

Construcción de un afinador para guitarra. Aplicación de la Transformada Rápida de Fourier

Juan Pablo Sánchez

Docente: Angel Augusto Agudelo Zapata

Universidad Tecnológica de Pereira

30 de mayo de 2024

Índice

Introducción	3
1. Fundamento teórico	4
1.1. Frecuencia y tono	4
1.2. Afinación de la guitarra	5
1.3. Procesamiento de señales digitales	6
1.4. Tecnologías usadas	7
2. Diseño del afinador	8
2.1. Estructura general del programa	8
3. Implementación	9
3.1. Importaciones	9
3.2. Captura de audio	9
3.3. Frecuencias de referencia y tolerancia	9
3.4. Inicialización de PyAudio	9
3.5. Configuración de la GUI	10
3.6. Etiquetas para mostrar afinación	10
3.7. Botón de control de afinación	10
3.8. Control del estado de afinación	10
3.9. Procesamiento de la señal	10
3.9.1. FFT: Transformada Rápida de Fourier	10
3.9.2. Frecuencia dominante	11
3.9.3. Cuerda más cercana a la frecuencia escuchada	11
3.9.4. Filtro pasa-bajos	11
3.10. Lógica de afinación	12
3.11. Configuración y ejecución del stream de audio	13
3.12. Utilidad de la ventana de Hamming	13
4. Resultados	14
4.1. Descripción del proceso de prueba	14
4.2. Comparación con afinadores comerciales	14
5. Conclusiones	16
Bibliografía	17

Introducción

Para la ejecución de un instrumento musical, es necesario que el sonido de este instrumento este dentro de los parámetros y reglas musicales que son teóricamente generales para cualquier instrumento, solo cambiando la ejecución respectivamente. Esto quiere decir que, si un pianista desea tocar cualquier acorde, éste tocará el acorde según convencionalmente y por reglas musicales debe tocarse. Sin embargo, ¿qué pasaría si algún día el pianista toca el acorde, dentro de las reglas, pero esta vez le suena mal, le suena tensionado y distinto? Muy seguramente, el instrumento sufrió un proceso llamado desafinación. En algunas ocasiones las cuerdas de una guitarra o de un piano de cola se pueden tensionar o destensionar haciendo que al ser ejecutadas, suenen a una mayor o menor frecuencia de lo que debería sonar originalmente y se dice entonces que el instrumento está desafinado. Muy pocas personas en el mundo cuentan con una capacidad auditiva y mental llamada *oído absoluto*, esta es la capacidad de reconocer frecuencias y asociarlas con una nota musical específica que suena a esa frecuencia, sin embargo, no necesariamente por ser músico, se tiene dicha capacidad, por lo tanto se hace necesaria una herramienta que permita tensionar o destensionar las cuerdas de un instrumento para hacer coincidir la frecuencia a la que suena, con la frecuencia establecida para que el instrumento esté *afinado*.

En este documento se presenta la construcción de un programa que identifica frecuencias dominantes a través de la transformada rápida de Fourier y permite compararlas con las frecuencias de las cuerdas de una guitarra clásica afinada en Mi menor (Em) anunciando si la cuerda esta tensionada o si está destensionada para finalmente quedar afinada.

1. Fundamento teórico

Este proyecto se basa en el procesamiento de señales de audio con el objetivo de construir un programa que permita afinar una guitarra clásica, en su afinación estándar.

1.1. Frecuencia y tono

La frecuencia mide el número de repeticiones de un evento periódico en una unidad de tiempo. De esta manera, la frecuencia de onda indica el número de oscilaciones que realiza la onda por unidad de tiempo. En un instrumento musical el sonido que se produce al ejecutarlo genera ondas de sonido a diferentes frecuencias que producen armonía en el ambiente. En muchos lugares cotidianos está la frecuencia, por ejemplo el medidor de rpm del motor de un carro, el latido del corazón de una persona y claro que en las ondas de sonido que influyen según su frecuencia en la forma en que escuchamos. La relación de la física y la frecuencia es muy estrecha. Los fundamentos de la música pueden descomponerse en una secuencia de notas que se identifican a través del tono o **tasa de frecuencia** percibida en cada onda sonora individual.

Concepto de tono

Un tono es la frecuencia percibida de un sonido, y musicalmente se pueden asignar distintos valores de nota en función de la frecuencia de un tono, estos se organizan en series de nombres, notas, escalas, melodías que juntas generan armonía según las reglas musicales. El tono se mide por la frecuencia, de esta manera, los tonos más altos son aquellos cuyas frecuencias son más rápidas y lo contrario serían los tonos más bajos. Todo esto frente a que el rango de frecuencia que puede escuchar el oído humano es de 20 Hz a 20000 Hz.

Solo 1 de cada 10000 personas poseen la capacidad natural de identificar tonos basándose en un punto de referencia o una canción, el oído absoluto.

Diferencia entre tono y afinación

Tono es un término más subjetivo, y principalmente hace referencia a una gama específica de frecuencias en Hz, que a su vez crean el sonido de una concreta. En un instrumento musical, según el tono que se esté tocando, debe sonar a la frecuencia establecida para ese tono, por el contrario, el instrumento está desafinado. Así que la afinación se refiere a la coincidencia de una frecuencia producida por una onda de sonido con una frecuencia establecida de un tono.

Partiendo de este principio de frecuencia y tono, se entiende que los instrumentos musicales necesitan estar afinados para poder ejecutar los tonos y que estos suenen en armonía con los demás instrumentos e incluso con la voz. Para este proyecto, el instrumento a usar será una guitarra clásica.

1.2. Afinación de la guitarra

La guitarra, al ser un instrumento de cuerda, es muy propenso a sufrir desafinación, ya que la afinación de este instrumento funciona por la tensión de sus cuerdas de distinto grosor para generar las frecuencias adecuadas, Las cuerdas de una guitarra se pueden definir durante su ejecución, se pueden desafinar cuando hace más frío en una región, se pueden desafinar al agarrar el instrumento y constantemente deben ser revisadas para que estén afinadas y a la hora de ejecutar el instrumento este suene armónicamente.

Notas estándar de afinación (EADGBE)

Musicalmente, las notas musicales son expresadas a través del cifrado americano así:

- DO: C
- RE: D
- MI: E
- FA: F
- SOL: G
- LA: A
- SI: B

Sin embargo cabe recalcar que para una nota en especial, no solo existe una frecuencia, ya que estas dependen del armónico de la nota. Si una nota se mira en su segundo armónico, esta será la misma nota, solo que más aguda y así según avanza en sus armónicos. De esta manera una nota C es lo mismo que C2 sin embargo, su frecuencia es más rápida y por lo tanto la nota suena más aguda. En la guitarra, se tienen seis cuerdas con seis notas diferentes, dos de ellas, la primera y la última cuerda en distintos armónicos: E4, B3, G3, D3, A2, E2, donde el número indica el armónico o como musicalmente se le llama, la octava de la nota. Esto permite conocer a que frecuencia deben sonar estas notas para decir que la guitarra está afinada.

Descripción de las frecuencias correspondientes a cada cuerda

La afinación estándar para una guitarra clásica indica que las frecuencias (Hz) correspondientes a cada cuerda son:

- E4: 329.63 Hz
- B3: 246.94 Hz
- G3: 196.00 Hz
- D3: 146.83 Hz

- A2: 110.00 Hz
- E2: 82.41 Hz

Estas frecuencias serán el parámetro a seguir para realizar la afinación del instrumento.

1.3. Procesamiento de señales digitales

Para este proyecto, es fundamental usar la Transformada Rápida de Fourier para encontrar a través del espectro de frecuencias, las frecuencias dominantes al capturar audio en tiempo real. Y los filtros pasabanda, en especial el filtro pasa bajo para eliminar frecuencias altas y trabajar en un rango específico de frecuencias.

FFT (Fast Fourier Transform)

Con el análisis FFT, se pueden determinar particularidades de una señal en mucha mayor medida que cuando se inspeccionan los datos del dominio del tiempo. En el dominio de la frecuencia, las características de la señal se describen mediante componentes de frecuencia independientes, mientras que el dominio del tiempo se describe mediante una forma de onda que contiene la suma de todas las características.

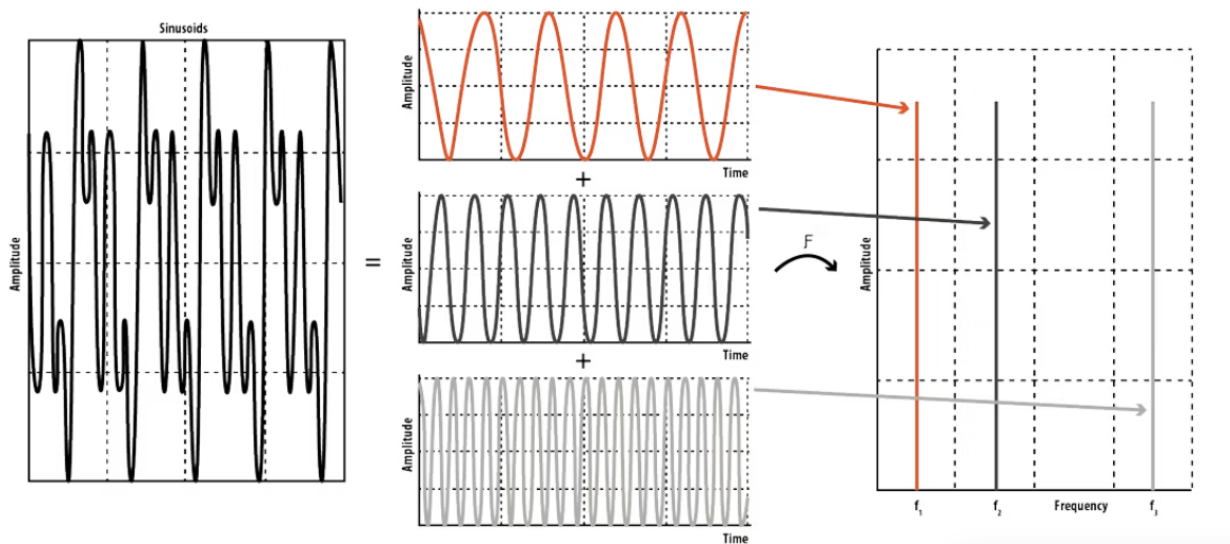


Figura 1: Dominio del tiempo y dominio de la frecuencia en una onda sinusoidal

El análisis de Fourier transforma señales del dominio del tiempo al dominio de la frecuencia, pero más correctamente, es un método matemático para transformar una función de tiempo finito $a(t)$ de N muestras de tiempo igualmente espaciadas en una función de frecuencia $A(f)$ de N muestras de frecuencia complejas igualmente espaciadas.

La FFT es un desarrollo de la transformada discreta de Fourier (DFT), pero la FFT, elimina

los términos duplicados en el algoritmo matemático para reducir el número de operaciones matemáticas realizadas, teniendo una complejidad de $N/\log(N)$ mientras que para DFT es N^2 .

Para este proyecto, la FFT convierte la señal de audio capturada, que está en el dominio del tiempo, al dominio de la frecuencia. De esta manera, se descompone la señal compleja de audio en sus componentes de frecuencia individuales y así se hace más fácil identificar la frecuencia dominante. Con la ayuda de la FFT, se puede analizar el espectro de frecuencia hallando la frecuencia con mayor magnitud para posteriormente comparar esta frecuencia con la frecuencia que debe tener la cuerda tocada. Además, gracias a la rapidez de este algoritmo, se hace más útil para la afinación en tiempo real. A través de la FFT, se puede implementar filtros digitales como filtros pasabajos y pasabanda para eliminar el ruido y las frecuencias no deseadas.

Filtro pasabajo

El filtro pasabajos permite el paso de frecuencias bajas, atenúa o rechaza las frecuencias superiores a la frecuencia de corte y para este proyecto en específico permite eliminar ruidos de alta frecuencia que puedan interferir en la afinación del instrumento y el reconocimiento de la frecuencia fundamental.

1.4. Tecnologías usadas

Para el diseño del programa que permite construir el afinador para la guitarra el lenguaje usado fue Python3.11 a través del cual con la ayuda de algunas librerías se facilitó el cálculo de la FFT y el procesamiento del audio.

Librerías usadas

- **PyAudio:** Esta librería permite capturar audio en tiempo real para luego tratarlo.
- **Numpy:** Esta es una librería matemática que permite tratar los datos del audio capturado y manipularlos a través de la FFT.
- **Tkinter:** Este módulo permite construir una interfaz gráfica para entender mejor el funcionamiento del programa.
- **Scipy.Signal:** Este módulo permite la construcción de los filtros digitales para eliminar ruidos y frecuencias no deseadas.

2. Diseño del afinador

Ya que este afinador aún no cuenta con una implementación de hardware que pueda usarse, por ahora es un programa en Python el cual a través de una interfaz gráfica permite al usuario interactuar con él para afinar su instrumento.

2.1. Estructura general del programa

De manera muy general, el programa se basa en capturar audio en tiempo real que será procesado en tiempo real para ir a la lógica de afinación y posteriormente a través de una interfaz gráfica lograr interactuar con el usuario que desee afinar su guitarra.

Se podría decir que el programa se basa en cuatro grandes partes:

- Captura de audio: Es importante capturar el audio en tiempo real.
- Procesamiento de las señales: A través del cual se encuentran las frecuencias dominantes que serán comparadas con las establecidas.
- Lógica de afinación: Donde se realizan las comparaciones necesarias para indicar si la cuerda está afinada, alta o baja
- GUI: La interfaz gráfica del usuario con la cual se logra mostrar al usuario el comportamiento de la frecuencia de sus cuerdas.

3. Implementación

3.1. Importaciones

Traen las librerías necesarias para la manipulación de audio (pyaudio), procesamiento de señales (numpy, scipy) y la construcción de la interfaz gráfica (tkinter).

```
1 import pyaudio
2 import numpy as np
3 import tkinter as tk
4 from tkinter import ttk
5 from scipy.signal import butter, lfilter
```

3.2. Captura de audio

Estos parámetros definen la configuración del audio: formato, número de canales, tasa de muestreo y tamaño del buffer.

```
1 # Definir parámetros de audio
2 FORMAT = pyaudio.paInt16 # Formato de los datos de audio (16-bit PCM)
3 CHANNELS = 1             # Número de canales (mono)
4 RATE = 22050              # Tasa de muestreo (samples per second)
5 CHUNK = 32768             # Tamaño del buffer (número de frames por
                           # buffer)
```

3.3. Frecuencias de referencia y tolerancia

Este diccionario define las frecuencias de referencia para cada cuerda de la guitarra. La tolerancia define el rango aceptable de desviación de la frecuencia detectada con respecto a la frecuencia de referencia. Notar que E2 se define como el segundo armónico en una frecuencia distinta a la original (multiplicada por 2) debido a que la frecuencia dominante es tomada como el siguiente armónico.

```
1 REFERENCE_FREQUENCIES = {
2     "E2": 164.82,
3     "A2": 110.00,
4     "D3": 146.83,
5     "G3": 196.00,
6     "B3": 246.94,
7     "E4": 329.63
8 }
9
10 TOLERANCE = 1.0
```

3.4. Inicialización de PyAudio

Para capturar y procesar audio en tiempo real.

```
1 p = pyaudio.PyAudio()
```

3.5. Configuración de la GUI

Se configura la ventana principal fijando su tamaño y fondo

```
1 root = tk.Tk()
2 root.title("Afinador de Guitarra")
3 root.geometry("600x400") # Fijar tamaño de la ventana
4 root.resizable(False, False) # Hacer la ventana no redimensionable
5 root.configure(bg="lightblue") # Fondo de la ventana
```

3.6. Etiquetas para mostrar afinación

Se crean etiquetas para mostrar el estado de afinación de cada cuerda añadiéndose a un contenedor que esta centrado.

```
1 labels_frame = ttk.Frame(root)
2 labels_frame.pack(pady=20)
3
4 labels = {}
5 for string in REFERENCE_FREQUENCIES.keys():
6     labels[string] = ttk.Label(labels_frame, text=f"{string}:
7     Desconocido", font=("Helvetica", 16), background="lightblue")
8     labels[string].pack(anchor=tk.CENTER, pady=5)
```

3.7. Botón de control de afinación

```
1 start_button = ttk.Button(root, text="Iniciar", command=lambda:
2     control_tuning())
3 start_button.pack()
```

3.8. Control del estado de afinación

Esta función controla el estado de afinación. Cambia el texto del botón dependiendo de si la afinación está activa o no, y llama a tuneGuitar para iniciar la afinación si se activa.

```
1 def control_tuning():
2     global is_tuning
3     if is_tuning:
4         is_tuning = False
5         start_button.configure(text="Iniciar")
6     else:
7         is_tuning = True
8         start_button.configure(text="Detener")
9         tune_guitar()
```

3.9. Procesamiento de la señal

3.9.1. FFT: Transformada Rápida de Fourier

Para retornar el resultado transformado al dominio de la frecuencia.

```

1 def fft(data):
2     fft_result = np.fft.rfft(data)
3     return fft_result

```

3.9.2. Frecuencia dominante

Aplicando la FFT, se encuentra la frecuencia dominante para luego seleccionar la frecuencia con la mayor magnitud en un rango de frecuencias que es relevante para este contexto.

```

1 def dominant_freq(data):
2     frequencies = np.fft.rfftfreq(len(data), 1.0 / RATE)
3     magnitude = np.abs(fft(data))
4
5     # Frecuencias de interés
6     min_freq = 70.0
7     max_freq = 350.0
8     relevant_indices = np.where((frequencies >= min_freq) & (frequencies
9 <= max_freq))
10    relevant_frequencies = frequencies[relevant_indices]
11    relevant_magnitudes = magnitude[relevant_indices]
12
13    if len(relevant_frequencies) > 0:
14        dominant_index = np.argmax(relevant_magnitudes)
15        dominant_freq = relevant_frequencies[dominant_index]
16        dominant_magnitude = relevant_magnitudes[dominant_index]
17
18    return dominant_freq, dominant_magnitude
19 else:
20    return None, None

```

3.9.3. Cuerda más cercana a la frecuencia escuchada

```

1 def find_closest_string(frequency, references):
2     closest_string = None
3     min_distance = float('inf')
4
5     for string, ref_freq in references.items():
6         distance = abs(frequency - ref_freq)
7         if distance < min_distance:
8             min_distance = distance
9             closest_string = string
10
11    return closest_string, min_distance

```

3.9.4. Filtro pasa-bajos

Se busca aplicar un filtro pasa-bajos para eliminar frecuencias altas no deseadas y mejorar la precisión de la detección de la frecuencia dominante.

```

1 def butter_lowpass(cutoff, fs, order=5):
2     nyquist = 0.5 * fs

```

```

3     normal_cutoff = cutoff / nyquist
4     b, a = butter(order, normal_cutoff, btype='low', analog=False)
5     return b, a
6
7 def lowpass_filter(data, cutoff, fs, order=5):
8     b, a = butter_lowpass(cutoff, fs, order=order)
9     y = lfilter(b, a, data)
10    return y

```

3.10. Lógica de afinación

Esta función es el núcleo del afinador de guitarra. Lee datos de audio en tiempo real, aplica el filtro pasa-bajos y la ventana de Hamming, calcula la frecuencia dominante y actualiza la interfaz gráfica para indicar si la cuerda está afinada, alta o baja.

```

1 def tune_guitar():
2     if not is_tuning:
3         return
4
5     # Leer datos del stream
6     data = stream.read(CHUNK)
7
8     # Convertir los datos a un array de numpy
9     audio_data = np.frombuffer(data, dtype=np.int16)
10
11    # Verificar el tama o de audio_data
12    if len(audio_data) != CHUNK:
13        print(f"Tama o inesperado de audio_data: {len(audio_data)}")
14        return
15
16    # Aplicar un filtro pasa-bajo para eliminar frecuencias altas
17    filtered_data = lowpass_filter(audio_data, 350.0, RATE)
18
19    # Aplicar la ventana de Hamming
20    windowed_data = filtered_data * np.hamming(len(filtered_data))
21
22    dominant_frequency, _ = dominant_freq(windowed_data)
23    if dominant_frequency is not None:
24        closest_string, min_distance =
25        find_closest_string(dominant_frequency, REFERENCE_FREQUENCIES)
26
27        if min_distance <= TOLERANCE:
28            if closest_string == "E2":
29                dominant_frequency /= 2
30                labels[closest_string].config(text=f"{closest_string}:
31                Afinado ({dominant_frequency:.2f} Hz)", foreground="green")
32            elif dominant_frequency > REFERENCE_FREQUENCIES[closest_string]:
33                if closest_string == "E2":
34                    dominant_frequency /= 2
35                    labels[closest_string].config(text=f"{closest_string}
36                    Alta ({dominant_frequency:.2f} Hz)", foreground="red")

```

```

35         labels[closest_string].config(text=f"{closest_string} Alta
({dominant_frequency:.2f} Hz)", foreground="red")
36     else:
37         if closest_string == "E2":
38             dominant_frequency /= 2
39             labels[closest_string].config(text=f"{closest_string}
Baja ({dominant_frequency:.2f} Hz)", foreground="blue")
40             labels[closest_string].config(text=f"{closest_string} Baja
({dominant_frequency:.2f} Hz)", foreground="blue")
41
42     root.after(100, tune_guitar)

```

3.11. Configuración y ejecución del stream de audio

Estas líneas configuran el stream de audio, inician la captura de audio en tiempo real y lanzan la interfaz gráfica. Al cerrar la interfaz, el stream de audio se detiene y PyAudio se cierra correctamente.

```

1 stream = p.open(format=FORMAT,
2                 channels=CHANNELS,
3                 rate=RATE,
4                 input=True,
5                 frames_per_buffer=CHUNK)
6
7 print("Captura de audio en tiempo real iniciada. Presiona Ctrl+C para
detener.")
8
9 root.mainloop()
10
11 stream.stop_stream()
12 stream.close()
13 p.terminate()

```

3.12. Utilidad de la ventana de Hamming

Se usa la ventana de Hamming como una herramienta para eliminar discontinuidades al principio y al final de un bloque analizado, ya que cuando se aplica la FFT a una señal, se asume que la señal es periódica dentro del período analizado, sin embargo esto en la práctica no sucede así, las discontinuidades pueden generar errores en la representación de la señal en el dominio de la frecuencia, lo cual se conoce como **leakege espectral**. Con la ventana de Hamming se suavizan las discontinuidades y se mejora la capacidad para distinguir entre frecuencias cercanas. En la lógica de afinación del programa, esta herramienta es utilizada.

4. Resultados

Los resultados son satisfactorios, se logra dejar con un margen de error muy pequeño, muy bien afinada la guitarra. El proceso de captura y análisis de la señal que recibe el programa se diseñó de una manera óptima, teniendo en cuenta la tasa de muestreo y el tamaño del buffer para así tomar muestras necesarias en unidad de tiempo. En general, se podría decir que el afinador es preciso.

4.1. Descripción del proceso de prueba

- Generalmente, se afina una guitarra de abajo para arriba siendo E4 la primera cuerda. El reconocimiento de cada toque a la cuerda demora aproximadamente un segundo, no solo el reconocimiento sino el procesamiento de la señal y su análisis incluyendo la lógica para saber si la cuerda esta alta, baja o afinada.
- Al llegar a E2, la aplicación reconoce la frecuencia del segundo armónico de E2 que es $2 \times 82,41Hz$ es decir $164,82Hz$ y esto es fundamental entenderlo para lograr que el afinador reconozca dentro de su rango específico de frecuencias a escuchar, la cuerda E2.
- Se puede decir que el afinador es preciso ya que tiene una tolerancia de 1.0 lo cual hace que la frecuencia a la que finalmente suene cada cuerda sea muy poco distante de la frecuencia establecida e ideal.
- Al probarlo con las notas de un teclado, reconoce inmediatamente las notas y sus frecuencias iguales a las establecidas.

4.2. Comparación con afinadores comerciales

- Se hizo la comparación con la aplicación de afinación de instrumentos musicales *Guitar Tuna* la cual mostraba en solo dos o tres cuerdas que la cuerda habia quedado 1Hz mas arriba o mas abajo de lo que deberia estar, y ese unico hercio coincidia con lo que decia el afinador, por lo tanto da una idea de que el afinador construido es preciso.
- El afinador comercial es más rápido a la hora de procesar la señal y anunciar su resultado, sin embargo el tiempo de respuesta del afinador construido no es tan largo en comparación con el comercial

Ventajas y limitaciones

- Este afinador usa la transformada rápida de fourier, filtro pasa-bajos y ventana de Hamming lo cual hace que la determinación de la frecuencia dominante sea muy precisa.
- No necesita otro tipo de hardware para funcionar, únicamente el micrófono por defecto del computador

- La interfaz gráfica es intuitiva, sin embargo necesita ser mejorada para ser una excelente aplicación de afinación
- Es limitada a afinar únicamente guitarra clásica, sin embargo siguiendo el mismo concepto se podrían afinar otros instrumentos.

5. Conclusiones

A través del ejercicio de la aplicación de la FFT, se identifica la importancia del procesamiento de señales digitales para resolver un problema práctico. A lo largo del proyecto, se abordaron varios aspectos técnicos y conceptuales que permitieron construir una herramienta funcional y útil para músicos.

La efectividad de la FFT para analizar señales es notoria a lo largo de este proyecto, ya que en tiempo real, fue muy importante el uso de la FFT para encontrar frecuencias dominantes. La capacidad de descomponer una señal compleja en sus componentes frecuenciales permitió identificar con precisión las frecuencias dominantes correspondientes a las cuerdas de la guitarra. Aspecto que es fundamental para un afinador ya que requiere detectar pequeñas variaciones en las frecuencias para generar una mejor guía al usuario a la hora de afinar sus cuerdas.

Este proyecto, resalta la importancia de las ventanas de Hamming en la reducción de efectos no deseados en el análisis de señales discretas, esta ventana facilitó la minimización del efecto de los extremos de los segmentos de señal analizados, mejorando la precisión de la detección de frecuencias dominantes.

Para mejorar la usabilidad del afinador, al proporcionar una retroalimentación visual clara sobre el estado de afinación de cada cuerda, los usuarios pueden ajustar las cuerdas de su guitarra de manera más eficiente y con mayor precisión.

Se destaca la importancia de ajustar los parámetros de filtrado y la tasa de muestreo para así, obtener resultados precisos en tiempo real. Haber implementado filtros pasa-bajo eliminó las frecuencias no deseadas y permitió centrar el análisis en el rango relevante para la afinación de la guitarra.

Cabe resaltar el hecho de conocer los armónicos que tiene cada nota, ya que el afinador reconoce para una cuerda la frecuencia de su segundo armónico y no su frecuencia fundamental que es a la frecuencia que debería reconocerla. Saber esto, permitió solucionar el problema de que el afinador no reconociera la última cuerda de la guitarra, la más grave.

Bibliografía

1. Electricos, C. (2020, junio 4). ¿Qué es la frecuencia y cómo funciona? Circuitos Eléctricos. <https://www.circuitos-electricos.com/que-es-la-frecuencia-y-como-functiona/>
2. ¿Qué es el tono en la música? (s/f). Emastered.com. Recuperado el 30 de mayo de 2024, de <https://emastered.com/es/blog/what-is-pitch-in-music>
3. Guía para Análisis FFT. (s/f). Soluciones de Adquisición de Datos (DAQ). Recuperado el 30 de mayo de 2024, de <https://dewesoft.com/es/blog/guia-para-el-analisis-fft>
4. Por, M. (2023, junio 11). ¿Cómo funcionan los filtros pasa-bajos, pasa-altos, pasa-banda y rechaza-banda? Electricity - Magnetism. [Website](#)
5. Ventana Hamming. (s/f). WIKIDAT. Recuperado el 30 de mayo de 2024, de [Website](#)
6. Powered by Chat-GPT [Chat-GPT](#)