Familiarization with Basic CT/DT Functions

Lab Exercises on December 5, 2021

Department of Electronics and Computer Engineering

Pulchowk Campus, Lalitpur

Ashlesh Pandey

PUL074BEX007

1 Objectives

- Familiarization and review of basic continuous time and discrete time functions.

2 Background Theory

2.1 Some MATLAB commands

Command	Use Case
`who`	Lists current variables
`whos`	Lists current variables (long display)
`input`	Requests user input
`disp`	Displays contents of an array or string
`clear all`	Remove variables from the current workspace
`close all`	Close all plots
`pause`	Pause execution
`home`	Send cursor home
`length`	Computes number of elements
`plot`	Generates xy plot
`tiledlayout`	Creates a tiled chart layout for displaying multiple plots
`nexttile`	Creates an axes object and places it into the next empty tile
`hold`	Freezes current plot
`grid`	Displays grid lines
`stem`	Creates stem plot
`legend`	Places legend on figure
`xlabel`	Adds text label to x-axis
`ylabel`	Adds text label to y-axis

`real`	Real part of complex number
`imag`	Imaginary part of complex number
`abs`	Absolute value and complex magnitude
`angle`	Phase angle in interval $[-\pi, \pi]$
`zeros`	Creates an array of zeros
`ones`	Creates an array of ones
`exp`	Returns the exponential of arguments
`sin`	Returns the sine of argument in radians
`cos`	Returns the cosine of argument in radians

Table 1: Some MATLAB commands with their use cases

3 Lab Exercises

Problem 1

Plot the baic signal using MATLAB

a. Impulse Response

```
function y = impulse_response(x)
 y = x==0;
end
```

Listing 1: Matlab function to return impulse response

b. Unit-step

```
function y = unit_response(x)
    y = x>=0;
end
```

Listing 2: Matlab function to return unit-step response

c. Ramp

```
function y = ramp_response(x)
  y = 2 .* x;
end
```

Listing 3: Matlab function to return ramp response

d. Rectangular

```
function y = rectangular_response(x)
  y = (x>=-1/2) & (x<=1/2);
end
```

Listing 4: Matlab function to return rectangular response

```
function y = basic_plot_selector(basic_plot_name,x)
  switch basic_plot_name
      case 'impulse'
          y=impulse_response(x);
      case {'unit','unit-step'}
          y=unit_response(x);
      case 'ramp'
          y=ramp_response(x);
      case {'rectangular','rect'}
          y=rectangular_response(x/2);
      otherwise
          error('The function you want is not found.');
  end
end
```

Listing 5: Matlab function to return desired response based on user input

```
basic_plot_name=input('Enter function you want to plot: ','s');
n=-5:5;
t=-5:0.01:5;
y=basic_plot_selector(basic_plot_name,t);
z=basic_plot_selector(basic_plot_name,n);
l=tiledlayout(2,1);
title(l,{sprintf('%s',['Plot for ',basic_plot_name, ' function'])
    ,'(PUL074BEX007)'})
nexttile
plot(t,y,'LineWidth',1.5)
xlabel('t','interpreter','latex')
```

```
11 ylabel('y(t)','interpreter','latex
       ')
12 title('Continuous Time Function')
13 nexttile
14 stem(n,z,'Linewidth', 1.5)
15 xlabel('n','interpreter','latex')
16 ylabel('y[n]','interpreter','latex
       ')
17 title('Discrete Time Function')
18 print(basic_plot_name,'-depsc')
```

Listing 6: Matlab script to plot desired response based on user input

```
>> lab_2_1
Enter function you want to plot: impulse
```

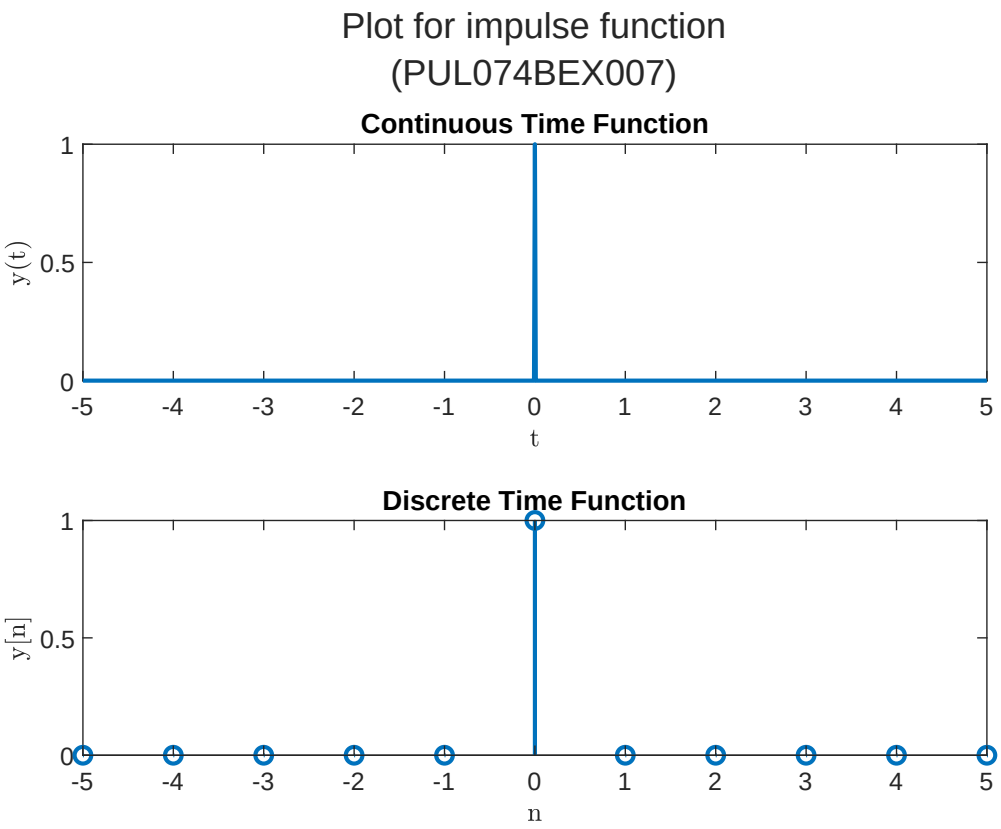

Figure 1: Plot for impulse function

```
>> lab_2_1

Enter function you want to plot: unit-step
```

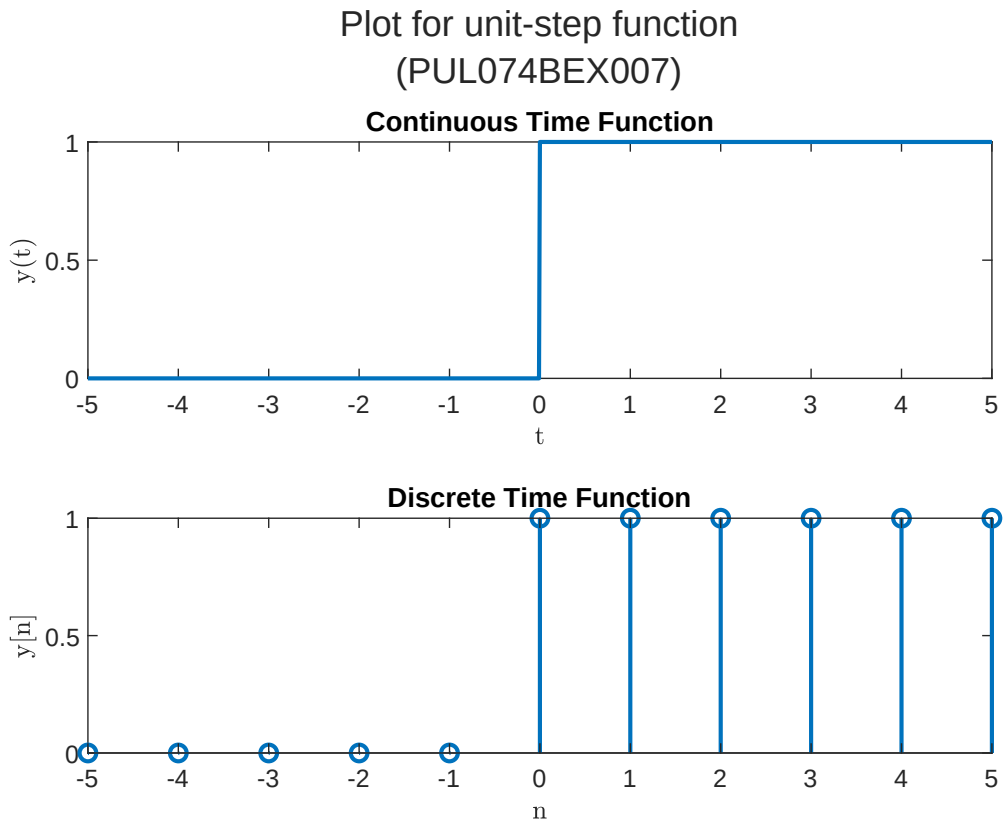

Figure 2: Plot for unit-step function

```
>> lab_2_1

Enter function you want to plot: ramp
```

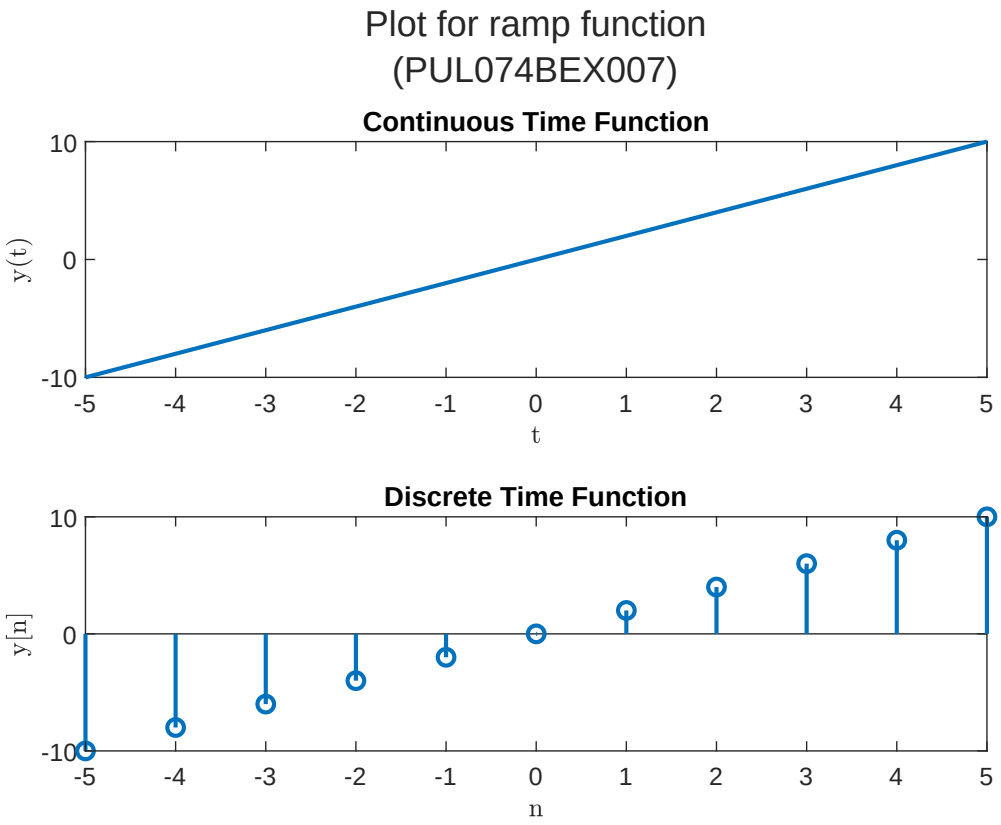

Figure 3: Plot for ramp function

```
>> lab_2_1
Enter function you want to plot: rectangular
```

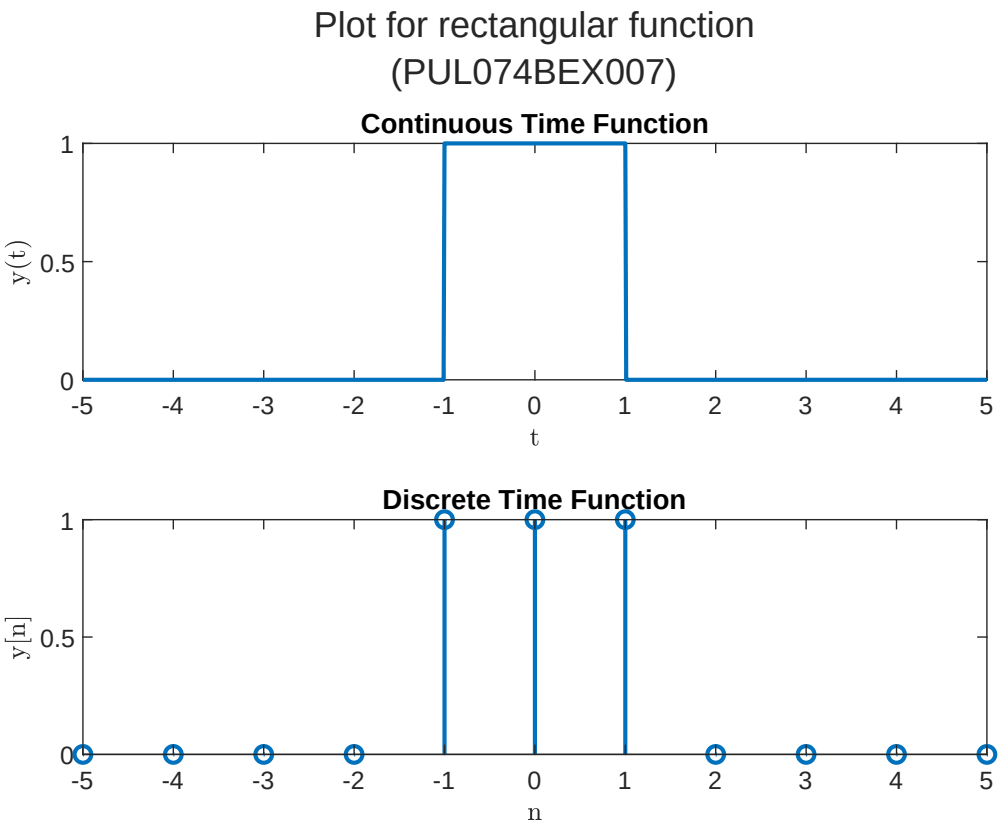

Figure 4: Plot for rectangular function

Problem 2

Plot the following continuous-time signals.

a. $x(t) = Ce^{at}$ **where** C **and** a **are real numbers and choose** C **and** a **both positive and negative.**

```
function y =
    exponential_response_ct(C,a,t)
  y = C .* exp(a.*t);
end
```

Listing 7: Matlab function to return CT exponential response

```
t=-5:0.01:5;
C=1;
a=1;
y1=exponential_response_ct(C,a,t);
C=-1;
a=-1;
y2=exponential_response_ct(C,a,t);
l=tiledlayout(2,1);
title(l,{'Plot for continuous time exponential function', '(PUL074BEX007)'})
nexttile
plot(t,y1,'Linewidth',1.5)
title('For C and a both positive')
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
nexttile
plot(t,y2,'Linewidth',1.5)
title('For C and a both negative')
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
print('lab_2_2_a','-depsc');
```

Listing 8: Matlab script to plot CT exponential function with C and a both real

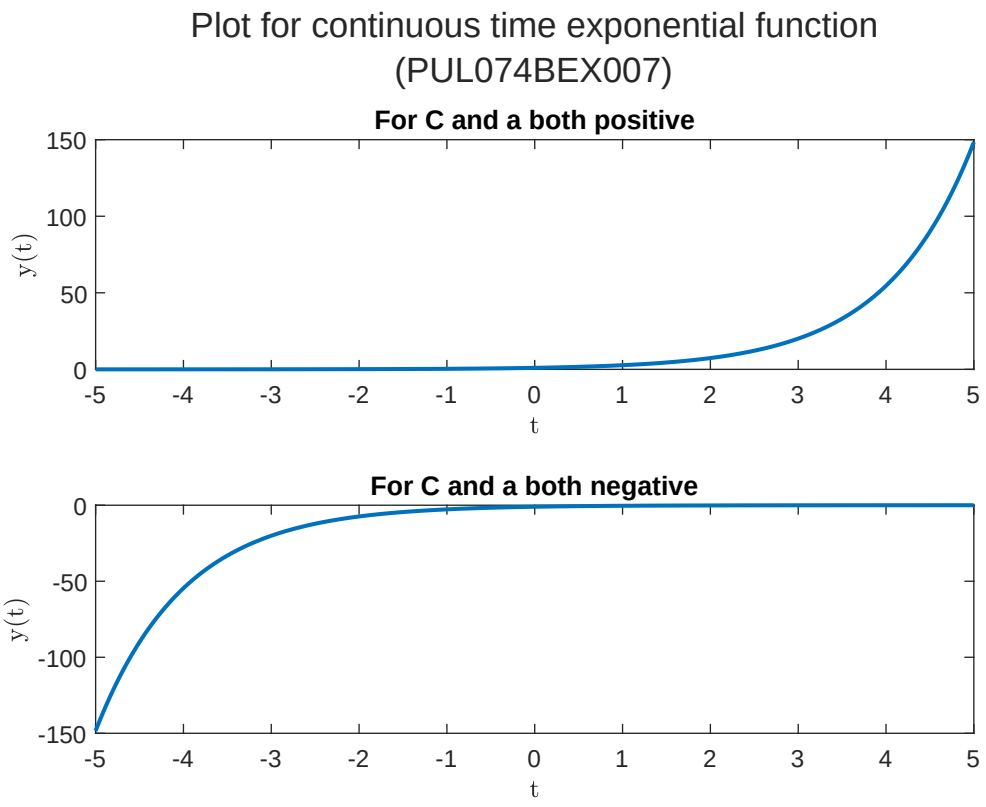

Figure 5: Plot for CT exponential function with C and a both real

b. Plot the same signal taking a as pure imaginary number.

```
t=-5:0.01:5;
C=1;
a=1i;
y=exponential_response_ct(C,a,t);
l=tiledlayout(2,1);
title(l,{'Plot for continuous time
    exponential function', 'with a
    as pure imaginary number','(
   PUL074BEX007)'})
nexttile
plot(t,real(y),'Linewidth',1.5)
title('Real part')
ylabel('Real [x(t)]')
xlabel('t')
nexttile
plot(t,imag(y),'Linewidth',1.5)
title('Imaginary part')
ylabel('Imag [x(t)]')
xlabel('t')
print('lab_2_2_b','-depsc')
```

Listing 9: Matlab script to plot CT exponential function with a purely imaginary

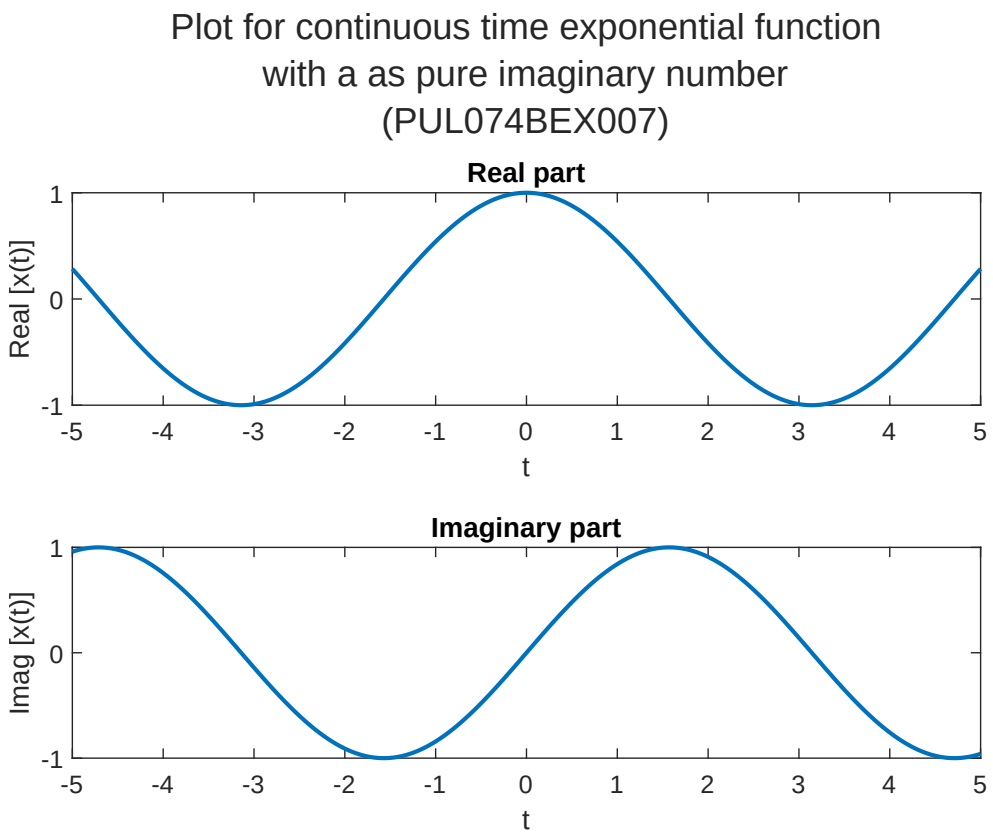

Figure 6: Plot for CT exponential function with C, a both positive and negative

c. Consider complex exponential signal as specified in b) where C is expressed in polar form i.e., $C = |C|e^{j\theta}$ and a in rectangular form i.e., $a = r + j\omega_o$. Then function $x(t)$, on simplification, becomes

$$x(t) = |C|e^{rt}[\cos(\omega_o t + \theta) + j\sin(\omega_o t + \theta)]$$

Now, plot the signal for different values of r and comment on the results.

i. r=0 ii. r<0 iii. r>0

```
function y =
    exponential_response_polar(C,r,
    w,theta,t)
    y = abs(C).* exp(r .* t).*(cos
    (w.*t+theta)+1i.*sin(w.*t+theta
    ));
end
```

Listing 10: Matlab function to return CT exponential (polar form) response

```
r=input('Enter the value for r: ')
    ;
t=-5:0.01:5;
C=1;
w=pi;
theta=pi;
y=exponential_response_polar(C,r,w
    ,theta,t);
l=tiledlayout(2,1);
title(l,{sprintf('Plot for CT
    exponential function with r=%d'
    ,r),'(PUL074BEX007)'})
nexttile
plot(t,real(y),'Linewidth',1.5)
title('Real part')
ylabel('Real [x(t)]')
xlabel('t','interpreter','latex')
nexttile
plot(t,imag(y),'Linewidth',1.5)
title('Imaginary part')
ylabel('Imag [x(t)]')
xlabel('t','interpreter','latex')
print(sprintf('lab_2_2_c_%d',r),'-
    depsc')
```

Listing 11: Matlab script to plot the function for values of r based on user input

```
>> lab_2_2_c

Enter the value for r: 0
```

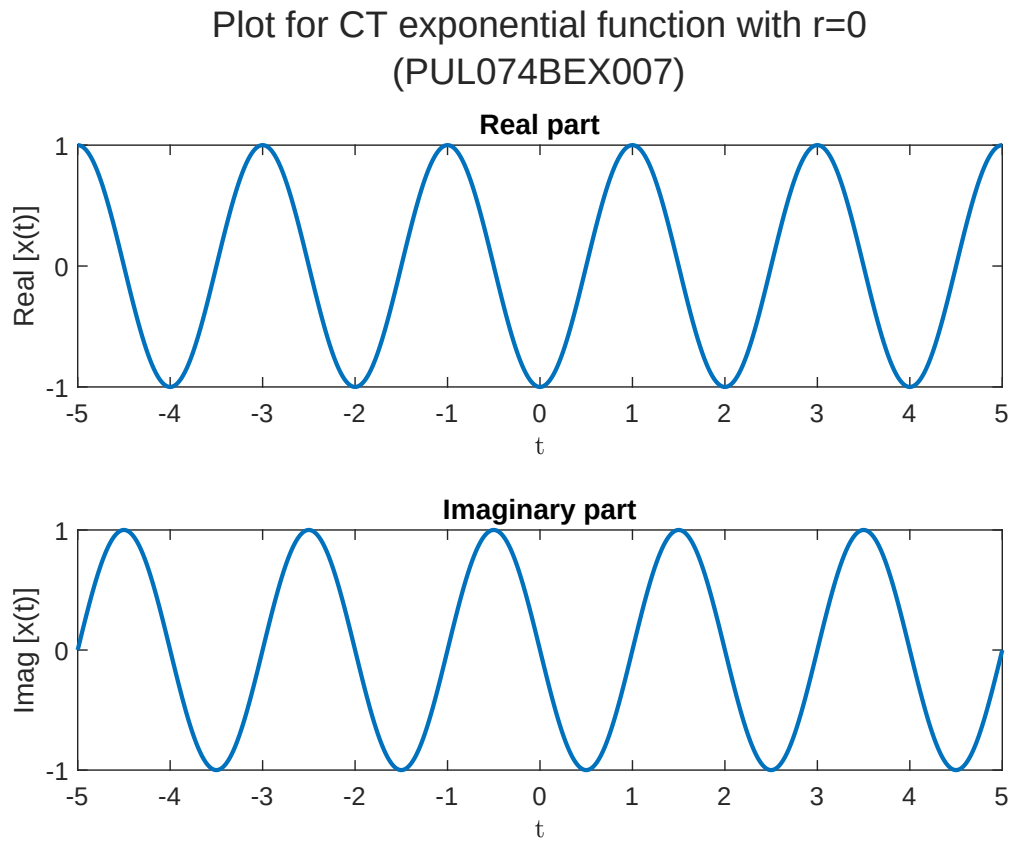

Figure 7: Plot for CT exponential function with $r = 0$

```
>> lab_2_2_c

Enter the value for r: -1
```

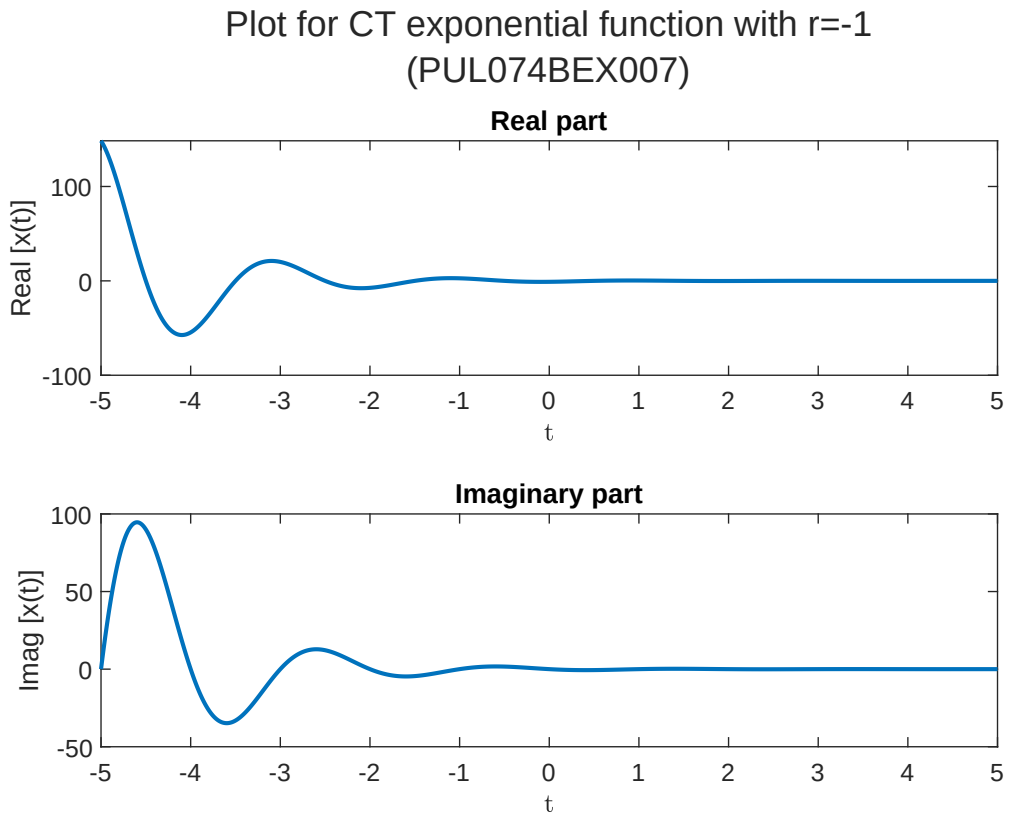

Figure 8: Plot for CT exponential function with $r = -1$

```
>> lab_2_2_c
Enter the value for r: 1
```

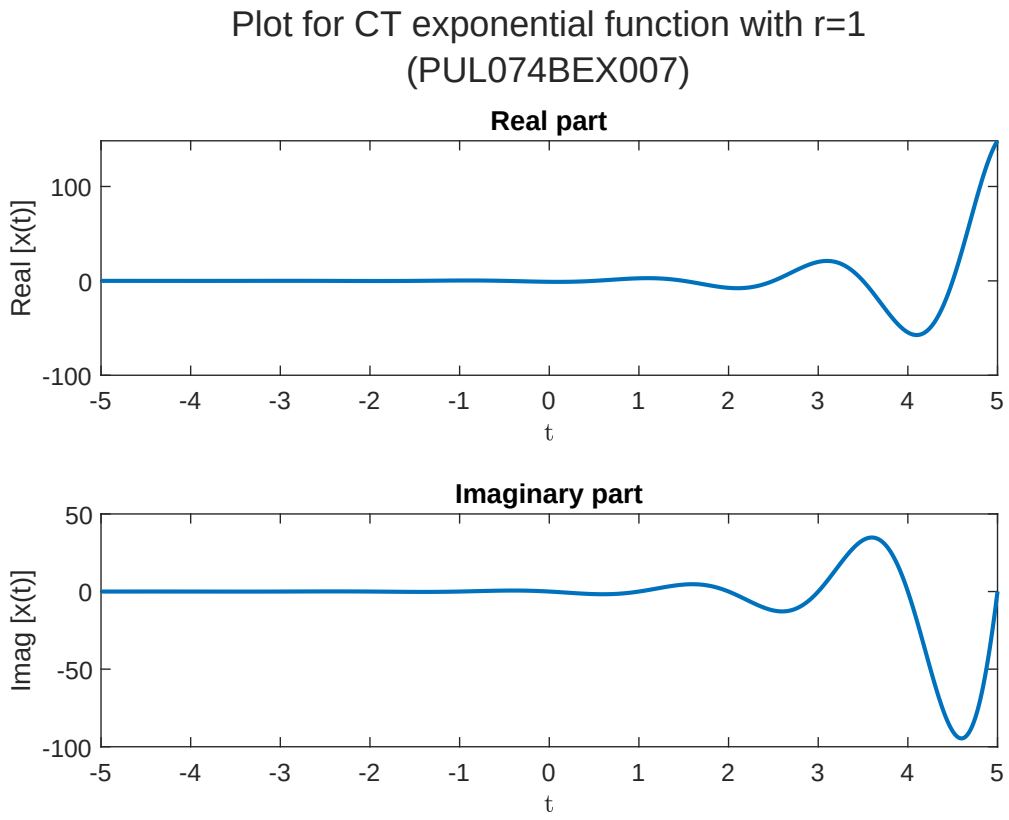

Figure 9: Plot for CT exponential function with $r = -1$

Problem 3

Plot the DT exponential function $x[n] = a^n$, $a = |a|e^{j\theta}$. Choose the suitable value of a and θ.

```
function y =
    exponential_response_dt(a,theta
    ,n)
  y = (abs(a).*exp(1i.*theta)).^n;
end
```

Listing 12: Matlab function to return DT exponential response

```
n=-5:5;
a=2;
theta=pi;
y=exponential_response_dt(a,theta,n);
l=tiledlayout(2,1);
title(l,{'Plot for DT exponential function','(PUL074BEX007)'})
nexttile
stem(n,abs(y),'Linewidth',1.5)
title('Magnitude')
ylabel('Magnitude of x[n]')
xlabel('n','interpreter','latex')
nexttile
stem(n,angle(y)*180/pi,'Linewidth',1.5)
title('Phase')
yticks([-180 0 180])
ylim([-180 180])
ylabel('Phase of x[n] in degree')
xlabel('n','interpreter','latex')
print('lab_2_3','-depsc')
```

Listing 13: Matlab script to plot DT exponential function

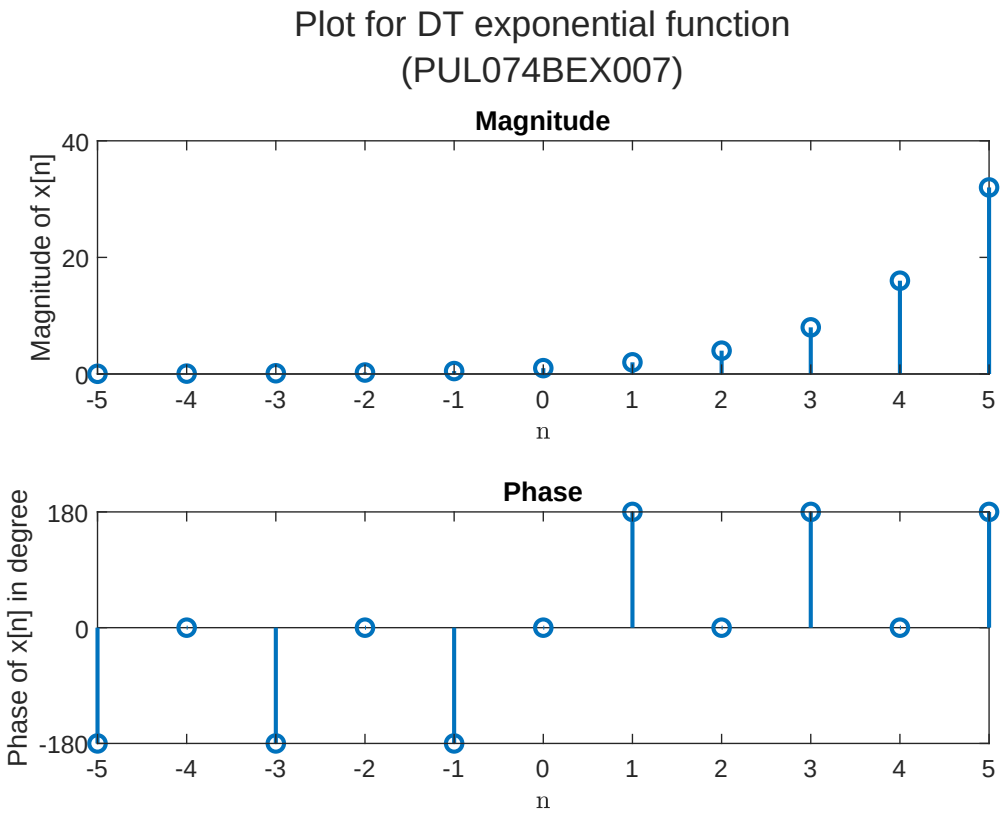

Figure 10: Plot for DT exponential function

Problem 4

Synthesize the signal from the FS coefficients as $C_0 = 1$, $C_1 = C_{-1} = \frac{1}{4}$, $C_2 = C_{-2} = \frac{1}{2}$, $C_3 = C_{-3} = \frac{1}{3}$.

```
function y = synthesize_signal(C,t)
    len = length(C);
    w = -floor(len/2):floor(len/2);
    y = sum(C' .* exp(1i.* w' .* t));
end
```

Listing 14: Matlab function to return synthesized signal from the FS coefficients

```
t = -5:0.01:5;
C=[1/3 1/2 1/4 1 1/4 1/2 1/3];
y=synthesize_signal(C,t);
l=tiledlayout(1,1);
title(l,{'Signal synthesis from the FS coefficients','(PUL074BEX007)'})
nexttile
plot(t,real(y),'Linewidth',1.5)
xlabel('t','interpreter','latex')
ylabel('Amplitude');
print('lab_2_4','-depsc')
```

Listing 15: Matlab script to plot the synthesized signal

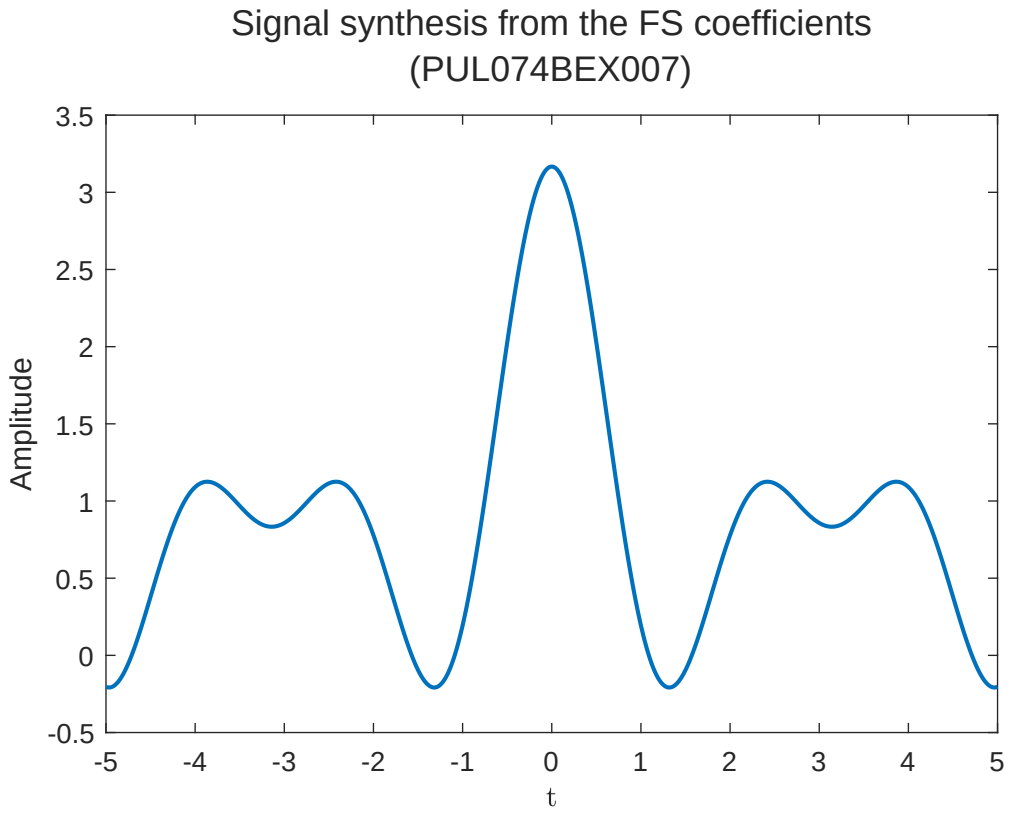

Figure 11: Plot for synthesized signal from the FS coefficients

Problem 5

Plot fundamental sinusoidal signal, its higher harmonics up to 5th harmonics and add all of them to see the result. Comment on the result.

```
function [fn, y] = harmonic_sum(w,
    n,t)
  no = 1:n;
  fn = sin(no.'.*w.*t);
  y = sum(fn,1);
end
```

Listing 16: Matlab function to return sinusoidal harmonics and their sum

```
figure()
t=-5:0.01:5;
w=pi/6;
n=5;
[fn, y] = harmonic_sum(w,n,t);
l=tiledlayout(2,1);
```

```
title(1,{'Plot for sinusoidal harmonics and their sum','(PUL074BEX007)'})
nexttile
p1=plot(t,fn,'Linewidth',1.5);
title('Sinusoidal Harmonics')
ylabel('Amplitude')
xlabel('t','interpreter','latex')
ax2=nexttile;
p2=plot(t,y,'Linewidth',1.5,'color','blue');
title('Sum of Harmonics')
ylabel('Amplitdue')
xlabel('t','interpreter','latex')
labels = {'1st Harmonic','2nd Harmonic','3rd Harmonic','4th Harmonic','5th Harmonic','Sum of Harmonics'};
legend([p1;p2],labels,'Location','northoutside','Orientation', 'horizontal','NumColumns',3)
print('lab_2_5','-depsc')
```

Listing 17: Matlab script to plot sinusoidal harmonics and their sum

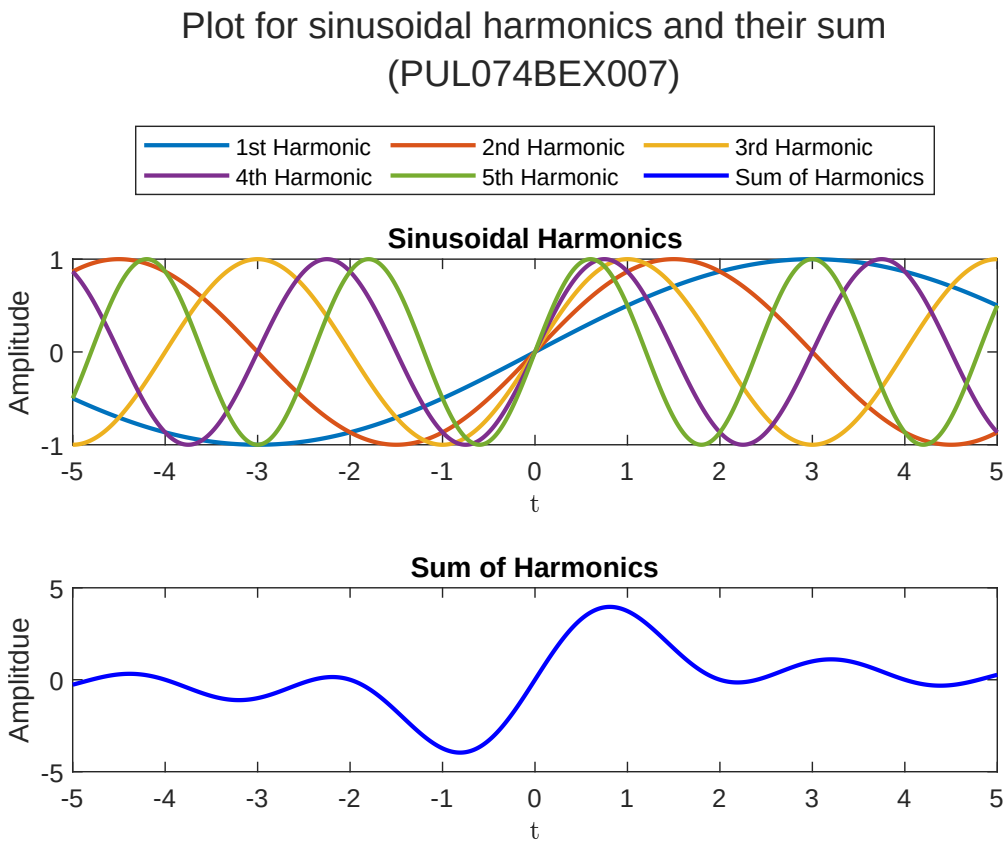

Figure 12: Plot for sinusoidal harmonics and their sum