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:

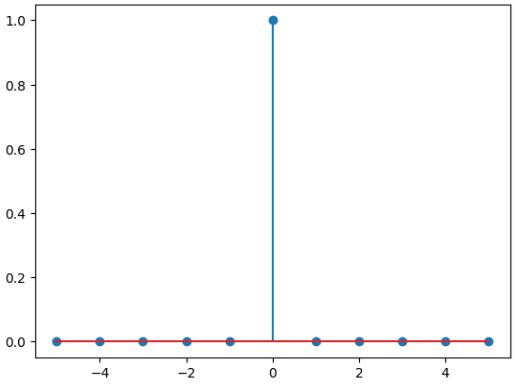


Figura 38. Señal impulso discreta.

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

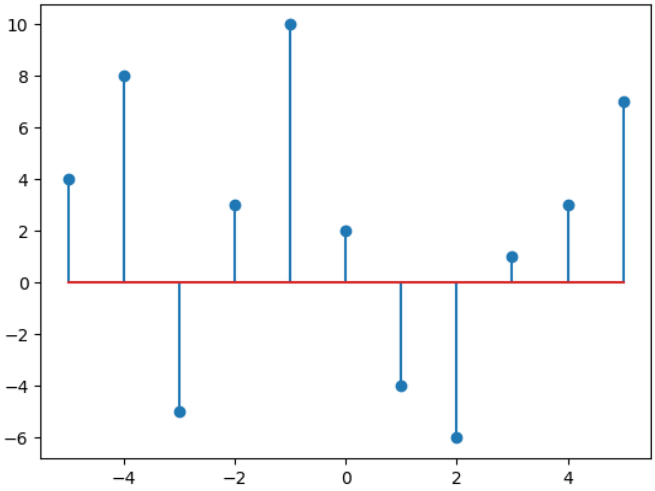


Figura 40. Señal x[n] del Ejemplo 18.

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:

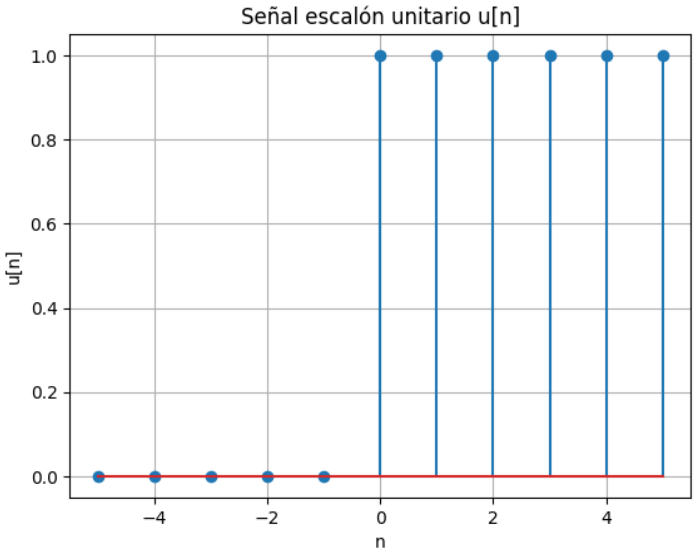


Figura 42. Señal u[n], visualización entre [-5 5].

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 ⊗ 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

$$x[n] \otimes h_1[n] = \{3\delta[n] + 2\delta[n-1] - 4\delta[n-2]\} \otimes \{2\delta[n]\}$$

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

$$= 6\delta[n] + 4\delta[n-1] - 8\delta[n-2]$$

Y el de,

$$x[n] \otimes h_2[n] = \{3\delta[n] + 2\delta[n-1] - 4\delta[n-2]\} \otimes \{-2\delta[n-1]\}$$

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

$$= -6\delta[n-1] - 4\delta[n-2] + 8\delta[n-3]$$

Sumando los dos resultados anteriores, obtenemos:

$$y[n] = 6\delta[n] + 4\delta[n-1] - 8\delta[n-2] - 6\delta[n-1] - 4\delta[n-2] + 8\delta[n-3]$$

$$y[n] = 6\delta[n] + \{4\delta[n-1] - 6\delta[n-1]\} + \{-8\delta[n-2] - 4\delta[n-2]\} + 8\delta[n-3]$$

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

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:

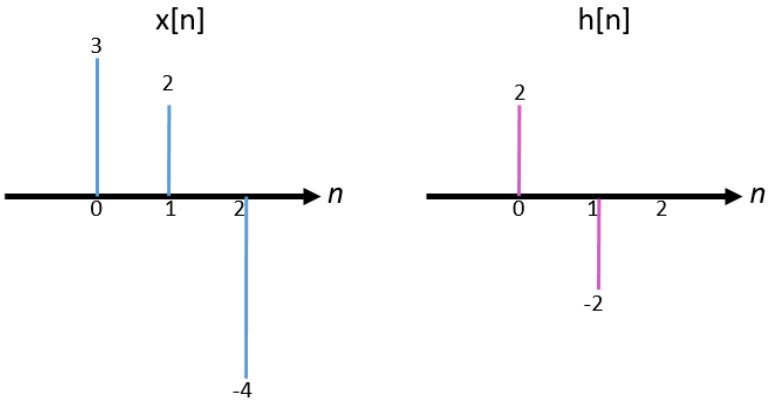


Figura 43. Ejemplo 20, convolución: señal x[n] y h[n].

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

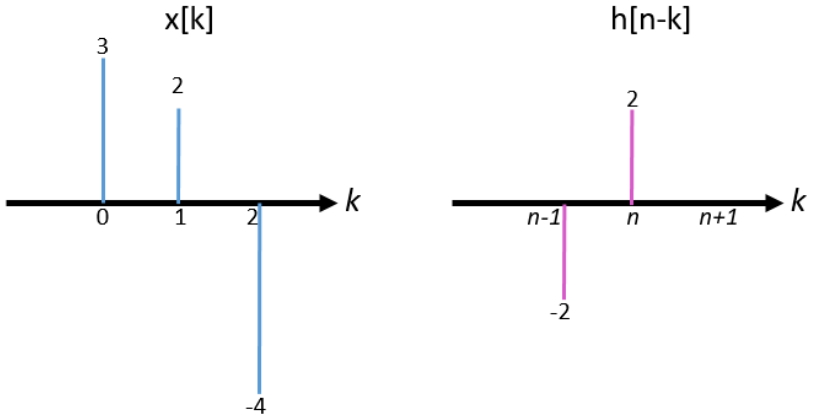


Figura 44. Ejemplo 20: Paso 1 de convolución discreta.

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

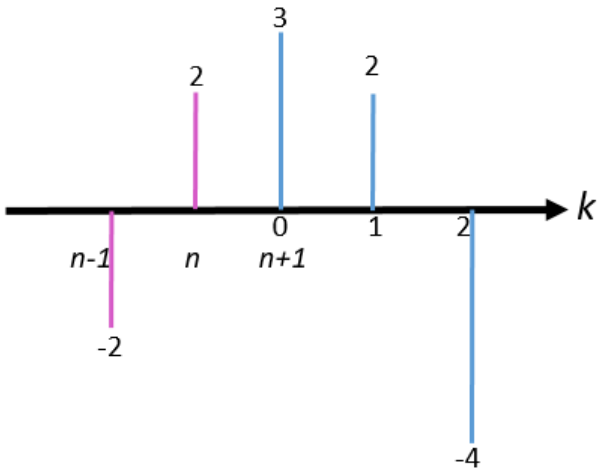


Figura 45. Ejemplo 20: Paso 2 de convolución discreta.

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:

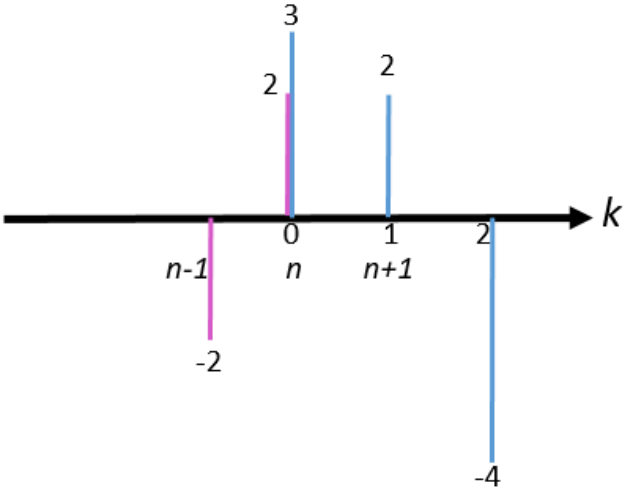


Figura 46. Ejemplo 20: Paso 3 de convolución discreta.

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.

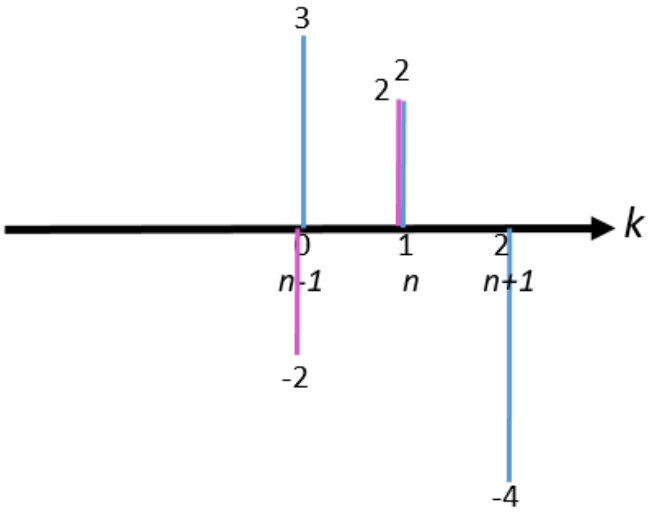


Figura 47. Ejemplo 20: Paso 4 de convolución discreta.

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.

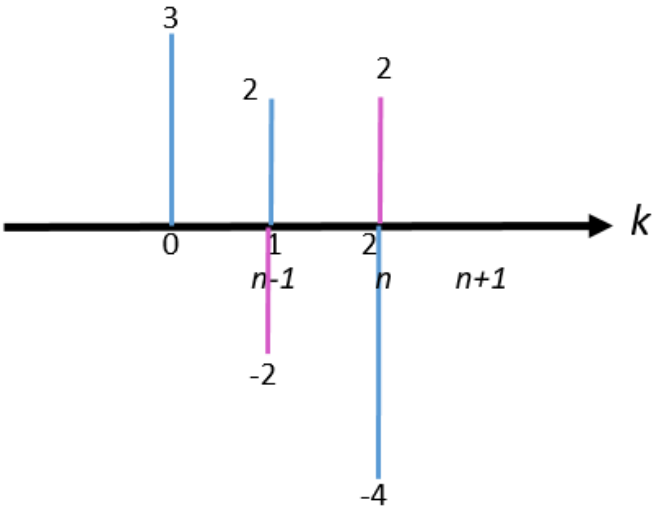


Figura 48. Ejemplo 20: Paso 5 de convolución discreta.

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.

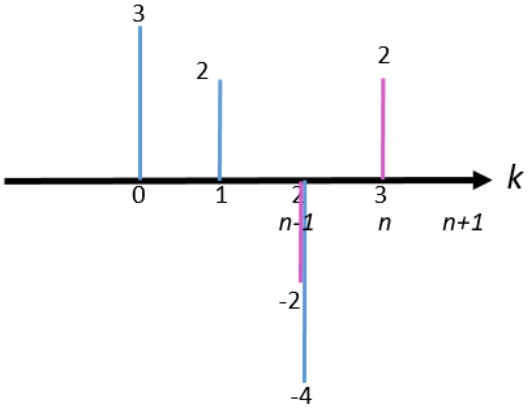


Figura 49. Ejemplo 20: Paso 6 de convolución discreta.

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.

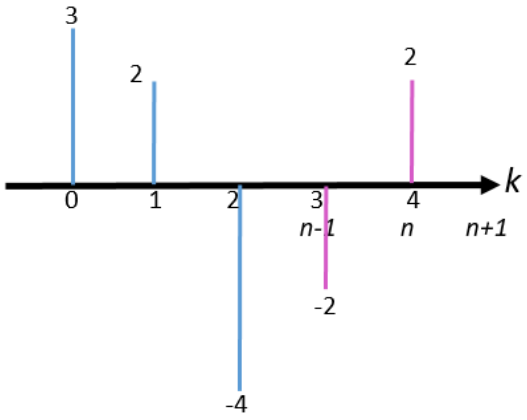


Figura 50. Ejemplo 20: Paso 7 de convolución discreta.

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:

```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:

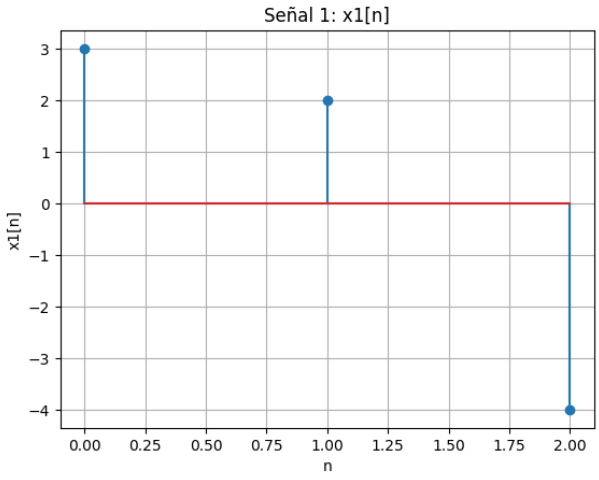
$$n\_conv = np.arange(n1[0] + n2[0], n1[-1] + n2[-1] + 1)$$

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.

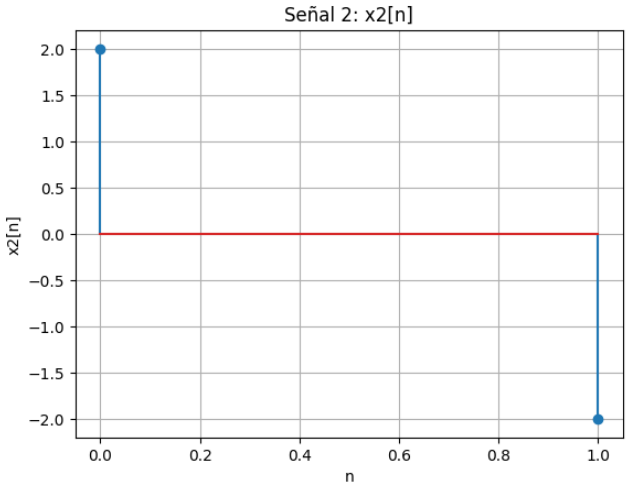
Finalmente, se grafica la señal convolucionada, lo que permite visualizar correctamente su duración y su desplazamiento en el tiempo discreto.

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

A)



B



C

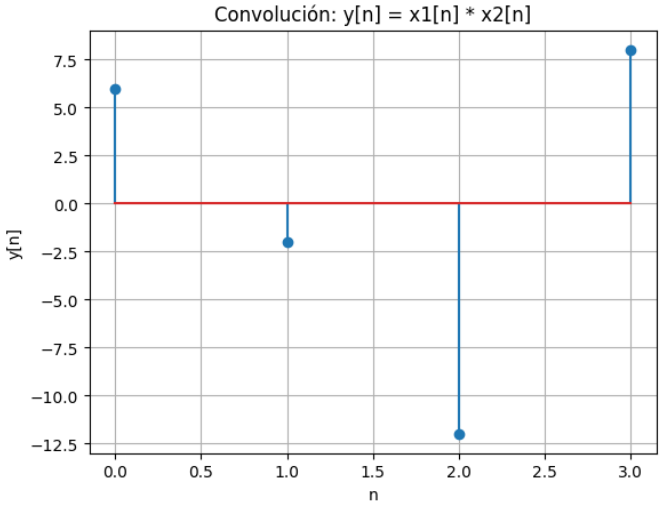


Figura 52. Representación gráfica del Ejemplo 19: a) señal de entrada x1[n], b) señal de entrada x2[n], c) resultado de la convolución, y[n].

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:

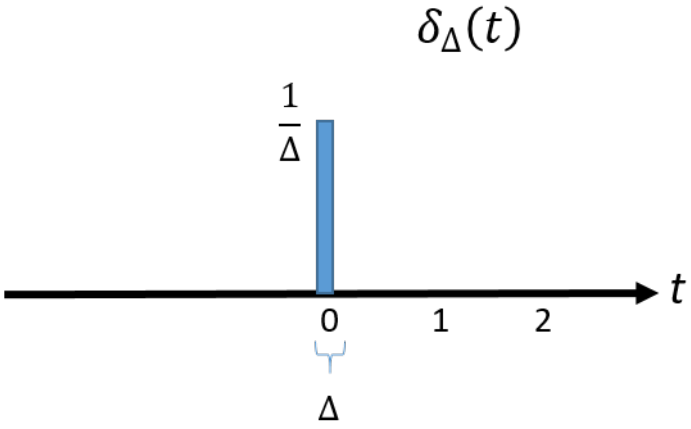


Figura 53. Señal $\delta_\Delta(t)$ en tiempo continuo.

Esta señal existe en un rango de tiempo muy pequeño, denominado Δ, 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 Δ. 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, δ(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.

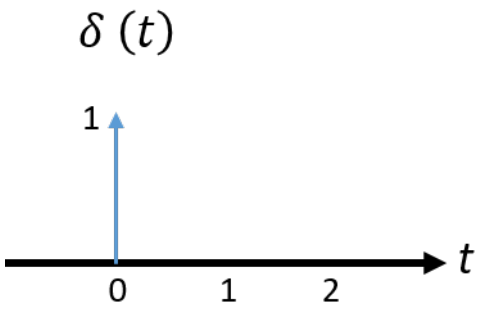


Figura 54. Señal impulso, $\delta(t)$, en tiempo continuo.

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:

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

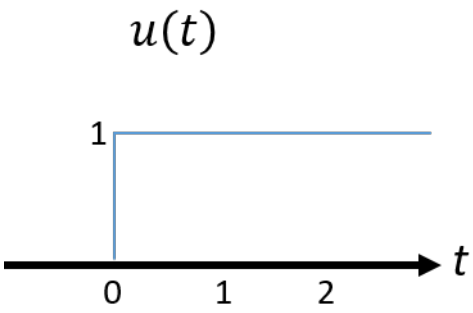


Figura 55. Señal escalón, $u(t)$, en tiempo continuo.

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 $t \to \tau$; mientras que, en la señal $h(t)$ se realiza el cambio de $t \to t - \tau$ (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:

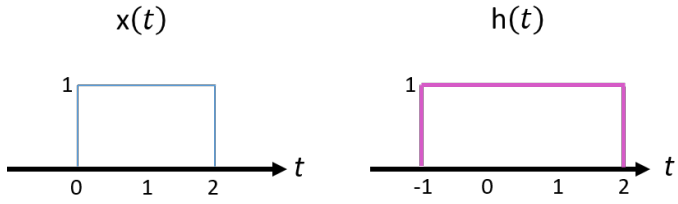


Figura 56. Señal $x(t)$ y $h(t)$: ejemplo de convolución en tiempo continuo.

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

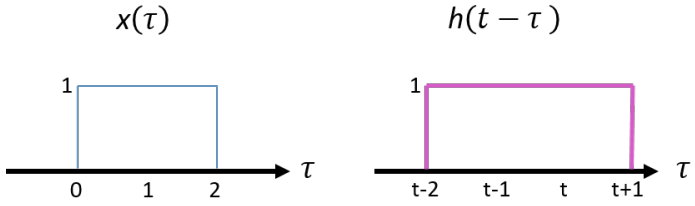


Figura 57. Señal $x(\tau)$ y $h(t-\tau)$: ejemplo de convolución en tiempo continuo.

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 $x(\tau)$.

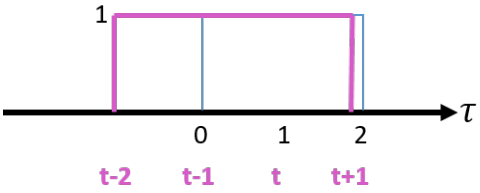


Figura 59. Paso 2: ejemplo de convolución en tiempo continuo.

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 $t+1 < 2 \quad \therefore \quad t < 1$. 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 $x(\tau)$.

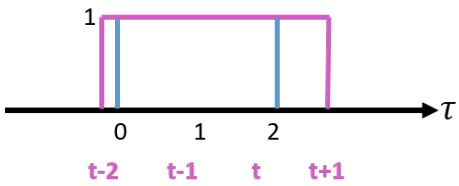


Figura 60. Paso 3: ejemplo de convolución en tiempo continuo.

La salida la obtenemos con la siguiente integral,

$$y(t) = \int_0^2 d\tau = \tau /_0^2 = 2$$

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 $t-2 < 0 \quad \therefore \quad t < 2$. 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 $x(\tau)$.

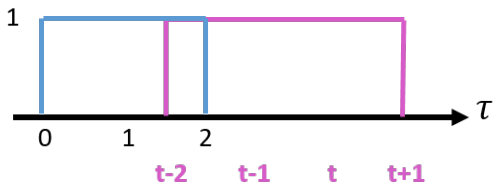


Figura 61. Paso 4: ejemplo de convolución en tiempo continuo.

La salida la obtenemos con la siguiente integral,

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

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 $t - 2 < 2 \quad \therefore \quad t < 4$. 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.

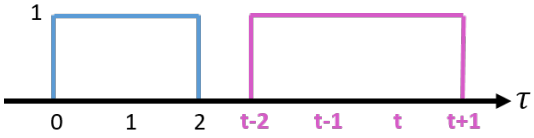


Figura 62. Paso 5: ejemplo de convolución en tiempo continuo.

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

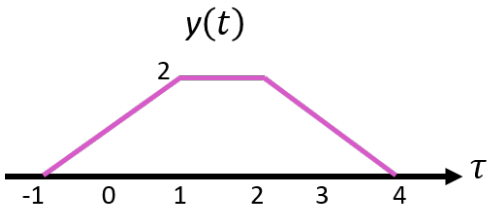


Figura 63. Señal y(t): ejemplo de convolución en tiempo continuo.

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,

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