Informe de Laboratorio Señales y Sistemas Modelado de Sistemas

David Vallejo, Xelena Maiguel, and Gerald Mass
davallejo@uninorte.edu.co, Xmaiguel@uninorte.edu.co, gmass@uninorte.edu.co

***Resumen*—In this laboratory report, 5 input signals and their impulsehttps://www.overleaf.com/project/634716961025215cd5d0c856 response will be evidenced, for each one its respective output was found, using the convolution method both in the discrete time domain and in the continuous time domain, where the signals to convolve are: Increasing and decreasing exponential, triangular, sinusoidal, ramp and rectangular.**

***Index Terms*—Continuous, discrete, amplitude, domain, convolution, GUI, time.**

I. Introducción

EL objetivo principal de este laboratorio es desarrollar procesos convolucionales en dominios de tiempo continuo y discreto. Esto se realizó para validar el concepto teórico de convolución de operaciones matemáticas adquirido en el curso de Señales y Sistemas. También es muy importante en el campo de las señales digitales, ya que a partir de la señal de entrada y la respuesta al impulso se puede obtener la señal de salida del sistema. El Laboratorio se divide en diferentes áreas para permitir al lector relacionar esta operación a nivel de computadora con el escenario gráfico. Primero, la Sección II proporciona los conceptos básicos necesarios para comprender más claramente el manejo y el comportamiento de cada tipo de señal (exponenciales ascendentes y descendentes, triángulos, sinusoides, rampas y rectángulos). Su objetivo es promover las habilidades que los ingenieros necesitan en situaciones problemáticas. La Sección III luego detalla los pasos tomados para el diseño, manipulación de la señal y parámetros funcionales (amplitud, puntos de inicio y fin) que son relevantes para la correcta ejecución de la señal.

II. Marco Teórico

II-A. *Convolución*

Una operación matemática con dos funciones, que es la representación más general del proceso de filtrado lineal (invariante). La convolución puede ser aplicada a dos funciones cualesquiera de tiempo o espacio (u otras variables) para arrojar una tercera función, la salida de la convolución. Si bien la definición matemática es simétrica con respecto a las dos funciones de entrada, es común en el procesamiento de las señales decir que una de las funciones es un filtro que actúa sobre la otra función.

II-A1. Convolución continua: Para la convolución continua se parte de la representación matemática de una señal de entrada arbitraria, de tal manera que la salida sea el resultado de la interacción existente entre ella y la respuesta en el tiempo propio del sistema. la señal salida $y(t)$ de un sistema cualquiera, es el resultado del proceso de transformación que este realiza sobre la señal de entrada $x(t)$, representado matemáticamente como $y(t) = \Gamma x(t)$ [1]. Para trabajar en el sistema continuo se presenta la operación conocida como la integral de convolución, cuya notación matemática se muestra en la ecuación. $y(t) = (x * h)(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$

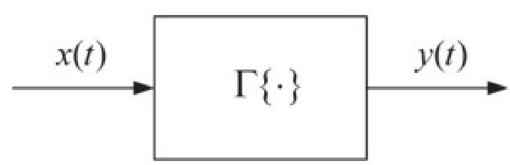

Figura 1. Representación de un sistema continuo.

II-A2. Convolución discreta: Para la convolución discreta la respuesta ante una entrada $x[n]$ será la superposición de las respuestas escaladas del sistema a cada uno de los impulsos desplazados [1]. Para trabajar en el tiempo discreto se conoce como la operación de convolución, dicho de otra manera, la suma de convolución. Matemáticamente se representa como. $y(n) = x[n] * h[h] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$

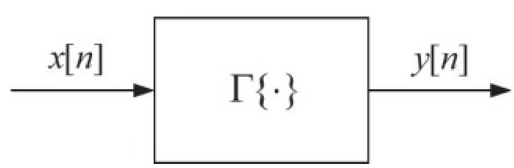

Figura 2. Representación de un sistema discreto.

II-A3. GUI: La interfaz gráfica de usuario (conocida también por su acrónimo GUI), la interfaz gráfica de usuario es un programa informático el cual actúa de interfaz de usuario, mediante el uso de un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz las cuales están guardados mediante código. Su principal uso consiste en proporcionar un entorno visual sencilla y fácil de comprender para el usuario así mismo permite la comunicación con el sistema operativo de una maquina o computador. Habitualmente las acciones se realizan mediante manipulación directa, para facilitar la interacción del usuario con la computadora. La interfaz gráfica de usuario es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

III. Desarrollo

Durante el desarrollo de la práctica 2, se dividirá en dos partes para el desarrollo de las convoluciones teórica y experimentalmente. En esta primera parte de practica de laboratorio se realizará la creación de un algoritmo usando como medio el sistema de MATLAB así mismo se creará una Interfaz Gráfica de Usuario (GUI) en el que se desarrolle el proceso de convolución tanto en el dominio del tiempo discreto como el dominio del tiempo continuo. Con esta interfaz se le permitirá al usuario escoger a su criterio una señal desde un menú desplegable con la posibilidad de introducir los valores necesarios para el tipo de convolución que se desea desarrollar, ya sea tipo discreta o tipo continua. Como se observa en la Figura 1. El diseño de la interfaz tiene que ser simple y fácil de comprender para el usuario; también permite elegir cualquier función (de las que se le permite escoger) y representar gráficamente el proceso de convolución según lo establezca el usuario. En primer lugar, se le solicita al usuario por medio del menú desplegable el dominio de la función en la que desea trabajar; esto se realiza por medio de la primera interacción con la GUI. Seguidamente, se le indica que seleccione la señal que desea anclar al eje horizontal, sea x(t) o x[n], en donde se encuentran las distintas señales:

- Señal triangular.
- Señal cuadrada o rectangular.
- Señal exponencial (Creciente y Decreciente).
- Señal senoidal.
- Señal rampa.

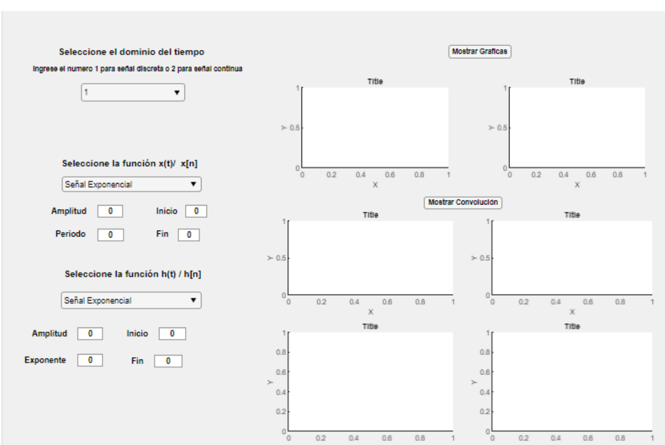

Figura 3. Diseño Interfaz Grafic (GUI).

Como continuación de esta primera parte es necesario digitar los parámetros de las funciones a convolucionar, tales como la amplitud, punto inicial y punto final, etc. Esto para que se genere correctamente la señal diseñada por el usuario.

Este procedimiento se realiza nuevamente para la creación de una segunda señal inicial h(t) o h[n] y finalmente obtener las gráficas originales y en el proceso de convolución en el tiempo. Para la generación de las funciones correspondientes.

III-A. *Señal Triangular*

La onda triangular es un tipo de señal por lo general simétrica (presenta subidas y bajadas constantes) la cual presenta velocidades de subida y bajada constantes. El diseño de esta señal se encuentra clasificado según el dominio de la señal. A través de la instrucción "y = tripuls(t)", la cual devuelve un pulso triangular continuo, aperiódico, simétrico, de altura de unidad en el tiempo de muestreo indicado en el arreglo t, centrado en t = 0, se realizó el desarrollo de la señal. Para el dominio discreto, se ejecuta el siguiente código para la generación de la señal (ver Figura 4.).

```
%-------triangular------------------
if xcon==1
    disp("Hola x_con=1")
    if dom==1
        disp("Hola x_con=1 stem")
        x=Ax*tripuls(tx-(pix+pfx)/2,pfx-pix,0);
        subplot(2,1,1)
        stem(tx,x)
        title("x[n]") %Para untiempo discreto
    end
    if dom==2
        disp("Hola x_con=1 plot")
        x=Ax*tripuls(tx-(pix+pfx)/2,pfx-pix,0);
        subplot(2,1,1)
        plot(tx,x)
        title("x(t)") %Para un tiempo continuo
    end
end
```

Figura 4. Codigo de la señal triangular para x.

Para realizar esta señal se establece a xcon como la primera función a operar (en este caso la señal triangular), se establece a "Ax" como la amplitud de la señal, a "pix" como punto inicial, a "pfx" punto final, el periodo se establece de la siguiente forma "tx = pix : delta : pfx" y delta = 0,0001. De la misma forma, para generar la señal en el dominio continuo se realiza el mismo procedimiento anterior, pero con la diferencia de que delta = 1.

Teniendo en cuenta, que todas las funciones se pueden convolucionar entre sí, se establece el siguiente código para h. Este presenta el mismo funcionamiento que el anterior, solo que se modifica la variable x por la h (también se modifica el nombre de las variables), es decir que el periodo se establece como "th = pih : delta : pfh." Esta segunda función se establece como la respuesta al impulso; siendo h(t) en continuo y h[n] en discreto.

```
if hcon==1
    if dom==1
        h=Ah*tripuls(th-(pih+pfh)/2,pfh-pih,0);
        subplot(2,1,2)
        stem(th,h)
```

```
        title("h[n]")
    end
   if dom==2
        h=Ah*tripuls(th-(pih+pfh)/2,pfh-pih,0);
        subplot(2,1,2)
        plot(th,h)
        title("h(t)")
    end
end
```

III-B. Señal Rectangular

Este tipo de señal alterna su valor entre dos valores máximo y mínimo, sin pasar por los valores intermedios. De igual forma, el diseño de esta señal se encuentra clasificado según el dominio de la señal. En el siguiente código se puede contemplar cuando se genera la señal en tiempo continuo y tiempo discreto; además cuando dicha señal es seleccionada como señal de entrada x(t) o x[n] y respuesta al impulso h(t) o h[n]. En este se tiene en cuenta la definición de las variables establecidas en la sección III-B. Además, se recalca que dom cuando es igual a 1, es porque se está desarrollando en el tiempo discreto y cuando es 2 en el continuo. Además, en este caso se define xcon y hcon como 2 debido a que en el input se le solicito una primera entrada al usuario y la señal cuadrada se encuentra en la opción número 2. La señal está definida como $x = (Ax)(ones(1, length(tx)))$, es decir, un vector de ones con una longitud que es igual a el vector del tiempo, es decir que la longitud de la señal empieza desde el punto inicial hasta el punto final con pasos de delta y esto es multiplicado por la amplitud. Además, para que esta pueda ser observado se pueda ver el impulso de subida y bajada de manera clara, se añade ceros a los costados con el fin de mostrar la señal completa.

```
if xcon==2 || hcon==2
    if xcon==2
        x=Ax*(ones(1,length(tx)));
    end
    if hcon==2
        h=Ah*(ones(1,length(th)));
    end
    if dom==1 && xcon==2
        subplot(2,1,1)
        tc=pix-delta:delta:(pfx+delta);
        xc=[zeros(1,1) x zeros(1,1)];
        stem(tc,xc)
        title("x[n]")
    elseif dom==2 && xcon==2
        subplot(2,1,1)
        tc=pix-delta:delta:(pfx+delta);
        xc=[zeros(1,1) x zeros(1,1)];
        plot(tc,xc)
        title("x(t)")
    end
     if dom==1 && hcon==2
        subplot(2,1,2)
        tc=pih-delta:delta:(pfh+delta);
        hc=[zeros(1,1) h zeros(1,1)];
        stem(tc,hc)
        title("h[n]")
    elseif dom==2 && hcon==2
        subplot(2,1,2)
        tc=pih-delta:delta:(pfh+delta);
        hc=[zeros(1,1) h zeros(1,1)];
        plot(tc,hc)
        title("h(t)")
    end
end
```

III-C. Señal Exponencial

Para esta señal se dividirá en dos señales (señal exponencial decreciente y señal exponencial creciente), inicialmente comenzaremos con la señal exponencial decreciente, esta señal se trabajará de manera similar a las anteriores, esto es debido a que su diseño se encuentra clasificado según el dominio de la señal. A través de la instrucción $exp(X)$, se posibilita la operación exponencial en MATLAB y se establece una función de la forma $x(t) = e^{-bt}$ con la finalidad de estudiar el comportamiento de esta en los distintos posibles parámetros establecidos. En el siguiente código se observa como la señal está definida como $x = exp((-tx)(Ax))$. Para el caso de la función exponencial, se debe tener en cuenta que en el continuo Ax es de tipo e^{-at} y en el discreto $\frac{1^n}{a}$.

```
if xcon==3 || hcon==3
    if dom==1 && xcon==3
        x=exp(-tx*Ax);
        subplot(2,1,1)
        stem(tx,x)
        title("x[n]")
    elseif dom==2 && xcon==3
        x=exp(-tx*Ax);
        subplot(2,1,1)
        plot(tx,x)
        title("x(t)")
    end
```

Así mismo, como todas las funciones se pueden convolucionar entre sí, se establece el siguiente código para h. En donde el periodo $\backslash th = pih : delta : pfh$". Esta segunda función se establece como una respuesta al impulso; siendo h(t) en continuo y h[n] en discreto.

```
    if dom==1 && hcon==3
        h=exp(-th*Ah);
        subplot(2,1,2)
        stem(th,h)
        title("h[n]")
    elseif dom==2 && hcon==3
        h=exp(-th*Ah);
        subplot(2,1,2)
        plot(th,h)
        title("h(t)")
    end
end
```

Ahora continuaremos con la señal creciente, esta señal funciona de manera similar a las anteriores debido a que su diseño se encuentra clasificado según el dominio de la señal. A través de la instrucción $\backslash exp(X)$”, se posibilita la operación exponencial en MATLAB y se establece una función de la forma $x(t) = e^{bt}$ con la finalidad de estudiar el comportamiento de esta en los distintos posibles parámetros establecidos. En el siguiente código se observa como la señal está definida como x = exp((tx)(Ax)). Para el caso se debe tener en cuenta que en el continuo Ax es de tipo e^{-at} y en el discreto $\frac{1^n}{a}$.

```
if xcon==4 || hcon==4
    if dom==1 && xcon==4
        x=exp(tx*Ax);
        subplot(2,1,1)
        stem(tx,x)
        title("x[n]")
    elseif dom==2 && xcon==4
        x=exp(tx*Ax);
        subplot(2,1,1)
        plot(tx,x)
        title("x(t)")
    end
    if dom==1 && hcon==4
        h=exp(th*Ah);
        subplot(2,1,2)
        stem(th,h)
        title("h[n]")
    elseif dom==2 && hcon==4
        h=exp(th*Ah);
        subplot(2,1,2)
        plot(th,h)
        title("h(t)")
    end
end
```

III-D. *Señal Rampa*

Para este tipo de señal se tiene en cuenta que en un periodo de tiempo t, la señal se divide en tres partes o intervalos, en el primer intervalo de la señal vale cero, en el segundo intervalo de la señal es una señal rampa que llega a uno, despues, la señal sigue constante en uno en el tercer intervalo. Para este caso se toma que la señal inicia en t y es cero hasta cero menos y es una rampa hasta uno en el intervalo de cero a uno, para los numeros mayores de uno la señal rampa se mantiene constante en uno hasta el final de t.

```
t=-5:5;
m=1;
r1=(t>=0).*t*m;
r2=(t>=1).*(t-1)*-m;
rt=r1+r2;
plot(t,rt,'b ','LineWidth',2)
grid on
hold on
```

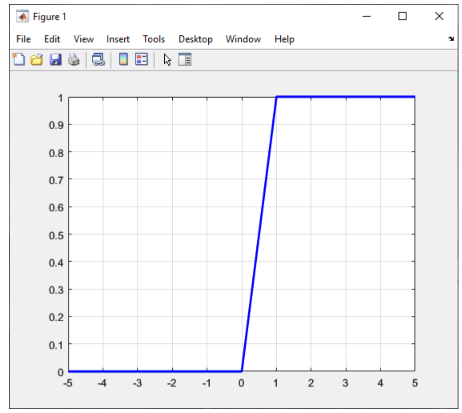

Figura 5. representacion grafica de la señal rampa

III-E. *Señal senoidal*

Para esta señal se realizará una señal senoidal la cual se extenderá a través del tiempo, pero se establecerán los limites de dicha señal como su punto de inicio su punto de terminación o punto final, así como su frecuencia y la amplitud, garantizando una buena visualización de la señal como se observa en la Figura 6.

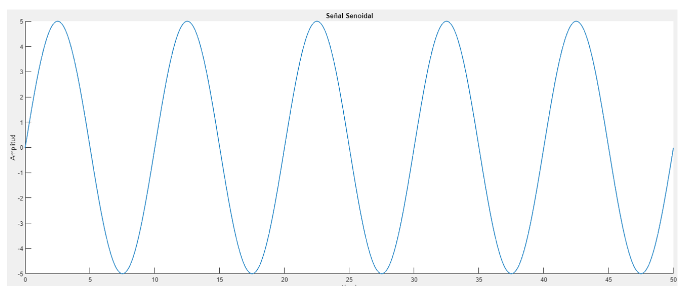

Figura 6. representacion grafica de la señal senoidal

III-F. *Proceso de convolución*

Una vez se tienen definidas todas las funciones, se establece la variable de control del for como “i” y se tiene en cuenta que es aquella que aumenta a medida que cambie el ciclo como se muestra de la siguiente forma.

```
for i=0:lx+lh-2
```

Después de esto como se observa en el siguiente código, se define la señal de las seleccionadas en términos de K o tao y se delimita la función fija x[k] o x(tao) dependiendo del dominio. Además, se utiliza la función zeros de MATLAB, con la finalidad de crear, ampliar y rellenar el vector para que al operar la función h se refleje, desplace y tenga lugar la nueva.

```
        pause(0.5)
%----defino la señal x en terminos de K o Tao
    x_k=[zeros(1,lh*1/delta)
    x zeros(1,lh*1/delta)];
    subplot(221)
    if dom==1
        stem(eje,x_k)
```

```
    title("x[k]")
elseif dom==2
    plot(eje,x_k)
    title("x(tao)")
end
```

Ahora a continuación, en el siguiente código se genera un corrimiento sobre la señal reflejada y se obtiene $h[n-k]$ o $h(t-tao)$. En él, se invertirá la función y se desplazara; luego se realizara el mismo proceso anterior con relación a la función zeros, la única diferencia, en este caso, esta en que se verá afectada por la variable del control for i con la finalidad de que el desplazamiento sea simultáneo con la función.

```
 h_menosk=[zeros(1,(i+1)/delta)
 h(end:-1:1) zeros(1,(lx+lh-1-i)/delta)]
subplot(223)
if dom==1
    stem(eje,h_menosk)
    title("h[n-k]")
elseif dom==2
    plot(eje,h_menosk)
    title("h(t-tao)")
end
```

Por otro lado, se continuara con el proceso, realizando la multiplicación de las funciones que están involucradas, es decir $x[k]$ y $h[n-k]$ para el dominio discreto o $x(tao)$ y $h(t-tao)$ para el dominio continuo como se muestra en el siguiente código. Y, además, define la convolución como la suma de los productos en cada uno de los puntos planteados en el mismo código.

```
%-defino la multiplicacion de x_k y h n-k
    subplot(222)
    if dom==1
        stem(eje,(x_k.*h_menosk));
        title("x[k]h[n-k]")
        xlabel("k")
    else
        plot(eje,(x_k.*h_menosk));
        title("x[tao]h[t-tao]")
        xlabel("k")
    end
```

Finalmente, para que se puedan generar las gráficas correspondientes al sistema, se hace uso del siguiente código, siendo la instrucción stem para el dom=1 que es tiempo discreto y la instrucción plot para el dom=2 que es tiempo continuo.

```
 %------defino la convolucion-------
y(i+1)=sum(x_k.*h_menosk);
subplot(224)
if dom==1
    stem((pix+pih):(pfx+pfh),
    [y zeros(1,lx+lh-2-i)])
    title("y[n]")
elseif dom==2
    plot((pix+pih):(pfx+pfh),
    ([y zeros(1,lx+lh-2-i)])*delta)
    title("y(t)")
    end
```

III-G. Graficacion del proceso de convolucion

Como una muestra de lo aprendido y explicado anteriormente realizaremos un ejemplo en el que mostraremos su proceso de convolución en el que se graficara las señales tanto en el tiempo discreto como en el tiempo continuo, comenzaremos inicialmente con el tiempo discreto. Para este ejemplo escogeremos una señal triangular y una señal rectangular, La señal de entrada x[n] la respuesta al impulso h[t] pueden ser observadas en la Figura 6.

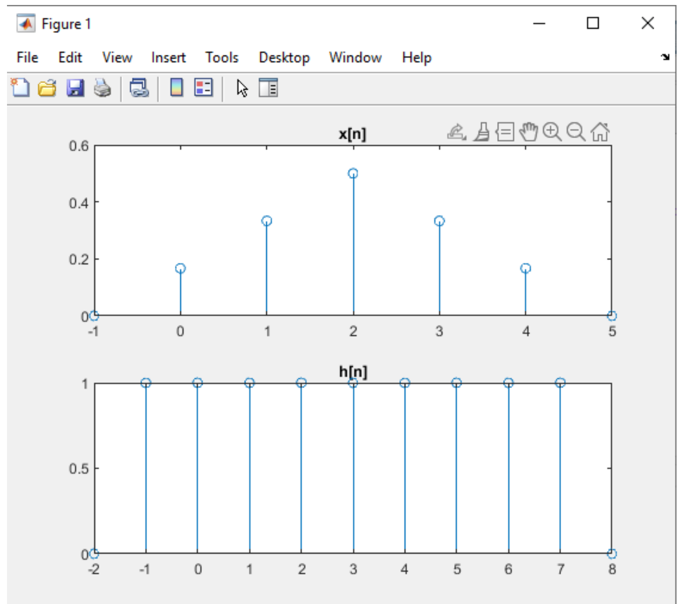

Figura 7. Señales de entrada en discretas.

A continuación, se mostrará todo el proceso de convolución, en donde la señal se trasladará en torno al eje y, se mueve hasta terminar la convolución de las dos graficas. En la Figura 7, se observa cuando aún no se ha comenzado el proceso de convolución.

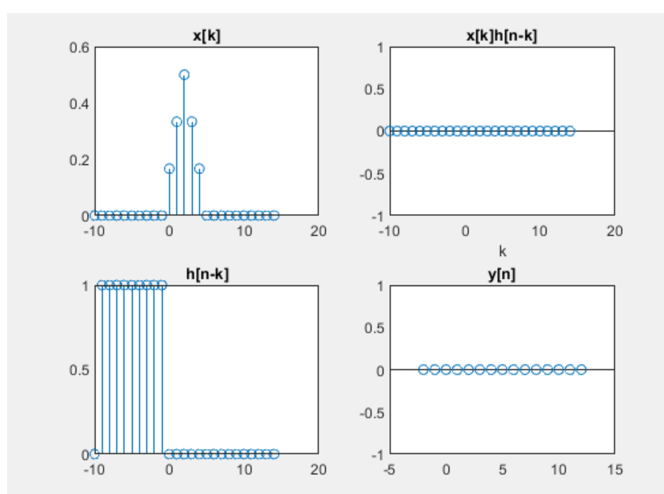

Figura 8. Inicio del proceso de convolucion.

Se puede observar a la señal x[k] la cual es la señal triangular y la señal h[n-k] muestra la señal rectangular invertida siendo esta ultima la que se moverá para realizar la convolución asimismo en la gráfica x[k]h[n-k] nos mostrara cuando existen valores para comparar durante el traslado de la señal h[n-k], finalmente en la grafica y[n] nos mostrara la convolución de ambas señales, en la Figura 8 se puede observar el proceso de la convolución terminada.

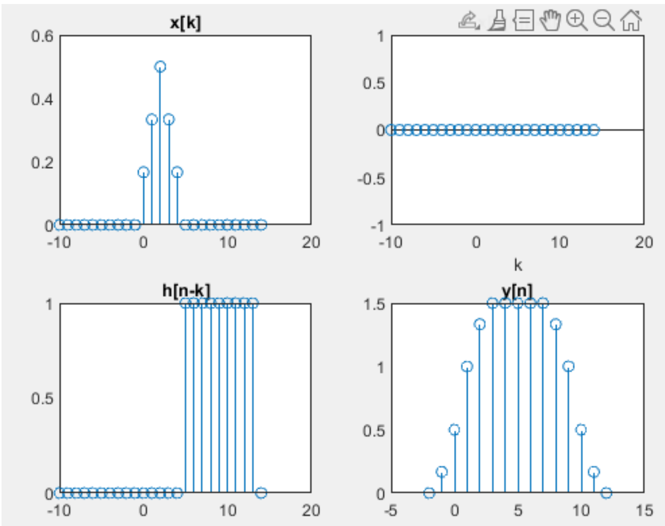

Figura 9. Final del proceso de convolucion.

A continuación, realizaremos el mismo proceso de convolución utilizando las mismas graficas con los mismos parámetros (una señal triangular y una señal rectangular) y realizaremos el proceso de convolución en el tiempo continuo, La señal de entrada x[n] y la respuesta al impulso h[t] se pueden observar en la Figura 9.

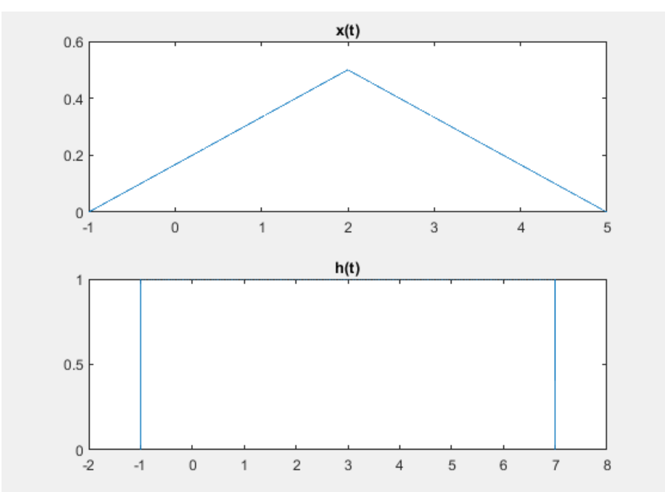

Figura 10. Final del proceso de convolucion.

A continuación, se mostrará todo el proceso de convolución, en donde la señal se trasladará en torno al eje y, se mueve hasta terminar la convolución de las dos graficas. En la Figura 10, se observa cuando aún no se ha comenzado el proceso de convolución.

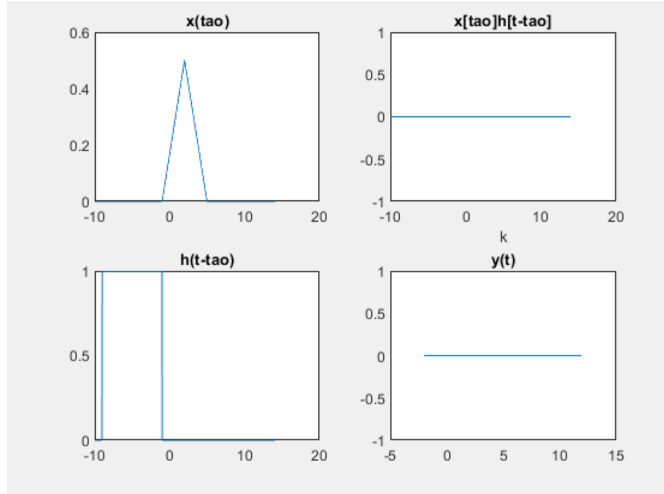

Figura 11. Inicio del proceso de convolucion.

Finalmente se puede observar al igual que en la convolucion son discretas la señal x[k] la cual es la señal triangular y la señal h[n-tao] muestra la señal rectangular invertida siendo esta ultima la que se moverá para realizar la convolución asimismo en la gráfica x[k]h[n-tao] nos mostrara cuando existen valores para comparar durante el traslado de la señal h[n-tao], finalmente en la grafica y[n] nos mostrara la convolución de ambas señales, en la Figura 11 se puede observar el proceso de la convolución terminada.

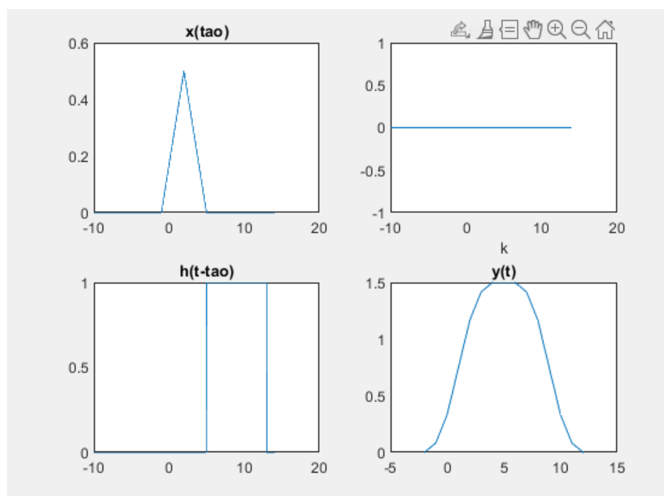

Figura 12. Final del proceso de convolucion.

IV. Conclusiones

En la primera parte de este laboratorio se construye un interfaz donde se puede ver el proceso entero de convolución gráfico. Para esto, se definen 5 tipos de señales, una triangular, una cuadrada, una exponencial, una sinusoidal y una rampa. Las anteriores señales se usan a pares para poder hacer la convolución, usando una como la respuesta de impulso y otra como la señal inicial. Para poder realizar este proceso, se crea un bucle el cual recorre la suma de las longitudes de ambas señales. En este bucle se grafica la señal que se deja anclada, la señal que se desplaza, el producto de las convolución y finalmente la suma de estos productos. Además, se agregan ceros a los lados de la señal para graficar así el desplazamiento de la misma y poder ver de manera más clara todo el proceso, para esto se usan vectores creados con la función zeros, los cuales van cambiando su tamaño dependiendo de la variable de control del bucle. Con el fin de evidenciar la convolución, se realiza la integral de convolución como se ve en el marco experimental, esto con el fin de poder ingresar dicha señal a matlab y poder graficarla, finalmente en un subplot se grafican la señal manual y la generada con matlab, la cual se multiplica por delta, para tener na mayor precisión.

REFERENCIAS

[1] J. Tello Portillo, Introduccion a las senales y sistemas. Baranquilla (Colombia): Universidad del Norte Editorial, 2017.