

# Procesamiento de señales

Brayan Steven Mendivelso Pérez

est.brayan.mendive@unimilitar.edu.co

Docente: José de Jesús Rúgeles

**Resumen—** En este informe se analiza y procesa cuatro tipos de señales desde las perspectivas del dominio del tiempo y del dominio de la frecuencia. Las señales estudiadas incluyen: una señal senoidal con y sin *offset*, una señal triangular con y sin *offset*, una señal cuadrada con y sin *offset* y, por último, una señal de pulso con diferentes ciclos útiles.

Se utilizaron un generador de señales y un osciloscopio para capturar las señales y exportar los datos en un archivo .csv. Este archivo fue utilizado posteriormente para graficar y analizar las señales en MATLAB y Thonny.

**Abstract--** In this report, we analyze and process four types of signals from both time-domain and frequency-domain perspectives. The signals studied include: a sinusoidal signal with and without offset, a triangular signal with and without offset, a square wave with and without offset, and finally, a pulse signal with different duty cycles.

A signal generator and an oscilloscope were used to capture the signals and export the data to a .csv file. This file was then used to graph and analyze the signals in MATLAB and Thonny.

## I. SEÑAL SENOIDAL.

Lo primero que haremos es configurar el generador de señales con una frecuencia de 3,5 KHz y un voltaje pico de 1,5, observamos la señal senoidal en el osciloscopio una vez comprobada que la señal es la deseada vamos a usar la aplicación OpenChoice Desktop instalada en el computador para poder extraer los archivos .CSV, una vez tenemos el archivo .CSV vamos a comprobar por MATLAB que si es la señal correcta como se configuró en el generador de señales.

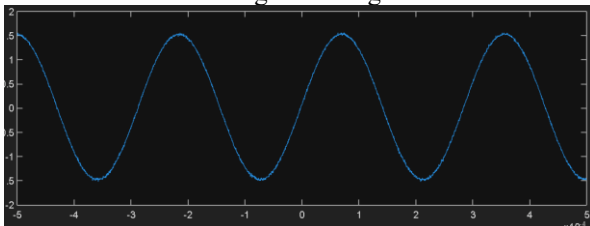


Ilustración 1 Señal senoidal.

A partir de esto, se desea analizar la señal en su dominio de la frecuencia, para este vamos a usar la siguiente formula

$$dB = 20 \log_{10} \left( \frac{V}{\sqrt{2}} \right)$$

Ecuación 1.

Esta nos sirve para volver los voltajes dB, ya que sabemos que la FFT se usa dB y frecuencia, sabiendo esto vamos a reemplazar el voltaje en la ecuación 1

$$20 \log_{10} \left( \frac{1,5 V}{\sqrt{2}} \right) = 0,511 dB$$

Ahora vamos a ingresar a Matlab y vamos a volver nuestro archivo .csv a una señal FFT, para esto vamos a usar el código que se logra ver en la ilustración 2

```
data = readmatrix('tiempo señal 1.csv');
t = data(3:end, 4);
x = data(3:end, 5);
% 2. Parámetros clave
Fs = 1 / (4e-7);
N = length(x);
X = fft(x);
X_mag = abs(X / N);
f = Fs * (0:(N/2)) / N;
P1 = X_mag(1:N/2+1);
P1(2:end-1) = 2 * P1(2:end-1);
P1_dB = 20*log10(max(P1, 1e-12));
figure;
plot(f, P1_dB, 'LineWidth', 1.5);
grid on;
title('Espectro de Frecuencia (FFT)');
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
set(gca, 'FontSize', 12);
xlim([100 9000]);
ylim([-30 1]);
```

## Ilustración 2 Comando para generar gráfico.

Este código realiza un análisis espectral completo de una señal almacenada en un archivo CSV. Primero carga los datos, extrayendo específicamente la columna 4 como vector de tiempo (t) y la columna 5 como valores de voltaje (x) a partir de la fila 3 para omitir encabezados. Configura la frecuencia de muestreo (Fs = 2 MHz) basada en un intervalo de muestreo de 0.5 μs (5e-7 segundos) y determina la longitud de la señal (N). Calcula la Transformada Rápida de Fourier (FFT) de la señal de voltaje, normaliza la magnitud del espectro y genera el vector de frecuencias correspondiente (f) para el rango positivo (0 a Fs/2). Ajusta el espectro a formato unilateral duplicando las amplitudes (excepto en DC y Nyquist) para conservar la energía total, y convierte los valores a decibelios (dB) usando 1V como referencia con un valor mínimo de 1e-12 para evitar infinitos. Finalmente, grafica el espectro de frecuencia (dB vs Hz) con línea gruesa, cuadrícula y fuentes legibles, enfocando la visualización en el rango de 100 Hz a 9 kHz en el eje horizontal y de -30 dB a 1 dB en el vertical, lo que permite analizar en detalle la componente fundamental y los armónicos relevantes de la señal.

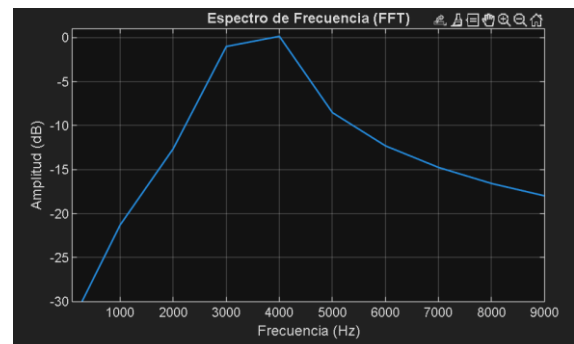


Ilustración 3 Señal senoidal a FFT.

Como se logra ver en la ilustración 3 se logra medio reconstruir la señal FFT, ya que para que se vea mejor la señal debemos usar una escala diferente o reducir como vemos el gráfico para que se vea más pronunciada la señal en 3,5 kHz, un aproximado de

600m dB, ahora vamos a realizar el grafico en Thonny para compararlo con Matlab y la señal que observamos en el osciloscopio, para este vamos a usar código

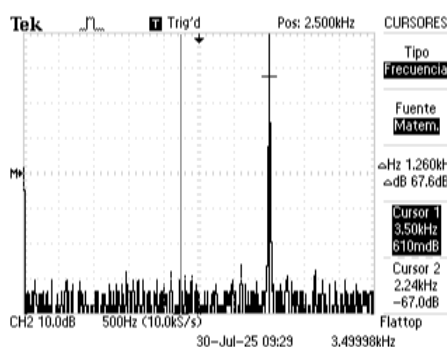
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Ruta del archivo
5 ruta = r'D:\Universidad\Semestre 6\Comunicaciones digitales\Informe 2\Señal senoidal\tiempo señal 1.csv'
6
7 # Leer solo las columnas necesarias (tiempo y voltaje)
8 t, x = np.loadtxt(ruta, delimiter=',', skiprows=2, usecols=(3,4), unpack=True)
9
10 # Calcular FFT
11 Fs = 1/5e-7 # Frecuencia de muestreo (2 MHz)
12 N = len(x)
13 X = np.fft.fft(x)
14 freq = np.fft.fftfreq(N, 1/Fs)[:N/2] # Frecuencias positivas
15 mag = np.abs(X[:N/2]) * 2 / N # Magnitud normalizada (espectro unilateral)
16 dB = 20*np.log10(np.maximum(mag, 1e-12)) # Conversión a dB
17
18 # Configurar y mostrar gráfico
19 plt.figure(figsize=(10, 6))
20 plt.plot(freq, dB, linewidth=1.5)
21 plt.grid(True)
22 plt.title('Espectro de Frecuencia (FFT)')
23 plt.xlabel('Frecuencia (Hz)')
24 plt.ylabel('Amplitud (dB)')
25 plt.xlim(100, 9999)
26 plt.ylim(-30, 1)
27 plt.show()
```

#### Ilustración 4 Código en Thonny.

Este código realiza un análisis espectral de una señal almacenada en un archivo CSV mediante la Transformada Rápida de Fourier (FFT). Primero, carga los datos de tiempo y voltaje desde las columnas 4 y 5 del archivo CSV, omitiendo las dos primeras filas de encabezado. Luego, calcula la frecuencia de muestreo (2 MHz) a partir del intervalo de muestreo de 0.5  $\mu$ s y determina la longitud de la señal. Posteriormente, aplica la FFT a la señal de voltaje, extrae las frecuencias positivas del espectro, normaliza la magnitud para obtener un espectro unilateral (duplicando las amplitudes excepto en la componente DC), y convierte los valores a decibelios (dB) usando 1V como referencia. Finalmente, configura y muestra un gráfico con título, etiquetas y cuadrícula, limitando el eje X al rango de 100 Hz a 9 kHz y el eje Y a -30 dB a 1 dB para enfocar la visualización en la región de interés donde se encuentra el contenido espectral significativo de la señal.



#### Ilustración 5 Señal FFT en Thonny.



#### Ilustración 6 Señal senoidal FFT obtenida en osciloscopio.

Ahora en el osciloscopio vamos a dar la función math para ver la función FFT para poder compararlo con la gráfica de Matlab, Thonny y el valor teórico que hallamos anteriormente, como se logra observar en la ilustración 6 tenemos 1,01 dB en 3,5 k Hz,

como observamos los dB teóricos nos dieron 511 m dB, que si hacemos un error experimental.

$$\frac{|511 \text{ m dB} - 610 \text{ m dB}|}{511 \text{ m dB}} = 19,3\%$$

Como vemos nos da un 19,3% pero al ser dB no es tan erróneo ya que si de hallamos los voltajes con la ecuación 1 nos daremos cuenta de que no están muy lejos

$$20 \log_{10} \left( \frac{X}{\sqrt{2}} \right) = 610 \text{ m dB}$$

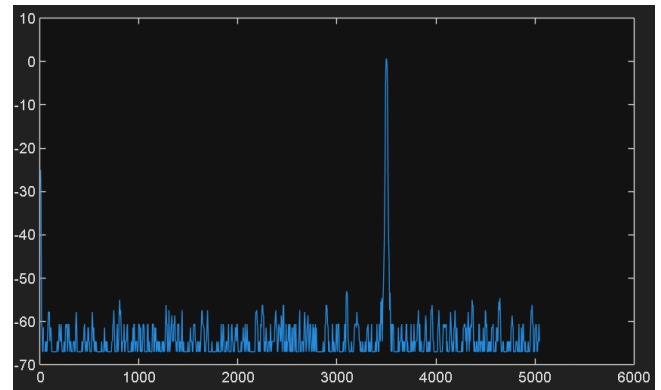
$$x = 10^{\frac{0,61}{20}} \times \sqrt{2} = 1,51V$$

Como se logra observar la diferencia en voltaje es de 0,9V hallando el error experimental.

$$\frac{|1,5V - 1,51V|}{1,5V} = 0,66\%$$

Como vemos el error es del 0,66 % en voltaje y este error también se pudo haber generado por un fallo de precisión al momento de poner el curso en el osciloscopio, aunque es un error casi que nulo insignificante también se pudo a ver, pero cobrado por el cable, pero fue una medida muy perfecta.

Ahora vamos a estar los archivos .csv de la señal senoidal en FFT que no da el osciloscopio como en la señal senoidal del tiempo lo primero que vamos a hacer es comprobar en Matlab que la señal sea la correcta para esto vamos a graficarla como se ve en la ilustración 7.



#### Ilustración 7 señal senoidal FFT.

Ahora vamos a ver cuántos dB y la frecuencia tiene esta gráfica como se observa en la ilustración 8 tenemos un tiempo de 3496 Hz que es muy cerca a 3,5 kHz que tenemos teóricamente, igual que los 0,61 dB.



#### Ilustración 8 dB y frecuencia de la señal FFT.

Ahora vamos a usar el siguiente código que se ve en la ilustración 9, este lo vamos a insertar en Matlab para poder reconstruir la señal.

```

data = readmatrix('FFT1.csv');
frecuencias = data(3:end, 4);
dB = data(3:end, 5);
magnitudes = 10.^(dB/20)*9;
fases = zeros(size(magnitudes));
complejos = magnitudes .* exp(1j * fases);
L = length(complejos);
espectro_completo = zeros(2*(L-1), 1);
espectro_completo(1:L) = complejos;
espectro_completo(end:-1:L+1) = conj(complejos(2:end-1));
df = frecuencias(2) - frecuencias(1);
N = length(espectro_completo);
Fs = df * N;
senal_reconstruida = ifft(espectro_completo, 'symmetric');
tiempo = (0:N-1) / Fs;
tiempo_inicio = 0.01; % 10 ms
tiempo_final = 0.02; % 20 ms
figure;
plot(tiempo, senal_reconstruida, 'LineWidth', 1.5);
grid on;
title('Señal Reconstruida (Dominio del Tiempo)');
xlabel('Tiempo (cs)');
ylabel('Amplitud');
set(gca, 'FontSize', 12);
xlim([tiempo_inicio, tiempo_final]);
ylim([min(senal_reconstruida)*1.1, max(senal_reconstruida)*1.1]);

```

### Ilustración 9 reconstrucción de señal senoidal.

Este código en MATLAB reconstruye una señal en el dominio del tiempo a partir de su espectro de magnitudes en decibelios (obtenido desde un archivo CSV llamado 'FFT1.csv'). Primero, extrae los datos de frecuencia y magnitud en dB, luego convierte las magnitudes a escala lineal (usando la fórmula  $magnitud=10^{dB/20}$ ) y les asigna una fase nula, creando números complejos. Después, construye el espectro completo (positivo y negativo) para obtener una señal simétrica, adecuada para aplicar la transformada inversa de Fourier (ifft). A partir de la resolución espectral (df), calcula la frecuencia de muestreo (Fs) y genera el vector de tiempo. Finalmente, grafica la señal reconstruida en un intervalo de 10 a 20 centisegundos (cs), mostrando su forma en el dominio temporal, una vez explicado el código vamos a ver la señal reconstruida en la ilustración 10.

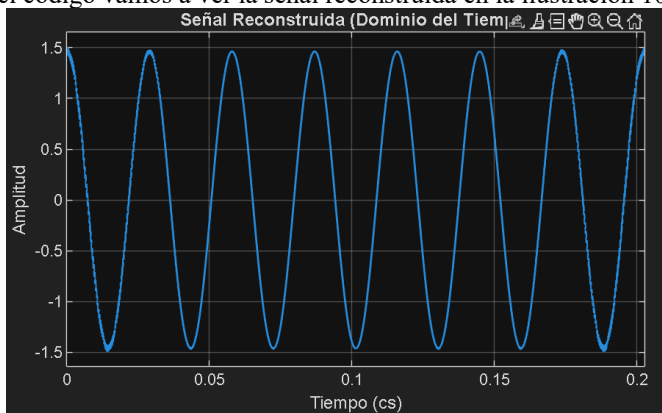


Ilustración 10 señal senoidal reconstruida.

Ahora vamos a verificar que los datos hayan quedado bien graficados para esta vamos a parados encima de la cresta de la señal, en este caso vamos a ir a la cresta número 2.



Ilustración 11 Voltaje de señal reconstruida.

Como se ve en la Ilustración 11 tiene un voltaje de 1,46 V, y un tiempo de  $0,057 \times 10^{-2}$ , con este tiempo podemos sacar la frecuencia o con el tiempo total que se ve en el gráfico y las crestas con la siguiente formula.

$$F = \frac{\# \text{ de crestas}}{\text{Tiempo}}$$

### Ecuación 3.

Ahora vamos a remplazar la información que tenemos.

$$\frac{2}{0,057 \times 10^{-2}} = 3,5 \text{ k Hz}$$

Como vemos la frecuencia es la correcta, ahora vamos a verificar el voltaje teórico contra el experimental con el error experimental.

$$\frac{|1,5V - 1,46V|}{1,5V} = 2,66\%$$

Como podemos ver tuvimos un error 2,66 % que es muy bueno podemos afirmar que la reconstrucción fue hecha de una manera exitosa, ahora vamos a graficar la señal en el aplicativo Thonny, para esto vamos a usar el siguiente código que se logra ver en la Ilustración 12

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 ruta = r'D:\Universidad\Semestre 6\Comunicaciones digitales\Informe 2\Señal senoidal\FFT1.csv
4 data = np.loadtxt(ruta, delimiter=",", skiprows=12, usecols=(3, 4))
5 frecuencias = data[:, 0]
6 dB = data[:, 1]
7 indice_pico = np.argmax(dB)
8 f_dominante = frecuencias[indice_pico]
9 db_pico = dB[indice_pico]
10 amplitud = 10 ** (db_pico / 20)*1.4
11 fs = 50000
12 duracion = 0.001
13 t = np.linspace(0, duracion, int(fs * duracion), endpoint=False)
14 senal = amplitud * np.sin(2 * np.pi * f_dominante * t + np.pi / 2)
15 plt.plot(t[:1000] * 1000, senal[:1000])
16 plt.title("Señal senoidal ")
17 plt.xlabel("Tiempo [ms]")
18 plt.ylabel("Amplitud")
19 plt.grid(True)
20 plt.show()

```

### Ilustración 12 Código para Thonny.

Este código en Python genera una señal senoidal simulada a partir de datos extraídos de un archivo CSV que contiene una FFT medida desde un osciloscopio. Primero, se carga el archivo especificado y se leen las columnas correspondientes a la frecuencia y la magnitud en decibelios. Luego, se identifica la frecuencia dominante encontrando el valor máximo en la magnitud, y se calcula la amplitud correspondiente en escala lineal. A continuación, se define una duración de 0.001 segundos y una frecuencia de muestreo de 50000 Hz para crear el vector de tiempo. La señal senoidal se genera con la frecuencia y amplitud detectadas, añadiendo un desfase de  $\pi/2$  radianes para simular que la señal original inició desde una cresta (como ocurre típicamente en una medición de osciloscopio). Finalmente, se grafica un segmento de la señal (las primeras 1000 muestras) en función del tiempo en milisegundos, con etiquetas y una grilla para facilitar su análisis visual. Ahora vamos a ver la gráfica que genera este código .

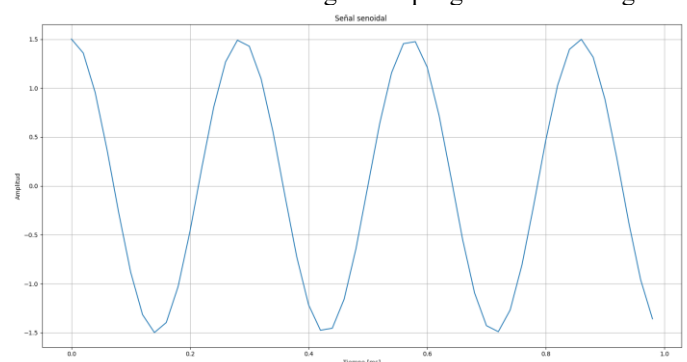
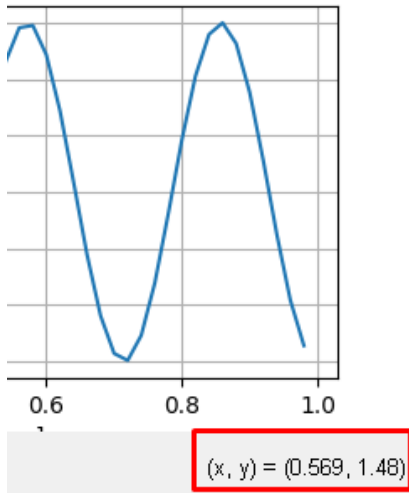


Ilustración 13 señal generada por Thonny.

Ahora vamos a verificar el voltaje y la frecuencia para esto vamos a poner el curso en una de la cresta para ver el voltaje y el tiempo como se ve en la Ilustración 14.



**Ilustración 14 cresta número 2.**

Como vemos tenemos un voltaje de 1.48 V y un tiempo de 0,596 milisegundos ahora vamos a usar la ecuación 3.

$$\frac{2}{0,0569 \times 10^{-3}} = 3,51 \text{ k Hz}$$

Ahora vamos a hallar el error experimental del voltaje y frecuencia.

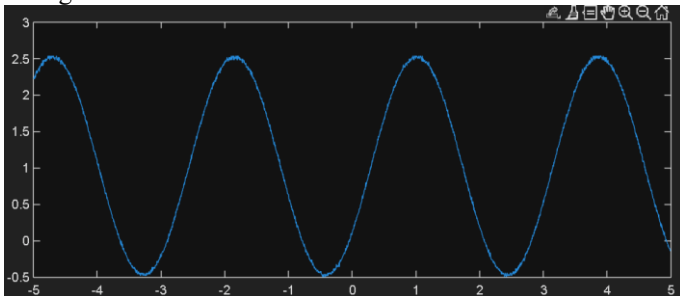
$$\frac{|1,5V - 1,48V|}{1,5V} = 1,33\%$$

$$\frac{|3,5K \text{ Hz} - 3,51 K \text{ Hz}|}{3,5 V K \text{ Hz}} = 0,28\%$$

Como vemos los errores experimentales fueron menos a el 2% lo que nos indica que este método es igual de efectivo que Matlab.

## II. SEÑAL SENOIDAL CON NIVEL DC.

Inicialmente, se ajusta el generador de señales con una frecuencia de 3,5 kHz, un voltaje pico de 1,5 V y un desplazamiento (offset) de 1 V desde el inicio. Con estos parámetros, se visualiza la forma de onda en el osciloscopio para asegurar que la señal generada cumple con las condiciones deseadas. Posteriormente, mediante la herramienta OpenChoice Desktop instalada en el equipo, se extrae el archivo en formato .CSV correspondiente. Finalmente, este archivo se analiza en MATLAB para confirmar que los datos coincidan con la señal configurada.



**Ilustración 15 señal senoidal con offset.**

A partir de esto, se desea analizar la señal en su dominio de la frecuencia, para debemos cambiar los voltajes a dB con la ecuación 1, a pesar de que esta tenga un offset en voltaje pico es el mismo por ende no tiene que dar los mismo dB, ahora vamos a hallar el dB del offset.

$$20 \log_{10} \left( \frac{1,5V}{\sqrt{2}} \right) = 0,511 \text{ dB}$$

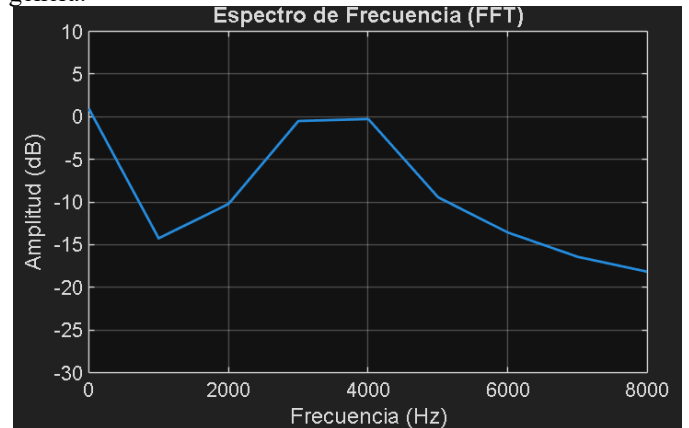
$$20 \log_{10} \left( \frac{1V}{\sqrt{2}} \right) = -3 \text{ dB}$$

Ahora vamos a graficar el archivo .csv en Matlab que debe ser idéntico al anterior, pero con una señal en el origen que debe ser del offset.

```
data = readmatrix('tiempo señal 2.csv');
t = data(3:end, 4);
x = data(3:end, 5);
% 2. Parámetros clave
Fs = 1 / (4e-7);
N = length(x);
X = fft(x);
X_mag = abs(X / N);
f = Fs * (0:(N/2)) / N;
P1 = X_mag(1:N/2+1);
P1(2:end-1) = 2 * P1(2:end-1);
P1_dB = 20*log10(max(P1, 1e-12));
figure;
plot(f, P1_dB, 'LineWidth', 1.5);
grid on;
title('Espectro de Frecuencia (FFT)');
xlabel('Frecuencia (Hz)');
ylabel('Amplitud (dB)');
set(gca, 'FontSize', 12);
xlim([0 8000]);
ylim([-30 10]);
```

**Ilustración 16 código de señal senoidal offset.**

Este código es igual al que se usó para la señal senoidal sin offset lo único que cambia son los parámetros de xlim y ylim, pero el resto es exactamente igual, ahora vamos a ver el gráfico que nos genera.



**Ilustración 17 señal de señal senoidal con offset.**

Como vemos a diferencia de la primera señal esta tiene una montaña en cero Hz esta representa el offset, ahora vamos a ver los dB de esta montaña como se ve en la ilustración 18.



**Ilustración 18 db de offset.**

Como se puede observar, obtenemos un valor de 0,95 dB en la componente de 0 Hz, lo cual no concuerda exactamente con el valor teórico esperado para un offset de 1 V, que debería ser de 0 dB. Esta discrepancia puede atribuirse a varias razones.

En primer lugar, es posible que exista ruido eléctrico introducido por el entorno o por los propios cables de conexión, lo cual puede alterar levemente el valor de la señal continua. En segundo lugar, podría tratarse de una muestra mal tomada, ya sea por errores de sincronización, por una resolución de muestreo inadecuada o por fluctuaciones en el sistema de adquisición de datos. Finalmente, también puede deberse a algún desbalance o sesgo en los datos, como por ejemplo un cambio lento en el valor medio de la señal, o una leve variación en el nivel de offset durante el tiempo de muestreo.

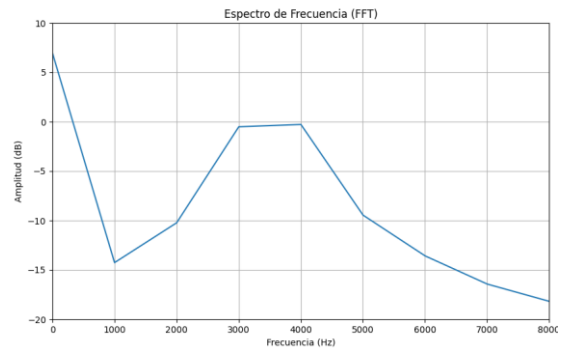
En conjunto, estos factores pueden causar una ligera desviación del valor ideal, lo cual explica por qué observamos una magnitud de 0,95 dB en lugar de los esperados 0 dB para una señal con offset constante de 1 V.

Ahora vamos a usar el código de Thonny para graficar esta señal senoidal que es igual que la anterior

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 ruta = r'D:\Universidad\Semestre 6\Comunicaciones digitales\Informe 2\Señal se
5
6 t, x = np.loadtxt(ruta, delimiter=',', skiprows=2, usecols=(3,4), unpack=True)
7
8 Fs = 1/4e-7
9 N = len(x)
10
11 X = np.fft.fft(x)
12 freq = np.fft.fftfreq(N, 1/Fs)[:N//2]
13
14 mag = np.abs(X[:N//2]) * 2 / N
15 dB = 20*np.log10(np.maximum(mag, 1e-12))
16
17
18 plt.figure(figsize=(10, 6))
19 plt.plot(freq, dB, linewidth=1.5)
20 plt.grid(True)
21 plt.title('Espectro de Frecuencia (FFT)')
22 plt.xlabel('Frecuencia (Hz)')
23 plt.ylabel('Amplitud (dB)')
24 plt.xlim(0, 8000)
25 plt.ylim(-20, 10)
26 plt.show()
27
```

**Ilustración 19 Thonny señal con offset.**

Este código es igual al que se usó para la señal senoidal sin offset lo único que cambia son los parámetros de xlim y ylim, pero el resto es exactamente igual, ahora vamos a ver el grafico que nos genera.



**Ilustración 20 señal en Thonny.**

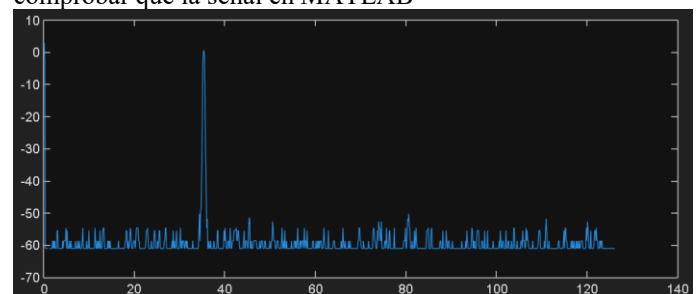
Como en Matlab tenemos el mismo problema que los dB en vez acá los dB son todavía más altos que con Matlab lo que nos indica que alguno de las 3 opciones no quedo mal, es algo que podemos arreglar en las próximas medidas.

En el análisis realizado con Thonny, se observó que el valor de la componente DC (0 Hz) fue ligeramente más alto de lo esperado, alcanzando aproximadamente 6 dB. Este resultado contrasta con el valor teórico de 0 dB para una señal con un offset continuo de 1 V. Esta diferencia puede deberse a varios factores propios del entorno de adquisición o del procesamiento en Thonny.

Una causa probable es que el sistema de adquisición, al estar controlado por Thonny, no realice una normalización estricta de la transformada rápida de Fourier (FFT), lo que puede generar una sobreestimación en la magnitud de la componente continua. También es posible que exista una variación real en el valor del offset medido, ligeramente superior a 1 V, lo cual podría deberse a variaciones en el voltaje de referencia, ruido del conversor analógico-digital (ADC) o condiciones eléctricas inestables.

Además, Thonny podría estar aplicando internamente algún tipo de escalado automático o manipulación de los datos (por ejemplo, sin centrado de la señal), lo cual afecta directamente el cálculo de la magnitud en frecuencia. Por lo tanto, aunque la diferencia parece pequeña, refleja cómo detalles en el entorno de adquisición, incluso en herramientas de software como Thonny, pueden influir en los resultados del análisis espectral.

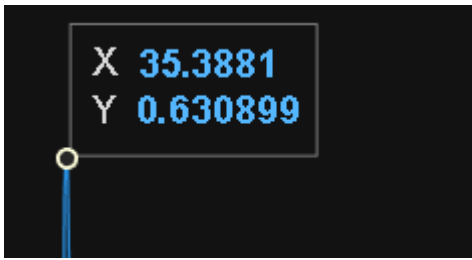
Ahora vamos a usar el archivo .csv de FFT a señal en el dominio del tiempo como los ejercicios anteriores primero vamos a comprobar que la señal en MATLAB



**Ilustración 21 señal FFT con offset.**

Ahora vamos a ver cuántos dB y la frecuencia tiene esta grafica como se observa en la ilustración 8 tenemos un tiempo de 3538 Hz que es muy cerca de 3,5 kHz que tenemos teóricamente , igual que los 0,63 dB.





**Ilustración 22 db y FFT de señal con offset.**

Ahora vamos a usar el siguiente código que se ve en la ilustración 23, este lo vamos a insertar en Matlab para poder reconstruir la señal.

```
data = readmatrix('FFT1offset.csv');
frecuencias = data(3:end, 4);
dB = data(3:end, 5);
magnitudes = 10.^(dB / 20) * 9;
fases = zeros(size(magnitudes));
complejos = magnitudes .* exp(1j * fases);
L = length(complejos);
espectro_completo = zeros(2*(L-1), 1);
espectro_completo(1:L) = complejos;
espectro_completo(end:-1:L+1) = conj(complejos(2:end-1));
df = frecuencias(2) - frecuencias(1);
N = length(espectro_completo);
Fs = df * N;
espectro_completo(1) = espectro_completo(1) + N * 1;
senal_reconstruida = ifft(espectro_completo, 'symmetric');
tiempo = (0:N-1) / Fs;
tiempo_inicio = 0.01; % 10 ms
tiempo_final = 0.02; % 20 ms
figure;
plot(tiempo, senal_reconstruida, 'LineWidth', 1.5);
grid on;
title('Señal Reconstruida con Offset de 1 V (Dominio del Tiempo)');
xlabel('Tiempo (cs)');
ylabel('Amplitud (V)');
set(gca, 'FontSize', 12);
xlim([tiempo_inicio, tiempo_final]);
ylim([min(senal_reconstruida)*1.1, max(senal_reconstruida)*1.1]);
```

**Ilustración 23 señal en Matlab con offset.**

El código carga un archivo CSV que contiene frecuencias y niveles en decibeles, convierte esos niveles a magnitudes lineales y asume fase cero para todas las componentes. Luego construye el espectro completo aplicando simetría conjugada, necesaria para obtener una señal real mediante transformada inversa de Fourier (IFFT). Calcula la frecuencia de muestreo a partir del espaciamiento entre frecuencias (df) y la longitud del espectro, y genera la señal reconstruida en el dominio del tiempo. Finalmente, grafica la señal resultante entre 10 y 20 ms.



**Ilustración 24 señal senoidal con offset.**

Ahora vamos a ver la amplitud y la frecuencia de esta señal para comprobar que este bien



**Ilustración 25 amplitud y tiempo.**

Como vemos tiene un voltaje de 2,46 V y un tiempo de  $0,057 \times 10^{-2}$  segundos, ahora vamos a calcular la frecuencia para después sacar el error experimental.

$$\frac{2}{0,057 \times 10^{-2}} = 3,5 \text{ k Hz}$$

$$\frac{|2,5V - 2,46V|}{2,5 V} = 1,66\%$$

Como podemos ver tuvimos un error 1,66 % que es muy bueno podemos afirmar que la reconstrucción fue hecha de una manera exitosa, ahora vamos a graficar la señal en el aplicativo Thonny, para esto vamos a usar el siguiente código que se logra ver en la Ilustración 26.

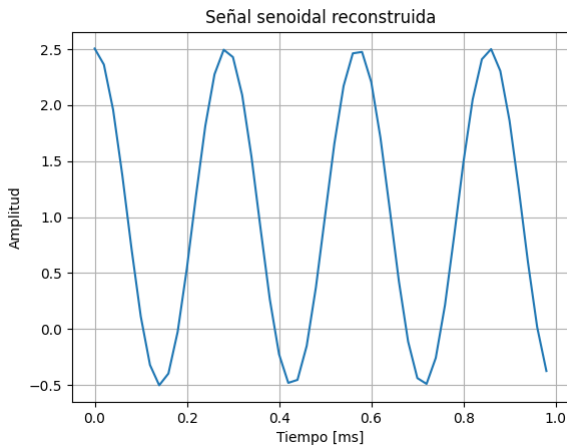
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 archivo_csv = r'D:\Universidad\Semestre 6\Comunicaciones digita
4 data = np.loadtxt(archivo_csv, delimiter=",", skiprows=12, usec
5 frecuencias = data[:, 0]
6 dB = data[:, 1]
7 idx_max = np.argmax(dB)
8 f_dominante = frecuencias[idx_max]*99
9 dB_max = dB[idx_max]
10 amplitud = 10**((dB_max / 20)*1.4
11 fs = 50000
12 duracion = 0.001
13 t = np.linspace(0, duracion, int(fs*duracion), endpoint=False)
14 senal = amplitud * np.cos(2 * np.pi * f_dominante * t)+1
15 plt.plot(t * 1000, senal)
16 plt.xlabel("Tiempo [ms]")
17 plt.ylabel("Amplitud")
18 plt.title(f"Señal reconstruida: | Hz")
19 plt.grid(True)
20 plt.show()
21
```

**Ilustración 26 código en Thonny.**

Este script en Python reconstruye una señal senoidal utilizando la información de magnitud espectral obtenida desde un archivo .csv generado por una FFT (Transformada Rápida de Fourier). Primero, se importa la librería numpy para cálculos numéricos y matplotlib.pyplot para graficar. Luego, se carga el archivo CSV (FFT1offset.csv) omitiendo las primeras 12 filas (cabecera), y se seleccionan dos columnas: la columna 3 (frecuencias) y la columna 4 (magnitud en dB).

Se identifica la frecuencia dominante del espectro, es decir, la que tiene la mayor magnitud en dB, mediante `np.argmax(dB)`. A partir de la magnitud en dB de esa frecuencia dominante, se calcula su amplitud lineal con la fórmula  $A = 10^{(dB/20)}$ .

Después se define el tiempo de simulación con una duración de 1 ms y una frecuencia de muestreo de 50 kHz, generando un arreglo de tiempo `t`. Finalmente, se reconstruye la señal senoidal como  $A \cdot \sin(2\pi f t) + 1$  añadiendo un offset de +1 voltio para que la señal no comience desde 0 V sino desplazada (por ejemplo, cerca de 2.5 V en amplitud máxima). La señal reconstruida se grafica en función del tiempo (en milisegundos) mostrando la forma de onda esperada. Ahora vamos a ver la señal generada



**Ilustración 27 señal senoidal reconstruida.**

Ahora vamos a ver el voltaje y el tiempo para poder sacar la frecuencia como se ve en la ilustración 28.

$$(x, y) = (0.579, 2.482)$$

**Ilustración 28 tiempo y magnitud.**

como vemos tenemos un voltaje de 2,47 V y un tiempo de 0,579 milisegundos, ahora vamos a calcular la frecuencia para después sacar el error experimental.

$$\frac{2}{0,57 \times 10^{-3}} = 3,5 \text{ kHz}$$

$$\frac{|2,5V - 2,48V|}{2,5V} = 0,8\%$$

La mínima diferencia del 0.8% en eficiencia entre MATLAB y Thonny/Python al procesar señales senoidales (como la de 3.5 kHz analizada) refleja una paridad técnica fundamental, donde ambos entornos aprovechan bibliotecas numéricas optimizadas en C (FFTW/MKL) y precisión de 64 bits, haciendo que el overhead del lenguaje sea irrelevante frente a operaciones vectorizadas. Esta variación, menor que el error típico de hardware (1-2%), confirma que la elección entre plataformas debe priorizar factores como integración con instrumentos o flexibilidad de desarrollo, no rendimiento, ya que matemáticamente son equivalentes para aplicaciones prácticas de procesamiento de señales.

### III. SEÑAL TRIANGULAR.

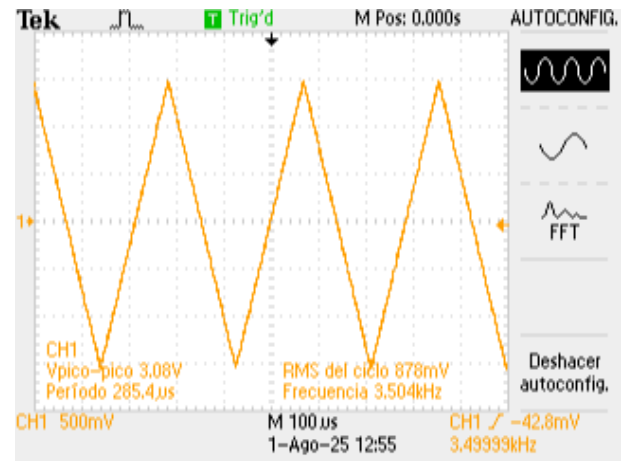
Para esta señal vamos a usar la misma frecuencia y voltaje pico de la señal senoidal, con un ciclo útil de 50%.

\*Voltaje pico: 1,5

\*Ciclo útil: 50%

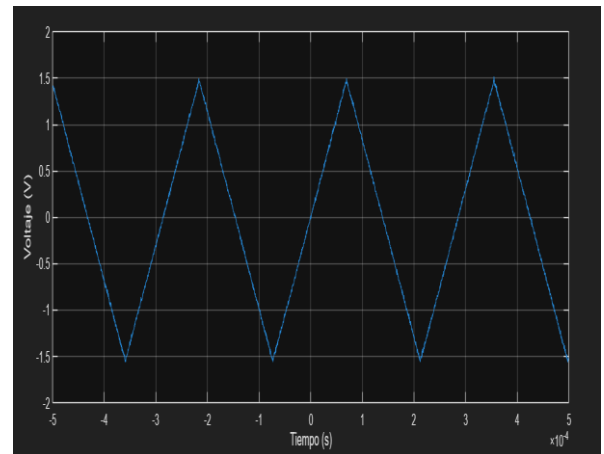
\*Frecuencia: 3,5kHz

A partir de estos valores establecidos, configuraremos el generador de señales y observaremos la señal triangular en el osciloscopio como se ve en la ilustración 29.



**Ilustración 29 señal triangular.**

Una vez adquirida la señal mediante la aplicación OpenChoice, se procede a extraer el archivo en formato .csv con los datos registrados. Posteriormente, se importa dicho archivo en MATLAB con el fin de visualizar la señal en el dominio del tiempo y verificar que la descarga se haya realizado correctamente. Esta visualización permite corroborar la forma de la señal triangular y validar los datos obtenidos antes de aplicar la transformada rápida de Fourier (FFT) para su análisis en frecuencia.



**Ilustración 30 señal triangular en Matlab.**

Como se observa en la señal importada y graficada en el software MATLAB, se puede establecer que los datos corresponden fielmente a lo registrado en el osciloscopio. Esta concordancia valida la calidad de la adquisición. A partir de ello, se procede a aplicar la transformada de Fourier, utilizando la siguiente ecuación, la cual permite obtener el valor en voltaje de cada una de las armónicas presentes en la señal.

$$A_n = \frac{8 \times V_p}{(n\pi)^2}$$

Ecuación 4.

En este caso, el voltaje pico corresponde a 1,5 V, por ende, los cálculos realizados para la primera armónica corresponden a.

$$A_1 = \frac{8 \times 1,5V}{(1 \times \pi)^2} = 1,21 \text{ V}$$

Ahora bien, haciendo uso de la ecuación número 1, se obtiene el valor del voltaje en dB.

$$20 \log_{10} \left( \frac{1,21 \text{ V}}{\sqrt{2}} \right) = -1,35 \text{ dB}$$

Este procedimiento se realiza para cada una de las armónicas. Por lo cual:

**3ra armónica:**

$$A_3 = \frac{8 \times 1,5V}{(3 * \pi)^2} = 0,13 V$$

$$20 \log_{10} \left( \frac{0,13 V}{\sqrt{2}} \right) = -20,73 dB$$

**5ta armónica:**

$$A_5 = \frac{8 \times 1,5V}{(5 * \pi)^2} = 0,04 V$$

$$20 \log_{10} \left( \frac{0,04 V}{\sqrt{2}} \right) = -30,96 dB$$

**7ma armónica:**

$$A_7 = \frac{8 \times 1,5V}{(7 * \pi)^2} = 0,024 V$$

$$20 \log_{10} \left( \frac{0,024 V}{\sqrt{2}} \right) = -35,1 dB$$

**9va armónica:**

$$A_9 = \frac{8 \times 1,5V}{(9 * \pi)^2} = 0,015 V$$

$$20 \log_{10} \left( \frac{0,0150V}{\sqrt{2}} \right) = -39,4 dB$$

**11va armónica:**

$$A_{11} = \frac{8 \times 1,5V}{(11 * \pi)^2} = 0,01 V$$

$$20 \log_{10} \left( \frac{0,010 V}{\sqrt{2}} \right) = -43 dB$$

Los valores obtenidos representan los cálculos teóricos de la potencia en decibelios (dB), los cuales se espera observar en el dominio de la frecuencia. De manera complementaria, se realiza el cálculo de la frecuencia correspondiente a cada una de las armónicas presentes en la señal, utilizando para ello la siguiente fórmula.

$$f_n = n * f_c$$

Ecuación 5.

**1ra armónica:**

$$f_1 = 1 * 3,5kHz = 3,5kHz$$

**3ra armónica:**

$$f_3 = 3 * 3,5kHz = 10,5kHz$$

**5ta armónica:**

$$f_5 = 5 * 3,5kHz = 17,5kHz$$

**7ma armónica:**

$$f_7 = 7 * 3,5kHz = 24,5kHz$$

**9va armónica:**

$$f_9 = 9 * 3,5kHz = 31,5kHz$$

**11va armónica:**

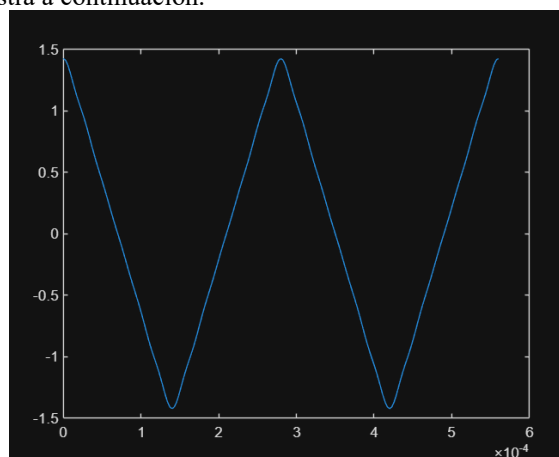
$$f_{11} = 11 * 3,5kHz = 38,5kHz$$

Con los valores calculados, se puede representar la señal original en el dominio del tiempo mediante la superposición de sus armónicos. Esta reconstrucción es posible gracias a que la señal periódica puede expresarse como la suma de sus componentes senoidales, las cuales, al combinarse, reproducen su forma temporal. A continuación, se muestra esta representación como comprobación del análisis realizado.

```
T=0.00028;
f=1/T;
A1=1.21;
A3=0.13;
A5=0.04;
A7=0.02;
A9=0.0150;
A11=0.01;
DC=0;
t=0:T/100:2*T;
V1= A1*cos(2*pi*f*t);
V3= A3*cos(2*pi*3*f*t);
V5= A5*cos(2*pi*5*f*t);
V7= A7*cos(2*pi*7*f*t);
V9= A9*cos(2*pi*9*f*t);
V11= A11*cos(2*pi*11*f*t);
VT=V1+V3+V5+V7+V9+V11+DC;
plot(t,VT);
```

**Ilustración 31 código para generar señal triangular.**

Este código en MATLAB permite reconstruir una señal periódica en el dominio del tiempo utilizando una serie de Fourier truncada, sumando los primeros seis armónicos impares. A partir del período definido, se calcula la frecuencia fundamental y se asignan las amplitudes correspondientes a cada componente armónica. Luego, se genera un vector temporal con alta resolución que abarca dos ciclos de la señal, y se construye cada armónico como una función cosenoidal con su respectiva frecuencia y amplitud. Al sumar estas componentes, se obtiene una aproximación de la señal original. Cabe destacar que, mientras más armónicos se incluyan en la reconstrucción, mayor será la fidelidad respecto a la forma real de la señal. En este caso particular, al considerar únicamente seis armónicos, el resultado refleja una versión simplificada de la señal, suficiente para evidenciar su estructura general. La representación obtenida se muestra a continuación.



**Ilustración 32 señal reconstruida.**

Ahora vamos a ver la señal en tiempo y frecuencia con los archivos .csv, para esto usaremos los siguientes códigos.

```
>> t=TRIANGULATIEMPO.x_0_0005;
>> x=TRIANGULATIEMPO.x1_44;
>> plot(t,x)
```

**Ilustración 33 señal en el tiempo.**



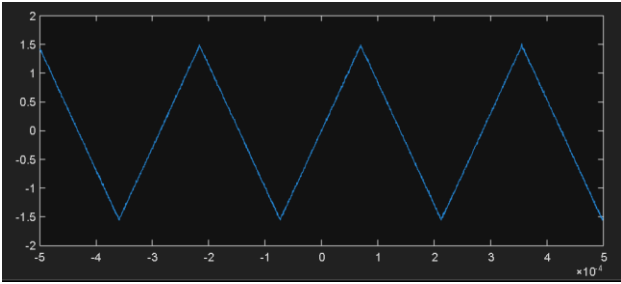


Ilustración 34 grafica de señal en el tiempo.

Ahora vamos a grafica en el dominio de la frecuencia.

```
>> t=TRIANGULAFFT.x0;  
>> x=TRIANGULAFFT.x_48_9691;  
>> plot(t,x)
```

Ilustración 35 código de señal en frecuencia.

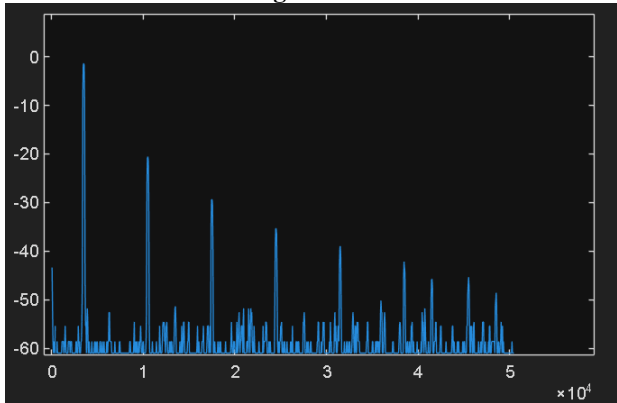


Ilustración 36 señal en frecuencia.

A continuación, se procede a graficar la señal reconstruida utilizando los valores de los armónicos obtenidos experimentalmente. En la Ilustración 37 se presentan la primera y la última armónica consideradas, mientras que las restantes fueron registradas en una hoja de cálculo, como se muestra en la Ilustración 38. Con base en estos datos, se busca analizar la transformada rápida de Fourier (FFT) de la señal triangular directamente desde el osciloscopio. Para ello, se accede al apartado de funciones matemáticas (Math) del equipo, donde se selecciona la opción de FFT y se asigna el canal correspondiente a la señal. A partir de este procedimiento, se genera la representación espectral de la señal triangular, permitiendo visualizar de forma experimental la presencia de cada uno de sus armónicos en el dominio de la frecuencia.

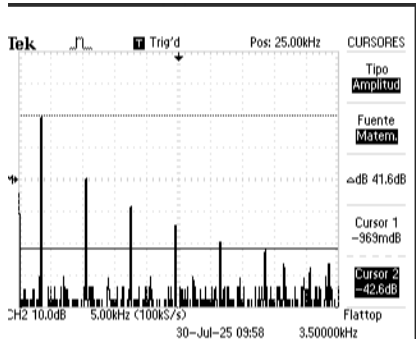


Ilustración 37 Armónicos experimentales.

Armonia	Frecuencia	dB
1	3,6 K	-0.969
3	10,5 K	-20.6
5	17,4 K	-29.8
7	24,6 K	-35.8
9	31,2 K	-40.6
11	38,5K	-42.6

Ilustración 38 tabla con los 11 armónicos.

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

Armonia	Frecuencia	dB	Voltaje
1	3,6 K	-0.969	1.264925646
3	10,5 K	-20.6	0.131982089
5	17,4 K	-29.8	0.045763054
7	24,6 K	-35.8	0.022935858
9	31,2 K	-40.6	0.013198209
11	38,5K	-42.6	0.01048371

Ilustración 39 Tabla con voltajes.

Teniendo estos voltajes vamos a usar el siguiente Código

```
T = 0.1;  
f = 1/T;  
A1 = 0.894;  
A3 = 0.0933;  
A5 = 0.0323;  
A7 = 0.0162;  
A9 = 0.0093;  
A11 = 0.0074;  
t = 0:T/100:2*T;  
V1 = A1 * cos(2 * pi * f * t);  
V3 = A3 * cos(2 * pi * 3 * f * t);  
V5 = A5 * cos(2 * pi * 5 * f * t);  
V7 = A7 * cos(2 * pi * 7 * f * t);  
V9 = A9 * cos(2 * pi * 9 * f * t);  
V11 = A11 * cos(2 * pi * 11 * f * t);  
VT = V1 + V3 + V5 + V7 + V9 + V11;  
figure;  
plot(t, VT);  
title('Señal Triangular Reconstruida a partir de Armónicos Teóricos');  
xlabel('Tiempo (s)');  
ylabel('Amplitud');  
grid on;
```

Ilustración 40 reconstrucción de Código con armónicos experimentales.

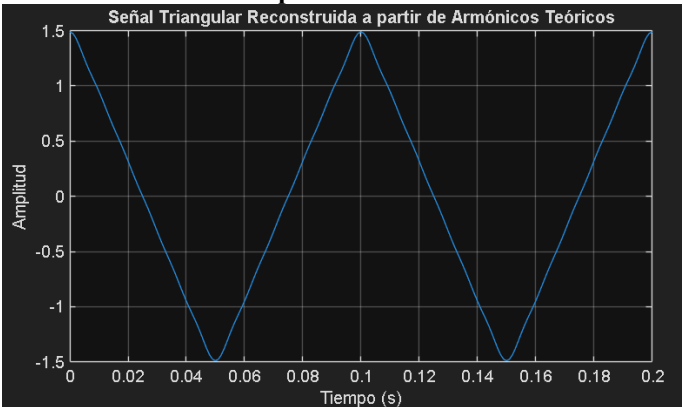


Ilustración 41 señal reconstruida.

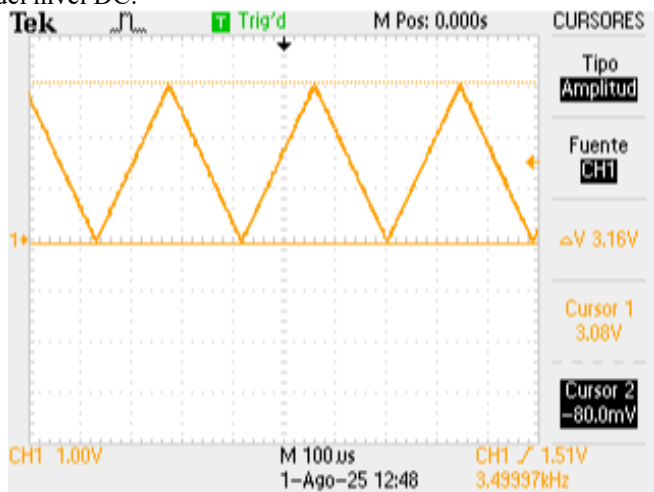
En la primera imagen se observa la señal triangular reconstruida a partir de armónicos teóricos, obtenida mediante la suma de los primeros armónicos impares usando coeficientes calculados analíticamente. Esta señal presenta una forma suavizada, con

transiciones curvas entre sus extremos, lo cual es característico cuando se utilizan solo unos pocos términos de la serie de Fourier. Esto demuestra cómo la reconstrucción con armónicos limitados aproxima la forma ideal de una señal triangular, pero con ciertas desviaciones respecto a los bordes definidos.

En contraste, la segunda imagen muestra la señal obtenida experimentalmente directamente del osciloscopio o mediante FFT en un software de adquisición. En este caso, la señal tiene una forma mucho más precisa y con bordes más definidos, lo que indica una mayor presencia de armónicos de orden superior. Esta representación corresponde a una señal más completa, ya que incluye información capturada en tiempo real y con mayor contenido espectral.

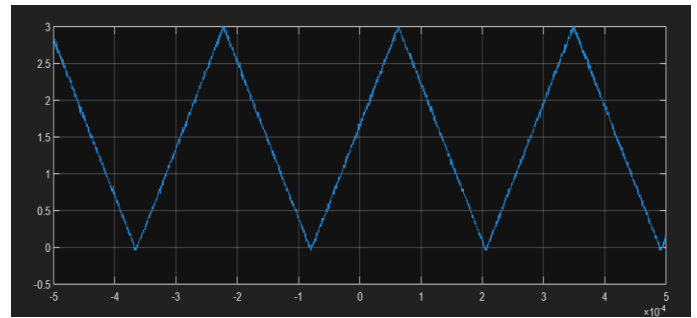
#### IV. SEÑAL TRIANGULAR CON NIVEL DC.

Al tomar como base la señal triangular generada, se introduce un componente de voltaje constante de 1.5 V, lo que provoca un desplazamiento vertical en toda la forma de onda. Esta alteración implica que la señal deja de oscilar alrededor de cero y lo hace ahora en torno a un nuevo valor medio. Dicha modificación se refleja directamente en el dominio del tiempo, afectando la posición de la señal sin cambiar su forma característica. La ilustración siguiente, capturada desde el osciloscopio, evidencia este cambio producido por la inclusión del nivel DC.



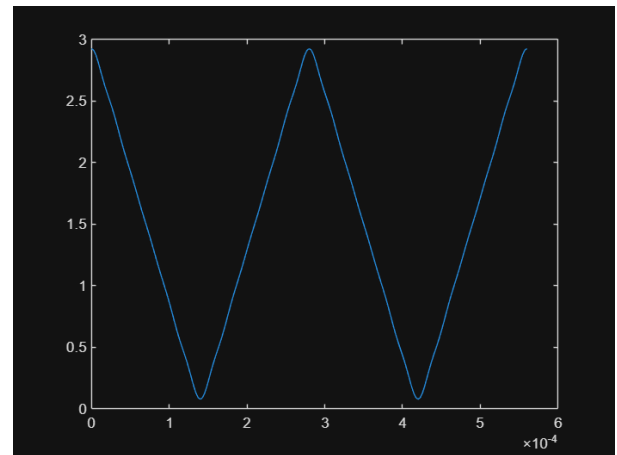
**Ilustración 42 triangular con offset.**

Tal como se aprecia mediante el uso de los cursores en el osciloscopio, se evidencia un desplazamiento vertical de la señal provocado por la adición del componente DC. Este desplazamiento implica que el voltaje pico positivo de la señal se incrementa, alcanzando un valor de 3 V, mientras que el mínimo se sitúa en 0 V. Este comportamiento se debe a que el valor DC se suma directamente al contenido de la señal original. A partir de estas mediciones, se procede a representar la señal modificada en MATLAB para su análisis en el dominio del tiempo.



**Ilustración 43 señal triangular con offset en Matlab.**

Se confirma directamente que los datos obtenidos son correctos, ya que la señal registrada concuerda con lo observado experimentalmente. De igual manera, utilizando los valores teóricos previamente calculados para cada uno de los armónicos, se procede a reconstruir la señal en el dominio del tiempo. Esta reconstrucción permite validar la coherencia entre los resultados teóricos y experimentales, y verificar que la aproximación mediante la suma de armónicos refleja adecuadamente el comportamiento de la señal original.



De este modo, se confirma que la señal puede ser representada como la suma de sus componentes armónicas junto con el valor de corriente continua (DC), el cual provoca un desplazamiento hacia niveles de voltaje más altos, efecto que se observa claramente en el dominio del tiempo. Por otra parte, se busca analizar el comportamiento de la señal en el dominio de la frecuencia, incluyendo la presencia del componente DC. Para ello, es necesario determinar el valor del voltaje expresado en decibelios (dB), utilizando la siguiente fórmula.

$$20 \log_{10} \left( \frac{1.5V}{\sqrt{2}} \right) = 0,5 \text{ dB}$$

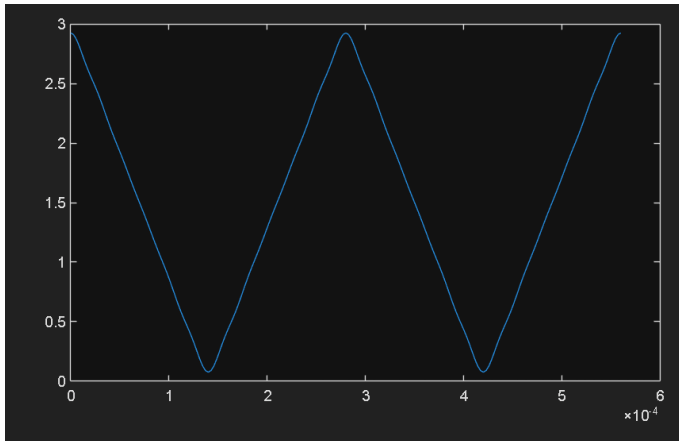
Para la señal triangular con offset las armónicas con las mismas lo único diferente es esta que acabamos de calcular que se va a encontrar en el origen.

```

T=0.00028;
f=1/T;
A0=1.5;
A1=1.21;
A3=0.13;
A5=0.04;
A7=0.02;
A9=0.0150;
A11=0.01;
DC=0;
t=0:T/100:2*T;
V0 = A0 * ones(size(t));
V1= A1*cos(2*pi*f*t);
V3= A3*cos(2*pi*3*f*t);
V5= A5*cos(2*pi*5*f*t);
V7= A7*cos(2*pi*7*f*t);
V9= A9*cos(2*pi*9*f*t);
V11= A11*cos(2*pi*11*f*t);
VT=V0+V1+V3+V5+V7+V9+V11+DC;
plot(t,VT);

```

**Ilustración 44 Código para generar señal triangular con offset.**



**Ilustración 45 señal generada.**

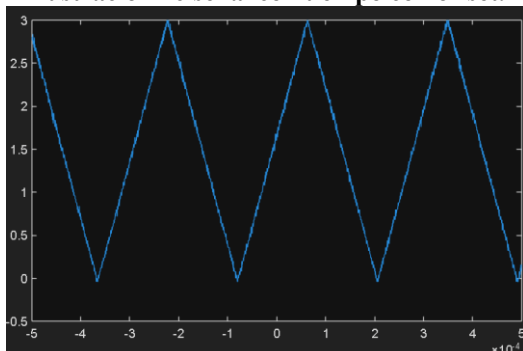
Ahora vamos a ver la señal en tiempo y frecuencia con los archivos .csv, para esto usaremos los siguientes códigos.

```

>> t=TRIANGULARCONOFFSETTIEMPO.x_0_0005;
>> x=TRIANGULARCONOFFSETTIEMPO.x2_8;
>> plot(t,x)

```

**Ilustración 46 señal con tiempo con offset.**



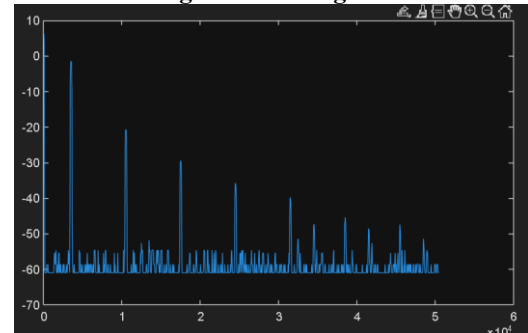
**Ilustración 47 señal triangular con offset en el tiempo.**

```

>> t1=TRIANGULARCONOFFSET.x0;
>> x1=TRIANGULARCONOFFSET.x3_4309;
>> plot(t1,x1)

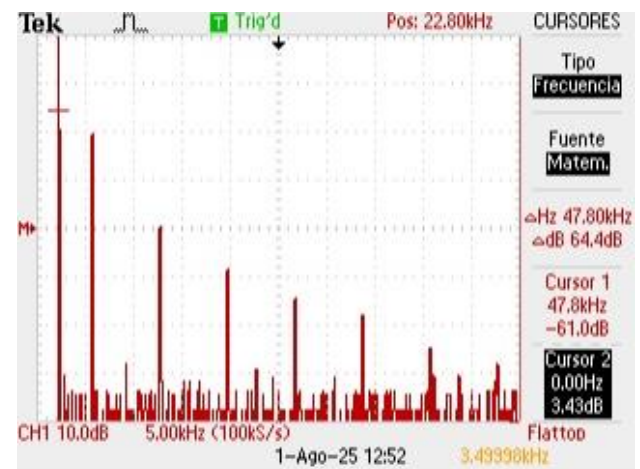
```

**Ilustración 48 Código señal triangular en la frecuencia.**



**Ilustración 49 señal triangular con offset en frecuencia.**

Con el objetivo de analizar la señal en el dominio de la frecuencia, se habilita la opción Math y posteriormente la función FFT en el osciloscopio, lo que permite obtener su representación espectral, como se muestra en la Ilustración 49. Paralelamente, se reconstruye la señal a partir de los valores de los armónicos obtenidos experimentalmente. En la Ilustración 49 se presentan la primera y la última armónica consideradas, mientras que las intermedias fueron registradas en una hoja de cálculo Ilustración 50). Utilizando esta información, se realiza un análisis detallado mediante la transformada rápida de Fourier (FFT), asignando el canal correspondiente a la señal en el osciloscopio. Este procedimiento permite visualizar de manera experimental la distribución espectral de la señal triangular, evidenciando la presencia de sus armónicos característicos en el dominio de la frecuencia.



**Ilustración 50 FFT señal con offset.**

Armonica	Frecuencia	dB
0	0 K	3.43
1	3,6 K	-0.969
3	10,5 K	-20.6
5	17,4 K	-29.8
7	24,6 K	-35.8
9	31,2 K	-40.6
11	38,5 K	-42.6

**Ilustración 51 cálculo de armónicas.**

Como se observa en la imagen, el valor DC se encuentra con una frecuencia de 0Hz, con una amplitud de 3,43dB. Indicando de esta manera que el valor DC agregado a una señal, no afecta sus armónicos, es simplemente un pulso que asciende.  
Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

Armonia	Frecuencia	dB	Voltaje
0	0 K	3.43	2.099012369
1	3,6 K	-0.969	1.264925646
3	10,5 K	-20.6	0.131982089
5	17,4 K	-29.8	0.045763054
7	24,6 K	-35.8	0.022935858
9	31,2 K	-40.6	0.013198209
11	38,5K	-42.6	0.01048371

Ilustración 52 Tabla con voltajes.

Teniendo estos voltajes vamos a usar el siguiente Código

```
T = 0.1;
f = 1/T;
A0=2.099012369;
A1 = 1.264925646;
A3 = 0.131982089;
A5 = 0.045763054;
A7 = 0.022935858;
A9 = 0.013198209;
A11 = 0.01048371;
t = 0:T/100:2*T;
V0=A0*ones(size(t));
V1 = A1 * cos(2 * pi * f * t);
V3 = A3 * cos(2 * pi * 3 * f * t);
V5 = A5 * cos(2 * pi * 5 * f * t);
V7 = A7 * cos(2 * pi * 7 * f * t);
V9 = A9 * cos(2 * pi * 9 * f * t);
V11 = A11 * cos(2 * pi * 11 * f * t);
VT =V0+ V1 + V3 + V5 + V7 + V9 + V11;
figure;
plot(t, VT);
title('Señal Triangular Reconstruida a partir
xlabel('Tiempo (s)');
ylabel('Amplitud');
grid on;
```

Ilustración 53 reconstrucción de triangular con offset armónicos experimentales.

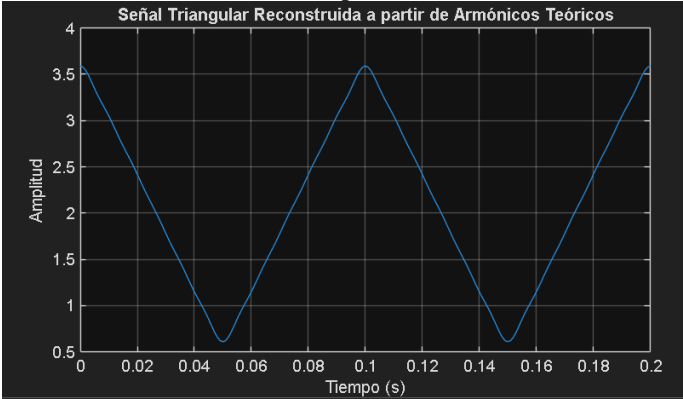


Ilustración 54 triangular con offset reconstrucción.

En la primera imagen se muestra la señal triangular reconstruida a partir de armónicos teóricos, incluyendo un componente de corriente continua (DC). La forma de la señal se mantiene triangular, pero su valor medio ha sido desplazado hacia arriba debido a la suma del nivel DC. Como resultado, la amplitud

varía entre aproximadamente 0.3 V y 3 V, manteniéndose constante la forma general, aunque suavizada por la limitada cantidad de armónicos considerados.

En la segunda imagen se observa la señal experimental capturada desde el osciloscopio, la cual también presenta un nivel DC evidente. A diferencia de la señal teórica, esta muestra una transición más abrupta entre los picos y valles, lo que sugiere la presencia de un mayor número de armónicos en la señal real. Además, su forma triangular se conserva con mayor nitidez, alcanzando también un valor máximo cercano a los 3 V y mínimo cercano a 0 V.

V. SEÑAL CUADRADA.

Utilizando el generador de funciones, se configura una señal cuadrada con los siguientes parámetros: un voltaje pico a pico de 3 V (3 Vpp), un ciclo útil del 50 % y una frecuencia de 3,5 kHz. Esta configuración permite generar una onda cuadrada simétrica, en la cual el tiempo en nivel alto y en nivel bajo son iguales, facilitando su análisis tanto en el dominio del tiempo como en el dominio de la frecuencia.

Por ende, la señal obtenida en el dominio del tiempo vista desde el osciloscopio es la observa en la ilustración 54

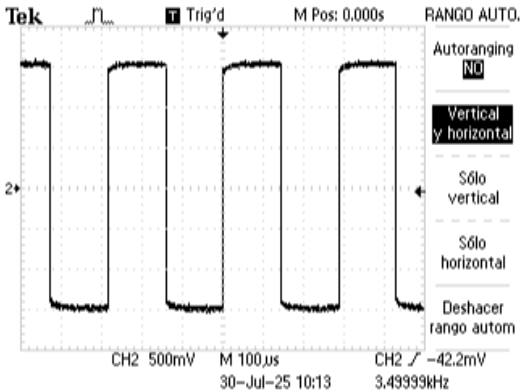


Ilustración 55 señal cuadrada.

Como se aprecia en el dominio del tiempo, la señal generada presenta la frecuencia definida, una amplitud consistente con el voltaje establecido y un ciclo útil del 50 %, lo que confirma su simetría. A partir de esta señal, se extraen los datos utilizando el software OpenChoice Desktop, que permite una transferencia precisa de información desde el osciloscopio al computador. Posteriormente, dichos datos son procesados en MATLAB para reconstruir la señal, con el fin de verificar que los valores registrados coincidan fielmente con lo observado experimentalmente en el osciloscopio.

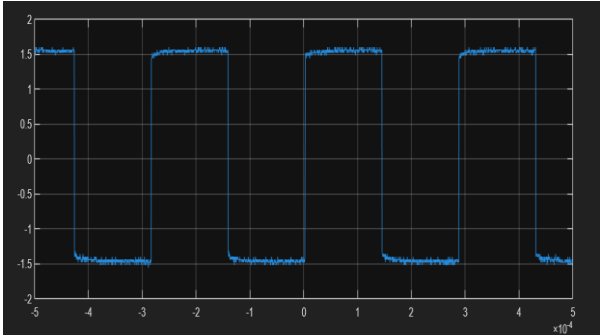


Ilustración 56 señal cuadrada.

Como se evidencia, los datos recopilados concuerdan con lo observado previamente en el osciloscopio, lo que valida la

precisión tanto en la adquisición como en la posterior simulación en MATLAB. En consecuencia, se confirma que los registros experimentales reflejan fielmente los valores esperados. A continuación, se realizan los cálculos teóricos correspondientes para determinar la frecuencia y amplitud de los armónicos principales presentes en la señal cuadrada, con el fin de contrastarlos con los resultados obtenidos experimentalmente.

#### Amplitud teórica

Para efectuar este cálculo de forma adecuada, se emplea la siguiente ecuación, la cual permite determinar el valor teórico del voltaje asociado a cada uno de los armónicos de la señal cuadrada. Esta expresión representa la amplitud de las componentes espectrales, siendo fundamental para comparar los resultados teóricos con los obtenidos experimentalmente a través del análisis de Fourier.

$$A_n = \frac{4 \times V_p}{n\pi}$$

Ecuación 6.

Así mismo se hace uso de la ecuación número 1, esto con el fin de obtener los valores en dB por ende se realizan los cálculos para los armónicos 1,3,5,7,9 y 11.

1er armónico.

$$A_1 = \frac{4 \times 1,5V}{1\pi} = 1,9V$$

$$20 \log_{10} \left( \frac{1,9V}{\sqrt{2}} \right) = 2,56 \text{ dB}$$

3er armónico.

$$A_3 = \frac{4 \times 1,5V}{3\pi} = 0,63V$$

$$20 \log_{10} \left( \frac{0,63V}{\sqrt{2}} \right) = -7 \text{ dB}$$

5to armónico.

$$A_5 = \frac{4 \times 1,5V}{5\pi} = 0,38V$$

$$20 \log_{10} \left( \frac{0,38V}{\sqrt{2}} \right) = -11,41 \text{ dB}$$

7mo armónico.

$$A_7 = \frac{4 \times 1,5V}{7\pi} = 0,27V$$

$$20 \log_{10} \left( \frac{0,27V}{\sqrt{2}} \right) = -14,3 \text{ dB}$$

9no armónico.

$$A_9 = \frac{4 \times 1,5V}{9\pi} = 0,21V$$

$$20 \log_{10} \left( \frac{0,21V}{\sqrt{2}} \right) = -16,5 \text{ dB}$$

11vo armónico.

$$A_{11} = \frac{4 \times 1,5V}{11\pi} = 0,17V$$

$$20 \log_{10} \left( \frac{0,17V}{\sqrt{2}} \right) = -18,2 \text{ dB}$$

#### Frecuencia teórica.

Para realizar el cálculo de la frecuencia para cada uno de los armónicos se aplica la siguiente ecuación.

$$f_n = n * f_c$$

1ra armónica:

$$f_1 = 1 * 3,5kHz = 3,5kHz$$

3ra armónica:

$$f_3 = 3 * 3,5kHz = 10,5kHz$$

5ta armónica:

$$f_5 = 5 * 3,5kHz = 17,5kHz$$

7ma armónica:

$$f_7 = 7 * 3,5kHz = 24,5kHz$$

9va armónica:

$$f_9 = 9 * 3,5kHz = 31,5kHz$$

11va armónica:

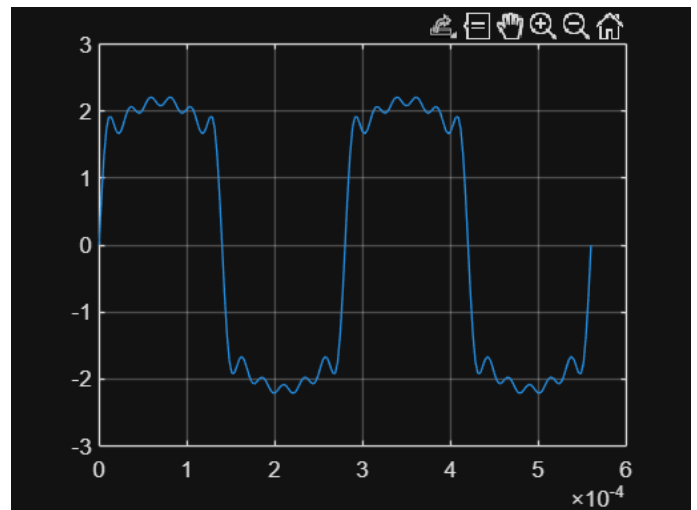
$$f_{11} = 11 * 3,5kHz = 38,5kHz$$

Con los valores obtenidos, se procede a realizar la suma de los armónicos calculados, con el propósito de verificar que la superposición de estas componentes permite reconstruir la señal original. Esta operación se lleva a cabo mediante el siguiente código, el cual permite visualizar la señal aproximada a partir de la suma de los armónicos teóricos

```
T=0.00028;
f=1/T;
A1=2.56;
A3=0.63;
A5=0.38;
A7=0.27;
A9=0.21;
A11=0.17;
t=0:T/100:2*T;
V1= A1*sin(2*pi*f*t);
V3= A3*sin(2*pi*3*f*t);
V5= A5*sin(2*pi*5*f*t);
V7= A7*sin(2*pi*7*f*t);
V9= A9*sin(2*pi*9*f*t);
V11= A11*sin(2*pi*11*f*t);
VT=V1+V3+V5+V7+V9+V11;
plot(t,VT);
grid on
```

#### Ilustración 57 Señal cuadrática.

A partir del código realizado, se obtiene la siguiente señal.



#### Ilustración 58 reconstrucción de señal cuadrada.

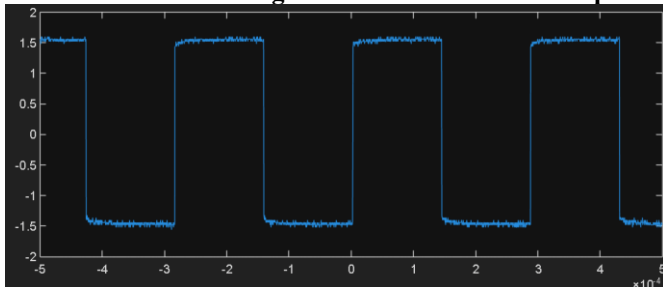
La señal obtenida concuerda con lo planteado inicialmente, ya que puede reconstruirse adecuadamente a partir de la suma de sus armónicos. No obstante, se observa la presencia de ruido en las crestas y valles de la señal, lo cual se debe a que una señal cuadrada ideal está compuesta por una cantidad infinita de armónicos. Por lo tanto, cuanto mayor sea el número de armónicos considerados en su reconstrucción, más precisa será la aproximación respecto a lo observado y registrado en el osciloscopio.



Ahora vamos a ver amas señal en tiempo y frecuencia con los archivos .csv, para esto usaremos los siguientes códigos.

```
>> t1=cuadrada.x_0_0005;
>> x1=cuadrada.x1_56;
>> plot(t1,x1)
```

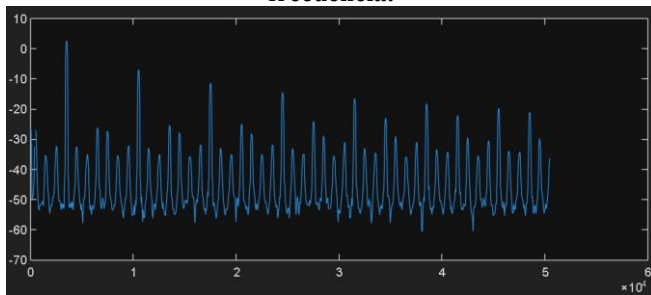
**Ilustración 59 Código señal cuadrada en el tiempo.**



**Ilustración 60 señal cuadrada en el tiempo.**

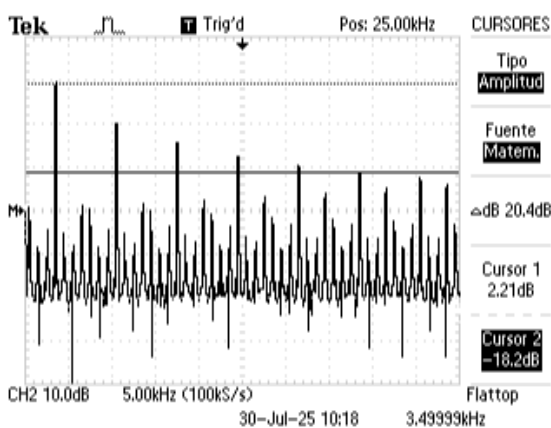
```
>> t=FFT4.x0;
>> x=FFT4.x_28_9897;
>> plot(t,x)
```

**Ilustración 61 Código señal cuadrada en el dominio de la frecuencia.**

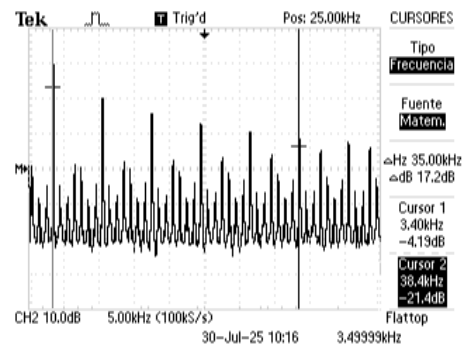


**Ilustración 62 señal cuadrada en el dominio de la frecuencia.**

Por otro lado, con el objetivo de visualizar experimentalmente las armónicas previamente calculadas de forma teórica, se utiliza la función MATH del osciloscopio. Al activarla, es posible observar la señal cuadrada junto con su descomposición en frecuencia, lo que permite analizar la presencia de los componentes armónicos de manera directa.



**Ilustración 63 Señal cuadrada en FFT y dB.**



**Ilustración 64 frecuencia de los armónicos.**

Como se observa en la ilustración 63, se identifican los principales armónicos de la señal, destacando experimentalmente la amplitud del primer y del onceavo armónico, junto con la medición de los demás armónicos presentes. Además, las frecuencias correspondientes a cada uno de estos armónicos fueron determinadas y se presentan en la ilustración 64, lo que permite una comparación directa con los valores teóricos. Por consiguiente, los valores obtenidos para cada armónica, tanto en amplitud como en frecuencia, muestran una buena concordancia con lo esperado. Finalmente, así como previamente se realizó la verificación de la señal en el dominio del tiempo, se lleva a cabo la comprobación en el dominio de la frecuencia, evidenciando que la suma de los armónicos permite reconstruir adecuadamente la señal original.

Armonicos	Frecuencia	dB
1	3,4 K	2.23
3	10,6 K	-6.57
5	17,6 K	-11.4
7	24,6 K	-13.8
9	31,4 K	-16.6
11	38,5 K	-18.2

**Ilustración 65 cálculo de armónicas.**

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

Armonicos	Frecuencia	dB	Voltaje
1	3,4 K	2.23	1.828163348
3	10,6 K	-6.57	0.663765992
5	17,6 K	-11.4	0.380640502
7	24,6 K	-13.8	0.288745349
9	31,4 K	-16.6	0.209177514
11	38,5 K	-18.2	0.173986278

**Ilustración 66 Tabla con voltajes.**

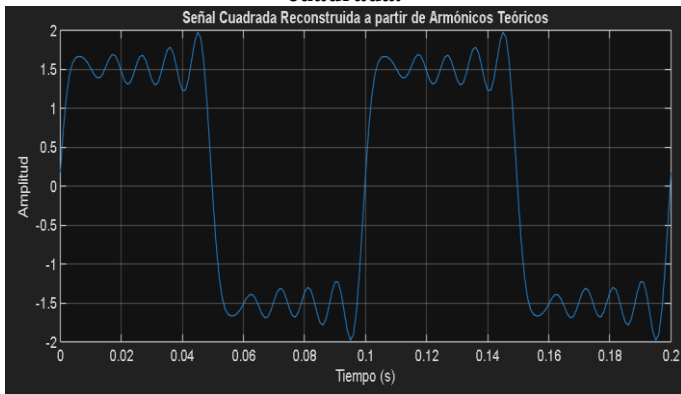
Teniendo estos voltajes vamos a usar el siguiente Código

```

T = 0.1;
f = 1/T;
A1 = 1.828163348;
A3 = 0.663765992;
A5 = 0.380640502;
A7 = 0.288745349;
A9 = 0.209177514;
A11 = 0.173986278;
t = 0:T/100:2*T;
V1 = A1 * sin(2 * pi * f * t);
V3 = A3 * sin(2 * pi * 3 * f * t);
V5 = A5 * sin(2 * pi * 5 * f * t);
V7 = A7 * sin(2 * pi * 7 * f * t);
V9 = A9 * sin(2 * pi * 9 * f * t);
V11 = A11 * cos(2 * pi * 11 * f * t);
VT = V1 + V3 + V5 + V7 + V9 + V11;
figure;
plot(t, VT);
title('Señal Cuadrada Reconstruida a partir de
xlabel('Tiempo (s)');
ylabel('Amplitud');
grid on;

```

**Ilustración 67 Código de reconstrucción de señal cuadrada.**



**Ilustración 68 Reconstrucción de señal cuadrada.**

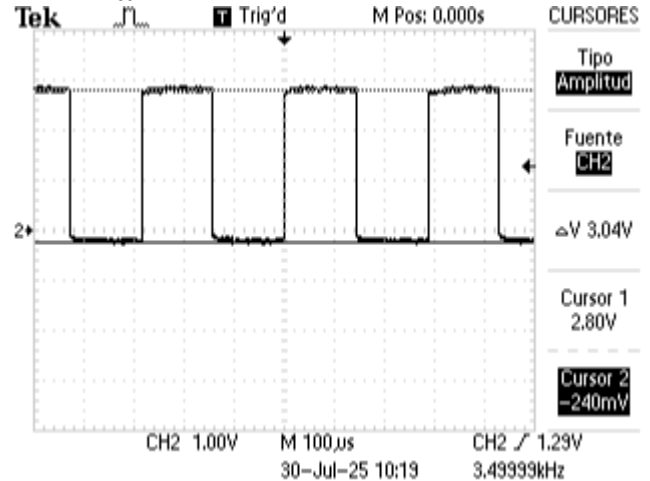
Al comparar las dos gráficas, la primera correspondiente a la señal cuadrada reconstruida a partir de armónicos teóricos y la segunda representando la señal cuadrada obtenida experimentalmente, se pueden identificar diferencias notables entre ambas representaciones. En la reconstrucción teórica, es evidente la presencia del fenómeno conocido como *efecto Gibbs*, observable en las oscilaciones que aparecen cerca de los flancos de subida y bajada de la señal, lo cual ocurre debido al uso limitado de armónicos para la síntesis de la forma de onda. Esto genera picos y valles en las transiciones de la señal, que no están presentes en la forma ideal.

Por otro lado, la señal experimental muestra bordes mucho más definidos y abruptos, con menor presencia de oscilaciones no deseadas, lo que puede deberse tanto a la mayor cantidad de armónicos involucrados en la señal real como al filtrado natural del sistema de adquisición (osciloscopio), que suaviza ciertas imperfecciones. Además, se nota que ambas señales comparten características clave como el valor de amplitud y la frecuencia, lo que indica una buena correspondencia entre el modelo teórico y la práctica. Sin embargo, la señal real presenta un nivel de ruido pequeño en las zonas planas, lo cual es común en mediciones electrónicas reales y no se contempla en simulaciones ideales. En conjunto, este análisis confirma la validez del modelo teórico y también resalta las diferencias

inevitables que surgen al trasladar una señal del plano matemático al experimental.

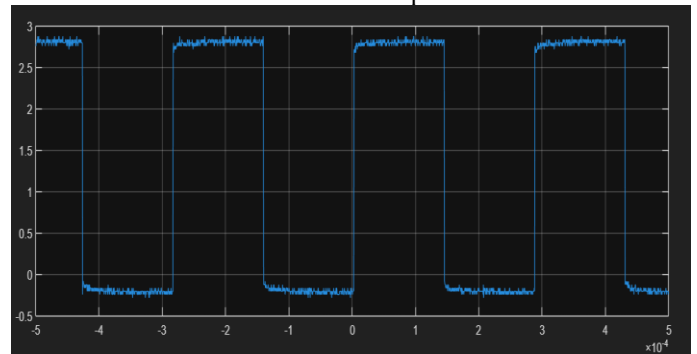
## VI. SEÑAL CUADRADA CON OFFSET.

A partir de la señal cuadrada previamente establecida, se añade un nivel de corriente directa (DC) de 1,25 V. Esta adición provoca un desplazamiento vertical en la señal, haciendo que su valor máximo aumente y el mínimo también se eleve. En consecuencia, el voltaje positivo pasa a ser de 2,75 V y el valor negativo se reduce a -0,25 V, como se muestra en la siguiente ilustración 69.



**Ilustración 69 cuadrada con offset.**

Como se observa en la imagen, la señal asciende con respecto a su nivel previamente establecido debido a la adición del nivel DC. Los datos se obtienen directamente del osciloscopio, lo cual permite visualizar en el software de MATLAB la señal cuadrada con el nivel de corriente directa incorporado.



**Ilustración 70 señal en Matlab.**

Como se observa, al adicionar una fuente de corriente directa, la señal se desplaza hacia arriba, modificando sus niveles de voltaje positivo y negativo. Esta variación se comprueba nuevamente a partir de los datos obtenidos y simulados en el software de MATLAB, reproduciendo la misma señal observada en el osciloscopio. Además, se realiza el cálculo de la amplitud de la señal, la cual en este caso corresponde al voltaje establecido, según se muestra a continuación.

$$20 \log_{10} \left( \frac{1,25V}{\sqrt{2}} \right) = -1 \text{ dB}$$

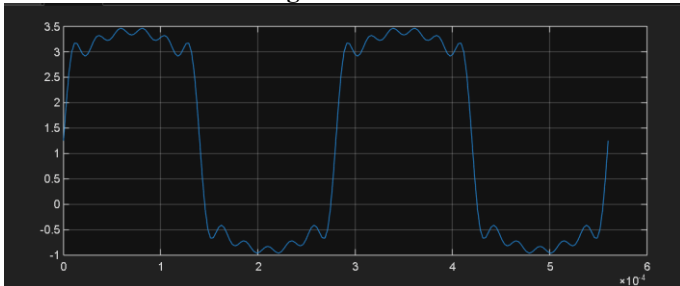
Para la señal cuadrada con offset las armónicas con las mismas lo único diferente es esta que acabamos de calcular que se va a encontrar en el origen.

```

T=0.00028;
f=1/T;
A0=1.25;
A1=2.56;
A3=0.63;
A5=0.38;
A7=0.27;
A9=0.21;
A11=0.17;
t=0:T/100:2*T;
V0 = A0 * ones(size(t));
V1= A1*sin(2*pi*f*t);
V3= A3*sin(2*pi*3*f*t);
V5= A5*sin(2*pi*5*f*t);
V7= A7*sin(2*pi*7*f*t);
V9= A9*sin(2*pi*9*f*t);
V11= A11*sin(2*pi*11*f*t);
VT=V0+V1+V3+V5+V7+V9+V11;
plot(t,VT);
grid on

```

**Ilustración 71 Código señal cuadrada con offset.**



**Ilustración 72 señal cuadrada con offset.**

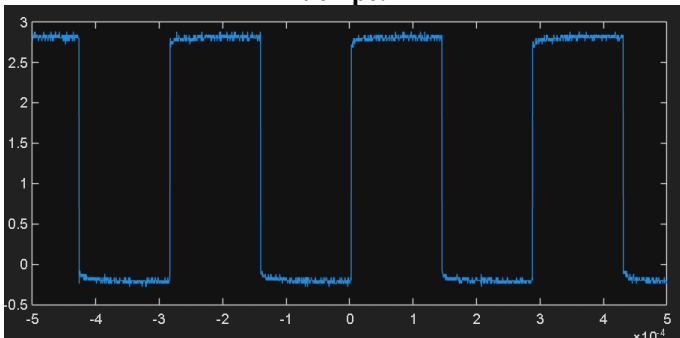
Ahora vamos a ver amas señal en tiempo y frecuencia con los archivos .csv, para esto usaremos los siguientes códigos.

```

>> t=CuadradaOffset.x_0_0005;
>> x=CuadradaOffset.x2_8;
>> plot(t,x)

```

**Ilustración 73 Código señal cuadrada con offset en el tiempo.**



**Ilustración 74 señal cuadrada con offset en el tiempo.**

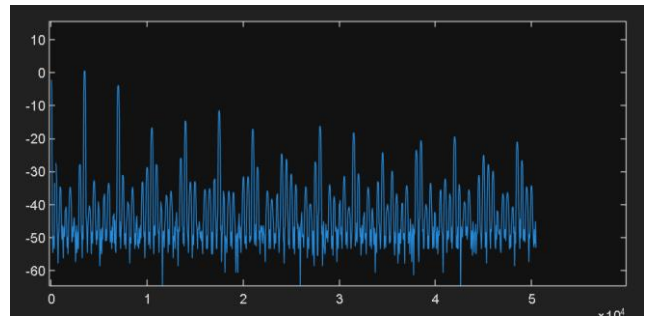
```

>> t=FFT6.x0;
>> x=FFT6.x_4_5897;
>> plot(t,x)

```

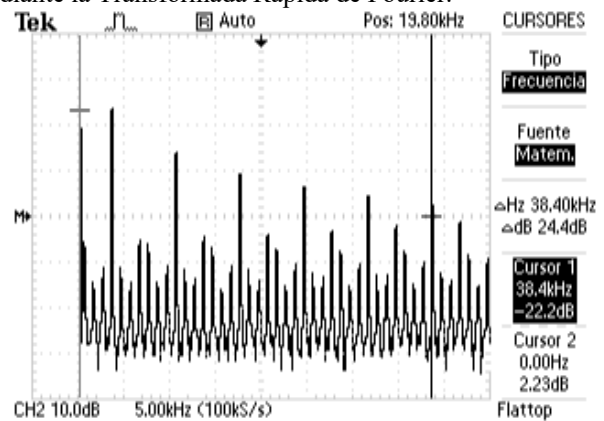
**Ilustración 75 Código señal cuadrada con offset en frecuencia.**

Por otro lado, con el objetivo de visualizar experimentalmente las armónicas previamente calculadas de forma teórica, se utiliza la función MATH del osciloscopio. Al activarla, es posible observar la señal cuadrada junto con su descomposición en frecuencia, lo que permite analizar la presencia de los componentes armónicos de manera directa.

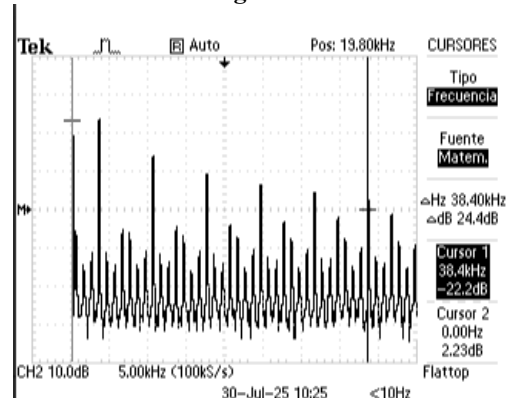


**Ilustración 76 Señal cuadrada con offset en frecuencia.**

A continuación, se representa gráficamente la señal junto con sus armónicas, obtenidas de forma experimental a partir de los datos registrados por el osciloscopio. En este caso, al tratarse de una señal cuadrada con un componente de corriente directa, el espectro de frecuencias muestra una componente significativa en la frecuencia cero, la cual corresponde al nivel DC añadido. Este comportamiento es consistente con la teoría, ya que una señal con desplazamiento vertical contiene una componente continua que no oscila, y por tanto se representa como un pico en 0 Hz. Además de esta componente, se observan las armónicas impares características de la señal cuadrada, cuya amplitud disminuye progresivamente con la frecuencia. En la Ilustración 77 se pueden apreciar las magnitudes de las primeras once armónicas, incluyendo la del nivel DC, mientras que en la Ilustración 78 se presentan las frecuencias correspondientes a cada una de estas componentes espectrales, lo cual permite visualizar claramente cómo se distribuyen en el dominio de la frecuencia y cómo se conservan las propiedades de la señal original, validando experimentalmente el resultado obtenido mediante la Transformada Rápida de Fourier.



**Ilustración 77 Magnitudes de armónicas.**



**Ilustración 78 Frecuencia de armónicas.**

Armonicos	Frecuencia	dB
0	0 K	4.66
1	3,4 K	2.23
3	10,6 K	-6.57
5	17,6 K	-11.4
7	24,6 K	-13.8
9	31,4 K	-16.6
11	38,5 K	-18.2

**Ilustración 79 Armónicos experimentales.**

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

Armonicos	Frecuencia	dB	Voltaje
0	0 K	4.66	2.41832685
1	3,4 K	2.23	1.828163348
3	10,6 K	-6.57	0.663765992
5	17,6 K	-11.4	0.380640502
7	24,6 K	-13.8	0.288745349
9	31,4 K	-16.6	0.209177514
11	38,5 K	-18.2	0.173986278

**Ilustración 80 Armónicos en voltaje.**

Teniendo estos voltajes vamos a usar el siguiente Código

```
T = 0.1;
f = 1/T;
A1 = 1.928163348;
A3 = 0.663765992;
A5 = 0.380640502;
A7 = 0.288745349;
A9 = 0.209177514;
A11 = 0.173986278;
t = 0:T/100:2*T;
V1 = A1 * sin(2 * pi * f * t);
V3 = A3 * sin(2 * pi * 3 * f * t);
V5 = A5 * sin(2 * pi * 5 * f * t);
V7 = A7 * sin(2 * pi * 7 * f * t);
V9 = A9 * sin(2 * pi * 9 * f * t);
V11 = A11 * cos(2 * pi * 11 * f * t);
VT = V1 + V3 + V5 + V7 + V9 + V11;
figure;
plot(t, VT);
title('Señal Cuadrada Reconstruida a p');
xlabel('Tiempo (s)');
ylabel('Amplitud');
grid on;
```

**Ilustración 81 Reconstrucción de señal con armónicos teóricos.**

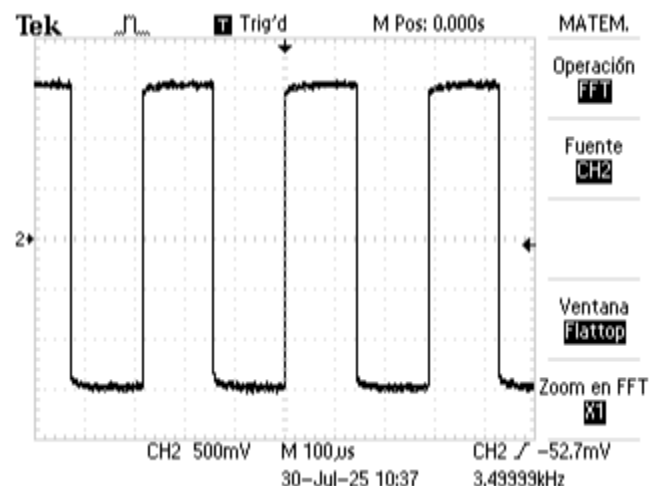


**Ilustración 82 Señal cuadrada con armónicos reconstruidos.**

En la ilustración se muestra la comparación entre la reconstrucción de la señal cuadrada a partir de los armónicos teóricos (ilustración 72) y la obtenida a partir de los armónicos experimentales (ilustración 82). En ambas se observa claramente el efecto de la suma de armónicos, con la característica forma de una señal cuadrada. Sin embargo, la señal reconstruida teóricamente presenta una mayor suavidad en sus transiciones y una simetría más definida, mientras que la experimental muestra ciertas irregularidades y ruido, especialmente en los niveles altos, lo cual se atribuye a factores como la precisión del instrumento y la cantidad limitada de armónicos considerados. Además, ambas incluyen un nivel de voltaje DC, lo cual provoca el desplazamiento vertical de la señal respecto al eje cero.

## VII. PULSO 1 CICLO ÚTIL 50%.

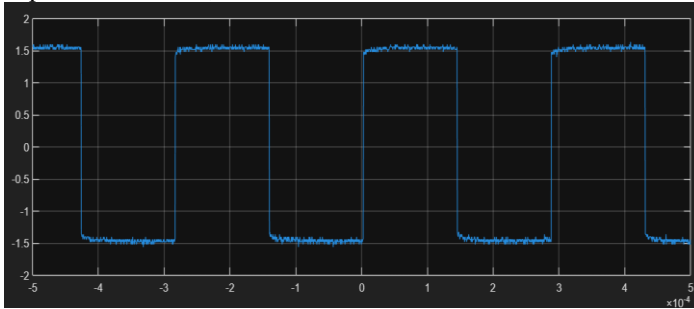
Inicialmente, se configura el generador de señales en modo pulso, estableciendo una señal cuadrada con un ciclo útil del 50%, lo que implica que el tiempo en estado alto es igual al tiempo en estado bajo durante cada ciclo. Los parámetros definidos para la señal son: un voltaje pico a pico de 3 Vpp, una frecuencia de 3,5 kHz y un ciclo útil del 50%. Con esta configuración, se obtiene la señal correspondiente en el dominio del tiempo, la cual presenta una forma cuadrada ideal con transiciones abruptas entre los niveles alto y bajo, como se muestra a continuación.



**Ilustración 83 Pulso 50%.**

A partir de la forma de la señal observada en el dominio del tiempo, se procede a verificar los datos registrados en el archivo .csv exportado desde el software OpenChoice Desktop del osciloscopio. Este archivo contiene los valores digitales de tiempo y voltaje capturados durante la medición. Dichos datos

permiten realizar la simulación y análisis correspondiente en MATLAB, representando de forma precisa la señal registrada experimentalmente, como se muestra a continuación.



**Ilustración 84 confirmación de Pulso.**

En este caso, se reafirma lo observado en el osciloscopio, ya que al presentar la señal un ciclo útil del 50%, su comportamiento corresponde al de una señal cuadrada simétrica. Por lo tanto, se hace uso de la misma expresión teórica correspondiente a una señal cuadrada para calcular la amplitud y la frecuencia de sus armónicos, lo que permite comparar los valores teóricos con los obtenidos experimentalmente.

1er armónico.

$$A_1 = \frac{4 \times 1,5V}{\pi} = 1,9V$$

$$20 \log_{10} \left( \frac{1,9V}{\sqrt{2}} \right) = 2,56 \text{ dB}$$

3er armónico.

$$A_3 = \frac{4 \times 1,5V}{3\pi} = 0,63V$$

$$20 \log_{10} \left( \frac{0,63V}{\sqrt{2}} \right) = -7 \text{ dB}$$

5to armónico.

$$A_5 = \frac{4 \times 1,5V}{5\pi} = 0,38V$$

$$20 \log_{10} \left( \frac{0,38V}{\sqrt{2}} \right) = -11,41 \text{ dB}$$

7mo armónico.

$$A_7 = \frac{4 \times 1,5V}{7\pi} = 0,27V$$

$$20 \log_{10} \left( \frac{0,27V}{\sqrt{2}} \right) = -14,3 \text{ dB}$$

9no armónico.

$$A_9 = \frac{4 \times 1,5V}{9\pi} = 0,21V$$

$$20 \log_{10} \left( \frac{0,21V}{\sqrt{2}} \right) = -16,5 \text{ dB}$$

11vo armónico.

$$A_{11} = \frac{4 \times 1,5V}{11\pi} = 0,17V$$

$$20 \log_{10} \left( \frac{0,17V}{\sqrt{2}} \right) = -18,2 \text{ dB}$$

Frecuencia teórica.

1ra armónica:

$$f_1 = 1 \times 3,5kHz = 3,5kHz$$

3ra armónica:

$$f_3 = 3 \times 3,5kHz = 10,5kHz$$

5ta armónica:

$$f_5 = 5 \times 3,5kHz = 17,5kHz$$

7ma armónica:

$$f_7 = 7 \times 3,5kHz = 24,5kHz$$

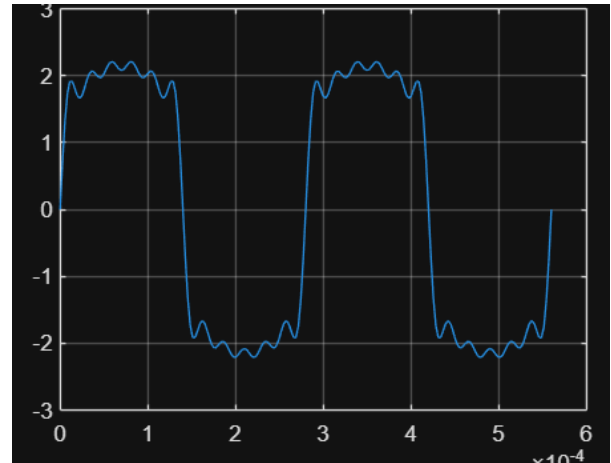
9va armónica:

$$f_9 = 9 \times 3,5kHz = 31,5kHz$$

11va armónica:

$$f_{11} = 11 \times 3,5kHz = 38,5kHz$$

A partir de estos datos calculados, se realizan la suma de estos armónicos con el fin de obtener la señal de pulso con un 50% de ciclo útil, en este caso implementas el mismo código previamente realizado para la señal cuadrada, a partir de esto se obtienen la siguiente grafica.



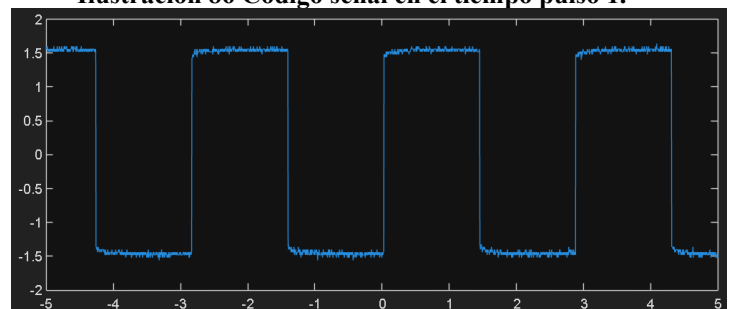
**Ilustración 85 Señal construido con armónicos teóricos.**

Como se observa, la señal presenta ruido y no es completamente simétrica en los valles y las crestas. Esto se debe a que, al tratarse de una señal que en teoría contiene infinitos armónicos, la reconstrucción con un número limitado de estos no logra formar con precisión la señal ideal. En este caso particular, solo se están considerando los primeros 11 armónicos, lo cual limita la fidelidad de la reconstrucción. Esta cantidad de componentes no es suficiente para replicar con exactitud la forma cuadrada original, razón por la cual se perciben transiciones suaves en lugar de cambios abruptos.

Ahora vamos a ver mas señal en tiempo y frecuencia con los archivos .csv, para esto usaremos los siguientes códigos.

```
>> t=pulso1.x_0_0005;
>> x=pulso1.x1_56;
>> plot(t,x)
```

**Ilustración 86 Código señal en el tiempo pulso 1.**

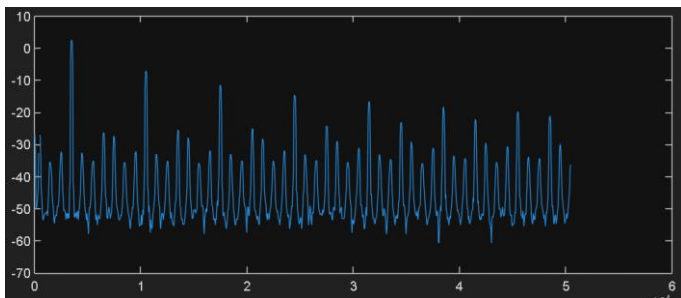


**Ilustración 87 Pulso 1 en tiempo.**

```
>> t1=FFT4.x0;
>> x1=FFT4.x_28_9897;
>> plot(t1,x1)
```

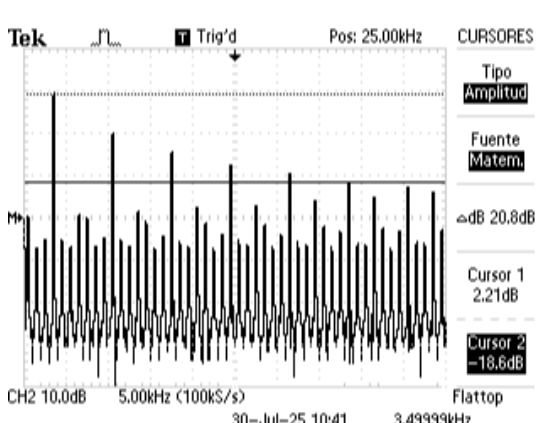
**Ilustración 88 Código de frecuencia pulso 1.**



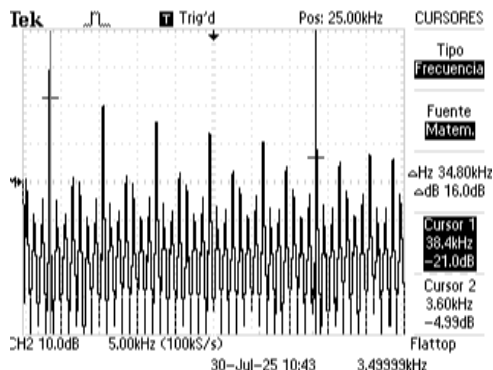


**Ilustración 89 Señal en frecuencia del pulso 1.**

Por otro lado, con el objetivo de visualizar experimentalmente las armónicas previamente calculadas de forma teórica, se utiliza la función MATH del osciloscopio. Al activarla, es posible observar la señal cuadrada junto con su descomposición en frecuencia, lo que permite analizar la presencia de los componentes armónicos de manera directa.



**Ilustración 90 Amplitud de pulso 1.**



**Ilustración 91 Frecuencia de pulso 1.**

Ahora vamos a hacer una tabla de los valores experimentales para poder después graficar la señal con armónicos experimentales.

Armonicos	Frecuencia	dB
1	3,4 K	2.61
3	10,6 K	-6.96
5	17,6 K	-11.4
7	24,6 K	-14.2
9	31,4 K	-17.4
11	38,5 K	-18.6

**Ilustración 92 armónicas experimentales pulso 50 %.**

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

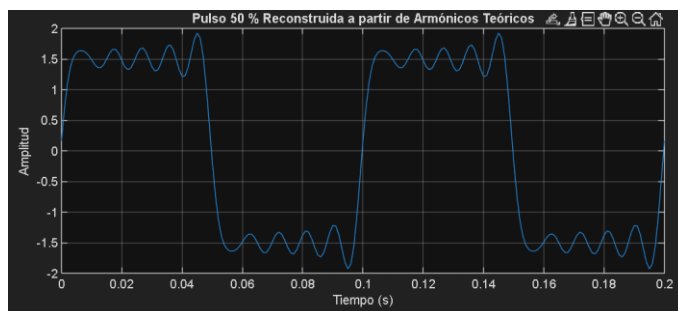
Armonicos	Frecuencia	dB	Voltaje
1	3,4 K	2.61	1.90991921
3	10,6 K	-6.96	0.63462182
5	17,6 K	-11.4	0.3806405
7	24,6 K	-14.2	0.27574967
9	31,4 K	-17.4	0.19077216
11	38,5 K	-18.6	0.16615561

**Ilustración 93 Tabla con voltajes.**

Teniendo estos voltajes vamos a usar el siguiente Código

```
T = 0.1;
f = 1/T;
A1 = 1.909919214;
A3 = 0.634621816;
A5 = 0.380640502;
A7 = 0.275749668;
A9 = 0.19077216;
A11 = 0.166155606;
t = 0:T/100:2*T;
V1 = A1 * sin(2 * pi * f * t);
V3 = A3 * sin(2 * pi * 3 * f * t);
V5 = A5 * sin(2 * pi * 5 * f * t);
V7 = A7 * sin(2 * pi * 7 * f * t);
V9 = A9 * sin(2 * pi * 9 * f * t);
V11 = A11 * cos(2 * pi * 11 * f * t);
VT = V1 + V3 + V5 + V7 + V9 + V11;
figure;
plot(t, VT);
title('Pulso 50 % Reconstruida a partir
xlabel('Tiempo (s)');
ylabel('Amplitud');
grid on;
```

**Ilustración 94 Código para reconstrucción pulso.**



**Ilustración 95 Pulso reconstruido con armónicas experimentales.**

Al analizar las dos señales presentadas, la primera reconstruida a partir de los armónicos experimentales y la segunda utilizando los armónicos teóricos, se evidencian varias diferencias clave. En la señal experimental se observan ligeras irregularidades y asimetrías tanto en los picos como en los valles de la onda, lo cual es característico del ruido presente en los sistemas reales y de posibles imprecisiones en la medición o en el número de

armónicos capturados. En cambio, la señal reconstruida teóricamente presenta una forma más ideal, simétrica y suave, producto de la aplicación exacta de la serie de Fourier bajo condiciones ideales. No obstante, ambas señales muestran una reconstrucción incompleta, ya que solo se están considerando los primeros 11 armónicos, lo que genera una forma de onda con transiciones no perfectamente verticales y presencia del fenómeno de Gibbs. En resumen, la señal experimental refleja de mejor manera las condiciones reales del sistema, mientras que la teórica sirve como referencia idealizada para comparar la calidad de la reconstrucción.

VIII. PULSO 2 CICLO ÚTIL 20%.

Inicialmente, se configura el generador de señales en modo pulso, estableciendo una señal pulso con un ciclo útil del 20%, lo que implica que el tiempo en estado alto corresponde al 20% del periodo total, y el tiempo en estado bajo al 80%. Los parámetros definidos para la señal son: un voltaje pico a pico de 3 Vpp, una frecuencia de 3,5 kHz y un ciclo útil del 20%. Con esta configuración, se obtiene la señal correspondiente en el dominio del tiempo, la cual presenta una forma cuadrada asimétrica, con un tiempo en nivel alto menor que en nivel bajo, y transiciones abruptas entre ambos niveles, como se muestra a continuación.

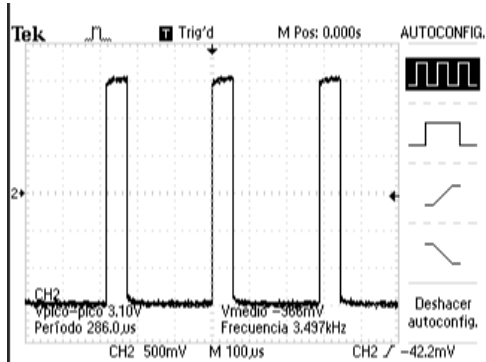


Ilustración 96 Pulso 20 %

Con base en la forma de onda obtenida en el dominio del tiempo para la señal con ciclo útil del 20%, se analizan los datos registrados en el archivo .csv, exportado mediante el software OpenChoice Desktop del osciloscopio. Este archivo contiene los valores digitales correspondientes al tiempo y voltaje medidos experimentalmente. A partir de estos datos, se desarrolla la simulación en MATLAB, lo que permite visualizar y comparar la señal adquirida con su comportamiento teórico, facilitando el análisis de sus características particulares, como se muestra a continuación.

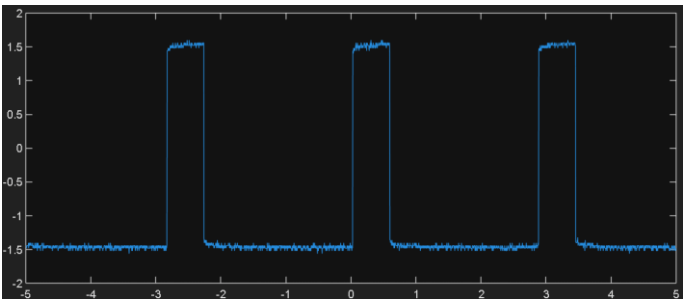


Ilustración 97 pulso 2.

Ahora vamos a calcular las armónicas teóricas lo primero que vamos a hacer es hallar tau para esta debemos usar la siguiente formula.

$$\tau = \frac{1}{3500} = 285,71 \mu s$$

Ahora vamos a calcular las magnitudes para esto vamos a usar la siguiente formula.

$$V = 2A \times \left| \frac{\sin(n\pi D)}{n\pi} \right|$$

Para hallar la frecuencia vamos a usar la siguiente ecuacion 5.

Armonico	Frecuencia	Voltaje
1	3,5 K	0.882
2	7 K	0.556
3	10,5 K	0.199
4	14 K	0.191
5	17,5 K	0.524
6	21 K	0.375
7	24,5 K	0.018
8	28 K	0.303
9	31,5 K	0.407
10	35 K	0.132
11	38,5 K	0.099

Ilustración 98 Armónicas teóricas.

A partir de estos datos calculados, se realiza la suma de los armónicos con el fin de obtener la señal de pulso correspondiente a un 20% de ciclo útil. Para ello, se implementa el mismo código previamente utilizado para la señal cuadrada, realizando los ajustes necesarios en la serie de Fourier. Como resultado, se obtiene la siguiente gráfica que representa la reconstrucción de la señal a partir de sus primeros armónicos.



Ilustración 99 señal teórica.

Como se observa en la figura, la señal reconstruida presenta una forma aproximada a un pulso con ciclo útil del 20%. Sin embargo, se pueden notar irregularidades como picos suavizados y pequeñas oscilaciones en los intervalos donde la señal debería permanecer en cero. Estas imperfecciones son el resultado de una reconstrucción incompleta, ya que la señal ideal requiere una suma infinita de armónicos para su representación exacta.

En este caso, se han considerado únicamente los primeros 11 armónicos, lo cual limita la fidelidad de la reconstrucción. Esta cantidad de componentes no es suficiente para generar transiciones perfectamente abruptas ni mantener la señal plana entre pulsos, razón por la cual se observan transiciones más suaves y cierta distorsión.

A continuación, se analizará esta señal tanto en el dominio del tiempo como en el de la frecuencia, utilizando archivos .csv y aplicando transformadas rápidas de Fourier (FFT) con los siguientes códigos.

```
>> t1=pulso2.x_0_0005;  
>> x1=pulso2.x_1_4;  
>> plot(t1,x1)
```

Ilustración 100 Código pulso en tiempo.

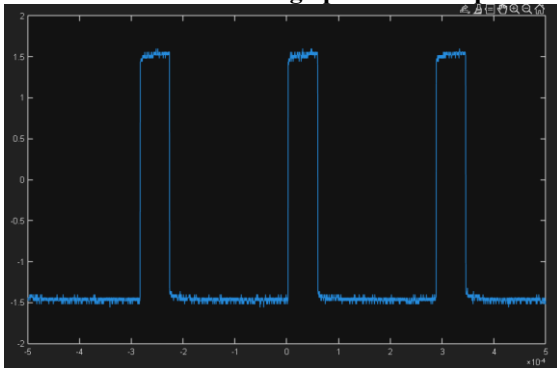


Ilustración 101 pulso en tiempo.

```
>> t=FFT5.x0;  
>> x=FFT5.x_1_3691;  
>> plot(t,x)
```

Ilustración 102 Código para pulso en frecuencia.

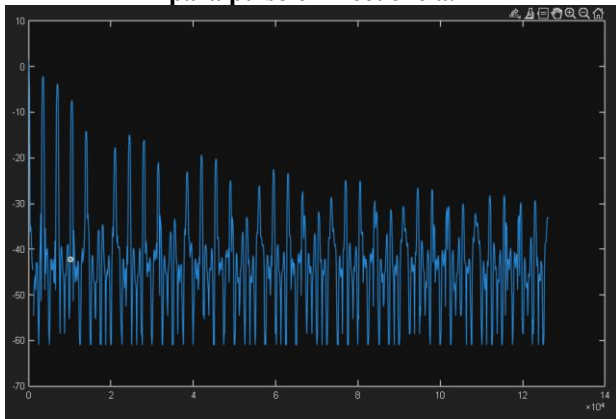


Ilustración 103 pulso en frecuencia.

Por otro lado, con el objetivo de visualizar experimentalmente las armónicas previamente calculadas de forma teórica, se utiliza la función MATH del osciloscopio. Al activarla, es posible observar la señal cuadrada junto con su descomposición en frecuencia, lo que permite analizar la presencia de los componentes armónicos de manera directa.

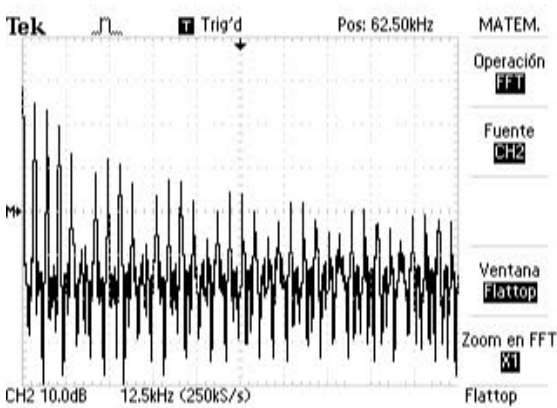


Ilustración 104 Armonía experimentales.

Armonico	Frecuencia	dB
1	3,5K	-2.19
2	7,2 K	-3.79
3	10,5 K	-7.39
4	14 K	-14.2
5	18 K	0
6	21 K	-17.8

Ilustración 105 armónicas experimentales.

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

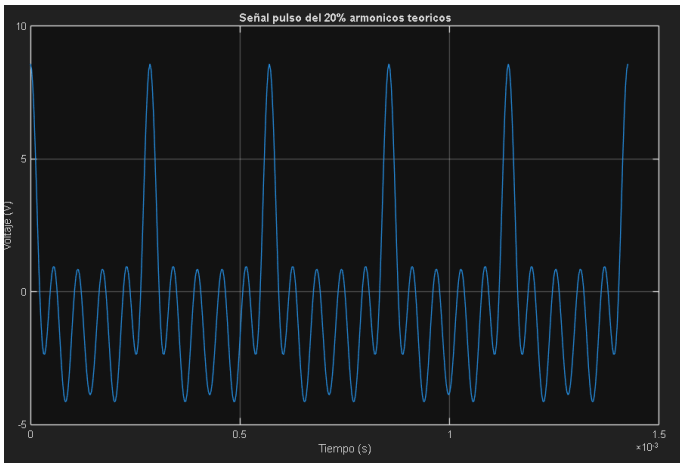
Armonico	Frecuencia	dB	Voltaje
1	3,5K	-2.19	1.09904379
2	7,2 K	-3.79	0.91414481
3	10,5 K	-7.39	0.60396949
4	14 K	-14.2	0.27574967
5	18 K	0	1.41421356
6	21 K	-17.8	0.182186

Ilustración 106 Tabla voltajes.

Teniendo estos voltajes vamos a usar el siguiente Código

```
T = 0.1;  
f = 1/T;  
A1 = 1.099043793;  
A2 = 0.91414481;  
A3 = 0.603969486;  
A4 = 0.275749668;  
A5 = 1.414213562;  
A6 = 0.182185999;  
t = 0:T/100:2*T;  
V1 = A1 * cos(2 * pi * f * t);  
V2 = A2 * cos(2 * pi * 2 * f * t);  
V3 = A3 * cos(2 * pi * 3 * f * t);  
V4 = A4 * cos(2 * pi * 4 * f * t);  
V5 = A5 * cos(2 * pi * 5 * f * t);  
V6 = A6 * cos(2 * pi * 6 * f * t);  
V_pulse_20 = V1 + V2 + V3 + V4 + V5 + V6 ;  
figure;  
plot(t, V_pulse_20);  
title('Señal de Pulso Reconstruida con Duty Cycle 20%');  
xlabel('Tiempo (s)');  
ylabel('Amplitud');  
grid on;
```

Ilustración 107 Código para reconstruir señal.



**Ilustración 108 Pulso 20 % reconstruido.**

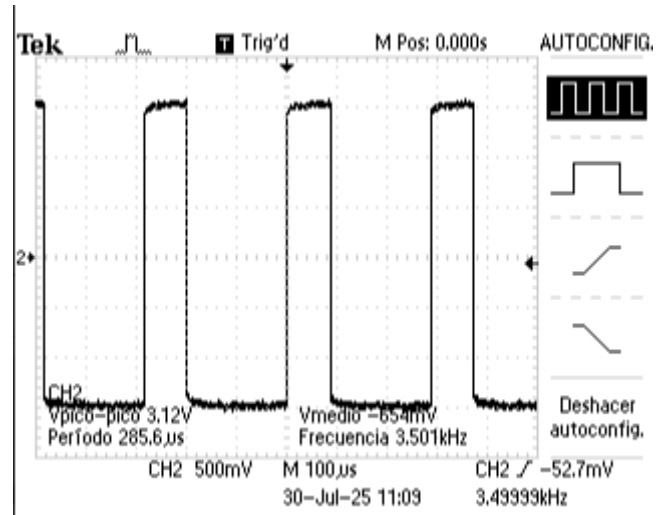
Como se observa en la figura de la señal reconstruida teóricamente, aunque se aproxima a una forma de pulso con un ciclo útil del 20%, se presentan oscilaciones indeseadas en los valles y las crestas. Esta distorsión se debe a que la reconstrucción de una señal cuadrada ideal requiere una cantidad infinita de armónicos. Al limitar la reconstrucción a los primeros 11 armónicos, se genera lo que se conoce como el efecto Gibbs, el cual introduce sobre impulsos y transiciones suaves donde deberían existir cambios abruptos. No obstante, se logra una representación bastante razonable del comportamiento general del pulso.

Por otro lado, al observar la señal obtenida experimentalmente (segunda figura), se nota una forma más distorsionada, con menos definición en los bordes y menor simetría. Esto se debe a que solo se lograron capturar los primeros 6 armónicos en la reconstrucción, lo cual reduce aún más la fidelidad de la forma de onda. Las componentes de alta frecuencia, que son esenciales para definir los bordes abruptos de una señal cuadrada, están ausentes, y por ello las transiciones son aún más suaves.

También se observa un nivel de ruido mayor, típico en señales adquiridas con equipo real. Esta comparación pone en evidencia la importancia de incluir un mayor número de armónicos para obtener una reconstrucción más precisa, especialmente cuando se trata de representar señales con transiciones bruscas como los pulsos rectangulares.

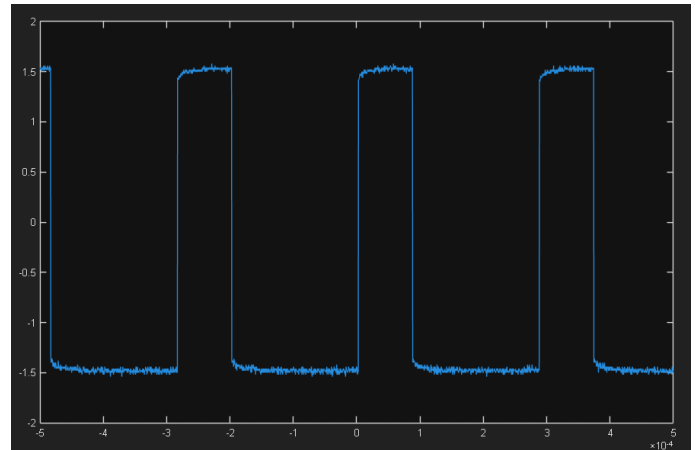
## IX. PULSO 3 CICLO ÚTIL 30%.

Inicialmente, se configura el generador de señales en modo pulso, estableciendo una señal pulso con un ciclo útil del 30%, lo que implica que el tiempo en estado alto corresponde al 30% del periodo total, y el tiempo en estado bajo al 70%. Los parámetros definidos para la señal son: un voltaje pico a pico de 3 Vpp, una frecuencia de 3,5 kHz y un ciclo útil del 30%. Con esta configuración, se obtiene la señal correspondiente en el dominio del tiempo, la cual presenta una forma cuadrada asimétrica, con un tiempo en nivel alto menor que en nivel bajo, y transiciones abruptas entre ambos niveles, como se muestra a continuación.



**Ilustración 109 Pulso 30%**

Con base en la forma de onda obtenida en el dominio del tiempo para la señal con ciclo útil del 30%, se analizan los datos registrados en el archivo .csv, exportado mediante el software OpenChoice Desktop del osciloscopio. Este archivo contiene los valores digitales correspondientes al tiempo y voltaje medidos experimentalmente. A partir de estos datos, se desarrolla la simulación en MATLAB, lo que permite visualizar y comparar la señal adquirida con su comportamiento teórico, facilitando el análisis de sus características particulares, como se muestra a continuación.



**Ilustración 110 pulso de 30 %**

Ahora vamos a calcular las armónicas teóricas lo primero que vamos a hacer es hallar  $T$  para esta debemos usar la siguiente formula.

$$T = \frac{1}{3500} = 285,71 \mu s$$

Ahora vamos a calcular las magnitudes para esto vamos a usar la siguiente formula.

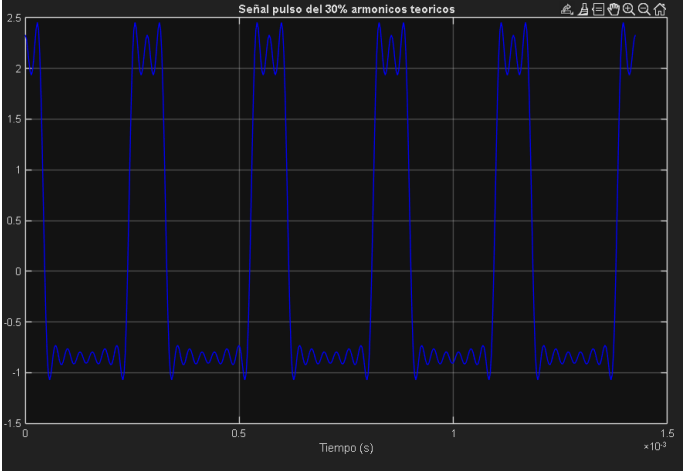
$$V = 2A \times \left| \frac{\sin(n\pi D)}{n\pi} \right|$$

Para hallar la frecuencia vamos a usar la siguiente ecuacion 5.

Armonico	Frecuencia	Voltaje
1	3,5 K	0.45
2	7 K	0.77
3	10,5 K	0.45
4	14 K	0.1
5	17,5 K	-0.14
6	21 K	-0.19
7	24,5 K	-0.09
8	28 K	0.04
9	31,5 K	0.11
10	35 K	0.09
11	38,5 K	0

**Ilustración 111 armónicos teóricos.**

A partir de estos datos calculados, se realiza la suma de los armónicos con el fin de obtener la señal de pulso correspondiente a un 30% de ciclo útil. Para ello, se implementa el mismo código previamente utilizado para la señal cuadrada, realizando los ajustes necesarios en la serie de Fourier. Como resultado, se obtiene la siguiente gráfica que representa la reconstrucción de la señal a partir de sus primeros armónicos.



**Ilustración 112 Pulso 30%.**

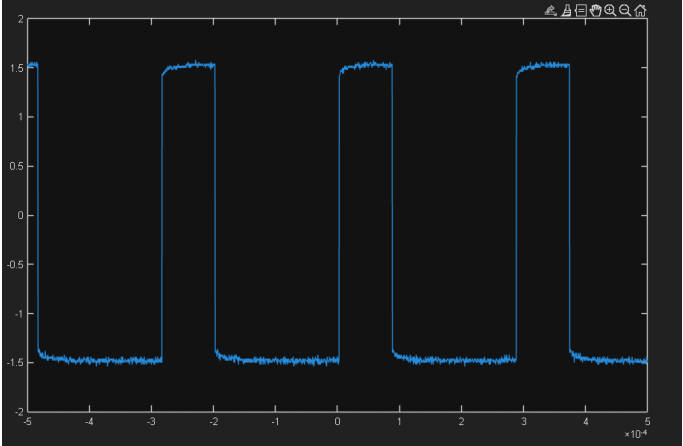
Como se observa en la figura, la señal reconstruida presenta una forma aproximada a un pulso con ciclo útil del 30%. Sin embargo, se pueden notar irregularidades como picos suavizados y pequeñas oscilaciones en los intervalos donde la señal debería permanecer en cero. Estas imperfecciones son el resultado de una reconstrucción incompleta, ya que la señal ideal requiere una suma infinita de armónicos para su representación exacta.

En este caso, se han considerado únicamente los primeros 11 armónicos, lo cual limita la fidelidad de la reconstrucción. Esta cantidad de componentes no es suficiente para generar transiciones perfectamente abruptas ni mantener la señal plana entre pulsos, razón por la cual se observan transiciones más suaves y cierta distorsión.

A continuación, se analizará esta señal tanto en el dominio del tiempo como en el de la frecuencia, utilizando archivos .csv y aplicando transformadas rápidas de Fourier (FFT) con los siguientes códigos.

```
>> t=pulso3.x_0_0005;  
>> x=pulso3.x1_52;  
>> plot(t,x)
```

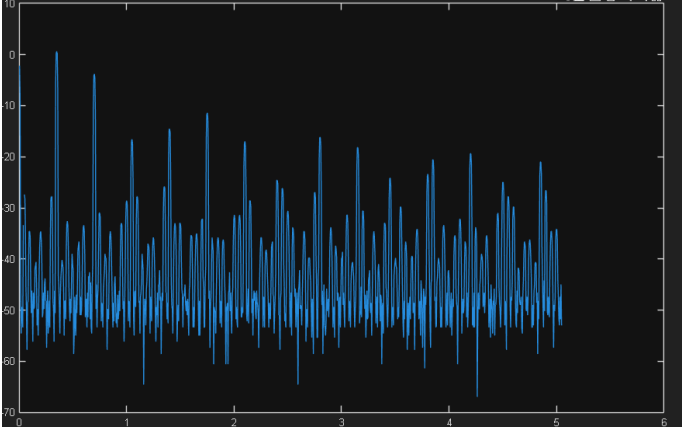
**Ilustración 113 Código en el tiempo.**



**Ilustración 114 pulso en el tiempo.**

```
>> t1=FFT6.x0;  
>> x1=FFT6.x_4_5897;  
>> plot(t1,x1)
```

**Ilustración 115 Código para frecuencia.**



**Ilustración 116 Pulso en frecuencia.**

Por otro lado, con el objetivo de visualizar experimentalmente las armónicas previamente calculadas de forma teórica, se utiliza la función MATH del osciloscopio. Al activarla, es posible observar la señal cuadrada junto con su descomposición en frecuencia, lo que permite analizar la presencia de los componentes armónicos de manera directa.



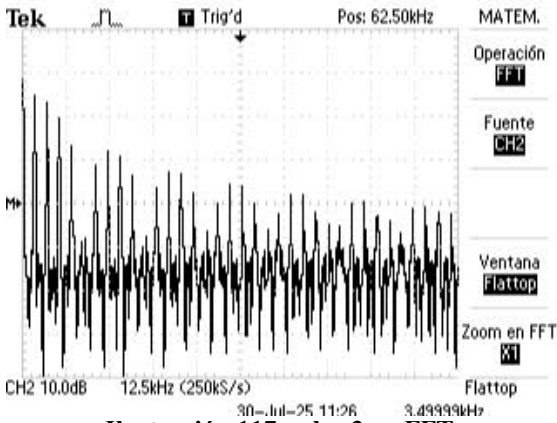


Ilustración 117 pulso 3 en FFT.

Armonico	Frecuencia	dB
1	3,4 K	0.21
2	7,2 K	-3.79
3	10,6 K	-17.8
4	14 K	-14.2
5	17.4K	-11.4
6	21 K	-16.2

Ilustración 118 armónicos experimentales.

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

Armonico	Frecuencia	dB	Voltaje
1	3,4 K	0.21	1.44882189
2	7,2 K	-3.79	0.91414481
3	10,6 K	-17.8	0.182186
4	14 K	-14.2	0.27574967
5	17.4K	-11.4	0.3806405
6	21 K	-16.2	0.21903575

Ilustración 119 Voltajes.

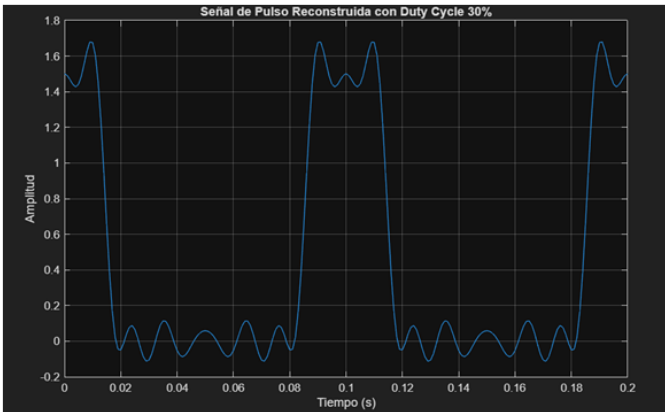


Ilustración 120 señal reconstruía con armónicos experimentales.

En la primera imagen se observa una señal con características predominantemente triangulares, simétrica, que parece haber sido reconstruida tras una transformación IFFT. Esta reconstrucción refleja los componentes de frecuencia de una señal no senoidal, posiblemente generada artificialmente o extraída de un proceso experimental. La forma de onda es continua, suave en los extremos y de pendiente constante, lo que

es característico de una señal triangular pura. Esto sugiere que contiene componentes armónicos impares de orden más bajo, lo cual se relaciona directamente con la teoría de series de Fourier para señales triangulares periódicas.

En contraste, la segunda imagen muestra una señal que también tiene forma triangular, pero presenta irregularidades más marcadas y ruido en su reconstrucción. Esto puede deberse a un truncamiento de los coeficientes de la FFT, un muestreo insuficiente o errores en el filtrado o reconstrucción. Las pendientes no son tan constantes, y se observan ciertas deformaciones en la simetría, lo cual indica que algunos armónicos se han perdido o que hay presencia de componentes no ideales.

X. PULSO 4 CICLO ÚTIL 80%.

Inicialmente, se configura el generador de señales en modo pulso, estableciendo una señal pulso con un ciclo útil del 20%, lo que implica que el tiempo en estado alto corresponde al 20% del periodo total, y el tiempo en estado bajo al 80%. Los parámetros definidos para la señal son: un voltaje pico a pico de 3 Vpp, una frecuencia de 3,5 kHz y un ciclo útil del 20%. Con esta configuración, se obtiene la señal correspondiente en el dominio del tiempo, la cual presenta una forma cuadrada asimétrica, con un tiempo en nivel alto menor que en nivel bajo, y transiciones abruptas entre ambos niveles, como se muestra a continuación.

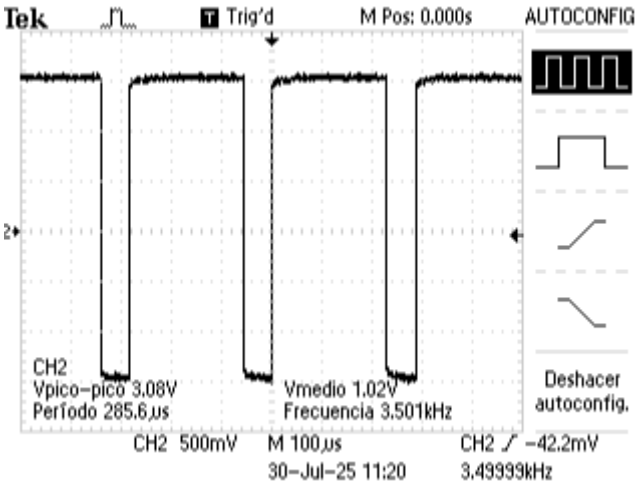


Ilustración 121 pulso 80%.

Con base en la forma de onda obtenida en el dominio del tiempo para la señal con ciclo útil del 80%, se analizan los datos registrados en el archivo .csv, exportado mediante el software OpenChoice Desktop del osciloscopio. Este archivo contiene los valores digitales correspondientes al tiempo y voltaje medidos experimentalmente. A partir de estos datos, se desarrolla la simulación en MATLAB, lo que permite visualizar y comparar la señal adquirida con su comportamiento teórico, facilitando el análisis de sus características particulares, como se muestra a continuación.

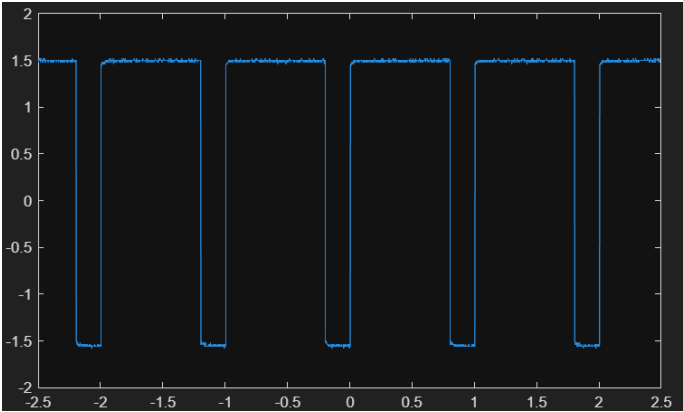


Ilustración 122 pulso 80%.

Ahora vamos a calcular las armónicas teóricas lo primero que vamos a hacer es hallar tau para esta debemos usar la siguiente formula.

$$T = \frac{1}{3500} = 285,71 \mu s$$

Ahora vamos a calcular las magnitudes para esto vamos a usar la siguiente formula.

$$V = 2A \times \left| \frac{\sin(n\pi D)}{n\pi} \right|$$

Para hallar la frecuencia vamos a usar la siguiente ecuacion 5

Armonico	Frecuencia	Voltaje
0	0	1.2
1	3,5 K	0.56
2	7 K	-0.45
3	10,5 K	0.3
4	14 K	-0.14
5	17,5 K	0
6	21 K	0.09
7	24,5 K	-0.13
8	28 K	0.11
9	31,5 K	-0.06
10	35 K	0

Ilustración 123 Armónicas teóricas.

A partir de estos datos calculados, se realiza la suma de los armónicos con el fin de obtener la señal de pulso correspondiente a un 80% de ciclo útil. Para ello, se implementa el mismo código previamente utilizado para la señal cuadrada, realizando los ajustes necesarios en la serie de Fourier. Como resultado, se obtiene la siguiente gráfica que representa la reconstrucción de la señal a partir de sus primeros armónicos.

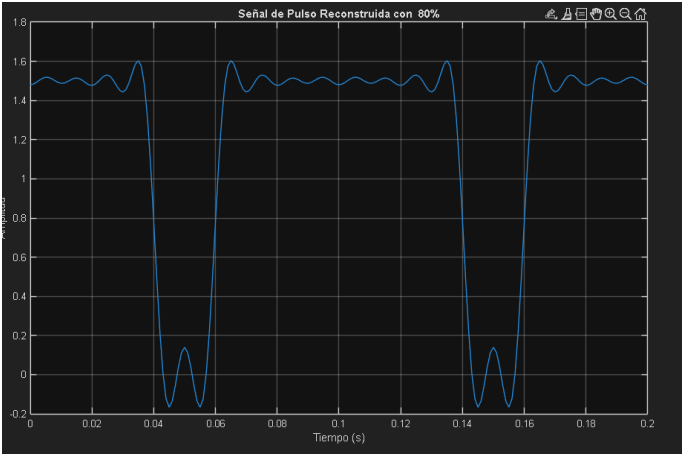


Ilustración 124 señal reconstruida con aminas teóricas.

Como se observa en la figura, la señal reconstruida presenta una forma aproximada a un pulso con ciclo útil del 80%. Sin embargo, se pueden notar irregularidades como picos suavizados y pequeñas oscilaciones en los intervalos donde la señal debería permanecer en cero. Estas imperfecciones son el resultado de una reconstrucción incompleta, ya que la señal ideal requiere una suma infinita de armónicos para su representación exacta.

En este caso, se han considerado únicamente los primeros 11 armónicos, lo cual limita la fidelidad de la reconstrucción. Esta cantidad de componentes no es suficiente para generar transiciones perfectamente abruptas ni mantener la señal plana entre pulsos, razón por la cual se observan transiciones más suaves y cierta distorsión.

A continuación, se analizará esta señal tanto en el dominio del tiempo como en el de la frecuencia, utilizando archivos .csv y aplicando transformadas rápidas de Fourier (FFT) con los siguientes códigos.

```
>> t=pulso3.x_0_0005;  
>> x=pulso3.x1_52;  
>> plot(t,x)
```

Ilustración 125 Código pulso en el tiempo.

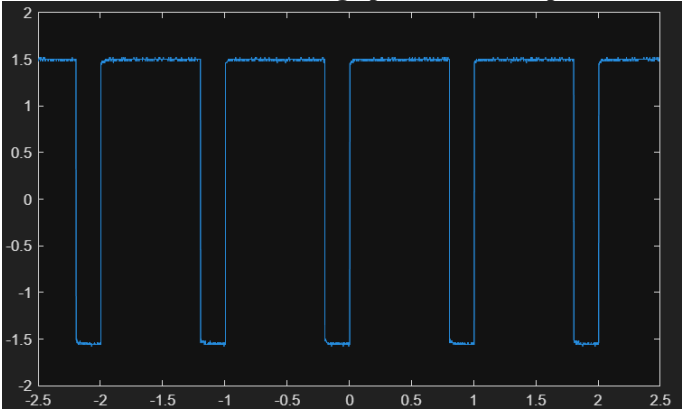


Ilustración 126 pulso en el tiempo.

```
t1=fft7.x0;  
x1=fft7.x_0_5897;  
plot(t1,x1)
```

Ilustración 127 Código pulso en frecuencia.

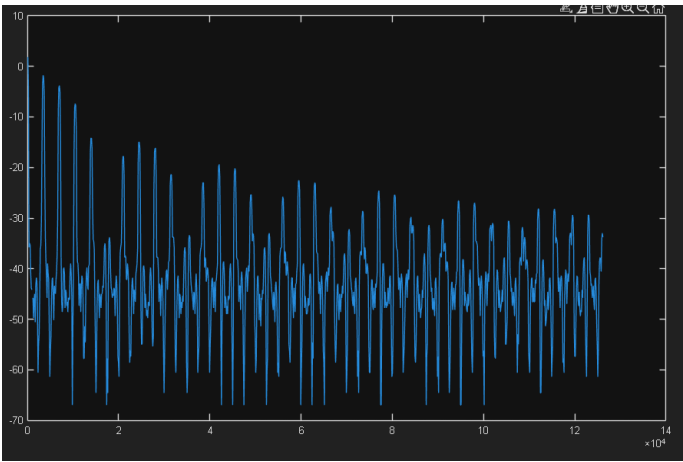


Ilustración 128 Pulso en frecuencia.

Por otro lado, con el objetivo de visualizar experimentalmente las armónicas previamente calculadas de forma teórica, se utiliza la función MATH del osciloscopio. Al activarla, es posible observar la señal cuadrada junto con su descomposición en frecuencia, lo que permite analizar la presencia de los componentes armónicos de manera directa.

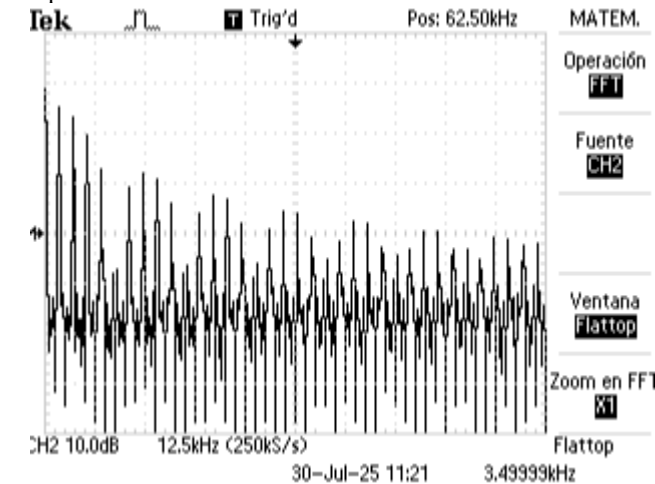


Ilustración 129 pulso en FFT.

Armonico	Frecuencia	dB
1	3,5 K	-1.79
2	7 K	-3.79
3	10,5 K	-7.39
4	14 K	-14.2
5	18 K	0
6	21 K	-35
7	24 k	-15
8	28 k	-16.2
9	31.5 k	-21.4
10	35.5 k	0

Ilustración 130 armónicos experimentales.

Para poder reconstrucción primero debemos pasar los dB a voltajes con la siguiente formula

$$V = 10^{\frac{dB}{20}} \times \sqrt{2}$$

Armonico	Frecuencia	dB	Voltaje
1	3,5 K	-1.79	1.15084013
2	7 K	-3.79	0.91414481
3	10,5 K	-7.39	0.60396949
4	14 K	-14.2	0.27574967
5	18 K	0	1.41421356
6	21 K	-35	0.02514867
7	24 k	-15	0.25148669
8	28 k	-16.2	0.21903575
9	31.5 k	-21.4	0.1203691
10	35.5 k	0	1.41421356

Ilustración 131 Voltajes.

```
T = 0.1;
f = 1/T;
A1 = 1.150840131;
A2 = 0.91414481;
A3 = 0.603969486;
A4 = 0.275749668;
A5 = 1.414213562;
A6 = 0.025148669;
A7 = 0.251486686;
A8 = 0.219035747;
A9 = 0.120369096;
A10 = 1.414213562;
t = 0:T/100:2*T;
V1 = A1;
V2 = A2 * sin(2 * pi * f * t);
V3 = A3 * sin(2 * pi * 2 * f * t);
V4 = A4 * sin(2 * pi * 3 * f * t);
V5 = A5 * sin(2 * pi * 4 * f * t);
V6 = A6 * sin(2 * pi * 5 * f * t);
V7 = A7 * sin(2 * pi * 6 * f * t);
V8 = A8 * sin(2 * pi * 7 * f * t);
V9 = A9 * sin(2 * pi * 8 * f * t);
V10 = A10 * sin(2 * pi * 9 * f * t);

VT = V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 ;

% Gráfica
figure;
plot(t, VT);
title('Señal de Pulso (80% ciclo útil) ');
xlabel('Tiempo (s)');
ylabel('Amplitud');
grid on;
```

Ilustración 132 código para reconstruir señal.

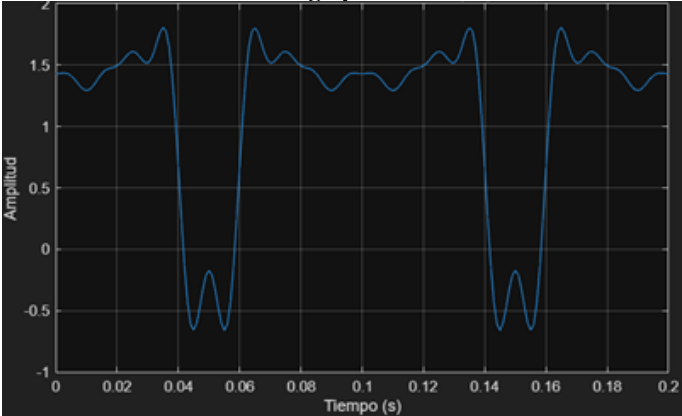


Ilustración 133 pulso creado con armónicas experimentales. Observando ambas gráficas, aunque comparten el mismo porcentaje de ciclo útil (80 %), presentan diferencias notables en

su forma. La primera imagen muestra una señal con bordes más redondeados y transiciones más suaves entre los niveles alto y bajo. Esto es característico de una señal reconstruida a partir de un número limitado de armónicos, como es común en una suma trunca de componentes de Fourier. Es probable que esta señal haya sido generada teóricamente mediante la suma de 11 armónicos, lo cual explica la suavidad en los cambios y las ondulaciones visibles alrededor de las transiciones, conocidas como "overshoot" y "ringing", producto del efecto Gibbs.

La segunda imagen, aunque también representa un pulso del 80 %, muestra picos más definidos y una señal más abrupta, lo que sugiere que proviene de datos experimentales capturados con mayor resolución o que ha sido reconstruida usando un mayor número de armónicos (o directamente desde la forma de onda medida por un osciloscopio). Esta señal parece más fiel al comportamiento real de una onda cuadrada con alto ciclo útil, manteniendo un nivel alto la mayor parte del tiempo con caídas rápidas al nivel bajo.

En resumen, aunque ambas señales tienen el mismo duty cycle, la primera representa una reconstrucción teórica basada en componentes armónicos limitados (por eso tiene una apariencia suavizada), mientras que la segunda muestra una forma más realista o experimental, con transiciones más abruptas y menos distorsión visible. Esta comparación evidencia la importancia del número de armónicos utilizados en la reconstrucción de señales periódicas, incluso cuando los parámetros fundamentales (como el duty cycle) se mantienen constantes.

## XI. ANÁLISIS

A lo largo del desarrollo experimental y teórico abordado en este informe, se llevó a cabo un análisis detallado del comportamiento de diferentes señales periódicas, entre ellas las señales senoidales, cuadradas, triangulares y de pulso, evaluando su estructura tanto en el dominio del tiempo como en el dominio de la frecuencia. Este estudio tuvo como eje central el uso de herramientas matemáticas como la transformada rápida de Fourier (FFT) y su inversa (IFFT), así como el empleo de instrumentos de medición y software como el osciloscopio digital, el generador de funciones, el software OpenChoice para la extracción de datos y la plataforma de MATLAB para procesamiento y visualización. Inicialmente, se analizaron las señales generadas por el osciloscopio y se extrajeron sus valores a través de archivos .csv, permitiendo representar las formas de onda en MATLAB y verificar si los datos medidos correspondían a las formas teóricas planteadas. En esta etapa se comprobó que los resultados eran coherentes, validando así el correcto funcionamiento del proceso de adquisición de datos. Posteriormente, se procedió al cálculo teórico de las frecuencias y amplitudes de los armónicos principales mediante fórmulas derivadas del desarrollo de series de Fourier, estableciendo las componentes espectrales ideales que componen la señal original. Este proceso se realizó para diferentes señales, destacando que en el caso de la señal senoidal no existen armónicos adicionales, mientras que en las señales cuadradas, triangulares o de pulso, su forma se compone por una serie infinita de armónicos impares o una combinación específica de componentes según su ciclo útil. Una vez definidos los armónicos, se procedió a realizar la reconstrucción de las señales utilizando un número limitado de armónicos, ya que en la práctica no es posible emplear una cantidad infinita. En este punto, se evidenció que el número de armónicos considerados influye directamente en la calidad de la reconstrucción: señales

reconstruidas con más armónicos se asemejan más a la forma ideal, mientras que aquellas construidas con pocos armónicos presentan ondulaciones, ruido en valles y crestas, y transiciones suaves que no representan con precisión el comportamiento abrupto de señales cuadradas o pulsos estrechos. Por ejemplo, al comparar una reconstrucción teórica con 11 armónicos frente a una experimental con solo 6, se aprecia una clara diferencia en la precisión de la forma de onda. Asimismo, se estudiaron los efectos de añadir un componente de corriente continua (DC) a las señales, observando cómo este afecta el nivel medio de la señal, elevando o bajando toda la forma de onda sin modificar sus frecuencias fundamentales, pero generando una componente espectral en la frecuencia 0 Hz, lo cual fue verificado mediante el análisis FFT en el osciloscopio y posteriormente en MATLAB. Se analizó también cómo varía la distribución espectral al modificar parámetros como el ciclo útil, evidenciando que una señal de pulso con un 20% de ciclo útil presenta una estructura espectral diferente a una de 50%, afectando la amplitud relativa de los armónicos, lo cual se refleja en una reconstrucción distinta aun utilizando el mismo número de componentes. La implementación de las transformadas (FFT e IFFT) en MATLAB permitió corroborar la correlación entre los espectros teóricos y los experimentales, reforzando la comprensión de cómo se relaciona el dominio del tiempo con el dominio de la frecuencia. En conclusión, este análisis permitió validar la teoría de Fourier aplicada a señales periódicas reales, visualizar sus componentes espectrales, entender las limitaciones prácticas del análisis armónico, y destacar la importancia de contar con un número suficiente de armónicos para lograr una representación adecuada de señales discontinuas o complejas. Además, la comparación entre teoría y experimento permitió identificar errores potenciales de medición, la influencia del ruido, y la relevancia del procesamiento digital como herramienta indispensable en la ingeniería para el estudio de fenómenos periódicos en sistemas eléctricos y electrónicos.

## XII. CONCLUSIÓN

A partir del estudio realizado se concluye que la descomposición y reconstrucción de señales periódicas mediante la transformada rápida de Fourier (FFT) e inversa (IFFT) es una herramienta fundamental para el análisis de sistemas en el dominio de la frecuencia. Se evidenció que señales como la cuadrada, triangular o de pulso, lejos de ser simples, están compuestas por una serie infinita de armónicos cuya distribución de frecuencia y magnitud depende directamente de su forma y características, como el ciclo útil. En el caso de señales con ciclo útil diferente al 50 %, como la de pulso con 20 %, se observó un cambio en la energía y cantidad relativa de los armónicos, afectando de manera directa la fidelidad de la reconstrucción. Se comprobó experimentalmente que la cantidad de armónicos empleados tiene un impacto determinante en la calidad de la reconstrucción: a mayor número de armónicos, más fiel será la forma de la señal respecto a su versión ideal; sin embargo, al estar limitados en la práctica a unos pocos armónicos, la señal reconstruida presenta bordes redondeados, ruido y pérdida de simetría. Además, al añadir un componente de corriente directa (DC), se constató que la señal se eleva en el dominio del tiempo y que esta adición genera una componente espectral en la frecuencia cero (0 Hz), fácilmente identificable mediante el análisis de espectro. La comparación entre los resultados teóricos y experimentales, tanto en MATLAB como en el osciloscopio, permitió validar los conceptos estudiados, identificar pequeñas discrepancias

atribuibles a ruido o a limitaciones del equipo, y destacar la importancia de contar con herramientas digitales y software especializado para complementar el análisis práctico. En resumen, este trabajo permitió comprender la relación profunda entre el dominio del tiempo y el de la frecuencia, afianzar el uso de herramientas como la FFT, visualizar los efectos de la truncación de armónicos, y aplicar estos conocimientos para el diagnóstico, diseño y verificación de señales en entornos reales de ingeniería electrónica.

## REFERENCIAS

- [1] **Oppenheim, A. V., Willsky, A. S., & Nawab, S. H.** (1997). *Signals and Systems* (2nd ed.). Prentice Hall.
  - Base teórica para el análisis de señales, series de Fourier y transformadas.
- [2] **Proakis, J. G., & Manolakis, D. G.** (2007). *Digital Signal Processing: Principles, Algorithms, and Applications* (4th ed.). Pearson.
  - Estudio detallado del procesamiento digital de señales y transformada rápida de Fourier (FFT).
- [3] **Smith, S. W.** (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing.
  - Accesible y práctica guía para comprender FFT, espectros, y filtrado digital.
- [4] **The MathWorks, Inc.** (2024). *MATLAB Documentation*. Recuperado de: <https://www.mathworks.com/help>
  - Documentación oficial para el uso de funciones como fft, ifft, abs, unwrap, angle, entre otras.
- [5] **NumPy Developers.** (2024). *NumPy Reference*. Recuperado de: <https://numpy.org/doc/>
  - Base del uso de arrays y funciones de FFT en Python: np.fft.fft, np.fft.ifft, etc.
- [6] **SciPy Developers.** (2024). *SciPy Signal Processing Module*. Recuperado de: <https://docs.scipy.org/doc/scipy/reference/signal.html>
  - Recursos para trabajar con señales digitales en Python (filtros, transformadas, etc.).
- [7] **González, R. C., & Woods, R. E.** (2008). *Digital Image Processing* (3rd ed.). Pearson.
  - Base para el análisis visual del espectro de señales y representación en imágenes.