LAB REPORT

Communication Lab I
(ELC 3920)

Experiment No.: 1

S. No:	**1 2**
F. No:	2 0 E L B 0 8 4
Name:	Y U S U F A H M E D K H A N

Object:

(a). Determine the first 10 harmonics of a square wave of given frequency and amplitude using spectrum analyser. Compare the experimental values with the theoretical values of harmonics obtained from Fourier series.
(b). Synthesize the above square wave by summing the first 3, 5, and 10 harmonics.

Group Members: Syed Imaduddin, ***Yusuf Ahmed Khan***, Sarim Khan and Pratyush Kaushik.

SIMULATION ATTACHED

Date of performing the experiment: 14/11/2022

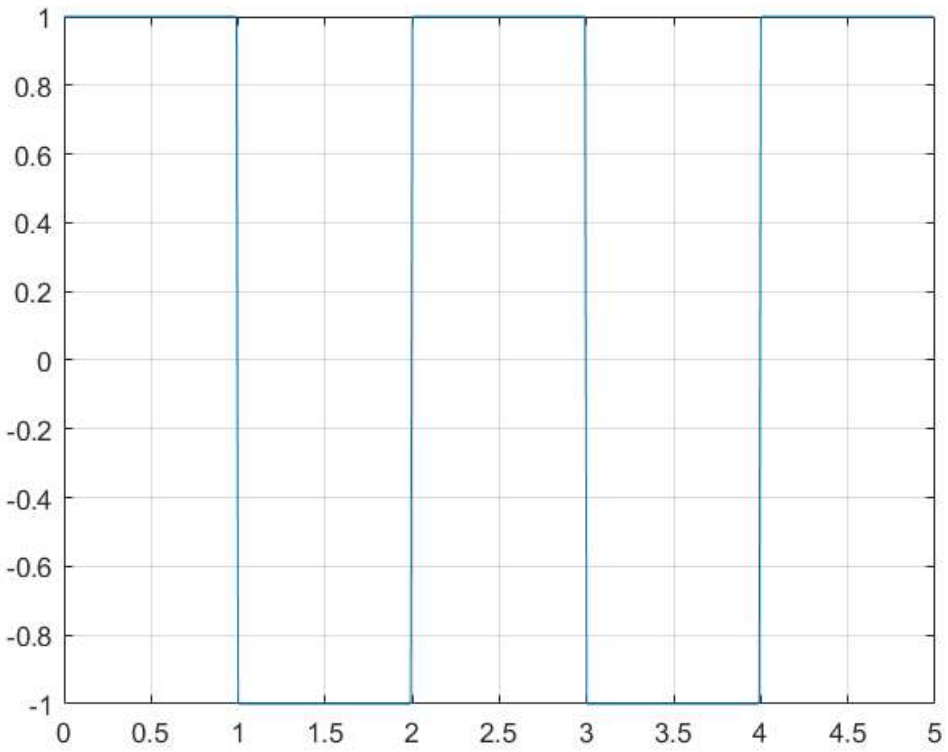
Date of report submission: 21/11/2022

Experiment 1

By Yusuf Ahmed Khan, Serial Number: 12

Approximation of Square Wave using Fourier Series

```
fs=100;
T=2;
w0=2*pi/T;

k=0:1/fs:5-1/fs;
y=square(w0*k, 50);

figure
plot(k,y)
grid on
```




```
syms t
N=5;
n=1:N;

a0=(2/T)*(int(1,t,0,1)+int(-1,t,1,2))
```

```
a0 = 0
```

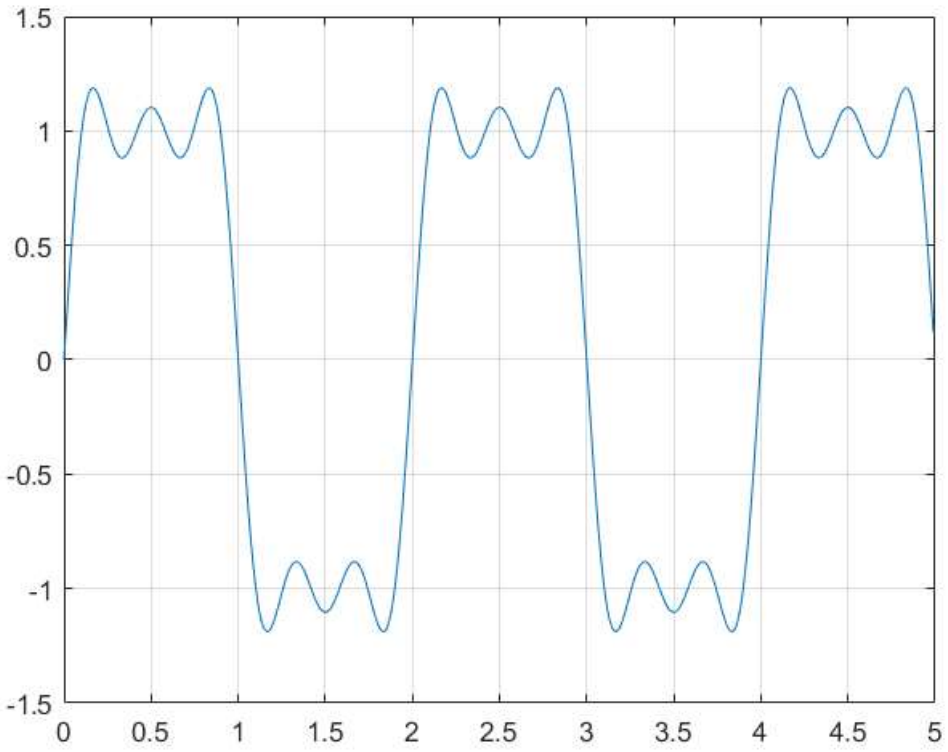
```
an=(2/T)*(int(1*cos(n*w0*t),t,0,1)+int(-1*cos(n*w0*t),t,1,2))
```

```
an = (0  0  0  0  0)
```

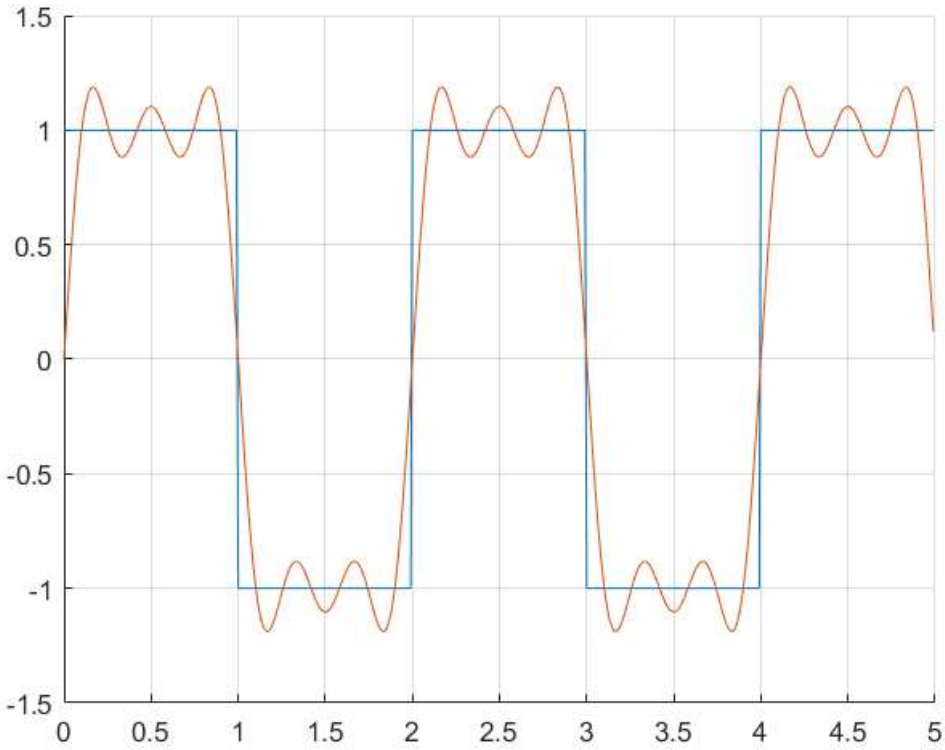
```
bn=(2/T)*(int(1*sin(n*w0*t),t,0,1)+int(-1*sin(n*w0*t),t,1,2))
```

```
bn =
(4/π  0  4/(3π)  0  4/(5π))
```

```
F=a0/2;

for i=1:N
    F=F+an(i)*cos(i*w0*k)+bn(i)*sin(i*w0*k);

end


figure
plot(k,F)
grid on
```




```
figure
hold on
plot(k,y)
plot(k,F)
hold off
grid on
```


1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Experiment 1

Approach 2 (Python)

By Yusuf Ahmed Khan, Serial Number:12

HARMONIC ANALYSIS AND WAVEFORM SYNTHESIS

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("ggplot")
```

Amplitude and Frequency of the square wave

In [3]:
```
#Enter the Amplitude
amp = 2

#Enter the Frequency
freq = 3000
```

In [4]:
```
def count_zeros(number):
    count = 0
    while number > 9:
        count += int(number % 10 == 0)
        number //= 10
    return count

q = count_zeros(freq)

t = 1/freq
new_time = t*pow(10,q+1)
print(new_time)

T = new_time

x_axis = []
y_axis = []


x_ = np.linspace(0,20,500)
f = x_[1]-x_[0]
```

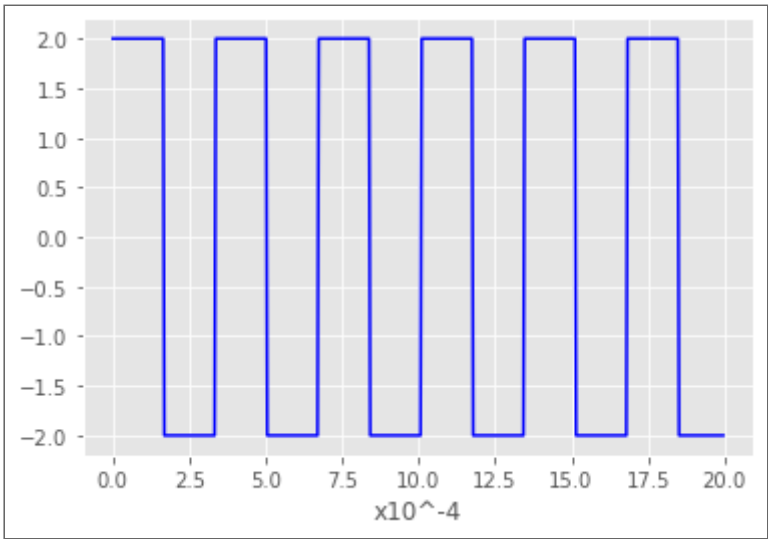
```
dt =f
ttemp = 0 #this is to count periods
```

```
3.333333333333333
```

Plotting the square wave

In [5]:

```
            while t<20:
  if ttemp>new_time:
    x_axis.append(t)
    y_axis.append(amp)
    ttemp=0


  if ttemp>new_time/2:

    x_axis.append(t)
    b=-amp
    y_axis.append(b)

  else:

    x_axis.append(t)
    y_axis.append(amp)


  t=t+dt
  ttemp=ttemp+dt

plt.xlabel(f"x10^-{q+1}")
plt.plot(x_axis,y_axis,color="blue")
plt.show()
```



For a signal x(t), the Fourier Series Representation is given by

$$\Rightarrow \quad x(t) = a_0 + \sum_{n=1}^{k} a_n \cos n\omega_0 t + b_n \sin n\omega_0 t \ldots (1)$$

Since the given square wave x(t) is an odd signal, Therefore a0 and an will be zero And it's a also a half wave symmetric signal, therefore bn will only contain odd harmonics

Number of harmonics

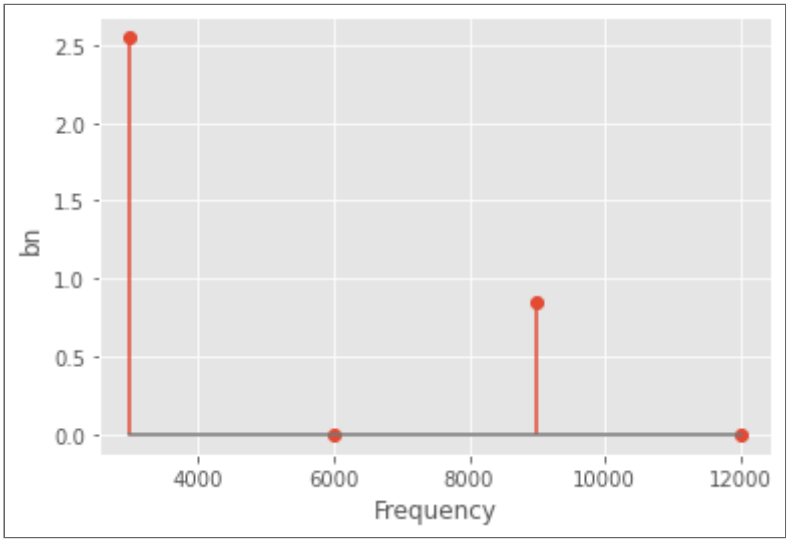
In [6]:
```
# Enter the no. of harmonics
Harmonics = 5
```

Plotting the coefficient bn

In [7]:
```
# Wn
def wn(n):
    global T
    wn = (2*np.pi*n)/T
    return wn


# Bn coefficients
def bn(n):
    n = int(n)
    if (n%2 != 0):
        return (4*amp)/(np.pi*n)
    else:
        return 0

bnh=[]
for n in range(1,Harmonics):
    bnh.append(bn(n))

print(bnh)


bnh_x=[]
for i in range(1,Harmonics):
    h = freq*i
    bnh_x.append(h)

plt.xlabel("Frequency")
```

```
plt.ylabel("bn")
plt.stem(bnh_x,bnh)
plt.show()
```

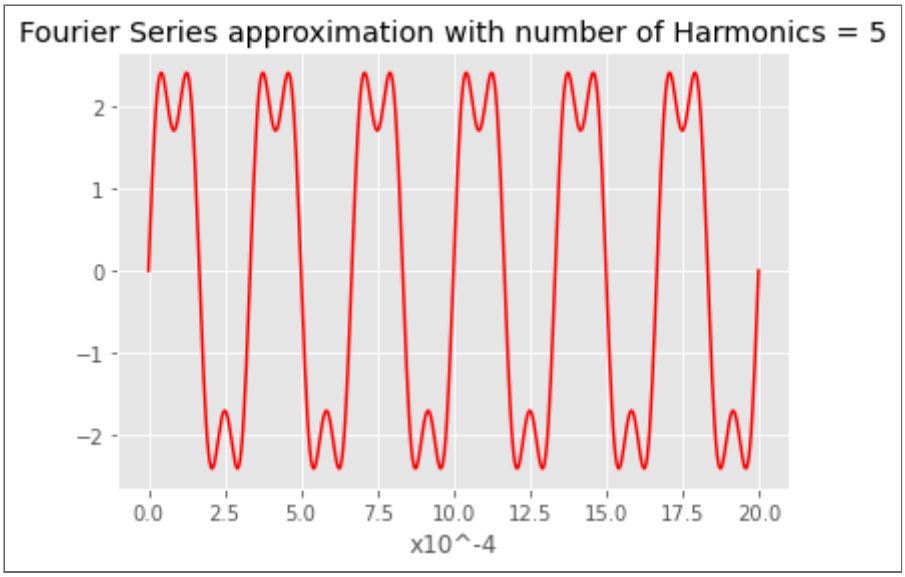
```
[2.5464790894703255, 0, 0.8488263631567752,
0]
```



Synthesizing the square from it's Harmonic components

In [10]:

```
# Fourier Series function
def fourierSeries(n_max,x):
    a0 = 0
    partialSums = a0
    for n in range(1,n_max):
        try:
            partialSums = partialSums + bn(n)*np.sin(wn(n)*x)
        except:
            print("pass")
            pass
    return partialSums



f = []

x_ = np.linspace(0,20,500)
for i in x_:
    f.append(fourierSeries(Harmonics,i))



plt.xlabel(f"x10^-{q+1}")
plt.plot(x_,f,color="red")
plt.title("Fourier Series approximation with number of Harmonics = "+str(Harmonics))
plt.show()
```


Fourier Series approximation with number of Harmonics = 5
2
1
0
−1
−2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
x10^-4