



# Análisis de Señales Utilizando Python

**DORA MARÍA BALLESTEROS**

UNIVERSIDAD MILITAR NUEVA GRANADA,  
FACULTAD DE INGENIERÍA.  
INGENIERÍA EN TELECOMUNICACIONES



editorial  
**redipe**

**Título original:**

Libro de investigación.

**ANÁLISIS DE SEÑALES UTILIZANDO PYTHON**

**Autor:** DORA MARÍA BALLESTEROS

**ISBN:** 978-1-957395-65-4

**Primera edición:** MAYO DE 2026

**SELLO Editorial**

Editorial REDIPE (95857440), Nueva York – Cali

Red de Pedagogía S.A.S. NIT: 900460139-2

© de la ilustración de la cubierta

**Comité Editorial:**

**Valdir Heitor Barzotto**, Universidad de Sao Paulo, Brasil

**Carlos Arboleda A.** PhD Investigador Southern Connecticut State University, Estados Unidos

**Agustín de La Herrán Gascón**, Ph D. Universidad Autónoma de Madrid, España

**Mario Germán Gil Claros**, Grupo de Investigación Redipe

**Rodrigo Ruay Garcés**, Chile. Coordinador Macroproyecto Investigativo Iberoamericano  
Evaluación Educativa

**Julio César Arboleda**, Ph D. Dirección General Redipe. Grupo de investigación Educación y  
Desarrollo humano, Universidad de San Buenaventura

Queda prohibida, salvo excepción prevista en la ley, la reproducción (electrónica, química, mecánica, óptica, de grabación o de fotocopia), distribución, comunicación pública y transformación de cualquier parte de ésta publicación -incluido el diseño de la cubierta- sin la previa autorización escrita de los titulares de la propiedad intelectual y de la Editorial. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Los Editores no se pronuncian, ni expresan ni implícitamente, respecto a la exactitud de la información contenida en este libro, razón por la cual no puede asumir ningún tipo de responsabilidad en caso de error u omisión.

Red Iberoamericana de Pedagogía

[editorial@rediberoamericanadepedagogia.com](mailto:editorial@rediberoamericanadepedagogia.com)

[www.redipe.org](http://www.redipe.org)

# Análisis de Señales utilizando Python

Universidad Militar Nueva Granada, Facultad de Ingeniería.  
Ingeniería en Telecomunicaciones.

**Dora Maria Ballesteros**  
dora.ballesteros@unimilitar.edu.co





AUTOR

# Dora María Ballesteros

Profesora e investigadora de la Universidad Militar Nueva Granada (UMNG) desde 2007. Es Ingeniera Electrónica, con Maestría y Doctorado en Ingeniería Electrónica, y cuenta con más de veinte años de experiencia en el procesamiento de señales, área en la que ha desarrollado una amplia labor académica e investigativa. En la última década, su trabajo se ha extendido hacia la inteligencia artificial y la ciencia de datos, integrando estos enfoques con el análisis clásico de señales.

A lo largo de su carrera ha publicado cinco libros y cerca de 90 artículos científicos en revistas nacionales e internacionales. Su investigación se caracteriza por la integración de fundamentos teóricos, experimentación reproducible y aplicaciones reales, con especial interés en el análisis de señales mediante técnicas de inteligencia artificial y en el desarrollo de herramientas orientadas a enfrentar problemáticas actuales como los *deepfakes* y el uso no ético de contenido sintético.

Este libro surge de su experiencia en docencia e investigación en la UMNG y de la convicción de que el análisis de señales se comprende mejor cuando se vincula con la práctica. Su propósito es ofrecer una visión clara y estructurada del análisis de señales, utilizando Python como un entorno sencillo de visualización y experimentación, donde los conceptos matemáticos se transforman en señales, gráficos y simulaciones que fortalecen la intuición y su aplicación en problemas reales de ingeniería.





## ANTES DE COMENZAR

Durante más de veinte años he trabajado con señales. Las he estudiado, las he enseñado, las he programado, las he visto transformarse una y otra vez entre el dominio del tiempo y la frecuencia. Han sido parte de mi vida académica, pero también de mi forma de pensar como ingeniera.

Y, sin embargo, durante mucho tiempo sentí que había algo pendiente.

Este libro nace de esa sensación. De saber que, a pesar de haber recorrido este camino durante años, me debía a mí misma sentarme a organizar, a escribir, a explicar con calma aquello que tantas veces compartí en clase, pero que nunca había dejado plasmado de manera completa. Y también se lo debía a mis estudiantes.

A cada grupo que me hizo preguntas, a quienes no entendieron a la primera (y eso estaba bien), a quienes necesitaban ver una gráfica más, un ejemplo adicional, una explicación distinta. A quienes confiaron en que, detrás de cada ecuación, había una idea que sí podía ser comprendida.

Este no es un libro escrito desde la perfección, sino desde la experiencia. Desde los aciertos, pero también desde los errores que me enseñaron cómo explicar mejor. Desde la convicción de que las señales no son solo expresiones matemáticas, sino formas de representar fenómenos reales que vale la pena entender.

Si este libro logra que alguien vea una señal con más claridad, que entienda mejor lo que antes parecía abstracto, o que pierda un poco el miedo a este tema, entonces habrá cumplido su propósito.

Y si además acompaña a mis estudiantes, tanto a los actuales como a los futuros, en su proceso de aprendizaje, entonces habrá valido completamente la pena.

*Dora Ma. Ballesteros*





## CONTENIDO

	PAG
CAPITULO 1.	
<b>Fundamentos de Señales Análogas y Señales continuas .....</b>	<b>11</b>
CAPITULO 2.	
<b>Sistemas lti (lineales, invariantes en el tiempo).....</b>	<b>55</b>
CAPITULO 3.	
<b>Señal exponencial compleja y series de fourier.....</b>	<b>81</b>
CAPITULO 4.	
<b>Transformada de fourier de Señales Periódicas y Aperiódicas...</b>	<b>123</b>





## CAPÍTULO I

# Fundamentos de Señales Análogas y Señales Continuas

En este capítulo encontrarás una breve introducción a los conceptos de señales continuas, discretas y digitales. Adicionalmente, se explicarán las transformaciones que podemos hacerles a las señales en el dominio del tiempo, y las propiedades de los sistemas.

Al finalizar el capítulo, deberás estar en capacidad de:

1. Diferenciar señales análogas, continuas, discretas y digitales.
2. Escribir código en Python para dibujar señales continuas y discretas.
3. Aplicar transformaciones (desplazamiento, inversión, escalamiento) a señales continuas y discretas.
4. Escribir código en Python para realizar desplazamientos, inversión, y/o escalamiento a señales continuas y discretas.
5. Identificar si un sistema cumple con las propiedades de: memoria, causalidad, estabilidad, invarianza, linealidad e invertibilidad.

En este capítulo introductorio del curso, vamos a conocer las generalidades de las señales continuas y discretas, y las transformaciones de la variable independiente. Adicionalmente, identificaremos en qué casos un sistema continuo o discreto cumple con las siguientes propiedades: linealidad, invarianza, estabilidad, memoria, causalidad e invertibilidad.

## 1.1. SEÑALES CONTINUAS Y SEÑALES DISCRETAS Y SU VISUALIZACIÓN EN PYTHON

### 1.1.1. Señales continuas vs. señales discretas

Una señal de tiempo continuo es aquella cuya variable independiente puede tomar cualquier valor real dentro de un intervalo determinado. Matemáticamente, este tipo de señales se representan como:

$$x(t), t \in R$$

donde  $t$  representa el tiempo continuo. En la naturaleza, la mayoría de las señales físicas son continuas, por ejemplo: la voz humana, una señal eléctrica analógica o la temperatura ambiente medida en función del tiempo. Desde un punto de vista teórico, una señal continua posee un número infinito de valores en el tiempo. Sin embargo, esta característica impide su almacenamiento y procesamiento directo en un computador.

Por otro lado, una señal de tiempo discreto es aquella definida únicamente para instantes de tiempo separados, usualmente equiespaciados. Se representa matemáticamente como:

$$x[n], n \in Z$$

donde  $n$  es un índice entero que identifica cada muestra de la señal. Este tipo de señales surge naturalmente cuando una señal continua es muestreada para su almacenamiento o procesamiento digital. A diferencia de las señales continuas, las señales discretas contienen un número finito o infinito de muestras, pero definidas en instantes discretos, lo que las hace adecuadas para ser manipuladas mediante algoritmos computacionales.

En este libro, utilizaremos la siguiente notación estándar:

Tipo de señal	Notación	Descripción
Tiempo continuo	$x(t)$	La variable independiente es el tiempo continuo, representado por la letra $t$ . Se utilizan paréntesis redondos.
Tiempo discreto	$x[n]$	La variable independiente es el tiempo discreto, representado por la letra $n$ . Se utilizan paréntesis cuadrados.

### 1.1.2. Visualización de señales

Aunque el análisis matemático distingue claramente entre señales continuas y discretas, en la práctica todas las señales representadas en un computador son discretas, ya que solo se pueden almacenar y procesar un número finito de muestras.

#### **Nota importante**

Cuando se utiliza Python para graficar señales, la diferencia entre señal continua y discreta no depende únicamente de la gráfica, sino del modelo matemático asumido. La instrucción `plt.plot` interpola visualmente entre muestras, generando una apariencia continua, mientras que `plt.stem` representa explícitamente cada muestra discreta.

#### **Ejemplo 1: señal senoidal**

Consideremos una señal senoidal de frecuencia  $f = 10 \text{ Hz}$

y duración de  $0.1 \text{ s}$ . A continuación, se muestra su generación y visualización en Python.

```

import numpy as np
import matplotlib.pyplot as plt

f = 10 # frecuencia [Hz]
fs = 1000 # frecuencia de muestreo [Hz]
t = np.linspace(0, 0.1, int(0.1*fs))

x = np.sin(2*np.pi*f*t)

plt.plot(t, x)
plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.title("Señal senoidal visualizada como continua")
plt.show()

```

Figura 1. Código en Python del Ejemplo 1 para crear y visualizar una señal senoidal.

Y obtenemos la siguiente figura:

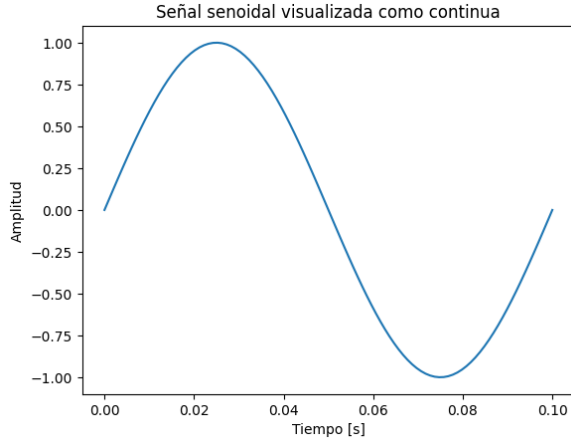


Figura 2. Gráfica señal senoidal del Ejemplo 1, visualizada como continua.

Aunque la señal se muestra como continua, en realidad está compuesta por un conjunto finito de muestras. Si se desea enfatizar su naturaleza discreta, se utiliza la función stem:

```
plt.stem(t, x)
plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.title("Señal senoidal muestreada")
plt.show()
```

Figura 3. Código en Python del Ejemplo 1 para visualizar una señal como discreta.

Obteniendo la siguiente gráfica:

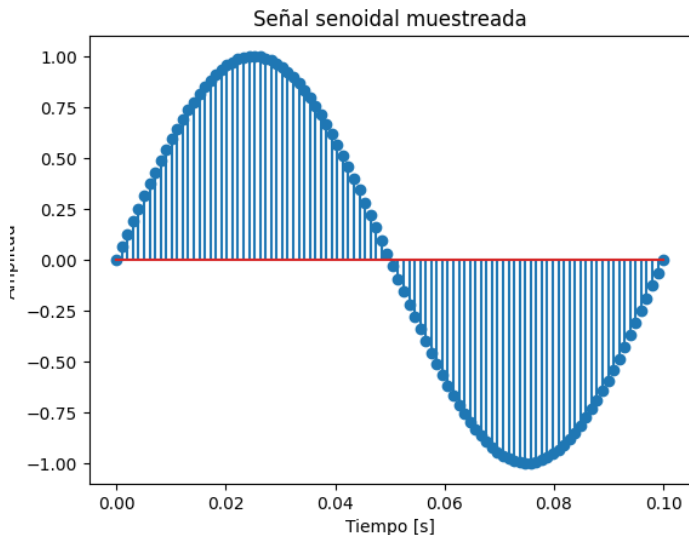


Figura 4. Gráfica señal senoidal del Ejemplo 1, visualizada como discreta.

### Ejemplo 2: señales discretas con índice entero

Para representar una señal estrictamente discreta, el eje horizontal debe expresarse mediante un índice entero  $n$ . Por ejemplo, una señal senoidal discreta con  $m$  muestras y  $n_c$  ciclos se puede escribir como:

```

m = 10 # número de muestras
nc = 1 # número de ciclos
n = np.arange(m)

x = np.sin(2*np.pi*nc*n/m)

plt.stem(n, x)
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal senoidal discreta x[n]")
plt.show()

```

Figura 5. Código en Python del Ejemplo 2 para visualizar señal senoidal discreta con índice entero.

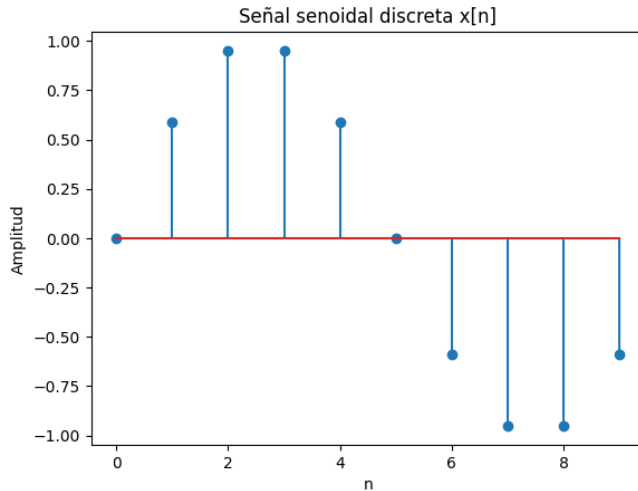


Figura 6. Gráfica señal senoidal del Ejemplo 2, visualizada como discreta con índice entero.

En este caso, la señal está definida únicamente para los valores enteros de  $n$ , lo que refleja correctamente el concepto de señal de tiempo discreto.

Al comprender esta distinción entre señales continuas y discretas, se establecen las bases necesarias para el estudio de las transformaciones en el dominio del tiempo y el análisis de sistemas, temas que abordaremos en las siguientes secciones.

## 1.2. TRANSFORMACIÓN DE LA VARIABLE INDEPENDIENTE

En el análisis de señales, es común modificar la variable independiente (tiempo continuo o índice discreto) con el fin de analizar cómo cambia la forma de una señal. Estas modificaciones reciben el nombre de transformaciones en el dominio del tiempo.

En esta sección estudiaremos tres transformaciones:

- Desplazamiento temporal
- Inversión temporal
- Escalamiento temporal

Cada transformación se analizará tanto para señales de tiempo continuo como de tiempo discreto, resaltando sus similitudes y diferencias.

### 1.2.1. Desplazamiento de señales continuas y discretas

El desplazamiento temporal consiste en adelantar o atrasar una señal respecto al eje del tiempo. Matemáticamente, esta transformación se expresa como:

	Señal original	Señal desplazada
Tiempo continuo	$x(t)$	$x(t - t_0)$
Tiempo discreto	$x[n]$	$x[n - k]$

En señales en tiempo continuo, el parámetro  $t_0$  es un número real (positivo o negativo). En señales discretas, el desplazamiento  $k$  es un número entero.

- Si el valor de desplazamiento es positivo, la señal se desplaza hacia la derecha.
- Si el valor de desplazamiento es negativo, la señal se desplaza hacia la izquierda.

#### Ejemplo 3: desplazamiento en tiempo continuo

Consideremos una señal con  $f = 1 \text{ Hz}$ , definida en el intervalo  $t \in [0, 1] \text{ s}$ . Para su representación gráfica se utiliza una frecuencia de

muestreo  $f_s = 50 \text{ Hz}$ . En este caso, al utilizar  $t_0 = 0.5 \text{ s}$ , la señal se desplaza hacia la derecha, quedando en el intervalo  $t \in [0.5, 1.5] \text{ s}$ .

Si se utilizara un valor negativo de  $t_0$ , el desplazamiento ocurriría hacia la izquierda.

```
import numpy as np
import matplotlib.pyplot as plt

f = 1      # frecuencia [Hz]
fs = 50    # frecuencia de muestreo [Hz]
ti = 0
tf = 1
t0 = 0.5   # desplazamiento temporal [s] (retardo)

# Tiempo de la señal original (duración finita)
t = np.linspace(ti, tf, int((tf - ti)*fs), endpoint=False)
x = np.sin(2*np.pi*f*t)

# Tiempo de la señal desplazada (misma forma, pero empieza después)
t_shift = t + t0
x_shift = np.sin(2*np.pi*f*(t_shift - t0)) # equivalente a sin(2π f t)

plt.plot(t, x, label="x(t), soporte [0,1]")
plt.plot(t_shift, x_shift, label="x(t - t0), soporte [0.5,1.5]")

plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.title("Señal de duración finita y su desplazamiento temporal")
plt.legend()
plt.grid(True)
plt.show()
```

Figura 7. Gráfica señal senoidal del Ejemplo 3.

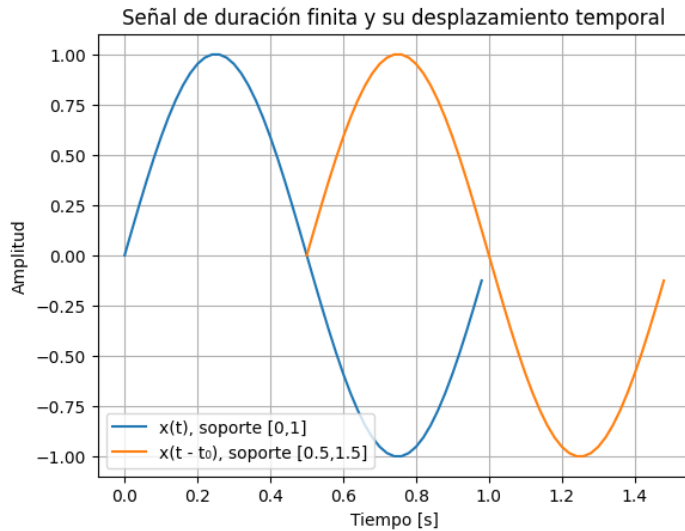


Figura 8. Gráficas del Ejemplo 3: señal original (azul) y señal desplazada (naranja).

#### Ejemplo 4: desplazamiento en tiempo discreto

Para una señal discreta con  $m = 40$  muestras y  $n_c = 2$  ciclos, el desplazamiento se realiza modificando el índice  $n$ , a partir del valor de  $k$ . Este desplazamiento no altera las amplitudes de la señal, únicamente su ubicación temporal.

```

import numpy as np
import matplotlib.pyplot as plt

# Parámetros de la señal
m = 40
nc = 2
k = 5

# Eje temporal discreto
n = np.arange(m)

# Señal original
x = np.sin(2 * np.pi * nc * n / m)

# Señal desplazada: x[n-k]
n_shift = n + k
x_shift = x

# Gráficas (cada plt.stem en una sola línea)
plt.stem(n, x, linefmt='C0-', markerfmt='C0o', basefmt=" ", label='x[n], soporte [0, 39]')
plt.stem(n_shift, x_shift, linefmt='C1-', markerfmt='C1o', basefmt=" ", label='x[n-k], soporte [5, 44]')

# Formato
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Desplazamiento temporal de una señal discreta (soporte finito)")
plt.legend()
plt.grid(True)
plt.show()

```

Figura 9. Código en Python del Ejemplo 4.

Obteniendo la siguiente gráfica, en la cual la señal original definida entre  $n = 0$  y  $n = 39$  se desplaza 5 posiciones hacia la derecha, quedando definida entre  $n = 5$  y  $n = 44$ .

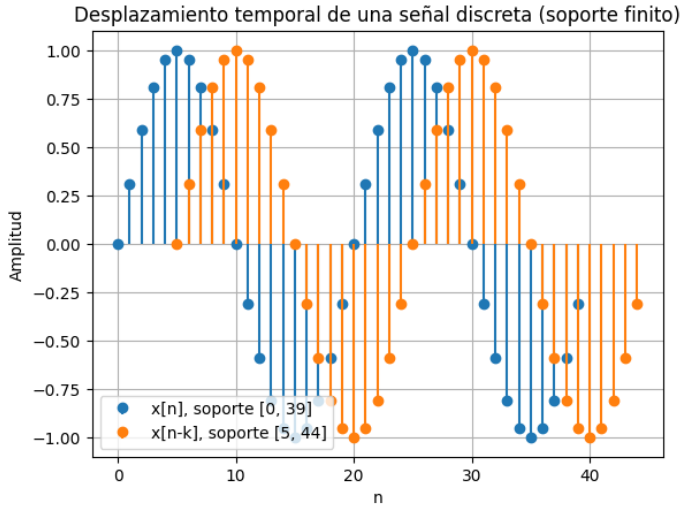


Figura 10. Gráficas del Ejemplo 4: señal original (azul) y señal desplazada (naranja).

### 1.2.2. Inversión temporal en señales de tiempo continuo y discreto

La inversión temporal produce un efecto de espejo de la señal con respecto al origen del eje temporal y se define matemáticamente como:

	Señal original	Señal invertida
Tiempo continuo	$x(t)$	$x(-t)$
Tiempo discreto	$x[n]$	$x[-n]$

Donde el origen temporal se fija en  $t = 0$ , ó  $n = 0$ . Esta transformación intercambia la parte de la señal ubicada a la derecha del origen temporal con la que se encuentra a la izquierda, manteniendo fija la muestra definida en el origen.

### Ejemplo 5. Inversión temporal de señales de soporte finito en tiempo continuo

En este ejemplo se ilustra el efecto de la inversión temporal sobre una señal de soporte finito, en tiempo continuo. Se compara la señal original con su versión invertida, resaltando el cambio en el soporte temporal y el efecto de espejo con respecto al origen del eje del tiempo.

Supongamos que tenemos una señal con  $f = 1 \text{ Hz}$ , definida en el intervalo  $t \in [0, 1] \text{ s}$ . Para su representación gráfica se utiliza una frecuencia de muestreo  $f_s = 200 \text{ Hz}$ . Al aplicar la inversión temporal, la señal resultante queda definida en el intervalo  $t \in [-1, 0] \text{ s}$ , como un reflejo de la señal original respecto al origen temporal. Se observa que el valor de la señal en  $t = 0$  se mantiene invariante en ambas representaciones.

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros
f = 1 # frecuencia de la señal [Hz]
fs = 200 # frecuencia de muestreo para visualización [Hz]
ti = 0
tf = 1

# Señal original (soporte finito en [ti, tf])
t = np.linspace(ti, tf, int((tf - ti) * fs), endpoint=False)
x = np.sin(2 * np.pi * f * t)

# Señal invertida temporalmente
# La inversión x(-t) se implementa invirtiendo el eje temporal
# y reordenando las muestras para mantener un eje creciente
t_inv = -t[::-1] # eje temporal invertido
x_inv = x[::-1] # muestras invertidas

# Gráficas
plt.plot(t, x, label="x(t), soporte [0, 1]")
plt.plot(t_inv, x_inv, label="x(-t), soporte [-1, 0]")

plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.title("Inversión temporal en tiempo continuo: x(t) vs x(-t)")
plt.legend()
plt.grid(True)
plt.show()
```

Figura 11. Código en Python del Ejemplo 5: inversión temporal en tiempo continuo.

A partir de la señal original  $x(t)$ , se implementa la inversión temporal construyendo la señal  $x(-t)$ . Esta operación se realiza invirtiendo el eje temporal y reordenando las muestras del arreglo. Adicionalmente, las amplitudes también se recorren en orden inverso del arreglo.

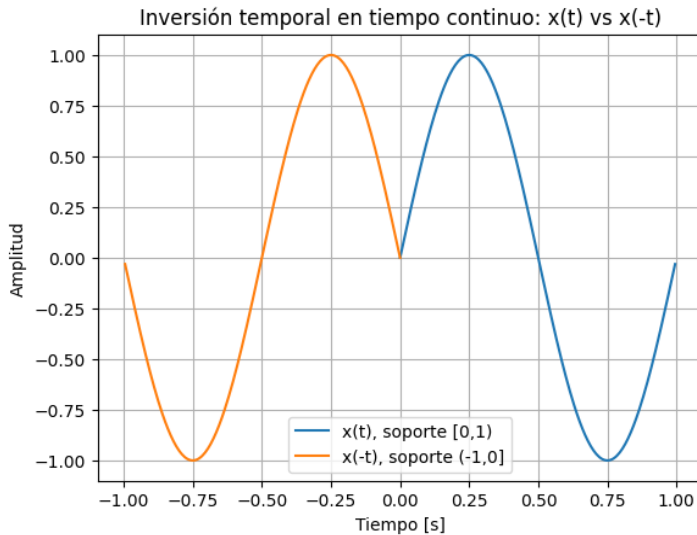


Figura 12. Representación gráfica del Ejemplo 5: señal original  $x(t)$  (azul) y señal invertida  $x(-t)$  (naranja).

### Ejemplo 6. Inversión temporal de señales de soporte finito en tiempo continuo y discreto

Ahora, supongamos que tenemos una señal  $x[n]$  de soporte finito definida para  $n \in [0 \ 19]$ , es decir, una señal discreta con 20 muestras que inicia en el origen. Al aplicar la inversión temporal, se obtiene una señal  $x[-n]$ , cuyo soporte queda definido para  $n \in [-19 \ 0]$ , lo cual implica que el orden temporal de las muestras se invierte.

En este caso, las amplitudes de la señal se asignan en orden inverso con respecto a la señal original. De manera análoga, las posiciones temporales discretas se invierten previamente y se reflejan con respecto al

origen, garantizando la correcta correspondencia entre cada muestra y su nuevo instante temporal.

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros
m = 20      # número de muestras
nc = 1      # número de ciclos en la ventana de m muestras

# Señal original (tiempo discreto) con soporte finito en [0, m-1]
n = np.arange(m)
x = np.sin(2 * np.pi * nc * n / m)

# Inversión temporal: x[-n]
# Para graficar de izquierda a derecha, invertimos el eje y reordenamos las muestras
n_inv = -n[::-1]  # soporte [- (m-1), 0]
x_inv = x[::-1]  # mismas muestras, en orden inverso

# Gráfica
plt.stem(n, x, linefmt='C0-', markerfmt='C0o', basefmt=" ",
        label=f"x[n], soporte [0, {m-1}]")
plt.stem(n_inv, x_inv, linefmt='C1-', markerfmt='C1o', basefmt=" ",
        label=f"x[-n], soporte [-(m-1), 0]")

plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Inversión temporal en tiempo discreto: x[n] vs x[-n]")
plt.legend()
plt.grid(True)
plt.show()
```

Figura 13. Código en Python del Ejemplo 6: inversión temporal en tiempo discreto.

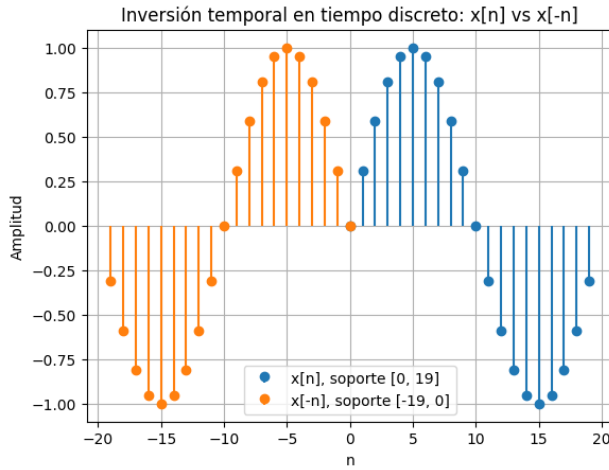


Figura 14. Representación gráfica del Ejemplo 6: señal original  $x[n]$  (azul) y señal invertida  $x[-n]$  (naranja).

### 1.2.3. Escalamiento en el dominio del tiempo continuo

La tercera transformación aplicada a la variable independiente corresponde al escalamiento temporal. A diferencia de las transformaciones de desplazamiento e inversión, el escalamiento presenta diferencias fundamentales cuando se aplica a señales de tiempo continuo y de tiempo discreto.

En el caso del tiempo continuo, el escalamiento modifica la duración temporal de la señal, produciendo un efecto de compresión o dilatación del eje temporal, según el valor del factor de escalamiento  $\alpha$ :

Señal original	Compresión de la señal (menor duración)	Dilatación de la señal (mayor duración)
$x(t)$	$x(\alpha t)$ para $\alpha > 1$	$x(\alpha t)$ para $0 < \alpha < 1$

#### Ejemplo 7. Compresión de una señal en tiempo continuo

Supongamos que tenemos una señal de frecuencia  $f = 1 \text{ Hz}$ , definida en el intervalo temporal  $t \in [1, 2] \text{ s}$ , es decir, con una duración de un segundo.

Al aplicar un escalamiento temporal con  $\alpha > 1$ , la señal se comprime, reduciendo su duración.

El siguiente código ilustra el efecto de compresión, para diferentes valores de  $\alpha$ .

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros
f = 1      # frecuencia de la señal [Hz]
fs = 50   # frecuencia de muestreo para visualización [Hz]
ti = 1    # tiempo inicial [s]
tf = 2    # tiempo final [s]

# Vector temporal original ( $\alpha = 1$ )
t = np.linspace(ti, tf - 1/fs, int((tf - ti) * fs))
x = np.sin(2 * np.pi * f * t)

# Compresión temporal
alpha2 = 2
alpha4 = 4

# Nuevos ejes temporales creados a partir de t
t2 = t / alpha2
t4 = t / alpha4

# Señales comprimidas
x2 = np.sin(2 * np.pi * f * (alpha2 * t2))
x4 = np.sin(2 * np.pi * f * (alpha4 * t4))

# Gráfica
plt.plot(t, x, label='x(t),  $\alpha = 1$ ', linewidth=2)
plt.plot(t2, x2, label='x( $\alpha t$ ),  $\alpha = 2$ ', linewidth=2)
plt.plot(t4, x4, label='x( $\alpha t$ ),  $\alpha = 4$ ', linewidth=2)

plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.title("Compresión temporal en tiempo continuo")
plt.legend()
plt.grid(True)
plt.show()
```

Figura 15. Código en Python del Ejemplo 7: compresión temporal en tiempo continuo, para  $\alpha > 1$ .

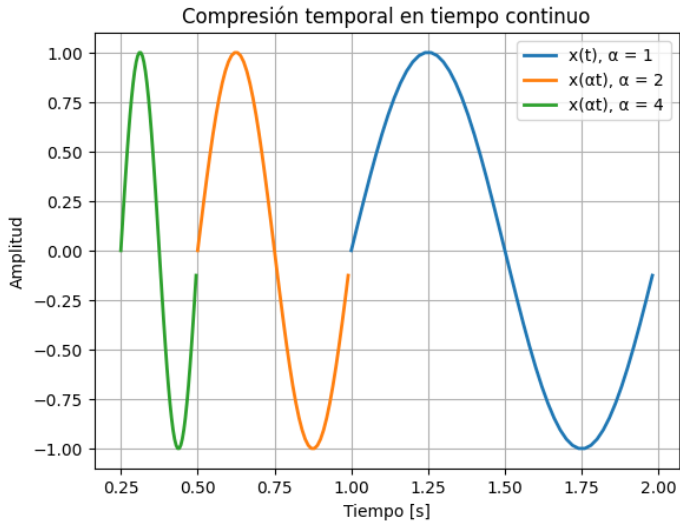


Figura 16. Representación gráfica del Ejemplo 7: señal original  $x(t)$  (azul), señal comprimida con  $\alpha=2$  (naranja) y  $\alpha=4$  (verde).

Al comprimir la señal se pueden apreciar dos efectos:

- La duración de la señal comprimida es igual a la duración de la señal inicial dividida en  $\alpha$ .
- El tiempo inicial de la señal comprimida es igual al tiempo inicial de la señal inicial, dividido en  $\alpha$ .
- El tiempo final de la señal comprimida es igual al tiempo final de la señal inicial, dividido en  $\alpha$ .

### Ejemplo 8. Dilatación de una señal en tiempo continuo

Vamos a utilizar los mismos datos del ejemplo 7, pero ahora, en este

caso  $0 < \alpha < 1$ .

```

import numpy as np
import matplotlib.pyplot as plt

# Parámetros
f = 1      # frecuencia de la señal [Hz]
fs = 50   # frecuencia de muestreo para visualización [Hz]
ti = 1    # tiempo inicial [s]
tf = 2    # tiempo final [s]

# Vector temporal original ( $\alpha = 1$ )
t = np.linspace(ti, tf - 1/fs, int((tf - ti) * fs))
x = np.sin(2 * np.pi * f * t)

# Dilatación temporal
alpha2 = 0.5
alpha4 = 0.25

# Nuevos ejes temporales creados a partir de t
t2 = t / alpha2
t4 = t / alpha4

# Señales dilatadas
x2 = np.sin(2 * np.pi * f * (alpha2 * t2))
x4 = np.sin(2 * np.pi * f * (alpha4 * t4))

# Gráfica
plt.plot(t, x, label='x(t),  $\alpha = 1$ ', linewidth=2)
plt.plot(t2, x2, label='x( $\alpha t$ ),  $\alpha = 0.5$ ', linewidth=2)
plt.plot(t4, x4, label='x( $\alpha t$ ),  $\alpha = 0.25$ ', linewidth=2)

plt.xlabel("Tiempo [s]")
plt.ylabel("Amplitud")
plt.title("Compresión temporal en tiempo continuo")
plt.legend()
plt.grid(True)
plt.show()

```

Figura 17. Código en Python del Ejemplo 8: dilatación en tiempo continuo, para  $0 < \alpha < 1$ .

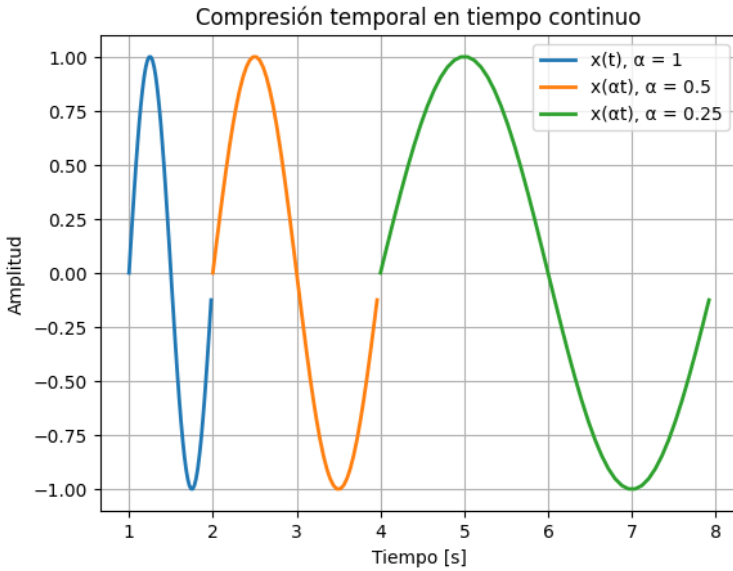


Figura 18. Representación gráfica del Ejemplo 8: señal original  $x(t)$  (azul), señal dilatada con  $\alpha=0.5$  (naranja) y  $\alpha=0.25$  (verde).

En este caso, a medida que  $\alpha$  se va haciendo más pequeño, la duración de la señal se va haciendo mayor. El vector de tiempos se divide entre  $\alpha$ .

#### 1.2.4. Escalamiento en el dominio del tiempo discreto

A diferencia del caso continuo, estas operaciones (compresión/dilatación) no son equivalentes exactos, sino aproximaciones prácticas para su implementación digital.

En particular, cuando se realiza una compresión temporal, se descartan muestras de la señal original que no pueden recuperarse posteriormente. Por el contrario, al efectuar una dilatación temporal, se introducen muestras adicionales con valor cero en posiciones específicas del eje temporal.

## Ejemplo 9: Compresión señal discreta

Supongamos que tenemos un ciclo de una señal senoidal con 40 muestras, empezando en la muestra  $n = 0$  y terminando en  $n = 39$ . Esta señal se comprime con un  $\alpha = 3$ , como se presenta con el siguiente código:

```
import numpy as np
import matplotlib.pyplot as plt

m = 40
nc = 1
alpha = 3 # prueba con 2, 3, 4...

n = np.arange(m) # entero
x = np.sin(2 * np.pi * nc * n / m)

x_new = x[::alpha]
n_new = np.arange(len(x_new)) # 0..len(x_new)-1

plt.stem(n, x)
plt.axis([n.min(), n.max()+1, x.min(), x.max()])
plt.show()

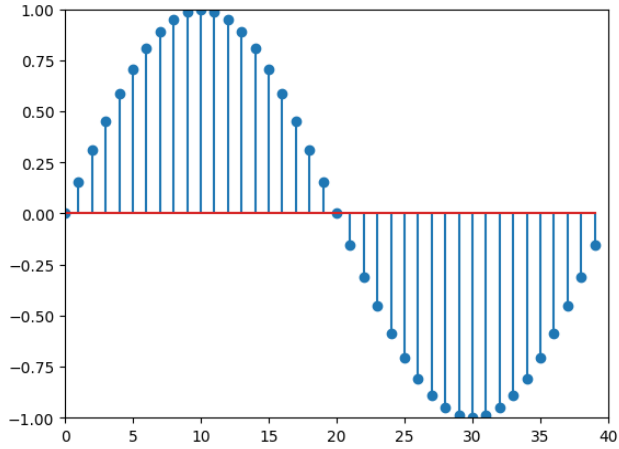
plt.stem(n_new, x_new)
plt.axis([n_new.min(), n_new.max()+1, x_new.min(), x_new.max()])
plt.xticks(n_new) # ticks enteros
plt.show()
```

Figura 19. Código en Python del Ejemplo 9: compresión en tiempo discreto, para  $\alpha=3$ .

Las amplitudes las tomamos cada  $\alpha$  posiciones, empezando desde la posición  $n=0$ . Así, para un vector de 40 posiciones que empieza en 0 y termina en 39, tendremos 14 muestras, empezando en 0 y terminando en 13 (que se obtiene de dividir 39 entre 3).

Obtenemos las siguientes gráficas:

A)



B)

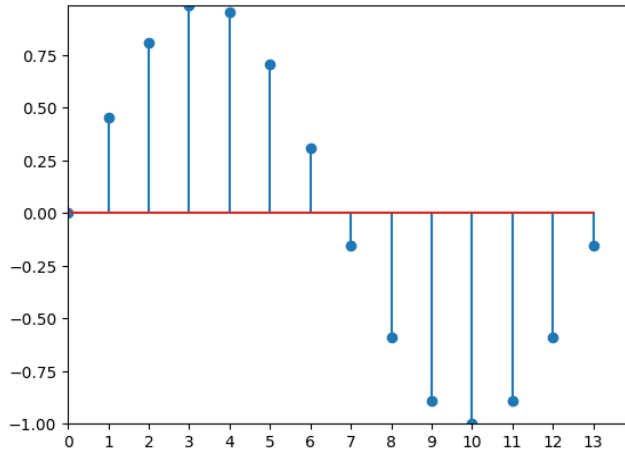


Figura 20. Representación gráfica del Ejemplo 9: a) señal  $x[n]$ , b) señal  $x[3n]$ .

**Ejemplo 10: Dilatación señal discreta, caso  $\alpha=1/q$ .**

Vamos ahora a realizar un ejemplo para el caso discreto, cuando hacemos una transformación de dilatación. En este caso  $\alpha = 1/q$ , donde  $q \in \mathbb{Z}$ .

```
import numpy as np
import matplotlib.pyplot as plt

m = 10      # número de muestras
nc = 1      # número de ciclos

q = 3      # q es entero
alpha = 1/ q

# Señal original (tiempo discreto entero)
n = np.arange(m)
x = np.sin(2 * np.pi * nc * n / m)

# Dilatación
L = int(round(1 / alpha))      # factor entero
n_new = np.arange(m * L)     # eje discreto entero
x_new = np.zeros(m * L)
x_new[::L] = x                # insertar ceros

# Señal original
plt.stem(n, x)
plt.axis([n.min(), n.max()+1, x.min(), x.max()])
plt.xticks(n)
plt.show()

# Señal dilatada
plt.stem(n_new, x_new)
plt.axis([n_new.min(), n_new.max()+1, x_new.min(), x_new.max()])
plt.xticks(n_new[::L])      # 🖱 ticks cada L
plt.show()
```

Figura 21. Código en Python del Ejemplo 10: dilatación en tiempo discreto, para  $\alpha=1/3$ .

En este código es importante utilizar el entero del redondeo de  $1/\alpha$ , para conocer las posiciones nuevas de la señal dilatada, y de esta manera se insertan ceros por medio de  $x\_new[:,L] = x$ .

Y se obtiene:

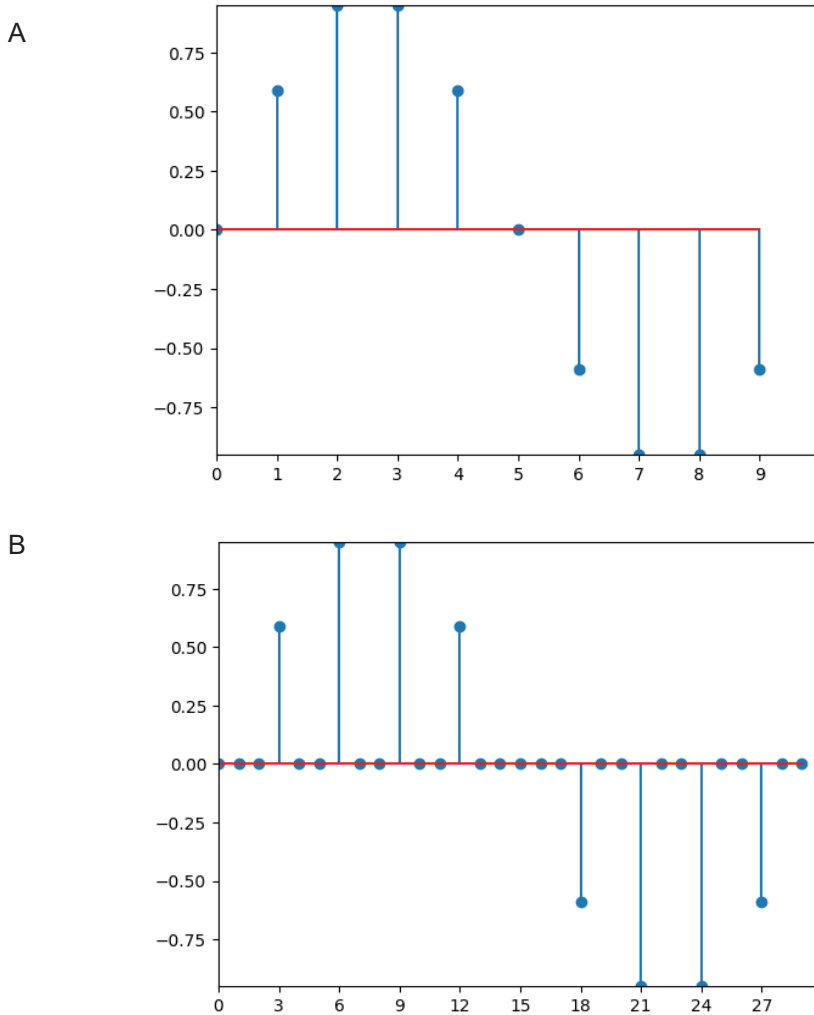


Figura 22. Representación gráfica del Ejemplo 10: a) señal  $x[n]$ , b) señal  $x[n/3]$ .

La señal  $x[n]$  está comprendida en el intervalo  $[0, 9]$  (último valor no nulo en  $n=9$ ); mientras que, la señal  $x[n/3]$  está comprendida en el intervalo  $[0, 29]$ , pero con valores no nulos hasta  $n=27$ .

### **Ejemplo 11: Dilatación señal discreta, caso $\alpha=p/q$ .**

Vamos ahora a analizar un caso “especial” de dilatación, que ocurre cuando el valor de  $p$  no es igual a 1. En esta situación, no solo se insertan ceros dentro de la señal dilatada, sino que, adicionalmente, algunas muestras de la señal original se pierden. Por lo tanto, este es un claro ejemplo de que el escalamiento temporal en tiempo discreto no es una operación invertible en general: la compresión nunca es reversible y la dilatación solo lo es cuando  $\alpha = 1/q$ , para  $q$  entero.

Trabajaremos con el siguiente código:

```
import numpy as np
import matplotlib.pyplot as plt

# Señal original
n = np.arange(0, 11)      # 0..10
x = np.arange(len(n))    # ejemplo: x[n] = n

# Factor de expansión
alpha = 2/3
ratio = 1/alpha          # 3/2

# Mapeo de índices
n_targets = ratio * n
n_int_mask = np.isclose(n_targets, np.round(n_targets))
n_targets_int = np.round(n_targets[n_int_mask]).astype(int)

# Nueva señal expandida
N_new = n_targets_int.max() + 1
x_new = np.zeros(N_new)
x_new[n_targets_int] = x[n_int_mask]

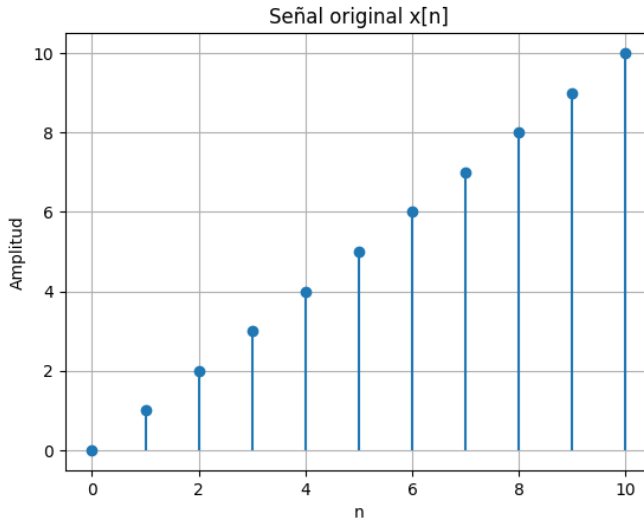
# Gráfica señal original
plt.figure()
plt.stem(n, x, basefmt=" ")
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal original x[n]")
plt.grid(True)
plt.show()

# Gráfica señal expandida
plt.figure()
plt.stem(np.arange(N_new), x_new, basefmt=" ")
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal expandida x[n/α], α = 2/3")
plt.grid(True)
plt.show()
```

Figura 23. Código en Python del Ejemplo 11: dilatación en tiempo discreto, para  $\alpha=2/3$ .

Y obtenemos la señal de salida:

A



B

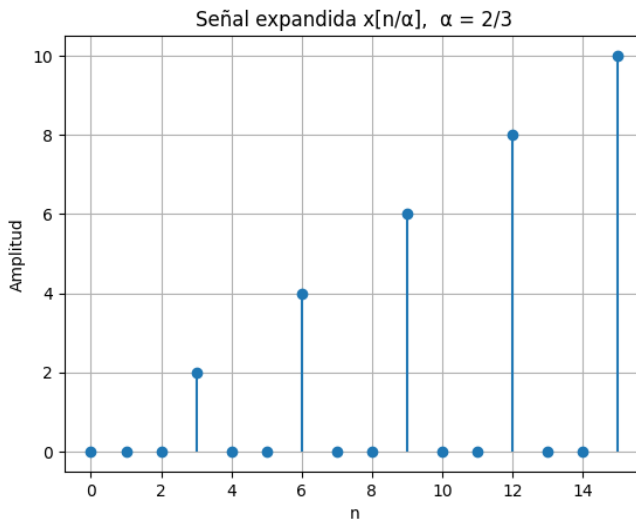


Figura 24. Representación gráfica del Ejemplo 11: a) señal  $x[n]$ , b) señal  $x[n/3]$ .

¿Cuáles son los datos que se conservan después de la dilatación? Si al realizar  $n/(\alpha)$  el resultado es entero, entonces se conserva; en caso contrario, se pierde. Por ejemplo, el dato ubicado en  $n=1$  se pierde, dado que, al realizar  $1/(2/3)$ , el resultado es 1.5; mientras que, el dato ubicado en  $n=2$  se conserva, porque al realizar  $2/(2/3)$  obtenemos como resultado 3.

### 1.2.5. Combinación de transformaciones

En esta última sección de transformaciones, se aborda la aplicación simultánea de dos o más transformaciones sobre la variable independiente de una señal. Este tipo de operaciones es frecuente en el análisis de señales y, aunque existen distintos procedimientos para realizarlas, a continuación, se propone un método sistemático que facilita su correcta interpretación y evita errores comunes.

El procedimiento recomendado es el siguiente:

1. Factorizar la variable independiente, de manera que el coeficiente que acompaña a la variable tenga magnitud uno y signo positivo. Esto permite identificar de forma clara el tipo de transformaciones involucradas.
2. Aplicar las transformaciones siguiendo el orden de los paréntesis, de afuera hacia adentro. En la práctica, esto implica realizar primero la inversión temporal o el escalamiento (dilatación o compresión), y aplicar el desplazamiento siempre como la última transformación.

Este método garantiza una lectura correcta de la señal transformada y resulta especialmente útil cuando se combinan inversión, escalamiento y desplazamiento en una misma expresión.

#### **Ejemplo 12: Combinación de transformaciones en señal discreta**

Vamos a ilustrar el procedimiento mediante un ejemplo. Supongamos que tenemos la siguiente señal discreta  $x[n]$ :

```

import numpy as np
import matplotlib.pyplot as plt

m = 30      # número de muestras de la señal
nc = 2      # número de ciclos de la señal
n = np.arange(0, m)      # eje temporal discreto
x = np.sin(2 * np.pi * nc * n/m) # señal discreta

plt.stem(n, x)
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal x[n]")
plt.grid(True)

```

Figura 25. Código en Python del Ejemplo 12: generación de la señal discreta.  
La señal obtenida se presenta a continuación:

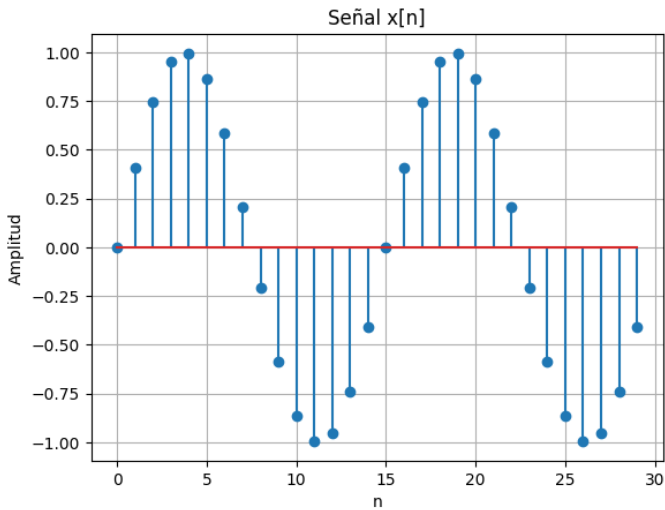


Figura 26. Representación gráfica del Ejemplo 12: señal x[n].

Esta señal es de soporte finito y está comprendida en el intervalo  $[0, 29]$ .

A esta señal se le aplicará la siguiente transformación:

$$x[-2n + 8]$$

Siguiendo el procedimiento descrito anteriormente, se realizan los siguientes pasos.

### Paso 1: Factorización de la variable independiente

$$x[-2n + 8] = x[-2(n - 4)]$$

Esta forma permite identificar claramente las transformaciones involucradas.

### Paso 2: Escalamiento temporal (compresión, $\alpha=2$ )

Se realiza primero la compresión temporal, tomando una de cada  $\alpha$  muestras de la señal original:

```
alpha = 2

n_new = np.arange(0, int(np.ceil(m/alpha)))
x_new = x[::alpha]

plt.stem(n_new, x_new)
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal comprimida ( $\alpha = 2$ )")
plt.grid(True)
```

Figura 27. Código en Python del Ejemplo 12: compresión,  $\alpha=2$ .

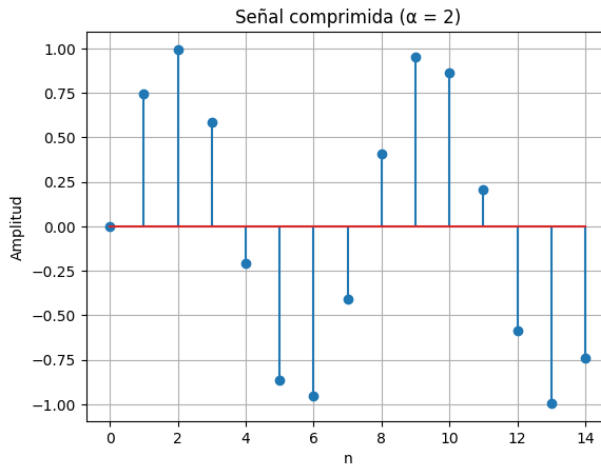


Figura 28. Representación gráfica del Ejemplo 12: compresión,  $\alpha=2$ .

En este primer resultado es importante resaltar que la duración de la señal cambió a la mitad, es decir, por un factor de  $1/\alpha$ . De las 30 muestras iniciales, ahora quedaron 15, ubicadas en el intervalo  $[0, 14]$ .

### Paso 3: Inversión temporal

A continuación, se realiza la inversión temporal reflejando el eje discreto con respecto al origen:

```
n_inv = -n_new

plt.stem(n_inv, x_new)
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal invertida")
plt.grid(True)
```

Figura 29. Código en Python del Ejemplo 12: inversión de la señal.

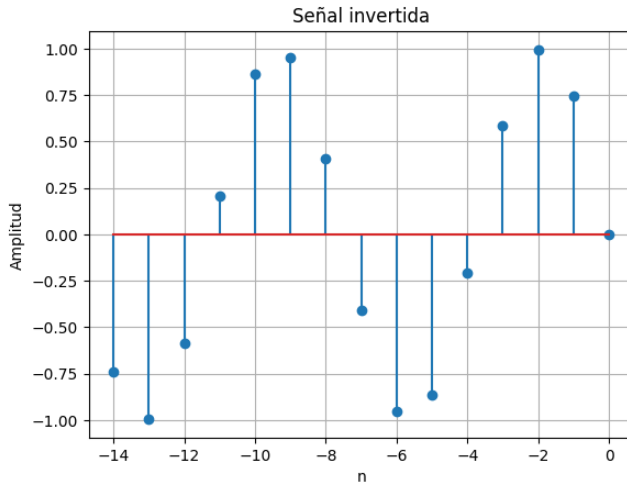


Figura 30. Representación gráfica del Ejemplo 12: inversión de la señal.

La señal resultante tiene la misma cantidad de muestras de la obtenida en el paso anterior, pero con efecto espejo respecto al eje vertical.

#### Paso 4: Desplazamiento temporal, $k=4$

Como paso final, vamos a realizar el desplazamiento temporal, con  $k=4$ .

```
k = 4
n_disp = n_inv + k

plt.stem(n_disp, x_new)
plt.xlabel("n")
plt.ylabel("Amplitud")
plt.title("Señal transformada final")
plt.grid(True)
plt.show()
```

Figura 31. Código en Python del Ejemplo 12: desplazamiento temporal,  $k=4$ .

Obteniendo:

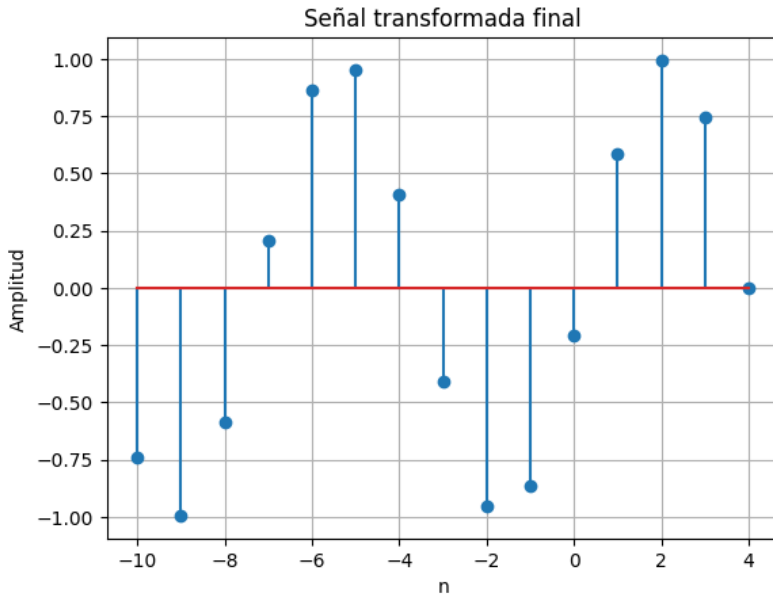


Figura 32. Representación gráfica del Ejemplo 12: resultado final de la transformación.

Finalmente, la señal queda en in intervalo  $[-10, 4]$ .

### Verificación del soporte temporal

Para verificar el soporte de la señal transformada, se evala la expresión  $-2n+8-2n$  en los extremos del soporte original:

$$-2n + 8 = 0 \quad \Rightarrow \quad n = 4$$

$$-2n + 8 = 29 \quad \Rightarrow \quad n = -10.5 \text{ muestra no válida en tiempo discreto}$$

$$-2n + 8 = 28 \quad \Rightarrow \quad n = -10$$

Por lo que la señal transformada queda definida en el intervalo:

$$n \in [-10, 4]$$

La amplitud de la señal transformada en  $n = -10$  corresponde al valor original de  $x(28)$ , mientras que, la amplitud en  $n = 4$  corresponde al valor original de  $x(0)$ .

Este resultado confirma que la transformación ha sido aplicada correctamente y evidencia que, en tiempo discreto, algunas muestras pueden perderse cuando el argumento de la señal no toma valores enteros.

### 1.3. PROPIEDADES DE LOS SISTEMAS

En esta última sección del primer capítulo se introducen las propiedades fundamentales de los sistemas. Un sistema puede ser un dispositivo físico, un circuito, un algoritmo o cualquier mecanismo que permita transformar una señal de entrada en una señal de salida.

Existen:

- Sistemas de tiempo continuo, en los cuales tanto la señal de entrada como la señal de salida son continuas.
- Sistemas de tiempo discreto, en los que tanto la entrada como la salida corresponden a señales discretas.
- Sistemas híbridos, donde la entrada es una señal continua y la salida es una señal discreta, como ocurre, por ejemplo, en un conversor analógico–digital (A/D).

A continuación, se describen algunas de las propiedades más importantes de los sistemas.

#### 1.3.1. Memoria

Se dice que un sistema es sin memoria si el valor de la señal de salida en un instante determinado depende únicamente del valor de la señal de entrada en ese mismo instante.

Un ejemplo de un sistema sin memoria en tiempo continuo es:

$$y(t) = 2x(t)$$

Donde  $y(t)$  es la salida del sistema y  $x(t)$  es la señal de entrada. En este caso, el sistema amplifica la señal por un factor de 2, es decir, por ejemplo, si ingresa una señal constante de 10 [V], la salida será otra señal constante de 20 [V].

En el caso discreto, un ejemplo de sistema sin memoria es:

$$y[n] = 2x[n]$$

Donde  $y[n]$  es la salida del sistema y  $x[n]$  es la señal de entrada discreta.

Por el contrario, un sistema con memoria es aquel en el que la salida depende de valores pasados o futuros de la señal de entrada. Por ejemplo:

$$y(t) = x(t - 2)$$

En ambos casos, para calcular la salida en el instante actual necesito conocer la señal de entrada en un instante pasado. Por ejemplo, para calcular  $y(0)$  es necesario conocer  $x(-2)$ .

### 1.3.2. Causalidad

Se dice que un sistema es causal, si la salida depende únicamente de valores presentes y/o pasados de la señal de entrada.

Un ejemplo de un sistema causal, es:

$$y(t) = x(t) + x(t - 2)$$

Por otro lado, un sistema es no causal si la salida depende de valores futuros de la señal de entrada, como en el caso de:

$$y(t) = x(t + 2)$$

Los sistemas no causales no pueden implementarse en tiempo real, aunque pueden utilizarse en aplicaciones de procesamiento fuera de línea.

### 1.3.3. Estabilidad

Un sistema es estable, si para una entrada acotada (de amplitud finita), la salida también es acotada. Esta propiedad se conoce como estabilidad BIBO (*Bounded Input, Bounded Output*).

La verificación de la estabilidad es fundamental, ya que un sistema inestable puede producir salidas no controladas o no ser físicamente realizable.

Por ejemplo, consideremos el siguiente sistema:

$$y(t) = \frac{x(t)}{t}$$

En este caso, para  $t=0$  la salida no está definida y puede crecer sin límite, incluso si la entrada es acotada. Por lo tanto, ese sistema no es estable.

En contraste, el sistema:

$$y(t) = x^2(t)$$

Sí es estable, dado que, para una entrada de amplitud finita, la salida permanece acotada.

#### 1.3.4. Invertibilidad

Se dice que un sistema es invertible si existe otro sistema (llamado sistema inverso) tal que, al conectarlos en serie, se recupera exactamente la señal de entrada. Es decir, si la salida del primer sistema es  $w(t)$  o  $w[n]$ , entonces el sistema inverso permite volver a obtener  $x(t)$  o  $x[n]$ :

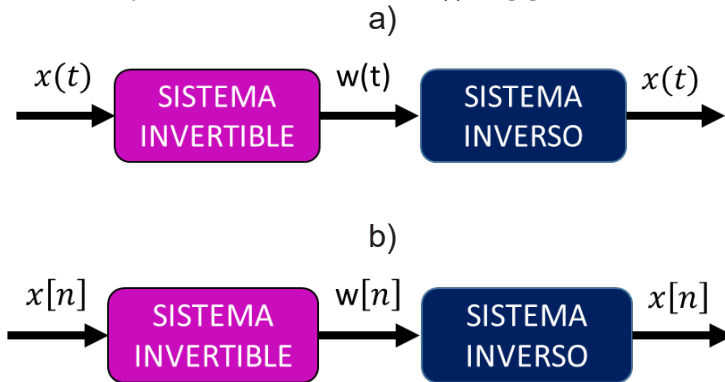


Figura 33. Interconexión sistema invertible-inverso: a) continuo, b) discreto.

Equivalentemente, un sistema es invertible si existe una correspondencia uno a uno entre entrada y salida: entradas diferentes no pueden producir la misma salida.

**Ejemplo 13: Sistema invertible en tiempo continuo**

Considere el sistema:

$$S_1: \quad y(t) = x(2t)$$

Cuyo sistema inverso, es:

$$S_2: \quad y(t) = x(t/2)$$

El primer sistema ( $S_1$ ) comprime la señal de entrada, y el segundo sistema ( $S_2$ ) la dilata, de manera que al aplicar ambos en cascada se recupera nuevamente la señal original  $x(t)$ .

**Ejemplo 14: Sistema discreto que no es invertible**

Ahora analicemos el sistema discreto:

$$S_1: \quad y[n] = x[2n]$$

Aquí también ocurre una compresión en el eje temporal. Sin embargo, a diferencia del caso continuo, en tiempo discreto esta operación descarta muestras: la salida solo conserva los valores de  $x[n]$  en índices pares (0, 2, 4, ...), por lo que se pierde la información correspondiente a las muestras de índices impares (1, 3, 5, ...).

En consecuencia, no es posible recuperar  $x[n]$  únicamente a partir de  $y[n]$ , ya que múltiples señales de entrada distintas pueden producir exactamente la misma salida (por ejemplo, cualquier modificación en las muestras impares de  $x[n]$  no afecta a  $y[n]$ ). Por lo tanto, este sistema no es invertible.

**1.3.5. Linealidad**

Un sistema se dice lineal si cumple el principio de superposición, el cual incluye dos propiedades fundamentales: aditividad y homogeneidad.

Esto significa que, si al sistema se le aplican dos señales de entrada y se combinan linealmente, la salida resultante es la misma combinación lineal de las salidas individuales.

**Ejemplo 15: sistema lineal**

Considere el sistema:

$$y(t) = 2x(t)$$

Definamos dos señales de entrada constantes:

$$x_1(t) = 2, \quad \text{y} \quad x_2(t) = 3$$

Al evaluar cada señal en el sistema, se obtiene:

$$y_1(t) = 2x_1(t) = 4 \quad \text{y} \quad y_2(t) = 2x_2(t) = 6$$

Ahora definimos una nueva señal como la suma de las dos señales de entrada:

$$x_3(t) = x_1(t) + x_2(t) = 5$$

Y al evaluar la señal en el sistema, se obtiene:

$$y_3(t) = 2x_3(t) = 10$$

Finalmente, comparamos:

$$y_3(t) = y_1(t) + y_2(t) = 10$$

Dado que se cumple el principio de superposición, se concluye que el sistema es lineal.

**Ejemplo 16: sistema no lineal**

Consideremos ahora el sistema

$$y(t) = 2x(t) + 1$$

Utilizando las mismas señales de entrada

$$x_1(t) = 2, \quad \text{y} \quad x_2(t) = 3$$

Obtenemos:

$$y_1(t) = 5 \quad \text{y} \quad y_2(t) = 7$$

Definiendo nuevamente

$$x_3(t) = x_1(t) + x_2(t) = 5$$

Y al evaluar la señal en el sistema:

$$y_3(t) = 11$$

Sin embargo,

$$y_1(t) + y_2(t) = 12 \neq y_3(t),$$

Por lo tanto, este sistema no cumple el principio de superposición y se clasifica como no lineal.

### 1.3.6. Invarianza

Un sistema es invariante en el tiempo si al desplazar la señal de entrada y luego evaluarla en el sistema, se obtiene el mismo resultado que al evaluar primero la señal en el sistema y después desplazar la señal de salida en la misma cantidad.

Formalmente, si una señal de entrada  $x[n]$  produce una salida  $y[n]$ , el sistema es invariante si para cualquier desplazamiento  $n_0$  se cumple:

$$x[n - n_0] \xrightarrow{\text{sistema}} y[n - n_0]$$

Es decir, el sistema no depende del instante específico en el que se aplica la señal, sino únicamente de su forma.

#### Ejemplo 17: sistema invariante

Supongamos que tenemos una señal  $x[n]$  la cual se encuentra en el intervalo  $n \in [0, 3]$ , con las amplitudes  $x[n] = \{1, 2, 1, 2\}$ .

Esta señal ingresa al sistema

$$y[n] = x[2n]$$

El cual realiza una compresión de la señal de entrada, por factor de  $\alpha=2$ , así:

```

import numpy as np
import matplotlib.pyplot as plt

m = 4
n = np.linspace(0, m-1, m)
x = np.array([1, 2, 1, 2])

α = 2
n_new = np.linspace(0, int((m-1)/α), int(m/α))
x_new = x[::α]

# Señal original
plt.figure()
plt.stem(n, x)
plt.xlabel("n")
plt.ylabel("x[n]")
plt.title("Señal original x[n]")
plt.grid(True)

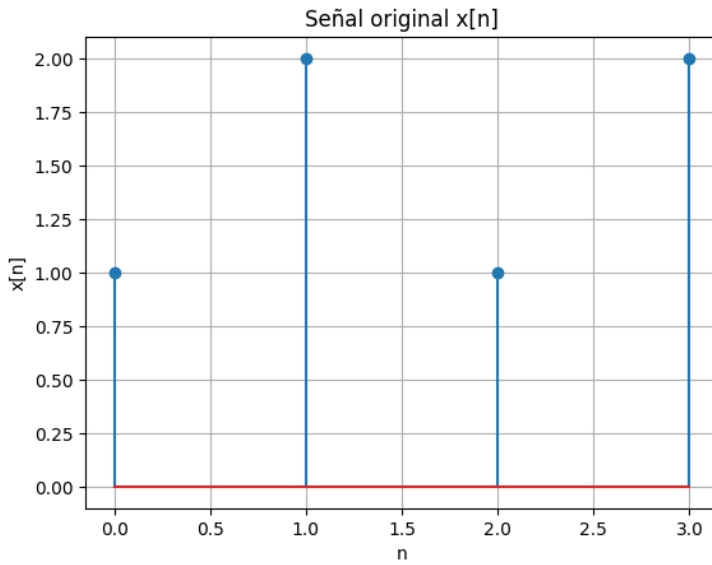
# Señal comprimida
plt.figure()
plt.stem(n_new, x_new)
plt.xlabel("n")
plt.ylabel("y[n]")
plt.title("Señal comprimida y[n] = x[2n]")
plt.grid(True)

```

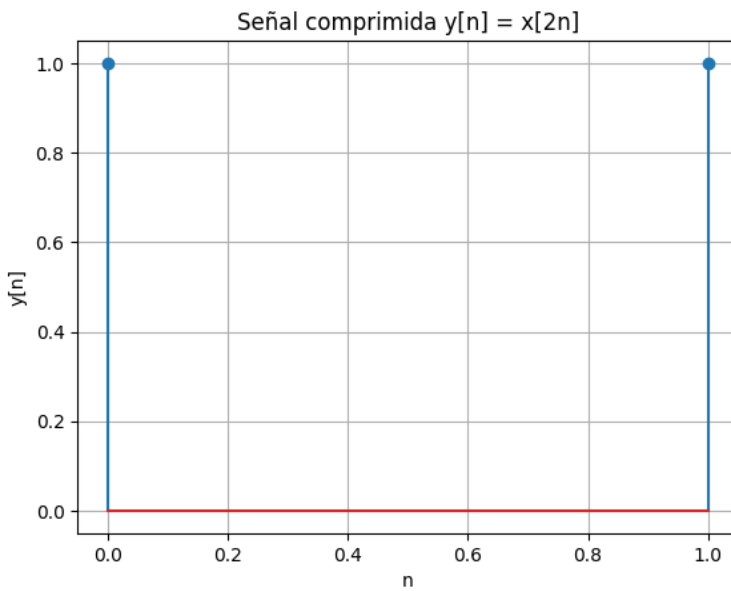
Figura 34. Código en Python señal del Ejemplo 17: señal  $x[n]$  y  $y[n]$ .

Obteniendo esta señal de salida del sistema, para la señal de entrada  $x[n]$ :

A)



B)

Figura 35. Representación gráfica del Ejemplo 17: a) señal  $x[n]$ , b) señal  $y[n]$ .

Ahora, desplacemos la señal de entrada una posición a la derecha, es decir:

$$x_1[n] = x[n - 1]$$

La cual ingresaremos al mismo sistema, obteniendo:

```
# Señal desplazada x1[n] = x[n-1]
m = 4
n = np.arange(0, m+1)
x1 = np.array([0, 1, 2, 1, 2])

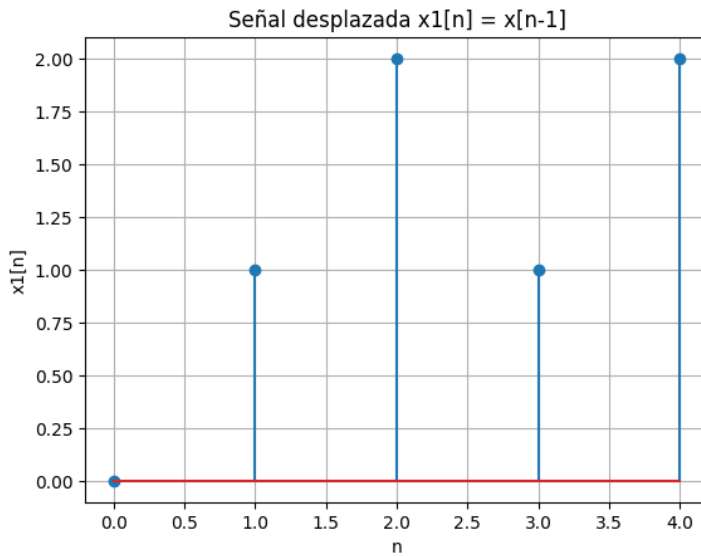
# Sistema: y[n] = x[2n]
alpha = 2
n_new = np.arange(0, int(m/alpha) + 1)
y1 = x1[::alpha]

# Señal desplazada
plt.figure()
plt.stem(n, x1)
plt.xlabel("n")
plt.ylabel("x1[n]")
plt.title("Señal desplazada x1[n] = x[n-1]")
plt.grid(True)

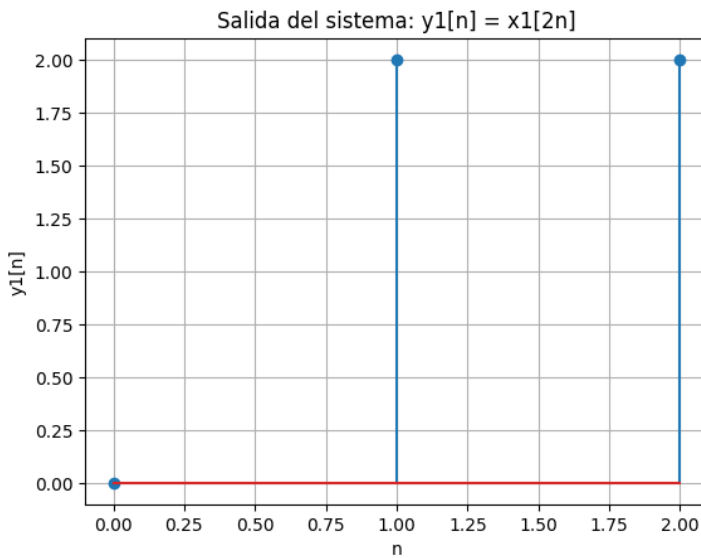
# Salida del sistema para la señal desplazada
plt.figure()
plt.stem(n_new, y1)
plt.xlabel("n")
plt.ylabel("y1[n]")
plt.title("Salida del sistema: y1[n] = x1[2n]")
plt.grid(True)
```

Figura 36. Código en Python señal del Ejemplo 17: señal  $x_1[n]$  y  $y_1[n]$ .

A)



B)

Figura 37. Representación gráfica del Ejemplo 17: a) señal  $x[n]$ , b) señal  $y[n]$ .

Si aplicamos desplazamiento a la derecha de la señal  $y[n]$  obtenida en la Figura 35b, nos damos cuenta que NO es la misma  $y_1[n]$ . Entonces, el sistema NO es invariante.

**Nota conceptual**

Un error común al analizar la invarianza en el tiempo es asumir que toda operación sobre el eje temporal conserva los desplazamientos. Sin embargo, los sistemas que realizan escalamiento temporal (compresión o dilatación) modifican la relación entre las muestras, de modo que un desplazamiento en la señal de entrada no se traduce en el mismo desplazamiento en la señal de salida. Por esta razón, los sistemas del tipo  $y[n]=x[\alpha n]$ , con  $\alpha \neq 1$ , no son invariantes en el tiempo.





## CAPÍTULO II

# Sistemas LTI

## ( Lineales, invariantes en el tiempo)

En este capítulo nos enfocaremos en los sistemas LTI, que son aquellos que cumplen tanto la propiedad de linealidad (principio de superposición) e invarianza temporal (respuesta independiente del tiempo absoluto). Estos sistemas permiten determinar la salida para cualquier entrada mediante la operación de convolución con la respuesta al impulso. Conoceremos en qué consiste la convolución, como calcularla en sistemas continuos y discretos, y cómo nos podemos apoyar en Python para su cálculo.

Al finalizar el Capítulo, deberás estar en capacidad de:

1. Representar una señal discreta en términos de impulsos desplazados.
2. Realizar la convolución entre dos señales discretas aplicando la propiedad distributiva.
3. Realizar la convolución entre dos señales discretas aplicando la ecuación de convolución.
4. Realizar la convolución entre dos señales discretas utilizando Python.
5. Realizar la convolución entre dos señales continuas aplicando la ecuación de convolución.
6. Identificar si un sistema LTI cumple con las propiedades de memoria, causalidad, invertibilidad y estabilidad, a partir del análisis de la respuesta al impulso.

Un sistema es Lineal e Invariante en el Tiempo (LTI) si cumple simultáneamente las propiedades de linealidad e invarianza temporal. Este tipo de sistemas ocupa un lugar central en el análisis de señales, ya que permite describir completamente su comportamiento a partir de una única señal: la respuesta al impulso.

En efecto, un sistema LTI queda completamente caracterizado por su respuesta al impulso. Esto significa que, si se conoce cómo responde el sistema ante una señal impulso, es posible determinar su salida frente a cualquier señal de entrada mediante la operación de convolución.

## 2.1. SEÑAL IMPULSO Y SEÑAL ESCALÓN EN TIEMPO DISCRETO

La señal impulso, denotada como  $\delta[n]$ , es la señal “más importante” cuando estamos trabajando con señales discretas, porque permite descomponer cualquier señal discreta como combinación de impulsos desplazados.

Esta señal impulso, equivale a:

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

Es decir, solamente existe en el origen, y para los demás valores de  $n$  es cero.

Gráficamente, esta señal es:

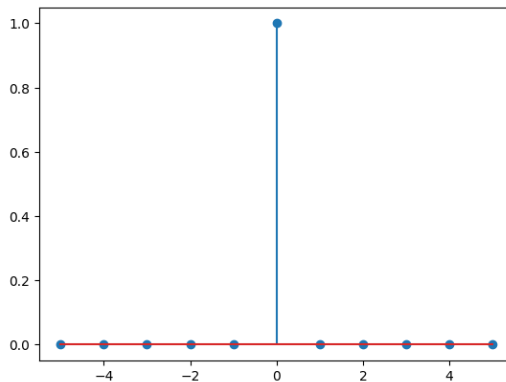


Figura 38. Señal impulso discreta.

La cual se generó con el siguiente código en Python:

```

import numpy as np
import matplotlib.pyplot as plt

n = np.linspace(-5, 5, 11)
impulso = np.array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0])
plt.stem(n, impulso) # visualización señal impulso

```

Figura 39. Código en Python para dibujar la señal impulso discreta.

Ahora bien, aplicando desplazamiento a la señal impulso, tendremos que:

$$\delta[n-1] = \begin{cases} 1 & n = 1 \\ 0 & n \neq 1 \end{cases}$$

$$\delta[n+1] = \begin{cases} 1 & n = -1 \\ 0 & n \neq -1 \end{cases}$$

...

$$\delta[n-2] = \begin{cases} 1 & n = 2 \\ 0 & n \neq 2 \end{cases}$$

Y de forma general,

$$\delta[n-k] = \begin{cases} 1 & n = k \\ 0 & n \neq k \end{cases}$$

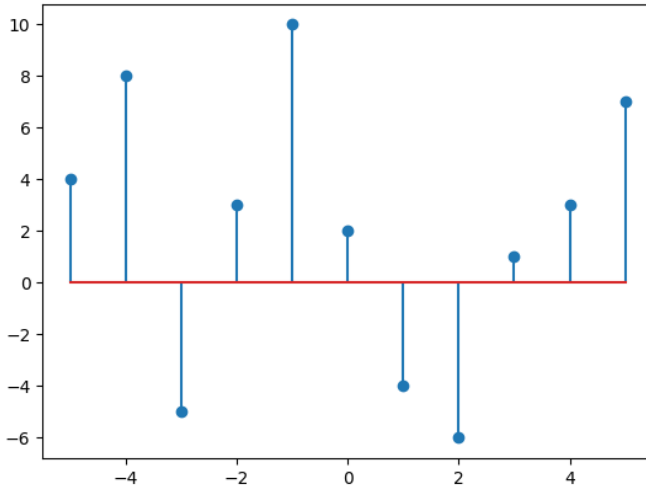
De tal forma que, cualquier señal discreta se puede representar a partir de impulsos desplazados, así:

$$x[n] = \sum_{k=-\infty}^{\infty} a_k \delta[n-k]$$

Donde  $a_k$  corresponde a la amplitud del impulso ubicado en  $n = k$ .

### Ejemplo 18: representación señal discreta a partir de impulsos desplazados

A manera de ejemplo, vamos a partir de la siguiente señal  $x[n]$ , la cual se presenta a continuación

Figura 40. Señal  $x[n]$  del Ejemplo 18.

La cual se dibuja a partir del siguiente código en Python:

```
# Señal discreta x[n]

import numpy as np
import matplotlib.pyplot as plt

n = np.linspace(-5, 5, 11)
impulso = np.array([4, 8, -5, 3, 10, 2, -4, -6, 1, 3, 7])
plt.stem(n, impulso) # visualización señal x[n]
```

Figura 41. Código en Python del Ejemplo 18.

Esta señal  $x[n]$  se puede representar así:

$$x[n] = 4\delta[n+5] + 8\delta[n+4] - 5\delta[n+3] + 3\delta[n+2] + 10\delta[n+1] + 2\delta[n] - 4\delta[n-1] - 6\delta[n-2] \\ + \delta[n-3] + 3\delta[n-4] + 7\delta[n-5]$$

Existe otra señal que también es importante cuando modelamos sistemas discretos, y corresponde a la señal escalón unitario. Esta señal está conformada por infinitos impulsos desplazados, los cuales inician en el origen y

terminan en infinito. Matemáticamente, esta señal la expresamos como:

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

Estrictamente hablando, no podemos graficar esta señal en Python, dado que necesitaríamos infinitos impulsos, pero vamos a graficar una parte de la señal, así:

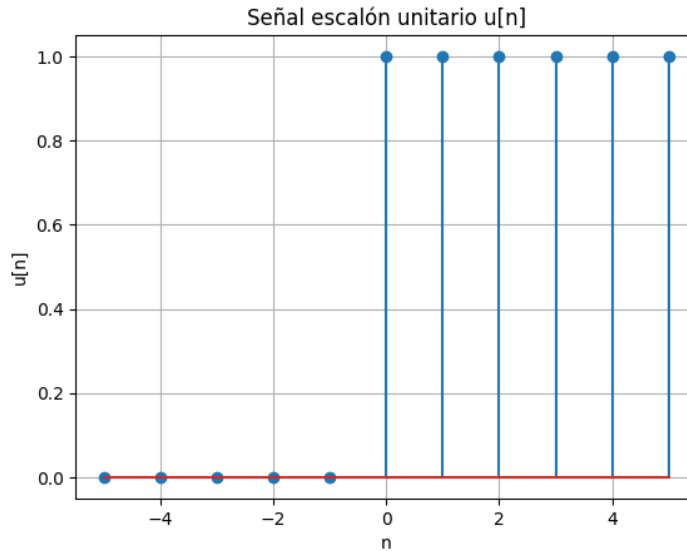


Figura 42. Señal u[n], visualización entre [-5 5].

#### Nota aclaratoria

Aunque la señal escalón unitario está definida para un número infinito de muestras, en la práctica se visualiza sobre un intervalo finito del eje temporal.

Esta señal  $u[n]$ , la podemos también expresar en términos de impulsos desplazados, así:

$$u[n] = \sum_{k=0}^{\infty} \delta[n - k]$$

Adicionalmente, podemos expresar la señal impulso en términos de la señal escalón, así:

$$\delta[n] = u[n] - u[n - 1]$$

La representación de señales discretas a partir de impulsos desplazados permite entender cómo un sistema responde ante excitaciones elementales. En particular, en sistemas LTI, el conocimiento de la respuesta del sistema ante un impulso unitario resulta suficiente para caracterizar completamente su comportamiento.

A partir de esta idea, surge naturalmente la operación de convolución, la cual permite determinar la salida de un sistema LTI ante cualquier señal de entrada, combinando las contribuciones de impulsos desplazados ponderados por la señal de entrada.

## 2.2. CONVOLUCIÓN EN TIEMPO DISCRETO

Es una operación matemática que se realiza entre dos señales, típicamente la señal de entrada a un sistema LTI, denotada como  $x[n]$ , y la respuesta al impulso del sistema, denotada como  $h[n]$ .

La salida del sistema,  $y[n]$ , se define como la convolución entre ambas señales, es decir:

$$y[n] = x[n] \otimes h[n]$$

donde  $\otimes$  es el operador de convolución. La salida del sistema puede interpretarse como una suma ponderada de respuestas al impulso desplazadas y escaladas, en la que cada valor de la señal de entrada actúa como el peso de cada contribución.

Para poder entender la operación de convolución, partiremos de los siguientes resultados de convolucionar una señal de entrada por un impulso desplazado:

$$x[n] \otimes \delta[n] = x[n]$$

$$x[n] \otimes \delta[n - 1] = x[n - 1]$$

$$x[n] \otimes \delta[n - 2] = x[n - 2]$$

$$x[n] \otimes \delta[n + 1] = x[n + 1]$$

$$x[n] \otimes \delta[n + 2] = x[n + 2]$$

...

Y de forma general,

$$x[n] \otimes \delta[n - k] = x[n - k]$$

De tal forma que, cuando se convoluciona una señal discreta por un impulso desplazado  $k$  posiciones, como resultado se tiene la señal desplazada por el mismo valor de desplazamiento  $k$ .

A continuación, vamos a conocer dos métodos para realizar la convolución discreta. El primero, utiliza la propiedad distributiva de la convolución, y el segundo, aplica directamente la definición matemática de convolución.

### 2.2.1. Método 1 (Propiedad Distributiva)

**Ejemplo 19: convolución de señales discretas, método propiedad distributiva.**

Supongamos que tenemos una señal  $x[n]$  y una señal  $h[n]$ , definidas como:

$$x[n] = 3\delta[n] + 2\delta[n - 1] - 4\delta[n - 2]$$

$$h[n] = 2\delta[n] - 2\delta[n - 1]$$

Entonces la salida del sistema LTI, la podemos expresar como:

$$y[n] = x[n] \otimes h[n] = x[n] \{h_1[n] + h_2[n]\} = x[n] \otimes h_1[n] + x[n] \otimes h_2[n]$$

Es decir, que para nuestro ejemplo tenemos que realizar dos convoluciones, dado que, la respuesta al impulso está conformada por dos impulsos.

Hacemos,

$$h[n] = h_1[n] + h_2[n]$$

Donde,

$$h_1[n] = 2\delta[n]$$

$$h_2[n] = -2\delta[n-1]$$

El resultado de

$$\begin{aligned} x[n] \otimes h_1[n] &= \{3\delta[n] + 2\delta[n-1] - 4\delta[n-2]\} \otimes \{2\delta[n]\} \\ &= 2\{3\delta[n] + 2\delta[n-1] - 4\delta[n-2]\} \\ &= 6\delta[n] + 4\delta[n-1] - 8\delta[n-2] \end{aligned}$$

Y el de,

$$\begin{aligned} x[n] \otimes h_2[n] &= \{3\delta[n] + 2\delta[n-1] - 4\delta[n-2]\} \otimes \{-2\delta[n-1]\} \\ &= -2\{3\delta[n-1] + 2\delta[n-2] - 4\delta[n-3]\} \\ &= -6\delta[n-1] - 4\delta[n-2] + 8\delta[n-3] \end{aligned}$$

Sumando los dos resultados anteriores, obtenemos:

$$\begin{aligned} y[n] &= 6\delta[n] + 4\delta[n-1] - 8\delta[n-2] - 6\delta[n-1] - 4\delta[n-2] + 8\delta[n-3] \\ y[n] &= 6\delta[n] + \{4\delta[n-1] - 6\delta[n-1]\} + \{-8\delta[n-2] - 4\delta[n-2]\} + 8\delta[n-3] \\ y[n] &= 6\delta[n] - 2\delta[n-1] - 12\delta[n-2] + 8\delta[n-3] \end{aligned}$$

### 2.2.2. Método 2 (aplicando la ecuación de convolución)

La convolución entre dos señales, se define a continuación:

$$y[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

Es decir que, una de las dos señales la dejamos “quieta” ( $x[n]$ ), y la otra la invertimos y la desplazamos ( $h[n]$ ). Vamos a aplicar este método con el mismo ejemplo que utilizamos en el método 1.

**Ejemplo 20: convolución de señales discretas, método ecuación de convolución.**

**Paso 0:** grafica de las dos señales:

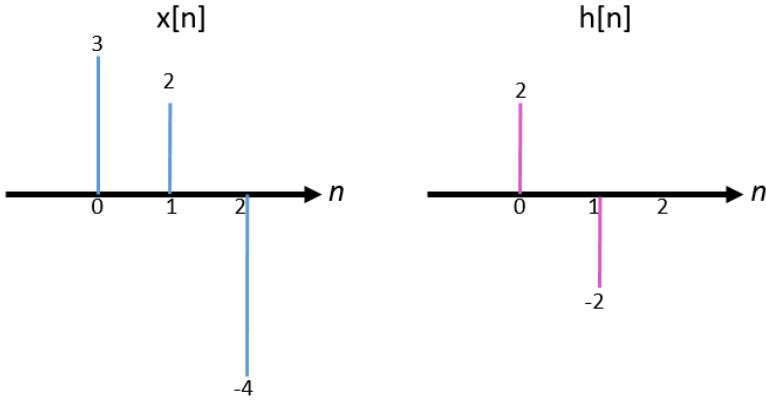


Figura 43. Ejemplo 20, convolución: señal  $x[n]$  y  $h[n]$ .

**Paso 1:** se realiza cambios de variable en las dos señales, y la inversión en una de ellas, así:

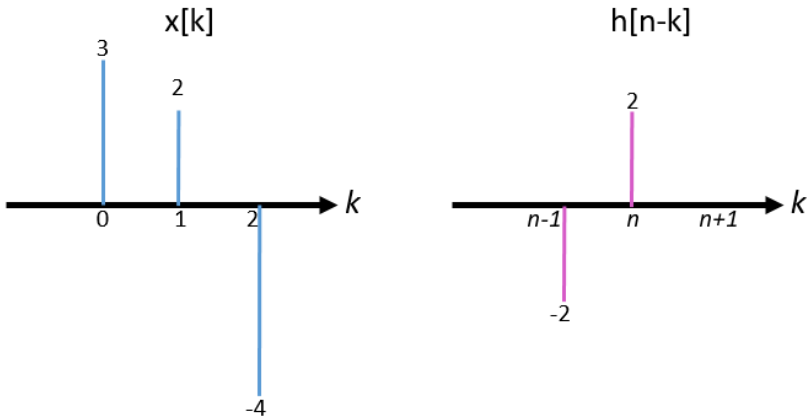


Figura 44. Ejemplo 20: Paso 1 de convolución discreta.

**Paso 2:** ubicar la señal  $h[n-k]$  a la izquierda de la señal  $x[k]$ , de la siguiente forma:

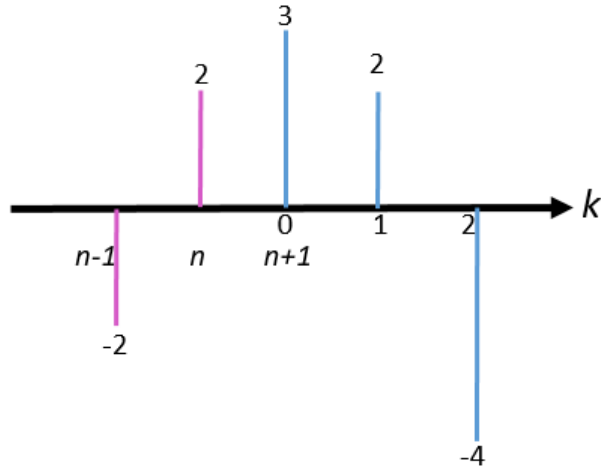


Figura 45. Ejemplo 20: Paso 2 de convolución discreta.

Dado que las señales no están superpuestas, la salida es  $y[n] = 0$ , para  $n < 0$ .

**Paso 3:** desplazar la señal  $h[n - k]$  hasta que un impulso quede superpuesto en la señal  $x[n]$ , así:

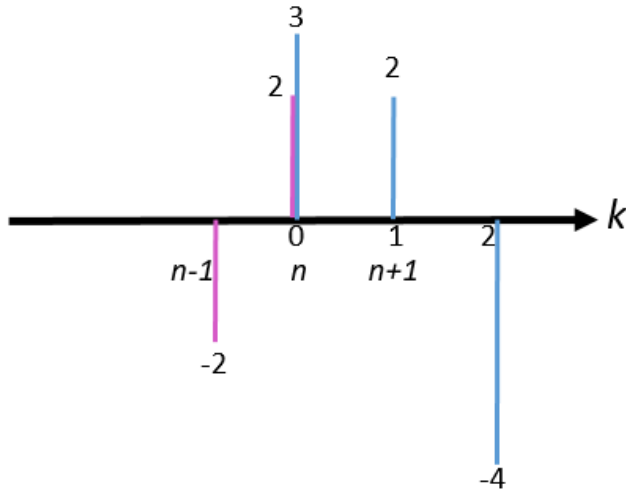


Figura 46. Ejemplo 20: Paso 3 de convolución discreta.

El resultado es igual a multiplicar las amplitudes del impulso que tienen en común ambas señales. Entonces  $y[n] = 6$ , para  $n = 0$ .

**Paso 4:** desplazar la señal  $h[n - k]$  una posición a la derecha.

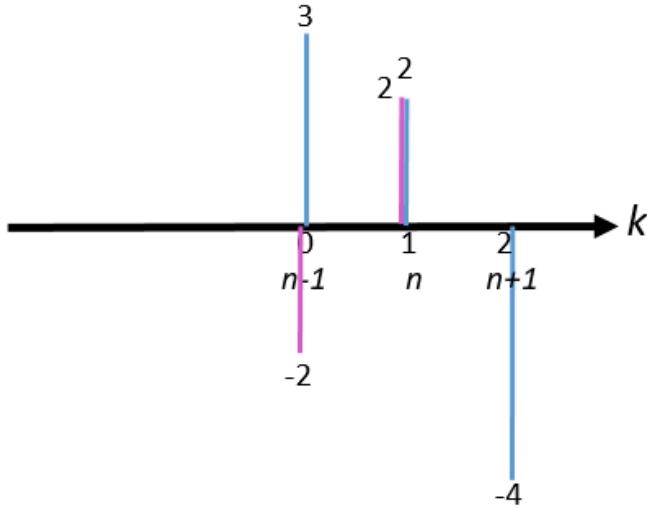


Figura 47. Ejemplo 20: Paso 4 de convolución discreta.

Para este caso, dos impulsos están superpuestos, por lo que se deben realizar dos multiplicaciones y posteriormente la suma de estos resultados. De tal forma que,  $y[n] = (3 \times -2) + (2 \times 2) = -6 + 4 = -2$ , para  $n = 1$ .

**Paso 5:** desplazar la señal  $h[n - k]$  dos posiciones a la derecha.

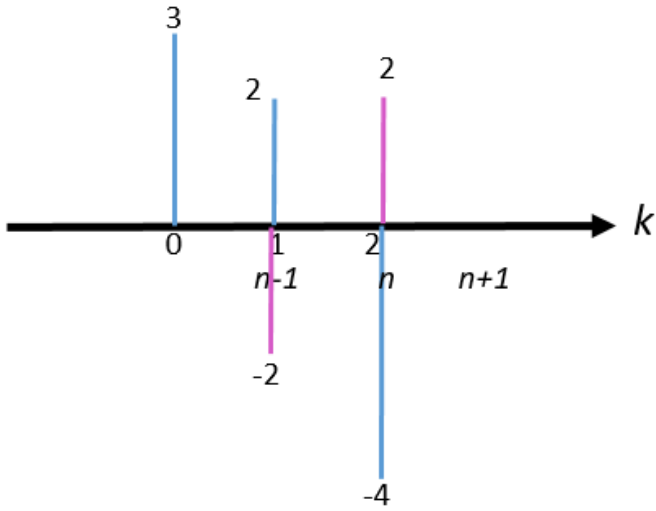


Figura 48. Ejemplo 20: Paso 5 de convolución discreta.

De nuevo, dos impulsos están superpuestos, por lo que se deben realizar dos multiplicaciones y posteriormente la suma de estos resultados. De tal forma que,  $y[n] = (-4 \times 2) + (2 \times -2) = -8 - 4 = -12$ , para  $n = 2$ .

**Paso 6:** desplazar la señal  $h[n - k]$  tres posiciones a la derecha.

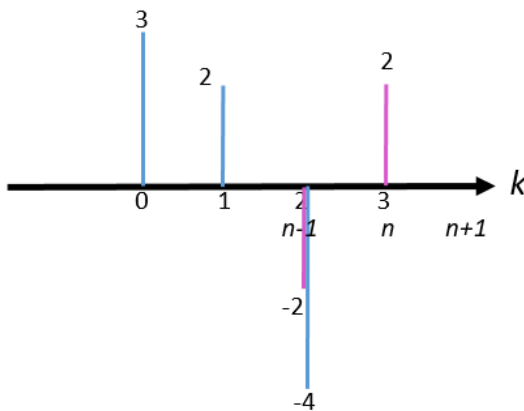


Figura 49. Ejemplo 20: Paso 6 de convolución discreta.

Para este caso, las señales se superponen en un impulso. De tal forma que,  $y[n] = (-4 \times -2) = 8$ , para  $n = 3$ .

**Paso 7:** desplazar la señal  $h[n - k]$  cuatro posiciones a la derecha.

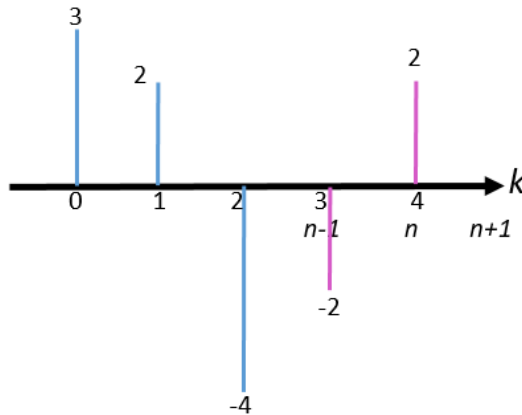


Figura 50. Ejemplo 20: Paso 7 de convolución discreta.

Como se aprecia en la figura anterior, las señales no están superpuestas, por lo que su salida es  $y[n] = 0$ , para  $n \geq 4$ .

En resumen,

$$y[n] = \begin{cases} 0 & n < 0 \\ 6 & n = 0 \\ -2 & n = 1 \\ -12 & n = 2 \\ 8 & n = 3 \\ 0 & n \geq 4 \end{cases}$$

O, de forma equivalente,

$$y[n] = 6\delta[n] - 2\delta[n - 1] - 12\delta[n - 2] + 8\delta[n - 3].$$

Si comparas el resultado con el obtenido con el método 1, es el mismo. Aunque este método puede ser un poco más largo que el método 1, nos servirá para entender la convolución en señales continuas.

Como se observa, el resultado obtenido mediante la aplicación directa de la ecuación de convolución coincide con el obtenido utilizando la propiedad distributiva. Aunque este segundo método resulta más extenso, permite comprender de manera más profunda el significado geométrico y operacional de la convolución, lo cual será fundamental en el análisis de señales continuas.

### 2.2.3. Convolución de señales discretas utilizando Python

A continuación, vamos a realizar la convolución del Ejemplo 19 utilizando la librería numpy de Python:

```
import numpy as np

# Primera señal
n1 = np.arange(0, 3)
x1 = np.array([3, 2, -4])

# Segunda señal
n2 = np.arange(0, 2)
x2 = np.array([2, -2])

plt.stem(n1, x1)
plt.title("Señal 1: x1[n]")
plt.xlabel("n")
plt.ylabel("x1[n]")
plt.grid(True)
plt.show()

plt.stem(n2, x2)
plt.title("Señal 2: x2[n]")
plt.xlabel("n")
plt.ylabel("x2[n]")
plt.grid(True)
plt.show()

# Convolución usando NumPy
y = np.convolve(x1, x2)

# Eje de tiempo discreto para la convolución
n_conv = np.arange(n1[0] + n2[0], n1[-1] + n2[-1] + 1)

# Gráfica de la convolución
plt.stem(n_conv, y)
plt.title("Convolución: y[n] = x1[n] * x2[n]")
plt.xlabel("n")
plt.ylabel("y[n]")
plt.grid(True)
plt.show()
```

Figura 51. Código en Python del Ejemplo 19.

Este código define dos señales en tiempo discreto, primero realiza las graficas de forma individual y luego calcula y visualiza su convolución discreta. Primero, se crean los vectores de tiempo  $n_1$  y  $n_2$  junto con las amplitudes de las señales  $x_1[n]$  y  $x_2[n]$ . A continuación, cada señal se representa gráficamente utilizando *stem*, mostrando sus muestras y su soporte temporal.

Posteriormente, se calcula la convolución discreta usando la función *np.convolve*.

A continuación, se construye el eje de tiempo de la señal resultante sumando los índices iniciales y finales de ambas señales originales, de la siguiente forma:

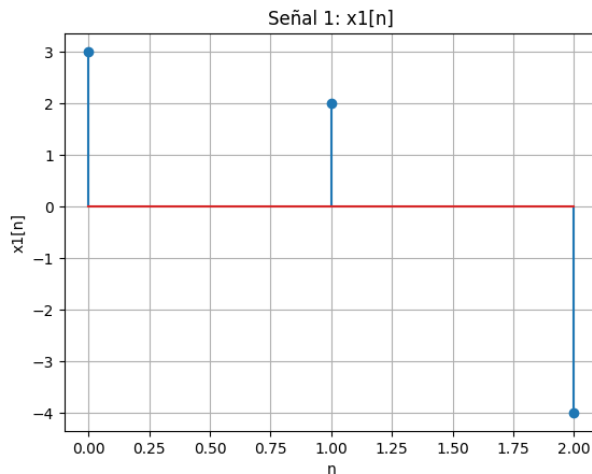
$$n_{conv} = np.arange(n1[0] + n2[0], n1[-1] + n2[-1] + 1)$$

Esto se debe a que el índice inicial de la convolución corresponde a la suma de los índices iniciales de los vectores de tiempo discreto  $n_1$  y  $n_2$ , mientras que el índice final se obtiene a partir de la suma de sus índices finales.

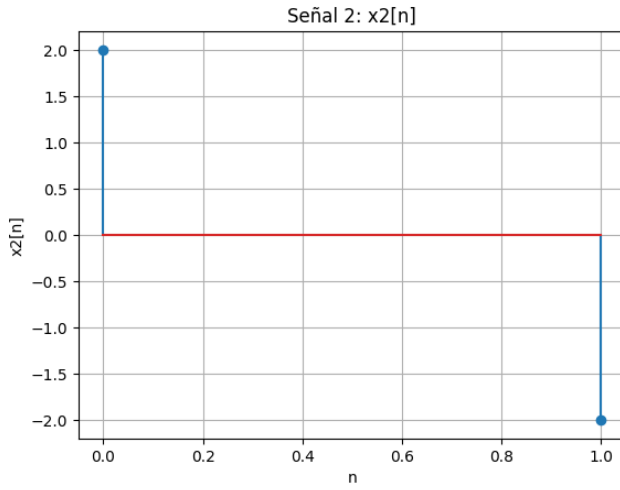
Finalmente, se grafica la señal convolucionada, lo que permite visualizar correctamente su duración y su desplazamiento en el tiempo discreto.

Las señales  $x_1[n]$  y  $x_2[n]$  y la salida de la convolución,  $y[n]$ , se presentan a continuación.

A)



B



C

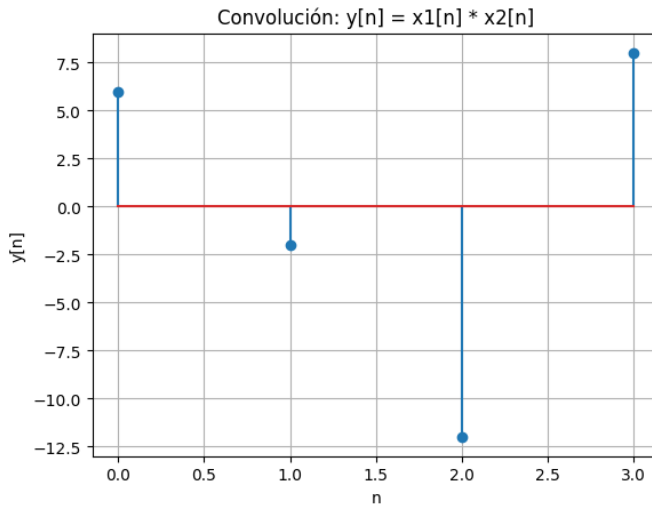


Figura 52. Representación gráfica del Ejemplo 19: a) señal de entrada  $x_1[n]$ , b) señal de entrada  $x_2[n]$ , c) resultado de la convolución,  $y[n]$ .

Este ejemplo en Python permite comprender de forma clara y práctica el proceso de convolución discreta, evidenciando cómo el soporte temporal y la forma de la señal resultante dependen directamente de las señales originales.

### 2.3. SEÑAL IMPULSO Y SEÑAL ESCALÓN EN TIEMPO CONTINUO

De forma similar al caso en tiempo discreto, también existe la señal impulso continuo y la señal escalón continua.

Primero que todo, vamos a partir de la siguiente señal:

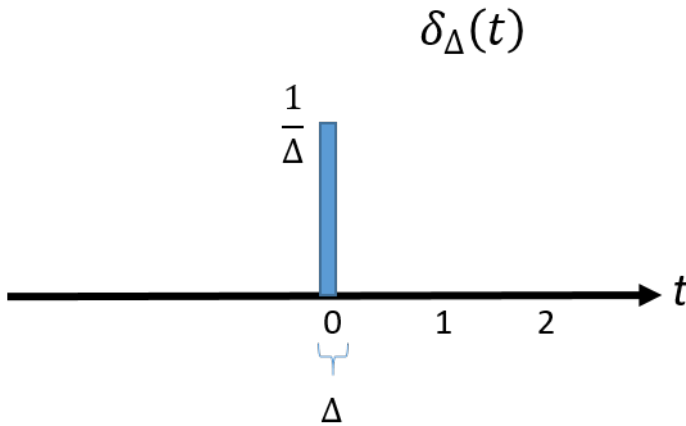


Figura 53. Señal  $\delta_{\Delta}(t)$  en tiempo continuo.

Esta señal existe en un rango de tiempo muy pequeño, denominado  $\Delta$ , y la amplitud es  $1/\Delta$ . De tal forma que, el área de esta señal siempre será de 1, independiente de qué tan pequeña sea  $\Delta$ . Este tipo de señal no representa una señal físicamente realizable, sino una construcción matemática que permite modelar eventos de duración extremadamente corta y analizar sistemas dinámicos.

Ahora bien, cuando  $\Delta \rightarrow 0$ , se tiene una señal que “existe” solamente en  $t = 0$ , conocida como señal impulso en tiempo continuo,  $\delta(t)$ .

En el límite cuando  $\Delta \rightarrow 0$ , se obtiene la señal impulso en tiempo continuo,  $\delta(t)$ , la cual no es una función ordinaria, sino un modelo matemático ideal. Su principal propiedad es que su área es unitaria y se encuentra concentrada en  $t = 0$ , es decir:

$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

De manera intuitiva, el impulso se interpreta como una señal nula en todo instante, excepto en  $t=0$ , donde concentra toda su área.

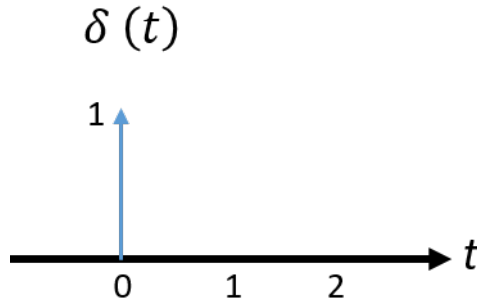


Figura 54. Señal impulso,  $\delta(t)$ , en tiempo continuo.

Es común representar gráficamente el impulso mediante un pulso centrado en  $t = 0$  con amplitud unitaria. Esta elección no implica que el impulso ideal tenga un valor definido en  $t = 0$ , sino que corresponde a una normalización conveniente para fines de visualización. En todos los casos, la propiedad fundamental que se preserva es que el área bajo la señal sea igual a uno; por esta razón, se utiliza la flecha para hacer énfasis en el área, a diferencia del impulso en tiempo discreto.

Por otra parte, la señal escalón unitario la podemos encontrar a partir de la señal impulso, así:

$$u(t) = \int_{-\infty}^t \delta(\tau) d\tau$$

La señal escalón unitario tiene amplitud igual a 1 para todos los valores de  $t$  mayores o iguales que cero, y amplitud nula para valores negativos de  $t$ , es decir:

$$u(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

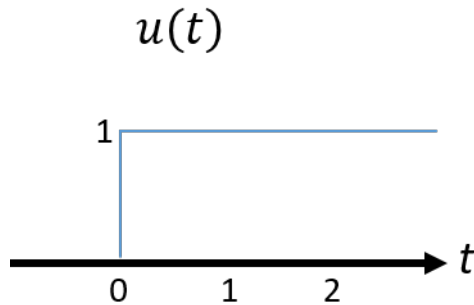


Figura 55. Señal escalón,  $u(t)$ , en tiempo continuo.

Adicionalmente, se puede expresar la señal  $\delta(t)$  a partir de la señal

$u(t)$ , así:

$$\delta(t) = \frac{d(u(t))}{dt}$$

Es decir, la señal impulso puede interpretarse como la derivada de la señal escalón unitario.

## 2.4. CONVOLUCIÓN EN TIEMPO CONTINUO

La convolución es una operación fundamental en el análisis de sistemas, ya que permite determinar la respuesta de un sistema a cualquier señal de entrada a partir de su respuesta al impulso. Desde un punto de vista geométrico, la convolución mide el grado de solapamiento entre una señal y una versión invertida y desplazada de otra señal.

La convolución de la señal  $x(t)$  con la señal  $h(t)$  se expresa matemáticamente, como:

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau = x(t) \otimes h(t)$$

Es decir, se debe hacer cambio de variable en la señal  $x(t)$  de  $t \rightarrow \tau$ ; mientras que, en la señal  $h(t)$  se realiza el cambio de  $t \rightarrow t - \tau$  (esto implica

hacer una inversión en la señal y desplazamiento). Este cambio permite analizar la interacción entre ambas señales mediante una inversión temporal y un desplazamiento progresivo de una de ellas.

Vamos a ilustrar este proceso, con un ejemplo:

Sea  $x(t)$  y  $h(t)$ , las siguientes señales:

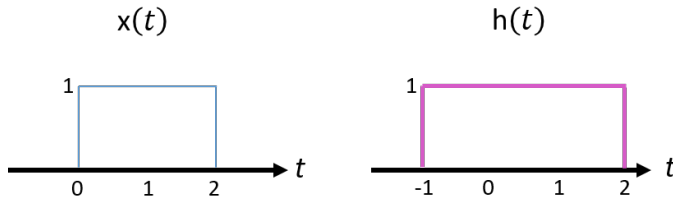


Figura 56. Señal  $x(t)$  y  $h(t)$ : ejemplo de convolución en tiempo continuo.

El paso preliminar consiste en el cambio de variable en las dos señales, quedando así:

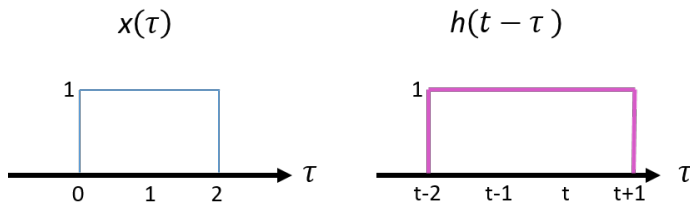


Figura 57. Señal  $x(\tau)$  y  $h(t - \tau)$ : ejemplo de convolución en tiempo continuo.

Nota aclaratoria:

En el análisis gráfico que sigue, se denomina borde externo al extremo inicial de la señal y borde interno al extremo final de la misma, considerando el sentido del desplazamiento.

**Paso 1:** la señal  $h(t - \tau)$  se ubica a la izquierda de la señal  $x(\tau)$ .



Figura 58. Paso 1: ejemplo de convolución en tiempo continuo.

Para este caso, se obtiene que  $y(t) = 0$  para  $t+1 < 0$ ,  $\therefore t < -1$ .

En este intervalo no existe solapamiento entre las señales, por lo que la integral de convolución es nula.

**Paso 2:** la señal  $h(t - \tau)$  se desplaza a la derecha hasta que el borde externo de  $h(t - \tau)$  se superpone con el borde externo de  $x(\tau)$ .

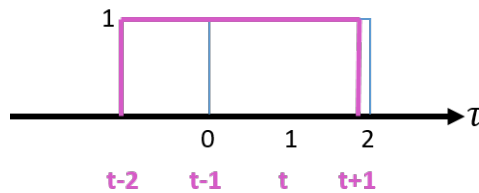


Figura 59. Paso 2: ejemplo de convolución en tiempo continuo.

La salida la obtenemos con la siguiente integral,

$$y(t) = \int_0^{t+1} d\tau = \tau/0^{t+1} = t + 1$$

Nota aclaratoria:

Los límites de integración se determinan observando el intervalo en el que ambas señales presentan solapamiento distinto de cero.

Para este caso, los límites de la integral corresponden al valor inicial y final de la zona en común que tienen las dos señales, que en este caso inicia en 0 y termina en  $t + 1$ .

El resultado es válido para,  $t > -1$  y  $t + 1 < 2 \therefore t < 1$ . Es decir, agrupando las dos condiciones se tiene que:

$$-1 < t < 1$$

**Paso 3:** la señal  $h(t - \tau)$  se desplaza a la derecha hasta que el borde interno de  $h(t - \tau)$  se superpone con el borde interno de  $x(\tau)$ .

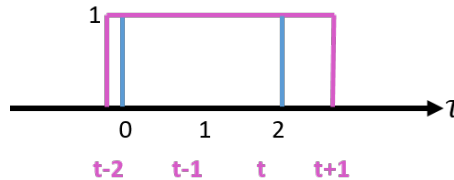


Figura 60. Paso 3: ejemplo de convolución en tiempo continuo.

La salida la obtenemos con la siguiente integral,

$$y(t) = \int_0^2 d\tau = \tau/0 = 2$$

Los límites de la integral corresponden al valor inicial y final de la zona en común que tienen las dos señales, que en este caso inicia en 0 y termina en 2. El resultado es válido para  $t > 1$ , y  $t - 2 < 0 \therefore t < 2$ . Es decir, agrupando las dos condiciones se tiene que:

$$1 < t < 2$$

**Paso 4:** la señal  $h(t - \tau)$  se desplaza a la derecha hasta que el borde interno de  $h(t - \tau)$  se superpone con el borde externo de  $x(\tau)$ .

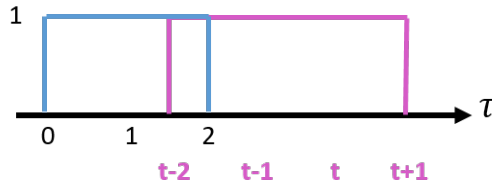


Figura 61. Paso 4: ejemplo de convolución en tiempo continuo.

La salida la obtenemos con la siguiente integral,

$$y(t) = \int_{t-2}^2 d\tau = \tau \Big|_{t-2}^2 = 2 - t + 2 = 4 - t$$

Los límites de la integral corresponden al valor inicial y final de la zona en común que tienen las dos señales, que en este caso inicia en  $t - 2$  y termina en 2. El resultado es válido para  $t > 2$ , y  $t - 2 < 2 \therefore t < 4$ . Es decir, agrupando las dos condiciones se tiene que:

$$2 < t < 4$$

**Paso 5:** la señal  $h(t - \tau)$  se desplaza a la derecha y las dos señales no tienen zona en común.

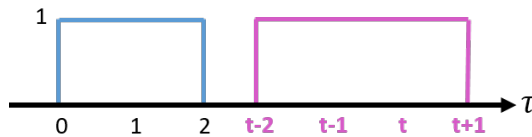


Figura 62. Paso 5: ejemplo de convolución en tiempo continuo.

Para este caso, se obtiene que  $y(t) = 0$  para  $t - 2 > 2, \therefore t > 4$ .

En resumen, se tiene que:

$$y(t) = \begin{cases} 0 & t < -1 \\ t+1 & -1 < t < 1 \\ 2 & 1 < t < 2 \\ 4-t & 2 < t < 4 \\ 0 & t > 4 \end{cases}$$

Este procedimiento puede reproducirse de forma numérica y visual mediante herramientas de simulación, lo que permite verificar cada uno de los tramos obtenidos analíticamente. En este libro, dicha verificación se realizará utilizando Python como entorno de experimentación.

Y se obtiene la siguiente gráfica de la señal de salida:

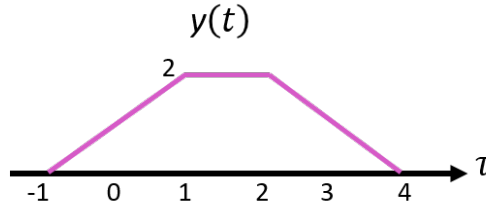


Figura 63. Señal  $y(t)$ : ejemplo de convolución en tiempo continuo.

## 2.5. PROPIEDADES DE LOS SISTEMAS LTI

Una vez definida la respuesta al impulso, podemos analizar propiedades fundamentales del sistema. Cuando tenemos un sistema LTI (Lineal e Invariante en el Tiempo), la verificación de las propiedades se realiza examinando la respuesta al impulso,  $h[n]$ .

### 2.5.1. Memoria

Se dice que un sistema LTI no tiene memoria, si se cumple que:

$$h[n] = C\delta[n]$$

donde  $C$  es una constante. En caso contrario, el sistema tiene memoria. Por ejemplo, supongamos que el sistema es:

$$y[n] = 2x[n] + 3x[n - 1]$$

Lo primero que debemos hacer es identificar la respuesta al impulso del sistema. Para ello nos preguntamos cuál debe ser el valor de  $h[n]$ , para que al convolucionarla con  $x[n]$  se obtenga  $2x[n] + 3x[n - 1]$ . La respuesta es:

$$h[n] = 2\delta[n] + 3\delta[n - 1]$$

Teniendo en cuenta que el valor obtenido no cumple con  $h[n] = C\delta[n]$ , decimos entonces que el sistema tiene memoria.

### 2.5.2. Causalidad

Un sistema LTI es causal si su respuesta al impulso puede expresarse como:

$$h[n] = \sum_{k=0}^m a_k \delta[n - k]$$

Donde  $a_k$  corresponde a la amplitud del impulso ubicado en  $\delta[n - k]$ ; mientras que  $k$  está definido para valores positivos, incluidos el 0.

Supongamos que tenemos el sistema del ejemplo anterior, en el que la respuesta al impulso es  $h[n] = 2\delta[n] + 3\delta[n - 1]$ , entonces, el sistema es causal. Ahora bien, si tuviésemos el sistema  $y[n] = 2x[n] + 3x[n + 1]$ , cuya respuesta al impulso es  $h[n] = 2\delta[n] + 3\delta[n + 1]$ , entonces este sistema no sería causal, dado que el valor de desplazamiento,  $k$ , es negativo.

### 2.5.3. Estabilidad

Un sistema LTI es estable, si se cumple que:

$$\sum_n |h[n]| < \infty$$

Cuando la cantidad de impulsos de la respuesta al impulso es finita, la suma  $|h[n]|$  es finita, y por lo tanto, el sistema es estable. En caso contrario, será inestable.

#### 2.5.4. Invertibilidad

Un sistema LTI con respuesta al impulso  $h_1[n]$  es invertible, si y solo si, existe un sistema inverso con respuesta al impulso  $h_2[n]$ , tal que:

$$h_1[n] \otimes h_2[n] = \delta[n]$$

Por ejemplo, si el sistema  $y_1[n] = x[n - 1]$ , entonces, el sistema inverso es  $y_2[n] = x[n + 1]$ , dado que:

$$h_1[n] = \delta[n - 1]$$

$$h_2[n] = \delta[n + 1]$$

Y,

$$h_1[n] \otimes h_2[n] = \delta[n - 1] \otimes \delta[n + 1] = \delta[n]$$



## CAPÍTULO III

# Señal exponencial compleja y series de Fourier

El análisis en series de Fourier se enfoca en señales periódicas y utiliza como base la señal exponencial compleja. Para lo cual, en este capítulo, repasaremos las señales exponenciales, y conoceremos cómo expresar señales básicas (ej. senoidal, cuadrada periódica) en términos de la suma de exponenciales complejas relacionadas armónicamente.

Al finalizar el capítulo, deberás estar en capacidad de:

1. Explicar por qué la Serie de Fourier (SF) utiliza la señal exponencial compleja, y no la señal exponencial real.
2. Calcular la SF de señales senoidales o cosenoidales aplicando la ecuación de Euler y el concepto de armónica.
3. Escribir código en Python para dibujar señales senoidales o cosenoidales, así como sus coeficientes de la SF.
4. Explicar el fenómeno de Gibbs.
5. Explicar las condiciones de Dirichlet y su relación con la existencia de la Serie de Fourier.
6. Calcular la SF de señales periódicas aplicando la ecuación de análisis y/o propiedades de la SF.
7. Escribir código en Python para calcular y dibujar la SF de la suma de señales senoidales o cosenoidales.

### 3.1. SEÑAL EXPONENCIAL

La Serie de Fourier se basa en la señal exponencial compleja; por esta razón, iniciaremos este capítulo con un breve recordatorio de los diferentes tipos de señales exponenciales.

#### 3.1.1. Señal exponencial continua real

Esta señal se define matemáticamente como:

$$x(t) = Ae^{\alpha t} \text{ donde } A, \alpha \in \mathbb{R}$$

Por ejemplo, consideremos la señal exponencial  $x(t) = 2e^{t/2}$ ,

donde  $A = 2$ , y  $\alpha = 1/2$ .

Este tipo de señales exponenciales puede ser creciente o decreciente, dependiendo del valor de  $\alpha$ . Si  $\alpha < 0$ , la señal exponencial es decreciente; mientras que, si  $\alpha > 0$ , la señal exponencial es creciente.

Sin embargo, en ambos casos, este tipo de señales no es periódica, es decir, no se cumple que  $x(t) = x(t + T)$ .

A continuación, estudiaremos a la señal exponencial compleja.

#### 3.1.2. Señal exponencial continua compleja

Esta señal se representa de la siguiente forma:

$$x(t) = Ae^{j\omega t} \text{ donde } A \in \mathbb{R}$$

Por simplicidad, y para facilitar el desarrollo matemático que veremos a continuación, asumiremos que  $A = 1$  (sin pérdida de generalidad, ya que el valor de A solo escala la señal).

A diferencia del caso anterior, este tipo de señales exponenciales sí es periódica, lo que significa que se repite con un período fundamental  $T_0$ , el cual se obtiene a partir de la frecuencia angular fundamental  $\omega_0$ , así:

$$T = \frac{2\pi}{\omega_0}$$

El periodo fundamental,  $T$ , se expresa en unidades de segundos [s]; mientras que, la frecuencia angular fundamental  $\omega_0$  en [rad/s].

Ahora, vamos a relacionar la señal exponencial continua con la señal cosenoidal y senoidal. Para ello, utilizamos la fórmula de Euler:

$$e^{j\omega_0 t} = \cos(\omega_0 t) + j\text{sen}(\omega_0 t)$$

Donde se obtiene que:

$$\cos(\omega_0 t) = \frac{1}{2} \{e^{j\omega_0 t} + e^{-j\omega_0 t}\}$$

Y,

$$\text{sen}(\omega_0 t) = \frac{1}{2j} \{e^{j\omega_0 t} - e^{-j\omega_0 t}\}$$

De esta forma:

$$\cos(\omega_0 t) = \mathbb{R}\{e^{j\omega_0 t}\}$$

$$\text{sen}(\omega_0 t) = \mathbb{I}\{e^{j\omega_0 t}\}$$

Existen dos diferencias importantes entre la representación de la señal cosenoidal y la señal senoidal:

1. En el caso de la señal cosenoidal las señales exponenciales se suman; mientras que en la señal senoidal, se restan.
2. La amplitud de la señal cosenoidal es real; mientras que, en la señal senoidal es puramente imaginaria.

Ahora, vamos a revisar el siguiente concepto:

“Un conjunto de señales exponenciales complejas relacionadas armónicamente corresponde a exponenciales con frecuencias fundamentales múltiplos de  $\omega_0$ ”.

- **Conjunto**, significa que se trata de varias señales.
- **Señales exponenciales complejas**, corresponden a expresiones del tipo  $e^{j\omega_0 t}$ .
- **Relacionadas armónicamente** significa que sus frecuencias pueden expresarse como  $k\omega_0$ , donde  $k$  es un entero. En otras palabras, todas las frecuencias presentes en la señal son **múltiplos enteros de una misma frecuencia fundamental**.

**Ejemplo 21: manipulación de exponenciales complejas.**

Supongamos que tenemos la siguiente señal,

$$x(t) = e^{j2t} + e^{j3t}$$

Entonces, aplicamos los siguientes pasos:

1. Identificar las frecuencias angulares de cada término:

$$\omega_1 = 2, \quad \omega_2 = 3$$

2. Calculamos la frecuencia angular promedio:

$$\bar{\omega} = \frac{2 + 3}{2} = \frac{5}{2}$$

3. Factorizamos el término  $e^{j\bar{\omega}t}$ :

$$x(t) = e^{j2.5t} \{e^{-j0.5t} + e^{j0.5t}\}$$

4. Re-escribimos la expresión para identificar una señal cosenoidal

Nota aclaratoria: introducimos el factor  $2/2$  con el fin de completar la forma necesaria para aplicar directamente la identidad del coseno a partir de las exponenciales complejas.

$$x(t) = e^{j2.5t} \frac{2}{2} \{e^{-j0.5t} + e^{j0.5t}\}$$

Por lo tanto:

$$x(t) = 2e^{j2.5t} \cos(0.5t)$$

Para dibujar la magnitud de la señal, observamos que

$$|e^{j2.5t}| = 1,$$

Y, en consecuencia:

$$|x(t)| = 2|\cos(0.5t)|$$

Ahora, vamos a utilizar el siguiente código en Python para graficar

$|x(t)|$ .

Primero, graficamos la envolvente cosenoidal  $x(t)=\cos(0.5t)$ , asociada a la señal original.

```
import numpy as np
import matplotlib.pyplot as plt

# Definir el rango de tiempo
t = np.linspace(0, 8 * np.pi, 1000)

# Calcular la señal x(t)
x = np.cos(0.5 * t)

# Graficar la señal
plt.figure(figsize=(8, 5))
plt.plot(t, x, label=r'\cos(0.5t)\$')
plt.xlabel('t')
plt.ylabel('Amplitud')
plt.title('x(t) = cos(0.5t)')
plt.grid(True)
plt.legend()
plt.show()
```

Figura 64. Ejemplo 21: código en Python señal  $x(t) = \cos(0.5t)$ .

Obteniendo la siguiente figura:

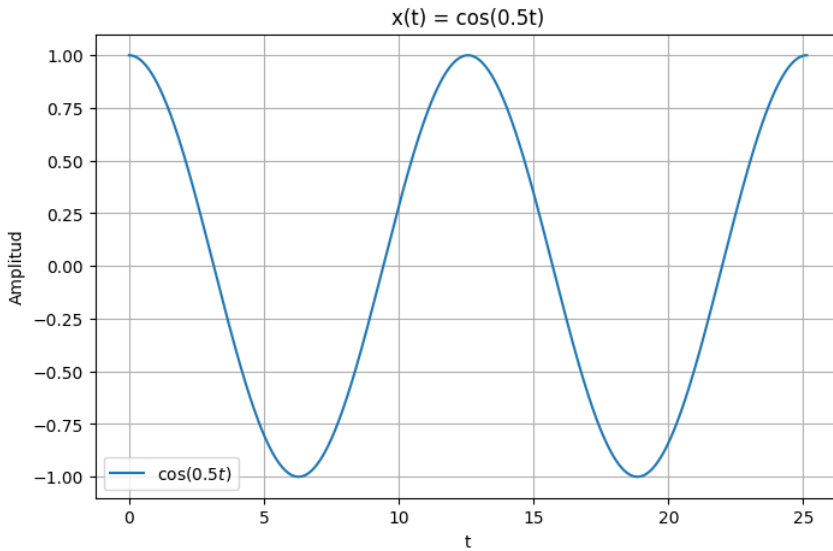


Figura 65. Ejemplo 21: gráfica de la señal  $x(t) = \cos(0.5t)$ .

Teniendo en cuenta que la frecuencia fundamental de esta señal es  $\omega_0 = 0.5 \text{ [rad/s]}$ , tenemos que el periodo, definido como  $T = 2\pi/(\omega_0)$  es igual a  $T = 2\pi/0.5 = 4\pi$ , y entonces, cada  $4\pi$  ( $\sim 12.56$ ) segundos, tenemos un ciclo de la señal. En el caso del código utilizado, al graficar de 0 a  $8\pi$ , tenemos dos ciclos de la señal.

Ahora, vamos a graficar  $|x(t)| = 2|\cos(0.5t)|$  con el siguiente código en Python:

```

# Calcular la magnitud de la señal
x = 2 * np.abs(np.cos(0.5 * t))

# Graficar la señal
plt.figure(figsize=(8, 5))
plt.plot(t, x, label=r'$\cos(0.5t)$')
plt.xlabel('t')
plt.ylabel('Amplitud')
plt.title('|x(t)| = 2|cos(0.5t)|')
plt.grid(True)
plt.legend()
plt.show()

```

Figura 66. Ejemplo 21: código en Python señal  $|x(t)| = 2|\cos(0.5t)|$ .  
Y obtenemos:

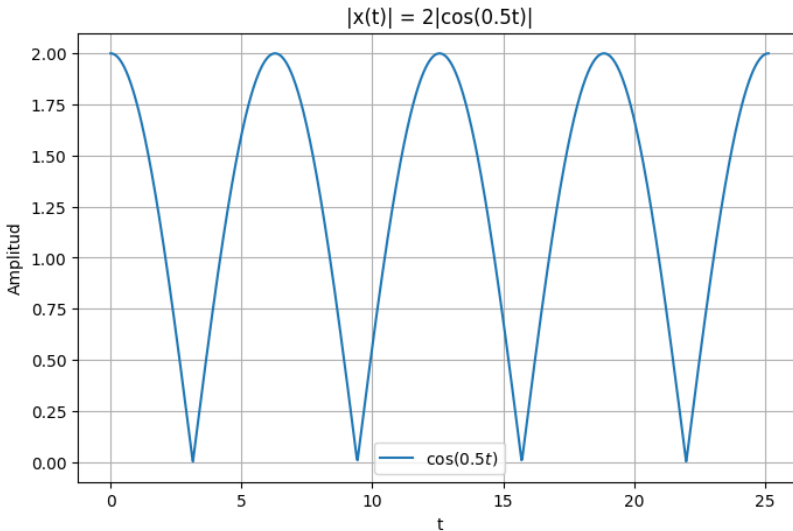


Figura 67. Ejemplo 21: gráfica de la señal  $|x(t)| = 2|\cos(0.5t)|$ .

Al comparar las dos figuras anteriores, nos damos cuenta que, el periodo de la segunda señal es la mitad del periodo de la primera señal, debido a que se obtuvo su valor absoluto, por lo que las amplitudes negativas ahora son positivas.

### 3.1.3. Forma general de la exponencial compleja

La forma general de una señal exponencial continua compleja es:

$$x(t) = Ae^{(r+j\omega_0)t}$$

donde,

$$A \in \mathbb{C}, \quad r \in \mathbb{R}, \quad \omega_0 \in \mathbb{R}$$

El término  $r$  determina si la señal crece o decrece en en tiempo:

- Si  $r > 0$ , la señal es creciente.
- Si  $r < 0$ , la señal es decreciente.

Sin embargo, en cualquiera de estos dos casos la señal no es periódica, debido a la presencia del factor exponencial real  $e^{rt}$ .

Por esta razón, cuando se utilizan exponenciales complejas para representar señales periódicas, se considera el caso particular en el que:

$$r = 0$$

Lo que conduce a la forma:

$$x(t) = Ae^{j\omega_0 t}$$

que corresponde a la exponencial compleja periódica estudiada en la sección anterior.

Nota aclaratoria:

Por esta razón, la exponencial compleja  $e^{j\omega t}$  constituye la **base natural para representar señales periódicas**, ya que es la única exponencial cuya magnitud permanece constante y cuya fase evoluciona de forma periódica en el tiempo.

### 3.1.4. Ejercicios propuestos

En los siguientes ejercicios se analiza la magnitud de señales formadas por la suma de exponenciales complejas. El propósito es observar cómo las

diferentes frecuencias angulares afectan la forma y el período de  $|x(t)|$  a partir de su representación gráfica.

Grafique  $|x(t)|$  de las siguientes señales:

1.  $x(t) = e^{j4\pi t} + e^{j8\pi t}$
2.  $x(t) = e^{j2\pi t} + e^{j8\pi t}$
3.  $x(t) = e^{j2\pi t} + e^{j4\pi t}$
4.  $x(t) = e^{j4\pi t} + e^{j16\pi t}$

### 3.2. SERIES DE FOURIER DE SEÑALES SENOIDALES

Antes de introducir formalmente la ecuación de la Serie de Fourier, analizaremos de manera intuitiva cómo una señal periódica puede representarse como la suma de senos y cosenos, o de forma equivalente, como la suma de exponenciales complejas. A partir de ejemplos sencillos, se identificará qué frecuencias están presentes en la señal y cómo cada una contribuye a su forma, utilizando la fórmula de Euler como herramienta de apoyo.

**Ejemplo 22: Series de Fourier de una señal senoidal: enfoque intuitivo con armónicas consecutivas.**

Sea  $x(t) = 1 + \text{sen}(4\pi t) + 2 \cos(4\pi t) + \cos\left(8\pi t + \frac{\pi}{4}\right)$ . Encontrar los  $\alpha_k$  y dibujar  $|a_k|$ .

1. Determinar la frecuencia fundamental e identificar cuáles armónicas están presentes, y si existe o no el coeficiente  $\alpha_0$ :

$$\text{M.C.D. } (4\pi, 4\pi, 8\pi) = 4\pi$$

Donde M.C.D. representa el Máximo Común Divisor.

Entonces, la frecuencia fundamental es  $\omega_0 = 4\pi$  y se tendría la

primera ( $a_1, a_{-1}$ ) y segunda armónica ( $a_2, a_{-2}$ ). Adicionalmente, como existe un nivel de D.C. también se tiene  $a_0$ .

Nota aclaratoria:

La frecuencia fundamental corresponde al **máximo común divisor de las frecuencias presentes en la señal**, ya que todas las componentes deben poder expresarse como múltiplos enteros de dicha frecuencia.

2. Aplicar Euler:

Recordando que  $\cos(\omega_0 t) = \frac{1}{2} \{e^{j\omega_0 t} + e^{-j\omega_0 t}\}$  y que

$\sen(\omega_0 t) = \frac{1}{2j} \{e^{j\omega_0 t} - e^{-j\omega_0 t}\}$ , entonces se tiene que:

$$x(t) = 1 + \frac{1}{2j} \{e^{j4\pi t} - e^{-j4\pi t}\} + \{e^{j4\pi t} + e^{-j4\pi t}\} + \frac{1}{2} \{e^{j(8\pi t + \frac{\pi}{4})} + e^{-j(8\pi t + \frac{\pi}{4})}\}$$

Resolviendo, se llega a:

$$x(t) = 1 + \frac{e^{j4\pi t}}{2j} - \frac{e^{-j4\pi t}}{2j} + e^{j4\pi t} + e^{-j4\pi t} + \frac{e^{j8\pi t} * e^{j\frac{\pi}{4}}}{2} + \frac{e^{-j8\pi t} * e^{-j\frac{\pi}{4}}}{2}$$

Ahora, se agrupan por exponenciales, y se obtiene que:

$$x(t) = 1 + e^{j4\pi t} \left\{1 + \frac{1}{2j}\right\} + e^{-j4\pi t} \left\{1 - \frac{1}{2j}\right\} + e^{j8\pi t} \left\{\frac{e^{j\frac{\pi}{4}}}{2}\right\} + e^{-j8\pi t} \left\{\frac{e^{-j\frac{\pi}{4}}}{2}\right\}$$

3. Identificar los coeficientes de la SF:

Recordando que de forma general  $x(t) = \sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t}$ , y que en nuestro caso solamente existe  $a_0, a_1, a_{-1}, a_2, a_{-2}$ , entonces la señal es de la forma (haciendo  $\omega_0 = 4\pi$ ):

$$x(t) = a_0 + a_1 e^{j4\pi t} + a_{-1} e^{-j4\pi t} + a_2 e^{j8\pi t} + a_{-2} e^{-j8\pi t}$$

Y al comparar término a término con lo obtenido en el paso 2, entonces:

$$a_0 = 1$$

$$a_1 = 1 + \frac{1}{2j}$$

$$a_{-1} = 1 - \frac{1}{2j}$$

$$a_2 = \frac{e^{j\frac{\pi}{4}}}{2}$$

$$a_{-2} = \frac{e^{-j\frac{\pi}{4}}}{2}$$

4. Encontrar la magnitud de los coeficientes de la SF:

$$|a_0| = |1| = 1$$

$$|a_1| = |a_{-1}| = \left| 1 + \frac{1}{2j} \right| = \left| 1 - \frac{j}{2} \right| = \sqrt{1^2 + 0.5^2} = 1.118$$

$$|a_2| = |a_{-2}| = \left| \frac{e^{j\frac{\pi}{4}}}{2} \right| = \left| \frac{\cos(\frac{\pi}{4}) + j\text{sen}(\frac{\pi}{4})}{2} \right| = \left| \frac{0.707 + j0.707}{2} \right| = \frac{\sqrt{0.707^2 + 0.707^2}}{2} = 0.5$$

5. Dibujar la magnitud de los coeficientes de la SF:

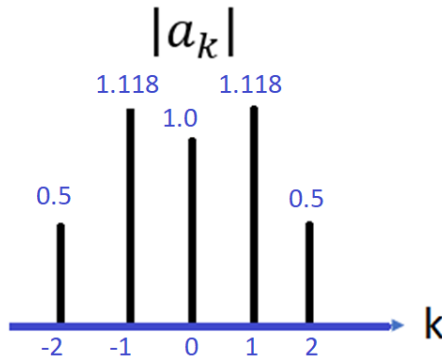


Figura 68. Magnitud de los coeficientes de la SF, ejemplo 22.

Cada valor de  $|a_k|$  representa la magnitud de una armónica (en este caso armónica 1, armónica 2 y nivel de d.c.) de la señal.

**Ejemplo 23: Series de Fourier de una señal senoidal: enfoque intuitivo con armónicas no consecutivas.**

Sea  $x(t) = 1 + 2\text{sen}(\pi t) - \cos(\pi t) + 3\cos\left(3\pi t + \frac{\pi}{2}\right)$ . Encontrar los  $a_k$  y dibujar  $|a_k|$ .

- 1.
1. Determinar la frecuencia fundamental e identificar cuáles armónicas están presentes, y si existe o no el coeficiente  $a_0$ :

$$\text{M.C.D}(\pi, \pi, 3\pi) = \pi$$

Donde M.C.D. representa el Máximo Común Divisor.

Entonces, la frecuencia fundamental es  $\omega_0 = \pi$  y se tendría la primera ( $a_1, a_{-1}$ ) y tercera armónica ( $a_3, a_{-3}$ ). Adicionalmente, como existe un nivel de D.C. también se tiene  $a_0$ .

2. Aplicar Euler:

Recordando que  $\cos(\omega_0 t) = \frac{1}{2}\{e^{j\omega_0 t} + e^{-j\omega_0 t}\}$

y que  $\text{sen}(\omega_0 t) = \frac{1}{2i}\{e^{j\omega_0 t} - e^{-j\omega_0 t}\}$ , entonces se tiene que:

$$x(t) = 1 + \frac{1}{j}\{e^{j\pi t} - e^{-j\pi t}\} - \frac{1}{2}\{e^{j\pi t} + e^{-j\pi t}\} + \frac{3}{2}\{e^{j(3\pi t + \frac{\pi}{2})} + e^{-j(3\pi t + \frac{\pi}{2})}\}$$

Resolviendo, se llega a:

$$x(t) = 1 + \frac{e^{j\pi t}}{j} - \frac{e^{-j\pi t}}{j} - \frac{e^{j\pi t}}{2} - \frac{e^{-j\pi t}}{2} + \frac{3e^{j3\pi t} * e^{j\frac{\pi}{2}}}{2} + \frac{3e^{-j3\pi t} * e^{-j\frac{\pi}{2}}}{2}$$

Ahora, se agrupan por exponenciales, y se obtiene que:

$$x(t) = 1 + e^{j\pi t} \left\{ -\frac{1}{2} + \frac{1}{j} \right\} + e^{-j\pi t} \left\{ -\frac{1}{2} - \frac{1}{j} \right\} + e^{j3\pi t} \left\{ \frac{3e^{j\frac{\pi}{2}}}{2} \right\} + e^{-j3\pi t} \left\{ \frac{3e^{-j\frac{\pi}{2}}}{2} \right\}$$

3. Identificar los coeficientes de la SF:

Recordando que de forma general  $x(t) = \sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t}$ , y que en nuestro caso solamente existe  $a_0, a_1, a_{-1}, a_3, a_{-3}$ , entonces la señal es de la forma (haciendo  $\omega_0 = \pi$ ):

$$x(t) = a_0 + a_1 e^{j\pi t} + a_{-1} e^{-j\pi t} + a_3 e^{j3\pi t} + a_{-3} e^{-j3\pi t}$$

Y al comparar término a término con lo obtenido en el paso 2, entonces:

$$a_0 = 1$$

$$a_1 = -\frac{1}{2} + \frac{1}{j}$$

$$a_{-1} = -\frac{1}{2} - \frac{1}{j}$$

$$a_3 = \frac{3e^{j\frac{\pi}{2}}}{2}$$

$$a_{-3} = \frac{3e^{-j\frac{\pi}{2}}}{2}$$

4. Encontrar la magnitud de los coeficientes de la SF:

$$|a_0| = |1| = 1$$

$$|a_1| = |a_{-1}| = \left| -\frac{1}{2} + \frac{1}{j} \right| = \left| -\frac{1}{2} - j \right| = \sqrt{0.5^2 + 1^2} = 1.118$$

$$|a_3| = |a_{-3}| = \left| \frac{3e^{j\frac{\pi}{2}}}{2} \right| = \left| 3 \frac{\cos(\frac{\pi}{2}) + j\sin(\frac{\pi}{2})}{2} \right| = \left| 3 \frac{0+j}{2} \right| = 3 \frac{\sqrt{0^2+1^2}}{2} = 1.5$$

5. Dibujar la magnitud de los coeficientes de la SF:

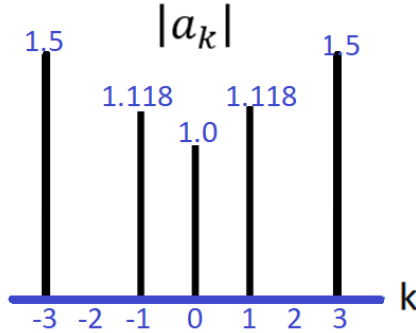


Figura 69. Magnitud de los coeficientes de la SF, Ejemplo 23.

Cada valor de  $|a_k|$  representa la magnitud de una armónica (en este caso armónica 1, armónica 3 y nivel de d.c.) de la señal.

Nota aclaratoria:  
 Tener en cuenta que en este ejercicio no existe la segunda armónica, dado que ninguna de las señales seno o coseno tienen frecuencia  $2\omega_0$ .

### 3.3. SERIES DE FOURIER DE SEÑALES PERIÓDICAS

A continuación, se presentan dos ecuaciones que nos permitirán, por una parte, representar una señal periódica en términos de los coeficientes de la SF (SF); y, por otra parte, encontrar los coeficientes de la SF de una señal periódica en tiempo continuo.

Partimos de la **ecuación de síntesis**, la cual se presenta a continuación:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t} = \sum_{k=-\infty}^{\infty} a_k e^{jk\left(\frac{2\pi}{T}\right)t}$$

donde cada término  $a_k$  representa una componente armónica de la señal, ubicada en múltiplos de la frecuencia fundamental  $\omega_0$ .

Con esta ecuación, representamos una señal periódica a partir de la suma de exponenciales complejas relacionadas armónicamente. Es importante tener en cuenta que todas las exponenciales tienen como frecuencia un múltiplo entero (denominado  $k$ ) de la frecuencia fundamental de la señal. Cuando la señal es real, las armónicas aparecen en pares conjugados; es decir, si existe  $k$ , también existe  $-k$ . Esta propiedad garantiza que la señal resultante en el dominio del tiempo sea real.

Por otra parte, tenemos la **ecuación de análisis** que nos permite encontrar los coeficientes de la SF de una señal periódica en tiempo continuo, así

$$a_k = \frac{1}{T} \int_0^T x(t) e^{-jk\omega_0 t} dt = \frac{1}{T} \int_0^T x(t) e^{-jk\left(\frac{2\pi}{T}\right)t} dt$$

Los coeficientes de Fourier equivalen a la porción de la señal representada por cada frecuencia. El coeficiente  $a_0$  corresponde al nivel de d.c. de la señal.

#### Ejemplo 24. Serie de Fourier de una señal cuadrada periódica

A continuación, vamos a aplicar la ecuación de análisis para encontrar los SF de una señal periódica que no es de tipo senoidal. Específicamente, para la señal “cuadrada periódica”, la cual la definimos así:

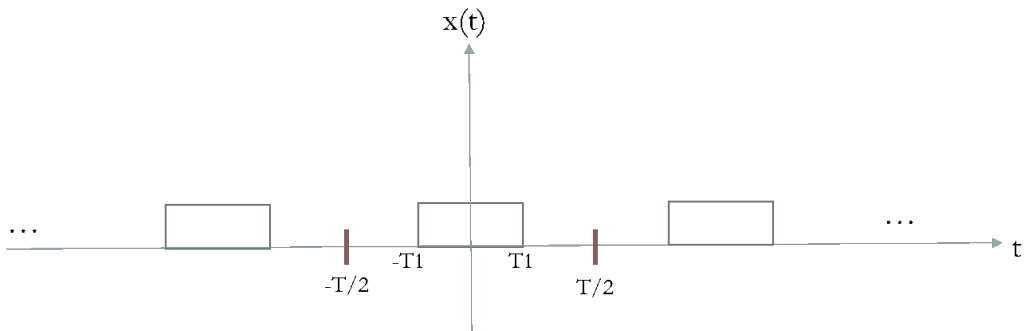


Figura 70. Señal cuadrada periódica.

Asumimos que la amplitud de la señal es 1, es decir que,

$$x(t) = \begin{cases} 1 & |t| < T_1 \\ 0 & T_1 < |t| < \frac{T}{2} \end{cases}$$

Esta definición corresponde a un periodo fundamental de la señal.

Para el coeficiente  $a_0$ , hacemos  $k = 0$  en la ecuación de análisis, y tenemos que:

$$a_0 = \frac{1}{T} \int_{-T_1}^{T_1} dt = \frac{2 * T_1}{T}$$

Para los demás valores de  $k$ , tenemos que:

$$a_k = \frac{1}{T} \int_{-T_1}^{T_1} e^{-jk\omega_0 t} dt = -\frac{1}{jk\omega_0 T} e^{-jk\omega_0 t} \Big|_{-T_1}^{T_1}$$

$$a_k = \frac{2}{k\omega_0 T} \left[ \frac{e^{jk\omega_0 T_1} - e^{-jk\omega_0 T_1}}{2j} \right] = \frac{2 * \text{sen}(k\omega_0 T_1)}{k\omega_0 T} = \frac{\text{sen}(k\omega_0 T_1)}{k * \pi}$$

Esta expresión muestra que la envolvente de los coeficientes tiene forma tipo sinc (a medida que  $k$  se hace más grande, el coeficiente se hace más pequeño).

Vamos a continuación a graficar los coeficientes de la SF de una señal cuadrada, con los siguientes valores:

1. Amplitud de la señal = 1
2.  $T_1 = 0.01$ ,  $T = 10 * T_1$
3. El rango de la gráfica va desde  $k = -15$  hasta  $k = 15$ .

Utilizaremos el siguiente código en Python:

```

import numpy as np
import matplotlib.pyplot as plt
num=20
coef = np.arange(-num,num+1)
a = np.zeros((len(coef),1))

T1=0.01
T = 10*T1
w0=2*np.pi/T
a0=2*T1/T
print("el valor de D.C. de la señal, es:", a0)

count=0
for k in range(-num,num+1):
    if k==0:
        a[count]=a0
    else:
        a[count]=np.sin(k*w0*T1)/(k*np.pi)
        count=count+1

plt.rcParams["figure.figsize"] = (8,4)
plt.stem(coef, a)

```

Figura 71. Ejemplo 24: código en Python para calcular la SF de una señal cuadrada periódica.

Del código anterior, vemos que:

- La variable “coef” corresponde a valores enteros en el rango [-20 20], creada a partir de la instrucción *np.arange*, con incremento (por defecto) de 1. Esta variable corresponde a “k” de la ecuación de análisis.
- La variable “a” corresponde a los coeficientes de la SF. Se crea con *np.zeros*, de la misma longitud que “coef”. Posteriormente, con el ciclo for, a medida que se obtienen los coeficientes, se reemplaza el cero por el valor correspondiente.
- La frecuencia fundamental  $\omega_0$  la obtenemos a partir del periodo de la señal, T, con la ecuación  $\omega_0 = 2\pi/T$ .

- El nivel de d.c. de la señal, es decir, el coeficiente  $a_0$  se calcula con la ecuación  $a_0 = 2T_1/T$ ; mientras que, para los demás valores de  $k$ , se utiliza la ecuación sinc que obtuvimos previamente.

Con este código obtenemos la siguiente gráfica:

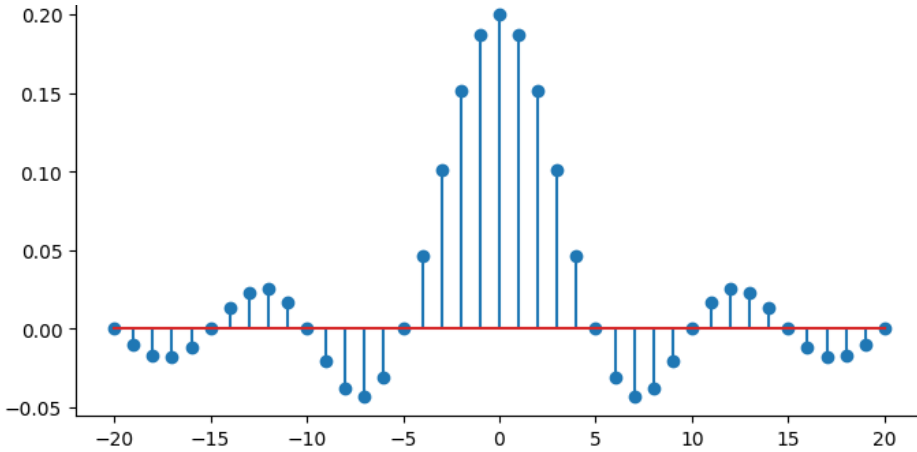


Figura 72. Ejemplo 24: gráfica de los coeficientes de la SF de una señal

cuadrada periódica para  $T_1 = 0.01$ ,  $T = 10 * T_1$ ,  $k = [-20 \ 20]$ .

Para este ejercicio el nivel de d.c. de la señal es 0.2, dado que  $2T_1/T = 2T_1/(10T_1) = 0.2$ . Adicionalmente, apreciamos que la “envolvente” de los coeficientes de la SF es de tipo sinc; a medida que se aleja de  $k=0$ , los coeficientes de la SF disminuyen en amplitud.

Vamos ahora a graficar los coeficientes de la SF, pero de la señal cuadrada periódica con  $T_1 = 0.01$  y  $T = 20 * T_1$ .

Y obtenemos:

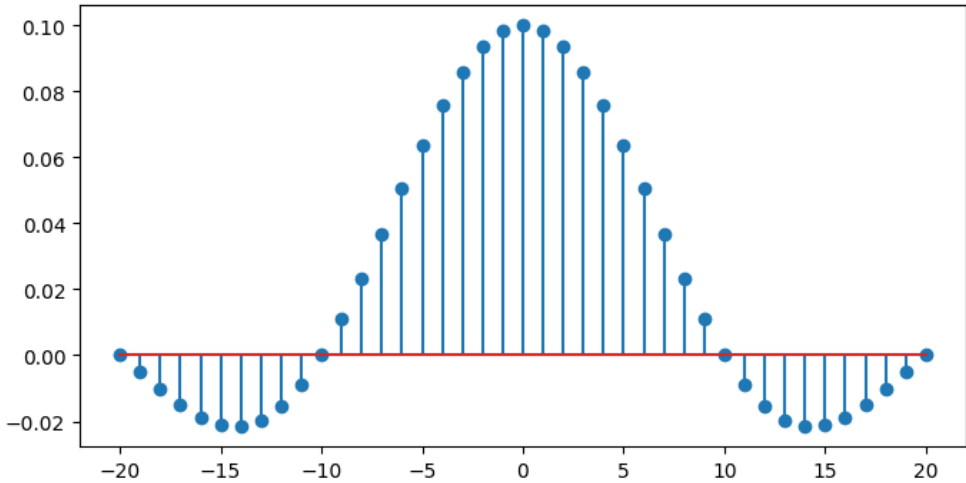


Figura 73. Ejemplo 24: gráfica de los coeficientes de la SF de una señal cuadrada periódica para  $T_1 = 0.01$ ,  $T = 20 * T_1$ ,  $k = [-20 \ 20]$ .

En este caso, el nivel de d.c. de la señal es  $2T_1/T = 2T_1/(20T_1) = 0.1$ . Adicionalmente, podemos observar que, la cantidad de “cruces por cero” ha disminuido. El primer cruce por cero lo obtenemos en el coeficiente  $k = 10$  (y su contraparte) y el segundo cruce por cero en  $k = 20$  (y su contraparte).

Nota aclaratoria:

De forma general, los cruces por cero de los coeficientes de la SF se presentan cuando se cumple la condición  $k\omega_0 T_1 = n\pi$ , lo que implica que dichos cruces dependen directamente de la relación entre el período de la señal  $T$  y el ancho del pulso  $T_1$ . En este caso, como  $T = MT_1$  con  $M = 20$ , los cruces por cero aparecen cada  $M/2$  coeficientes.

Pero, ¿qué ocurre si  $M$  es un valor impar? Vamos a ver su comportamiento con un ejemplo.

$$T = 9 * T_1$$

Obtenemos la siguiente gráfica de coeficientes de la SF:

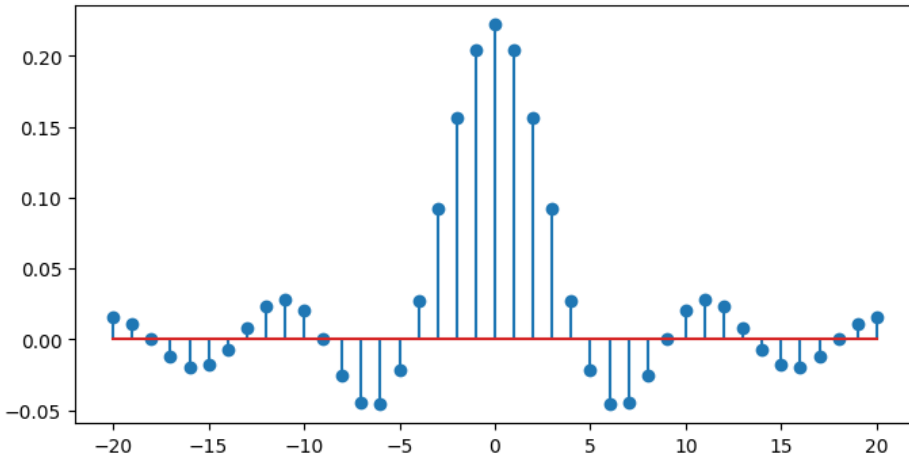


Figura 74. Coeficientes de la SF de la señal cuadrada periódica,

$$T_1 = 0.01, T = 9 * T_1, k = [-20 \ 20].$$

El primer cruce por cero debería ocurrir en  $k = 4.5$ . Sin embargo, recordemos que el índice armónico  $k$  solo puede tomar valores enteros. Por lo tanto, el cruce por cero no aparece exactamente en un coeficiente, sino que se observa como un cambio de signo entre coeficientes consecutivos, específicamente entre el coeficiente  $a_4$  (positivo) y el coeficiente  $a_5$  (negativo).

El segundo cruce por cero debería ocurrir en  $k = 9$ . Como este valor sí es entero, se obtiene directamente que  $a_9=0$ .

Entonces, siguiendo la misma explicación anterior, ¿qué esperamos que ocurra si  $M = 5$ ? El primer cambio de polaridad se presenta entre el coeficiente  $a_2$  (coeficiente positivo) y  $a_3$  (coeficiente negativo). Y el primer cruce

por cero lo tendremos en  $a_5$ .

Hacemos en el código:

```
T = 5*T1
```

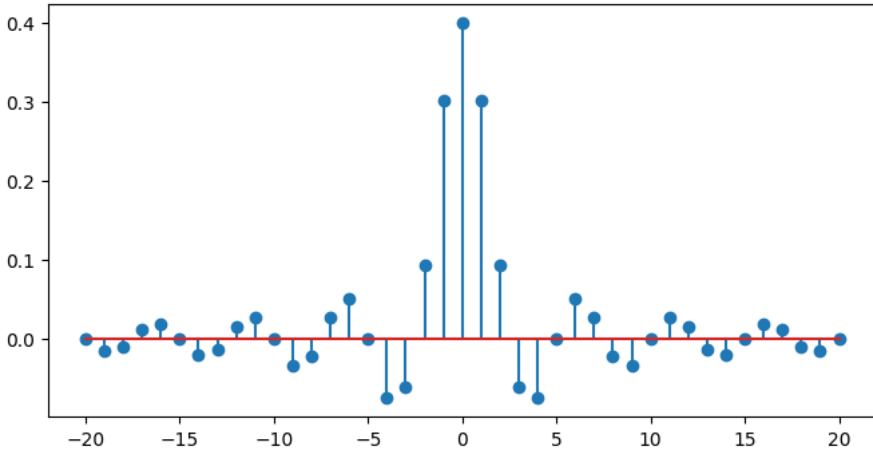


Figura 75. Coeficientes de la SF de la señal cuadrada periódica,  $T_1 = 0.01$ ,

$$T = 5 * T_1, k = [-20 \ 20].$$

El resultado obtenido coincide con lo esperado. El nivel de d.c es de  $2T_1/T = 2T_1/(5T_1) = 0.4$ .

### Ejemplo 25. Serie de Fourier de un tren de impulsos

Una de las señales más importantes en el análisis de señales y en sistemas de comunicaciones es el tren de impulsos. Aunque a primera vista puede parecer una construcción abstracta, esta señal permite modelar procesos fundamentales como el muestreo de señales continuas.

El tren de impulsos se define como:

$$x(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Donde  $T$  corresponde al periodo de la señal, es decir, al espaciamiento entre impulsos consecutivos. Todos los impulsos están separados entre sí por el mismo valor, que corresponde a  $T$ .

Los coeficientes de la SF de esta señal son:

$$a_k = \frac{1}{T} \quad -\infty < k < \infty$$

Es decir, el número de coeficientes de la SF es infinito, definido para todos los valores de  $k$ .

Vamos a utilizar el siguiente código en Python para dibujar una porción del tren de impulsos

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros
T = 10
n = np.arange(-5, 6) # 11 impulsos: desde -5T hasta 5T
t_imp = n * T        # posiciones de los impulsos

# Gráfica
plt.figure()
plt.stem(t_imp, np.ones_like(t_imp))

plt.title('Tren de impulsos (T = 10)')
plt.xlabel('t')
plt.ylabel('Amplitud')
plt.grid()

plt.show()
```

Figura 76. Código en Python Ejemplo 25. Señal tren de impulsos, para  $T=10$ .

Obteniendo la siguiente señal

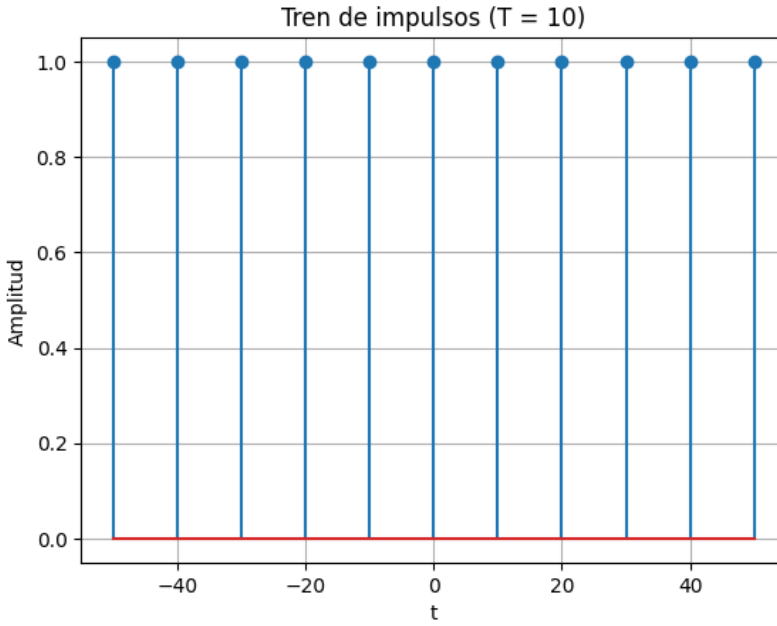


Figura 77. Señal tren de impulsos desde  $-5T$  hasta  $5T$ , para  $T=10$  [s].

Posteriormente, escribimos el código en Python para graficar una porción de los coeficientes de la SF, desde  $k=-5$  hasta  $k=5$ , así:

```
# Valores de k
k = np.arange(-5, 6)

# Coeficientes ak
ak = (1/T) * np.ones_like(k)

# Gráfica
plt.figure()
plt.stem(k, ak)

plt.title('Coeficientes de la SF del tren de impulsos, con T=10')
plt.xlabel('k')
plt.ylabel('|a_k|')
plt.grid()

plt.show()
```

Figura 78. Código en Python Ejemplo 25. Coeficientes de la SF de la señal tren de impulsos, para  $T=10$ .

Y obtenemos la siguiente gráfica:

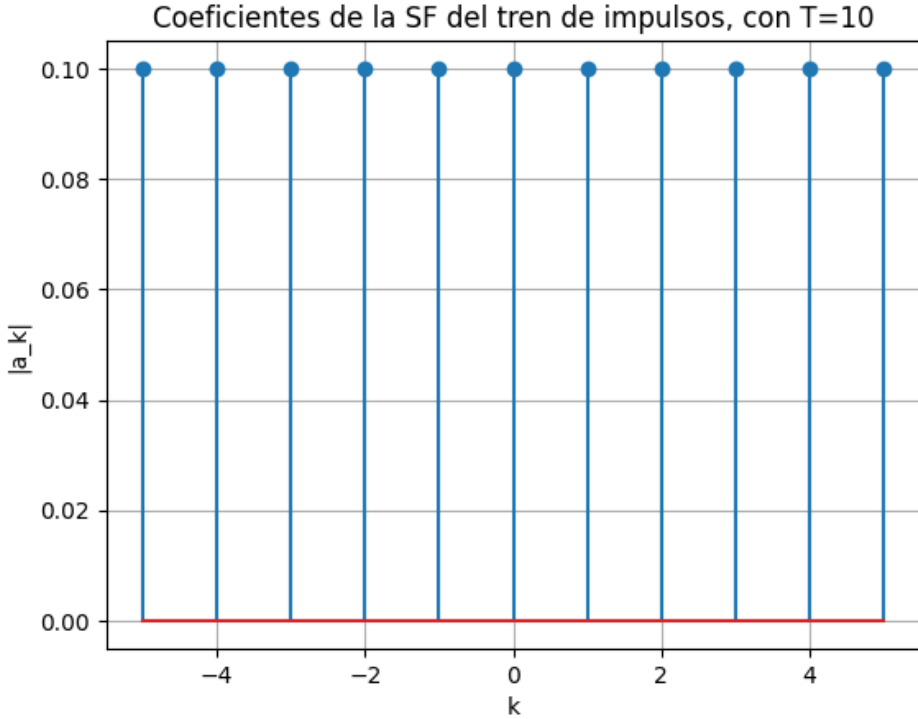


Figura 79. Coeficientes de la SF de la señal tren de impulsos, para T=10 [s].

Cada valor de  $a_k$  representa la amplitud de una armónica (en este caso infinitas) de la señal.

### 3.4. FENÓMENO DE GIBBS

El fenómeno de Gibbs aparece en señales que presentan discontinuidades, como la señal cuadrada periódica, y se manifiesta cuando la señal se reconstruye utilizando un número finito de coeficientes de la SF. En este caso, aparecen oscilaciones o “rizos” alrededor de las regiones donde se presentan las discontinuidades. La amplitud de estas oscilaciones es aproximadamente del **9 % del salto en la discontinuidad**, independientemente del número de términos utilizados en la reconstrucción.

Por ejemplo, para una señal cuadrada de amplitud 1, los rizados alcanzan valores cercanos a 0.09. Este fenómeno ocurre independientemente del número de términos de la Serie de Fourier utilizados en la reconstrucción de la señal  $x(t)$ .

### Ejemplo 26. Fenómeno de Gibbs señal cuadrada

Para ilustrar este fenómeno, vamos a reconstruir una señal cuadrada periódica con un número finito de parejas de coeficientes (5, 10, 50) incluido el nivel de d.c de la señal, utilizando el siguiente código en Python:

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros de la señal
T = 1          # Periodo
T1 = T / 4    # Ancho del pulso
w0 = 2 * np.pi / T  # Frecuencia fundamental

# Eje de tiempo
t = np.linspace(-T, T, 2000)

# Señal cuadrada ideal
x = np.where(np.abs(t) < T1, 1, 0)

# Función para reconstrucción con N armónicas
def fourier_reconstruction(t, N):
    x_rec = np.zeros_like(t, dtype=float)

    # Coeficiente DC
    a0 = 2 * T1 / T
    x_rec += a0

    # Armónicas
    for k in range(1, N + 1):
        ak = np.sin(k * w0 * T1) / (k * np.pi)
        x_rec += 2 * ak * np.cos(k * w0 * t)

    return x_rec

# Número de armónicas a usar
N_values = [5, 10, 50]

# Gráfica
plt.figure(figsize=(9, 5))
plt.plot(t, x, 'k', linewidth=2, label='Señal cuadrada ideal')

for N in N_values:
    plt.plot(t, fourier_reconstruction(t, N),
             label=f'Reconstrucción con N = {N}')

plt.xlabel('t')
plt.ylabel('Amplitud')
plt.title('Fenómeno de Gibbs en la reconstrucción de una señal cuadrada')
plt.legend()
plt.grid(True)
plt.show()
```

Figura 80. Código en Python del Ejemplo 25,  $T_1 = T/4$ .

Obteniendo las siguientes señales reconstruidas:

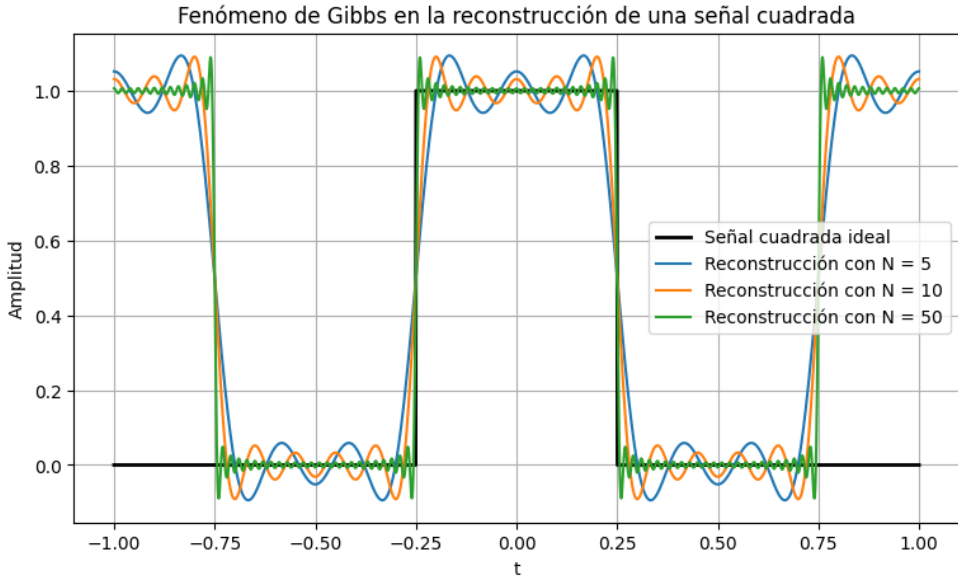


Figura 81. Fenómeno de Gibbs en la reconstrucción de una señal cuadrada periódica: (azul)  $-5 \leq k \leq 5$ , (naranja)  $-10 \leq k \leq 10$ , verde ( $-50 \leq k \leq 50$ ), (negra) ideal.

En la figura anterior se observa el fenómeno de Gibbs al reconstruir una señal cuadrada periódica utilizando un número finito de coeficientes de la SF.

A medida que aumenta el número de armónicas, la reconstrucción mejora significativamente en las regiones continuas. No obstante, en las proximidades de las discontinuidades persisten oscilaciones cuya amplitud no tiende a cero, sino que converge a un valor constante cercano al **9 % del salto de la señal**. Aunque el ancho de la región donde aparecen los rizados se reduce al aumentar el número de armónicas, su amplitud máxima no desaparece.

Este fenómeno ocurre **incluso si se aumenta indefinidamente el número de armónicas utilizadas en la reconstrucción**, lo cual constituye una de las características más distintivas del fenómeno de Gibbs.

Nota de énfasis:

Este fenómeno no desaparece al aumentar el número de armónicas, sino que se hace más localizado alrededor de la discontinuidad

### 3.5. CONDICIONES SUFICIENTES PARA LA EXISTENCIA DE LA SERIE DE FOURIER

Aunque una amplia variedad de señales periódicas admite representación en Serie de Fourier, no todas admiten esta representación en el sentido clásico.

De manera general, una condición suficiente para que una señal periódica tenga representación en Serie de Fourier es que sea cuadráticamente integrable en un periodo, es decir:

$$\int_0^T |x(t)|^2 dt < \infty$$

A partir de esta condición general, se derivan un conjunto de condiciones más específicas, conocidas como las **Condiciones de Dirichlet**, las cuales garantizan la existencia de la Serie de Fourier en el sentido clásico.

- Condición 1:  $x(t)$  debe ser absolutamente integrable sobre cualquier periodo.

Es decir:

$$\int_0^T |x(t)| dt < \infty$$

Si se cumple con esta condición, se tendrá que  $|a_k| < \infty$ .

Como ejemplo, una señal que NO cumple con esta condición es

$x(t) = \frac{1}{t}$ , periódica con periodo  $T$ . Dado que en  $t = 0, T, 2T, \dots, \infty$  tendrá discontinuidades, generando que el área bajo la curva no sea finita.

- Condición 2: en cualquier rango finito de tiempo, la cantidad de máximos y mínimos de  $x(t)$  debe ser finita.

Como ejemplo, una señal que NO cumple con esta condición es una

señal del tipo  $x(t) = \text{sen}\left(\frac{1}{t}\right)$ , definida en un intervalo finito, ya que presenta un número infinito de oscilaciones en un intervalo acotado de tiempo.

- Condición 3: en cualquier rango finito de tiempo, la cantidad de discontinuidades debe ser finita.

Como ejemplo, una señal que NO cumple con esta condición es una señal que divide su amplitud a la mitad, en la mitad del tiempo restante,

y esto lo repite desde  $t = 0$  hasta  $t = T$ , periódica con periodo igual a  $T$ .

Nota aclaratoria:

La condición cuadrática garantiza convergencia en el sentido de energía, mientras que las condiciones de Dirichlet permiten asegurar convergencia puntual en los puntos de continuidad de la señal.

### 3.6. PROPIEDADES DE LA SF

A continuación, se presentan las propiedades de la SF.

#### 3.6.1. Linealidad:

Sea  $x_1(t)$ ,  $x_2(t)$ , señales periódicas con coeficientes de la SF:

$$x_1(t) \xrightarrow{SF} a_k$$

$$x_2(t) \xrightarrow{SF} b_k$$

Si estas señales se suman, y a su vez cada una está multiplicada en amplitud por un escalar A y B, de la forma:

$$x_3(t) = Ax_1(t) + Bx_2(t)$$

Entonces, los coeficientes de la SF de la nueva señal se calculan de acuerdo con:

$$x_3(t) \xrightarrow{SF} Aa_k + Bb_k$$

Nota: para poder identificar correctamente los armónicos de la señal  $x_3(t)$  es necesario determinar la frecuencia fundamental de la señal, a través de la operación de MCD (máximo común divisor) entre las frecuencias de cada uno de sus términos.

### 3.6.2. Desplazamiento en el dominio del tiempo

Sea  $x_1(t)$  una señal periódica con coeficientes de la SF:

$$x_1(t) \xrightarrow{SF} a_k$$

Si esta señal se desplaza en el tiempo,

$$x_2(t) = x_1(t - t_0)$$

Entonces, los coeficientes de la SF de la nueva señal se obtienen como:

$$x_2(t) \xrightarrow{SF} a_k e^{-jk\omega_0 t_0}$$

Es decir, la magnitud de los coeficientes originales no se afecta, solamente su fase.

### 3.6.3. Inversión en el dominio del tiempo

Sea  $x_1(t)$  una señal periódica con coeficientes de la SF:

$$x_1(t) \xrightarrow{SF} a_k$$

Si esta señal se invierte en el tiempo:

$$x_2(t) = x_1(-t)$$

Entonces, los coeficientes de la SF de la nueva señal se obtienen como:

$$x_2(t) \xrightarrow{SF} a_{-k}$$

Es decir, al invertir una señal, también se invierten sus coeficientes de la SF.

De esta propiedad se tiene que:

- Si  $x(t)$  es par, entonces sus coeficientes son también par.
- Si  $x(t)$  es impar, entonces sus coeficientes son también impar.

### 3.6.4. Multiplicación en el dominio del tiempo

Sea  $x_1(t)$  y  $x_2(t)$  señales periódicas con coeficientes de la SF:

$$x_1(t) \xrightarrow{SF} a_k$$

$$x_2(t) \xrightarrow{SF} b_k$$

Si estas señales se multiplican, así:

$$x_3(t) = x_1(t) \cdot x_2(t)$$

Entonces, los coeficientes de la SF de la nueva señal se obtienen como:

$$x_3(t) \xrightarrow{SF} a_k \otimes b_k$$

Donde  $\otimes$  es el operador de convolución. Es decir, se realiza la convolución continua entre los espectros de ambas señales.

### Relación de Parseval

Nos sirve para calcular la potencia promedio de la señal, tanto en el dominio del tiempo, como a partir de sus coeficientes de la SF, por medio de la siguiente ecuación:

$$\frac{1}{T} \int_T |x(t)|^2 dt = \sum_{k=-\infty}^{\infty} |a_k|^2$$

En muchas ocasiones es más sencillo aplicar la parte derecho de la ecuación que realizar la integral del cuadrado de la señal (ej. si  $x(t)$  es una señal cosenoidal), por lo que la relación de Parseval es de gran utilidad.

### 3.7. EJEMPLOS DE SF DE SEÑALES PERIÓDICAS

A continuación, se presentan algunos ejemplos de aplicación de la Serie de Fourier y sus propiedades.

#### Ejemplo 27. Reconstruir la señal $x(t)$ a partir de los coeficientes de la SF

Escriba la señal  $x(t)$  en términos de señales senoidales o cosenoidales,

sabiendo que  $\omega_0 = 200\pi \left[ \frac{\text{rad}}{\text{s}} \right]$ , y los coeficientes de la SF son:

$$a_0 = 5 \qquad a_1 = 2 \qquad a_{-1} = 2 \qquad a_3 = j \qquad a_{-3} = -j$$

**Solución:**

Nuestra señal  $x(t)$  tendrá un nivel de d.c de 5, la primera armónica correspondiente a una señal cosenoidal (dado que  $a_1$  y  $a_{-1}$  son enteros), y la tercera armónica correspondiente a una señal senoidal (dado que  $a_3$  y  $a_{-3}$  son estrictamente imaginarios).

Aplicando la ecuación de síntesis, tenemos que:

$$x(t) = 5 + 2\{e^{j\omega_0 t} + e^{-j\omega_0 t}\} + j\{e^{j3\omega_0 t} - e^{-j3\omega_0 t}\}$$

O de forma equivalente,

$$x(t) = 5 + 2\{e^{j\omega_0 t} + e^{-j\omega_0 t}\} - 1/j \{e^{j3\omega_0 t} - e^{-j3\omega_0 t}\}$$

Ahora, teniendo en cuenta la fórmula de Euler, podemos re-escribir la señal  $x(t)$  como:

$$x(t) = 5 + 4\cos(\omega_0 t) - 2\text{sen}(3\omega_0 t)$$

Y reemplazando el valor de  $\omega_0$ ,

$$x(t) = 5 + 4\cos(200\pi t) - 2\text{sen}(600\pi t)$$

**Ejemplo 28. Reconstruir la señal  $x(t)$  a partir de los coeficientes de la SF (sólo nos proporcionan los valores de  $k$  positivos)**

Se tiene una señal  $x(t)$  real y periódica con que  $\omega_0 = 100\pi \left[ \frac{\text{rad}}{\text{s}} \right]$ . Esta señal contiene un nivel de d.c., la primera y la cuarta armónica, cuyos coeficientes de la SF, son:

$$a_0 = 2$$

$$a_1 = 5$$

$$a_4 = 2j$$

Encuentre  $x(t)$  expresada de la forma  $x(t) = A + B\cos(\omega_0 t) + C\text{sen}(4\omega_0 t)$

**Solución:**

Aunque no nos proporcionaron el valor de  $a_{-1}$ , ni el de  $a_{-4}$ , a partir de la fórmula de Euler sabemos que si un coeficiente es estrictamente entero,

su pareja será también entero de la misma magnitud y signo; mientras que, si un coeficiente es estrictamente imaginario, su pareja también será imaginario (misma magnitud), pero de signo contrario. Es decir, para este ejercicio tenemos que  $a_{-1} = 5$ , y  $a_{-4} = -2j$ .

Entonces, aplicando la ecuación de síntesis, tenemos que:

$$x(t) = 2 + 5\{e^{j\omega_0 t} + e^{-j\omega_0 t}\} + 2j\{e^{j4\omega_0 t} - e^{-j4\omega_0 t}\}$$

O de forma equivalente,

$$x(t) = 2 + 5\{e^{j\omega_0 t} + e^{-j\omega_0 t}\} - 2/j \{e^{j4\omega_0 t} - e^{-j4\omega_0 t}\}$$

Ahora, teniendo en cuenta la fórmula de Euler, podemos re-escribir la señal  $x(t)$  como:

$$x(t) = 2 + 10\cos(\omega_0 t) - 4\text{sen}(4\omega_0 t)$$

Reemplazando el valor de la frecuencia fundamental  $\omega_0 = 100\pi \left[ \frac{\text{rad}}{\text{s}} \right]$ , obtenemos finalmente

$$x(t) = 2 + 10\cos(100\pi t) - 4\text{sen}(400\pi t)$$

**Ejemplo 29. Cálculo de los coeficientes de la SF de una señal periódica formada por componentes senoidales armónicamente relacionados.**

Dibuje la señal en el dominio del tiempo, calcule y grafique la magnitud de los coeficientes de la SF de la señal,

$$x(t) = 2\text{sen}\left(\frac{\pi}{2}t\right) + \cos\left(\frac{\pi}{2}t\right) + \text{sen}(2\pi t)$$

A diferencia de los dos primeros ejercicios, en este caso tenemos la señal en el dominio del tiempo, y debemos encontrar sus coeficientes de la SF.

Lo primero que debemos hacer es encontrar el M.C.D. (máximo común divisor) de las frecuencias de los tres términos de  $x(t)$ , es decir:

$$\text{M.C.D} (\pi/2, \pi/2, 2\pi) = \pi/2$$

Es decir, tenemos que  $\omega_0 = \pi/2$ , entonces tenemos dos términos de aportan a la primera armónica, y un término que aporta a la cuarta armónica (dado que  $2\pi/0.5\pi=4$ ).

Al aplicar Euler, tenemos que:

$$a_1 = 0.5 - j$$

$$a_{-1} = 0.5 + j$$

$$a_4 = -j/2$$

$$a_{-4} = j/2$$

Como siguiente paso, vamos a calcular la magnitud de los coeficientes obtenidos, y nos queda así

$$|a_1| = |a_{-1}| = \sqrt{0.5^2 + 1} = \sqrt{1.25} = 1.118$$

$$|a_4| = |a_{-4}| = \sqrt{0.5^2} = \sqrt{0.25} = 0.5$$

Para el código en Python, primero vamos a re-escribir la señal  $x(t)$  en términos de la frecuencia fundamental ( $f_0$ ), así:

$$x(t) = A * \sin(k_1 * 2\pi * f_0 * t) + B * \cos(k_2 * 2\pi * f_0 * t) + C * \sin(k_3 * 2\pi * f_0 * t)$$

Donde A, B y C son amplitudes de las señales senoidales o cosenoidales; mientras que  $k_1, k_2, k_3$  corresponden al valor de la armónica de esa señal senoidal o cosenoidal.

Quedando para este ejemplo, así:

$$x(t) = 2 * \sin(1 * 2\pi * f_0 * t) + 1 * \cos(1 * 2\pi * f_0 * t) + 1 * \sin(4 * 2\pi * f_0 * t)$$

Ahora, vamos a utilizar el siguiente código en Python:

```

import numpy as np
import matplotlib.pyplot as plt

# Parámetros de la señal
fo = 1/4 # frecuencia fundamental de la señal [Hz]
fs = 40 * fo # frecuencia de muestreo (cumple Nyquist)
ciclos = 1 # cantidad de ciclos a visualizar
stop = ciclos / fo # tiempo final
ns = int(stop * fs) # número de muestras

# Vector de tiempo (alineado exactamente con la fs)
t = np.arange(0, stop, 1/fs)

# Definición de la señal
x = 2*np.sin(2*np.pi*fo*t) + np.cos(2*np.pi*fo*t) + np.sin(4*2*np.pi*fo*t)

# Configuración de la figura
plt.rcParams["figure.figsize"] = (8,4)

# Gráfica
plt.plot(t, x, label='Señal continua')
plt.plot(t, x, 'ko', label='Muestras')

plt.xlabel('Tiempo (s)')
plt.ylabel('$x(t)$')
plt.title(r'Señal $x(t)=2\sin(2\pi f_0 t)+\cos(2\pi f_0 t)+\sin(8\pi f_0 t)$')

plt.grid(True)
plt.legend()
plt.show()

```

Figura 82. Código en Python Ejemplo 28, señal

$$x(t) = 2\sin(2\pi f_0 t) + \cos(2\pi f_0 t) + \sin(8\pi f_0 t).$$

Del código anterior, lo primero que se definió fue la frecuencia fundamental de la señal, expresada en Hertz, que corresponde a  $f_0 = 0.25 \text{ Hz}$ , dado que la frecuencia angular fundamental es  $\omega_0 = \frac{\pi}{2} [\text{rad/s}]$ .

Como segundo parámetro, definimos la frecuencia de muestreo de la señal. De acuerdo con el criterio de Nyquist, esta debe ser al menos el doble de la frecuencia máxima presente en la señal. En este ejercicio, la máxima armónica es la cuarta, por lo que  $f_{m\acute{a}x} = 4f_0 = 4(0.25) = 1 \text{ [Hz]}$ . Como hemos definido  $f_s = 40f_0 = 40(0.25) = 10 \text{ [Hz]}$ , se cumple ampliamente con el criterio de Nyquist.

Como tercer parámetro se define la cantidad de ciclos de la señal a visualizar, que en este caso se ha fijado en un solo ciclo.

El tiempo final de visualización (stop) se calcula a partir del número de ciclos y de la frecuencia fundamental de la señal. Posteriormente, el número de muestras se obtiene multiplicando el tiempo total de visualización por la frecuencia de muestreo.

A continuación, se construyen dos vectores: el primero corresponde al vector de tiempo de la señal y el segundo a los valores de amplitud de  $x(t)$ . El vector de tiempo se genera utilizando la instrucción `np.arange`, la cual crea muestras separadas exactamente por el periodo de muestreo  $1/f_s$ . Finalmente, la señal se calcula utilizando las funciones `np.sin` y `np.cos`, y se grafica con la función `plt.plot`.

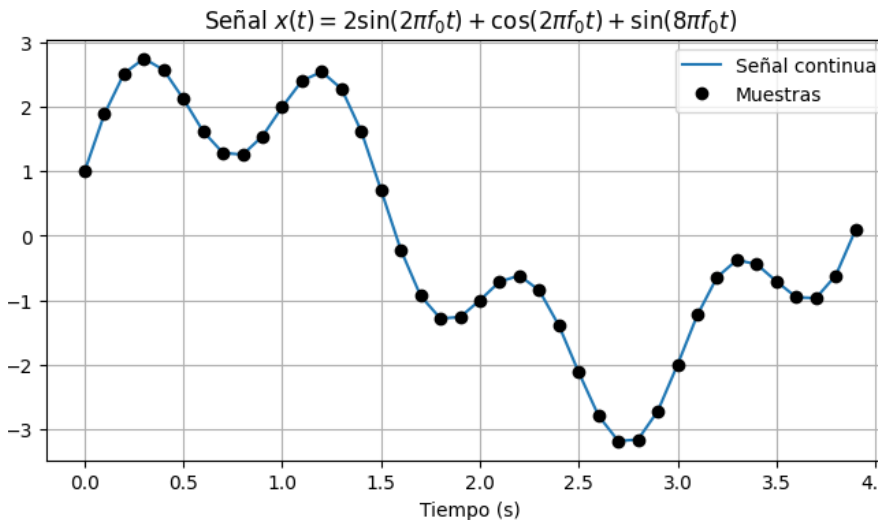


Figura 83. Gráfica del Ejemplo 28, señal  $x(t) = 2\sin\left(\frac{\pi}{2}t\right) + \cos\left(\frac{\pi}{2}t\right) + \sin(2\pi t)$

A continuación, vamos a utilizar el siguiente código, para graficar la magnitud de los coeficientes de la SF:

```

from scipy.fft import fft, fftfreq
import numpy as np
import matplotlib.pyplot as plt

# número de muestras
n = len(x)

# Serie de Fourier
ak = fft(x) / n
k = fftfreq(n, 1/fs)

# posiciones armónicas
k = k / fo

# Configurar la visualización
plt.stem(k, np.abs(ak))
plt.xlim(-6, 6)
plt.ylim(0, 2)

plt.xlabel('k')
plt.ylabel(r'$|a_k|$')
plt.title('Magnitud de los coeficientes de la Serie de Fourier')

plt.grid(True)
plt.show()

```

Figura 84. Código en Python Ejemplo 28, coeficientes de la SF.

Lo primero que hacemos es calcular la longitud de la señal (es decir, la cantidad de muestras de la señal  $x(t)$  en el dominio del tiempo) y asignarla al parámetro  $n$ . Posteriormente, calculamos la transformada rápida de Fourier de la señal con la función  $fft$ , y normalizamos el resultado dividiendo entre  $n$ .

Como siguiente paso, utilizamos la función `fftfreq` para encontrar las frecuencias asociadas a cada coeficiente obtenido. Dado que la Serie de Fourier se expresa en términos del índice armónico  $k$ , dividimos estas frecuencias entre la frecuencia fundamental  $f_0$ , obteniendo así las posiciones correspondientes de  $k$ .

Finalmente, utilizamos la función `plt.stem` para graficar la magnitud de los coeficientes  $|a_k|$ .

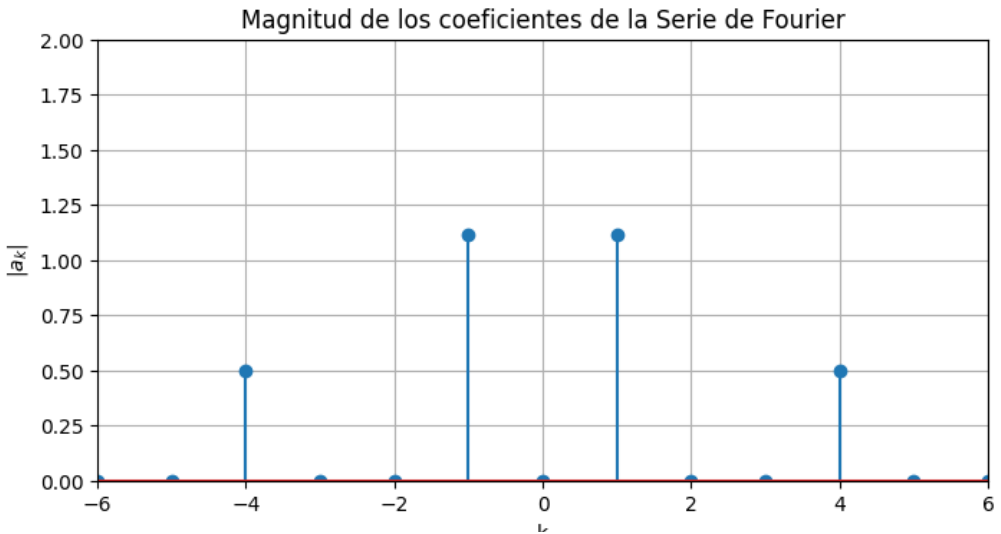


Figura 85. Código en Python Ejemplo 28, coeficientes de la SF de la señal

$$x(t) = 2\sin\left(\frac{\pi}{2}t\right) + \cos\left(\frac{\pi}{2}t\right) + \sin(2\pi t).$$

Al comparar las amplitudes obtenidas en la figura anterior, con los valores obtenidos en los cálculos teóricos, nos damos cuenta que corresponden al mismo resultado.

### Ejemplo 30. Cálculo de los coeficientes de la SF de una señal periódica expresada como producto de cosenos

Calcule y dibuje los coeficientes de la SF de la señal  
 $x(t) = 2\cos(\omega_0 t)\cos(2\omega_0 t)$ .

#### Solución:

En este ejercicio aplicaremos la propiedad de multiplicación de la Serie de Fourier, la cual establece que cuando dos señales periódicas se multiplican en el dominio del tiempo, los coeficientes de la SF de la señal resultante corresponden a la convolución de los coeficientes de la SF de cada señal.

Para aplicar esta propiedad, primero separamos la señal en dos factores:

$$x(t) = x_1(t) \cdot x_2(t)$$

Donde

$$x_1(t) = 2\cos(\omega_0 t) \xrightarrow{SF} a_k$$

$$x_2(t) = \cos(2\omega_0 t) \xrightarrow{SF} b_k$$

$$x(t) \xrightarrow{SF} c_k = a_k \otimes b_k$$

$$a_1 = 1 \quad \text{y} \quad a_{-1} = 1$$

Y con la señal  $x_2(t)$ , tenemos, que:

$$b_2 = 1/2 \quad \text{y} \quad b_{-2} = 1/2$$

Debemos tener en cuenta que en la señal  $x_2(t)$  aparece la segunda armónica, dado que la frecuencia del coseno es  $2\omega_0$ . De forma similar, en la señal  $x_1(t)$  aparece la primera armónica, porque la frecuencia es  $\omega_0$ .

A continuación, se presenta la implementación en Python para ilustrar la convolución de coeficientes de la SF.

```

import numpy as np
import matplotlib.pyplot as plt

COL_A, COL_B, COL_C = "#D32F2F", "#5E35B1", "#1565C0"

def impulses(ax, k, a, col, title=""):
    ax.axhline(0, color="black", lw=2)
    ax.set_xlim(-4,4); ax.set_ylim(-0.25,1.25)
    ax.set_xticks([]); ax.set_yticks([])
    for s in ax.spines.values(): s.set_visible(False)

    for ki, ai in zip(k,a):
        ax.vlines(ki,0,ai,color=col,lw=5)
        txt = r"$\frac{1}{2}$" if np.isclose(ai,0.5) else f"{ai:g}"
        ax.text(ki,ai+0.06,txt,ha="center",color=col)
        ax.text(ki,-0.12,f"{int(ki)}",ha="center",color=col)

    ax.text(3.9,0.02,"$k$",ha="right")
    ax.set_title(title)

# coeficientes
ka, aa = np.array([-1,1]), np.array([1,1])
kb, bb = np.array([-2,2]), np.array([0.5,0.5])

# convolución
k1, c1 = kb-1, bb
k2, c2 = kb+1, bb

# resultado final
k = np.array([-3,-1,1,3])
c = np.array([0.5,0.5,0.5,0.5])

# figura
fig = plt.figure(figsize=(10,7))
gs = fig.add_gridspec(3,2, height_ratios=[1,1,1.2])

ax1 = fig.add_subplot(gs[0,0])
ax2 = fig.add_subplot(gs[0,1])
ax3 = fig.add_subplot(gs[1,0])
ax4 = fig.add_subplot(gs[1,1])
ax5 = fig.add_subplot(gs[2,:]) # ocupa todo el ancho

impulses(ax1,ka,aa,COL_A,"$a_k$")
impulses(ax2,kb,bb,COL_B,"$b_k$")
impulses(ax3,k1,c1,COL_C,"Conv. con $a_{-1}$")
impulses(ax4,k2,c2,COL_C,"Conv. con $a_{1}$")
impulses(ax5,k,c,COL_C,"Resultado $c_k = a_k \circledast b_k$")

plt.tight_layout()
plt.show()

```

Figura 86. Código en Python Ejemplo 29, aplicación de la propiedad de multiplicación.

Al realizar la convolución de los coeficientes  $a_k$  con los coeficientes  $b_k$ , obtenemos:

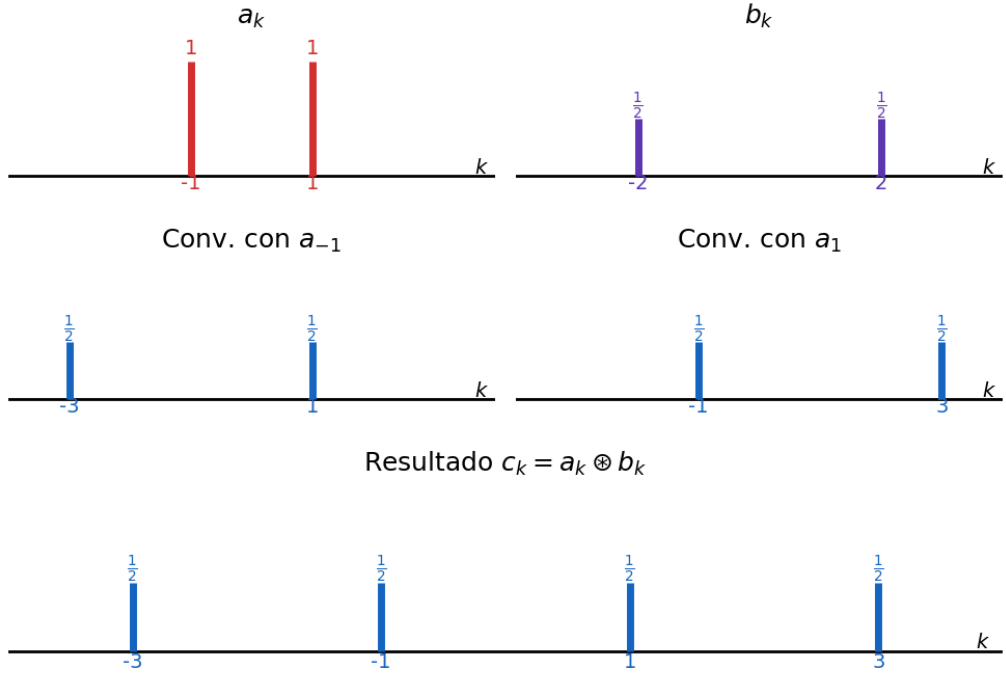


Figura 87. Gráfica del Ejemplo 29, aplicación de la propiedad de multiplicación.

Con

$$c_1 = 0.5$$

$$c_3 = 0.5$$

$$c_{-1} = 0.5$$

$$c_{-3} = 0.5$$

Observamos que, como resultado de la convolución de los coeficientes  $a_k$  y  $b_k$ , aparecen componentes espectrales en  $k = \pm 1$  y  $k = \pm 3$ . Esto ilustra cómo la multiplicación de señales en el dominio del tiempo genera nuevas componentes armónicas en el dominio de la frecuencia.

**Ejemplo 31. Aplicación de la Relación de Parseval**

Determine la potencia promedio de la señal  $x(t)$  real y periódica, a partir de sus coeficientes de la SF

$$a_0 = 2 \qquad a_1 = 1 - 2j \qquad a_{-1} = 1 + 2j$$

$$a_2 = 0.5 - j \qquad a_{-2} = 0.5 + j$$

**Solución:**

Recordando la relación de Parseval, tenemos que la potencia promedio de una señal (es decir,  $\bar{P}$ ), se calcula a partir de:

$$\bar{P} = \sum_{k=-\infty}^{\infty} |a_k|^2$$

Primero, calculamos la magnitud de los coeficientes de la SF, así:

$$|a_0| = |2|$$

$$|a_1| = |a_{-1}| = \sqrt{1^2 + 2^2} = \sqrt{5}$$

$$|a_2| = |a_{-2}| = \sqrt{0.5^2 + 1^2} = \sqrt{1.25}$$

Reemplazando los valores, se tiene:

$$\bar{P} = |a_0|^2 + |a_1|^2 + |a_{-1}|^2 + |a_2|^2 + |a_{-2}|^2$$

$$\bar{P} = |a_0|^2 + 2|a_1|^2 + 2|a_2|^2$$

$$\bar{P} = 4 + 2(5) + 2(1.25) = 16.5$$

**3.8. RESUMEN DEL CAPITULO DE SERIES DE FOURIER**

En este capítulo estudiamos la Serie de Fourier, una herramienta fundamental del análisis de señales que permite representar señales periódicas como la suma de componentes sinusoidales relacionadas armónicamente.

Inicialmente, introdujimos la señal exponencial compleja, la cual constituye la base matemática para la representación de señales periódicas en

términos de exponenciales complejas. Esta representación facilita el análisis en el dominio de la frecuencia y permite expresar cualquier señal periódica como una combinación de armónicas de la frecuencia fundamental.

Posteriormente, presentamos la expresión general de la Serie de Fourier, que permite descomponer una señal periódica  $x(t)$  en una suma infinita de exponenciales complejas:

$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 t}$$

donde  $\omega_0$  corresponde a la frecuencia fundamental de la señal y  $a_k$  representa a los coeficientes de la serie, los cuales determinan tanto la amplitud como la fase de cada componente armónica.

A lo largo del capítulo analizamos diversos ejemplos que nos permitieron comprender cómo calcular los coeficientes de la Serie de Fourier y cómo interpretar su representación en el dominio de la frecuencia. Observamos que esta descomposición produce un espectro discreto, compuesto por líneas espectrales cuya amplitud y fase están determinadas por los coeficientes  $a_k$ .

También estudiamos algunas propiedades de la Serie de Fourier que facilitan el cálculo de los coeficientes cuando la señal presenta ciertas simetrías o cuando se realizan transformaciones en el dominio del tiempo. Finalmente, analizamos el fenómeno de Gibbs, el cual aparece al aproximar señales con discontinuidades mediante un número finito de armónicas.

En conjunto, estos resultados nos permiten comprender cómo se distribuye la energía de una señal periódica en el dominio de la frecuencia, a través de un espectro discreto compuesto por armónicas. Sin embargo, muchas señales presentes en aplicaciones reales no son periódicas. En estos casos, al considerar que el periodo tiende a infinito, esta representación discreta en frecuencia se transforma en una representación continua, dando origen a la Transformada de Fourier, herramienta que permite analizar señales no periódicas en el dominio de la frecuencia.



## CAPÍTULO IV

# Transformada de Fourier de Señales periódicas y aperiódicas

La Transformada de Fourier extiende el análisis al caso de señales no periódicas, permitiendo representarlas en términos de un continuo de frecuencias. En este capítulo, partiremos de señales matemáticas y avanzaremos hasta una señal real, en particular una señal de voz, con el objetivo de comprender cómo la Transformada de Fourier revela su contenido espectral y permite identificar componentes como interferencias o ruido, tanto desde el punto de vista teórico como a través de su implementación en Python.

Al finalizar el capítulo, deberás estar en capacidad de:

- Explicar la relación entre la Serie de Fourier y la Transformada de Fourier, y su aplicación en señales periódicas y aperiódicas.
- Interpretar la Transformada de Fourier como una representación en frecuencia continua e identificar los principales componentes espectrales de una señal.
- Comprender y aplicar las principales propiedades de la Transformada de Fourier y su efecto en la relación tiempo–frecuencia.
- Aplicar la Transformada de Fourier mediante la FFT en Python para analizar señales.
- Interpretar el espectro de señales reales, en particular señales de voz, identificando interferencias y reconociendo su utilidad como base del Procesamiento Digital de Señales.

#### 4.1. MOTIVACIÓN DE LA TRANSFORMADA DE FOURIER

En el capítulo anterior estudiamos la Serie de Fourier, una herramienta que permite representar una señal periódica como la suma de exponenciales complejas relacionadas armónicamente. Esta representación nos permitió analizar cómo se distribuye la energía de la señal en el dominio de la frecuencia, obteniendo un espectro discreto compuesto por líneas espectrales ubicadas en múltiplos enteros de la frecuencia fundamental.

Sin embargo, muchas señales presentes en aplicaciones reales no son periódicas. Por ejemplo, una señal de voz, una vibración mecánica o una señal biomédica suelen tener duración finita o variar de manera irregular en el tiempo. En estos casos, no es posible aplicar directamente la Serie de Fourier, ya que esta requiere que la señal sea periódica, es decir, que se repita indefinidamente en el tiempo.

**Nota aclaratoria:**

Matemáticamente, una señal periódica satisface la condición

$$x(t) = x(t - T)$$

donde  $T$  corresponde al periodo de la señal. Como consecuencia, una señal periódica tiene **soporte infinito**, ya que su forma se repite para todos los valores de  $t$ .

A partir de esta limitación surge una pregunta natural: ¿cómo podemos analizar en el dominio de la frecuencia señales que no son periódicas?

Una forma de abordar este problema consiste en construir una versión periódica de una señal no periódica. Para ello, imaginemos que tomamos una señal de duración finita y la repetimos periódicamente en el tiempo, obteniendo así una señal de periodo  $T$ , la cual puede representarse mediante la Serie de Fourier. Si aumentamos progresivamente el valor de  $T$ , las repeticiones de la señal se separan cada vez más en el eje temporal. En el límite, cuando  $T \rightarrow \infty$ , las repeticiones desaparecen y la señal resultante se comporta como una señal no periódica.

La Transformada de Fourier puede interpretarse como el límite de la Serie de Fourier cuando el periodo tiende a infinito.

Este proceso también tiene un efecto importante en el dominio de la frecuencia. En la Serie de Fourier, una señal periódica se representa mediante un conjunto discreto de frecuencias. A medida que el periodo de la señal aumenta, estas frecuencias se vuelven cada vez más cercanas entre sí. En el límite, cuando el periodo tiende a infinito, las líneas espectrales se vuelven infinitamente densas y el espectro deja de ser discreto para convertirse en un espectro continuo.

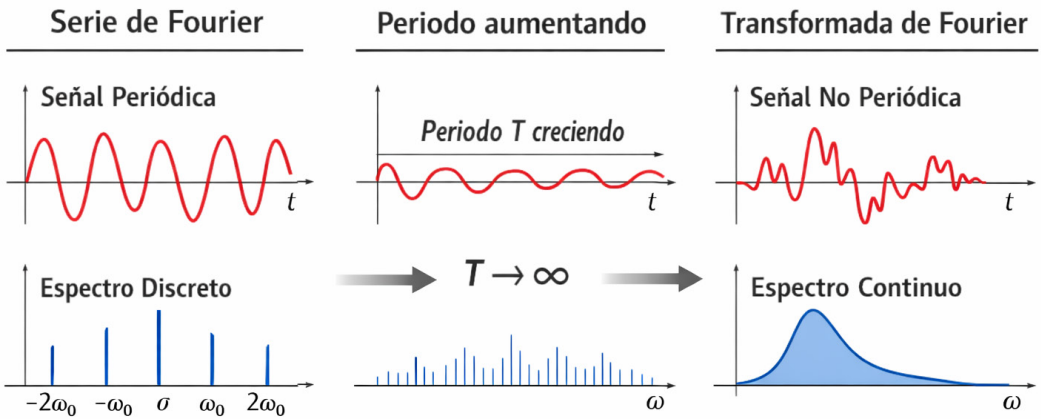


Figura 88. Transición conceptual entre la Serie de Fourier y la Transformada de Fourier. Inicialmente, la señal periódica (suma de senoidales) presenta un espectro discreto (parte izquierda de la figura). A medida que el periodo aumenta, las componentes espectrales discretas se acercan entre sí (como se ilustra en la parte central de la figura). Finalmente, en el límite  $T \rightarrow \infty$ , el espectro discreto se transforma en un espectro continuo (parte derecha de la figura).

Este razonamiento conduce naturalmente a la Transformada de Fourier, una herramienta que permite representar señales no periódicas mediante una distribución continua de frecuencias.

En este capítulo utilizaremos tanto la notación  $X(j\omega)$  como  $X(jf)$ , donde  $\omega = 2\pi f$ . La primera se expresa en [rad/s] y la segunda en [Hz].

El espectro describe cómo se distribuye la amplitud (y fase) de la señal en función de la frecuencia.

#### 4.2. TRANSFORMADA DE FOURIER DE SEÑALES PERIÓDICAS

Antes de iniciar el estudio de la Transformada de Fourier (TF) de señales aperiódicas, es importante analizar cómo se obtiene el espectro de una señal periódica a partir de los coeficientes de la Serie de Fourier (SF). Lo más importante para resaltar en este momento es que **el espectro de una señal periódica es siempre discreto**, y se calcula mediante alguna de las siguientes ecuaciones:

$$X(j\omega) = \sum_{k=-\infty}^{\infty} 2\pi a_k \delta(\omega - k\omega_0)$$

$$X(jf) = \sum_{k=-\infty}^{\infty} a_k \delta(f - kf_0)$$

En el primer caso, el espectro se expresa en **[rad/s]**; mientras que en el segundo en unidades de **[Hz]**.

Nota aclaratoria:

En este libro se utilizará la notación  $X(jf)$  para enfatizar que el espectro de una señal puede ser complejo. En muchos textos también se utiliza la notación  $X(f)$ ; ambas representan la Transformada de Fourier de la señal.

No obstante, en ambos casos los armónicos de la señal, los armónicos de la señal, que en la SF se denominan  $a_k$ , se convierten en **impulsos en frecuencia** que conservan la amplitud de la armónica y que están ubicados en  $k\omega_0$ , o  $kf_0$ , según corresponda.

Como diferencia importante, no solo cambia la ubicación de los impulsos en el eje de frecuencia, sino también su amplitud. Al comparar ambas

representaciones espectrales, se observa que a partir de  $X(j\omega)$  es posible obtener  $X(jf)$  realizando dos operaciones: la frecuencia se divide entre  $2\pi$ , y la amplitud de los impulsos también se divide entre  $2\pi$ .

### Ejemplo 32. TF de suma de señales cosenoidales

Supongamos que tenemos la señal

$$x(t) = 10\cos(100\pi t) + 8\cos(300\pi t) + 12\cos(500\pi t),$$

y deseamos graficar su Transformada de Fourier, tanto  $X(j\omega)$ , como  $X(jf)$ .

Lo primero que vamos a realizar es calcular los coeficientes  $a_k$ .

Identificamos la  $\omega_0$  calculando el M.C.D. entre las frecuencias angulares  $\{100\pi, 300\pi, 500\pi\} = 100\pi$ . Entonces, nuestra señal tiene la primera armónica, la tercera armónica y la quinta armónica, dado que la señal se puede re-escribir como:

$$x(t) = 10\cos(\omega_0 t) + 8\cos(3\omega_0 t) + 12\cos(5\omega_0 t)$$

Entonces, los valores de  $a_k$  distintos de cero, son:

$$a_1 = a_{-1} = 5$$

$$a_3 = a_{-3} = 4$$

$$a_5 = a_{-5} = 6$$

Correspondientes a tres armónicas y seis coeficientes. Recordemos que, al tratarse de señales cosenoidales (y no senoidales), los coeficientes son **reales y pares**.

Ahora, aplicaremos la ecuación que relaciona  $X(jf)$  con los seis valores de  $a_k$ , quedando:

$$X(jf) = a_1\delta(f - f_0) + a_{-1}\delta(f + f_0) + a_3\delta(f - 3f_0) + a_{-3}\delta(f + 3f_0) + a_5\delta(f - 5f_0) + a_{-5}\delta(f + 5f_0)$$

Y reemplazando  $f_0 = 100\pi/(2\pi) = 50$  [Hz], con los valores de  $a_k$ , obtenemos:

$$X(jf) = 5\delta(f - 50) + 5\delta(f + 50) + 4\delta(f - 150) + 4\delta(f + 150) + 6\delta(f - 250) + 6\delta(f + 250)$$

Finalmente, obtenemos  $X(j\omega)$ , multiplicando la amplitud por  $2\pi$ , al igual que las frecuencias, quedando:

$$X(j\omega) = 10\pi\delta(\omega - 100\pi) + 10\pi\delta(\omega + 100\pi) + 8\pi\delta(\omega - 300\pi) + 8\pi\delta(\omega + 300\pi) + 12\pi\delta(\omega - 500\pi) + 12\pi\delta(\omega + 500\pi)$$

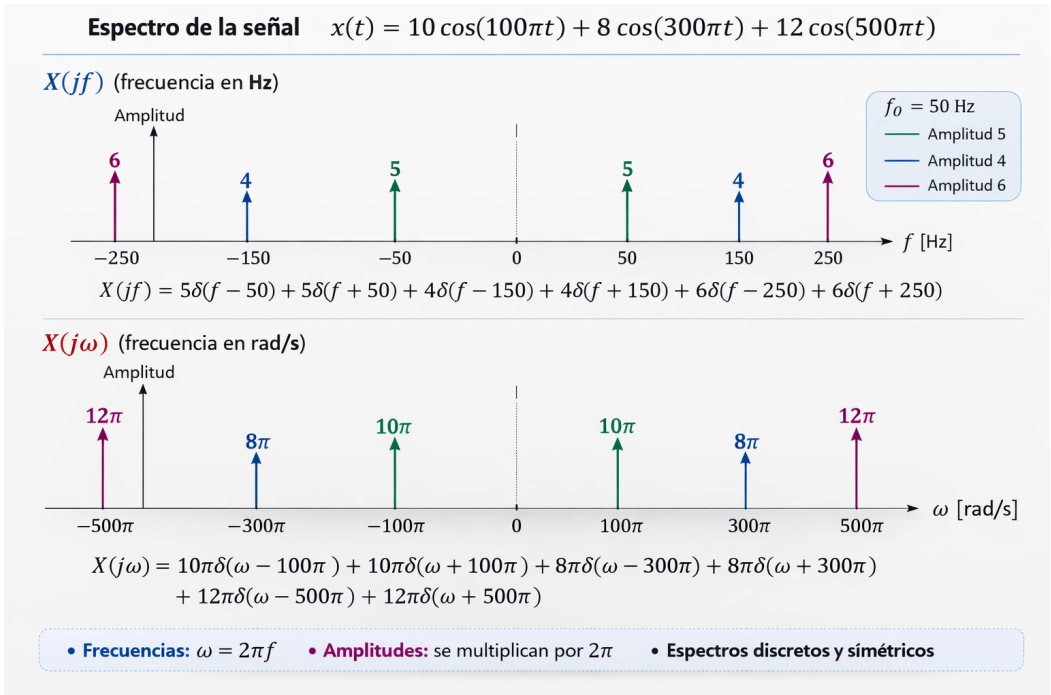


Figura 89. Ejemplo 31: Espectro  $X(jf)$  y  $X(j\omega)$  de la señal

$$x(t) = 10\cos(\omega_0 t) + 8\cos(3\omega_0 t) + 12\cos(5\omega_0 t).$$

**Ejemplo 33. TF de señal cuadrada periódica**

Considere la siguiente señal periódica

$$x(t) = \begin{cases} 1 & -T_1 \leq t \leq T_1 \\ 0 & \text{en otro caso} \end{cases}$$

con periodo

$$T = 20T_1$$

Esta señal corresponde a una señal cuadrada periódica.

Vamos entonces a calcular los coeficientes de la SF, recordando la ecuación que identificamos en el Capítulo 3, así:

$$a_0 = \frac{2T_1}{T}$$

$$a_k = \frac{\text{sen}(k\omega_0 T_1)}{k\pi}, \quad k \neq 0$$

Reemplazando los valores, tenemos

$$a_0 = \frac{2T_1}{20T_1} = \frac{2}{20} = 0.1$$

Y teniendo en cuenta que  $\omega_0 = 2\pi/T = 2\pi/(20T_1) = \pi/(10T_1)$ .

Entonces  $\omega_0 T_1 = (\pi T_1)/(10T_1) = \pi/10$

$$a_k = \frac{\text{sen}\left(\frac{k\pi}{10}\right)}{k\pi}, \quad k \neq 0$$

Como los coeficientes de la Serie de Fourier son diferentes de cero para infinitos valores de  $k$ , el espectro de la señal está compuesto por infinitos impulsos ubicados en las frecuencias armónicas  $f = kf_0$ , así:

$$X(jf) = \sum_{k=-\infty}^{\infty} \frac{\text{sen}\left(\frac{k\pi}{10}\right)}{k\pi} \delta(f - kf_0), \quad k \neq 0$$

Mientras que,

$$X(jf) = 0.1\delta(f)$$

Y en el caso de,

$$X(j\omega) = \sum_{k=-\infty}^{\infty} \frac{2 * \text{sen}\left(\frac{k\pi}{10}\right)}{k} \delta(\omega - k\omega_0), \quad k \neq 0$$

$$X(j\omega) = 0.2\pi\delta(\omega)$$

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros del ejemplo
T1 = 0.01
T = 20*T1
f0 = 1/T

# 1. Señal rectangular periódica
t = np.linspace(-2*T,2*T,2000)
x = np.zeros_like(t)

for n in range(-3,4):
    x[(t>=-T1+n*T) & (t<=T1+n*T)] = 1

plt.figure(); plt.plot(t,x)
plt.title("Señal rectangular periódica")
plt.xlabel("t (s)"); plt.ylabel("x(t)")
plt.grid(); plt.show()

# 2. Coeficientes de Fourier
k = np.arange(-20,21)
ak = np.zeros(len(k))

for i,ki in enumerate(k):
    if ki==0:
        ak[i] = 2*T1/T
    else:
        ak[i] = np.sin(ki*np.pi/10)/(ki*np.pi)

plt.figure(); plt.stem(k,ak)
plt.title("Coeficientes de la SF")
plt.xlabel("k"); plt.ylabel("a_k")
plt.grid(); plt.show()

# 3. Espectro X(jf)
f = k*f0

plt.figure(); plt.stem(f,ak)
plt.title("X(jf)"); plt.xlabel("Frecuencia (Hz)")
plt.ylabel("Amplitud")
plt.grid(); plt.show()
```

Figura 90. Ejemplo 32: Código enb Python del Ejemplo 32.

Obteniendo las siguientes gráficas:

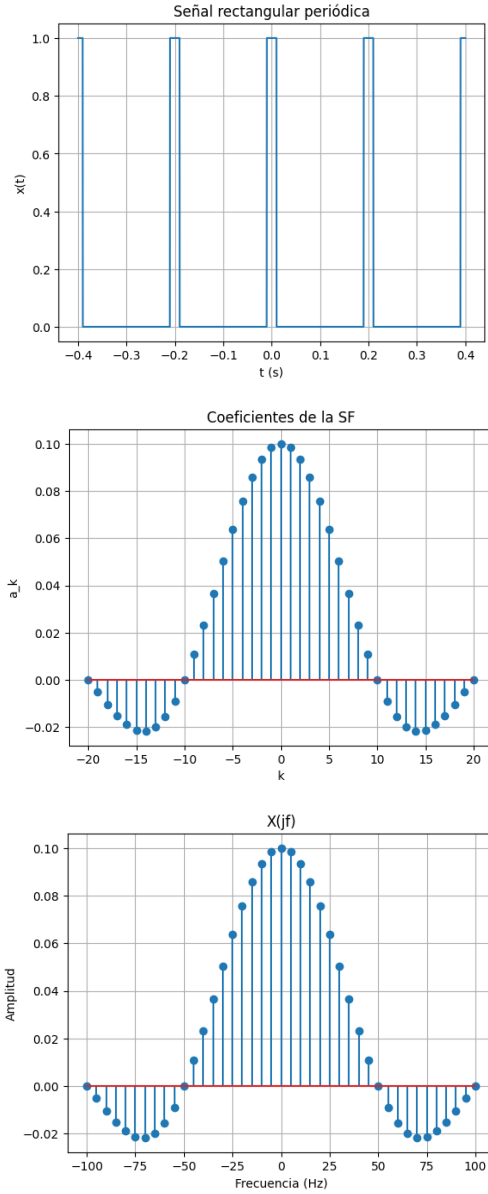


Figura 91. Ejemplo 32: a) Señal cuadrada periódica con  $T = 20T_1$ , b)  $a_k$ , c)  $X(jf)$ .

La señal cuadrada generada tiene  $T_1 = 0.01$  [s], por lo que el tiempo total durante el cual la señal permanece en alto es  $2T_1 = 0.02$  [s]. Por otra parte, el periodo de la señal es de  $T = 0.2$  [s].

En la gráfica de los coeficientes de la Serie de Fourier se han representado los valores de  $-20 \leq k \leq 20$ , lo que corresponde a un total de 41 términos (es decir,  $2k_{m\acute{a}x} + 1$  con  $k_{m\acute{a}x} = 20$ ), incluyendo el término de nivel de d.c. de la señal.

Una vez obtenidos los coeficientes de la Serie de Fourier, el espectro de la señal periódica puede construirse ubicando cada coeficiente  $a_k$  en la frecuencia armónica correspondiente  $f = kf_0$ .

Teniendo en cuenta que  $f_0 = 1/T$ , la frecuencia fundamental de la señal es  $5$  [Hz]. Por consiguiente, el coeficiente de la SF correspondiente a  $k = 20$  se ubica en el espectro en la frecuencia  $20f_0 = 100$  [Hz].

### 4.3. CONDICIONES SUFICIENTES PARA LA EXISTENCIA DE LA TRANSFORMADA DE FOURIER

Antes de aplicar la Transformada de Fourier a una señal aperiódica, es necesario analizar si dicha transformada existe. Aunque un amplio número de señales aperiódicas admiten Transformada de Fourier, existen algunas para las cuales no es posible calcular su espectro, ya que no poseen energía finita.

Por esta razón, se han establecido ciertas condiciones matemáticas que permiten determinar si una señal aperiódica admite representación en el dominio de la frecuencia. Estas condiciones se conocen como **condiciones de Dirichlet**. En particular, una de las condiciones fundamentales es que la señal tenga energía finita, es decir:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt < \infty$$

Específicamente, las condiciones de Dirichtlet son tres:

- Condición 1: Se cumpla que la señal sea absolutamente integrable, es decir,

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty$$

- Condición 2: la señal tenga un número finito de máximos y mínimos en cualquier intervalo finito.
- Condición 3: la señal tenga un número finito de discontinuidades en cualquier intervalo finito. Cada discontinuidad debe ser finita.

Nota aclaratoria:

Para el caso de señales periódicas, se aplicarán las condiciones de Dirichtlet de la SF.

#### 4.4. TRANSFORMADA DE FOURIER DE SEÑALES APERIÓDICAS

En la sección anterior vimos que la Transformada de Fourier de una señal periódica consiste en un conjunto de impulsos en frecuencia ubicados en múltiplos enteros de la frecuencia fundamental (es decir  $(k\omega_0)$ , o  $(kf_0)$  para  $X(j\omega)$  o  $X(jf)$ , respectivamente).

Sin embargo, muchas señales de interés en aplicaciones de ingeniería en telecomunicaciones no son periódicas. Por lo cual, es necesario extender el concepto de Transformada de Fourier hacia señales aperiódicas.

Como se mencionó en la sección 4.1. esto puede interpretarse como el caso límite en el cual el periodo de una señal periódica crece indefinidamente, es decir,  $T \rightarrow \infty$ . En esta situación, los impulsos del espectro dejan de estar separados por intervalos discretos y pasan a formar una **distribución continua en frecuencia**.

La Transformada de Fourier de señales aperiódicas se define como:

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

o también,

$$X(jf) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

A diferencia de la SF, donde aparecen únicamente frecuencias discretas (armónicas), en este caso la frecuencia puede tomar **cualquier valor continuo**.

Ahora bien, la señal  $x(t)$  puede reconstruirse a partir de  $X(j\omega)$  o  $X(jf)$  así, utilizando la Transformada Inversa de Fourier (ITF):

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) e^{j\omega t} d\omega$$

Y,

$$x(t) = \int_{-\infty}^{\infty} X(jf) e^{j2\pi ft} df$$

La relación entre la Serie de Fourier y la Transformada de Fourier puede entenderse observando cómo cambia la representación espectral de una señal dependiendo de su periodicidad. La siguiente figura resume estas representaciones para señales periódicas y aperiódicas.

## Relación entre Serie de Fourier y Transformada de Fourier

— Cómo se representa una señal en el dominio de la frecuencia —

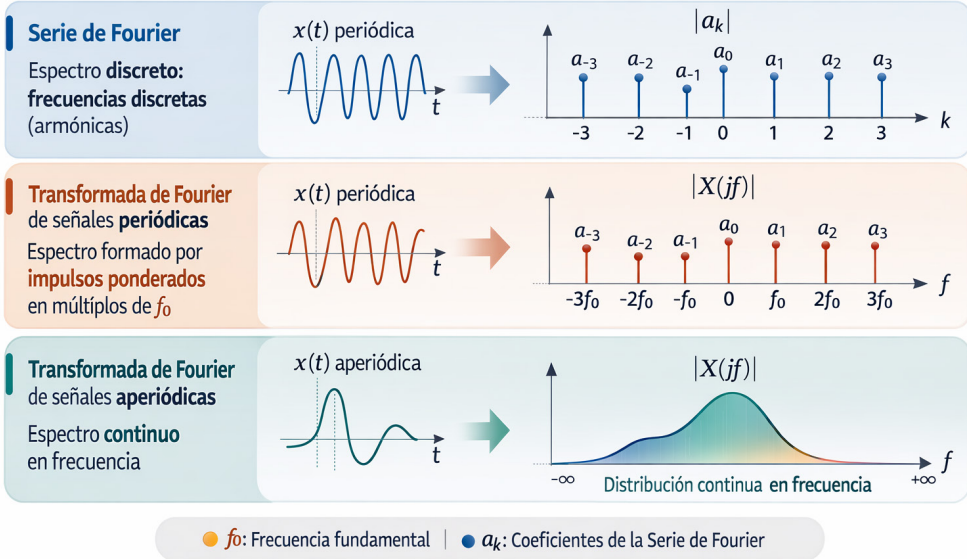


Figura 92. Relación entre la Serie de Fourier y la Transformada de Fourier. Para señales periódicas se obtiene un espectro discreto, mientras que para señales aperiódicas la Transformada de Fourier produce un espectro continuo.

A continuación, realizaremos la TF de algunas señales aperiódicas.

### Ejemplo 34. TF de señal exponencial real unilateral

Sea la siguiente señal:

$$x(t) = e^{-at}u(t), \quad a > 0$$

Recordando que la señal escalón unitario existe para los valores de tiempo positivos (incluido el cero),  $x(t)$  puede escribirse como:

$$x(t) = \begin{cases} e^{-at} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Entonces, en la ecuación de la TF de la señal, solamente se tendrá la integral a partir de cero, es decir:

$$X(j\omega) = \int_0^{\infty} x(t) e^{-j\omega t} dt = \int_0^{\infty} e^{-at} e^{-j\omega t} dt$$

Dado que para ese rango de tiempo la señal  $x(t)$  toma el valor  $e^{-at}$ .

Resolviendo la integral, obtenemos que:

$$X(j\omega) = \left[ \frac{e^{-(a+j\omega)t}}{-(a+j\omega)} \right]_0^{\infty}$$

Ahora, evaluamos los límites:

$$\lim_{t \rightarrow \infty} e^{-(a+j\omega)t} = 0 \quad (\text{dado que } a > 0)$$

Y

$$e^{-(a+j\omega)0} = 1$$

Por lo tanto:

$$X(j\omega) = \frac{0 - 1}{-(a+j\omega)} = \frac{1}{(a+j\omega)}$$

Teniendo en cuenta que el espectro contiene una componente imaginaria, es necesario calcular su magnitud para graficarla, obteniendo:

$$|X(j\omega)| = \frac{1}{\sqrt{a^2 + \omega^2}}$$

Para realizar la gráfica de  $|X(j\omega)|$ , vamos a utilizar dos valores de referencia:

- Cuando  $\omega = 0$ , tenemos que  $|X(0)| = \frac{1}{\sqrt{a^2}} = \frac{1}{a}$
- Cuando  $\omega = a$ , tenemos que  $|X(a)| = \frac{1}{a\sqrt{2}}$

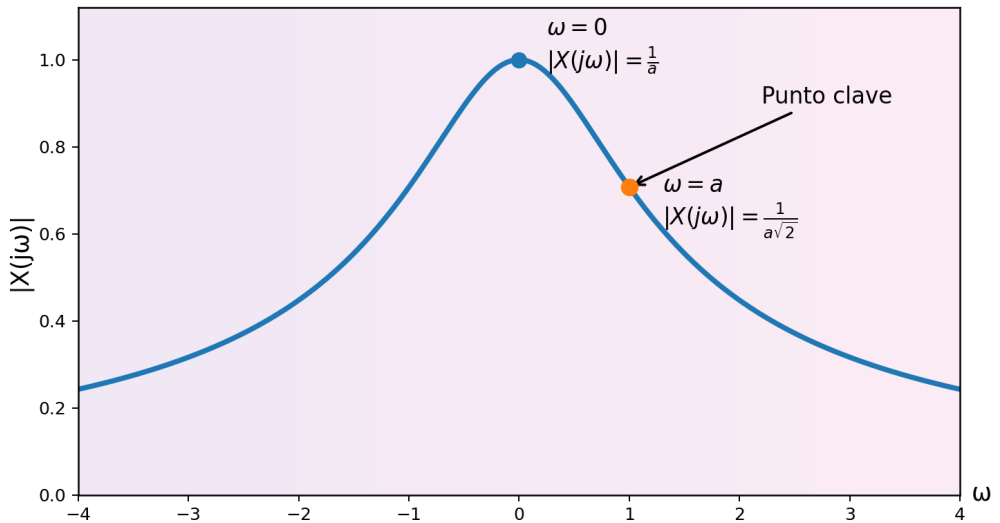


Figura 93. Ejemplo 34: Magnitud del espectro de la señal  $e^{-at}u(t)$ .

Se observa que el valor máximo de la magnitud del espectro ocurre en  $\omega = 0$ , y corresponde a  $1/a$ . A medida que la frecuencia aumenta, la magnitud disminuye de forma suave. En particular, cuando  $\omega = a$ , la magnitud se reduce a  $1/(a\sqrt{2})$ , lo que indica una caída significativa en el contenido espectral.

**Nota aclaratoria:**

Esta señal es importante dentro de comunicaciones, dado que su espectro presenta un comportamiento similar al de un filtro pasa-bajo. En particular, concentra su mayor energía en bajas frecuencias (alrededor de  $\omega=0$ ) y atenúa progresivamente las componentes de alta frecuencia a medida que  $\omega$  aumenta.

**Ejemplo 35. TF de señal exponencial real bilateral**

En este ejemplo, vamos a trabajar también con la señal exponencial real, pero en este caso, la versión bilateral. Esta señal exponencial tiene dos componentes, una señal creciente para los valores negativos de tiempo, y una señal decreciente para los valores positivos de tiempo.

Es decir,

$$x(t) = \begin{cases} e^{at} & t < 0 \\ e^{-at} & t > 0 \end{cases}$$

Esta señal se puede escribir de forma compacta, así:

$$x(t) = e^{-a|t|}$$

Dado que para los tiempos positivos  $|t| = t$ ; mientras que, para los tiempos negativos  $|t| = -t$ .

Como esta señal tiene dos componentes, aplicamos la ecuación de la TF en dos partes, así:

$$X(j\omega) = \int_{-\infty}^0 e^{at} e^{-j\omega t} dt + \int_0^{\infty} e^{-at} e^{-j\omega t} dt$$

Resolviendo,

$$\int_{-\infty}^0 e^{(a-j\omega)t} dt = \left[ \frac{e^{(a-j\omega)t}}{(a-j\omega)} \right]_{-\infty}^0 = \frac{1}{(a-j\omega)}$$

$$\int_0^{\infty} e^{-(a+j\omega)t} dt = \frac{1}{(a+j\omega)}$$

Y entonces, tenemos que:

$$X(j\omega) = \frac{1}{(a-j\omega)} + \frac{1}{(a+j\omega)} = \frac{2a}{(a^2 + \omega^2)}$$

A diferencia de la señal exponencial unilateral, cuyo espectro es complejo, en este caso el espectro es completamente real y simétrico, por lo que  $|X(j\omega)| = X(j\omega)$ . Esto se debe a que la señal en el dominio del tiempo es par.

Para realizar la gráfica de  $|X(j\omega)|$ , vamos a utilizar dos valores de referencia:

- Cuando  $\omega = 0$ , tenemos que  $|X(0)| = \frac{2a}{a^2} = \frac{2}{a}$
- Cuando  $\omega = a$ , tenemos que  $|X(a)| = \frac{2a}{2a^2} = \frac{1}{a}$

Vamos ahora a comparar la magnitud de los espectros de la señal exponencial unilateral real, con el de la señal exponencial bilateral real.

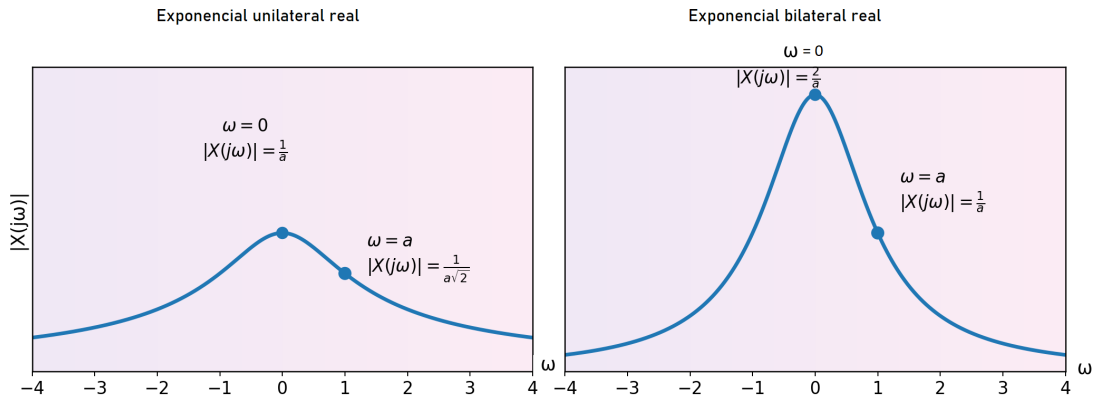


Figura 94. Ejemplo 35: Comparación de espectros de señales exponenciales unilateral y bilateral.

Comparando el espectro de la señal exponencial bilateral real con el de la señal exponencial unilateral real, se observa que cuando  $\omega = 0$  la amplitud de la señal exponencial bilateral es el doble. Por otro lado, cuando  $\omega = a$ , la amplitud es  $1/a$ , en comparación con aproximadamente  $0.707/a$  en la señal exponencial unilateral.

**Ejemplo 36. TF de señal impulso unitario en tiempo continuo**

Sea la señal impulso unitario en tiempo continuo definida como:

$$x(t) = \delta(t)$$

La TF de la señal, se expresa de la siguiente forma:

$$X(j\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt$$

De manera intuitiva,  $\delta(t)e^{-j\omega t}$  puede interpretarse considerando que el impulso unitario está completamente concentrado en  $t = 0$ . En ese punto, la exponencial compleja toma el valor de  $e^{-j\omega 0}$ , por lo que dentro de la integral el impulso selecciona únicamente ese valor.

Por lo tanto,

$$X(j\omega) = e^{-j\omega 0}$$

Y en consecuencia,

$$X(j\omega) = 1$$

Este ejercicio es muy importante en comunicaciones, ya que permite comprender una idea fundamental: la señal más concentrada en el dominio del tiempo, correspondiente al impulso unitario, posee el espectro más ancho posible, es decir, una constante de valor 1 para todas las frecuencias.

**Ejemplo 37. TF de una señal constante en el tiempo**

En este caso, la señal está definida como:

$$x(t) = 1,$$

Y su TF se expresa como:

$$X(j\omega) = \int_{-\infty}^{\infty} e^{-j\omega t} dt$$

Para obtener su espectro, utilizamos la propiedad de dualidad de la Transformada de Fourier.

$$\text{Si } \delta(t) \xrightarrow{TF} 1$$

$$\text{Entonces por dualidad, } 1 \xrightarrow{TF} 2\pi\delta(\omega)$$

Por lo tanto,

$$X(j\omega) = 2\pi\delta(\omega)$$

Este resultado indica que una señal constante en el tiempo contiene únicamente la componente de frecuencia cero, es decir, toda su información espectral está concentrada en  $\omega = 0$ .

#### Nota aclaratoria

De manera intuitiva, la propiedad de dualidad permite intercambiar el papel del tiempo y la frecuencia (es decir,  $t \rightarrow \omega$ , y  $\omega \rightarrow t$ ). Matemáticamente, dicho intercambio introduce un factor  $2\pi$  y una inversión en la variable. En este caso, esta inversión no altera el resultado, ya que el impulso es una función par.

En particular, para el caso de la señal impulso, se tiene que:

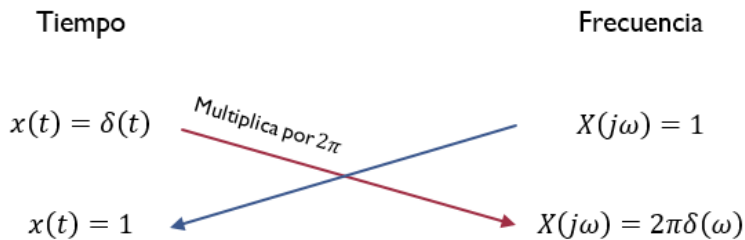


Figura 95. Ejemplo 37: Aplicación conceptual de la propiedad de dualidad para los pares  $\delta(t) \rightarrow 1$ , y  $1 \rightarrow 2\pi\delta(\omega)$ .

### Ejemplo 38. TF de la señal cuadrada aperiódica

Previamente, habíamos calculado el espectro de una señal cuadrada periódica, el cual es discreto, con infinitos impulsos, cuya envolvente es de la forma sinc. Ahora, calcularemos el espectro de la cuadrada aperiódica.

Sea  $x(t)$  una señal definida como:

$$x(t) = \begin{cases} 1 & |t| < T_1 \\ 0 & \text{en otro caso} \end{cases}$$

Donde  $2T_1$  corresponde a la duración total de la señal (tiempo en alto).

Nota aclaratoria:

Esta definición es consistente con el caso de la señal cuadrada periódica, en la cual  $2T_1$  representa el tiempo en que la señal permanece en nivel alto dentro de un período.

Entonces,

$$X(j\omega) = \frac{2 \operatorname{sen}(\omega T_1)}{\omega} = \frac{2T_1 \operatorname{sen}(\omega T_1)}{\omega T_1}$$

Este resultado indica que una señal limitada en el tiempo tiene un espectro continuo cuya forma es tipo sinc.

En particular:

- La mayor concentración de energía se encuentra alrededor de  $\omega = 0$ .
- El espectro presenta ceros en:

$$\omega = \pm \frac{\pi}{T_1}, \pm \frac{2\pi}{T_1}, \pm \frac{3\pi}{T_1}, \dots$$

- A mayor duración de  $T_1$ , el espectro se hace más angosto.
- A menor duración de  $T_1$ , el espectro se ensancha.

La señal rectangular aperiódica mantiene la misma estructura temporal que la señal cuadrada periódica en su intervalo activo, pero sin repetición. Su espectro revela cómo esta duración finita se traduce en una distribución continua en el dominio de la frecuencia.

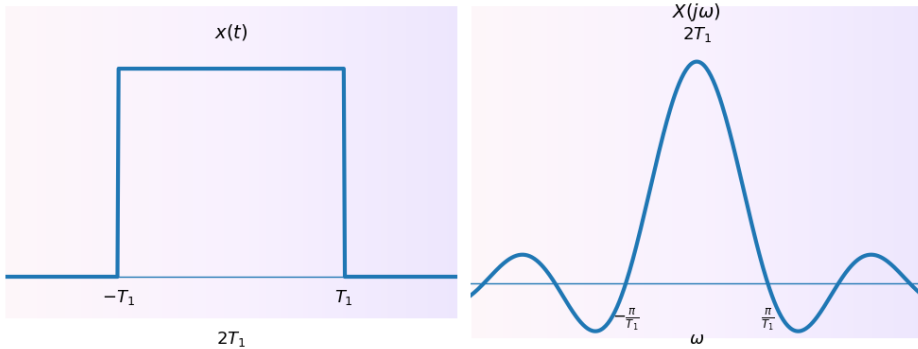


Figura 96. Ejemplo 38: señal cuadrada aperiódica,  $x(t)$  y  $X(j\omega)$ .

#### 4.5. TABLA RESUMEN TF DE SEÑALES PERIÓDICAS Y APERIÓDICAS

Como cierre de esta sección, se presentan dos tablas resumen, una para señales periódicas y otra para señales aperiódicas, junto con sus respectivas representaciones en frecuencia. Estas síntesis permiten integrar los resultados obtenidos y ofrecen una visión global de la Transformada de Fourier, resaltando la relación entre el dominio del tiempo y el dominio de la frecuencia.

	Tiempo $x(t)$		Frecuencia $X(j\omega)$
<b>Senoidal</b>	$\sin(\omega_0 t)$	$\leftrightarrow$	$\frac{\pi}{j}[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]$
<b>Cosenoidal</b>	$\cos(\omega_0 t)$	$\leftrightarrow$	$\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$
<b>Tren de impulsos</b>	$\sum_{k=-\infty}^{\infty} \delta(t - kT)$	$\leftrightarrow$	$\frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_0)$
<b>Cuadrada periódica</b>	$x(t) = \begin{cases} 1 & -T_1 \leq t \leq T_1, \\ 0 & \text{en otro caso} \end{cases}$ $a_0 = \frac{2T_1}{T}, \quad a_k = \frac{\sin(k\omega_0 T_1)}{k\pi}, \quad k \neq 0$	$\leftrightarrow$	$2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\omega - k\omega_0)$

Figura 97. Pares fundamentales de la Transformada de Fourier para señales periódicas. Las señales periódicas presentan espectros discretos compuestos por impulsos en múltiplos de la frecuencia fundamental.

	Tiempo $x(t)$		Frecuencia $X(j\omega)$
<b>Impulso</b>	$\delta(t)$	$\leftrightarrow$	1
<b>Constante</b>	1	$\leftrightarrow$	$2\pi\delta(\omega)$
<b>Exp. unilateral</b>	$e^{-at}u(t), a > 0$	$\leftrightarrow$	$\frac{1}{a + j\omega}$
<b>Exp. bilateral</b>	$e^{-a t }, a > 0$	$\leftrightarrow$	$\frac{2a}{a^2 + \omega^2}$
<b>Rectangular</b>	$x(t) = 1,  t  \leq T_1$	$\leftrightarrow$	$2T_1 \frac{\sin(\omega T_1)}{\omega T_1}$

Figura 98. Pares fundamentales de la Transformada de Fourier para señales aperiódicas. Se observa cómo señales localizadas en el dominio del tiempo, como el impulso unitario, presentan espectros extendidos, mientras que señales extendidas en el tiempo, como la constante, se concentran en frecuencia. La señal rectangular ilustra un caso intermedio, evidenciando la relación entre duración temporal y ancho espectral.

## 4.6. PROPIEDADES DE LA TRANSFORMADA DE FOURIER

En esta parte final del capítulo abordaremos las propiedades de la Transformada de Fourier. Conocerlas permite determinar el espectro de señales más complejas a partir de transformadas básicas, aplicando de manera estratégica aquellas propiedades que simplifican el análisis.

Estas propiedades establecen una relación directa entre transformaciones en el dominio del tiempo y sus efectos en el dominio de la frecuencia, lo que permite interpretar y construir resultados sin necesidad de recalcular la integral de la Transformada de Fourier.

A continuación, se presentan las propiedades más importantes y representativas para comunicaciones.

### 4.6.1. Linealidad

Esta propiedad es muy similar a la de linealidad de la Serie de Fourier (el lector puede compararlas). Básicamente, si se conoce el espectro de dos señales y estas se suman en el dominio del tiempo, entonces sus espectros también se suman.

Adicionalmente, si cada señal se multiplica por un escalar, su espectro se multiplica por el mismo escalar.

Sea  $x_1(t)$ ,  $x_2(t)$ , con TF:

$$x_1(t) \xrightarrow{TF} X_1(j\omega)$$

$$x_2(t) \xrightarrow{TF} X_2(j\omega)$$

Si estas señales se combinan linealmente de la forma:

$$x_3(t) = Ax_1(t) + Bx_2(t)$$

Entonces, su Transformada de Fourier se calcula así:

$$x_3(t) \xrightarrow{TF} AX_1(j\omega) + BX_2(j\omega)$$

Esta propiedad ya la hemos aplicado de forma intuitiva en algunos ejercicios, por ejemplo, cuando sumamos señales cosenoidales.

Cuando el espectro se expresa en Hz, la propiedad se escribe como:

$$x_2(t) \xrightarrow{TF} AX_1(jf) + BX_2(jf)$$

#### 4.6.2. Desplazamiento en el dominio del tiempo

Sea  $x_1(t)$  una señal con TF:

$$x_1(t) \xrightarrow{TF} X_1(j\omega)$$

Si esta señal se desplaza en el tiempo de la forma:

$$x_2(t) = x_1(t - t_0)$$

Entonces, la TF de la nueva señal está dada por:

$$x_2(t) \xrightarrow{TF} e^{-j\omega t_0} X_1(j\omega)$$

De lo anterior, se deduce que un desplazamiento en el tiempo no altera la magnitud del espectro, pero sí introduce un cambio en la fase.

Cuando el espectro se expresa en Hz, la propiedad se escribe como:

$$x_2(t) \xrightarrow{TF} e^{-j2\pi f t_0} X_1(jf)$$

#### 4.6.3. Escalamiento en el tiempo

Sea  $x_1(t)$  una señal con TF:

$$x_1(t) \xrightarrow{TF} X_1(j\omega)$$

Si esta señal se escala en el tiempo:

$$x_2(t) = x_1(at)$$

Entonces, la TF de la nueva señal está dada por:

$$x_2(t) \xrightarrow{TF} \frac{1}{|a|} X_1\left(\frac{j\omega}{a}\right)$$

El escalamiento en el dominio del tiempo genera un efecto contrario en el dominio de la frecuencia. Por ejemplo, si comprimo una señal en el tiempo, su espectro se dilata; y viceversa.

Nota aclaratoria:

Si te preguntas qué significa  $X_1\left(\frac{j\omega}{a}\right)$ , su interpretación es directa: en la expresión del espectro original  $X_1(j\omega)$ , se reemplaza la variable  $\omega$  por  $\frac{\omega}{a}$ .

Si el espectro lo expresas en Hz, la propiedad de escalamiento queda, así:

$$x_2(t) \xrightarrow{TF} \frac{1}{|a|} X_1\left(\frac{jf}{a}\right)$$

### Ejemplo 39. Compresión de una señal cosenoidal

Supongamos que tenemos la señal  $x_1(t) = \cos(100\pi t)$ , cuya frecuencia fundamental es  $\omega_0 = 100\pi$ . De acuerdo con nuestra tabla de TF de señales básicas periódicas, tenemos que el espectro es:

$$\cos(100\pi t) \xrightarrow{TF} \pi\{\delta(\omega - 100\pi) + \delta(\omega + 100\pi)\}$$

Ahora, vamos a suponer que esta señal se comprime por un factor de  $\alpha = 2$ , entonces, la TF de la señal comprimida, que llamaremos  $x_2(t)$  es:

$$x_2(t) \xrightarrow{TF} \frac{\pi}{2} \left\{ \delta\left(\frac{\omega}{2} - 100\pi\right) + \delta\left(\frac{\omega}{2} + 100\pi\right) \right\}$$

El espectro de la señal lo podemos re-escribir así:

$$X_2(j\omega) = \frac{\pi}{2} \left\{ \delta\left(\frac{\omega - 200\pi}{2}\right) + \delta\left(\frac{\omega + 200\pi}{2}\right) \right\}$$

Pero, ¿qué significa  $\delta\left(\frac{\omega}{2}\right)$ ?

Recordando la interpretación del impulso continuo como el límite de una señal de ancho  $\Delta$  muy pequeño y altura  $1/\Delta$ , se tiene que su área es igual a 1. Si esta señal se dilata en el eje de la variable (en este caso  $\omega$ ), su ancho pasa a ser  $2\Delta$ , por lo que se obtiene:

$$\delta\left(\frac{\omega}{2}\right) = 2\delta(\omega)$$

Y de forma general, la propiedad fundamental de escalamiento de la señal impulso, es:

$$\delta\left(\frac{\omega}{a}\right) = |a| \delta(\omega)$$

Teniendo en cuenta lo anterior, el espectro queda como:

$$X_2(j\omega) = \frac{\pi}{2} \{2\delta(\omega - 200\pi) + 2\delta(\omega + 200\pi)\}$$

Y cancelando la constante de 2, tenemos que:

$$X_2(j\omega) = \pi\{\delta(\omega - 200\pi) + \delta(\omega + 200\pi)\}$$

Se observa que al comprimir la señal en el tiempo por un factor de 2, las frecuencias se expanden por el mismo factor, duplicando la frecuencia fundamental. Este resultado es consistente con el espectro de la señal  $\cos(200\pi t)$ , el cual es la versión comprimida de la señal  $\cos(100\pi t)$ , por un factor de  $a = 2$ .

Nota aclaratoria:

En la señal original el periodo es  $T = \frac{2\pi}{100\pi} = 0.02$ ; mientras que, en la señal comprimida el periodo es  $T = \frac{2\pi}{200\pi} = 0.01$ , es decir, la mitad del periodo de la señal original.

#### Ejemplo 40. Dilatación de una señal cosenoidal

Partimos de la misma señal del ejemplo anterior  $x_1(t) = \cos(100\pi t)$ , con  $\omega_0 = 100\pi$ , cuyo espectro es:

$$\cos(100\pi t) \xrightarrow{TF} \pi\{\delta(\omega - 100\pi) + \delta(\omega + 100\pi)\}$$

Ahora, vamos a suponer que esta señal se dilata la señal por un factor  $a = 0.5$ , entonces, la TF de la señal dilatada, que llamaremos  $x_2(t)$  es:

$$x_2(t) \xrightarrow{TF} \frac{\pi}{0.5} \left\{ \delta\left(\frac{\omega}{0.5} - 100\pi\right) + \delta\left(\frac{\omega}{0.5} + 100\pi\right) \right\}$$

El espectro de la señal lo podemos re-escribir así:

$$X_2(j\omega) = 2\pi \left\{ \delta\left(\frac{\omega - 50\pi}{0.5}\right) + \delta\left(\frac{\omega + 50\pi}{0.5}\right) \right\}$$

Aplicando la propiedad de escalamiento de la señal impulso continua, tenemos que:

$$\delta\left(\frac{\omega}{0.5}\right) = 0.5\delta(\omega)$$

Teniendo en cuenta lo anterior, el espectro queda como:

$$X_2(j\omega) = \frac{2\pi}{2} \{ \delta(\omega - 50\pi) + \delta(\omega + 50\pi) \}$$

Y cancelando la constante de 2, tenemos que:

$$X_2(j\omega) = \pi \{ \delta(\omega - 50\pi) + \delta(\omega + 50\pi) \}$$

Se observa que al dilatar la señal en el tiempo por un factor de 0.5, las frecuencias se comprimen por el mismo factor, reduciendo a la mitad la frecuencia fundamental. Este resultado es consistente con el espectro de la señal  $\cos(50\pi t)$ , el cual es la versión dilatada de la señal  $\cos(100\pi t)$ , por un factor de  $\alpha = 1/2$ .

#### 4.6.4. Multiplicación en el dominio del tiempo

Sea  $x_1(t)$ ,  $x_2(t)$ , con TF:

$$x_1(t) \xrightarrow{TF} X_1(j\omega)$$

$$x_2(t) \xrightarrow{TF} X_2(j\omega)$$

Si estas señales se multiplican, así:

$$x_3(t) = x_1(t) \cdot x_2(t)$$

Entonces, la TF de la nueva señal se obtiene como:

$$X_3(j\omega) = \frac{1}{2\pi} [X_1(j\omega) \otimes X_2(j\omega)]$$

Donde  $\otimes$  es el operador de convolución. Es decir, se realiza la convolución discreta entre los espectros de la señales.

De forma equivalente, si el espectro se expresa en Hz:

$$X_3(jf) = [X_1(jf) \otimes X_2(jf)]$$

Es importante notar que, a diferencia de lo que podría esperarse, el producto en el tiempo no produce un producto en frecuencia, sino una convolución.

### Ejemplo 41. Modulación en amplitud

La propiedad de multiplicación de la Transformada de Fourier es fundamental en sistemas de comunicación, particularmente en **modulación en amplitud (AM)**. Multiplicar una señal por una portadora cosenoidal en el dominio del tiempo produce un **desplazamiento de su espectro en frecuencia**, permitiendo transportar información hacia bandas de frecuencia más altas.

Vamos a aplicar esta propiedad, con la multiplicación de dos señales, de la primera conocemos su dominio del tiempo, y de la segunda, su espectro.

Sea,

$$x_1(t) = \cos(100\pi t),$$

y una señal  $x_2(t)$  cuyo espectro es:

- Triángulo centrado en  $\omega = 0$ , de amplitud 1, en el rango de  $-10\pi$  hasta  $10\pi$ , como se presenta en la siguiente figura.

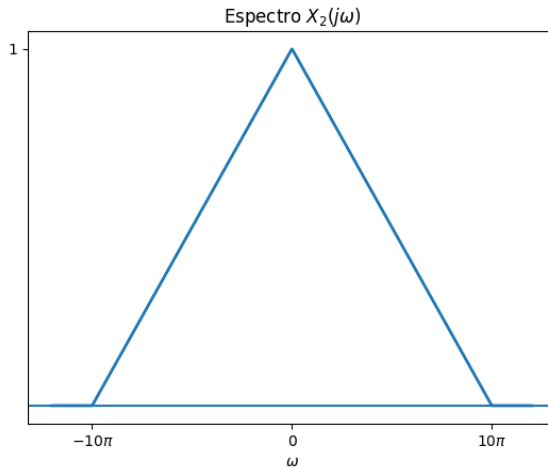


Figura 99. Ejemplo 41: Espectro de la señal  $x_2(t)$ .

Se desea graficar  $X_3(j\omega)$  y  $X_3(jf)$ .

Como paso inicial, vamos a obtener el espectro de  $x_1(t)$ , el cual es:

$$X_1(j\omega) = \pi\{\delta(\omega - 100\pi) + \delta(\omega + 100\pi)\}$$

Entonces, aplicamos la propiedad de multiplicación de la TF, y realizamos la convolución entre cada impulse de  $X_1(j\omega)$  con el espectro  $X_2(j\omega)$ .

Es decir, desplazamos el espectro de  $X_2(j\omega)$  a cada una de las ubicaciones en frecuencia de  $X_1(j\omega)$ , quedando:

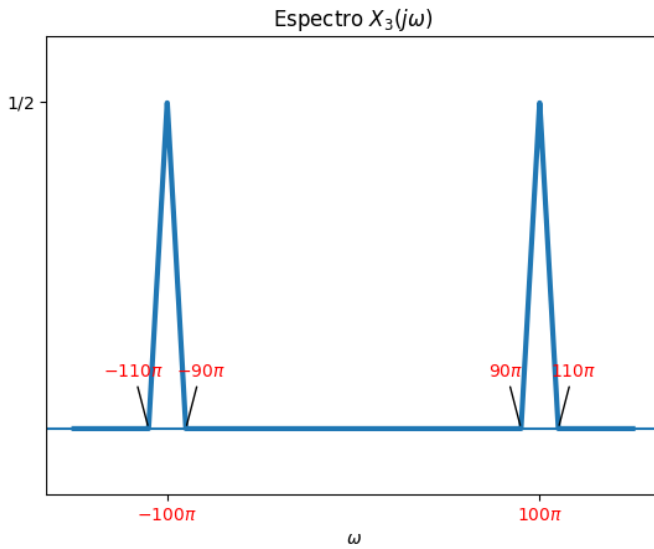


Figura 100. Ejemplo 41: Espectro  $X_3(j\omega)$ .

A manera de resumen de la aplicación de la propiedad de multiplicación:

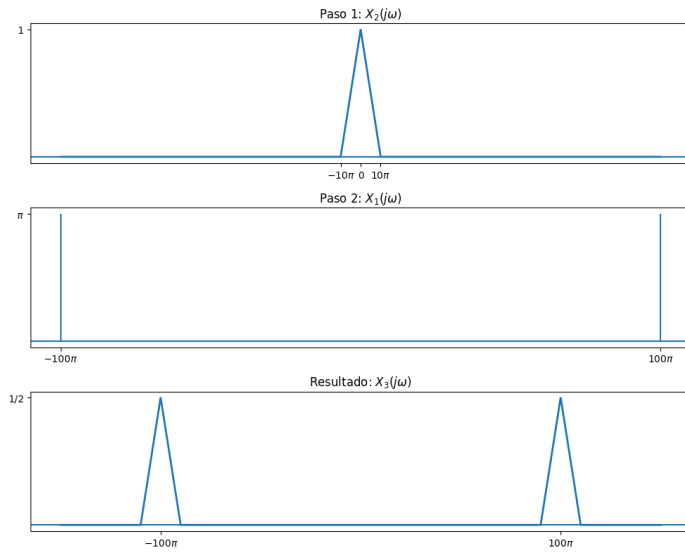


Figura 101. Ejemplo 41: pasos de aplicación de la propiedad de multiplicación.

#### 4.6.5. Convolución en el dominio del tiempo

Esta propiedad es la contraparte de la propiedad de multiplicación de la TF. En este caso, las señales se convolucionan en el tiempo, mientras que, sus espectros se multiplican.

Sea  $x_1(t)$ ,  $x_2(t)$ , con TF:

$$x_1(t) \xrightarrow{TF} X_1(j\omega)$$

$$x_2(t) \xrightarrow{TF} X_2(j\omega)$$

Si estas señales se convolucionan, así:

$$x_3(t) = x_1(t) \otimes x_2(t)$$

Entonces, la TF de la nueva señal se obtiene como:

$$X_3(j\omega) = X_1(j\omega)X_2(j\omega)$$

Ahora, si el espectro lo queremos expresar en [Hz], el resultado en frecuencia se representa así:

$$X_3(jf) = X_1(jf)X_2(jf)$$

**Nota aclaratoria:**

En la propiedad de convolución en el dominio del tiempo, no aparece ningún escalar que diferencie la expresión del espectro entre  $X_3(j\omega)$  y  $X_3(jf)$ , ya que en frecuencia el resultado corresponde a una multiplicación directa.

En contraste, en la propiedad de multiplicación en el dominio del tiempo, si aparece un factor de escala al expresar el espectro en  $\omega$ , debido a que la operación en frecuencia es una convolución, la cual implica una integral que se ve afectada por el cambio de variable entre  $f$  y  $\omega$ .

### 4.6.6. Dualidad

La propiedad de dualidad intercambia los dominios de tiempo y frecuencia, pero requiere especial cuidado en el manejo de constantes y en la interpretación de las variables.

Esta propiedad ya la habíamos trabajado de forma intuitiva en el Ejemplo 37. Ahora la veremos formalmente. En esta propiedad podemos “intercambiar” las señales en el dominio del tiempo y el dominio de la frecuencia, aplicando las siguientes “reglas”:

- Partiendo de  $x(t)$  se reemplaza  $t \rightarrow \omega$  y se multiplica su amplitud por un factor de escala  $2\pi$ . Posteriormente se realiza una inversión en el espectro (este paso se omite si la señal  $x(t)$  era par) y así obtenemos  $x(j\omega)$ .
- Partiendo de  $X(j\omega)$  se reemplaza  $\omega \rightarrow t$ , y así obtenemos  $X(t)$ .

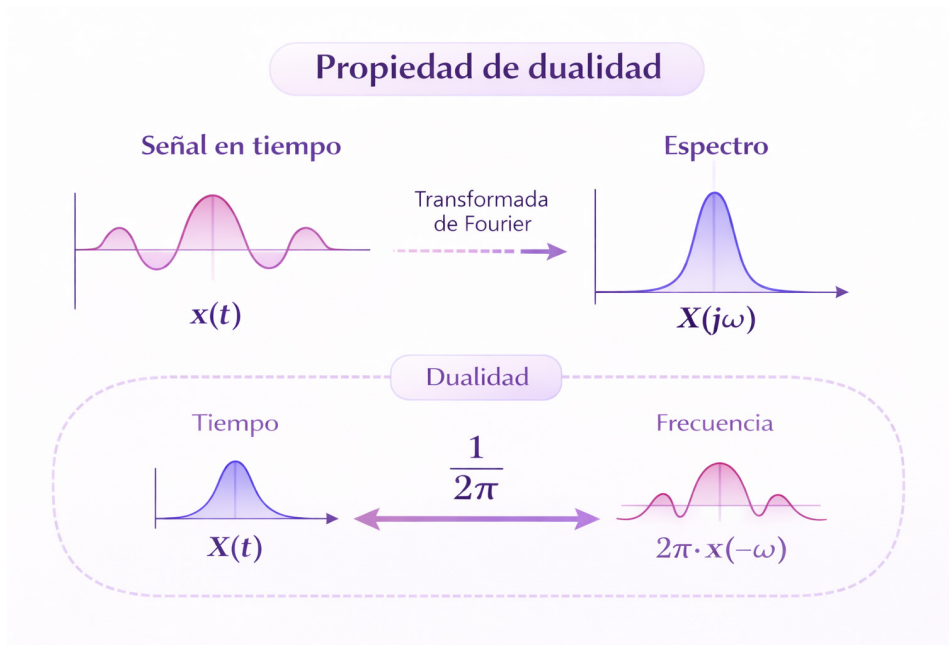


Figura 102. Propiedad de Dualidad.

Nota aclaratoria:

Cuando la señal  $x(t)$  es par, no hay necesidad de realizar inversión en el espectro al momento de sustituir  $t \rightarrow \omega$ .

### Ejemplo 42. Propiedad de dualidad de la señal exponencial bilateral

Queremos identificar la señal en el dominio del tiempo, cuyo espectro es  $X(j\omega) = e^{-a|\omega|}$ . Este espectro es simétrico respecto al origen, es decir, es par.

Partimos de la pareja conocida:

$$x(t) = e^{-a|t|} \xrightarrow{TF} X(j\omega) = \frac{2a}{(a^2 + \omega^2)}$$

Nos damos cuenta que nuestro espectro es una señal exponencial bilateral, pero está en el dominio de la frecuencia, y no del tiempo. Entonces, se hace necesario aplicar la propiedad de dualidad, para realizar “intercambio” de señales.

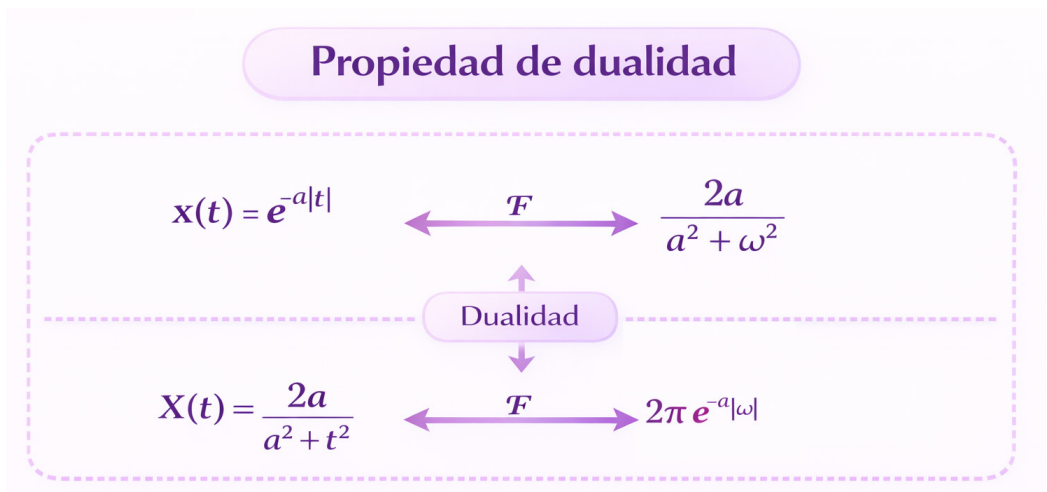


Figura 103. Ejemplo 42: propiedad de dualidad de la señal exponencial bilateral.

Qué pasos realizamos:

- Escribimos en el dominio de la frecuencia la señal exponencial bilateral, multiplicando por  $2\pi$ , es decir, nos queda como  $X(j\omega) = 2\pi e^{-a|\omega|}$ .
- Posteriormente, realizamos el cambio de variable de  $\omega \rightarrow t$ , llegando a  $x(t) = \frac{2a}{a^2+t^2}$
- Dado que la pareja obtenida por dualidad produce  $X(j\omega) = 2\pi e^{-a|\omega|}$ , y el espectro dado es  $e^{-a|\omega|}$ , se requiere un factor de escala  $\frac{1}{2\pi}$  para ajustarlo.

Por lo tanto, la señal en el dominio del tiempo cuyo espectro es  $X(j\omega) = e^{-a|\omega|}$  es:

$$x(t) = \frac{1}{2\pi} \frac{2a}{a^2 + t^2}$$

#### 4.6.7. Relación de Parseval

Nos sirve para calcular la energía total de una señal a partir del dominio del tiempo o del dominio de la frecuencia, por medio de la siguiente ecuación:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(jf)|^2 df$$

O su equivalente cuando el espectro se expresa en [rad/s], así:


$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(j\omega)|^2 d\omega$$

¿De dónde sale ese factor en la Relación de Parseval entre  $X(jf)$  y  $X(j\omega)$ ?

Proviene del cambio de variable:

$$\omega = 2\pi f \quad \Rightarrow \quad d\omega = 2\pi df$$

Cuando pasas de la versión en  $f$  a  $\omega$ , la integral "absorbe" ese  $2\pi$ , y aparece:



¡Una idea clave para entender cómo se mide la frecuencia en radianes por segundo!

Figura 104. Relación de Parseval:

explicación del factor de escala entre  $X(jf)$  y  $X(j\omega)$ .

#### 4.7. Transformada de Fourier utilizando Python

Hasta ahora, hemos trabajado la Transformada de Fourier con señales analíticas y ejemplos controlados. Sin embargo, una de sus aplicaciones más importantes en ingeniería aparece cuando analizamos señales reales, como la voz. En este caso, la TF permite identificar cómo se distribuye la energía de la señal en función de la frecuencia, revelando componentes que no siempre son evidentes al observar únicamente la forma de onda en el tiempo.

En una señal de voz, el dominio del tiempo muestra cómo varía la amplitud a lo largo del tiempo, mientras que el dominio de la frecuencia permite observar qué componentes espectrales están presentes. Esta representación es especialmente útil para estudiar características como el tono, la presencia de armónicos y la concentración de energía en ciertas bandas de frecuencia.

En el caso de señales digitales, como las que se procesan en Python, se trabaja con una versión discreta de la señal, por lo que en la práctica se utiliza la Transformada Discreta de Fourier (DFT), calculada eficientemente mediante la Fast Fourier Transform (FFT).

### Ejemplo 43. Análisis espectral señal de voz con ruido externo tipo tono

Vamos a suponer que tenemos una grabación de una señal de voz, pero que tuvo una interferencia debido a la adición (superposición) de un tono. Queremos conocer tanto el comportamiento en frecuencia del mensaje original, así como, identificar la posición en frecuencia de ese tono. En otras palabras, en el dominio del tiempo tuvimos que:

$$x_{noise}(t) = x(t) + n(t)$$

Donde  $x(t)$  es el mensaje de voz original,  $n(t)$  corresponde a una señal cosenoidal de frecuencia desconocida, y  $x_{noise}(t)$  es la señal resultante, en la cual al escucharla notaremos un “pitido”.

Conociendo la propiedad de linealidad de la TF, sabemos que el espectro de  $x_{noise}$  es la suma lineal del espectro del mensaje de voz original y el espectro de la señal cosenoidal, el cual, como bien sabemos, es un “tono” en su frecuencia fundamenta.

Inicialmente, cargamos el audio e incluimos un botón de reproducción para poderlo escuchar dentro de Colaboratory.

```

import numpy as np
import matplotlib.pyplot as plt
import librosa
from IPython.display import Audio, display

# =====
# 1. Cargar archivo de audio
# =====
audio_path = "audio.wav" # Cambia por el nombre o la ruta de tu archivo
x, fs = librosa.load(audio_path, sr=None)

print("frecuencia de muestro del audio:", fs)

# =====
# 2. Reproducir audio
# =====
display(Audio(x, rate=fs))

```

Figura 105. Ejemplo 43: código en Python para el cargue y reproducción del audio.

Posteriormente, utilizando la librería de numpy, calculamos la fft de la señal, seleccionamos solamente sus frecuencias positivas, calculamos su magnitud, y graficamos.

```

# =====
# 3. Calcular la FFT
# =====
N = len(x)                    # Número de muestras
X = np.fft.fft(x)            # Transformada de Fourier
f = np.fft.fftfreq(N, d=1/fs) # Vector de frecuencias

# =====
# 4. Tomar solo frecuencias positivas
# =====
mask = f >= 0
f_pos = f[mask]
X_pos = X[mask]

# =====
# 5. Magnitud del espectro
# =====
X_mag = np.abs(X_pos) / N

# =====
# 6. Graficar espectro
# =====
plt.figure(figsize=(10, 5))
plt.plot(f_pos, X_mag)
plt.xlabel("Frecuencia [Hz]")
plt.ylabel("Magnitud")
plt.title(r"$|X_{noise}(jf)|$")

plt.grid(True)
plt.show()

```

Figura 106. Ejemplo 43: código en Python para el cálculo de la fft de una señal de voz.

Obteniendo:

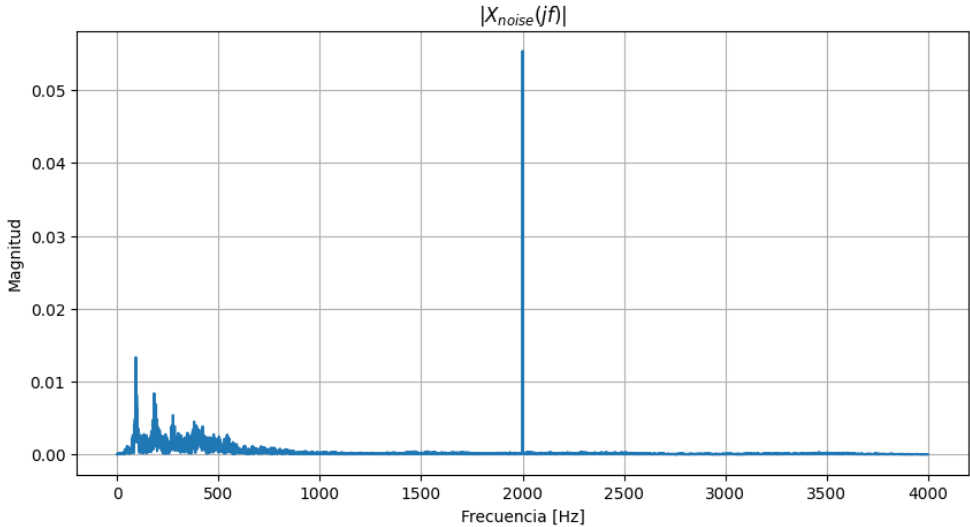


Figura 107. Ejemplo 43: espectro de la señal de voz ruidosa.

A partir de la gráfica del espectro obtenida, podemos apreciar que el tono se ubica alrededor de los 2000 Hz, mientras que el espectro del mensaje de voz original se concentra por debajo de los 800 Hz, aproximadamente.

En este punto, es natural preguntarse:

¿qué utilidad tiene esta representación en frecuencia dentro de un sistema de comunicaciones?

Los sistemas modernos incorporan en su receptor unidades de preprocesamiento que permiten reducir el ruido presente en la señal, como ocurre aquí con el tono interferente. El diseño de filtros digitales para eliminar este tipo de perturbaciones constituye uno de los pilares del Procesamiento Digital de Señales.

Quiero que te quedes con algo más que una gráfica: la intuición de que toda señal puede ser entendida... y también transformada. Lo que acabamos de hacer (escuchar, analizar e identificar una interferencia) es solo el comienzo.

En el próximo semestre daremos ese siguiente paso: diseñar sistemas que modifiquen las señales de manera intencional, por ejemplo, eliminando ese “pitido” que hoy identificamos. Ese camino lo recorreremos en Procesamiento Digital de Señales, apoyándonos en el libro de mi autoría titulado “*Procesamiento digital de señales utilizando Python*” de la editorial Redipe.

Por ahora, cierra este capítulo con la satisfacción de haber aprendido a escuchar los datos... en este caso, literalmente.

