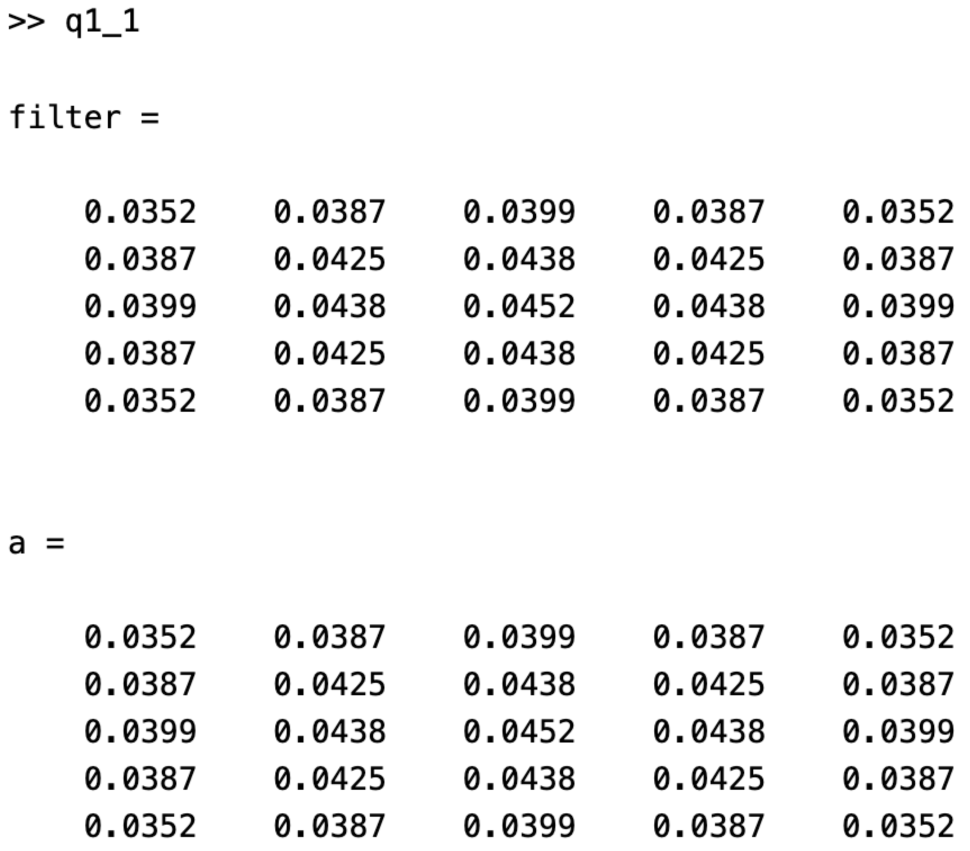
Praffullitt Jain
20171142
Vineeth Gandhi
DSAA

Assignment 2

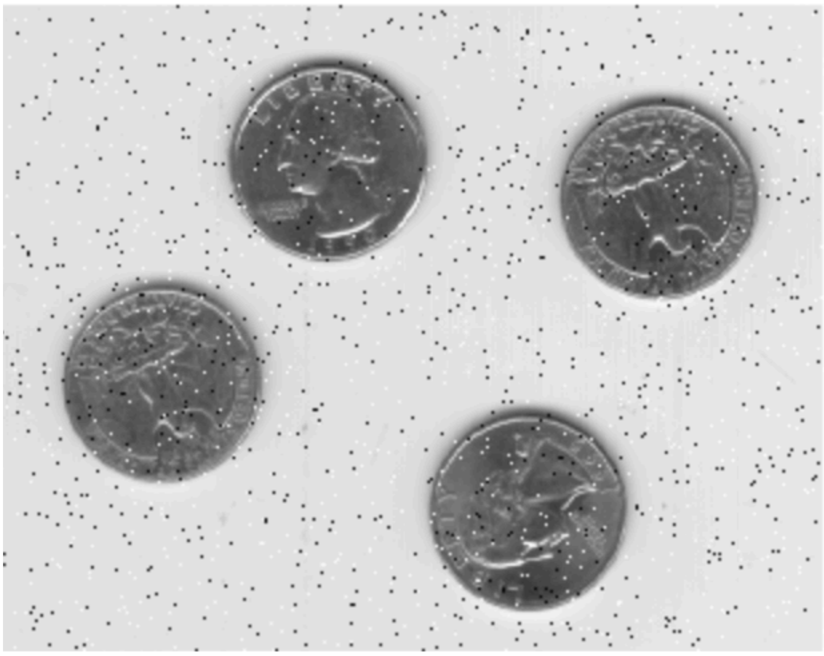
Problem 1

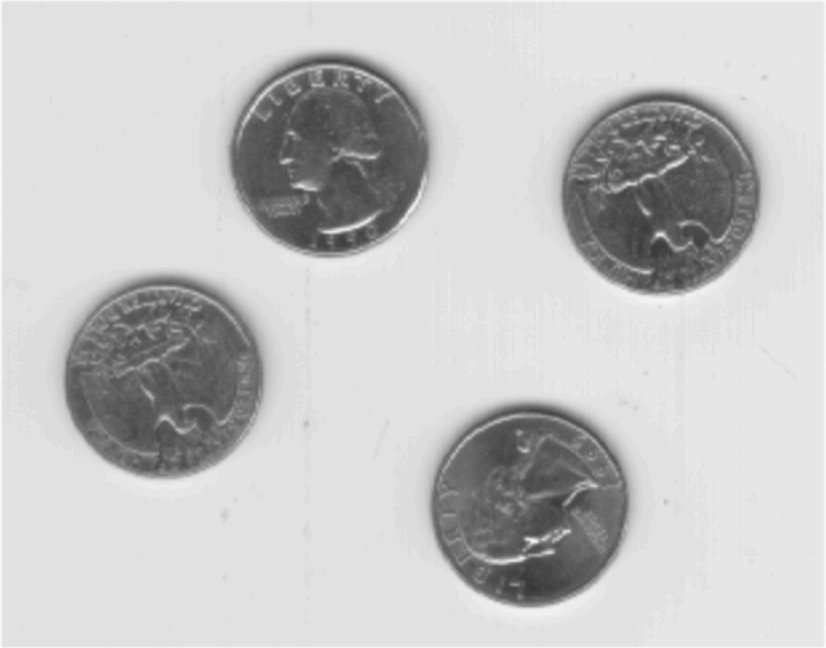
1.



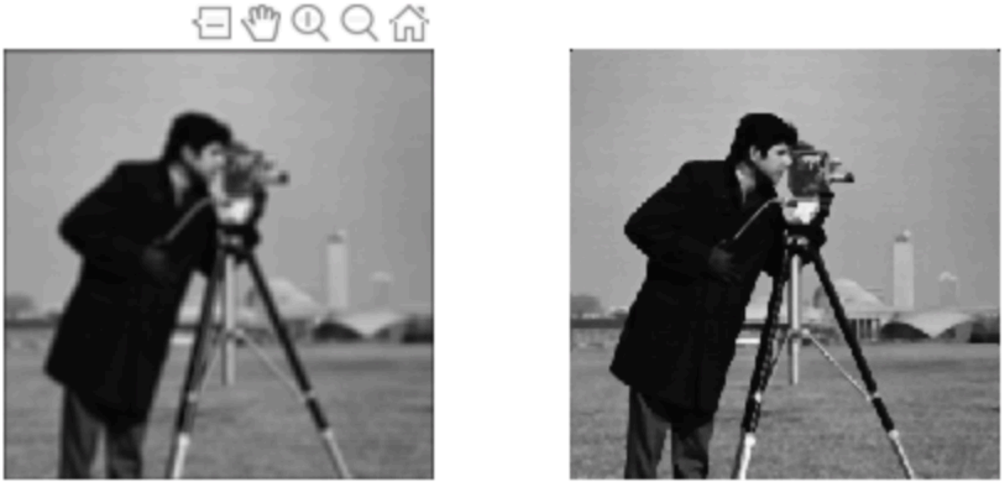
```
>> q1_1

filter =

    0.0352    0.0387    0.0399    0.0387    0.0352
    0.0387    0.0425    0.0438    0.0425    0.0387
    0.0399    0.0438    0.0452    0.0438    0.0399
    0.0387    0.0425    0.0438    0.0425    0.0387
    0.0352    0.0387    0.0399    0.0387    0.0352


a =

    0.0352    0.0387    0.0399    0.0387    0.0352
    0.0387    0.0425    0.0438    0.0425    0.0387
    0.0399    0.0438    0.0452    0.0438    0.0399
    0.0387    0.0425    0.0438    0.0425    0.0387
    0.0352    0.0387    0.0399    0.0387    0.0352
```

2.

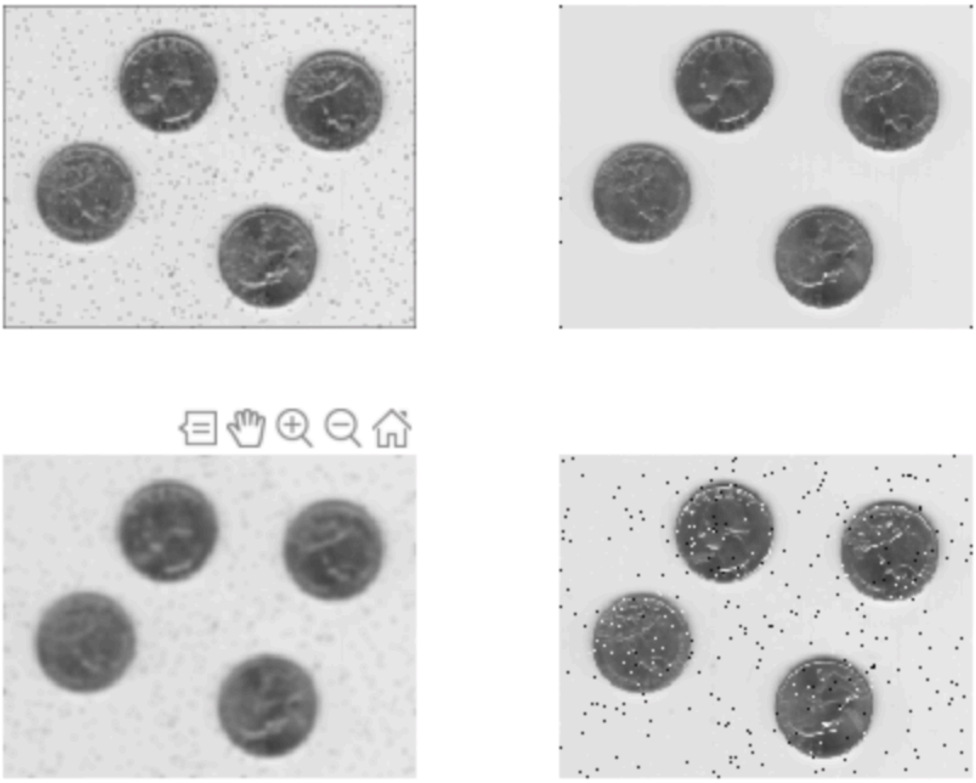




3.

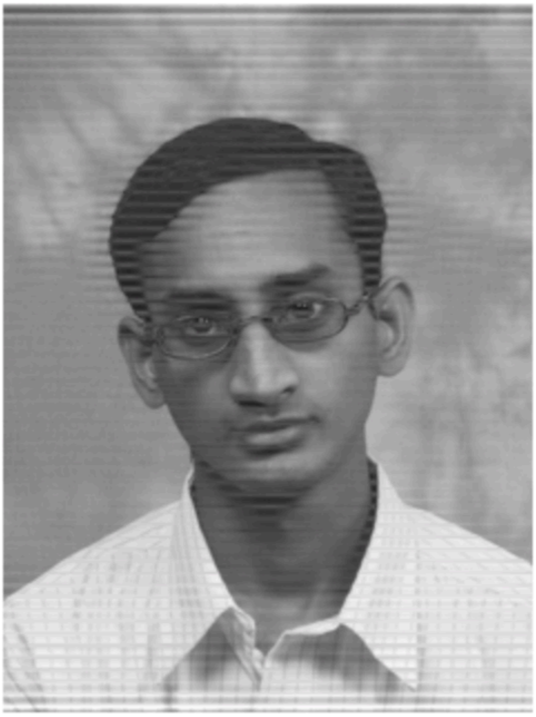


4.



The median filter is the best choice for ‘salt and pepper’ noise.

5.



Problem 2

PART A:

ceil((Width+2*Z -F+1)/S) * ceil((Height+2*Z -F+1)/S) * C

PART B:

F*F*C point-wise multiplications and F*F*C point-wise additions are done for obtaining a single value. Overall values obtained will be equal to the dimensions of the output of the convolution (same as above).

Total additions =
F*F*C* ceil((Width+2*Z -F+1)/S) * ceil((Height+2*Z -F+1)/S) * C.

Total multiplications =
F*F*C* ceil((Width+2*Z -F+1)/S) * ceil((Height+2*Z -F+1)/S) *C.

Problem 4

I am using a function called fir1 which takes 3 arguments n , Wn and type.

Wn is the ratio of the maximum frequency or cutoff frequency to the sampling frequency of the signal.

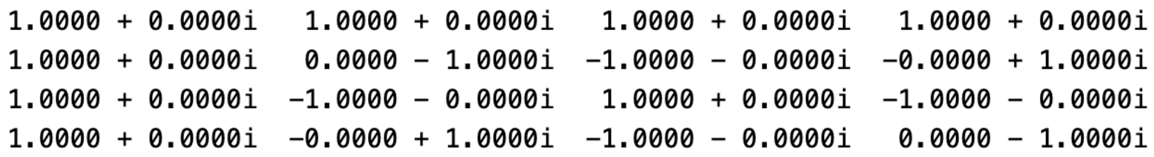
FType is lowpass filter

And I am using filter function to apply the obtained filter to the noisy signal to filter it.

Problem 5

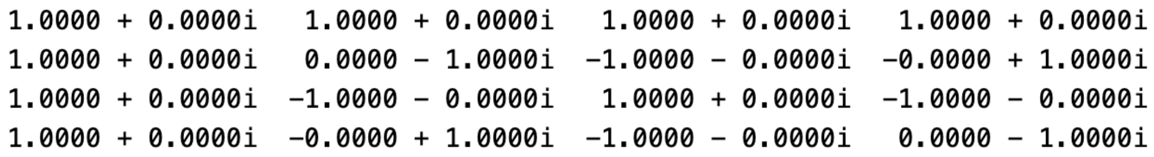
2D Discrete Fourier Transform-

- Using dftmtx-



```
 1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
 1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
 1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
 1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i   0.0000 - 1.0000i
```

- Using q5_dft-

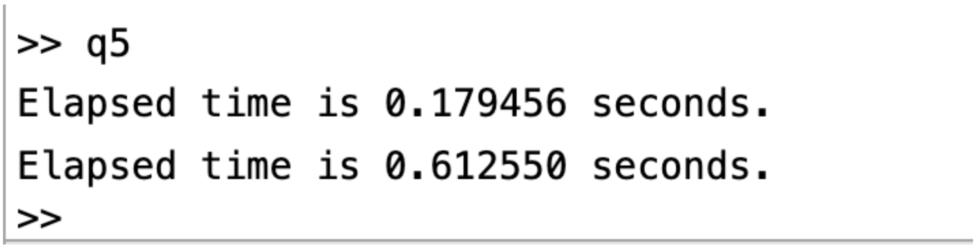


```
 1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
 1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
 1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
 1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i   0.0000 - 1.0000i
```

2D Fast Fourier Transform-

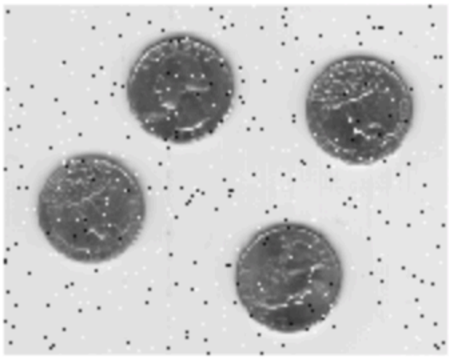
- Using fft2
- Using q5_fft-

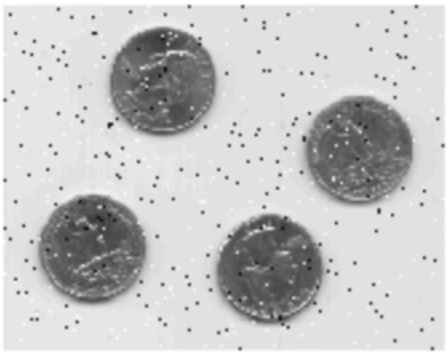
Implemented using recursion



```
>> q5
Elapsed time is 0.179456 seconds.
Elapsed time is 0.612550 seconds.
>>
```

Problem 6





img = imread('inp1.png');

// I(mxn)

x = fft2(img);

// x = Wm * I * Wn

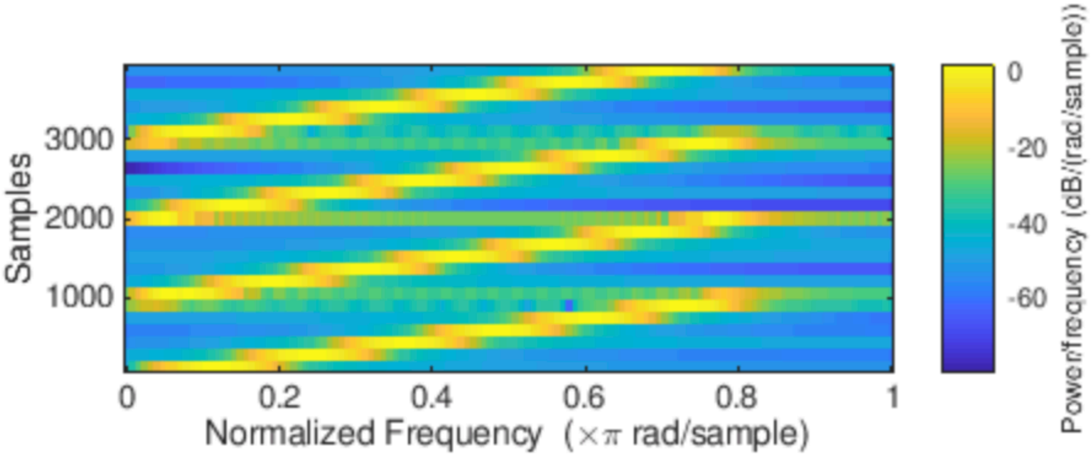
x1 = fft2(x);

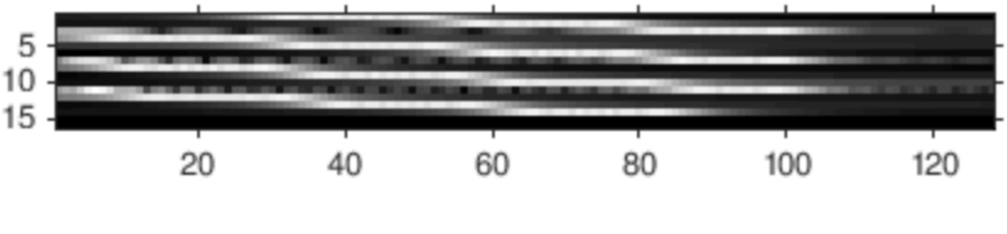
// x1 = Wm * (Wm * I * Wn) * Wn

The final image is flipped along both axes.

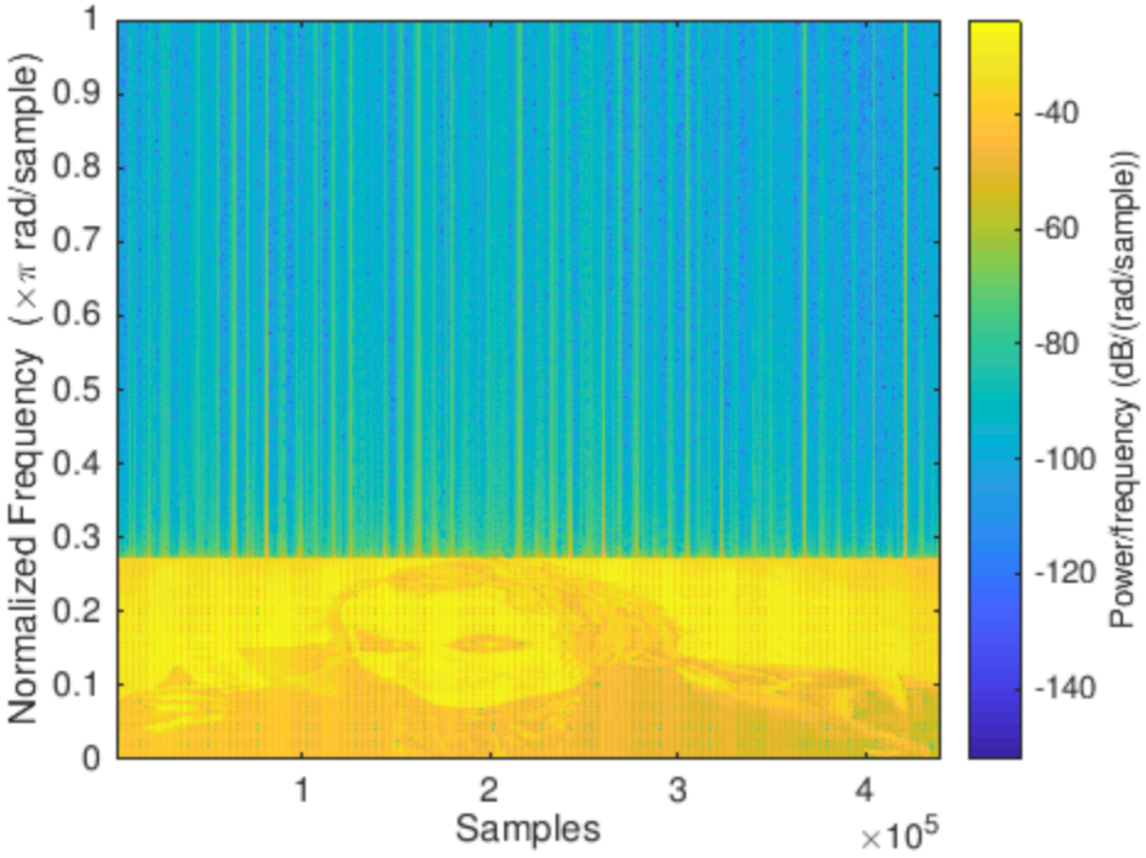
Problem 7

1.



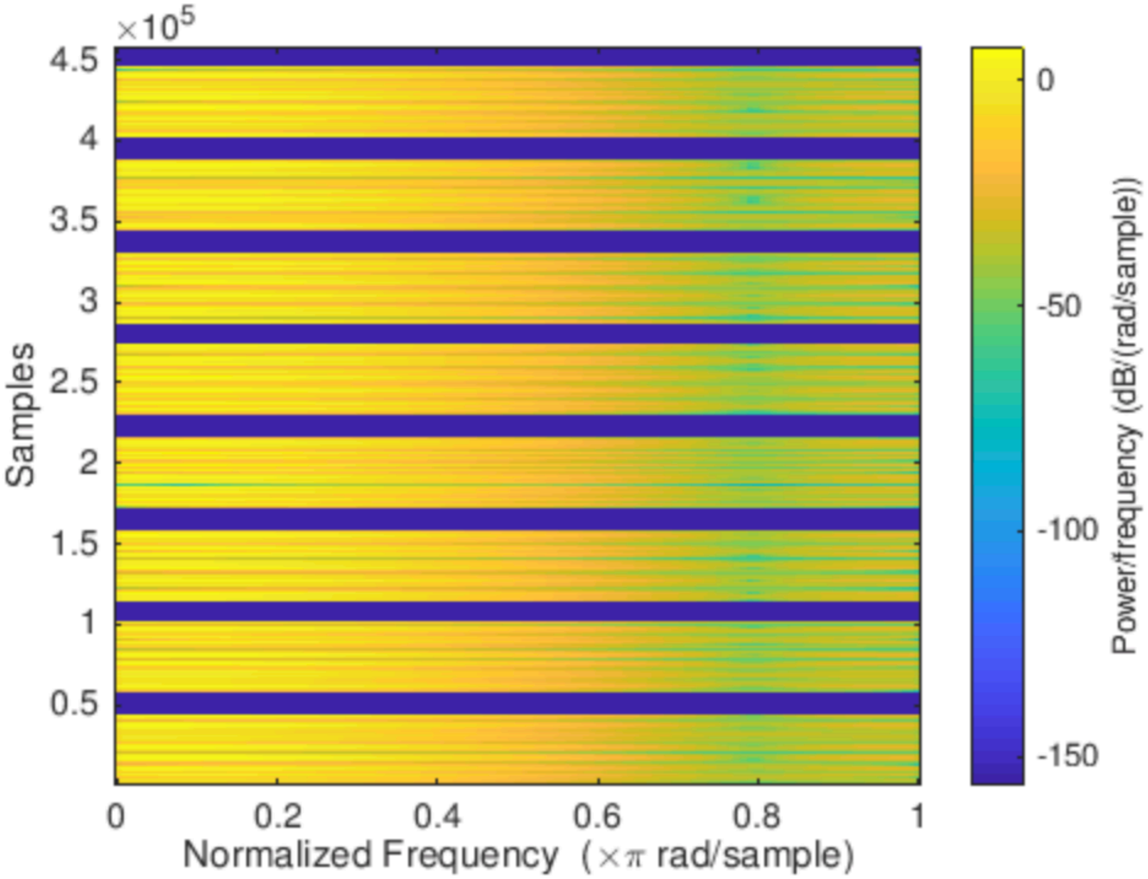


2.



Normalized Frequency (×π rad/sample)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
Samples
×10^5
-40
-60
-80
-100
-120
-140
Power/frequency (dB/(rad/sample))

3.


×10^5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
Samples
0
0.2
0.4
0.6
0.8
1
Normalized Frequency (×π rad/sample)
0
-50
-100
-150
Power/frequency (dB/(rad/sample))