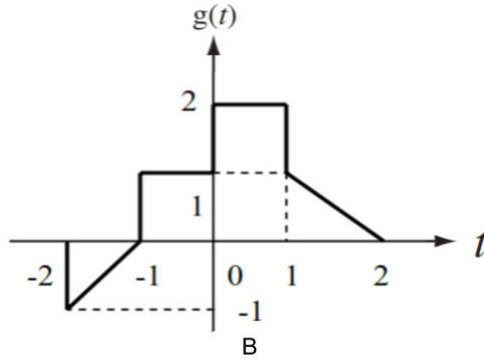
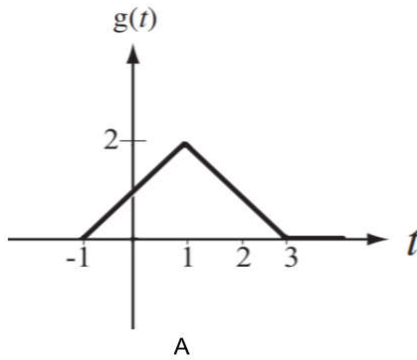


Trabajo Práctico N°1
Representación matemática de señales

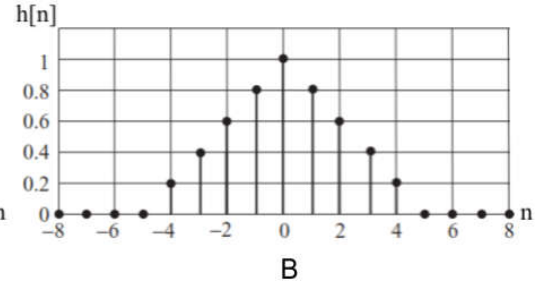
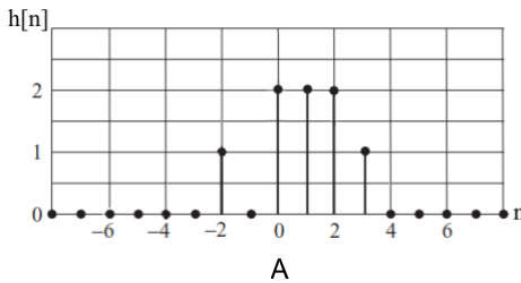
1) Para cada función $g(t)$, represente:

- a. $g(t - 1)$
- b. $-g(3t)$
- c. $g(-t - 5)$
- d. $g(t/3)$
- e. $g(2t - 3)$
- f. $2g(-1/2(t + 10))$



2) Para cada función $h[n]$, represente:

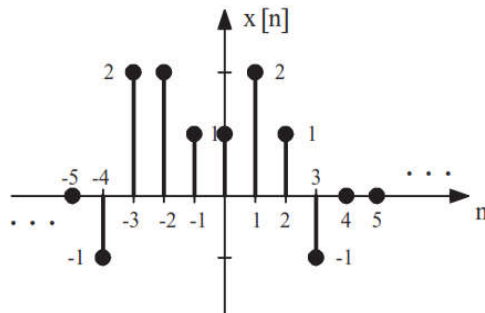
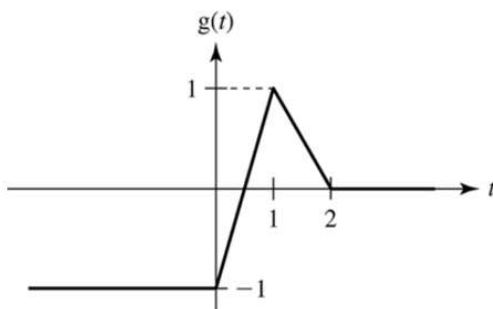
- a. $h[n + 3]$
- b. $h[2n]$
- c. $h[-n + 2]$
- d. $h[n/2 + 4]$
- e. $h[-4n + 4]$
- f. $h[n - 1]\delta[n - 2]$



$$h[n] \begin{cases} -2 & \text{si } n < -4 \\ n & \text{si } -4 \leq n < 1 \\ 4/n & \text{si } 1 \leq n \end{cases}$$

C

3) Dibujar las partes par e impar de las siguientes funciones:



Trabajo Práctico N°1
Representación matemática de señales

4) Determine las partes par e impar de estas funciones:

a. $g(t) = 2t^2 - 3t + 6$

b. $g(t) = 20 \cos(40\pi t - \pi/4)$

c. $g(t) = \text{sinc}(t)$

d. $g(t) = t(2 - t)(1 + 4t)$

e. $g[n] = u[n] - u[n - 4]$

f. $g[n] = e^{-(n/4)}u[n]$

g. $g[n] = \cos[2\pi n/4]$

h. $g[n] = \text{sen}[2\pi n/4] u[n]$

5) Determinar el período fundamental y la frecuencia fundamental de cada una de las siguientes funciones:

a. $g(t) = 10 \cos(50\pi t)$

b. $g(t) = 10 \cos(50\pi t + \pi/4)$

c. $g(t) = 10 \cos(50\pi t) + \text{sen}(15\pi t)$

d. $g(t) = \cos(2\pi t) + \text{sen}(3\pi t) + \cos(5\pi t - 3\pi/4)$

e. $g(t) = 2 \cos(4\pi t/5) * \text{sen}^2(16t/3)$

6) Determine el período N_0 y la frecuencia f_0 fundamental de las siguientes funciones:

a. $g[n] = \cos[2\pi n/10]$

b. $g[n] = \cos[\pi n/10]$

c. $g[n] = \cos[2\pi n/5] + \cos[2\pi n/7]$

d. $g[n] = e^{j[2\pi n/20]} + e^{-j[2\pi n/20]}$

e. $g[n] = e^{-j[2\pi n/3]} + e^{-j[2\pi n/4]}$

7) Determine cuál es el período fundamental de la señal $x[n] = A \cdot \cos[\omega n]$ para los siguientes valores de ω :

a. $\omega = 0,75\pi$

b. $\omega = 0,15\pi$

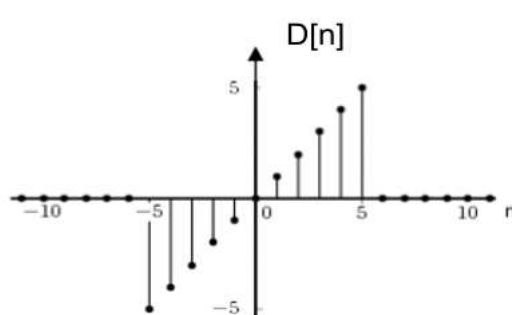
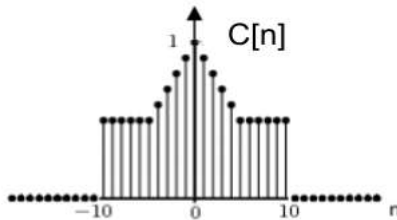
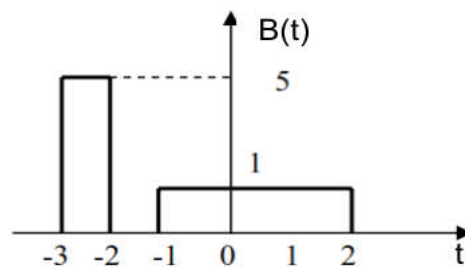
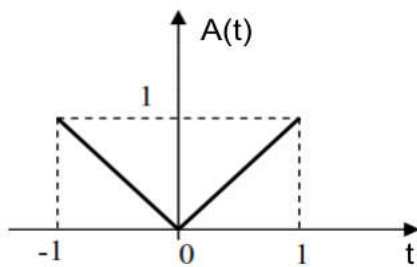
c. $\omega = 0,48\pi$

d. $\omega = (2)^{1/2}\pi$

e. $\omega = 0,23985\pi$

f. $\omega = 0,33$

8) Construir con funciones elementales (escalón, rampa, etc.) las siguientes funciones:



Trabajo Práctico N°1
Representación matemática de señalesOCTAVE APLICADO

- I. Para verificar el ejercicio 1 para la función A en el entorno de Octave, colocamos *edit* en la línea de comandos para abrir un editor, donde vamos a definir una función en dicho entorno. La función posee una cabecera en la primera línea del archivo con la palabra *function* y un cuerpo de definición que contiene las sentencias que sean necesarias. Para nuestro ejercicio tecleamos:

```
% Definición de la función A del ejercicio 1 (el porcentual
indica comentario)
function s=f1A(t)
s=(t+1).*(-1<=t & t<=1)+(-t+3).*(1<t & t<=3);
endfunction
```

Aquí se define una función de nombre f1A, que tiene un argumento (t) y devuelve un valor s correspondiente a la función matemática. La función matemática vale la recta (t+1) cuando el primer paréntesis vale 1 (verdadero), lo cual ocurre si se verifican las desigualdades lógicas de dicho paréntesis. Si se verifican las desigualdades lógicas del segundo paréntesis, éste toma el valor 1 con lo cual la función vale la recta (-t+3). Si no se cumple ninguna regla lógica, ambos paréntesis valen cero (falso) al igual que la función para dichos valores de t. A continuación, guardamos la función con el nombre f1A seguido de la extensión .m, en el directorio preestablecido por Octave.

Luego abrimos un nuevo archivo en el editor y colocamos la secuencia de comandos siguientes:

```
%Archivo para graficar la función f1A(t) y sus
transformaciones
fplot('f1A(t)', [-5 +5],200);
title ('f1A(t)');
figure; fplot('f1A(t-1)', [-5 +5],200);
title ('f1A(t-1)');
figure; fplot('-f1A(3*t)', [-5 +5],200);
title ('-f1A(3t)');
figure; fplot('f1A(-t-5)', [-9 -1],200);
title ('f1A(-t-5)');
figure; fplot('f1A(t/3)', [-4 +10],200);
title ('f1A(t/3)');
figure; fplot('f1A(2*t-3)', [-5 +5],200);
title ('f1A(2*t-3)');
figure; fplot('2*f1A(-1/2*(t+10))', [-20 0],200);
title ('2*f1A(-1/2*(t+10))');
```

El comando fplot grafica una función dentro de los límites especificados por el corchete, con un mínimo de puntos dado por el valor entero posterior. El comando figure abre una nueva ventana para graficar, y se preserva la figura previa. Por último la sentencia title añade un título al dibujo. Almacenamos el archivo con el nombre Ej1A.m y ahora podemos usar el nombre de este archivo como un comando más para graficar la función f1A(t) y sus transformaciones.

Trabajo Práctico N°1**Representación matemática de señales**

Como actividad desarrolle la función y el archivo para graficar la función B y las transformaciones correspondientes del ejercicio 1.

II. Verificamos las transformaciones discretas del ejercicio 2.

Colocamos *edit* en la línea de comandos para abrir un editor y definimos la siguiente función f2A:

```
%Definición de la función A discreta del ejercicio 2.
function y=f2A(n)
s=find((round(n)~=n));
n(s)=NaN;
y=0.*(n<-2)+1.*(n==-2)+0.*(n==-1)+2.*(n>=0 &
n<=2)+1.*(n==3)+0.*(n>=4);
endfunction
```

La tercera línea analiza el vector n en busca de valores no enteros, devolviendo la ubicación de estos en un vector s. La cuarta línea asigna a dichas posiciones el valor NaN (not a number) para que su evaluación no genere errores. La línea siguiente genera los valores de la función utilizando las sentencias lógicas como vimos anteriormente. Dicha función se guarda en el entorno de Octave bajo el nombre f2A.m. Armamos un archivo de comandos y lo almacenamos con el nombre Ej2A.m.

```
%Archivo para graficar la función f2A(t) y sus
transformaciones
n=-10:1:10;
g0=f2A(n);
figure;stem(n,g0,'filled');
title('función f2A[n]');
g1=f2A(n+3);
figure;stem(n,g1,'filled');
title('función f2A[n+3]');
g2=f2A(2*n);
figure;stem(n,g2,'filled');
title('función f2A[2*n]');
g3=f2A(-n+2);
figure;stem(n,g3,'filled');
title('función f2A[-n+2]');
g4=f2A(n/2+4);
figure;stem(n,g4,'filled');
title('función f2A[n/2+4]');
g5=f2A(-4*n+4);
figure;stem(n,g5,'filled');
title('función f2A[-4*n+4]');
g6=f2A(n-1).*(n==2);
figure;stem(n,g6,'filled');
title('función f2A[n-1]*d[n-2]');
```

Trabajo Práctico N°1**Representación matemática de señales**

La función `stem(x,y,'fill')` grafica el vector y versus el vector x , mientras 'fill' especifica el llenado del círculo en el extremo de la línea. Como actividad desarrolle las funciones y los archivos para graficar las funciones B y C del ejercicio 2.

- III. Para el ejercicio 4a, definimos y almacenamos el nombre de la función `f4a` seguido de la extensión `.m`, en el directorio preestablecido por Octave.

```
% Definición de la función f4a
function s=f4a(t)
s=2*t.^2-3*t+6;
endfunction
```

Armamos un archivo de comandos, y lo almacenamos con el nombre `Ej4a.m`

```
%Gráfica de la parte par e impar de la función f4a
figure;fplot('f4a(t)',[-5 +5],200);
title('f4a(t)');
figure;fplot('(f4a(t)+f4a(-t))/2',[-5 +5],200);
title('Par de f4a(t)');
figure;fplot('(f4a(t)-f4a(-t))/2',[-5 +5],200);
title('Impar de f4a(t)');
```

Al ejecutar dicho archivo en la línea de comando, se grafica la parte par e impar del ejercicio 4a.

Para el ejercicio 4e, definimos la función `ud` (escalón discreto) y la función `f4e` seguido de la extensión `.m`, en el directorio preestablecido por Octave.

```
%Definición de la función ud
function y=ud(n)
s=find((round(n)~=n));
n(s)=NaN;
y=1.*(0<=n);
endfunction
```

```
%Definición de la función discreta f4e
function y=f4e(n)
y=ud(n)-ud(n-4);
endfunction
```

Armamos un archivo de comandos, y lo almacenamos con el nombre `Ej4e.m`.

```
%Gráfica de la función par e impar de la función f4e
n=-10:1:10;
par=(f4e(n)+f4e(-n))./2;
figure;stem(n,par,'filled');
title('función par de la función f4e');
impar=(f4e(n)-f4e(-n))./2;
figure;stem(n,impar,'filled');
title('función impar de la función f4e');
```

Trabajo Práctico Nº1**Representación matemática de señales**

Desarrolle las funciones y los archivos para graficar las partes pares e impares de los puntos restantes del ejercicio 4.

- IV. Verifique la periodicidad de las funciones graficando los ejercicios del punto 5.
- V. Verifique la periodicidad de las funciones graficando los ejercicios del punto 6.
- VI. Interpretación gráfica de la frecuencia angular discreta del ejercicio 7. Grabe esta secuencia de comandos como Ej7a.m

```
n=-30:1:30;  
grafical=cos(3/4*pi.*n);  
figure(1);stem(n,grafical,'filled');  
title('ejercicio 7a');  
hold on;  
fplot('cos(3/4*pi.*t)',[-30 30],400);
```

Se puede observar que la fracción racional determina que el denominador es el período real de la función discreta y el numerador la cantidad de veces que se debe repetir la señal analógica para obtener el período discreto.

Verifique lo anterior para los ejercicios del punto 7b y 7c.