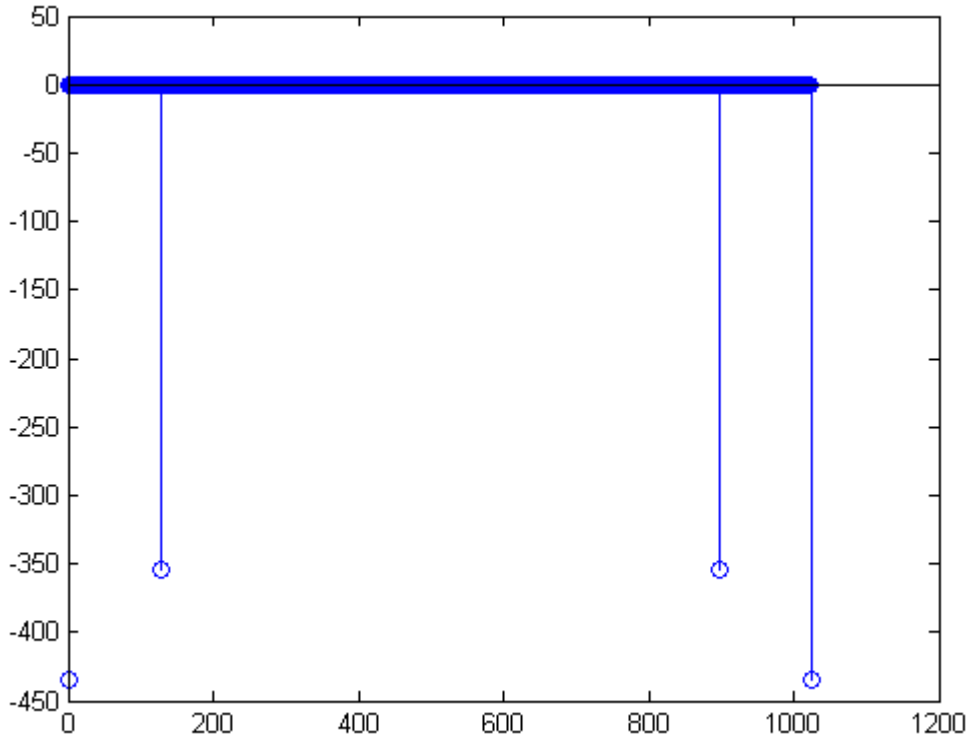
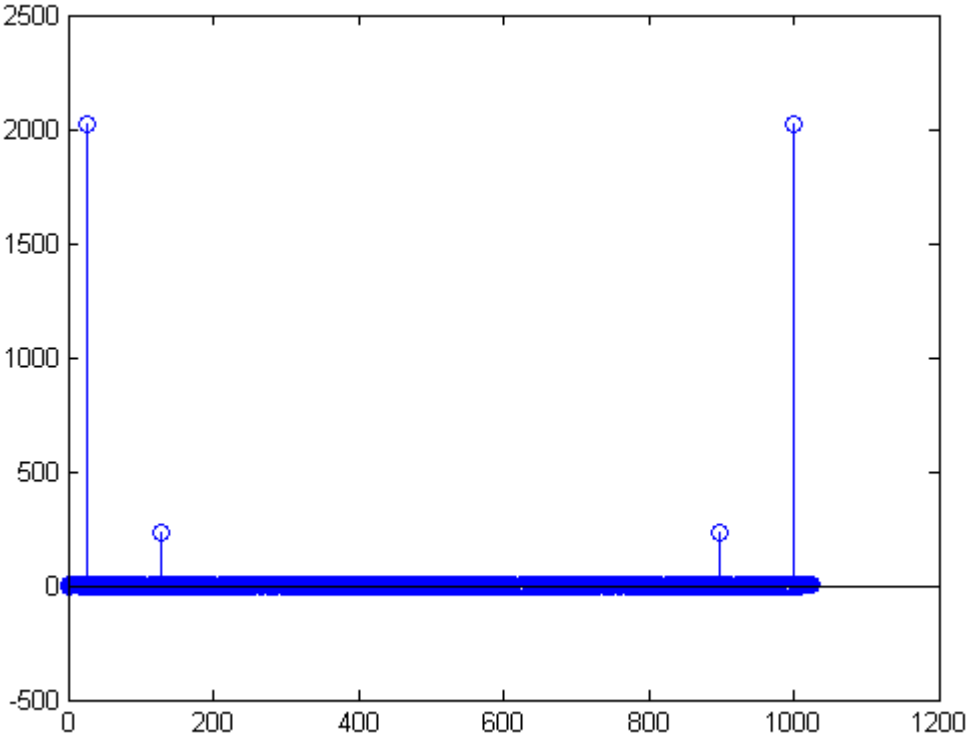
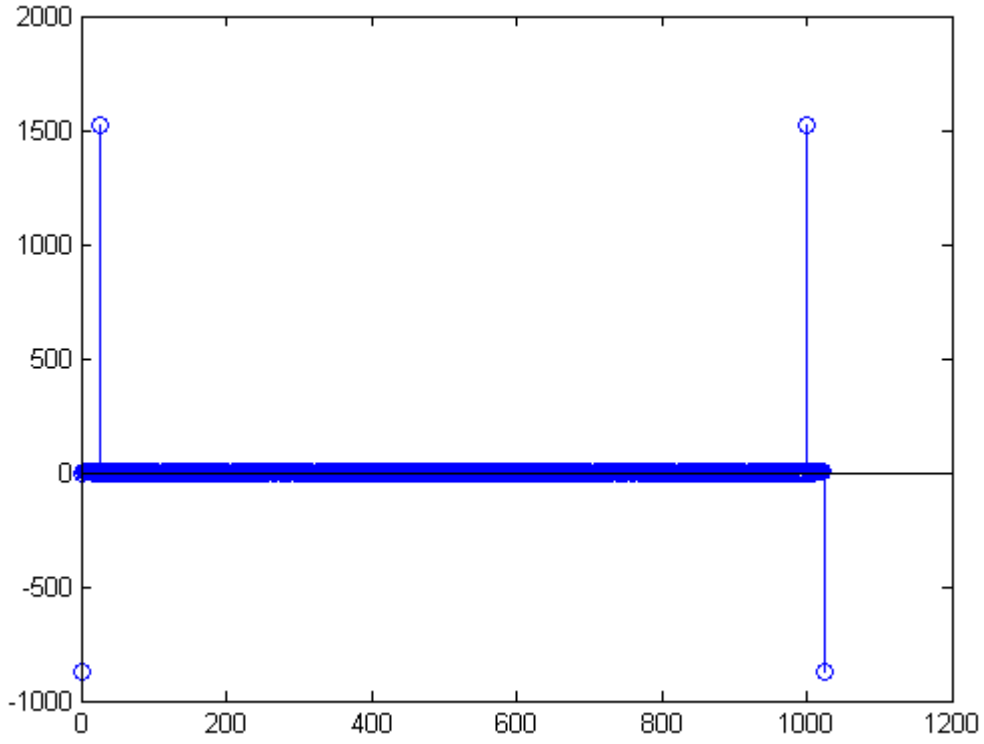

50
0
-50
-100
-150
-200
-250
-300
-350
-400
-450
0
200
400
600
800
1000
1200

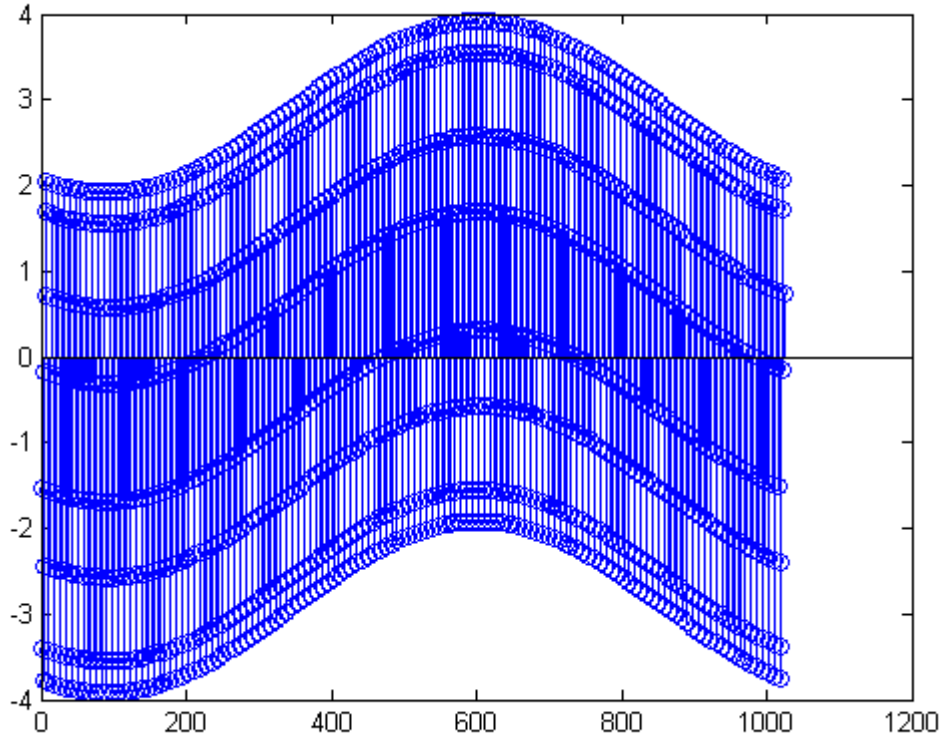
Z3


2500
2000
1500
1000
500
0
-500
0
200
400
600
800
1000
1200

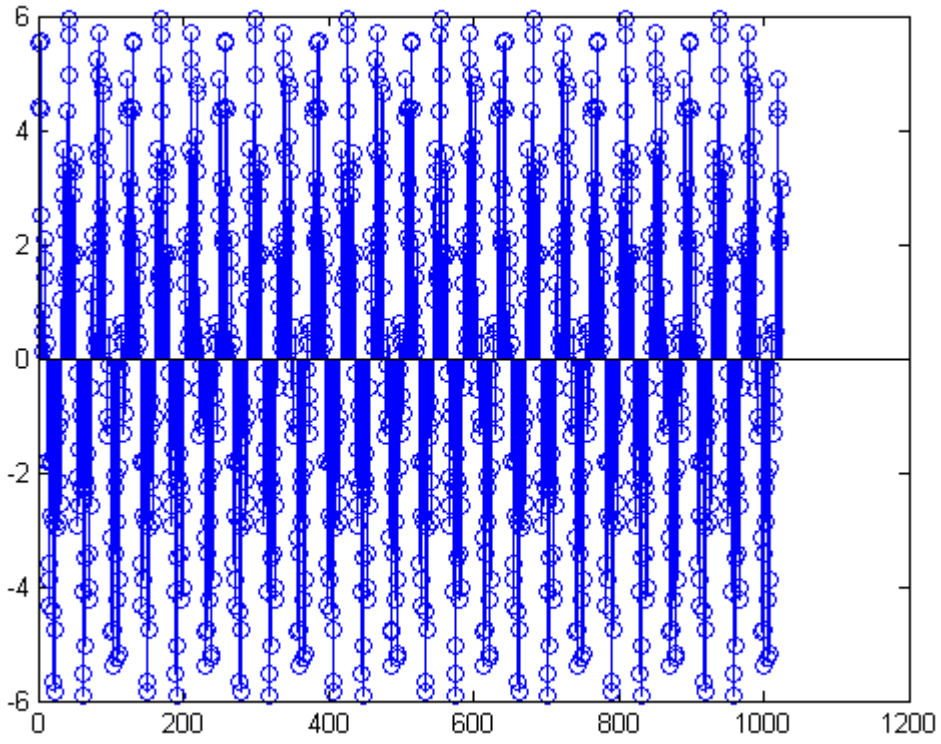
Z2


2000
1500
1000
500
0
-500
-1000
0
200
400
600
800
1000
1200

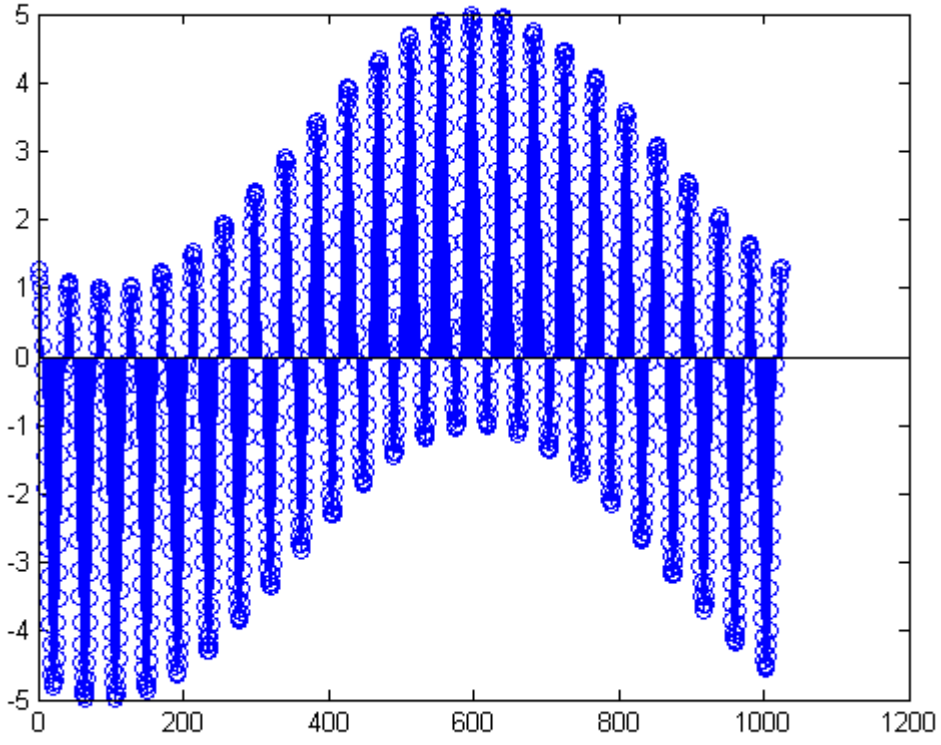
Z1


4
3
2
1
0
-1
-2
-3
-4
0
200
400
600
800
1000
1200

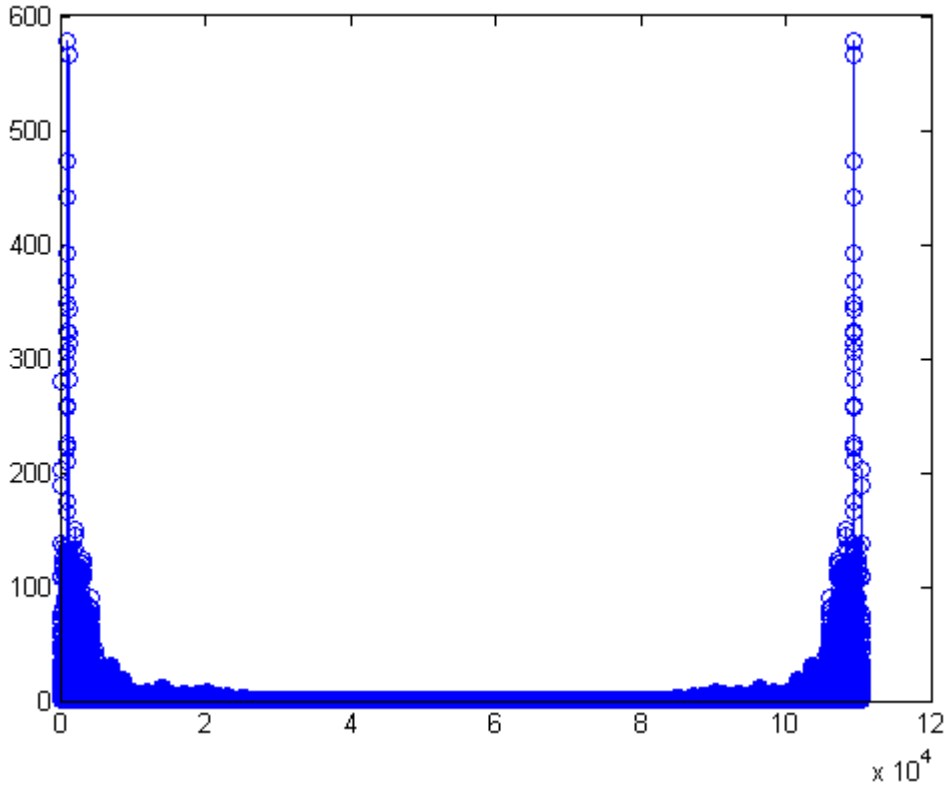
Y3


6
4
2
0
-2
-4
-6
0
200
400
600
800
1000
1200

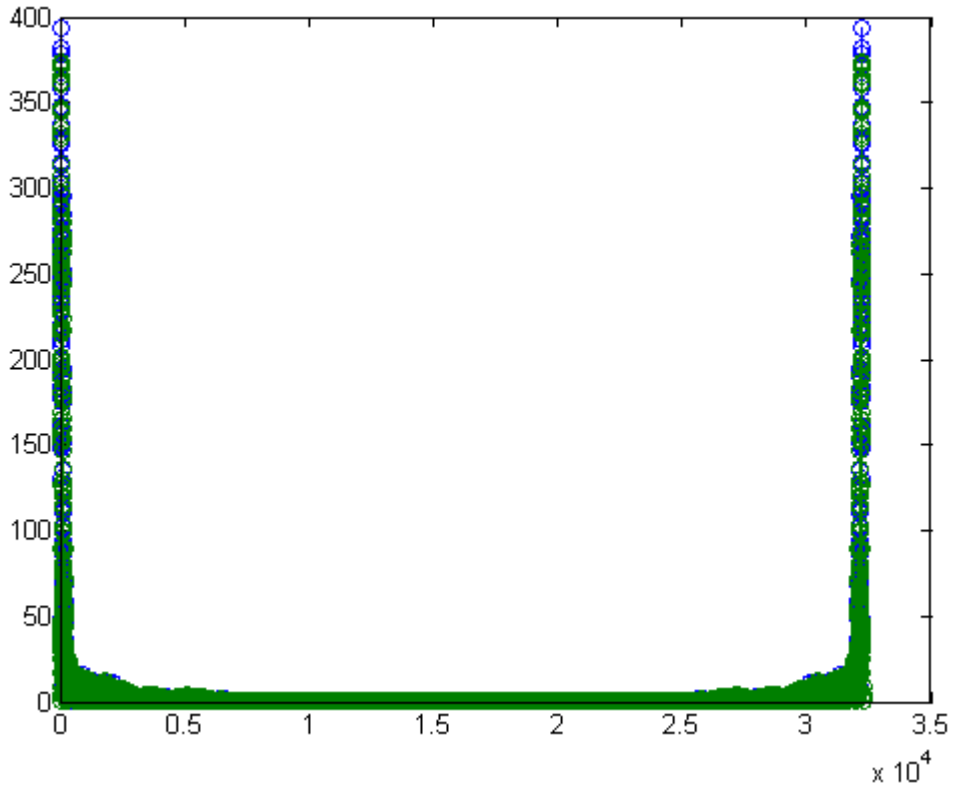
Y2


5
4
3
2
1
0
-1
-2
-3
-4
-5
0
200
400
600
800
1000
1200

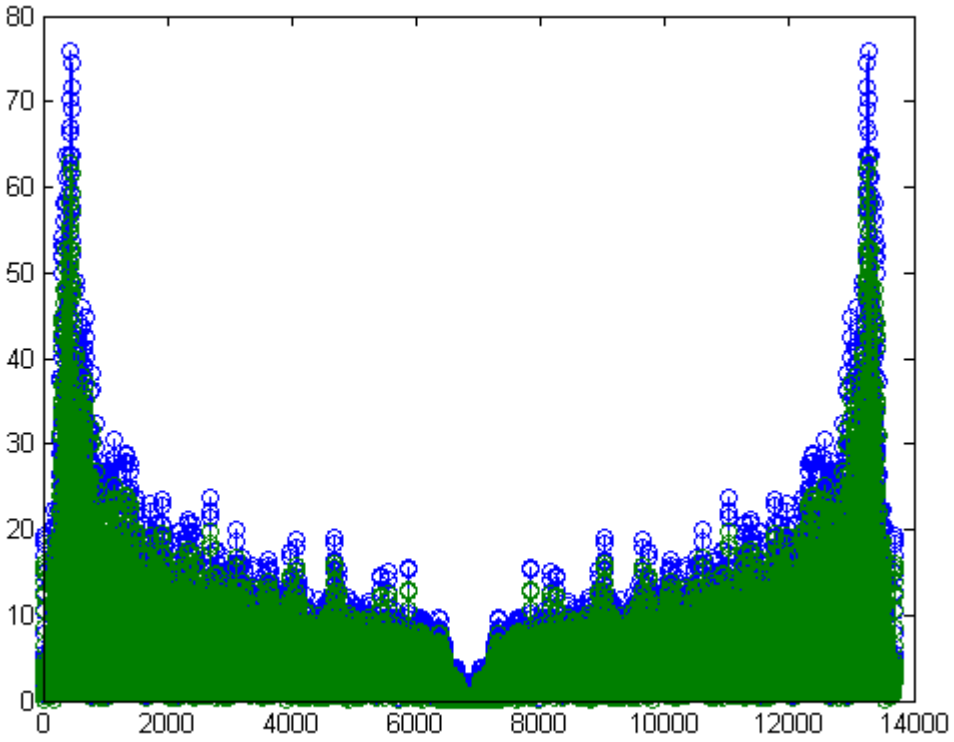
Y1


600
500
400
300
200
100
0
0
2
4
6
8
10
12
x 10^4

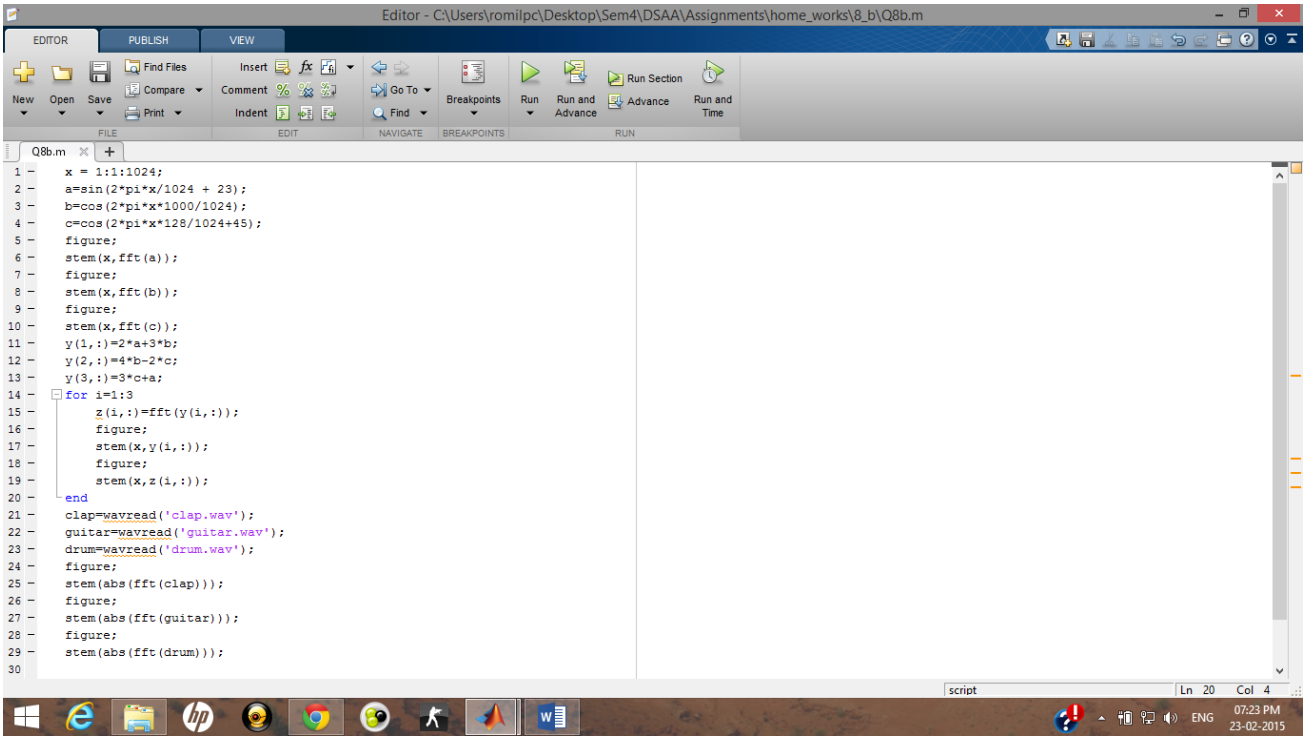
guitar


400
350
300
250
200
150
100
50
0
0
0.5
1
1.5
2
2.5
3
3.5
x 10^4

drum


80
70
60
50
40
30
20
10
0
0
2000
4000
6000
8000
10000
12000
14000

clap


Editor - C:\Users\romilpc\Desktop\Sem4\DSAA\Assignments\home_works\8_b\Q8b.m
EDITOR
PUBLISH
VIEW
New
Open
Save
Find Files
Compare
Print
Insert
Comment
Indent
Go To
Find
Breakpoints
Run
Run and Advance
Run Section
Advance
Run and Time
FILE
EDIT
NAVIGATE
BREAKPOINTS
RUN
Q8b.m
x = 1:1:1024;
a=sin(2*pi*x/1024 + 23);
b=cos(2*pi*x*1000/1024);
c=cos(2*pi*x*128/1024+45);
figure;
stem(x,fft(a));
figure;
stem(x,fft(b));
figure;
stem(x,fft(c));
y(1,:)=2*a+3*b;
y(2,:)=4*b-2*c;
y(3,:)=3*c+a;
for i=1:3
z(i,:)=fft(y(i,:));
figure;
stem(x,y(i,:));
figure;
stem(x,z(i,:));
end
clap=wavread('clap.wav');
guitar=wavread('guitar.wav');
drum=wavread('drum.wav');
figure;
stem(abs(fft(clap)));
figure;
stem(abs(fft(guitar)));
figure;
stem(abs(fft(drum)));
script
Ln 20
Col 4
ENG
07:23 PM
23-02-2015

Code

a) Zi is the fft for the corresponding Yi.

```
[     a=sin(2*pi*x/1024 + 23);
      b=cos(2*pi*x*1000/1024);
      c=cos(2*pi*x*128/1024+45);
      y(1,:)=2*a+3*b;
      y(2,:)=4*b-2*c;
      y(3,:)=3*c+a;
]
```

The plots of the absolute values of the Fast Fourier Transform thus result in the various frequencies with the values of the coefficient being their multipliers.

b) The plots are thus explained by the fact there is an aggregation of simple frequencies which make up the sounds data.