**Experiment-7**

**Name of the experiment:** To study and implement Gibbs phenomenon, Parseval’s theorem and Fourier transform.

**Theory:** Parseval’s theorem: Total power in a periodic signal equals to sum of squared amplitude of each harmonic. If x(t)->Cn, Power=Summation (Cn\*Cn) {from –Infinity to Infinity}

Gibb’s phenomenon: The sinusoidal components of the signal that occur at multiples of the fundamental frequency are called harmonics.

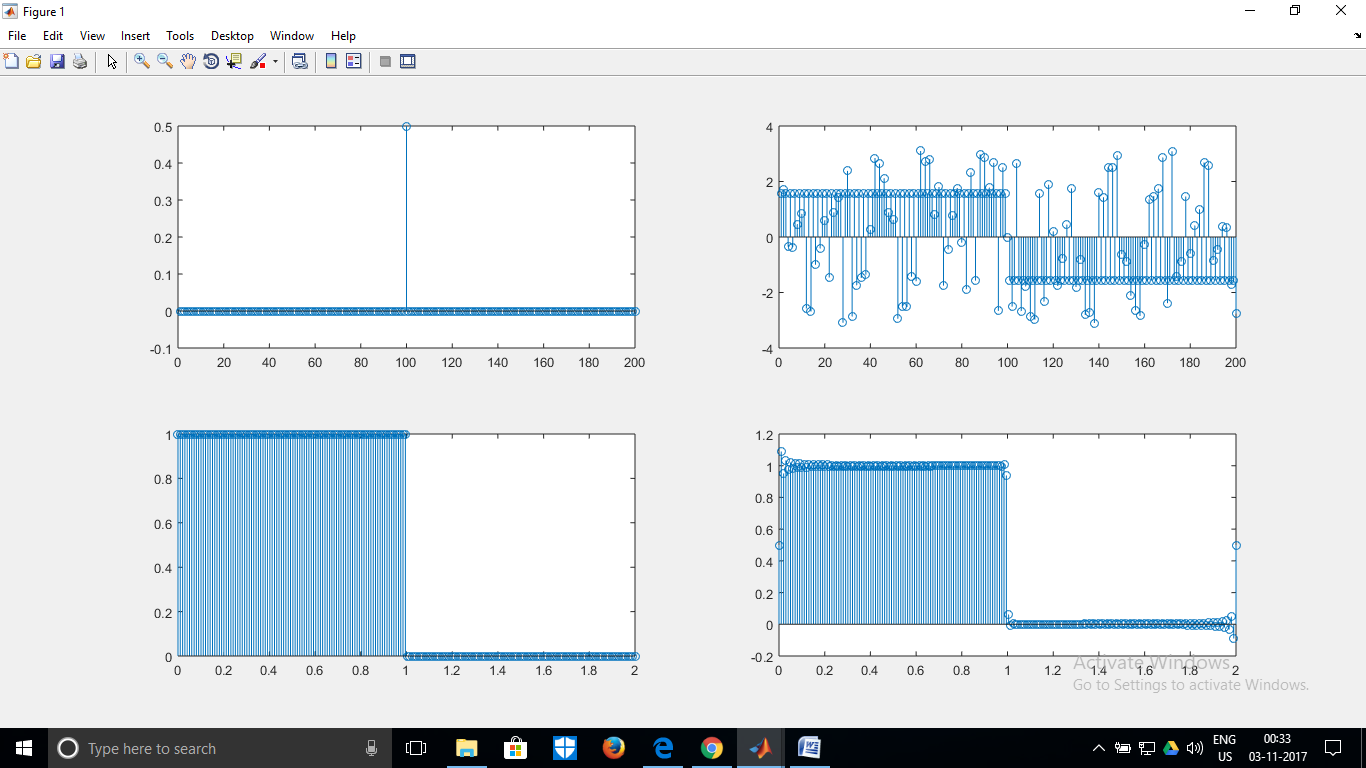
For well-behaved (continuous) periodic signals, a sufficiently large number of harmonics can be used to approximate the signal reasonably well. For periodic signals with discontinuities, however, such as a periodic square wave, even a large number of harmonics will not be sufficient to reproduce the square wave exactly. This effect is known as Gibbs phenomenon and it manifests itself in the form of ripples of increasing frequency and closer to the transitions of the square signal.

Fourier transform: The Fourier transform of a signal f (t) is calculated as

(𝜔) = ∫ 𝑓(𝑡) 𝑒−𝑗𝜔𝑡𝑑𝑡 [Infinity to -Infinity]

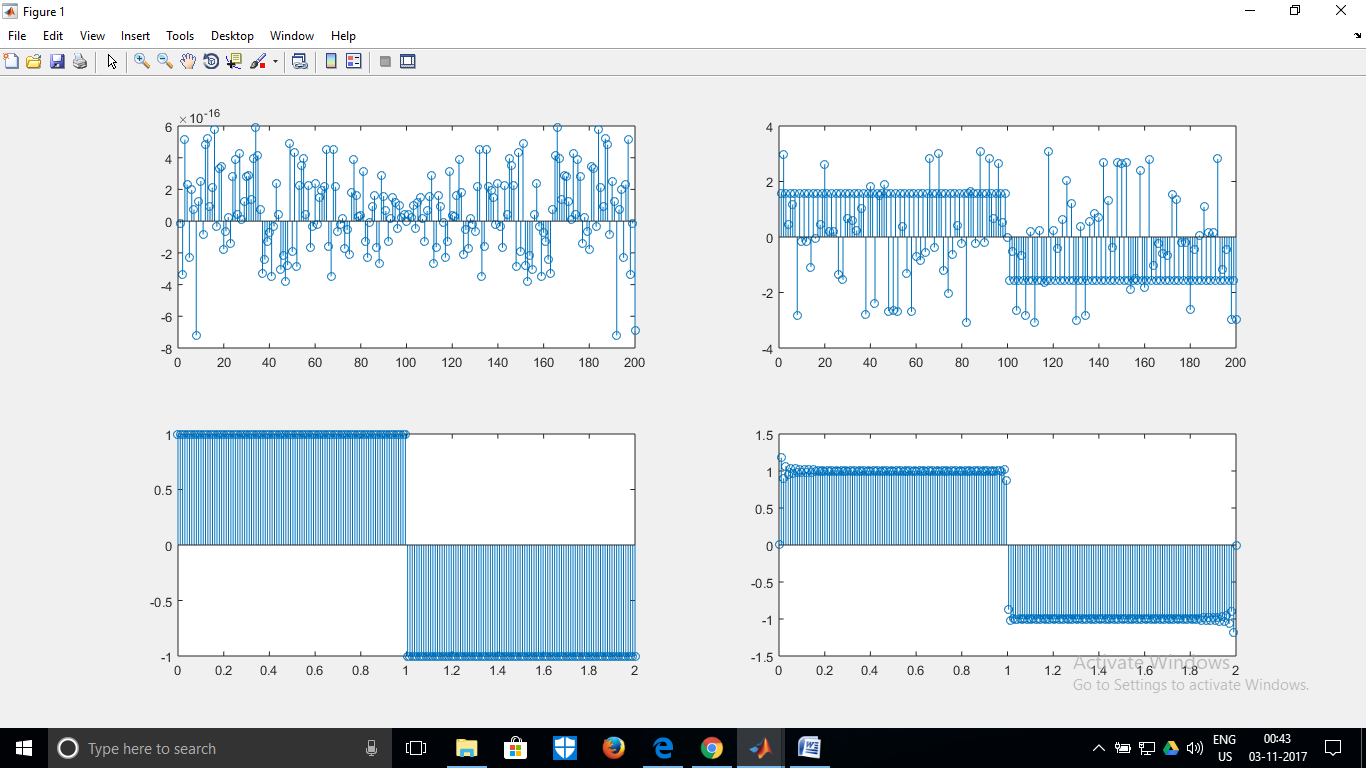
Magnitude Spectrum= |F(w)|, Phase spectrum = Angle(F(w))

**Observation and Results:**

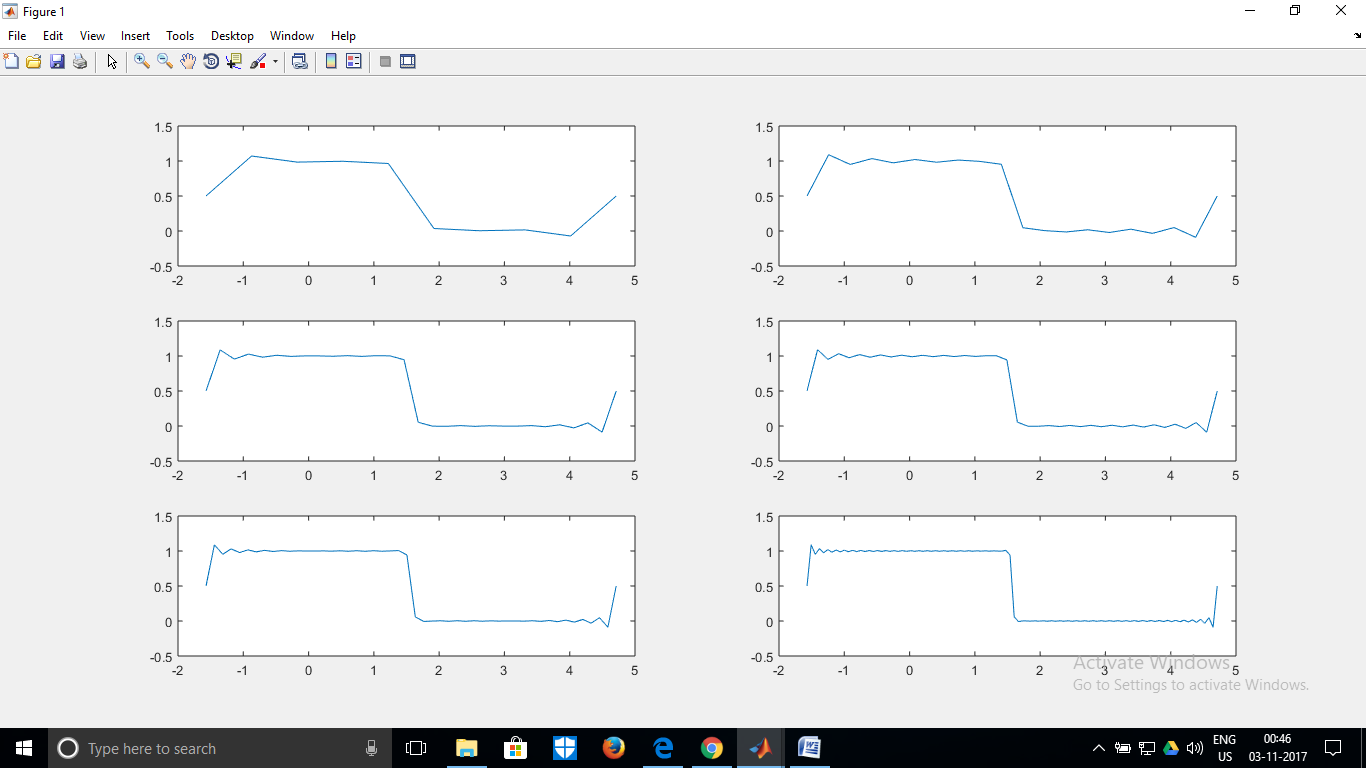


1. The amplitude, phase spectrum, actual signal and Fourier reconstruction (in order)

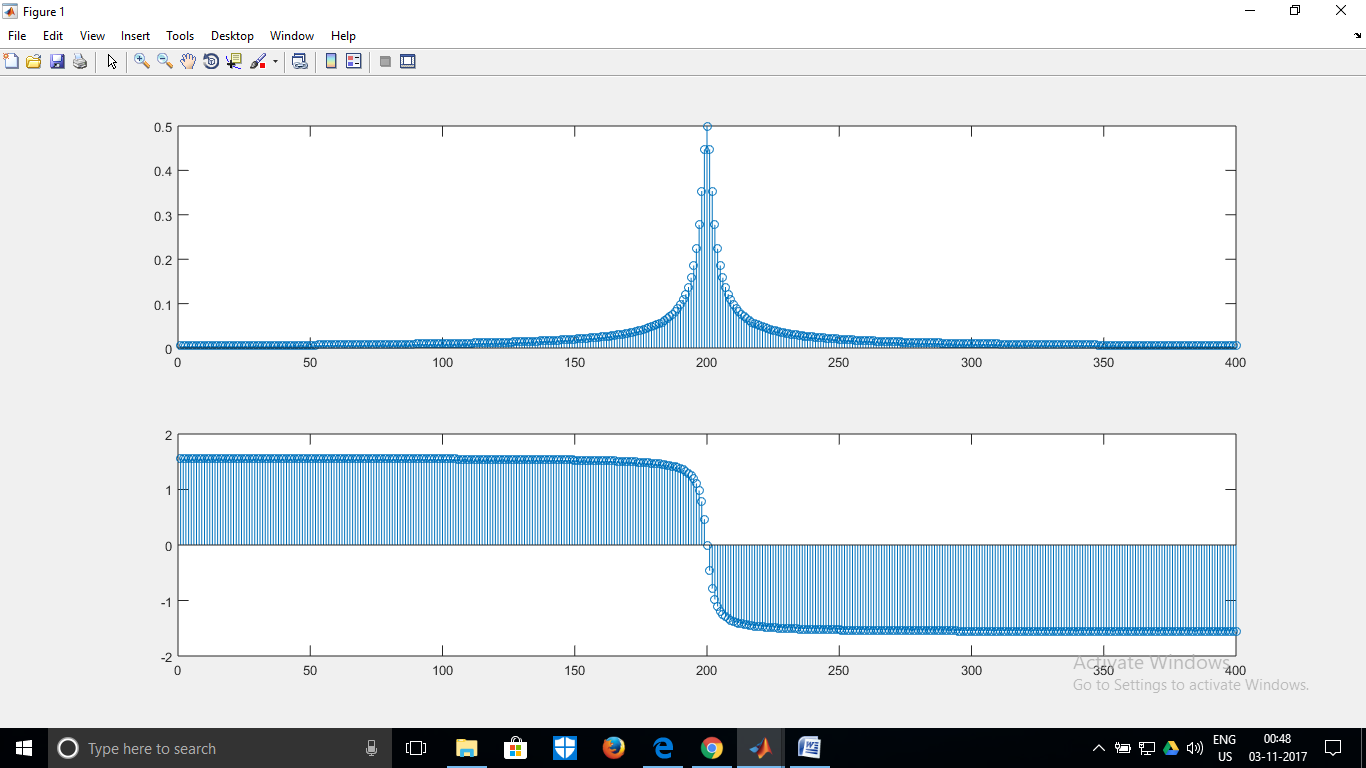
Total power=0.4990, power upto second harmonic=0.4526



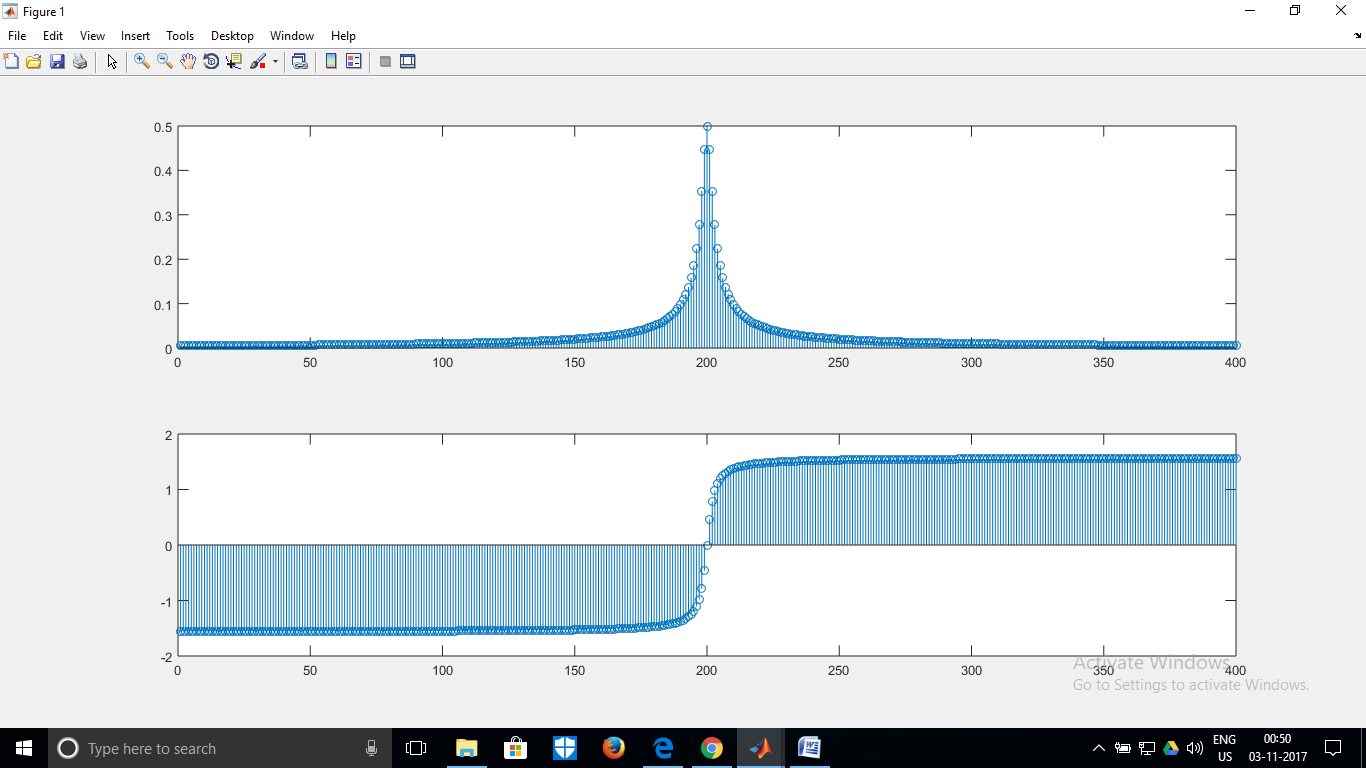
Ratio of power in 7th and 5th harmonic= 0.5102



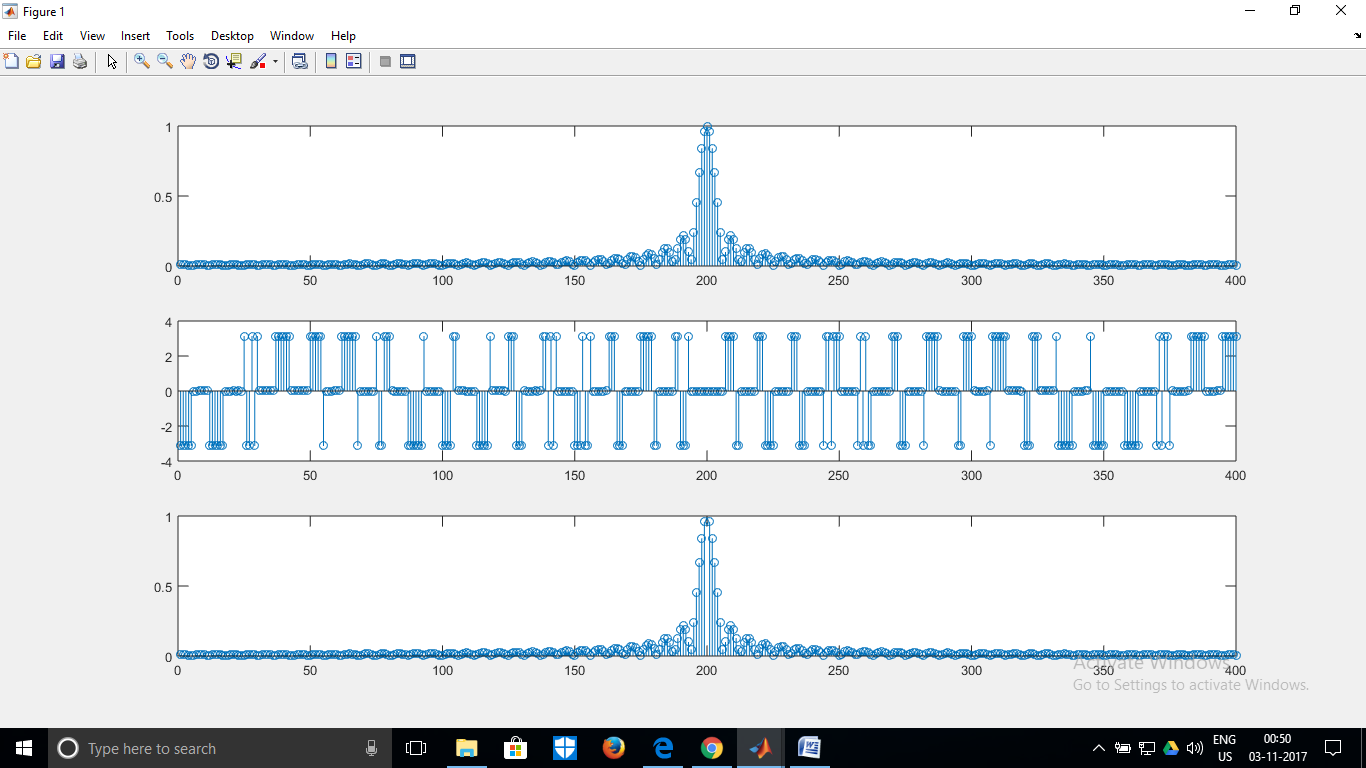
Gibbs Phenomenon, demonstrated. Clearly, the overshoots don’t vanish even as approximations keep getting better. The error energy=2.6456, overshoot=84.2136%



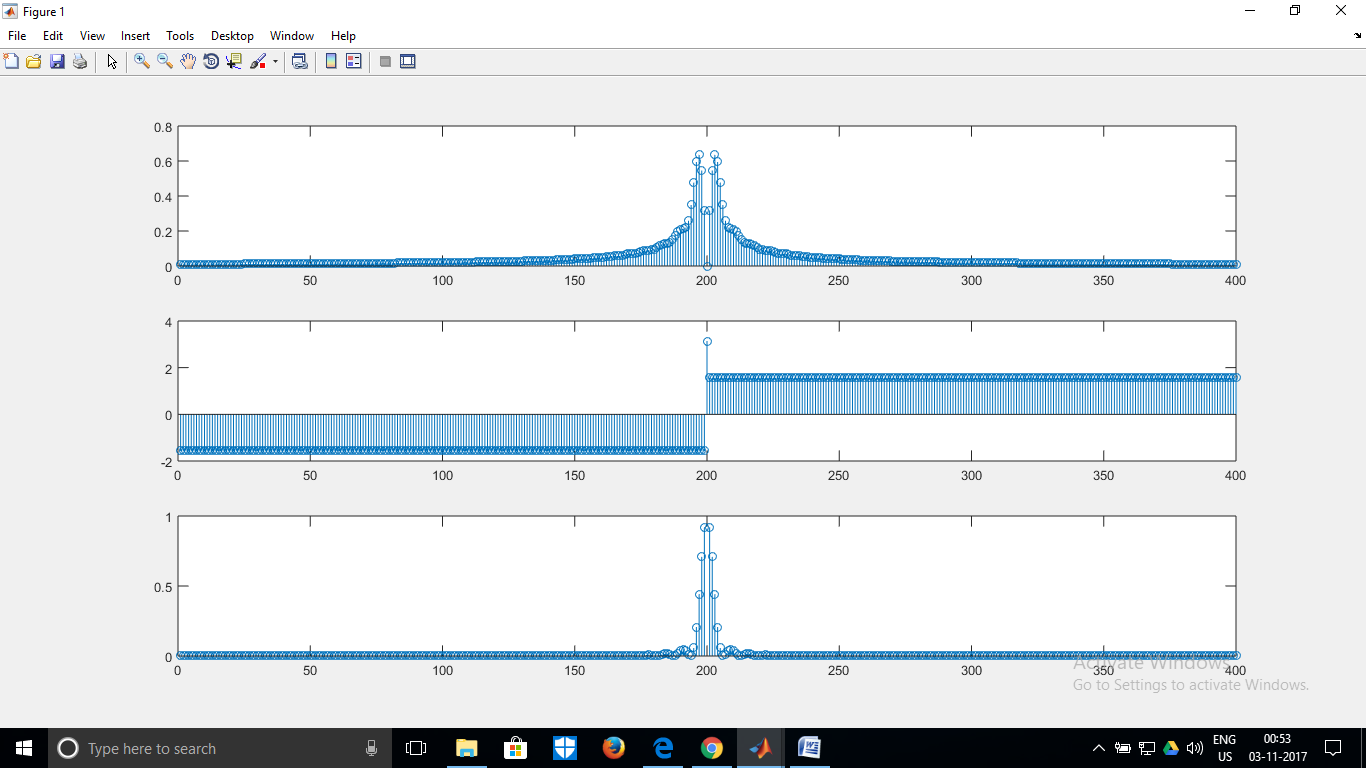
4. a. Amplitude and phase spectrum of e-atu(t), a=2



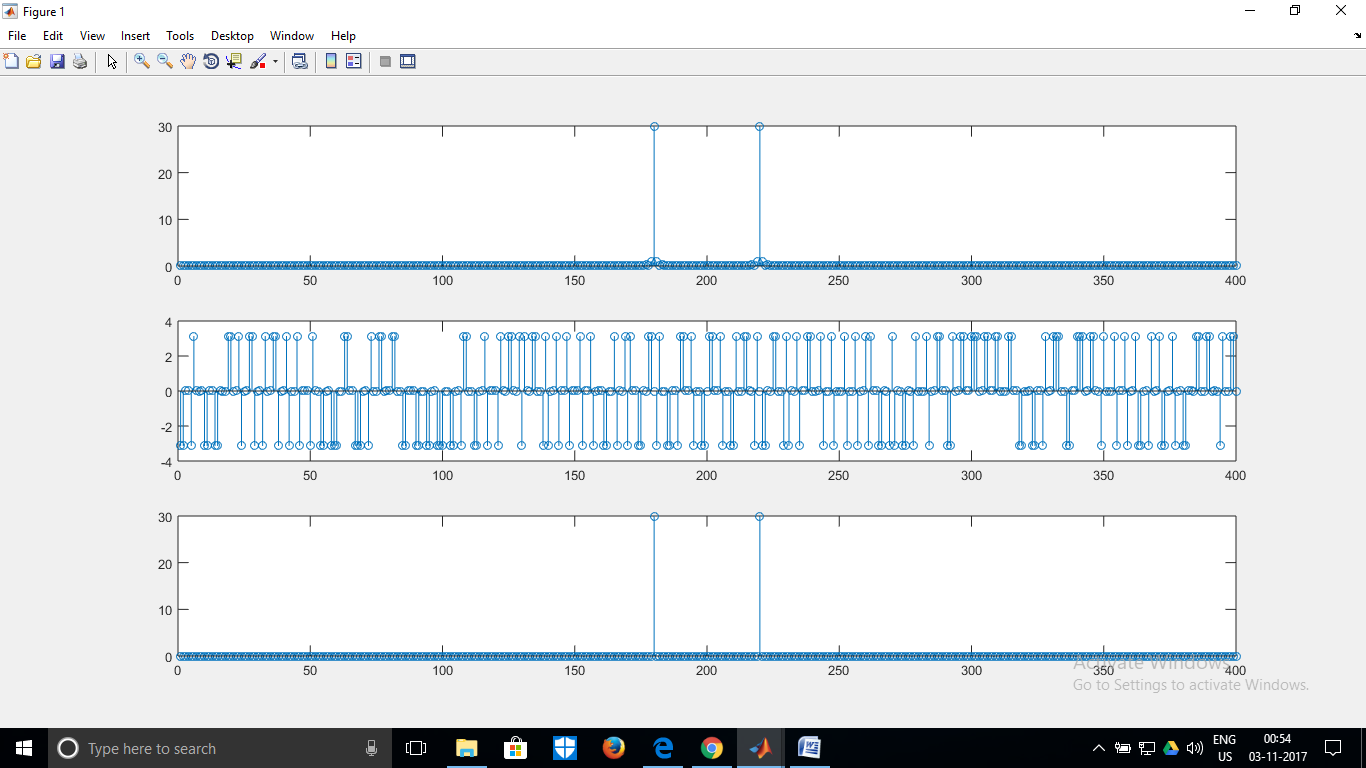
4. b. Amplitude and phase spectrum of eatu(t), a=2



4. c. Amplitude and phase spectrum of rect (t/tau), tau=1. Third plot=Verification by inbuilt Fourier transform.



4. d. Amplitude and phase spectrum of tri(t/tau), tau=1. Third plot=Verification by inbuilt Fourier transform.



4. e. Amplitude and phase spectrum of cos(wt), w=20. Third plot=Verification by inbuilt Fourier transform.

**Conclusion:** This experiment helped clear our concepts about Gibbs phenomenon, Fourier Transform and Parseval’s theorem. It was interesting to observe how nicely a better and better approximation to square wave was obtained as we took more and more terms of the Fourier series. However, while trying to use the inbuilt Fourier transform for verification, sometimes we ran into trouble as the dirac function can’t be plotted; imaginary part are ignored and the makeFunction() command can’t always generate the exact function handle. However, we will try to ensure that these limitations are overcome by manipulating the input arguments.

**Appendix:** MATLAB code:

clc;

clear all;

close all;

% w = warning('query','last');

% id = w.identifier;

% warning('off',id);

n=input('Enter choice ');

switch(n)

case 1

cn=zeros(1,200);

pn=zeros(1,200);

power=0;

for i=1:1:200

sign1= @(t) 1.\*exp(-1j\*((i-100)\*pi\*t));

cn(i)=integral(sign1,0,1)/2;

pn(i)=angle(cn(i));

power=power+abs(cn(i))\*abs(cn(i));

end;

disp(power);

disp(2\*abs(cn(99))\*abs(cn(99))+2\*abs(cn(98))\*abs(cn(98))+abs(cn(100))\*abs(cn(100)));

figure;

subplot(2,2,1);

stem(cn);

subplot(2,2,2);

stem(pn);

t=linspace(0,2,200);

res=zeros(1,200);

for i=1:1:200

if(t(i)<=1)

res(i)=1;

else

res(i)=0;

end;

end;

subplot(2,2,3);

stem(t,res);

subplot(2,2,4);

g=zeros(1,200);

for i=1:1:200

for k=1:1:200

g(i)=g(i)+(cn(k))\*exp(((k-100)\*1j\*pi\*t(i)));

end;

end;

stem(t,g);

case 2

cn=zeros(1,200);

pn=zeros(1,200);

power=0;

for i=1:1:200

sign1= @(t) 1.\*exp(-1j\*((i-100)\*pi\*t));

sign2=@(t) -1.\*exp(-1j\*((i-100)\*pi\*t));

cn(i)=1\*(integral(sign1,0,1)+integral(sign2,1,2))/2;

pn(i)=angle(cn(i));

power=power+2\*abs(cn(i))\*abs(cn(i));

end;

disp(power);

disp(abs(cn(93))\*abs(cn(93))/(abs(cn(95))\*abs(cn(95))));

figure;

subplot(2,2,1);

stem(cn);

subplot(2,2,2);

stem(pn);

t=linspace(0,2,200);

res=zeros(1,200);

for i=1:1:200

if(t(i)<=1)

res(i)=1;

else

res(i)=-1;

end;

end;

subplot(2,2,3);

stem(t,res);

subplot(2,2,4);

g=zeros(1,200);

for i=1:1:200

for k=1:1:200

g(i)=g(i)+(cn(k)).\*exp(((k-100)\*1j\*pi\*t(i)));

end;

end;

stem(t,g);

case 3

figure

cn=zeros(1,10);

for i=1:1:10

sign1= @(t) 1.\*exp(-1j\*((i-5)\*1\*t));

cn(i)=1\*(integral(sign1,-pi/2,pi/2))/(2\*pi);

end;

g=zeros(1,10);

t=linspace(-pi/2,3\*pi/2,10);

for i=1:1:10

for k=1:1:10

g(i)=g(i)+(cn(k)).\*exp(((k-5)\*1j\*t(i)));

end;

end;

subplot(3,2,1);

plot(t,g);

cn=zeros(1,20);

for i=1:1:20

sign1= @(t) 1.\*exp(-1j\*((i-10)\*1\*t));

cn(i)=1\*(integral(sign1,-pi/2,pi/2))/(2\*pi);

end;

g=zeros(1,20);

t=linspace(-pi/2,3\*pi/2,20);

for i=1:1:20

for k=1:1:20

g(i)=g(i)+(cn(k)).\*exp(((k-10)\*1j\*t(i)));

end;

end;

subplot(3,2,2);

plot(t,g);

cn=zeros(1,30);

for i=1:1:30

sign1= @(t) 1.\*exp(-1j\*((i-15)\*1\*t));

cn(i)=1\*(integral(sign1,-pi/2,pi/2))/(2\*pi);

end;

g=zeros(1,30);

t=linspace(-pi/2,3\*pi/2,30);

for i=1:1:30

for k=1:1:30

g(i)=g(i)+(cn(k)).\*exp(((k-15)\*1j\*t(i)));

end;

end;

subplot(3,2,3);

plot(t,g);

cn=zeros(1,40);

t=linspace(-pi/2,3\*pi/2,40);

for i=1:1:40

sign1= @(t) 1.\*exp(-1j\*((i-20)\*1\*t));

cn(i)=1\*(integral(sign1,-pi/2,pi/2))/(2\*pi);

end;

g=zeros(1,40);

for i=1:1:40

for k=1:1:40

g(i)=g(i)+(cn(k)).\*exp(((k-20)\*1j\*t(i)));

end;

end;

subplot(3,2,4);

plot(t,g);

cn=zeros(1,50);

power=0;

for i=1:1:50

sign1= @(t) 1.\*exp(-1j\*((i-25)\*1\*t));

cn(i)=1\*(integral(sign1,-pi/2,pi/2))/(2\*pi);

power=power+abs(cn(i))\*abs(cn(i));

end;

t=linspace(-pi/2,3\*pi/2,50);

g=zeros(1,50);

for i=1:1:50

for k=1:1:50

g(i)=g(i)+(cn(k)).\*exp(((k-25)\*1j\*t(i)));

end;

end;

subplot(3,2,5);

plot(t,g);

cn=zeros(1,100);

for i=1:1:100

sign1= @(t) 1.\*exp(-1j\*((i-50)\*1\*t));

cn(i)=1\*(integral(sign1,-pi/2,pi/2))/(2\*pi);

end;

t=linspace(-pi/2,3\*pi/2,100);

g=zeros(1,100);

for i=1:1:100

for k=1:1:100

g(i)=g(i)+(cn(k)).\*exp(((k-50)\*1j\*t(i)));

end;

end;

subplot(3,2,6);

plot(t,g);

disp('Error: ');

disp(pi-power);

disp('Overshoot ');

disp((pi-power)/(pi)\*100);

case 4

ch=input('Enter choice ');

switch(ch)

case 1

w=linspace(-199,200,400);

k=zeros(1,400);

for i=1:1:400

y=@(t) exp(-2\*t).\*exp(-1j\*w(i)\*t);

k(i)=integral(y,0,50);

end;

subplot(2,1,1);

stem(abs(k));

subplot(2,1,2);

stem(angle(k));

case 2

w=linspace(-199,200,400);

k=zeros(1,400);

for i=1:1:400

y=@(t) exp(2\*t).\*exp(-1j\*w(i)\*t);

k(i)=integral(y,-50,0);

end;

subplot(2,1,1);

stem(abs(k));

subplot(2,1,2);

stem(angle(k));

case 3

w=linspace(-199,200,400);

k=zeros(1,400);

for i=1:1:400

y=@(t) 1.\*exp(-1j\*w(i)\*t);

k(i)=integral(y,-0.5,0)+integral(y,0,0.5);

end;

subplot(3,1,1);

stem(abs(k));

subplot(3,1,2);

stem(angle(k));

syms x;

f=rectangularPulse(x);

ft\_f = fourier(f);

disp(ft\_f);

hand=matlabFunction(ft\_f);

ib=zeros(1,400);

for i=1:1:400

ib(i)=hand(w(i));

end;

subplot(3,1,3);

stem(abs(ib));

case 4

w=linspace(-199,200,400);

k=zeros(1,400);

for i=1:1:400

y1=@(t) (t+1).\*exp(-1j\*w(i)\*t);

y2=@(t) (t-1).\*exp(-1j\*w(i)\*t);

k(i)=integral(y1,-1,0)+integral(y2,0,1);

end;

subplot(3,1,1);

stem(abs(k));

subplot(3,1,2);

stem(angle(k));

syms x;

f=triangularPulse(x);

ft\_f = fourier(f);

disp(ft\_f);

hand=matlabFunction(ft\_f);

ib=zeros(1,400);

for i=1:1:400

ib(i)=hand(w(i));

end;

subplot(3,1,3);

stem((ib));

case 5

w=linspace(-199,200,400);

k=zeros(1,400);

for i=1:1:400

y=@(t) cos(20\*t).\*exp(-1j\*w(i)\*t);

k(i)=integral(y,-30,30);

end;

subplot(3,1,1);

stem(abs(k));

subplot(3,1,2);

stem(angle(k));

syms x;

f=cos(20\*x);

ft\_f = fourier(f);

disp(ft\_f);

hand=matlabFunction(ft\_f);

ib=zeros(1,400);

for i=1:1:400

ib(i)=hand(w(i));

if(ib(i)==Inf)

ib(i)=30;

end;

end;

subplot(3,1,3);

stem(abs(ib));

end;

end;