Suma de convolución

Edgar Tello Paleta

13 de diciembre de 2016

Índice general

1.1 Objetivo de aprendizaje

El alumno comprenderá cómo un sistema discreto lineal e invariante (SDLI), puede ser caracterizado mediante su respuesta al impulso unitario con ayuda de la convolución entre secuencias discretas.

1.2 Introducción Teórica

Sistemas de tiempo discreto

Un sistema de tiempo discreto se define matemáticamente como una transformación que mapea una secuencia de entrada $x[n]$ en una secuencia de salida $y[n]$. Lo cual se puede denotar mediante la ecuación

$$y[n] = T\{x[n]\}, \tag{1}$$

y se puede representar gráficamente mediante la figura 1.

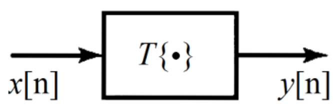


Figura 1. Representación de un sistema discreto.

Sistema lineal de tiempo discreto

Un sistema discreto T es lineal, si y sólo si cumple con el principio de superposición:

$$T(a_1x_1[n] + a_2x_2[n]) = a_1T(x_1[n]) + a_2T(x_2[n]) \tag{2}$$

para cualesquiera señales de entrada $x_1[n]$ y $x_2[n]$, y cualesquiera constantes arbitrarias a_1 y a_2.

Sistema invariante de tiempo discreto

Un sistema discreto T es invariante en el tiempo, si y sólo si

$$y[n] = T\{x[n]\} \tag{3}$$

implica que

$$y[n-k] = T\{x[n-k]\} \tag{4}$$

para toda señal de entrada $x[n]$ y todo desplazamiento k en el tiempo.

1.2.1 Respuesta al impulso unitario de sistemas SDLI y la suma de convolución

Cualquier secuencia se puede expresar como una suma de impulsos unitarios escalados y desplazados mediante la ecuación

$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{5}$$

donde

$$\delta[n-k] = \begin{cases} 1, & n = k, \\ 0, & n \neq k. \end{cases} \tag{6}$$

De acuerdo a la ecuación (1), la respuesta de un ***sistema T de tiempo discreto***, a la secuencia $x[n]$ representada mediante la ecuación (5), está dada por la ecuación

$$y[n] = T\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\} \tag{7}$$

Si el sistema *T* es ***lineal***, entonces de acuerdo con la ecuación (2) se tiene que:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]\, T\{\, \delta[n-k]\, \} \tag{8}$$

Si además, el sistema T es ***invariante*** y se considera que $h[n] = T\{\delta[n]\}$, entonces se tiene que

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n] \tag{9}$$

La ecuación anterior supone la definición de la suma convolución entre las secuencias $x[n]$ y $h[n]$. La ecuación (9) se ilustra en la figura 3, considerando que x[n] y h[n] son las mostradas en las figuras 2a y 2b, respectivamente.

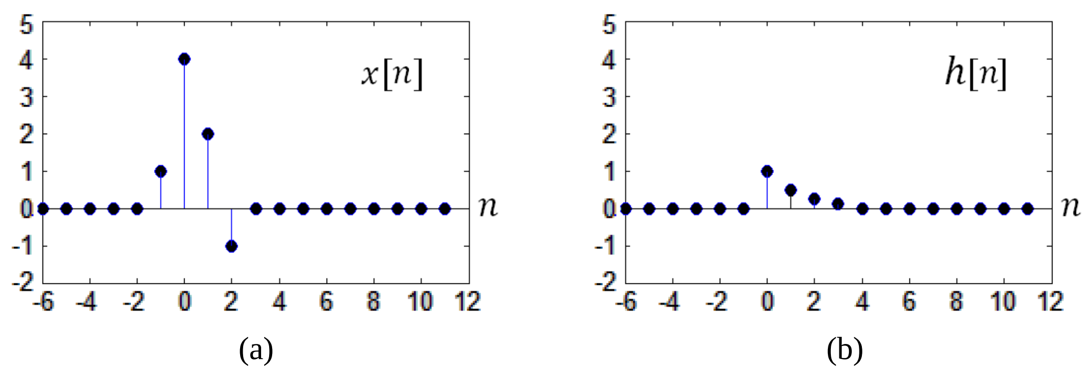
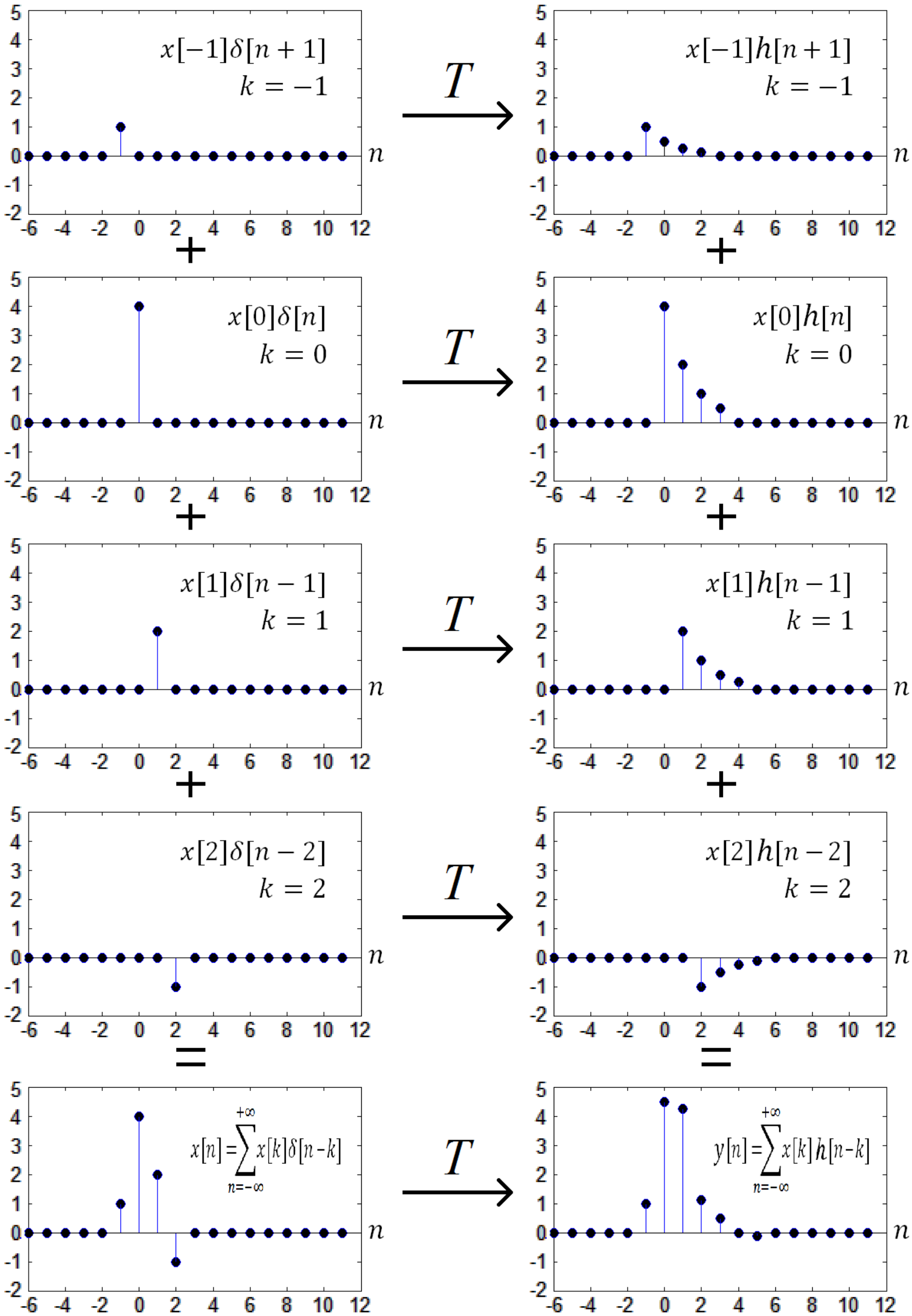


Figura 2. (a) Señal x[n]. (b) Respuesta al impulso de un sistema ***T*** SDLI.



(a) (b)

Figura 3. (a) x[n] según la ecuación (5). (b) y[n] según la ecuación (9).

1.3 Material y equipo

- Computadora con programa Matlab.

1.4 Actividad de investigación previa

1. Dada la secuencia $x[n] = 1\delta[n] + 2\delta[n-1] + 1\delta[n-2] + 2\delta[n-3]$ y el SDLI con respuesta al impulso $h[n] = 4\alpha^n(u[n] - u[n-3])$, $\alpha = 0.5$:

   1.1 Exprese a $x[n]$ y $h[n]$ de la forma:

$$x[n] = \{\ \underset{\uparrow}{x[0]},\ x[1],\ x[2],\ x[3]\ \}, \qquad h[n] = \{\ \underset{\uparrow}{h[0]},\ h[1],\ h[2]\ \},$$

   Represente gráficamente a $x[n]$ y $h[n]$ en las figuras 4(a) y 4(b), respectivamente.

   1.2 Calcule $y[n] = x[n] * h[n]$ utilizando la ecuación (9):

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[-k] \quad = \sum_{k=-2}^{0} x[k]h[-k]$$
$$y[1] = \sum_{k=-\infty}^{\infty} x[k]h[1-k] = \sum_{k=-1}^{1} x[k]h[1-k]$$
$$y[2] = \sum_{k=-\infty}^{\infty} x[k]h[2-k] = \sum_{k=0}^{2} x[k]h[2-k]$$
$$y[3] = \sum_{k=-\infty}^{\infty} x[k]h[3-k] = \sum_{k=1}^{3} x[k]h[3-k]$$
$$y[4] = \sum_{k=-\infty}^{\infty} x[k]h[4-k] = \sum_{k=2}^{4} x[k]h[4-k]$$
$$y[5] = \sum_{k=-\infty}^{\infty} x[k]h[5-k] = \sum_{k=3}^{5} x[k]h[5-k]$$

   Exprese $y[n]$ de la forma:

$$y[n] = \{\ \underset{\uparrow}{y[0]},\ y[1],\ y[2],\ y[3],\ y[4],\ y[5]\ \}$$

   Represente gráficamente a $y[n]$ en la figura 4(c).

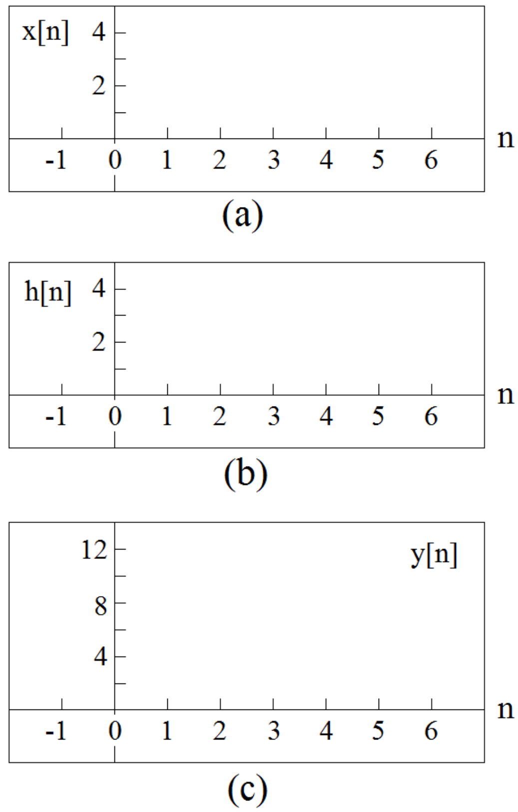


Figura 4.

2. Analice el código de Matlab mostrado en la figura 5. Éste implementa la convolución entre los arreglos x y h. En la figura 6 se ilustra la operación del código mostrado en la figura 5.

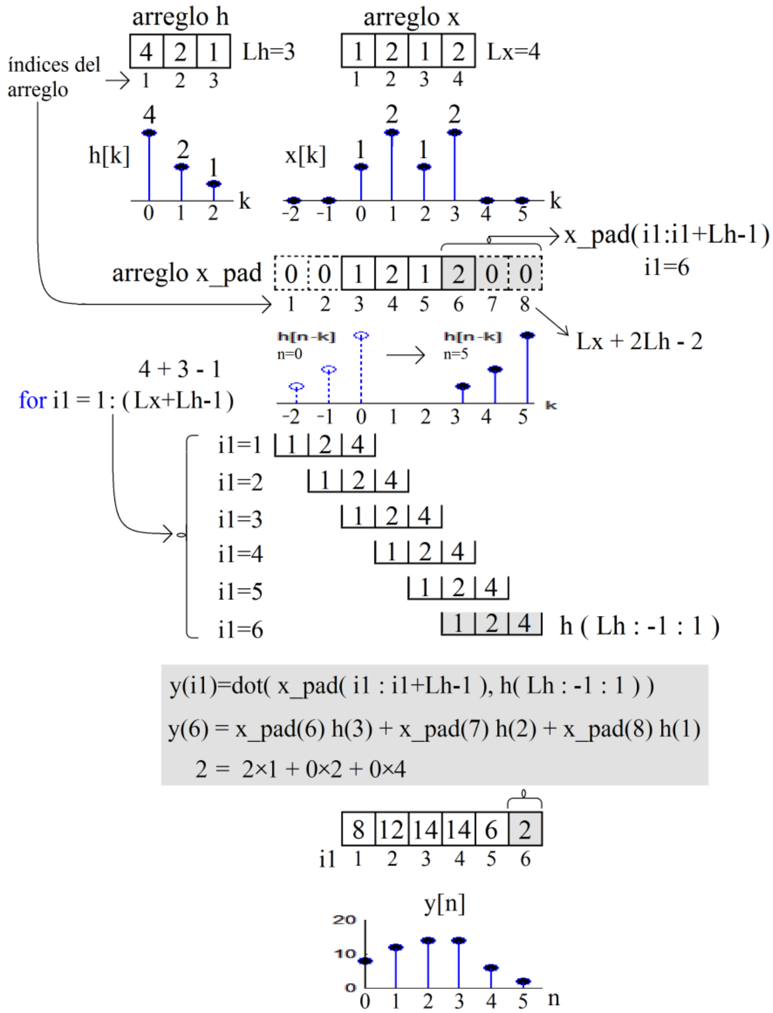
```
function y=Asys_conv1(x,h)

Lx = max(size(x)); %Longitud del arreglo x
Lh = max(size(h)); %Longitud del arreglo h

x_pad =zeros( 1, Lx+2*Lh-2 );
x_pad( Lh: Lh+Lx-1 ) = x;
y=zeros( 1, Lx+Lh-1 );

%implementación de la ecuación (9) para las secuencias
%x y h de longitud finita
for i1 = 1 : ( Lx+Lh-1 )
    y(i1)=dot( x_pad( i1:i1+Lh-1 ), h( Lh:-1:1 ) );
end
```

Figura 5. Función en Matlab para la convolución de 2 arreglos



$$y[5] = \sum_{k=-\infty}^{\infty} x[k]h[5-k] = \sum_{k=3}^{5} x[k]h[5-k]$$

$$y[5] = x[3]h[2] + x[4]h[1] + x[5]h[0]$$

$$y[5] = 2\times 1 + 0\times 2 + 0\times 4 = 2$$

Figura 6. Ilustración del programa en la figura 5.

1.5 Desarrollo

1.5.1 Experimento 1:

Convolución "conceptual" de señales causales de duración finita.

Dada la secuencia

$$x[n] = \{ \underset{\uparrow}{1},\ 2, 1,\ 2 \}$$

de longitud Lx=4, y el sistema SDLI con respuesta al impulso

$$h[n] = \{ \underset{\uparrow}{4},\ 2,\ 1 \}$$

de longitud Lh=3, se pretende determinar $y[n] = x[n] * h[n]$, de longitud Ly=Lx+Lh-1.

a) De acuerdo con la ecuación (9) y teniendo en cuenta que $x[n]$ tiene un número finito de puntos, $y[n]$ puede escribirse como:

$$y[n] = x[0]h[n] + x[1]h[n-1] + x[2]h[n-2] + x[3]h[n-3] \qquad (10)$$

Con las secuencias x[n] y h[n] cree, respectivamente, los arreglos x y h de longitud Lx y Lh respectivamente:

```
x = [1  2  1  2];
h = [4  2  1];
```

Para crear cada una de las cuatro secuencias $yk = x[k]h[n-k]$ en la ecuación (10), escriba para $k = 0,1,2,3$, el siguiente código de Matlab:

```
hk = zeros(1,Ly);
hk(1+k:Lh+k) = h;
yk = x(k+1)*hk;
```

Para obtener y[n] a partir de las cuatro secuencias $x[k]h[n-k]$ en la ecuación (10), sume los arreglos yk:

```
y = y0 + y1 + y2 + y3;
```

Cree el arreglo de índices de tiempo:

n = 0 : Ly ;

y represente en una figura con cinco gráficas (utilizando `subplot(5,1, )` y `stem(n, )`) las cuatro señales $yk[n]$ y la suma $y[n]$.

b) Utilice la función conv( ) de Matlab para obtener el arreglo

```
yb=conv(x,y);
```

Ahora utilice la función Asys_conv1( ) para obtener el arreglo

```
yc=Asys_conv1(x,y);
```

c) Abra una figura y represente con tres gráficas (utilizando `subplot(3,1, )` y `stem(n, )`) los arreglos y, yb e yc. Observe que los tres representan la misma secuencia $y[n]$, para $n = 0,1,2,3,4,5$.

1.5.2 Experimento 2

Propiedades de la suma de convolución

Dadas las secuencias causales:

$$x[n] = \{ \underset{\uparrow}{1},\ 1, 1, -1, -1, -1,\ 1,\ 1,\ 1, -1, -1,\ -1, \}$$

$$h_1[n] = \{ 0,\ 0\ \underset{\uparrow}{1},\ 0 \}$$

$$h_2[n] = \{ \underset{\uparrow}{1},\ 2,\ 2,\ 1 \}$$

Cree los arreglos x, h1 y h2.Defina el arreglo de índices de tiempo `n1=0:Ly1-1;` donde Ly1 = Ly2 = Lx+Lh1-1. Defina el arreglo de índices de tiempo `n2=0:Ly-1;` donde Ly=(Lx+Lh1-1)+Lh2-1.

Utilizando la función Asys_conv1( ):

a) Ley conmutatividad
Obtenga las señales $y_a[n] = x[n] * h_1[n]$ y $y_b[n] = h_1[n] * x[n]$. En una figura con ayuda de subplot(3,1,) y stem(n1,) represente de forma gráfica las secuencias $x[n]$, $y_a[n]$ e $y_b[n]$ para verificar que $h_1[n] = \delta[n-2]$ y que $y_a[n] = y_b[n] = x[n-2]$.

b) Ley distributiva
Obtenga $y_1[n] = x[n] * h_1[n]$, $y_2[n] = x[n] * h_2[n]$ e $y_a[n] = y_1[n] + y_2[n]$. Para $h_3[n] = h_1[n] + h_2[n]$, obtenga $y_b[n] = h_3[n] * x[n]$. En una figura con ayuda de subplot(2,1,) y stem(n1,) represente de forma gráfica las secuencias $y_a[n]$ e $y_b[n]$ para verificar que $y_a[n] = y_b[n]$.

c) Ley asociativa
Para verificar que

$$(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$$

Obtenga

$$y_1[n] = x[n] * h_1[n]$$

$$y_a[n] = y_1[n] * h_2[n]$$

$$h_3[n] = h_1[n] * h_2[n]$$

$$y_b[n] = x[n] * h_3[n]$$

En una figura con ayuda de subplot(2,1,) y stem(n2,) represente de forma gráfica las secuencias $y_a[n]$ e $y_b[n]$ para verificar que $y_a[n] = y_b[n]$.

1.5.3 Experimento 3

Aplicación de la convolución para el filtrado de una señal de audio

Genere los tres tonos siguientes:

```
fs=44100;   %frecuencia de muestreo

n  = 0:1:2*fs;

x1 = 0.5*sin(2*pi*n/42);    %1050Hz

x2 = 0.3*sin(3*2*pi*n/42);  %2100Hz

x3 = 0.2*sin(6*2*pi*n/42);  %6300Hz
```

Abra una figura y represente con 3 gráficas los arreglos x1,x2 y x3:

```
ejes=[0 200 -1 1];

figure
```

Para i=1,2,3

```
subplot(3,1,i)

plot(xi)

AXIS(ejes)

title('xi[n]')
```

Obtenga la señal

```
x = x1 + x2 + x3;
```

En el archivo hn_por2ala15.txt se encuentra la secuencia $2^{15}h[n]$.Escriba el siguiente código de Matlab para guardar la secuencia h[n], en el arreglo h:

```
fid1 = fopen('hn_por2ala15.txt','r');
```

```
h = (1/2^15)* fscanf(fid1, '%d');
fclose(fid1);
```

Para escuchar la señal $x[n]$ escriba el siguiente código de Matlab:

```
wavplay(x,fs);
```

calcule $y[n] = x[n] * h[n]$ y escuche la secuencia $y[n]$ mediante el código de Matlab:

```
y = Asys_conv1(x,h);
wavplay(y,fs);
```

Abra una figura y represente con 3 gráficas los arreglos x, h e y:

```
ejes=[0 200 -1 1];
figure
subplot(3,1,1)
plot(x)
AXIS(ejes)
title('x[n]')

subplot(3,1,2)
plot(h)
AXIS(ejes)
title('h[n]')

subplot(3,1,3)
plot(y)
AXIS(ejes)
title('y[n]')
```

1.6 Bibliografía

OPPENHEIM, A.v., SCHAFER, R.w., BUCK, J. R., Tratamiento de señales en tiempo discreto, Madrid, Pearson Educación, 2000