

Análisis de Calidad de Potencia

Grupo 3

NADIA ROTBI PRADO

PABLO SEGURA FERNANDEZ

ENCARNACIÓN CERVANTES REQUENA

Universidad de Almería

Primer Cuatrimestre

Curso Académico 2025–2026



Problema Práctico 1. Análisis en el dominio del tiempo

0.1 Ejercicio 2.2.1: Cálculo del Valor RMS

Objetivo

Implementar una función en MATLAB que calcule el valor RMS (*Root Mean Square*) de una señal periódica y verificar su precisión frente al valor teórico.

Fundamento teórico

El valor eficaz o RMS de una señal $v(t)$ se define como:

$$V_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^N v_i^2} \quad (1)$$

Para una señal senoidal pura de amplitud pico V_p , el valor teórico del RMS es:

$$V_{\text{RMS,teo}} = \frac{V_p}{\sqrt{2}} \quad (2)$$

Metodología

- Se genera una señal senoidal de 50 Hz y amplitud pico de 325 V.
- Se calcula el valor del RMS teórico y práctico
- Se compara con el valor teórico (≈ 230 V) verificando un error menor al 1 %.
- Se muestran los resultados

Código MATLAB

Listing 1: Código MATLAB del Ejercicio 2.2.1: Cálculo del Valor RMS

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 2.2.1 - C lculo del Valor RMS
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez
clear; close all; clc;

%% --- Funci n para calcular RMS ---
function Vrms = calcularRMS(v)
    % Calcula el valor RMS de un vector de se al
    % Entrada: v      vector de valores de la se al
    % Salida: Vrms    valor RMS calculado
    N = length(v);
    Vrms = sqrt(sum(v.^2) / N);
end
```

```

%% --- Generaci n de se al de prueba ---
fs = 1000;           % Frecuencia de muestreo [Hz]
t = 0:1/fs:0.1;      % Duraci n de 100 ms
f = 50;              % Frecuencia [Hz]
Vp = 325;             % Amplitud pico [V]

v = Vp * sin(2*pi*f*t); % Se al senoidal

%% --- C lculo del valor RMS ---
Vrms_calc = calcularRMS(v);
Vrms_teorico = Vp / sqrt(2); % Valor RMS te rico

error_rel = abs(Vrms_calc - Vrms_teorico)/Vrms_teorico * 100;

%% --- Resultados ---
fprintf('Valor_RMS_calculado: %.2f V\n', Vrms_calc);
fprintf('Valor_RMS_te rico: %.2f V\n', Vrms_teorico);
fprintf('Error_relativo: %.4f%%\n', error_rel);

%% --- Gr fica ---
figure;
plot(t*1000, v);
xlabel('Tiempo [ms]');
ylabel('Tensi n [V]');
title(sprintf('Se al Senoidal de 50 Hz - Vrms = %.2f V (Error %.2f%%)', Vrms_calc, error_rel));
grid on;

```

Resultados y análisis

En la Figura 1 se muestra la señal generada, donde se puede apreciar que es senoidal con una amplitud pico de 325 V y un periodo de 20 ms (es decir, una frecuencia de 50 Hz). Se muestra además el valor del RMS calculado (228,67 V) y el error que es aproximadamente del 0.5 %. Con estos resultados, queda demostrado que el cálculo del RMS es el adecuado para una señal senoidal pura.

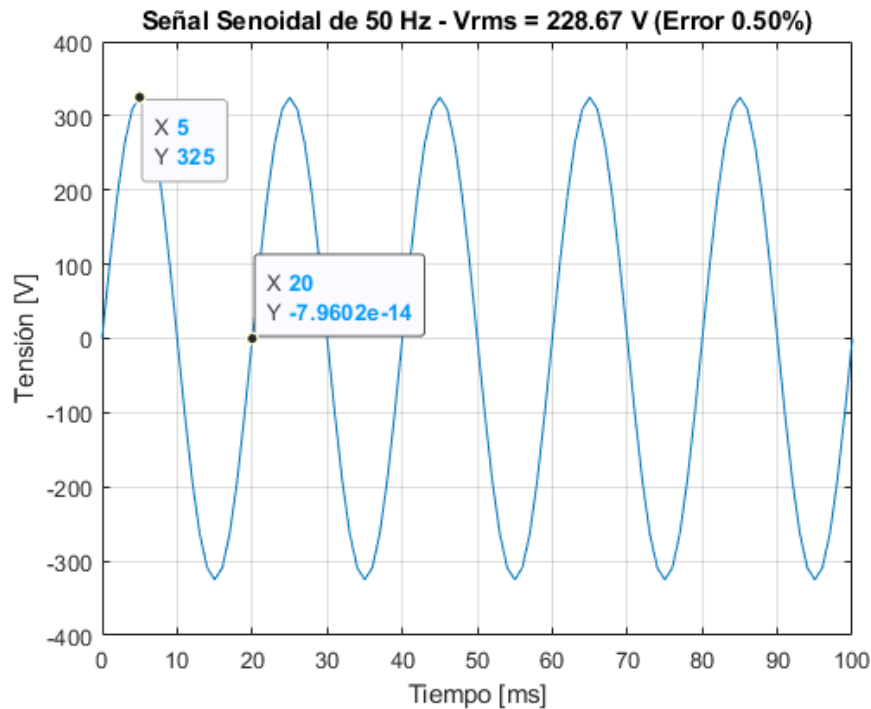


Figura 1: Señal senoidal utilizada para el cálculo del valor RMS.

0.2 Ejercicio 2.2.2: Detección de Cruces por Cero

Objetivo

Desarrollar una función que detecte los puntos de cruce por cero en una señal periódica y calcular su frecuencia.

Fundamento teórico

Un cruce por cero ocurre cuando la señal cambia de signo. Si se identifican los instantes de cruce t_i , el período medio se obtiene a partir de los intervalos entre cruces alternos:

$$T = 2 \cdot \text{mean}(t_{i+2} - t_i) \Rightarrow f = \frac{1}{T} \quad (3)$$

Código MATLAB

Listing 2: Código MATLAB del Ejercicio 2.2.2: Detección de Cruces por Cero

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 2.2.2 - Detecci n de Cruces por Cero
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez
clear; close all; clc;

%% --- Funci n para detectar cruces por cero ---
```

```

function [indices_cero, tiempos_cruce_interp, f_detectada] =
    detectarCrucesCero(v, t)
% Detecta cruces por cero con interpolación lineal para evitar
% acumular
% errores
% Entradas:
%   v : vector de la señal
%   t : vector de tiempos (misma longitud que v)
% Salidas:
%   indices_cero      : índices i tales que v(i) y v(i+1)
%   tienen distinto signo
%   tiempos_cruce_interp : tiempos estimados (en s) de los
%   cruces por cero (interpolados)
%   f_detectada       : frecuencia estimada [Hz]

s = sign(v);
s(s==0) = 1;
idx = find(diff(s) ~= 0);

if isempty(idx)
    indices_cero = [];
    tiempos_cruce_interp = [];
    f_detectada = NaN;
    return;
end

indices_cero = idx;

% Interpolación lineal para estimar el instante exacto del
% cruce v=0
tiempos_cruce_interp = zeros(size(idx));
for k = 1:length(idx)
    i = idx(k);
    t1 = t(i); t2 = t(i+1);
    v1 = v(i); v2 = v(i+1);
    % tiempo aproximado donde v = 0 por interpolación lineal
    tiempos_cruce_interp(k) = t1 - v1*(t2-t1)/(v2-v1);
end

% Estimación de la frecuencia:
dt_all = diff(tiempos_cruce_interp); % medio-período
if length(dt_all) >= 1
    T = 2 * mean(dt_all); % período completo
    f_detectada = 1 / T;
else
    f_detectada = NaN;
end
end

```

```

%% --- Generaci n de se al de prueba ---
fs = 10000;                % Frecuencia de muestreo [Hz]
t = 0:1/fs:0.1;           % Duraci n de 100 ms
f = 50;                    % Frecuencia [Hz]
Vp = 325;                  % Amplitud pico [V]

v = Vp * sin(2*pi*f*t); % Se al senoidal

%% --- Detecci n ---
[indices, t_cruce, f_est] = detectarCrucesCero(v, t);

%% --- Resultados ---
fprintf('Frecuencia_estimada: %.4f Hz\n', f_est);

%% --- Gr fica ---
figure;
plot(t*1000, v, 'b', 'LineWidth', 1.2); hold on;
plot(t_cruce*1000, zeros(size(t_cruce)), 'ro', 'MarkerFaceColor',
     'r'); % marca en y=0
xlabel('Tiempo [ms]'); ylabel('Tensi n [V]');
title(sprintf('Cruces por cero (interpolados) - f = %.4f Hz',
             f_est));
grid on;

```

Resultados

La función detecta correctamente los puntos donde la señal cruza el eje horizontal, marcándolos en la Figura 2. La frecuencia estimada es de 50.0 Hz, confirmando el correcto funcionamiento del algoritmo. Cabe destacar que se tuvo que utilizar una interpolación lineal entre puntos en los que cambia el signo de la señal porque utilizando directamente la función "sign" de Matlab, se acumulaban errores. Dichos errores llevaban a una detección defectuosa al representar más periodos de la señal (que es la misma que en el ejercicio anterior). Además, se tuvo que aumentar la frecuencia de muestreo a 10000 Hz para facilitar la búsqueda de los cruces por cero.

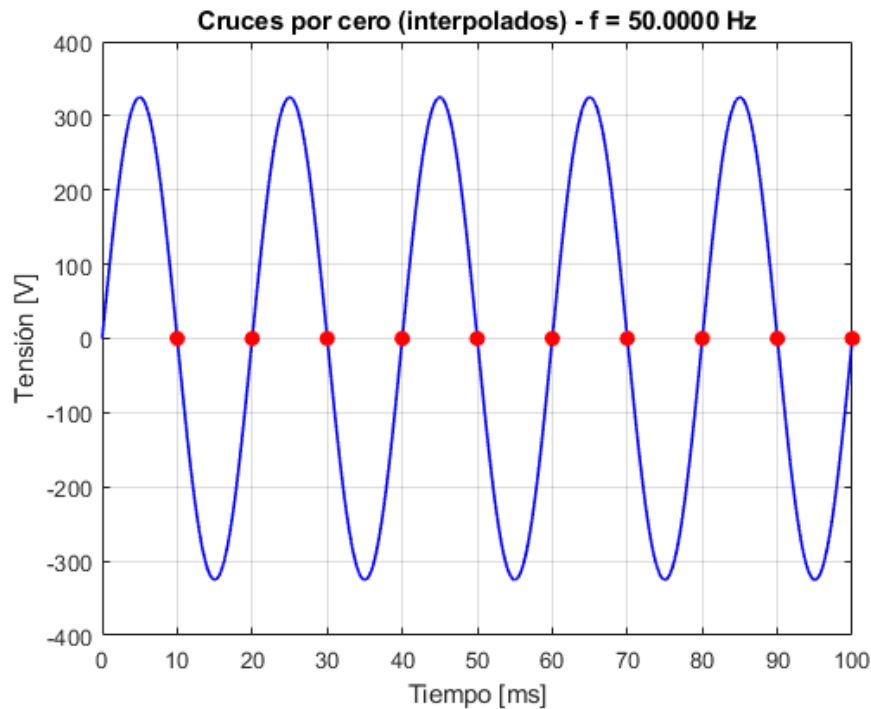


Figura 2: Detección de cruces por cero en una señal senoidal de 50 Hz.

0.3 Ejercicio 2.2.3: Detección de Variaciones de Tensión (RMS Deslizante)

Objetivo

Implementar un cálculo de RMS con ventana deslizante para detectar huecos o sobretensiones en una señal.

Código MATLAB

Listing 3: Código MATLAB del Ejercicio 2.2.3: RMS Deslizante

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 2.2.3 - Detecci n de Variaciones de Tensi n
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez
clear; close all; clc;

%% --- Funci n RMS deslizante ---
function [v_rms, t_rms] = rmsDeslizante(v, fs, ventana_ms)
    % Calcula el RMS deslizante de una se al
    % Entradas:
    %   v       se al
    %   fs      frecuencia de muestreo [Hz]
    %   ventana_ms  tama o de la ventana [ms]
```

```

% Salidas:
%   v_rms      vector de valores RMS
%   t_rms      vector de tiempos asociados

Nw = round((ventana_ms / 1000) * fs); % muestras por
    ventana
v2 = v.^2;
v_rms = sqrt(movmean(v2, Nw));
t_rms = (0:length(v)-1)/fs;
end

%% --- Se al con hueco de tensi n para ver el c lculo de RMS
    en varios casos ---
fs = 2000; t_total = 0.2;
t = 0:1/fs:t_total;
f = 50; Vp = 325;

v = Vp * sin(2*pi*f*t);
% Hueco del 50% entre 50 y 100 ms
v((t >= 0.05) & (t < 0.1)) = 0.5 * Vp * sin(2*pi*f*t((t >= 0.05)
    & (t < 0.1)));

%% --- Calcular RMS deslizando ---
[vrms, t_rms] = rmsDeslizando(v, fs, 20); % ventana 20 ms

%% --- Gr ficas ---
figure;
subplot(2,1,1);
plot(t*1000, v, 'k');
xlabel('Tiempo [ms]'); ylabel('Tensi n [V]');
title('Se al con hueco de tensi n (50%-50ms)'); grid on;

subplot(2,1,2);
plot(t_rms*1000, vrms, 'b', 'LineWidth', 1.2);
hold on;
yline(230, 'g--', 'Nominal (230V)');
yline(207, 'r--', 'L mite -10% (207V)');
xlabel('Tiempo [ms]'); ylabel('RMS [V]');
title('Valor RMS deslizando (Ventana 20ms)');
legend('RMS', 'Nominal', 'L mite -10%', 'Location', 'best');
grid on;

```


Resultados

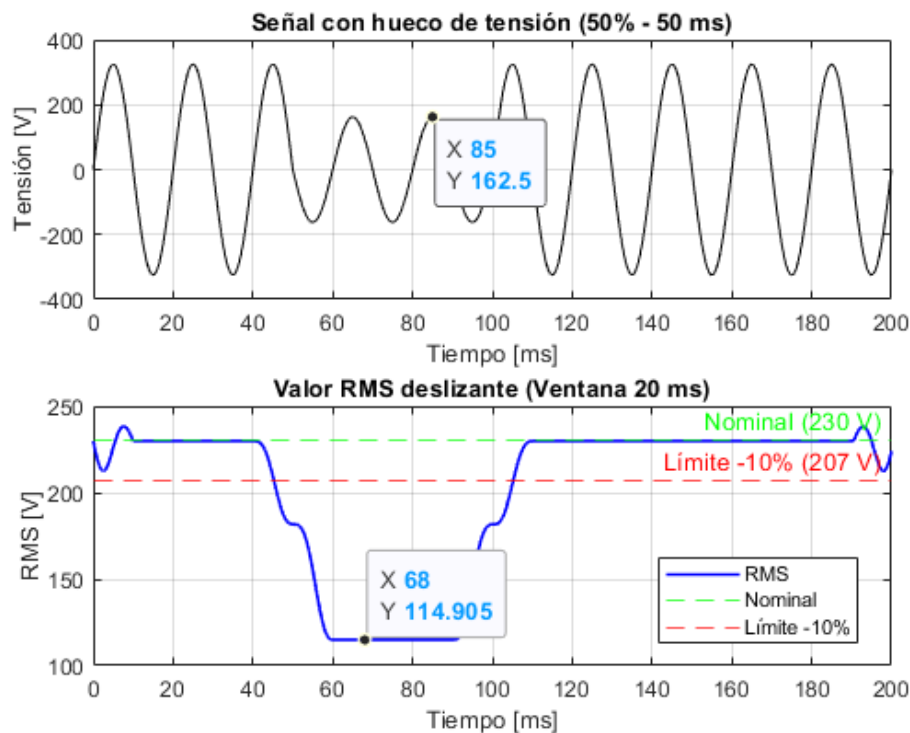


Figura 3: Evolución del valor RMS deslizante ante un hueco de tensión.

La señal presenta un hueco del 50 % durante 50 ms. El valor RMS deslizante disminuye aproximadamente hasta 115 V, recuperándose al valor nominal tras el evento. La ventana de 20 ms permite seguir la variación con buena resolución temporal. Como se puede apreciar en la Figura 3, si se eligiese un límite del 10 % para la detección de fallos en la red, el algoritmo sería capaz de producir el valor del RMS buscado a tiempo real y con una simple lógica condicional, cuando se da la condición de que dicho valor baje del límite establecido, se podría lanzar una señal de alarma y registrar el evento.

0.4 Actividad Guiada 2.3: Análisis de un Hueco de Tensión

Objetivo

Generar y analizar una señal con un hueco de tensión típico en una red de 230 V y 50 Hz.

Código MATLAB

Listing 4: Código MATLAB de la Actividad Guiada 2.3: Hueco de Tensión

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Actividad Guiada 2.3 - An lisis de un Hueco de Tensi n
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
Pablo Segura
```

```

% Fernandez
clear; close all; clc;

%% --- Parámetros básicos ---
fs = 2000;           % Frecuencia de muestreo [Hz]
t_total = 0.2;       % Duración total [s]
t = 0:1/fs:t_total;
f = 50; Vp = 325;

%% --- Generar señal base ---
v = Vp * sin(2*pi*f*t);

%% --- Introducir hueco de tensión ---
inicio_hueco = 0.05; % 50 ms
duracion_hueco = 0.05; % 50 ms (2.5 ciclos)
fin_hueco = inicio_hueco + duracion_hueco;

idx_hueco = (t >= inicio_hueco) & (t < fin_hueco);
v(idx_hueco) = 0.5 * Vp * sin(2*pi*f*t(idx_hueco)); % profundidad
50%

%% --- Función RMS deslizante (reutilizada del ejercicio
anterior) ---
function [v_rms, t_rms] = rmsDeslizante(v, fs, ventana_ms)
    Nw = round((ventana_ms / 1000) * fs);
    v2 = v.^2;
    v_rms = sqrt(movmean(v2, Nw));
    t_rms = (0:length(v)-1)/fs;
end

%% --- Cálculo del RMS deslizante ---
[vrms, t_rms] = rmsDeslizante(v, fs, 20);

%% --- Cálculos de análisis ---
[Vrms_min, idx_min] = min(vrms);
t_inicio_evento = t(find(vrms < 230*0.99, 1, 'first'));
t_fin_evento = t(find(vrms > 230*0.99 & t > fin_hueco, 1, 'first'
));
duracion_evento = t_fin_evento - t_inicio_evento;

%% --- Mostrar resultados ---
fprintf('Inicio del hueco detectado: %.2f ms\n', t_inicio_evento
*1000);
fprintf('Valor RMS mínimo: %.2f V\n', Vrms_min);
fprintf('Duración hasta recuperación: %.2f ms\n',
duracion_evento*1000);
fprintf('Hueco supera límite -10%: %s\n', string(Vrms_min <
207));

%% --- Visualización ---
figure;

```

```

subplot(2,1,1);
plot(t*1000, v, 'k');
xlabel('Tiempo [ms]'); ylabel('Tensi n [V]');
title('Se al con Hueco de Tensi n (50%, 50 ms)');
grid on;

subplot(2,1,2);
plot(t_rms*1000, vrms, 'b', 'LineWidth', 1.3); hold on;
yline(230, 'g--', 'Nominal 230 V');
yline(207, 'r--', 'L mite -10%');
xline(inicio_hueco*1000, 'k--', 'Inicio Hueco');
xline(fin_hueco*1000, 'k--', 'Fin Hueco');
xlabel('Tiempo [ms]'); ylabel('RMS [V]');
title('Valor RMS Deslizante (Ventana 20 ms)');
legend('RMS', 'Nominal', 'L mite -10%', 'Ubicaci n del Hueco');
grid on;

```

Resultados

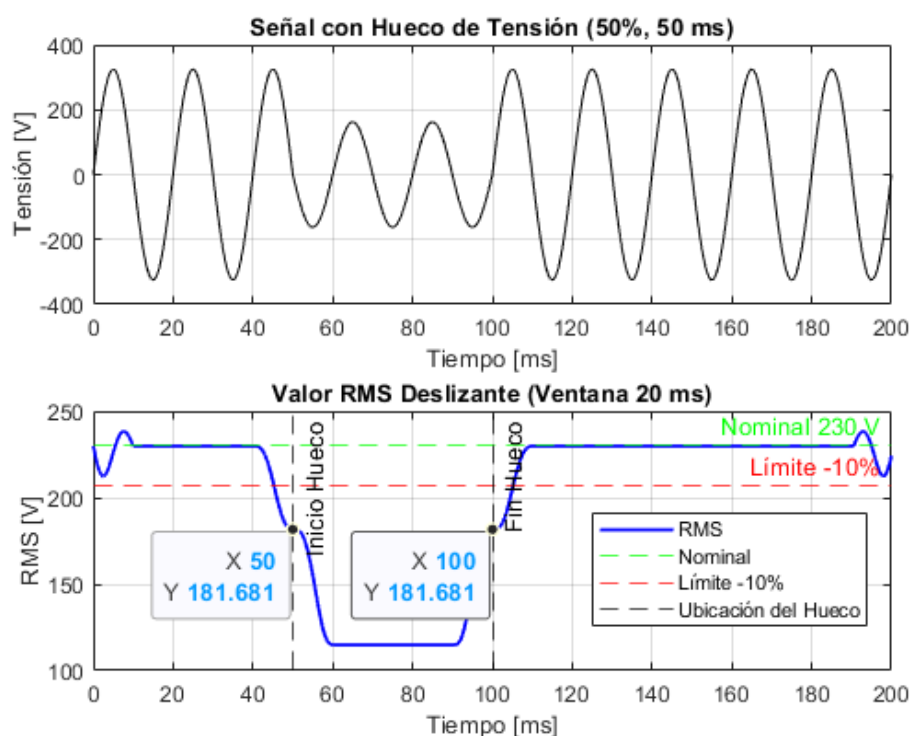


Figura 4: Análisis del hueco de tensión mediante RMS deslizante.

El análisis RMS deslizante permite identificar con precisión los huecos de tensión y evaluar su severidad. El evento analizado reduce el valor eficaz a la mitad del nominal y se mantiene por 2.5 ciclos (50 ms), siendo claramente visible en la evolución del RMS. Este tipo de análisis resulta esencial en estudios de **calidad de energía eléctrica**. Como se ha reutilizado la señal generada en el ejercicio anterior y la misma función, la única variación introducida ha sido el cálculo de los tiempos del hueco, que como se comentaba en el

ejercicio anterior, se hace mediante lógico condicional simple (guardando los tiempos en los que el valor de RMS baja con respecto al límite propuesto).

Preguntas propuestas

- ¿En qué momento se detecta el inicio del hueco?: A los 50 ms.
- ¿Cuál es el valor RMS mínimo alcanzado?: 114,905 V.
- ¿Cuánto tarda la señal en volver al valor nominal?: Aproximadamente 60 ms (desde 45 ms a 105 ms).
- ¿El hueco supera el límite del-10 %? ¿Por cuánto tiempo?: Sí, durante unos 50 ms.

Problema Práctico 2. Análisis en el dominio de la frecuencia

0.5 Ejercicio 2.1: Análisis Espectral Básico

El objetivo de este apartado es calcular y representar el espectro de amplitud de una señal senoidal pura empleando la Transformada Rápida de Fourier (FFT). Se sigue un procedimiento estandarizado para obtener el *semiespectro*, normalizar por el número de muestras y duplicar las componentes intermedias (excepto DC y $f_m/2$), de modo que la amplitud de una señal real se interprete correctamente en el dominio frecuencial.

0.5.1 Fundamentos teóricos

Sea $x(t)$ una señal temporal muestreada a una frecuencia f_s durante un intervalo T , con $N = T f_s$ muestras y vector temporal definido como:

$$t_n = \frac{n}{f_s}, \quad n = 0, 1, \dots, N - 1. \quad (4)$$

En esta práctica se genera una señal senoidal pura de frecuencia $f_0 = 50$ Hz y amplitud pico $V_{\text{pico}} = 325$ V, de la forma:

$$x(t) = V_{\text{pico}} \sin(2\pi f_0 t). \quad (5)$$

La Transformada Rápida de Fourier (FFT) de la señal discreta $x[n]$ permite representar la información temporal en el dominio de la frecuencia:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N - 1. \quad (6)$$

El espectro de amplitud normalizado se define como:

$$M_2[k] = \frac{|X[k]|}{N}, \quad (7)$$

donde $|X[k]|$ representa la magnitud del componente complejo de la FFT y N es el número total de muestras, empleado para la normalización de amplitudes.

Dado que la señal $x(t)$ es real, el espectro resultante es simétrico. Por tanto, sólo se considera la mitad positiva (*semiespectro*) mediante:

$$k = 0, 1, \dots, \left\lfloor \frac{N}{2} \right\rfloor.$$

Para conservar la energía total del espectro original, se duplican las amplitudes de las componentes intermedias, excluyendo la componente de continua ($k = 0$) y, si N es par, la componente de Nyquist ($k = N/2$):

$$M[0] = M_2[0], \quad M[k] = 2 M_2[k] \text{ para } 1 \leq k \leq \left\lfloor \frac{N}{2} \right\rfloor - 1, \quad (8)$$

$$M\left[\frac{N}{2}\right] = M_2\left[\frac{N}{2}\right]. \quad (9)$$

El vector de frecuencias asociado a cada componente espectral se calcula como:

$$f[k] = \frac{k}{N} f_s, \quad k = 0, 1, \dots, \left\lfloor \frac{N}{2} \right\rfloor, \quad (10)$$

y la resolución frecuencial, es decir, la distancia entre dos componentes consecutivas del espectro, se obtiene mediante:

$$\Delta f = \frac{f_s}{N}. \quad (11)$$

Finalmente, para determinar la posición exacta de un armónico o componente frecuencial, se busca el **índice más cercano** al valor teórico nf_0 dentro del vector de frecuencias calculado, siendo n el orden del armónico y f_0 la frecuencia fundamental.

0.5.2 Metodología

- **Generación de la señal:** se define una senoidal pura de frecuencia $f_0 = 50$ Hz, amplitud pico $V_{\text{pico}} = 325$ V.
- **Parámetros de muestreo:** $f_s = 2000$ Hz, duración $T = 1$ s, por lo que $N = T f_s = 2000$ y $\Delta f = 1$ Hz.
- **Cálculo de FFT:** se computa $X[k] = \text{FFT}\{x[n]\}$ y se obtiene $M_2[k] = |X[k]|/N$.
- **Semiespectro:** se conserva sólo la primera mitad (frecuencias no negativas) y se duplica la amplitud de las componentes intermedias.
- **Representación:** se grafica M frente a f mediante **stem** hasta 500 Hz.
- **Configuración empleada:**

Frecuencia de la señal	$f_0 = 50$ Hz
Amplitud pico	$V_{\text{pico}} = 325$ V
Frecuencia de muestreo	$f_s = 2000$ Hz
Duración	$T = 1$ s
Número de muestras	$N = 2000$
Resolución en frecuencia	$\Delta f = 1$ Hz

0.5.3 Resultados

Dado que la señal es monofrecuencial y N proporciona $\Delta f = 1$ Hz, se espera:

- Un **pico único** en $f = 50$ Hz.
- La componente DC (0 Hz) prácticamente nula.
- El resto de frecuencias con magnitud despreciable (ruido numérico).

Para ello, se emplea el siguiente código de Matlab:

Listing 5: Código MATLAB del Ejercicio 2.1

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 2.1 - An lisis Espectral B sico
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez

clear; close all; clc;

% --- Par metros de la se al ---
fs = 2000;      % Frecuencia de muestreo [Hz]
f0 = 50;       % Frecuencia de la se al [Hz]
Vp = 325;      % Amplitud pico [V]
T = 1;         % Duraci n [s]
t = 0:1/fs:T-1/fs;
N = length(t);

% --- Generar se al senoidal pura ---
v = Vp * sin(2*pi*f0*t);

% --- Calcular FFT y normalizar ---
V = fft(v);
M = abs(V)/N;      % Normalizaci n por N

% --- Quedarse solo con frecuencias positivas ---
M = M(1:N/2+1);   % Primera mitad del espectro
M(2:end-1) = 2*M(2:end-1); % Duplicar las intermedias
f = fs*(0:N/2)/N;  % Vector de frecuencias

% --- Graficar el espectro hasta 500 Hz ---
figure;
stem(f, M, 'filled');
xlim([0 500]); grid on;
xlabel('Frecuencia (Hz)');
ylabel('Magnitud (V)');
title('Espectro de una senoidal pura de 50 Hz');
```

En la Figura 5 se incluirá el semiespectro de amplitud hasta 500 Hz, obtenido con `stem`:

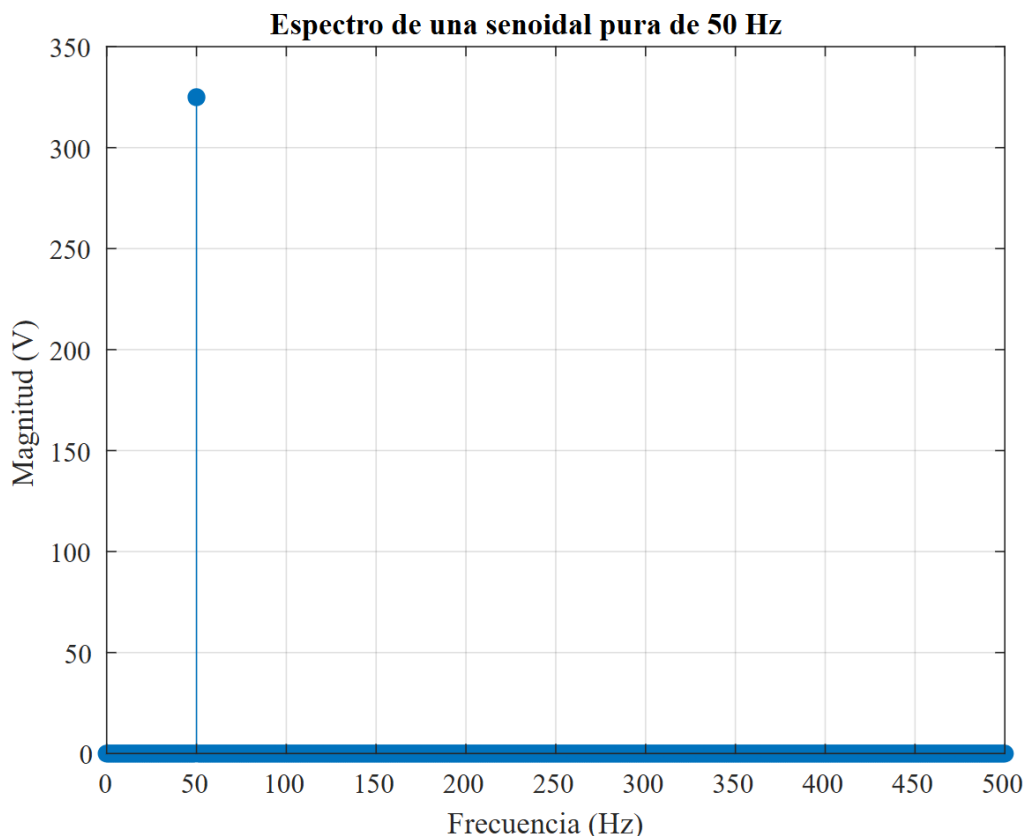


Figura 5: Semiespectro de amplitud de la señal senoidal pura de 50 Hz (hasta 500 Hz).

0.6 Ejercicio 2.2: Análisis de Armónicos

El objetivo de este apartado es diseñar una **función en MATLAB** capaz de identificar y cuantificar los armónicos presentes en una señal periódica, y calcular su distorsión armónica total (*THD*). Esta herramienta servirá como base para el análisis del siguiente ejercicio (2.3), donde se aplicará a una señal compuesta con armónicos conocidos.

0.6.1 Fundamento teórico

Los armónicos son componentes de frecuencia múltiplos enteros de la fundamental, definidos por:

$$f_n = n f_0, \quad n = 1, 2, 3, \dots$$

Para determinar sus magnitudes, se aplica la Transformada Rápida de Fourier (FFT), que descompone la señal en sus componentes sinusoidales.

La distorsión armónica total se calcula mediante:

$$\text{THD} = 100 \times \frac{\sqrt{\sum_{n=2}^N V_n^2}}{V_1} \quad (12)$$

donde V_1 es la amplitud de la componente fundamental y V_n las amplitudes de los armónicos de orden superior.

0.6.2 Descripción de la función

La función `analisis_armonicos.m` recibe como entrada:

- **x**: vector de la señal temporal.
- **fs**: frecuencia de muestreo.
- **f0**: frecuencia fundamental (por defecto 50 Hz).
- **maxH**: número de armónicos a analizar (por defecto 15).

y devuelve:

- Una **tabla** con el número de armónico, frecuencia exacta y magnitud.
- El valor de **THD** expresado en porcentaje.
- La resolución frecuencial $\Delta f = f_s/N$.

La función implementa el cálculo de la FFT, selecciona el semiespectro positivo, normaliza por el número de muestras y duplica las componentes intermedias, siguiendo el criterio de análisis de señales reales.

0.6.3 Código de la función en MATLAB

Listing 6: Función `analisis_armonicos.m`

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 2.2: An lisis de Arm nicos
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez

function [tabla, THD_pct, df] = analisis_armonicos(x, fs, f0,
maxH)
% ANALISIS_ARMONICOS - Identifica y cuantifica arm nicos
% espec ficos.
% -----
% Entradas:
%   x   : se al temporal
%   fs   : frecuencia de muestreo [Hz]
%   f0   : frecuencia fundamental [Hz]
%   maxH: n mero de arm nicos (por defecto, 15)
%
% Salidas:
%   tabla   : arm nico , frecuencia y magnitud
%   THD_pct : distorsi n arm nica total (%)
%   df      : resoluci n frecuencial (fs/N)

if nargin < 4 || isempty(maxH), maxH = 15; end
if nargin < 3 || isempty(f0), f0 = 50; end

x = x(:); N = numel(x); df = fs / N;

% --- FFT y semiespectro ---
X = fft(x); M2 = abs(X)/N;
```



```

if mod(N,2)==0
    M = M2(1:N/2+1);
    if numel(M)>2, M(2:end-1)=2*M(2:end-1); end
    f = fs*(0:N/2)/N;
else
    M = M2(1:(N+1)/2);
    if numel(M)>1, M(2:end)=2*M(2:end); end
    f = fs*(0:(N-1)/2)/N;
end

% --- B squeda de arm nicos ---
orden = (1:maxH).';
f_obj = orden*f0; f_obj(f_obj>f(end))=f(end);
idx = arrayfun(@(ft) nearest_index(f,ft), f_obj);
f_medida = f(idx); Vn = M(idx);

% --- C lculo del THD ---
V1 = Vn(1);
THD_pct = 100*sqrt(sum(Vn(2:end).^2))/V1;

% --- Tabla de salida ---
tabla = table(orden, f_medida(:), Vn(:), ...
    'VariableNames', {'Armonico','Frecuencia_Hz','Magnitud_V'});
end

function k = nearest_index(vec, value)
    [~,k] = min(abs(vec - value));
end

```

La función determina con precisión la magnitud de cada armónico y permite evaluar el grado de distorsión presente en una señal eléctrica. Se empleará en el ejercicio siguiente para analizar una señal con armónicos del 3º y 5º orden.

0.7 Ejercicio 2.3: Generar y Analizar Señal con Armónicos

0.7.1 Objetivo

Generar una señal compuesta por varios armónicos conocidos (3º y 5º orden) y analizar su contenido frecuencial mediante la función `analisis_armonicos`. Además, calcular y comparar el THD obtenido con el valor teórico esperado del 18 %.

0.7.2 Procedimiento

1. Se define la señal base de 50 Hz con amplitud pico de 325 V.
2. Se añaden los armónicos:
 - 3º armónico (150 Hz) = 15 % del fundamental.
 - 5º armónico (250 Hz) = 10 % del fundamental.

3. Se calcula la FFT con frecuencia de muestreo de 2000 Hz y duración de 0,5s.
4. Se usa la función `analisis_armonicos` para obtener la tabla de magnitudes y el THD.
5. Se representan las gráficas:
 - **Superior:** señal temporal (primeros 4 ciclos).
 - **Inferior:** espectro de los primeros 10 armónicos.

0.7.3 Código MATLAB empleado

Listing 7: Script de análisis del Ejercicio 2.3

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 2.3 - Generar y Analizar Se al con Arm nicos
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez

clear; close all; clc;

fs = 2000; f0 = 50; Vp = 325; T = 0.5;
t = 0:1/fs:T-1/fs; N = numel(t);

% Se al compuesta: fundamental + 3 (15%) + 5 (10%)
v = Vp*sin(2*pi*f0*t) ...
    + 0.15*Vp*sin(2*pi*3*f0*t) ...
    + 0.10*Vp*sin(2*pi*5*f0*t);

% An lisis de arm nicos (1..10)
[tabla, THD_pct, df] = analisis_armonicos(v, fs, f0, 10);

% Calcular % respecto al fundamental
V1 = tabla.Magnitud_V(tabla.Armonico==1);
Porc = 100 * (tabla.Magnitud_V / V1);
tabla.Porcentaje = round(Porc,2);

disp(tabla);
fprintf('THD medido = %.2f%% | THD te rico = 18%%\n', THD_pct);

% --- Gr ficas ---
figure('Name','Ejercicio_2.3');
subplot(2,1,1);
t_ms = t*1000;
plot(t_ms(t_ms<=4*(1000/f0)), v(t_ms<=4*(1000/f0)), 'LineWidth',1);
xlabel('Tiempo(ms)'); ylabel('Voltaje(V)');
title('Se al temporal(primeros 4 ciclos)'); grid on;

subplot(2,1,2);
```

```
bar(tabla.Armonico, tabla.Magnitud_V);
xlabel('Número de Armónico'); ylabel('Magnitud (V)');
title('Espectro de Armónicos (1..10)'); grid on;
```

0.7.4 Resultados esperados

La señal presenta una forma de onda distorsionada, con picos en las frecuencias:

$$f_1 = 50 \text{ Hz}, \quad f_3 = 150 \text{ Hz}, \quad f_5 = 250 \text{ Hz}.$$

El valor calculado de THD debe aproximarse al teórico de:

$$\text{THD}_{\text{teórico}} = \sqrt{0,15^2 + 0,10^2} \times 100 = 18 \, \%.$$

0.7.5 Figuras

En este apartado se muestra la figura generadas con MATLAB:

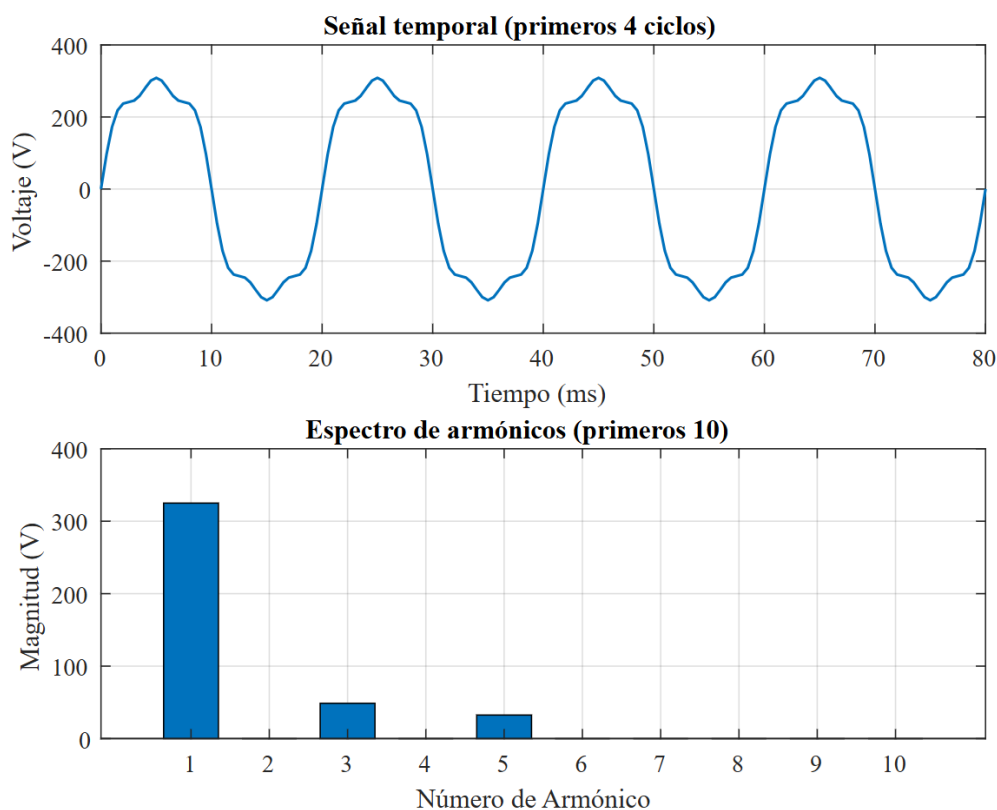


Figura 6: Señal temporal (superior) y espectro de armónicos (inferior).

El uso de la función `análisis_armonicos` permite automatizar el cálculo de las magnitudes y del THD de una señal distorsionada. Los resultados obtenidos concuerdan con la teoría, observándose un valor de THD 18 % y una clara distorsión de la forma de onda debido a los armónicos del 3º y 5º orden.

0.8 Actividad 3.4: Comparar Diferentes Cargas

Comparar cuantitativamente el contenido armónico de tres señales que simulan distintos tipos de carga:

1. **Carga lineal (ideal):** solo fundamental de 50 Hz.
2. **Carga con distorsión moderada:** fundamental + 3º (10 %) + 5º (5 %).
3. **Carga altamente distorsionada:** fundamental + 3º (25 %) + 5º (15 %) + 7º (10 %).

Se calculará el *Total Harmonic Distortion* (THD) y se verificará el cumplimiento del límite del 8 % indicado en normativa.

0.8.1 Diseño del experimento y parámetros

- Fundamental $f_0 = 50$ Hz, amplitud pico $V_{\text{pico}} = 325$ V.
- Muestreo: $f_s = 2000$ Hz, duración $T = 1$ s ($\Rightarrow \Delta f = 1$ Hz).
- Señales:
 - **Lineal:** $v_{\text{lin}}(t) = V_p \sin(2\pi f_0 t)$.
 - **Moderada:** $+ 0,10 V_p \sin(2\pi \cdot 3f_0 t) + 0,05 V_p \sin(2\pi \cdot 5f_0 t)$.
 - **Alta:** $+ 0,25 V_p \sin(2\pi \cdot 3f_0 t) + 0,15 V_p \sin(2\pi \cdot 5f_0 t) + 0,10 V_p \sin(2\pi \cdot 7f_0 t)$.

0.8.2 Procedimiento en MATLAB

1. Generar las tres señales anteriores.
2. Aplicar la función `analisis_armonicos` (o *fallback* equivalente) para obtener:
 - Tabla por señal con orden, frecuencia exacta y magnitud de armónicos.
 - THD (%).
3. Construir una **tabla comparativa** con THD y magnitudes de 1º, 3º, 5º y 7º armónico.
4. Representar los **tres espectros** (0–500 Hz) en subplots.
5. Evaluar qué cargas superan el límite del 8 % de THD.

0.8.3 Código MATLAB (autocontenido)

Listing 8: Script de comparación de cargas (Actividad 3.4)

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% 3.4 - Actividad Guiada: Comparar Diferentes Cargas
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez

clear; close all; clc;
```

```

% --- Par metros comunes ---
fs = 2000; f0 = 50; Vp = 325; T = 1.0;
t = 0:1/fs:T-1/fs;

% --- Se ales a comparar ---
v_lin = Vp*sin(2*pi*f0*t); %
    Lineal (ideal)
v_mod = Vp*sin(2*pi*f0*t) + 0.10*Vp*sin(2*pi*3*f0*t) ... %
    Moderada
    + 0.05*Vp*sin(2*pi*5*f0*t);
v_alt = Vp*sin(2*pi*f0*t) + 0.25*Vp*sin(2*pi*3*f0*t) ... % Alta
    + 0.15*Vp*sin(2*pi*5*f0*t) + 0.10*Vp*sin(2*pi*7*f0*t);

% --- An lisis de arm nicos (1..15) ---
maxH = 15;
use_func = exist(' analisis_armonicos ', 'file')==2;

if use_func
    [Tlin, THDlin] = analisis_armonicos(v_lin, fs, f0, maxH);
    [Tmod, THDmod] = analisis_armonicos(v_mod, fs, f0, maxH);
    [Talt, THDalt] = analisis_armonicos(v_alt, fs, f0, maxH);
else
    [Tlin, THDlin] = analisis_armonicos_fb(v_lin, fs, f0, maxH);
    [Tmod, THDmod] = analisis_armonicos_fb(v_mod, fs, f0, maxH);
    [Talt, THDalt] = analisis_armonicos_fb(v_alt, fs, f0, maxH);
end

% --- Tabla comparativa (se rellena en MATLAB y se copia a LaTeX)
% ---
getV = @(Tab,n) Tab.Magnitud_V(Tab.Armonico==n);
V1 = [getV(Tlin,1); getV(Tmod,1); getV(Talt,1)];
V3 = [getV(Tlin,3); getV(Tmod,3); getV(Talt,3)];
V5 = [getV(Tlin,5); getV(Tmod,5); getV(Talt,5)];
V7 = [getV(Tlin,7); getV(Tmod,7); getV(Talt,7)];
THD = [THDlin; THDmod; THDalt];

Carga = {'Lineal (ideal)', 'Distorsion moderada', 'Altamente
    distorsionada'};
TablaComp = table(Carga, round(THD,2), round(V1,3), round(V3,3),
    ...
    round(V5,3), round(V7,3), ...
    'VariableNames', {'Carga', 'THD_pct', 'V1_V', 'V3_V', 'V5_V', 'V7_V'
    });
disp(TablaComp);

% --- Gr ficas: espectros 0..500 Hz en subplots ---
[f_lin,M_lin] = one_sided_spectrum(v_lin, fs);
[f_mod,M_mod] = one_sided_spectrum(v_mod, fs);
[f_alt,M_alt] = one_sided_spectrum(v_alt, fs);

figure('Name','Espectros','Color','w');

```

```

subplot(3,1,1); stem(f_lin,M_lin,'filled'); xlim([0 500]); grid
on;
title(sprintf('Lineal □ THD=%.2f%%',THDlin)); xlabel('Hz');
ylabel('V');

subplot(3,1,2); stem(f_mod,M_mod,'filled'); xlim([0 500]); grid
on;
title(sprintf('Moderada □ THD=%.2f%%',THDmod)); xlabel('Hz');
ylabel('V');

subplot(3,1,3); stem(f_alt,M_alt,'filled'); xlim([0 500]); grid
on;
title(sprintf('Alta □ THD=%.2f%%',THDalt)); xlabel('Hz');
ylabel('V');

saveas(gcf,'espectros_cargas.png');

% ----- Helpers (semiespectro y fallback) -----
function [f,M]=one_sided_spectrum(x,fs)
x=x(:); N=numel(x); X=fft(x); M2=abs(X)/N;
if mod(N,2)==0
    M=M2(1:N/2+1); if numel(M)>2, M(2:end-1)=2*M(2:end-1); end
    f=fs*(0:N/2)/N;
else
    M=M2(1:(N+1)/2); if numel(M)>1, M(2:end)=2*M(2:end); end
    f=fs*(0:(N-1)/2)/N;
end
end

function [tabla,THD_pct]= analisis_armonicos_fb(x,fs,f0,maxH)
[f,M]=one_sided_spectrum(x,fs);
ord=(1:maxH)'; fobj=ord*f0; fobj(fobj>f(end))=f(end);
idx=arrayfun(@(u) find(abs(f-u)==min(abs(f-u)),1), fobj);
f_med=f(idx); Vn=M(idx); V1=Vn(1);
THD_pct=100*sqrt(sum(Vn(2:end).^2))/V1;
tabla=table(ord,f_med(:),Vn(:),'VariableNames', ...
    {'Armonico','Frecuencia_Hz','Magnitud_V'});
end

```

0.8.4 Discusión y Resultados

En la Figura 7 se presentan los valores de distorsión armónica total (THD) y las magnitudes principales de los armónicos obtenidos para las tres cargas analizadas. Los resultados experimentales confirman las tendencias teóricas previstas.

Carga	THD_pct	v1_v	v3_v	v5_v	v7_v
{'Lineal (ideal)'} }	0	325	0	0	0
{'Distorsión moderada'} }	11.18	325	32.5	16.25	0
{'Altamente distorsionada'}	30.82	325	81.25	48.75	32.5

Límite normativo de THD = 8.0 %

- Carga lineal: Cumple
- Carga con distorsión moderada: SUPERA
- Carga altamente distorsionada: SUPERA

Figura 7: Tabla comparativa con los resultados de MATLAB.

En la Figura 8, se observa que para la **carga lineal (ideal)**, la señal está compuesta exclusivamente por la componente fundamental de 50 Hz con una amplitud de 325 V, sin presencia de armónicos. El valor de THD obtenido es **0 %**, lo que indica una forma de onda perfectamente senoidal y, por tanto, libre de distorsión. Esta situación representa el caso ideal de una carga puramente resistiva, donde la corriente y el voltaje son proporcionales y sinusoidales.

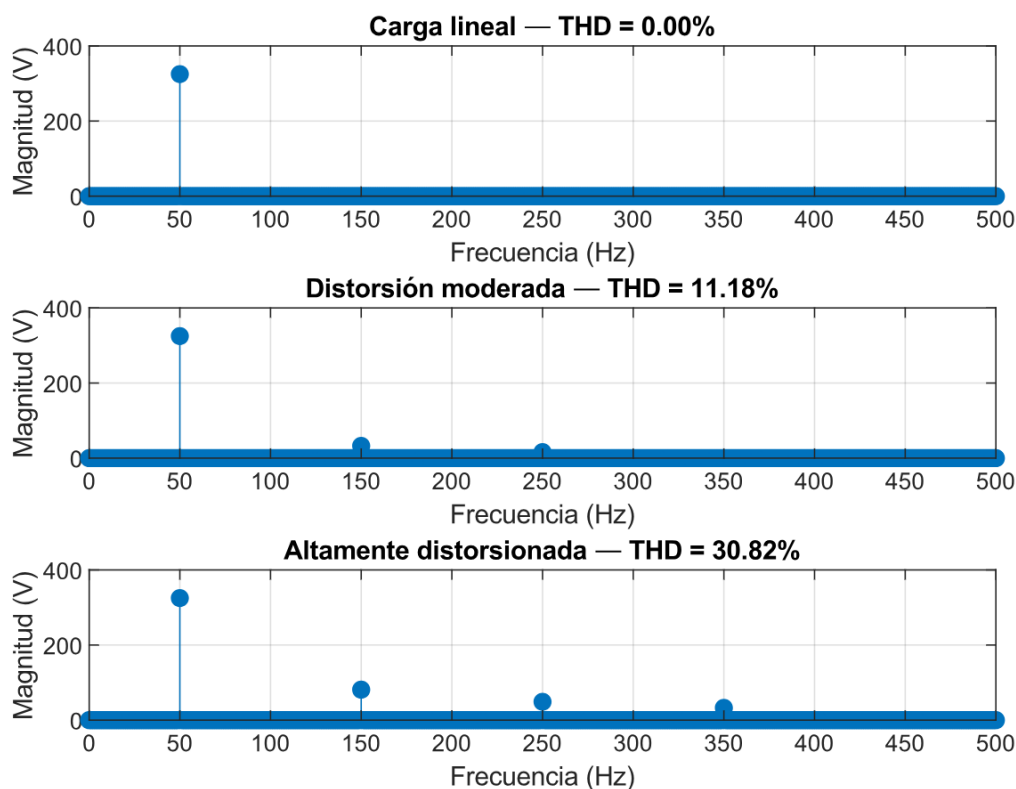


Figura 8: Espectros de las tres señales.

En la **carga con distorsión moderada**, se introducen componentes armónicas del 3º y 5º orden, con magnitudes del 10 % y 5 % del fundamental, respectivamente. El espectro

resultante muestra picos en las frecuencias de 150 Hz y 250 Hz, correspondientes a los armónicos esperados. El THD calculado fue de **11,18 %**, un valor ligeramente superior al 11,2 % teórico estimado a partir de la expresión:

$$\text{THD}_{\text{teórico}} = \sqrt{(0,10)^2 + (0,05)^2} \times 100 = 11,18 \%$$

Este resultado valida la correcta implementación del método de análisis y refleja una distorsión leve, aunque suficiente para **superar el límite normativo del 8 %** establecido por las normas de calidad de energía eléctrica.

Finalmente, la **carga altamente distorsionada** incluye armónicos de 3º, 5º y 7º orden, con amplitudes relativas del 25 %, 15 % y 10 % del fundamental, respectivamente. El espectro muestra máximos pronunciados en 150 Hz, 250 Hz y 350 Hz, reflejando una distorsión severa de la forma de onda. El THD obtenido, **30,82 %**, coincide estrechamente con el valor teórico de 30,4 %, confirmando la coherencia entre simulación y predicción analítica.

La comparación de resultados permite establecer una relación directa entre la amplitud de los armónicos introducidos y el valor del THD medido. A medida que se incrementa el número y magnitud de los armónicos, la forma de onda se aleja de la idealidad senoidal y la distorsión total aumenta de forma cuadrática.

En cuanto al **cumplimiento normativo**, solo la carga lineal cumple con el límite del 8 %, mientras que las cargas con distorsión moderada y alta **lo superan significativamente**, con un exceso del 3,18 % y 22,82 %, respectivamente. Estas dos últimas representarían cargas no lineales que degradan la calidad de potencia, pudiendo provocar pérdidas adicionales, sobrecalentamiento en transformadores o interferencias en sistemas de comunicación.

En conjunto, los resultados obtenidos demuestran la efectividad del procedimiento de análisis implementado en MATLAB y la utilidad de la función `analisis_armonicos.m` como herramienta para evaluar la distorsión armónica en sistemas eléctricos.

1 Ejercicio 4.3: Sobretensión por Conmutación de Condensadores

1.1 Objetivo

El propósito de este ejercicio es analizar el fenómeno de **sobretensión transitoria** producido por la conexión de un banco de condensadores en una red de baja tensión. Se estudiará la señal temporal, el cálculo del valor eficaz (RMS) mediante ventanas deslizantes de diferente duración y se discutirán los efectos sobre equipos electrónicos sensibles, así como posibles medidas correctivas.

1.2 Especificaciones de la señal

- Duración total: 400 ms
- Frecuencia fundamental: 50 Hz
- Tensión nominal: 325 V (valor pico)

- Sobretensión del 25 % durante 80 ms
- Inicio del evento: 150 ms

1.3 Fundamento teórico

Cuando un banco de condensadores se conecta a una red eléctrica, se produce un **transitorio de conmutación** debido a la interacción entre la inductancia del sistema y la capacitancia del banco. Esto genera una sobretensión temporal que puede alcanzar valores significativamente superiores al nominal. En el caso ideal analizado, el evento se modela como un aumento del 25 % de la amplitud de la señal durante un intervalo determinado.

El cálculo del valor eficaz o RMS de una señal $v(t)$ se define como:

$$V_{\text{RMS}} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} v^2(t) dt} \quad (13)$$

Para el análisis temporal, se utiliza una **ventana deslizando** de tamaño fijo T_w , desplazada muestra a muestra, que permite observar la evolución instantánea del valor RMS y detectar variaciones bruscas debidas a eventos transitorios.

1.4 Metodología

1. Se genera una señal senoidal de 50 Hz y amplitud pico 325 V.
2. Entre 150 ms y 230 ms se introduce un incremento del 25 % en la amplitud.
3. Se calcula el valor RMS deslizando con ventanas de 10 ms y 20 ms.
4. Se comparan los resultados obtenidos y se analiza qué ventana permite detectar mejor el evento.
5. Se discuten los posibles efectos en equipos electrónicos y las medidas correctivas recomendadas.

1.5 Código MATLAB

Listing 9: Código MATLAB del Ejercicio 4.3: Sobretensión por Conmutación de Condensadores

```
% Pr ctica 2. Itinerario de El ctrica
% M ster en Ingenier a Industrial. Universidad de Almer a
% Ejercicio 4.3: Sobretensi n por Conmutaci n de Condensadores
% Grupo 3: Nadia Rotbi Prado, Encarnaci n Cervantes Requena y
% Pablo Segura
% Fern ndez
clear; close all; clc;

%% 1. Generar la se al de sobretensi n
% Par metros b sicos
fs = 10e3; % Frecuencia de muestreo (10 kHz)
t_total = 0.4; % Duraci n total [s]
t = 0:1/fs:t_total; % Vector de tiempo
```

```

f = 50; % Frecuencia [Hz]
Vp = 325; % Amplitud nominal [V pico]

% Sobretensi n
inicio_evento = 0.150; % [s]
duracion_evento = 0.080; % [s]
fin_evento = inicio_evento + duracion_evento;

% Se al base
v = Vp * sin(2*pi*f*t);

% Aumento del 25% durante el evento
indice_evento = (t >= inicio_evento) & (t <= fin_evento);
v(indice_evento) = 1.25 * Vp * sin(2*pi*f*t(indice_evento));

figure;
plot(t*1000, v);
xlabel('Tiempo [ms]');
ylabel('Tensi n [V]');
title('Se al con sobretensi n de 25% durante 80 ms');
grid on;

%% 2. Calcular RMS deslizando con ventanas de 10 ms y 20 ms
% Ventanas de 10 ms y 20 ms
ventana1 = round(0.010 * fs);
ventana2 = round(0.020 * fs);

% RMS deslizando
v_rms_10 = sqrt(movmean(v.^2, ventana1));
v_rms_20 = sqrt(movmean(v.^2, ventana2));

figure;
plot(t*1000, v_rms_10, 'b', 'DisplayName', 'Ventana 10 ms');
hold on;
plot(t*1000, v_rms_20, 'r', 'DisplayName', 'Ventana 20 ms');
xlabel('Tiempo [ms]');
ylabel('Tensi n RMS [V]');
title('C lculo de RMS deslizando');
legend;
grid on;

figure;
plot(t*1000, v, 'k', 'DisplayName', 'Se al original');
hold on;
plot(t*1000, v_rms_10, 'b', 'DisplayName', 'RMS 10 ms');
plot(t*1000, v_rms_20, 'r', 'DisplayName', 'RMS 20 ms');
xlabel('Tiempo [ms]');
ylabel('Tensi n [V]');
title('Comparaci n de RMS con diferentes ventanas');
legend;
grid on;

```

1.6 Resultados

En la Figura 9 se observa la señal temporal con la sobretensión del 25 %. En la Figura 10 se muestran los valores RMS calculados con ventanas de 10 ms y 20 ms. La última figura es la combinación de las dos anteriores.

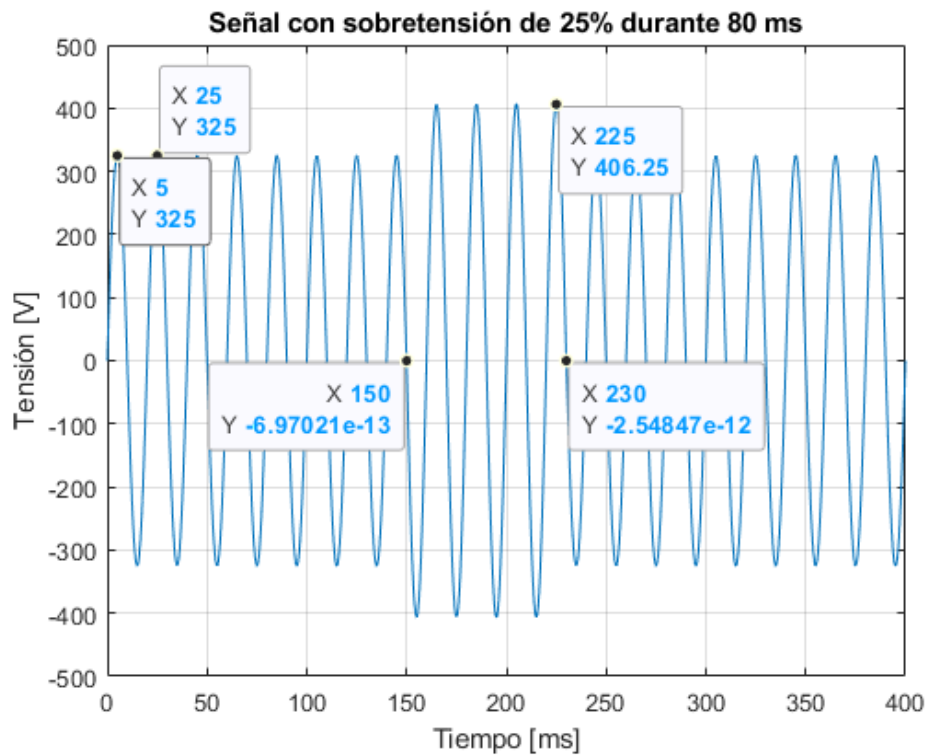


Figura 9: Señal con sobretensión.

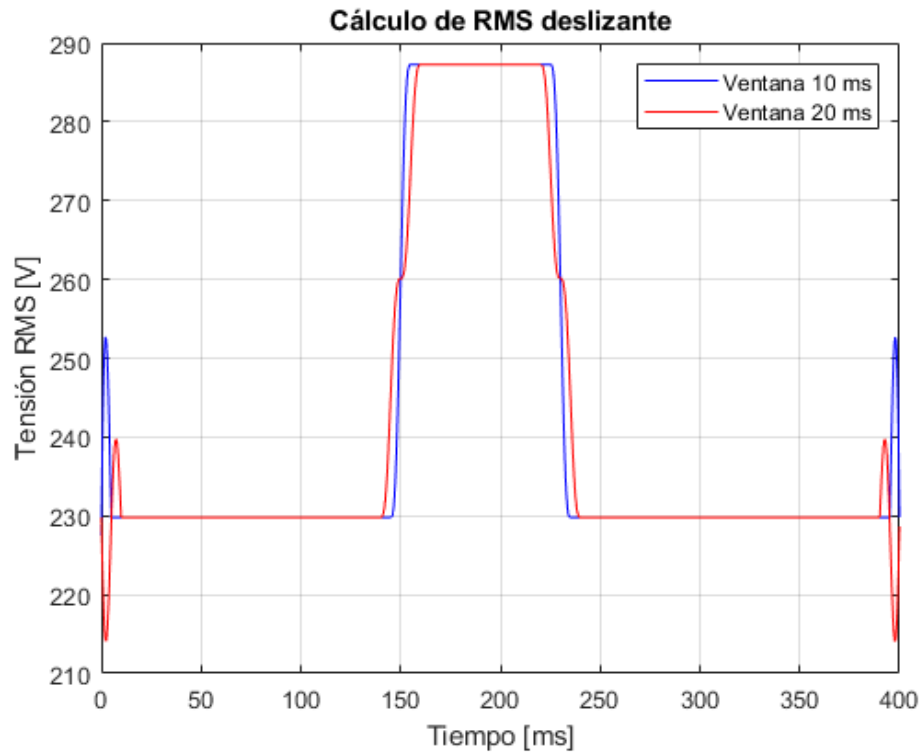


Figura 10: Valores RMS deslizantes.

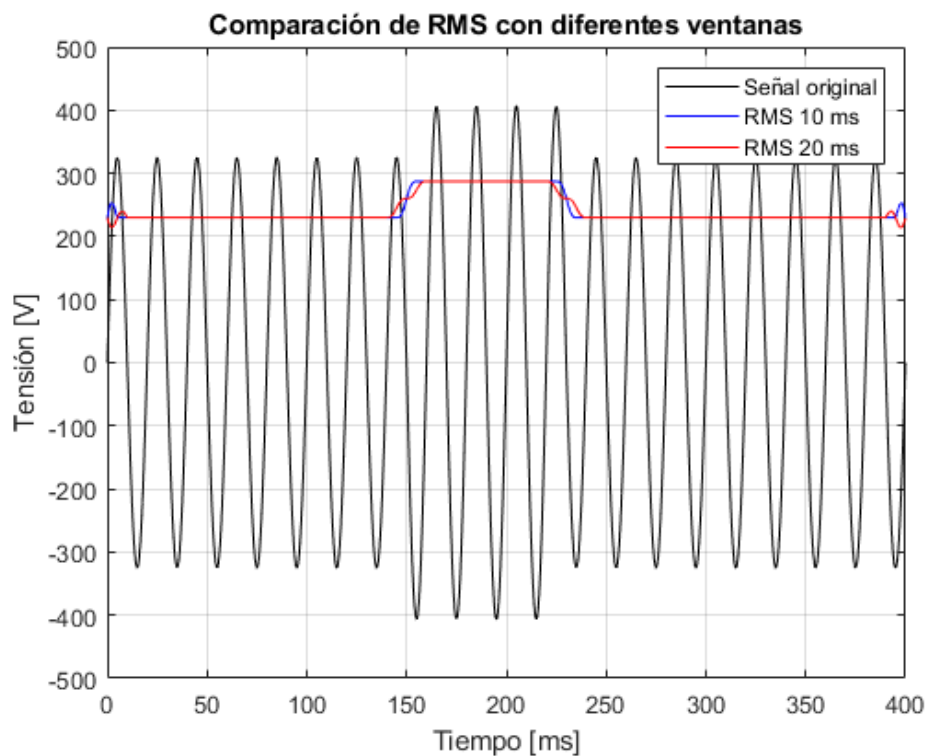


Figura 11: Señal con sobretensión y valores RMS deslizantes.

La ventana de **10 ms** presenta una respuesta más rápida y permite identificar claramente el inicio y fin del evento. En cambio, la ventana de **20 ms** suaviza la respuesta, ofreciendo

mayor estabilidad pero menor sensibilidad ante transitorios breves.

Por tanto, una ventana corta es preferible para la **detección temprana de sobretensiones**, mientras que una más larga es útil para el **análisis promedio** o de tendencia.

1.7 Impacto en equipos electrónicos

La sobretensión del 25 % eleva la tensión pico de 325 V a aproximadamente 406 V, lo que puede ocasionar:

- **Estrés dieléctrico** en condensadores y componentes de aislamiento.
- **Corrientes de irrupción elevadas** en fuentes de alimentación conmutadas.
- **Activación errónea** de protecciones electrónicas.
- **Daños acumulativos** por repetición del evento.

Estos efectos son especialmente críticos en dispositivos sensibles como sistemas informáticos, controladores industriales o equipos médicos.

1.8 Medidas correctivas recomendadas

Para mitigar las sobretensiones transitorias producidas por la conmutación de condensadores se proponen las siguientes estrategias:

- Empleo de **reactores de preinserción** o resistencias limitadoras.
- Uso de **contactores sincronizados** con el paso por cero de la tensión.
- Instalación de **supresores de transitorios** (varistores o TVS).
- Implementación de un **monitoreo RMS en tiempo real** con ventana corta.
- Incorporación de **filtros LC** o **snubbers** para limitar picos de conmutación.

1.9 Conclusiones

El análisis realizado muestra que la conexión abrupta de un banco de condensadores puede generar sobretensiones transitorias significativas, incluso en condiciones de operación normales. El uso de un RMS deslizante con ventana de 10 ms permite detectar eficazmente el evento, aportando una herramienta útil para la **supervisión dinámica de la calidad de potencia**. La implementación de medidas preventivas adecuadas es esencial para garantizar la fiabilidad y longevidad de los equipos electrónicos conectados a la red.