Análisis de Señales — Curso 2024

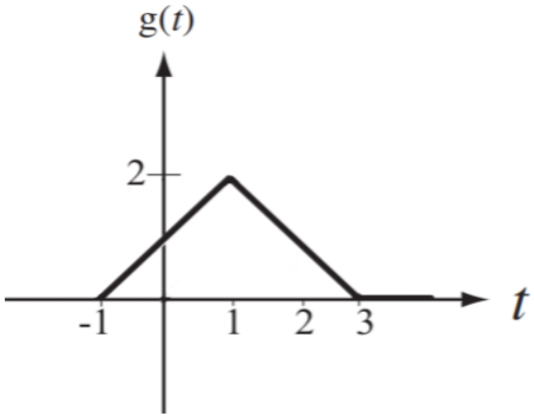
Trabajo Práctico Nº1

Representación matemática de señales – con Octave aplicado

Ejemplo1

Para la función $g(t)$, represente:

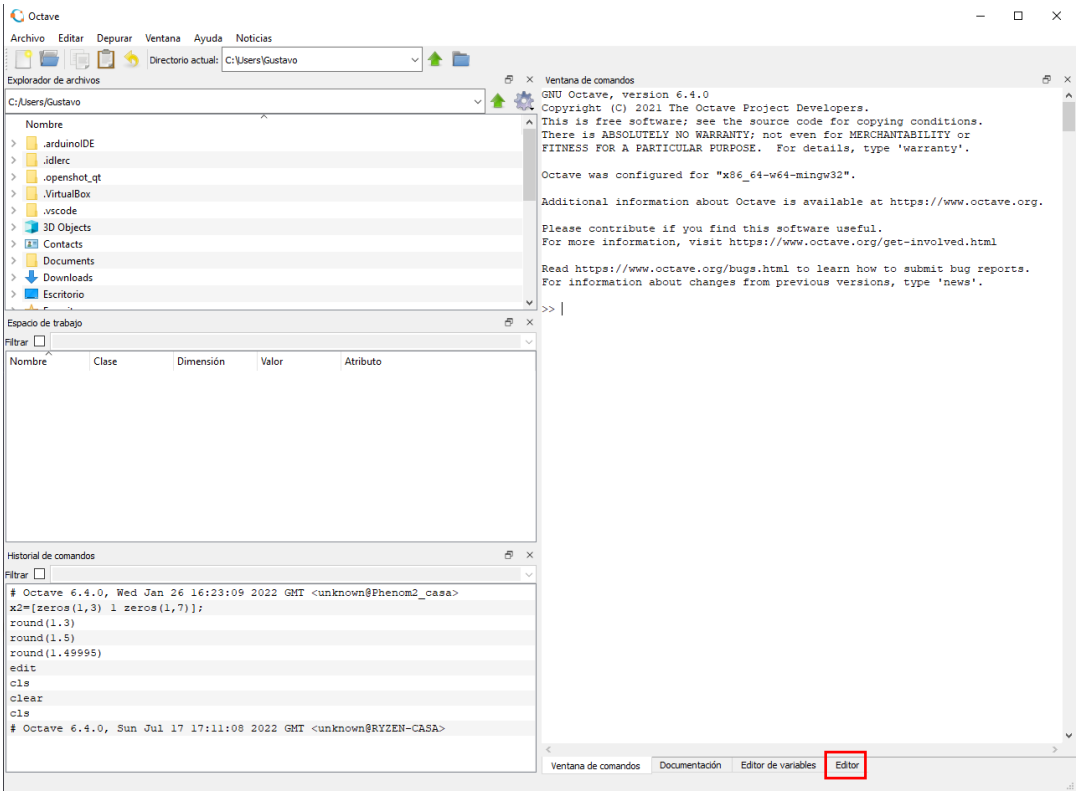
a. $g(t-1)$
b. $-g(3t)$
c. $g(-t-5)$
d. $g(t/3)$
e. $g(2t-3)$
f. $2g(-1/2(t+10))$



Primero para utilizar el Octave tenemos dos opciones:

- Instalar el software desde la página oficial https://octave.org para el sistema operativo utilizado.
- Utilizar Octave online desde https://octave-online.net

En nuestro caso instalamos Octave y utilizamos el entorno gráfico, el cual se muestra en la figura siguiente:



La ventana derecha es la de *comandos* y debajo de esta hay 4 solapas, la última es la de *editor*. En la zona izquierda hay 3 ventanas, de arriba hacia abajo tenemos la de: *explorador de archivos*, *espacio de trabajo y de historial de comandos.*

Trabajo Práctico Nº1
Representación matemática de señales – con Octave aplicado

Para desarrollar el ejemplo 1 con la función g(t), nos situamos en el 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 g(t)del ejemplo 1 (% indica comentario)
function s=g(t)
s=(t+1).*(-1<=t & t<=1)+(-t+3).*(1<t & t<=3);
endfunction
```

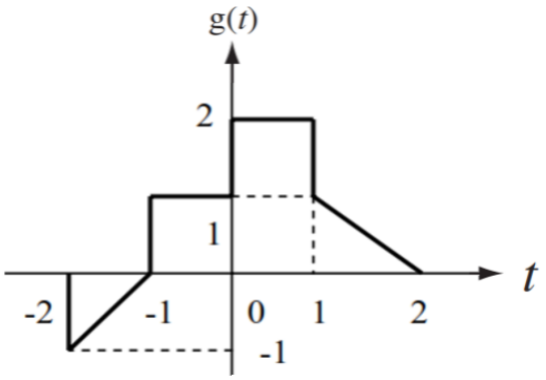
Aquí se define una función de nombre g, que tiene un argumento (t) y devuelve un valor s correspondiente a la función matemática. La función matemática vale la recta (t+1) cuando el primer paréntesis vale 1 (verdadero), lo cual ocurre si se verifican las desigualdades lógicas de dicho paréntesis. Si 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 g 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 (script):

```
%Archivo para graficar la función g(t) y sus transformaciones
fplot('g(t)', [-5 +5],200);
title ('g(t)');
figure; fplot('g(t-1)', [-5 +5],200);
title ('g(t-1)');
figure; fplot('-g(3*t)', [-5 +5],200);
title ('-g(3t)');
figure; fplot('g(-t-5)', [-9 -1],200);
title ('g(-t-5)');
figure; fplot('g(t/3)', [-4 +10],200);
title ('g(t/3)');
figure; fplot('g(2*t-3)', [-5 +5],200);
title ('g(2*t-3)');
figure; fplot('2*g(-1/2*(t+10))', [-20 0],200);
title ('2*g(-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 Ej1.m y ahora podemos usar el nombre de este archivo como un comando más para graficar la función g(t) y sus transformaciones.

Actividad 1

Desarrolle la función y el archivo en Octave para graficar la función g(t) con las transformaciones del punto anterior.



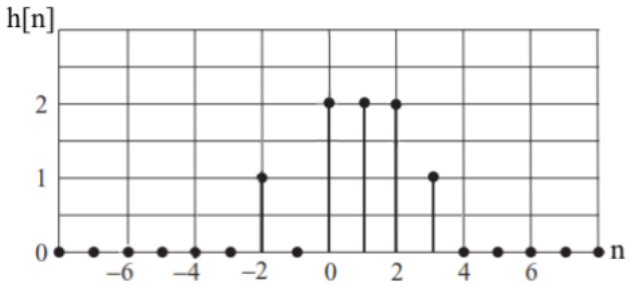
Trabajo Práctico Nº1

Representación matemática de señales – con Octave aplicado

Ejemplo 2

Para la 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. $h[n-1]\delta[n-2]$



Para desarrollar las transformaciones discretas del ejemplo 2 abrimos el editor y definimos la siguiente función h[n]:

```
%Definición de la función h[n]discreta del ejemplo 2.
function y=h(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 h.m.

Luego armamos un archivo de comandos y lo almacenamos con el nombre Ej2.m.

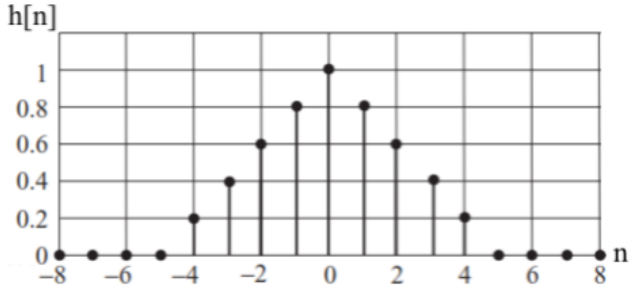
```
%Archivo para graficar la función h(n) y sus transformaciones
n=-10:1:10;
g0=h(n);
figure;stem(n,g0,'filled');
title('función h[n]');
g1=h(n+3);
figure;stem(n,g1,'filled');
title('función h[n+3]');
g2=h(2*n);
figure;stem(n,g2,'filled');
title('función h[2*n]');
g3=h(-n+2);
figure;stem(n,g3,'filled');
title('función h[-n+2]');
g4=h(n/2+4);
figure;stem(n,g4,'filled');
title('función h[n/2+4]');
g5=h(-4*n+4);
figure;stem(n,g5,'filled');
title('función h[-4*n+4]');
g6=h(n-1).*(n==2);
figure;stem(n,g6,'filled');
title('función h[n-1]*d[n-2]');
```

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.

Trabajo Práctico Nº1
Representación matemática de señales – con Octave aplicado

Actividad 2

Desarrolle la función y el archivo en Octave para graficar la función h[n] con las transformaciones del punto anterior.



Actividad 3

Desarrolle las funciones y los archivos en Octave para graficar las partes pares e impares de las siguientes funciones:

a. $g(t) = 2t^2 - 3t + 6$
b. $g(t) = 20\cos(40\pi t - \pi/4)$
c. $g(t) = t(2-t)(1+4t)$
d. $g[n] = e^{-(n/4)}u[n]$
e. $g[n] = sen[2\pi n/4]\,u[n]$
f. $g[n] = u[n] - u[n-4]$

AYUDA: Para las funciones discretas, definimos la función ud (escalón discreto) con 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
```

Actividad 4

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

a. $g(t) = 10\cos(50\pi t) + sen(15\pi t)$
b. $g(t) = \cos(2\pi t) + sen(3\pi t) + \cos(5\pi t - 3\pi/4)$
c. $g(t) = 2\cos(4/5\pi t) * sen^2(2/3\pi t)$

Actividad 5

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

a. $g[n] = \cos[2\pi n/5] + \cos[2\pi n/7]$
b. $g[n] = e^{j[2\pi n/20]} + e^{-j[2\pi n/20]}$
c. $g[n] = e^{-j[2\pi n/3]} + e^{-j[2\pi n/4]}$

Actividad 6

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

A	B	C	D	E	F
Ω = 0,75π	Ω = 0,15π	Ω = 0,48π	Ω = $(2)^{1/2}$π	Ω =0,23985π	Ω = 0,33

Trabajo Práctico Nº1

Representación matemática de señales – con Octave aplicado

Ejemplo 3

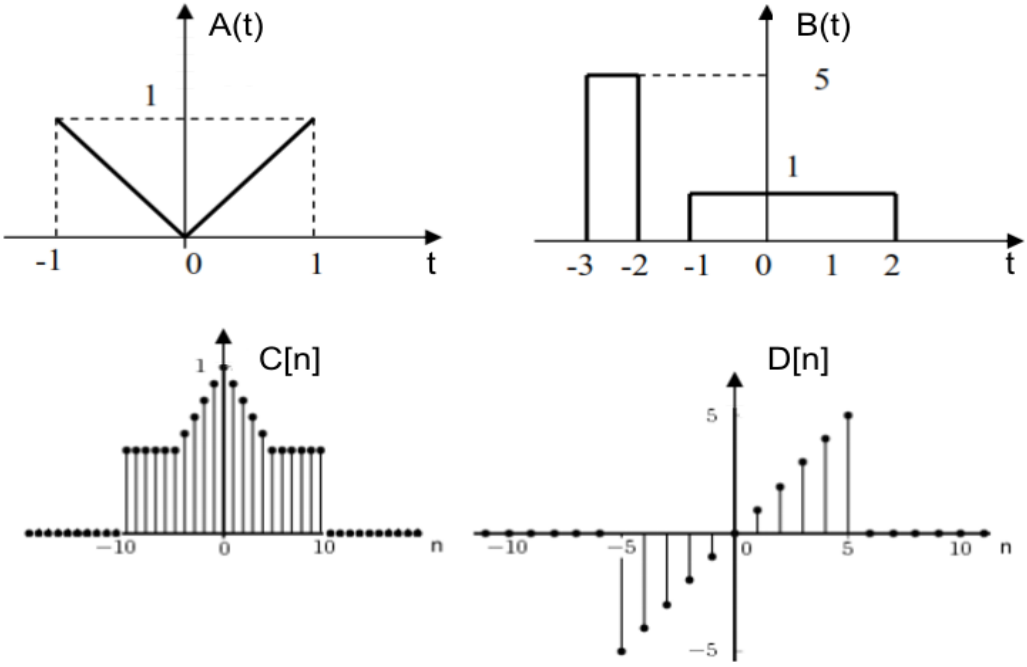
La interpretación gráfica de la frecuencia angular discreta de la actividad 6 la podemos ver utilizando Octave. Grabe esta secuencia de comandos como Ej6a.m

```
n=-30:1:30;
grafica1=cos(3/4*pi.*n);
figure(1);stem(n,grafica1,'filled');
title('ejemplo 6a');
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.

Actividad 7

Desarrolle un archivo de código en Octave para la actividad 6b y 6c.

Actividad 8

Construir con funciones elementales (escalón, rampa, cajón, etc.) las siguientes funciones, realizando las transformaciones necesarias en cada caso.