DEPARTAMENTO DE
INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Laboratorio 2

PROCESAMIENTO DE SEÑALES E IMÁGENES

Profesores:

- Violeta Chang C.
- Leonel E. Medina

Ayudante: Luis Corral

Repaso y más sobre MATLAB

Un aspecto importante en Matlab es la posibilidad de realizar cálculos mediante vectorización. Esta característica hace uso de la expansión implícita donde las dimensiones no congruentes son interpretadas de todas maneras:

```
clearvars

a = [1:5;6:10]        % 2x5
```

```
a = 2×5
     1     2     3     4     5
     6     7     8     9    10
```

```
b = a.*(ones(1,5)*2)  % 2x5 * 1x5 ? -> Multiplicación por filas.
```

```
b = 2×5
     2     4     6     8    10
    12    14    16    18    20
```

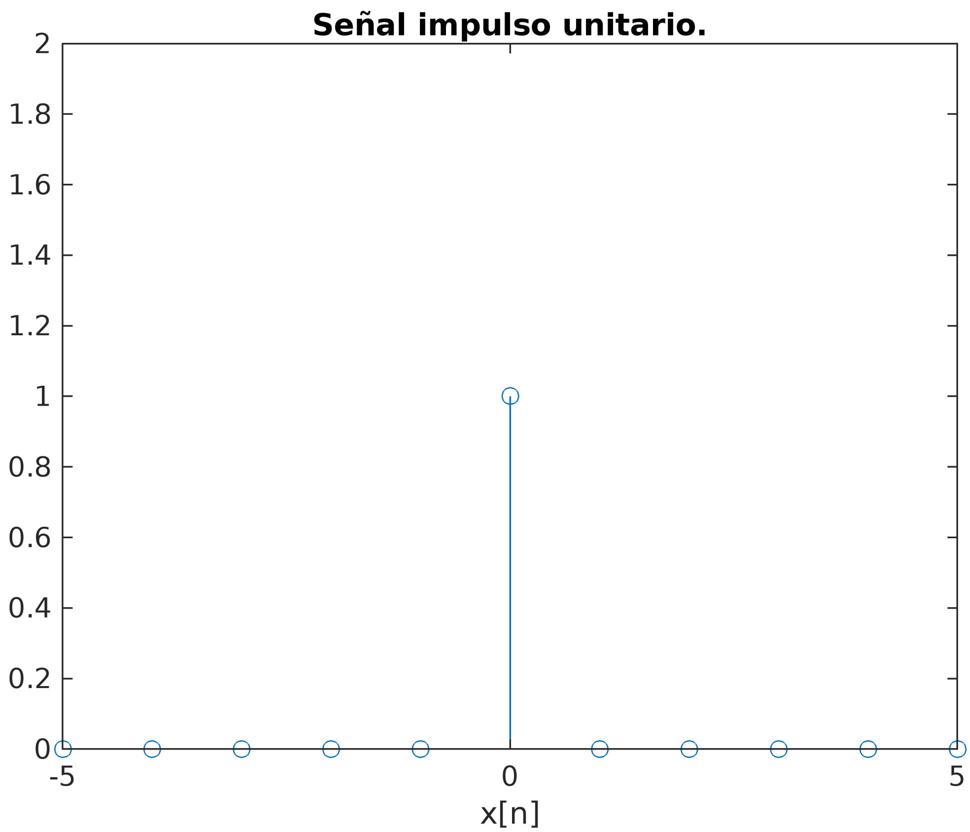
```
c = a.*([1;1]*3)      % 2x5 * 2x1 ? ->      ''      por columnas.
```

```
c = 2×5
```

```
     3     6     9    12    15
    18    21    24    27    30
```

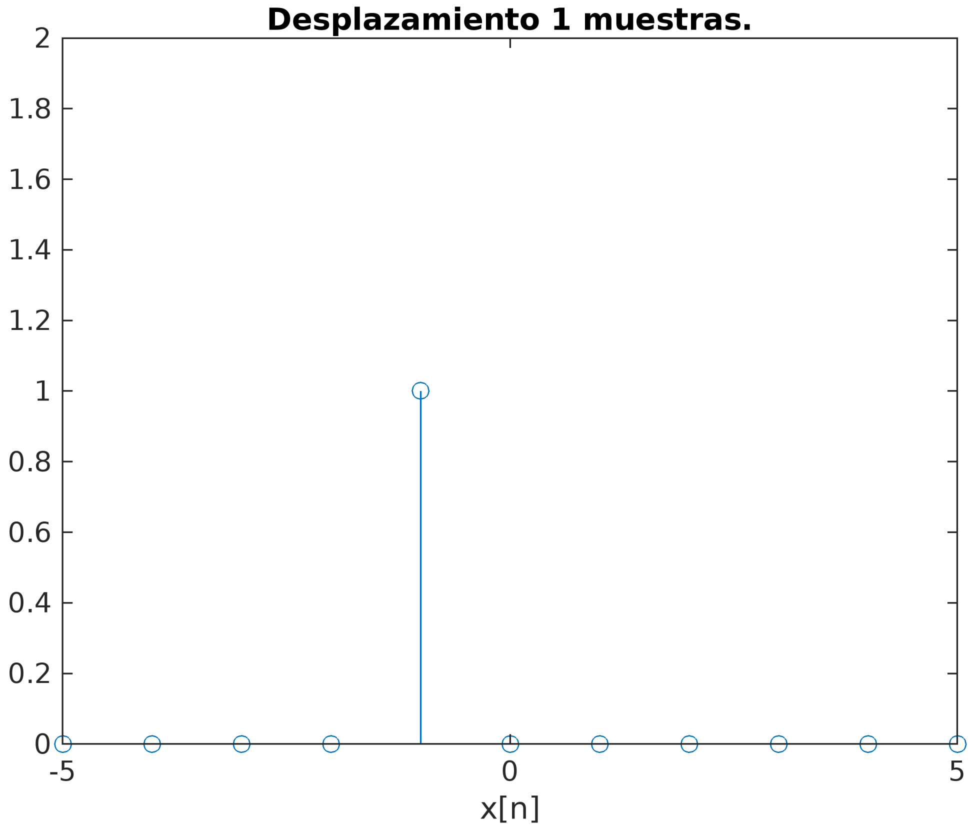
Además, podemos utilizar operadores condicionales para modificar vectores y matrices [1]:

```
clearvars

n = -5:5;
x_n = n==0;                         % Arreglo tipo lógico 1 en n==0.

figure
stem(n,x_n)
title('Señal impulso unitario.')
xlabel('x[n]')
ylim([0 2])
```

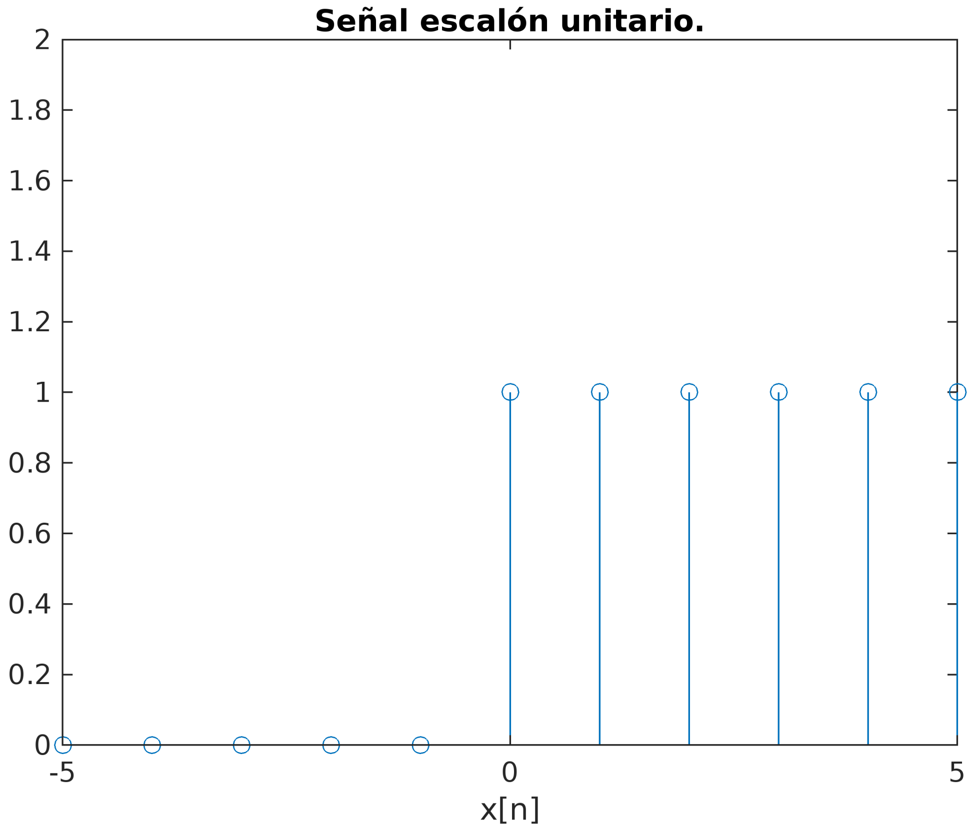



```
% x_n = (n + delta)==0;
delta = 1;    % Deslizador gráfico muy cool.
x_n = (n + delta)==0;               % Arreglo tipo lógico 1 en n+delta==0.

figure
stem(n,x_n)
title(['Desplazamiento ' num2str(delta) ' muestras.'])
```

```
xlabel('x[n]')
ylim([0 2])
```

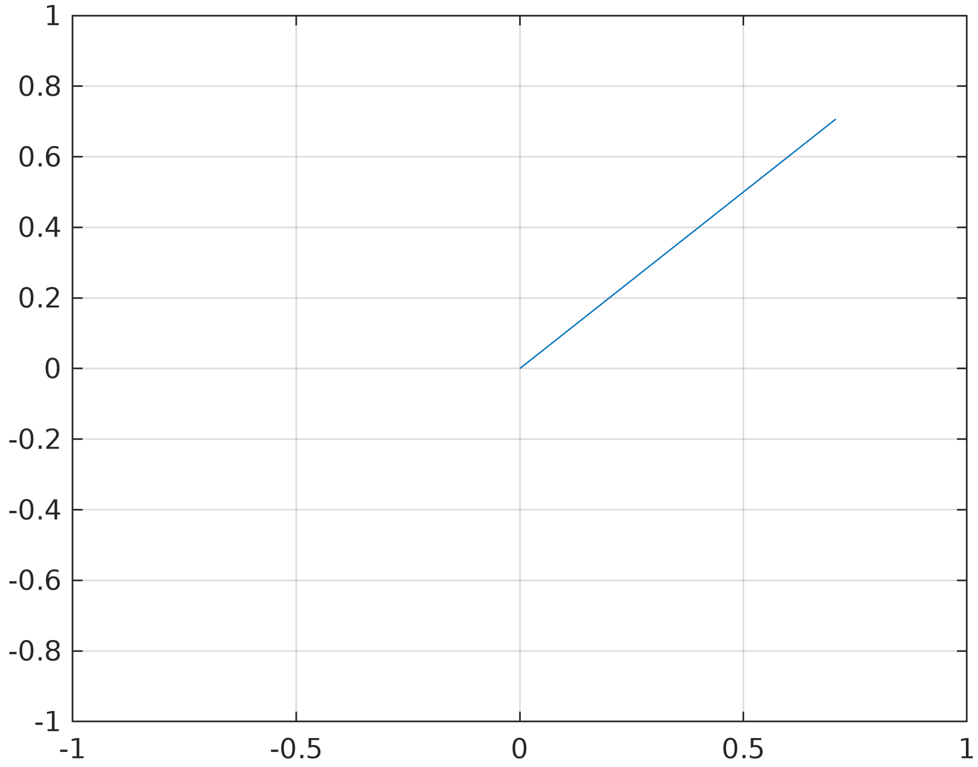



```
x_n = n>=0;                          % Arreglo tipo lógico 1 en n>=0.

figure
stem(n,x_n)
title('Señal escalón unitario.')
xlabel('x[n]')
ylim([0 2])
```



Finalmente, la representación de números complejos en Matlab se realiza mediante el operador 1i:

```
a = 0.707;
b = 0.707;
C_1 = a+1i*b;                         % Variable tipo double (complex).

figure
plot([0 real(C_1)], [0 imag(C_1)])   % Partes reales e imaginarias.
xlim([-1 1])
ylim([-1 1])
grid on
```



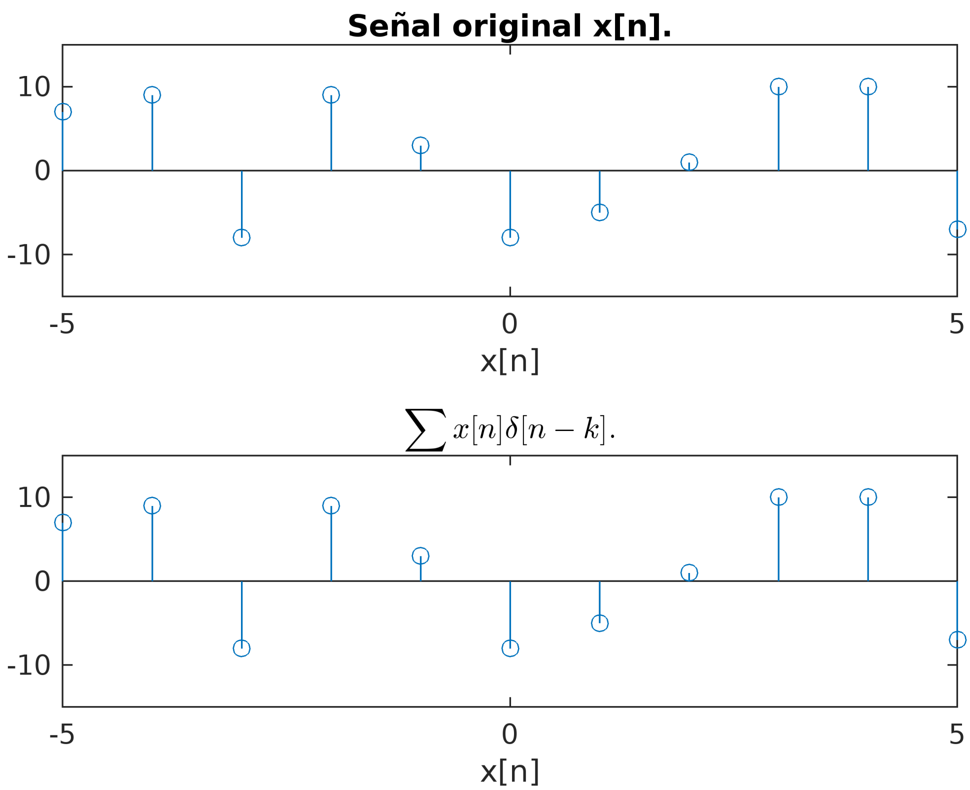
Representación en impulsos unitarios

La idea es multiplicar cada valor de una secuencia discreta $x[n]$ por un impulso unitario $\delta[n]$ desplazado k muestras entre el intervalo $-\infty < n < \infty$:

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

```
clearvars

l = 5;
n = -l:l;
N = length(n);
x_n_1 = randi([-10 10],1,N);   % N números aleatorios enteros entre -10 y 10.
x_n_2 = zeros(1,N);

% Ciclos
for i = 1:N
    d = (n - n(i))==0;
    x_n_2 = x_n_2 + (d * x_n_1(i));
end

% % Vectorizado
% x_n_3 = eye(N)*x_n_1';
```

```
figure
subplot(2,1,1)
stem(n,x_n_1)
title('Señal original x[n].')
xlabel('x[n]')
ylim([-15 15])
subplot(2,1,2)
stem(n,x_n_2)
title('$$\sum x[n]\delta[n-k]$$.','interpreter','latex')
xlabel('x[n]')
ylim([-15 15])
```



Suma de convolución

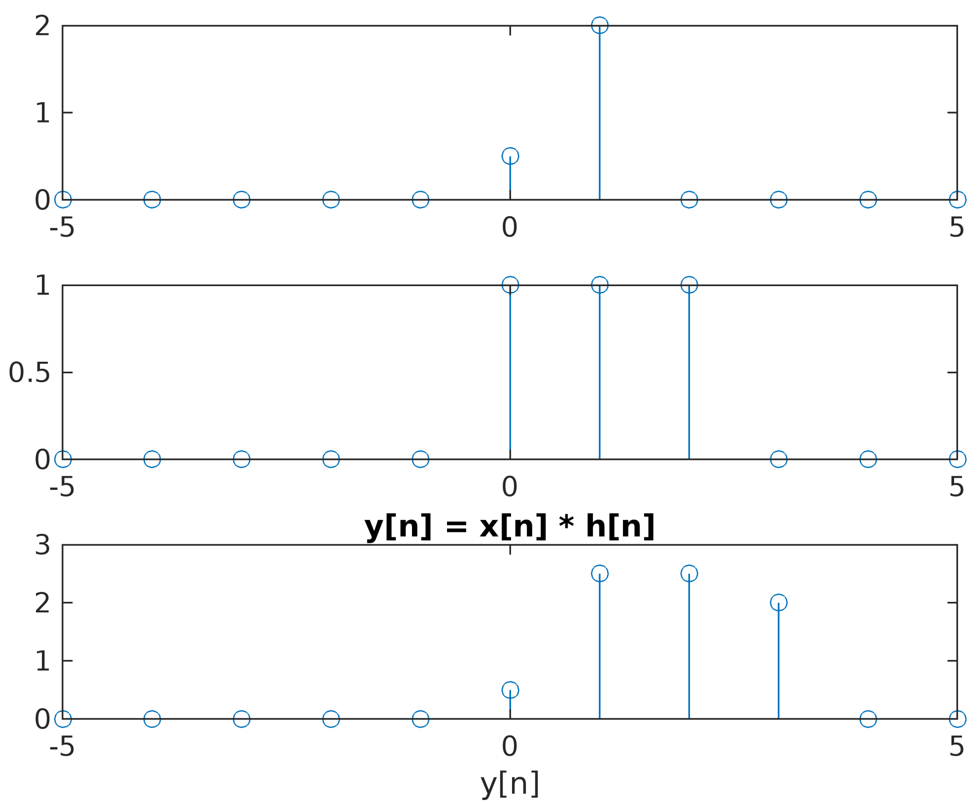
La suma de convolución entre las señales $h[n]$ y $x[n]$ está definida como (ver ejemplo 2.1):

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

$$y[n] = x[n] * h[n]$$

```
clearvars
```

```
n = -5:5;
N = length(n);
h_n = n>=0&n<=2;
x_n = zeros(1,N);
zv = find(n==0);
x_n(1,zv:zv+1) = [0.5 2];
y_n = zeros(1,N);
for i = 0:1
    y_n = y_n + x_n(1,zv+i)*circshift(h_n,i);
end

figure
subplot(3,1,1)
stem(n,x_n)
subplot(3,1,2)
stem(n,h_n)
subplot(3,1,3)
stem(n,y_n)
ylim([0 3])
title('y[n] = x[n] * h[n]')
xlabel('y[n]')
```



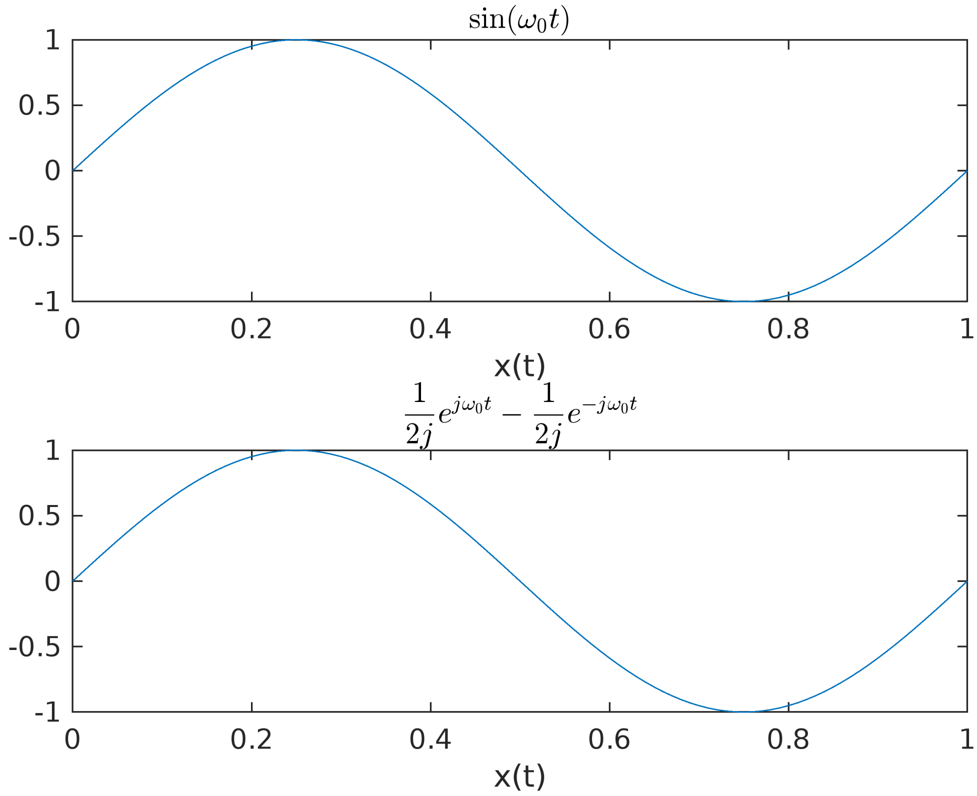
Recordar la similitud con la integral de convolución:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

Representación en series de Fourier

Considere el ejemplo 3.3 para la señal $x(t) = \sin(\omega_0 t)$ con $\omega_0 = 2\pi$, para la cual tenemos coeficientes $a_1 = 1/2j$, $a_{-1} = -1/2j$ y cero para todos los otros valores de k:

```
clearvars

t = 0:0.01:1;                                   % OJO, tiempo continuo discretizado.
x_t_1 = sin(2*pi*t);                            % Señal original.
x_t_2 = (1/(2*1i))*exp(1i*2*pi*t) - ...
        (1/(2*1i))*exp(-1i*2*pi*t);             % Señal en series de Fourier.

figure
subplot(2,1,1)
plot(t,x_t_1)
title('$${\sin(\omega_0 t)}$$','interpreter','latex')
xlabel('x(t)')
subplot(2,1,2)
plot(t,real(x_t_2))
title('$${\frac{1}{2j} e^{j \omega_0 t} -  \frac{1}{2j} e^{-j \omega_0 t}}$$', ...
      'interpreter','latex')
xlabel('x(t)')
```




```
P_prom_1 = (1/(2*pi))*trapz(2*pi*t,abs(x_t_1).^2)  % Integración mediante la
```

```
P_prom_1 = 0.5000
```

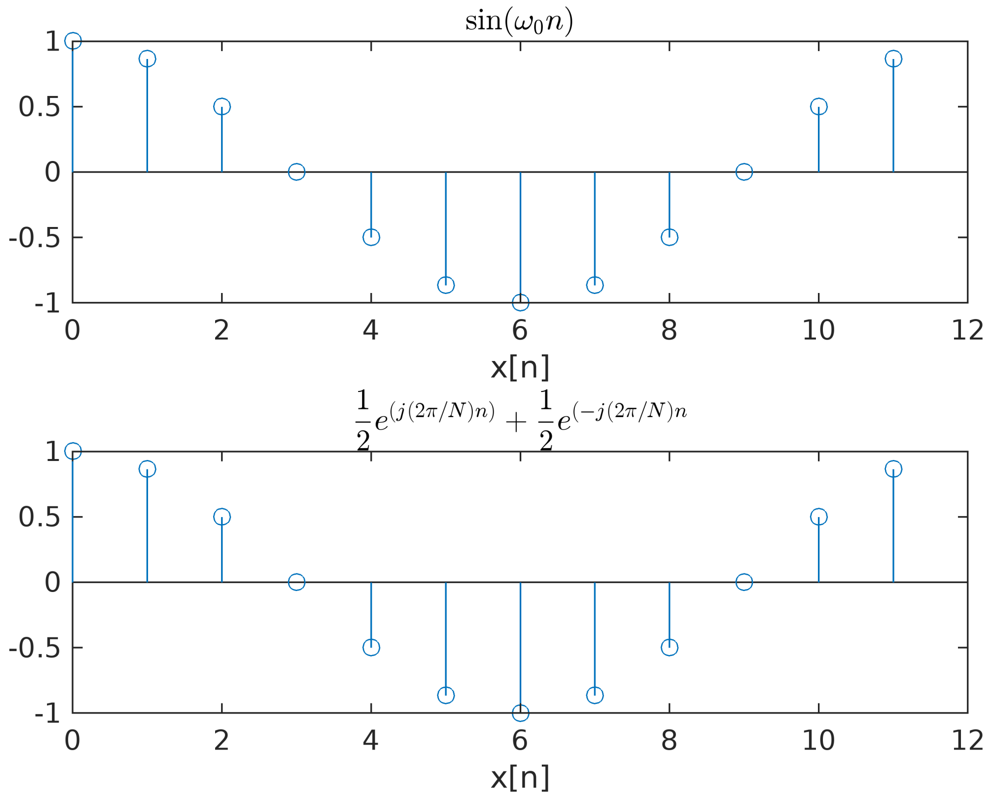
```
                                                   % regla del trapecio.
P_prom_2 = (abs(1/(2*1i)))^2 + (abs(-1/(2*1i)))^2
```

```
P_prom_2 = 0.5000
```

Para la versión discreta, considere la señal periodica del laboratorio 1 $x[n] = \cos(2\pi n/12)$:

```
clearvars

n = 0:11;
N = length(n);
x_t_1 = cos(2*pi*n/12);                   % Señal original.
x_t_2 = (1/2)*exp(1i*(2*pi/N)*n) + ...
        (1/2)*exp(-1i*(2*pi/N)*n);        % Señal en series de Fourier.

figure
subplot(2,1,1)
stem(n,x_t_1)
title('$${\sin(\omega_0 n)}$$','interpreter','latex')
xlabel('x[n]')
subplot(2,1,2)
```

```
stem(n,real(x_t_2))
title('$${\frac{1}{2} e^{(j (2 \pi / N) n)} +  \frac{1}{2} e^{(-j (2 \pi / N) n}}$$', ...
      'interpreter','latex')
xlabel('x[n]')
```




```
P_prom_d_1 = (1/N)*sum(abs(x_t_1).^2)
```

```
P_prom_d_1 = 0.5000
```

```
P_prom_d_2 = (abs(1/(2*1i)))^2 + (abs(-1/(2*1i)))^2
```

```
P_prom_d_2 = 0.5000
```

La transformada continua de Fourier

El par de transformadas continuas de Fourier nos permiten obtener el espectro $X(j\omega)$ de una señal $x(t)$ y su reconstrucción con la operación inversa:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega$$

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

Para la función $x(t) = \sin(\omega_0 t)$ con $\omega_0 = 2\pi$ se tiene:

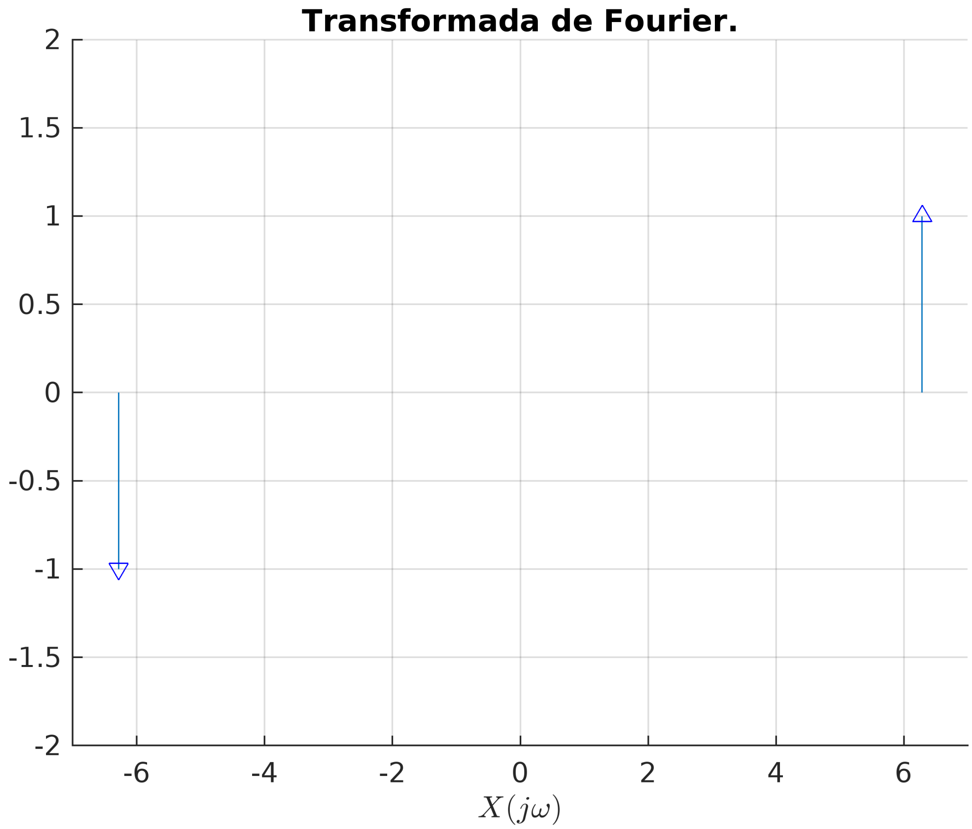
```
clearvars

w_sin = 2*pi;
w_1 = -w_sin                    % Ver eq. 4.10, 2 coeficientes 2 valores para w.
```

```
w_1 = -6.2832
```

```
w_2 = w_sin
```

```
w_2 = 6.2832
```

```

figure
w = [w_1 w_1];
y = [0 -1];
line(w,y);
hold on
plot(w_1, -1, 'bv')
ylim([-2 2])
xlim([-7 7])
hold on
w = [w_2 w_2];
y = [0 1];
line(w,y);
hold on
plot(w_2, 1, 'b^')
title('Transformada de Fourier.')
xlabel('$$X(j\omega)$$','interpreter','latex')
grid on
```



Ejercicio

Realice la transformada de Fourier para la señal:

$x(t) = sin(2\pi t) + cos(4\pi t)$.

Genere un gráfico y haga un pequeño análisis de sus resultados. Se evalúa los conceptos en formato de texto, los comentarios dentro del código, la exactitud del cálculo y la calidad de los gráficos generados. Muestre solo los valores más importantes.

Referencias

[1] https://www.cpp.edu/~zaliyazici/ece308/MatLab.html

[2] Oppenheim, A.V. & Willsky, A.S. & Nawab, S.H. (1997). Señales y sistemas (2nd ed.). Prentice Hall.