INSTITUTE OF ENGINEERING
CENTRAL CAMPUS,PULCHOWK

DIGITAL SIGNAL PROCESSING

LAB #2

# Familiarization with basic CT/DT functions

**Submitted BY:**
AMRIT PRASAD PHUYAL
IOE Roll: PULL074BEX004

**Submitted To:**
Anila Kansakar
Department of Electronics and
Computer Engineering

January 15, 2022

# Table of Contents

## List of MATLAB codes



## List of Figures

# 1 Title

Familiarization with basic CT/DT functions

# 2 Objective

Familiarization with basic CT/DT functions

# 3 Theory

```
who          >> List variables in workspace
whos         >> List variables in workspace, with sizes and types
input()      >> Read input from user
disp()       >> Display value of variable
subplot()    >> Plot multiple graphs in one figure
figure()     >> Create new figure window
clear all    >> Clear all variables from workspace, freeing up system memory
close all    >> Close all figures
home         >> Send cursor to home position
hold on      >> Retain plot data for multiple plots in one figure
grid on      >> Turn on grid lines
grid off     >> Turn off grid lines
grid         >> Turn on and off grid lines and set grid spacing
demo         >> Access product examples in Help browser
ver          >> Display version of Matlab and operating system information
lookfor      >> Search for keyword in all help files
length()     >> Length of largest dimension of array
pause        >> Stop MATLAB execution temporarily
plot()       >> 2-D line  plotting
stem()       >> Plot discrete data as stems
real()       >> Return real part of complex number
imag()       >> Return imaginary part of complex number
zeros()      >> Create array of all zeros
ones()       >> Create array of  all ones
exp()        >> Return exponential of complex number
for          >> Loop to repeat specified numbers of times
end          >> End  For loop
if-else      >> If-else statement execute if statement is true
```

# 4 Lab Problems

## 4.1 Problem 1

### Plot the basic signal using Matlab

#### 4.1.1 Impulse response

Codes:

```
n=-5:5;
t=-5:0.01:5;
y=impulse_response(t);
z=impulse_response(n);
l=tiledlayout(2,1);
```

```
title(l,{sprintf('%s',['Plot for Impulse function'])})
nexttile
plot(t,y,'LineWidth',1.5)
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
title('Continuous Time Function')
nexttile
stem(n,z,'Linewidth', 1.5)
xlabel('n','interpreter','latex')
ylabel('y[n]','interpreter','latex')
title('Discrete Time Function')


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

Code 1: Matlab code for plotting Impulse function

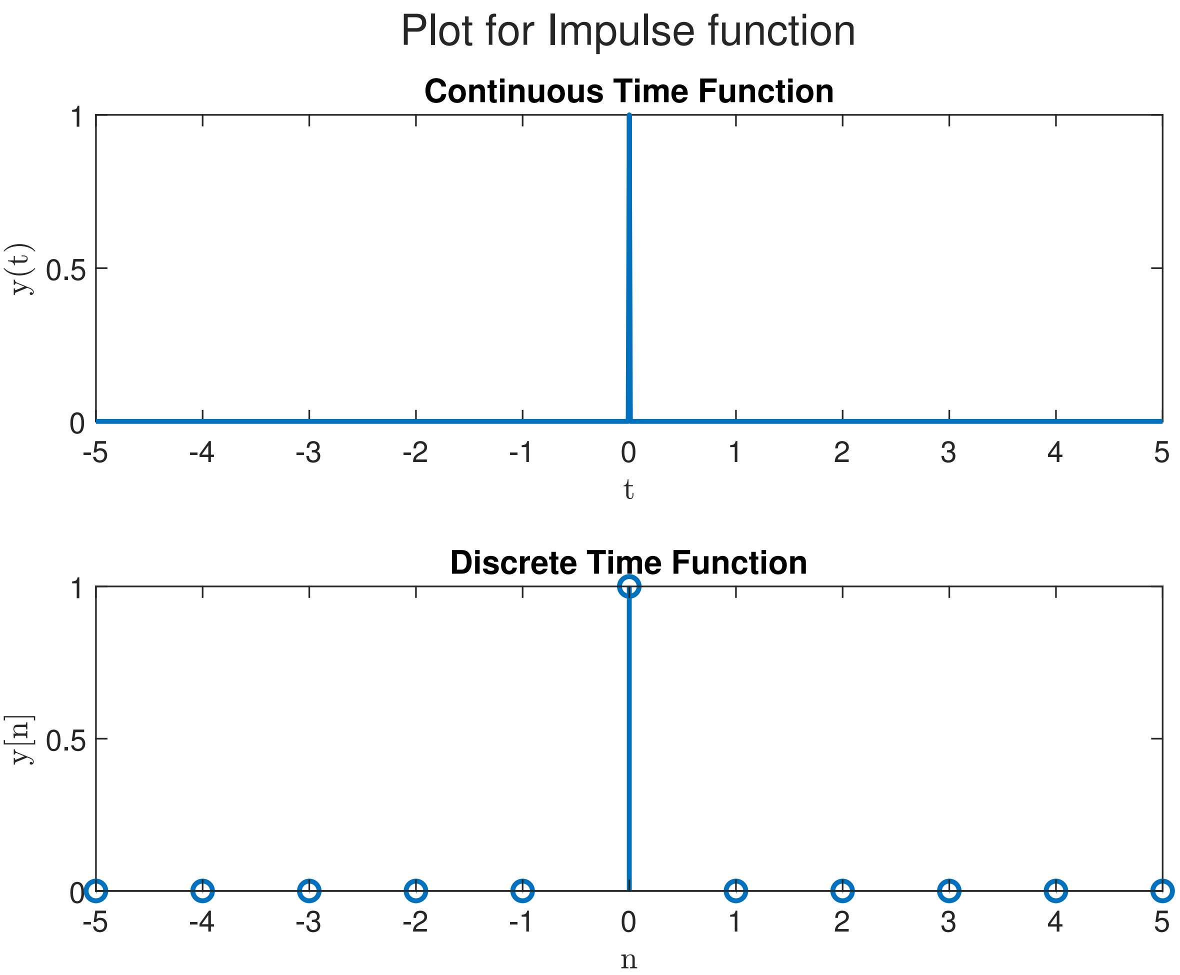


Figure 1: Plot for Impulse function

#### 4.1.2 Unit step response

```
n=-5:5;
t=-5:0.01:5;
```

```
y=unit_response(t);
z=unit_response(n);
l=tiledlayout(2,1);
title(l,{sprintf('%s',['Plot for Unit Step function'])})
nexttile
plot(t,y,'LineWidth',1.5)
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
title('Continuous Time Function')
nexttile
stem(n,z,'Linewidth', 1.5)
xlabel('n','interpreter','latex')
ylabel('y[n]','interpreter','latex')
title('Discrete Time Function')

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

Code 2: Matlab code for plotting Unit Step function

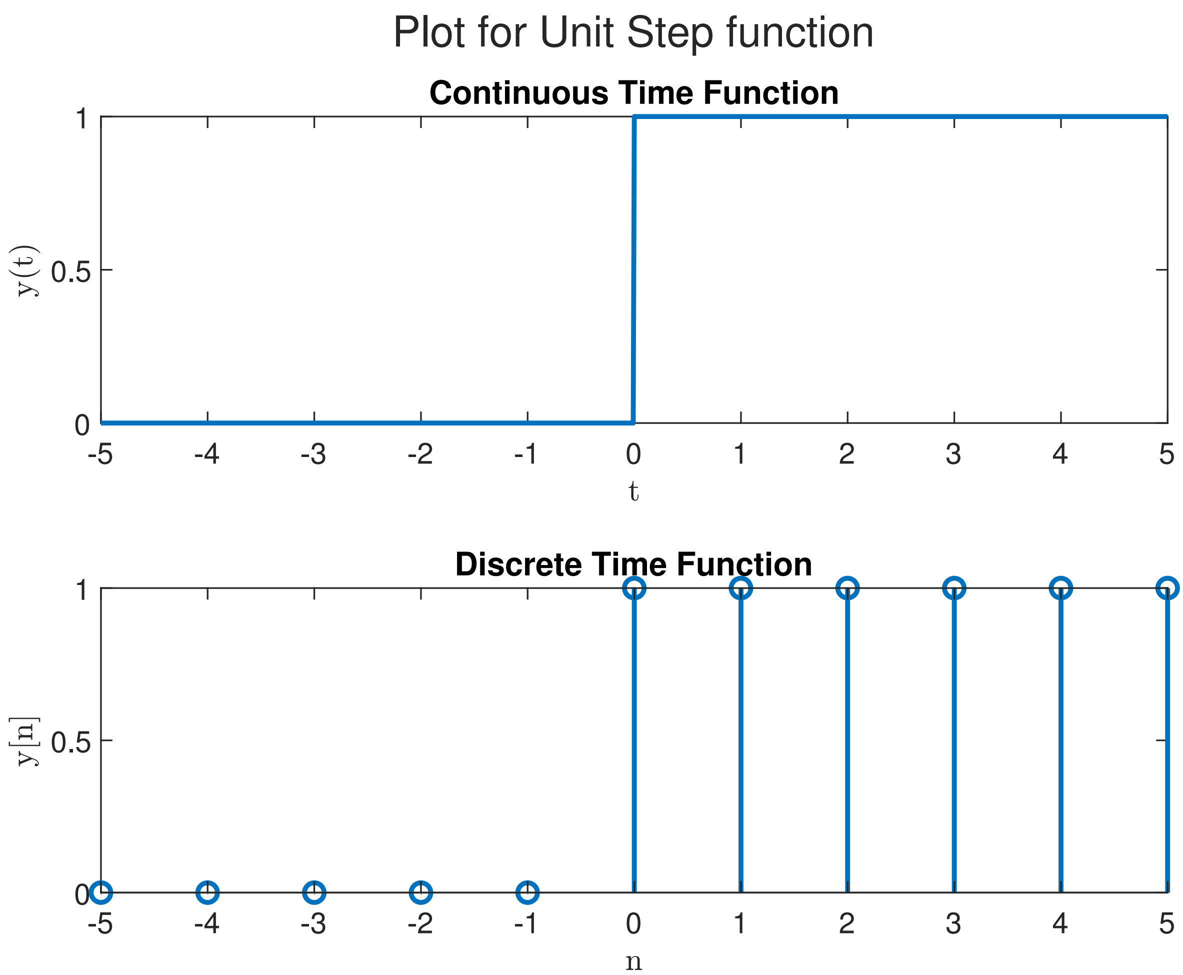


Figure 2: Plot for Unit Step function

### 4.1.3 Ramp response

```matlab

n=-5:5;
t=-5:0.01:5;
y=ramp_response(t);
z=ramp_response(n);
l=tiledlayout(2,1);
title(l,{sprintf('%s',['Plot for Ramp function'])})
nexttile
plot(t,y,'LineWidth',1.5)
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
title('Continuous Time Function')
nexttile
stem(n,z,'Linewidth', 1.5)
xlabel('n','interpreter','latex')
ylabel('y[n]','interpreter','latex')
title('Discrete Time Function')

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

Code 3: Matlab code for plotting Ramp function

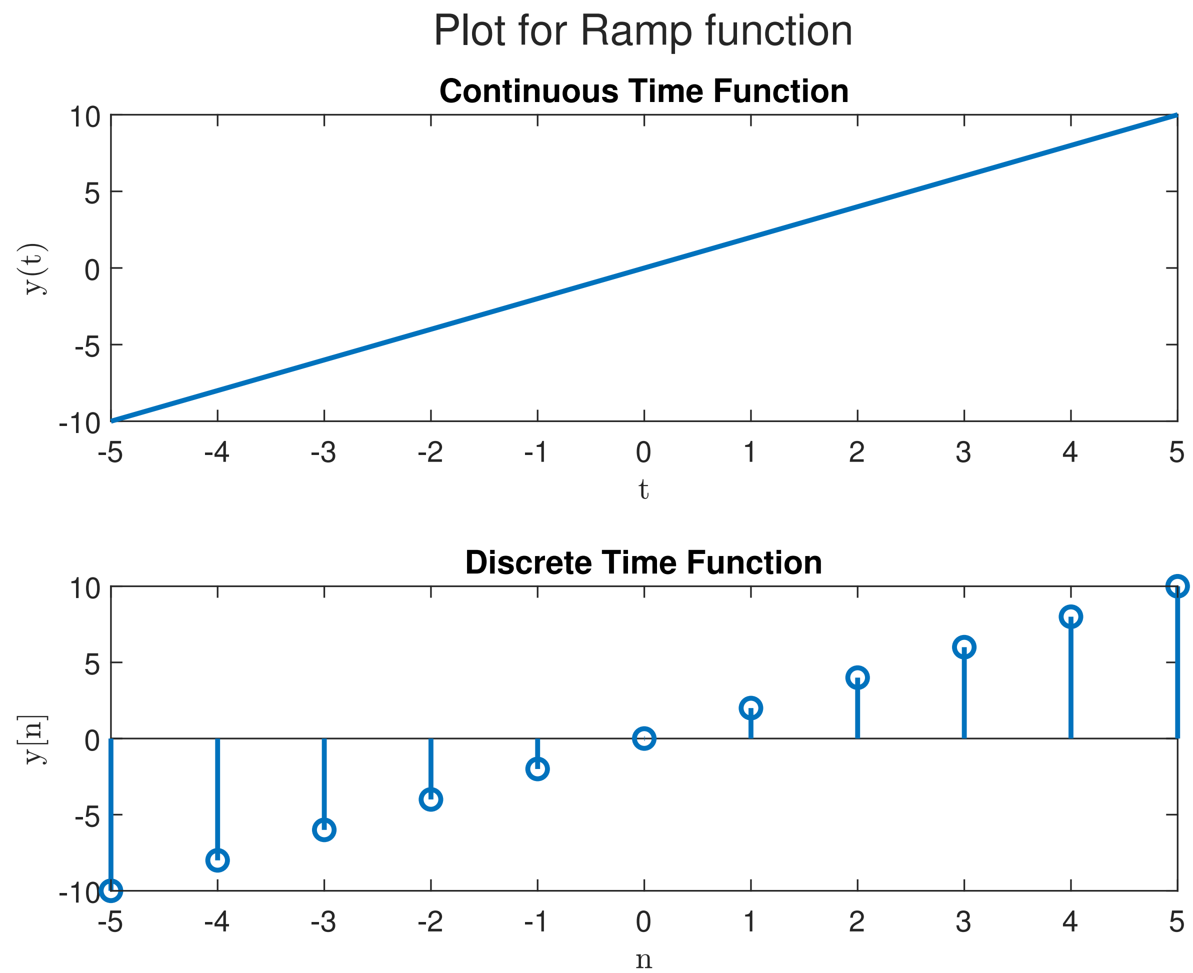


Figure 3: Plot for Ramp function

### 4.1.4 Rectangular pulse response

```
n=-5:5;
t=-5:0.01:5;
y=rectangular_response(t/2);
z=rectangular_response(n/2);
l=tiledlayout(2,1);
title(l,{sprintf('%s',['Plot for Rectangular function'])})
nexttile
plot(t,y,'LineWidth',1.5)
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
title('Continuous Time Function')
nexttile
stem(n,z,'Linewidth', 1.5)
xlabel('n','interpreter','latex')
ylabel('y[n]','interpreter','latex')
title('Discrete Time Function')

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

Code 4: Matlab code for plotting Rectangular pulse function

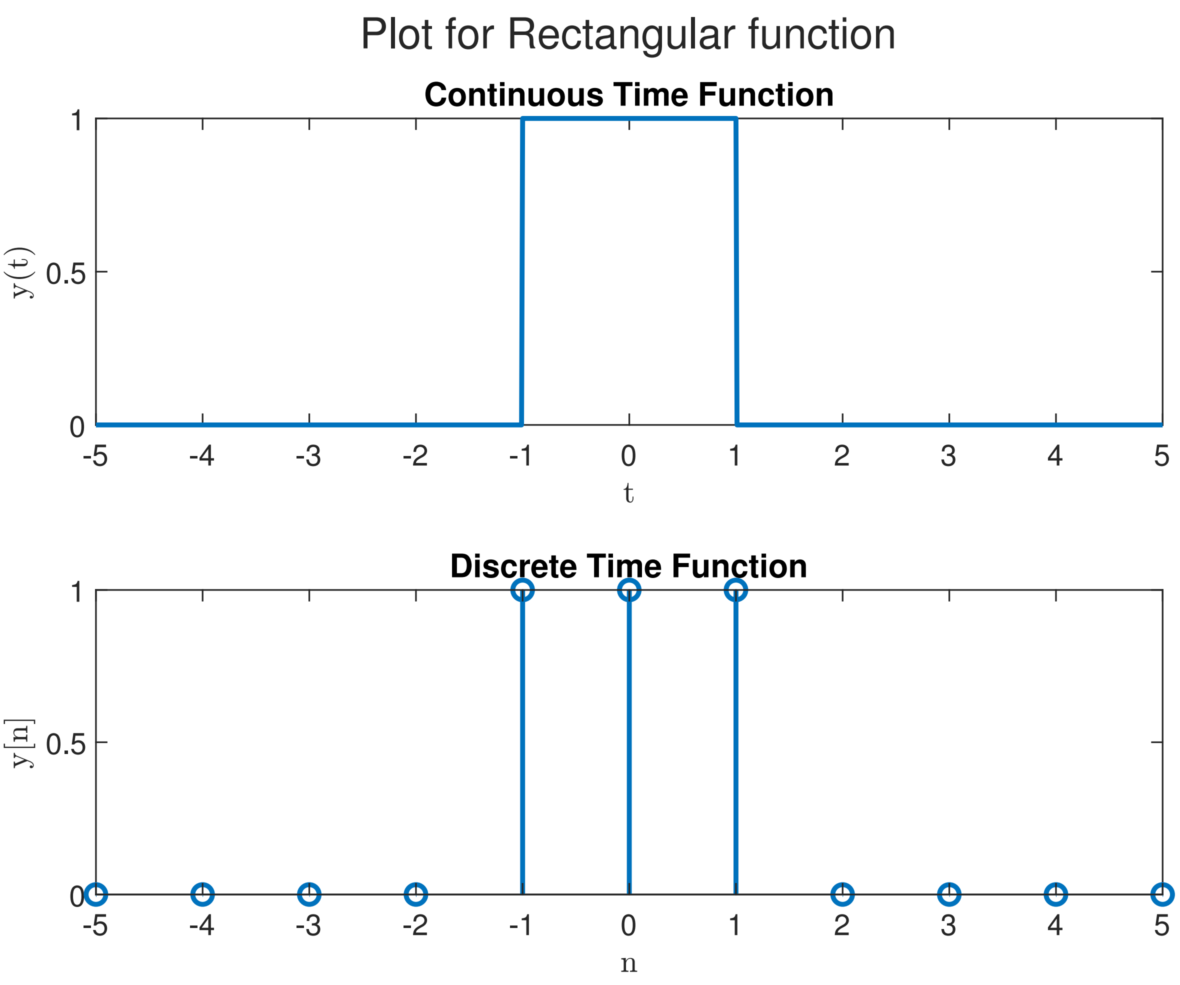


Figure 4: Plot for Rectangular pulse function

### 4.2 Problem 2

#### Plot the following continuous-time signals

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

```
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);
C=1;
a=-1;
y3=exponential_response_ct(C,a,t);
C=1;
a=1;
y4=exponential_response_ct(C,a,t);
l=tiledlayout(2,2);
title(l,{'Plot for continuous time exponential function'})
nexttile
plot(t,y1,'Linewidth',1.5)
title('For C and a both -ve')
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
nexttile
plot(t,y2,'Linewidth',1.5)
title('For C -ve and a +ve')
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
nexttile
plot(t,y3,'Linewidth',1.5)
title('For C +ve and a -ve')
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')
nexttile
plot(t,y4,'Linewidth',1.5)
title('For C and a both +ve')
xlabel('t','interpreter','latex')
ylabel('y(t)','interpreter','latex')


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

Code 5: Matlab code for C and a Both real

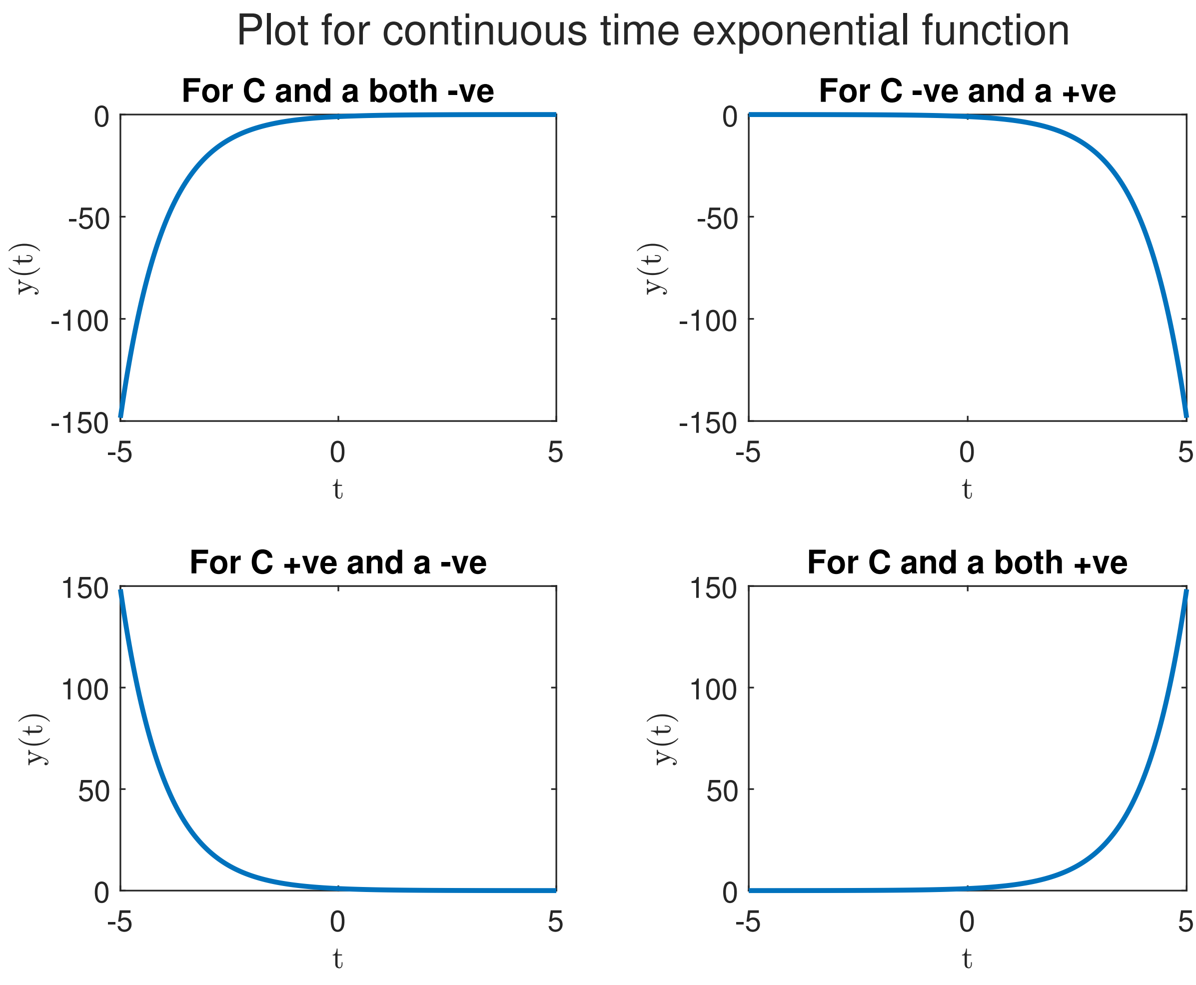


Figure 5: Plot for for C and a Both real

**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'})
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')

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

Code 6: Matlab code for C real a Imaginary

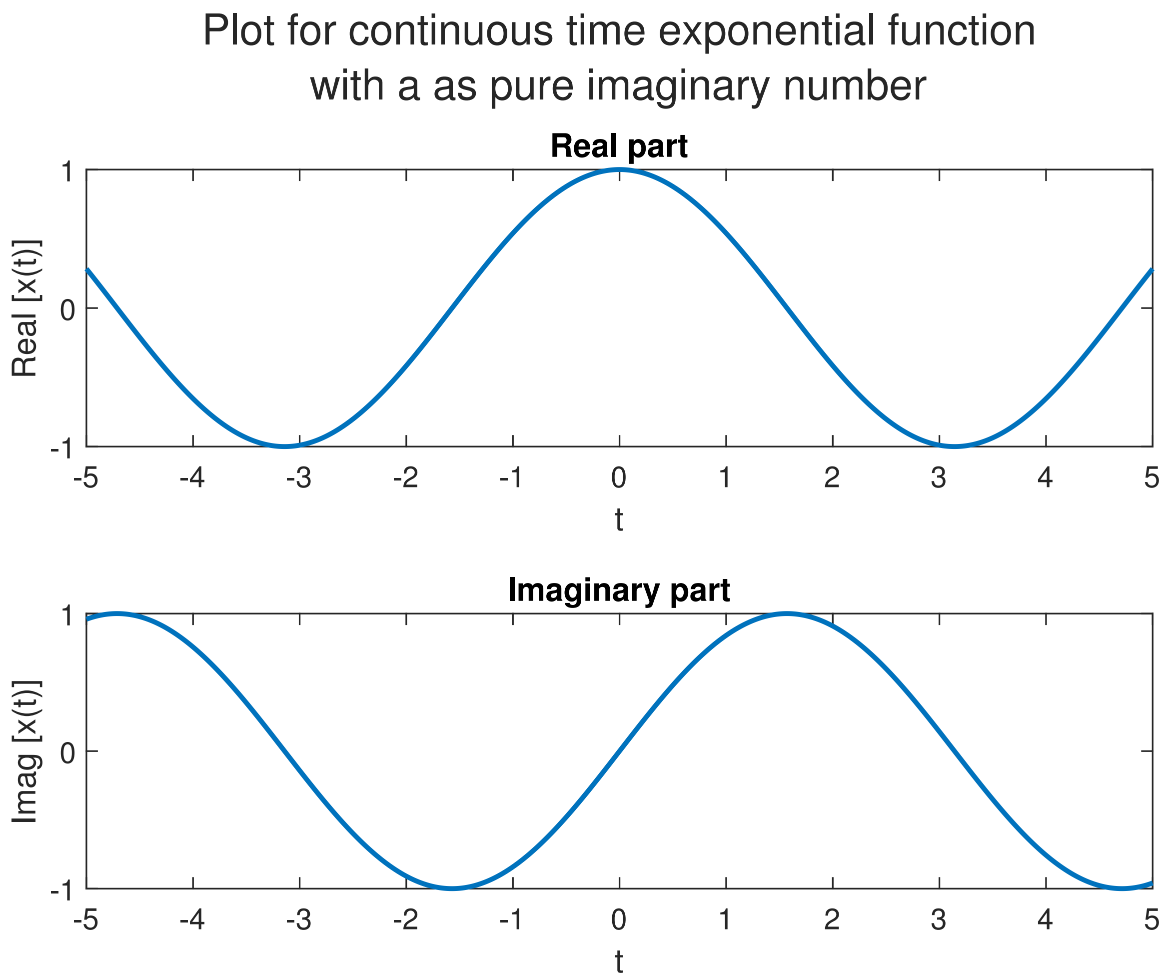


Figure 6: Plot for C real a Imaginary

**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**

```
r=input('Enter acceptable value any 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)})
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')

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
```

Code 7: Matlab code for different value of r

### 4.2.1 For r=0

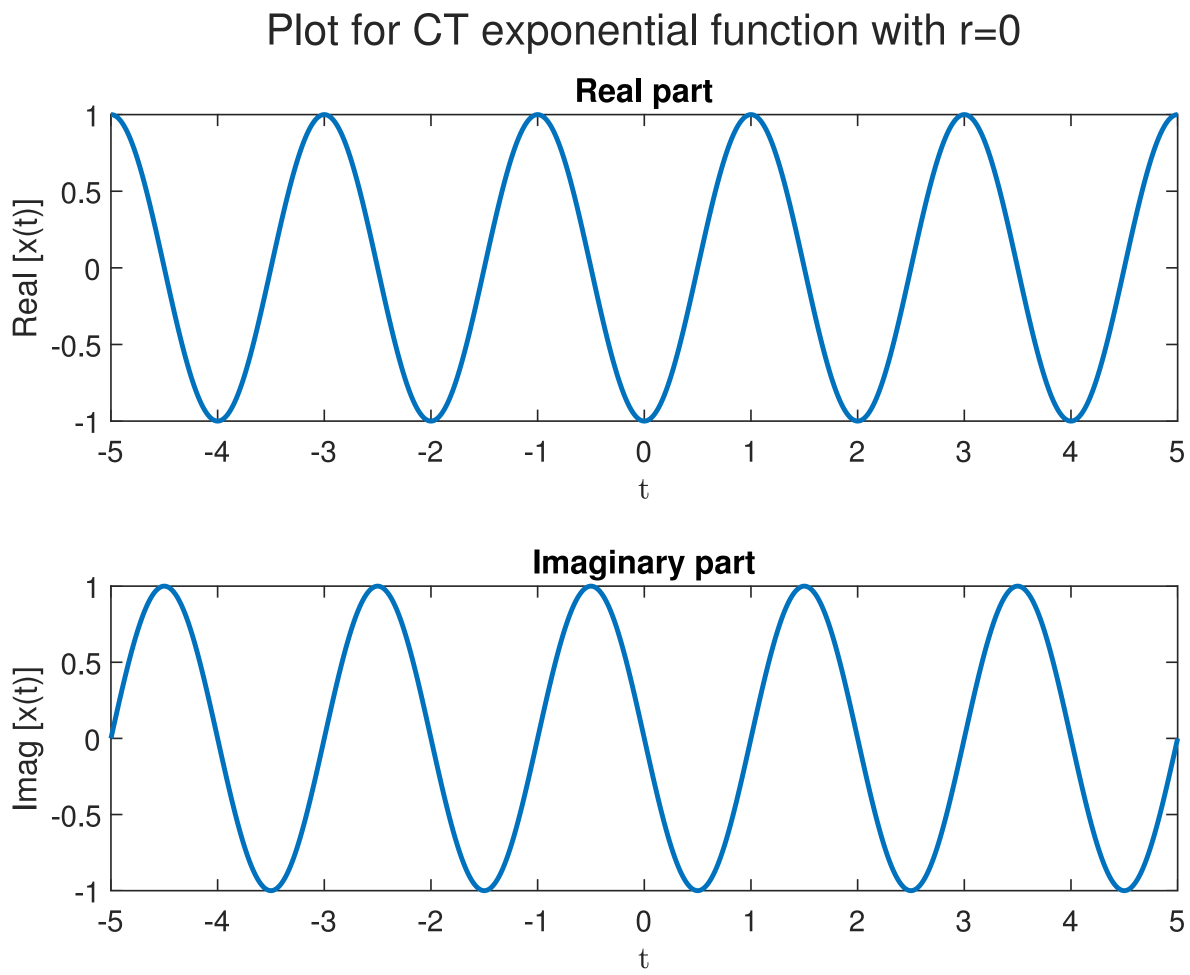


Figure 7: Plot for r=0

#### 4.2.2 For r<0

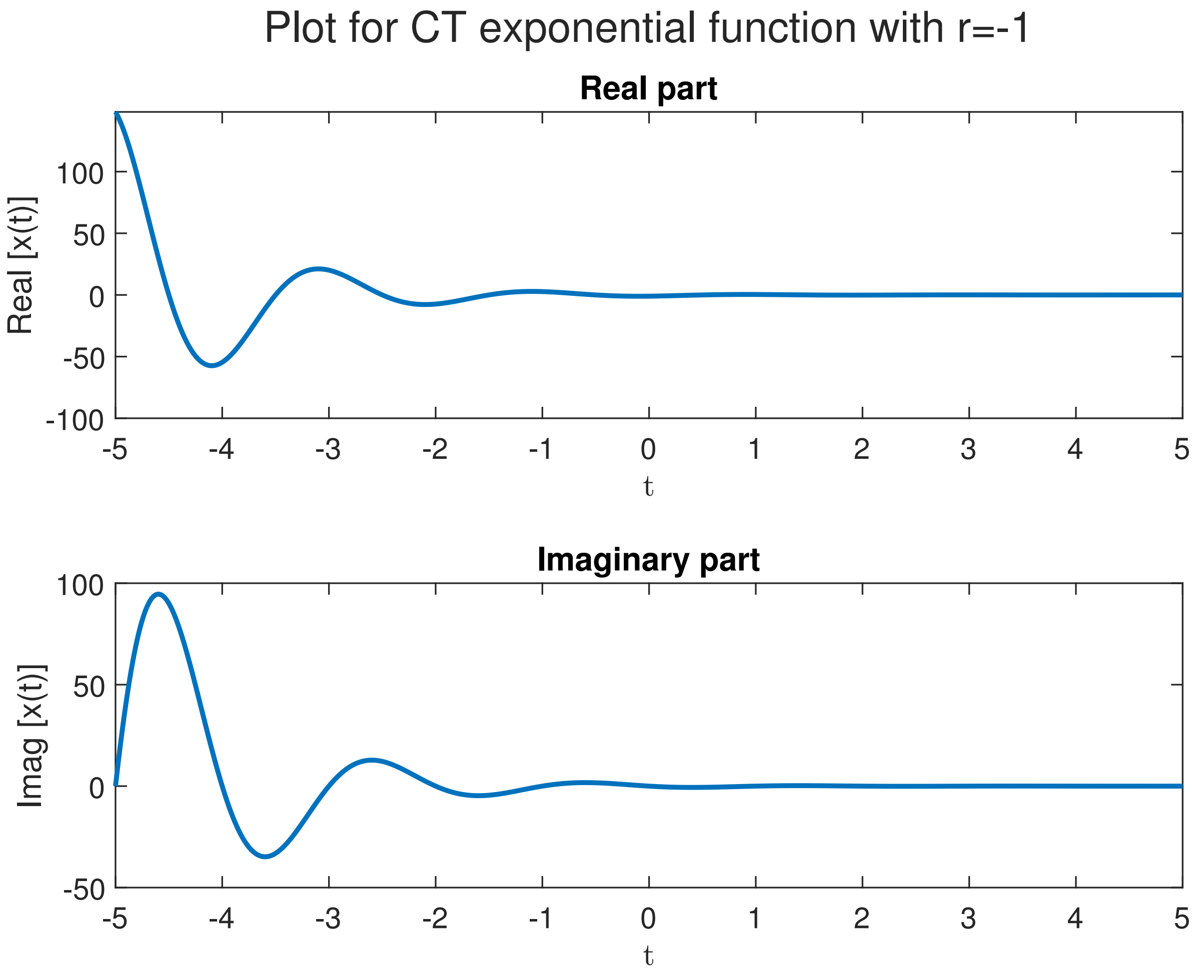


Figure 8: Plot for r<0

#### 4.2.3 For r>0

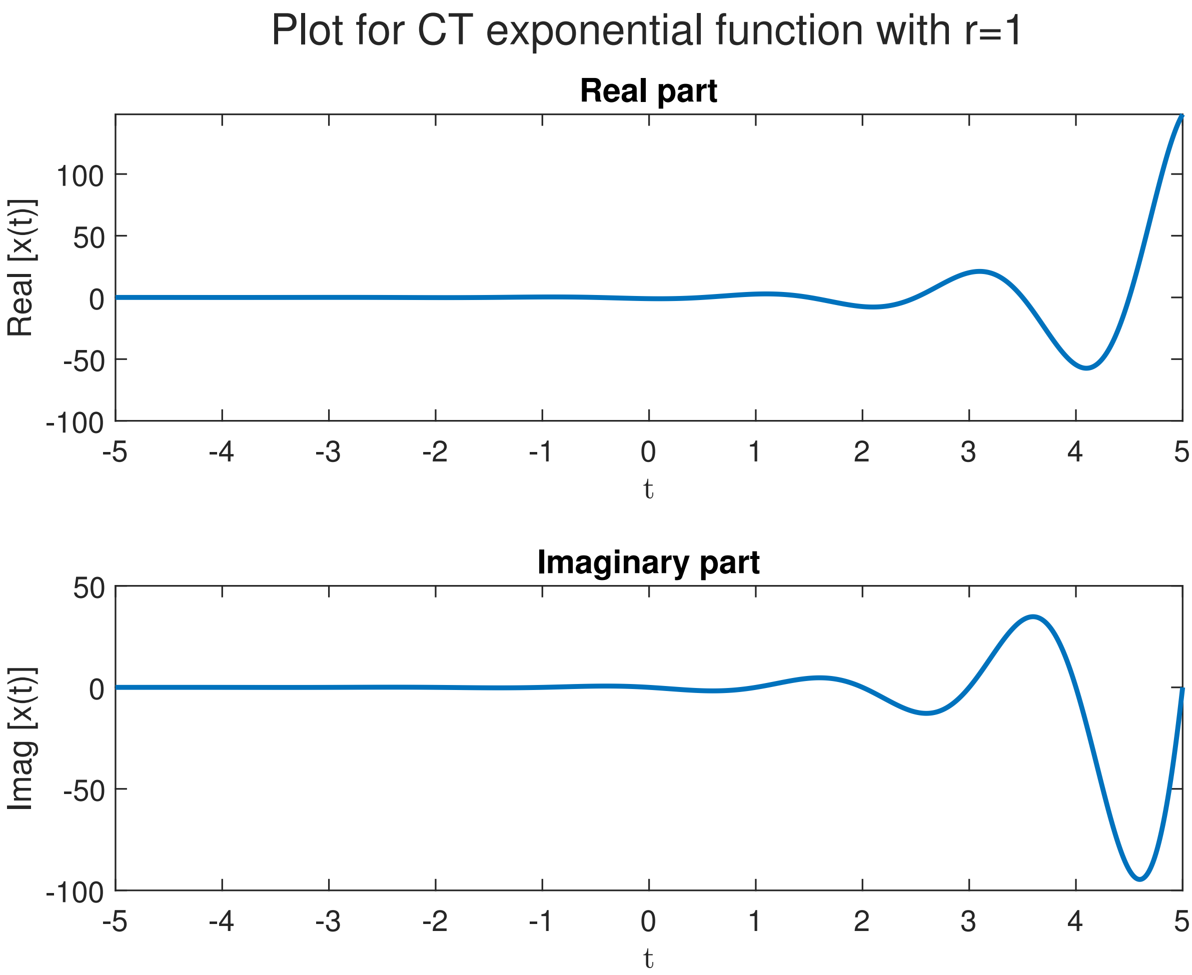


Figure 9: Plot for r>0

### 4.3 Problem 3

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

```
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'})
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')

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

Code 8: Matlab code for calculation and plot DT exponential function

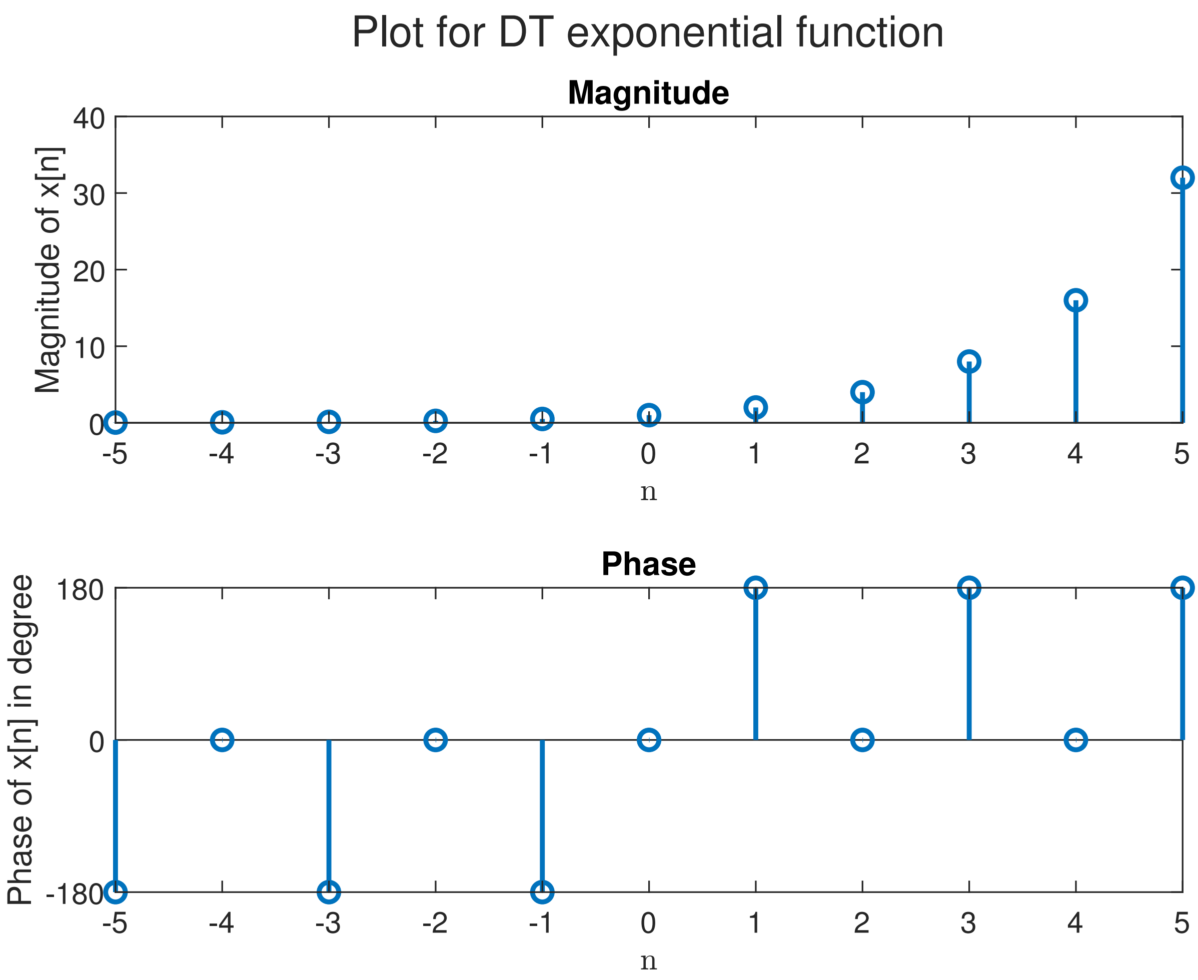


Figure 10: Plot for DT exponential function

### 4.4 Problem 4

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

```
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'})
nexttile
plot(t,real(y),'Linewidth',1.5)
xlabel('t','interpreter','latex')
ylabel('Amplitude');
```

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

Code 9: Matlab code for synthesizing and plotting signal

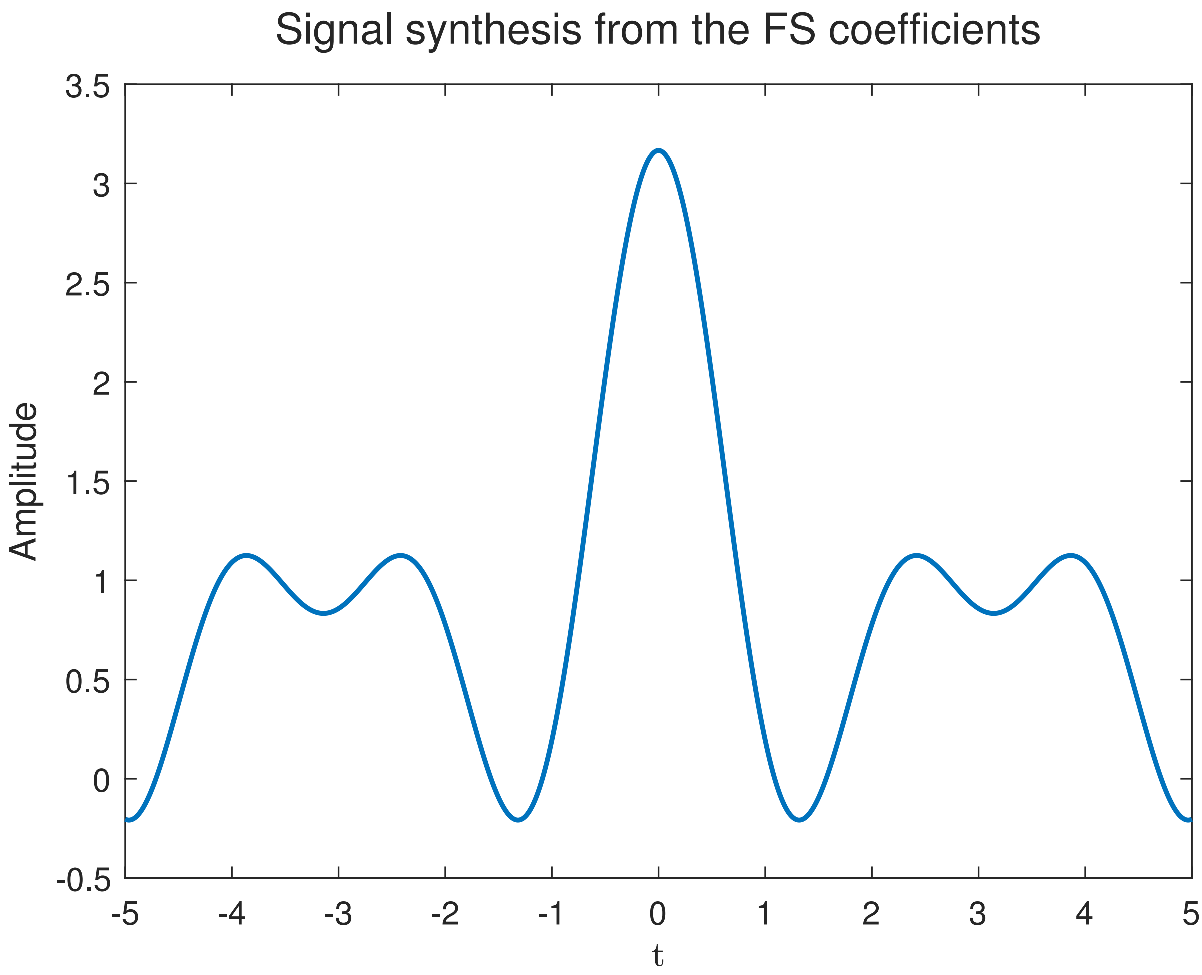


Figure 11: Plot for synthesized signal

### 4.5 Problem 5

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

```
figure()
t=-5:0.01:5;
w=pi/6;
n=5;
[fn, y] = harmonic_sum(w,n,t);
l=tiledlayout(2,1);
title(l,{'Plot for sinusoidal harmonics and their sum'})
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)

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

Code 10: Matlab code for calculation and plot of Sinusoidal harmonics and thier sum

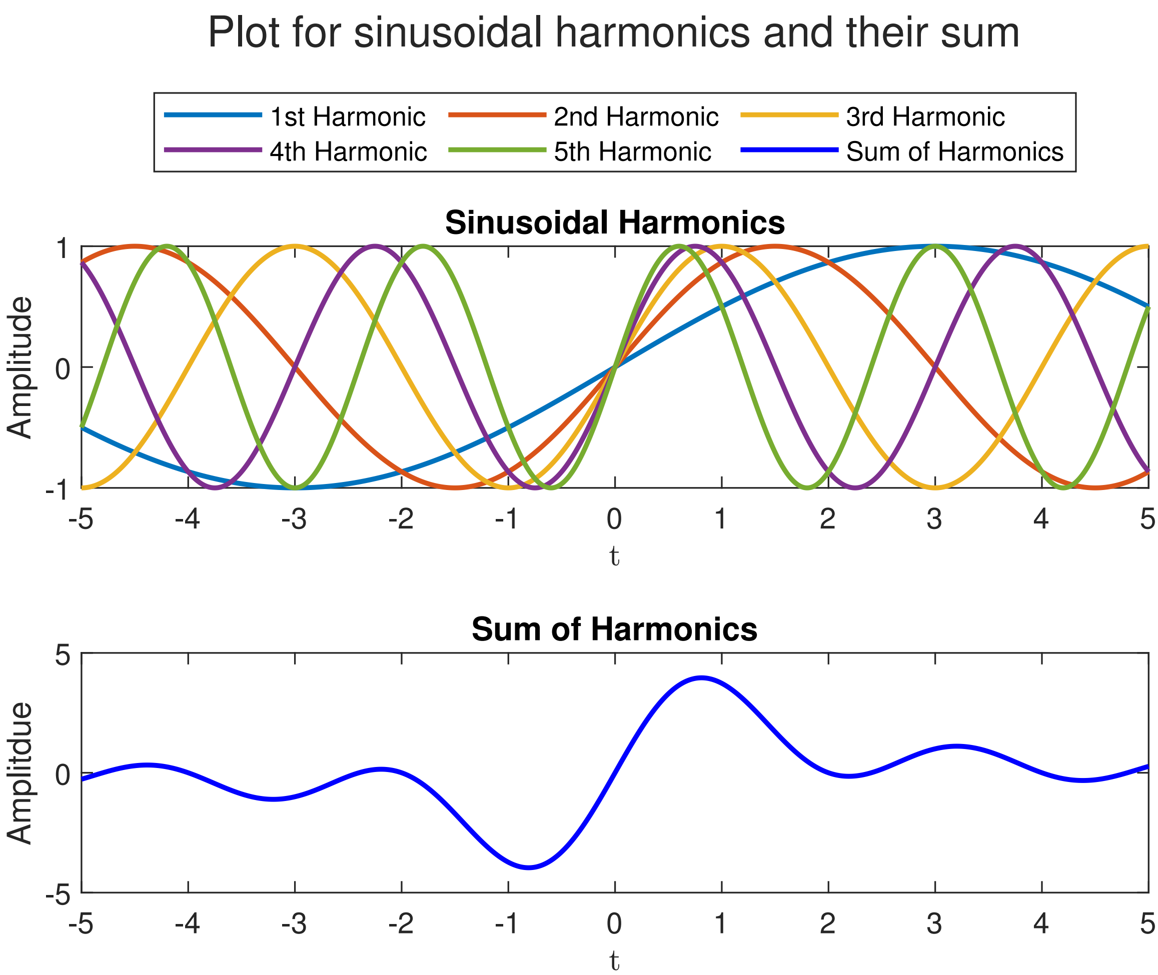


Figure 12: Plot for Sinusoidal harmonics and thier sum

# 5 Discussion and Conclusion

In this Lab we familiarize ourself with Matlab Programming with basic of Continous time and Discrete time signals. We also learn about their basic operations in Matlab including plotting and analyzing the signals. We learn to use online help for different commands and use them to calculate different DT and CT functions and plot them.