

Informe de Interfaz Gráfica para la Visualización de Señales Periódicas y su Espectro Frecuencial

Autor: Diego Andrés Rodríguez, Dilan Camilo Acosta, Steven Alejandro Urbina.

Fecha: 16 octubre 2024

Introducción

El propósito de esta interfaz gráfica es permitir la visualización de distintas señales periódicas y su análisis en el dominio de la frecuencia mediante la serie de Fourier. El usuario tiene la opción de manipular el periodo de las señales utilizando un control deslizante, lo cual afecta la señal en el tiempo y modifica su espectro frecuencial. De esta forma, la interfaz muestra cómo el espectro de Fourier evoluciona conforme se ajusta el periodo de la señal, lo que facilita el análisis de las componentes de frecuencia de manera interactiva.

Marco Conceptual

- Señales periódicas

Las señales periódicas son aquellas que se repiten de forma regular con el tiempo, lo que las hace fundamentales en campos como la telecomunicación y el procesamiento digital. Estas señales se pueden descomponer en componentes básicas como ondas senoidales a diferentes frecuencias, lo que permite analizarlas en el dominio de la frecuencia. La Transformada Rápida de Fourier (FFT) es una técnica computacional eficiente para obtener el espectro de una señal en el dominio de la frecuencia. Con esta herramienta, es posible transformar la señal desde el dominio del tiempo a una representación en el dominio de la frecuencia, lo que facilita el análisis de las componentes de frecuencia presentes en la señal.

- Serie de Fourier

La Serie de Fourier es una herramienta matemática que permite descomponer una señal periódica compleja en una suma de senos y cosenos. Esta técnica es muy utilizada en el análisis de señales, ya que proporciona una representación del contenido de la señal en términos de frecuencia, lo que facilita su estudio en diferentes aplicaciones tecnológicas.

- Transformada rápida de Fourier FFT

La FFT es una versión optimizada de la Transformada de Fourier que permite obtener el espectro de frecuencias de una señal de manera rápida y eficiente. Esto es especialmente útil cuando se requiere analizar las componentes frecuenciales de señales periódicas y verificar cómo varían en función del tiempo o del periodo de la señal.

- DTFT

La Transformada de Fourier en Tiempo Discreto (DTFT) convierte una señal discreta en una representación continua en el dominio de la frecuencia. A diferencia de la FFT, que trabaja con un número limitado de frecuencias, la DTFT proporciona un espectro continuo que describe todas las frecuencias posibles en la señal. Es ideal para estudiar señales periódicas o de larga duración, ya que permite observar cómo están distribuidas las frecuencias en toda la señal. Aunque es una herramienta más teórica, en la práctica se usa la FFT para obtener una versión aproximada de la DTFT al trabajar con señales reales

Implementación de la Interfaz

La interfaz gráfica desarrollada permite cargar y visualizar diferentes tipos de señales periódicas, así como su representación en el dominio de la frecuencia a través de la serie de Fourier. Las señales disponibles en la interfaz incluyen: señal seno, cuadrada, trapezoidal, triangular y de diente de sierra.

El usuario puede manipular el periodo de estas señales mediante un control deslizante (fader), lo que permite ver cómo cambia la señal en el tiempo y, de forma simultánea, cómo varía su espectro de frecuencias. El espectro es calculado utilizando la Transformada Rápida de Fourier (FFT), y se muestra en un gráfico con escala logarítmica.

La implementación se realizó utilizando Python y las bibliotecas PyQt5 para la interfaz gráfica, y Matplotlib para la representación gráfica de las señales y su espectro. Además, se emplea Numpy para el procesamiento de las señales y la realización de la FFT.

Conclusiones

La interfaz gráfica desarrollada permite visualizar de manera eficiente señales periódicas y su espectro de frecuencias. La manipulación del periodo de la señal brinda una herramienta útil para observar cómo la frecuencia de una señal está relacionada con su periodo, y cómo esta relación afecta su representación en el dominio de la frecuencia.

El uso de la FFT ha permitido realizar un análisis rápido y eficiente del contenido frecuencial de las señales, mostrando su transformación desde el dominio del tiempo. La interfaz proporciona una base para el estudio y análisis de señales periódicas en campos como el procesamiento de señales y la ingeniería de sonido.

Anexos

Código principal de cálculo de señales periódicas Y función CTFT, en este caso DTFT ya que nos encontramos en tiempos discretos.

```
fs = 192000
tipo_senal = "seno" # Valor por defecto
octave_bands = np.array([31.5, 63, 125, 250, 500, 1000, 2000, 4000, 8000, 16000, 32000])

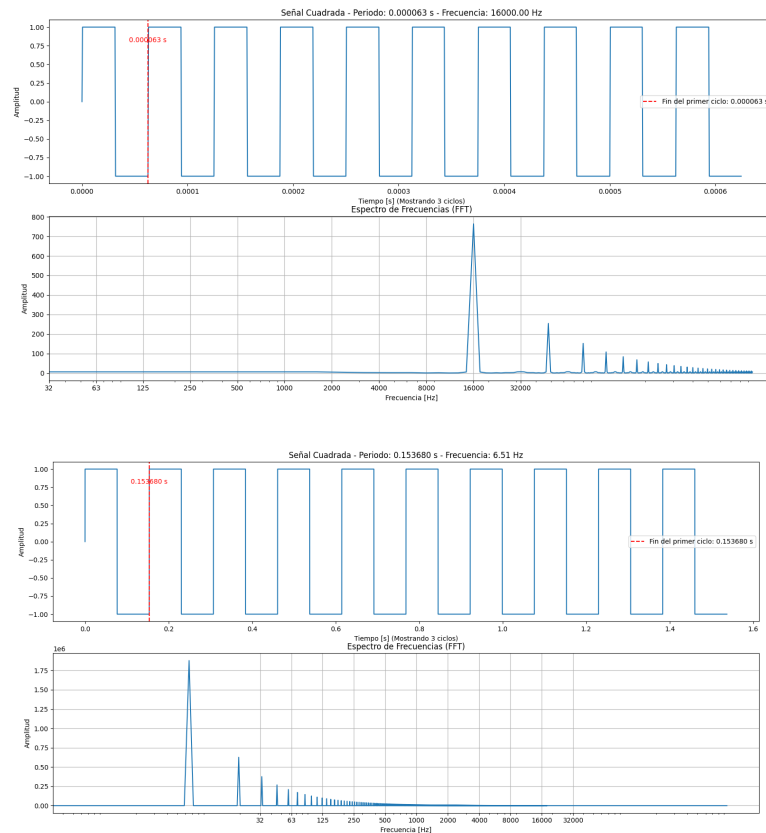
# lista de periodos discretos proporcionada
periodos = [
    5e-05, 6.25e-05, 8e-05, 0.0001, 0.000125, 0.00015873015873015873,
    0.0002, 0.00025, 0.00031746031746031746, 0.0004, 0.0005, 0.000625,
    0.0008, 0.001, 0.00125, 0.0015873015873015873, 0.002, 0.0025,
    0.0031746031746031746, 0.004, 0.005, 0.00625, 0.008, 0.01, 0.0125,
    0.015873015873015872, 0.02, 0.025, 0.031746031746031744, 0.04, 0.05
]

# Función para generar diferentes señales periódicas
def generar_senal(tipo, frecuencia, ciclos=10): # Aumentamos los ciclos a 10
    num_puntos = int(fs * ciclos / frecuencia) # Aseguramos suficientes puntos
    t = np.linspace(0, ciclos / frecuencia, num_puntos, endpoint=False)

    if tipo == "seno":
        senal = np.sin(2 * np.pi * frecuencia * t)
    elif tipo == "cuadrada":
        senal = np.sign(np.sin(2 * np.pi * frecuencia * t))
    elif tipo == "trapezoidal":
        senal = np.piecewise(t, [np.mod(t, 1/frecuencia) < (1/frecuencia)/2, np.mod(t, 1/frecuencia) >= (1/frecuencia)/2],
                             [lambda x: 4 * frecuencia * x - 1, lambda x: -4 * frecuencia * x + 3])
    elif tipo == "triangular":
        senal = 2 * np.abs(2 * (t * frecuencia - np.floor(0.5 + t * frecuencia))) - 1
    elif tipo == "diente-de-sierra":
        senal = 2 * (t * frecuencia - np.floor(0.5 + t * frecuencia))

    return t, senal

# Función para calcular la FFT sin normalización
def calcular_fft(senal):
    N = len(senal) # Usamos la longitud completa de la señal
    espectro = np.fft.fftfreq(N, 1/fs) # Eliminamos la normalización
    frecuencias = np.fft.fftfreq(N, 1/fs)
    espectro_positivas = np.abs(espectro[:N // 2]) # Solo la mitad positiva
    frecuencias_positivas = frecuencias[:N // 2]
    return frecuencias_positivas, espectro_positivas
```



Se puede observar como a medida que se aumenta el periodo de la señal en este caso cuadrada. La función tiende mas a ser continua, sin embargo, se logra apreciar unos cambios de amplitud.

Referencias

- Oppenheim, A. V., & Schafer, R. W. (2010). Discrete-Time Signal Processing (3rd ed.). Pearson Education.
- Lyons, R. G. (2011). Understanding Digital Signal Processing (3rd ed.). Pearson Education.