

EQUIPO DE MEDICIÓN DE ENERGÍA

ANTEPROYECTO

Fernando Fortunati
e-mail: nfortunati@gmail.com

Sebastian Martínez
e-mail: sebajm1363@gmail.com

RESUMEN: Se aplicarán los conceptos aprendidos en la cátedra Procesamiento Embebido de Señales para el diseño y realización de un medidor de energía.

PALABRAS CLAVE: Calidad de energía, True RMS, THD, FFT, Ventana Hamming, Python.

1 INTRODUCCIÓN

El objetivo de la 1ra Etapa de este proyecto, consistirá en hacer una primera aproximación al código a implementarse en un microcontrolador, ya que es el objetivo que nos ponemos para la Etapa 2.

En el mismo se abarcaran los conocimientos aprendidos sobre muestreo de señales, aplicación de Ventana y FFT.

Al desear aplicarle la FFT a una señal continua, y que el hecho de muestrearla hace que se pierda información; se le aplicara previamente a la función una ventana de Hanning para suavizarla y evitar grandes discontinuidades.

2 ALCANCE

Se propone la adquisición de una señal mediante un microcontrolador, con el objetivo de realizar su procesamiento interno para determinar parámetros característicos, tales como el verdadero valor eficaz (True RMS) y la distorsión armónica total (THD).

Para el cálculo de la THD, se aplicará una Transformada Rápida de Fourier (FFT) a la señal previamente muestreada, obteniendo así los componentes espectrales necesarios para la estimación del valor mencionado.

Posteriormente, los valores correspondientes a la señal muestreada, los parámetros utilizados en la Transformada Rápida de Fourier (FFT), así como los resultados del cálculo de True RMS y del índice de Distorsión Armónica Total (THD), serán transmitidos mediante una interfaz de comunicación UART.

Finalmente, la representación gráfica de la señal en el dominio temporal, el espectro en frecuencia y los valores calculados se realizará a través de una aplicación desarrollada en Python.

3 MARCO TEÓRICO

3.1 Muestreo:

El muestreo consiste en convertir una señal continua en una secuencia discreta de valores tomados en instantes periódicos. Según el Teorema de Nyquist, para evitar aliasing, la frecuencia de muestreo f_s debe ser al menos el doble de la máxima frecuencia presente en la señal ($f_s \geq 2f_{max}$).

Para la selección de la frecuencia de muestreo se seguirán los siguientes lineamientos:

- Definir hasta qué armónico se desea mostrar en la FFT.
- Con la frecuencia máxima que se desea ver en el espectro, se seleccionará la frecuencia de muestreo de acuerdo al criterio de Nyquist.
- Se deberá adecuar este valor de frecuencia para que sea potencia de 2, de esta forma evitamos el leakage.

EJEMPLO:

Fundamental: 50 Hz

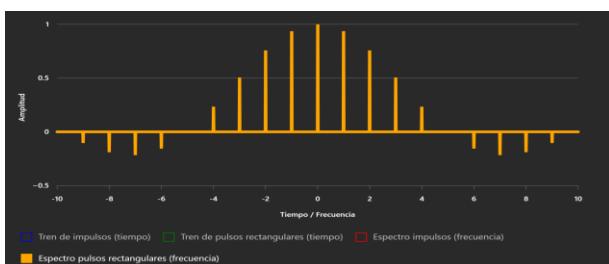
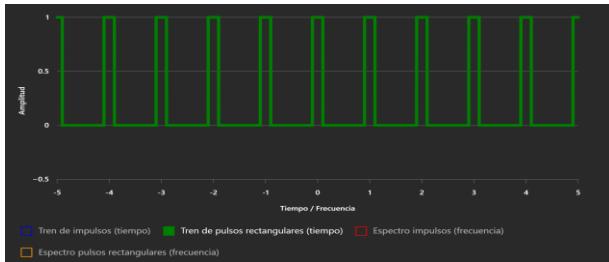
Cantidad de armónicos: 25

Frecuencia Máxima (Fmax): $25 * 50 \text{ Hz} = 1250 \text{ Hz}$

Frecuencia de Muestreo (Fs): $2 * F_{max} = 2500 \text{ Hz}$

Fs corregida: 4096 Hz (exactamente 64 ciclos)

En la práctica, el muestreo no se realiza con impulsos ideales, sino con un tren de pulsos rectangulares, lo que introduce un efecto adicional: la señal muestreada se convoluciona con una función sinc en frecuencia, debido al ancho finito de los pulsos.



3.2 Ventana de Hanning:

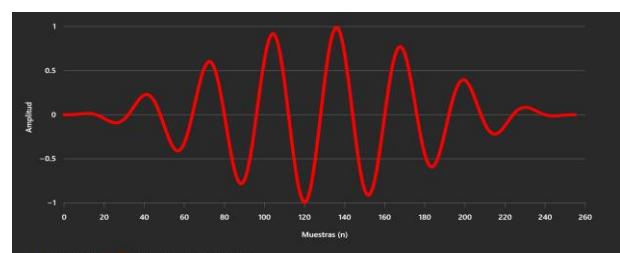
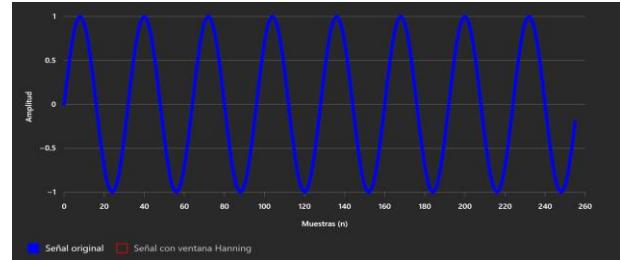
La **Ventana de Hanning** (o Hann) es una función utilizada en procesamiento digital de señales para reducir el **efecto de fuga espectral** cuando se aplica la Transformada de Fourier a señales finitas. En lugar de cortar la señal abruptamente, se multiplica por una ventana que suaviza los extremos, disminuyendo las discontinuidades.

Su fórmula es:

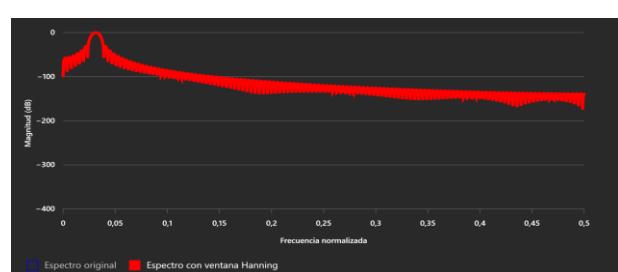
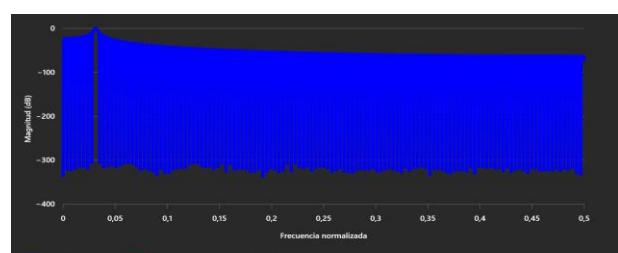
$$\omega[n] = 0,5 * \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), 0 \leq n \leq N-1$$

Características:

- Reduce la fuga espectral mejor que una ventana rectangular.
- Atenua los extremos de la señal a cero.
- Se usa en análisis espectral y filtrado.



Para mas claridad se analizan ahora, los espectros antes y después de aplicarle la ventana:



Como se puede apreciar, cuando no se aplica la ventana hay un alto nivel de componentes de alta frecuencia que no pertenecen a la señal original. Aunque el pico principal está en la frecuencia correcta, la energía se dispersa en frecuencias adyacentes, dificultando la interpretación.

En cambio, luego de aplicar la ventana, se aprecia una atenuación de las componentes de alta frecuencia. Esto se debe a que se suavizaron los extremos de la señal (Figura 4), lo que disminuye la Fuga Espectral. El resultado es un espectro más limpio, con menos energía fuera de la frecuencia principal.

3.3 Transformada Rápida Fourier (FFT):

La Transformada Rápida de Fourier (FFT) es un algoritmo eficiente para calcular la Transformada Discreta de Fourier (DFT), que convierte una señal del dominio del tiempo al dominio de la frecuencia. Esto permite analizar qué componentes frecuenciales forman la señal.

La DFT se define como:

$$X[k] = \sum_{n=0}^{N-1} x[n] * e^{-j2\pi kn/N}, k = 0, 1, 2, \dots, N - 1$$

Diferencias:

- La DFT directa requiere $O(N^2)$ operaciones para N muestras.
- La FFT reduce esto a $O(N \log N)$, lo que permite análisis rápido incluso para señales largas.
- El punto anterior, en el caso de sistemas embebidos. Mejora la velocidad de procesamiento y la utilización de recursos.

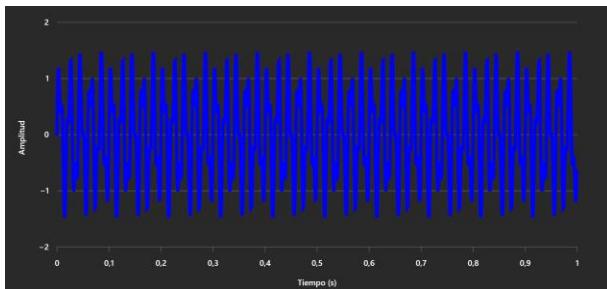


Figura 7: Señales senoidales de 50 y 120 Hz

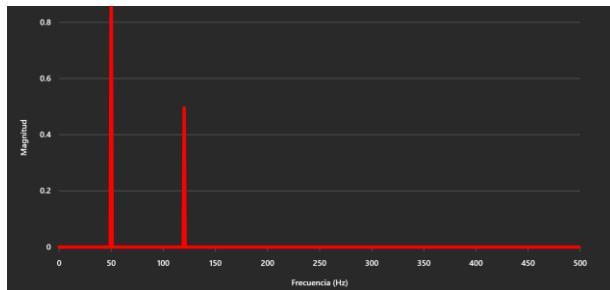


Figura 8: FFT de la señal senoidal

4 DESARROLLO

Para la metodología de trabajo en esta 1ra Etapa, se utilizará un Notebook de Python, para probar el código que luego se intentará implementar embebido en un microcontrolador.

Para simular todos los pasos antes mencionados, se generaran distintas señales (conocidas); de esta forma se conocerá de antemano el espectro resultante y se podrá contrastarlo posteriormente en la Etapa 2.

A estas señales se las muestreara, de acuerdo a los conceptos antes mencionados; para obtener la señal discreta que luego se procesara.

Esta señal se convolucionara con la respuesta al impulso de la Ventana de Hanning, para evitar discontinuidades y por lo tanto fuga espectral.

Para concluir, se le aplicara la FFT para obtener los componentes de la señal. Y con estos componentes se obtendrá el valor de la THD.

Aprovechando los beneficios de Python, se graficaran todas las señales antes mencionadas para poder seguir el proceso y validar los resultados.

5 BIBLIOGRAFIA

[1] Documentación Librería Numpy [En Línea]
<https://numpy.org/doc/stable/>

[2] “Muestreo y ADC” – Apunte de clase para Procesamiento Embebido de Señales – Facultad Regional Avellaneda, Universidad Tecnológica Nacional

[3] “Análisis Espectral” – Apunte de clase para Procesamiento Embebido de Señales – Facultad Regional Avellaneda, Universidad Tecnológica Nacional