerza de contorno y σ la densidad superficial de la membrana (kg/m^2). Este tipo de vibraciones se dan en muchos de los instrumentos de percusión así como en las superficies de las cajas de resonancia de muchos otros instrumentos musicales como el violín, la guitarra, el arpa, etc. No sucede de la misma forma en instrumentos de viento.

■ Tubos sonoros

En los apartados anteriores se han analizado las vibraciones elásticas, producidas en sistemas de una y dos dimensiones donde oscilaban cuerdas, placas o membranas.

Este apartado contempla las oscilaciones locales de la presión en el aire, es decir las ondas sonoras, cuya propagación sucede en las tres dimensiones, ondas esféricas en el espacio, y particularmente se hablará, para conocer el funcionamiento acústico de los instrumentos de viento, de las variaciones de presión producidas en el interior de tubos acústicos abiertos y cerrados por un extremo.

La velocidad de propagación del sonido en el aire queda, en función de las variables termodinámicas:

$$v = \sqrt{\frac{\gamma \cdot R \cdot T_0}{M}} \quad (2.10)$$

Donde M es la masa molecular del aire ($28,9Kg/Kmol$) aproximadamente, R es la constante de los gases ideales, γ Constante adiabática de los gases ($\gamma_{aire} = 1042$) y T_0 la temperatura absoluta del aire en grados kelvin.

Comúnmente para cálculos prácticos se utiliza la siguiente fórmula más reducida:

$$v = 331,6 + 0,6 \cdot t \quad (2.11)$$

Expresando la temperatura en grados centígrados en este caso.

la perturbación de presión en el interior de un tubo puede originarse al soplar tangente hacia la abertura, y esta perturbación viene condicionada por sus dimensiones, la longitud y su radio de abertura, y al igual que en cuerdas se pueden producir varios modos de vibración, para tubos abiertos, y otros diferentes para tubos cerrados por un extremo:

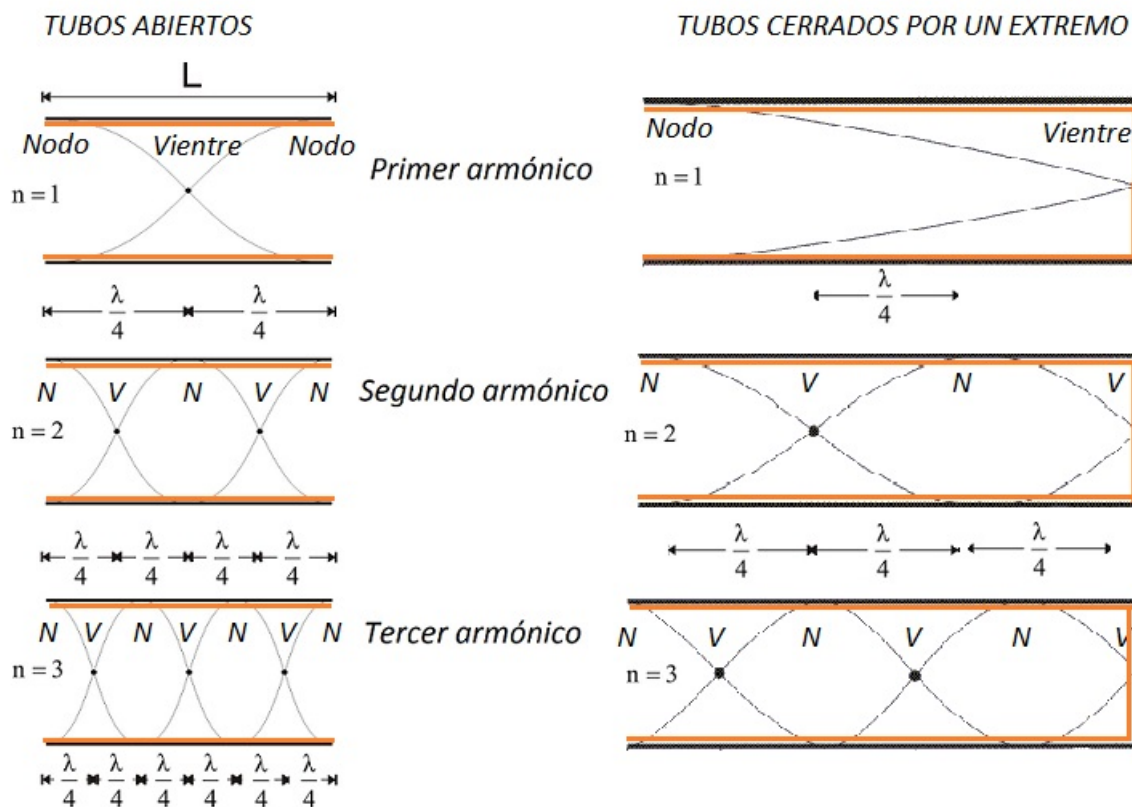


Figura 2.10: Primeros modos de vibración, tubos acústicos

En tubos abiertos, la fórmula para obtener las frecuencias de los diferentes armónicos es la siguiente:

$$f_n = n \cdot \frac{v}{2 \cdot L} \quad (2.12)$$

Por otra parte en tubos cerrados usaremos:

$$f_n = (2n - 1) \cdot \frac{v}{4 \cdot L} \quad (2.13)$$

Tomando n siempre valores enteros y positivos.

2.2.4. Instrumentos musicales de cuerda frotada

Habiendo explicado el fenómeno vibratorio de las cuerdas en los apartados anteriores, podemos comprender mejor el funcionamiento y la generación de sonido de instrumentos como el violín, guitarra o similares.

Concretamente la familia de los instrumentos de cuerda frotada occidentales poseen algunas características propias relevantes que les diferencian del resto de instrumentos de cuerda.

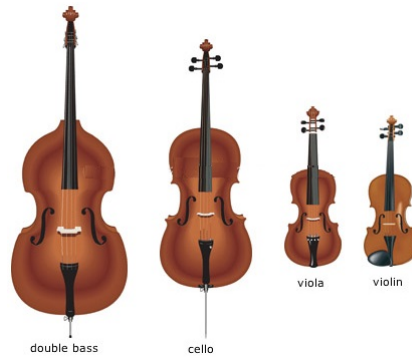


Figura 2.11: Contrabajo, violonchelo, viola y violín

Estos instrumentos utilizan el arco, una vara de madera la cual tensa varias fibras (tradicionalmente pelo de caballo), que por fricción provoca una perturbación continua en la cuerda, es decir, a diferencia de otros instrumentos de cuerda como el piano o la guitarra, la vibración no se amortigua hasta que deja de haber movimiento relativo y perpendicular arco-cuerda.



Figura 2.12: Arco utilizado en los instrumentos occidentales de cuerda frotada

Este especial sistema de excitación de una cuerda dota a estos instrumentos de un timbre característico y merece un estudio en profundidad dada su relación directa con el presente proyecto.

A continuación vamos a analizar este proceso de fricción.

Herman von Helmholtz (1821-1894), aprovechando el microscopio de vibraciones propuesto por Lissajous (1822-1880), observó este movimiento llamado "stick and slip." "pegar y deslizar" que sucede al producirse el contacto arco-cuerda perpendicularmente y con cierta velocidad, también llamado movimiento de Helmholtz.

Cabe destacar que a las fibras del arco se les aplica regularmente una sustancia coloidal tipo resina para aumentar su coeficiente de rozamiento. Esta resina especial para arcos de instrumentos tiene la característica de permanecer seca y sólida a temperatura ambiental, pero con la fricción se calienta aumentando su viscosidad y su temperatura más de 20 grados centígrados, proporcionando un agarre (fibra-cuerda) mucho mayor.

En la figura siguiente podemos observar cómo en la fase "slip", el arco y la cuerda tienen movimientos en sentido opuesto, y en la fase "stick." ambos movimientos tienen la misma velocidad y sentido.

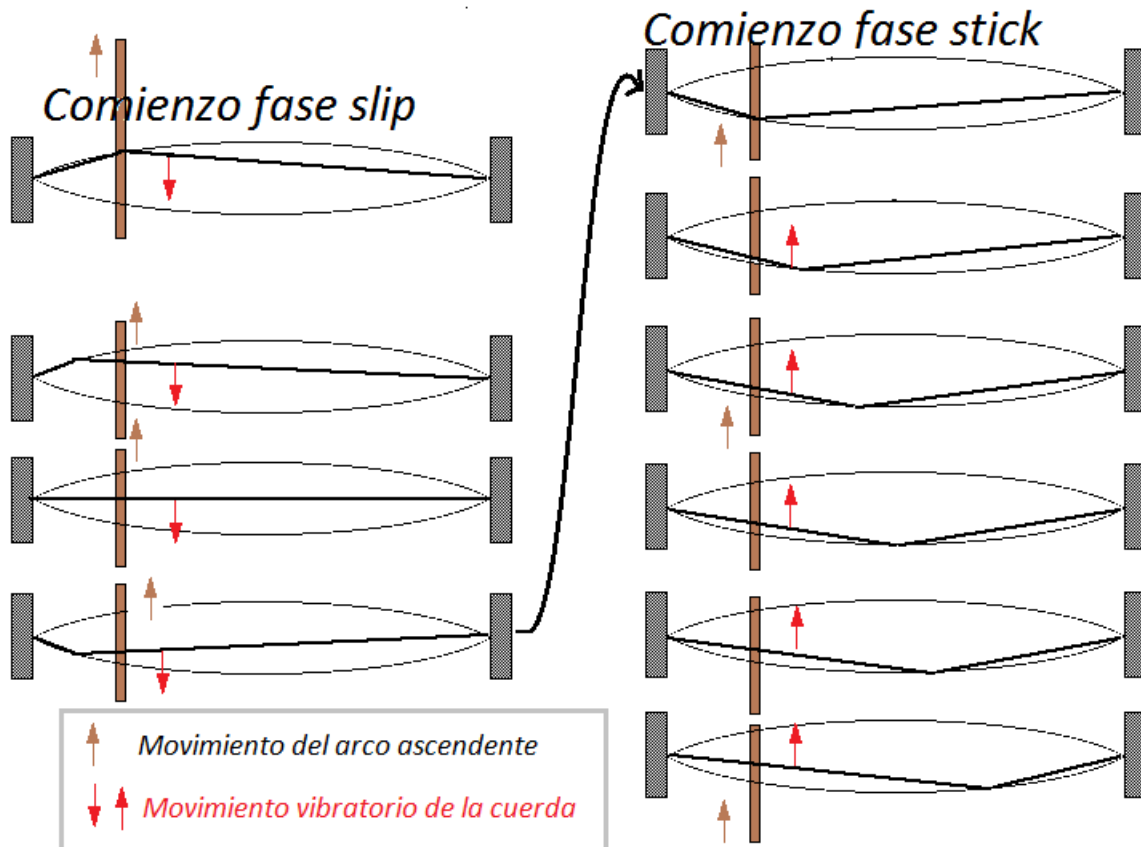


Figura 2.13: Movimiento de Helmholtz: "stick and slip"

Resultando de este movimiento, una forma de onda tipo diente de sierra.

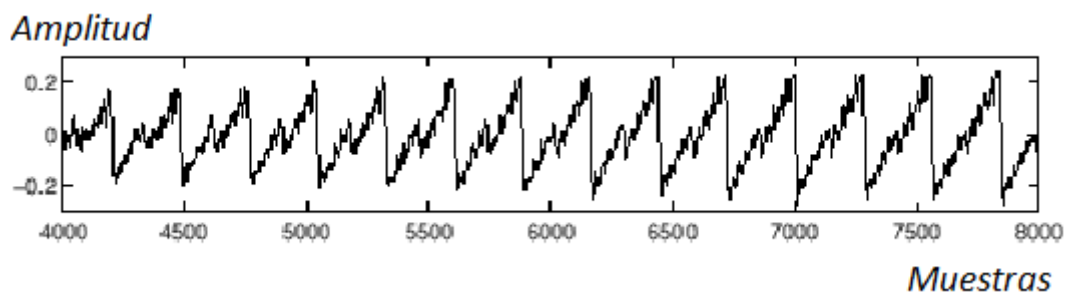


Figura 2.14: Forma de onda "diente de sierra" provocada por el movimiento de Helmholtz

Para obtener una vibración válida, es decir, con características tonales y armónicas, en un violín u otro instrumento de su familia, se debe conseguir un equilibrio entre la presión aplicada del arco sobre la cuerda y la velocidad del arco, de lo contrario se producirán sonidos inarmónicos considerablemente desagradables, pero esta cuestión pertenece más al estudio de la interpretación de dichos instrumentos.

Otra característica importante en los instrumentos de cuerda frotada es la ausencia de trastes, al contrario que en las guitarras o instrumentos derivados. Esto proporciona la capacidad de emitir, cualquier frecuencia dentro de la tesitura o registro del instrumento, podría decirse que la resolución en frecuencia de un instrumento de esta familia, es infinita, no como en una guitarra cuya resolución frecuencial es de semitono y se atribuye a la existencia de trastes.

La vibración en las cuerdas de un violín o instrumento similar se transfiere también al puente, que es una pieza de madera en uno de los extremos de la longitud vibrante, y este a su vez transmite dicha vibración a la barra armónica y a el alma, de esta forma , la tapa inferior y superior de la caja de resonancia vibran en consonancia perturbando el aire contenido en ella. Esta onda sonora es emitida al exterior por los orificios de la caja llamados Efes .

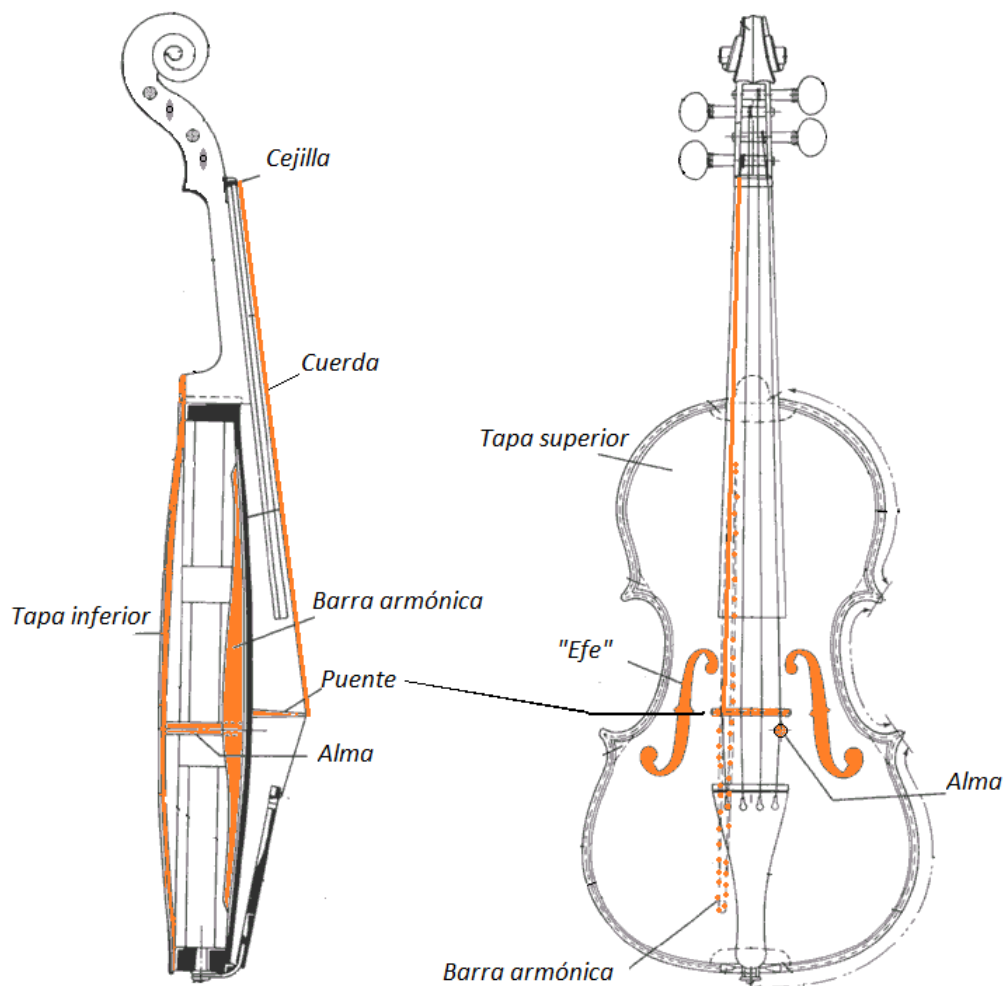


Figura 2.15: Elementos transmisores de la vibración en un violín

2.2.5. Acústica y fisiología de la audición

■ Fisiología del oído humano

El oído humano es el órgano encargado de la percepción del sonido, que sucede en cadena comenzando con la llegada del sonido al oído externo, a continuación pasa por el oído medio y sigue hasta el oído interno, donde se convierte en una señal nerviosa o neurológica que el cerebro puede interpretar.

A continuación se va a explicar brevemente la función de cada uno de los procesos de este sistema en cadena.

El oído externo está compuesto del pabellón auditivo y el conducto auditivo exterior, la función de el pabellón (oreja) es focalizar el sonido, logrando un incremento hasta de 5 dB, esta vibración sonora viaja a continuación por el conducto auditivo hasta llegar al tímpano.

El oído medio formado por el tímpano, la caja timpánica y la cadena de huesecillos (martillo yunque y estribo). El tímpano es una membrana fina que separa el conducto auditivo de la caja timpánica, su función es la transformación de las ondas sonoras en vibraciones mecánicas. La cadena de huesecillos comunican físicamente el tímpano con la ventana oval del oído interno transmitiendo la vibración de una membrana a otra y consiguiendo una amplificación en presión casi 20 veces mayor que la detectada en el tímpano, esto es en parte debido a la diferencia entre las áreas de ambas membranas, todo el sistema actúa como un adaptador de impedancias acústicas.

En el oído medio también existe un conducto que comunica la caja timpánica con la garganta llamado **trompa de Eustaquio** y su función es garantizar un equilibrio de presiones dentro y fuera del oído medio, normalmente está cerrada pero cuando se traga o se bosteza se abre, de ahí la necesidad de producir estos movimientos cuando se tiene sensación de oído taponado al subir o bajar repentinamente una distancia considerable (cambio de presión).

El oído interno posee dos partes con funciones bien diferenciadas, en primer lugar encontramos los canales semicirculares, órgano sensor del equilibrio, y la segunda parte es la cóclea o caracol (llamada así por su forma en espiral), que transforma las vibraciones mecánicas provenientes del estribo en señales neurológicas. Es considerado por muchos uno de los procesos más complejos del cuerpo humano, este órgano es capaz de realizar un "análisis en frecuencia" biológicamente, esto sucede de la siguiente manera:

la cóclea posee una membrana que la divide aproximadamente por la mitad, esta membrana posee distintas frecuencias de resonancia a lo largo de su longitud debido al aumento progresivo de su elasticidad y la disminución de su ancho. Si esta membrana es excitada con diferentes tonos puros de diversas frecuencias, se generarán ondas viajeras cuyos máximos estarán localizados en diferentes posiciones. Esta teoría de la localización espacial de las frecuencias en la cóclea fue propuesta por Von Helmholtz y se acercaba bastante a lo que hoy en día sabemos del oído interno.

El último paso de este sistema auditivo es la generación de la señal neurológica que a continuación interpreta el cerebro. Esta transformación es producida por las células ciliadas situadas

sobre la membrana basilar y en contacto directo con la membrana tectoria, el movimiento relativo entre las dos membranas genera un potencial eléctrico generando una señal nerviosa que el cerebro recibe.

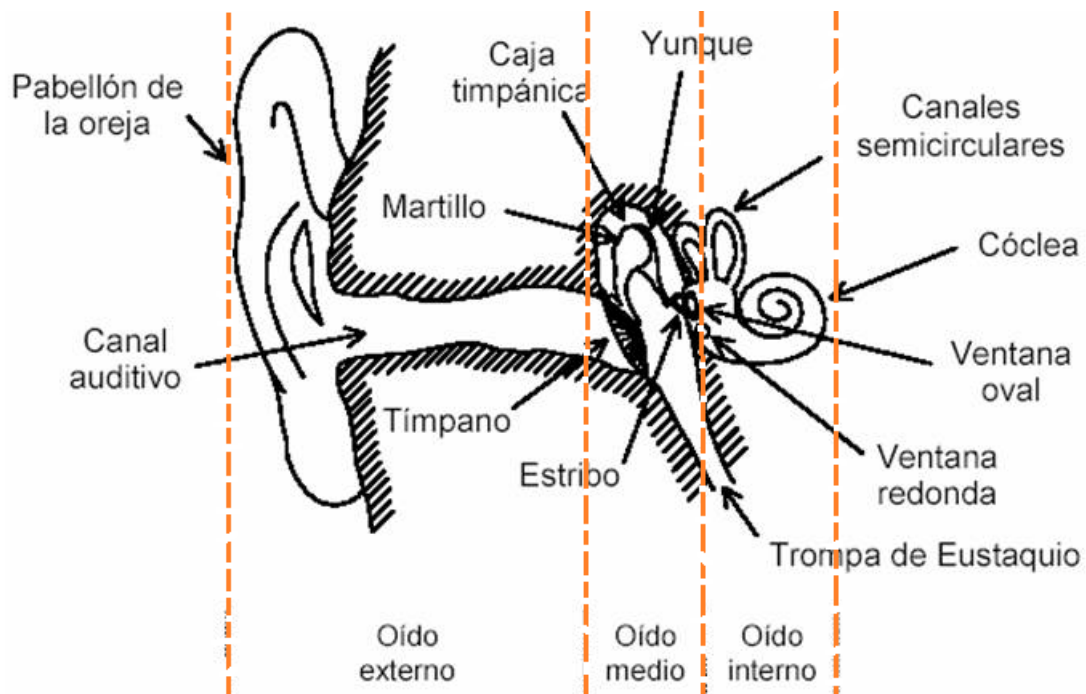


Figura 2.16: Partes del oído humano

■ Sensación sonora

La sensación sonora se mide en fonos, una unidad empleada para indicar la sonoridad.

Los valores mínimos perceptibles por el oído para diferentes frecuencias constituyen una curva llamada umbral de audición, y los valores máximos se sitúan a su vez otra curva superior denominada umbral del dolor, pues los valores percibidos superiores a esa curva producen en el humano una sensación dolorosa.

El oído no percibe igual tonos de la misma intensidad pero de distinta frecuencia, estudiando y midiendo esta fenomenología se ha llegado al establecimiento de curvas de igual sonoridad.

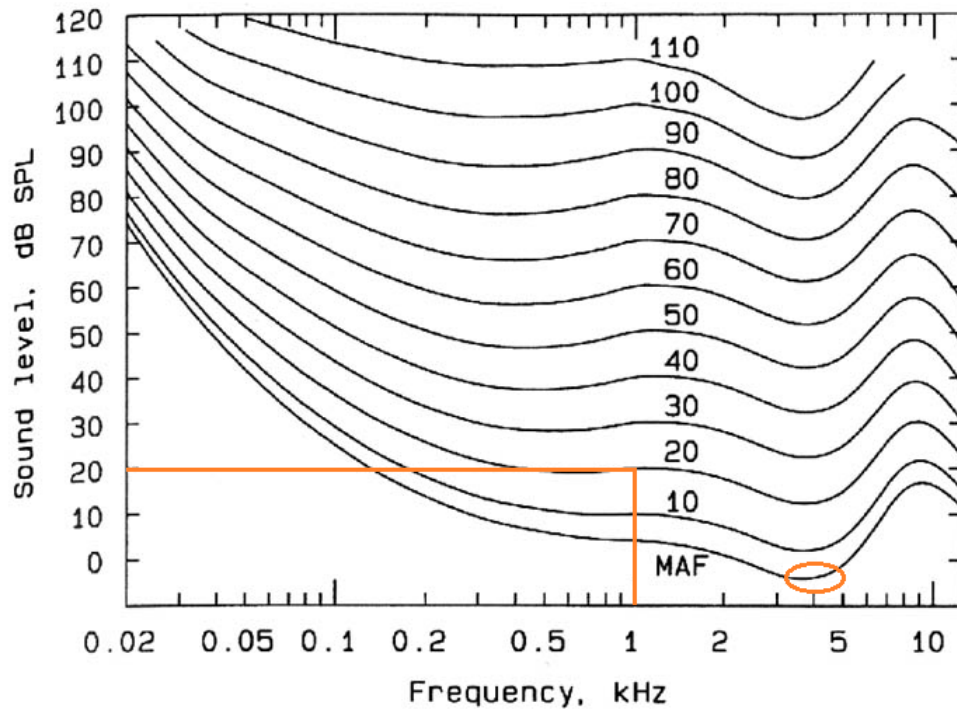


Figura 2.17: Curvas de igual sonoridad

Se puede apreciar en el anterior gráfico que alrededor de 3200 Hz el oído posee la máxima sensibilidad sonora. También cabe destacar que a la frecuencia de 1000 Hz el nivel de sensación sonora coincide con el nivel de intensidad sonora en decibelios.

Esta desigual respuesta del oído ha llevado a realizar ciertas correcciones en los aparatos de medida sonora (Sonómetros) dadas las diferencias entre el sistema del aparato comparado con el oído, se establecen por tanto las ponderaciones en medidas del ruido A, B y C, dependiendo el uso de una u otra del nivel de presión recibido. A para niveles bajos, B para niveles intermedios y C para niveles altos.

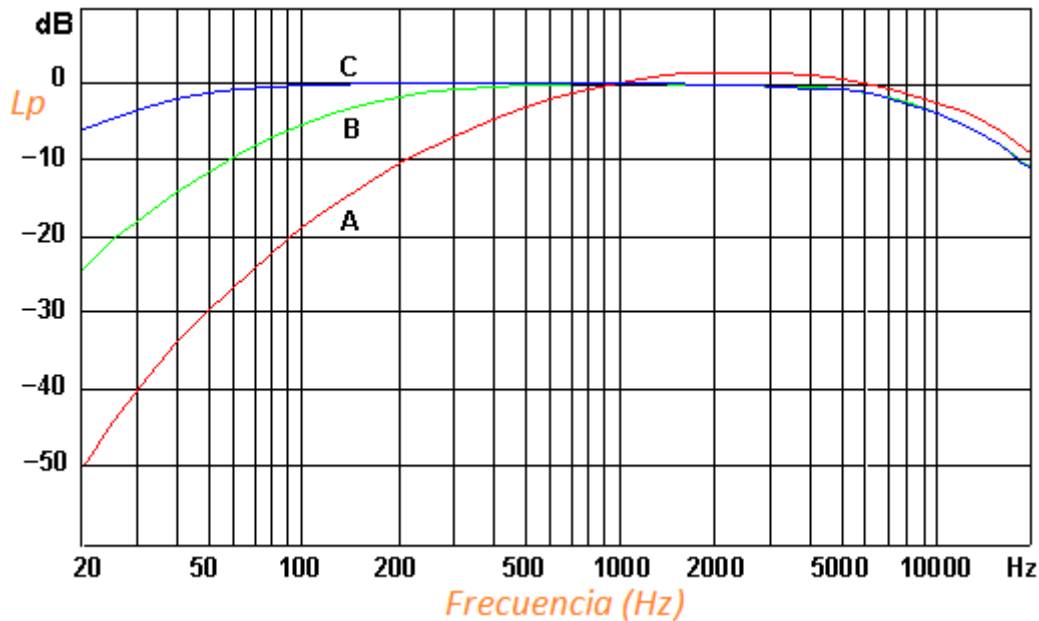


Figura 2.18: Curvas de ponderación A,B y C

2.3. Captación y transmisión eléctrica del sonido: Transductores.

Un transductor es cualquier dispositivo que convierte una forma de energía en otra, en este caso un transductor electroacústico convierte energía eléctrica en acústica y viceversa.

Existen dos grandes tipos de transductores electro-acústicos, emisores (altavoces) donde la cadena de transformación es electro-mecánico-acústica. Y los inversos o receptores (micrófonos) con transformación acústico-mecánico-eléctrica, los últimos serán objeto de estudio en esta memoria debido a que poseen mayor relevancia en el contexto del proyecto. (el violín electroacústico construido posee un transductor receptor piezoeléctrico).

Los transductores se pueden clasificar también según las bases físicas de su funcionamiento, podemos distinguir 4 sub-clases de transductores:

2.3.1. Micrófonos dinámicos

Se basan en la vibración de una bobina, por la cual circula una corriente eléctrica, esta se encuentra alojada en el seno de un campo magnético.

En el caso de los micrófonos dinámicos, la vibración producida por las ondas sonoras es recibida por el diafragma, una membrana muy sensible y de dimensiones reducidas, adherida a la suspensión del diafragma y con carácter móvil se encuentra la bobina, que vibra en consonancia con la membrana, y lo hace además bajo el efecto de un campo magnético creado a su alrededor por un imán, finalmente debido al efecto auto-inductivo, se genera en el conductor

de la bobina una corriente alterna, proporcional al movimiento del sistema diafragma-bobina, que a su vez es transmitida por un cable, esta señal es, al fin y al cabo, una vibración eléctrica, traducida de una vibración sonora, es decir, ha ocurrido la transducción acústico-eléctrica

Este fenómeno suele ser reversible, para reproducir el sonido captado en un altavoz dinámico se produce el proceso inverso.

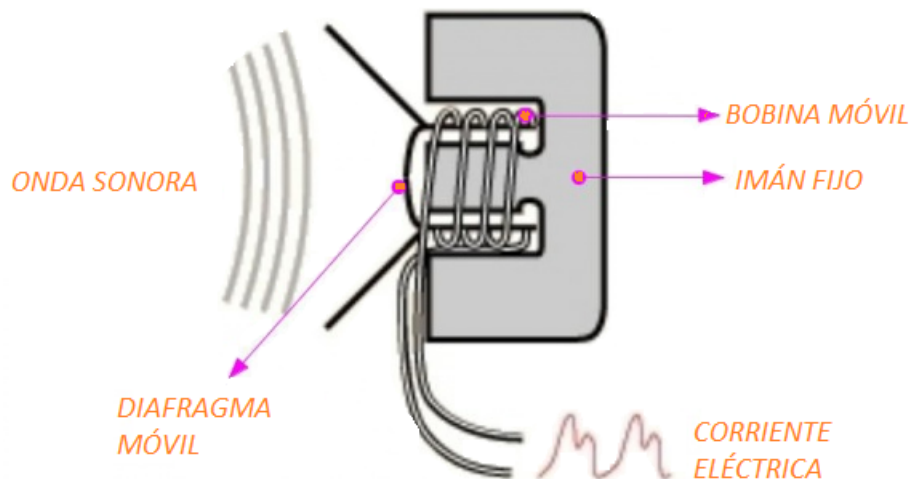


Figura 2.19: Esquema micrófono dinámico

Estos micrófonos son muy robustos y suelen utilizarse para amplificar la voz en radiodifusión, estudio y grabaciones domésticas.

2.3.2. Micrófonos electrostáticos

El elemento más distintivo de este tipo de transductores es el condensador utilizado, formado por dos placas paralelas separadas por un material dieléctrico, una de estas placas es móvil y sensible a las ondas sonoras incidentes.

Si se aplica una fuerza sobre la placa móvil, esta se desplazará una distancia x con respecto a la distancia de reposo y tendrá lugar una variación de la carga q provocando así una corriente eléctrica proporcional al movimiento de la placa móvil.

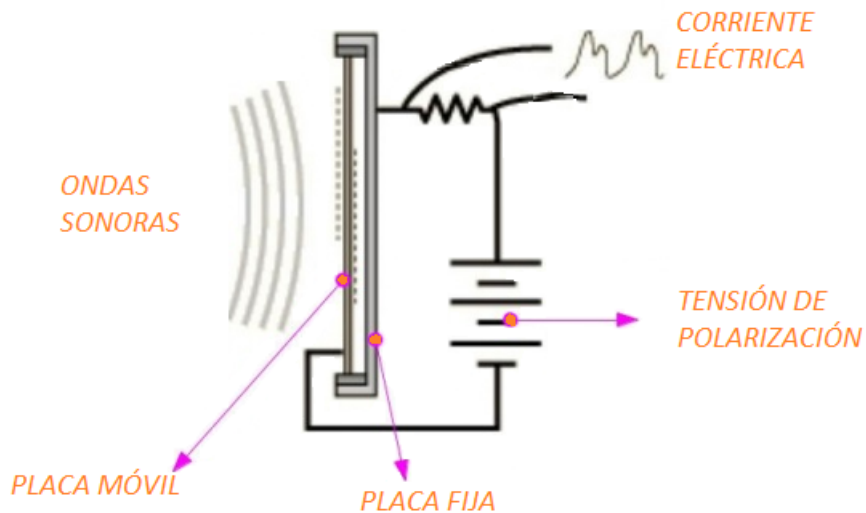


Figura 2.20: Esquema micrófono electrostático

Para que el proceso de transducción sea lineal, los micrófonos de condensador o electrostáticos, necesitan una tensión de polarización elevada, que obtienen del dispositivo al que están conectados.

Poseen un rango frecuencial de respuesta plana mucho mayor que los dinámicos. Se trata de micrófonos delicados, utilizados en calibración, instrumentación y grabación de alta calidad.

2.3.3. Micrófonos piezoeléctricos

Dispositivos que se basan en el fenómeno de la piezoelectricidad. Al comprimir ciertos cristales se provoca una polarización eléctrica en las caras normales a la dirección del esfuerzo. Si en vez de comprimir se expande, las cargas inducidas invierten su signo.

La aplicación de estos transductores se tiene en la emisión y recepción de ultrasonidos en investigación e industria, así como en micrófonos de contacto directo con superficies vibrantes. El transductor instalado en el violín electroacústico pertenece a esta clase de micrófonos piezoeléctricos de contacto.

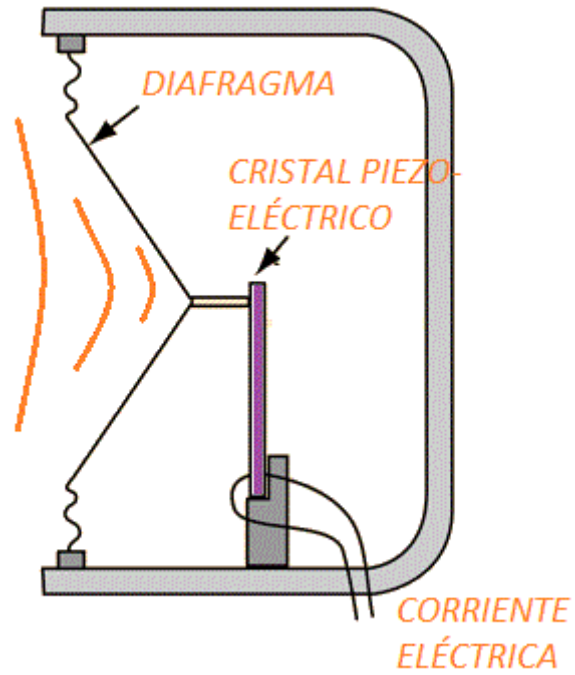


Figura 2.21: Esquema micrófono piezoeléctrico.

Los cristales de cuarzo, la sal de rochela, el fosfato ácido de amonio y el titanato de bario son ejemplos de materiales con propiedades piezoeléctricas.



Figura 2.22: Piezoeléctrico instalado en el violín.

2.3.4. Transmisión eléctrica del sonido: Señal de audio

El audio es una señal analógica eléctricamente exacta a una señal sonora; normalmente está acotada al rango de frecuencias audibles, una señal de audio se puede caracterizar por su dinámica (valor de pico, rango dinámico, potencia, relación señal-ruido) o por su composición espectral (ancho de banda, frecuencia fundamental, armónicos, distorsión armónica, etc.).

La señal de audio puede viajar a través de un cable de dos formas:

- Señal no balanceada

A través de dos cables conductores se envían la señal y referencia neutra respectivamente, es la manera más sencilla de transmitir audio por cable. Como es lógico los conectores no balanceados tienen dos pines (phono y cinch) llamados habitualmente, un ejemplo de estos conductores de audio no balanceados es el clásico *RCA* utilizado comúnmente en el ámbito doméstico para conexiones de audio. Y el conector de 1/4" llamado también mini-jack, de uso habitual en dispositivos de sonido tales como, reproductores mp3, teléfonos móviles, altavoces *plugandplay*, tarjetas de sonido, etc.



Figura 2.23: Cable de audio tipo RCA y conector de 1/4 de pulgada (mini-jack)

Esta configuración sencilla es sensible al ruido producido por interferencias electro-magnéticas que superen el apantallamiento del cable, sobre todo en cables largos.

- Señal balanceada

Esta variante de transmisión de audio es más compleja, se realiza a través de 3 conductores paralelos, uno empleado para la referencia neutra y los otros dos llevan la señal copiada y desfasada entre sí 180 grados, o con polarización inversa, esta es una forma muy efectiva de neutralizar el ruido añadido por interferencias.

A la salida del dispositivo a interconectar, se desfasa una copia de la señal, y se transmite a la vez que la original por un segundo conductor con las mismas características, en caso de ruido por interferencias, se ven afectadas las dos señales, desfasadas entre sí, y finalmente en la

entrada del dispositivo de destino la señal se vuelve a polarizar, quedando entonces el ruido desfasado, seguidamente se suman ambas señales, los valores de ruido desfasados 180 grados se anulan en dicha suma, y la señal se auto-amplifica y queda protegida ante interferencias, tenemos pues, una señal balanceada.

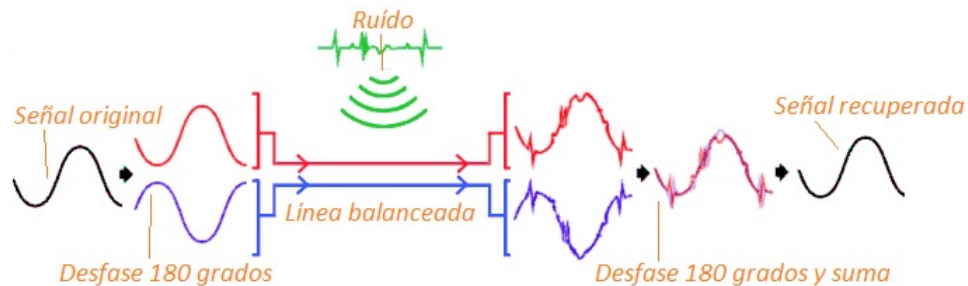


Figura 2.24: Gráfico: Señal de audio balanceada

Para portar este tipo de señal se necesitan cables de tres pines y tres conductores como por ejemplo el conector XLR o el conector de 1/4 de pulgada o Mini-jack (estéreo). Habitualmente los tres terminales se nombran (+) Hot (caliente), (-) Cold (frío), y malla o masa (Sleeve o ground).

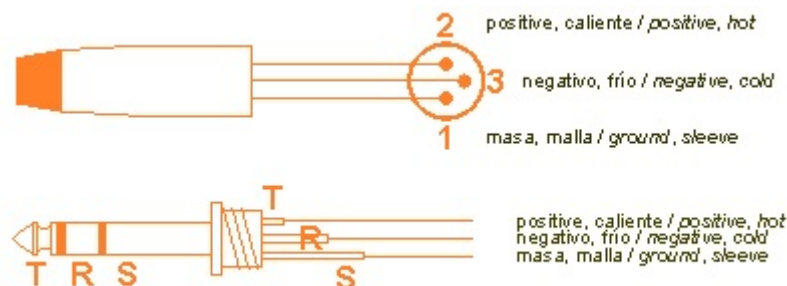


Figura 2.25: Cables de audio balanceado

2.4. Digitalización del sonido

Hasta ahora hemos hablado del audio analógico, que en el sistema del proyecto se genera en el micrófono piezoeléctrico de contacto, esta señal de audio no balanceada se transmite a través de un cable monofónico hasta la entrada de línea de la tarjeta de sonido de una computadora portátil por el puerto de 1/4 pulgada o mini-jack. Esta tarjeta de sonido digitaliza los parámetros eléctricos provenientes del cable, el resto de la trayectoria que sigue señal de audio a partir de este momento, se efectúa en un contexto digital (procesador).

Frente a los inconvenientes del audio analógico como: La distorsión armónica en el caso de que el proceso de generación y/o amplificación no fuese lineal, o el problema de la degradación de la señal por la suma de ruido, siendo esta degradación inseparable de la señal están las ventajas del audio digital, que soporta procesos ilimitados sin degradación, copias sin pérdidas, corrección de errores, posibilidad de compresión, facilidad de trabajo, entre otras.

El proceso de digitalización implica el muestreo y la cuantificación posterior, dos sub-procesos de notable importancia en el contexto del proyecto, el resultado de esta conversión analógica-digital (A-D) es una señal digital binaria, es decir una secuencia de unos y ceros que codifican los valores muestreados de la señal original.

2.4.1. Muestreo (S)

El muestreo de una señal continua consiste en tomar muestras de dicha señal en ciertos instantes de tiempo, se trata de una discretización temporal, si se toman muestras equidistantes en el tiempo resulta un muestreo periódico, siendo T_s el periodo de muestreo.

$$x[n] = x(t)_{t=nT_s} = x(nT_s)$$

- Teorema de muestreo de Nyquist

Dada una señal $x(t)$ limitada en banda, si se muestrea cada T_s segundos, es posible recuperarla completa y exacta a partir de sus muestras siempre que se cumpla:

$$T_s \leq \frac{1}{2f_{max}}$$

Siendo f_{max} la frecuencia máxima de la señal $x(t)$. Denominamos frecuencia de muestreo f_s a la frecuencia con que se van obteniendo muestras de la señal original.

A la supuesta mínima frecuencia de muestreo necesaria para recuperar la señal analógica exacta se le llama frecuencia de Nyquist.

$$f_{Nyquist} = 2f_{max}$$

En el muestreo, la señal analógica $x(t)$ es multiplicada por un tren de deltas de Dirac $s_\delta(t)$ cuyas deltas están situadas en múltiplos de T_s , esta señal $s_\delta(t)$ recibe el nombre de señal muestreadora original.

$$s_\delta(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

El resultado de este producto es otro tren de deltas de dirac con el mismo periodo que la señal muestreadora cuyos valores de amplitud o pesos son los de las muestras $x(nT_s)$ que han sido capturadas.

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \cdot \delta(t - nT_s)$$

Quedando así, la señal discretizada en el tiempo.

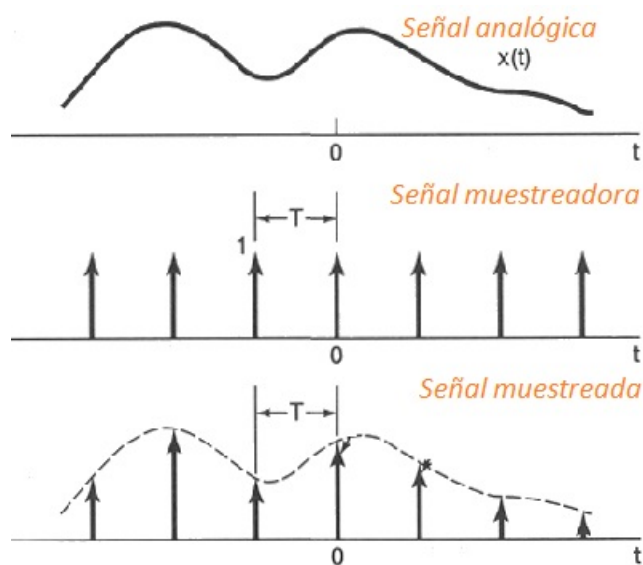


Figura 2.26: Muestreo de una señal analógica

2.4.2. Cuantificación (Q)

El proceso de cuantificación Q consiste en asignar valores digitales a las muestras obtenidas en función de su amplitud original, se trata de una discretización de la amplitud, quedando así una señal discreta en tiempo y amplitud, codificada binariamente, una señal de audio digital.

la unidad mínima digital es el bit, cada grupo de b bits representa una muestra de la señal analógica, podremos representar un total de $L = 2^b$ niveles de señal diferentes, por ello, tras muestrear la señal analógica, se aproxima este valor muestreado al más cercano de los niveles de señal existentes. Esta labor la realiza el cuantificador.

Básicamente la función del cuantificador es efectuar unas operaciones de redondeo o truncamiento, este proceso no lineal viene determinado por su función característica de cuantificación $Q(x)$

Cuantificación "Midthread"

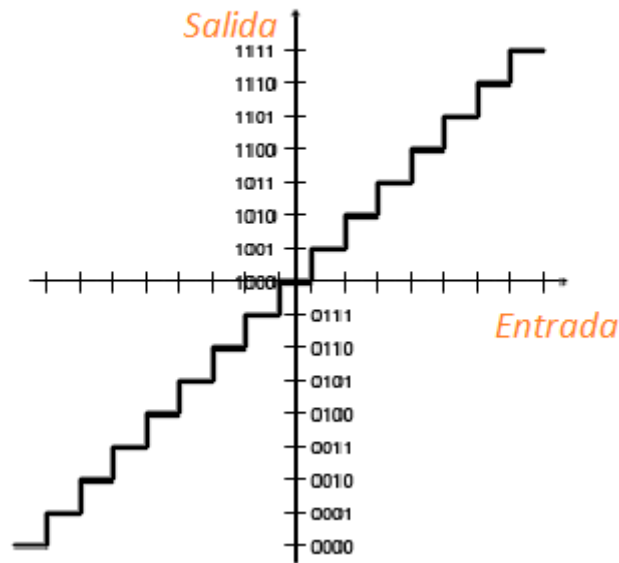


Figura 2.27: función de cuantificación $Q(x)$

Este redondeo o truncamiento implica un error, llamado error de cuantificación e_q cuyo valor para cada muestra será:

$$e_q = x_q - x = Q(x) - x$$

Existen unos límites por encima y por debajo de los cuales, no existe nivel digital, llamados x_{min} y x_{max} cuando la señal original se encuentra fuera de los límites establecidos, esta señal se encuentra en saturación o sobrecarga, en cambio si los valores de la señal se encuentran dentro del intervalo mencionado, se encuentra en zona granular.

Podemos definir entonces el margen dinámico del cuantificador.

$$2X_m = x_{max} - x_{min}$$

la diferencia entre valores contiguos de cuantificación, recibe el nombre de escalón de cuantificación Δ

En un cuantificador se tiene que cumplir:

$$L \leq 2^b$$

$$\Delta = \frac{x_{max} - x_{min}}{L}$$

$$\Delta = \frac{x_{qmax} - x_{qmin}}{L - 1}$$

En el caso de que el tamaño del escalón de cuantificación sea siempre constante, estaremos ante un cuantificador uniforme, si este escalón varía de acuerdo a una ecuación determinada, se trata de un cuantificador no uniforme.

En definitiva, el proceso de digitalización de una señal analógica se realiza por medio de un conversor A-D, cuyos dos procesos generales son el muestreo y la cuantificación.

2.4.3. Tarjetas de sonido

Una tarjeta de sonido es un dispositivo en forma de tarjeta de expansión de computadora, puede ser interna o externa, y su función principal es la gestión de las entradas y salidas de las diferentes señales de audio regulada y controlada por un programa informático.

La parte más importante de una tarjeta de sonido y lo que sin duda condicionará su calidad es el conversor analógico-digital (AD) y el digital-analógico(DA). Dicho elemento, como hemos mencionado anteriormente, tiene como misión transformar la señal analógica que entra en la tarjeta en digital, para poder ser procesada y tratada en el ordenador, y a su vez, devolverla transformándola de nuevo en analógica para poder ser reproducida.

Como breve contextualización histórica, cabe mencionar que la primera tarjeta de sonido compatible con computadoras se inventó en 1987. Esta tarjeta de sonido recibe su nombre de la empresa Adlib, fundada por Martin Prevel en Quebec (Canadá). Tras la creación de este nuevo producto, Martin Prevel paso más de un año tratando de promocionarlo entre los desarrolladores de IBM, sin embargo, su inexperiencia en el campo de la industria fue la causa de su frustrado intento de fomentar su producto. Finalmente estableció contacto con el presidente de Top Star (una empresa de Nueva Jersey), que tras quedar satisfecho después de una demostración del producto participó en su divulgación y promoción.

La AdLib usaba el chip de sonido Yamaha YM3812, de la empresa Yamaha, que producía sonido mediante la Síntesis FM. Con la tarjeta AdLib, el software de la PC podía generar música multitímica y efectos de sonido, aunque la calidad acústica era distintivamente sintética.

Tras la aparición de la tarjeta AdLib, vino la tarjeta S de la casa Creative Labs, totalmente compatible con la anterior, pero que además de la síntesis FM, incorporaba la posibilidad de grabar y reproducir audio digital. Esto permitía usar en los juegos sonidos reales (voces, ruidos, etc). Creative se hizo con el mercado, consiguiendo desde entonces ser el estándar de hecho.

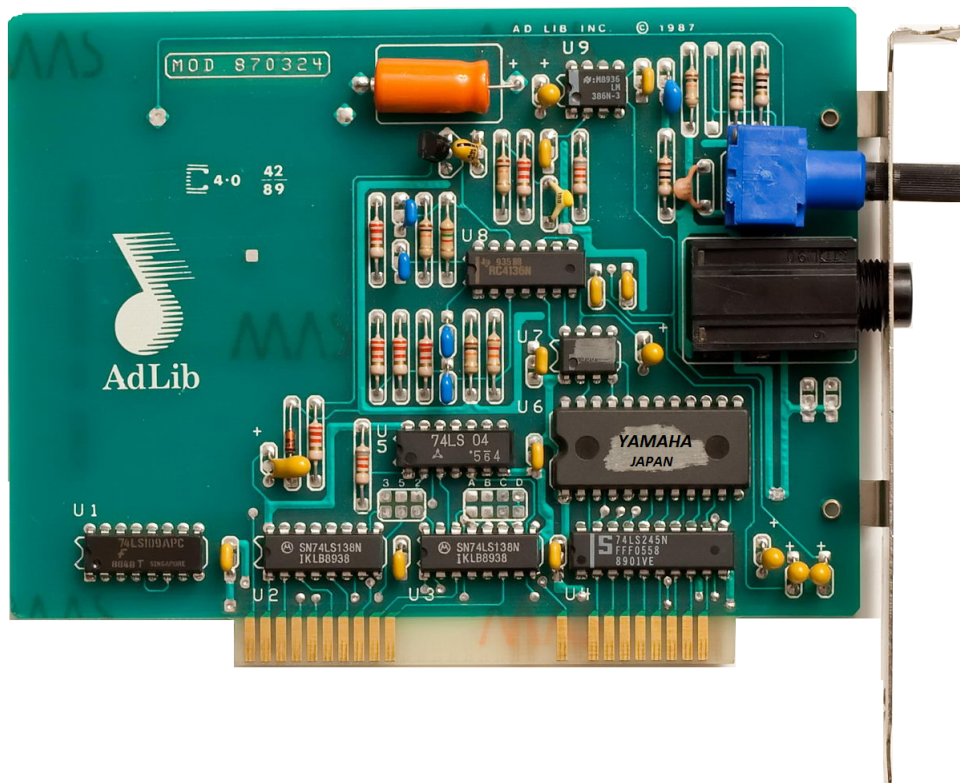


Figura 2.28: Primera tarjeta de sonido alojada en un PC

Las tarjetas de sonido actuales permiten muestrear las señales de audio con varias frecuencias de muestreo, siendo las más utilizadas 44.100 Hz, Calidad de compact disc y 80.000 Hz calidad de dvd, el usuario puede variar también la profundidad de bits de la cuantificación, comúnmente se utilizan 16,24 ó 32 bits.

Una vez digitalizada, a la señal de audio se le pueden aplicar multitud de procesos utilizando programas editores de audio, programación sonora, reproductores, y una gran variedad de aplicaciones informáticas enfocadas al tratado de señales de audio digital.

Esta conversión también nos proporciona la capacidad de analizar la señal de audio tanto en el dominio del tiempo como en el dominio de la frecuencia.

2.5. Análisis de una señal de audio digital

El análisis o caracterización de una señal de audio digital se realiza tanto en el dominio del tiempo como en el dominio transformado de la frecuencia.

2.5.1. Análisis en el dominio temporal

A continuación se detallarán los parámetros más importantes que caracterizan una señal (de audio en nuestro caso) en el dominio temporal:

- Valor de pico x_p^+ (positiva) y x_p^- (negativa)
- Valor de pico a pico X_{pp}

$$X_{pp} = x_p^+ - x_p^-$$

- Media $\langle x(t) \rangle$

$$\langle x(t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cdot dt$$

$\langle x(t) \rangle^2$ Representa la potencia de la componente continua de la señal.

- Valor cuadrático medio $\langle x^2(t) \rangle$

$$\langle x^2(t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x^2(t) \cdot dt$$

$\langle x^2(t) \rangle$ Representa la potencia total de la señal.

- Valor eficaz X_{eff}

$$X_{eff}^2 = \langle x^2(t) \rangle - \langle x(t) \rangle^2$$

X_{eff}^2 Representa la potencia de las componentes alternas de la señal.

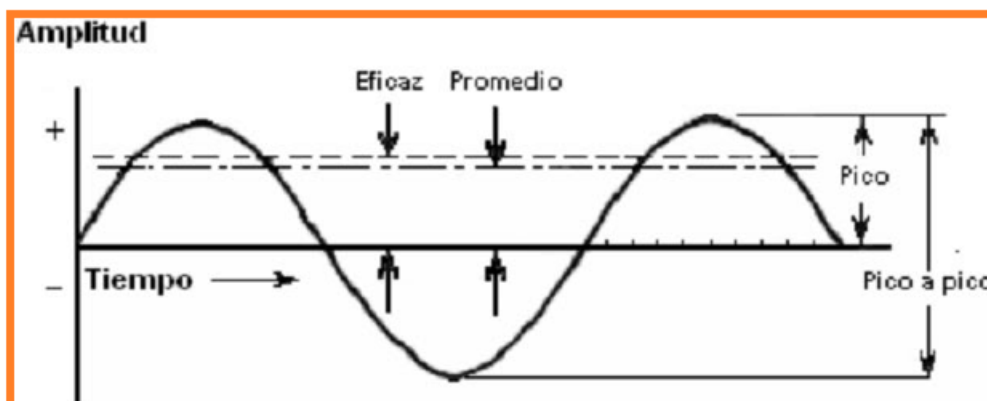


Figura 2.29: Representación temporal de una señal periódica

2.5.2. Análisis en el dominio frecuencial

En el dominio de la frecuencia representamos las señales mediante una combinación lineal de exponenciales complejas, emplearemos la transformada (discreta en el caso del audio digital) de Fourier.

La transformada discreta de Fourier (DFT) transforma una función matemática en otra, obteniendo una representación en el dominio de la frecuencia, este proceso requiere que la señal de entrada en el dominio del tiempo sea discreta y de duración limitada, por eso para analizar en el dominio de la frecuencia una señal de audio digital, es imprescindible acotar el número de muestras que se van a analizar por medio de un enventanado, en el caso del proyecto actual este análisis se efectúa prácticamente en tiempo real, proporcionando el parámetro de la frecuencia fundamental mediante un proceso de pitchtracking que analizaremos posteriormente.

Ecuación de síntesis para señales discretas periódicas en el tiempo:

$$x(n) = \sum_{k=0}^{N-1} c_k \cdot e^{j2\pi kn/N}$$

Ecuación de análisis para señales discretas periódicas en el tiempo:

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi kn/N}$$

Una representación en el dominio frecuencial o representación espectral muestra el contenido frecuencial de un sonido. Las componentes de frecuencias individuales del espectro pueden denominarse armónicos o parciales. Las frecuencias armónicas son múltiplos enteros simples de la frecuencia fundamental.

Cualquier frecuencia puede denominarse parcial, sea o no múltiplo de la frecuencia fundamental. De hecho, muchos sonidos no tienen una fundamental clara (percusión, ruido, sonidos inarmónicos).

El contenido frecuencial de un sonido puede mostrarse de diversas maneras. Una forma estándar es la de dibujar en un sistema de ejes cartesianos cada parcial como una línea sobre el eje x. (a su respectiva frecuencia) La altura de cada línea correspondería a la amplitud de cada componente frecuencial.

Una señal sinusoidal (un tono puro) viene representada por una sola componente frecuencial.

Para adaptar el análisis de Fourier al dominio de las señales muestreadas, de duración finita y variantes respecto al tiempo, se define la transformada de Fourier localizada o a corto plazo (Short-Time Fourier Transform o STFT). En este caso, se divide la señal en pequeñas porciones o segmentos, que se denominan tramas de análisis (analysis frames). El contenido frecuencial o espectro se calcula en cada una de las tramas utilizando la transformada de Fourier.

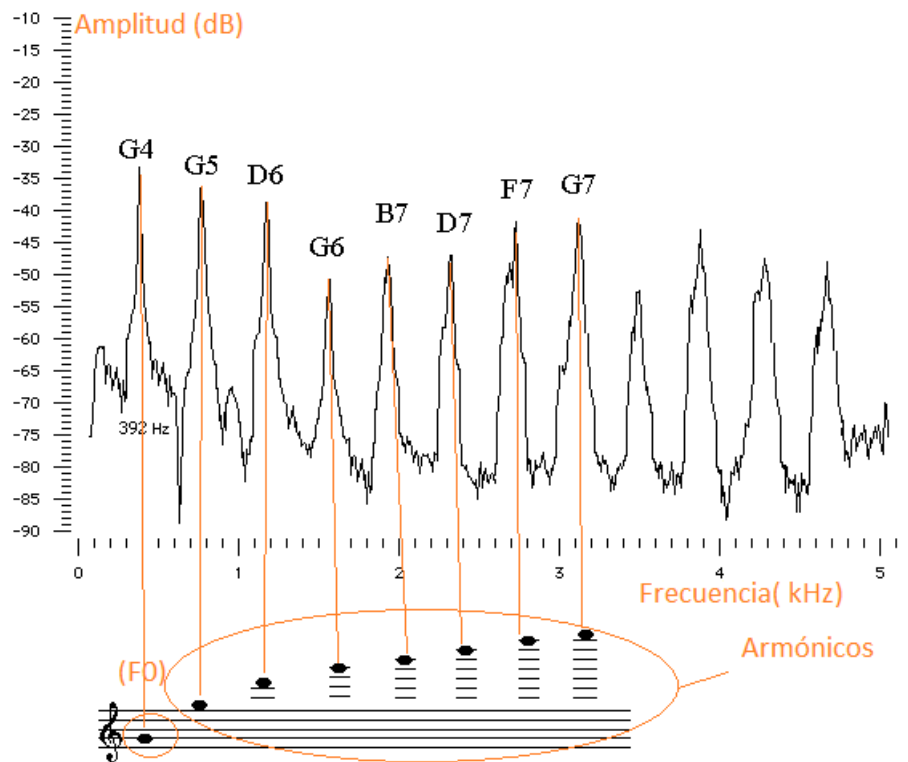


Figura 2.30: Representación frecuencial de un sol4 reproducido por un violín

Como se puede apreciar, además del máximo esperado en la frecuencia fundamental $F_0 = 392\text{Hz}$ se observan varios máximos locales en frecuencias múltiplos de F_0 , se trata de los armónicos producidos por los diferentes elementos vibrantes del violín, estos armónicos definen el timbre del sonido, caracterizan el sonido de violín.

Existen dos categorías de análisis espectral, el análisis armónico que se realiza mediante la mencionada Transformada de Fourier, la cual calcula las amplitudes y las frecuencias de las componentes espectrales, y el análisis de formantes, que estima la forma global de la envolvente de amplitudes en el espectro, un análisis predictivo lineal por ejemplo.

Como hemos dicho, la STFT impone una secuencia de ventanas temporales de la señal de entrada, es decir, divide la señal en fragmentos cortos (short time) delimitados en el tiempo por una función ventana. Una ventana no es nada más que un tipo específico de envolvente que se aplica para un análisis espectral. La duración de la ventana en este proyecto es de aproximadamente 0.01 segundos. (512 muestras)

A través del análisis espectral individual de cada segmento enventanado, se puede obtener una secuencia de medidas (de espectros) que constituyen un espectro variable a lo largo del tiempo llamado espectrograma o sonograma.

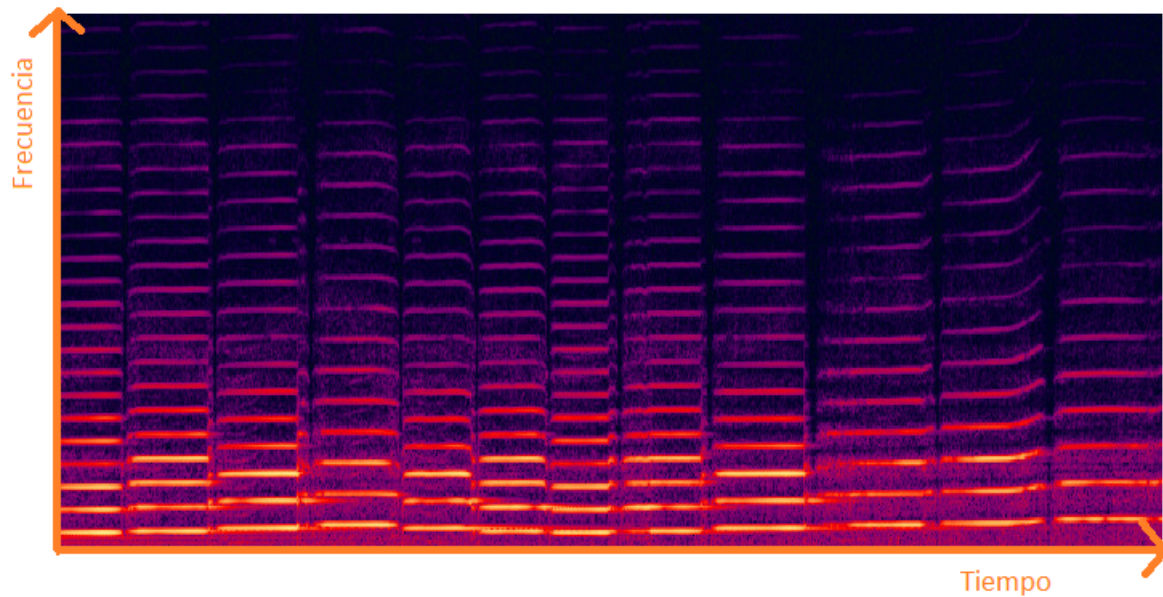


Figura 2.31: Espectrograma o sonograma

Analizando el espectro de una porción de señal de audio, podremos determinar sus frecuencias fundamentales, el peso de sus armónicos, su armonicidad, su ancho de banda, así como el efecto de filtros aplicados anteriormente a la transformación.

2.6. Caracterización de una señal de violín eléctrico

Se pretende estudiar las características de la señal de audio producida por un violín electroacústico, para poder posteriormente procesarla y extraer de ella los parámetros más interesantes en cuanto al objetivo del proyecto, que serán , amplitud y frecuencia fundamental (FO).

2.6.1. Registro

Tanto el violín eléctrico como el acústico tienen normalmente 4 cuerdas que, vibrando libremente (al aire) producen las siguientes frecuencias. La distancia en frecuencia entre los sonidos de cada una de las cuerdas es de una quinta justa, la relación de quinta justa equivale como ya mencionamos al comienzo de este capítulo, a $3/2$ de la frecuencia base.

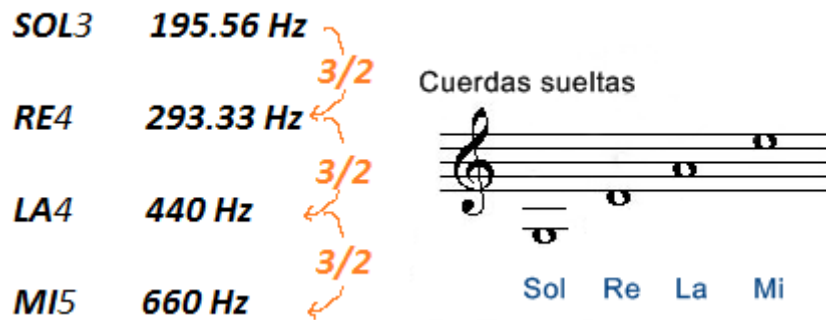


Figura 2.32: Frecuencias de las cuerdas de un violín

Por lo tanto se establece la frecuencia más baja alrededor de 195 Hz, el límite superior es más difícil de establecer, puesto que sobre la cuerda Mi5, se pueden colocar los dedos acortando la cuerda y en consecuencia, aumentando la frecuencia de vibración, pero llega un punto en el cual el sonido, deja de ser suficientemente armónico para ser utilizado como nota musical, ocurre alrededor de 3500 Hz.

2.6.2. Amplitud

la sensación sonora posee un rango muy grande en cuanto a la percepción de distintas variaciones de presión, desde $20\mu Pa$ hasta $20Pa$, por eso, a la hora de medir la intensidad de un sonido conviene definir un intervalo logarítmico en vez de lineal, se definen las siguientes magnitudes.

Nivel de potencia sonora $L_W(dB) = 10 \log_{10} \frac{W}{W_0}$ siendo $W_0 = 10^{-12}W$ la mínima potencia que puede percibir el oído humano (potencia umbral)

Nivel de intensidad sonora $L_I(dB) = 10 \log_{10} \frac{I}{I_0}$ siendo $I_0 = 10^{-12}W \cdot m^{-2}$ la intensidad umbral.

Nivel de presión sonora $L_p(dB) = 20 \log_{10} \frac{p}{p_0}$ siendo la presión umbral $p_0 = 2 \cdot 10^5 Pa$

Un violín acústico puede producir un nivel de presión sonora L_p máximo de unos 65 dB Aproximadamente, pero en el caso de los violines eléctricos, se puede amplificar o atenuar este valor dependiendo de la ganancia que le apliquemos, por lo tanto, la amplitud depende de la microfónica empleada, el modelo de violín eléctrico, y sobre todo de la ganancia aplicada sobre el canal por el cual circula la señal.

En CSound, los valores de amplitud se miden siempre relativos a $0dBfs$, que representa el valor de pico justo antes de saturar.

Un violín eléctrico de la marca y modelo: YAMAHA Silent violín, produce una amplitud sonora (una vez digitalizado el audio y medido a la entrada del programa C-SOUND en dB relativos al intervalo full-scale (0dBfs)) que puede variar entre -35dB y 0 dB.

Sin embargo, el violín construido, debido a la baja sensibilidad del micrófono piezoeléctrico, produce en el mismo caso una amplitud de entre -35dB y -20 dB, por lo tanto, en el caso de utilizar el violín de construcción propia, se necesitará una pre-amplificación para poder trabajar en un rango superior de valores de amplitud, esta amplificación viene implementada en el algoritmo diseñado.

2.6.3. Contenido espectral, Timbre

El espectro es un importante caracterizador del timbre, en concreto la posición y valor de la fundamental (normalmente la altura percibida) y sus parciales armónicos.

También las zonas de mayor amplitud del espectro, con su particular ancho de banda (intervalo de frecuencias en que están comprendidas) llamadas formantes, son las principales responsables de la caracterización del timbre.

A diferencia de el estudio de los parciales armónicos, el análisis de las formantes estima globalmente la envolvente de amplitudes en el espectro.

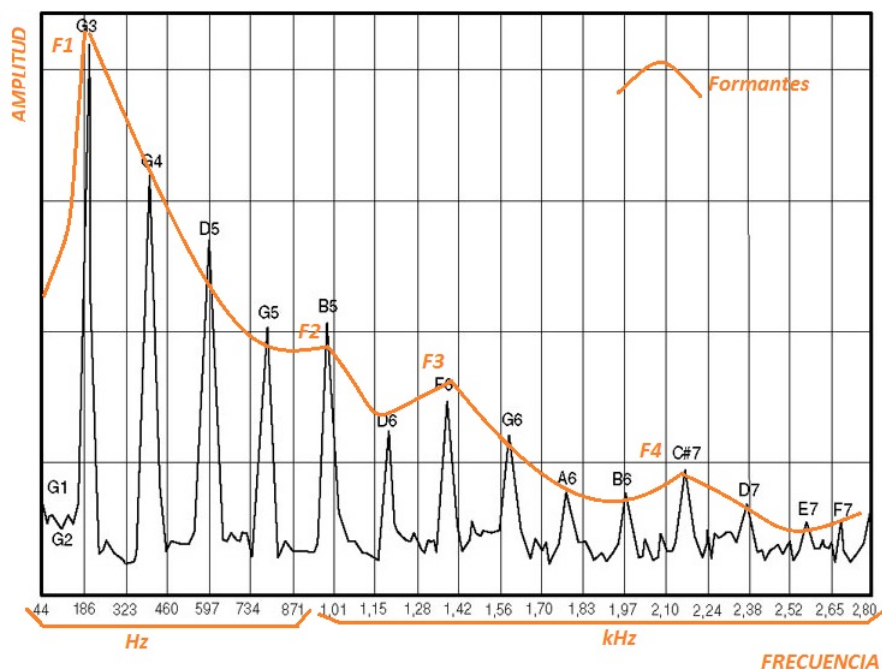


Figura 2.33: Espectro de la cuerda más grave de un violín (Sol_3)

2.7. Técnicas de seguimiento de tono: Pitchtracking

2.7.1. Introducción

La detección de la frecuencia fundamental (F0) ha sido desde hace tiempo un problema estudiado profundamente, considerado prácticamente resuelto para grabaciones monofónicas de voz o instrumentos musicales, al trabajar con audio grabado, se pueden resolver errores de detección después del proceso para lograr una mayor precisión. El proceso se vuelve más complejo si se trata de la detección de tono en tiempo real de una señal, y mucho más aún si se trata de una señal polifónica.

Uno de los grandes problemas añadidos en el pitchtracking es la existencia masiva de armónicos, inherentes a una señal producida por la voz o un instrumento musical acústico, que dificultan el proceso a la hora de discriminar las frecuencias fundamentales, en algunos casos con amplitud igual o incluso superior a la F(0) correspondiente

El proceso de detección de F(0) de nuestro algoritmo, requiere por lo tanto: la capacidad de funcionar en tiempo real (eficiencia computacional), producir la mínima latencia o retardo a la salida, detección en presencia de ruido, y sensibilidad a las variaciones propias de la ejecución musical.

Dado que el algoritmo implementado funciona en tiempo real, únicamente con señales monofónicas, a continuación se detallarán las diferentes técnicas que solucionan la detección de tono de este tipo de señales.

Las técnicas o algoritmos de detección de tono *P.D.A.*, (Pitch Detection Algorithm) se pueden categorizar según su naturaleza, temporal o frecuencial.

2.7.2. Técnicas PDA en el dominio del tiempo

■ Método de Autocorrelación

Este método se basa en la comparación de la forma de onda (máximos) de una señal, con retrasadas versiones de la misma señal, con el fin de determinar su periodo fundamental, Dada una señal temporal $x(t)$ de característica estacionaria (estadísticamente constante) es posible definir una expresión para su autocorrelación $r_x(\tau)$ como función del retardo τ a partir de:

$$r_x(\tau) = \int x(t) \cdot x(t + \tau) dt$$

Dicha función posee un máximo global para $r_x(\tau) = 0$, por tanto si existe otro máximo global diferente al origen, la señal puede llamarse periódica para un retardo o periodo T_0 tal que todos sus máximos se localicen a intervalos discretos espaciados en un factor de nT_0 para todo entero n , donde se cumpla que $r_x(nT_0) = r_x(0)$. Resultando una frecuencia fundamental $f_0 = \frac{1}{T_0}$.

Para funciones no estacionarias, se aproxima la autocorrelación en intervalos cortos de tiempo (segmentación y enventanado), siendo más acertado el resultado a medida que aumenta el número de muestras por ventana.

A pesar de la facilidad en cuanto a la implementación de esta técnica, se descarta por su poca robustez frente al ruido, no obstante una variante del método de autocorrelación llamado AMDF ha sido implementado en nuestro algoritmo.

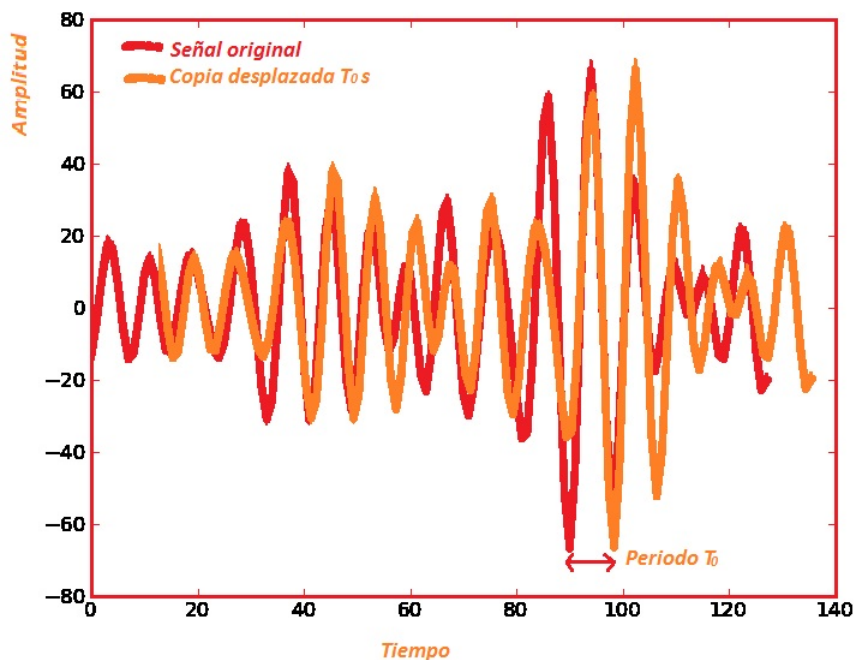


Figura 2.34: Autocorrelación

2.7.3. Técnicas PDA en el dominio de la frecuencia

Estas técnicas utilizan la transformación en frecuencia FFT (Fast Fourier Transform) sobre segmentos cortos de tiempo ajustados con una función ventana de la señal de entrada para enfatizar la información de sus armónicos y realizar posteriormente la construcción de su respectivo histograma (representación espectral).

■ Método HPS (Harmónic Product Spectrum)

El producto de armónicos espectrales HPS asume como frecuencia fundamental de una señal periódica al máximo común divisor de sus frecuencias armónicas, entonces, si se comprime el espectro de la señal por un factor de 2, consecuentemente el segundo armónico del espectro comprimido se encontrará en la posición de la frecuencia fundamental en el espectro original, por lo tanto, si multiplicamos ambos espectros (comprimido y original) se enfatizará el peso de la fundamental, este proceso de diezmo espectral se puede utilizar varias veces, con diversos

factores enteros de compresión, aprovechando así la presencia de los distintos armónicos presentes en el espectro original.

Debido a que los máximos no correlacionados (inarmónicos) del espectro original, tienden a cancelarse en este proceso, el HPS resulta un algoritmo muy robusto frente al ruido.

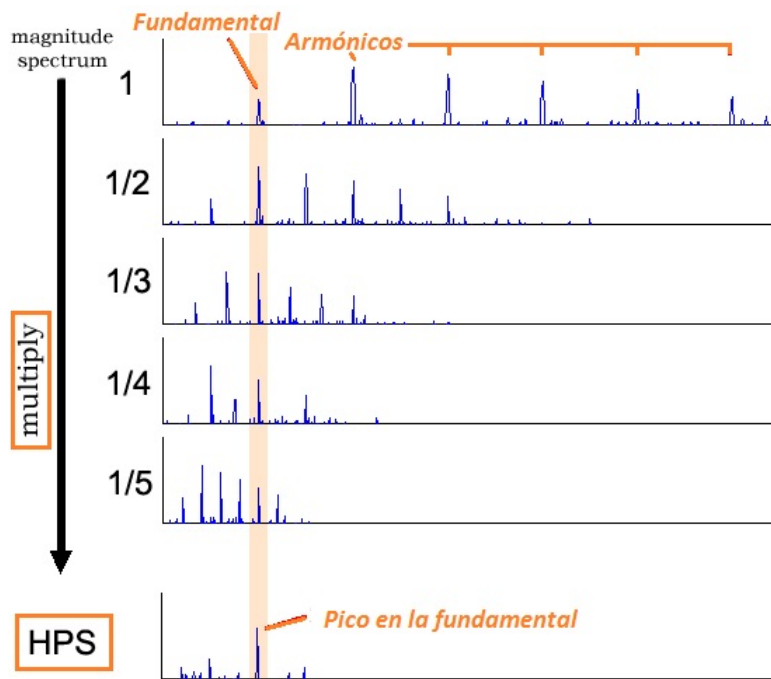


Figura 2.35: HPS

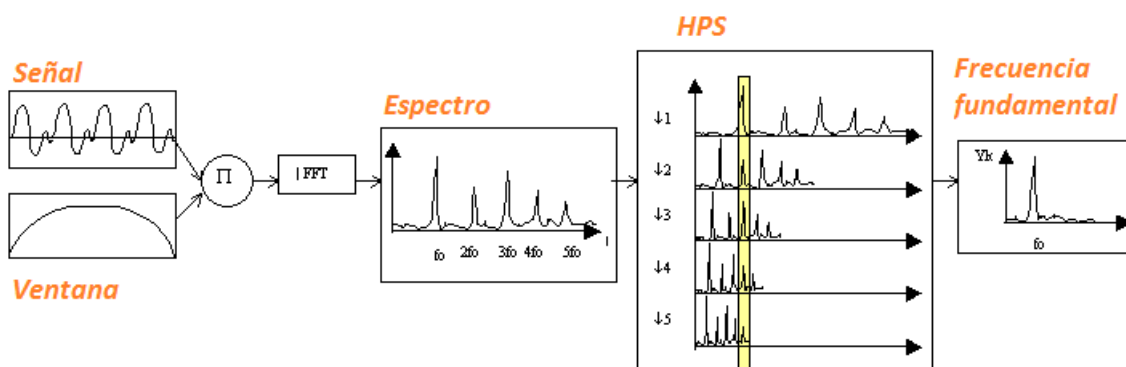


Figura 2.36: Estructura método HPS

- Método ML (Máximo Likelihood)

Una vez obtenido el espectro de la señal a analizar el algoritmo ML busca a través de un conjunto de posibles espectros ideales y elige el que más se parece a la forma del espectro

de la señal de entrada. El espectro ideal se define como un tren de impulsos con frecuencia f convolucionada con el espectro de la señal ventana.

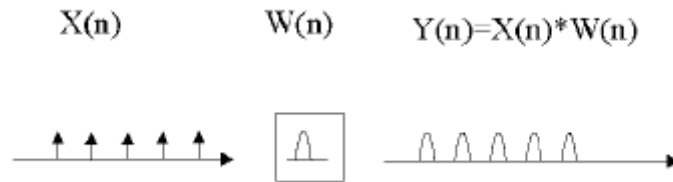


Figura 2.37: Generación del espectro ideal

Este proceso trata de minimizar el error entre el espectro de la señal de entrada y un posible espectro candidato como se ilustra en las siguientes ecuaciones.

$$E(w) = |Y - Y_w|^2 = |Y|^2 + |Y_w|^2 - 2YY_w$$

Donde Y es el espectro de entrada obtenido, y Y_w es un espectro ideal con una frecuencia fundamental situada en w , se debe encontrar el máximo valor del producto de los dos espectros, una vez encontrado el espectro ideal \hat{Y} se demuestra que:

$$\hat{Y} = \min_w \{E(w)\} = \max_w \{YY_w\}$$

Dado que el algoritmo ML no necesita ser espectralmente interpolado como en el caso del HPS, se pueden utilizar tamaños más pequeños de ventana, esto puede acelerar el algoritmo considerablemente. Sin embargo, la eficiencia del algoritmo depende de la resolución en frecuencia que se requiera.

El algoritmo ML funciona bien si la fuente de entrada tiene una afinación fija. Teclados e instrumentos de viento son mucho más apropiados para el uso de esta solución que las cuerdas o la voz, estos últimos instrumentos pueden producir fácilmente errores de detección de tono, sobre todo en presencia de vibrato.

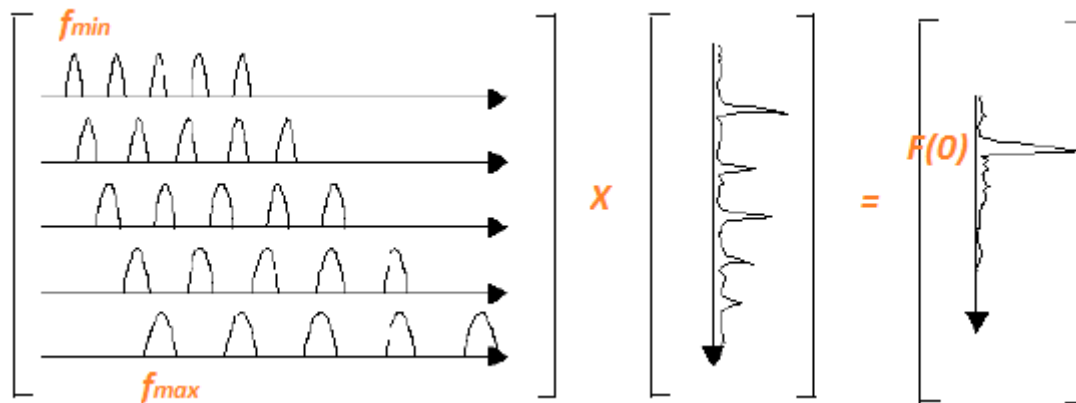


Figura 2.38: Maximum Likelihood

De las 3 técnicas mencionadas, el algoritmo HPS, es el que más se ajusta a los objetivos del proyecto, dada su robustez frente al ruido y su precisión en situaciones críticas como vibrato o transitorios.

Existen técnicas que combinan procesos en los dos dominios, llamados algoritmos híbridos de detección de tono, un ejemplo es la detección vía Wavelets, pero se ha descartado su uso por ser demasiado sensible al ruido, este método se suele utilizar para detección de tono en archivos de audio previamente adecuados y filtrados, por lo tanto no se ha tenido en cuenta siendo fundamental el funcionamiento en tiempo real para este proyecto.

2.8. Síntesis digital del sonido - Técnicas de síntesis

Una vez detectado la altura (pitch) y la amplitud de una porción de la señal mediante un algoritmo PDA y un detector de envolventes (fase de detección), la señal del violín ya analizada deja de ser útil, y el algoritmo comienza una nueva fase de síntesis del sonido cuyo desarrollo se efectuará paralelamente a la fase de detección de forma continua, utilizando los 2 parámetros obtenidos como variables de entrada de los diferentes instrumentos virtuales programados en lenguaje C-SOUND.

2.8.1. Introducción

La síntesis clásica del sonido consiste en la generación por medios electrónicos de sonidos, sin la participación de ninguna fuente acústica externa. A lo largo de la historia, la síntesis del sonido ha sufrido un desarrollo vertiginoso, comenzando por la creación de sistemas que utilizaban bobinas rotantes, generadoras de corrientes alternas por inducción (ej: Telharmonium, Theremin, Hammond), más adelante con la aparición de los sintetizadores electrónicos analógicos basados principalmente en osciladores controlados por tensión (VCO) (ej: MkII, Moog 55),

antecesores a su vez de los sintetizadores de osciladores digitales (DCO) (ej: Yamaha DX7), y por último la nueva generación de sintetizadores digitales software, donde cabe destacar dos grandes familias: Los sintetizadores gráficos y los lenguajes de síntesis.

CSound es uno de los más potentes lenguajes de síntesis, siendo la opción elegida para la implementación de todas las fases del proyecto.

Las técnicas de síntesis sonora se pueden organizar en tres grupos generales, síntesis lineal, síntesis no lineal, y síntesis computacional.

2.8.2. Síntesis lineal

En este grupo de técnicas, los procesos aplicados a la señal de partida generada son lineales y no generan nuevos parciales armónicos (operaciones algebraicas, filtros lineales, etc.)

■ Síntesis por tabla de ondas

Esta técnica se basa en la reproducción secuenciada de formas de onda contenidas en una tabla previamente almacenada, generando así una señal de oscilación periódica, cuya componente espectral depende de la forma de onda empleada.

Además de poder utilizar formas de ondas básicas (sinusoidal, cuadrada, triangular...) generadas por funciones matemáticas, se suele utilizar para emular sonidos de instrumentos tradicionales, cuyas formas de onda han sido previamente obtenidas y almacenadas en tablas.

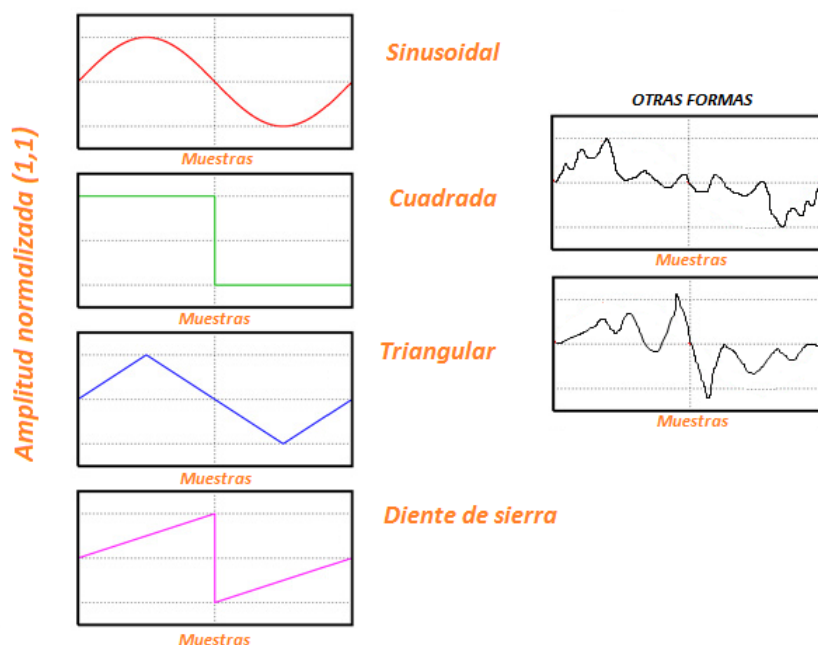


Figura 2.39: Diferentes tablas de onda

Algunos de los instrumentos implementados en el código del algoritmo, utilizan síntesis por tabla de ondas.

■ Síntesis sustractiva

Este tipo de síntesis trabaja eliminando componentes espectrales de un sonido inicialmente rico en contenido espectral. Las señales a las que se les sustraen porciones del espectro pueden ser sintéticas o grabadas.

Es usual aplicar sustracción de parciales a señales complejas sintéticas como por ejemplo las ondas “diente de sierra”, “cuadradas” o el “ruido blanco” (su densidad espectral de potencia es una constante para todas las frecuencias, espectro plano).

La sustracción de parciales de un tono complejo se realiza mediante algún tipo de filtro, como por ejemplo el filtro paso banda, rechaza banda, filtro peine, o también mediante una combinación de filtros, incluso algún filtro implementado mediante la edición gráfica del espectro.

Una analogía que pretende explicar el concepto de la síntesis sustractiva de manera más gráfica puede ser la escultura, se trata de moldear y dar la forma deseada partiendo de un bloque tosco (análogo al espectro rico en parciales), eliminando porciones no necesarias para el resultado final.

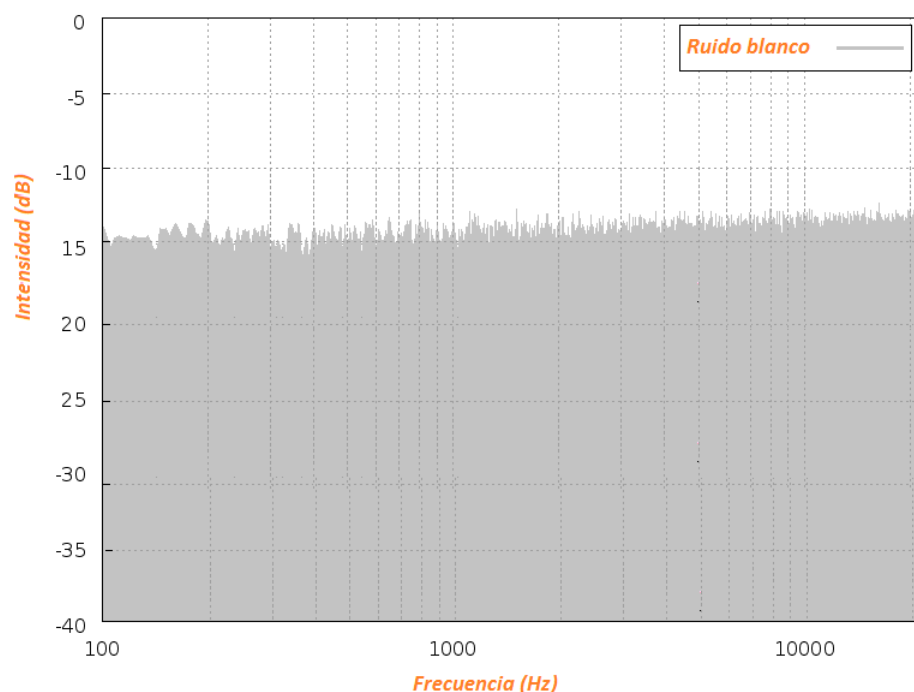


Figura 2.40: Espectro de ruido blanco

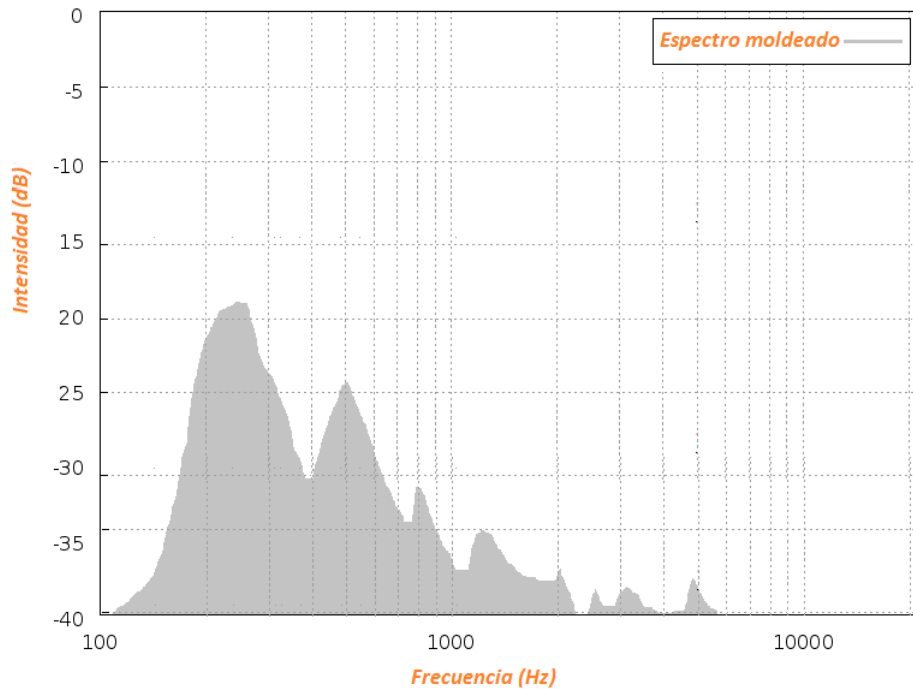


Figura 2.41: Espectro filtrado

- Síntesis aditiva

Esta técnica de síntesis en frecuencia tiene como fundamento el desarrollo en serie de Fourier de señales periódicas. Toda señal puede ser descompuesta como una suma de oscilaciones sinusoidales simples.

$$x(t) = \sum_{k=0}^{\infty} A_k \sin(k2\pi f_0 t + \Phi_k)$$

donde: A_k es la amplitud de la componente k , f_0 es la frecuencia fundamental de la señal y Φ_k es la fase de la componente k .

En la síntesis aditiva, se utilizan osciladores simples que producen las frecuencias deseadas, a estos osciladores también se les puede asignar una tabla de onda determinada, con mayor o menor contenido espectral. La disposición de las diferentes frecuencias producidas por la suma de osciladores determinarán el timbre característico del instrumento diseñado.

Como el oído no percibe la diferencia de fase de las oscilaciones, la ecuación anterior puede ser simplificada para su uso práctico.

$$x(t) = \sum_{k=1}^n A_k(t) \sin(k2\pi f_0 t)$$

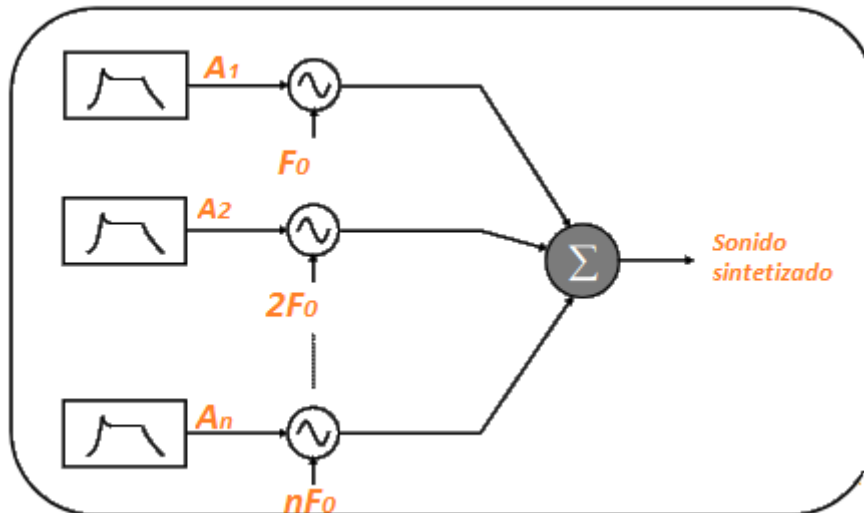


Figura 2.42: Esquema: síntesis aditiva

Es necesario entonces definir los valores de las frecuencias fundamentales y sus amplitudes respectivas, así como el número de osciladores n utilizados.

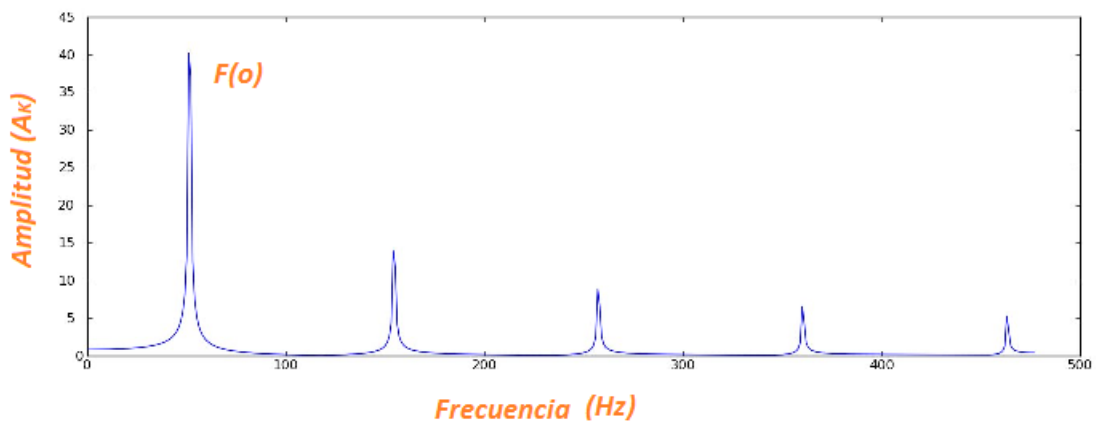


Figura 2.43: Espectro de una señal generada por síntesis aditiva

■ Síntesis multiplicativa o cruzada

Este tipo de síntesis se basa en la “impresión” de la envolvente espectral de una señal, sobre otro espectro rico en parciales, o señal con espectro constante, por medio de una convolución espectral, que multiplica uno a uno los valores del contenido en frecuencia de una señal por los de otra, produciendo un efecto sonoro de fusión entre los dos timbres, llamamos a la primera señal, señal moduladora, y a la señal rica en contenido espectral, señal portadora.

Un producto comercial que se basaba en este tipo de síntesis fueron los vocoders, que utilizaban una especie de órgano con el cual se creaba la señal portadora, y disponían de un micrófono, siendo la voz la señal moduladora en este caso.



Figura 2.44: Vocoder

2.8.3. Síntesis no lineal

Las operaciones que utilizan estas técnicas, sí que pueden alterar completamente el contenido en frecuencias de las señales de partida.

Estos tipos de técnicas de síntesis son muy eficientes a la hora de conseguir una relativa complejidad tímbrica con pocos osciladores. Pero uno de los inconvenientes es que son poco intuitivos de controlar.

Mencionaremos tres tipos de síntesis no lineal, los de mayor uso práctico.

- Síntesis por modulación en amplitud (AM)

Definición: La modulaciónn hace referencia al fenómeno por el cual, dadas dos señales, se hace que la primera varíe de acuerdo a la segunda, obteniéndose como resultado una nueva señal.

Llamamos a la primera señal: Portadora, a la segunda: Moduladora, y Señal modulada al resultado de la modulación entre la primera y segunda.

La modulación en amplitud consiste en modificar la amplitud de la portadora en función de la señal moduladora

La portadora suele ser una señal periódica de alta frecuencia (tono). La moduladora es una señal periódica de frecuencia entre 20Hz-20Khz (frecuencias audibles).

$$c(\text{portadora}) = A_c \cos(2\pi f_c t)$$

$$m(\text{moduladora}) = A_m \cos(2\pi f_m t) + A_m$$

$$c \cdot m = (A_c \cdot A_m) \cdot \cos(2\pi f_c t) \cdot \cos(2\pi f_m t) + A_m \cdot A_c \cos(2\pi f_c t)$$

$$c \cdot m = \frac{A_c \cdot A_m}{2} \cos((2\pi f_c + 2\pi f_m)t) + \frac{A_c \cdot A_m}{2} \cos((2\pi f_c - 2\pi f_m)t) + A_m \cdot A_c \cos(2\pi f_c t)$$

Como resultado, de modular ambas señales, se obtiene una señal compuesta por la suma de tres señales sinusoidales, analizando su espectro, la componente frecuencial de la portadora se conserva, y se añaden dos bandas laterales, con frecuencias iguales a la suma y a la resta de las frecuencias moduladora y portadora.

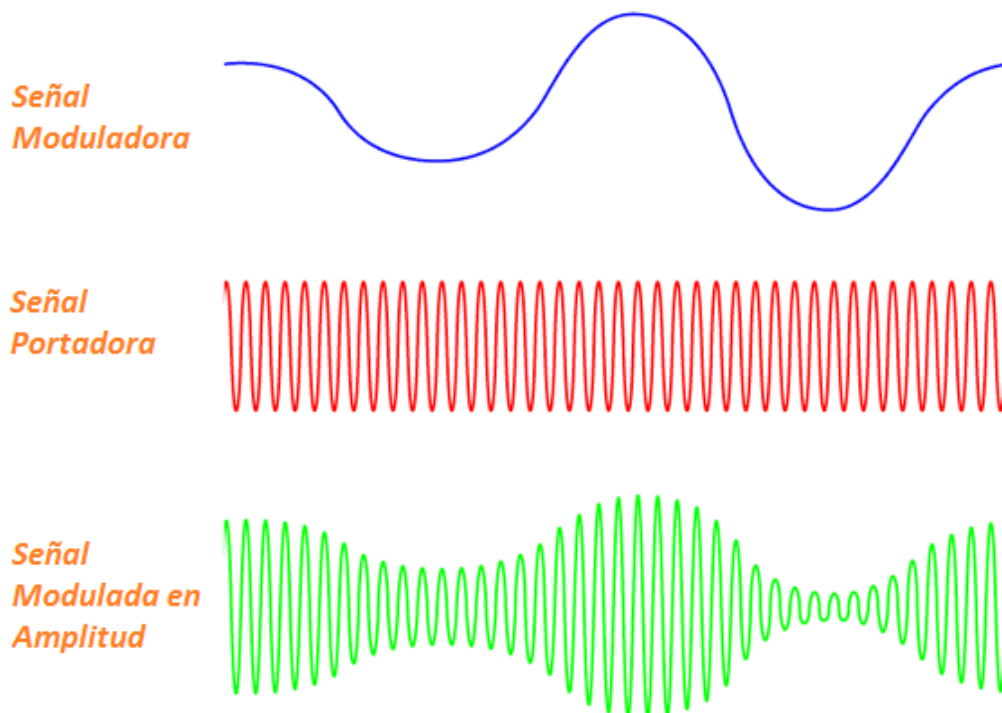


Figura 2.45: Representación temporal de las señales Moduladora, Portadora y Modulada AM

Este tipo de técnica de síntesis podría definirse en términos musicales como un trémolo extremo.

- Síntesis por modulación en frecuencia (FM)

Del conjunto de las técnicas de síntesis por modulación la más potente y flexible es la síntesis FM.

Está basada en variar la frecuencia de una señal (portadora) en función de otra señal (moduladora).

Varios de los instrumentos implementados en el algoritmo presentado, utilizan síntesis FM.

La síntesis FM fue descubierta por John Chowning en la universidad de Stanford a finales de los años 60; Chowning se encontraba experimentando con las técnicas de vibrato y descubrió que cuando la frecuencia de vibrato aumentaba hasta alcanzar la banda de frecuencias audibles el vibrato se convertía en un cambio de timbre (aparecían nuevas componentes espectrales).

Yamaha patentó la implementación hardware de la FM y más tarde, en los 80 lanzó al mercado el sintetizador DX-7.

Sean A_m, A_p, f_m, f_p las amplitudes y frecuencias de la moduladora y de la portadora. Una señal f_m viene definida por:

$$f_m = A_p \sin(2\pi f_p + A_m \sin(2\pi f_m))$$

Se define el índice P:M (o índice C:M) como: $\beta = \frac{f_p}{f_m}$.

La desviación máxima en frecuencia sobre la portadora es D .

El índice de modulación se define como: $I = \frac{D}{f_m}$.

Al igual que en el caso de la modulación AM, al modular una señal mediante FM se genera en el espectro una serie de bandas laterales. La diferencia es que mientras en la síntesis AM se generan dos bandas laterales, en FM se producen una n bandas laterales dadas por:

$$f_{m_s} = f_p \pm n \cdot f_m$$

El ancho de banda de una señal FM viene dado por:

$$f_m BW \approx 2 \cdot (D + f_m)$$

La amplitud de cada banda lateral varía de acuerdo a un tipo de funciones matemáticas conocidas como funciones Bessel tipo I y orden n , $J_n(I)$

El argumento de la función es el índice de modulación.

Teniendo en cuenta las amplitudes mencionadas, podemos reescribir la función de una señal f_m como:

$$\sum_{n=-\infty}^{\infty} J_n(I) \cdot \sin(2\pi[f_p \pm n \cdot f_m]t)$$

En FM, la frecuencia fundamental se obtiene a partir de la siguiente relación:

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

Donde N_1 y N_2 son enteros sin factores comunes. La frecuencia fundamental viene dada entonces por:

$$f_0 = \frac{f_c}{N_1} = \frac{f_m}{N_2}$$

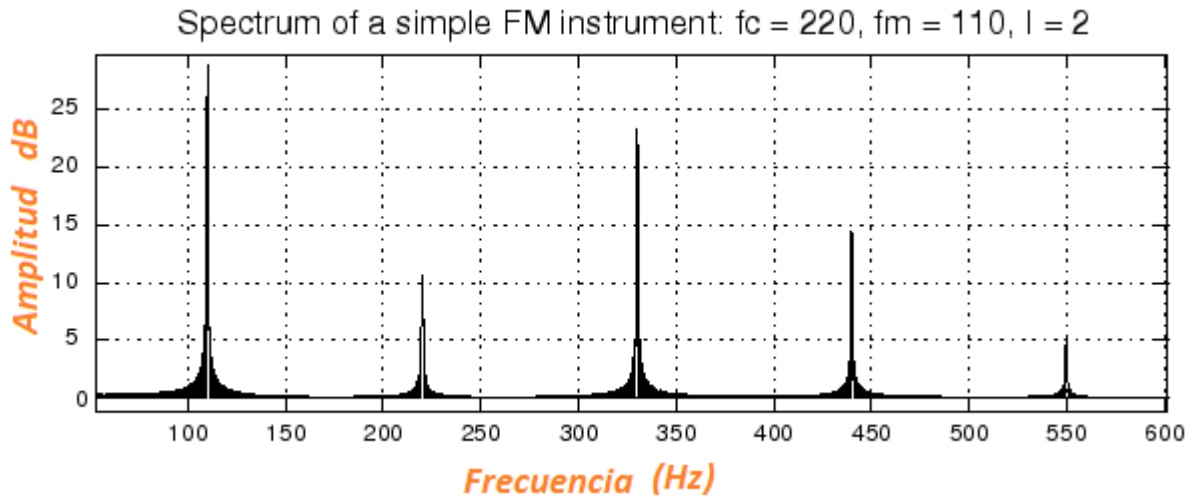


Figura 2.46: Espectro de una señal modulada en frecuencia

Una representación gráfica de las distintas funciones de Bessel se encuentra en los apéndices de la memoria.

Este tipo de técnica de síntesis podría definirse en términos musicales como un vibrato extremo.

- Síntesis por moldeado de ondas (Waveshaping)

Es una técnica de síntesis por distorsión que produce un espectro con una evolución dinámica de sus componentes, pero de banda limitada, es decir, se establece un número máximo de armónicos.

La distorsión o moldeado de la forma de onda de la señal original, es producida por la función moldeadora, que recibe la señal original como parámetro de entrada, obteniendo un resultado a la salida que depende de la forma de onda de la señal y de la función de transferencia utilizada.

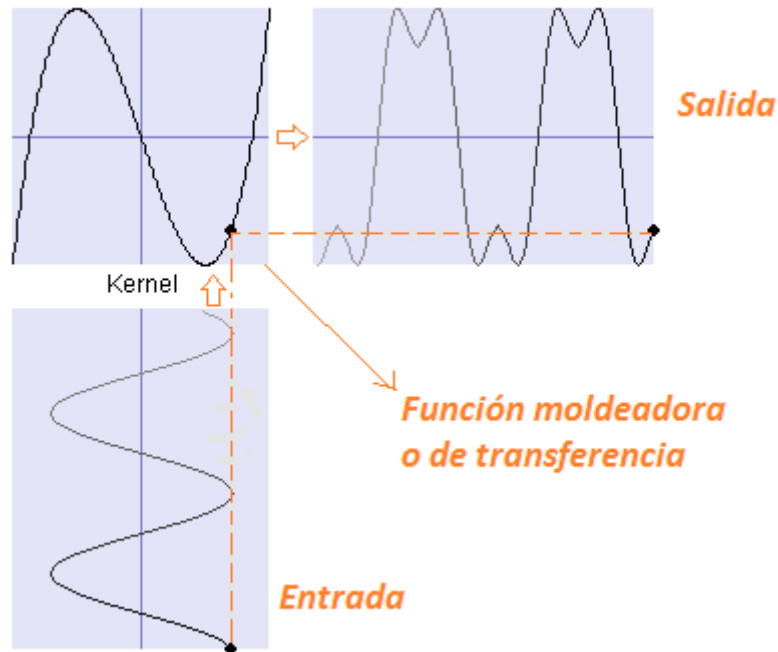


Figura 2.47: Ejemplo de distorsión de una señal por Waveshaping: comportamiento no lineal

Es un muy interesante método, pues produce un comportamiento no lineal, que depende de la amplitud de la señal original, como en muchos instrumentos acústicos, cuya evolución del timbre en función de la potencia sonora no es lineal.

2.8.4. Síntesis computacional

Estas técnicas son inconcebibles sin la gran capacidad de los ordenadores para el rápido cálculo de funciones complejas y para procesar y manipular modelos matemáticos.

- Síntesis por modelado físico

Técnica de síntesis del sonido que utiliza un modelos matemáticos para simular las propiedades físicas de la generación del sonido de instrumentos reales o modificados por conveniencia.

El sonido se genera usando resonadores (cómo y con que frecuencias vibran los distintos elementos físicos del instrumento), y generadores de la excitación que produce el sonido.

Por ejemplo, para emular el sonido de una guitarra, un instrumento de síntesis por modelado físico implementa todas las ecuaciones implicadas en la vibración de la cuerda y en las propiedades acústicas de la caja, así como la zona de excitación y su técnica de excitación (Dedos, púa, slider, etc).

Se trata de un método de síntesis que requiere el uso de microprocesadores específicos ya que realiza un elevado volumen de cálculo en tiempo real. La ventaja es que los sonidos generados

son muy realistas.

Karplus-Strong es un ejemplo de algoritmo de modelado físico, genera sonidos tipo pluck, (cuerda pulsada), y está implementado en varios instrumentos de nuestro algoritmo.

Estos instrumentos virtuales suelen tener interfaces gráficas, que muestran el esquema del instrumento a emular, y proporcionan la posibilidad de variar los parámetros físicos más importantes de una forma muy intuitiva y directa.



Figura 2.48: Interfaz de un sintetizador por modelado físico.

Existen más técnicas de síntesis, algunas de ellas combinan algoritmos de diferentes naturalezas, el sistema propuesto admite la implementación de muchos más instrumentos de prácticamente cualquier tipo de síntesis, ampliando así las posibilidades sonoras, y permitiendo una continua evolución y mejora de las prestaciones del mismo.

La señal producida por estos instrumentos sintetizadores digitales, es ya un resultado sonoro muy interesante, que a través de la salida de audio del software utilizado (CSOUND) y a de la salida analógica del procesador (Line Out) se encamina hacia los altavoces, habiendo sufrido previamente un proceso de conversión Digital-analógico (D/A) realizado por la tarjeta de sonido.

Además, en el algoritmo implementado, se puede habilitar la conversión a eventos MIDI en tiempo real de los parámetros obtenidos en la fase de análisis, posibilitando el uso de estas secuencias de información musical en otros programas editores, secuenciadores o sintetizadores de sonido, sean software o hardware, por ejemplo, creación de una partitura musical en diversos formatos, que refleje la secuencia interpretada por el músico.

2.9. MIDI

En esta sección hablaremos resumidamente del lenguaje entre dispositivos musicales MIDI (acrónimo de: Musical Instruments Data Interfaz).

Cabe aclarar un error de concepto comúnmente cometido al hablar de sonido y MIDI; Se suele relacionar el sonido pobre sintetizado por las tarjetas de sonido convencionales con el MIDI, el sonido MIDI no existe, una secuencia MIDI no suena, tan sólo es un protocolo de comunicación universal y digital entre equipos o dispositivos musicales. Con este protocolo se pueden transmitir secuencias digitales que serán interpretadas por equipos como tarjetas de sonido, y, la calidad o complejidad del sonido solo depende de las prestaciones o la gama del dispositivo, (Convencional, Semi-Profesional, Profesional...),

El algoritmo implementado, como se adelantó al final de la sección anterior, permite convertir las notas musicales detectadas (parámetros de frecuencia y duración) a eventos MIDI, lo que conlleva una ampliación de posibilidades de edición musical.

En los apéndices de la memoria se proporciona información detallada de diferentes aspectos del protocolo MIDI.

2.9.1. Introducción

A mediados de los años sesenta surgen los primeros sintetizadores comerciales. Eran instrumentos analógicos y (por su alto coste) monofónicos, es decir, capaces de emitir una sola nota a la vez. Por ello, se pensó en formas de conectarlos para permitir el control de diferentes instrumentos por ejemplo desde un teclado y conseguir, de este modo, más notas simultáneas, así como sonidos más ricos.

Las primeras comunicaciones entre estos aparatos fueron también analógicas; en ellas el voltaje de la señal era proporcional a la frecuencia deseada (y, por consiguiente, a la altura de la nota pulsada). Pero debido a que los equipos de distintos fabricantes no obedecían de la misma forma a estas señales de control y que de esta forma se seguía sin poder controlar más de una nota (pues la suma de dos voltajes produciría una nota más aguda, no dos notas separadas); Se determinó la necesidad de crear una solución de protocolo digital, y en 1981 surgió el primero, bautizado como USI (Universal Synthesizer Interface). Pero no tuvo demasiado éxito y por lo tanto no fue precisamente universal, por lo que durante los años siguientes una comisión de fabricantes japoneses y norteamericanos de instrumentos electrónicos se consagró a definir el protocolo estándar que iba a permitir la conexión entre estos aparatos, independientemente del fabricante.

Así nació el MIDI, en 1983, y fue tal el éxito obtenido y el importante mercado que se generó que, aunque las especificaciones técnicas hayan quedado obsoletas para las posibilidades tecnológicas de hoy, la normativa no ha cambiado en ningún punto, si acaso se han añadido detalles no contradictorios con la especificación original.

Además de resolver muchos problemas, tras la creación del MIDI, se han abierto multitud de posibilidades inimaginables antes de su aparición.

Una ventaja razonable es la posibilidad de separar el dispositivo controlador (teclado en la

mayoría de los casos) del propio sintetizador, anteriormente unidos, así se pueden controlar varios sintetizadores desde el mismo controlador.

2.9.2. El estándar MIDI

La norma MIDI se refiere entre otros aspectos a el tipo de cables y conectores utilizados para la conexión de dispositivos, la estructura de la señal midi, el tipo de mensajes y su codificación.

■ Cables y conectores

Los cables utilizados deben estar blindados para hacer frente a las posibles interferencias y no deben medir más de 15 metros. En cada extremo del cable debe existir un conector tipo DIN, así como en los puertos de entrada a los dispositivos.

Estos cables se pueden conectar en tres tipos de puertos con el mismo conector, indicados como MIDI IN (entrada, recibe la señal desde otros dispositivos), MIDI THRU (a través, se hace una copia de la entrada que se envía a través de ese puerto, sirve para enlazar varios dispositivos que utilicen la misma señal) , y MIDI OUT (salida, transmisión de mensajes generados por el dispositivo).

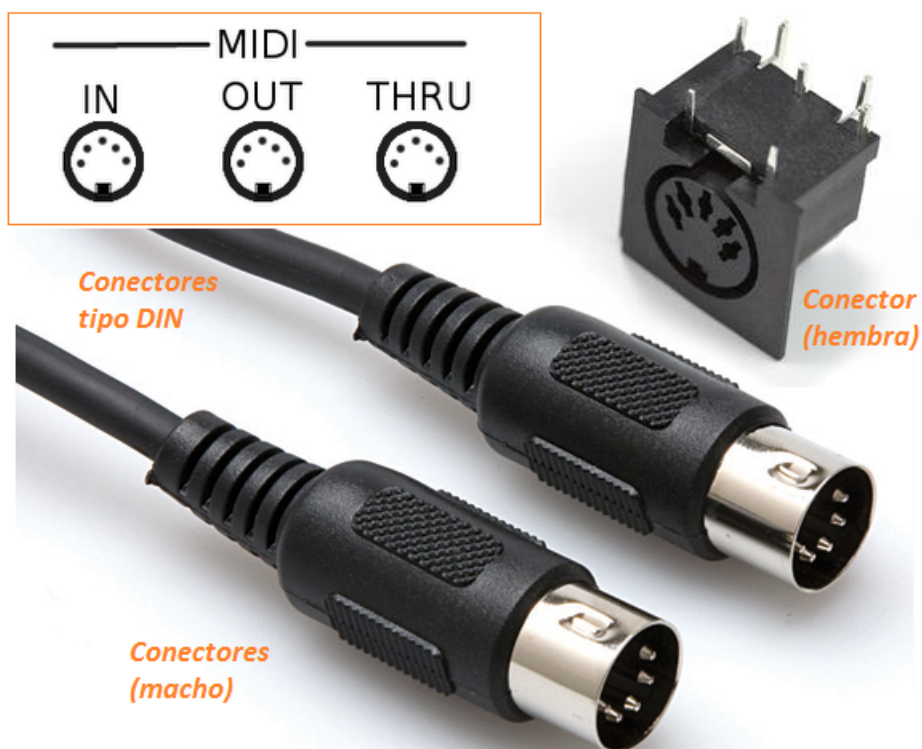


Figura 2.49: Conectores tipo DIN

Por un cable MIDI se transmiten mensajes codificados en código binario, son las órdenes a interpretar por el dispositivo de destino, la velocidad de transmisión de esta señal binaria es de

31.250 baudios que es el resultado de dividir 1 MHz entre 32, y el tipo de transmisión es asíncrona, necesitando por ello cada mensaje un bit de comienzo (un cero) y otro de fin (un uno) del evento o mensaje, la información útil de un mensaje se codifica en palabras código de 8 bits(1 byte), luego la velocidad real de transmisión de información útil es de 3.125 bytes/segundo.

Existen múltiples tipos de configuraciones de interconexión de dispositivos MIDI: En cadena, en bucle, en estrella, y combinaciones de las anteriores.

El estándar MIDI define el concepto de canal, para posibilitar una conexión multi-banda, un cable MIDI puede transmitir simultáneamente hasta 16 canales distintos, representados por la numeración del 0 al 15.

■ Mensajes MIDI

La estructura de un byte MIDI es la siguiente:

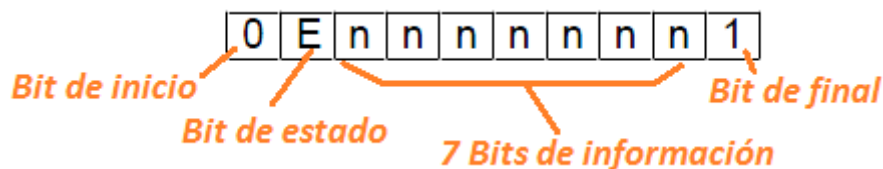


Figura 2.50: Estructura de un byte MIDI

Donde el primero y el último bit son de inicio y final de mensaje respectivamente, el bit E es el bit de estado, (valdrá 1 para los bytes de estado y 0 para los de datos), y el resto de bits son los de información, como éstos son siete, los valores de los diferentes parámetros expresados en código estarán dentro del rango (0 - 127).

Dependiendo del byte de estado que preceda a la secuencia, esta se interpretará de una forma u otra, un byte de estado se caracteriza por tener el segundo bit con un valor de 1, organizando así la secuencia asíncrona de cada canal.

Podemos distinguir dos grandes grupos en cuanto al tipo de mensajes transmitidos según la norma MIDI: Mensajes de canal y mensajes de sistema, a continuación se detallará la función de cada uno de ellos.

Mensajes de canal: Estos mensajes actúan sobre un único canal, cuyo número de referencia viene indicado en los 4 últimos bits del byte de estado, y controlan tanto la emisión o el cese de notas musicales, sus parámetros de frecuencia, modulación, etc. Como el cambio de programa, que en los sintetizadores generadores habituales se interpreta por un cambio de preset de timbre.

Podemos organizar estos tipos de mensajes de canal en 2 subgrupos: Mensajes de voz y mensajes de modo, y a continuación se detallan sus principales características.

Mensajes de voz: Los mensajes de voz hacen referencia a las órdenes de interpretación de la secuencia musical transmitida como: inicio de nota, fin de nota, especificación de la altura de nota, modulación, cambio de programa, etc. A continuación se proporciona una tabla con los tipos de mensajes de voz y su codificación.

MENSAJES DE CANAL DE VOZ	byte de estado			1º byte de datos	2º byte de datos
	Dec.	Hex.	Bin.	valor	valor
nota off	128	8cH	1000cccc	nº de nota	velocidad
nota on	144	9cH	1001cccc	nº de nota	velocidad (si 0 = nota off)
postpulsación polifónica	160	AcH	1010cccc	nº de nota	presión
cambio de control (en general)	176	BcH	1011cccc	nº de controlador	valor numérico
cambio de programa	192	CcH	1100cccc	nº de programa	
postpulsación de canal	208	DcH	1101cccc	presión	
variación de altura	224	EcH	1110cccc	menos significativo	más significativo

Bits de canal

Figura 2.51: Tabla: Mensajes de canal (voz)

Mensajes de modo: Indican la forma de utilización y gestión de los mensajes de modo y los canales de un instrumento, órdenes como: restablecer valores de controladores, activación de las entradas y salidas locales o remotas, silenciado del canal, etc. A continuación se proporciona una tabla con los tipos de mensajes de modo y su codificación:

MENSAJES DE CANAL DE MODO	byte de estado			1º byte de datos	2º byte de datos
	Dec.	Hex.	Bin.	valor	valor
restaurar controladores	176	BcH	1011cccc	121	0
control local on / off	176	BcH	1011cccc	122	127 / 0
desactivar todas las notas	176	BcH	1011cccc	123	0
desactivar modo omni	176	BcH	1011cccc	124	0
activar modo omni	176	BcH	1011cccc	125	0
modo mono on	176	BcH	1011cccc	126	0-16
modo mono off	176	BcH	1011cccc	127	0

Figura 2.52: Tabla: Mensajes de canal (modo)

Mensajes de sistema: Son mensajes para todo el sistema MIDI por lo tanto no se especifica su número de canal, en todos ellos el byte de estado comienza con la secuencia (1111), estos mensajes son útiles entre otras funciones, para coordinar y sincronizar varios dispositivos MIDI interconectados.

Los mensajes de sistema se subdividen en mensajes System Common, System Real Time, o System Exclusive.

Los primeros están destinados a todos los receptores del sistema; los segundos se utilizan para la sincronización de los elementos que funcionan mediante un reloj temporizador. Por su parte, los mensajes de Sistema Exclusivo incluyen un código de identificación del fabricante y se utilizan para transferir bytes de datos formateados según una especificación diseñada por ese fabricante.

A continuación se proporciona una tabla con los tipos de mensajes de sistema y su codificación:

MENSAJE DE SISTEMA	Byte de estado		Primer byte de datos	Segundo byte de datos
	Dec. Hex.	Bin.		
COMUNES				
trama temporal MTC	241	F1H	11110001	número de mensaje (0-7) / dígito decimal de tiempo (0-9), 8 mensajes forman un tiempo SMPTE en horas:minutos:segundos:frames (por ejemplo 02:48:37:12)
puntero de posición de canción	242	F2H	11110010	el menos significativo
selección de canción	243	F3H	11110011	el más significativo
no definido	244	F4H	11110100	Número de canción
no definido	245	F5H	11110101	
petición de afinación	246	F6H	11110110	
DE TIEMPO REAL				
reloj temporizador	248	F8H	11111000	
no definido	249	F9H	11110101	
empezar	250	FAH	11111010	
continuar	251	FBH	11111011	
parar	252	FCH	11111100	
active sensing	254	FEH	11111110	
system reset	255	FFH	11111111	
	240	F0H	11110000	(inicio de "Sysex")
DE SISTEMA EXCLUSIVO	código de fabricante y modelo			
 cualquier número de bytes de datos			
	247	F7H	11110111	(fin "Sysex")

Figura 2.53: Tabla: Mensajes de sistema

- Tarjeta de implementación MIDI (MIDI implementation chart)

Como hemos visto, existen multitud de mensajes diferentes en la norma MIDI, por lo tanto es razonable pensar que existirán dispositivos que no reconozcan algunos de ellos, ya sea por su antigüedad, o por tratarse de mensajes especiales como los Sysex (System exclusive), si un dispositivo no reconoce algún mensaje, lo ignora por defecto, una manera de saber previamente si puede o no un dispositivo reconocer ciertos mensajes es consultando la tarjeta de implementación, Formato tabla estándar, es muy recomendable consultarlo antes de la adquisición de un dispositivo MIDI.

A continuación se proporciona un ejemplo de tarjeta de implementación MIDI:

MIDI Implementation Chart

Function		Tx	Rx	Remarks
Basic Channel	default	x	1-16	set by Channel Pins
	changed	x	x	
Mode	default	x	x	
	Messages	x	x	
	changed	x	x	
Note Number		x	36-107	C2-B7 (65.405-3951 Hz)
	True Voice	*****	x	
Velocity	Note On	x	o	
	Note Off	x	x	
After Touch	Keys	x	x	
	Channels	x	x	
Pitch Bender		x	x	
Control Change	7	x	o	Main Volume
	9	x	o	Sine / Rectangle Select
	64-67	x	o	Osc0: A, D, S, R
	68-71	x	o	Osc1: A, D, S, R
	72-75	x	o	Osc2: A, D, S, R
	76-79	x	o	Osc3: A, D, S, R
	80-83	x	o	Osc4: A, D, S, R
	84-87	x	o	Osc5: A, D, S, R
	88-91	x	o	Osc6: A, D, S, R
	92-95	x	o	Osc7: A, D, S, R
	102-109	x	o	individual Osc Volumes
110-117	x	o	individual Osc Pans	
Program Change		x	x	
	True #	x	x	
System Exclusive		x	x	
System Common	Song Pos	x	x	
	Song Sel	x	x	
	Tune	x	x	
System Realtime	Clock	x	x	
	Commands	x	x	
Aux Messages	Reset	x	o	

o : implemented
x : not implemented

Figura 2.54: Tarjeta de implementación MIDI

2.9.3. General MIDI

General MIDI es una extensión del protocolo MIDI.

La especificación General MIDI es adoptada en 1991 por la Asociación de Fabricantes MIDI (MIDI Manufacturers Association: MMA) y establece unas características mínimas para la compatibilidad de los instrumentos MIDI. Entre otras cuestiones establece, por ejemplo:

- 24 voces como mínimo de polifonía, que puedan tener un volumen independiente.
- 16 canales MIDI, que puedan enviar distintos número de voces (polifonía) y/o tocar diferentes instrumentos. El canal 10 se establece como el canal básico para la percusión.
- Un conjunto mínimo de 128 instrumentos y la posibilidad de que al menos 16 puedan sonar simultáneamente (multi-tímbrico).

Estas especificaciones permiten garantizar que las asignaciones que hayas hecho de instrumentos en una obra musical se respeten, aunque cambies de dispositivo MIDI. Es decir, si decides, por ejemplo, que la melodía principal la debe interpretar un saxofón alto, ése será el instrumento que suene, aunque reproduzcas el archivo MIDI a través de otro dispositivo MIDI o en otra tarjeta de sonido de un ordenador distinto. Lo que no se garantiza es que el instrumento 66 (saxofón alto) suene con la misma calidad siempre, pues esto dependerá de la calidad de los sonidos que genere el sintetizador o la tarjeta de sonido que los reproduzca.

Con respecto a los sonidos de percusión, se asignan al canal 10, y cada tecla va asociada a un instrumento percusivo. En MIDI cada tecla de un piano (teclado) está numerada.

Tanto los 128 instrumentos, como los sonidos de percusión de General MIDI, están especificados en los apéndices de la memoria.

Pero la rápida evolución de la tecnología musical hizo que pronto esta norma resultara insuficiente y que algunos fabricantes de instrumentos ampliaran sus especificaciones (aunque siempre respetando lo establecido por la especificación GM, para mantener la compatibilidad):

El primer fabricante fue Roland, introduciendo muchos controles e instrumentos extra y lo denominó **GS Standard** El siguiente fabricante fue Yamaha, con requisitos superiores aún a los establecidos en GS. Lo denominaron **XG**. En este panorama, el propio GM ha sufrido una evolución, o mejor dicho, una ampliación en cuanto a sus exigencias. De este modo, la primeras especificaciones del General MIDI pasan a denominarse GM nivel 1 (**GM1**) y a las nuevas GM nivel 2 (**GM2**), y continúan actualizándose desde que surgieron en 1999.



Figura 2.55: Logos: General MIDI, GS Standard y XG MIDI

2.9.4. Controladores

El MIDI ha permitido que haya una clara separación entre dispositivos controladores (maestros) y dispositivos productores de sonido (esclavos). Surge así una disociación entre el modo en que se introduce la información musical y los sonidos resultantes. Es decir, hasta ahora, cuando veíamos a un pianista o trompetista tocar su instrumento, ya preveíamos, antes de que sonase, cuál iba a ser el timbre del mismo.

Sin embargo, el mundo MIDI ha permitido que los controladores puedan adoptar el mecanismo de ejecución de prácticamente cualquier instrumento musical, permitiendo así que los distintos músicos (no sólo los pianistas) puedan tener acceso al mundo MIDI, aprovechando la técnica del instrumento que han aprendido. Así, una guitarra MIDI, por ejemplo, podría utilizar el sonido de un órgano, de una flauta o de cualquier sonido electrónico, que no pretenda imitar a ninguno de los instrumentos musicales existentes.



Figura 2.56: Distintos controladores MIDI

En la actualidad estos controladores pueden emplear conexiones MIDI o utilizar directamente **conexiones USB**, lo que facilita su conexión al ordenador y nos permite prescindir del puerto MIDI de la tarjeta de sonido (que podría no tenerlo). A través de estos controladores pueden introducirse y grabar directamente interpretaciones musicales en el ordenador, mediante un programa de secuenciación. También pueden emplearse para introducir información en los edi-

tores de partituras.

Uno de los avances que proporciona este proyecto, es conseguir el control MIDI a través de un instrumento electroacústico, es decir, la señal de audio genera a su vez una señal de control MIDI vía procesado informático.

Terminamos así el repaso de conocimientos previos relativos al presente proyecto, mucha información se ha obviado, por no tener demasiada relevancia con el contexto del sistema diseñado, aún así en los apéndices de la presente memoria, se proporciona información complementaria adicional de algunos de los temas anteriormente tratados.

Capítulo 3

Sistemas actuales relacionados

Este apartado analizará las propiedades, ventajas y desventajas de algunos sistemas comerciales o prototipos relacionados con el sistema propuesto.

3.1. Sistemas Software

Como el algoritmo presentado, estos sistemas no requieren equipo hardware para realizar las distintas funciones, son aplicaciones informáticas que trabajan con señales de audio digital como fuente de información.

3.1.1. RTPITCH Versión 1.3

Herramienta de Windows para obtener seguimiento de tono en tiempo real.

RTPITCH es un programa enfocado a la visualización del pitch (frecuencia fundamental) en tiempo real para una señal de voz. Con RTPITCH puede visualizar la forma de onda (Dominio del tiempo) y el tono (Frecuencia) de los sonidos del habla que se está reproduciendo en el micrófono de la computadora o de los puertos de entrada de línea. RTPITCH también tiene una opción para comparar dos secuencias de pitch, lo cual es utilizado para la práctica de la entonación.

RTPITCH no es software de dominio público, la propiedad intelectual es propiedad de Marcos Huckvale, University College London. Sin embargo la aplicación RTPITCH puede ser usada y copiada sin cargo alguno siempre que el programa y el archivo de ayuda se mantengan sin cambio y sea visible este aviso de copyright.

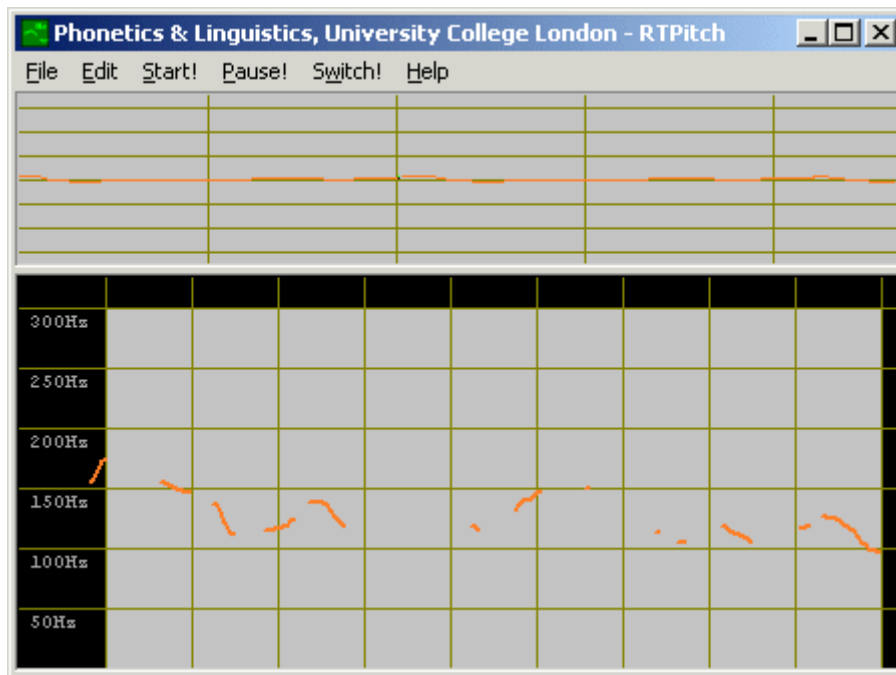


Figura 3.1: Interfaz RTPITCH 1.3

Una ventaja de este software es la visualización de la variación temporal del pitch o (FO), es muy útil para reconocimiento básico de voz o en un contexto pedagógico, para practicar la entonación.

Una desventaja con respecto al algoritmo presentado, es que no incluye técnicas de re-síntesis, no ofrece herramientas creativas.

3.1.2. Dalton Harts Harmonizer

Se trata de un plug-in (Algoritmo independiente que puede utilizarse en varios entornos de edición de audio en este caso) armonizador vocal basada en MAX MSP. Su funcionamiento y utilidades son las siguientes: El usuario canta en un micrófono mientras se toca un teclado MIDI, y el software modifica automáticamente la frecuencia fundamental de su voz para que coincida con cada una de las notas pulsadas en el controlador MIDI. Es polifónico (32 voces) o monofónico, Posee controles de envolvente de volumen, pitch bend, vibrato, deslizamiento, y algunos más.

Utiliza **sigmund**, un algoritmo de seguimiento de tono, así como **shifter**, un algoritmo que produce el cambio de tono de las diferentes formantes de la señal de voz, posibilitando la creación de armonías vocales muy realísticas en tiempo real.

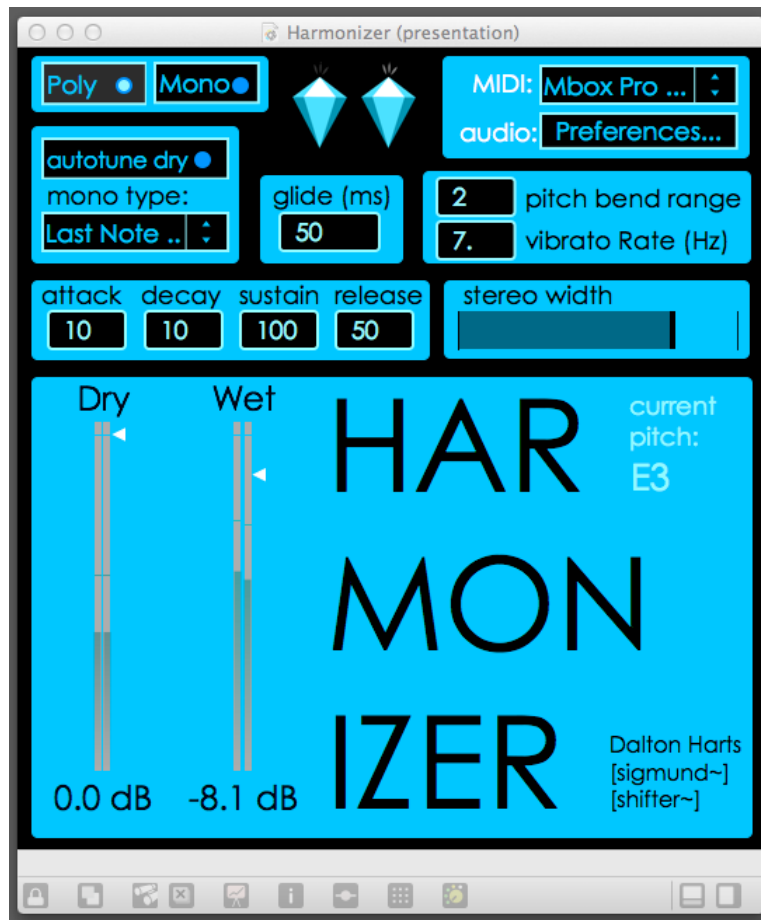


Figura 3.2: Interfaz Harmonizer

Este sistema sí que proporciona una interesante herramienta para el desarrollo creativo musical, su funcionamiento recuerda al vocoder, sólo que no existen modulaciones en este caso, la señal del controlador MIDI sólo determina las posiciones de las frecuencias fundamentales a las que se se ajustan las copias de la señal de voz.

3.2. Sistemas Hardware (MIDI pickups)

Estos sistemas hacen un seguimiento de la frecuencia fundamental a partir de audio analógico y con diferentes técnicas de detección de tono implementadas electrónicamente o mediante procesadores hardware.

las pastillas MIDI suelen disponer de tantos sensores como cuerdas posea el instrumento en el que esté instalado el sistema, de esta forma pueden generar señales MIDI polifónicas, estos sensores son los dispositivos encargados de captar las señales independientes a analizar.

En instrumentos electroacústicos como guitarras o bajos, algunas pastillas MIDI (llamadas hexafónicas en el caso de las guitarras)son simplemente instaladas o conectadas al puente, junto a las pastillas del instrumento, mientras que otros modelos sustituyen a las pastillas reales por los piezoeléctricos del sistema, a la salida del mismo se obtienen directamente los diferentes eventos MIDI, así que tan sólo se tiene que conectar a un sintetizador MIDI, secuenciador u ordenador para poder asignar los sonidos o efectos pertinentes.

3.2.1. MIDI Shadow

El sistema Shadow SH 075 es uno de los primeros conversores audio-MIDI, y hasta ahora sigue siendo popular entre los músicos profesionales. Es caro en comparación con las otras marcas, pero esto se justifica con su buen funcionamiento de baja latencia y funciones adicionales. Aparte de conseguir notas precisas, posee un montón de características prácticas, como MIDI polifónicos y monofónicos modo, MIDI transponer y un teclado MIDI que permite cambios en el programa, la señal MIDI que se envía directamente desde la guitarra sin necesidad de pedal.



Figura 3.3: Shadow , Conversor MIDI

Este sistema es muy robusto, pero no demasiado asequible, y por si solo no proporciona control de la síntesis del sonido.

3.2.2. Roland GK3

Como el sistema anterior, se puede instalar sobre el cuerpo de la guitarra, posee una pastilla Hexafónica y un módulo de procesado. El equipo Roland GK3 Permite obtener señales individuales de cada una de las cuerdas de guitarra, habilitando la interpretación de acordes y

polifonías, también detecta las diferentes dinámicas interpretadas. Este sistema posee un interruptor de selección para mezclar o cambiar entre sintetizador y el tono de la guitarra normal.

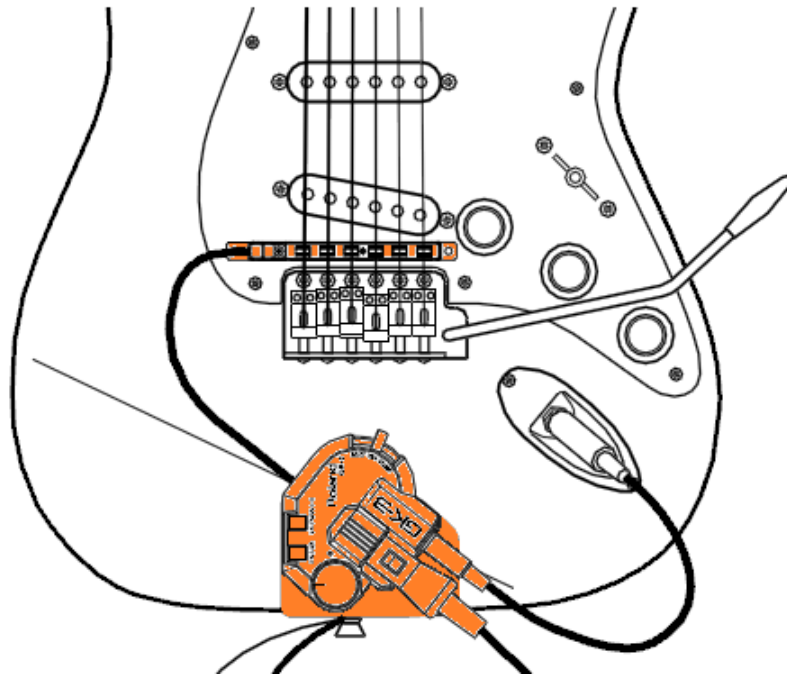


Figura 3.4: Esquema de instalación en guitarra , Roland Gk3

Los sistemas Hardware son muy interesantes, pues no necesitan conexión con un procesador externo, pero no proporcionan control de la síntesis por si mismos, se limitan a la conversión Audio-MIDI.

Capítulo 4

Tecnologías utilizadas

Este apartado de la memoria pretende dar a conocer los programas o aplicaciones utilizadas, que han servido de recurso o plataforma para la creación y el desarrollo del sistema propuesto.

4.1. Matlab

Las primeras pruebas de eficiencia de distintas técnicas de pitchtracking, han sido implementadas y desarrolladas en entorno Matlab, facilitando el proceso de selección del tipo de algoritmo utilizado en el sistema propuesto.

Matlab es un paquete de software orientado hacia el cálculo numérico científico e ingenieril, integra cálculo numérico, computación de matrices y gráficos en un entorno de trabajo cómodo para el usuario.

Su nombre significa Laboratorio de Matrices y fue escrito inicialmente en base a los ya existentes paquetes de cálculo matricial LINPACK y EISPACK. Posteriormente se han anadido librerías, denominadas Toolboxes, especializadas en diferentes áreas científicas.

Matlab ha evolucionado y crecido con las aportaciones de multitud de usuarios, en entornos universitarios se ha convertido en una herramienta instructora básica para cursos de matemáticas aplicadas así como para cursos avanzados en otras áreas. En entornos industriales se utiliza para investigar y resolver problemas prácticos y cálculos de ingeniería.

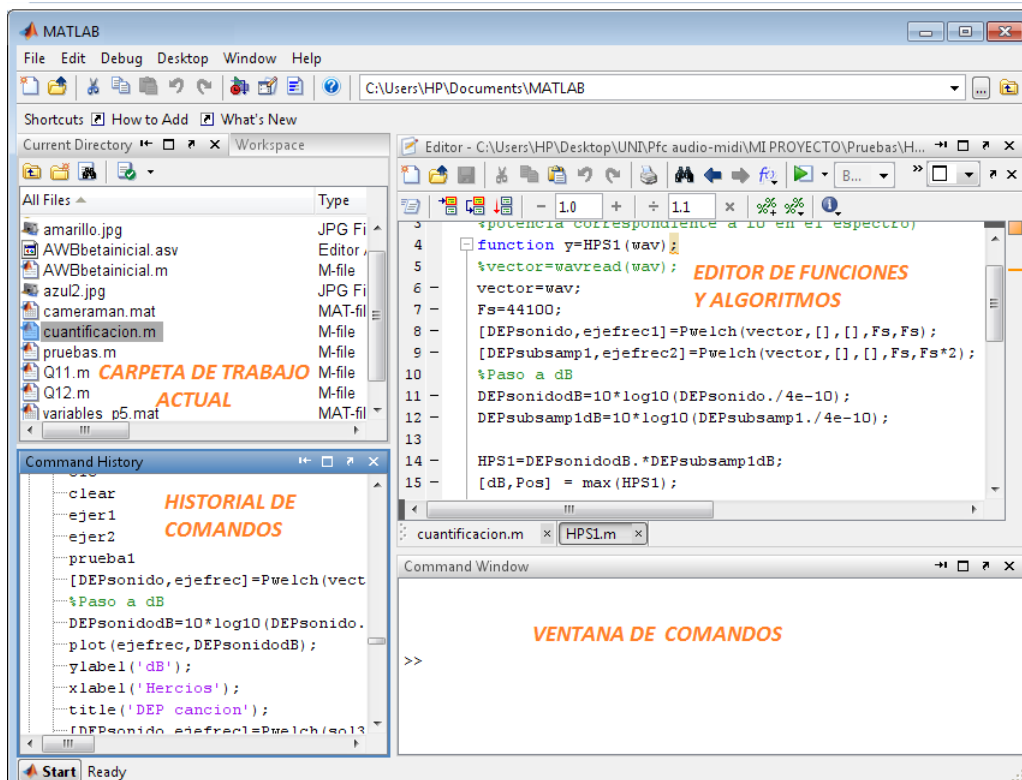


Figura 4.1: Interfaz Matlab

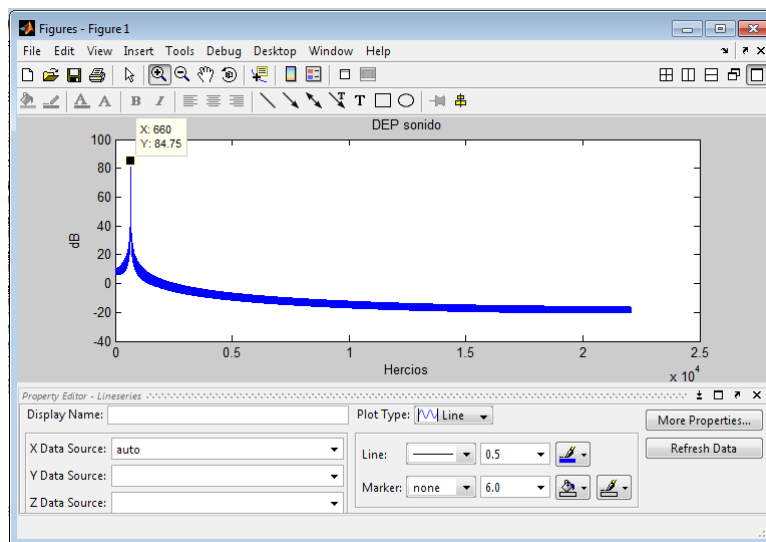


Figura 4.2: Editor de figuras Matlab

Sus aplicaciones más comunes son el cálculo numérico, la implementación de algoritmos, la resolución de problemas con formulación matricial, la estadística, la optimización, etc. Es de

destacar la aplicación en el estudio, simulación y diseños de los sistemas dinámicos y de control.

4.2. CSound

Una vez he comprendido el funcionamiento de el algoritmo de detección de tono a bajo nivel en entorno matlab, inicio la implementación del sistema propuesto en lenguaje CSound, basado en lenguaje C.

El cuerpo principal del proyecto se desarrollará en el entorno CSound, un lenguaje de programación de computadores, libre, abierto, y gratuito, orientado a sintetizar, editar, analizar música y sonido en formato de audio digital, su nombre viene dado por que su compilador está programado en lenguaje C.

CSound permite gestionar las entradas y salidas de audio digital y MIDI en tiempo real, de una manera intuitiva y directa, así como generar una interfaz gráfica donde se pueden controlar diferentes parámetros necesarios para el funcionamiento del prototipo planteado.

Este lenguaje de programación sonora es multi-plataforma y está disponible para todos los sistemas operativos: Linux, Mac OS X, Windows, etc.

CSound compila ficheros de tipo texto que contienen programas escritos en lenguaje CSound y produce sonido en tiempo real o un fichero de audio para su reproducción posterior.

Al compilar, CSound primero traduce un conjunto de instrumentos descritos en texto plano, que se encuentran en un archivo de orquesta, a una estructura de datos que reside en la máquina (la computadora). Después ejecuta estos instrumentos (definidos por el usuario) interpretando (como notas y otros parámetros) una lista de datos que el programa puede "leer.^{en} varios formatos: un archivo de partitura descrita como texto plano, un archivo MIDI, un controlador MIDI en tiempo real, audio en tiempo real (es el caso del sistema propuesto), o un dispositivo (no MIDI) como un teclado o ratón de computadora.

4.2.1. Archivos .csd

Los dos tipos de ficheros mencionados (Orquesta y partitura) pueden estar unificados en un fichero conjunto con extensión **.csd**, con una estructura determinada por una serie de comandos de inicio y final de las diferentes secciones que a continuación detallaremos. El algoritmo propuesto está programado en este tipo de ficheros **.csd**



Figura 4.3: Estructura básica de un documento .csd

4.2.2. Inicio y fin del algoritmo

Csound comienza a interpretar y compilar los ficheros a partir del comando `< CsoundSynthesizer >`, y deja de hacerlo cuando lea el comando de cierre de fichero `< /CsoundSynthesizer >` es decir, toda la programación del algoritmo viene delimitada por estos dos comandos, la información que se sitúe fuera de estos límites será obviada por el compilador, este espacio puede utilizarse para disponer información del autor de la aplicación o para hacer comentarios previos para la comprensión del algoritmo.

4.2.3. Sección de opciones

Delimitadas por los comandos `< CsOptions >`, y `< /CsOptions >` se incluyen las opciones específicas de funcionamiento: activación/desactivación de entradas y salidas de audio, MIDI, generación de archivos de diferentes formatos según las necesidades de la aplicación etc. En el código fuente que se proporcionará en capítulos posteriores se detallarán en forma de comentarios de código, todas las opciones características del algoritmo propuesto.

4.2.4. Sección de instrumentos

Entre los comandos `< CsInstruments >` y `< /CsInstruments >` se encuentra la programación de los diversos instrumentos que posea el archivo, es la sección de orquesta, posee varios comandos que delimitan el código referido a cada instrumento `instr` y `endin` en el inicio de esta sección de instrumentos se deben especificar las condiciones de trabajo o configuración.


```

sr    = int    ; Frecuencia de muestreo, es por defecto 44100.
kr    = int    ; Frecuencia de control, es por defecto 4410.
ksmps = int    ; Define una cantidad de muestreos en un periodo de control, el valor es sr/kr.
nchnls = int   ; Es el numero de canales que tendra el archivo, por ejemplo monoaural o estereo.
Odbfs = int    ; Define el valor de 0 decibelios para una amplitud de escala completa.

```

COMENTARIOS, Después de ";"

Figura 4.4: Cabecera de la sección de instrumentos: Configuración

Después de esta configuración previa se incluye la codificación en lenguaje CSound de los procesos matemáticos que vamos a realizar para generar los sonidos de los diferentes instrumentos, cada uno entre los comandos *instry endin*, donde cabe mencionar la utilización de diferentes códigos de operación llamados **opcodes**, son funciones prediseñadas por CSound o por usuarios expertos, que ayudan y aligeran el proceso de programación, éstos requieren unos parámetros de entrada y generan una determinada salida, todo ello con una sintaxis específica para cada opcode. También existe la posibilidad de crear nuestros propios opcodes. En el código fuente que se proporcionará en capítulos posteriores se detallarán en forma de comentarios de código, la función de cada uno de los opcodes utilizados, y en los apéndices de la memoria consta una lista con información detallada de estas funciones o códigos de operación.

4.2.5. Sección de partitura

En último lugar y entre los comandos $\langle CSscore \rangle$ y $\langle /CSscore \rangle$ se encuentra la sección de partitura, donde se generan las diferentes tablas de ondas anteriormente mencionadas y donde se especifica el orden de actuación de los diferentes instrumentos, en el caso particular del proyecto propuesto, no existe una partitura, sino que se dejan activados los diferentes instrumentos y es la propia señal de audio de entrada la que determina las diferentes notas a interpretar mediante la utilización de un umbral cuyo funcionamiento se explicará en los siguientes capítulos.

4.2.6. Tipos de variables en CSound

En CSound se utilizan diferentes tipos de variables dependiendo el uso que se les vaya a asignar:

- Variables tipo g-: globales.

Han de ser de algún tipo de los que siguen, por lo que después de la letra g debe haber alguna de las otras tres letras indicadoras de tipo. Toman valor al inicio de la compilación.

- Variables tipo i-: nota.

Toma valor al inicio de cada una de las notas que interpreta el instrumento.

- Variables tipo k-: control.

Toman un nuevo valor cada **kr** veces por segundo.

- Variables tipo a-: audio.

Toman valor **sr** veces por segundo. Son vectores (no se sustituye el valor anterior por el nuevo, sino que se almacena uno a continuación del otro).

4.3. CsoundQt

Este software es un front-end de CSound. Definimos front-end como la parte del software que interactúa con el o los usuarios y el back-end es la parte que procesa la entrada desde el front-end. La separación del sistema en front-ends y back-ends es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas. La idea general es que el front-end sea el responsable de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas, y procesarlas de una manera conforme a la especificación que el back-end pueda usar.

CsoundQt, antes llamado QuteCsound, es un editor multiplataforma y de front-end para CSound con control gráfico en tiempo real, resaltado de sintaxis, autocompletado, widgets interactivos, ayuda interactiva integrada y puesta en marcha automática de CSound desde un shell o la API.

Definimos Widget en CsoundQt como una interfaz gráfica de usuario personalizada (GUI) la cual puede controlar una orquesta en tiempo real.

Pretende ser un entorno sencillo pero potente y completo desarrollo de CSound. CsoundQT acerca CSound a un grupo más amplio de personas, reduciendo la curva de aprendizaje inicial.

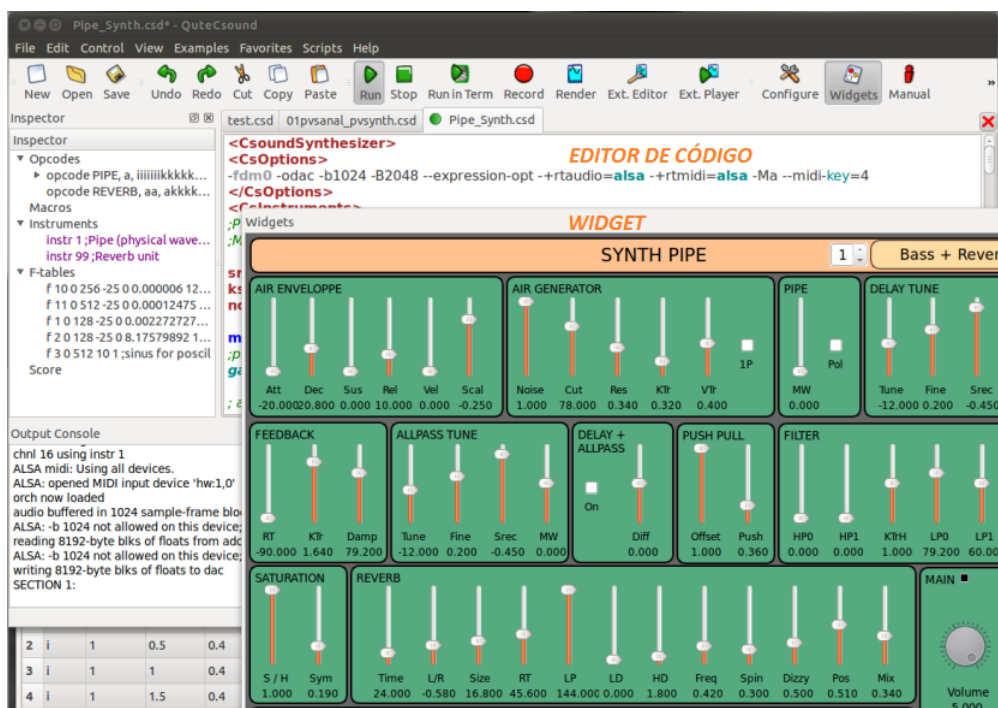


Figura 4.5: Interfaz y widget CsoundQt

4.4. ASIO4ALL

Se trata de un controlador de audio específico. Un controlador es un Software que permite al ordenador gestionar de manera óptima los componentes que tiene instalados. Cada componente necesita su propio controlador (Tarjeta de Sonido, Módem, CD-ROM, etc.).

ASIO4ALL es un controlador inteligente de baja latencia en entrada y salida de audio, emula los controladores ASIO, con funcionamiento en tiempo real para la mayoría de los dispositivos de audio en ordenadores.

Esta herramienta ha ganado muchísima popularidad en los últimos años convirtiéndose en un elemento indispensable en equipos semi-profesionales de producción y edición sonora, consigue reducir mucho la latencia, optimizando el encaminamiento de la señal de audio digital.

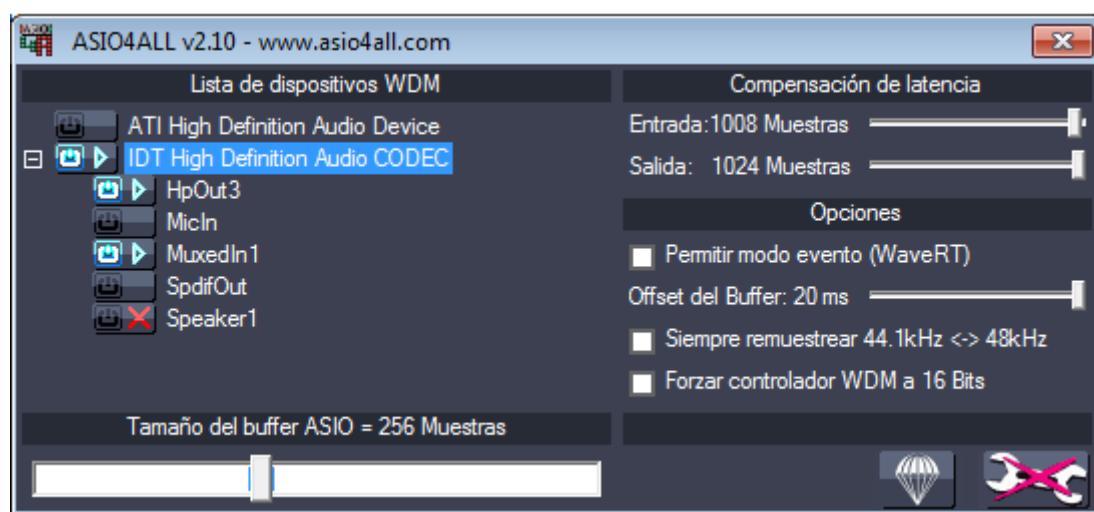


Figura 4.6: Interfaz ASIO4ALL

Como se puede ver en la figura superior, dispone de una consola gráfica o interfaz, donde se pueden seleccionar los diferentes dispositivos, también es posible la activación o desactivación de las diferentes entradas y salidas, así como controlar parámetros para obtener un rendimiento y una latencia óptimos en los dispositivos de audio.

4.5. LoopMIDI

Por último cabe mencionar el uso de este software, necesario en caso de la activación del envío MIDI del algoritmo, se encarga de gestionar y mapear las entradas y salidas software MIDI de los diferentes dispositivos emisores o receptores dentro del mismo equipo, es decir, funciona como un cable MIDI posibilitando interconexiones entre dispositivos, pero en un entorno virtual, lo que permite enviar señales MIDI desde una aplicación a otra, ambas en el mismo equipo.

Los puertos virtuales creados son únicos para cada usuario y sólo existen mientras que la aplicación loopMIDI se está ejecutando.

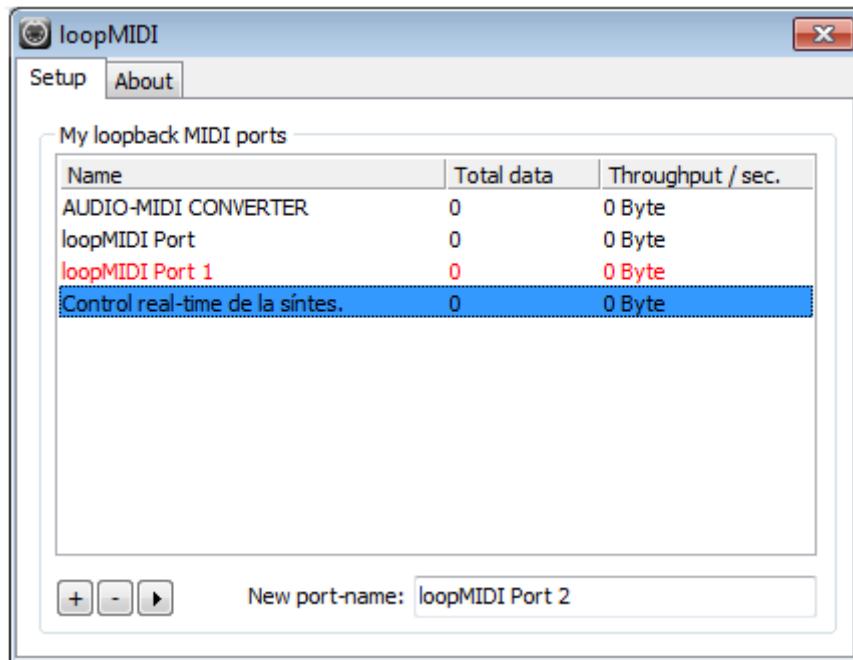


Figura 4.7: Interfaz LoopMIDI

Capítulo 5

Diseño e implementación del algoritmo.

En este capítulo abordaremos la fase de experimentación en matlab y la fase de implementación del algoritmo en lenguaje CSound, determinaremos su estructura y características, y se proporcionará el código fuente de la aplicación.

5.1. Estructura del sistema

Antes de profundizar en la programación de la aplicación, es conveniente conocer el esquema general del sistema propuesto.

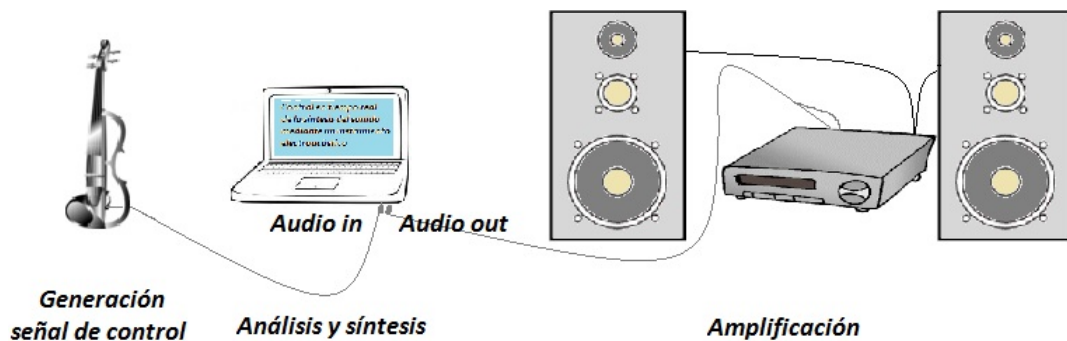


Figura 5.1: Esquema del sistema

La señal generada por el violín electroacústico se transmite por un cable de audio con conectores de 1/4 de pulgada (Mini-jack) y se conecta a la entrada de línea de la tarjeta de sonido de el ordenador, en el cual y mediante las aplicaciones anteriormente mencionadas, dicha señal se analiza y se extraen los parámetros de amplitud y altura tonal, a continuación produce la síntesis del sonido, el resultado depende de las opciones, el instrumento y el efecto seleccionado. Esta señal se envía al puerto de salida de audio de la tarjeta de sonido, finalmente y mediante otro cable (esta vez mini-jack - RCA) se conecta a un sistema de amplificación y transducción electro-acústica (Amplificador-Altavoces) que amplifica y reproduce la señal de salida.

5.2. Experimentación en matlab

La elección de técnica de pitchtracking, implica una de las decisiones previas más importantes. En los capítulos de la memoria que hacen referencia a esta fase del procesado, ya se han comentado algunos indicios que reducen las posibilidades de elección, basándose en la documentación previa y teniendo en cuenta los requisitos del sistema. (Robustez frente al ruido y funcionamiento en tiempo real).

Aún así se han efectuado una serie de pruebas previas en Matlab, que ayudan a comprender el proceso de detección de frecuencia fundamental, cuyos códigos fuente se incluyen en los apéndices de la memoria.

Algunas de las técnicas de detección de la frecuencia fundamental probadas en matlab estaban previamente descartadas para su implementación en nuestro sistema, pero se incluyen por su carácter ilustrativo del proceso.

- Detección de altura tonal de tonos puros mediante zerocrossing

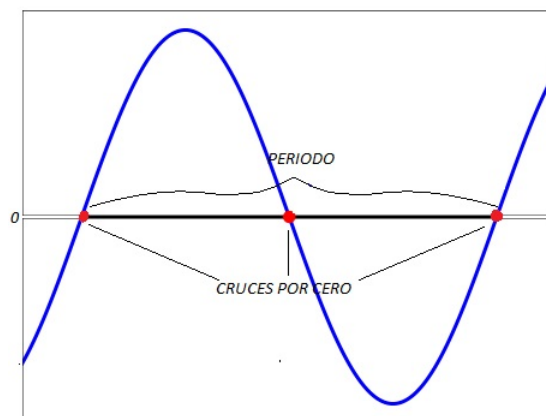


Figura 5.2: Cruces por cero

Se trata de hallar el periodo de la oscilación mediante el conteo de cruces por cero de la forma de onda de la señal en un determinado periodo de tiempo.

la señal analizada es una concatenación de tonos puros generados en la misma función (Prueba1), los tonos se corresponden con las frecuencias emitidas por las cuerdas del violín sin variación de longitud, (SOL3, RE4, LA4, MI5) y los resultados de la detección son los siguientes:

		SOL 3	RE4	LA4	MI5
Frecuencia	Original:	195.56	293,34.	440	660
(Hz)	Análisis:	195.75	293.50	440.25	660.25

Figura 5.3: Tabla comparativa zerocrossing

- Detección de frecuencia de tonos puros mediante análisis en frecuencia: Densidad espectral en potencia (DEP)

la función depmod es una función de análisis densidad espectral de potencia, modificada para que calcule la frecuencia fundamental y su potencia sonora en dB (en el caso en que tenga un máximo de potencia correspondiente a f_0 en el espectro) encuentra el máximo y devuelve la posición que se corresponde con la frecuencia.

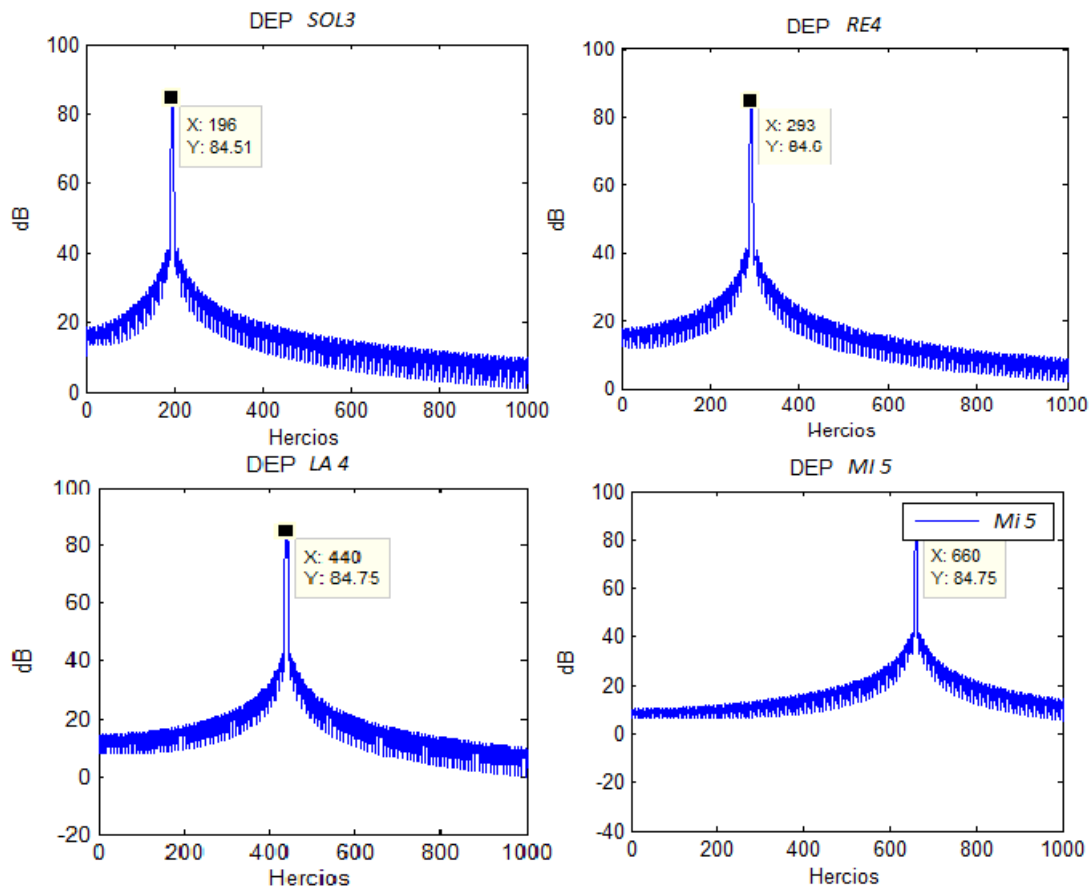


Figura 5.4: Detección (F0) DEP

los resultados de la detección son los siguientes:

		SOL 3	RE4	LA4	MI5
Frecuencia (Hz)	Original:	195.56	293,34.	440	660
	Análisis:	196	293	440	660

Figura 5.5: Tabla comparativa MAX DEP

Añadiendo 2 parciales armónicos a el tono generado y analizando mediante DEP, obtenemos la siguiente gráfica

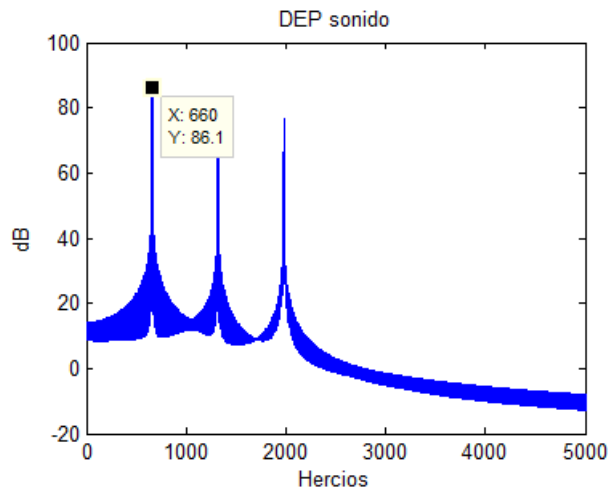


Figura 5.6: Densidad espectral de potencia MI5 (2 armónicos añadidos)

- Detección de frecuencia de tonos con 2 armónicos añadidos, mediante máx, HPS (1 submuestreo)

Implementación del algoritmo HPS (Harmonic Product Spectrum) en matlab.

Esta última prueba discrimina más eficientemente la frecuencia fundamental, y ha sido la solución elegida por otros trabajos con PDA

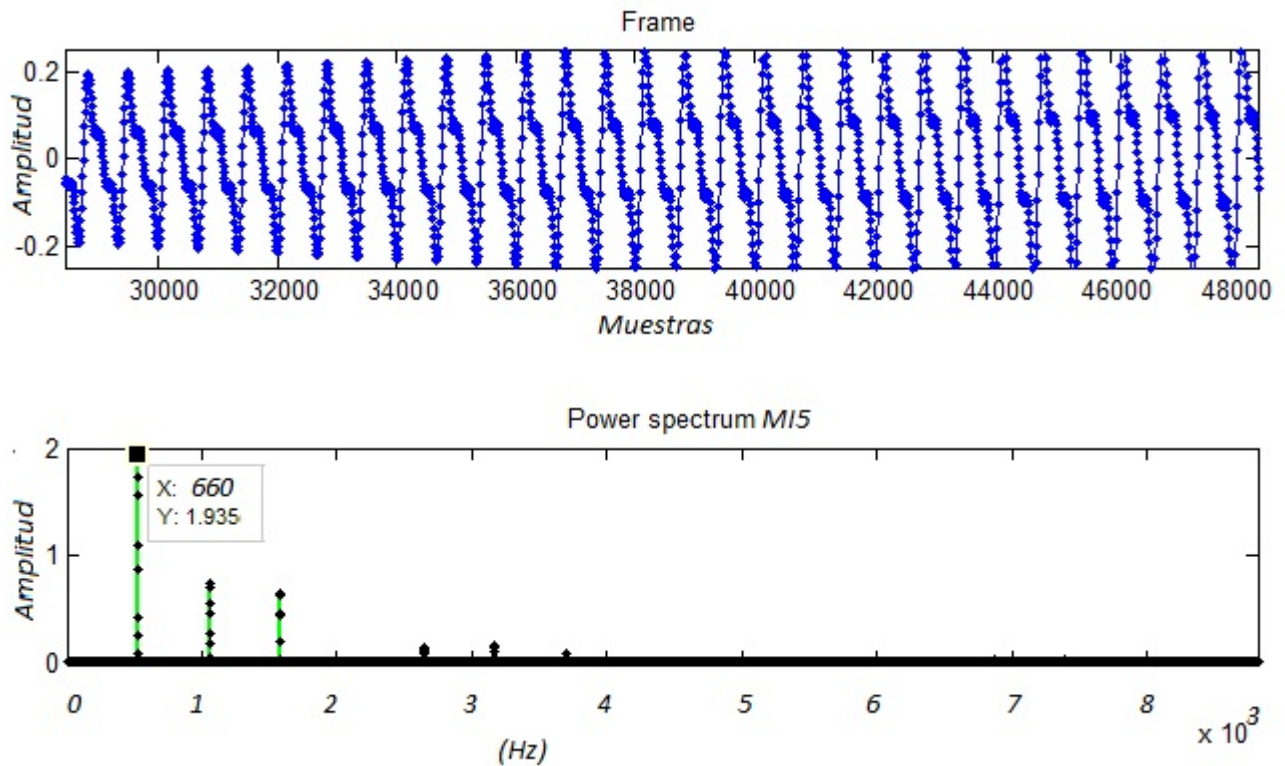


Figura 5.7: Detección (F0) HPS MI5

los resultados de la detección son los siguientes:

		SOL 3	RE4	LA4	MI5
Frecuencia (Hz)	Original:	195.56	293,34.	440	660
	Análisis:	196	293	440	660

Figura 5.8: Tabla comparativa HPS

HPS finalmente será una de las dos técnicas que utilizaremos, teniendo en cuenta que en lenguaje CSound existe un opcode que implementa esta técnica en frecuencia de detección de tono, opcode: ptrack.

5.3. Implementación en CSound

En esta sección se detallará la implementación del algoritmo en lenguaje CSound, utilizando el front-end CsoundQt.

5.3.1. Estructura del algoritmo

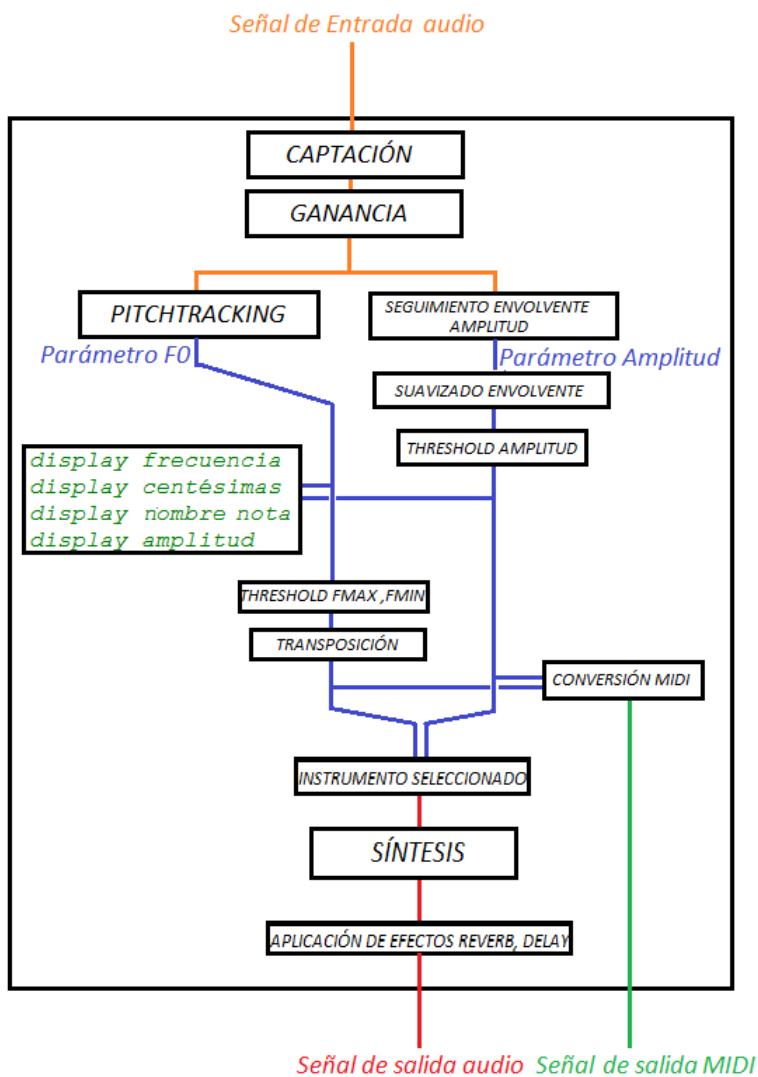


Figura 5.9: Esquema de la estructura del algoritmo

Como se puede observar en el gráfico anterior, la estructura del algoritmo es bastante simple, a continuación describiremos las 3 fases principales del proceso.

5.3.2. Fases del proceso

- Fase de captación y análisis de la señal de audio de entrada:

La señal que proviene del violín electroacústico es recibida y leída en tiempo real por CSound mediante el opcode **inch**, al que se le especifica el canal de entrada y se asigna un nombre para su salida, en nuestro caso se ha llamado *al*

En caso de necesitar amplificación adicional, se aplica una ganancia constante a la señal de entrada para que el sistema pueda trabajar en un rango aceptable de valores de amplitud.

A continuación se extrae la envolvente de amplitud de la señal mediante el opcode **follow2**, al que hay que especificar los parámetros de tiempo de ataque y decaimiento máximos, a la salida del opcode obtenemos el parámetro *amp*, que se utilizará posteriormente como control de amplitud.

Seguidamente se procede a la detección de tono con opción a elegir técnica de pitchtracking los opcodes **ptrack** y **pitchamdf**, que implementan HPS (Harmonic Product Spectrum) y AMDF basada en la autocorrelación (Average Magnitude Difference Function) respectivamente, estas funciones requieren parámetros de entrada como: tamaño de enventanado (window size), solapamiento (overlap), rango de frecuencias estimadas, y algunas otras variables. El funcionamiento de estos dos opcodes y los valores de los parámetros de entrada están detallados en posteriores secciones.

A la salida tanto de *ptrack* como de *pitchamdf*, se obtienen variables de frecuencia fundamental en hercios y de amplitud en decibelios (full-scale 0dB), estas variables son de tipo control (*k*), y se actualizan cada *sr/hopsize* (siendo *hopsize* el solapamiento = $\frac{1}{2}$ del tamaño de la ventana de análisis), les hemos llamado *kf* y *ka* respectivamente.

Algunos instrumentos funcionan mejor con variables de amplitud de tipo control, y otros con tipo audio (*a*) es por esto que nos interesan dos parámetros diferentes que expresan la misma magnitud.

La forma de la envolvente de la señal de amplitud, se suaviza mediante el opcode **port** para evitar ruidos impulsivos.

Después de obtener estos parámetros esenciales, y sólo si la amplitud detectada en ese instante supera un umbral establecido desde el interfaz, (Puerta de ruido), se relaciona el valor de frecuencia obtenido con el nombre de la nota (en notación inglesa) contenido en una cadena de caracteres, y sus centésimas de tono (afinación precisa), para poder imprimirlo por pantalla mediante objetos tipo `display` dispuestos en la sección de análisis del interfaz gráfico.

Se imprime por pantalla el nombre de la nota, las centésimas de tono, la amplitud (*ka*) en dB y la frecuencia en Hz, de esta forma podemos hacer una monitorización de las señales de control y comprobar el funcionamiento de la fase de análisis.

- Fase de adecuación y filtrado de los parámetros de control (F0 y amplitud):

En esta fase, los parámetros de control sufren una adecuación para limitar posibilidades de error o para conseguir un efecto sonoro determinado.

La señal de control de frecuencia se filtra para establecer un intervalo entre frecuencia máxima y mínima (ancho de banda de trabajo controlado) Con valores seleccionados por el usuario desde el interfaz (sección de control), de esta forma evitamos errores de detección producidos por posibles interferencias.

También podemos transportar el parámetro kf a la octava deseada y seleccionada desde la sección de opciones del interfaz, de esta forma la frecuencia del sonido a la salida no está restringida al registro propio del instrumento conectado, el resultado es muy interesante para el desarrollo creativo-musical.

Otra modificación de la señal de control kf puede ser el suavizado de las variaciones bruscas en frecuencia, utilizando el opcode **port**, que produce un efecto de glissando o portamento más o menos acusado dependiendo del valor establecido para el factor de suavizado. (parámetro de entrada del opcode)

Se permite aplicar una ganancia determinada a la señal de control de amplitud ka , obteniendo valores de más adecuados si se precisa.

- Fase de síntesis y efectos y/o conversión a eventos MIDI:

Mediante varios switches, se pueden activar tanto el bloque de conversión MIDI como el bloque de síntesis, dependiendo de las necesidades del intérprete.

Conversión MIDI: Para la correcta conversión de las señales de control en una secuencia de eventos MIDI, hay que adecuarlas al entorno.

Los valores que va tomando la variable kf , son trasladados al rango MIDI (0-127), mediante la ecuación:

$$knota = 69,5 + \left(12 \cdot \frac{\log_{10}\left(\frac{kf}{440}\right)}{0,301}\right)$$

En los apéndices se encuentra la tabla con los valores MIDI relacionados con las distintas notas musicales.

Lo mismo sucede para los valores de amplitud ka , que se escalan al intervalo correcto mediante el opcode **scale**, a la salida del opcode obtenemos el valor traducido y se almacena en una variable que hemos llamado $ksc1$

Una vez adecuados los parámetros necesarios y sólo si el botón de activación MIDI está seleccionado, se procede a la conversión MIDI utilizando el opcode **midion**. Cabe mencionar que este opcode no envía nunca mensajes tipo NOTE OFF, sino que detiene los mensajes NOTE

ON cada vez que cambia algún parámetro de entrada.

La señal MIDI se envía a la salida de CSound y puede ser recibida por cualquier otro software de edición musical a través de un enlace software llamado LoopMidi explicado en anteriores capítulos. Esta señal puede utilizarse como señal disparadora o trigger en instrumentos VST alojados en aplicaciones externas, o para generar una partitura tipo piano-roll, escritura musical convencional, etc.

Síntesis y efectos

Síntesis: Los diferentes sintetizadores software programados reciben las señales de control *ka*, *kf* y *amp* como parámetros de entrada en el caso de que esté activado el bloque de síntesis y seleccionado un instrumento en concreto.

Dependiendo del instrumento seleccionado en el menú del interfaz se emplea una técnica de síntesis u otra, cada opcode de síntesis recibe los parámetros de entrada mencionados anteriormente así como diversas variables que controlarán su funcionamiento como pueden ser: número de tabla de onda (Almacenada previamente en la sección de partitura del documento .csd), Tiempo de ataque, entre otras.

A continuación se van a describir los diferentes instrumentos sintetizadores software implementados:

- Oscil1 y Oscil2

2 Instrumentos que contienen osciladores simples con diferentes tablas de ondas, implementados mediante el opcode **oscili**, son un ejemplo de síntesis por tabla de ondas.

- Vco

Instrumento que utiliza el opcode **vco** oscilador de modelado analógico con banda limitada que produce un sonido rico en parciales armónicos.

- Pluck

Incluye el opcode **pluck**, basado en el algoritmo de Karplus-Strong, un ejemplo de síntesis por modelado físico.

- Organ

Emulación del sonido tipo órgano, se basa en la combinación de 9 osciladores simples (**oscil**); síntesis aditiva.

- Buzz

Mediante el opcode **buzz** se generan una serie de señales coseno, cuyas frecuencias son parciales armónicos de F0, sonido muy rico en contenido espectral; síntesis aditiva.

- Woodwind

Pretende emular el sonido de instrumentos de la familia viento-madera, utiliza el opcode **wgclar**; síntesis por modelado físico.

- Bell

Pretende emular el sonido de campana tubular, utiliza el opcode **fbell** ; síntesis por modulación en frecuencia (FM).

- Percflute

Produce un sonido similar al efecto picado de una flauta travesera, utiliza el opcode **fmpercfl**; síntesis FM.

- Voice

Simple imitación del sonido de la voz humana, utiliza el opcode **fmvoice** ; síntesis FM.

La aplicación está abierta a la inclusión de nuevos y más complejos instrumentos sintetizadores software.

Efectos: El sistema también implementa los efectos de Reverberación y Delay (Eco), que son globales e independientes, los efectos activados se aplican a la señal de salida de los diferentes instrumentos.

- Reverberación

La reverb implementada utiliza el opcode **reverb2**, un algoritmo consistente en un reverberador que combina 6 filtros paso-bajo en paralelo y 5 retardos.

Este efecto se activa desde la sección de efectos del interfaz gráfico, donde se disponen objetos tipo potenciómetro, que hacen posible el control de los parámetros básicos del efecto desde el interfaz.

Los 3 parámetros controlables son: Tiempo de reverberación, Mezcla (wet-dry), e índice de difusión en altas frecuencias.

- Delay

El efecto de delay o eco implementado utiliza los opcodes: **delayr**, **deltap** y **delayw**, que proporcionan un sistema de escritura y lectura de la señal en una línea de retardo.

De este efecto se pueden controlar 3 parámetros desde el interfaz: Tiempo de retardo (en segundos), Mezcla y nivel de realimentación (feedback).

Finalmente la señal de audio generada en la fase de síntesis y efectos es enviada a la salida de la tarjeta de sonido, utilizando en nuestro caso el controlador de audio ASIO4ALL el cual reduce considerablemente la latencia comparado con otros controladores convencionales.

5.3.3. Interfaz

La aplicación front-end CsoundQt posibilita la creación de un interfaz gráfico o widget desde el cual el usuario controla el funcionamiento del algoritmo de una manera visual e intuitiva.

En CsoundQt, la comunicación Interfaz-Algoritmo y viceversa se puede lograr de diferentes formas, que pueden producir mayor o menor carga del sistema, dado que interesa que el sistema sea ligero, la solución elegida para optimizar el funcionamiento ha sido la creación de un bus software con diferentes canales al que tienen acceso ambas partes, así como la utilización de un único display genérico, donde se muestran según se vayan modificando, los valores de los diferentes parámetros de entrada al algoritmo.

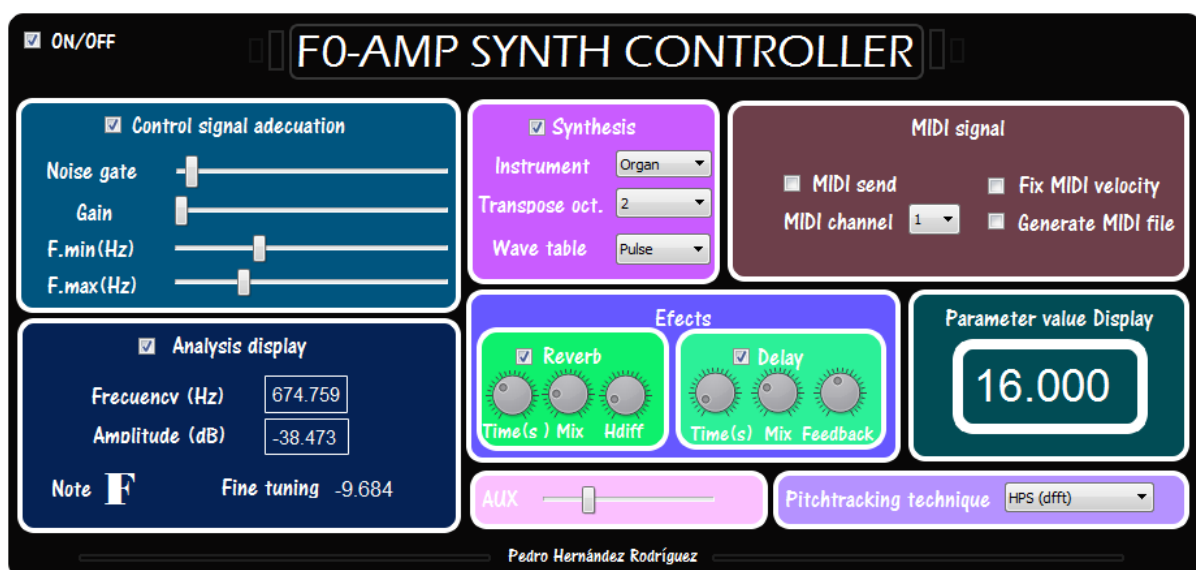


Figura 5.10: Interfaz gráfico *FO-AMP*

En el interfaz gráfico se distinguen diferentes bloques que se detallarán a continuación:

Bloque de adecuación De la señal: En el cual el usuario puede controlar el valor del umbral de la puerta de ruido, la pre-amplificación de la señal, y los límites máximo y mínimo de valor

de frecuencia estimada.

Bloque de displays de análisis: En este bloque se visualizan los parámetros fundamentales de la señal de control como son: Frecuencia, Amplitud, Nota musical y afinación precisa (Centésimas de tono).

Bloque de síntesis: En el cual se elige entre los diferentes instrumentos, se determina la transposición y se selecciona la tabla de ondas en el caso de que el instrumento seleccionado lo requiera.

Bloque de efectos: En esta sección se pueden activar o desactivar, y controlar los parámetros básicos de los efectos Reverberación y Delay.

Bloque MIDI: En el cual se puede activar el envío MIDI, seleccionar el canal MIDI por el cual se va a enviar y varias opciones más.

Se dispone de un objeto tipo slider que se utilizará como controlador de parámetros auxiliares en el caso de que el instrumento lo requiera.

También se puede elegir entre 2 técnicas de Pitchtracking DFFT (HPS) y Autocorrelación (AMDF), (Las 2 que mejor resultado han dado).

5.3.4. Restricciones del sistema

Debido a la necesidad de trabajar en tiempo real, se ha optado por un funcionamiento **monaural**, es decir, tanto la lectura de la señal de entrada como la escritura de la señal de salida se realizan en un solo canal de audio, de esta forma se reduce prácticamente a la mitad el coste computacional, lo que se traduce en una respuesta con menor latencia. El resultado es una señal de audio tipo mono.

El sistema no es capaz de detectar **señales polifónicas**, produciendo errores de detección cuando se envían por la señal de entrada sonidos reproducidos simultáneamente, por lo tanto no soporta acordes, sólo melodías.

5.3.5. Adecuación de parámetros

Para que el algoritmo implementado funcione correctamente, es decir, para que produzca los mínimos errores de detección, la mínima latencia, y genere una señal de audio sin interrupciones ni ruido adicional, se han tenido que adecuar algunos parámetros globales y específicos, para conseguir el equilibrio LATENCIA-CALIDAD de respuesta del sistema.

- Parámetros globales

Determinación de la frecuencia de muestreo sr : En el archivo .csd y más concretamente en la cabecera de la sección de instrumentos, hay que especificar la frecuencia de muestreo sr (sample rate) a la que se desea trabajar. La frecuencia de muestreo se define como el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta. La unidad utilizada es el hercio. En nuestro caso se ha decidido establecer 44100 Hz.

Por una parte se ha de cumplir el criterio de Nyquist.

El violín puede generar sonidos cuya frecuencia fundamental máxima es de 3500 Hz aproximadamente, en el caso de que solo hubiera que registrar esa señal la frecuencia de muestreo mínima sería 7000 Hz aproximadamente, ya que no interesa precisamente la captación de parciales armónicos habitualmente superiores a la fundamental.

En el caso de la generación del sonido las frecuencias máximas de los sonidos sintetizados pueden ser mucho mayores contemplando los parciales armónicos.

Como el espectro audible tiene su límite máximo en 20 kHz, la frecuencia de muestreo convencional de 44.1 que implica calidad similar al audio en compact-disc o CD, es más que suficiente para el correcto funcionamiento del sistema, la variación de este parámetro puede resultar en una variación de la latencia obtenida debido a su relación con el parámetro $ksmps$, periodo en muestras de la actualización de las diferentes variables de control (tipo k).

Frecuencia de refresco de datos en las señales tipo k , $ksmps$: Se trata del periodo en muestras de la actualización de las diferentes variables de control (tipo k). Dependiendo de la carga de CPU que se experimente en el momento de activar el algoritmo se han establecido diferentes valores para dicho parámetro con la finalidad de optimizar la latencia y asegurar una señal de audio continua y sin ruidos adicionales. es conveniente que los valores adoptados sean potencia de 2.

En el equipo del cual se dispone, se han obtenido buenos resultados en un intervalo de 8 a 64 muestras por ciclo de control, resultando una mayor latencia mientras mayor sea el ciclo.

Tamaño de los buffer Hardware y Software: Para lograr la latencia más baja posible sin rupturas de audio, tienen que ajustarse los valores de estos dos buffer. El resultado final dependerá directamente del sistema, y también de la complejidad de los cálculos realizados en el algoritmo.

Para obtener un buen equilibrio CALIDAD-LATENCIA es recomendable seguir el siguiente procedimiento:

Una vez establecido el valor de *ksmps* que produce un buen funcionamiento, hay que ir dando valores potencia negativa de dos al tamaño del buffer interno o software, abreviado como *-b*, comenzando con valores pequeños como -1,-2,-4... y sucesivamente hasta que se comprometa la continuidad de la salida de audio. El valor real de *-b* será el valor absoluto de $-b \cdot ksmps$.

A continuación reducir el tamaño predeterminado del buffer Hardware abreviado como *-B*, (1024 comúnmente) dividiéndolo sucesivamente entre 2, hasta que se comprometa la continuidad de la salida de audio o aparezcan ruidos tipo glitch.

Se ha experimentado con distintos tamaños de buffer y se ha determinado que para el equipo existente, la combinación óptima de valores de tamaño que aseguran calidad y baja latencia es de $-b = -2$ y $-B = 6$ muestras.

- Parámetros específicos

Dependiendo del registro del instrumento electroacústico controlador, se establecerán unos determinados valores a los parámetros siguientes:

Tamaño de enventanado: la señal a analizar se fragmenta en bloques de un determinado tamaño en muestras.

En la fase de pitchtracking se establece el tamaño en muestras del solapamiento entre los bloques analizados mediante el parámetro *ihopsize*, el opcode **ptrack** define por defecto el tamaño de enventanado como: $window\ size = 2 \cdot ihop\ size$, lo cual produce un solapamiento del 50 por ciento.

De tal forma que el tamaño en muestras del solapamiento indicado por el usuario determinará el tamaño de enventanado.

Hay que plantearse algunas consideraciones para determinar el tamaño óptimo de enventanado, por ejemplo el periodo máximo de la señal que se puede detectar, es decir, la frecuencia mínima que se prevé (195Hz en el caso del violín), para que se tengan suficientes muestras y pueda ser captado adecuadamente.

$$T_{max} = \frac{1}{f_{min}} = \frac{1}{195} = 5,12\text{milisegundos}$$

Para que el análisis sea correcto se necesitarán un mínimo de 4 periodos de oscilación por bloque, por lo tanto el tamaño de enventanado mínimo debe ser igual o superior a 4 veces el periodo máximo.

$$T_{max\ window} = 4 \cdot T_{max} = 4 \cdot 5,12 \cdot 10^{-3} = 20,51\text{milisegundos}$$

Que multiplicado por la frecuencia de muestreo resulta el tamaño mínimo de enventanado

$$20,51 \cdot 10^{-3} \cdot 44100 \approx 905\text{muestras}$$

Como hemos dicho anteriormente, el tamaño de enventanado es $ihop\ size \cdot 2$ por lo tanto, y teniendo en cuenta que este valor debe ser potencia de 2, el valor mínimo de *ihopsize* que garantiza la detección de la frecuencia mínima esperada es de 512 muestras.

Resulta un tamaño de enventanado: **window\ size = 1024** muestras.

Frecuencia máxima y mínima Con el fin de limitar los errores de detección producidos por ruidos interferentes en la señal, o por armónicos de altas frecuencias con demasiada potencia, a esta señal de control se le aplica un filtro paso-banda, cuyas frecuencias límite máxima y mínima se determinan desde la sección de adecuación en el interfaz mediante objetos tipo *slider*.

La frecuencia mínima obviamente se establecerá alrededor de 195 Hz y la máxima puede variar dependiendo de las necesidades del músico o las características de la obra musical interpretada. Valores por defecto: $f_{min} = 190Hz$ y $f_{max} = 2000Hz$.

Capítulo 6

Construcción de un violín electroacústico

6.1. Introducción

La luthería o lutería es un arte que consiste en la construcción de instrumentos musicales, principalmente cordófonos compuestos -de cuerda, con caja de resonancia y mástil. Al artesano se le denomina luthier o lutier.

La luthería alcanza la categoría de ciencia, no en vano quienes la practican dedican gran parte de su tiempo a investigar sobre nuevos métodos de trabajo, técnicas de elaboración, materiales y resultados sonoros. Se trata de optimizar el trabajo y conseguir la excelencia musical que todo artista ansía.

Un buen luthier bebe conocimientos de la organología -ciencia que estudia la morfología de los instrumentos musicales- y tiene en cuenta en sus creaciones factores físicos, acústicos y químicos.

La construcción de instrumentos musicales que produzcan sonidos con relativa calidad implica un proceso de larga duración, complejo y costoso.

Sin embargo, teniendo en cuenta que la calidad del sonido (Espectro relativamente rico en parciales) producido por el instrumento controlador no es relevante para el buen funcionamiento del sistema, siendo tan sólo la clara presencia de la vibración fundamental el requisito más importante de la señal de entrada, el proceso de construcción del instrumento de control, un violín electroacústico en nuestro caso, resulta mucho más fácil e implica un coste mucho menor.

La consideración anterior es importante si en un futuro, se decidiera comercializar un producto basado en el sistema propuesto.

6.2. Cálculos y medidas básicas

Con el fin de conseguir una interpretación cómoda y una morfología similar a la de un violín acústico, se han respetado las medidas básicas detalladas de un manual de luthería (Manual del luthier; Tratado práctico sobre la construcción de violines. de Ramón Pinto Comas).

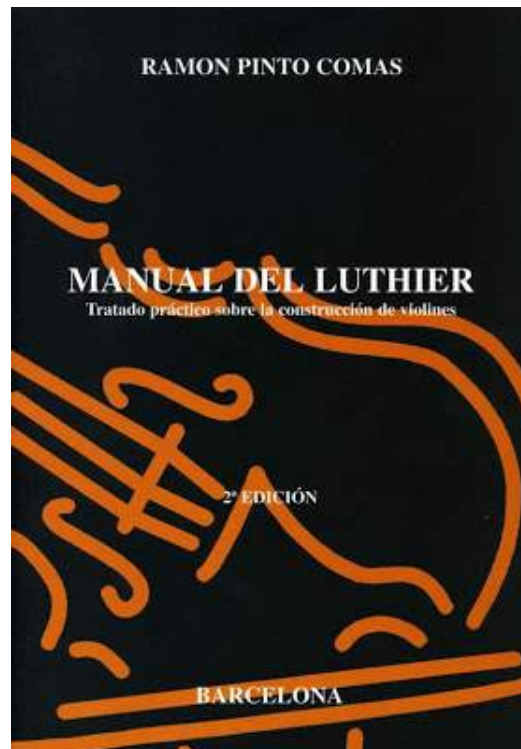


Figura 6.1: Portada del tratado práctico de luthería

Un violín para que pueda ser utilizado por un violinista precisa que sea construido según unos cánones invariables en sus medidas y dimensiones, pues prácticamente todas las variaciones posibles han sido ya comprobadas y experimentadas.

Únicamente y en nuestro caso podemos determinar la forma de la estructura análoga a la caja de resonancia, a nuestro gusto, así como diseñar el sistema captación del sonido por tratarse de un violín no convencional.

A continuación se detalla el diseño y medidas fundamentales del prototipo.

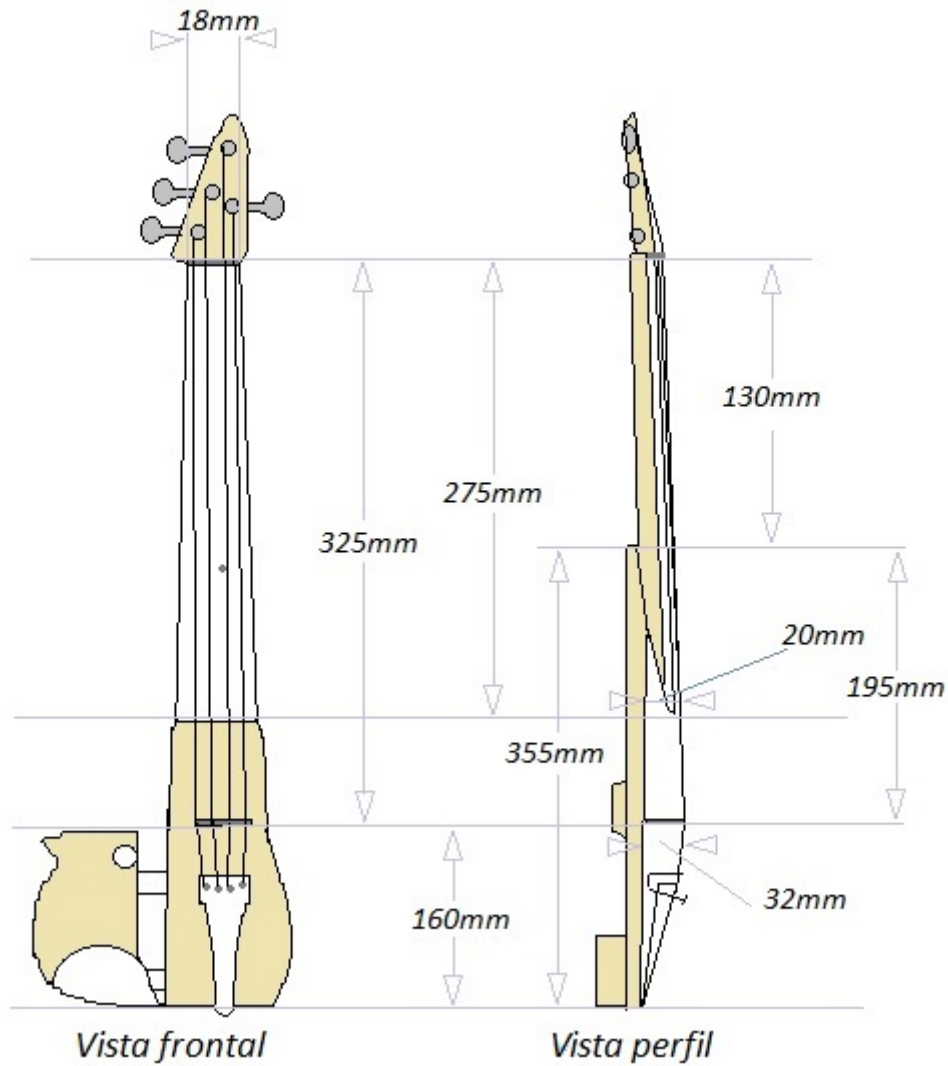


Figura 6.2: Diseño y medidas del prototipo

Medidas:

- Largo de la caja (bloque estructural 1): 355mm
- Desde el borde superior del bloque1 hasta el punto donde se coloca el puente: 195mm
- Desde el puente hasta el borde inferior del bloque1: 160mm
- El mango desde la cejilla hasta el inicio del bloque1: 130mm
- Largo del batedor (Diapasón): 275mm
- Altura del batedor sobre el bloque1: 20mm
- Altura del puente: 32mm
- Ancho de la cejilla desde la 4ª cuerda hasta la 1ª: 18mm
- Cuerda vibrante, desde cejilla hasta puente: 325mm

6.3. Detalle del proceso de construcción

Una vez claro el diseño y las medidas básicas, se selecciona el material con el cual se va a construir el cuerpo del instrumento, he optado por la madera de pino, considerando propiedades de ligereza y facilidad de tallado y trabajo en general. Para instrumentos más grandes como violas o violonchelos habría que considerar la resistencia a la tensión del material, pero en el caso del violín, debido a sus reducidas dimensiones, se prevé factible en madera-pino.



Figura 6.3: Muestras de madera de pino empleadas

A continuación se traslada la forma diseñada a los diferentes bloques de madera, es conveniente dejar alrededor de 1 o 2 cm de sobrante, siempre hay tiempo para perfilar o lijar.



Figura 6.4: Bloques 1 2 y 3

Seguidamente se cortan los bloques con la forma deseada mediante una sierra eléctrica y se lijan hasta conseguir una silueta semi-definida.



Figura 6.5: Bloques cortados y lijados

La unión del bloque 1 con el bloque 2 es uno de los procesos más críticos de la construcción, ha de ser muy sólida. Basándonos en técnicas de luthería de trabajo con clavos de madera, se resolvió solucionar el problema mediante la inyección a presión y rotación de 5 varillas gruesas de madera de más densidad con la ayuda de un taladro eléctrico. El resultado ha sido más satisfactorio de lo esperado.



Figura 6.6: Clavos de unión

Una vez eliminado el sobrante de los clavos de unión, se colocan las clavijas en la parte superior del bloque 2, en este caso se han utilizado clavijas de guitarra, puesto que el sistema tradicional en los violines se presentaba muy laborioso, dado los limitados recursos del improvisado taller.



Figura 6.7: Clavijas instaladas

El bloque 3 es desmontable para ahorrar espacio en el transporte del instrumento, va unido al cuerpo principal mediante 2 barras introducidas a presión en los orificios tipo muesca que poseen tanto el bloque1 como el 3.



Figura 6.8: Vista trasera

Otro de los elementos cuya instalación implica una considerable complejidad es el batedor, originalmente se talla en madera de ébano, en este prototipo se trata de una lámina de goma dura que adopta la forma del diapasón. Se decidió el uso de este material por su relativa flexibilidad, propiedad que simplifica el proceso de pegado al mango. Aun así se tuvo que idear un sistema de fijación para que secase la cola empleada.



Figura 6.9: Batedor de goma dura

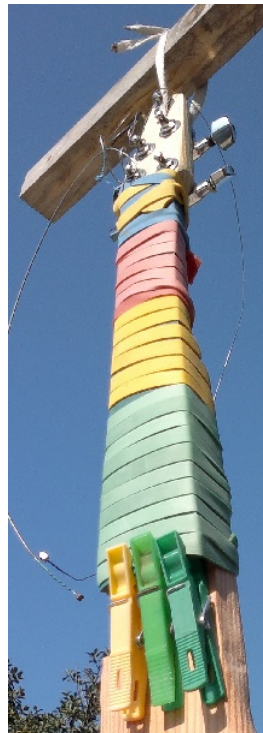


Figura 6.10: Sistema de fijación para un correcto secado y pegado

El secado de la unión se completa a los 2 días transcurridos.

A continuación y tras un largo estudio de las posibles formas de amplificación, se determinó utilizar un sistema análogo a la transmisión de la vibración en los violines acústicos (Detallada

en el capítulo 2 de la presente memoria) utilizando una pieza clave llamada *Alma* una varilla cilíndrica de madera de pino que se determinó colocar debajo de la patilla izquierda del puente, y que atraviesa el bloque1, para ello se necesita realizar un orificio en un punto determinado del bloque1.

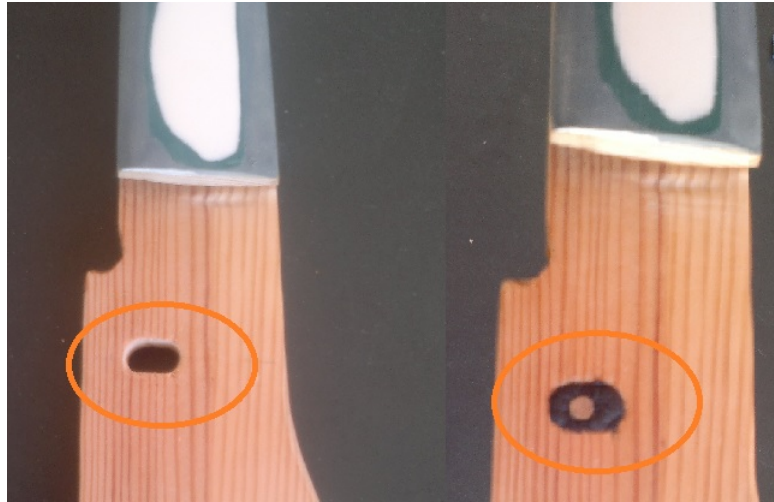


Figura 6.11: Detalle del orificio

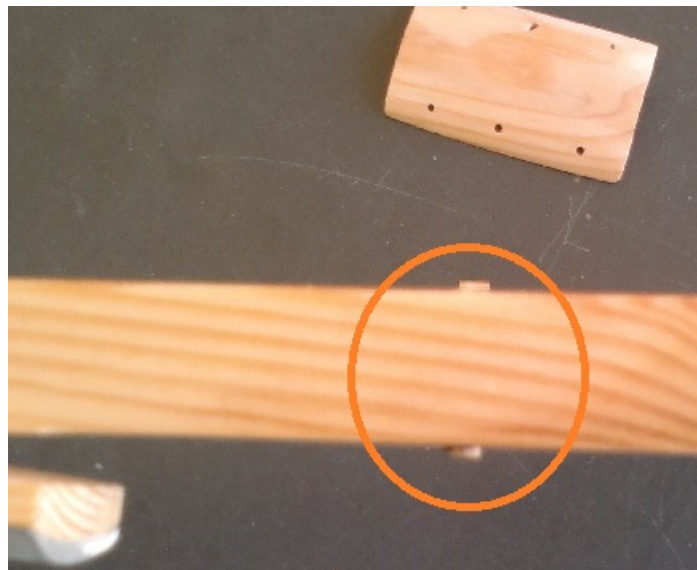


Figura 6.12: Alma atravesando el bloque1

Esta varilla transmitirá la vibración del puente hasta llegar a una placa sujeta por los extremos, que cuya vibración en consonancia con el alma será captada por un micrófono de contacto piezoeléctrico cuyo funcionamiento se ha detallado en el capítulo 2.

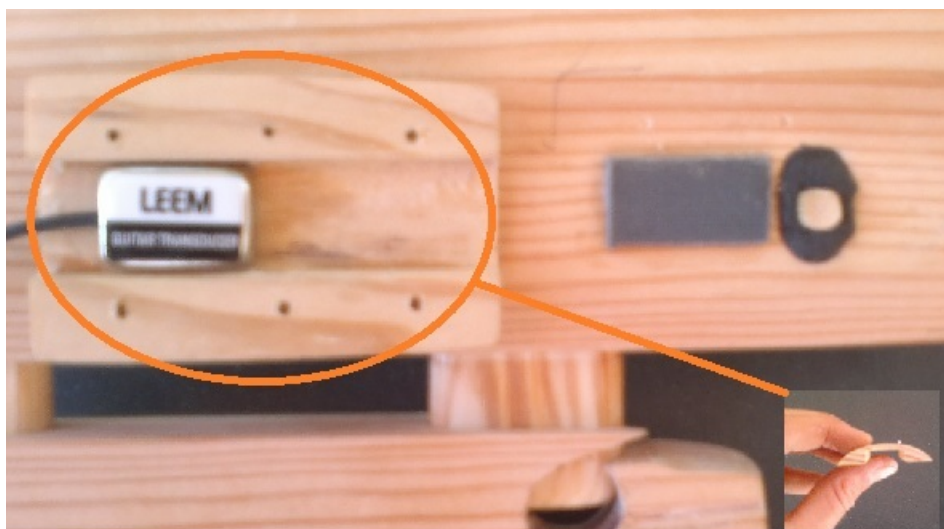


Figura 6.13: Detalle placa y micrófono de contacto

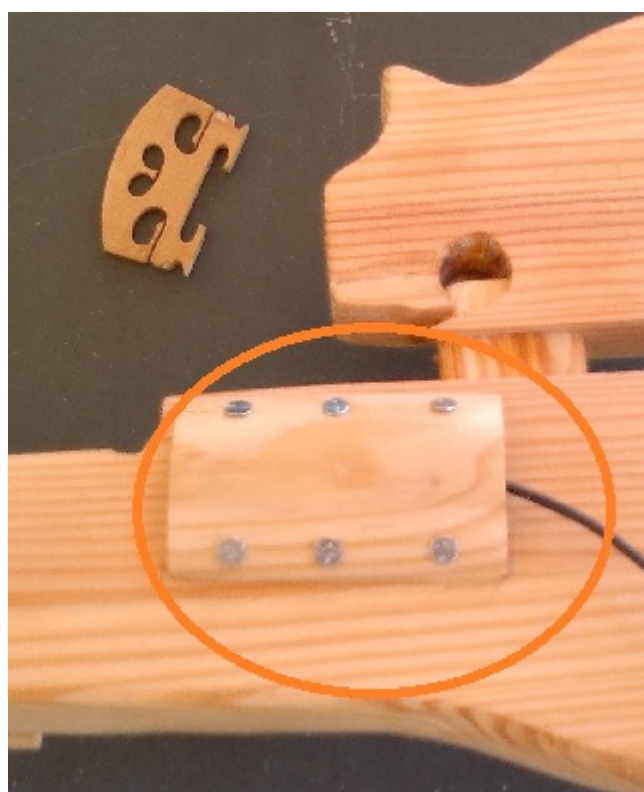


Figura 6.14: instalación sistema de captación

Una vez comprobado un funcionamiento que se ajusta a los objetivos planteados inicialmente, se lija al detalle y se pinta el prototipo. A continuación se montan las piezas complementarias: (Barbada, Cordal, Tensores, Almohadilla, Puente...)



Figura 6.15: pintado

Los resultados sonoros del prototipo son satisfactorios, su manejo es muy similar a un instrumento acústico, su respuesta afinada, la amplitud de la señal generada no es demasiado elevada dadas las características del transductor y el sistema de captación empleado, este problema se puede solucionar con un circuito amplificador previo, o bien como se ha comentado anteriormente, aplicando una ganancia sobre la señal de audio registrada desde el software.

Capítulo 7

Test del sistema

En este capítulo se evaluarán los siguientes aspectos del resultado obtenido al ejecutar el sistema: Errores de detección, Latencia y salida MIDI.

Los sonidos producidos por los sintetizadores software no entrarán en la discusión, por ser subjetiva e irrelevante su evaluación.

Al hacer sonar el sistema, se han obtenido unos resultados muy satisfactorios, una detección de tono bastante precisa por parte de las dos técnicas implementadas y un seguimiento perfecto de la amplitud.

7.1. Errores de detección

Para poder evaluar los errores de detección, se ha obtenido la secuencia de valores que toma la variable de control de frecuencia fundamental, kf , y se ha representado gráficamente.

Se distinguirá entre el modo de interpretación de las notas, ligado o suelto, para posteriormente sacar conclusiones al respecto. A continuación se proporcionan representaciones gráficas que resultan muy útiles a la hora de valorar el funcionamiento del algoritmo.

Escala de SolM interpretado con diferentes técnicas de arco (Algoritmo Ptrack):

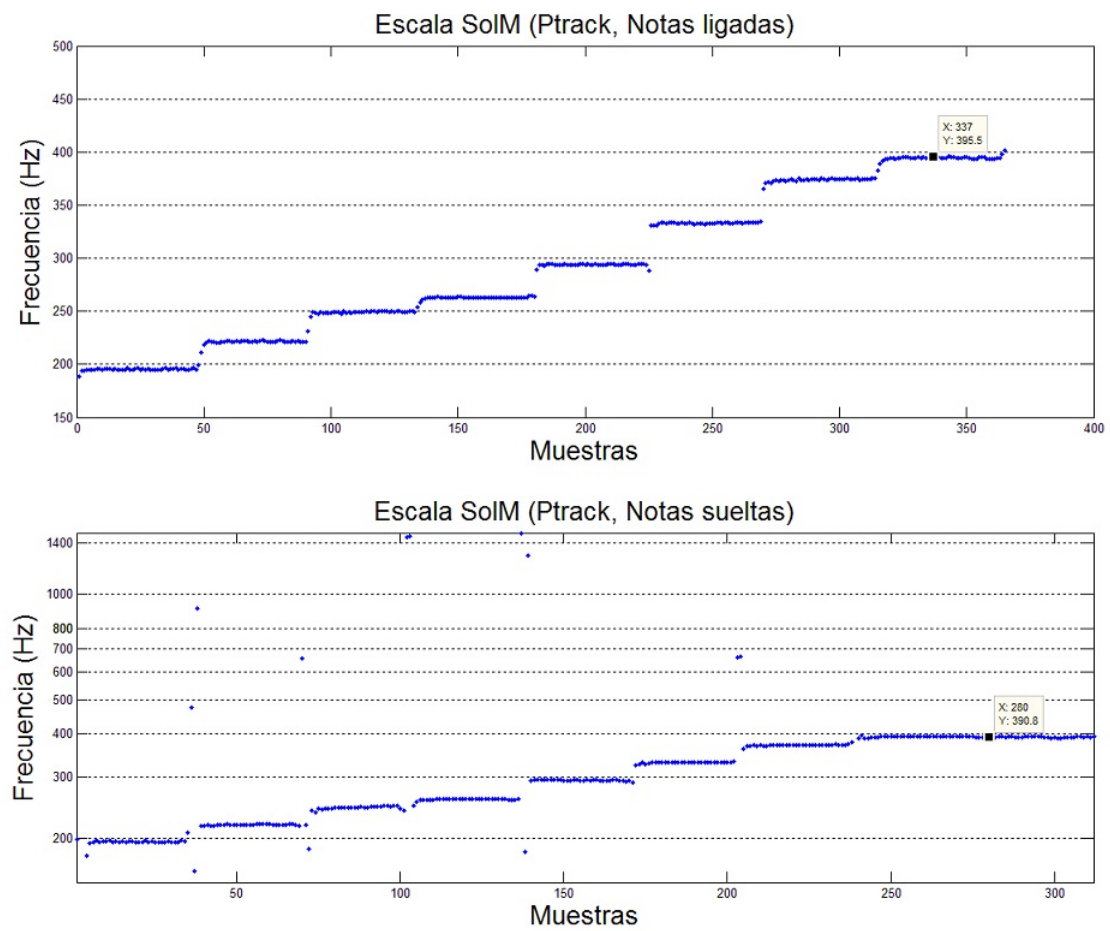


Figura 7.1: Representación respuesta Ptrack a una escala

Como puede observarse, existen muy pocos errores de detección de tono.

Arpegio de SolM interpretado con diferentes técnicas de arco (Algoritmo Ptrack):

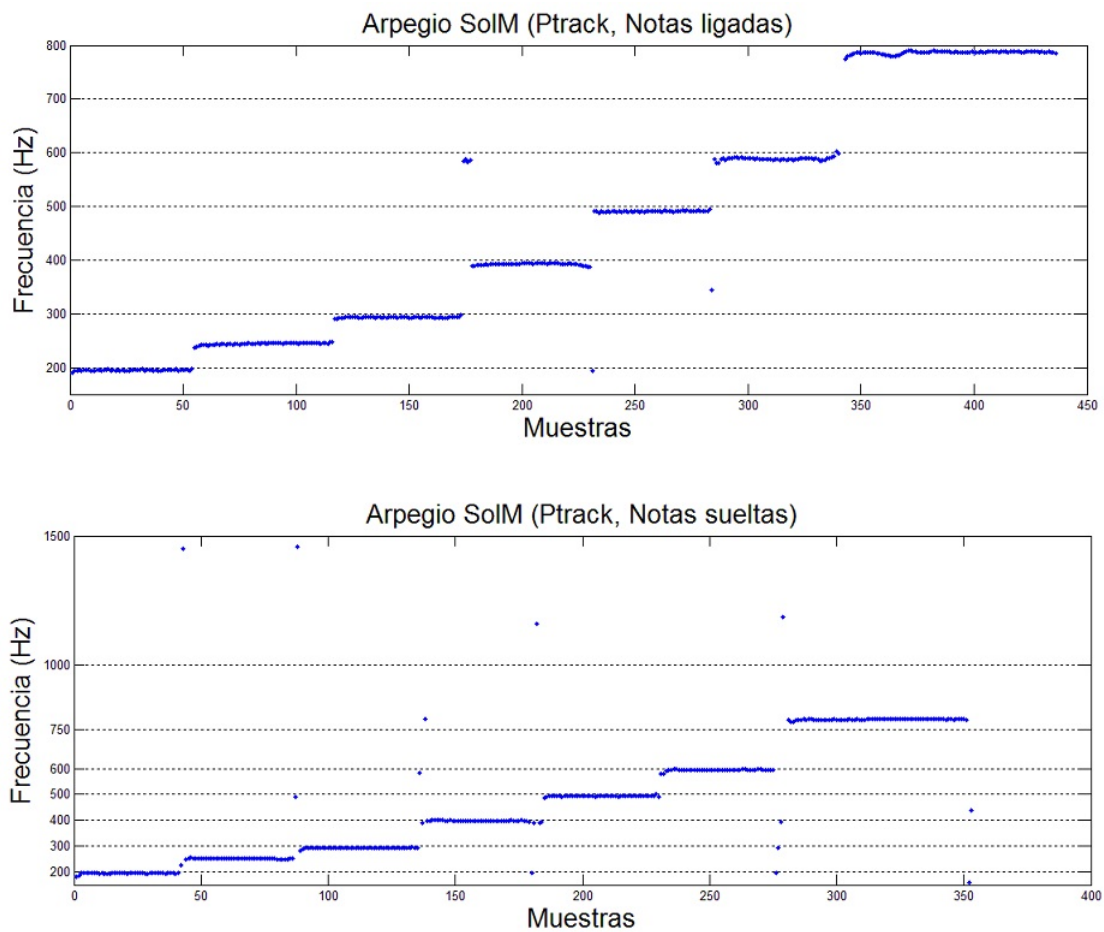


Figura 7.2: Representación respuesta Ptrack a un arpegio

Se puede apreciar en los gráficos anteriores que existen menos errores de detección cuando se interpreta con la técnica de notas ligadas, es decir en el mismo arco.

Cabe mencionar que la frecuencia de muestreo de estas señales no coincide con la del sistema (44100 Hz), sino que se obtiene una muestra cada vez que la señal de control k_f varía.

A continuación se realizan representaciones gráficas de la respuesta a una melodía por parte de los dos algoritmos utilizados.

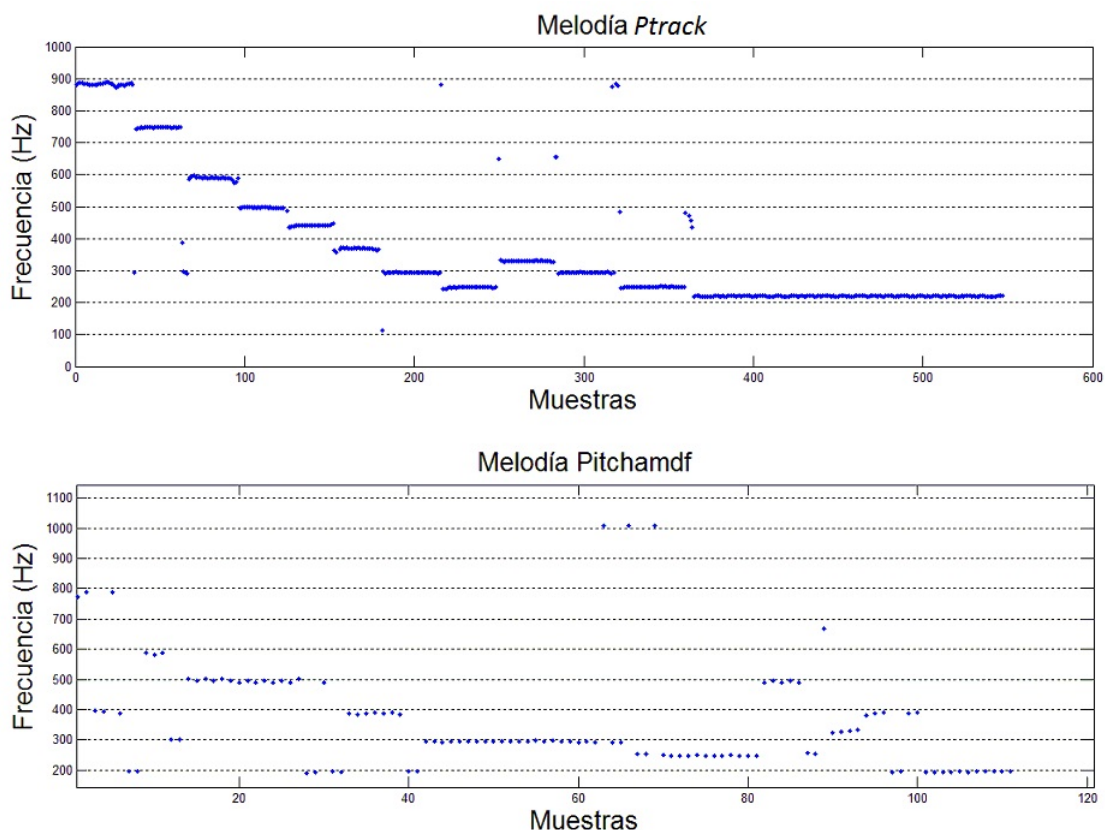


Figura 7.3: Diferentes melodías con diferentes Técnicas (*ptrack*, *pitchamdf*)

Se puede apreciar que las muestras de salida del algoritmo *pitchamdf* están más distanciadas, luego tiene una resolución temporal menor que *ptrack*, sin embargo el parámetro entre muestras no varía, por lo tanto el resultado es más constante, (Sensación tonal más clara).

También se puede sacar la conclusión de que, *ptrack* suele tener más errores de detección en los instantes transitorios de inicio de nota, *pitchamdf* es más estable en los transitorios, pero estadísticamente falla más en general, se recomienda su utilización en el caso de que se deba interpretar una obra o fragmento con la mayoría de notas no ligadas, o con técnicas percusivas. *stacatto*, *pizicatto* etc.

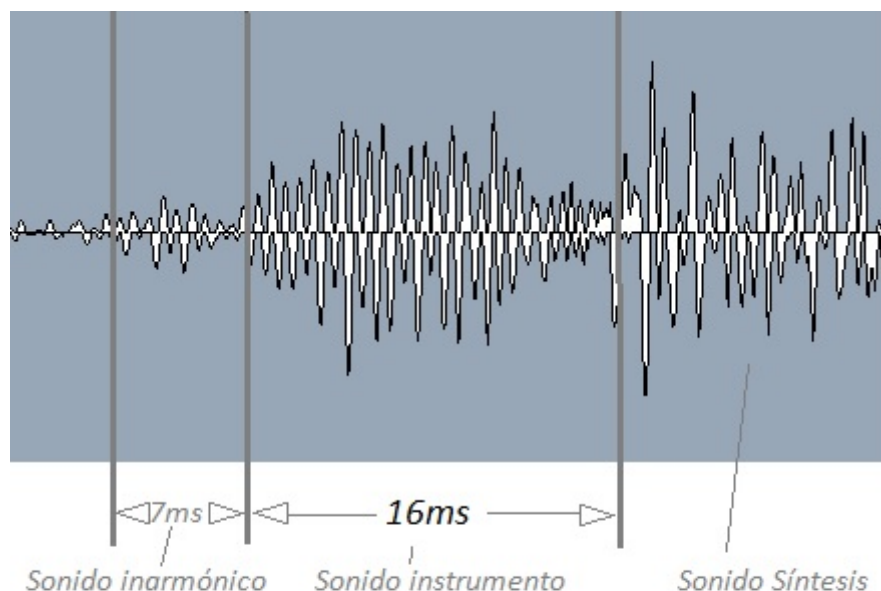
La respuesta sonora en mayoría de los errores de detección en transitorios ha sido atenuada mediante la utilización del parámetro de tiempo de ataque en muchos de los instrumentos implementados, también se puede limitar el rango de error utilizando un intervalo de frecuencias (delimitado por f_{max} y f_{min}) menor.

7.2. Latencia

En esta sección se evaluará la latencia obtenida, tiempo que transcurre desde que se inicia una nota en el instrumento hasta el inicio de su reproducción análoga sintetizada, lo ideal pero utópico sería una latencia de 0 segundos. sin embargo consideraremos aceptable un retraso de menos de 50 milisegundos.

Hay varios factores que pueden intervenir en la obtención de una mayor o menor latencia: el tipo de controlador de audio y el ajuste de sus opciones de compensación de latencia (Se recomienda utilizar Asio4all), el parámetro de actualización de los valores de control, **ksmps** (Número de muestras en un periodo de control), la carga en CPU registrada en ese mismo instante.

Utilizando una grabación que contenga los sonidos producidos por el instrumento y la respuesta sonora del sistema, y midiendo la separación entre el inicio de ambos sonidos con las herramientas de un secuenciador, (Cubase en nuestro caso) Se determinará la latencia obtenida bajo ciertas condiciones de trabajo.



Condiciones de trabajo:

ksmps = 32
sr = 44100 Hz

Window size = 1024 muestras
15% de carga CPU

Figura 7.4: Medición de latencia-condiciones de trabajo

El resultado de la medición es de aproximadamente 16 milisegundos de latencia obtenidos, un valor razonablemente satisfactorio, aunque ya lo era el resultado percibido por el oído, incapaz

de distinguir este intervalo medido.

Dependiendo de: Velocidad de procesador, Tarjeta de sonido, Controladora de audio, etc. Podremos trabajar con un determinado valor del parámetro **ksmps** u otro, mientras más bajo sea, menor latencia se producirá.

7.3. Salida MIDI

Para la evaluación de la señal de salida MIDI del algoritmo se ha utilizado el programa **Ableton live 8**, al cual se le ha enviado dicha señal por medio del software mapeador **loopMIDI**.

Esta señal se visualiza en formato **pianoroll**, para la prueba se ha interpretado un arpeggio de SolM y el resultado es el siguiente:

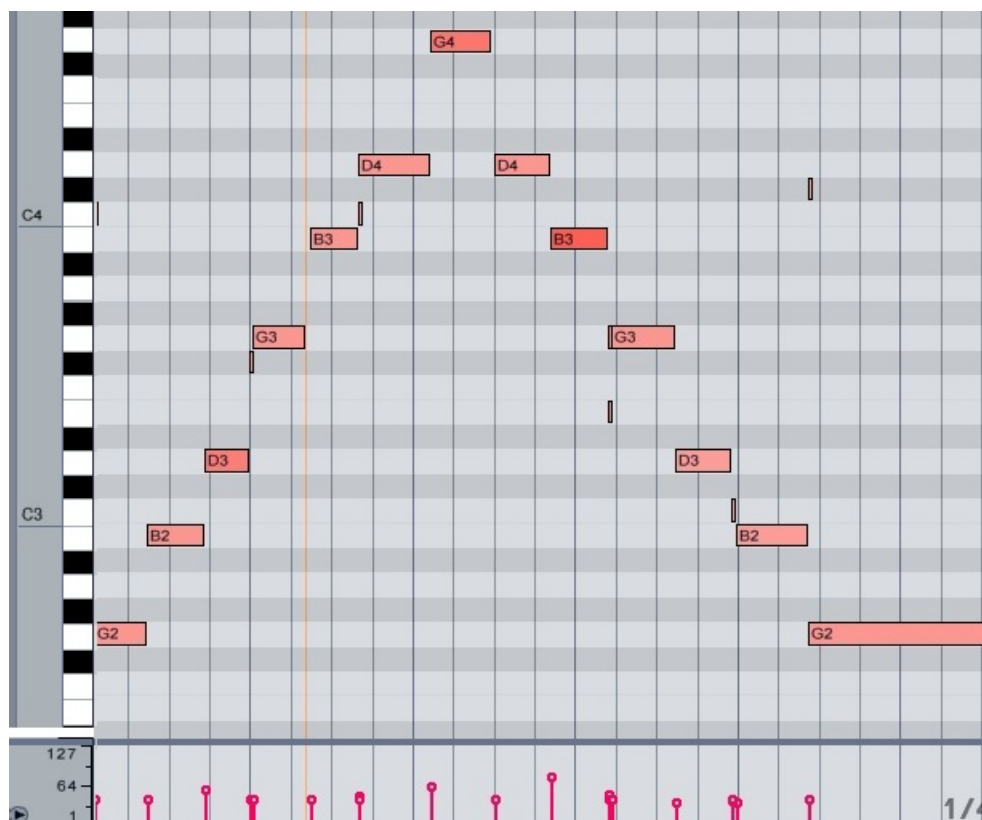


Figura 7.5: Partitura pianoroll de la señal de prueba (arpeggio SolM)

Se pueden apreciar algunos errores de conversión, pero en general el resultado es bastante útil, dado que una vez captada la señal MIDI se puede modificar en el software de edición musical, aplicar cuantización de duraciones y eliminar manualmente los diferentes mensajes erróneos.

Cabe destacar que la latencia del proceso de envío de la señal MIDI es prácticamente nula, sin embargo la captación y posterior reproducción a cargo del software de edición musical puede

producir latencia añadida que depende de la configuración MIDI del software utilizado.

Una vez cuantizada y limpiada la señal de prueba anterior, podemos generar una partitura en notación musical.

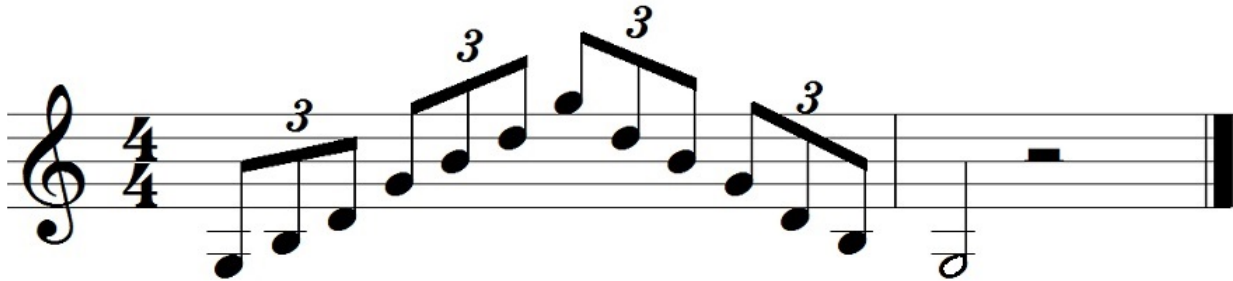


Figura 7.6: Partitura en notación musical de la señal de prueba previamente adecuada

Con un poco de dedicación en tareas de filtrado de errores y cuantización, se puede obtener la transcripción del pasaje interpretado en notación musical, resultado muy interesante en el contexto de la producción musical.

Capítulo 8

Conclusiones y posibles mejoras

■ Conclusiones

Tras haber evaluado el funcionamiento práctico del sistema mediante la audición y más exhaustivamente en el capítulo de test del sistema, el resultado obtenido es muy satisfactorio, se ha conseguido proporcionar una herramienta informático-musical práctica y complementaria que aumenta las posibilidades creativas del músico.

El sistema se ha probado también con diferentes instrumentos controladores (Guitarra eléctrica, Voz humana amplificada, y Saxofón) con resultados muy similares al del violín construido, con menos errores cuando la técnica de interpretación es lo menos percusiva posible.

Los objetivos de control de la síntesis del sonido en tiempo real, y conversión MIDI han sido cumplidos con una considerable precisión, aunque se pueden corregir y mejorar algunos aspectos.

Utilizando el sistema, un violinista por ejemplo, puede interpretar secuencias atribuidas tradicionalmente a otros instrumentos, por ejemplo una línea de bajo, walking bass, un solo de rock, funk, etc. Además permite registrar las diferentes melodías en formato MIDI para un procesamiento posterior, habilitando la posibilidad de transcribir las secuencias registradas a escritura en notación musical.

El sistema implementado puede ser la base esencial de un futuro producto musical y pedagógico más elaborado.

Este proyecto es un claro ejemplo de la utilidad de la informática en el mundo de la música, una fusión interdisciplinar que hoy en día se investiga a fondo produciendo grandes avances, que provocan una vez más en la historia, la evolución de las tendencias musicales.

■ Posibles mejoras

Se me ocurren muchísimas mejoras posibles y complementos para futuros desarrollos del algoritmo, a continuación se citan las que considero más importantes.

El funcionamiento monofónico del sistema es una limitación considerable, en futuros proyectos se tratará de conseguir una respuesta del sistema similar con señales de entrada polifónicas, por medio de bancos de filtros aplicados a la señal y procesado en paralelo.

El diseño del algoritmo propuesto queda abierto a la inclusión de nuevos instrumentos sintetizadores, más complejos y por lo tanto de más calidad, así como la implementación de nuevas técnicas como samplers, vocoders, etc.

Una mejora inmediata puede ser el control de los diferentes parámetros del interfaz (Cambio de instrumento, transposición, efectos, etc.) mediante un dispositivo hardware con comunicación vía MIDI como por ejemplo una pedalera. De esta forma el músico no necesita interrumpir la interpretación para modificar dichos parámetros.

Otra mejora inmediata puede ser la grabación o registro de las secuencias de control (Amplitud y $F(0)$) para su posterior modificación o reproducción.

Para un uso más manejable, el algoritmo puede ser convertido a una aplicación VST por medio del software **cabbage**, de esta forma se podría alojar como un plugin complementario en programas de edición musical.

También puede incorporarse un visualizador que muestre en tiempo real la trayectoria de la frecuencia fundamental con respecto al tiempo, sería este un complemento muy interesante en un contexto pedagógico para la comprensión y asimilación de conceptos musicales (Melodía, intervalos, etc.)

Por último, la idea de poder tocar o ensayar con músicos que se encuentren distanciados, vía internet y sin latencia, es más cercana, dado que la cantidad de información a enviar y recibir ($Amp, F(0)$) es mucho menor que en el caso del audio digital, a día de hoy se han creado algunas aplicaciones que resuelven este problema pero todavía hay mucho por mejorar.

Bibliografía

- J. Wang, S. Wang, W. Chen, K. Chang, H. Chen

“Real-time pitch training system for violin learners”. IEEE International Conference on Multimedia and Expo Workshops, pp. 163-168. (2012)

- Naotoshi Seo

“Pitch detection”. ENEE632 Project4 Part I: March 24, 2008

- Ricardo Alzate, Germán Castellanos.

“Implementación de algoritmos de estimación de pitch en tiempo real sobre procesadores DSP”. Grupo de control y procesamiento digital de señales. Universidad Nacional de Colombia.

- Patricio de la Cuadra, Aaron Master, Craig Sapp

“Efficient pitch detection techniques for interactive music” Center for Computer Research in Music and Acoustics, Stanford University

- Philip McLeod, Geoff Wyvill

“A smarter way to find pitch” University of Otago Department of Computer Science pmcleod, geoff@cs.otago.ac.nz

- Gareth Middleton

“Pitch detection algorithms” Connexions module: m11714 1

- Barry Vercoe

“The canonical Csound reference manual” Version 5.19.0, MIT Media Lab et. al.

- Sergi Jordà Puig

“Audio digital y MIDI”, Guías Monográficas Anaya Multimedia, Madrid 1997

- Stefan Upgard

“Implementation and analysis of pitch tracking algorithms” (Master of Science Thesis Project)
Departamento de Señales, Sensores y Sistemas de KTH, Estocolmo, Suecia.

- Ghulam Muhammad

“Extended Average Magnitude Difference Function Based Pitch Detection” Department of
Computer Engineering, King Saud University, Saudi Arabia 2010

- Ramon Pinto Comas

“Tratado práctico sobre la construcción de violines” Barcelona 1985

- Antonio Durá Doménech

“Temas de acústica” Publicaciones Universidad de Alicante

- Basilio Pueo Ortega, Miguel Romá Romero

“Electroacústica, Altavoces y Micrófonos” Pearson Educación, Madrid 2003

PÁGINAS WEB:

- www.midi.org

Portal de la unión de fabricantes creadora del estándar MIDI.

- www.borg.com/jglatt/tech/midifile.htm

Especificación de un archivo MIDI estándar

- www.iainmccurdy.org/csound.html

Ejemplos de implementación de opcodes Csound

Código fuente

```
1 ;Control en tiempo real de la síntesis del sonido mediante un instrumento
;electroacústico.
2 ;Pedro Hernández Rodríguez Alicante 2013, Proyecto fin de carrera.
3 ;Ingeniería técnica en telecomunicaciones, sonido e imagen
4
5 <CsoundSynthesizer>
6
7 <CsOptions>
8
9 -odac1          -iadcl          -d ;entrada y salida de audio en tiempo real
10 ;-Q2           ;salida midi rtmidi
11 ;--midioutfile=midiout.mid      ;opción de generación de archivo midi
12
13 </CsOptions>
14
15 <CsInstruments>
16
17 sr= 44100                ;Opciones de cabecera.
18 ksmps = 32
19 nchnls= 1
20 0dbfs = 1
21
22 gSnotas = "CcDdEfffGgAaB" ;cadena de caracteres nombre notas
23
24 gasig init 0
25
26 ;//////////declaración de canales del bus software//////////
27
28 chn_k "fmin", 1          ;frecuencia mínima
29 chn_k "fmax", 1          ;frecuencia máxima
30 chn_k "gain", 1          ;Ganancia
31 chn_k "tresh", 1         ;Umbral amplitud
32 chn_k "transp", 1        ;Transposición
33 chn_k "reverb", 1        ;Activación reverb
34 chn_k "reverbtime", 1    ;Tiempo de reverberación
35 chn_k "reverbmix", 1     ;Mezcla de reverberación
36 chn_k "hdiff" ,1         ;reverb difusión de las altas frecuencias
37 chn_k "delay", 1         ;Activacion eco
38 chn_k "timedelay",1      ;Tiempo de retardo eco
39 chn_k "mixdelay",1       ;Mezcla delay
40 chn_k "feedback",1      ;Realimentación eco
41 chn_k "tabla" ,1        ;Tabla de ondas
42 chn_k "vocal",1         ;Parámetro para instrumento sinte. voz
43 chn_k "P_Display", 2     ;Display genérico
44 chn_k "frecmod",1        ;Frecuencia de modulación
45 chn_k "midichnl",1      ;Canal de envío MIDI
46
47
```

```

48 ;\\\\\\\\ opcodes definidos por el usuario\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
49
50 ; opcode que imprime en el display genérico los parámetros que se modifican
51 opcode Parameter_Display, 0, k
52
53     kParam,      xin
54     ktrig        changed kParam
55     if (ktrig == 1) then
56         chnset   kParam, "P_Display"
57     endif
58 endop
59
60 ; opcode para escalar magnitudes utilizado en la sección de envío MIDI
61 opcode Scalk, k, kkkkk
62 kval, kinmin, kinmax, koutmin, koutmax xin
63 kres = (((koutmax - koutmin) / (kinmax - kinmin)) * (kval - kinmin)) + koutmin
64 xout kres
65 endop
66
67 ;\\\\\\\\\\\\Captación de parámetros del bus software\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
68 instr 1 ;GUI
69
70 ktrig      metro 10 ; Refresco de parámetros
71
72     if (ktrig == 1) then
73         ;obtención de parámetros de los canales.
74
75         gkfmin      chnget "fmin"
76         gkfmax      chnget "fmax"
77         gkgain      chnget "gain"
78         gktresh     chnget "tresh"
79         gktransp    chnget "transp"
80         gkreverb     chnget "reverb"
81         gkreverbtime chnget "reverbtime"
82         gkreverbmix  chnget "reverbmix"
83         gkhdiff     chnget "hdiff"
84         gkdelay     chnget "delay"
85         gktimedelay chnget "timedelay"
86         gkfeedback  chnget "feedback"
87         gkmixdelay  chnget "mixdelay"
88         gktabla     chnget "tabla"
89         gkvocal     chnget "vocal"
90         gkfrecmod   chnget "frecmod"
91         gkmidichnl  chnget "midichnl"
92     endif
93 endin
94
95
96 instr 2
97 gal init 0 ;inicialización de la cantidad de señal directa.
98
99 ;\\\\\\\\\\\\\\\\\\\\COMUNICACION INTERFAZ-ALGORITMO\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
100
101 konof invalúe "S" ;switch on off
102 kcontrol invalúe "C" ;botón sección de control
103 kanalisi invalúe "A" ;botón sección análisis
104 ktprueba invalúe "T" ;botón activación tono de prueba
105 kinst invalúe "instrument" ; instrumento de prueba
106 ksize invalúe "winsize" ; tamaño de la ventana de analisis fft
107 kchnl invalúe "chnl"
108 kmode invalúe "mode" ; modo de detección, fft ó convolución (AMDF)
109 kmidi invalúe "M" ;Activación envío MIDI

```

```

113 ;selección del tamaño de inventanado FFT
114 if (ksize= 0) then
115     iwinsize = 512
116 elseif(ksize= 1) then
117     iwinsize = 1024
118 elseif(ksize= 2) then
119     iwinsize = 2048
120 elseif(ksize= 3) then
121     iwinsize = 4096
122 endif
123
124 ;\\\\\\\\\\\\\\\\\\\\\\FASE DE CAPTACIÓN Y ADECUACIÓN\\\\\\\\\\\\\\\\\\\\\\
125
126 if (konof= 1) then ;condicional global , on off dispositivo
127     a1 inch 1 ;señal de entrada de audio
128     amp follow2 a1,0.1,0.01 ;seguim. amplitud
129     if kmode = 0 then
130         kf,ka ptrack a1,512 ;seguim. de tono y amplitud HPS .
131     elseif kmode = 1 then
132         kf,ka pitchamdf a1,180,4000,0 ;Técnica convolutiva
133         ka=ka-20
134     endif
135     ;kf tonek kf , 64,0
136     if ka > gktresh then
137
138
139         ;ka port ka, 5 ;adecuación de la amplitud.
140         ka = ka + gkgain ;Ganancia añadida
141         amp = amp + gkgain
142
143
144
145     ;\\\\\\\\\\\\\\\\\\\\\\GENERACIÓN NOMBRE DE NOTA Y ESCALADO FO \\\\\\\\\\\\\\\\\\\\\\\
146
147     knota = 69.5 + (12 *( log10(kf/440) / 0.301 ) );de cps a |MIDI (0-127)
148     kpchclass = int( knota% 12 )
149     kfrac = frac(knota)
150     if kfrac > 0.8 then
151         kfrac = kfrac - 1
152         kpchclass = kpchclass + 1
153     endif
154
155     kcents = kfrac * 100
156     kcents portk kcents, 0.02
157     Snombre strsubk gSnotas, kpchclass, kpchclass + 1
158     kchr strchark Snombre
159     if kchr > 95 then ;sostenido o no
160         Snombre strupperk Snombre
161         Snombre strcatk Snombre,"#"
162     endif
163
164 ;\\\\\\\\\\\\\\\\\\\\\\COMUNICACIÓN ALGORITMO-INTERFAZ SECCIÓN DE ANÁLISIS\\\\\\\\\\\\\\\\\\\\\\
165
166     if (kanalisi = 1) then ;activación sección de análisis
167         outvalue "freq", kf ;display frecuencia
168         outvalue "cents", kcents ;display centésimas de tono
169         outvalue "nota", Snombre ;display Nombre de nota
170         outvalue "amp", ka ;display amplitud
171         outvalue "nmidi", knota ;display nota midi
172     endif
173
174

```



```

175 ; Filtros fmin y fmax
176     if kf < gkfmax then
177     if kf > gkfmin then
178
179 ;printk2 kf,2 ; en caso de imprimir señal de control
180
181 ;////////////////////TRANSPOSICIÓN////////////////////
182     if (gktransp ==0) then
183     kf=kf/0.25
184     elseif (gktransp ==1) then
185     kf = kf/0.5
186     elseif (gktransp ==2) then
187     kf = kf
188     elseif (gktransp ==3) then
189     kf = kf/2
190     elseif (gktransp ==4) then
191     kf = kf/4
192     endif
193
194 ;////////////////////MIDI////////////////////
195
196 kcentinela changed kf
197 if (kcentinela = 0) then
198     kscl Scalk ka, -45, 0,1,127
199 endif
200 ;printk2 kscl,2
201 if kmidi =1 then ;envío midi
202     midion gkmidichnl,knota,40 ;en el último parámetro poner kscl en su caso
203     endif
204
205 ;\\\\\\\\\\\\\\\\\\FASE DE SÍNTESIS\\\\\\\\\\\\\\\\\\
206
207     if ktprueba = 1 then ;Activacion sección de síntesis
208
209         ;Instrumento oscil1
210         if (kinst = 0) then
211         aout oscili amp , kf, 1
212         ;Instrumento oscil2
213         elseif (kinst = 1) then
214         aout oscili amp, kf, 2
215         ;Instrumento vco
216         elseif (kinst= 2) then
217         aout vco2 ampdb(ka-5), kf
218         ;Instrumento pluck
219         elseif (kinst = 3) then
220         aout pluck ampdb(ka-12), kf,0,5,2,1,1
221         ;Instrumento organ
222         elseif (kinst = 4) then
223         kenv linseg 0, .01, 0.2, 200-.02, 0.2, .01, 0
224
225         a1 oscil amp, 1 * kf, 3
226         a2 oscil amp, 2 * kf, 3
227         a3 oscil amp, 3 * kf, 3
228         a4 oscil amp, 4 * kf, 3
229         a5 oscil amp/2.5, 6 * kf, 3
230         a6 oscil amp/4, 8 * kf, 3
231         a7 oscil amp/8, 10* kf, 3
232         a8 oscil amp/8, 12* kf, 3
233         a9 oscil amp/2, 16* kf, 3
234
235         aorgan = kenv* (a1+a2+a3+a4+a5+a6+a7+a8+a9)

```

```

236         aout = aorgan
237         ;Instrumento buzz
238         elseif (kinst = 5) then
239         aout buzz amp*2, kf,3,2
240         ;Instrumento flute
241         elseif (kinst = 6) then
242         aout1 oscili amp , kf, 1
243         aout2 wgflute ampdb(ka), kf/1.5,0.08,0.1,0.1,0.1,4,0.05,1
244         aout = (2*aout1+aout2)/3
245         ;Instrumento woodwind
246         elseif (kinst = 7) then
247         aout wgclar ampdb (ka+5), kf,-0.2,5,0.5,0.3,0,0.05,1
248         ;Instrumento bell
249         elseif (kinst = 9) then
250         aout fmbell ampdb (ka), kf, 10, 5, 0.005, 6, 1, 1, 1, 1, 1,100
251         out aout
252         elseif (kinst = 10) then
253         ;Instrumento percflute
254         aout fmpercfl ampdb (ka), kf/0.75, 10, 5, .01, 6, 1,1,1,1,1
255         ;Instrumento voice
256         elseif (kinst = 11) then
257         aout fmvoice ampdb (ka), kf, gkvocal, 90, 0.005, 6, 1,1,1,1,1
258         endif ;fin condicional de selección de instrumentos
259
260         out aout ;salida de audio
261
262         gal = gal + aout ;añadir señal global reverb
263         vincr gasig, aout ;añadir señal global delay
264
265         endif ;fin condicional tono prueba
266         endif ;fin condicional filtro max
267         endif ;fin condicional filtro min
268         endif ;fin condicional umbral (kth)
269     endif ;fin condicional on off dispositivo.
270     endin
271
272 ;Envío de parámetros al display genérico
273 instr 4
274
275     Parameter_Display gkfmin
276     Parameter_Display gkfmax
277     Parameter_Display gkgain
278     Parameter_Display gktresh
279     Parameter_Display gkreverbtime
280     Parameter_Display gkreverbmix
281     Parameter_Display gkhdifff
282     Parameter_Display gktimedelay
283     Parameter_Display gkfeedback
284     Parameter_Display gkmixdelay
285     Parameter_Display gkvocal
286
287     endin
288
289 ;////////////////////SECCIÓN DE EFECTOS////////////////////////////////////
290
291
292 ;BLOQUE DELAY
293 instr 90
294
295 if gkdelay = 1 then ;Activación bloque delay
296 aBasura delayr 2,100

```

```

297 aBufOut deltap gktimedelay
298     delayw gasig+(aBufOut*gkfeedback)
299     out gasig+(aBufOut*gkmixdelay)
300     clear gasig ;reinicialización delay
301 endif
302 endin
303
304 ;BLOQUE REVERBERACIÓN.
305 instr 99
306
307 krevs invalúe "rs" ;switch reverb on of
308
309 if gkreverb = 1 then ;Activación bloque reverb
310 ;gal butterbp gal ,440,1000 ;utilizar en caso de filtrado reverb
311 arev reverb2 gal*gkreverbmix , gkreverbtime ,gkhdiff
312     out arev
313     gal = 0 ;reinicialización reverb
314 endif
315 endin
316
317 </CsInstruments>
318
319 <CsScore>
320
321 ; tablas de onda
322 f 1 0 131072 10 1 ;Sinusoidal
323 f 2 0 256 10 1 .2 .15 .1 .2 .1 .0.02 .01 ;Sinusoidal compleja
324 f 3 0 8192 10 1 .02 .01 ;C.sine 2
325 f 4 0 16384 10 1 1 1 0.7 0.5 0.3 0.1 ; Pulse
326 f 5 0 16384 10 1 0 0.3 0 0.2 0 0.14 0 .111 ; Square
327 f 6 0 16384 10 1 0.5 0.3 0.25 0.2 0.167 0.14 0.125 .111 ; Sawtooth
328 f 101 0 512 7 0 256 1 256 0 ; triángulo
329 f 102 0 512 10 1 ; senoide simple
330 f 103 0 512 7 0 56 0.4 200 0.4 225 0.2 31 0
331 f 7 0 256 1 "fwavblnk.aiff" 0 0 0 ;ruido blanco utilizado para algunos instrm.
332
333 ;instrumentos
334 i1 0 3600
335 i2 0 3600
336 i4 0 3600
337 i90 0 3600
338 i99 0 3600 ;mantiene sonando los instrumentos durante una hora.
339 e
340
341 </CsScore>
342
343 </CsSoundSynthesizer>

```


Opcodes CSound utilizados ordenados alfabéticamente

ampdb

Es un opcode convertidor de valor: se encarga de transformar un valor dado en decibelios a su equivalente valor PCM.

Sintaxis:

ampdb(x)

Siendo x el valor en dB a convertir.

buzz

Es un opcode de generación de señal: genera una serie de parciales coseno armónicamente relacionados.

Sintaxis:

asal **buzz** xamp, xfrec, knarm, ifn

siendo:

asal: Señal sintetizada.

xamp: Amplitud de la señal.

xfrec: Frecuencia fundamental de la señal.

changed

Detector de cambios en señales tipo k, devuelve 1 en caso de cambio.

Sintaxis:

ktrig **changed** kvar1 [, kvar2,..., kvarN]

siendo:

ktrig: señal de salida, 1 si hay cambios kvar1....kvarN: distintas variables a detectar.

chn-k

Declara y nombra un canal de bus software tipo k.

Sintaxis:

chn-k Sname, imode[, itype, idft, imin, imax]

siendo:

Sname: Nombre del canal declarado

imode: Modo de funcionamiento (1=entrada, 2=salida)

chnget

Lectura de datos del canal de bus software especificado en la sintaxis.

Sintaxis:

kval **chnget** Sname

siendo:

kval: Valor leído de tipo k Sname: Nombre del canal.

chnset

Escribe datos en el canal con el nombre especificado, del cual se requiere previa declaración.

Sintaxis:

chnset kval, Sname

siendo:

Kval: Variable de control que se desea escribir

Sname: Nombre del canal

clear

Resetea a 0 los valores de la señal de audio determinada.

Sintaxis:

clear avar1

siendo:

avar1: Señal de audio a resetear.

delayr

Es un opcode de modificación de señal: permite leer una línea digital de retardo en la que la señal ha estado guardada un tiempo.

Sintaxis:

ares **delayr** idlt

siendo:

ares: Valor retardado

idlt: Tiempo de retardo

delayw

Es un opcode de modificación de señal: permite escribir la variable a la que hace referencia en una línea de retardo creada antes por el opcode delayr que siempre le precede.

Sintaxis:

delayw asig

siendo:

asig: Señal a introducir en la línea de retardo

deltap

Lee el parámetro específico de tiempo de delay introducido desde el interfaz.

Sintaxis:

ares **deltap** kdlt

siendo:

ares: Señal de salida

kdl: Tiempo de retardo en segundos

fbell

Opcod sintetizador de sonido, utiliza síntesis FM para crear el sonido de campana.

Sintaxis:

ares **fbell** kamp, kfreq, kc1, kc2, kvdepth, kvrate, ifn1, ifn2, ifn3, ifn4

siendo:

ares: Señal de audio sintetizado

kamp: Amplitud de la nota

kfreq: Frecuencia fundamental de la nota

kc1,kc2: Controles del sintetizador, (kc1= índice de modulación, kc2= mezcla de la salida)

kvdepth: Profundidad vibratoria (LFO)

kvrate: Frecuencia de vibración (LFO)

ifn1,ifn2,ifn3,ifn4: Ondas sinusoidales almacenadas en tablas

fmpercfl

Opcod que utiliza síntesis FM para emular el sonido percusivo de una flauta.

Sintaxis:

ares **fmpercfl** kamp, kfreq, kc1, kc2, kvdepth, kvrate, ifn1, ifn2, ifn3, ifn4

siendo:

ares: Señal de audio sintetizado

kamp: Amplitud de la nota

kfreq: Frecuencia fundamental de la nota

kc1,kc2: Controles del sintetizador, (kc1= índice de modulación, kc2= mezcla de la salida)

kvdepth: Profundidad vibratoria (LFO)

kvrate: Frecuencia de vibración (LFO)

ifn1,ifn2,ifn3,ifn4: Ondas sinusoidales almacenadas en tablas

fmvoice

Opcod que utiliza síntesis FM para emular el sonido de la voz humana.

Sintaxis:

ares **fmvoice** kamp, kfreq, kvowel, ktilt, kvibamt, kvibrate, ifn1, ifn2, ifn3, ifn4, ivibfn

siendo:

ares: Señal de audio sintetizado

kamp: Amplitud de la nota

kfreq: Frecuencia fundamental de la nota

kvowel: Parámetro que determina la vocal, rango (0-64) ktilt: Parámetro que determina la envolvente espectral (0-99) kvibamt: Profundidad del vibrato

kvibrate: Frecuencia del vibrato

ifn1,ifn2,ifn3,ifn4: Ondas sinusoidales almacenadas en tablas

ivibfn: Onda del LFO almacenada en tabla

follow2

Extractor de envolvente controlable, utilizado para seguimiento de amplitud.

Sintaxis:

ares **follow2** asig, katt, krel

siendo:

ares: Envolvente extraída, (señal tipo audio)

asig: Señal analizada

katt: Tiempo de ataque (relativo a 60dB) en segundos

krel: Tiempo de relajación en segundos

inch

Lee del canal especificado una señal de audio externa.

Sintaxis:

ain1 **inch** kchan1

siendo:

ain1: Señal de audio captada

kchan1: Canal especificado (1 por defecto)

init

Opcod que inicializa la variable de audio o control al valor especificado.

Sintaxis:

ares **init** iarg

siendo:

ares: Señal a inicializar

iarg: Valor especificado

invalue

Lee señales tipo k de un canal definido por el usuario, utilizado para la comunicación interfaz-algoritmo.

Sintaxis:

kvalue **invalue** "channel name"

siendo:

kvalue: Variable tipo k donde se almacena el dato leído

channel name: Nombre del canal

linseg

Trazador de segmentos rectilíneos entre determinados puntos, utilizado para crear envolventes.

Sintaxis:

kres **linseg** ia, idur1, ib [, idur2] [, ic] [...]

siendo:

kres: Señal envolvente

ia,ib...: Puntos entre los cuales se traza el segmento

idur: Duración en segundos de los diferentes segmentos

metro

Generador de señal metronómica en el caso de necesitar disparador asíncrono, en nuestro caso determina la velocidad de refresco en los parámetros de comunicación algoritmo-interfaz.

Sintaxis:

ktrig **metro** kfreq

siendo:

ktrig: Señal metronómica, trigger

kfreq: Frecuencia de los diferentes disparos

midion

Genera mensajes MIDI tipo NOTE ON con frecuencia de control (k-rate), en el caso de que algún parámetro de entrada cambie, se silencia la nota y se genera la siguiente con el valor actualizado.

Sintaxis:

midion kchn, knum, kvel

siendo:

kchn: Canal de envío MIDI

knum: Número asociado a la frecuencia fundamental de la nota

kvel: Velocidad MIDI (Amplitud en nuestro caso)

oscil

Es un opcode generador de señal que permite crear sonidos en base a la repetición de una cierta tabla de onda. (Oscilador simple)

Sintaxis:

ares **oscil** xamp, xcps, ifn

siendo:

ares: Señal de audio sintetizada

xamp: Valor de la amplitud

xcps: Valor de la frecuencia fundamental en ciclos por segundo

ifn: Tabla de onda a emplear

oscili

Oscilador simple con interpolación lineal.

Sintaxis:

ares **oscili** xamp, xcps, ifn

siendo:

ares: Señal de audio sintetizada

xamp: Valor de la amplitud

xcps: Valor de la frecuencia fundamental en ciclos por segundo

ifn: Tabla de onda a emplear

out

Escribe señal de audio monofónica en dispositivo de salida externo, es decir, hace sonar la señal indicada.

Sintaxis:

out asig

siendo:

asig: Señal de salida a reproducir

outvalue

Envío de una señal tipo k a un canal definido por el usuario. Utilizado para la comunicación algoritmo-interfaz **Sintaxis:**

outvalue “channel name”, kvalue

siendo:

channel name: Nombre del canal

kvalue: parámetro a enviar

pitchamdf

Opcod de seguimiento de pitch de una señal basado en el método AMDF (Average Magnitude of Difference Function), una función de autocorrelación, devuelve pitch y valor de amplitud. El método es muy rápido y funciona en tiempo real. Esta técnica suele funcionar mejor para señales monofónicas.

Sintaxis:

kcps, krms **pitchamdf** asig, imincps, imaxcps

siendo:

kcps: Valor de la frecuencia fundamental detectada en ciclos por segundo (Hz)

krms: Valor de la amplitud detectada

asig: Señal analizada

imincps, imaxcps: Valores mínimo y máximo de frecuencia estimados en la señal

pluck

Opcod de síntesis que emula el sonido de decaimiento de una cuerda pulsada o una membrana.

Sintaxis:

ares **pluck** kamp, kcps, icps, ifn, imeth

siendo:

ares: Señal de audio sintetizada

kamp: Valor de amplitud

kcps: Valor de frecuencia

icps: Anticipación del valor kcps

ifn: Tabla de ondas utilizada

imeth: Método de decaimiento natural (1-6)

portk

Aplica portamento a una señal definida por intervalos, suavizado en intervalos.

Sintaxis:

kres **portk** ksig, khtim

siendo:

kres: Señal suavizada

ksig: Señal a suavizar

khtim: Factor de suavizado

ptrack

Opcod de seguimiento de pitch y amplitud, utiliza las técnicas de análisis y pitchtracking: STFT (Short Time Fourier Transform) y HPS (Harmónic Product Spectrum), explicadas anteriormente.

Sintaxis:

kcps, kamp **ptrack** asig, ihopsize

siendo:

kcps: Valor de la frecuencia fundamental en ciclos por segundo (Hz)

kamp: Valor de la amplitud

asig: Señal analizada

ihopsize: Valor en muestras del solapamiento, (tamaño de enventanado = 2ihopsize)

reverb2

Opcod reverberador que utiliza 11 filtros con determinados retardos.

Sintaxis:

ares **reverb2** asig, ktime, khdif

siendo:

ares: Señal reverberada

asig: Señal original

ktime: Tiempo de reverberación en segundos

khdif: Difusión en altas frecuencias

strcatk

Opcode concatenador de cadenas tipo k.

Sintaxis:

Sdst **strcatk** Ssrc1, Ssrc2

siendo:

Sdst: Cadena resultante

Ssrc1, Ssrc2: Cadenas a concatenar

strchark

Devuelve el código ASCII de un carácter en una cadena.

Sintaxis:

kchr **strchark** Sstr

siendo:

Kchr: Código ASCII

Sstr: Sub-cadena, o carácter

strsubk

Devuelve una sub-cadena dentro de una cadena fuente.

Sintaxis:

Sdst **strsubk** Ssrc, kstart, kend

siendo:

Sdst: Sub-cadena de salida

Ssrc: Cadena fuente de entrada

kstart,kend: posiciones inicio y fin en la cadena fuente

strupperk

Convierte el carácter al inmediatamente superior en una cadena.

Sintaxis:

Sdst **strupperk** Ssrc

siendo:

Sdst: Variable cadena en la que se almacena el carácter superior

Ssrc: Carácter seleccionado a convertir

vco2

Opcod que implementa un oscilador de banda limitada usando tablas de onda predefinidas.

Sintaxis:

ares **vco2** kamp, kcps

siendo:

ares: Señal de audio sintetizada

kamp: Valor de amplitud

kcps: Valor de frecuencia

vincr

Opcod incrementador acumulador.

Sintaxis:

vincr accum, aincr

siendo:

accum: Incremento

aincr: Señal incrementada

wgclar

Opcod sintetizador que crea tonos similares a un clarinete.

Sintaxis:

ares **wgclar** kamp, kfreq, kstiff, iatt, idetk, kngain, kvibf, kvamp

siendo:

ares: Señal de audio sintetizada

kamp: Valor de amplitud
kfreq: Valor de frecuencia
kstiff: Parámetro de control de rigidez
iatt: Tiempo de ataque (Soplido total)
idetk: Tiempo de parada de soplado
kngain: Ganancia de la componente de ruido
kvibf: Frecuencia de vibrato
kvamp: Amplitud del vibrato

xin

Recibe variables de entrada a un opcode definido por el usuario

Sintaxis:

xinarg1 **xin**

siendo:

xinarg1: Argumento de entrada

xout

Envía variables de salida desde un opcode definido por el usuario

Sintaxis:

xout xoutarg1

siendo:

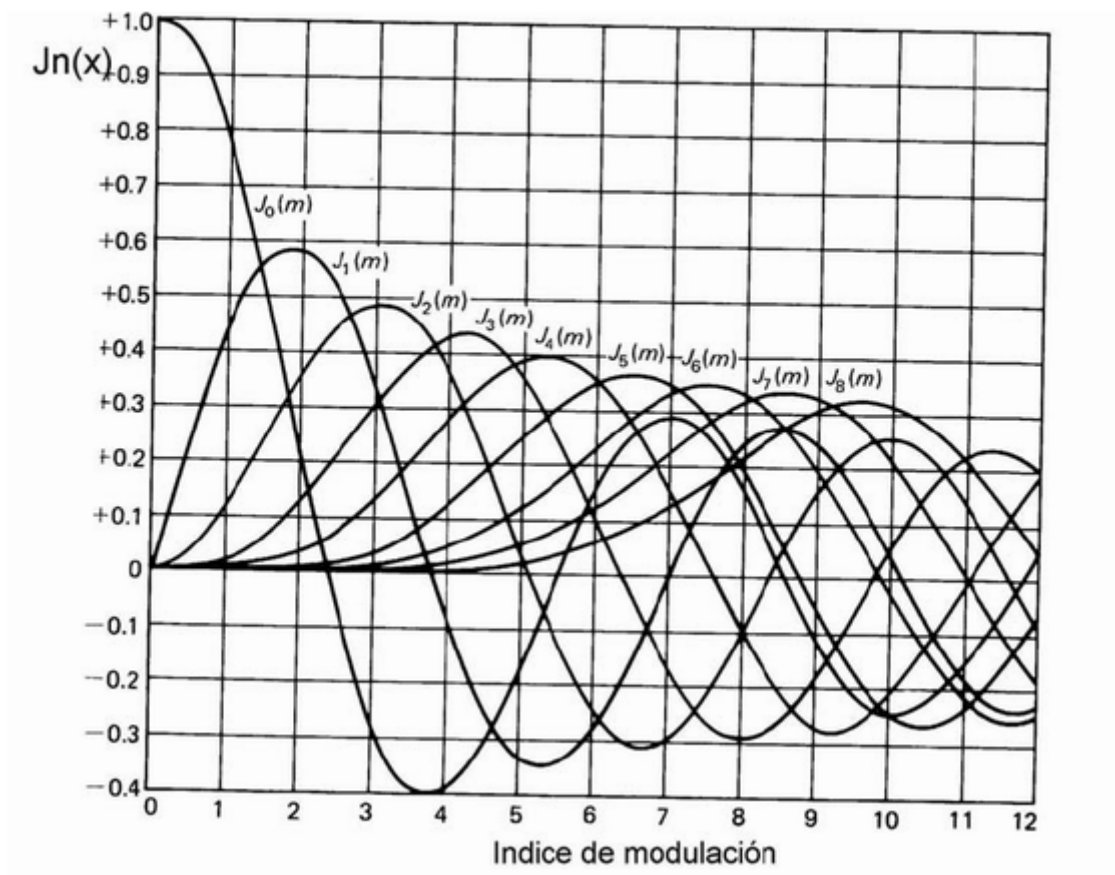
xoutarg1: Argumento de salida

Tabla de intervalos musicales

Nombre del intervalo	Valor en tonos y semitonos	Relación de frecuencias
Unísono ² / Primera justa	0	1
Primera aumentada / Semitono cromático ²	1 semitono	
Segunda disminuida	0	
Segunda menor	1 semitono	
Segunda mayor	1 tono	9/8
Segunda aumentada	1 1/2 tono	
Tercera disminuida	1 tono	
Tercera menor	1 1/2 tono	
Tercera mayor	2 tonos	5/4
Tercera aumentada	2 1/2 tono	
Cuarta disminuida	2 tonos	
Cuarta justa	2 1/2 tonos	4/3
Cuarta aumentada (llamada <i>tritono</i>) ³	3 tonos	
Quinta disminuida (llamada <i>falsa quinta</i>) ³	3 tonos	
Quinta justa	3 1/2 tonos	3/2
Quinta aumentada	4 tonos	
Sexta disminuida	3 1/2 tonos	
Sexta menor	4 tonos	
Sexta mayor	4 1/2 tonos	5/3
Sexta aumentada	5 tonos	
Séptima disminuida	4 1/2 tonos	
Séptima menor	5 tonos	
Séptima mayor	5 1/2 tonos	15/8
Séptima aumentada	6 tonos	
Octava justa	6 tonos	2

Funciones Bessel

Funciones de Bessel



Alturas MIDI

MIDI note to frequency conversion chart

MIDI Note	Frequency	MIDI Note	Frequency	MIDI Note	Frequency
Octave -1		Octave 0		Octave 1	
C 0	8.1757989156	12	16.3515978313	24	32.7031956626
Db 1	8.6619572180	13	17.3239144361	25	34.6478288721
D 2	9.1770239974	14	18.3540479948	26	36.7080959897
Eb 3	9.7227182413	15	19.4454364826	27	38.8908729653
E 4	10.3008611535	16	20.6017223071	28	41.2034446141
F 5	10.9133822323	17	21.8267644646	29	43.6535289291
Gb 6	11.5623257097	18	23.1246514195	30	46.2493028390
G 7	12.2498573744	19	24.4997147489	31	48.9994294977
Ab 8	12.9782717994	20	25.9565435987	32	51.9130871975
A 9	13.7500000000	21	27.5000000000	33	55.0000000000
Bb 10	14.5676175474	22	29.1352350949	34	58.2704701898
B 11	15.4338531643	23	30.8677063285	35	61.7354126570
Octave 2		Octave 3		Octave 4	
C 36	65.4063913251	48	130.8127826503	60	261.6255653006
Db 37	69.2956577442	49	138.5913154884	61	277.1826309769
D 38	73.4161919794	50	146.8323839587	62	293.6647679174
Eb 39	77.7817459305	51	155.5634918610	63	311.1269837221
E 40	82.4068892282	52	164.8137784564	64	329.6275569129
F 41	87.3070578583	53	174.6141157165	65	349.2282314330
Gb 42	92.4986056779	54	184.9972113558	66	369.9944227116
G 43	97.9988589954	55	195.9977179909	67	391.9954359817
Ab 44	103.8261743950	56	207.6523487900	68	415.3046975799
A 45	110.0000000000	57	220.0000000000	69	440.0000000000
Bb 46	116.5409403795	58	233.0818807590	70	466.1637615181
B 47	123.4708253140	59	246.9416506281	71	493.8833012561
Octave 5		Octave 6		Octave 7	
C 72	523.2511306012	84	1046.5022612024	96	2093.0045224048
Db 73	554.3652619537	85	1108.7305239075	97	2217.4610478150
D 74	587.3295358348	86	1174.6590716696	98	2349.3181433393
Eb 75	622.2539674442	87	1244.5079348883	99	2489.0158697766
E 76	659.2551138257	88	1318.5102276515	100	2637.0204553030
F 77	698.4564628660	89	1396.9129257320	101	2793.8258514640
Gb 78	739.9888454233	90	1479.9776908465	102	2959.9553816931
G 79	783.9908719635	91	1567.9817439270	103	3135.9634878540
Ab 80	830.6093951599	92	1661.2187903198	104	3322.4375806396
A 81	880.0000000000	93	1760.0000000000	105	3520.0000000000
Bb 82	932.3275230362	94	1864.6550460724	106	3729.3100921447
B 83	987.7666025122	95	1975.5332050245	107	3951.0664100490
Octave 8		Octave 9			
C 108	4186.0090448096	120	8372.0180896192		
Db 109	4434.9220956300	121	8869.8441912599		
D 110	4698.6362866785	122	9397.2725733570		
Eb 111	4978.0317395533	123	9956.0634791066		
E 112	5274.0409106059	124	10548.0818212118		
F 113	5587.6517029281	125	11175.3034058561		
Gb 114	5919.9107633862	126	11839.8215267723		
G 115	6271.9269757080	127	12543.8539514160		
Ab 116	6644.8751612791				
A 117	7040.0000000000				
Bb 118	7458.6201842894				
B 119	7902.1328200980				

Lista de archivos en el CD adjunto:

- Memoria del proyecto en formato .pdf
- algoritmo FO-AMP en formato .csd
- Vídeo representativo: funcionamiento del sistema en formato .wmv
- Carpeta con diferentes archivos de audio: resultados sintetizados en formato .wav
- Carpeta con diferentes archivos MIDI: resultados convertidos en formato .mid

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A

copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text. The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this

License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between

the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>. Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any

later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.