Table of Contents


```
%Ryan Plante
%DSP Homework 2
%14 February 2018
```

Question 1

```
%See paper
```

Question 2

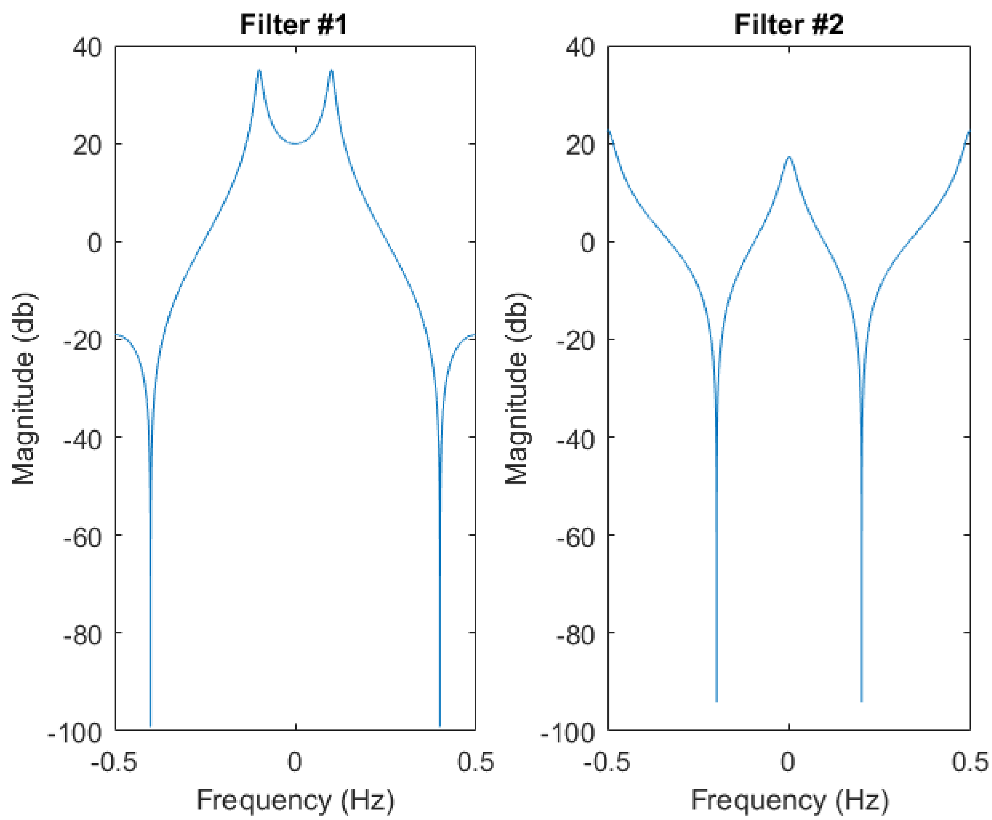
```
%See paper
```

Question 3

```
%Filter 1

%Declare Difference Eq Coefficients
a1 = [1 -1.5371 0.9025];
b1 = [1 1.6180 1];

%frequency array, ~10000 point resolution
f = -0.5:.0001:0.5;

z = exp(i*2*pi*f); %input

%Evaluate H(z) = Y(z)/X(z) at all points of Z
hz1 = polyval(b1, z) ./polyval(a1, z);
hz1db = 20*log10(hz1); %convert from magnitude to db
subplot(1,2,1);
plot(f, hz1db)
xlabel('Frequency (Hz)');
ylabel('Magnitude (db)')
title('Filter #1');
```

```
%Filter 2
a2 = [1 0 -0.81];
b2 = [1 -0.618 1];

hz2 = polyval(b2, z) ./polyval(a2, z);
hz2db = 20*log10(hz2);
subplot(1,2,2);
plot(f, hz2db)
xlabel('Frequency (Hz)');
ylabel('Magnitude (db)')
title('Filter #2');
```

Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored

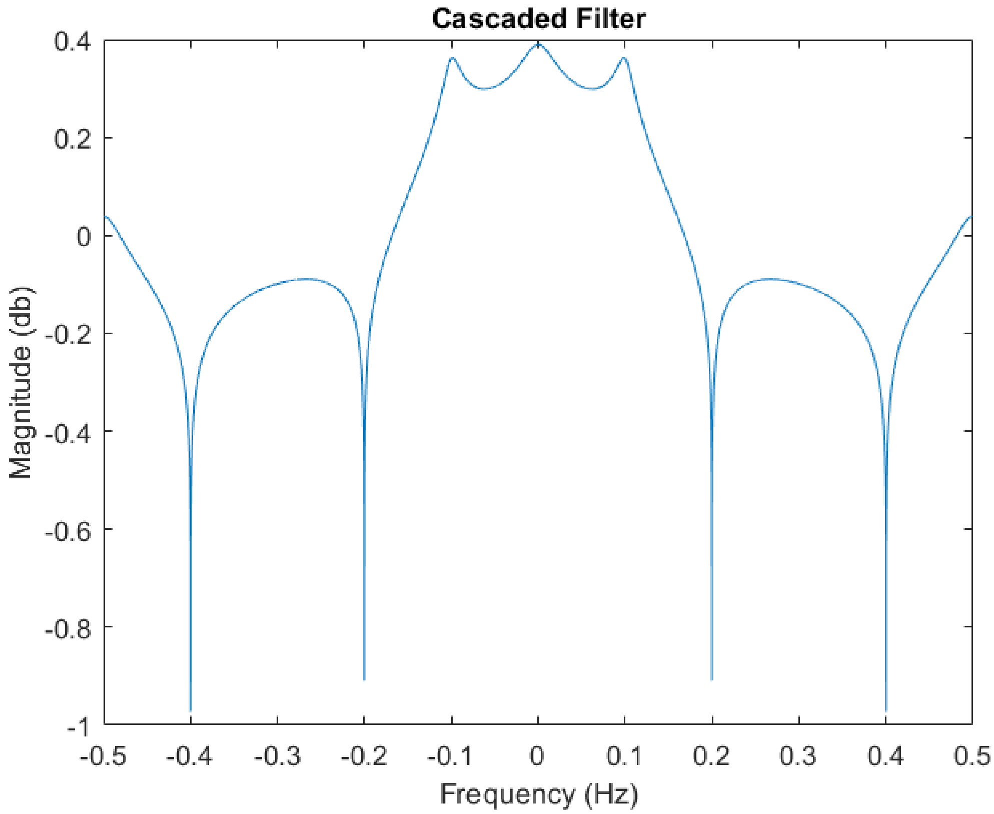


Question 4

```
%a
casc =(hz1db+hz2db); %cascade the two filters together
gain = 1/abs(max(casc)) %find value needed for max 1db gain

%b
cascadeWithGain = gain*casc; %apply our gain
figure(2)
plot(f,cascadeWithGain)
xlabel('Frequency (Hz)');
```

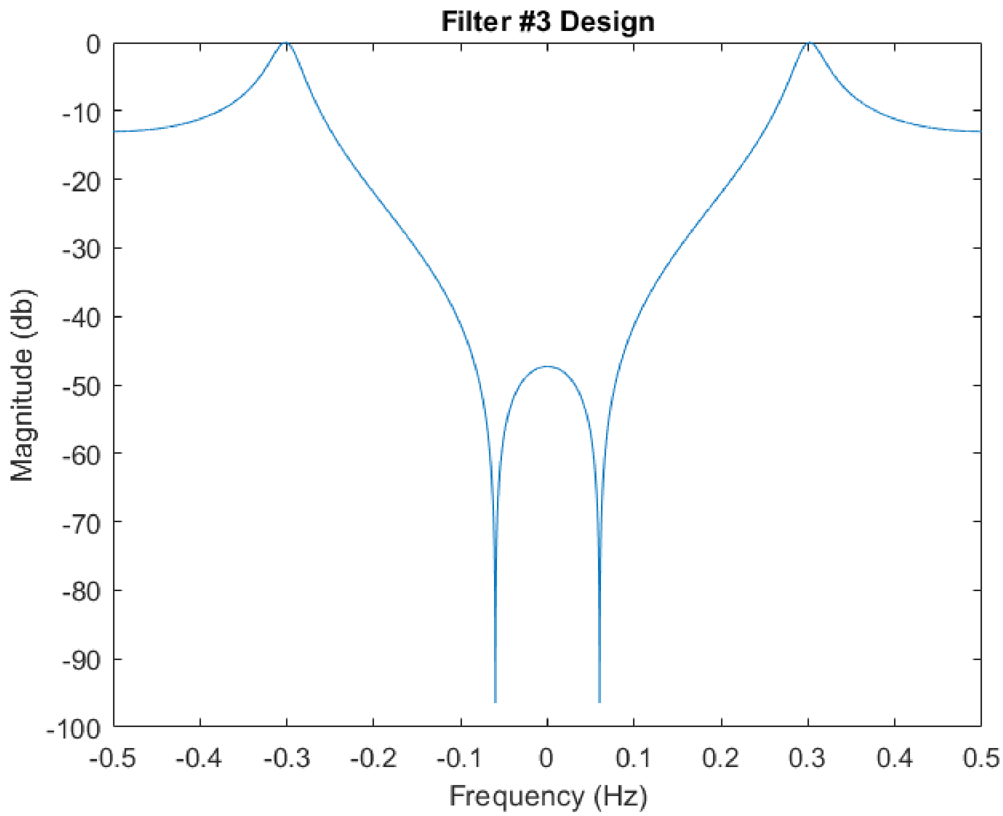
```
ylabel('Magnitude (db)');
title('Cascaded Filter');
%c
x = [0 0 0 0 0 0 0 0 0 3 1.5 -1.8 0.2 0 0 0 0 0 0 0 0]; %input array of
 20 values
stage1 = filter(b1, a1, x*gain) %output of stage one
stage2 = filter(b2, a2, stage1) %output of stage two
```

```
gain =

    0.0105

Warning: Imaginary parts of complex X and/or Y arguments ignored

stage1 =

  Columns 1 through 7

         0         0         0         0         0         0         0

  Columns 8 through 14

         0    0.0314    0.1150    0.1863    0.1700    0.0776   -0.0320

  Columns 15 through 20

   -0.1192   -0.1544   -0.1297   -0.0600    0.0248    0.0923


stage2 =

  Columns 1 through 7

         0         0         0         0         0         0         0

  Columns 8 through 14

         0    0.0314    0.0955    0.1722    0.2471    0.2984    0.2902

  Columns 15 through 20

    0.2198    0.1223    0.0245   -0.0352   -0.0479   -0.0116
```



Question 5

```
%See paper for derivation of coefficients
a = [1 -1.8*cos(2*pi*(0.3)) 0.81];
b = [1 -2*cos(2*pi*(0.06)) 1];

%Evaluate H(z) = Y(z)/X(z) at all points of Z
hz3 = polyval(b, z) ./ polyval(a, z);
gain2 = 1./abs(max(hz3)) %gain of 0 db
hz3 = 20*log10(hz3*gain2); %convert to db, add in gain
figure(3)
plot(f, hz3)
xlabel('Frequency (Hz)');
ylabel('Magnitude (db)');
title('Filter #3 Design');


gain2 =

    0.0726

Warning: Imaginary parts of complex X and/or Y arguments ignored
```



Question 6

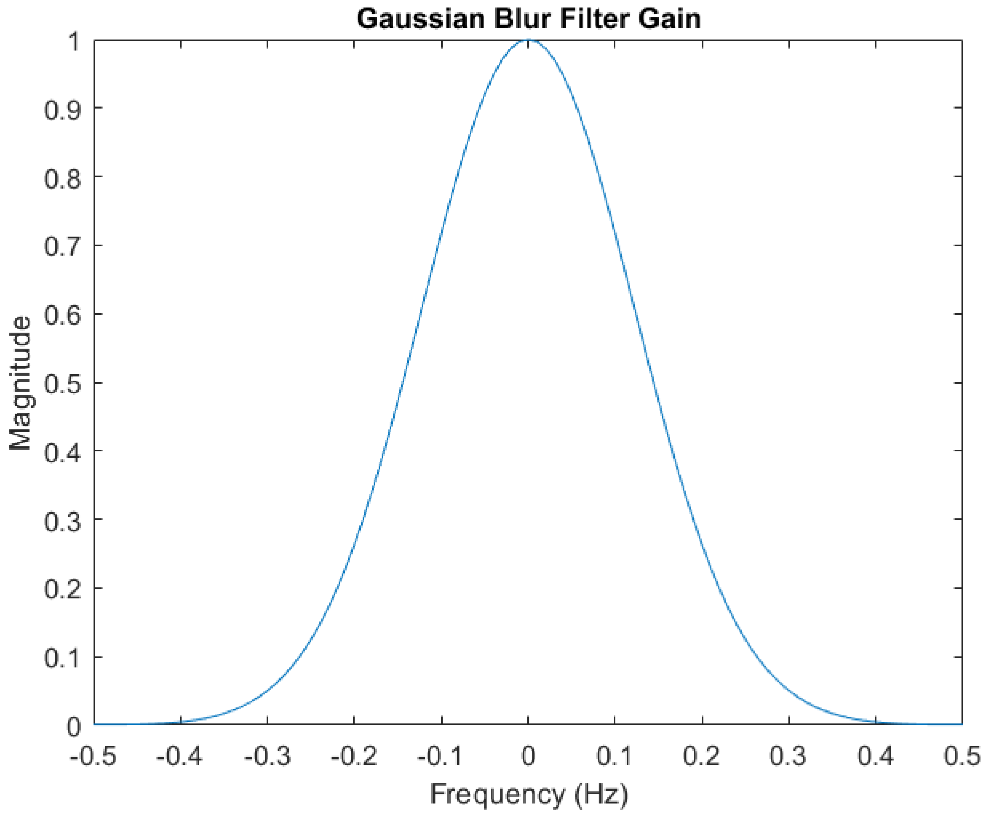
```
clear
```

A

```
M = 4;
sigma = 1.3;
G = 1; %placeholder value

%ii
%compute sum of impulse response to find DC gain
sum = 0;
for n = -M:M
  sum = sum + exp(-0.5*(n/sigma)^2);
end

G = 1/sum %gain coefficient for DC gain of 1

%iii
f= (-0.5:0.01:0.5); %frequency, 100 samples
window = (-M:1:M); %window in which filter is non-zero

%define our filter symbolically
syms z freq n
```

```
h(n) = piecewise((n >= -M) & (n <= M), G*exp(-0.5*(n/sigma)^2),0);
H(z) = 0*z;
for i = -M:M
  H(z) = H(z) + h(i)*(z^-i);
end

%substitute our exponential in for z
H(freq) = subs(H(z), z, exp(1j*2*pi*freq));
%plot our filter gain
figure(4)
plot(f, abs(double(H(f))))
xlabel('Frequency (Hz)');
ylabel('Magnitude')
title('Gaussian Blur Filter Gain');

%iv
% Load the image, and convert to doubles for processing.
p1 = double(imread('Cavvy_bw.jpg'));

%calculate number of rows in image array
rows = size(p1(:,1));

%for all rows in image create new image processed through gaussian
 blur
%filter
for i= 1:rows
  p2(i,:) = conv(p1(i,:), double(h(window)));
end
figure(5)
%orignal image
imshow(p1,[0 255])
title('Original Image');

figure(6)
%gaussian blur processed image
imshow(p2,[0 255])
title('Gaussian Blur Filtered Image');
```

```
G =

    0.3070
```



Original Image

Gaussian Blur Filtered Image

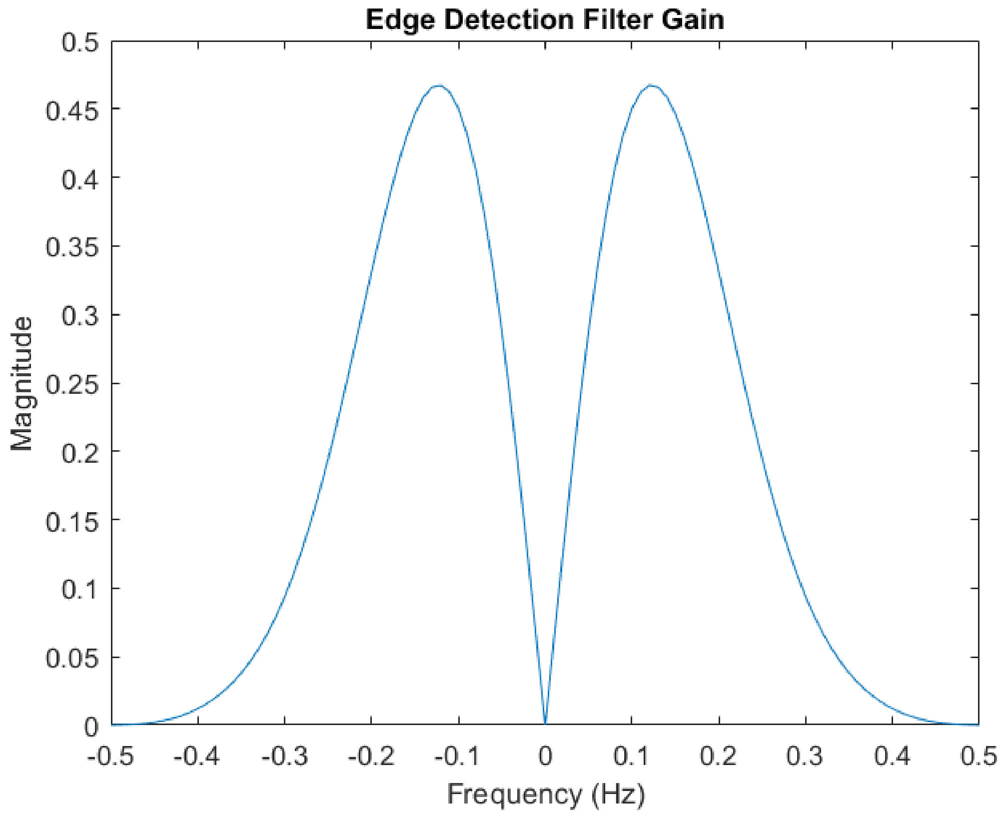
B

```
%define our edge detection filter
h2(n) = piecewise((n >= -M) & (n <= M), -G*(n/(sigma^2))*exp(-0.5*((n/
sigma)^2)),0);

%i
%once again find our filter gain
sum = 0;
for i = -M:M
  sum = sum + (i/(sigma^2))*exp(-0.5*((i/sigma)^2));
end
%create new gain coefficient
G = -1/sum

H(z) = 0*z;
for i= -M:M
  H(z) = H(z) + h2(i)*(z^-i);
end

%substitue our exponential in for z
H(freq) = subs(H(z), z, exp(1j*2*pi*freq));

%plot filter gain vs frequency
figure(7)
plot(f, abs(double(H(f))))
```

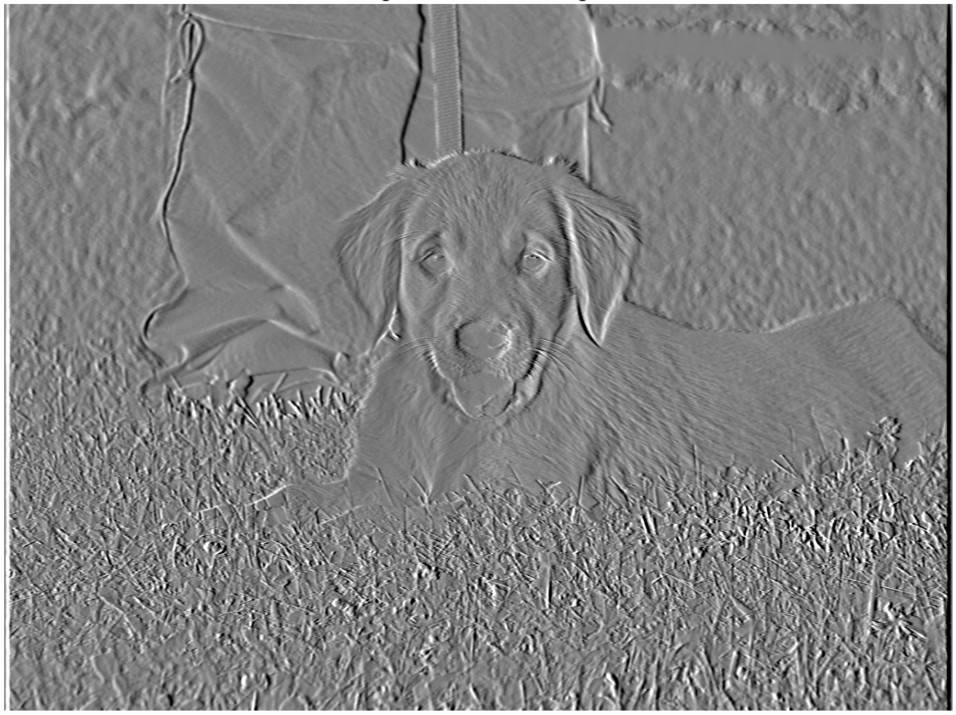
```
xlabel('Frequency (Hz)');
ylabel('Magnitude')
title('Edge Detection Filter Gain');

%ii
%for all rows in image create new image processed through edge
 detection
%filter
for i= 1:rows
  p3(i,:) = conv(p1(i,:), double(h2(window)));
end

figure(8)
%edge detection
imshow(p3, [-20 20])
title('Edge Detection Filtered Image');
```

G =

7.7900e+15



Edge Detection Filtered Image



Question 7

```
%See paper
```

Published with MATLAB® R2017a