

# Universidad del Norte

## Laboratorio 2

### Señales y Sistemas - Modelado de Sistemas

**Pablo Martínez, Samuel Solano**

{pmanning, sdsolano}@uninorte.edu.co

**Abstract** – This report shows the detailed process of what has been carried out in the programming and creation of a graphic interface capable of performing convolution between different signals. These signals are exponential signal, sinusoidal signal, triangular signal, rectangular signal and three additional ramp signals. Likewise, the results obtained when using this program.

**Key words-** Convolution, signals, systems.

#### INTRODUCCIÓN

Se denomina convolución al proceso de distribución de una función  $f$ , en cada punto de (y ponderada por) otra función  $g$  (Hecht, 2000) [1], En otras palabras, La convolución es la forma más adecuada y conveniente (en cuanto a procesamiento de señales compete) de operar dos señales, y así mismo, se considera como la base de cualquier sistema de procesamiento digital. Entre las aplicaciones de la convolución se encuentran: Filtrado de señales (ayuda a cualquier tipo de procesamiento más que todo audiovisual), Comprobar la correlación de una señal con otra, encontrar patrones en las señales, entre otras. [2].

Teniendo en cuenta lo anterior y bien conocida la importancia e ideas preliminares a cerca de la convolución, en el presente informe de laboratorio, se tiene la intención de ampliar el concepto de convolución antes mencionado mediante el uso de un programa elaborado en python cuya función sea aplicar el proceso de convolución entre las señales “base” cómo lo

son: Señal exponencial, Señal sinusoidal, Señal triangular, Señal rectangular y en adicional tres señales rampa que se ilustrarán más adelante.

Es bien sabido que Python contiene la opción de realizar convoluciones sin necesidad de crear algoritmo alguno, solo mediante el uso de la función de numpy “np.convolve”, Sin embargo, dado el objetivo ya mencionado de este laboratorio, se requiere de un algoritmo codificado en Python para demostrar la convolución entre todas y cada una de las señales previamente mencionadas, teniendo así la opción de ser utilizadas, graficadas y convolucionadas tanto en su forma continua como en su forma discreta según sea la voluntad del usuario.

#### METODOLOGÍA

##### 1. Interfaz Gráfica

Como parte visual y básicamente la cara del trabajo a realizar, se optó por utilizar la herramienta streamlit, la cual facilita una gran variedad de opciones gráficas aptas para el objetivo de este laboratorio.

Se hace uso de dos apartados para graficar en los cuales se podrá observar en uno la generación de la señal y el sistema original y en el otro el resultado de la convolución

Por otro lado, en cuanto a los inputs del usuario es necesario conocer en un principio la señal y sistema a utilizar, la amplitud de estas, el punto de inicio y de finalización de las mismas (en el tiempo  $t$ ), y entre otros parámetros que varían dependiendo de la señal, como es el caso del seno siendo este de la forma “ $\text{Asen}(\omega t)$ ”.

Para mayor facilidad y minimalismo de la interfaz, se utiliza un dropdown en el caso de la selección de señales y para el resto de parámetros se hace uso de sliders. (ver figura 1)

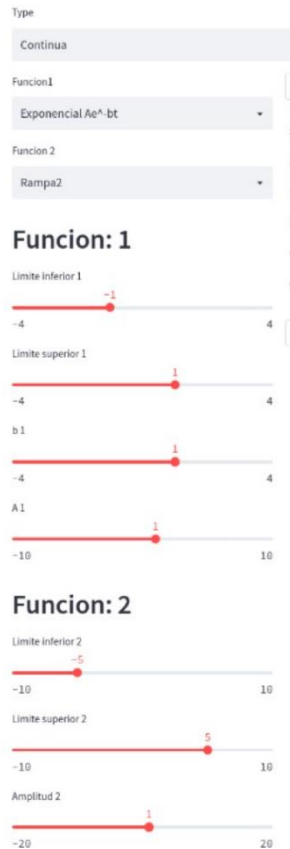


Fig 1. Interfaz del usuario

## 2. Generación de Señales

Teniendo en cuenta que el presente laboratorio puede ser considerado como una secuela o avance del antes realizado Laboratorio 1 “Generación y Operación de Señales”, no representa dificultad el generar la representación gráfica de las señales, sin embargo, se tuvo en cuenta que en este caso se requiere una representación de las señales de forma continua y en adicional discreta, por lo que se realizó un ajuste de código para lograr este requerimiento. Para conseguir lo anterior, bastó con declarar el vector del eje independiente  $t$  siempre teniendo en cuenta el teorema de Nyquist, el cual nos dice que para una reconstrucción precisa de una señal “la frecuencia de muestreo  $f_s$  debe ser mayor que el doble del componente de interés de frecuencia más alto en la señal medida.” [3], para luego digitar la variable en función de  $t$  como eje dependiente a graficar  $y(t)$ .

Ya estando definidas las variables, solo bastó con aplicarle la función “Plot” en el caso de las continuas y “Stem” en el caso de las discretas, sin embargo, a la hora de discretizar las

variables con los ejes antes planteados, no se logra apreciar la buena discretización de las señales dado a la cercanía de los puntos del eje  $t$ , es por esto, que se optó por disminuir el paso del eje, o en otras palabras, hacer un tipo de diezmado o downsampling, que consiste en la reducción de ancho de banda y de la frecuencia de muestreo.

## 3. Algoritmo convolución

Teniendo a la convolución como tema central de este laboratorio, se considera a esta como la parte más importante de la metodología.

Matemáticamente, se entiende la convolución como la sumatoria de las multiplicaciones que surgen al pasar la señal por el sistema:

$$(h * g)(t) = \int_{-\infty}^{\infty} h(\tau)g(t - \tau)d\tau$$

En otras palabras, consiste en mantener una señal en su posición  $\tau$  y voltear la otra para pasarla sobre la estática y realizar multiplicaciones, por muy compleja que pueda parecer la explicación anterior, se creó una función en python, cuyo objetivo sea realizar esta operación, siendo así las variables “r” y “s” las señales a convolucionar:

```
def conv(r,s):
    n=0
    rl=len(r)
    rs=len(s)
    j=n
    i=0
    mult=0
    con=0
    conv=np.zeros(rl+rs)
    while n<rl+rs:
        while j>=0:
            try:
                mult=s[i]*r[j]
                con=con+mult
                j=j-1
```

i=i+1

except:

mult=0

j=j-1

i=i+1

conv[n]=con

i=0

n=n+1

j=n b

con=0

return conv

#### 4. Animaciones

En cuanto a las animaciones sucede lo mismo que con la generación de señales, anteriormente ya se había trabajado con esto. Es por esto que no representa gran grado de dificultad el hacerlas.

Para ello basta con un ciclo “for” cuya función sea graficar la función varias veces en distintos “t” dados por la convolución y así obtener una animación del proceso abordado.

Para mayor comprensión de lo explicado anteriormente, se puede sintetizar en las siguientes líneas de código:

for i in range(11):

rg=t+desp\*i

b=conv(y,y2)

graph(rg,y,t2,y2)

ta2=np.arange(t2[0]+t[0],r1\*i+t2[0]+0.04+t[0],0.04)

lt1=len(ta2)

b[ta>ta2[lt1-1]]=0

graph2(ta,b)

graph2(tb,a)

## RESULTADOS Y ANÁLISIS

Tal como se explicó en la metodología, y teniendo en cuenta el objetivo del presente laboratorio que es representar gráficamente el proceso de la convolución, se logran satisfactoriamente los siguientes resultados:

- En cuanto a la generación de señales se obtiene una alta precisión en lo esperado por el usuario tal como se logra observar en los ejemplos a continuación:

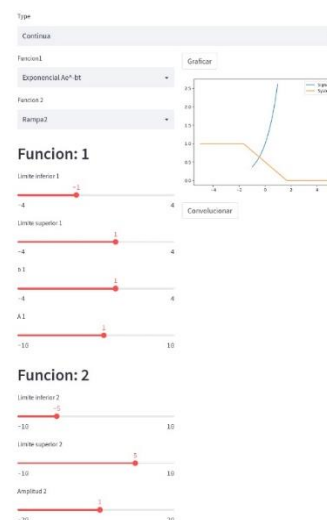


Fig 2. generación de señales continuas

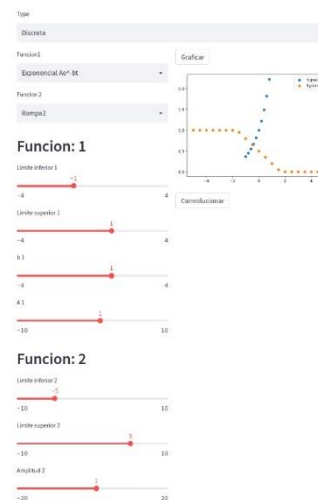


Fig 3. generación de señales discretas

- Por el lado de la convolución y siendo esto la parte principal del lab, se logra obtener satisfactoriamente la gráfica de convolución esperada en todas las combinaciones posibles de señales, aquí se puede visualizar lo anterior:

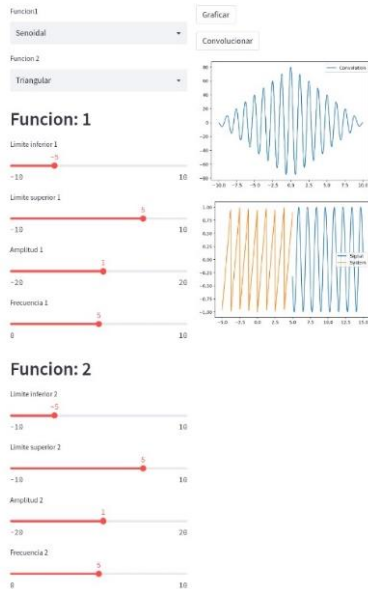


Fig 4. Convolución señales continuas

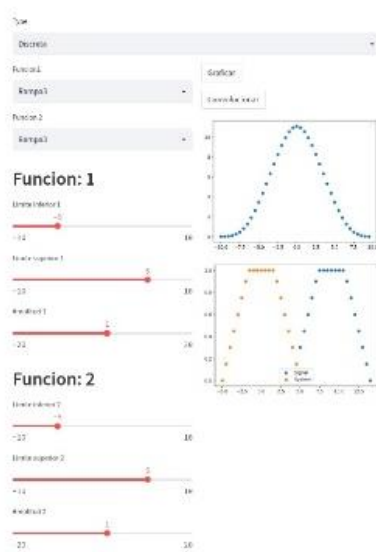


Fig 5. Convolución señales discretas

Cabe resaltar que en el presente documento no se logrará evidenciar lo que concierne a las animaciones. Sin embargo, en la plataforma *github* en la que este se encuentra, se anexa el código para libre prueba y visualización de estas.

## DISCUSIÓN Y CONCLUSIONES

En conclusión, se resalta el éxito de esta práctica al haber logrado generar un código capaz de realizar el proceso de convolución satisfactoriamente en python. Utilizando para ello la teoría antes estudiada sobre este tema y así mismo el algoritmo para convolucionar que se utiliza a la hora de realizar este proceso a mano.

En adicional, se pudo tener un gran acercamiento a diferentes plataformas muy importantes a a nivel de software mundial como lo es streamlit y de igual manera el lenguaje de programación python, ya que resultó muy útil aprender mucho sobre este lenguaje y al mismo tiempo interiorizar conceptos de las señales y sistemas.

Por otro lado, se demostró la “naturaleza” del proceso de la convolución y todos los parámetros que ha de haber detrás de esta para que se pueda realizar correctamente, tal como fue el caso de los ajustes en el eje independiente necesarios para operar de una mejor manera.

Pese a la claridad con la que se ha planteado la presente práctica, no fue exenta de problemas a la hora de su elaboración. La mayor dificultad estuvo más que todo en el planteamiento del algoritmo de convolución, ya que pese a que se tenía una idea de qué debía de hacer, algunos factores como el cuándo parar la multiplicación punto a punto, representaron un reto y a eso sumándole la falta de contacto previo con python y las librerías utilizadas. Más, sin

embargo, se obtuvo un gran rendimiento al final del día.

Como parte previa a la convolución, estuvo la generación de señales, pero tal como se ha mencionado, esta no representó un gran reto, ya que se había realizado algo similar anteriormente. Lo mismo sucedió con la animación, anteriormente se había realizado y por ende gracias a esa experiencia, bastó con adaptar ese concepto al laboratorio en cuestión.

A pesar de los problemas que presentamos, se logró cumplir con los objetivos del laboratorio al mismo tiempo que se adquirieron habilidades y destrezas de programación, se pudo comprender en mayor medida el proceso de la convolución y así mismo afianzar los conceptos y métodos previamente estudiados en la teoría.

#### REFERENCIAS

[1] HECHT E., ÓPTICA. 3ª ED, ESPAÑA: PEARSON, 537-540, 2000.

[2] Data Science Team. "Convolución — DATA SCIENCE". DATA SCIENCE. <https://datascience.eu/es/matematica-y-estadistica/convolucion/>

[3] Engineer Ambitiously.TM. "Adquirir una Señal Analógica: Ancho de Banda, Teorema de Muestreo de Nyquist y Aliasing". Engineer Ambitiously - NI. <https://www.ni.com/es-co/innovations/white-papers/06/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html>