Signals and Systems

MATLAB Assignment

Name - Devansh Tanna

ID No - 2019A3PS0158P

Tutorial Section No - 5

MATLAB Assignment - SAS

Question-1

1a)

Generating Signal

$$x(t) = \begin{cases} A\cos(2\pi f t) & 0 < t < 3\text{ms}, \\ 0 & 3\text{ ms} < t < 5\text{ ms} \end{cases}$$

Here

- Sampling Frequency `Fs = 800 kHz`
- Frequency `f = 1158 Hz`

Figure

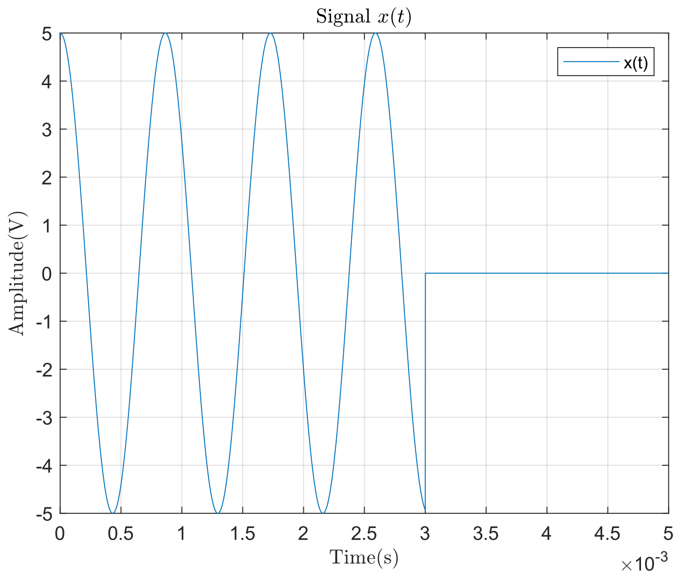


Figure-1 : Original Signal

1b)

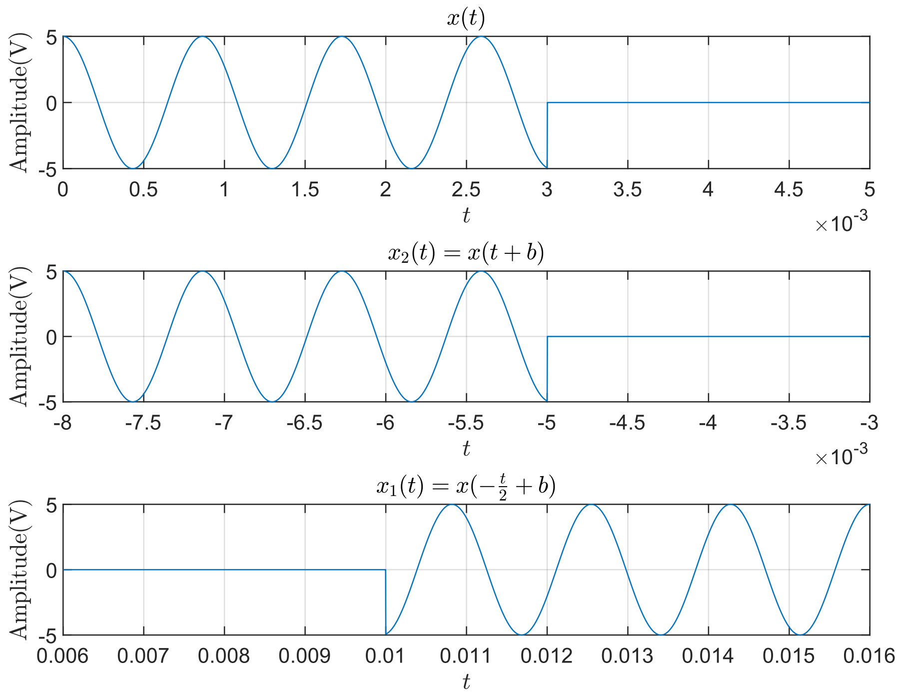
generating new signal $x_1(t)$:

$x_1(t) = x(\frac{-t}{2} + b)$, $b = 8$ms

Here following transforms have been performed:

1. $x(t) \to x(t+b)$: $x(t)$ is shifted by b units
2. $x(t+b) \to x(-\frac{t}{2}+b)$: $x(t+b)$ is reflected about Y-axis and time-scale is scaled up by 2

Figure



Transformed Signals $x(t)$, $x(t+b)$, $x_1(t)$

MATLAB Code

```
tstart = 0;
tend = 5e-3;
tmid = 3e-3;
A = 5; %% Amplitude
Fs = 800e3; %% Sampling Frequency
f = 1158; %% Signal Frequency
tp = tstart:1/Fs:tmid; %% 0 ≤ t < 3ses
ts = tstart:1/Fs:tend; %% 0 ≤ t ≤ 5sec
x = A*cos(2*pi*f*tp); %% A*cos(2π*f*t)
x = [x zeros([1 (tend-tmid)*Fs])]; %% for 3 ≤ t ≤ 5

figure %% initializing figure
plot(ts, x); xlabel("Time(s)");ylabel("Amplitude(V)");legend("x(t)"); grid;title("Signal $x(t)$", "Interpreter","latex")

b = 8e-3; %% Parameter "b" -> ID : 2019A3PS0158P => b = 8ms
t3 = (b-ts)*2; %% Tranformed time-range for signal x₁(t) = x(-t/2 + b)
t2 = ts-b; %% Transformed time-range for signal  x₂(t) = x(t+b)

figure %% initializing figure

subplot(3,1,1) %% Plot of x(t)
plot(ts, x); title("$x(t)$", "Interpreter","latex")
ylabel("Amplitude(V)","Interpreter","latex")
xlabel("$t$","Interpreter","latex")
grid

subplot(3,1,2) %% Plot of x₂(t) = x(t+b)
plot(t2, x); title("$x_2(t) = x(t + b)$", "Interpreter","latex")
ylabel("Amplitude(V)","Interpreter","latex")
xlabel("$t$","Interpreter","latex")
grid

subplot(3,1,3) %% Plot of x₁(t) = x(-t/2 + b)
plot(t3, x); title("$x_1(t) = x(-\frac{t}{2} + b)$", "Interpreter","latex")
grid
ylabel("Amplitude(V)","Interpreter","latex")
xlabel("$t$","Interpreter","latex")
```

To plot the $y(t) = x(at + t_0)$, we have to find the significant time range, note that signal $x(t)$ is only significant in range $[t_1, t_2]$ ms, so significant time range can be found by substituting

$$\tau = \frac{t - t_0}{a}$$

where $t \in [t_1, t_2]$, so $\tau \in [\frac{t_1 - t_0}{a}, \frac{t_2 - t_0}{a}]$ if $a > 0$, otherwise $\tau \in [\frac{t_2 - t_0}{|a|}, \frac{t_1 - t_0}{|a|}]$

Question-2

According to ID number pattern x(t) will have zeros at -2.5 and 2, and poles at -5 and $-8 + j2$.

Here it is given that $x(t)$ is real signal, so there must be another pole which is conjugate of $(-8 + j2)$, i.e. $(-8 - j2)$.

So $X(s)$ can be written as,

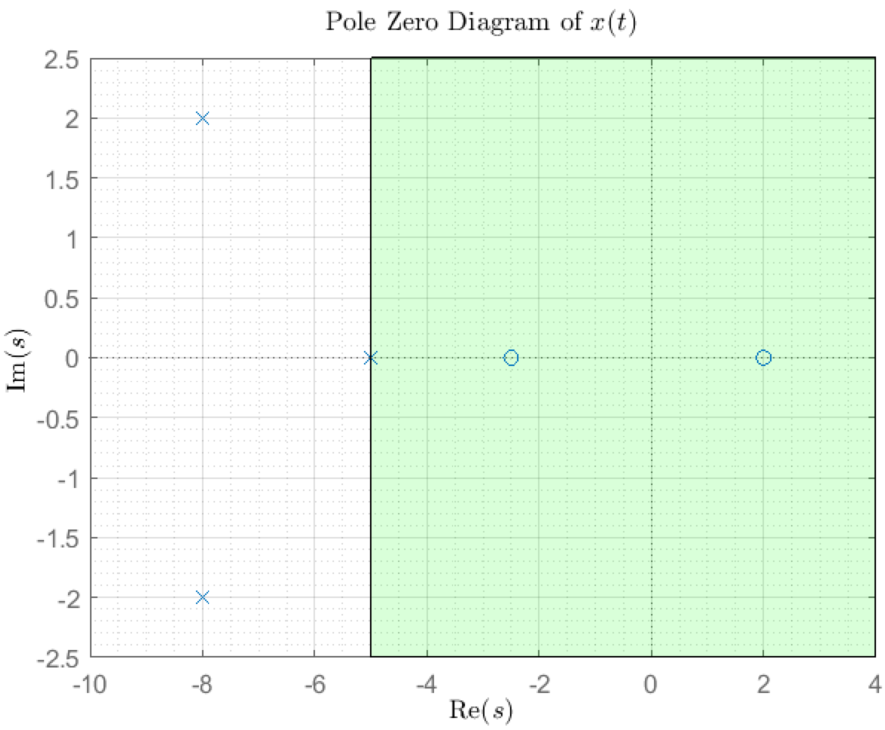
$$X(s) = A \frac{(s + 2.5)(s - 2)}{(s + 5)(s + 8 - j2)(s + 8 + j2)}$$

where A is some constant.

Assuming $A = 1$ for simplicity,

Pole Zero Diagram

Figure



As given, ROC for this signal $x(t)$, is given right to the <u>-X i.e. -5</u> (As shaded in figure).

Here we can observe that ROC is right to the rightmost pole, and Laplace transform of signal $x(t)$ i.e. $X(s)$ is rational, so we can conclude that signal $x(t)$ is right sided

MATLAB Code

```
s = tf('s'); % 's' parameter

% Poles = [-5, -8 + j2]
% Zeros = [-2.5, 2]

sys = (s+2.5)*(s-2)/(s+5)/(s+8-2i)/(s+8+2i);

% sys =
%              (s + 2.5)(s - 2)
%   --------------------------------------
%      (s + 5)(s + 8 - 2j)(s + 8 + 2j)
%
% Continous Time Transfer Function

figure;
h = pzplot(sys); % Generating pole-zero plot handle

h.AxesGrid.YUnits = '';
h.AxesGrid.XUnits = '';
h.AxesGrid.BackgroundAxes.Title.Interpreter = 'Latex';
h.AxesGrid.BackgroundAxes.XLabel.Interpreter = 'Latex';
h.AxesGrid.BackgroundAxes.YLabel.Interpreter = 'Latex';

p = getoptions(h); % getting properties option for customizing plot
p.XLim = {[-10, 4]}; % Setting X-Limits
p.YLim = {[-2.5, 2.5]}; % Setting Y-Limits

p.Title.String = "Pole Zero Diagram of $x(t)$";
p.Title.Interpreter = 'latex';

p.XLabel.Interpreter = 'latex';
```

```
p.XLabel.String = "$\mathrm{Re}(s)$";
p.YLabel.String = "$\mathrm{Im}(s)$";
p.YLabel.Interpreter = "latex";

setoptions(h, p); % Applying properties

ax = gca; % getting current axes for setting up Cartesian Grid
ax.XGrid = 'on'; ax.XMinorGrid = 'on'; % Turning Grid Lines On
ax.YGrid = 'on'; ax.YMinorGrid = 'on';

hold on
patch([-5 4 4 -5], [-2.5 -2.5 2.5 2.5], 'green', 'FaceAlpha', 0.15) % Drawing ROC region
hold off
```

Question-3

According to ID-No, $t_1 = 5$, $t_2 = 8$,

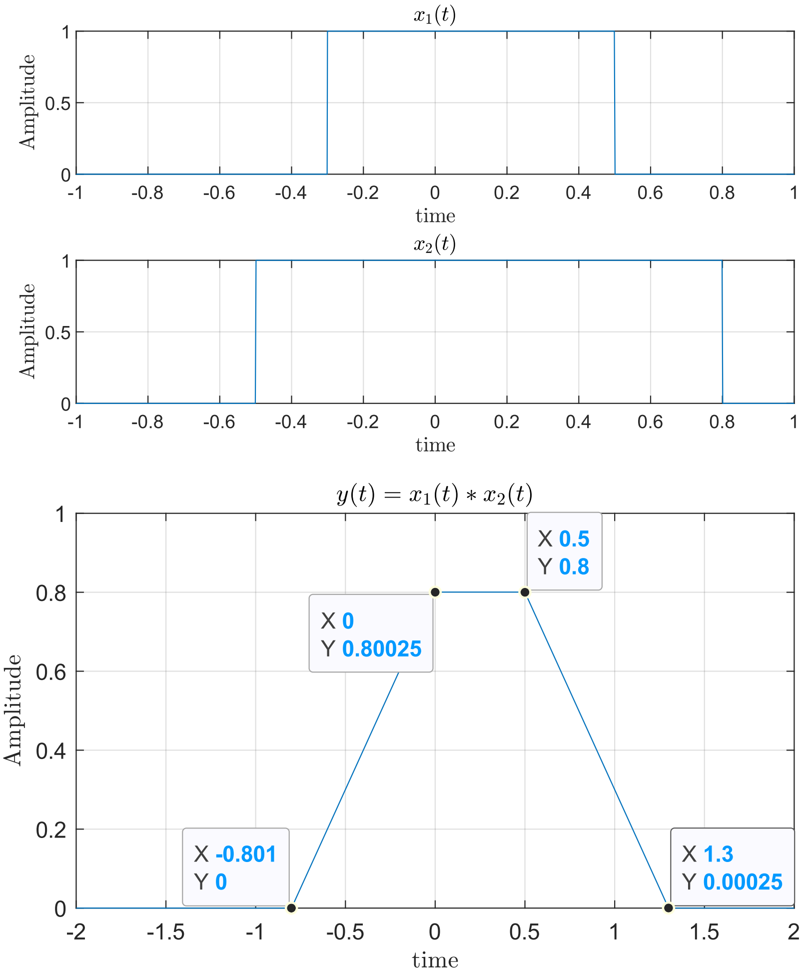
So signals $x_1(t)$ and $x_2(t)$ will be,

$$
\begin{aligned}
x_1(t) = u(t + 0.3) - u(t - 0.5) \\
x_2(t) = u(t + 0.5) - u(t - 0.8)
\end{aligned}
$$

where time vector, $t = -1 : 0.001 : 1$.

Here convolution of signal is given by $y(t) = x_1(t) * x_2(t)$,

Figure



Plot of $x_1(t)$, $x_2(t)$ and $y(t) = x_1(t) * x_2(t)$

Assuming $x_1(t-\tau)$ as moving signal sliding over $x_2(t)$

1. For $t \leq -0.8$

   There will be no overlap $\Rightarrow y(t) = 0$

2. For $0.8 < t \leq 0$,

   x_1 will be overlapped on x_2 for $\underline{-0.5 \text{ to } t+0.3}$.

   $\Rightarrow$ So the integral will be $t + 0.8$

3. For $0 < t \leq 0.5$

   x_1 will be completely overlapped on x_2.

   $\Rightarrow$ So the integral will be $(0.5 - (-0.3)) = 0.8$

4. For $0.5 < t \leq 1.3$

   x_1 will be overlapped on x_2 for $\underline{t - 0.5 \text{ to } 0.8}$

   $\Rightarrow$ So the integral will be $1.3 - t$

So by summing up,

$$y(t) = \begin{cases} 0, & t \le -0.8 \\ t + 0.8, & 0.8 < t \le 0 \\ 0.8, & 0 < t \le 0.5 \\ 1.3 - t, & 0.5 < t \le 1.3 \end{cases}$$

MATLAB Code

```
clear all;
clf;
clc;

t1 = 5; % Parameter
t2 = 8; % Parameter
t = -1:0.001:1; % Time range

x1 = heaviside(t + 0.3) - heaviside(t - 0.1*t1); % Signal x₁(t) = u(t+0.3) - u(t - 0.1t₁)
x2 = heaviside(t + 0.5) - heaviside(t - 0.1*t2); % Signal x₂(t) = u(t+0.5) - u(t - 0.1t₂)
y = conv(x1, x2)*0.001; % Convolution of x₁(t) and x₂(t)
tnet = -2:0.001:2; % Time vector for y(t),
% which can be obtained by summing endpoints of both the signals

f = gcf; % current figure
f.Position = f.Position + [0 -200 0 200];
ax1 = subplot(4,1,1);
p1 = plot(t, x1); % Plotting x₁(t)
grid on;
xlabel('time', "Interpreter","latex");
ylabel('Amplitude', "Interpreter","latex");
title("$x_1(t)$", "Interpreter","latex");

ax2 = subplot(4,1,2);
p2 = plot(t, x2); % Plotting x₂(t)
grid on;
xlabel('time', "Interpreter","latex");
ylabel('Amplitude', "Interpreter","latex");
title("$x_2(t)$", "Interpreter","latex");

ax3 = subplot(4,1,[3, 4]); % Merging two tiles for better visibility
p3 = plot(tnet, y); % Ploting y(t) = x1(t)*x2(t)
datatip(p3, -0.8, 0, 'Location', 'northwest');
datatip(p3, 0, 0.8, 'Location','southwest');
datatip(p3, 0.5, 0.8, 'Location','northeast');
datatip(p3, 1.3, 0, 'Location','northeast');
ax3pos = ax3.Position + [0 -0.03 0 0];

xlabel(ax3, 'time', "Interpreter","latex");
ylabel(ax3, 'Amplitude', "Interpreter","latex");
title(ax3, "$y(t)=x_1(t)*x_2(t)$", "Interpreter","latex");
ax3.XGrid = 'on';
ax3.YGrid = 'on';
ax3.Position = ax3pos;
```

Question-4

Butterworth filter $B(j\omega)$ is low-pass filter with cut-off frequency ω_c, with poles given as

$$p_k = \omega_c \exp\left(i\left(\frac{\pi(2k-1)}{2N} + \frac{\pi}{2}\right)\right), \qquad k = 1, 2 ... N$$

And $B(s)B(-s)$ is defined as,

$$B(s)B(-s) = |B(s)|^2 = \frac{1}{1 + \left(\frac{s}{j\omega_c}\right)^{2N}}$$

$B(j\omega)$ can be obtained by considering poles only of left half of the s-plane, because our system should be stable and causal so ROC must contain $j\omega$ axis,

So,

$$B(s) = \frac{A}{(s - p_1)(s - p_2)(s - p_3)...(s - p_k)}$$

Here $\omega_c = 1$ rad/s, so value of $A = 1$

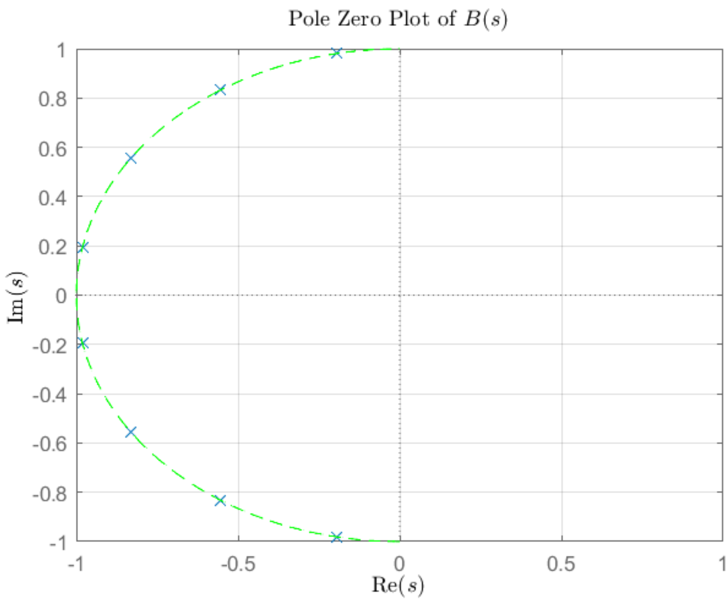
And $B(s)B(-s)$ will have these poles,

$$p_k = \omega_c \exp\left(i\left(\frac{\pi(2k-1)}{2N} + \frac{\pi}{2}\right)\right), \qquad k = 1, 2, ..., 2N$$

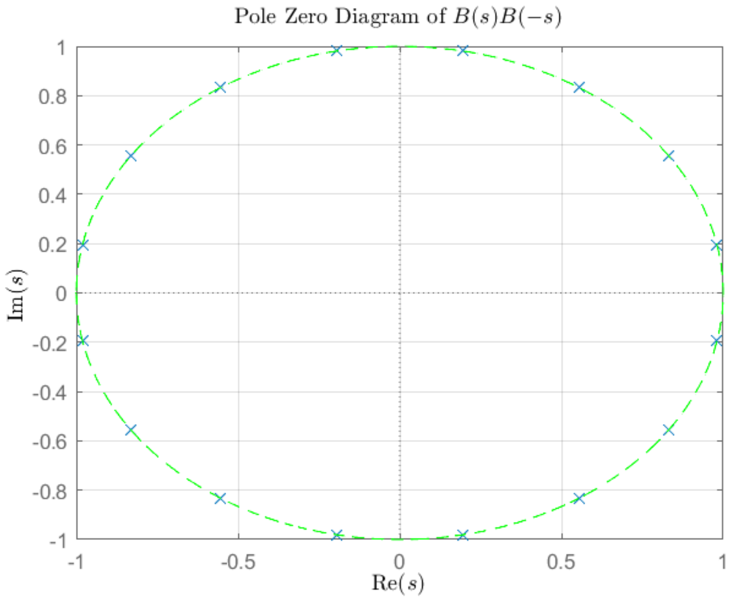
$B(s)$ for $N = 8$ will be,

$$B(s) = \frac{1}{s^8 + 5.126s^7 + 13.14s^6 + 21.85s^5 + 25.69s^4 + 21.85s^3 + 13.14s^2 + 5.126s + 1}$$

Figure



Pole-Zero plot of $B(s)$



Pole-Zero plot of $B(s)B(-s) = |B(s)|^2$

MATLAB Code

```
clear all;
clf;
clc;
s = tf('s');
N = 8; % Order of Butterworth Filter
p2 = exp(sqrt(-1)*(pi*(1:2:2*N-1)/(2*N) + pi/2)); % Poles of butterworth filter
Bs = 1;
for j = 1:N
    Bs = Bs * (1/(s-p2(j)));
end
b1 = Bs.Denominator{1};
Bs.Denominator = {real(Bs.Denominator{1})};  % ignoring imaginary terms because of order of 1e-15

% 4a) ----------------
h1 = pzplot(Bs); % Plotting Pole-Zero of B(s)
h1.AxesGrid.YUnits = '';
h1.AxesGrid.XUnits = '';
h1.AxesGrid.BackgroundAxes.Title.Interpreter = 'Latex';
h1.AxesGrid.BackgroundAxes.XLabel.Interpreter = 'Latex';
h1.AxesGrid.BackgroundAxes.YLabel.Interpreter = 'Latex';
p1 = getoptions(h1);
p1.Title.String = "Pole Zero Plot of $B(s)$";
p1.XLabel.Interpreter = "latex";
p1.XLabel.String = "$\mathrm{Re}(s)$";
p1.Title.Interpreter = "latex";
p1.YLabel.String = "$\mathrm{Im}(s)$";
p1.YLabel.Interpreter = 'latex';
p1.XLim = {[-1 1]};

ax1 = gca;
ax1.XGrid = 'on';
ax1.YGrid = 'on';
hold on;
theta = -pi/2:0.001:pi/2;
plot(-cos(theta), sin(theta), 'g--'); % Drawing left-half circle
hold off
setoptions(h1, p1);
Bs_ = 1; % B(-s)
for j = 1:N
    Bs_ = Bs_ * (1/(-s - p2(j)));
end
Bs_.Denominator = {real(Bs_.Denominator{1})};

B = Bs*Bs_;  % B(s)B(-s)
```

```
% 4b) ------------
figure;
h3 = pzplot(B);
B.Denominator = {B.Denominator{1}.*(abs(B.Denominator{1})>1e-7)}; %ignoring small terms

h3.AxesGrid.YUnits = '';
h3.AxesGrid.XUnits = '';
h3.AxesGrid.BackgroundAxes.Title.Interpreter = 'Latex';
h3.AxesGrid.BackgroundAxes.XLabel.Interpreter = 'Latex';
h3.AxesGrid.BackgroundAxes.YLabel.Interpreter = 'Latex';
p3 = getoptions(h3);
p3.Title.String = "Pole Zero Diagram of $B(s)B(-s)$";
p3.Title.Interpreter = 'latex';
p3.XLabel.Interpreter = 'latex';
p3.XLabel.String = "$\mathrm{Re}(s)$";
p3.YLabel.String = "$\mathrm{Im}(s)$";
p3.YLabel.Interpreter = "latex";
ax2 = gca;
ax2.XGrid = 'on';
ax2.YGrid = 'on';

hold on;
theta = -pi:0.001:pi;
plot(cos(theta), sin(theta), 'g--');
hold off
setoptions(h3, p3);

% 4c) ---------------------
  display(Bs) % Displaying B(s)
```