DEPARTAMENTO DE
INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Evaluación 1

PROCESAMIENTO DE SEÑALES E IMÁGENES

Profesores:

- Violeta Chang C.
- Leonel E. Medina

Ayudante: Luis Corral

Nombre Alumno: John Serrano C.

Problema 1

1-. Determine de manera grafica que señal se obtiene luego de la convolución $y_j(t) = x(t) * h_j(t)$, utilizando como frecuencias $\omega_i = 2\pi$ y $\omega_c = 4\pi$:

$$x(t) = \frac{\sin \omega_i t}{\pi t},$$

$$h_1(t) = \delta(t),$$

$$h_2(t) = \frac{\sin \omega_c t}{\pi t},$$

$$h_3(t) = sin(\pi t) + cos(4\pi t).$$

2-. Determine de manera grafica que señal se obtiene luego de la convolución $y_1(t) = x(t) * h_1(t)$ del ejercicio anterior cuando $x(t) = x(t + t_0)$ con $t_0 = 0.5$.

Problema 2

Compruebe los resultados del Problema 1 utilizando la función conv de Matlab para realizar la convolución $y_j(t) = x(t) * h_j(t)$ y la función sinc de Matlab para obtener las señales $x(t)$ y $h_2(t)$. Normalice los valores obtenidos de la convolución dividiendo por el valor máximo (utilizando la función max de matlab) de esta señal y la opción 'same' en la función conv. Adicionalmente, calcule el resultado para $y_4(t)$ con $T = 0.5$ y:

$$h_4(t) = \begin{cases} 1, & |t| < T \\ 0, & |t| > T \end{cases}.$$

Referencias

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

Desarrollo

Problema N°1 - Parte 1

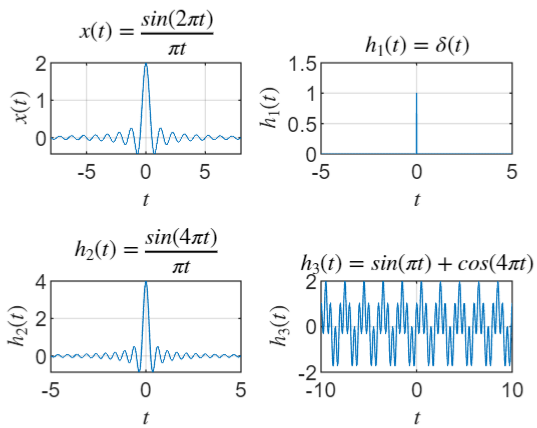
Como no podemos utilizar la función conv() de Matlab, vamos a aplicar la siguiente propiedad: **"La Transformada de Fourier de una convolución es la multiplicación de las Transformadas de Fourier individuales de las funciones".** Por lo tanto, debemos comenzar obteniendo las Transformadas de Fourier de $x(t)$, $h_1(t)$, $h_2(t)$ y $h_3(t)$, considerando además que debemos utilizar las frecuencias que nos da el enunciado cuando sea necesario.

Lo primero es definir las frecuencias junto con las señales del enunciado. También podemos graficarlas para poder entender con qué estamos trabajando.

```
% Definición de frecuencias y señales junto con sus gráficos
% correspondientes

clearvars

w_i = 2 * pi;               % Se define w_i = 2*pi
w_c = 4*pi;                 % Se define w_c = 4*pi

t = -5*pi:0.001*pi:5*pi;    % Se define un rango de tiempo en el que vamos a trabajar
h1 = t == 0;                % Se define h1(t) (Señal impulso unitario)
h3 = sin(pi*t) + cos(4*pi*t);

% Si intentamos graficar x(t) tal como está definida, tendremos problemas
% cuando t = 0. Para eso, podemos definir x(t) de otra manera

x = zeros(1,length(t));     % Se define un arreglo de 0s
```

```
for i = 1:length(t)          % Se recorre el arreglo
    x(i) = (sin(w_i*t(i)))/(pi*t(i));    % Se calcula cada valor de x(t)
end

% Lo mismo para el caso de h2(t):

h2 = zeros(1,length(t));     % Se define un arreglo de 0s
for i = 1:length(t)          % Se recorre el arreglo
    h2(i) = (sin(w_c*t(i)))/(pi*t(i));   % Se calcula cada valor de h2(t)
end

subplot(2,2,1);              % Se inicializa un subgráfico
plot(t,x)                    % Se grafica x(t)
xlim([-8 8])                 % Se definen los límites del eje x entre -8 y 8

% Configuraciones adicionales del subgráfico
title('$x(t) = \frac{sin(2\pi t)}{\pi t}$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$x(t)$$','interpreter','latex')
grid on

subplot(2,2,2);              % Se inicializa un subgráfico
plot(t,h1)                   % Se gráfica h1(t)
xlim([-5 5])                 % Se definen los límites del eje x entre -5 y 5
ylim([0 1.5])                % Se definen los límites del eje y entre 0 y 1.5

% Configuraciones adicionales del subgrafico
title('$h_1(t) = \delta(t)$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$h_1(t)$$','interpreter','latex')
grid on

subplot(2,2,3)               % Se inicializa un subgráfico
plot(t,h2)                   % Se gráfica h2(t)
xlim([-5 5])                 % Se definen los límites del eje x entre -5 y 5

% Configuraciones adicionales del subgráfico
title('$h_2(t) = \frac{sin(4\pi t)}{\pi t}$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$h_2(t)$$','interpreter','latex')
grid on

subplot(2,2,4)               % Se inicializa un subgráfico
plot(t,h3)                   % Se gráfica h3(t)
xlim([-10 10])               % Se definen los límites del eje x entre -10 y 10

% Configuraciones adicionales del subgráfico
title('$h_3(t) = sin(\pi t) + cos(4\pi t)$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$h_3(t)$$','interpreter','latex')
```

```
grid on
```

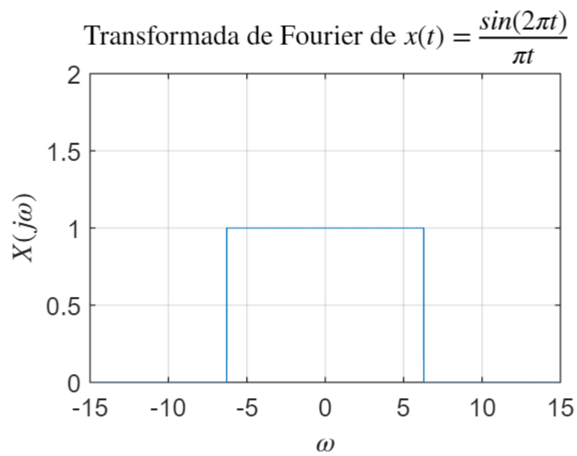


Ahora, vamos a obtener la Transformada de Fourier de $x(t)$ y a graficarla. Del libro, sabemos que la Transformada de Fourier para una Señal del estilo de $x(t)$ es:

$$\mathscr{F}\left(\frac{sin(Wt)}{\pi t}\right) = \begin{cases} 1, & |w| < W \\ 0, & |w| > W \end{cases}$$

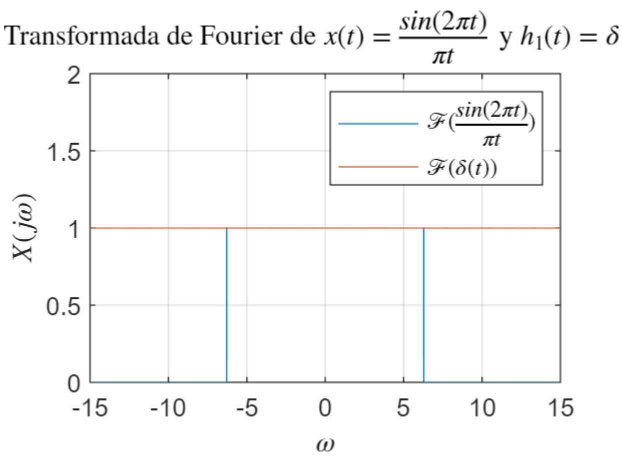
Siguiendo lo anterior, creamos en Matlab la Transformada de Fourier junto con su gráfico:

```
% Transformada de Fourier de x(t)

x_FT = zeros(1,length(t)); % Se crea un arreglo de 0 de largo t
for i = 1:length(t)        % Se recorre el arreglo de tiempos
    if abs(t(i)) <= w_i
        x_FT(i) = 1;       % Si |w| < W, entonces es 1
    end
end

% Gráfico de la Transformada de Fourier de x(t)

figure                     % Se inicializa una figura para graficar
plot(t,x_FT)               % Se crea el gráfico de la Transformada VS el tiempo
ylim([0 2])                % Se declaran los límites del eje y entre 0 y 2
xlim([-15 15])             % Se declaran los límites del eje x entre -15 y 15

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $x(t) = \frac{sin(2\pi t)}{\pi t}$', 'interpreter', 'latex')
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
grid on
```



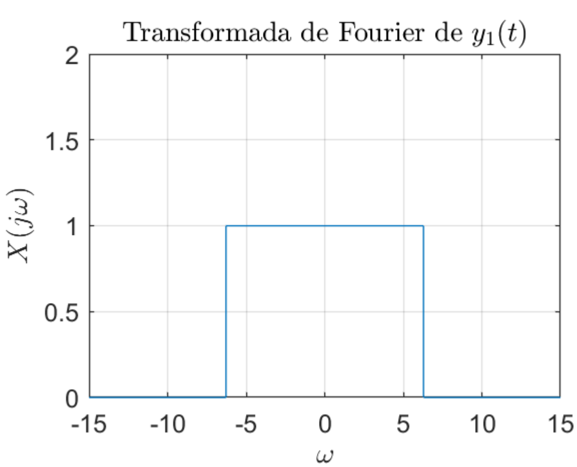
Luego, para el caso de $h_1(t)$, ya conocemos que la Transformada de Fourier de la Señal Impulso Unitario ($\delta(t)$) es 1. Por lo tanto, podemos graficarla junto con el gráfico anterior.

```
% Transformada de Fourier de la Señal Impulso Unitario (h1(t))

h1_FT = zeros(1,length(t));       % Se define un arreglo de 0s
for i = 1:length(t)               % Se recorre el arreglo de tiempos
    h1_FT(i) = 1;                 % La Transformada es 1 para todo t
end

% Gráfico de las Transformadas de Fourier de x(t) y h1(t)
figure           % Se inicializa una figura para graficar
plot(t,x_FT)     % Se grafica la Transformada de Fourier de x(t)
hold on          % Se utiliza hold on para no perder el gráfico anterior
plot(t, h1_FT)   % Se grafica la Transformada de Fourier de h1(t)
ylim([0 2])      % Se declara los límites del eje y entre 0 y 2
xlim([-15 15])   % Se declara los límites del eje x entre -15 y 15

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $x(t) = \frac{sin(2\pi t)}{\pi t}$ y $h_1(t) = \delta (t)$',
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
legend('$$\mathcal{F}(\frac{sin(2\pi t)}{\pi t})$$','$$\mathcal{F}(\delta(t))$$', 'interpreter
grid on
```



Ahora, aplicamos la multiplicación entre las Transformadas de Fourier de $x(t)$ y de $h_1(t)$ para obtener la Transformada de Fourier de $y_1(t)$.

```
% Transformada de Fourier de y1(t)

y1_FT = zeros(1,length(t));          % Se define un arreglo de 0s
for i = 1:length(t)                  % Se recorre el arreglo de tiempos
    y1_FT(i) = x_FT(i) * h1_FT(i);   % Se multiplica cada valor de x_FT con h1_FT
end

% Gráfico de la Transformada de Fourier de y1(t)

figure                      % Se inicializa una figura para graficar
plot(t, y1_FT)              % Se grafica y1_FT con t
ylim([0 2])                 % Se declaran los límites del eje y entre 0 y 2
xlim([-15 15])              % Se declaran los límites del eje x entre -15 y 15

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $y_1(t)$', 'interpreter', 'latex')
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
grid on
```



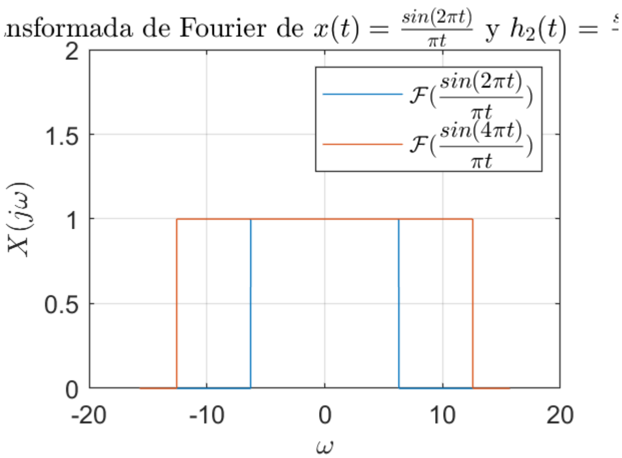
Si miramos bien el gráfico, nos daremos cuenta de que **obtuvimos exactamente el mismo gráfico de la Transformada de Fourier de** $x(t)$. Por lo tanto, eso nos da a concluir que la Anti Transformada de la Transformada de Fourier de $y_1(t)$ es $x(t)$. En otras palabras:

$$y_1(t) = x(t) = \frac{sin(2\pi t)}{\pi t}$$

El gráfico de $x(t)$ ya fue realizado más arriba, por lo que ya conocemos la forma de la Señal $y_1(t)$.

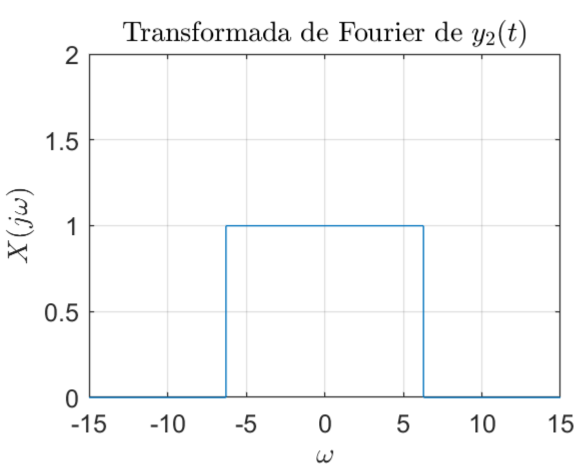
Ahora realizaremos el mismo proceso para obtener $y_2(t)$. Primero debemos obtener la Transformada de Fourier de $h_2(t)$.

```
% Transformada de Fourier de h2(t)

h2_FT = zeros(1,length(t));          % Se declara un arreglo de 0s
for i = 1:length(t)                  % Se recorre el arreglo de tiempos
    if abs(t(i)) < w_c
        h2_FT(i) = 1;                % Para cada valor de t que cumpla |w| < W, es 1
    end
end


% Gráfico de la Transformada de Fourier de h2(t) junto con x(t)

figure                               % Se inicializa una figura para graficar
plot(t,x_FT)                         % Se grafica t junto con la Transformada de x(t)
hold on                              % Se utiliza hold on para mantener el gráfico anterior
plot(t,h2_FT)                        % Se grafica t junto con la Transformada de h2(t)
ylim([0 2])                          % Se declaran los límites del eje y entre 0 y 2

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $x(t) = \frac{sin(2\pi t)}{\pi t}$ y $h_2(t)$ =  $\frac{sin(4
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
legend('$$\mathcal{F}(\frac{sin(2\pi t)}{\pi t})$$','$$\mathcal{F}(\frac{sin(4\pi t)}{\pi t})$$
grid on
```



Ahora, aplicamos la multiplicación entre las Transformadas de Fourier de $x(t)$ y de $h_2(t)$ para obtener la Transformada de Fourier de $y_2(t)$.

```
% Transformada de Fourier de y2(t)

y2_FT = zeros(1,length(t));          % Se declara un arreglo de 0s
for i = 1:length(t)                  % Se recorre el arreglo de tiempo
    y2_FT(i) = x_FT(i) * h2_FT(i);  % Se realiza la multiplicación de cada valor de x_FT y h2_F
end

% Gráfico de la Transformada de Fourier de y2(t)

figure                               % Se inicializa una figura para graficar
plot(t, y2_FT)                       % Se grafica t con la Transformada de y2(t)
ylim([0 2])                          % Se declaran los límites del eje y entre 0 y 2
xlim([-15 15])                       % Se declaran los límites del eje x entre -15 y 15

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $y_2(t)$', 'interpreter', 'latex')
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
grid on
```



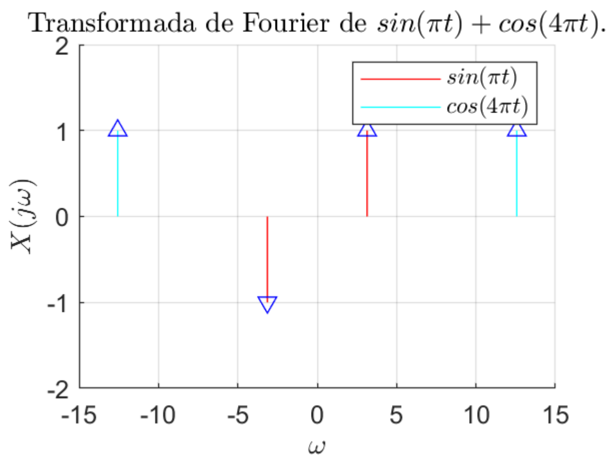
Nuevamente, **obtuvimos exactamente el mismo gráfico de la Transformada de Fourier de** $x(t)$. En otras palabras:

$$y_2(t) = y_1(t) = x(t) = \frac{sin(2\pi t)}{\pi t}$$

Repetimos el proceso para $y_3(t)$. Vamos a obtener la Transformada de Fourier de $h_3(t)$. Si nos damos cuenta $h_3(t)$ tiene una forma bastante similar a la Señal cuya Transformada de Fourier y Gráfico se realizaron en la Tarea 2 del curso. Por lo tanto, basándonos en esta, podemos realizar el gráfico de la Transformada de Fourier:

```
% Gráfico de Transformada de Fourier de sin(πt) + cos(4πt) (h3(t))

w_sin = pi;         % Definimos w_0, el cual para este caso es π
w_1s = -w_sin;        % Definimos w_1s, el cual equivale a -2π
w_2s = w_sin;         % Definimos w_2s el cual equivale a 2π

figure                % Inicializamos el gráfico
w = [w_1s w_1s];      % Se crea un array para graficar el primer resultado con w_1s
y = [0 -1];           % Se crea otro array con 0 y -1 para el primer resultado
linea_1s = line(w,y);  % Se grafica el primer resultado
linea_1s.Color = 'red';    % Se grafica la línea con color rojo
hold on               % Se utiliza hold on para que todo se grafique en un solo gráfico
plot(w_1s, -1, 'bv')  % Se grafica la flecha hacia abajo en el punto (-π, -1)
ylim([-2 2])          % Se definen los límites del eje y entre -2 y 2
xlim([-15 15])        % Se definen los límites del eje y entre -15 y 15
hold on               % Se utiliza hold on para que no se cree un nuevo gráfico
w = [w_2s w_2s];      % Se crea un array para graficar el segundo resultado con w_2s
y = [0 1];            % Se crea otro array entre con 0 y 1 para el segundo resultado
linea_2s = line(w,y);  % Se grafica el segundo resultado
linea_2s.Color = 'red';    % Se grafica la línea con color rojo
hold on               % Se utiliza hold on para que todo se grafique en un solo gráfico
plot(w_2s, 1, 'b^')   % Se grafica la flecha hacia arriba en el punto (π, 1)

w_cos = 4*pi;         % Definimos w_0, el cual para este caso es 4π
w_1c = -w_cos;        % Definimos w_1c, el cual equivale a -4π
w_2c = w_cos;         % Definimos w_2c el cual equivale a 4π

w = [w_1c w_1c];      % Se crea un array para graficar el primer resultado con w_1c
y = [0 1];            % Se crea otro array con 0 y 1 para el primer resultado
linea_1c = line(w,y); % Se grafica el primer resultado
linea_1c.Color = 'cyan';  % Se grafica la línea con color cian
hold on               % Se utiliza hold on para que todo se grafique en un solo gráfico
plot(w_1c, 1, 'b^')   % Se grafica la flecha hacia arriba en el punto (-4π, 1)
hold on               % Se utiliza hold on para que no se cree un nuevo gráfico
w = [w_2c w_2c];      % Se crea un array para graficar el segundo resultado con w_2c
y = [0 1];            % Se crea otro array entre con 0 y 1 para el segundo resultado
linea_2c = line(w,y);  % Se grafica el segundo resultado
```

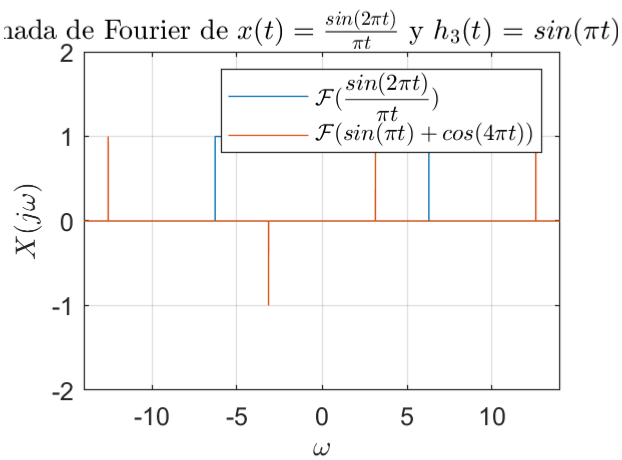
```
linea_2c.Color = 'cyan';  % Se grafica la línea con color cian
hold on                 % Se utiliza hold on para que todo se grafique en un solo gráfico
plot(w_2c, 1, 'b^')    % Se grafica la flecha hacia arriba en el punto (4π, 1)

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $sin(\pi t) + cos(4\pi t)$.', 'interpreter', 'latex')
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
legend('$$sin(\pi t)$$', '', '', '', '$$cos(4\pi t)$$','', 'interpreter', 'latex')
grid on
```



De lo anterior, podemos basarnos en el gráfico para poder obtener la Transformada de Fourier de $h_3(t)$ en forma de arreglo y así poder realizar la multiplicación con la Transformada de Fourier de $x(t)$.

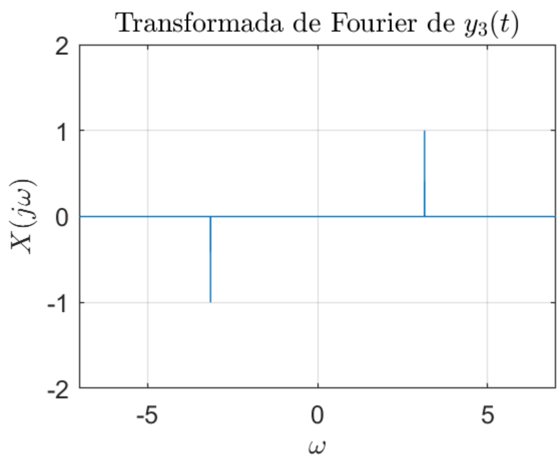
```
% Transformada de Fourier de h3(t)

h3_FT = zeros(1,length(t));             % Se inicializa un arreglo de 0s
for i = 1:length(t)                     % Se recorre el arreglo de 0s
    if t(i) == -4*pi                    % Si t == -4*pi, entonces es 1
        h3_FT(i) = 1;
    elseif t(i) == -pi                  % Si t == -pi, entonces es -1
            h3_FT(i) = -1;
    elseif t(i) == pi                   % Si t == pi, entonces es 1
            h3_FT(i) = 1;
    elseif t(i) == 4*pi                 % Si t == 4*pi, entonces es 1
            h3_FT(i) = 1;
    end
end


% Gráfico de la Transformada de Fourier de x(t) junto con h3(t)

figure                                  % Se inicializa una figura para poder graficar
plot(t, x_FT)                           % Se grafica t junto con x_FT
hold on                                 % Se utiliza hold on para no perder el gráfico anterior
plot(t,h3_FT)                           % Se realiza el grafico de 3 junto con h3_FT
```

```
ylim([-2 2])                            % Se declaran los límites del eje y entre -2 y 2
xlim([-14 14])                          % Se declaran los límites del eje x entre -14 y 14

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $x(t) = \frac{sin(2\pi t)}{\pi t}$ y $h_3(t)$ =  $sin(\pi t)
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
legend('$$\mathcal{F}(\frac{sin(2\pi t)}{\pi t})$$','$$\mathcal{F}(sin(\pi t) + cos(4\pi t))$$
grid on
```



Ahora queda realizar la multiplicación para encontrar la Transformada de Fourier de $y_3(t)$.

```
% Transformada de Fourier de y3(t)

y3_FT = zeros(1,length(t));            % Se define un arreglo de 0s
for i = 1:length(t)                    % Se recorre el tiempo
    y3_FT(i) = x_FT(i) * h3_FT(i);     % Se multiplica cada valor de x_FT con h3_FT
end

% Gráfico de la Transformada de Fourier de y3(t)

figure                                 % Se inicializa una figura para poder graficar
plot(t,y3_FT)                          % Se grafica t junto con y3_FT
ylim([-2 2])                           % Se declaran los límites del eje y entre -2 y 2
xlim([-7 7])                           % Se declaran los límites del eje x entre -7 y 7

% Configuraciones adicionales del gráfico
title('Transformada de Fourier de $y_3(t)$', 'interpreter', 'latex')
xlabel('$$\omega$$', 'interpreter', 'latex')
ylabel('$$X(j\omega)$$','interpreter','latex')
grid on
```

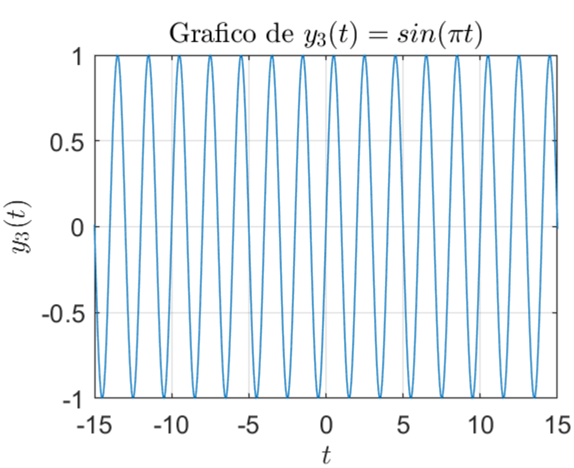


Notemos que el gráfico resultante es de una Transformada de Fourier conocida, pues se trata de la **Transformada de Fourier de** $sin(\pi t)$. Por lo tanto:

$$y_3(t) = sin(\pi t)$$

El gráfico de esta señal es el siguiente:

```
% Gráfico de y3(t) = sin(pi*t)

y3 = sin(pi*t);                         % Se define y3
figure                                  % Se inicializa una figura para graficar
plot(t, y3)                             % Se grafica t vs y3
xlim([-15 15])                          % Se declaran los límites del eje x entre -15 y 15

% Configuraciones adicionales del gráfico
title('Grafico de $y_3(t) = sin(\pi t)$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_3(t)$$','interpreter','latex')
grid on
```

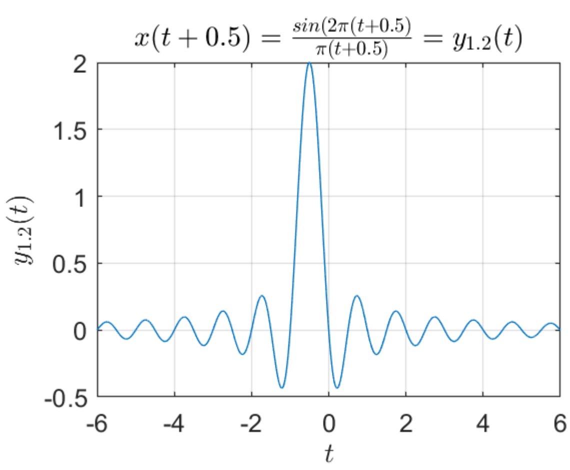


Problema N° 1 - Parte 2

Similar a la parte anterior, podemos aplicar la misma propiedad para así determinar la convolución. Sin embargo, del cálculo de $y_1(t)$, ya sabemos que al aplicar la propiedad de **"La Transformada de Fourier de una convolución es la multiplicación de las Transformadas de Fourier individuales de las funciones"** donde una de estas funciones es la Señal Impulso Unitario, nos da como resultado que la convolución entre la Señal Impulso Unitario junto con otra función es igual a esa otra función que no es la Señal Impulso Unitario. Por lo tanto, no necesitamos sacar la Transformada de Fourier de $x(t+0.5)$ para saber que la convolución entre $x(t+0.5)$ y $\delta(t)$ es exactamente $x(t+0.5)$. Por lo que solo nos quedaría definir $x(t+0.5)$ y graficarla para conocer como es la nueva Señal $y_{1.2}(t)$.

$$x(t+0.5) * \delta(t) = x(t+0.5) = y_{1.2}(t)$$

```
% Convolución de x(t+0.5) con la Señal Impulso Unitario

t2 = t + 0.5;                       % Se define t+0.5
x2 = zeros(1,length(t2));           % Se define un arreglo de 0s
for i = 1:length(t2)                % Se calcula el valor de x con el nuevo t
    x2(i) = (sin(w_i*t2(i)))/(pi*t2(i));
end

% Gráfico del resultado de la convolución (x(t+0.5))

figure                              % Se inicializa una figura para graficar
plot(t, x2)                         % Se grafica x(t+0.5)
xlim([-6 6])                        % Se declaran los límites del eje x entre -6 y 6

% Configuraciones adicionales del gráfico
title('$x(t+0.5) = \frac{sin(2\pi (t+0.5)}{\pi (t+0.5)} = y_{1.2}(t)$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_{1.2}(t)$$','interpreter','latex')
grid on
```



Si observamos bien el gráfico, se puede destacar de que al haber agregado 0.5 al tiempo, **la señal se desplazó hacia a la izquierda**, por lo tanto, ya no está centrada en el origen, si no que esta centrada levemente entre -2 y aproximadamente -1.

Problema N°2

Para esta parte, se pide utilizar la función sinc. Sinc corresponde al Seno Cardinal y se define de la siguiente manera:

$$sinc(x) = \begin{cases} 1, & x = 0 \\ \dfrac{sin(x\pi)}{\pi x}, & x \neq 0 \end{cases}$$

Basándonos en lo anterior, es posible reescribir $x(t)$ y $h_2(t)$ para así evitar la indeterminación cuando t = 0.

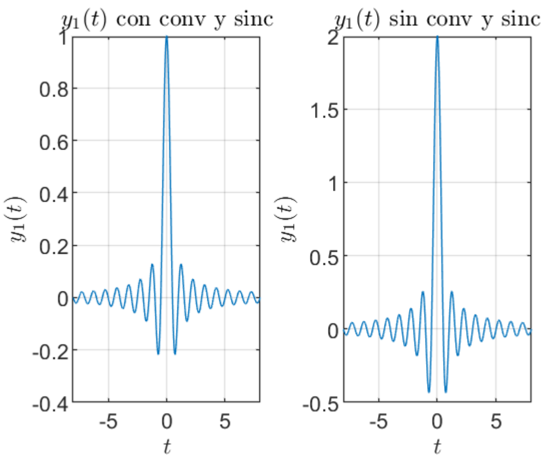
$$x(t) = \frac{sin(2\pi t)}{\pi t} = sinc(2t) \cdot 2$$

$$h_2(t) = \frac{sin(4\pi t)}{\pi t} = sinc(4t) \cdot 4$$

Ahora, podemos utilizar la función conv() para comprobar las convoluciones del problema anterior. Vamos a comenzar obteniendo $y_1(t)$ y graficándola lado a lado con la señal que obtuvimos en el problema anterior.

```
% Convoluciones de x(t) y h1(t) con conv y sinc

xsinc = sinc(2*t) * 2;        % Se define x(t) con sinc

y1 = conv(h1, xsinc, 'same'); % Se realiza la convolución con conv y sinc
y1 = y1 / max(y1);            % Se normalizan los valores de la convolución

% Gráfico lado a lado de y1(t) con conv y sinc y sin conv y sin sinc

figure                        % Se inicializa una figura para poder graficar
subplot(1,2,1)                % Se inicializa un subgrafico
plot(t, y1)                   % Se grafica y1 (con conv y sinc) en el primer subgrafico
xlim([-8 8])                  % Se declaran los límites del eje x entre -8 y 8

% Configuraciones adicionales del subgráfico 1
title('$y_1(t)$ con conv y sinc', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_1(t)$$','interpreter','latex')
grid on


subplot(1,2,2)                % Se inicializa un segundo subgráfico
```

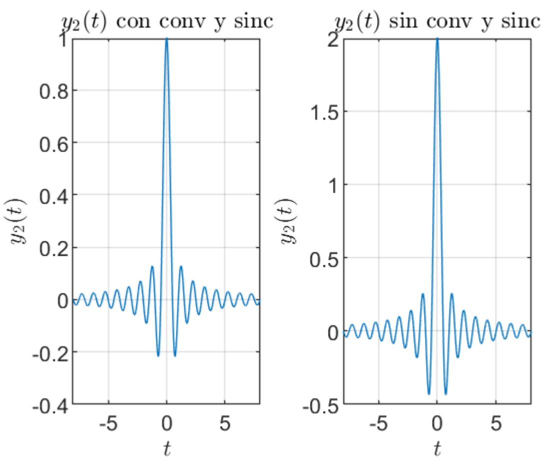
```
plot(t,x)                       % Se grafica y1 (sin conv y sin sinc) en el segundo subgrafico
xlim([-8 8])                    % Se declaran los límites del eje x entre -8 y 8

% Configuraciones adicionales del subgráfico 2
title('$y_1(t)$ sin conv y sinc', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_1(t)$$','interpreter','latex')
grid on
```



Claramente hay una diferencia en el eje y, **pues la amplitud de $y_1(t)$ al utilizar conv y sinc disminuye a 1,** mientras que sin conv y sinc la amplitud es 2. Sin embargo, con esa diferencia de lado, sigue siendo la misma señal. Ahora vamos a comprobar $y_2(t)$.

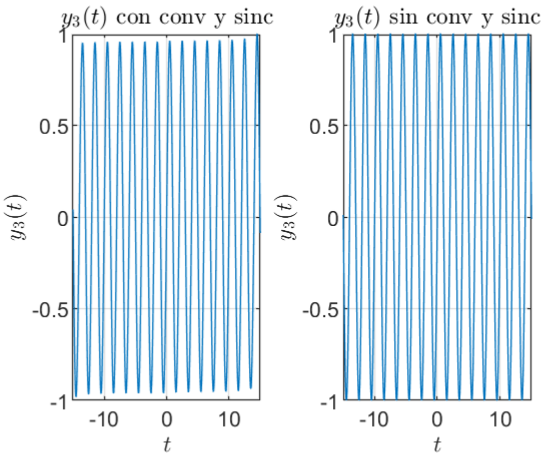
```
% Convolución de x(t) y h2() con conv y sinc

h2sinc = sinc(4*t)*4;     % Se define h2 utilizando sinc

y2 = conv(h2sinc, xsinc, 'same');  % Se realiza la convolución con conv
y2 = y2 / max(y2);                 % Se normalizan los valores de y2

% Gráfico lado a lado de y2 con conv y sinc y sin conv y sin sinc

figure                    % Se inicializa una figura para gráficar
subplot(1,2,1)            % Se inicializa un subgráfico
plot(t, y2)               % Se grafica y2 (con conv y sinc)
xlim([-8 8])              % Se definen los límites del eje x entre -8 y 8

% Configuraciones adicionales del subgráfico
title('$y_2(t)$ con conv y sinc', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_2(t)$$','interpreter','latex')
grid on

subplot(1,2,2)            % Se inicializa un subgráfico
plot(t,x)                 % Se grafica y2 (sin conv y sin sinc)
xlim([-8 8])              % Se definen los límites del eje x entre -8 y 8
```

```
% Configuraciones adicionales del subgráfico
title('$y_2(t)$ sin conv y sinc', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_2(t)$$','interpreter','latex')
grid on
```



De los gráficos podemos observar que **se repite exactamente el mismo caso anterior**, donde la amplitud de $y_2(t)$ disminuye a 1 al utilizar conv y sinc, pero sigue siendo la misma señal. Ahora realizaremos la comprobación de $y_3(t)$.

```
% Convolución de x(t) y h3() con conv y sinc

y3 = conv(h3, xsinc, 'same');  % Se realiza la convolución con conv
y3 = y3 / max(y3);             % Se normalizan los valores de y3(t)

figure                         % Se inicializa una figura para graficar
subplot(1,2,1)                 % Se inicializa un subgráfico
plot(t, y3)                    % Se grafica y3 (con conv y sinc)
xlim([-15 15])                 % Se definen los límites del eje x entre -15 y 15

% Configuraciones adicionales del subgráfico
title('$y_3(t)$ con conv y sinc', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_3(t)$$','interpreter','latex')
grid on

subplot(1,2,2)                 % Se inicializa un subgráfico
plot(t,sin(pi*t))              % Se grafica y3 (sin conv y sin sinc)
xlim([-15 15])                 % Se definen los límites del eje x entre -15 y 15

% Configuraciones adicionales del subgráfico
title('$y_3(t)$ sin conv y sinc', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_3(t)$$','interpreter','latex')
grid on
```



Para este caso, podemos notar diferencias muy leves, principalmente en la amplitud, donde **vemos que la amplitud de $y_3(t)$ con conv y sinc se acerca a 1, pero no alcanzar a tocarlo.** En cambio, en el caso sin conv y sinc, la amplitud es exactamente 1.

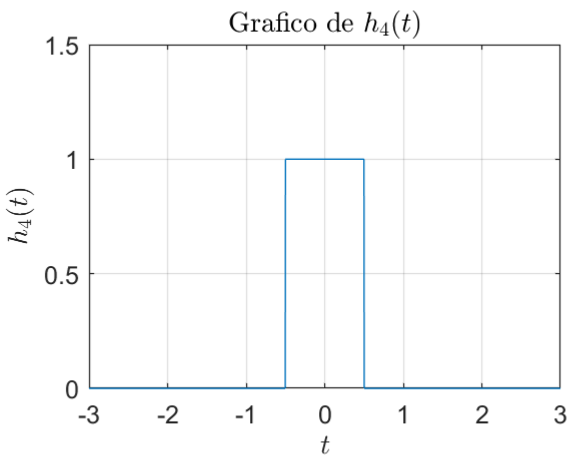
De las comprobaciones anteriores, podemos concluir que aplicar la propiedad si sirve para poder realizar una convolución sin tener que utilizar la función conv() de Matlab.

Por último, podemos realizar el proceso de calcular conv para $y_4(t)$, donde $h_4(t)$ es:

$$h_4(t) = \begin{cases} 1, & |t| < 0.5 \\ 0, & |t| > 0.5 \end{cases}$$

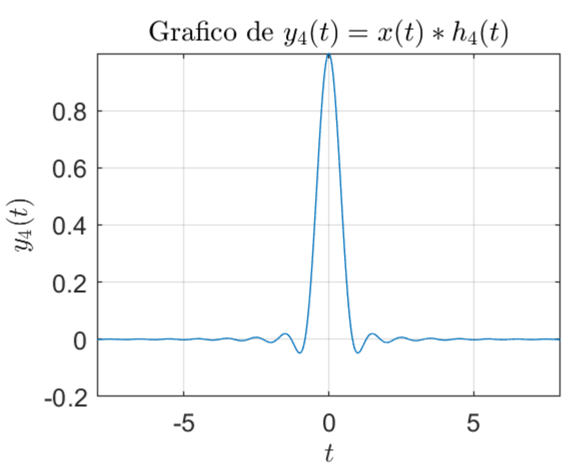
Podemos graficar la señal anterior para conocer bien con que estamos trabajando.

```
% Señal h4(t)

h4 = zeros(1,length(t)); % Se crea un arreglo de 0 de largo t
for i = 1:length(t)      % Se recorre el arreglo de tiempos
    if abs(t(i)) < 0.5
        h4(i) = 1;       % Si |t| < 0.5, entonces es 1
    end
end

% Gráfico de la señal h4(t)

figure                   % Se inicializa una figura para poder graficar
plot(t,h4)               % Se grafica h4(t)
xlim([-3 3])             % Se definen los límites del eje x entre -3 y 3
ylim([0 1.5])            % Se definen los límites del eje y entre 0 y 1.5

% Configuraciones adicionales del gráfico
title('Grafico de $h_4(t)$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$h_4(t)$$','interpreter','latex')
grid on
```



Ahora vamos a realizar la convolución para obtener $y_4(t) = x(t) * h_4(t)$. Además, vamos a graficarla para ver qué forma tiene la señal.

```
% Convolución de x(t) y h4(t)

y4 = conv(h4, xsinc,'same');  % Se realiza la convolución entre x(t) y h4(t) (con conv)
y4 = y4/max(y4);              % Se normalizan los valores de y4

% Gráfico de y4(t)

figure                        % Se inicializa una figura para graficar
plot(t,y4)                    % Se grafica y4(t)
xlim([-8 8])                  % Se definen los límites del eje x entre -8 y 8

% Configuraciones adicionales del gráfico
title('Grafico de $y_4(t) = x(t) * h_4(t)$', 'interpreter', 'latex')
xlabel('$$t$$', 'interpreter', 'latex')
ylabel('$$y_4(t)$$','interpreter','latex')
grid on
```



Si analizamos el gráfico anterior, podemos ver **que se parece un poco a** $y_1(t)$ e $y_2(t)$, con la diferencia **de que las oscilaciones cuando t < -0.5 y t > 0.5 no se notan tanto,** siendo más bien planas, a diferencias de las señales mencionadas anteriormente, donde claramente se pueden ver varias oscilaciones cuando $t < -0.5$ y $t > 0.5$.