191IT234_Niraj_Nandish

September 9, 2020

1 Lab Week 4

1.0.1 Name: Niraj Nandish

1.0.2 Roll no.: 191IT234

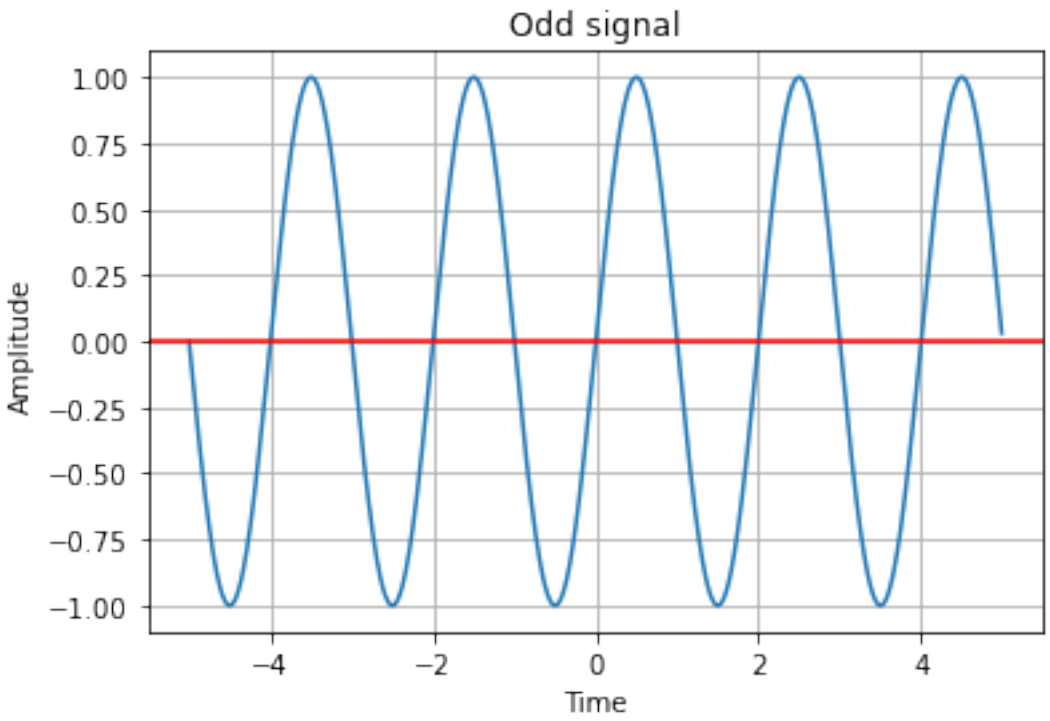
1.0.3 Semester: 3

[1]:
```
# Import all required libraries
import numpy as np                                    # Contains built-in␣
 ↪functions to work on arrays
import matplotlib.pyplot as plt                       # Used to plot graphs
from scipy import signal                              # Generate signals of␣
 ↪various types
from scipy.fft import fft                             # Generate Fourier␣
 ↪Transorm of given signal
```

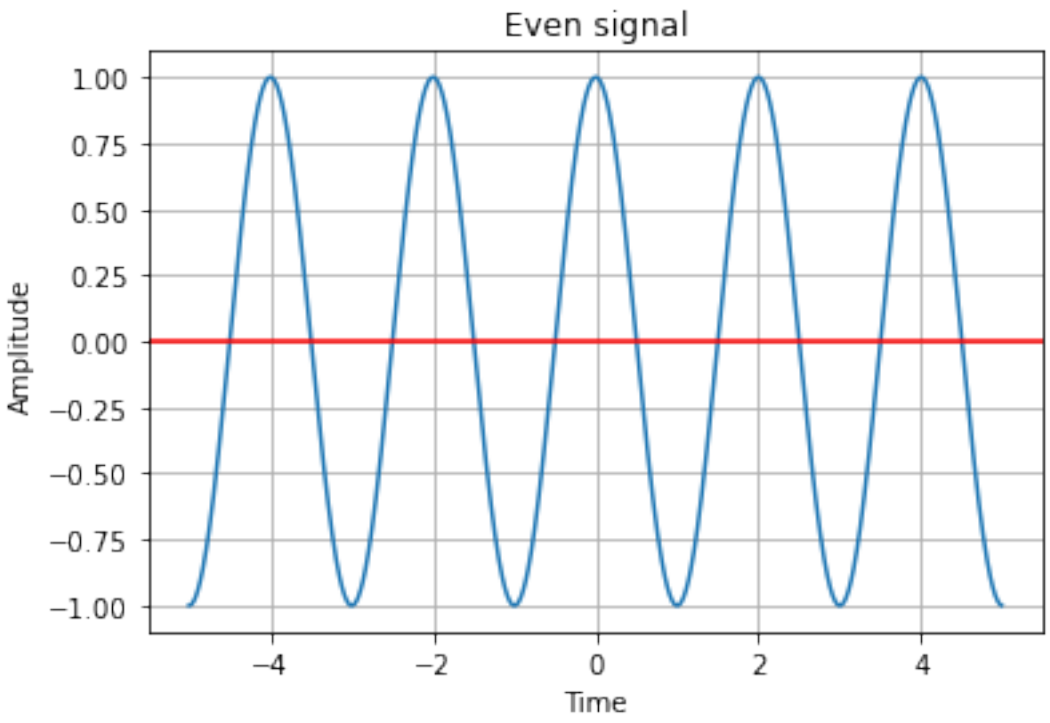
1.1 Question 1

Plot the following signals

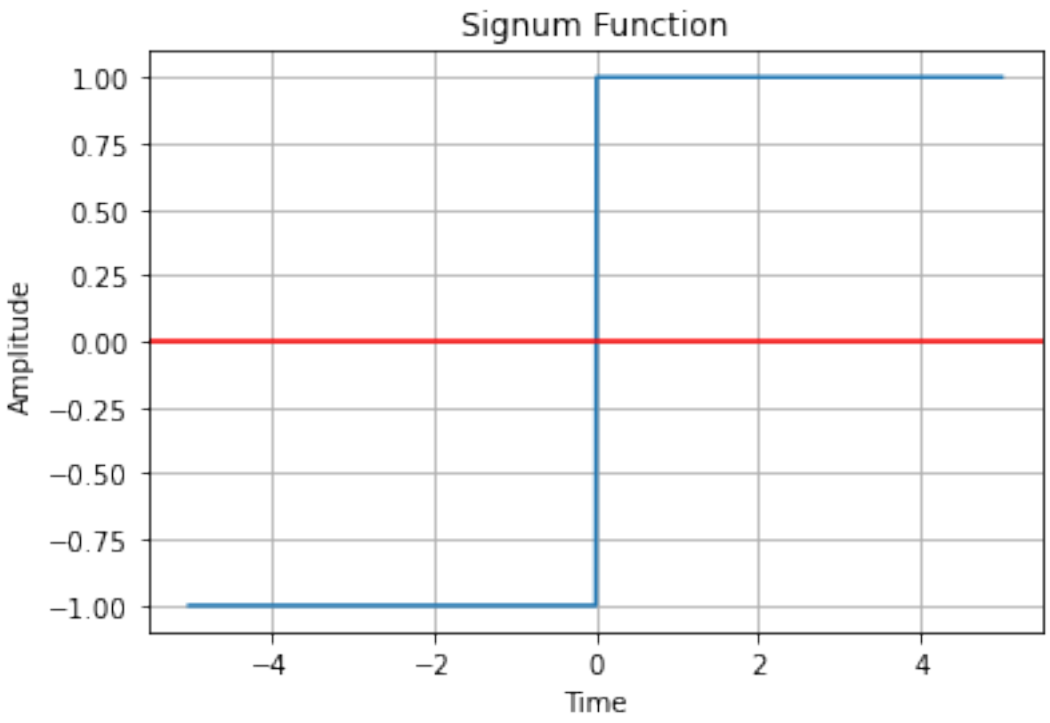
[2]:
```
# a. Odd Signal

time = np.arange(-5, 5, 0.01)
amp = np.sin(time*np.pi)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Odd signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



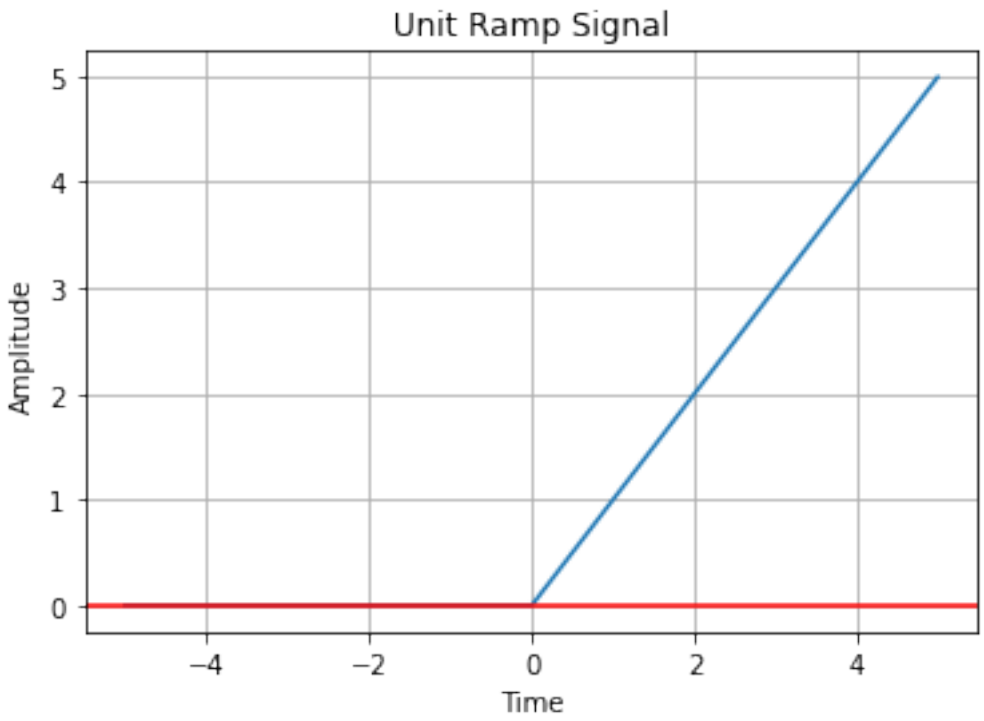
[3]:
```
# b. Even Signal

time = np.arange(-5, 5, 0.01)
amp = np.cos(time*np.pi)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Even signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



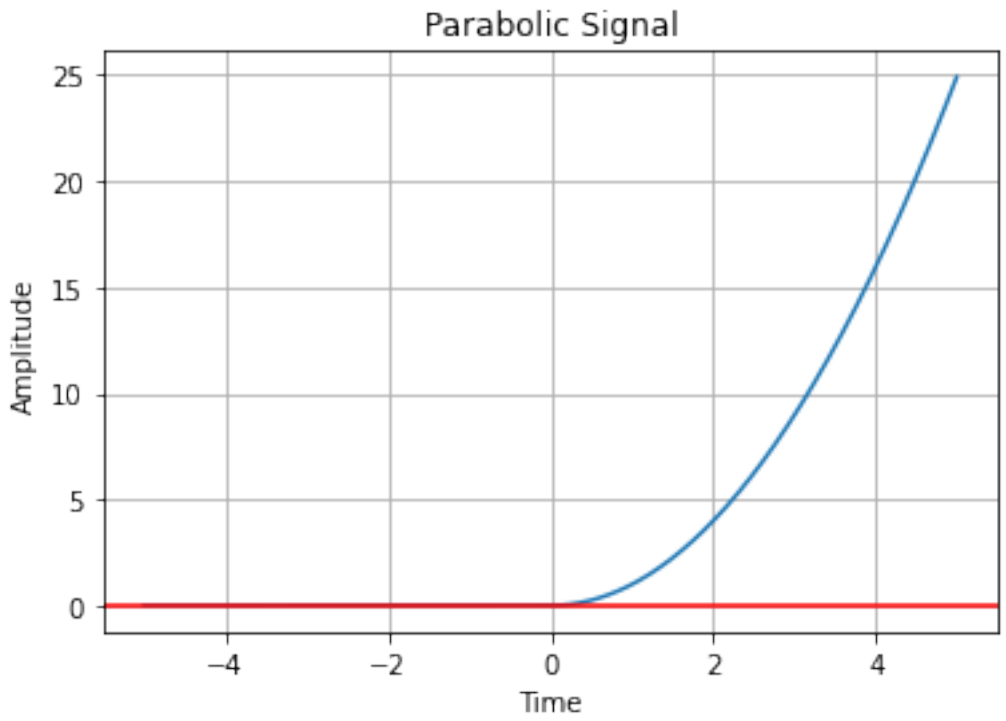
[4]:
```
# c. Signum Function

time = np.arange(-5, 5, 0.01)
amp = np.zeros(np.size(time))
index =  np.where(time < 0)
amp[index] = -1
index = np.where(time > 0)
amp[index] = 1
index = np.where(time == 0)
amp[index] = 0

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Signum Function')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



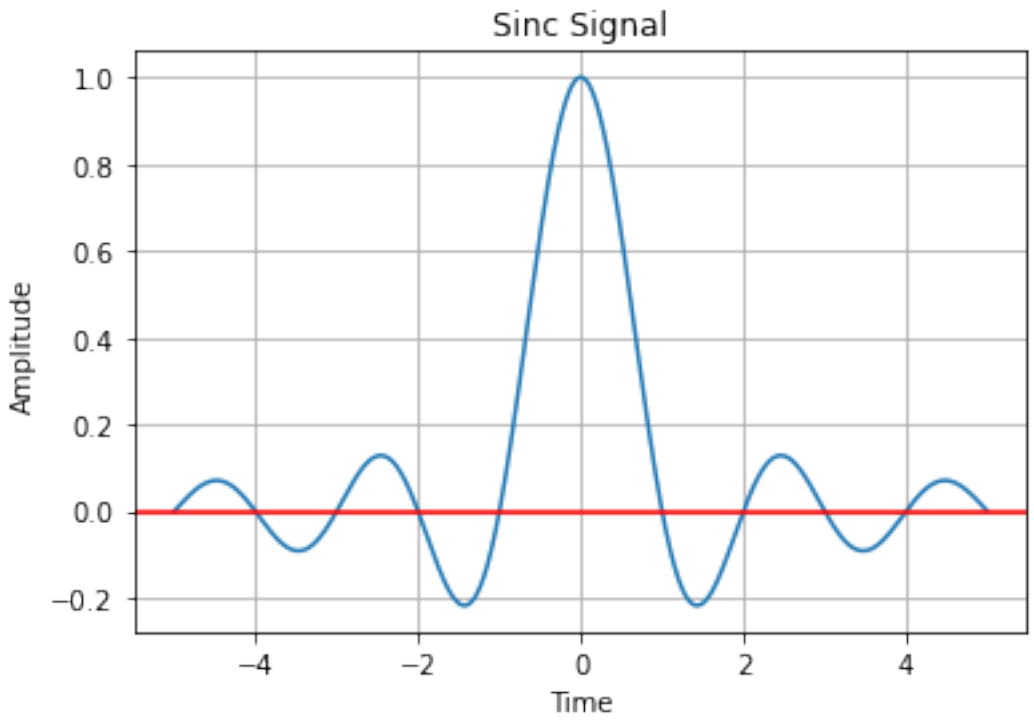
[5]:
```
# d. Unit Ramp Signal

time = np.arange(-5, 5, 0.01)
amp = np.zeros(np.size(time))
index = np.where(time >= 0)
amp[index] = time[index]

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Unit Ramp Signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



[6]:
```
# e. Parabolic Signal

time = np.arange(-5, 5, 0.01)
amp = np.zeros(np.size(time))
index = np.where(time >= 0)
amp[index] = time[index] ** 2

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Parabolic Signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```

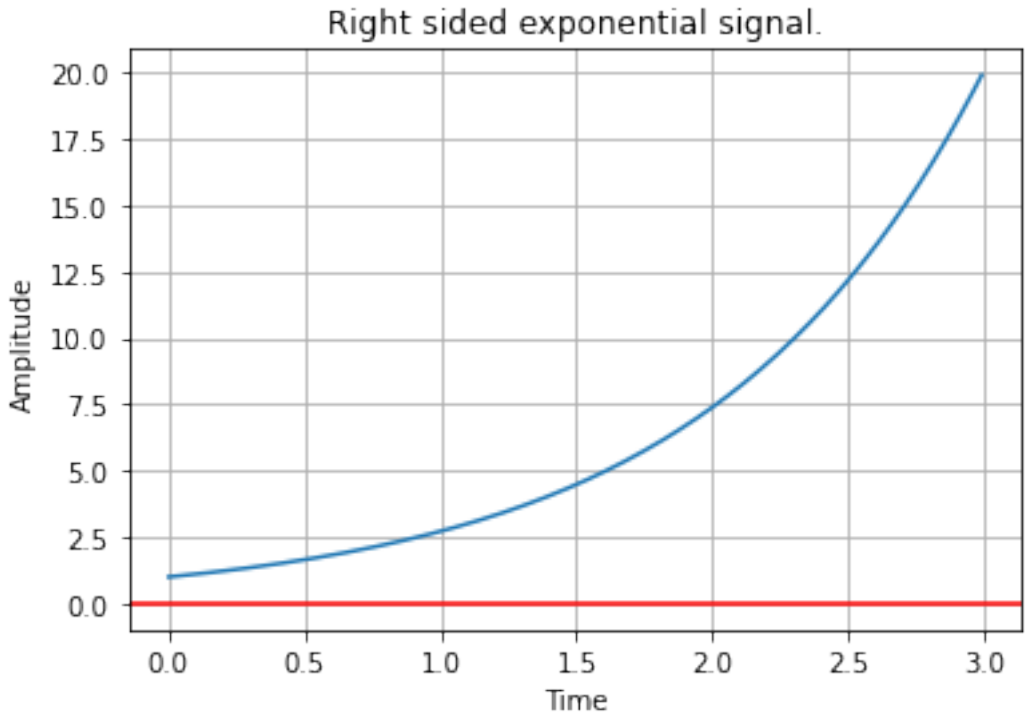


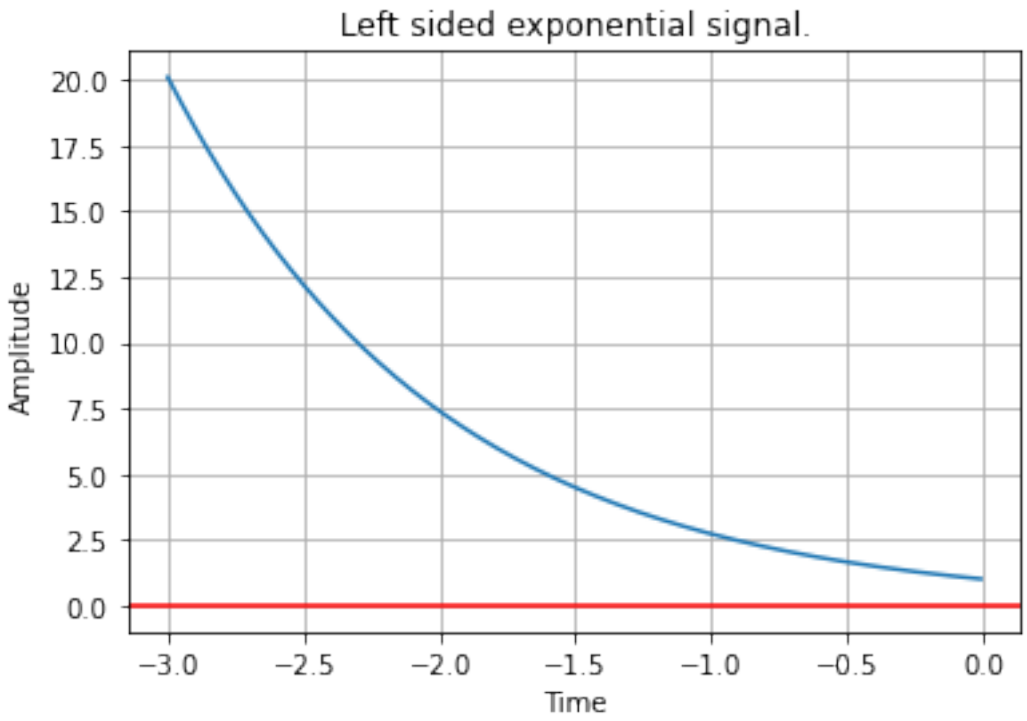
[7]:
```
# f. Sinc Signal

time = np.arange(-5, 5, 0.01)
amp = np.sin(time*np.pi)/(time*np.pi)

plt.figure()
plt.plot(time,amp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Sinc Signal')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```

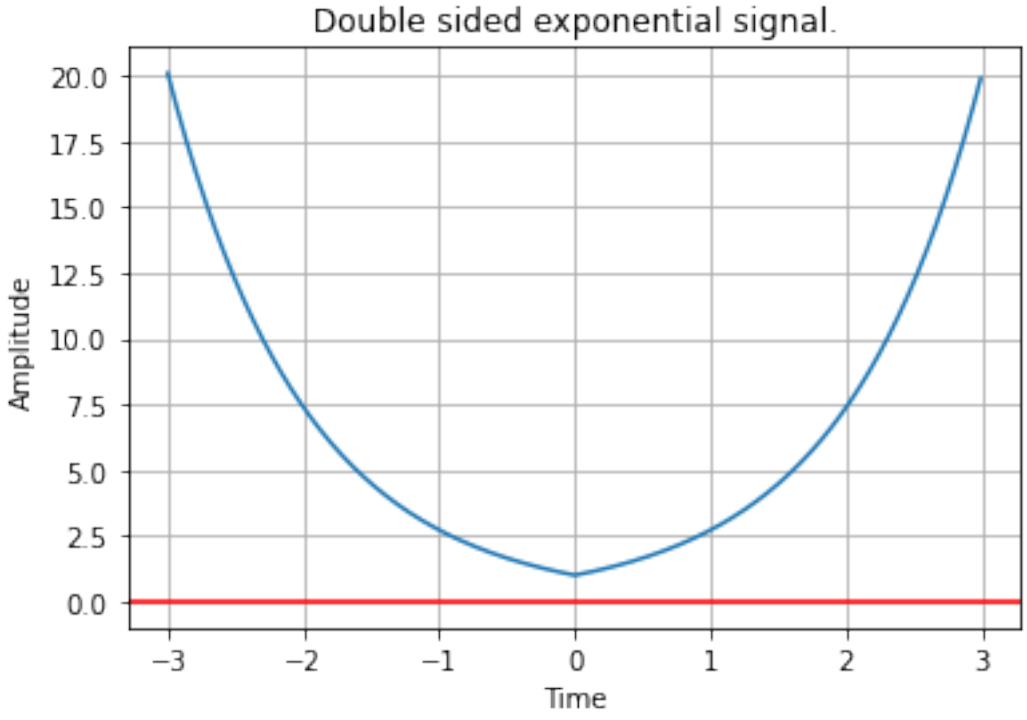


[8]:
```
# g. Right ,left and double sided exponential signal

rtime = np.arange(0, 3, 0.01)
ltime = np.arange(-3, 0, 0.01)
time = np.arange(-3, 3, 0.01)
rexp = np.exp(rtime)
lexp = np.exp(-ltime)

doub_exp = np.exp(time)
index = np.where(time < 0)
doub_exp[index] = np.exp(-time[index])
index

plt.figure()
plt.plot(rtime,rexp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Right sided exponential signal.')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
```

```
plt.plot(ltime,lexp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Left sided exponential signal.')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure()
plt.plot(time,doub_exp)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Double sided exponential signal.')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()
```



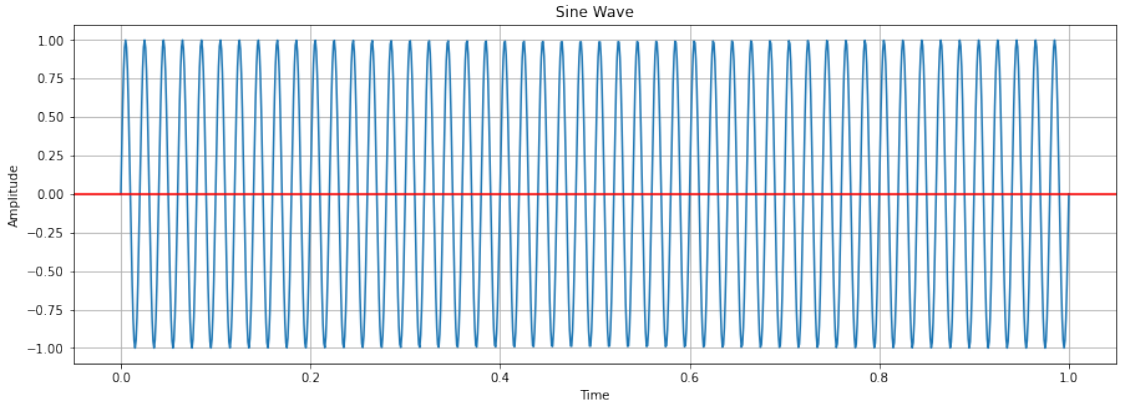

Left sided exponential signal.
Amplitude
Time
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0

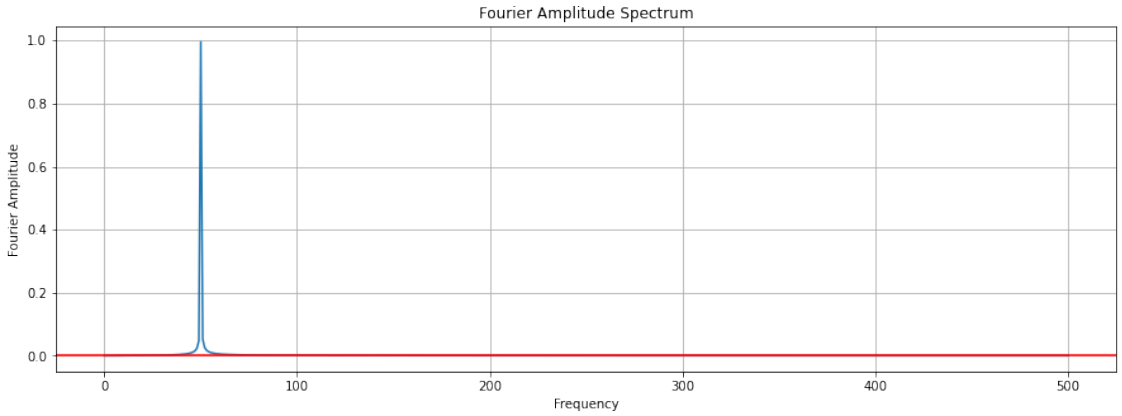

Double sided exponential signal.
Amplitude
Time
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−3
−2
−1
0
1
2
3

1.2 Question 2

Plot the Sine Wave and Fourier amplitude Spectrum of frequency 50Hz, sampled at 1ms.

[9]:
```
N = 1000
T = 1/1000
time = np.linspace(0, N*T, N)
y_sin = np.sin(2*np.pi*50*time)
yf = fft(y_sin)
xf = np.linspace(0, 1/(2*T), N//2)

plt.figure(figsize=(15,5))
plt.plot(time, y_sin)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Sine Wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure(figsize=(15,5))
plt.plot(xf, 2/N * np.abs(yf[0:N//2]))
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Fourier Amplitude Spectrum')
plt.xlabel('Frequency')
plt.ylabel('Fourier Amplitude')
plt.show()
```

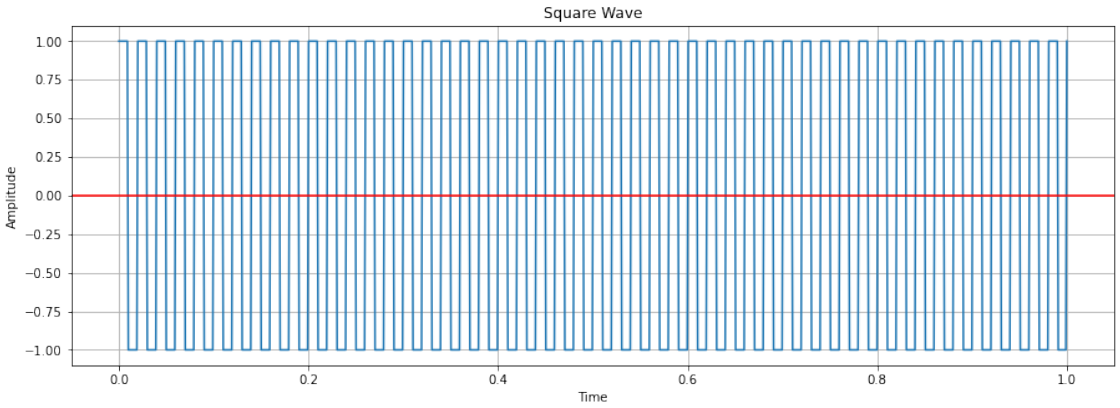


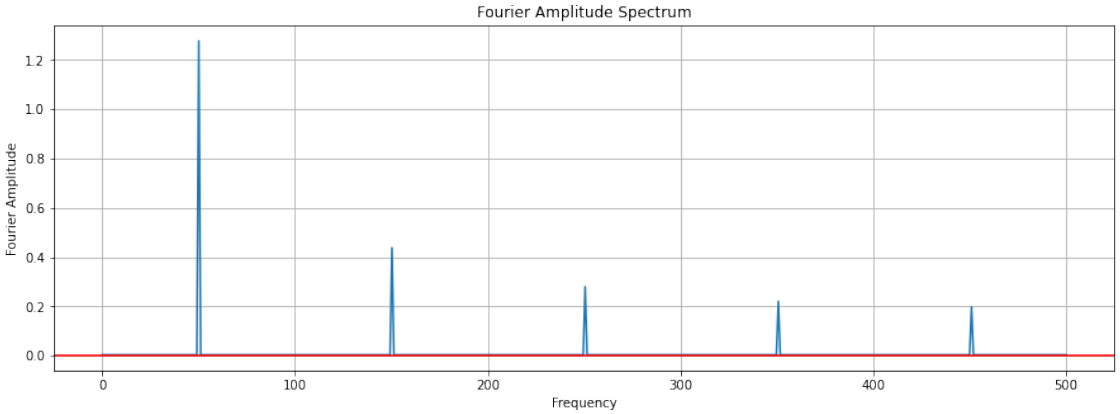


1.3 Question 3

Plot the Square Wave and Fourier amplitude Spectrum of frequency 50Hz, sampled at 1ms.

[10]:
```
N = 1000
T = 1/1000
time = np.linspace(0, N*T, N)
y_sq = signal.square(2*np.pi*50*time)
yf = fft(y_sq)
xf = np.linspace(0, 1/(2*T), N//2)

plt.figure(figsize=(15,5))
plt.plot(time, y_sq)
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Square Wave')
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.show()

plt.figure(figsize=(15,5))
plt.plot(xf, 2/N * np.abs(yf[0:N//2]))
plt.axhline(y=0, c='r')
plt.grid()
plt.title('Fourier Amplitude Spectrum')
plt.xlabel('Frequency')
plt.ylabel('Fourier Amplitude')
plt.show()
```


Square Wave
Amplitude
Time
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.2
0.4
0.6
0.8
1.0


Fourier Amplitude Spectrum
Fourier Amplitude
Frequency
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500