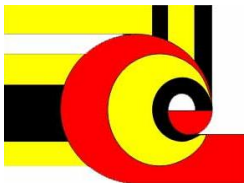
Instituto Politécnico Nacional



Centro de Investigación en Computación

Aplicación de métodos de procesamiento digital de voz a señales de audio para la extracción de características de interés para el lenguaje musical.

T E S I S

Que para obtener el grado de:

Maestría en Ciencias en Ingeniería de Cómputo

P R E S E N T A :

Ing. Pablo Tovar Castrejón.

Director de tesis.

Dr. Sergio Suárez Guerra.

ABRIL 2016

SIP-14

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:00 horas del día 30 del mes de marzo de 2016 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Aplicación de métodos de procesamiento digital de voz a señales de audio para la extracción de características de interés para el lenguaje musical"

Presentada por el alumno(a):

Tovar	Castrejón	Pablo
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

A	1	4	0	1	6	3

aspirante de: **MAESTRÍA EN CIENCIAS EN INGENIERÍA DE CÓMPUTO**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de Tesis

Dr. Sergio Suárez Guerra

Dr. Oleksiy Pogrebnyak

Dr. Luis Pastor Sánchez Fernández

Dr. Ricardo Barrón Fernández

M. en C. Pablo Manrique Ramírez

Dr. Francisco Hiram Calvo Castro

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

ESTADOS UNIDOS MEXICANOS
INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN COMPUTACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 21 del mes de Abril del año 2016, el (la) que suscribe Pablo Tovar Castrejón alumno (a) del Programa de Maestría en Ciencias en Ingeniería de Cómputo con número de registro A140163, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Sergio Suárez Guerra y cede los derechos del trabajo intitulado Aplicación de métodos de procesamiento digital de voz a señales de audio para la extracción de características de interés para el lenguaje musical, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección casvar.pablo@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Pablo Tovar Castrejón

Nombre y firma

Resumen

En el presente trabajo se propone un sistema para el análisis de pistas de audio generadas por un instrumento musical con el objetivo de extraer características de interés para el lenguaje musical a modo de obtener una representación gráfica de la misma con resultados comparables a otros trabajos del estado del arte. Entre las técnicas utilizadas se encuentran la extracción de los Mel Frequency Cepstral Coefficients (MFCC), el uso de Redes Neuronales y Modelos Ocultos de Markov. El instrumento musical que se eligió fue el piano, mismo que es considerado uno de los instrumentos musicales más completos debido al amplio rango de frecuencias que puede emitir, así como la enorme variedad de combinaciones de notas que puede generar.

Para la clasificación de 48 notas musicales y 96 acordes (144 emisiones sonoras en total) provenientes de un piano se utilizaron los MFCCs como vector de características, lo cual permitió la identificación no solo de notas musicales y acordes mayores y menores, sino también la octava a la que estos pertenecen. Esto permite dar un paso adelante en la extracción de características de la música a comparación de otros trabajos del estado del arte como el propuesto por Papadopoulos [Pap07] en donde se propone un método para clasificar acordes emitidos por el piano sin tomar en cuenta la octava, lo cual limita su conjunto de análisis a 24 acordes, o bien trabajos como el presentado por Mauch [Mau10] en donde se propone un método para la clasificación de acordes emitidos por la guitarra, en donde la octava del acorde no es de importancia, reportando una clasificación de 71% para 109 acordes.

Una vez extraídos los vectores de características de una pieza musical generada por un piano se desarrolló un sistema de redes neuronales que funciona como árbol de decisiones, el cual toma como entrada los Mel Frequency Cepstral Coefficients y entrega a la salida el código MIDI de la emisión sonora identificada. Este enfoque de niveles es innovador y permite aumentar el número de emisiones sonoras a clasificar sin reducir de manera drástica el porcentaje de clasificaciones correctas. Bajo la clasificación propuesta en el presente trabajo se llega a un 83% de emisiones sonoras correctamente clasificadas en el caso de acordes mayores o menores y de un 92% para notas musicales.

Una vez que se han obtenido los códigos MIDI se obtiene una secuencia con la información estimada de la señal de audio. Estas secuencias MIDI son susceptibles a presentar errores principalmente debidos a intervalos de análisis en los que se presentan cambios de nota o acorde. Es por ello que se proponen diversos métodos para reducir estos errores de clasificación. Entre los métodos presentados destacan los basados en Modelos Ocultos de Markov, mismos que se utilizan como filtros de secuencias MIDI. Como primera aproximación se toma un modelo similar al propuesto por Barbancho [Bar12] en donde se genera un único Modelo Oculto de Markov para la clasificación de acordes de guitarra en donde los símbolos observables corresponden al cromagrama de la señal de audio y cada estado oculto se corresponde con un acorde a clasificar. Bajo esta perspectiva Barbancho reporta un porcentaje de acordes correctamente clasificados del 87% para 330 posiciones en la guitarra. En el presente trabajo gracias al uso de un HMM como filtro se logra aumentar el porcentaje de notas correctamente clasificadas un 4%, llegando también a un 87% de clasificaciones correctas para 144 emisiones sonoras. En este punto cabe mencionar que a pesar de que el conjunto de emisiones sonoras aquí analizadas tiene una menor cardinalidad que el reportado por Barbancho éste incluye clasificaciones no solo de acordes, sino separación entre notas musicales y acordes, incluyendo en todos los casos la octava a la que estos pertenecen.

Por otro lado también se propone el uso de un Modelo Oculto de Markov por cada emisión sonora a identificar compuestos de dos estados ocultos cada uno, con lo que dada una secuencia MIDI se busca el modelo que la genera con una mayor probabilidad. De la misma manera fue posible hacer uso del segundo enfoque en el uso de HMM para la identificación de notas musicales a las que se le aplica un vibrato. Con esto se aporta una visión innovadora en lo que podría entenderse como una segunda etapa de clasificación de emisiones sonoras.

Abstract

In this work a system for analyzing tracks generated by a musical instrument is proposed, with the aim to extract characteristics of interest to the music language in order to obtain a graphic representation, with results comparables to other state-of-the-art papers. Among techniques used can be mentioned extraction of Mel Frequency Cepstral Coefficients (MFCC), artificial neural networks and hidden Markov models (HMM). The chosen musical instrument was the piano, same that is considered one of the most complete instrument due to the wide range of frequencies that can emit, as well as the variety of combinations of notes that can generate.

In order to classified 48 notes and 96 chords (144 emissions in total) MFCC were used as feature vector, which allowed the identification of the eighth they belong.

Once extracted the feature vectors a neural network system were developed that function as decision tree, which takes MFCC as input and delivers MIDI codes of emissions identified. This approach is innovative and increases the number of emissions to classify without reducing drastically the percentage of correct classifications. Under the classification proposed in this work a percentage of 83% on chords classification were reach and 92% in the case of musical notes.

After the analysis with neural networks system a MIDI sequences are obtained, same that present errors due intervals that contain changes of notes and chord principally. In order to reduce this errors two approaches on use of hidden Markov models were proposed. The first models is similar to the one proposed by Barbancho [Bar 12], where Viterbi algorithm was used. On the other hand a second approach were proposed, were every emission (note or chord) has a specific hidden Markov Model made of two states, then the forward algorithm was used in order to identify the model that generates a sequence of MIDI symbols with the higher probability. This approach allowed the identification of vibratos applied to music notes.

Agradecimientos

A mi madre, que ha sido la persona que siempre ha estado a mi lado apoyando todas las decisiones que he tomado y es por quien he logrado superarme. A mi padre, que me ha inculcado carácter y un interés particular en la música.

Agradezco al Dr. Sergio Suárez Guerra por su amable asesoramiento y a todos los profesores que me han apoyado a lo largo de mi trayectoria académica, así como al Consejo Nacional de Ciencia y Tecnología por el apoyo económico que me brindó para que pudiera realizar los estudios de Maestría.

Al Dr. Ernesto Rodrigo Vázquez Cerón, al Dr. Victor Rogelio Barrales Guadarrama y a todos los miembros del laboratorio de sensores y señales de la Universidad Autónoma Metropolitana, que me han apoyado y han creído en mí.

Índice de contenido

Índice de contenido vi
1 Introducción 1
1.1 Planteamiento del problema 1
1.1 Justificación 1
1.2 Objetivo general 3
1.3 Objetivos particulares 4
1.4 Alcances del trabajo 4
2 Estado del arte 5
2.1 Descripción de cómo se afronta esta problemática en la actualidad. 8
2.2 The Music Information Retrieval Evaluation eXchange (MIREX). 15
3 Marco Teórico 16
3.1 Sonido 16
3.2 Sensaciones sonoras 16
3.3 Notas musicales 16
3.4 Pentagrama 17
3.5 Frecuencia de las notas musicales 18
3.6 Semitono y Tono 20
3.7 Duración de una nota musical 21
3.8 Compás 22
3.9 Intervalos 23
3.10 Escala musical 23
3.11 Escala Mayor 24
3.12 Escala menor 25
3.13 Armadura 25
3.14 Melodía, Armonía y Ritmo 26
3.15 Vibrato 27
3.16 Acorde 27
3.17 MIDI 28
3.17.1 Header chunk 28
3.17.2 Track chunk 30
3.18 Técnicas y métodos 33
3.18.1 Señales 33
3.18.2 Señales de audio producidas por el piano 34
3.18.3 Función de autocorrelación 35
3.18.4 Espectro y transformada de Fourier 35
3.18.5 Ventaneo 36

3.18.6 ***Mel Frequency Cepstral Coefficients*** 38
3.18.7 Redes neuronales artificiales 39
3.18.8 Modelos Ocultos de Markov 41
3.18.9 Cromagrama 42
4 Propuesta de solución. 45
4.1 Algoritmo para la obtención de una representación musical en pentagrama 45
4.2 Sistema de redes neuronales para la clasificación de emisiones sonoras 48
4.3 Filtrado de secuencias de símbolos 49
4.4 Selección de la longitud de los segmentos a análisis y la ventana a utilizar 49
5 Pruebas y resultados 51
5.1 Análisis por autocorrelación 51
5.1.1 Análisis de melodía por autocorrelación 52
5.2 Análisis por transformada de Fourier 55
5.2.1 Puntualizaciones en el análisis de las señales haciendo uso de la transformada de Fourier 56
5.3 Uso de ***MFCCs*** y ***LPCs*** 57
5.4 Análisis del algoritmo propuesto en condiciones ideales 61
5.5 Limpieza de las instrucciones MIDI generadas. 63
5.6 Uso de Modelos Ocultos de Markov (HMM) para el reconocimiento de notas musicales. 67
5.6.1 Primera aproximación en el uso de HMM 67
5.6.2 Segunda aproximación en el uso de HMM 71
5.6.3 Reconocimiento de vibrato 73
5.7 Comparación con otros trabajos del estado del arte 76
6 Conclusiones, trabajos futuros y aportaciones científicas. 78
6.1 Conclusión del objetivo general. Aplicar métodos de procesamiento digital de voz a señales de audio generadas por un instrumento musical, adaptando y extendiendo los mismos para la extracción de características de interés en el lenguaje musical 78
6.1.1 Conclusión del objetivo específico 1. Identificación de notas musicales vía F0. 78
6.1.2 Conclusión del objetivo específico 2. Identificación de acordes usando MFCCs y redes neuronales. 78
6.1.3 Conclusión del objetivo específico 3. Uso de HMM para la corrección de secuencias de emisiones sonoras y reconocimiento de vibratos. 78
6.1.4 Conclusión del objetivo específico 4. Uso del análisis de energía de la señal para la estimación de la duración de las emisiones sonoras en una pista de audio a modo de obtener una representación en figuras propias del lenguaje musical. 79
6.1.5 Conclusión del objetivo específico 5. Desarrollo de una librería de funciones que permitan la generación de archivos MIDI. 79
6.2 Posibles aplicaciones inmediatas del presente trabajo 80
6.3 Trabajos a futuro 80
6.4 Aportaciones científicas 80
7 Referencias 81

Índice de figuras

Figura 1.1 Pentagrama de los primeros cuatro compases de la obra "Für Elise" de Ludwing van Beethoven. 1
Figura 1.2 Otras representaciones utilizadas para expresar piezas musicales. (a) Notación hindú, tomada de [Cou12]. (b) Notación gregoriana, fragmento de la obra "Pangue lingua gloriosi" de Santo Tomás de Aquino. 2
Figura 1.3 Correspondencia entre la frecuencia de un señal de audio y la notación propia de los músicos. Tomada de [Per03]. 2
Figura 1.4 Espectrograma de una melodía generada con una onda sinusoidal y su partitura correspondiente. Se muestran las correspondencias entre algunas bandas espectrales y las notas que producen. Tomada de [Per03]. 3
Figura 2.1 Sistema de transcripción de música propuesto por Kashino y Tanaka. Tomada de [Kas93]. 5
Figura 2.2 Sistema para la transcripción automática de música. Tomada de [Kas95]. 6
Figura 2.3 CNN para la extracción de patrones musicales. Tomada de [Li10]. 7
Figura 2.4 Curva de convergencia en entrenamiento de 200 épocas. 7
Figura 2.5 Ejemplo de lead sheet. Como puede observarse en la parte superior se especifican los acordes a interpretar usando la representación de un brazo de guitarra en lugar de colocar las notas en un pentagrama. Tomada de [Mau10]. 8
Figura 2.6 Ejemplo de posiciones para la emisión del acorde de Do mayor. Las líneas verticales indican las seis cuerdas y los números dentro de los círculos sobre la cuerdas indican la posición de los dedos. En el caso de b) se muestra una barra en donde el dedo índice presiona varias cuerdas al mismo tiempo. Tomada de [Bar12]. 10
Figura 2.7 Respuestas $Hb(k)$ aplicadas al blanqueamiento del espectro. Tomada de [Kla06b]. 10
Figura 2.8 Sistema propuesto por Barbancho. Tomada de [Bar12]. 11
Figura 2.9 HMM utilizado, se muestra tres estado únicamente. Cada estado corresponde a un único CFCs. 11
Figura 2.10 Transcripción y análisis manual de progresiones de acordes. (a) progresión de acordes $Do\ Mayor, La\ menor, Fa\ Mayor, Sol\ Mayor$ encontrada en muchas canciones populares de música pop. (b) Progresión encontrada en la canción "Sin documentos". Los números entre paréntesis indican la posición de la mano. Tomada de [Bar12]. 12
Figura 2.11 Número de cuerdas en la guitarra que emiten cierta nota musical en el acorde de Do mayor. (a) Primera posición. (b) Segunda posición. 13
Figura 2.12 Acorde de Do mayor interpretado en el piano en una octava cualquiera. 13
Figura 2.13 Número de teclas que se oprimen para la emisión del acorde de Do mayor en cualquier octava. 13
Figura 3.1 Evolución de la presión sonora en función del tiempo en un punto cualquiera del espacio. 16
Figura 3.2 Ubicación de las doce notas del sistema de música occidental. Tomada de [Nuñ07]. 17
Figura 3.3 Elementos constitutivos de la representación escrita de una pieza musical. 17
Figura 3.4 Claves habitualmente usadas en la representación de una pieza musical. (a) Clave de Sol. (b) Clave de Fa. (c) Clave de Do. 17
Figura 3.5 Ubicación de las notas musicales en el pentagrama. (a) Orden en clave de Sol. (b) Orden en clave de Fa. 18
Figura 3.6 Representación de notas alteradas. 18
Figura 3.7 Notación utilizada en la música para hacer referencia a la frecuencia de las notas musicales. 18
Figura 3.8 Bandas estandarizadas de octava y 1/3 de octava. 19
Figura 3.9 Diferencia de frecuencias entre una nota y su anterior y posterior. 21
Figura 3.10 Figuras usadas para indicar la duración de una nota. Tomada de [Jac58]. 22
Figura 3.11 Agrupación de corcheas, semicorcheas y figuras de duración menor. El agrupamiento se suele dar de modo que la duración de cada grupo sea el de una negra. 22
Figura 3.12 Compás de cuatro cuartos. El denominador de la fracción nos indica que se utilizarán cuartos para representar la duración de un pulso y el numerador nos indica que se agruparán cuatro pulsos por compás. 22
Figura 3.13 Intervalos musicales. Tomada de [Esl03]. 23

Figura 3.14 Escala cromática. Al centro se muestra las notas que componen la escala cromática de Do. En la parte superior los intervalos entre una nota y su posterior expresados en semitonos (S). En la parte inferior se muestran los intervalos entre la raíz de la escala y las demás notas expresados en tonos (T). 24
Figura 3.15 Escala de Do mayor. 24
Figura 3.16 Escala de Mi mayor. 24
Figura 3.17 Escala de La menor. 25
Figura 3.18 Armaduras de las escalas mayores y sus relativas menores para la clave de Sol. Tomada de [Rod14]. 26
Figura 3.19 Ejemplo de melodía. 26
Figura 3.20 Ejemplo de armonía. 26
Figura 3.21 Acorde de Do Mayor en estado original. Tomada de [Cre10]. 27
Figura 3.22 Acorde de Do Mayor. Primer acorde: Estado original. Segundo acorde: Primera inversión. Tercer acorde: Segunda inversión. 27
Figura 3.23 Formato para la cabecera de un archivo MIDI. (a) Valores hexadecimales para la cadena de caracteres "MThd". (b) Longitud del resto de la cabecera. (c) Tipo de archivo MIDI. (d) Número de pistas. (e) Velocidad de los acontecimientos. 29
Figura 3.24 Composición del Track chunk. (a) Cadena de caracteres "MTkr". (b) Longitud en bytes del resto del fragmento. (c) Instrucciones MIDI. (d) Instrucción de fin de track. 30
Figura 3.25 Sintaxis para las instrucciones MIDI. (a) Cadena de instrucciones MIDI. (b) Ejecución de dos eventos simultáneamente, en este caso los eventos evento2 y evento3, ambos se activarán un número de pulsos igual a delta2 después de la ejecución del evento evento1 30
Figura 3.26 Codificación para la activación de una nota. (a) Código de Note on. (b) Canal. (c) Nota a activar usando su código MIDI. (d) Velocidad de activación. 32
Figura 3.27 Codificación para la desactivación de una nota. (a) Código de Note off. (b) Canal. (c) Nota a desactivar usando su código MIDI. (d) Velocidad de desactivación. 32
Figura 3.28 Emisión de una nota Do4 por un periodo de 128 pulsos. (a) Delta time de activación. (b) Evento Note on. (c) Código MIDI de Do4. (d) Velocidad de activación. (e) Delta time de desactivación, 12810 pulsos. (f) Evento Note off. (g) Código MIDI de Do4. (h) Velocidad de desactivación. 32
Figura 3.29 Mensaje de cambio de control para la modulación de una nota. (a) Delta time de activación. (b) Instrucción de cambio de control. (c) Instrucción de modulación (d) Valor de la modulación. 33
Figura 3.30 Señal de la nota Do3 emitida por un piano. En la parte superior se observa el ruido emitido al comenzar la emisión de la nota. En la parte inferior se observa la naturaleza cuasi-periódica de la señal pasados unos 20 ms de comenzar la emisión. Tomada de [Haw93] 34
Figura 3.31 Espectro de la nota Do4. Tomada de [Haw93]. 35
Figura 3.32 Efecto de ventaneo en el espectro de una señal sinusoidal. Tomada de [Man11]. 37
Figura 3.33 Características de las funciones ventana en el dominio del tiempo y el dominio de la frecuencia. Tomada de [Man11]. 38
Figura 3.34 Ejemplo de filtros utilizados para la reducción de las dimensiones del logaritmo del espectro. 38
Figura 3.35 Procedimiento para la obtención de los MFCCs de una señal. 39
Figura 3.36 Neurona artificial. Los escalares wi son los pesos sinápticos, los cuales se multiplican por las componentes del vector de entrada correspondientes. El escalar b se conoce como bias. 40
Figura 3.37 Funciones usualmente usadas en las neuronas artificiales. (a) Función lineal. (b) Función límite duro. (c) Función sigmoidal. (d) Función tangente hiperbólica. Tomada de [IRT14]. 40
Figura 3.38 Capa de neuronas artificiales. Tomada de [Dem02]. 41
Figura 3.39 HMM compuesto de dos estados ocultos y tres símbolos observables. 42
Figura 3.40 Cómputo del cromagrama. (a) Transformación de escala de frecuencia a código MIDI vía ecuación (3.23). (b) Sumatoria vía ecuación (3.24), el resultado se normaliza. 44
Figura 4.1 Proceso de generación y análisis de pistas de audio. 45
Figura 4.2 Cambio de nota. 45
Figura 4.3 Emisión repetida de una nota musical. 46
Figura 4.4 Silencio entre notas. 46
Figura 4.5 Esquema general para la identificación de emisiones sonoras, para el caso de este esquema pueden identificarse notas musicales, acordes mayores y acordes menores. 48

Figura 4.6 Diagrama general para la obtención de partituras. 49
Figura 5.1 Análisis por autocorrelación. (a) Segmento de la señal. (b) Autocorrelación del segmento dado. (c) Límites de búsqueda para T0. 51
Figura 5.2 Análisis por autocorrelación. (a) Segmento de la señal ventaneada. (b) Autocorrelación del segmento dado. (c) Límites de búsqueda para T0. 52
Figura 5.3 Análisis de una melodía por autocorrelación. Parte superior: Señal de audio. Centro: Vector de T0 obtenidas. Parte inferior: Vector de F0 obtenidas. 52
Figura 5.4 Vector de frecuencias fundamentales con irregularidades. 53
Figura 5.5 Vector de frecuencias fundamentales libre de irregularidades. 53
Figura 5.6 Comparación entre partituras. (a) Partitura original, usada para la síntesis de la señal de audio. (b) Partitura obtenida a través del análisis por autocorrelación. 54
Figura 5.7 Transcripción realizada de un archivo de la base de datos LabROSA. (a) Archivo MIDI original. (b) Transcripción generada a partir del audio de la señal. 54
Figura 5.8 Análisis a través de la transformada de Fourier. Superior izquierda: Segmento de la señal a analizar. Inferior izquierda: Segmento ventaneado. Superior derecha: Espectro del segmento. Inferior derecha: Componentes del espectro que superan el umbral establecido. 55
Figura 5.9 Análisis del espectro del acorde de Do Mayor. Superior izquierda: Segmento de la señal a analizar. Inferior izquierda: Segmento ventaneado. Superior derecha: Espectro del segmento, se indica el umbral establecido (60%). Inferior derecha: Componentes del umbral establecido. 56
Figura 5.10 Resultados de la clasificación de una red neuronal entrenada con MFCCs 57
Figura 5.11 Salidas de la red neuronal al alimentar la misma con vectores MFCCs de notas fuera de la octava dos. 58
Figura 5.12 Resultados de la clasificación de notas haciendo uso de MFCCs (línea verde) y LPCs (línea azul). 58
Figura 5.13 Resultados para la red neuronal entrenada con cuarenta y ocho notas. 59
Figura 5.14 Primeros doce acordes mayores sintetizados. 59
Figura 5.15 Resultados de la clasificación de acordes mayores haciendo uso de MFCCs (línea verde) y LPCs (línea azul). 59
Figura 5.16 Resultados de la clasificación de 48 acordes mayores usando redes neuronales con diferentes números de neuronas en la capa oculta. 60
Figura 5.17 Señal a analizar haciendo uso del algoritmo propuesto. 61
Figura 5.18 Análisis de energía de la señal a intervalos de 0.5 s. Cada intervalo se numera en la parte superior. 61
Figura 5.19 Limpieza de las instrucciones MIDI. A la derecha se muestran las partituras y a la izquierda el correspondiente vector de frecuencias. (a) Pista original. (b) Resultado obtenido al hacer uso del algoritmo propuesto sin limpieza de instrucciones. (c) Limpieza de instrucciones eliminando intervalos de duración corta. (d) Limpieza de instrucciones haciendo uso de redondeo de duraciones. (e) Limpieza de instrucciones aplicando eliminación de intervalos de corta duración y posteriormente redondeo de intervalos. 65
Figura 5.20 Partituras obtenidas realizando la limpieza de las instrucciones MIDI. (a) Pista original. (b) Resultado obtenido al hacer uso del algoritmo propuesto sin limpieza de instrucciones. (c) Limpieza de instrucciones eliminando intervalos de duración corta. (d) Limpieza de instrucciones haciendo uso de redondeo de duraciones, tomando un octavo de negra como duración mínima (e) Limpieza de instrucciones haciendo uso de redondeo de duraciones, tomando un dieciseisavo de negra como duración mínima (f) Limpieza de instrucciones haciendo uso de redondeo de duraciones, tomando un treintaidosavo de negra como duración mínima (g) Limpieza de instrucciones aplicando eliminación de intervalos de corta duración y posteriormente redondeo de intervalos a octavos de negra. 66
Figura 5.21 Modelo oculto de Markov propuesto para la identificación de notas musicales. En la parte izquierda se observa el estado de emisión y en la derecha el estado de no emisión. 67
Figura 5.22 Diagrama del uso de HMM para el reconocimiento de notas musicales. 69
Figura 5.23 HMM propuesto para el reconocimiento de melodías de tres notas. 71
Figura 5.24 HMM simplificado para el reconocimiento de secuencias de tres notas. 72
Figura 5.25 Uso del HMM propuesto para la identificación de melodías. 72
Figura 5.26 Uso de la segunda aproximación de los HMM. A la izquierda se muestra la secuencia de códigos MIDI. A la derecha se muestran los estados (notas musicales) entregados por el algoritmo de Viterbi. 73

Figura 5.27 Secuencia MIDI obtenida al analizar una señal de audio de la nota Do 4 al aplicársele un vibrato..... 74
Figura 5.28 Densidad de probabilidades para el estado de emisión en HMM que identifica la emisión de vibrato sobre la nota Sol 4. 75
Figura 5.29 Densidad de probabilidades para el estado de no emisión en HMM que identifica la emisión de vibrato sobre la nota Sol 4. 75
Figura 5.30 Aplicación del algoritmo forward a secuencias de vibratos de diez símbolos a sus correspondientes HMM. 76
Figura 5.31 Identificación de vibrato haciendo uso del algoritmo forward, aplicando el mismo a cada HMM propuesto. 76

1 Introducción

1.1 Planteamiento del problema

Con este trabajo se busca aplicar diversas técnicas tomadas del procesamiento digital de voz a señales provenientes de un instrumento musical dado adaptando o extendiendo las mismas para su uso en la extracción de características de interés en el lenguaje musical.

1.1 Justificación

En su tesis doctoral Klapuri [Kla04] asegura que el problema de la extracción de características de una señal de audio, para su representación en un lenguaje propio de la música, es comparable al reconocimiento automático de voz, lo cual ha sido estudiado por más de cincuenta años, por lo que los métodos teóricos y enfoques del reconocimiento de voz pueden ser utilizados para su aplicación en señales de audio provenientes de instrumentos musicales. Para la representación gráfica de una pista musical a ser interpretada por un músico pueden utilizarse diversas notaciones. Seguramente la más utilizada es el pentagrama, comúnmente utilizada en occidente. La Figura 1.1 muestra un ejemplo de esta notación.

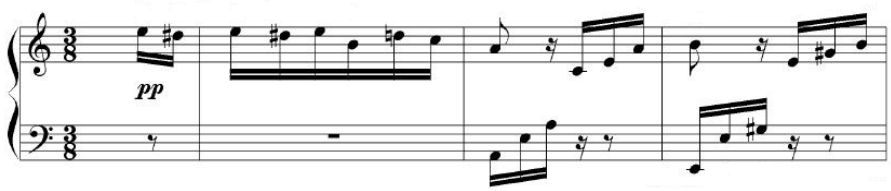
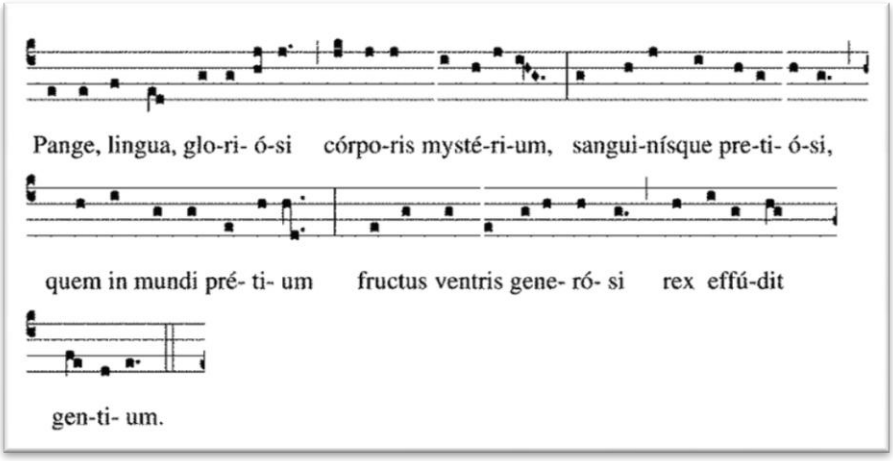


Figura 1.1 Pentagrama de los primeros cuatro compases de la obra "Für Elise" de Ludwing van Beethoven.

Cabe destacar que esta representación no es única, ejemplo de esto son las notaciones hindú y gregoriana representadas en la Figura 1.2.

(a)



(b)

Figura 1.2 Otras representaciones utilizadas para expresar piezas musicales. (a) Notación hindú, tomada de [Cou12]. (b) Notación gregoriana, fragmento de la obra "Pangue lingua gloriosi" de Santo Tomás de Aquino.

Cada una de estas notaciones responde a una necesidad dada, pero todas comparten características fundamentales, mismas que se pueden obtener de la señal de audio. Dos de las principales características necesarias para la representación simbólica de una pista musical son la altura y duración. La altura de un sonido (*pitch* en inglés) está principalmente relacionada con la frecuencia de sonidos periódicos, e indica si un sonido es más o menos agudo [Per03]. De acuerdo al estándar del ANSI (*American National Standards Institute*) en 1994, se define la altura como el atributo de la sensación auditiva a través de la cual los sonidos pueden ser ordenados en una escala que va de grave a agudo. Por otro lado la altura puede ser definida operacionalmente como sigue: se dice que un sonido tiene cierta altura si puede ser emparejado ajustando la frecuencia de un tono sinusoidal de amplitud arbitraria [Har97]. Con lo mencionado anteriormente puede encontrarse la correspondencia entre la frecuencia de una señal sonora y la notación utilizada en el lenguaje musical, como se muestra en la Figura 1.3, donde la relación entre la frecuencia de una nota y su consecutiva corresponde a un doceavo de octava.

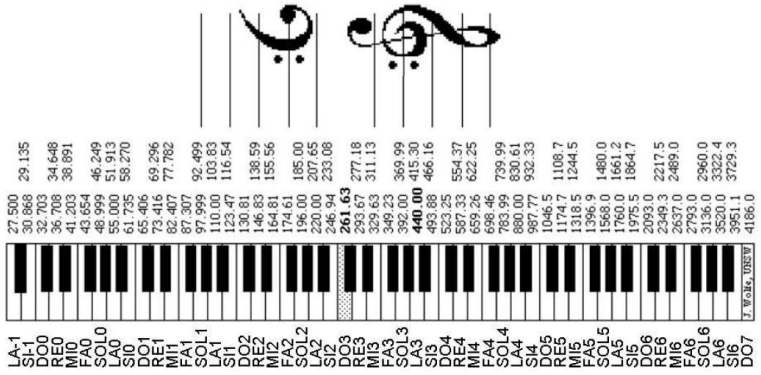


Figura 1.3 Correspondencia entre la frecuencia de un señal de audio y la notación propia de los músicos. Tomada de [Per03].

De esto queda en evidencia la necesidad de hacer uso de métodos de estimación de la frecuencia fundamental F0 de una señal monofónica o bien de múltiples F0 para pistas polifónicas.

Una problemática importante en la detección de múltiples F0 de pistas producidas por instrumentos reales, es que estos producen una señal de audio que contiene no solo la frecuencia correspondiente a la nota

generada, sino un amplio contenido espectral. A estas frecuencias se le conoce como armónicos. El contenido armónico para una misma nota musical varía de acuerdo al instrumento que la produce, lo que se traduce en la sensación sonora conocida como timbre, el cual nos permite identificar distintos instrumentos musicales.

El variado contenido armónico de cada instrumento musical dificulta la tarea de encontrar la F0, especialmente cuando se trabaja con pistas polifónicas.

Es importante considerar también que las características buscadas en una señal de audio cambian en función del tiempo, por lo que se hace necesario conocer los intervalos temporales en que determinada frecuencia está presente, es decir, conocer tanto las características temporales como espectrales. La Figura 1.4 muestra el espectrograma de una señal de audio generada con ondas sinusoidales y su correspondencia con la notación usada en occidente.

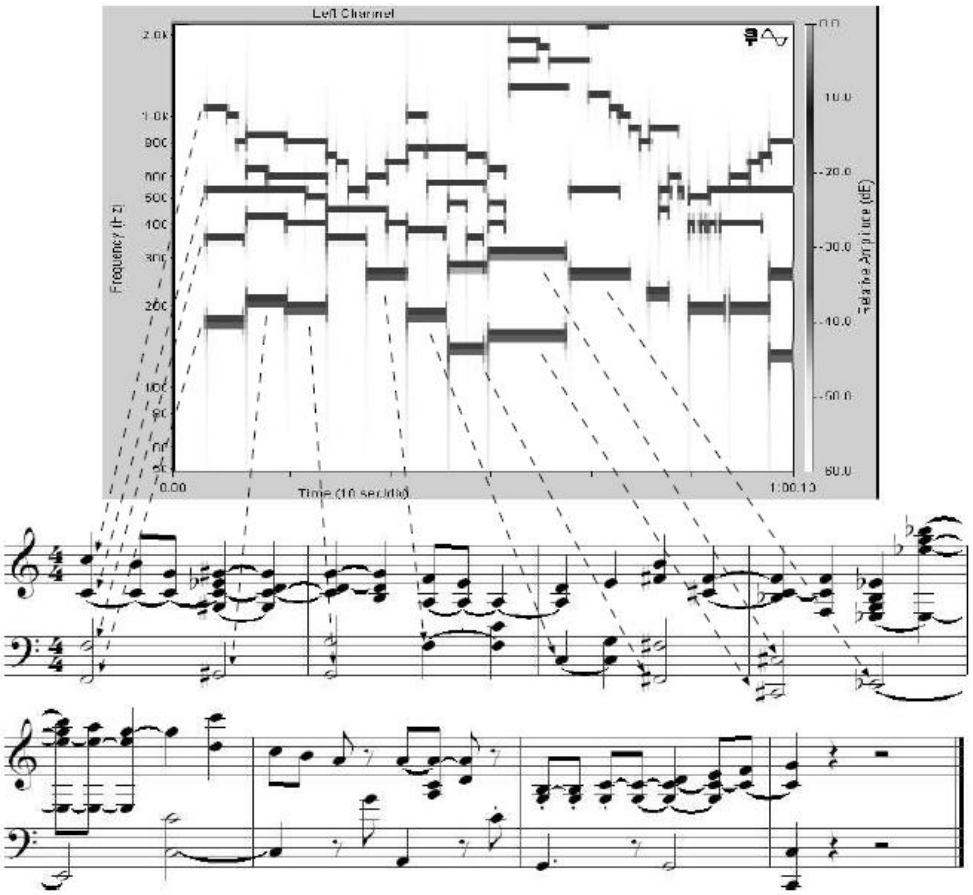


Figura 1.4 Espectrograma de una melodía generada con una onda sinusoidal y su partitura correspondiente. Se muestran las correspondencias entre algunas bandas espectrales y las notas que producen. Tomada de [Per03].

1.2 Objetivo general

Aplicar métodos de procesamiento digital de voz a señales de audio generadas por un instrumento musical, adaptando y extendiendo los mismos para la extracción de características de interés en el lenguaje musical.

1.3 Objetivos particulares

- Identificación de notas musicales vía análisis de F0.
- Identificación de acordes usando MFFCs y redes neuronales.
- Uso de HMM para la corrección de secuencias de emisiones sonoras y reconocimiento de vibratos.
- Uso del análisis de energía de la señal para la estimación de la duración de las emisiones sonoras en una pista de audio a modo de obtener su representación en figuras propias del lenguaje musical.
- Desarrollo de una librería de funciones que permitan la generación de archivos MIDI.

1.4 Alcances del trabajo

En el presente trabajo se muestra un sistema compuesto por una serie de redes neuronales que permiten realizar un análisis general de la estructura de una pista de audio clasificando las emisiones sonoras que lo integran. Este sistema se entrenó para el análisis de pistas provenientes de un único instrumento musical.

Una vez identificadas las emisiones sonoras como notas musicales, acordes y vibratos se proponen una serie de técnicas que permiten corregir posibles errores en la clasificación usando información del contexto de la señal.

Así mismo se propone un algoritmo compatible con las técnicas anteriores que permite la generación de instrucciones en formato MIDI las cuales permiten la representación de las mismas en forma de partitura, misma que será una aproximación de la señal analizada.

2 Estado del arte

El tema de la extracción de características útiles de señales de audio para su representación en un lenguaje propio de la música se remonta a los años setenta, cuando Moorer [Moo75] presentó un análisis de técnicas de procesamiento digital de señales que resultaban útiles para el propósito de la transcripción. Además de esto construyó un sistema para transcribir composiciones a dos voces. Este sistema sufría de fuertes limitaciones en cuanto a la relación de frecuencias permitidas entre dos notas simultáneas y en cuanto al rango de notas utilizadas. En los años 80´s un grupo de investigadores de Stanford continuó el trabajo de Moorer, pero hasta ese entonces la polifonía seguía limitada a dos voces.

A finales de los ochenta, la Universidad de Osaka en Japón inició un proyecto que tenía como objetivo la extracción de sentimientos de las señales musicales y la construcción de un robot que pudiera responder a la música de la misma manera en la que lo hace un humano. Durante el proyecto se diseñó un sistema de transcripción polifónica de hasta cinco voces simultáneas, pero que presentaba errores a la salida.

En 1993 Hawley [Haw93] publicó una tesis sobre el análisis computacional de la escena auditiva y se enfrentó al también con el problema de la transcripción polifónica de composiciones para piano. Hawley, entre otras cosas, propone la normalización de los sensores de audio a modo de que produzcan descriptores de eventos acústicos, como el formato MIDI (Musical Instrument Digital Interface).

Uno de los avance significativos en la extracción de características de señales de audio se logró cuando investigadores de la Universidad de Tokio publicaron un sistema de transcripción que incluía nuevas técnicas. El esquema del sistema mencionado puede observarse en la Figura 2.1.

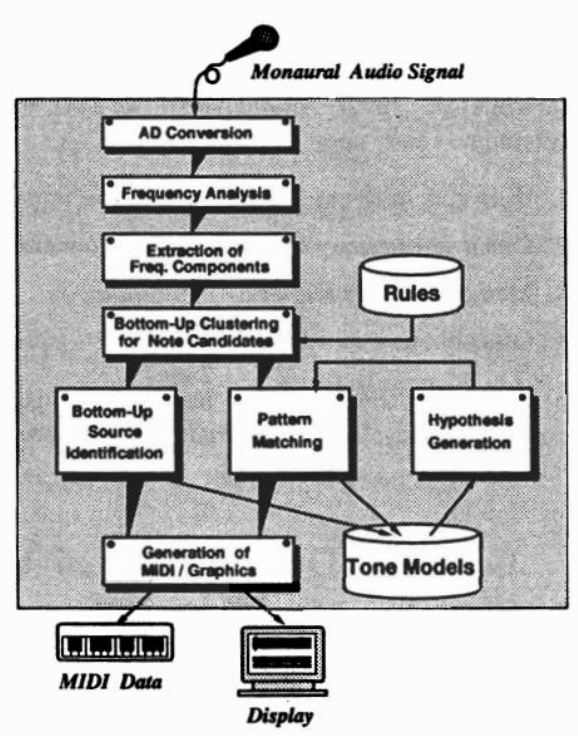


Figura 2.1 Sistema de transcripción de música propuesto por Kashino y Tanaka. Tomada de [Kas93].

Tanaka y Kashino fueron los primeros en detallar y utilizar de manera precisa reglas de separación auditiva, proponiendo un algoritmo para modelar tonos automáticamente. Posteriormente, en 1995 [Kas95]

mejoraron su sistema al emplear la llamada arquitectura *blackboard,* que parece favorecer el proceso de extracción y transcripción. En la Figura 2.2 se presenta un diagrama del sistema mencionado.

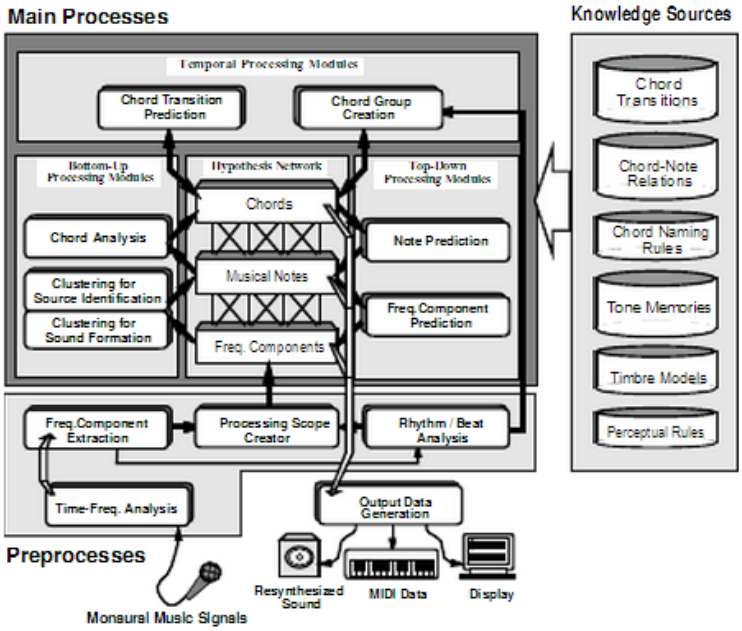


Figura 2.2 Sistema para la transcripción automática de música. Tomada de [Kas95].

La arquitectura antes mencionada utiliza redes Bayesianas de Pearl para propagar la información a través del sistema.

Para el año 2004 Klapuri [Kla04], en su tesis doctoral realiza un análisis de métodos de procesamiento digital de señales para la transcripción automática de música, en donde concluye que la transcripción automática de música es un problema "difícil", comparable con el reconocimiento de voz. Además asegura que faltan años, quizá décadas, antes de alcanzar, digamos un 95% de la precisión y flexibilidad de un músico profesional. En su trabajo Klapuri puntualiza que las señales musicales son candidatas naturales para el problema de la estimación de múltiples frecuencias fundamentales (F0), de la misma manera en que las señales de voz son candidatas naturales para la estimación de una única F0.

En 2007 Papadopoulos [Pap07] propone un modelo en donde la señal observada proviene del cromagrama de la señal (véase la sección 3.18.9 Cromagrama) y la progresión de acordes es representada usando Modelos Ocultos de Markov. Para comenzar son su análisis la señal es decimada hasta 11025 Hz, convertida a mono y llevada al domino de la frecuencia vía Trasformada Discreta de Fourier usando una ventana Blackman con una longitud de 0.48 s con 12.5% de traslape. Debido a la baja resolución en frecuencia en baja frecuencia Papadopoulos considera frecuencias por encima de 60 Hz y fija como límite superior 1 kHz debido a que las fundamentales y harmónicos en la música popular son usualmente más fuertes que las componentes no harmónicas por encima de 1 KHz. Para la clasificación Papadopoulos considera un HMM ergódico (existe probabilidad de transición entre un estado y cualquier otro) con cada estado representando un acorde, usando un léxico de 24 triadas mayores y menores en el conjunto $\{C\ mayor, C\#\ mayor, \ldots, B\ mayor, Do\ menor, \ldots, B\ menor\}$ aunque no especifica la octava a la que pertenecen. En cada estado en el modelo genera un vector observado, este es, la característica croma con cierta probabilidad. Papadopoulos propone tres aproximaciones para definir las probabilidades de observación. El primer método aprende las probabilidades por entrenamiento de un modelo Gaussiano en vectores croma normalizados. En el segundo método no utiliza un conjunto de entrenamiento sino que se definen las probabilidades con base en reglas de la teoría musical únicamente. El tercer enfoque es similar al

segundo pero las probabilidades se definen con base en una medición de correlación normalizada. En todos los casos el vector de probabilidades iniciales π asigna una probabilidad de $1/24$ para cada uno de los 24 estados, pues a priori no se conoce el acorde con el que una pieza musical comienza.

Por otro lado la probabilidad de transición entre dos acordes se basa en algunas reglas de teoría musical, más específicamente el círculo de quintas doblemente anidado. Éste círculo representa una relación existente entre las 12 notas de la escala cromática. Si bien no es posible saber qué estado o acorde sigue a otro Papadopoulos se basa en algunas reglas de la teoría musical para hacer hipótesis sobre qué estados son más probables. Como ejemplo menciona que en la música occidental después de emitirse un acorde $La\ mayor$ es más probable que se emita un acorde $F\#\ menor$ o un acorde $D\ mayor$ que un acorde $G\#\ mayor$. Con esto Papadopoulos reporta que los resultados utilizando estos tres métodos son muy similares entre sí, obteniendo un porcentaje de emisiones correctamente clasificadas de 70.96%

En el año 2010 Li, Chan y Chun [Li10] usaron Redes Neuronales Convolucionales (CNN por sus siglas en inglés) de cinco capas, las cuales tomaban vectores MFCCs como entrada. El objetivo de su modelo de CNN era la clasificación automática de género musical, tarea que en la actualidad sigue estudiándose. La Figura 2.3 muestra la arquitectura de su modelo.

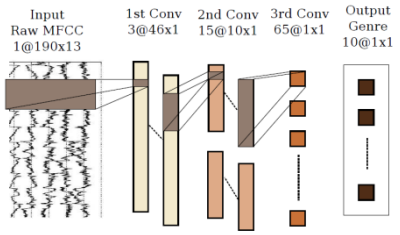


Figura 2.3 CNN para la extracción de patrones musicales. Tomada de [Li10].

En sus experimentos los vectores MFCC fueron extraídos de segmentos de $23\ ms$ con un 50% de sobreposición. Como base de datos utilizaron un conjunto de 1000 canciones con una duración de $30\ s$ a una frecuencia de muestreo de $22050\ Hz$ a $16\ bit$. Estas canciones se encuentran distribuidas entre diez diferentes géneros: Blues, Clásica, Country, Disco, Hip-hop, Jazz, Metal, Pop, Regguae y Rock.

La Figura 2.4 muestra la convergencia de la tasa de error de entrenamiento se su CNN a los cuales se iban agregando progresivamente subconjuntos de géneros. El subconjunto más pequeño consta de tres géneros musicales, mientras que el más grande consta de seis.

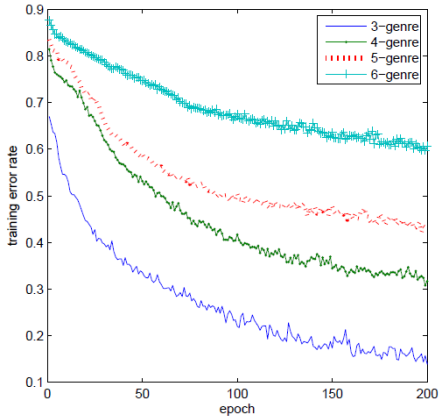


Figura 2.4 Curva de convergencia en entrenamiento de 200 épocas.

En sus resultados finales Li reporta una precisión del 84% con lo que concluyen que la variación en patrones musicales (después de algún tipo de transformación como la transformada de Fourier o MFCC) es similar a las presentadas en imágenes y por tanto pueden ser extraídos con CNN.

A pesar de esto al utilizar muestras de una base de datos distinta a la usada en el entrenamiento la precisión disminuya hasta 30%. Esto según Li revela que su modelo de extracción de características tiene deficiencia en generalizar patrones de aprendizaje para pistas no mostradas a su modelo. Por otro lado concluye que el vector de características MFCC es sensible a variaciones de timbre y armadura traduciéndose en una disminución en la precisión. Así mismo propone que una solución práctica a estos problemas es la utilización de una base de datos más grande con una mayor riqueza como cambios de tono y ligeros cambios de tempo.

También en 2010 Mauch y Dixon [Mau10] proponen un método para estimar, usando una señal de audio la secuencia de acordes presentes en la misma así como la nota más grave, posición de los acordes y clave. El núcleo de su método es una red Bayesiana dinámica de seis capas. Usando 109 acordes si método provee un detalle armónico sustancialmente mayor a enfoques previos manteniendo un nivel alto de precisión. De a cuerdo a Mauch con un 71% de notas correctamente clasificadas exceden de manera significativa el estado del arte al ser comparados con transcripciones hechas a mano usando las 176 pistas de audio del MIREX 2008, en el área de detección de acordes. Como objetivo de este trabajo Mauch busca ofrecer un sistema que eventualmente permita a los músicos generar las *lead sheet* (notación alternativa o complementaria a la partitura que especifica por medio de símbolos los acordes a interpretar) de manera automática. La Figura 2.5 muestra un ejemplo de *lead sheet*.

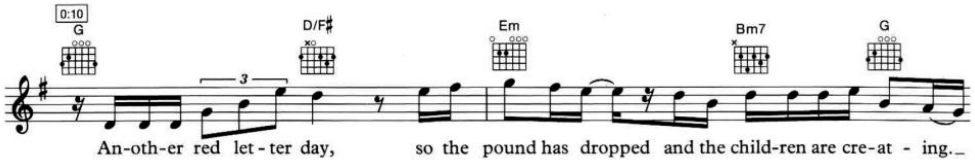


Figura 2.5 Ejemplo de lead sheet. Como puede observarse en la parte superior se especifican los acordes a interpretar usando la representación de un brazo de guitarra en lugar de colocar las notas en un pentagrama. Tomada de [Mau10].

Mauch explica que la selección de los 109 acordes que incluyó en su experimentación no es definitiva, pues por un lado MIREX en su tarea de detección de acordes incluye únicamente dos tipos de acordes, estos son, mayores y menores, mientras que por otro lado los acordes utilizados en la música pop son mucho más complejos.

2.1 Descripción de cómo se afronta esta problemática en la actualidad.

Actualmente se investigan nuevas técnicas de procesamiento digital de señales para la extracción de características de señales de audio. Benetos [Ben12] resume algunas de las técnicas que se investigan en la actualidad para la extracción de múltiples F0 de una señal de audio, como se muestra en la Tabla 2.1.

Tabla 2.1 Enfoques para la estimación de múltiples F0 de una señal de audio utilizando representaciones tiempo-frecuencia [Ben12].

Time-Frequency Representation	Citation
Short-Time Fourier Transform	[Abd02, AP04, AP06, BJ05, BED09a, BBJT04, BBFT10, BBST11, BKTB12, Bel03, BDS06, BMS00, BBR07, BD04, BS12, Bro06, BG10, BG11, CLLY07, OCR+08, OCR+09b, OCR+09a, OCQR10, OVC+11, CKB03, Cem04, CKB06, CSY+08, CJAJ04, CJJ06, CJJ07, CSJJ07, CSJJ08, Con06, DG03, DGI06, DCL10, Dix00, DR93, DZZS07, DHP09, DHP10, DPZ10, DDR11, EBD07, EBD08, EBD10, FHAB10, FK11, FCC05, Fon08, FF09, GBHL09, GS07a, GD02, GE09, GE10, GE11, Gro08, GS07a, Joh03, Kla01, Kla03, Kla04b, Kla06, Kla09a, Kla09b, KT11, LYLC10, LYC11, LYC12, LW07, LWB06, Lu06, MSH08, NRK+10, NRK+11, NLRK+11, NNLS11, NR07, Nie08, OKS12, OP11, ONP12, OS03, OBBC10, BQ07, QRC+10, CRV+10, PLG07, PCG10, PG11, Pee06, PI08, Per10, PI04, PI05, PI07, PI08, Per10, PI12, PAB+02, PEE+07, PE07a, PE07b, QCR+08, QCR+09, QCRO09, QRC+10, CRV+10, CQRSVC+10, ROS09a, ROS09b, RVBS10, Rap02, RFdVF08, RFF11, SM06, SC10, SC11, SB03, Sma11, Smi00, TL05, VK02, YSWJ10, WL06, Wel04, WS05, Yeh08, YR04, YRR05, YRR10, YSWS05, ZCJM10]
ERB Filterbank	[BBV09, BBV10, KT99, Kla04b, Kla05, Kla08, RK02, Ryy08, RK08, TK00, VR04, VBH07, VBH08, VBB10, ZLLX08]
Constant-Q Transform	[Bro92, CJ02, CPT09, CTS11, FBR11, KDK12, Mar12, MS09, ROS07, Sma09, Wag03, WVR+11b, WVR+11a]
Wavelet Transform	[FCC05, KNS04, KNS07, MKT+07, NEOS09, PHC06, SIOO12, WRK+10, YG10, YG12a]
Constant-Q Bispectral Analysis	[ANP11, NPA09]
Resonator Time-Frequency Image	[ZR07, ZR08, ZRMZ09, Zho06, BD10b, BD10a]
Multirate Filterbank	[CQ98, Got00, Got04]
Reassignment Spectrum	[HM03, Hai03, Pee06]
Modulation Spectrum	[CDW07]
Matching Pursuit Decomposition	[Der09]
Multiresolution Fourier Transform	[PGSMR12, KCZ09, Dre11]
Adaptive Oscillator Networks	[Mar04]
Modified Discrete Cosine Transform	[SC09]
Specmurt	[SKT+08]
High-resolution spectrum	[BLW07]
Quasi-Periodic Signal Extraction	[TS09]

Así mismo Benetos ofrece un compilado de métodos de procesamiento digital de señales para abordar la extracción de múltiples F0, ver Tabla 2.2.

Tabla 2.2 Técnicas de procesamiento digital de señales para la extracción de múltiples F0 [Ben12].

Technique	Citation
Signal Processing Techniques	[ANP11, BBJT04, BBFT10, BBST11, BKTB12, BLW07, Bro06, Bro92, CLLY07, OCR+08, OCR+09b, OCR+09a, Dix00, Dre11, DZZS07, FHAB10, CQ98, FK11, Gro08, PGSMR12, HM03, Hai03, Joh03, KT99, Kla01, Kla03, Kla04b, Kla05, Kla06, Kla08, LRPI07, LWB06, NPA09, BQ07, PHC06, PI07, PI08, Per10, PI12, QCR+09, QCRO09, CQRSVC+10, SKT+08, SC09, TK00, Wag03, WZ08, YSWJ10, WL06, WS05, YR04, YRR05, Yeh08, YRR10, YSWS05, ZLLX08, Zho06, ZR07, ZR08, ZRMZ09]
Maximum Likelihood	[BED09a, DHP09, DPZ10, EBD07, EBD08, EBD10, FHAB10, Got00, Got04, KNS04, KNS07, KT11, MKT+07, NEOS09, NR07, Pee06, SIOO12, WRK+10, WVR+11b, WVR+11a, YG10, YG12b, YG12a]
Spectrogram Factorization	[BBR07, BBV09, BBV10, OVC+11, Con06, CDW07, CTS11, DCL10, DDR11, FBR11, GE09, GE10, GE11, HBD10, HBD11a, HBD11b, KDK12, Mar12, MS09, NRK+10, NRK+11, NLRK+11, Nie08, OKS12, ROS07, ROS09a, ROS09b, SM06, SB03, Sma04b, Sma09, Sma11, VBB07, VBB08, VBB10, VMR08]
Hidden Markov Models	[BJ05, CSY+08, EP06, EBD08, EBD10, LW07, OS03, PE07a, PE07b, QRC+10, CRV+10, Rap02, Ryy08, RK05, SC10, SC11, VR04]
Sparse Decomposition	[Abd02, AP04, AP06, BBR07, BD04, OCQR10, CK11, Der06, GB03, LYLC10, LYC11, LYC12, MSH08, OP11, ONP12, PAB+02, QCR+08]
Multiple Signal Classification	[CJAJ04, CJJ06, CSJJ07, CJJ07, CSJJ08, ZCJM10]
Support Vector Machines	[CJ02, CPT09, EP06, GBHL09, PE07a, PE07b, Zho06]
Dynamic Bayesian Network	[CKB03, Cem04, CKB06, KNK198, ROS09b, RVBS10]
Neural Networks	[BS12, GS07a, Mar04, NNLS11, OBBC10, PI04, PI05]
Bayesian Model + MCMC	[BG10, BG11, DGI06, GD02, PLG07, PCG10, PG11, TL05]
Genetic Algorithms	[Fon08, FF09, Lu06, RFdVF08, RFF11]
Blackboard System	[BMS00, BDS06, Bel03, McK03]
Subspace Analysis Methods	[FCC05, VR04, Wel04]
Temporal Additive Model	[BDS06, Bel03]
Gaussian Mixture Models	[Kla09a, Mar07]
Least Squares	[Kla09b, KCZ09]

Como puede observarse la gama de enfoques utilizados en la actualidad es muy amplio para un único problema, la estimación de múltiples F0 de una señal de audio. De igual manera otros problemas como la detección automática de bits por minuto (BPM), escala, valor de las notas y detección de los valores del compás son objeto de investigaciones que arrojan una serie de enfoques distintos.

Por otro lado, también en el 2012, Barbancho [Bar12] propone un método para la extracción de posiciones de guitarra para señales de audio grabadas, en donde consideró 330 posiciones distintas, correspondientes a diferentes versiones de acordes mayores, menores, acordes mayores de séptima y acordes menores de séptima interpretados en una guitarra. Su método se formula como un Modelo Oculto de Markov (HMM) donde los estados ocultos corresponden a diferentes posiciones y las características ocultas son obtenidas por medio de estimadores de múltiple frecuencia fundamental. La transición entre posiciones se limitaron por un modelo musical entrenado a partir de una base de datos de secuencias de acordes y una función heurística de costo que mide la dificultad de ir de una posición dada a otra. La Figura 2.6 muestra dos posiciones para emitir el acorde de $Do\ mayor$ en la guitarra.

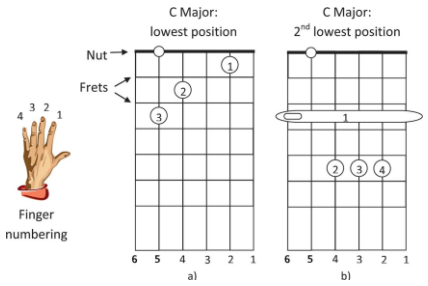


Figura 2.6 Ejemplo de posiciones para la emisión del acorde de Do mayor. Las líneas verticales indican las seis cuerdas y los números dentro de los círculos sobre la cuerdas indican la posición de los dedos. En el caso de b) se muestra una barra en donde el dedo índice presiona varias cuerdas al mismo tiempo. Tomada de [Bar12].

En la Figura 2.6 las líneas verticales representan las cuerdas de la guitarra, mismas que se encuentran numeradas del uno al seis y que corresponden a las notas Si, Mi, Sol, Re, La, Mi (siguiendo la numeración), por otro lado las líneas verticales separan el brazo de la guitarra en filas, donde cada fila representa el llamado traste en la guitarra. Para conocer la nota que se emite al presionar una cuerda en cierto traste es necesario realizar el recorrido sobre la escala cromática comenzando desde la nota a la que corresponde la cuerda hasta llegar al traste que se presiona, por ejemplo, en la Figura 2.7a se observa que se oprime la quinta cuerda en el tercer traste usando el dedo tres; la quinta cuerda emite la nota La cuando se rasga y no se oprime en ninguno de sus trastes (cuerda al aire), mientras que al oprimir el tercer traste se emite la nota Do, pues se cuentan tres notas de la escala cromática ($Do, Do\#, Re, Re\#, Mi, Fa, Fa\#, Sol, Sol\#, La, La\#, Si, Do$) a partir de la nota La exclusive. De esta manera para la Figura 2.7a, al rasguear todas las cuerdas, se emiten las notas Mi, Do, Sol, Mi, Do, Mi, mientras que para la Figura 2.7b al rasguear las seis cuerdas se emiten la notas $Sol, Mi, Do, Sol, Do, Sol$. En ambos casos siempre se emiten notas pertenecientes al acorde de $Do\ mayor$, es decir Do, Mi, Sol.

En el método propuesto se toma una señal grabada y se procesa con un estimador de múltiples frecuencias fundamentales el cual mide la fuerza (*salience*) de diferentes frecuencias candidatas a ser F0 por medio de la ecuación (2.3)

$$s(\tau) = \sum_{m=1}^{M} g(\tau, m)\left|Y\left(f_{\tau,m}\right)\right| \tag{2.3}$$

En donde la frecuencia fundamental $F0$ y el periodo τ se relacionan entre si por medio de la igualdad $F0 = f_s/\tau$, siendo f_s la frecuencia de muestreo. la función $g(\tau, m)$ es una función de peso y $Y(f_{\tau,m})$ es la Trnsformada de Fourier blanqueada por medio de un banco de filtros mostrado en la Figura 2.7.

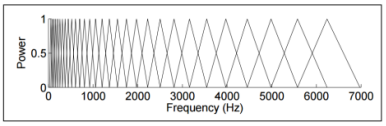


Figura 2.7 Respuestas $H_b(k)$ aplicadas al blanqueamiento del espectro. Tomada de [Kla06b].

Para llegar al espectro blanqueado $Y(k)$ se utiliza la igualdad $Y(k) = \gamma(k)X(k)$, donde $X(k)$ es el espectro de la señal y $\gamma(k)$ es la función obtenida al interpolar los coeficientes γ_b calculados por medio de la igualdad $\gamma_b = \sigma_b^{\nu-1}$ donde ν es un parámetro que mide el blanqueamiento del espectro y los escalares $\sigma_b$son obtenidos por medio de la ecuación (2.4).

$$\sigma_b = \left(\frac{1}{K}\sum_k H_b(k)|X(k)|^2\right)^{1/2} \tag{2.4}$$

En el caso de Barbancho el vector de observación $\boldsymbol{o}_t$ consistió en la estimación de 41 notas entre 82 Hz ($Mi2$) y 830 Hz (Sol#5). La Figura 2.8 muestra el sistema propuesto para la identificación de acordes.

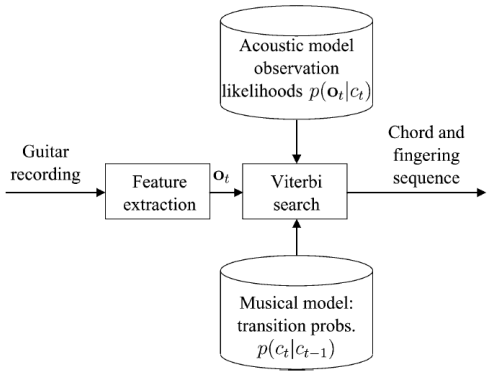


Figura 2.8 Sistema propuesto por Barbancho. Tomada de [Bar12].

La búsqueda de la secuencia de acordes y posición de los dedos en la guitarra (CFCs por sus siglas en inglés) se acotó por medio de dos modelos preentrenados: un modelo acústico que describe las probabilidades $p(\boldsymbol{o}_t|c_t)$ de observar un $\boldsymbol{o}_t$ dado un CFCs, y un modelo musicológico que determina la probabilidad $p(c_t|c_{t-1})$ de cambiar entre dos CFCs temporalmente sucesivos. De esta Barbancho formula un HMM donde cada estado oculto corresponde a un CFCs diferente. La Figura 2.9 muestra el HMM utilizado.

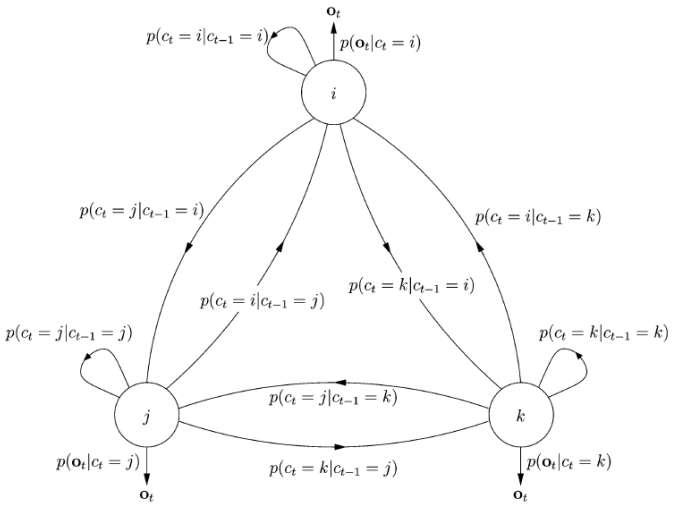
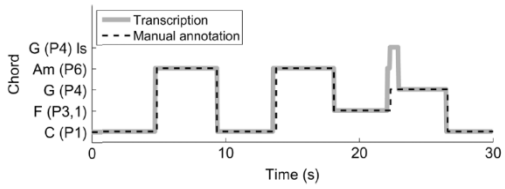
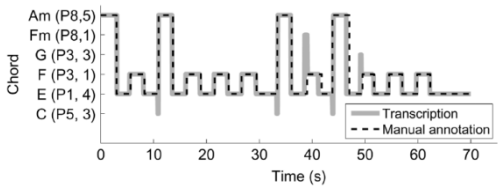


Figura 2.9 HMM utilizado, se muestra tres estado únicamente. Cada estado corresponde a un único CFCs.

La Figura 2.10 muestra los resultados obtenidos en la transcripción de un par de progresiones de acordes por medio del sistema descrito.



(a)



(b)

Figura 2.10 Transcripción y análisis manual de progresiones de acordes. (a) progresión de acordes $Do\ Mayor, La\ menor, Fa\ Mayor, Sol\ Mayor$ encontrada en muchas canciones populares de música pop. (b) Progresión encontrada en la canción "Sin documentos". Los números entre paréntesis indican la posición de la mano. Tomada de [Bar12].

Con esto Barbancho reporta los resultados mostrados en la Tabla 2.3, en donde se incluyen los resultados vía el método propuesto con una transición entre estados con distribución normal (PM) y una que toma en cuenta la dificultad física de ir de un acorde a otro (PHY), esto es, qué tan difícil es cambiar la mano de una posición a otra.

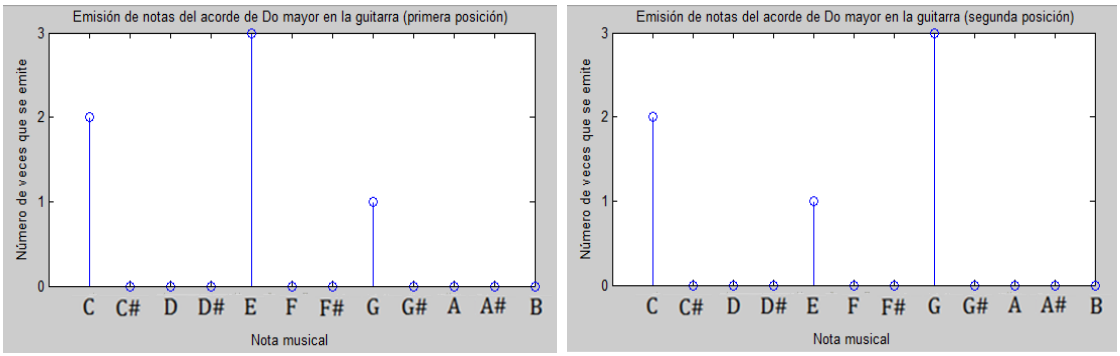
Tabla 2.3 Resultados del sistema propuesto por Barbancho.

Acordes y posiciones correctas (210 posibilidades, número de cuerdas rasgueadas no considerado)	
PM	PHY
88%	79%
Acordes, posición y número de cuerdas rasgueadas correctas (330 posibilidades)	
PM	PHY
87%	71%

La diferencia entre las 210 y 330 posibilidades radica en que un mismo acorde, en la guitarra tiene cuerdas que puedes ser o no rasgueadas, es decir el hacer vibrar cierta cuerda en particular es opcional.

En los trabajos citados anteriormente puede verse que el cromagrama es usado como vector de características, pero es importante señalar las limitaciones que se presentan en su uso, pues al ser éste una representación tan compacta pierde mucha información que podría ser de interés, principalmente el rango de frecuencias u octava a la que se genera alguna emisión sonora. Como se mencionó, las investigaciones que utilizan al cromagrama como vector de características se enfocan al análisis de señales de audio generadas por una guitarra, en la cual, para la identificación de algún acorde no es necesario, en principio, conocer las octavas de las notas que lo componen. Como ejemplo puede tomarse el acorde de $Do\ mayor$ en dos distintas posiciones, como se muestra en la Figura 2.6 y analizarse las notas que se están emitiendo.

Como puede observarse en ambos casos se emiten las tres notas que componen el acorde de $Do\ mayor$, éstas son Do, Re y Mi, pero en cada posición algunas notas se emiten en distinto número, por ejemplo, en la primera posición la nota Mi se emite mediante tres cuerdas, mientras que en la segunda posición solo se emite mediante una cuerda. Si se graficaran el número de cuerdas utilizadas para emitir una nota dentro del acorde de $Do\ mayor$ se obtendrían las gráficas mostradas en la Figura 2.11.



(a) (b)

Figura 2.11 Número de cuerdas en la guitarra que emiten cierta nota musical en el acorde de Do mayor. (a) Primera posición. (b) Segunda posición.

En principio las graficas de la Figura 2.11 serán una aproximación del cromagrama, evidentemente agregando cierto "ruido" debido a la estructura del instrumento y el contenido en frecuencia que ésta agrega. De esta manera puede verse que haciendo uso de este vector puede identificarse entre la primera y segunda posición del acorde de *Do mayor* sin necesidad de hacer un estudio de las octavas a las que pertenece cada nota, es decir, el cromagrama es suficiente como vector de características para la identificación de acordes generados por una guitarra.

Por otro lado se puede analizar las maneras en que se puede emitir el acorde de *Do mayor* en un instrumento distinto, digamos el piano. La forma fundamental de emitir un *Do mayor* consiste en presionar únicamente tres teclas (las correspondientes a las notas que componen el acorde) como se muestra en la Figura 2.12.

Figura 2.12 Acorde de Do mayor interpretado en el piano en una octava cualquiera.

Al analizar la Figura 2.12 puede observarse que no se especifica la octava en la que las notas se emitieron y es precisamente ahí donde radican las diferencias con el análisis de la guitarra, pues mientras en el piano es deseable conocer la octava a las que las notas pertenecen, en la guitarra en principio carece de importancia. Si se presenta una gráfica similar a la de la Figura 2.11, esta vez tomando en cuenta el número de teclas que se pulsan en el piano, se obtiene lo observado en la Figura 2.13.

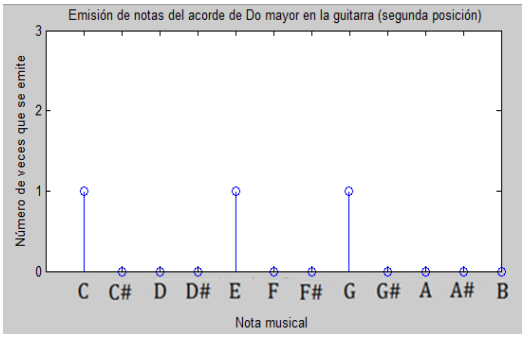


Figura 2.13 Número de teclas que se oprimen para la emisión del acorde de Do mayor en cualquier octava.

Aún más, en el piano el acorde de $Do\ mayor$ (de hecho cualquier acorde) puede interpretarse de distintas maneras haciendo que una o dos de las tres notas que lo componen se emitan en una octava más aguda que la de la nota Do, obteniéndose así las llamadas $inversiones$ del acorde. En todos los casos, ya sea al interpretar el acorde en distintas octavas o al interpretar una inversión del mismo la gráfica presentada en la Figura 2.13 será exactamente la misma y al ser ésta una aproximación del cromagrama puede verse que este vector no ofrece la suficiente información para realizar una clasificación lo suficientemente fina cuando se analizan emisiones sonoras provenientes de un piano. Es por ello que se presenta la necesidad de buscar un vector de características que ofrezca más información de interés a modo de poder dar una descripción más específica de una señal de audio.

A modo de resumen se presenta la Tabla 2.4, que muestra los trabajos más representativos con objetivos similares al del presente trabajo.

Tabla 2.4 Trabajos representativos del estado del arte con objetivos similares al del presente trabajo.

Trabajo	Instrumento analizado	Número de emisiones sonoras identificadas	Porcentaje máximo de emisiones correctamente clasificadas	Técnicas y métodos utilizados	Comentarios
Barbancho [Bar12]	Guitarra	330	87%	Cromagrama, HMM	En el análisis de Barbancho no se analiza la octava a la que pertenecen los acordes, pues hablando de la guitarra esto carece de importancia.
Böck [Böc11]	Piano	88	88.9%	transformada de Fourier, Redes Neuronales.	El sistema de Böck reconoce la activación de notas musicales, considerando una correcta identificación aún si se falla en identificar la octava.
Mauch [Mau10]	Piano	109	71%	Cromagrama, Redes Bayesianas Dinámicas (DBNs)	En este trabajo Mauch separa el espectro en dos subconjuntos representando cada uno las notas altas y bajas respectivamente. En su búsqueda limita la emisión de acordes solo al subconjunto de notas altas, limitando el mismo a unas dos octavas.
Papadopoulos [Pap07]	Piano	24	70.9%	Cromagrama, HMM.	El trabajo de Papadopoulos no especifica la octava a la que pertenecen los acordes analizados (12 acordes mayores y 12 acordes menores).

2.2 The Music Information Retrieval Evaluation eXchange (MIREX).

MIREX es una comunidad de evaluación de algoritmos de extracción de características y clasificación de señales de audio organizado por la Universidad de Illinois que año con año lanza una convocatoria para la resolución de ciertas tareas. Como ejemplo de éstas se citan a continuación algunas de ellas pertenecientes a la convocatoria 2013, mismas que se acompañan con la precisión más alta alcanzada ente los participantes.

- Estimación de acordes 2013. 76% [MIRa13].
- Detección de escala. 81.9% [MIRb13].
- Alineación automática de audio y partitura. 86.7% [MIRc13]

Como puede observarse aún las investigaciones más recientes no logran rebasar una precisión del 90%, lo que deja ver un campo de investigación en crecimiento.

3 Marco Teórico.

3.1 Sonido

El sonido consiste en una vibración mecánica que se propaga a través de un medio elástico y denso (habitualmente el aire) y que es capaz de producir una sensación auditiva. Cuando se habla de ondas sonoras, casi siempre se hace referencia a las ondas longitudinales cuya frecuencia fluctúa ente 20 y 20,000 Hz, es decir, el intervalo de audición humana [Res04]. Las ondas sonoras generan variaciones de presión en el medio en el que se propagan, esto puede ser representado a como se muestra en la Figura 3.1.

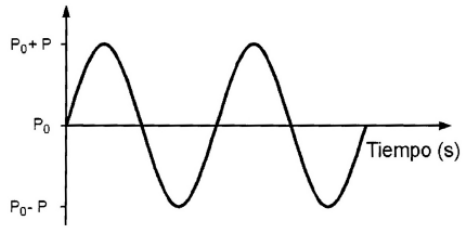


Figura 3.1 Evolución de la presión sonora en función del tiempo en un punto cualquiera del espacio.

3.2 Sensaciones sonoras

Las cualidades que se distinguen habitualmente en el sonido o mejor dicho, en las sensaciones sonoras, son tres: altura, intensidad y timbre [Ola54].

La altura de un sonido es la cualidad que se quiere expresar cuando se dice que un sonido es más agudo o más grave que otro; dependiendo principalmente de la frecuencia del movimiento vibratorio que lo origina, correspondiendo los sonidos agudos a frecuencias elevadas y los graves a las frecuencias bajas.

La intensidad del sonido es la cualidad que se quiere expresar cuando un sonido es más fuerte o más débil que otro; depende en primera aproximación de la amplitud del movimiento vibratorio que lo origina y en menor medida de la frecuencia de éste.

El timbre es la cualidad que permite diferenciar dos sonidos de igual altura e intensidad, pero de diversa procedencia; depende del grado de complejidad del movimiento vibratorio que origina el sonido.

3.3 Notas musicales

Actualmente en la música occidental se utiliza un sistema de doce notas con temperamento igual (esto quiere decir que la proporción existente entre una nota y la siguiente es constante) para representar la frecuencia de una onda sonora. En el sistema latino las doce notas musicales se conocen como Do, $Do\#\ (o\ bien\ \ Reb\ ), Re, Re\#\ (Mib), Mi, Fa,\ \ Fa\#\ (Solb), Sol, Sol\#\ (Lab), La, La\#\ (Sib), Si.$ Donde el símbolo # se lee como sostenido y el símbolo b como bemol. A las notas que aparecen seguidas de un sostenido o bemol se les llama notas $alteradas$ y con ello a estos símbolos se les designa el término de $alteraciones$. Como puede observarse las notas seguidas de alguna alteración pueden representarse de dos formas, ya sea con la nota anterior seguida de un sostenido o bien de la nota siguiente acompañada de un bemol, en el lenguaje musical a estos dos símbolos (que representan la misma nota) se les conoce como $enarmónicos$. Por otro lado en el sistema anglosajón se utilizan letras para referirse a estas mismas doce

notas, esto es, $C, C\#\ (Db), D, D\#\ (Eb), E, F, F\#\ (Gb), G, G\#\ (Ab), A, A\#\ (Bb), B$. En la Figura 3.2 se puede observar la ubicación de estas notas en el piano.

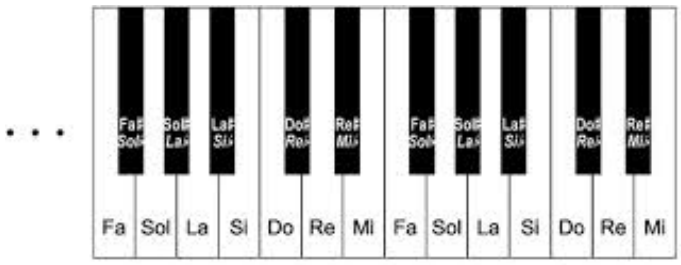


Figura 3.2 Ubicación de las doce notas del sistema de música occidental. Tomada de [Nuñ07].

3.4 Pentagrama

La representación por excelencia de una pieza musical, está dada por el $pentagrama$, éste consiste en un conjunto de cinco líneas horizontales usadas como base para la colocación de las notas musicales a ser interpretadas. La Figura 3.3 muestra un ejemplo de pentagrama indicando sus principales componentes.

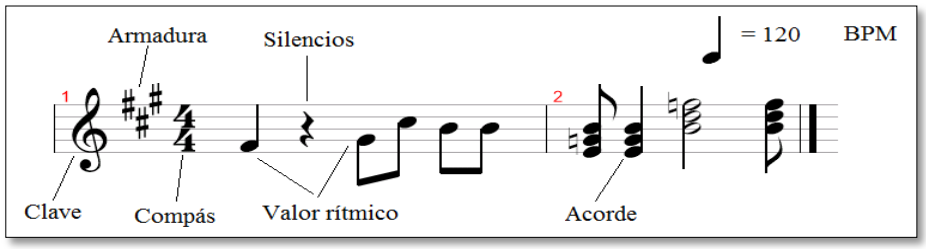


Figura 3.3 Elementos constitutivos de la representación escrita de una pieza musical.

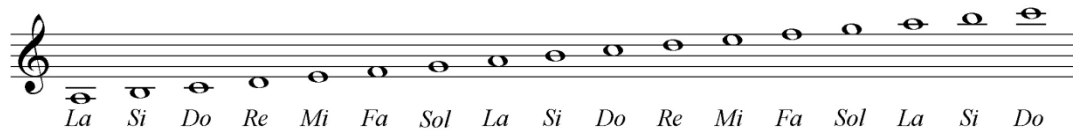
Para comprender el significado de cada uno de estos símbolos es necesario entender las bases de la teoría musical, misma que será presentada a lo largo del desarrollo de este marco teórico.

De momento puede introducirse el concepto de $clave$, símbolo con el que se comienza siempre la escritura en el pentagrama y que nos indica el rango de frecuencias sobre el que debe interpretarse una nota musical, en general se utilizan tres claves denominadas clave de Sol, clave de Fa y clave de Do. La Figura 3.4 muestra las mismas.

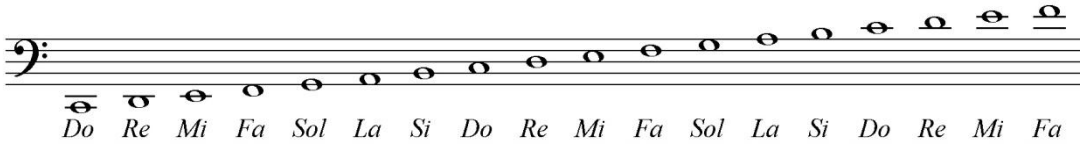
(a) (b) (c)

Figura 3.4 Claves habitualmente usadas en la representación de una pieza musical. (a) Clave de Sol. (b) Clave de Fa. (c) Clave de Do.

Las claves dan a las líneas del pentagrama el punto de referencia a partir del cual las notas serán ordenadas. La Figura 3.5 muestra el orden de las notas musicales en las claves de Sol y Fa.



(a)



(b)

Figura 3.5 Ubicación de las notas musicales en el pentagrama. (a) Orden en clave de Sol. (b) Orden en clave de Fa.

Como puede verse a veces las cinco líneas del pentagrama no son suficientes para representar una nota musical dada, es por ello que se agregan pequeñas líneas horizontales llamadas $auxiliares$, para continuar con la escritura. Así mismo el observador habrá notado que las notas alteradas no aparecen en la Figura anterior. Para representar una nota alterada se escribe la alteración deseada (sostenido o bemol) seguida de la nota a alterar. La Figura 3.6 ilustra la representación de un $Sol\#$ $(G\#)$ y un Lab (Ab).

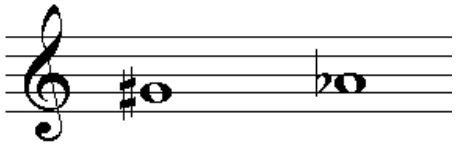


Figura 3.6 Representación de notas alteradas.

3.5 Frecuencia de las notas musicales

En la Figura 3.7 se representa la relación existente entre la frecuencia de una nota musical y el nombre correspondiente en el lenguaje musical, así como su representación en el pentagrama.

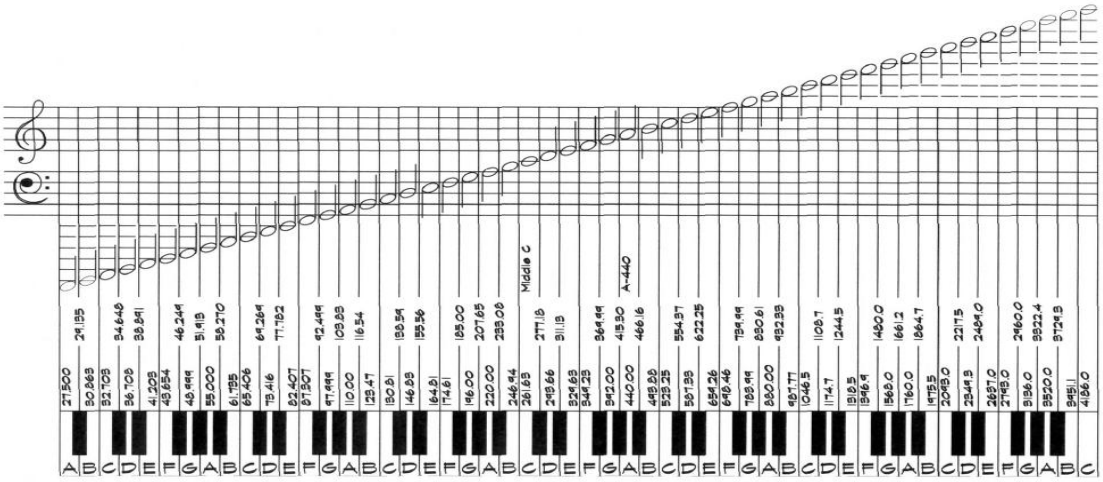


Figura 3.7 Notación utilizada en la música para hacer referencia a la frecuencia de las notas musicales.

Como puede observarse la clave de Sol es utilizada para representar notas de frecuencias "altas" y la clave de Fa cubre el rango de frecuencias "bajas".

En la Tabla 3.1 se presentan las frecuencias de las notas musicales separadas por octavas. Una octava se define como un rango de frecuencias dado por $[F_1, F_2]$ tales que $F_2 = 2F_1$

Tabla 3.1 Frecuencia de las notas musicales separadas por octava. Tomada de [Mon10].

	Oc. 0	Oc. 1	Oc. 2	Oc. 3	Oc. 4	Oc. 5	Oc. 6	Oc. 7	Oc. 8
Do		32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
Do#		34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	
Re		36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	
Re#		38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	
Mi		41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	
Fa		43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	
Fa#		46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	
Sol		49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	
Sol#		51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	
La	27,50	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	
La#	29,14	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	
Si	30,87	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	

Por otro lado, la norma ISO 266 estandariza las bandas de octava y sus divisiones en tercios de octava como se muestra en la Figura 3.8.

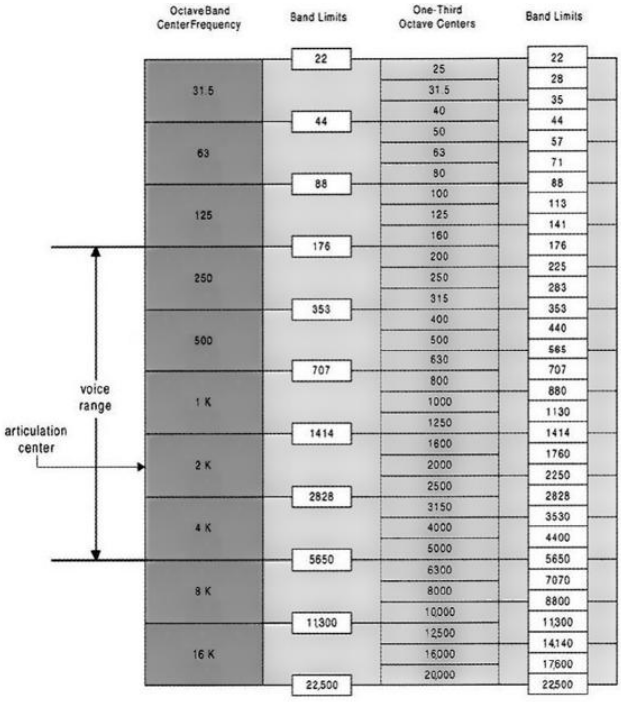


Figura 3.8 Bandas estandarizadas de octava y 1/3 de octava.

La definición de intervalos de octava obedece al amplio rango de audición del ser humano, así mismo, utilizando ésta se puede hacer referencia a una nota musical en una octava dada usando la notación latina o

anglosajona seguida del número de octava en la que se encuentra, por ejemplo, la nota La con una frecuencia de $440\ Hz$ se expresa como $A4$ o bien un Do sostenido a una frecuancia de $69.3\ Hz$ se expresa como $C\#2$.

Como se mencionó anteriormente la proporción entre una nota musical $N1$ y su consecutiva posterior $N2$ mantienen una proporción constante, esto es:

$$\frac{N2}{N1} = r \tag{3.1}$$

Para encontrar el valor de la constante r se puede tomar la nota $A4$ como referencia y notar que:

$$frecuencia(A\#4) = frecuencia(A4) \cdot r \tag{3.2}$$

Para la siguiente nota consecutiva, esta es $B4$, se tiene:

$$frecuencia(B4) = frecuencia(A\#4) \cdot r = fecuencia(A4) \cdot r^2 \tag{3.3}$$

Si se continúa de esta manera se encuentra que para $A5$ se cumple que:

$$frecuencia(A5) = frecuencia(A4) \cdot r^{12} \tag{3.4}$$

Por otro lado se sabe que $frecuencia(A4) = 440\ Hz$ y $frecuencia(A5) = 880\ Hz$ con lo que, sustituyendo en la ecuación (3.4), se obtiene que:

$$r = \sqrt[12]{2} \tag{3.5}$$

En general para dos notas musicales con frecuencias $F_1\ y\ F_2$ respectivamente, se tiene:

$$F_2 = 2^{\frac{d}{12}} \cdot F_1 \tag{3.6}$$

Donde d representa el número de notas que hay que recorrer de la primera nota a la segunda. Cabe destacar que el valor de d es entero, ya sea positivo si se va a notas de mayor frecuencia o negativo si se va a menores frecuencias.

3.6 Semitono y Tono

Una definición también fundamental en el lenguaje musical es la de *semitono*, que se utiliza para hacer referencia al recorrido que hay que hacer para ir de una nota a su consecutiva, por ejemplo, para ir de un $A4$ a un $A\#4$ se recorre un semitono. Al recorrido de dos semitonos se le conoce como $Tono$. Con esto para la ecuación (3.6), la variable d puede entenderse como el número de semitonos entre una nota y otra.

Es claro que en términos matemáticos el "tamaño" de un semitono varía dependiendo de la nota de la que se parte. Para encontrar la diferencia de frecuencias entre una nota F y sus adyacentes F_1 (anterior) y F_2 (posterior), con el fin de traducir un semitono a su "tamaño" en frecuencia, se tiene:

$$\Delta f_1 = F - F_1 \tag{3.7}$$

$$\Delta f_2 = F_2 - F \tag{3.8}$$

Haciendo uso de la ecuación (3.6) se obtiene que $F_1 = F\left(2^{\frac{-1}{12}}\right)$ y $F_2 = F\left(2^{\frac{1}{12}}\right)$ con lo que sustituyendo en las ecuaciones (3.7) y (3.8) respectivamente se concluye que:

$$\Delta f_1 = F\left(1 - 2^{\frac{-1}{12}}\right) \tag{3.9}$$

$$\Delta f_2 = F\left(2^{\frac{1}{12}} - 1\right) \tag{3.10}$$

De esta manera se tiene que las distancias entre la frecuencia F de una nota dada y su anterior y posterior son aproximadamente $\Delta f_1 \approx 0.0561F$ y $\Delta f_2 \approx 0.0595F$ respectivamente. La Figura 3.9 ilustra lo descrito antes.

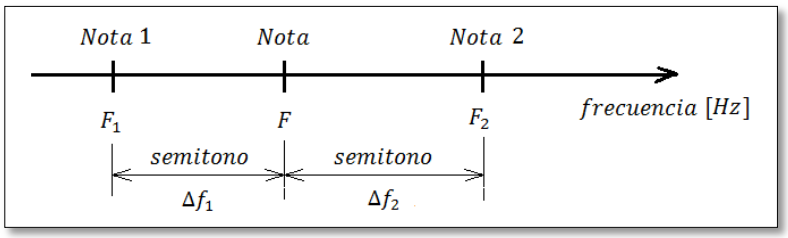


Figura 3.9 Diferencia de frecuencias entre una nota y su anterior y posterior.

Como puede observarse los músicos utilizan el término semitono para referirse al recorrido entre una nota cualquiera y su posterior. Desde un punto de vista matemático el "valor" del semitono es variable, dependiendo éste de la nota que se analice.

3.7 Duración de una nota musical

Como ya sabemos una pieza musical melódica toma a las notas como uno de sus elementos primarios para su construcción. Es evidente que para lograr una sensación agradable al oído es necesario organizar la emisión de estas notas. Es por ello que resulta fundamental definir una unidad básica para la medición del tiempo. En el caso de la música se utiliza el pulso o $beat$, en donde cada uno de estos pulsos se encuentra separado uno de otro en intervalos de tiempo iguales, este intervalo se encuentra ligado al $tempo$, que indica el número de $beats$ por minuto (bpm) y que en términos prácticos se refiere a la "velocidad" con la que se interpreta una pieza musical. Por ejemplo un tempo de $60\ bpm$ indica que cada minuto se dividirá en 60 pulsos, por lo que el tiempo entre cada pulso será de un segundo; así mismo un tempo de $120\ bpm$ indica una separación en segundos entre pulsos de $0.5\ s$.

La duración de una nota se indica por medio de la figura de la misma. La nota de mayor duración se llama $redonda$ representada por un óvalo. A partir de éste las siguientes figuras dividen al tiempo de la redonda en factores de dos. Comenzando con la $blanca$, figura parecida a la redonda pero a la que se le agrega una línea delgada horizontal, denominada $plica$ en uno de sus costado, generalmente el derecho apuntando hacia arriba o bien del lado izquierdo apuntando hacia abajo. La figura que tiene una duración igual a la mitad de la blanca (un cuarto de la redonda) se llama $negra$ o cuarto, que es similar a la blanca pero se rellena el óvalo. La figura con una duración igual a la mitad de la negra es la $corchea$ u octavo, parecida a la negra, pero a la que se agrega una pequeña tilde o corchete. La Figura 3.10 muestra la relación entre las figuras descritas anteriormente, agregando otras que siguen el mismo patrón.

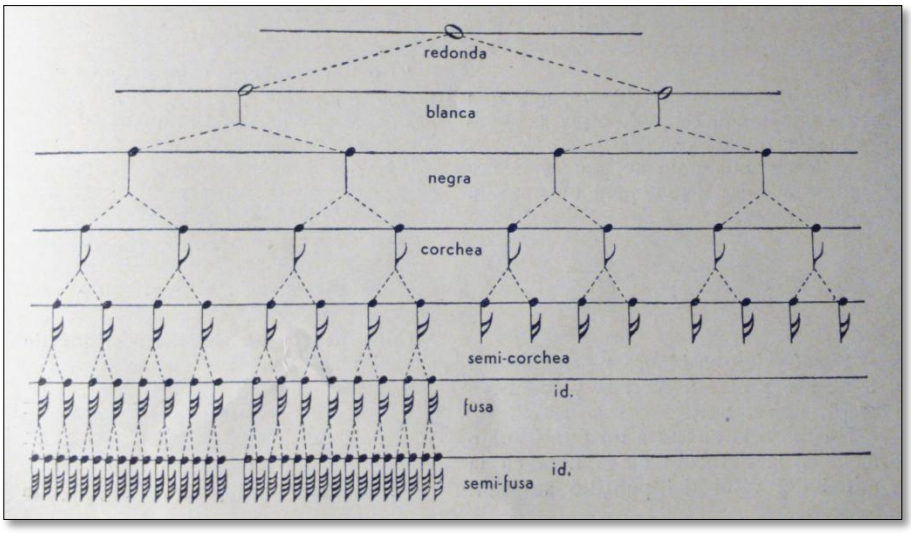


Figura 3.10 Figuras usadas para indicar la duración de una nota. Tomada de [Jac58].

Por convención cuando se tiene una sucesión de corcheas, éstas se agrupan uniendo sus corchetes en grupos equivalentes a una negra. La Figura 3.11 muestra lo descrito.

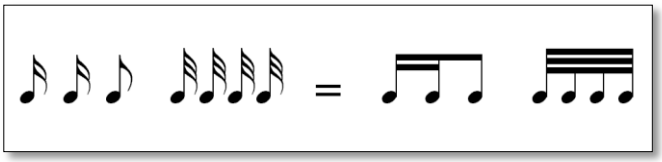


Figura 3.11 Agrupación de corcheas, semicorcheas y figuras de duración menor. El agrupamiento se suele dar de modo que la duración de cada grupo sea el de una negra.

3.8 Compás

En la sección anterior se mencionó que la figura redonda es la que presenta una mayor duración, pero no se especificó cuál era ésta. Para dar un valor específico en pulsos o beats a cualquier figura es necesario analizar el compás. Un compás consiste en la agrupación de pulsos en el que se acentúa el primer pulso de este grupo. Para especificar el número de pulsos por compás y la duración (en pulsos) de las figuras se utiliza una notación parecida a una fracción (sin la línea horizontal) en donde el numerador indica el número de pulsos por compás y el denominador indica el símbolo que se utilizará para representar la duración de un beat. Por ejemplo, el compás de cuatro cuartos $\frac{4}{4}$ indica que cada compás estará compuesto por cuatro pulsos y que la duración de un pulso será equivalente a un cuarto de redonda, esto es, una negra. De igual manera un compás de $\frac{3}{8}$ indica que se agrupan tres pulsos por compás y que la duración de un pulso será representada por un octavo de redonda, es decir, una corchea. La Figura 3.12 ilustra el uso del compás de cuatro cuartos.

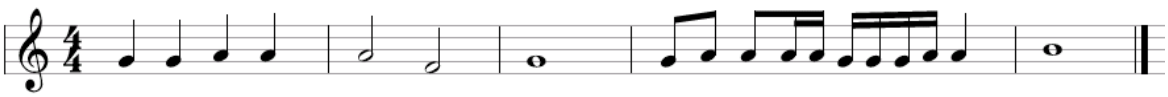


Figura 3.12 Compás de cuatro cuartos. El denominador de la fracción nos indica que se utilizarán cuartos para representar la duración de un pulso y el numerador nos indica que se agruparán cuatro pulsos por compás.

Como puede verse la fracción que nos da la información del compás se coloca enseguida de la clave o bien enseguida de la armadura, la cual se analizará más tarde. Así mismo cabe destacar que cada compás se delimita con una línea vertical que abarca las cinco líneas del pentagrama.

3.9 Intervalos

De la sección *Semitono y Tono* se concluye que la relación entre una nota y otra puede expresarse en semitonos y tonos, esta relación o distancia recibe el nombre de *intervalo*. En la música cada intervalo recibe un nombre, dependiendo del número de semitonos que lo compone. Por ejemplo, al intervalo formado por un semitono se le conoce como segunda menor, al formado por dos semitonos (o un tono) se le conoce como segunda mayor, al formado por tres semitonos se le conoce como tercera menor y así sucesivamente. La Figura 3.13 ilustra los trece primeros intervalos musicales.

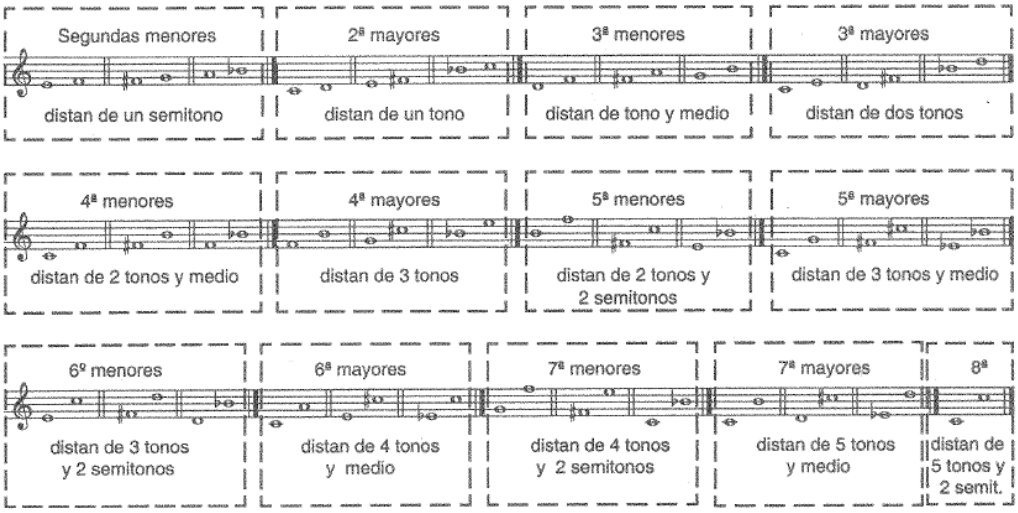


Figura 3.13 Intervalos musicales. Tomada de [Esl03].

En algunas bibliografías suelen sustituirse los términos quinta mayor y quinta menor por quinta justa y quinta disminuida respectivamente, algo similar ocurre con los intervalos de cuarta en los cuales la cuarta menor también se conoce como cuarta justa. De igual manera se designa el término *Unísono* cuando se permanece en la misma nota (esto puede entenderse como un recorrido de cero semitonos).

3.10 Escala musical

La escala musical o simplemente escala, puede entenderse como un conjunto de notas, de las cuales se elige una a la que se le llamará *raíz* y a partir de ella se fijan intervalos entre una nota y su posterior. La nota raíz es la que dará nombre a la escala.

Los intervalos a los que hace referencia la definición anterior, en el lenguaje musical, se expresan en semitonos y tonos.

La primera escala a definir es la *escala cromática* de *Do*, compuesta por las doce notas usadas en occidente, en donde la distancia entre una nota y su posterior es de un semitono, ver la Figura 3.14.

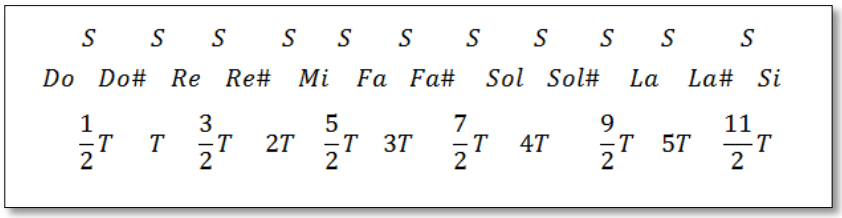


Figura 3.14 Escala cromática. Al centro se muestra las notas que componen la escala cromática de Do. En la parte superior los intervalos entre una nota y su posterior expresados en semitonos (S). En la parte inferior se muestran los intervalos entre la raíz de la escala y las demás notas expresados en tonos (T).

3.11 Escala Mayor

Quizá la escala más ampliamente usada en la música occidental es la escala mayor, compuesta por siete notas, en donde los intervalos entre una nota y su posterior están dados por la regla: T, T, S, T, T, T, S. En donde T representa un tono y S un semitono. La Figura 3.15 muestra la escala de Do mayor.

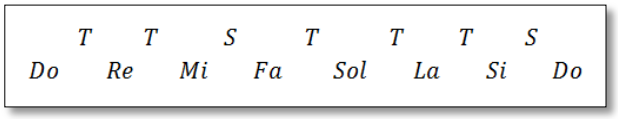


Figura 3.15 Escala de Do mayor.

Como puede observarse la escala de Do mayor no presenta ninguna alteración en sus notas. Como segundo ejemplo en la Figura 3.16 se muestra la escala de Mi mayor, la cual presenta cuatro alteraciones, éstas en las notas Fa, Sol, Do, Re.

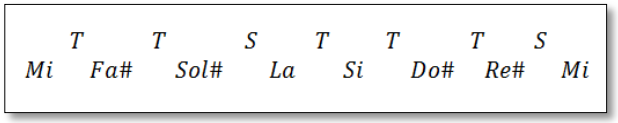


Figura 3.16 Escala de Mi mayor.

Una manera de identificar a las notas que componen una escala es a través del término $grado$, el cual hace referencia al lugar que ocupa una nota con respecto a la raíz, por ejemplo, en la escala de Do mayor, la nota Do es el primer grado, la nota Re es el segundo grado, Mi el tercer grado y así sucesivamente.

Haciendo uso de lo expuesto en la sección de $Intervalos$ se pueden clasificar las distancias en una escala mayor partiendo siempre de la raíz. Por ejemplo, el intervalo en la escala mayor de Do y su segundo grado (Re) es una segunda mayor, pues entre éstas notas se recorre un tono. Continuando con el mismo análisis se pueden obtener los nombres de los intervalos que componen a una escala mayor. La Tabla 3.2 muestra estos intervalos, usando la escala de Do mayor como guía.

Tabla 3.2 Intervalos presentes en la escala mayor. Las distancias para el análisis de intervalos se toman siempre a partir de la raíz.

Nota	Intervalo	Distancia [tonos]	Distancia [semitonos]
Do	Unísono	0	0
Re	Segunda mayor	1	2
Mi	Tercera mayor	2	4
Fa	Cuarta justa (menor)	2.5	5
Sol	Quinta justa (mayor)	3.5	7
La	Sexta mayor	4.5	9

Si	Séptima mayor	5.5	11
Do	Octava	6	12

3.12 Escala menor

Otra escala también ampliamente usada es la escala menor, en donde los intervalos necesarios para su construcción están dados por: T, S, T, T, S, T, T. La Figura 3.17 muestra la escala de La menor.

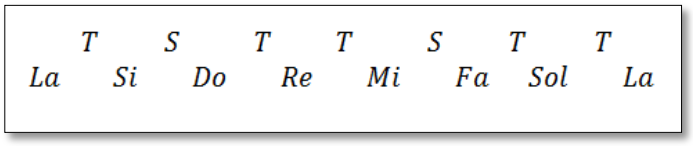


Figura 3.17 Escala de La menor.

Puede verse que la escala de La menor, como la escala de Do mayor, no presenta ninguna alteración. De hecho la escala mayor y menor guardan una estrecha relación entre sí. Si se toma el sexto grado de la escala mayor y se comienza a partir de ahí una escala utilizando las mismas notas se obtiene una escala menor. Por ejemplo, si se toman las notas que componen la escala de Mi mayor y se construye una escala con éstas comenzando a partir del sexto grado, este es $Do\#$, se obtiene la escala de $Do\#$ menor. A la escala menor que surge a partir del análisis del sexto grado de una escala mayor se le conoce como $relativa\ menor$. Por ejemplo, la escala de La menor es la relativa menor de la escala de Do mayor.

Por otro lado, siguiendo un análisis de intervalos similar al hecho con la escala mayor, se obtiene la Tabla 3.3, que muestra los intervalos presentes en la escala menor usando la escala de La menor como base.

Tabla 3.3 Intervalos presentes en la escala menor. Las distancias para el análisis de intervalos se toman siempre a partir de la raíz.

Nota	Intervalo	Distancia [tonos]	Distancia [semitonos]
La	Unísono	0	0
Si	Segunda mayor	1	2
Do	Tercera menor	3.5	3
Re	Cuarta justa (menor)	2.5	5
Mi	Quinta justa (mayor)	3.5	7
Fa	Sexta menor	4	8
Sol	Séptima menor	5	10
La	Octava	6	12

3.13 Armadura

Al analizar la construcción de escalas mayores y menores puede notarse que para cada escala mayor (y su relativa menor) aparece un número específico de alteraciones. En el caso de la escala de Do mayor (La menor) no aparece ninguna, para la escala de Sol mayor (Mi menor) aparece una alteración, esta es $Fa\#$. La escala mayor que presenta dos alteraciones es Re mayor (Si menor), la escala con tres alteraciones es La mayor ($Fa\#$ menor) y así sucesivamente. En otras palabras, si se conocen el número de alteraciones presentes en una escala, ya sea mayor o menor, puede saberse en qué escala está compuesta una pieza musical. Con esto los músicos definen la $armadura$, que consiste en el conjunto de alteraciones presentes en una escala mayor o menor, éstas se colocan enseguida de la clave en el pentagrama. La Figura 3.18 muestra las armaduras correspondientes a las escalas mayores y sus relativas menores.

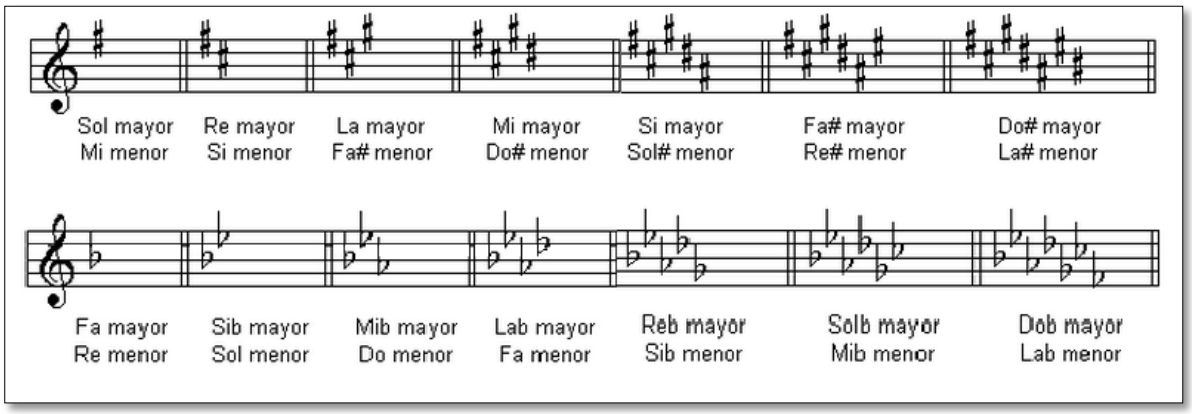


Figura 3.18 Armaduras de las escalas mayores y sus relativas menores para la clave de Sol. Tomada de [Rod14].

Como puede observarse cada alteración (sostenido o bemol) ocupa una posición específica, que se corresponde con la nota alterada. Para la clave de Fa también se utilizan armaduras, éstas en diferente posición dependiendo de las notas a alterar.

3.14 Melodía, Armonía y Ritmo

De acuerdo a Bernal Jiménez [Ber50], la melodía, armonía, ritmo y timbre son los elementos constitutivos de la música tal como se concibe hoy en día. Como ya se mencionó, el timbre es la cualidad del sonido que nos permite diferenciar un instrumento musical de otro.

En términos musicales, la $melodía$ se entiende como la sucesión de sonidos de diferentes alturas. En esta definición se sobreentiende que la emisión de cada uno de estos sonidos se hace de manera individual o bien que se ejecuta una nota a la vez. La Figura 3.19 muestra una melodía representada en el pentagrama.

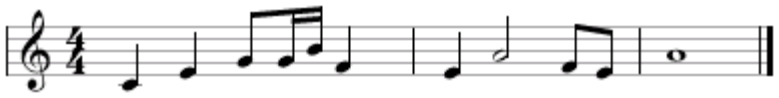


Figura 3.19 Ejemplo de melodía.

Se denomina $armonía$, en términos amplios, a la serie de producciones simultáneas de dos o más sonidos de diferente altura. Utilizando el concepto de acorde (que se analizará más adelante) la armonía puede entenderse como una sucesión de acordes y bicordes. La Figura 3.20 presenta una armonía representada en el pentagrama.

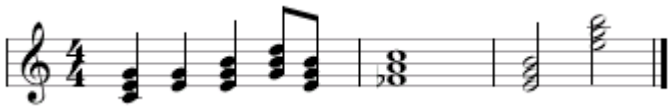


Figura 3.20 Ejemplo de armonía.

En un lenguaje más técnico suelen utilizarse los términos monofonía y polifonía para referirse a la melodía y armonía respectivamente.

El $ritmo$, por otra parte, puede entenderse como la forma en que se suceden una serie de sonidos que se repiten tomando como base un patrón regular de tiempo.

3.15 Vibrato

El vibrato es una técnica ampliamente usada en los instrumentos como la guitarra y el violín que consiste en hacer vibrar la cuerda sobre la cual se está emitiendo una nota. Esto puede lograrse ya sea haciendo vibrar la cuerda de arriba hacia abajo, rotando ligeramente la yema del dedo de derecha a izquierda o bien una combinación de ambos movimientos. De esta manera la frecuencia de la señal sonora generada varía alrededor de la frecuencia de la nota a la que se le aplica el vibrato dentro de un intervalo de por lo general dos semitonos por encima y dos semitonos por debajo.

3.16 Acorde

Se define el acorde como la emisión simultánea de tres a siete notas musicales a intervalos de tercera (menor o mayor). De acuerdo a esta definición la emisión simultánea de dos notas, estrictamente, no puede ser llamada acorde, en su lugar se denomina bicorde.

Existen un enorme número de combinaciones posibles para formar acordes pero por lo general se comienza con el análisis del acorde mayor y el acorde menor. El acorde mayor se forma por tres notas entre las cuales existen intervalos de tercera mayor (entre la primera y la segunda nota) y tercera menor (entre la segunda y la tercera nota), ejemplo de éste es el acorde de $Do\ Mayor\ (DoM)$, formado por las notas Do, Mi, Sol. La Figura 3.21 muestra lo descrito.

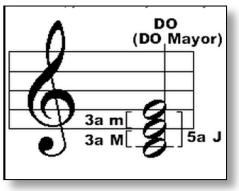


Figura 3.21 Acorde de Do Mayor en estado original. Tomada de [Cre10].

A la nota más grave del acorde, en este caso Do, se le conoce como $tónica$ y a las otras dos se les llama $tercera$ y $quinta$, en este caso Mi y Sol respectivamente. El acorde anterior se presenta en su llamado $estado\ original$, pues la nota que da nombre al acorde es la más grave. Cuando la tercera o quinta nota del acorde son las más graves se dice que el acorde presenta una $inversión$ llamadas primera y segunda respectivamente. La Figura 3.22 ilustra lo descrito.

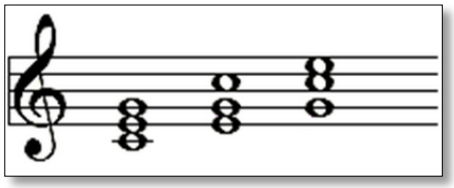


Figura 3.22 Acorde de Do Mayor. Primer acorde: Estado original. Segundo acorde: Primera inversión. Tercer acorde: Segunda inversión.

Como puede verse un acorde posee un número de inversiones igual al número de notas que lo componen menos uno.

Por su parte el acorde menor se forma por tres notas que guardan intervalos de tercera menor (entre la primera y segunda nota) y tercera mayor (entre la segunda y tercera nota), ejemplo de éste es el acorde de $La\ menor\ (Lam)$ compuesto por las notas La, Do, Mi.

3.17 MIDI

MIDI, abreviación de Interface Digital de Instrumento Musical ($Music\ Instrumet\ Digital\ Interface$), es un sistema que permite a instrumentos musicales electrónicos y computadoras enviar instrucciones o códigos unos a otros [Mid09]. Al ser MIDI todo un sistema de comunicación entre dispositivos electrónicos se compone de dos partes, hardware y software. En el caso particular de este trabajo será suficiente con analizar el software o formato MIDI, esto es, archivos con extensión $.mid$.

Cabe recalcar que los archivos MIDI contienen una serie de instrucciones que permiten la comunicación entre dispositivos. Así mismo es importante puntualizar que MIDI no es un formato de audio o compresión de audio que contenga la información de una señal acústica como un archivo wav o $mp3$. Si bien existen diversos reproductores de audio que permiten escuchar archivos $.mid$ como si se tratasen de audio, lo cierto es que éstos decodifican las instrucciones contenidas en el archivo MIDI y sintetizan señales sonoras para su reproducción.

Para analizar la comunicación entre instrumentos y la manera en que un archivo $.mid$ almacena esta información se puede comenzar examinando la manera en que se produce un sonido con un instrumento. Tomando el piano como ejemplo se ve que para emitir un sonido es necesario oprimir una tecla y para detener la emisión es necesario soltarla, con esto, se llega a dos instrucciones fundamentales, éstas son, comenzar la reproducción de una nota y detener la reproducción de la misma. Podemos también considerar la fuerza con la que se presiona una nota dándonos sensaciones de "ataques" veloces o lentos, con lo que al par de instrucciones mencionadas puede agregárseles este atributo a modo de indicar la velocidad con la que se emite un sonido.

Las dos instrucciones mencionadas son parte del repertorio del formato MIDI, en donde cada acontecimiento involucrado en la emisión de un sonido se denomina $evento$.

Un archivo MIDI se compone de "pedazos" (por la traducción literal del inglés $chunks$) o fragmentos, constituidos cada uno de estos por una serie de instrucciones. Estos fragmentos son el $Header\ chunk$ (fragmento de cabecera) y uno o más $Track\ chunks$ (fragmentos de pista). Para una representación más manejable se utilizan números hexadecimales para su escritura.

3.17.1 Header chunk

Un archivo MIDI siempre comienza con la cadena de caracteres "MThd" o $4D\ 54\ 68\ 64$ en hexadecimal, seguido de cuatro bytes representando el valor del seis decimal, esto es $00\ 00\ 00\ 06$, que representan la extensión del resto de la cabecera en bytes, misma que es constante para cualquier archivo $.mid$. Los dos bytes siguientes pueden contener los valores decimales cero, uno o dos, que indican el tipo de archivo MIDI utilizado. En términos generales el tipo cero de archivo MIDI almacena todas las instrucciones en un único $track$ o pista, mientras que los tipos uno y dos permiten el uso de diversas pistas. Los siguientes dos bytes hacen referencia al número de pistas contenidas en el archivo que, para el caso de un archivo tipo cero se limita a una. Finalmente los últimos dos bytes de la cabecera representan la velocidad con que los eventos

MIDI se sucederán, ésta puede estar dada de dos maneras, que se explicarán más adelante. La Figura 3.23 ilustra lo descrito anteriormente.

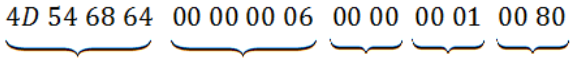


Figura 3.23 Formato para la cabecera de un archivo MIDI. (a) Valores hexadecimales para la cadena de caracteres "MThd". (b) Longitud del resto de la cabecera. (c) Tipo de archivo MIDI. (d) Número de pistas. (e) Velocidad de los acontecimientos.

Como se mencionó anteriormente la manera de indicar la velocidad de los eventos MIDI puede expresarse de dos maneras, éstas se diferencian entre sí por medio del bit que ocupa la posición quince (de derecha a izquierda y comenzando por cero) de (e) en la Figura 3.23.

Cuando el bit quince tiene el valor de uno, los catorce restantes se analizan a través del formato dado por el SMPTE ($Society\ of\ Motion\ Picture\ and\ Television\ Engineers$). Para el caso en el que el decimoquinto bit sea cero, caso utilizado en el presente trabajo, la velocidad se analiza usando divisiones o pulsos por cuarto (o figura negra en una partitura), que se representará por las siglas PPQ.

Para clarificar lo descrito en el párrafo anterior y encontrar una relación entre estos pulsos por cuarto (PPQ) y el tiempo, es necesario introducir una variable definida también en el formato MIDI, ésta es, microsegundos por cuarto, que se representará con las siglas $MSPQ$. Para el caso de un compás de cuatro cuartos se sabe que la duración de un cuarto o negra es igual a la duración de un $beat$, con esto, puede establecerse una relación entre el tempo de una canción y los $MSPQ$, ésta es:

$$tempo\ [bpm] = \frac{6 \times 10^7}{MSPQ\ \left[\frac{\mu s}{beat}\right]} \tag{3.11}$$

En la ecuación anterior, entre corchetes cuadrados, se adjuntan las unidades de cada variable para ayudar con el análisis dimensional. Las unidades $\frac{\mu s}{beat}$ representan microsegundos por $beat$ y, de la sección $Duración\ de\ una\ nota\ musical$, se sabe que bmp representa $beats$ por minuto (que pueden representarse en una notación más afín a las ciencias como $\frac{beat}{m}$). Con esto resulta evidente que éstas dos cantidades son inversamente proporcionales obteniéndose una constante de proporcionalidad de 6×10^7.

Para encontrar la duración en segundos de un pulso por cuarto, a la que se representará como TPP, como función de $MSPQ$ es necesario hacer uso de la variable PPQ de acuerdo a la ecuación:

$$TPP\ \left[\frac{\mu s}{pulso}\right] = \frac{MSPQ\ \left[\frac{\mu s}{beat}\right]}{PPQ\ \left[\frac{pulso}{beat}\right]} \tag{3.12}$$

Por default el valor asignado a $MSPQ$, en el formato MIDI, es de 5×10^5, pero puede ser modificado a través de un evento. Con el valor predeterminado de $MSPQ$ y a través de la ecuación (3.10), es claro que el tempo por default, para una canción con compases de cuatro cuartos, es de $120\ bpm$.

Para ejemplificar el uso de las ecuaciones anteriores, supongamos que se desea codificar una canción dada en compases de cuatro cuartos con un tempo de $60\ bpm$. Haciendo uso de (3.10) se concluye que es necesario establecer $MSPQ = 1 \times 10^6$, o bien, expresado en hexadecimal $MSPQ = 0F4240$.

Por otro lado, tomando las ecuaciones (3.10) y (3.11) se obtiene una tercera igualmente útil, ésta es:

$$PPS\left[\frac{pulso}{s}\right] = \frac{1}{60}\left[\frac{m}{s}\right] \cdot tempo\left[\frac{beat}{m}\right] \cdot PPQ\left[\frac{pulso}{beat}\right] \tag{3.13}$$

Donde PPS representa los pulsos (MIDI) por segundo. De nuevo se acompaña cada variable de sus respectivas unidades para dejar en claro la consistencia de unidades. Finalmente para conocer el número de pulsos P presentes en un intervalo de tiempo t dado en segundos se hace uso de:

$$P\,[pulso] = t[s] \cdot PPS\left[\frac{pulso}{s}\right] \tag{3.14}$$

Con esto podemos concluir que se tiene completo control en la velocidad en la que se interpretará un archivo MIDI haciendo modificaciones en las variables PPQ y $MSPQ$.

3.17.2 Track chunk

En este fragmento de los archivos MIDI se encuentran las instrucciones o eventos que permiten las emisiones de sonido, cambios de timbre, tiempo, $MSPQ$, etc.

Este fragmento comienza siempre con una cabecera compuesta de la cadena de caracteres "MTrk" o bien $4D\ 54\ 72\ 6B$ en hexadecimal, seguidos de cuatro bytes que indican la longitud del track. Después de la cabecera se encuentran todas las instrucciones MIDI necesarias para la generación de eventos. Para finalizar el fragmento de track se agregan los valores hexadecimales $00\ FF\ 2F\ 00$, que indican la salida del mismo. La Figura 3.24 muestra lo descrito.

$$\underbrace{4D\ 54\ 72\ 6B}_{(a)}\ \underbrace{XX\ XX\ XX\ XX}_{(b)}\ \underbrace{YY\ \cdots\ YY}_{(c)}\ \underbrace{00\ FF\ 2F\ 00}_{(d)}$$

Figura 3.24 Composición del Track chunk. (a) Cadena de caracteres "MTkr". (b) Longitud en bytes del resto del fragmento. (c) Instrucciones MIDI. (d) Instrucción de fin de track.

Las instrucciones MIDI también cumplen una sintaxis específica, ésta es, el número de pulsos o $delta\ time$ a esperar para activar un evento, seguido del evento a activar. Cabe destacar que este delta time se mide a partir de la activación de la instrucción anterior. Por ejemplo, para ejecutar dos eventos simultáneamente sus delta time deben ser, en la primera instrucción, el tiempo deseado a esperar para la activación y, en la segunda, el delta time deberá ser cero. La Figura 3.25 muestra una representación de lo descrito.

$$< delta\ time >< evento >< delta\ time >< evento > \cdots < delta\ time >< evento >$$
(a)

$$< delta1 >< evento1 >< delta2 >< evento2 >< 00 >< evento3 >$$
(b)

Figura 3.25 Sintaxis para las instrucciones MIDI. (a) Cadena de instrucciones MIDI. (b) Ejecución de dos eventos simultáneamente, en este caso los eventos evento2 y evento3, ambos se activarán un número de pulsos igual a delta2 después de la ejecución del evento evento1

Es importante señalar que la longitud de cada delta time es variable y puede ir de uno a cuatro bytes. Para conocer la longitud de cada delta time es necesario analizar el bit siete (contándose de derecha a izquierda y comenzando desde cero) de cada byte, si éste es cero se indica que es el primer byte y que finaliza el delta

time (los bytes se contarán de derecha a izquierda), en caso contrario, es decir, un valor de uno, se indica que el byte a la derecha es parte del delta time. A este tipo de representación se le conoce como $variable\ length\ value$ (VLV) o valor de longitud variable por su traducción al español. A continuación se muestran algunos ejemplos.

Se comenzará con un delta time de diez base diez, que se representará en distintas bases, esto es:

$$10_{10} = A_{16} = 0000\ 1010_2$$

Como puede verse el diez decimal puede representarse perfectamente con un byte. El bit siete es cero lo que indica que este byte es el último del delta time. Para fines prácticos se adjuntarán como subíndice las iniciales VLV para reconocer la diferencia entre un número expresado en decimal y otro expresado en longitud variable, esto nos permite escribir:

$$10_{10} = A_{16} = 0000\ 1010_2 = 0000\ 1010_{VLV}$$

Es importante señalar que la notación usada para expresar cantidades en VLV no llega más allá de este trabajo.

Como segundo ejemplo se tomará al 128 decimal, que expresado en distintas bases es:

$$128_{10} = 80_{16} = 1000\ 000_2$$

Si se desea expresar este número como VLV es necesario hacer uso de dos bytes, pues siempre el bit en la posición siete de cada byte nos indica el fin (o continuación) de la cantidad. Con esto puede verse que el único uno en la representación decimal ocupa este lugar, por lo que debe recorrerse un bit a la izquierda, esto nos deja con $1\ 0000\ 0000$. Ahora se tiene más de un byte. Para indicar la cantidad en VLV es necesario colocar un uno en el bit siete del último byte, para indicar que el byte a su derecha es parte del delta time, esto nos lleva a:

$$128_{10} = 80_{16} = 1000\ 000_2 = 1000\ 0001\ 0000\ 0000_{VLV} = (81\ 00_{16})_{VLV}$$

Ahora supóngase que se tiene el número $FF7F$ en VLV (expresado obviamente con caracteres hexadecimales), el cual, en la notación aquí utilizada es $1111\ 1111\ 0111\ 1111_{VLV}$. El bit siete del primer byte en el número dado es cero, es decir es el primer byte. Continuando con el análisis se observa un uno en la posición siete del segundo byte, lo que nos dice que es parte del delta time. Para expresar la cantidad dada en decimal se puede comenzar sustituyendo el bit en la posición quince por un cero, dándonos $0111\ 1111\ 0111\ 1111$, luego se elimina el bit en la séptima posición (recorriéndose todos los bits a su izquierda), esto es, $0011\ 1111\ 1111\ 1111$, la cual es la representación decimal, así;

$$1111\ 1111\ 0111\ 1111_{VLV} = 0011\ 1111\ 1111\ 1111_2 = 3FFF_{16} = 16383_{10}$$

Una vez entendida la manera en que se expresan los delta time en el formato MIDI se prosigue describiendo la sintaxis para los eventos. Para fines de este trabajo se describirán dos instrucciones, que son las necesarias para el desarrollo del mismo. Pueden consultarse [Has13], [Mid09a] y [Mid09b] para una lista más amplia y detallada.

Para la activación de una nota se utiliza el evento $Note\ on$, el cual se codifica, en binario, como se muestra en la Figura 3.26.

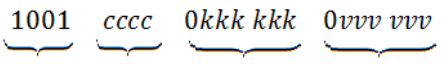


Figura 3.26 Codificación para la activación de una nota. (a) Código de Note on. (b) Canal. (c) Nota a activar usando su código MIDI. (d) Velocidad de activación.

Donde c, k y v representan dígitos binarios. El conjunto de cuatro bits compuesto por letras c, se especifica el canal en el que quiere activarse la nota (cada track cuenta con 16 canales). Los dígitos k representan el código MIDI de la nota a activar y las v establecen la velocidad con la que se emitirá la misma.

Para desactivar una nota o evento $Note\ off$ se utiliza una instrucción similar a la anterior dada como en la Figura 3.27.

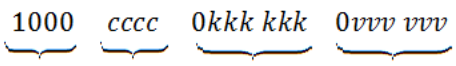


Figura 3.27 Codificación para la desactivación de una nota. (a) Código de Note off. (b) Canal. (c) Nota a desactivar usando su código MIDI. (d) Velocidad de desactivación.

En la Tabla 3.4 se muestran los valores MIDI para las nota musicales organizados por octavas.

Tabla 3.4 Códigos MIDI de la notas musicales expresados en decimal.

	Nota musical											
Octava	*Do*	*Do#*	*Re*	*Re#*	*Mi*	*Fa*	*Fa#*	*Sol*	*Sol#*	*La*	*La#*	*Si*
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127	-	-	-	-

Con esto puede ejemplificarse la emisión de una nota $Do4$ en formato MIDI por un periodo de 128_{10} pulsos, en hexadecimal, como se muestra en la Figura 3.28.

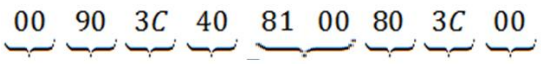


Figura 3.28 Emisión de una nota Do4 por un periodo de 128 pulsos. (a) Delta time de activación. (b) Evento Note on. (c) Código MIDI de Do4. (d) Velocidad de activación. (e) Delta time de desactivación, 128_{10} pulsos. (f) Evento Note off. (g) Código MIDI de Do4. (h) Velocidad de desactivación.

Para asemejarse más a la producción de música hecha por instrumentos acústicos y eléctricos el formato MIDI implementa los llamados *mensajes de cambio de control* (control change messages) que permiten, entre otras cosas, la emisión de notas moduladas, generando un efecto de vibrato. Este mensaje sigue una sintaxis similar a la de cualquier evento MIDI, como se muestra en la Figura 3.29.

$$\underbrace{00}_{(a)}\ \underbrace{B0}_{(b)}\ \underbrace{01}_{(c)}\ \underbrace{50}_{(d)}$$

Figura 3.29 Mensaje de cambio de control para la modulación de una nota. (a) Delta time de activación. (b) Instrucción de cambio de control. (c) Instrucción de modulación (d) Valor de la modulación.

En el caso de esta instrucción el valor de la modulación puede variar entre cero y 127_{10}.

3.18 Técnicas y métodos

En este apartado se dará una breve introducción a las técnicas, métodos y modelos utilizados en el presente trabajo.

3.18.1 Señales

Una señal se define como toda cantidad física que varía con respecto al tiempo, espacio o alguna otra variable o variables. Si una señal puede ser descrita por medio de una tabla o expresión matemática o una regla bien definida, se le llama determinista, por otro lado si se tiene una señal que evoluciona de manera impredecible se dice que ésta es aleatoria. Cabe destacar que la clasificación de las señales encontradas en el mundo real como deterministas o aleatorias no es siempre clara, pues algunos enfoques son capaces de arrojar resultados útiles, mientras que en otras ocasiones una mala clasificación puede llevar a resultados erróneos, pues algunas herramientas matemáticas pueden aplicarse únicamente a señales deterministas, mientras otras solo son aplicables a señales aleatorias [Pro07].

Otro concepto importante es el de señal periódica que es toda aquella señal s que cumple la igualdad $s(x) = s(x + T)$ donde se denomina a T como periodo de la señal. Una señal perfectamente periódica puede descomponerse en las llamadas series de Fourier como suma de funciones seno y coseno cuyas frecuencias son múltiplos enteros de alguna frecuencia base, llamada frecuencia fundamental del sonido.

Muchas señales son no periódicas, pero pueden sin embargo ser representadas como suma de sinusoides que no son armónicos. El término general para los sinusoides que compones una forma de onda, sean armónicos o no, es tono parcial o simplemente parcial.

Señales de audio como las producidas por instrumentos de cuerda producen señales quasi-periódicas, esto es, señales que no son periódicas pero que se aproximan [Moo75].

En el caso de los instrumentos musicales se puede hacer una clasificación general entre aquellos que producen señales con parciales casi armónicos y aquellos que no. Las señales provenientes de instrumentos casi armónicos pueden modelarse como sumas de sinusoides con variaciones lentas en frecuencia y amplitud, éste es el caso de instrumentos como el piano

3.18.2 Señales de audio producidas por el piano

En esta sección se mencionarán algunas características esenciales que poseen las señales de audio emitidas por un piano.

Para comenzar es necesario presentar el concepto de tono puro, que es aquella emisión sonora que se compone de una única frecuencia, ejemplo de este concepto son los sonidos generados por los diapasones, que sirven, entre otros propósitos como afinadores. Es claro que al escuchar un diapasón diseñado para emitir, digamos, una nota $La4$, podrá distinguirse de la misma nota emitida por un piano o una flauta, esto es debido al contenido armónico que estos instrumentos añaden a las notas que emiten. En el caso particular del piano se tiene que las ondas sonoras generadas por éste se componen de una onda sinusoidal con la frecuencia de la nota emitida f_0 más una serie de armónicos $f_n = nf_0$ (con $n = 2,3,\dots,N$). Cabe señalar que el piano también emite frecuencias cuasi-armónicas que se aproximan a un múltiplo de la frecuencia de la nota emitida a partir del decimoctavo armónico.

Continuando con el análisis de las notas en el piano, se observa que en general una nota comienza con un ruido tipo crujido para posteriormente presentar una oscilación debida a la cuerda percutida, seguida de una señal cuasi-periódica compuesta de la frecuencia de la nota emitida más frecuencias armónicas, cuasi-armónicas y ruido. La Figura 3.30 muestra lo descrito.

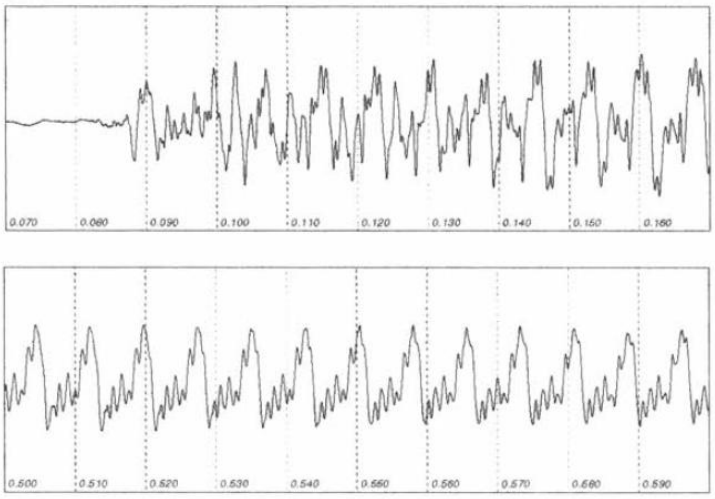


Figura 3.30 Señal de la nota Do3 emitida por un piano. En la parte superior se observa el ruido emitido al comenzar la emisión de la nota. En la parte inferior se observa la naturaleza cuasi-periódica de la señal pasados unos 20 ms de comenzar la emisión. Tomada de [Haw93]

En la Figura 3.31 puede observarse el espectro obtenido al analizar la nota $Do4$ de frecuencia igual a $262\ Hz$ y sus primeros 18 armónicos, luego de los cuales comienzan a aparecer parciales cuasi-armónicas.

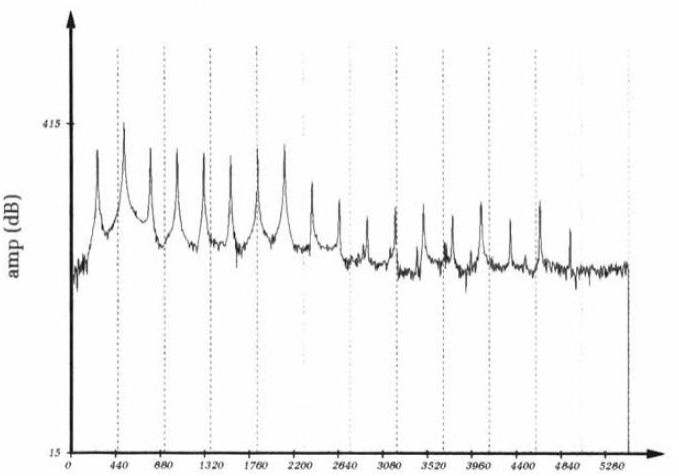


Figura 3.31 Espectro de la nota Do4. Tomada de [Haw93].

3.18.3 Función de autocorrelación

La función de autocorrelación (ACF por sus siglas en inglés) [Rab77] es una de los estimadores más utilizados para $F0$ definida como:

$$ACF[v] = \frac{1}{N} \sum_{n=0}^{N-v-1} x[n] \cdot x[n+v] \tag{3.15}$$

Donde $x[n]$ es la señal de entrada, N es la longitud de la señal y v denota el desplazamiento temporal. Para una señal periódica, el primer pico mayor en la ACF indica el periodo fundamental de la onda. Sin embargo cabe destacar que otros picos aparecen en múltiplos del período (conocidos como errores subarmónicos).

3.18.4 Espectro y transformada de Fourier

El espectro es una relación típicamente representada como la gráfica de una magnitud o valor relativo de algún parámetro contra la frecuencia.

Señales que típicamente son representadas en el dominio del tiempo y por cada función del tiempo $f(t)$, una función equivalente $F(\omega)$ puede ser encontrada de modo que específicamente describe el contenido en frecuencia requerido para generar $f(t)$. El estudio de la relación entre el dominio del tiempo y su correspondiente frecuencia es materia del análisis de Fourier y la transformada de Fourier, dada por la Ec. (3.16)

$$F[x(t)] = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt = X(\omega) \tag{3.16}$$

La transformada inversa de Fourier de la función $X(\omega)$ que lleva del domino de la frecuencia al tiempo está dada por:

$$F^{-1}[X(\omega)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{-j\omega t}d\omega = x(t) \tag{3.17}$$

Las ecuaciones (3.16) y (3.17) son válidas para funciones periódicas. En el caso de funciones aleatorias éstas pierden validez y por ello, con el objetivo de caracterizar el contenido en frecuencia de estas funciones se define la densidad espectral de potencia $S(f)$, misma que puede ser estimada por medio de la ecuación (3.18).

$$S(f) = \int_{-\infty}^{\infty} R_{xx}(\tau)e^{-j\omega t}d\tau \tag{3.18}$$

Siendo $R_{xx}(\cdot)$ la función de autocorrelación de la función de muestreo de un proceso estocástico $x(t)$.

A modo de poder realizar el análisis de Fourier a segmentos finitos de secuencias en tiempo discreto se formulan la transformada discreta de Fourier (DFT) y su inversa (IDFT) de acuerdo a las ecuaciones (3.19) y (3.20) respectivamente.

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j\frac{2\pi}{N}kn} \tag{3.19}$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j\frac{2\pi}{N}kn} \tag{3.20}$$

3.18.5 Ventaneo

Al procesar una señal digital $x[n]$ es claro que no es posible tomar un número infinito de muestras para el análisis de ésta, lo que hace necesario tomar solo un conjunto finito. El proceso de tomar un número discreto de muestras de la señal $x[n]$ es equivalente a multiplicar ésta por una función de longitud finita $w[n]$ conocida como ventana. Un ejemple de esta función es la llamada ventana rectangular definida como:

$$w[n] = \begin{cases} 1, & 0 \leq n < L \\ 0, & otro\ caso \end{cases}$$

De este modo puede obtenerse la señal ventaneada $\hat{x}[n]$ por medio de la igualdad $\hat{x}[n] = x[n]w[n]$.

La operación de ventanear en el domino del tiempo una señal $x_c(t)$ introduce dos tipos de distorsiones en el dominio de la frecuencia, por ejemplo, sea $x_c(t)$ la señal sinusoidal de la Figura 3.32(a) se observa que su transformada de Fourier $X_c(j\Omega)$ presenta dos impulsos localizados en las frecuencias $\Omega = \pm\Omega_1$. En la Figura 3.32(b) se observa una función ventana rectangular $w_c(t)$ y su correspondiente transformada de Fourier $W_c(j\Omega)$. Como puede observarse $W_c(j\Omega)$ presenta un pico pronunciado, denominado lóbulo principal, en el origen, seguido de una serie de picos de magnitud decreciente. En la Figura 3.32(c) como cada línea del espectro original es reemplazada por una copia del espectro de la ventana escalado por la amplitud de las correspondientes exponenciales complejas. La suma de las copias escaladas y trasladadas lleva a la transformada de Fourier de la señal sinusoidal ventaneada $\hat{X}_c(j\Omega)$, misma que se muestra en la Figura 3.32(d).

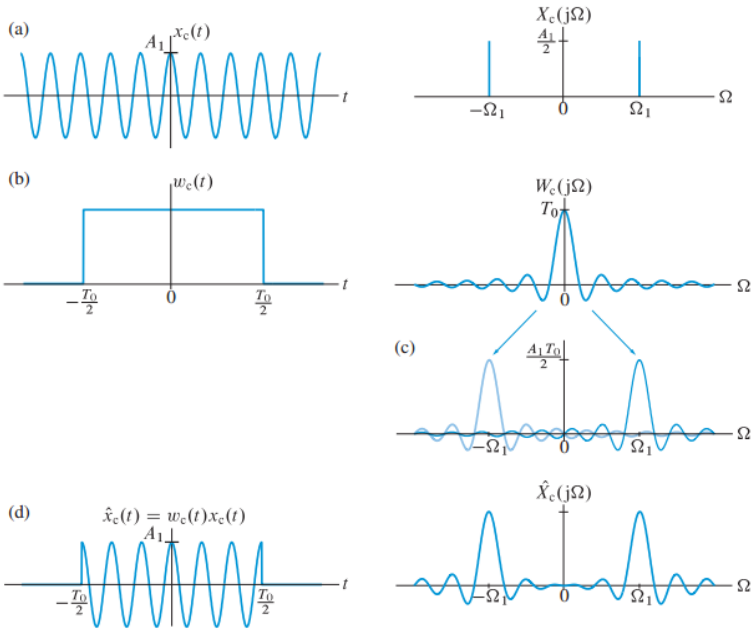


Figura 3.32 Efecto de ventaneo en el espectro de una señal sinusoidal. Tomada de [Man11].

De lo anterior puede verse que el ventanear una señal produce distorsiones en la transformada de Fourier de la señal dada, estas distorsiones pueden clasificarse en dos, *smearing* y *leakage*. El *smearing* hace referencia al efecto de "engrosamiento" que provoca el lóbulo principal a las líneas verticales que tendría el espectro ideal. Por otro lado el *leakage* hace referencia a la transferencia de potencia de bandas de frecuencia con una cantidad grande de potencia de la señal a bandas con poca o nula potencia, lo que se ve traducido en la aparición de picos en frecuencias erróneas, eliminación de picos o cambio de amplitud de picos existentes.

La Figura 3.33 muestra un conjunto de ventanas comúnmente usadas en la práctica y sus correspondientes espectros en escala lineal-lineal, lineal-logarítmica y logarítmica-logarítmica.

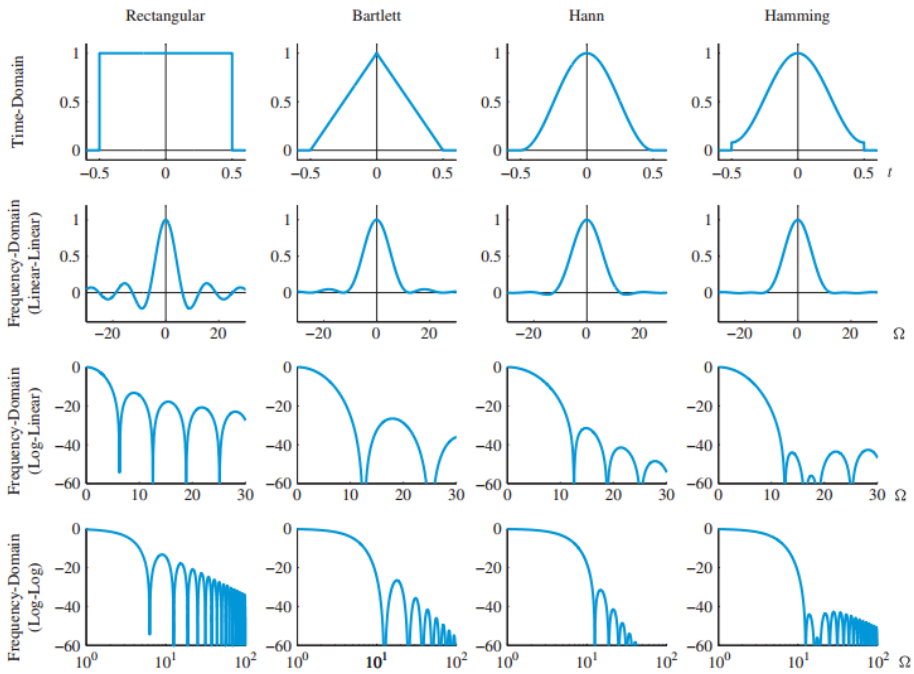


Figura 3.33 Características de las funciones ventana en el dominio del tiempo y el dominio de la frecuencia. Tomada de [Man11].

3.18.6 *Mel Frequency Cepstral Coefficients*

Los *Mel Frequency Cepstral Coefficients* (*MFCCs*) son uno de los vectores de características más utilizados en el reconocimiento de voz [Log00], esto debido a su habilidad de representar la amplitud del espectro en una forma compacta. El proceso de obtención de los *MFCCs* sigue un criterio basado en la manera en que el oído humano percibe los sonidos, esto es, en escala aproximadamente logarítmica.

Para la obtención de los *MFCCs* de una señal de audio, se comienza segmentando la misma en intervalos (generalmente de 20 *ms* a 40 *ms*) y aplicando una función ventana para eliminar el efecto de bordes. Después de esto se computa la transformada discreta de Fourier de cada uno de los segmentos obtenidos de las cuales se obtiene su correspondiente espectro. Una vez que se tiene este vector se procede a reducir la dimensión del mismo, tomando diferentes componentes y llevándolas a una sola a través de un banco de filtros. La Figura 3.34 muestra un banco de filtro típicamente usado para este propósito.

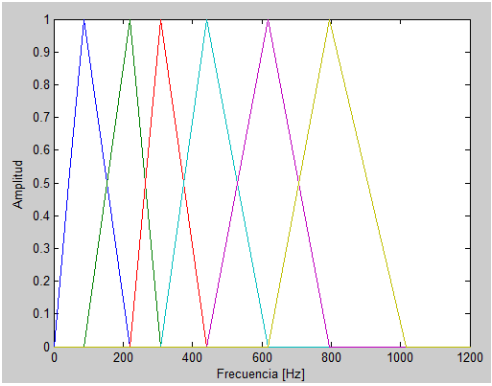


Figura 3.34 Ejemplo de filtros utilizados para la reducción de las dimensiones del logaritmo del espectro.

Como puede observarse cada filtro dará énfasis a un rango de frecuencias dado y eliminará toda frecuencia fuera del mismo. El diseño de estos filtros se lleva a cabo en la llamada $escala\ de\ Mel$. Primero se definen las frecuencias inferior y superior que se desean analizar del espectro, éstas generalmente corresponden a $0\ Hz$ y la frecuencia de $Nyquist$ $(F_s/2)$ respectivamente, donde F_s es la frecuencia de muestreo de la señal. una vez definidas las frecuencias superior e inferior éstas se les aplica la ecuación (3.21) para llevarlas a la llamada $escala\ de\ Mel$.

$$M(f) = 1125ln\left(1 + \frac{f}{700}\right) \tag{3.21}$$

Donde f representa la frecuencia en Hz. Una vez que se tienen las frecuencias inferior y superior en la escala de Mel se procede a dividir el rango en intervalos iguales, en donde el número de intervalos es igual al número de coeficientes $MFCC$ que se desea obtener, en el ejemplo mostrado en la Figura 3.28 es de seis. Cada intervalo estará delimitado por un par de frecuencias en la escala de Mel, éstas ahora deberán llevarse a la escala en Hertz aplicando a cada una de ellas la ecuación inversa a (3.22), ésta es:

$$f(M) = 700\left(\exp\left(\frac{M}{1125}\right) - 1\right) \tag{3.22}$$

En donde M representa el valor de cada frecuencia en la escala de Mel. Con esto se tienen las frecuencias que delimitan cada filtro en Hertz.

Una vez que se tiene el banco de filtros se procede a realizar la reducción de dimensionalidad antes mencionada, para ello se toma uno de los filtros y se multiplica por el vector obtenido al elevar el espectro al cuadrado, para posteriormente sumar los valores, obteniéndose así un escalar. El procedimiento anterior se repite con cada uno de los filtros, obteniéndose un vector de dimensión igual al número de filtros. Una vez que se tiene este vector se calcula el logaritmo de cada una de sus entradas y a este nuevo vector se le aplica la transformada discreta del coseno (DCT). El vector resultante son los $MFCCs$ de la señal dada. La Figura 3.35 muestra el procedimiento descrito.

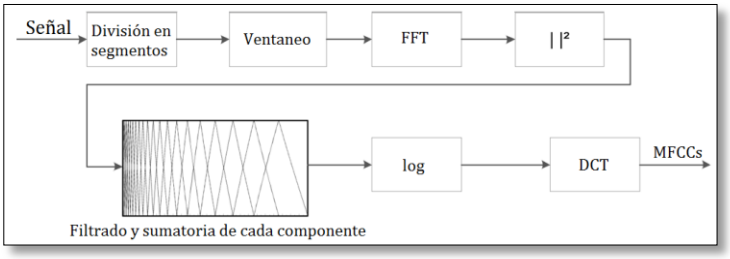


Figura 3.35 Procedimiento para la obtención de los MFCCs de una señal.

3.18.7 Redes neuronales artificiales

Para comenzar con las redes neuronales es necesario analizar su elemento constitutivo, éste es la neurona artificial. Una neurona artificial consiste en una función matemática que va del dominio $\mathbb{R}^n$ al conjunto de los números reales, es decir, contradominio $\mathbb{R}$. Una manera de representar gráficamente el mapeo que realiza la neurona se presenta en la Figura 3.36.

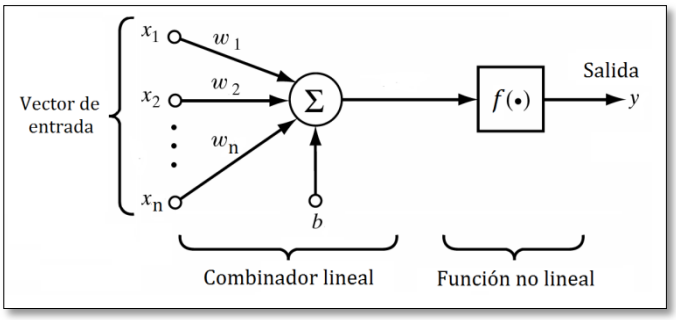


Figura 3.36 Neurona artificial. Los escalares w_i son los pesos sinápticos, los cuales se multiplican por las componentes del vector de entrada correspondientes. El escalar b se conoce como bias.

Como puede verse la neurona artificial consta de dos partes, una lineal y una no lineal. La parte lineal consiste en un producto punto entre el vector de entrada $\bar{x} = (x_1, \dots, x_n)$ y un vector $\bar{w} = (w_1, \dots, w_n)$ conformado por los llamados pesos sinápticos, que consisten en una serie de escalares de valores modificables (ajustables) para controlar el comportamiento de la neurona. Para un par de vectores de dimensión n, el producto punto entre ellos se define como:

$$\bar{x} \cdot \bar{w} = \sum_{i=1}^{n} x_i w_i = x_1 w_1 + \cdots + x_n w_n$$

Además de los pesos sinápticos una neurona artificial posee otro parámetro ajustable, éste es el llamado $bias$, el cual consiste en un número real que se suma al resultado del producto punto.

La denominada parte no lineal de la neurona consiste en una función $f(\cdot)$, generalmente no lineal, la cual toma como argumento el resultado de la parte lineal, esto es:

$$y = f(\bar{x} \cdot \bar{w} + b)$$

Donde y representa la salida de la neurona.

La selección de la función $f(\cdot)$ depende del objetivo que se persiga con la neurona artificial, aunque un conjunto de funciones comúnmente usadas son la función lineal, el límite duro, la función logística sigmoidal (comúnmente referida únicamente como sigmoidal) y la tangente hiperbólica. La Figura 3.37 muestra estas funciones.

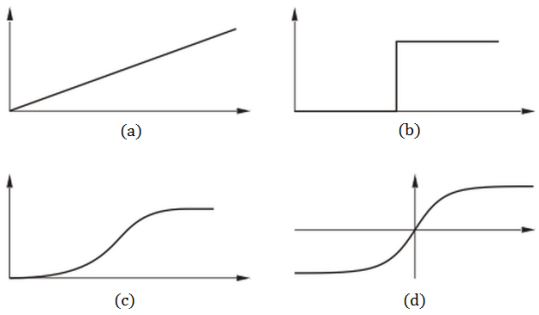


Figura 3.37 Funciones usualmente usadas en las neuronas artificiales. (a) Función lineal. (b) Función límite duro. (c) Función sigmoidal. (d) Función tangente hiperbólica. Tomada de [IRT14].

Las definiciones matemáticas de las que se parte para la función límite duro, sigmoidal y tangente hiperbólica son, respectivamente:

$$H(x) = \begin{cases} 0 \ ; & si\ x < 0 \\ 1 \ ; & si\ x \geq 0 \end{cases}$$

$$sig(x) = \frac{1}{1 + e^{-x}}$$

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Cabe destacar que para la aplicación práctica de las funciones anteriores en una neurona artificial, a éstas suelen aplicárseles traslaciones y escalamientos, como se mostró en la Figura 3.29. En particular a la neurona que instrumenta la función límite duro en su parte no lineal se le conoce como $perceptrón$.

Una vez entendido el concepto de neurona artificial puede pasarse al de $capa$, la cual consiste en un conjunto de neuronas artificiales que toman como entrada un mismo vector. Figura 3.38 ilustra lo descrito.

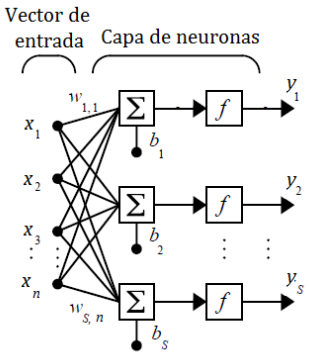


Figura 3.38 Capa de neuronas artificiales. Tomada de [Dem02].

A la arquitectura antes mostrada pueden agregarse más capas intermedias, conocidas como capas ocultas ($hidden\ layers$). Esta arquitectura de red neuronal recibe el nombre de $Feed\ Forward$, pues la transmisión de la información sigue un único sentido, es decir, la salida de una neurona no retroalimenta la entrada de otra en una capa anterior.

3.18.8 Modelos Ocultos de Markov

Los Modelos Ocultos de Markov (HMM por sus sigla en inglés) son un modelo probabilístico que asigna probabilidades a secuencias de símbolos, es decir, un HMM puede entenderse como una máquina que genera secuencias de símbolos [Smi04]. Para presentar la forma en que los HMM funcionan se tomará el ejemplo dado por Rabiner [Rab99]. Imagínese que se tienen N urnas, cada una conteniendo un número grande de pelotas de M colores distintos. Ahora asuma que se elige una de estas urnas aleatoriamente de acuerdo a una distribución de probabilidad inicial π. De la urna elegida se toma una pelota aleatoriamente, se registra su color y se regresa de donde se tomó. Como siguiente paso se elige otra urna de manera aleatoria siguiendo una distribución de probabilidades dada por la urna de la que se tomó la pelota, se toma una nueva pelota de la urna elegida, se registra el color y se regresa de donde se tomó. Continuando con el proceso descrito se

obtiene una secuencia de símbolos (los colores de las pelotas) que serán los llamados $símbolos\ observables$. Dada una secuencia de colores de pelotas no se conoce la urna de la cual se tomó cada pelota, por lo cual las urnas son los llamados $estados\ ocultos$. La Figura 3.39 muestra un HMM compuesto de dos estados ocultos (o urnas retomando el ejemplo anterior) y tres símbolos observables (colores de pelotas).

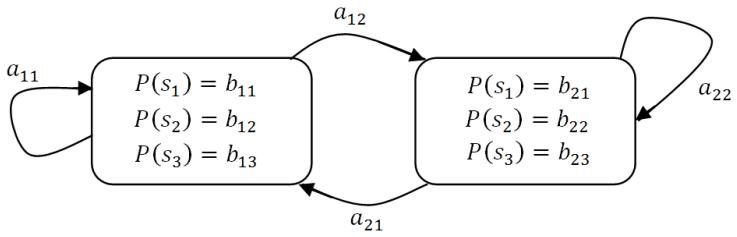


Figura 3.39 HMM compuesto de dos estados ocultos y tres símbolos observables.

En la Figura 3.39 cada a_{ij} representa la probabilidad de ir del estado i al estado j y cada b_{kl} representa la probabilidad de que el estado k emita el símbolo l. Éstas probabilidades pueden agruparse en dos matrices de la siguiente manera:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}$$

Es claro que la suma de los valores de cada una de las filas de las matrices anteriores debe ser igual a uno. Por otro lado la distribución de probabilidades iniciales puede representarse con un vector, en donde cada entrada representa la probabilidad de elegir cuál será el primer estado oculto que emitirá un símbolo. En el caso de la Figura anterior se tendrá un vector $\pi = [\pi_1, \pi_2]$ es donde de nuevo es claro que la sumatoria de las entradas del vector debe ser igual a uno.

Con lo anterior se ve que cualquier HMM puede describirse a través de dos matrices conocidas como matriz de transición A y matriz de observación B y un vector de probabilidades iniciales π. Por conveniencia se utiliza la notación compacta $\lambda = (A, B, \pi)$.

Una vez entendida la estructura de los modelos ocultos de Markov pueden presentarse los tres problemas básicos que permite la aplicación de los mismos en el modelado de problemas reales.

En el primer problema se busca conocer cuál será la probabilidad de que una secuencia $X = (x_1, x_2, \dots, x_T)$ sea generada por un HMM $\lambda = (A, B, \pi)$ dado. Para la resolución de este problema se utiliza el llamado algoritmo $forward$.

El segundo problema consiste en encontrar la secuencia de estados ocultos de un HMM $\lambda = (A, B, \pi)$ que con mayor probabilidad genera una secuencia de símbolos $X = (x_1, x_2, \dots, x_T)$ dada. el llamado algoritmo de $Viterbi$ es el usado para la resolución de este problema.

Finalmente el tercer problema consiste en encontrar los parámetros de un HMM $\lambda = (A, B, \pi)$ que maximicen la probabilidad de generar una secuencia de símbolos $X = (x_1, x_2, \dots, x_T)$ dada. Este problema se resuelve usando el al algoritmo $forward - backward$.

3.18.9 Cromagrama

El cromagrama es un vector de características ampliamente usado en el análisis de señales de audio producidas por instrumentos musicales, éste generalmente tiene una dimensionalidad de doce o veinticuatro.

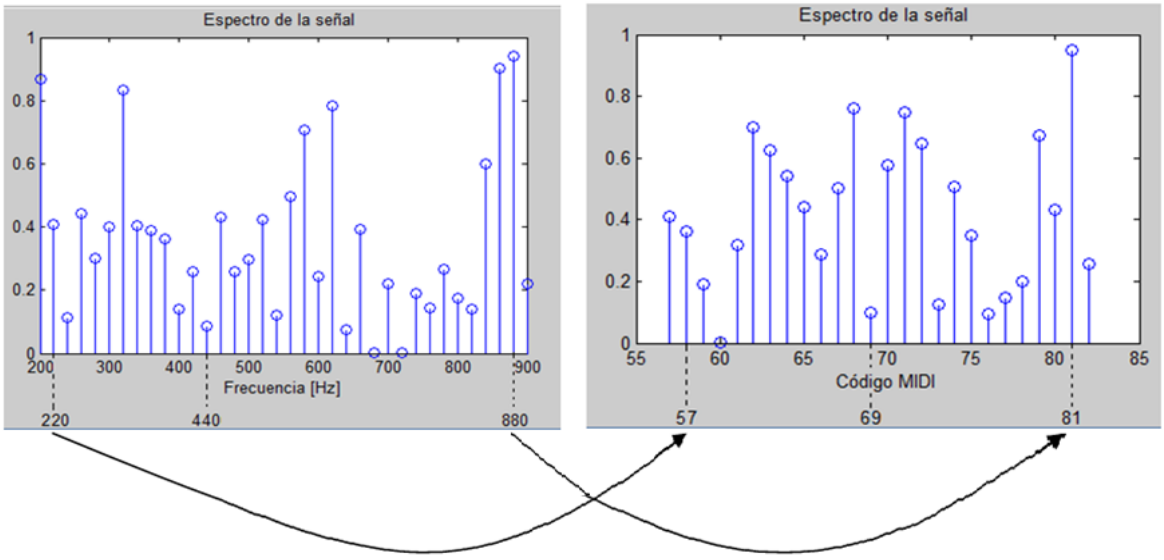
Para la obtención del cromagrama se comienza tomando la DFT de la señal y se mapea a un espectro de semitonos (códigos MIDI) usando la ecuación:

$$n(f_k) = 12 \log\left(\frac{f_k}{440}\right) + 69\,, \quad n \in \mathbb{R}^+ \tag{3.23}$$

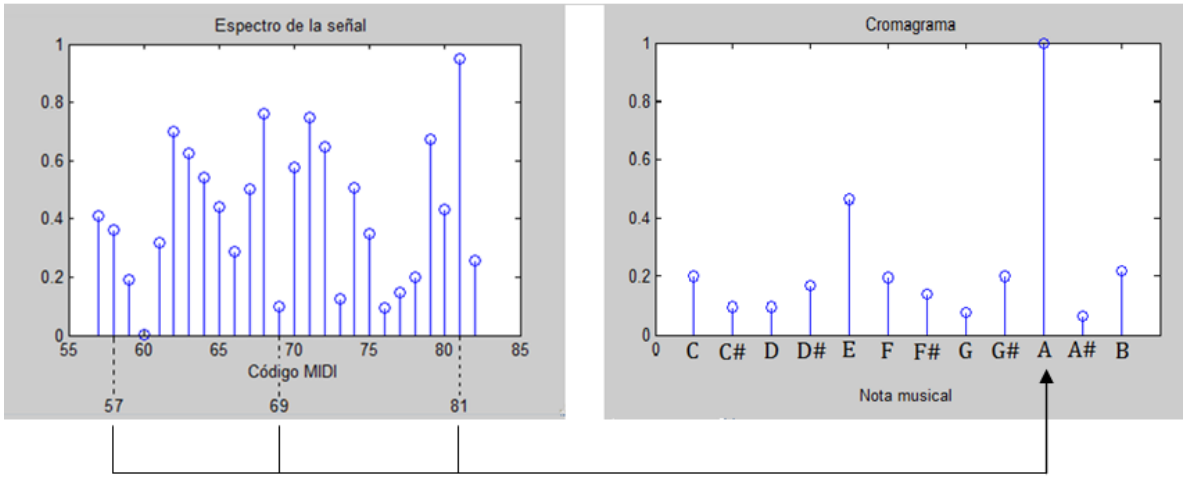
Donde f_k son las frecuencias de la transformada de Fourier y n corresponde a los valores de la escala de semitonos. Luego de esto el espectro de semitonos es suavizado a lo largo del tiempo usando un filtro de media para reducir el ruido lo que, según Papadopoulos [Pap07], mejora significativamente los resultados. Finalmente los n semitonos son mapeados a las clases de semitonos c con el mapeo dado por:

$$c(n) = mod(n, 12) \tag{3.24}$$

Con esto se obtienen los vectores de dimensionalidad doce los cuales sirven como vectores de características para la clasificación de emisiones sonoras. La Figura 3.40 muestra el uso práctico de las ecuaciones (3.23) y (3.24) usando como ejemplo ilustrativo las frecuencias correspondientes a la nota $A3$ y dos de sus harmónicos con frecuencias de 220 Hz, 440 Hz y 880 Hz respectivamente.



(a)



(b)

Figura 3.40 Cómputo del cromagrama. (a) Transformación de escala de frecuencia a código MIDI vía ecuación (3.23). (b) Sumatoria vía ecuación (3.24), el resultado se normaliza.

4 Propuesta de solución.

Antes de comenzar con las pruebas es necesario contar con una serie de pistas de audio que permitan el análisis de las mismas. La grabación de un instrumento musical real supone la necesidad de contar con dispositivos apropiados para tal tarea, como micrófonos, una mescladora de audio y el mismo instrumento en sí, sin dejar de lado a la persona que cuente con los conocimientos y habilidad necesaria para la interpretación. Tomando en cuenta estos requerimientos se optó por el uso de sintetizadores para la generación de las pistas de audio, mismos que toman la partitura de una pieza musical y emulan el sonido de instrumentos musicales reales. Con esto a partir de las pistas sintetizadas se puede hacer el análisis de las mismas para posteriormente llegar a una representación en partitura, la cual que puede ser comparada con la original para observar las similitudes y diferencias. Lo anteriormente descrito se muestra en la Figura 4.1.

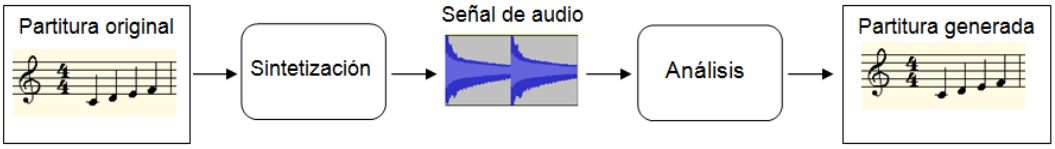


Figura 4.1 Proceso de generación y análisis de pistas de audio.

4.1 Algoritmo para la obtención de una representación musical en pentagrama

Para la obtener una representación ordenada y coherente de la información útil extraída de una señal de audio es necesario establecer un camino que se seguirá a modo de satisfacer los requerimientos mencionados. Para ello se desarrolló un algoritmo que utiliza características propias de una señal acústica generada por un instrumento dado. Para los fines de este trabajo se describirán tres propiedades que servirán como base para el desarrollo del algoritmo.

La primera característica consiste en un cambio de nota, es decir se deja de emitir una nota $N1$ para comenzar a emitir una nota $N2$. La Figura 4.2 muestra lo descrito.

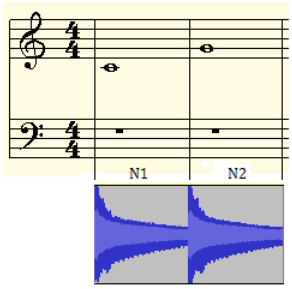


Figura 4.2 Cambio de nota.

La manera en que puede detectarse este cambio de nota es a través de la frecuencia fundamental, es decir, al tomar un intervalo perteneciente a la primera nota se encontrará cierta frecuencia f_1 y al analizar un segundo segmento perteneciente a la segunda nota se encontrará una frecuencia distinta f_2.

Como segunda característica se tomará la repetición de una nota, es decir, se emite repetidamente la misma señal de audio. La Figura 4.3 ilustra lo mencionado.

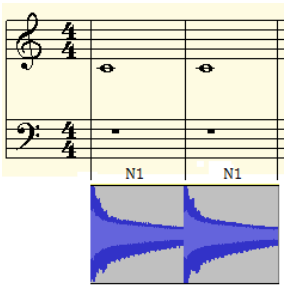


Figura 4.3 Emisión repetida de una nota musical.

En este caso es claro que la frecuencia obtenida en cualquier segmento de análisis será similar por lo que ahora puede tomarse la diferencia de energía para darse cuenta que se está pulsando de nuevo la nota encontrada.

Como característica final se tomarán los silencios en una pieza musical, es decir, intervalos en que ninguna nota es emitida. La Figura 4.4 muestra lo mencionado.

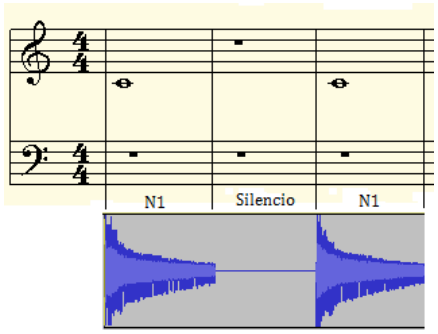


Figura 4.4 Silencio entre notas.

En este caso se puede definir un umbral de energía mínima que servirá para separar los segmentos en donde se encuentra una señal de audio emitida por un instrumento y los lugares en los que no. Con estas observaciones puede procederse al desarrollo del procedimiento para la representación coherente de una señal de audio usando como herramienta el formato MIDI.

Tomando en cuenta las características antes descritas se propone un algoritmo que segmente la señal de audio de entrada, analice uno a uno cada segmento calculando su energía y las frecuencias presentes en el mismo y, con base a los resultados de este análisis, genere la instrucción de activación o desactivación de una nota tomando en cuenta la duración de la misma. El pseudocódigo del algoritmo es el siguiente:

```
// Algoritmo para la representación de una señal de audio haciendo uso del
// formato MIDI

//Inicialización de variables

tAcum   <-  0;      // Variable para acumular el tiempo de duración de una nota
notaAct <- -1;      // Guarda el código numérico de la nota encontrada en el
                    // análisis actual
notaAnt <- -1;      // Guarda el código numérico de la nota encontrada en el
                    // Análisis anterior
tSil    <-  0;      // Tiempo de duración de silencio
umbralE <-  K;      // Se fija el umbral de energía mínima constante (K)
```

```
EAnt    <-  0;      // Guarda el valor de la energía del segmento anterior
fact    <-  C;      // Factor usado para comparar las energías entre segmentos
                    // (C > 1)
tInter  <-  40e-3; // Duración de los intervalos de análisis

Segmentar la señal de audio en intervalos no sobrepuestos de tiempo tInter;
Segm    <-  Primer segmento;
Mientras (Haya segmentos por analizar)
    EAct <- Calcula la energía del segmento Analizado E(s);
    Si (EAct < umbralE)
        //Se detecta silencio, es decir no se emita ninguna nota
        //Se acumula el tiempo del silencio
        tSil <- tsil + tInter;
        notaAct <-  -1;
        Si (notaAnt >= 0)
            //En el análisis anterior se encontró una nota
            //Desactiva la nota anterior
            off(notaAnt, tAcum);
            tAcum <-  0;
        FinSi
    DeOtroModo
        //Se detecta suficiente energía para analizar la nota musical emitida
        notaAct <- Analizar la nota del segmento Segm;
        Si (notaAct == notaAnt)
            //Se encuentra la misma nota que en el segmento anterior
            Si (EAct > fact*EAnt)
                //Se vuelve a pulsar la misma nota que en el segmento anterior
                //Desactiva la nota anterior
                off(notaAnt, tAcum);
                //Activa la nota actual
                on(notaAct, 0);
                tAcum <- tInter;
                tSil  <- 0;
            DeOtroModo
                //Es la misma nota emitida en el segmento anterior
                tAcum <- tAcum + tInter;
            FinSi
        DeOtroModo
            //La nota de intervalo anterior es distinta
            Si (notaAnt >= 0)
                //En el intervalo anterior no hay silencio
                //Desactiva la nota del intervalo anterior
                off(notaAnt, tAcum);
            FinSi
            //Activa la nota del segmento actual
            on(notaAct, tSil);
            tSil <- 0;
            tAcum <- tInter;
        FinSi
        //Reasignación de las variables para el siguiente ciclo de análisis
        EAnt    <- EAct;
        notaAnt <- notaAct;
        Segm    <- siguiente segmento;
    FinSi
FinMientras
```

En donde la instrucción "Analizar la nota del segmento Segm" hace referencia al sistema de redes neuronales que se describe más adelante, el cual recibe una señal de audio, en este caso un segmento de 40 ms, y a la salida entrega la clave MIDI de la o las notas detectadas. Así mismo puede observarse que las funciones en el algoritmo anterior $on()$ y $Off()$ se corresponden con las usadas en el formato MIDI para activar y desactivar una nota musical respectivamente. Además cabe señalar que el algoritmo anterior busca

en todo momento obtener un conjunto de instrucciones MIDI coherentes, es decir, asegura que una nota activada se desactive posteriormente o que una instrucción de desactivación vaya siempre precedida por una señal de activación de nota. En el capítulo 5 se presentará un análisis detallado del presente algoritmo.

4.2 Sistema de redes neuronales para la clasificación de emisiones sonoras

La Figura 4.5 muestra el esquema general propuesto para la identificación de distintas emisiones sonoras, esto es, identificar si el segmento mostrado corresponde a una nota musical o a un acorde, y generar la secuencia MIDI correspondiente.

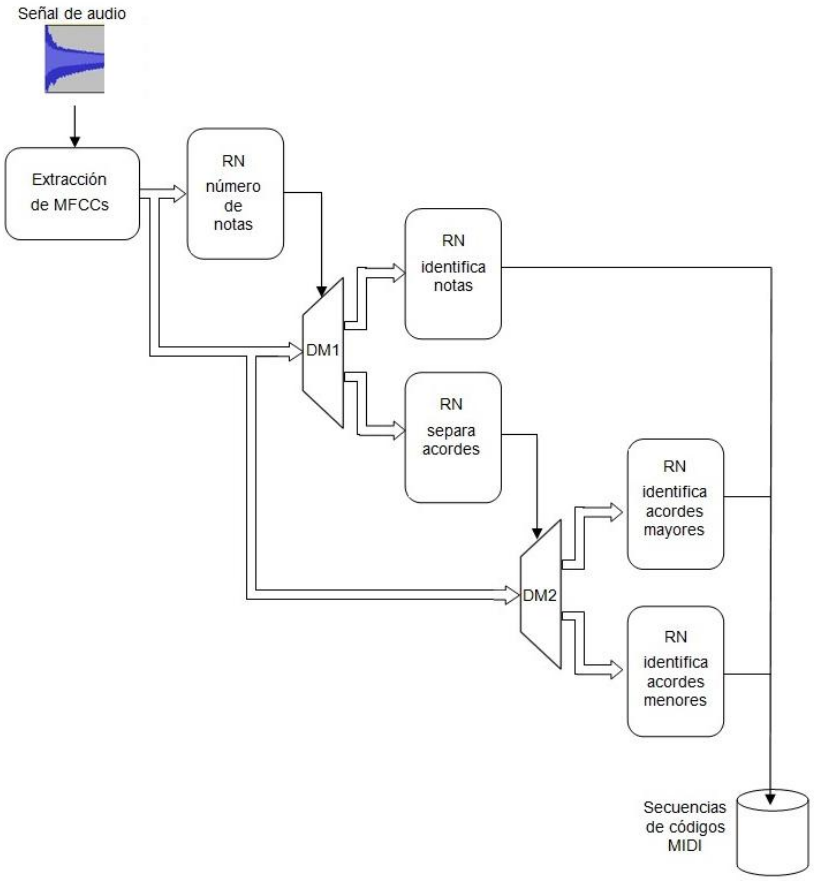


Figura 4.5 Esquema general para la identificación de emisiones sonoras, para el caso de este esquema pueden identificarse notas musicales, acordes mayores y acordes menores.

Como puede observarse, a la entrada del sistema propuesto se recibe una señal de audio, misma a la que se le extrae el vector MFCC. Cada vez que se obtiene un vector MFCC se hace pasar por una serie de redes neuronales, cada una entrenada con una tarea específica, para la identificación de la emisión sonora de la que se trata, en el caso de la Figura anterior se muestra un sistema que identifica entre notas musicales, acordes mayores y acordes menores. La primera red neuronal, etiquetada como "RN número de notas" se entrena para identificar el número de notas que componen al segmento, esto es, toma el vector MFCC como entrada y entrega a la salida un número uno para notas musicales o un tres para acordes. La salida de esta primera red neuronal se utiliza como selector para un demultiplexor, marcado como "DM1", que servirá para seleccionar la red neuronal correspondiente para continuar con el análisis, esto es, en caso de que el selector tome el valor de uno el vector MFCC es enviado a una red neuronal etiquetada como "RN identifica notas" que se entrenó

para la identificación de notas musicales, obteniéndose a la salida los códigos MIDI de las mismas; por otro lado, si el selector toma el valor de tres se envía el vector MFCC a una red neuronal etiquetada como "RN separa acordes" entrenada para la clasificación de acordes, que pueden ser mayores o menores. Una vez que se cataloga un acorde como mayor o menor se procede a identificar de qué acorde específico se trata usando el demultiplexor etiquetado como "DM2" para enviar el vector MFCC a la red neuronal correspondiente para la identificación del mismo. Con esto se obtienen una serie de secuencias MIDI a partir de la señal de audio de entrada. Es claro que el sistema únicamente será capaz de clasificar emisiones sonoras que hayan sido previamente usadas para el entrenamiento de las redes neuronales que lo componen.

4.3 Filtrado de secuencias de símbolos

Después del análisis haciendo uso del algoritmo de detección se procede a la eliminación de ruido dentro de las secuencias MIDI generadas haciendo uso de distintas técnicas que se describirán más adelante entre las que se destacan los Modelos Ocultos de Markov. La Figura 4.6 Diagrama general para la obtención de partituras.Figura 4.6 muestra el diagrama general para la obtención de una representación gráfica de la señal de audio, es decir, la partitura aproximada de la señal de audio de entrada.

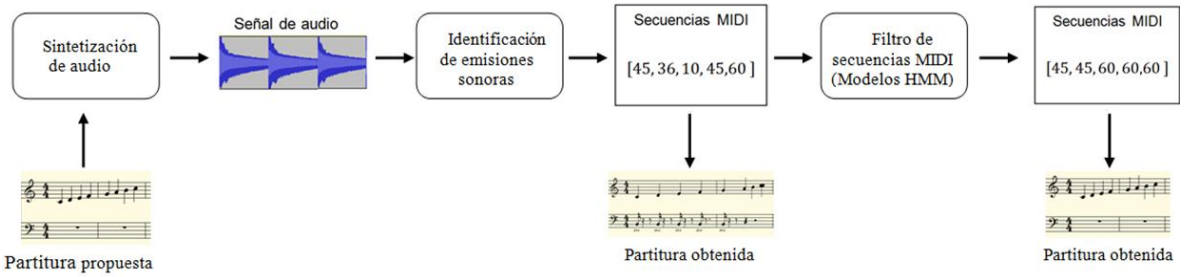


Figura 4.6 Diagrama general para la obtención de partituras.

En este punto puede hablarse de dos procesamientos de señales distintos, en primer lugar se tienen los segmentos de audio que, como ya se mencionó en la sección 3.18.2 Señales de audio producidas por el piano, pueden modelarse como señales cuasi-periódicas y por otro se tienen secuencias de símbolos (códigos MIDI) de naturaleza aleatoria, pues, en principio, una melodía puede estar compuesta de emisiones sonoras que no tienen por qué seguir una regla bien definida.

4.4 Selección de la longitud de los segmentos a análisis y la ventana a utilizar

La longitud de los segmentos a analizar se definió tomando en cuenta dos aspectos. El primero es el rango de frecuencias que se analizó, éste es de 60 a $20000\,Hz$ (la nota más grave incluida es el $Do2$, con una frecuencia de $65.4\,Hz$), con lo que un intervalo de $20\,ms$ es suficiente para albergar al menos un período de la señal de menor frecuencia. Como segundo punto se toma el principio de incertidumbre por el cual se sabe que al tomar un intervalo temporal de mayor longitud se gana resolución en frecuencia a costa de resolución en tiempo. Partiendo de este hecho se realizaron pruebas preliminares para verificar si las redes neuronales utilizadas para la clasificación mejoraban con el aumento en la longitud del segmento, para ello se tomaron segmentos de diferentes longitudes y se entrenaron 100 redes neuronales para cada uno utilizando 24 notas musicales, registrándose el porcentaje de notas correctamente clasificadas. Los resultados se muestran en la Tabla 4.1.

Tabla 4.1 Resultados de las pruebas preliminares para la selección de la longitud del segmento a analizar.

Longitud del segmento a analizar [ms]	Porcentaje de notas correctamente clasificadas
20	96.79%
40	99.68%
80	100%
100	100%

De esta manera se observa que efectivamente al aumenta la longitud del segmento a analizar se aumenta el porcentaje de notas correctamente clasificadas. A modo de buscar la mejor relación entre resolución en tiempo y frecuencia se tomó el intervalos de $40\,ms$, que permite aumentar el porcentaje de clasificación correcta en aproximadamente 3% sin necesidad de aumentar demasiado su longitud con respecto al intervalo de $20\,ms$.

Por otro lado se seleccionó la ventana de Hamming para realizar el ventaneo de los segmentos, esto debido a que reduce el $leakage$ en el espectro, atenuando así la distorsión del espectro en altas frecuencias.

5 Pruebas y resultados

En el presente capítulo se presentan las pruebas y resultados obtenidos haciendo uso de diferentes técnica describiendo los procedimientos seguidos en cada uno de ellos. Se comenzó con pruebas de técnicas básicas como la autocorrelación y la transformada de Fourier para poner en perspectiva las ventajas y desventajas de su uso.

5.1 Análisis por autocorrelación

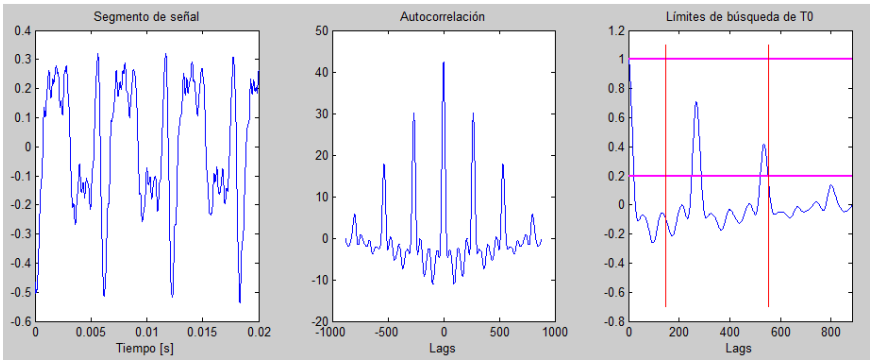
Dentro de las pruebas hechas se comenzó con la instrumentación de la autocorrelación de una señal de audio con el objetivo de encontrar la frecuencia fundamental $F0$ presente en la misma, la cual se tradujo posteriormente en la nota musical correspondiente haciendo uso de:

$$F0 = \frac{1}{T0} \tag{5.1}$$

$$T0 = \frac{nlag}{Fs} \tag{5.2}$$

En donde $T0$ es el periodo fundamental de la señal y $nlag$ representa el número de muestras o $lags$ del primer máximo local (dentro de los intervalos de búsqueda) posterior al máximo global de la función de autocorrelación. Fs representa la frecuencia de muestreo usada, que en lo posterior se fijará en $44100\ Hz$.

Este procedimiento resultó útil únicamente para melodías. La Figura 5.1 muestra el procesamiento descrito para un segmento de señal de $20\ ms$.

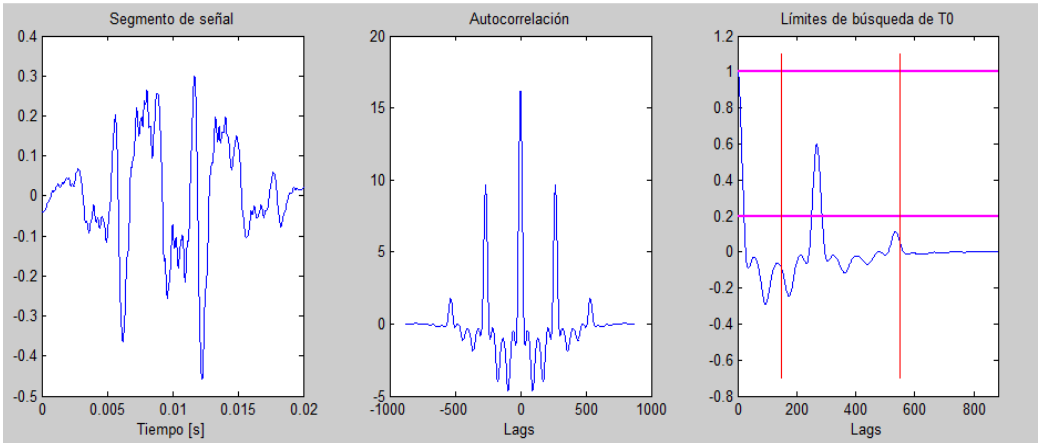


(a) (b) (c)

Figura 5.1 Análisis por autocorrelación. (a) Segmento de la señal. (b) Autocorrelación del segmento dado. (c) Límites de búsqueda para T0.

Las líneas horizontales en (c) de la Figura anterior representan los límites superior e inferior para la búsqueda del máximo local y las líneas horizontales muestran el intervalo de $lags$ permitido, que se traduce en el rango de frecuencias para $F0$. El máximo local dentro de estos intervalos, para el ejemplo dado, se encuentra en $nlag = 268$, con lo que, haciendo uso de la ecuación (5.2) se encuentra que $T0 = 0.0061\ s$ lo cual nos lleva a una frecuencia fundamental $F0 = 165.16\ Hz$. De la Tabla 1.1 puede observarse que la nota musical más cercana a la frecuencia calculada corresponde a un $Mi3$.

El mismo procedimiento antes descrito puede aplicarse esta vez a un segmento de señal ventaneado. La Figura 5.2 muestra esto, en donde se utilizó una ventana de Hamming.



(a) (b) (c)

Figura 5.2 Análisis por autocorrelación. (a) Segmento de la señal ventaneada. (b) Autocorrelación del segmento dado. (c) Límites de búsqueda para T0.

Haciendo un análisis igual al anterior se llega a un resultado similar, encontrándose la nota $Mi3$.

5.1.1 Análisis de melodía por autocorrelación

Dada una melodía se segmentó la misma en intervalos no superpuestos de $40\,ms$, mismos que se ventanearon haciendo uso de una ventana de Hamming. Posteriormente se aplicó a cada uno de éstos la función de autocorrelación de la cual se obtuvo el periodo fundamental $T0$ y frecuencia fundamental $F0$, misma que se tradujo en la nota musical correspondiente. La Figura 5.3 muestra una señal de audio de la escala mayor de Do, sintetizada haciendo uso del software $Guitar\ Pro\ v5.2$ [Gui15] seguida de su correspondiente análisis por autocorrelación.

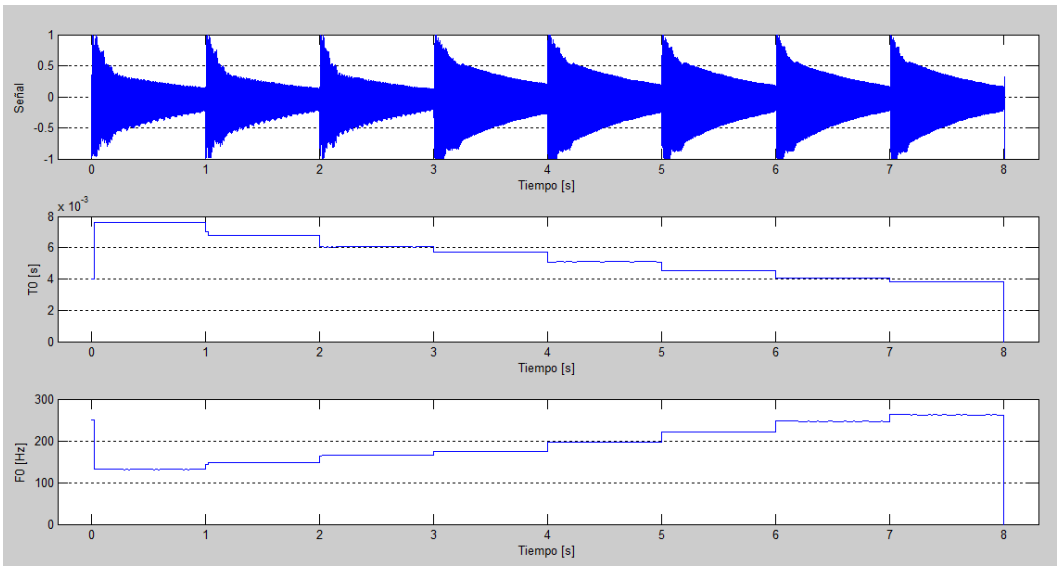


Figura 5.3 Análisis de una melodía por autocorrelación. Parte superior: Señal de audio. Centro: Vector de T0 obtenidas. Parte inferior: Vector de F0 obtenidas.

Como ya se mencionó la señal de audio sintetizada corresponde a la escala de Do mayor. Al analizar las $F0$ encontradas y traducirse las mismas en sus notas correspondientes se encuentran: $Do3, Re3, Mi3, Fa3, Sol3, La3, Si3, D04$, como se esperaba.

Existen casos en el que el vector de frecuencias fundamentales presenta ciertas irregularidades o "ruido" que se traducirá en notas falsas detectadas. La Figura 5.4 ilustra esta situación.

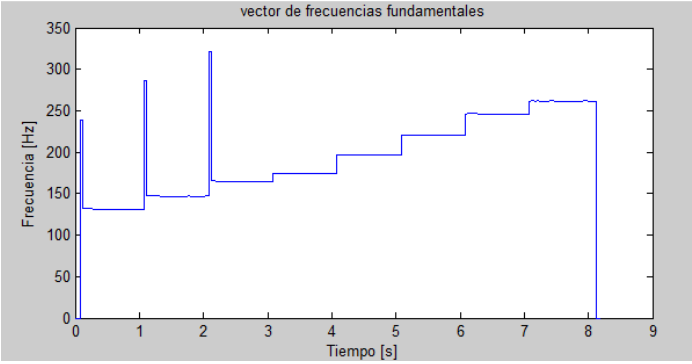


Figura 5.4 Vector de frecuencias fundamentales con irregularidades.

Una manera de solucionar este problema es notar que el tiempo de los "picos" en el vector de frecuencias aparecen por un intervalo de tiempo demasiado corto, es decir, no hay manera que un músico produzca este tipo de interpretación. Fijando un umbral de tiempo que delimite la duración mínima de una nota y llevando cada frecuencia a la frecuencia de la nota musical más cercana se obtiene un vector libre de "ruido", como se ilustra en la Figura 5.5.

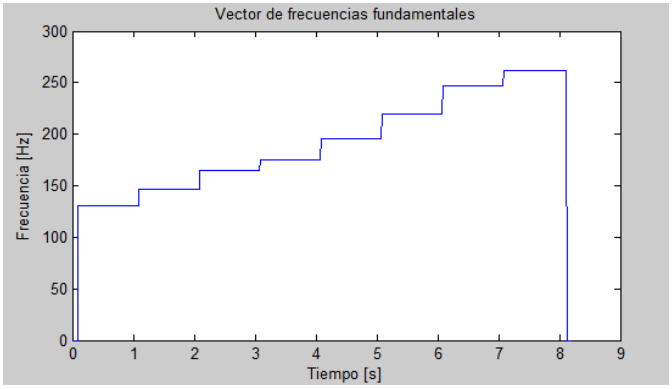
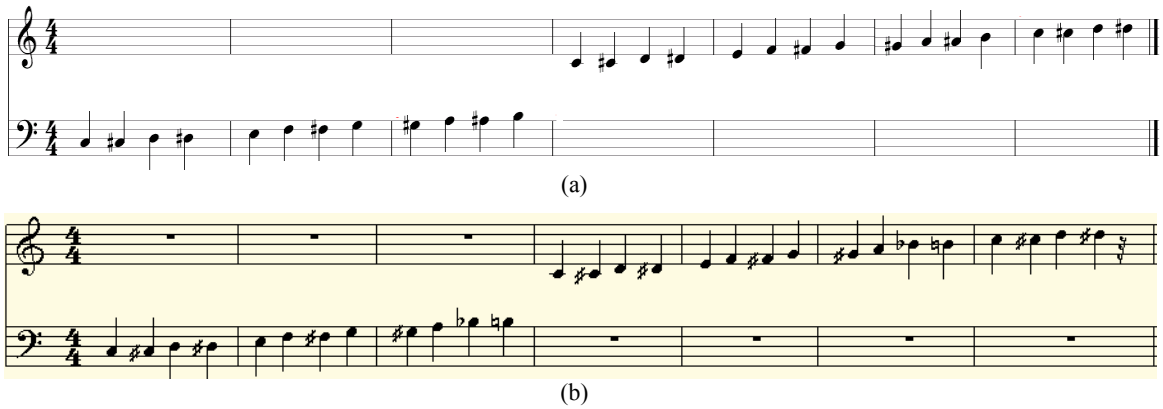


Figura 5.5 Vector de frecuencias fundamentales libre de irregularidades.

Una vez que se tiene un vector "limpio" puede procederse a la síntesis del archivo MIDI correspondiente. La Figura 5.6 muestra la comparación de dos partituras, la primera muestra la pieza original sintetizada y la segunda muestra la representación obtenida usando el procedimiento antes descrito. Para la obtención de la representación gráfica (partitura) del archivo MIDI generado se usó el software *Anvil Studio* [Anv14].



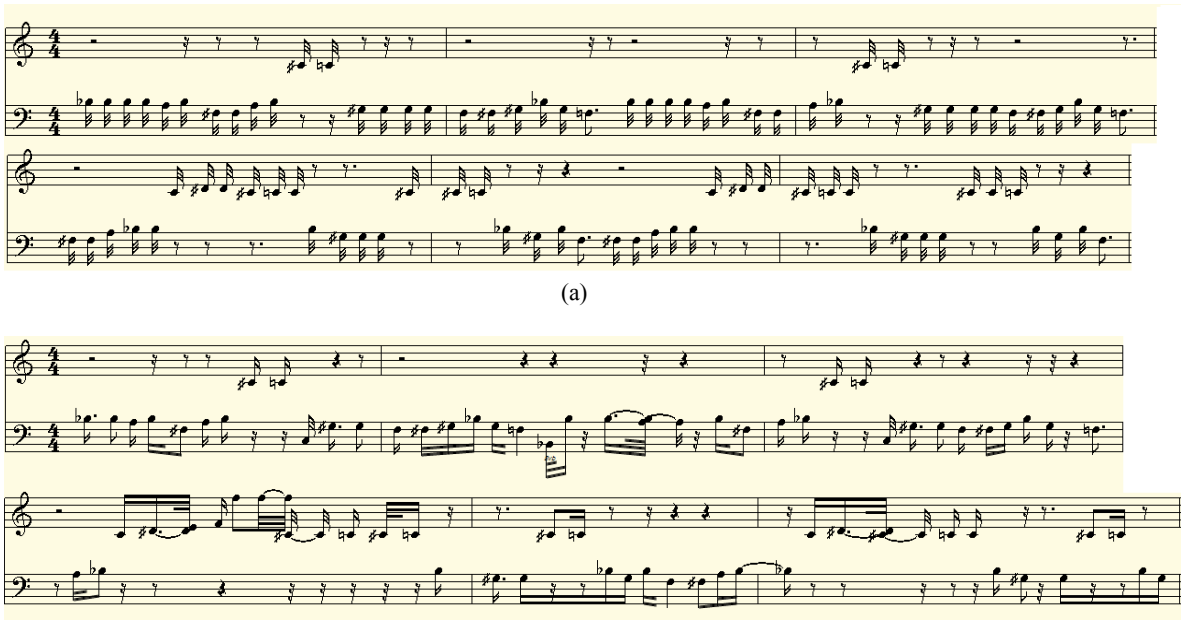
(a)

(b)

Figura 5.6 Comparación entre partituras. (a) Partitura original, usada para la síntesis de la señal de audio. (b) Partitura obtenida a través del análisis por autocorrelación.

En la Figura 5.6 puede observarse la aparición de enarmónicos (dos figuras distintas que hacen referencia a la misma nota) en una representación comparada con la otra, por ejemplo, en la primera partitura se tiene una nota $La\#$, que es la misma que Sib, la cual aparece en la segunda partitura. De esta manera pueden verse que el análisis por autocorrelación resulta muy útil para el análisis de melodías, encontrándose una eficiencia del 100% en señales de audio "sencillas" para las cuales se conoce el tempo.

La pieza musical anterior puede ser catalogada como "simple" para su uso en pruebas, es por ello que se realizaron otras pruebas utilizando una base de datos de archivos MIDI, en este caso, la base de datos $LabROSA$ [Lab08]. La Figura 5.7 muestra la transcripción obtenida a partir de los primeros compases del archivo $bal\_islameiMINp\_align.mid$.



(a)

(b)

Figura 5.7 Transcripción realizada de un archivo de la base de datos LabROSA. (a) Archivo MIDI original. (b) Transcripción generada a partir del audio de la señal.

Es claro que ambas partituras no son exactamente iguales, pero cabe destacar que la estructura general de la melodía es respetada, traduciéndose en dos piezas musicales muy similares entre sí, cuya principal diferencia aparece en los tiempos de duración de las notas, por ejemplo, dos fusas sustituidas por un dieciseisavo.

En conclusión la autocorrelación es un método útil de análisis de melodías para las cuales se conoce el tempo. Cabe destacar que en algunos casos este análisis presenta errores en la octava de la nota resultante del análisis, por ejemplo un $Do3$ en lugar de un $Do4$.

5.2 Análisis por transformada de Fourier

Si bien el análisis por autocorrelación muestra buenos resultados para melodías, el mismo no es útil para armonías. Debido a esto se comenzó con un análisis directo de la transformada de Fourier de una señal.

Para realizar un primer acercamiento se tomó un segmento de señal de audio de $40\,ms$, se ventaneo y se obtuvo su transformada de Fourier, de la que posteriormente se obtuvo el espectro. Una vez que se tiene el espectro se define un umbral, el cual servirá para obtener las componentes frecuenciales que tienen un mayor aporte, las cuales se traducirán en notas musicales. La Figura 5.8 muestra lo descrito utilizándose un umbral del 60% del pico máximo de frecuencia.

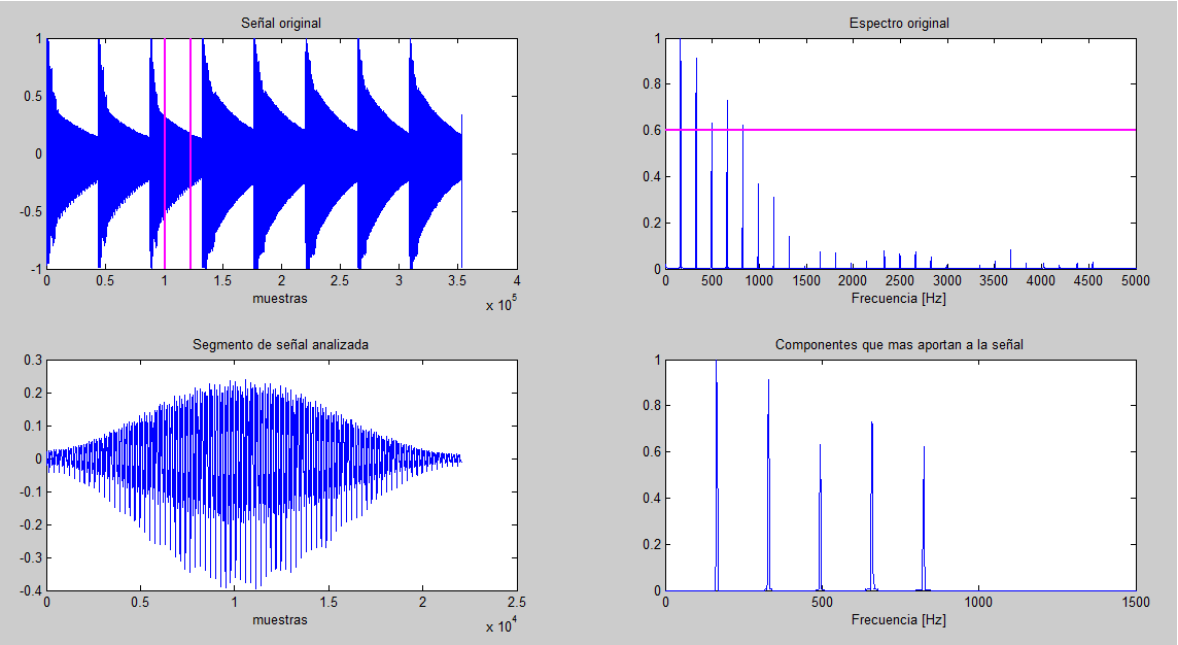


Figura 5.8 Análisis a través de la transformada de Fourier. Superior izquierda: Segmento de la señal a analizar. Inferior izquierda: Segmento ventaneado. Superior derecha: Espectro del segmento. Inferior derecha: Componentes del espectro que superan el umbral establecido.

De antemano se sabe que el segmento analizado corresponde a un $Mi4$, por lo que podría esperarse que la componente de frecuencia que más aporta a la señal fuera la correspondiente a esta nota. Sin embargo, listando las notas de mayor a menor contribución se tiene: $Mi2, Mi3, Mi4, Si4, Sol\#5$. Con esto puede verse que por contribución se puede identificar una nota Mi pero no puede asegurarse la octava a la que pertenece, pues no solo la componente de mayor aporte no corresponde precisamente a la esperada ($Mi4$), sino que aparecen armónicos que en un momento dado podrían dificultar la identificación de la nota emitida.

En el caso de análisis de acordes el panorama no es más alentador, pues podría esperarse que la tónica del mismo apareciera con un mayor aporte en el espectro. La Figura 5.9 muestra el análisis del espectro hecho a un acorde de $Do\ Mayor$.

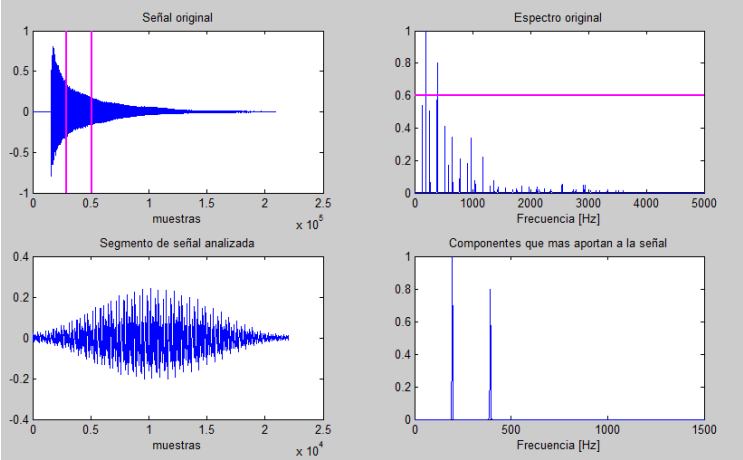


Figura 5.9 Análisis del espectro del acorde de Do Mayor. Superior izquierda: Segmento de la señal a analizar. Inferior izquierda: Segmento ventaneado. Superior derecha: Espectro del segmento, se indica el umbral establecido (60%). Inferior derecha: Componentes del umbral establecido.

Al fijar un umbral del 60 % únicamente se distinguen dos picos en frecuencia, correspondientes a las notas $Sol2$ y $Sol3$, las cuales se ubican como terceras del acorde. Ni la nota Do ni Mi (tónica y tercera) parecen contribuir de manera sustancial al espectro. Si se opta por reducir el umbral a, por ejemplo, 40% se encuentran, en orden de aportación, las notas: $Sol2, Sol3, Do2, Do3, Do4$. A simple vista podría parecer que se está analizando un bicorde en lugar de un acorde, pues únicamente aparecen dos notas. Si se continúa reduciendo el umbral a, digamos, 20%, se tienen las notas: $Sol2, Sol3, Do2, Do3, Do4, Mi4, Si5, Sol4, Re5$. Con este umbral ya aparecen las tres notas constitutivas del acorde (Do, Mi, Sol), pero a su vez aparecen las notas Si y Re, que en principio no pertenecen al acorde (aunque se sabe que esto ocurre debido al contenido armónico de la señal) y que pueden llevar a una mala clasificación de la señal.

En resumen la transformada de Fourier evidentemente arroja resultado útiles para el análisis de notas musicales, pero carece de la precisión necesaria para obtener una transcripción lo suficientemente fiel.

5.2.1 Puntualizaciones en el análisis de las señales haciendo uso de la transformada de Fourier

Cabe destacar que en esta aproximación en el análisis directo de la magnitud del espectro no se tomaron en cuenta efectos que afectan el comportamiento del mismo, como las distorsiones debidas al ventaneo o la aparición de armónicos producidos por el timbre del piano. En este sentido se puede menciona que el efecto de $smearing$, en el caso de señales de audio producidas por instrumentos musicales, no perjudica de manera significativa la clasificación de emisiones, pues las frecuencias de las notas musicales están bien definidas, por lo que se sabe de antemano los intervalos en donde hay que buscar picos en frecuencia. Por otro lado el $leakage$ y los armónicos propios del piano representan una complicación importante en el análisis directo de la magnitud del espectro para la identificación de emisiones sonoras, pues provocan la aparición, atenuación o incluso desaparición de picos en frecuencia que dificultan la tarea de identificación. Si bien el análisis detallado de estos efectos puede ser materia de un estudio minucioso, en el presente trabajo se adopta una perspectiva distinta para la clasificación de emisiones sonoras.

5.3 Uso de $MFCCs$ y $LPCs$

Como pudo observarse en el apartado anterior el uso de la transformada de Fourier no es lo suficientemente preciso para los requerimientos de la transcripción de una señal de audio, por este motivo se prosiguió con la extracción de $MFCCs$ y $LPCs$, mismos que fueron usados como vectores de características para posteriores métodos de clasificación.

La extracción de estos vectores de características se realizó a partir de una señal que sirvió de base para este objetivo, en este caso una señal de audio sintetizada compuesta de las notas musicales de las octavas $2, 3, 4$ y 5, emitidas cada una de éstas repetidas veces, dando un total de 48 notas. Cada uno de estos 48 intervalos se segmento en intervalos de $40\,ms$ superpuestos $20\,ms$ de los cuales se obtuvieron los correspondientes $MFCCs$ y $LPCs$ con una dimensionalidad de 30. Posteriormente haciendo uso de una segunda señal de audio se extrajeron de ella otro conjunto de vectores de características similares a los antes descritos con el objetivo de hacer uso de éstos en la evaluación de una posterior clasificación.

Con esto se obtuvo una base de datos de vectores de características para cuatro octavas, mismas que sirvieron para el entrenamientos de dos conjuntos de redes neuronales, la primera haciendo uso de los $MFCCs$ para su entrenamiento y la segunda con $LPCs$.

La red neuronal utilizada como base fue una $feed\ forward$ con una capa oculta de 20 neuronas con funciones de transferencia sigmoidales y una neurona de salida con una función lineal. Los vectores de objetivos se fijaron de tal modo que cada nota se correspondiera con un número entero, asignando a la nota $Do2$ el valor de uno, a $Do\#2$ el valor de dos y así sucesivamente hasta el número cuarenta y ocho.

Para comenzar el análisis comparativo se tomó un subconjunto compuesto de las primeras doce notas (todas pertenecientes a la octava dos), tomando sus correspondientes vectores de características y entrenando con ellos dos redes neuronales (una con $MFCCs$ y la otra con $LPCs$) repetidas veces. Cada que se completaba un entrenamiento se tomaban vectores de características del conjunto de prueba y se alimentaba con ellos a la red neuronal para evaluar la clasificación hecha por la misma. Para obtener el porcentaje de notas correctamente clasificadas se tomó la salida de la red neuronal para una nota dada y se redondeó su valor al entero más próximo (pues la red neuronal no emite siempre valores enteros) y se comparó con el objetivo fijado en el entrenamiento para la misma. Al final se restó el número de clasificaciones incorrectas al total de notas evaluadas y se dividió entre el total de notas clasificada multiplicando el resultado por 100%. La Figura 5.10 muestra el caso particular de los resultados obtenidos para una red neuronal entrenada con $MFCCs$ tomando un conjunto de doce notas. Una vez entrenada la red se introdujo como entrada cincuenta vectores $MFCC$ por cada nota, llegando a un total de 600.

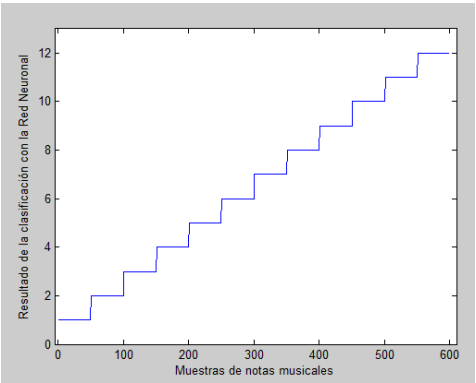


Figura 5.10 Resultados de la clasificación de una red neuronal entrenada con MFCCs

En el caso mostrado en la Figura 5.10 se tiene un porcentaje de notas correctamente clasificadas de 100%. Una vez entrenada la anterior red neuronal cabría preguntarse si es posible clasificar notas musicales fuera del

conjunto de entrenamiento, por ejemplo, alimentar la red neuronal con los $MFCCs$ de una nota $Do3$ (que no se usó para entrenar) y observar si existe algún patrón. Para esto se tomó el conjunto completo de $MFCCs$, tomando cincuenta vectores de cada una de las cuarenta y ocho notas, y se alimentó con ellos a la red neuronal. Los resultados obtenidos se muestran en la Figura 5.11.

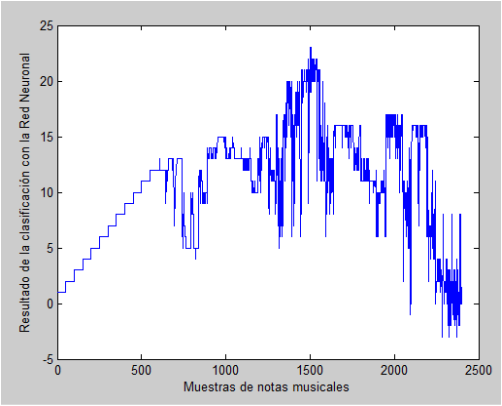


Figura 5.11 Salidas de la red neuronal al alimentar la misma con vectores MFCCs de notas fuera de la octava dos.

Como puede observarse la clasificación no presenta ningún patrón claro y, como era de esperarse, si se presentan notas fuera del conjunto que se utilizó para entrenar la red neuronal se obtienen clasificaciones incorrectas.

Una vez observado lo anterior se procedió a aumentar el conjunto de notas que conforman el conjunto de entrenamiento de uno en uno hasta llegar a un par de redes neuronales entrenadas con cuarenta y ocho notas de modo que pudieran analizarse dos cosas, la primera realizar un análisis comparativo de los vectores de características, es decir, discernir qué vectores de características ($MFCCs$ o $LPCs$) son mejores para el objetivo de reconocer notas musicales y en segundo lugar observar la reducción en el porcentaje de notas correctamente clasificadas conforme se aumenta el conjunto de notas dado, en este caso cuando se va de una a cuatro octavas. Los resultados obtenidos se ilustran en la Figura 5.12.

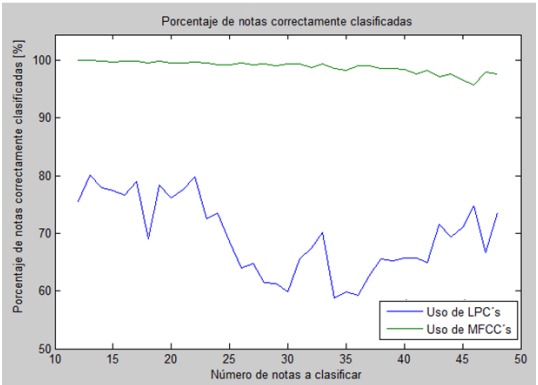


Figura 5.12 Resultados de la clasificación de notas haciendo uso de MFCCs (línea verde) y LPCs (línea azul).

Con este análisis resulta clara la superioridad de los $MFCCs$ por sobre los $LPCs$ para su uso en el reconocimiento de notas musicales. En el caso de $MFCCs$ el mayor y menor porcentaje obtenidos fueron, 100% y 95.95% respectivamente y de 80.15% y 58.76% para los $LPCs$. La Figura 5.13 muestra las salidas de la red neuronal con mejor porcentaje de clasificación correcta (97.58%) para cuarenta y ocho notas.

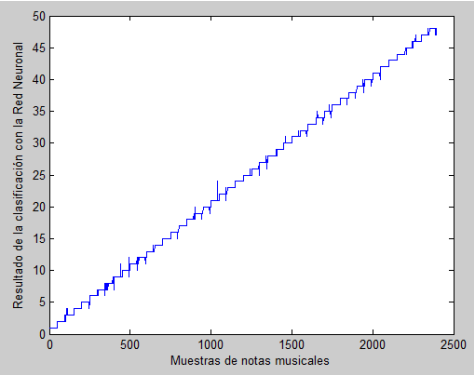


Figura 5.13 Resultados para la red neuronal entrenada con cuarenta y ocho notas.

Una vez hechas las pruebas de clasificación para notas musicales aisladas se procedió con las pruebas en acordes mayores. Para el análisis de los mismos se procedió de manera similar al que se siguió con las notas musicales aisladas, realizando la sintetización de los acordes mayores en posición original comenzando con el acorde de $Do\ Mayor$ en la octava dos seguido por el acorde de $Do\#\ Mayor$, $Re\ Mayor$, y así sucesivamente hasta el acorde de $Si\ Mayor$ en la octava cuatro, dando un total de 48 acordes mayores. La Figura 5.14 muestra una partitura en la que se representan los primeros 12 acordes de la serie descrita.

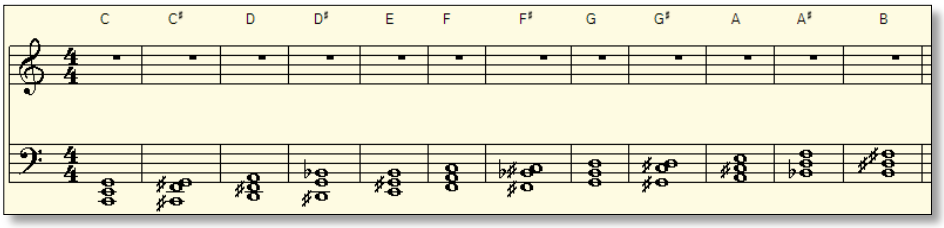


Figura 5.14 Primeros doce acordes mayores sintetizados.

Una vez sintetizados los acordes se procedió a extraer los $MFCCs$ y $LPCs$ de los mismos, segmentados en intervalos de 40 ms. Con ellos se entrenó una serie de redes neuronales, una con los $MFCCs$ y otra con $LPCs$ a modo de realizar una comparación en el porcentaje de notas correctamente clasificadas. Se comenzó alimentando ambas redes inicialmente con conjuntos compuestos por doce acordes y se fue incrementando el número de uno en uno hasta llegar a un par de redes entrenadas con los cuarenta y ocho acordes sintetizados.

La Figura 5.15 muestra los resultados obtenidos.

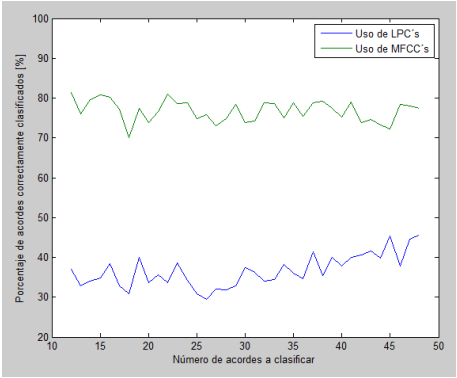


Figura 5.15 Resultados de la clasificación de acordes mayores haciendo uso de MFCCs (línea verde) y LPCs (línea azul).

De nuevo es claro que los $MFCCs$ son mejores par la clasificación de acordes, resultado que es consistente con la clasificación de notas musicales aisladas. En este caso el porcentaje mayor y menor de notas correctamente clasificadas son 81.33% y 70% respectivamente para la red entrenada con $MFCCs$, lo que significa una disminución importante con respecto a la clasificación de notas aisladas, lo cual era de esperarse, tomando en cuenta que el contenido armónico de un acorde es más compleja que la de una nota musical aislada.

Una alternativa usada para tratar de mejorar el porcentaje de notas correctamente clasificadas fue el de aumentar el número de neuronas en la capa oculta, comenzando con veinte neuronas y aumentando progresivamente este número de cinco en cinco, llegando hasta ochenta. La Figura 5.16 muestra los resultados obtenidos.

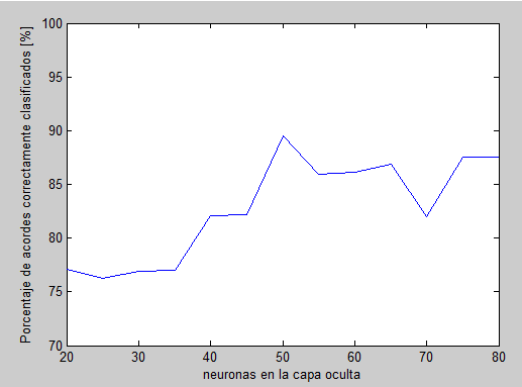


Figura 5.16 Resultados de la clasificación de 48 acordes mayores usando redes neuronales con diferentes números de neuronas en la capa oculta.

En la Figura 5.16 puede observarse el aumento, hasta cierto punto, en el porcentaje de acordes correctamente clasificados conforme se aumenta el número de neuronas, llegando a un máximo de 90% para una red neuronal artificial con una capa oculta compuesta de cincuenta neuronas.

De manera similar se entrenaron las redes neuronales que componen el sistema mostrado en la Figura 4.5 llegando a los porcentajes máximos de emisiones sonoras correctamente clasificadas mostradas en la Tabla 5.1.

Tabla 5.1 Porcentaje máximo de emisiones sonoras correctamente clasificadas.

Clasificación de la red neuronal	Porcentaje de emisiones correctamente clasificadas
Número de notas	96%
Tipo de acorde (mayor o menor)	96%
Clasificación de notas musicales	95%
Clasificación de acorde mayores	90%
Clasificación de acordes menores	90%

Con esto se pueden calcular los porcentajes de clasificación correcta para cada tipo de emisión sonora tomando en cuenta las etapas seguidas, esto es, para clasificar correctamente un acorde mayor, es necesario que sea clasificado correctamente en la red neuronal de número de notas, seguida de una clasificación correcta en el tipo de acorde y finalmente una clasificación correcta en el acorde dado, lo que daría un porcentaje de $0.96 * 0.96 * 0.90 = 82.9\%$, de la misma manera se llega a un porcentaje de 92.1% para notas musicales y 82.9% para acordes menores.

5.4 Análisis del algoritmo propuesto en condiciones ideales

Para ejemplificar el funcionamiento del algoritmo descrito en la propuesta de solución se presentará su aplicación en una señal de audio sencilla que permita observar el funcionamiento del mismo. La Figura 5.17 muestra la señal a analizar.

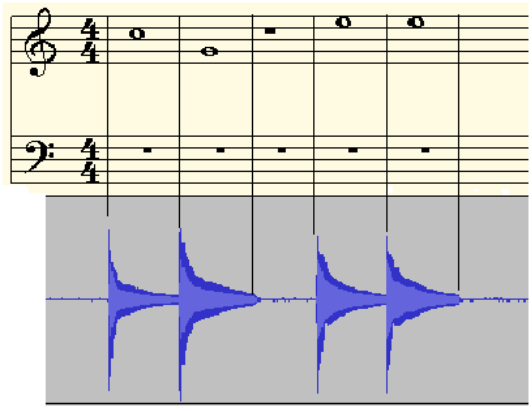


Figura 5.17 Señal a analizar haciendo uso del algoritmo propuesto.

La señal anterior se interpreta con un tempo de 240 bpm, lo que implica que la duración de cada redonda en de 1 s. Para realizar un análisis rápido se tomarán intervalos de 0.5 s para segmentar la señal. La Figura 5.18 muestra la señal a analizar acompañada de la energía normalizada de cada segmento y el límite de energía mínima que establece el algoritmo.

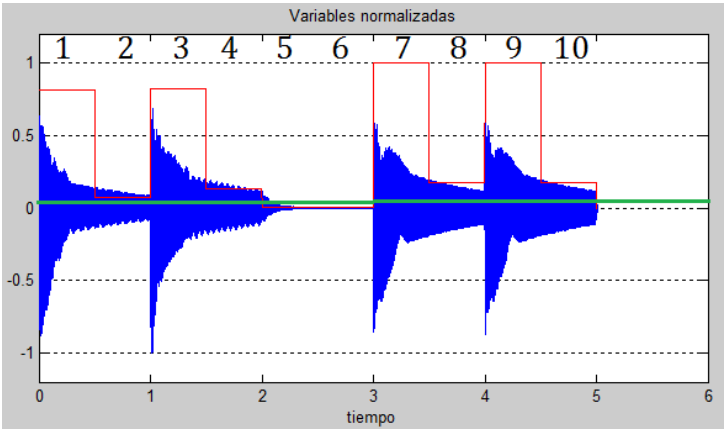


Figura 5.18 Análisis de energía de la señal a intervalos de 0.5 s. Cada intervalo se numera en la parte superior.

En total se tienen 10 intervalos que se numeraron en la parte superior de la señal. El algoritmo comienza con la inicialización de variables y la segmentación de la señal, una vez hecho esto se procede con el análisis ordenado de los segmentos hasta que ya no haya segmentos por analizar. El análisis se llevará a cabo respondiendo las condicionales del algoritmo e indicando las implicaciones de las mismas.

Primer segmento:

- La energía es mayor que el umbral ⇒ Se analiza la nota.
- La nota actual es diferente a la anterior.
- La nota anterior es menor a cero ⇒ No se desactiva ninguna nota.
- Se activa la nota actual, con un retardo de $0\ s$.
- Se asigna: Tiempo de silencio igual a cero.
- Se asigna: Tiempo acumulado igual a $0.5\ s$.
- Pasa al siguiente segmento

Segundo segmento:

- La energía es mayor que el umbral ⇒ Se analiza la nota.
- La nota actual es igual a la anterior.
- La energía actual es menor a la del segmento anterior (multiplicada por el factor "fact", usado para comparar las energías entre segmentos).
- Se asigna: Tiempo acumulado igual a $1\ s$.
- Pasa al siguiente segmento.

Tercer segmento:

- La energía es mayor que el umbral ⇒ Se analiza la nota.
- La nota actual es diferente a la anterior.
- La nota del intervalo anterior es mayor a cero ⇒ Se desactiva la nota anterior con un tiempo de duración de $1\ s$.
- Se activa la nota actual, con un retardo de $0\ s$.
- Se asigna: Tiempo de silencio igual a cero.
- Se asigna: Tiempo acumulado igual a $0.5\ s$.
- Pasa al siguiente segmento

Cuarto segmento: Igual que el segundo segmento.

Quinto segmento:

- La energía en menor que el umbral.
- Se asigna: Tiempo de silencio igual a $0.5\ s$
- Se asigna: Nota actual igual a -1.
- La nota anterior es mayor a cero ⇒ Se desactiva la nota anterior con un tiempo de duración de $1\ s$.
- Pasa al siguiente segmento.

Sexto segmento:

- La energía en menor que el umbral.
- Se asigna: Tiempo de silencio igual a $1\ s$.
- La nota anterior es menor a cero.
- Pasa al siguiente segmento.

Séptimo segmento:

- La energía es mayor que el umbral ⇒ Se analiza la nota.
- La nota actual es diferente a la anterior.
- La nota anterior es menor a cero ⇒ Se activa la nota actual con un retardo de $1\ s$.
- Se asigna: Tiempo de silencio igual a cero.
- Se asigna: Tiempo acumulado igual a $0.5\ s$.
- Pasa al siguiente segmento.

Octavo segmento: Igual que el segundo segmento.

Noveno segmento:

- La energía es mayor que el umbral ⇒ Se analiza la nota.

- La nota actual es igual a la anterior.
- La energía actual es mayor a la del segmento anterior (multiplicada por el factor).
- Se desactiva la nota anterior con un tiempo de duración de $1\ s$.
- Se activa la nota actual con un retardo de $0\ s$.
- Se asigna: Tiempo de silencio igual a cero.
- Se asigna: Tiempo acumulado igual a $0.5\ s$.
- Pasa al siguiente segmento

Décimo segmento: Igual al segundo segmento.

- No hay más segmentos por analizar ⇒ Fin del algoritmo.

Los eventos MIDI generados mediante el análisis anterior se resumen en la Tabla 5.2.

Tabla 5.2 Eventos MIDI obtenidos por el algoritmo especificando tiempo de ejecución y segmento del análisis del que provienen.

Tiempo [s]	Eventos MIDI	Segmento del que surge la instrucción MIDI
0	Activa nota ($Do5$)	1
0.5	-	-
1	Desactiva nota ($Do5$) Activa nota ($Sol4$)	3
1.5	-	-
2	Desactiva nota ($Sol4$)	5
2.5	-	-
3	Activa nota ($Mi5$)	7
3.5	-	-
4	Desactiva nota ($Mi5$) Activa nota ($Mi5$)	9
4.5	-	-
5	Desactiva nota ($Mi5$)	10

Como se mencionó anteriormente el algoritmo cumple con la necesidad de ser compatible con los eventos MIDI de activación y desactivación de las notas musicales. Así mismo al realizar la representación gráfica (partitura) de el conjunto de instrucciones anteriores se obtiene una igual a la que se mostró en la Figura 5.18.

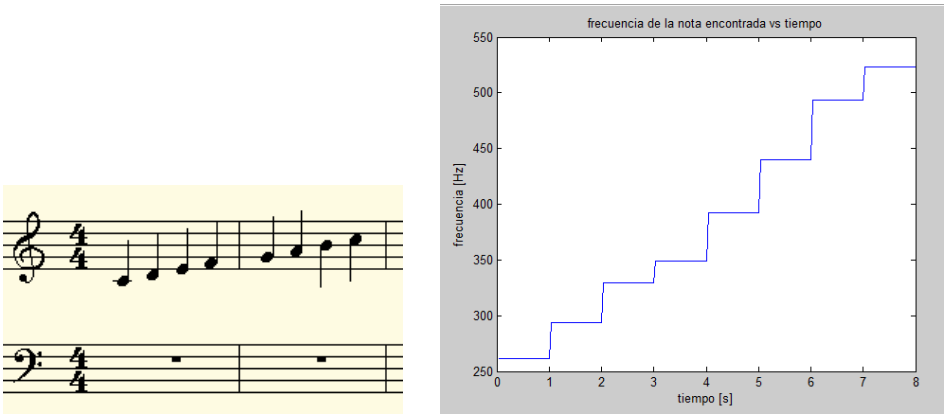
5.5 Limpieza de las instrucciones MIDI generadas.

Como se mencionó anteriormente se utilizaron intervalos de análisis de 40 ms. Este intervalo de tiempo puede o no coincidir con un múltiplo o submúltiplo del intervalo entre $beats$ de una melodía a analizar. Por ejemplo, supóngase que se tiene una pista de audio con un tempo de $60\ bpm$, con compases de cuatro cuartos. Con esto se puede saber la duración que cada figura tendrá, esto es, la negra tendrá una duración de un segundo, la figura blanca tendrá una duración de dos segundos y la redonda una duración de cuatro segundos. Por otro lado la duración de una corchea (mitad de una negra) será de $0.5\ s$, una semicorchea (mitad de una corchea) tendrá una duración de $0.25\ s$ y una fusa (mitad de una semicorchea) tendrá una duración de $0.125\ s$. El uso de semifusas no es común, por lo que podría tomarse a la fusa como límite inferior en la duración de la emisión de una nota musical o silencio. La Tabla 5.3 resume lo descrito.

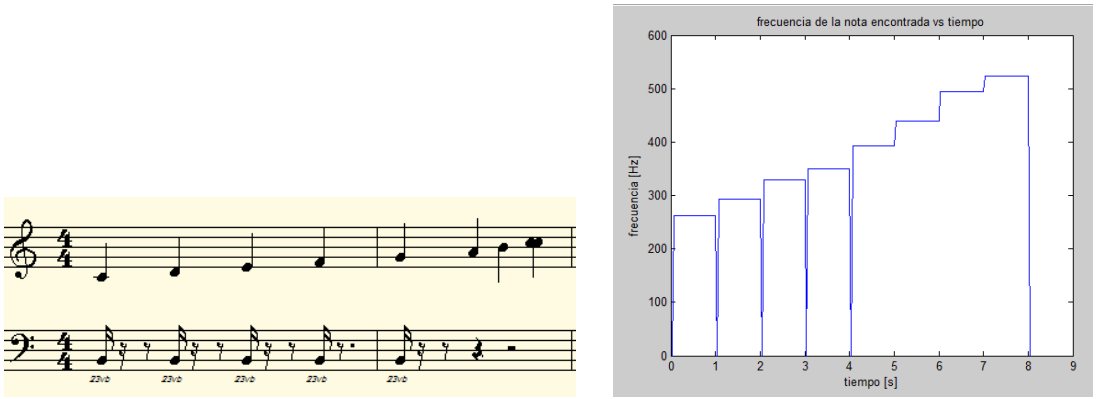
Tabla 5.3 Duración de las notas musicales para una pieza musical con compases de cuatro cuartos a 60 bpm.

Figura	Duración de la nota [s]
Negra	1
Blanca	2
Redonda	4
Corchea	0.5
Semicorchea	0.25
Fusa	0.125

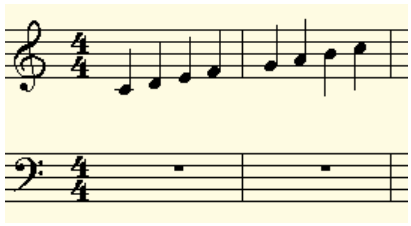
Como puede observarse usando intervalos de $40\ ms$, en general, no se obtienen múltiplos o submúltiplos de la duración de una figura, por ejemplo, supóngase que durante el análisis se encontraron 16 intervalos consecutivos con la nota Do, esto nos llevaría a una duración de $16 * 40\ ms = 0.64\ s$. Suponiendo además que la figura musical de menor duración dentro de la pista es una semicorchea, se puede observar que los $0.64\ s$ no se corresponden con una combinación de figuras musicales, hablando en cuestión de duración. En este sentido la generación de emisiones MIDI como múltiplos de intervalos de $40\ ms$ resulta inexacta, por lo cual se realizó un ajuste de redondeo presuponiendo que se conoce el tempo de la pista a analizar, ajustando los intervalos de análisis al múltiplo más cercano de la figura de duración mínima. En el presente ejemplo la duración de la nota encontrada fue de $0.64\ s$, mismos que pueden redondearse a una duración de $0.625\ s$, que es equivalente a la duración de cinco fusas (para un tempo de $60\ bpm$). La Figura 5.19 muestra una comparación de las partituras obtenidas haciendo uso del algoritmo para la obtención de una representación musical, seguida por el uso de la limpieza del vector de notas musicales mencionado en la sección de *análisis por autocorrelación*, eliminando intervalos de duración demasiado corta, y posteriormente haciendo uso del redondeo presentado en esta sección y finalmente una combinación de ambos métodos de limpieza (primero eliminando intervalos de duración pequeña y posteriormente redondeando).

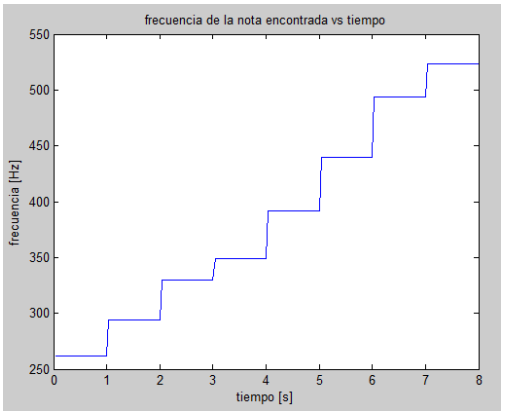


(a)

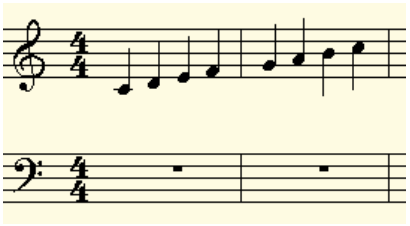


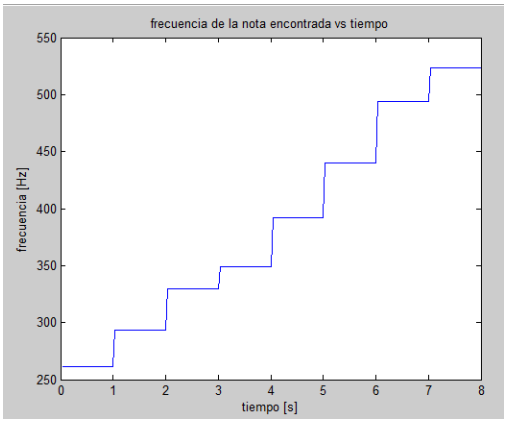
(b)



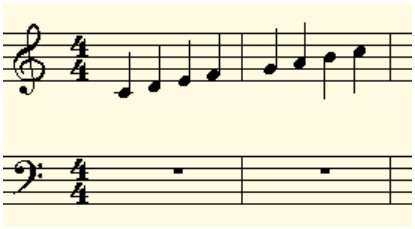


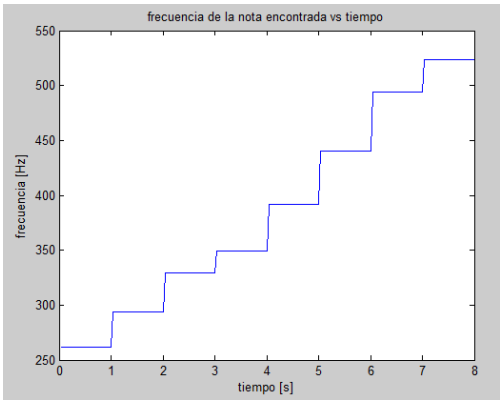
(c)





(d)

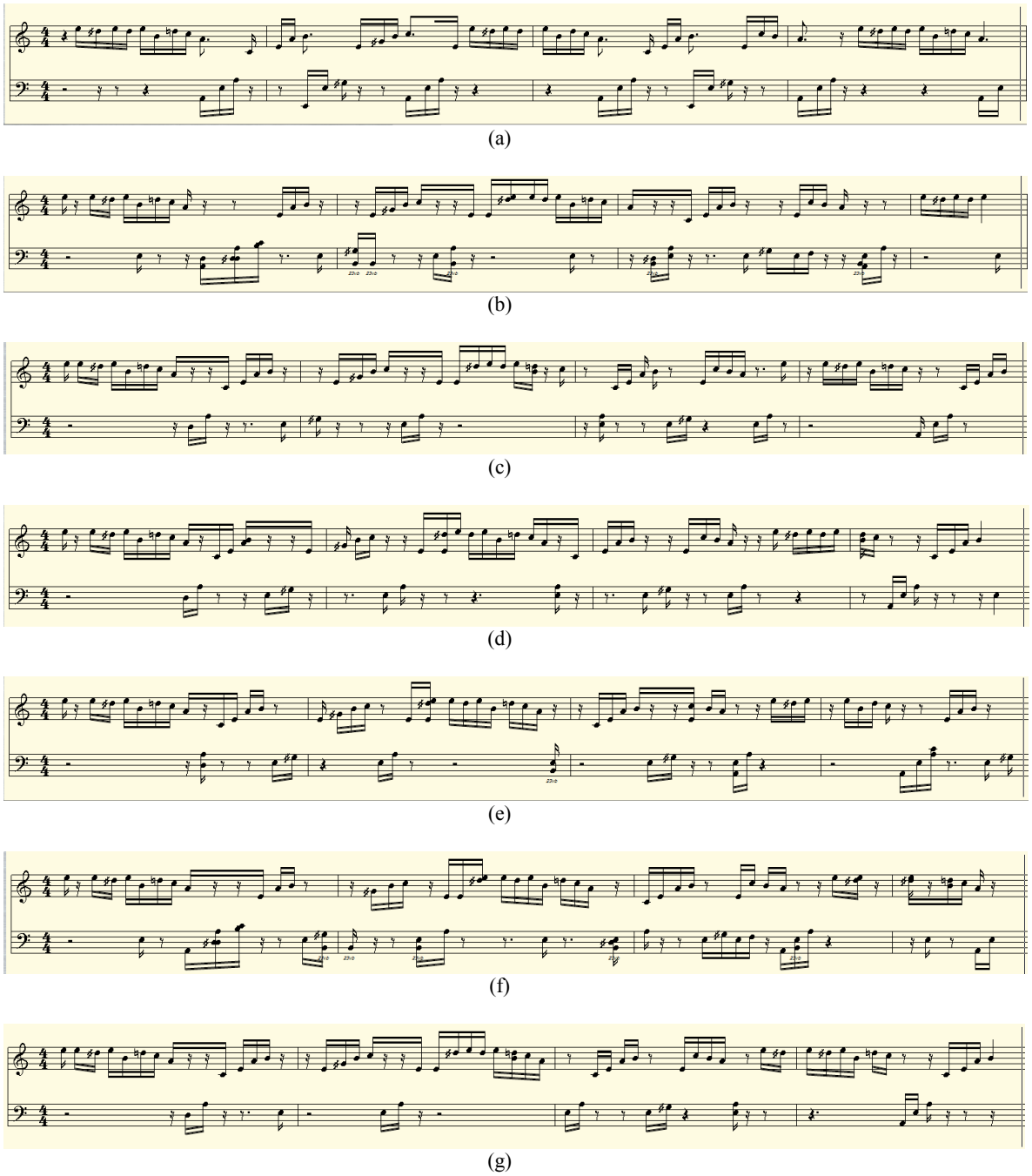




(e)

Figura 5.19 Limpieza de las instrucciones MIDI. A la derecha se muestran las partituras y a la izquierda el correspondiente vector de frecuencias. (a) Pista original. (b) Resultado obtenido al hacer uso del algoritmo propuesto sin limpieza de instrucciones. (c) Limpieza de instrucciones eliminando intervalos de duración corta. (d) Limpieza de instrucciones haciendo uso de redondeo de duraciones. (e) Limpieza de instrucciones aplicando eliminación de intervalos de corta duración y posteriormente redondeo de intervalos.

Como puede observarse es importante realizar la limpieza de las instrucciones MIDI obtenidas, pues de ello depende la generación de una partitura más legible. Cabe destacar que al reproducir las partituras en la Figura 5.19a y la Figura 5.19b las señales de audio son muy parecidas entres sí, pero al ser interpretadas por una persona pueden generar confusión, pues la aparición de semicorcheas provoca que no se respeten los intervalos de cuatro cuartos. Así mismo, en un principio, podría parecer que no existe diferencia en las diferentes técnicas de limpieza utilizadas, sin embargo hay que tomar en cuenta que la señal de audio utilizada no presenta un patrón "complicado". Al aplicar un procesamiento similar al anterior a la pista beet_eliseMINp_align.mid de la base de datos *LabROSA* [Lab08] se obtienen los resultados mostrados en la Figura 5.20.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figura 5.20 Partituras obtenidas realizando la limpieza de las instrucciones MIDI. (a) Pista original. (b) Resultado obtenido al hacer uso del algoritmo propuesto sin limpieza de instrucciones. (c) Limpieza de instrucciones eliminando

intervalos de duración corta. (d) Limpieza de instrucciones haciendo uso de redondeo de duraciones, tomando un octavo de negra como duración mínima (e) Limpieza de instrucciones haciendo uso de redondeo de duraciones, tomando un dieciseisavo de negra como duración mínima (f) Limpieza de instrucciones haciendo uso de redondeo de duraciones, tomando un treintaidosavo de negra como duración mínima (g) Limpieza de instrucciones aplicando eliminación de intervalos de corta duración y posteriormente redondeo de intervalos a octavos de negra.

Como puede observarse en una pista de audio más diversa el cambio entre los distintos procedimientos ya es notoria. En el caso del primer análisis la partitura presenta muchas notas que no deberían aparecer, pues el algoritmo activa toda nota sin importar su duración. En el caso de la limpieza por eliminación de intervalos muchos de estos ruidos desaparecen. Por otro lado se probaron tres distintos intervalos de redondeo, correspondientes a un octavo, un dieciseisavo y un treintaidosavo de negra, en el caso del primer intervalo se presenta en la pista obtenida una clara distorsión en el ritmo de la misma, mientras que al disminuir la duración del redondeo se aprecia una mejor aproximación a la señal de audio original. Finalmente para el caso de la limpieza combinada se asemeja mucho al resultado de limpieza por dieciseisavos de negra.

5.6 Uso de Modelos Ocultos de Markov (HMM) para el reconocimiento de notas musicales.

Una pieza musical dada, similar a lo que ocurre con una señal de voz, puede ser codificada como una secuencia de símbolos de naturaleza aleatoria, pues aunque una melodía está sujeta a algunas reglas armónicas hay que tomar en cuenta que el músico tiene la libertad de emitir sonidos de manera libre en el orden que mejor le parezca. Esto lleva al uso de HMM como herramienta natural para modelar la generación de melodías.

5.6.1 Primera aproximación en el uso de HMM

Para comenzar con la instrumentación de los HMM se propuso un modelo para el reconocimiento de notas musicales aisladas, es decir dada una cadena de patrones se presupone que corresponde a una única nota musical, donde los símbolos emitidos por el HMM se corresponden con los códigos MIDI de las notas. Como primera aproximación se planteó un modelo de dos estados ocultos, nombrados "emisión de nota" y "no emisión". En un principio se supone que la emisión de las notas musicales es exactamente la misma, esto es, la probabilidad inicial de todas las notas es la misma. Una vez que se emite una nota existe una alta probabilidad de permanecer en ella, aunque se puede presentar algún símbolo que no corresponda a la misma, este "ruido" se puede incluir dentro de las probabilidades de estado de emisión. Por otro lado el estado de no emisión se incluye para identificar secuencias de caracteres que no corresponden a la nota específica del modelo. De esta manera se puede ir de un estado de no emisión a un estado de emisión en cualquier momento y viceversa. Con lo anterior se llega al HMM mostrado en la Figura 5.21.

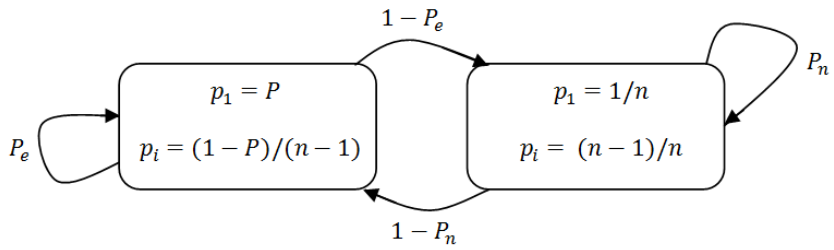


Figura 5.21 Modelo oculto de Markov propuesto para la identificación de notas musicales. En la parte izquierda se observa el estado de emisión y en la derecha el estado de no emisión.

En la Figura 5.21 n representa el número de símbolos (notas) que es posible observar, se generan tantos modelos como notas a identificar, en el presente trabajo 48 notas, p_1 indica la probabilidad de emisión de la nota a la que corresponde el modelo, p_i especifica las probabilidades de emitir el resto de las notas dentro del conjunto de símbolos con $i \in \{2, \dots, n\}$ y las constantes P, P_e y P_n representan la probabilidad de observar la nota a la que corresponde al modelo, la probabilidad de permanecer en el estado de emisión y la probabilidad de permanecer en el estado de no emisión respectivamente.

Con lo anterior se generan tantos modelos como notas a reconocer y dada una secuencia de símbolos se calcula la probabilidad de que ésta sea generada por alguno de los modelos usando el algoritmo $forward$. De esta manera el modelo que arroje una mayor probabilidad indicará la nota que se está emitiendo. Cabe destacar que antes de aplicar esta aproximación es necesario segmentar la señal de modo que la secuencia entregada a los HMM corresponda a una única emisión sonora (una única nota o acorde), pues como se mencionó anteriormente cada modelo identifica un único tipo de señal, por ejemplo, para lograr una mejor identificación de una nota $Do4$ es importante asegurarse que la secuencia de símbolos presentada sea únicamente de esta nota. Evidentemente no se conoce en un principio qué segmentos corresponden a una nota y qué segmentos corresponden a otra, pero se puede realizar una segmentación para separar distintas notas sin saber cuáles son éstas por medio de la energía de la señal, es decir, se buscará los lugares en donde la señal presente un aumento súbito de energía, con esto es posible segmentar la señal de modo que la secuencia de símbolos corresponda a una única emisión sonora. Una vez segmentada la señal se aplica el sistema de redes neuronales descrito en el Capítulo 4 a cada segmento y se presenta la secuencia de símbolos obtenida al algoritmo $fordward$ para su clasificación. La Figura 5.22 muestra el diagrama general de lo descrito aplicado a la secuencia de notas $Do4, Mi3$ y $\underline{Sol3}$.

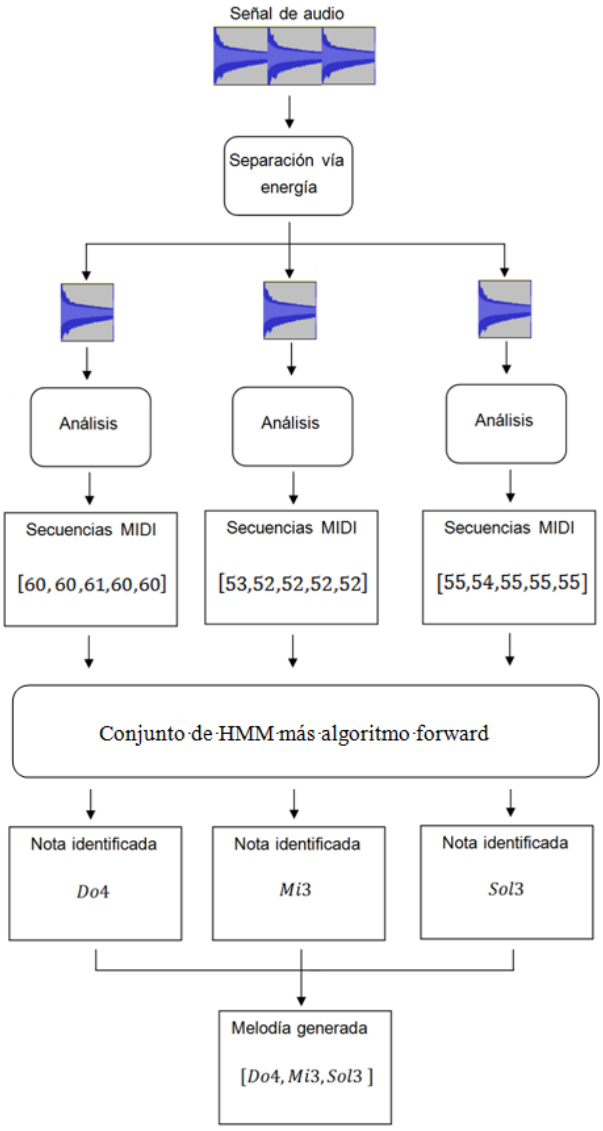


Figura 5.22 Diagrama del uso de HMM para el reconocimiento de notas musicales.

En la Figura 5.22 el bloque "Separación vía energía" hace referencia a la segmentación de la señal por medio de los cambios súbitos de energía de modo que cada segmento contenga una única emisión sonora, los bloques "Análisis" hacen referencia al procesamiento de la señal con el sistema de redes neuronales descrito en el Capítulo 4 . Por otro lado el bloque "Conjunto de HMM más algoritmo forward" hace referencia al uso del algoritmo $forward$ para la determinación del HMM que con mayor probabilidad genera la secuencia dada.

Para comenzar con las pruebas de lo descrito anteriormente se generaron 12 modelos que se corresponden con las notas de la escala cromática en la cuarta octava, esto es, las notas con código MIDI en el conjunto $\{60, 61, \dots, 71\}$ con lo que la variable n toma el valor de 12. Se propusieron los valores $P = 0.80$ y $P_e = P_n = 0.70$, con esto las probabilidades restantes, dentro del estado de emisión, son $p_2 = p_3 = \cdots = p_{12} = 0.0727$ en el caso de primer modelo, que reconocerá la primera nota, es decir, Do. Para el segundo modelo, en principio, basta con intercambiar las probabilidades del estado de emisión a modo que ahora sea el segundo

símbolo el que sea más probable observar, es decir, hacer $p_2 = 0.80$ y $p_1 = 0.0727$, mientras el resto de valores se conserva.

Con lo anterior se le presentó al programa una secuencia de caracteres, misma que puede ser generada por el sistema de redes neuronales antes descrito, y se calculó la probabilidad de que ésta fuera generada por cada uno de los modelos ocultos propuestos. La Tabla 5.4 muestra los resultados obtenidos para la secuencia [60 60 62 60 63 60 60 60] generada a partir del análisis de una nota $Do4$.

Tabla 5.4 Resultados obtenidos aplicando el algoritmo forward de la secuencia dada a los doce modelos propuestos.

Modelo	Nota a la que corresponde.	Código MIDI.	Probabilidad obtenida.
1	Do 4	60	7.80e-05
2	$Do\#$ 4	61	1.97e-12
3	Re 4	62	3.06e-11
4	$Re\#$ 4	63	3.06e-11
5	Mi 4	64	3.06e-11
6	Fa 4	65	1.97e-12
7	$Fa\#$ 4	66	1.97e-12
8	Sol 4	67	1.97e-12
9	$Sol\#$ 4	68	1.97e-12
10	La 4	69	1.97e-12
11	$La\#$ 4	70	1.97e-12
12	Si 4	71	1.97e-12

Como puede observarse la probabilidad de que el primer modelo generara la secuencia es mucho mayor que el resto de las probabilidades, por lo cual la secuencia se identifica como una nota Do 4.

Ahora se tomará una secuencia de diez símbolos generada a partir del análisis de una nota Fa 4, correspondiente al código MIDI 65, ésta es, [68 65 65 70 65 65 65 68 70 65] obteniéndose los resultados de la Tabla 5.5.

Tabla 5.5 Resultados obtenidos aplicando el algoritmo forward de la secuencia dada a los doce modelos propuestos.

Modelo	Nota a la que corresponde	Código MIDI	Probabilidad obtenida
1	Do 4	60	2.81e-15
2	$Do\#$ 4	61	2.81e-15
3	Re 4	62	2.81e-15
4	$Re\#$ 4	63	2.81e-15
5	Mi 4	64	2.81e-15
6	Fa 4	65	3.91e-08
7	$Fa\#$ 4	66	2.81e-15
8	Sol 4	67	2.81e-15
9	$Sol\#$ 4	68	1.92e-12
10	La 4	69	2.81e-15
11	$La\#$ 4	70	6.76e-13
12	Si 4	71	2.81e-15

Esta vez la mayor probabilidad se observa en el sexto modelo, que precisamente corresponde a la nota Fa 4.

Como puede observarse esta aproximación ayuda al reconocimiento de notas aisladas proponiendo un modelo para cada una, que en principio puede extenderse tanto como notas o acordes sean usados para el entrenamiento de las redes neuronales usadas para la generación de las secuencias MIDI.

Cabe destacar que otros métodos para la clasificación de secuencias de símbolos segmentadas por medio de energía pueden ser utilizados, por ejemplo el método de votación, tomando en cuentas que se trabaje con una señal similar a la del piano en el sentido de que cada emisión sonora pueda separarse de las demás haciendo un análisis de la energía. Éste análisis requiere un procesamiento extra de la señal, el cual haciendo uso de un enfoque distinto podría evitarse como se presentará en la siguiente sección.

5.6.2 Segunda aproximación en el uso de HMM

En este enfoque, a diferencia del anterior, se propone un único modelo de Markov para la identificación de notas musicales en donde los símbolos observados se corresponden con los códigos MIDI de las notas musicales y cada estado oculto es equivalente a una nota, es decir, la cadena podrá identificar tantas notas como estados ocultos posea. Esta vez no se utilizará el algoritmo $forward$ para la identificación de las notas sino el algoritmo de $Viterbi$, pues dada una secuencia de caracteres se quiere saber qué estado oculto la produjo con mayor probabilidad.

Como ejemplo se tomará un HMM que reconocerá tres notas, por lo que serán necesarios tres estados. Se supondrá que los símbolos observables corresponden únicamente a las notas a reconocer, por lo cual se tendrán tres símbolos por cada estado. Lo anterior se muestra en la Figura 5.23.

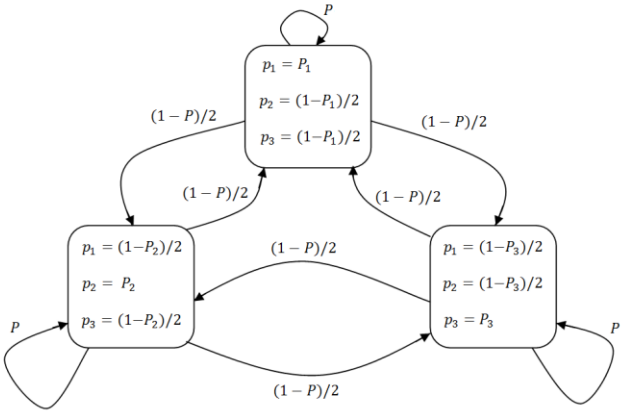


Figura 5.23 HMM propuesto para el reconocimiento de melodías de tres notas.

Como puede observarse a cada estado se le asignó una probabilidad de emisión de la nota a la que corresponde, en principio puede hacerse $P_1 = P_2 = P_3$ si es que no se cuenta con mayor información acerca de la pieza a analizar. Así mismo puede verse que las probabilidades de permanecer en un estado son, en todos los casos, iguales a P y que existen las probabilidades de cambiar a otro estado son iguales, éstas son $(1 - P)/2$.

Para simplificar el modelo anterior se puede comenzar aplicando las mismas probabilidades de emisión correspondientes a cada nota como se puntualizó en el párrafo anterior, así mismo supóngase que el primer estado reconocerá una nota con símbolo 1 y que se tiene la siguiente secuencia de símbolos $[1, 1, 1, 2, 1, 1, 1, 1, 1]$, que en principio indicaría la presencia de una única nota, es decir, puede entenderse la aparición del símbolo 2 como un ruido dentro de la secuencia, por lo que en este tipo de casos no es conveniente un cambio de estado en el HMM, sino la permanencia en el estado de la nota a reconocer. Tomando en cuenta la observación anterior es claro que la probabilidad de emitir un "ruido", dentro de un estado oculto dado, debe ser igual o mayor que la probabilidad de cambiar de estado, pues de otra manera, al observarse un símbolo como el 2 en la secuencia anterior, el algoritmo de $Viterbi$ concluiría que el estado

que emitió ese símbolo es el correspondiente a la segunda nota, en lugar de identificar que es el estado correspondiente a la primera.

Con lo antes descrito se llega a un HMM simplificado como el que se muestra en la Figura 5.24, en donde se hicieron $P = P_1 = P_2 = P_3$.

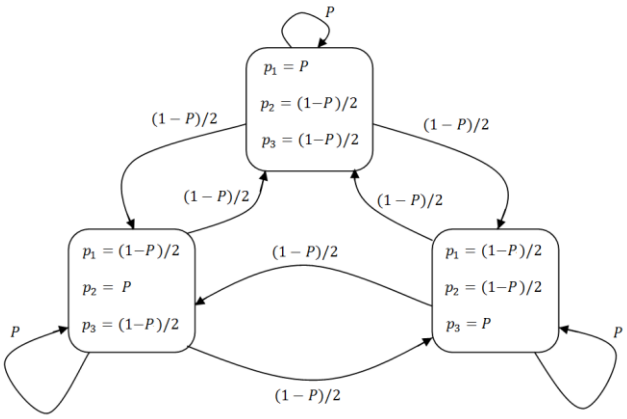


Figura 5.24 HMM simplificado para el reconocimiento de secuencias de tres notas.

Dado el modelo anterior, éste puede generalizarse para el reconocimiento de una pieza musical compuesta de n notas (acordes y bicordes) generando un HMM de n estados, en donde a cada estado se le asignará una nota. La probabilidad de emisión de la nota de cada estado será P y la del resto de los símbolos será $(1 - P)/(n - 1)$, mientras que la probabilidad de permanecer en un estado oculto es también P. Es claro que debe cumplirse la desigualdad $P > (1 - P)/(n - 1)$ con lo que también se observa que al presentarse un ruido dentro de una secuencia de caracteres es más probable permanecer en el estado correcto que cambiar de estado de emisión, lo cual se traduciría en una identificación errónea.

Con lo anterior puede observarse que de hecho, en esta aproximación, las matrices de transición y emisión son simétricas de orden n e iguales, con una forma general dada por:

$$\begin{bmatrix} P & (1-P)/(n-1) & \cdots & (1-P)/(n-1) \\ (1-P)/(n-1) & P & \cdots & (1-P)/(n-1) \\ \vdots & \vdots & \ddots & \vdots \\ (1-P)/(n-1) & (1-P)/(n-1) & \cdots & P \end{bmatrix} \tag{5.3}$$

La Figura 5.25 ejemplifica el uso del uso de esta aproximación, partiendo de la señal de audio, hasta la identificación de la melodía.

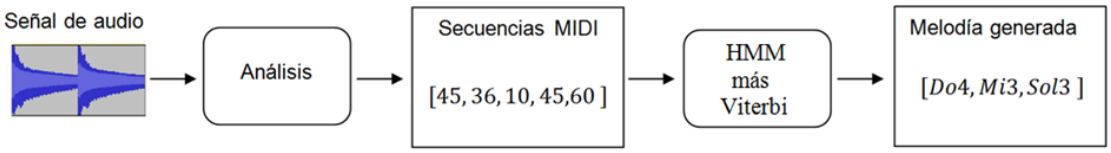


Figura 5.25 Uso del HMM propuesto para la identificación de melodías.

En la Figura 5.25 el bloque "Análisis" hace referencia al procesamiento de la señal haciendo uso del sistema de redes neuronales descrito en el Capítulo 4 "Propuesta de solución.". Por otro lado el bloque "HMM

más Viterbi" hace referencia al uso del algoritmo de $Viterbi$ para determinar los estados que con mayor probabilidad generan la secuencia de símbolos dada.

Para ejemplificar el uso de este método se tomó la señal se audio de la escala de $Do\ mayor$ y se generaron las secuencias MIDI de la misma. Para determinar el número de notas presentes en la secuencia, simplemente se restó el número máximo de la secuencia menos el mínimo de la misma, de modo que con esto se sabe el número de estados y símbolos que tendrá el HMM. Se fijó una probabilidad de emisión (o de permanencia en un estado) $P = 0.5$, con lo que las matrices de transición y emisión fueron iguales a:

$$\begin{bmatrix} 0.5 & 0.0385 & \cdots & 0.0385 \\ 0.0385 & 0.5 & \cdots & 0.0385 \\ \vdots & \vdots & \ddots & \vdots \\ 0.0385 & 0.0385 & \cdots & 0.5 \end{bmatrix}$$

La Figura 5.26 muestra el resultado obtenido al aplicar el algoritmo de $Viterbi$ a la secuencia de códigos MIDI de la escala de Do mayor con el HMM propuesto.

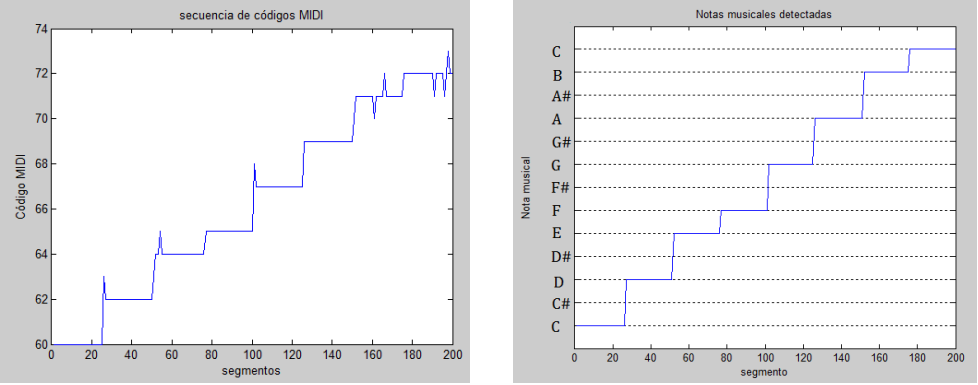


Figura 5.26 Uso de la segunda aproximación de los HMM. A la izquierda se muestra la secuencia de códigos MIDI. A la derecha se muestran los estados (notas musicales) entregados por el algoritmo de Viterbi.

En la Figura 5.26 puede observarse la identificación de las notas que componen la escala mayor de Do.

En este punto cabe destacar la ventaja de este segundo enfoque en el uso de HMM con respecto al anterior, pues no es necesario llevar a cabo una segmentación de la señal por medio de el análisis de la energía, esto permite evitar un procesamiento adicional en el sistema. Por otro lado al no identificarse segmentos específicos de cada emisión sonora métodos como el de votación ya no son aplicables.

5.6.3 Reconocimiento de vibrato

Para el reconocimiento de notas a las que se les aplica un vibrato se utilizaron HMM. Para esta aproximación se tomó en cuenta el patrón general que sigue este tipo de efecto, esto es, considerar que la frecuencia de la nota emitida varía con el tiempo en un intervalo de más menos dos semitonos. La Figura 5.27 muestra una secuencia de notas obtenida al analizar una nota Do 4, con código MIDI igual a 60, con un efecto de vibrato aplicado a través del mensaje de cambio de control de modulación (véase la Figura 3.29).

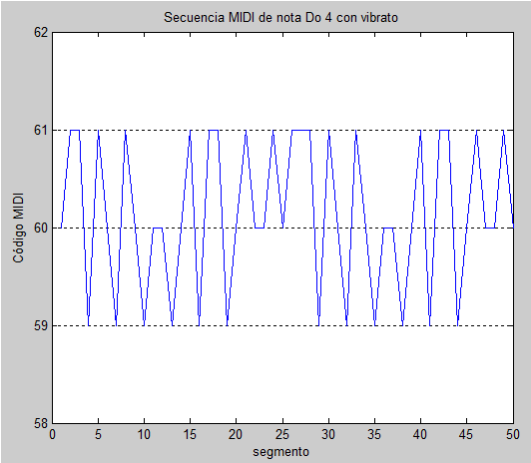


Figura 5.27 Secuencia MIDI obtenida al analizar una señal de audio de la nota Do 4 al aplicársele un vibrato.

Con base en lo anterior se propuso un HMM de dos estados y se siguió un procedimiento similar al presentado en la "*Primera aproximación en el uso de HMM*". Los estados ocultos pueden nombrarse como "emisión" y "no emisión", significando, como su nombre lo indica, la emisión de una nota con vibrato y la emisión de un ruido respectivamente. Para definir las probabilidades de emisión de un símbolo (código MIDI de una nota) se tomó en cuenta el comportamiento que presenta un vibrato, como se mostró en la Figura 5.27, es decir, si se emite una nota N y a esta se le aplica un vibrato existe una alta probabilidad de observar los símbolos de las notas $N-2$, $N-1$, $N+1$ y $N+2$ que representan las dos notas consecutivas anteriores y las dos notas consecutivas posteriores respectivamente o utilizando un lenguaje de la música dos semitonos por debajo y dos tonos por arriba de la nota N. También existe la posibilidad, aunque en menor medida, de observar los símbolos de las notas $N-3$ y $N+3$, que son las notas que se encuentran a una distancia de tres semitonos de la nota N. Con esto se proponen las probabilidades de emisión para el estado de emisión de vibrato como se muestran en la Tabla 5.6.

Tabla 5.6 Probabilidades de emisión propuestas para el estado oculto de emisión de vibrato para la nota N.

Símbolo	Probabilidad de emisión
$N-3$	$1/12$
$N-2$	$2/12$
$N-1$	$2/12$
N	$2/12$
$N+1$	$2/12$
$N+2$	$2/12$
$N+3$	$1/12$
Resto de símbolos	0

Como se mencionó en el marco teórico el vibrato no suele ir más allá de un intervalo de dos semitonos por arriba y dos semitonos por debajo de la nota a la que se aplica el efecto, por lo que en principio la probabilidad de emisión de las notas más allá del intervalo $[N-2, N+2]$ es pequeña o bien cero.

La Figura 5.28 muestra gráficamente la densidad de probabilidades descrita anteriormente para un HMM que puede emitir los símbolos de las notas MIDI dentro de la escala cromática de $Do4$, es decir los códigos del 60 al 71, y cuyo estado oculto de emisión busca identificar la emisión de un vibrato en la nota $Sol4$ (con código MIDI igual a 67).

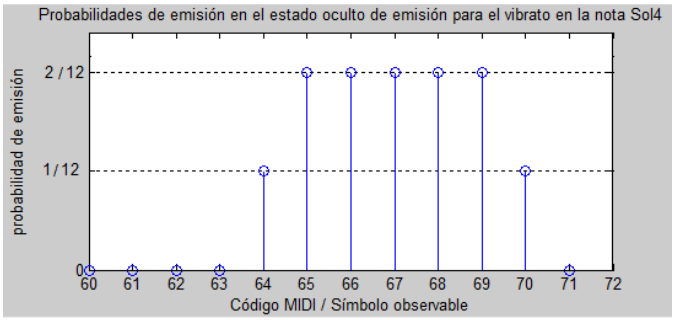


Figura 5.28 Densidad de probabilidades para el estado de emisión en HMM que identifica la emisión de vibrato sobre la nota Sol 4.

Por otro lado el estado oculto de no emisión tomará en cuenta que existe la posibilidad de observar símbolos de notas que en principio no tienen nada que ver con la emisión del vibrato, los cuales pueden ser considerados como ruido. Las probabilidades de emisión de cualquier nota en este estado se consideró es igual, por lo que las probabilidades están dadas por $1/n$, donde n representa el número total de símbolos observables. La Figura 5.29 muestra la densidad de probabilidades para el estado de no emisión para el HMM del ejemplo anterior.

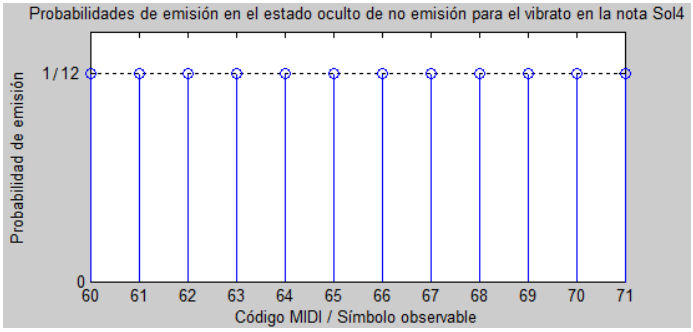


Figura 5.29 Densidad de probabilidades para el estado de no emisión en HMM que identifica la emisión de vibrato sobre la nota Sol 4.

Como ya se mencionó en este enfoque se generó un HMM por cada vibrato, para posteriormente utilizar el algoritmo $forward$ para identificar el modelo que mejor se ajusta. Usando esta idea se generaron 36 señales de audio, cada una con una única nota a la que se le aplica vibrato, yendo desde la nota $Do3$ con código MIDI igual a 48, hasta la nota Si5, con código igual a 83. Con esto se generaron 36 HMM cada uno de dos estados, con probabilidades de emisión para el estado de emisión siguiendo lo descrito en la Tabla 5.5 y probabilidades de emisión en el estado de no emisión iguales a $1/36$. Para probar qué tan buena es esta aproximación se aplicó el algoritmo $forward$ a cada HMM usando una serie de diez símbolos correspondiente a el vibrato que identifica, de modo que se observe la probabilidad de que cada HMM genere la secuencia para la que fue generado. Los resultados se muestran en la Figura 5.30.

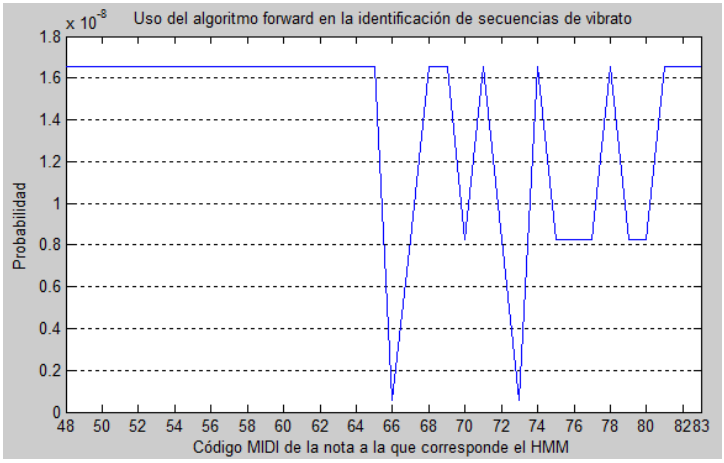


Figura 5.30 Aplicación del algoritmo forward a secuencias de vibratos de diez símbolos a sus correspondientes HMM.

Como puede observarse la aproximación propuesta genera probabilidades entre 5.5×10^{-10} y 1.6×10^{-8}. Por otro lado si se toma una serie, por ejemplo $Re\#3$, con código MIDI 51, y se le aplica el algoritmo $forward$ con cada uno de los HMM propuestos se obtiene el resultado mostrado en la Figura 5.31.

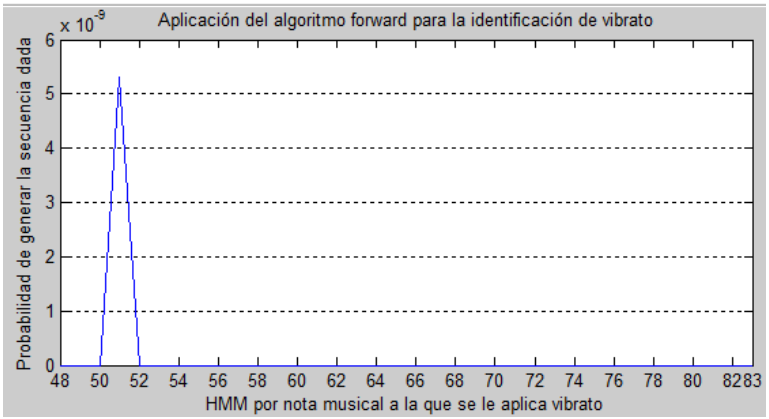


Figura 5.31 Identificación de vibrato haciendo uso del algoritmo forward, aplicando el mismo a cada HMM propuesto.

Como puede observarse se identifica el vibrato de la nota con código MIDI igual a 51, es decir $Re\#3$, como era de esperarse, obteniéndose una probabilidad de 5.3×10^{-9} contra probabilidades de cero.

5.7 Comparación con otros trabajos del estado del arte

En la Tabla 5.7 se muestran resultados de otros trabajos relacionados con la presente investigación, mostrando las similitudes y diferencias de los mismos.

Tabla 5.7 Comparación de trabajos representativos vs el presente trabajo.

Trabajo	Instrumento analizado	Número de emisiones sonoras identificadas	Porcentaje máximo de emisiones correctamente clasificadas	Técnicas y métodos utilizados	Comentarios
Barbancho [Bar12]	Guitarra	330	87%	Cromagrama, HMM	En el análisis de Barbancho no se analiza la octava a la que pertenecen los acordes, pues hablando de la guitarra esto carece de importancia.
Böck [Böc11]	Piano	88	88.9%	transformada de Fourier, Redes Neuronales.	El sistema de Böck reconoce la activación de notas musicales, considerando una correcta identificación aún si se falla en identificar la octava.
Mauch [Mau10]	Piano	109	71%	Cromagrama, Redes Bayesianas Dinámicas (DBNs)	En este trabajo Mauch separa el espectro en dos subconjuntos representando cada uno las notas altas y bajas respectivamente. En su búsqueda limita la emisión de acordes solo al subconjunto de notas altas, limitando el mismo a unas dos octavas.
Papadopoulos [Pap07]	Piano	24	70.9%	Cromagrama, HMM.	El trabajo de Papadopoulos no especifica la octava a la que pertenecen los acordes analizados (12 acordes mayores y 12 acordes menores).
Presente trabajo	Piano	144	87%	MFCCs, Redes Neuronales, HMM.	En el presente trabajo se realiza la clasificación de emisiones sonoras, éstas pueden ser notas musicales (48 en total), acordes mayores (48 en total) y acordes menores (48 en total), mismas que pueden ser emitidas en un rango de cuatro octavas. También se pone énfasis en que se considera una clasificación correcta únicamente si se identifica correctamente la octava a la que pertenecen estas emisiones.

6 Conclusiones, trabajos futuros y aportaciones científicas.

6.1 Conclusión del objetivo general. Aplicar métodos de procesamiento digital de voz a señales de audio generadas por un instrumento musical, adaptando y extendiendo los mismos para la extracción de características de interés en el lenguaje musical

En el presente trabajo se mostró el uso de distintas técnicas utilizadas en el procesamiento digital de voz aplicadas en el análisis de señales generadas por un instrumento musical, abarcando la identificación de emisiones sonoras como notas musicales, acordes y vibratos. Así mismo se presentó un algoritmo diseñado para la generación de instrucciones MIDI coherentes, mismas que permiten la representación gráfica aproximada en pentagrama de la información extraída de la señal de audio, acompañado de una librería de funciones. Por otro lado también se exponen diversos métodos para la eliminación de ruido y corrección de errores en la clasificación de emisiones sonoras haciendo uso de HMM a modo de aumentar el número de instrucciones MIDI correctamente clasificadas.

6.1.1 Conclusión del objetivo específico 1. Identificación de notas musicales vía F0.

Para la primera clasificación se utilizó la estimación de F0 haciendo uso de la autocorrelación, misma que sirvió para identificar las notas musicales pertenecientes a cinco octavas, lo que representa un amplio rango de frecuencias, además de esto la autocorrelación es un método que es robusto ante el cambio de timbre del instrumento analizado, aunque presenta errores en la correcta identificación en la octava a la que pertenece la nota en un 5% de los casos estudiados. Así mismo es importante puntualizar que el uso de la autocorrelación se restringe únicamente a identificación de notas musicales emitidas individualmente, pues al presentar acordes o bicordes este método falla.

6.1.2 Conclusión del objetivo específico 2. Identificación de acordes usando MFCCs y redes neuronales.

Para el caso de identificación de emisiones sonoras en las que se presenta más de una nota a la vez se propuso el uso de los MFCCs como vector de características, mismo que fue usado para la clasificaciones de 48 acordes mayores, 48 acordes menores y 48 notas musicales que en total suman 144 emisiones sonoras distintas. El uso de MFCC permite clasificar estas emisiones brindando información sobre a qué octava pertenecen, lo que representa una aportación importante con respecto a otros trabajos, como el presentado por Papadopoulos [Pap07], el cual se limita a un conjunto de 24 acordes en donde la octava no se analiza. De igual manera se propuso un sistema de redes neuronales a modo de árbol de decisiones que permite hacer uso de los vectores MFCCs como entrada y que permite, no solo la clasificación de las emisiones sonoras, sino que también sirve como decodificador de códigos MIDI. Es importante recalcar que el uso de varias redes neuronales en serie permite aumentar el conjunto de emisiones sonoras a clasificar sin presentarse una disminución drástica en el porcentaje de clasificaciones correctas.

6.1.3 Conclusión del objetivo específico 3. Uso de HMM para la corrección de secuencias de emisiones sonoras y reconocimiento de vibratos.

Una vez que se desarrolló todo el sistema de generación de secuencias MIDI a partir de señales de audio se propusieron dos enfoques en el uso de Modelos Ocultos de Markov para el filtrado de las mismas, estos puede

entenderse como correctores de secuencias MIDI, mismos que sirvieron para corregir ruidos dentro de las mismas. El primer enfoque toma un modelos similar al propuesto por Barbancho [Bar12], solo que a diferencia de éste, que fue utilizado para la clasificación de pisadas en la guitarra, el HMM propuesto en el presente trabajo se utilizó, como ya se mencionó para el filtrado de secuencias MIDI. Como segunda aproximación se presentó una aplicación innovadora en el uso de HMM, en donde se propuso el uso del algoritmo $forward$ aplicado a un conjunto de Modelos Ocultos de Markov, en donde cada uno de estos corresponde a una emisión sonora además de la identificación de secuencias sonoras generadas al aplicar vibratos a las notas musicales. A continuación se describen más detalladamente estos modelos.

La primera aproximación propone generar un HMM de dos estados ocultos por cada emisión sonora a reconocer. Haciendo uso del algoritmo $forward$ se busca de entre los modelos propuestos cuál de ellos genera la secuencia dada con mayor probabilidad. De esta manera se obtiene lo que se podría entender como un clasificador de secuencias MIDI, mismo que representa una aportación innovadora. Haciendo uso de este enfoque fue posible realizar el reconocimientos de vibratos aplicados a notas musicales. Es importante recalcar que el uso de este método requiere un procesamiento previo de las secuencias MIDI, pues la secuencia obtenida de una señal de audio debe segmentarse en secuencias más pequeñas que correspondan a emisiones sonoras distintas, por lo cual, valiéndose de la manera en que el piano emite sonidos distintos, este cometido se logra analizando los intervalos en que la energía de la señal presenta un cambios de un valor menor a uno mayor.

Por otro lado la segunda aproximación hace uso de un único HMM que puede entenderse como un filtro de secuencias MIDI, en donde cada estado oculto se corresponde con un acorde o nota musical a identificar y como símbolos observables se presentan los códigos MIDI provenientes del sistema de redes neuronales. Haciendo uso del modelo propuesto y del algoritmo de $Viterbi$ se logra el cometido de reducir las clasificaciones incorrectas hechas por el sistema de redes neuronales. Con esta aproximación se logró aumentar un 4% la correcta clasificación de emisiones sonoras. Además de esto se tiene la ventaja de que la secuencia de códigos MIDI puede presentarse directamente al HMM sin necesidad de realizar algún procedimiento adicional. Con esto se presenta un uso innovador en el uso de HMM, que en otros trabajos del estado del arte suelen ser utilizados como clasificadores.

6.1.4 Conclusión del objetivo específico 4. Uso del análisis de energía de la señal para la estimación de la duración de las emisiones sonoras en una pista de audio a modo de obtener una representación en figuras propias del lenguaje musical.

Acompañada de la clasificación por redes neuronales se realizó el análisis del comportamiento esencial de la energía en una señal de audio proveniente de un instrumento como el piano, lo que permitió la propuesta de un algoritmo que permite realizar la decodificación en secuencias MIDI con la cual fue posible realizar una representación gráfica aproximada de la señal de entrada, esto es, el pentagrama.

6.1.5 Conclusión del objetivo específico 5. Desarrollo de una librería de funciones que permitan la generación de archivos MIDI.

Al ser el formato MIDI parte importante de este trabajo se desarrolló una librería de funciones en MATLAB, que permite la generación de archivos .mid de manera más rápida. Esto representa una aportación en el desarrollo de código, pues actualmente no existe un conjunto de funciones similar para este lenguaje de programación.

De esta manera haciendo uso de las herramientas antes descritas fue posible llegar a una representación gráfica aproximada para señales de audio compuestas por 144 emisiones sonoras emitidas por un piano.

6.2 Posibles aplicaciones inmediatas del presente trabajo

De la misma manera pueden mencionarse las aplicaciones prácticas inmediatas derivadas del desarrollo del presente trabajo, éstas podrían ser:

1. Uso del análisis de F0 para el entrenamiento de cantantes, particularmente los estudiantes de solfeo, quienes podrían observar la precisión con la que emiten cierta nota musical.
2. Uso del sistema como apoyo para la generación de partituras de manera rápida, evitando así una transcripción manual y facilitando la generación de un archivo digital.
3. La librería de funciones desarrolladas para la generación de secuencias MIDI podría ser de interés para otros desarrolladores e investigadores, con lo que sería posible compartirla en plataformas como mathworks, dedicadas a la publicación de código.

6.3 Trabajos a futuro

Con el objetivo de llegar a un sistema más flexible y robusto pueden mencionarse los siguientes puntos a desarrollar:

1. Realizar los entrenamientos de las redes neuronales haciendo uso de MFCCs extraídos de fuentes sonora distintas, a modo de hacer más robusto el sistema ante ruido y sonidos con timbres diversos.
2. Incluir el cromagrama como vector de características para la clasificación de acordes de modo que sirva como complemento a los vectores MFCCs.
3. Aumentar el conjunto de emisiones sonoras a clasificar, incluyendo otros tipos de acordes básicos, como acordes de disminuidos o acordes compuestos por más de tres notas.
4. Explorar arquitecturas distintas para el árbol de decisiones propuesto en el presente trabajo y analizar cuál de ellas permite la mejor clasificación.
5. Proponer reglas para generalizar el algoritmo para la generación de secuencias MIDI de forma que sea posible incluir sonidos provenientes de instrumentos con diversos timbres, como instrumentos de viento o cuerda rasgueada.
6. Realizar un análisis más detallado en los valores de las probabilidades para los Modelos Ocultos de Markov, tratando de tomar en cuenta la dificultad física de ejecutar un acorde después de otro dado, de modo similar al trabajo propuesto por Barbancho [Bar12].

6.4 Aportaciones científicas

En el presente trabajo se realizaron aportaciones científicas a los trabajos del estado del arte, entre las que cabe mencionar:

1. Propuesta de un sistema novedoso de clasificación de emisiones sonoras compuesto de redes neuronales organizadas a modo de árbol de decisiones que hace uso de los MFCCs como vector de características.
2. Propuesta de HMM de dos estados para la identificación de secuencias de símbolos correspondientes a emisiones sonoras como notas musicales, acordes mayores y menores y vibratos haciendo uso del algoritmo $forward$.
3. Propuesta en el uso de un HMM similar al usado por Barbancho [Bar12] para el filtrado de secuencias MIDI. A diferencia del trabajo mencionado se usa el piano como fuente sonora y se usan como símbolos observables los códigos MIDI de las emisiones sonoras.

7 Referencias

[Anv14] Anvil Studio. Versión 2014.09.08. Willow software. Disponible en: http://www.anvilstudio.com/

[Bar12] A. Barbancho, A. Klapuri, L. Tardón, I. Barbancho. Automatic Transcription of Guitar Chords and Fingering From Audio. IEEE transactions on audio, speech, and lenguage processing, VOL. 20, No. 3, 2012.

[Ben12] E. Benetos. Automatic Transcription of Polyphonic Music Exploiting Temporal Evolution. PhD thesis, School of Electronic Engineering and Computer Science Queen Mary University of London, 2012.

[Ber50] Bernal Jiménez, Miguel (1950). La técnica de los compositores, el estilo melódico armónico. Editorial Jus. México, D.F.

[Böc11] S. Böck, M. Schedl, Polyphonic piano note transcription with recurrente neural networks. Departament of computational perception, Johannes Kepler University, Linz, Austria, 2011.

[Cou12] D. Courtney. North indian musical notation - an overview. Última actualización: 04-Feb-2012. [Consultado: 16-02-2014]. Disponible en: http://chandrakantha.com/articles/indian_music/lippi.html

[Cre10] Armonía. Crea música. 2010. [Consultado: 06-05-15]. Disponible en: http://www.creamusica.com.ar/clase_de_organo4.htm

[Dem02] Demuth, H., Beale, M. (2002). Neural Network Toolbox, For use with MATLAB.

[Esl03] Eslava, Hilarion (2003). Método completo de solfeo sin acompañamiento, Segunda parte. EDITAPSOL. México, D.F.

[Gui15] GuitarPro. Arobas Music. Disponible en: http://www.guitar-pro.com/en/index.php

[Har97] W. M. Hartmann. Signals, sound and sensation. Springer. pp. 283. 1997.

[Has13] Hass, Jeffrey. Introduction to computer music: Volume One. 2013. [Consultado: 20-12-2014].

[Haw93] J. Hawley. Structure out of Sound. Massachusetts Institute of Technology, 1993.

[Int13] IntelliScore Esemble. MP3 to MIDI converter. 2013. Disponible en: http://www.intelliscore.net/order.html

[IRT14] Artificial Neural Networks (2014). RWTH AACHEN Univesity. Disponible en: http://www.irt.rwth-aachen.de/en/research/methods/identifikation-dynamischer-systeme/knn/

[Jac58] Pahissa, Jaime (1956). Enciclopedia práctica Jackson. W. M. Jackson, Inc., Editores. Tomo 13. México, D.F.

[Kas93] K. Kashino, H. Tanaka. A sound source separation system with the ability of automatic tone modeling. En Proceedings of the International Computer Music Conference, 1993.

[Kas95] K. Kashino, K. Nakadai, T. Kinoshita, H. Tanaka. Application of Bayesian probability network to music scene analysis. Proceedings of the International Joint Conference on AI, CASA workshop, 1995.

[Kla04] A. Klapuri. Signal Processing Methods for the Automatic Transcription of Music. PhD thesis, Tampere University of Technology, 2004.

[Kla06b] A. Klapuri. Multiple fundamental Frequency estimation by summing harmonic amplitudes. Institute of signal processing, Tampere University if Technology, Finland, 2006.

[Lab08] LabROSA. Polimer, G. 2008. Disponible en: http://labrosa.ee.columbia.edu/projects/piano/

[Li10] T. Li, A. Chan, A. Chun. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network. Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I, IMECS 2010, March 17 - 19, Hong Kong. 2010.

[Log00] Logan, Beth (2000). Mel Frequency Cepstral Coefficients for Music Modeling.

[Man11] D. Manolakis, V. Ingle. Applied digital signal processing. Cambridge University Press. 2011.

[Mau10] M. Mauch, S. Dixon. Simultaneous Estimation of Chords and Musical Context From Audio. IEEE Transactions on audio, speech and lenguage processing, Vol. 18. 2010.

[Mid09] MIDI Manufacters Association (2009). An Introduction to MIDI. Roland Corporation U.S. Disponible en: http://midi.org/aboutmidi/intromidi.pdf

[Mid09a] MIDI Manufacters Association (2009). MIDI messages. Roland Corporation U.S. http://www.midi.org/techspecs/midimessages.php

[Mid09b] MIDI Manufacters Association (2009). The complete MIDI 3.0 detailed specification. Roland Corporation U.S. Disponible en: http://www.midi.org/techspecs/midispec.php

[MIRa13] The Music Information Retrieval Evaluation eXchange (MIREX). University of Illinois. 2013. Disponible en: http://www.music-ir.org/mirex/wiki/2013:Audio_Chord_Estimation_Results_Billboard_2013

[MIRb13] The Music Information Retrieval Evaluation eXchange (MIREX). University of Illinois. 2013. Disponible en: http://nema.lis.illinois.edu/nema_out/mirex2013/results/akd/summary.html

[MIRc13] The Music Information Retrieval Evaluation eXchange (MIREX). University of Illinois. 2013. Disponible en: http://www.music-ir.org/mirex/wiki/2013:Real-time_Audio_to_Score_Alignment_(a.k.a._Score_Following)_Results
[Mon10] Montoya, Juan. Frecuencias de las notas y escalas musicales. 2010. [Consultado: 15-12-2014]. Disponible en: http://juan960.blogspot.mx/2010/04/frecuencias-de-las-notas-y-escalas.html
[Moo75] J.A. Moorer. On the segmentation and Analysis of Continuous Musical Sound by Digital Computer. PhD thesis, Stanford University, 1975.
[Nuñ07] Nuño, Luis. Notas Musicales. 2007. [Consultado: 15-12-2014]. Disponible en: http://www.ruedaarmonica.com/5.php
[Ola54] de Olazabal, Tirzo (1954). Acústica Musical y Organología. 8a ed. Ricordi Americana S.A.E.C.
[Pap07] H. Papadopoulus, G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. Workshop Content-Based Multimedia Indexing. 2007.
[Per03] A. Pertusa. Transcripción de melodías polifónicas mediante redes neuronales dinámicas. Memoria de suficiencia investigadora, Universidad de Alicante, 2003.
[Pro07].J. Proakis, D. Manolakis. Digital Signal Processin, 4th edition, Pearson, 2007.
[Rab99] Rabiner, Lawrence (1999). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77:257-286.
[Res04]. Resnick, Halliday, Krane (2004). Física, Volumen 3. 5a ed. México, D.F. Grupo Patria Cultural.
[Rod14] Rodríguez Alvira, Armaduras de clave. 2014. [Consultado: 26-03-2015]. Disponible en: http://www.teoria.com/es/referencia/a/armaduras.php
[Smi04] Smith, Noah A (2004). Hidden Markov Models: All the Glorius Gory Details. Johns Hopkins University. Disponible en: http://www.cs.cmu.edu/~nasmith/papers/smith.tut04a.pdf
Disponible en: http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI.shtml