Zack Boyle and Jack Landers

Lab Report 7

Step 2

- What is the lowest value of the squared error? If the error goes to close to zero, how many iterations does it take?
- After L data samples how much does the shape of the fft of the filter look like the least squares filter fft from the class lecture slides?
  - The shape closely resembles the one from the class slides
- What is the maximum value of the adaptive filter's fft magnitude? How does that compare to the least squares fft?
  - All are roughly in the same ballpark as the magnitude from the slides except for the $\mu = 0.1$ example which is much higher in magnitude.

μ	Lowest Mean Squared Error Value	Num Iterations to Get Near Zero	Max Magnitude of FFT
0.00001	0.093	N/A	2.3e-5
0.0001	0.03	N/A	0.00018
0.01	0.0001	75	0.00039
0.1	0	0	7.35e80

- For a = 0.5, How do these results compare to Step 1 results? Why do you think the squared error curve is different?
    - There is much more noise in the squared error curve. This is due to the noise coefficient (a) being increased by a factor of 5.
- Repeat Step 1 with mu = 0.0001, a = 0.1, and L = 5000 instead of 1000. Compare the results to the result earlier in this step which had the same value of mu and a, but had L = 1000.
    - The graphs which did not ever reach zero (such as mu = 0.0001) actually had the time to reach 0.

Step 3

- With mu = 0.001, run your adaptive filter. In addition to the three plots used in Steps 1 and 2, also make a stem plot of the filter at the end of the loop.
- Compare the filter w with the unknown filter h. How similar is it?
    - For μ = 0.01, 0.001, and 0.005, filters w and h nearly match. However, μ = 0.08 distorted the filter beyond recognition, and μ = 0.0001 is somewhat close to h but still very distorted.
- What is the error level at the end of the adaptation loop?

μ	Error Level
0.001	0.04
0.01	0
0.005	0

0.008	246
0.0001	0.189

- Explore the importance of timing alignment. Repeat the filter with mu = 0.01, but change d to be d = x_unk(length(h):end); % make d the same length as x. Describe the output and explain.
    - The filter is roughly comparable to h, but very noisy. This is because the desired signal is truncated so there is less data to work with.

Step 4

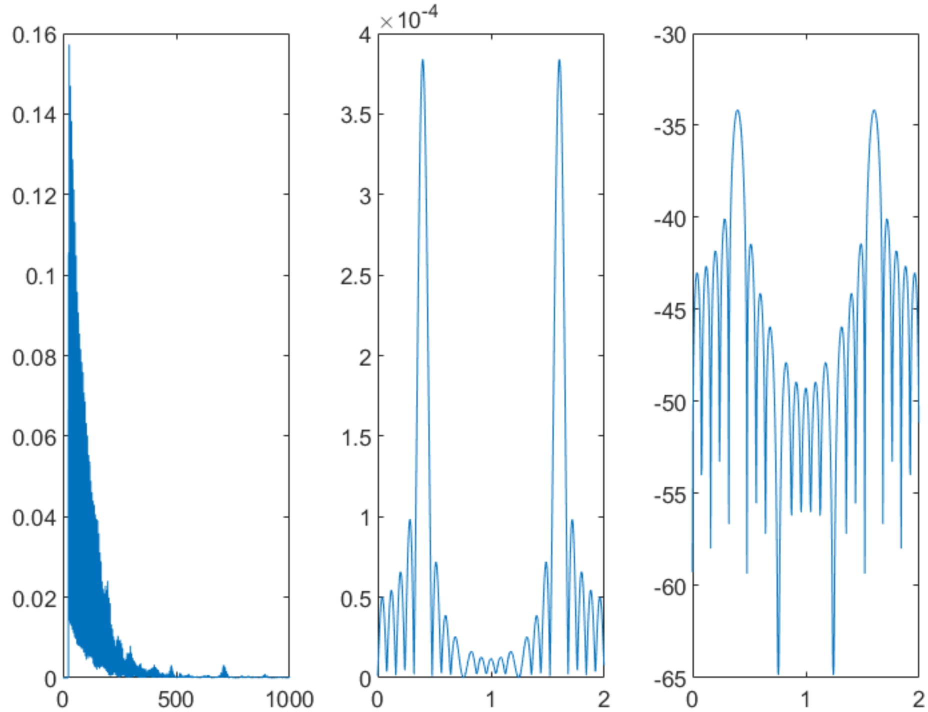
- Do ffts of x and d and compare magnitude fft plots before proceeding to make sure that your signals are correct.
    - The fft plots are as we expected.
- With mu = 0.01, run your adaptive filter as you did in Step 3. View the error plot, the ffts of w at the end of the adaptation loop, and a stem plot of w at the end of the loop. Does the error get reduced as the filter adapts? Try a few other values for mu and see if there is any improvement. Why is this filter not working well? Make a stem plot of the unknown filter b_pm and compare it to the stem plot or your filter w.
    - This filter is not working well as b_pm does not resemble w and the mean squared error remains large. This is because n_pm is much larger than M, meaning our filter cannot remember enough data to accurately match the desired filter.

- Repeat the steps above but change n_pm from 64 to 32 and use mu = 0.01. Explain why this change of n_pm made an improvement compared to the previous result. What other change might have improved the result?
    - Changing n_pm to 32 makes it much closer to M = 25, meaning that our filter will be much better at representing the mystery filter. Increasing L to give our loops more chances to lower the squared error would also help.

Part 1

```
clc
clear
close all

M = 25;
L = 1000;
a = 0.1;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);


[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.001, a = 0.1");
```

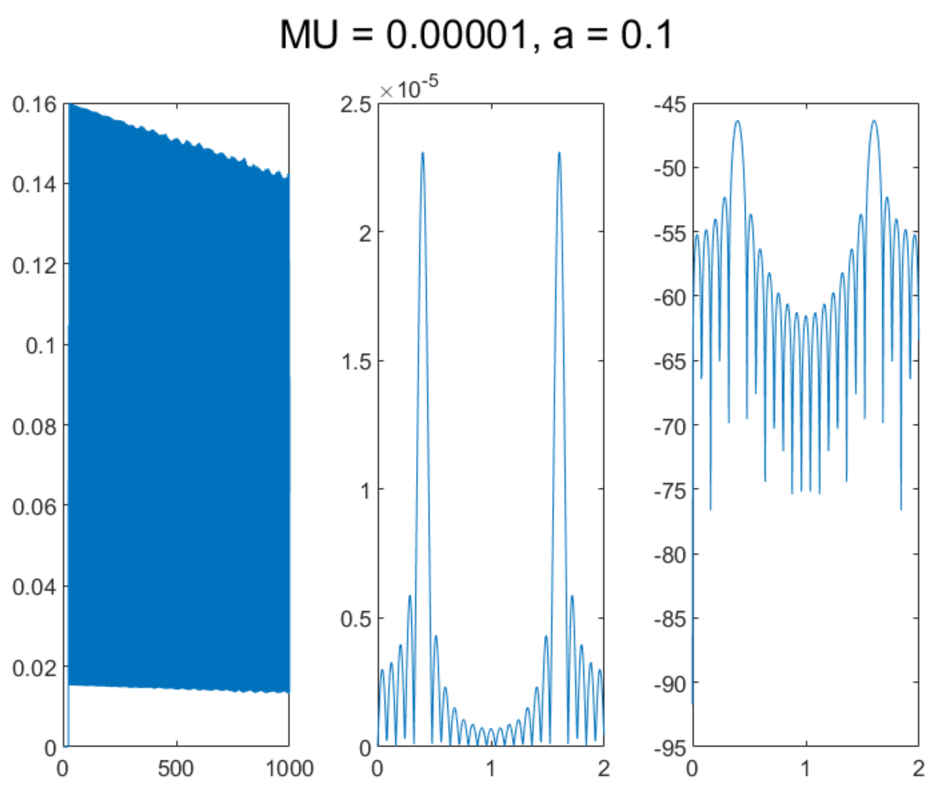
MU = 0.001, a = 0.1



Part 2

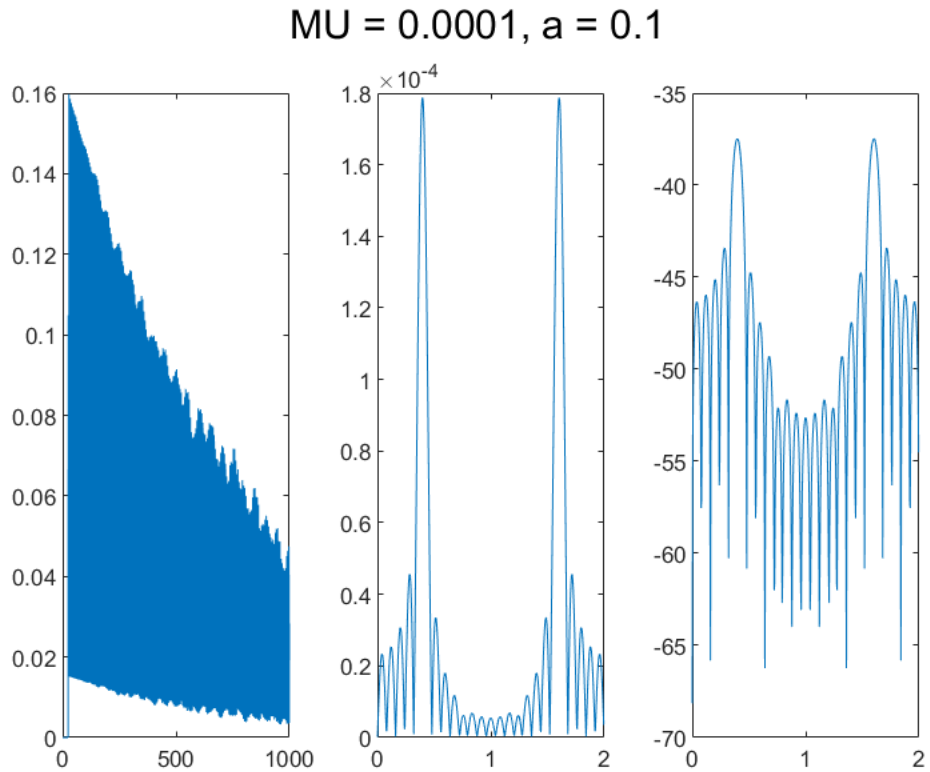
```
clc
clear
close all

%Mu = 0.00001
M = 25;
L = 1000;
a = 0.1;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.00001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
```

```
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.00001, a = 0.1");
```



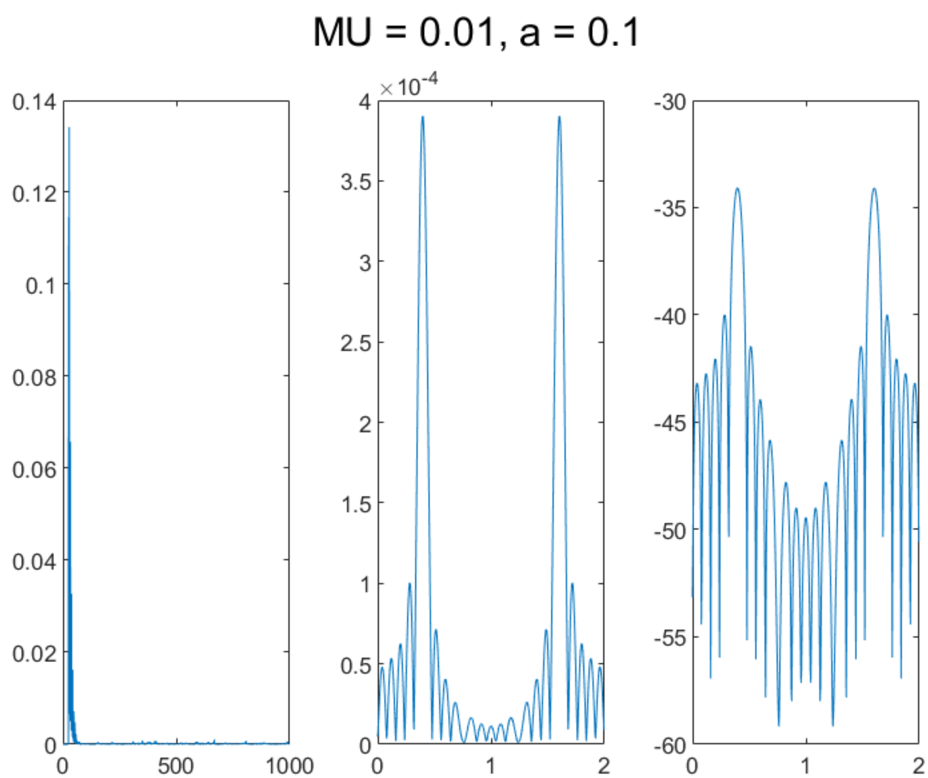

```
clc
clear
close all

%Mu = 0.0001
```

```
M = 25;
L = 1000;
a = 0.1;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.0001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.0001, a = 0.1");
```



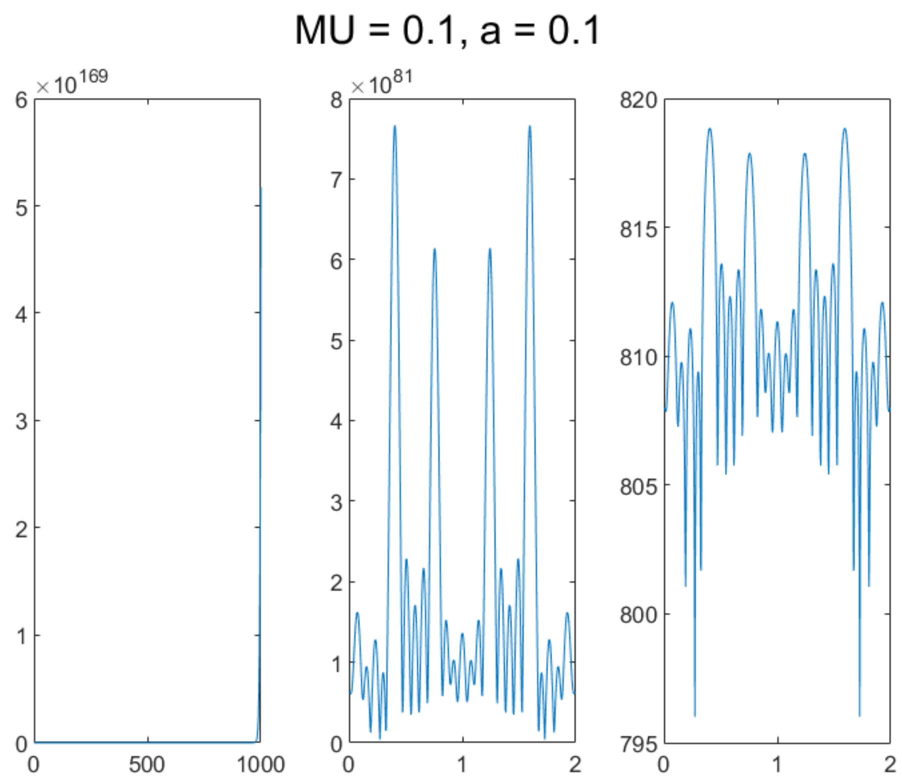

```
clc
clear
close all

%Mu = 0.01
M = 25;
L = 1000;
a = 0.1;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.01;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end
```

```
    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.01, a = 0.1");
```



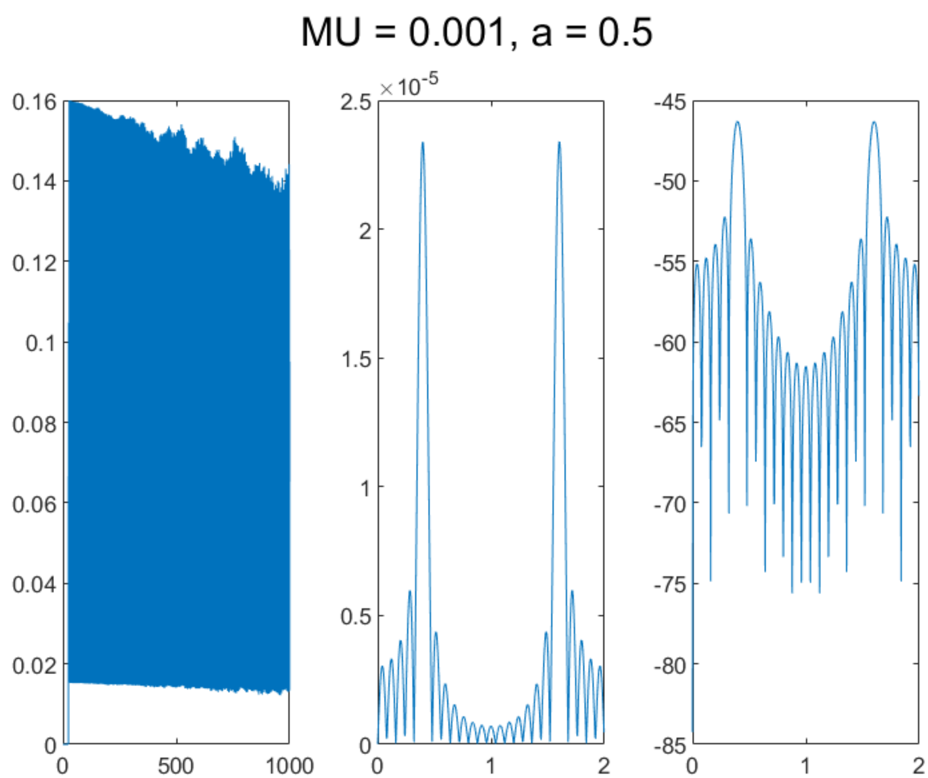

```
clc
clear
close all

%Mu = 0.1
M = 25;
L = 1000;
```

```
a = 0.1;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.1;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.1, a = 0.1");
```



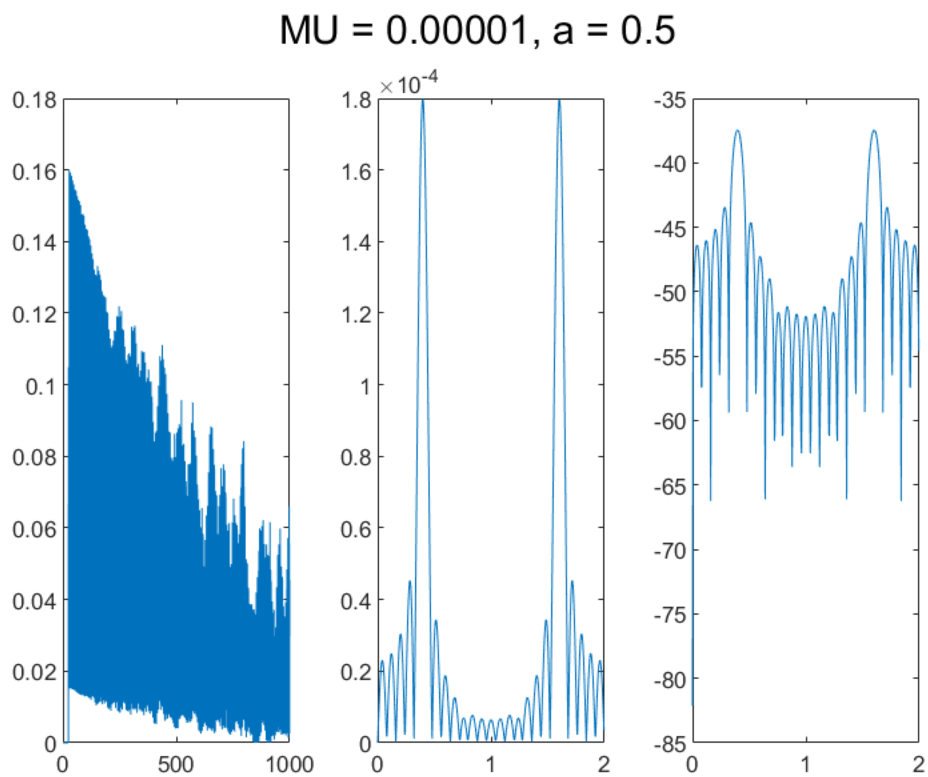

```
clc
clear
close all

%Mu = 0.00001
M = 25;
L = 1000;
a = 0.5;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.00001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end
```

```
    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.001, a = 0.5");
```



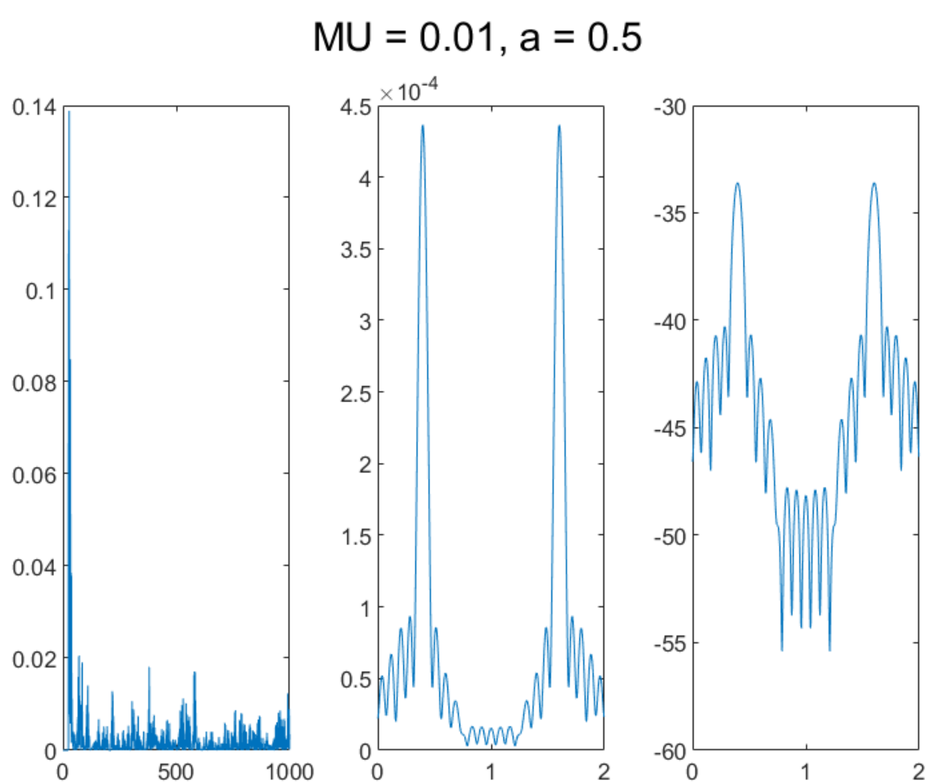

```
clc
clear
close all

%Mu = 0.0001
M = 25;
L = 1000;
```

```
a = 0.5;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.0001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.00001, a = 0.5");
```



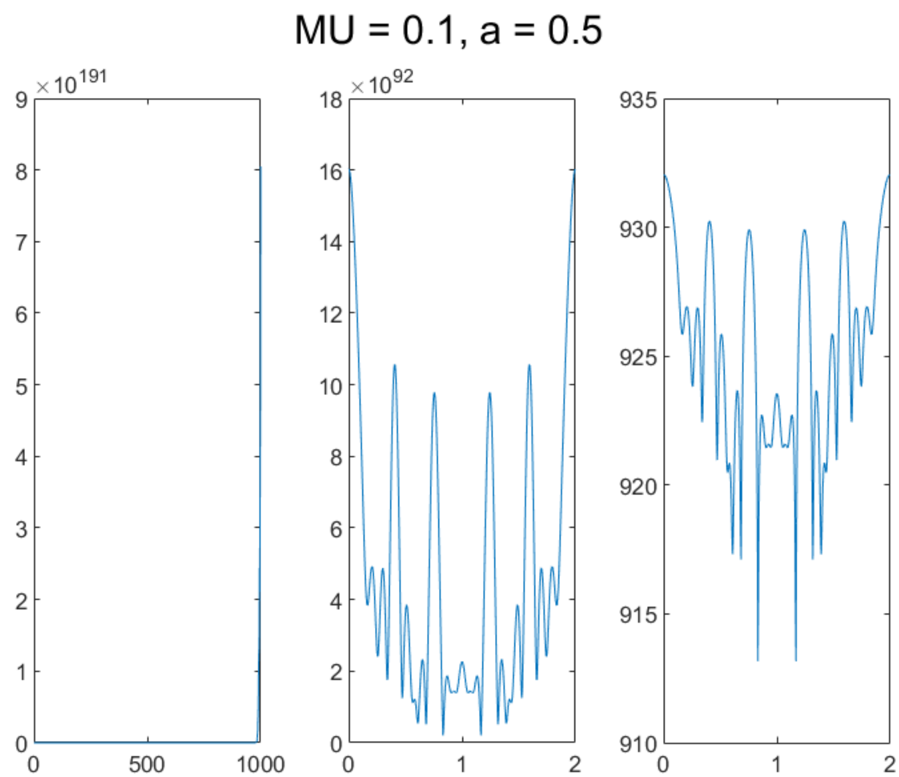

```
clc
clear
close all

%Mu = 0.01
M = 25;
L = 1000;
a = 0.5;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.01;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end
```

```
    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.01, a = 0.5");
```



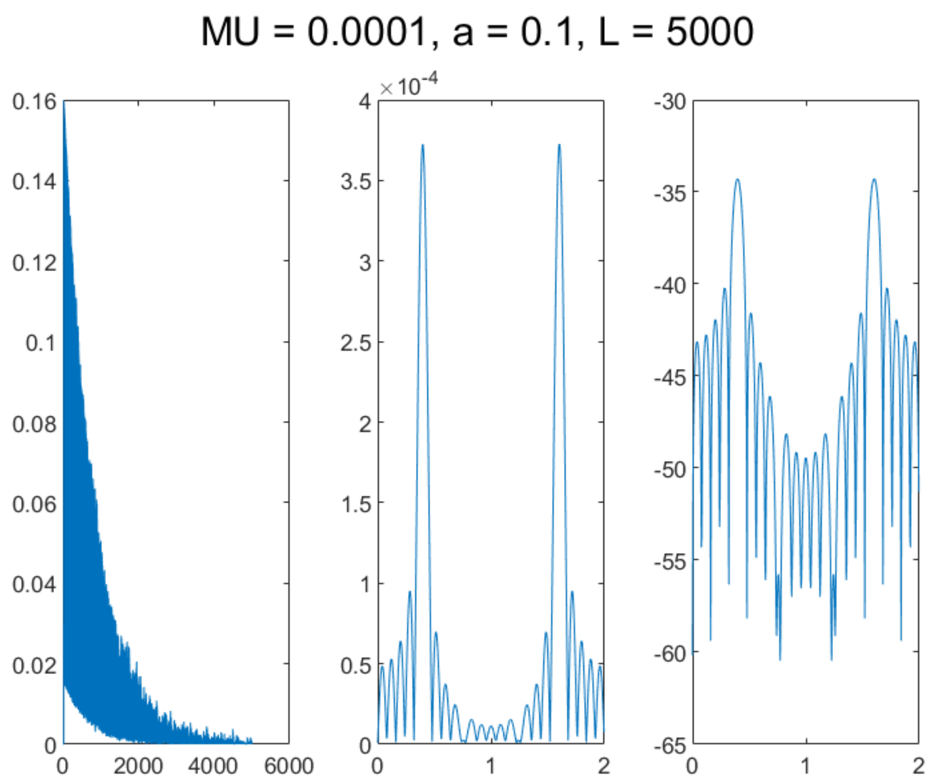

```
clc
clear
close all

%Mu = 0.1
M = 25;
L = 1000;
```

```
a = 0.5;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.1;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.1, a = 0.5");
```




```
clc
clear
close all

M = 25;
L = 5000;
a = 0.1;
x = cos(2*pi*0.2*(0:(L-1))) + cos(2*pi*0.38*(0:(L-1)))+a*randn(1,L);
d = 0.4*cos(2*pi*0.2*(0:(L-1))+pi/5);

y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.0001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);
```

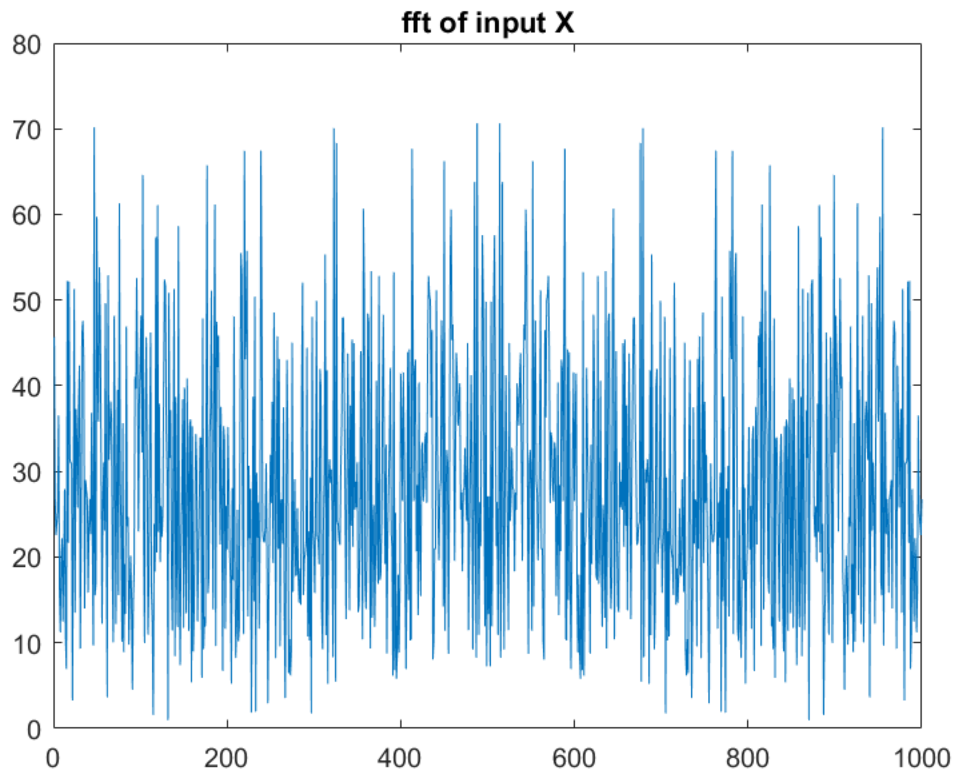
```
    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 3, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 3, 2);
plot(W/pi, abs(H));
subplot(1, 3, 3);
plot(W/pi, 10*log10(abs(H)));
sgtitle("MU = 0.0001, a = 0.1, L = 5000");
```



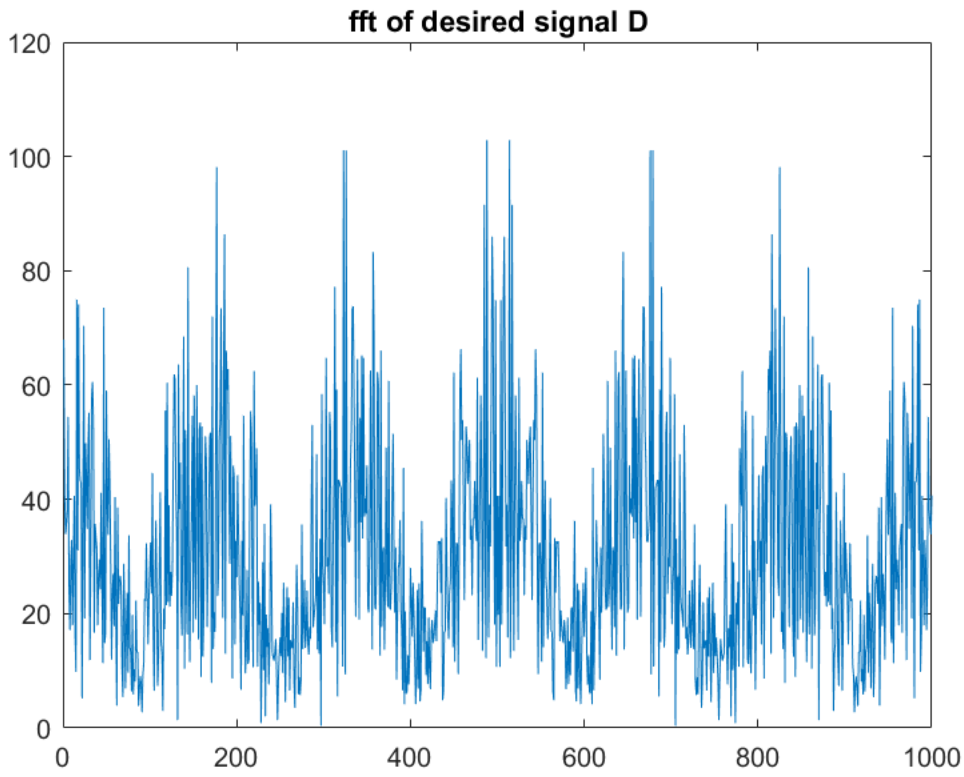
Part 3

```
M = 25;
L = 1000;
h = [1, 0, 0, 0, 0, 0, 0.5];
x = randn(1,L);
x_unk = conv(h,x);
```

```
d = x_unk(1: end- length(h)+1);

X = fft(x);
D = fft(d);


figure;
plot(abs(X));
title("fft of input X");
```

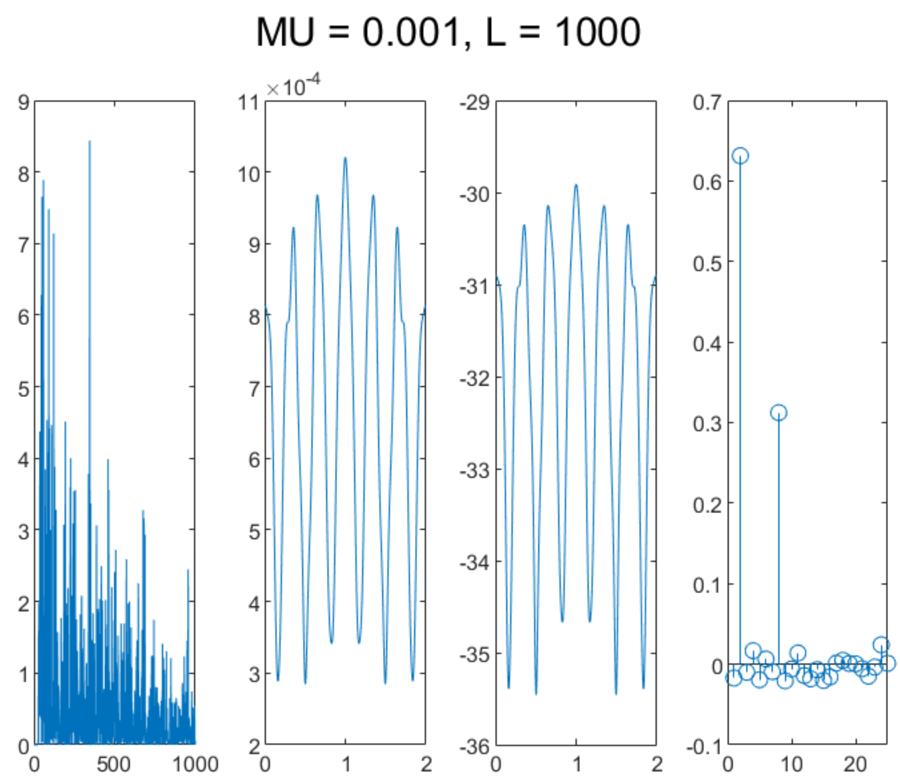



```
figure;
plot(abs(D));
title("fft of desired signal D");
```



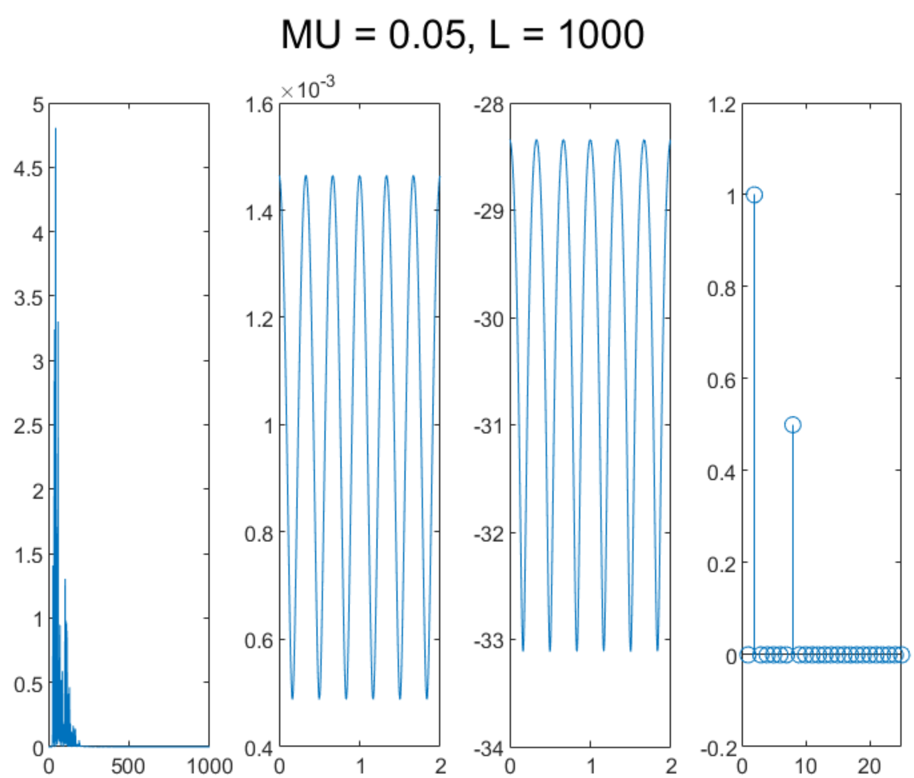

```
y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 4, 1);
```

```
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 4, 2);
plot(W/pi, abs(H));
subplot(1, 4, 3);
plot(W/pi, 10*log10(abs(H)));
subplot(1, 4, 4);
stem(w);
sgtitle("MU = 0.001, L = 1000");
```



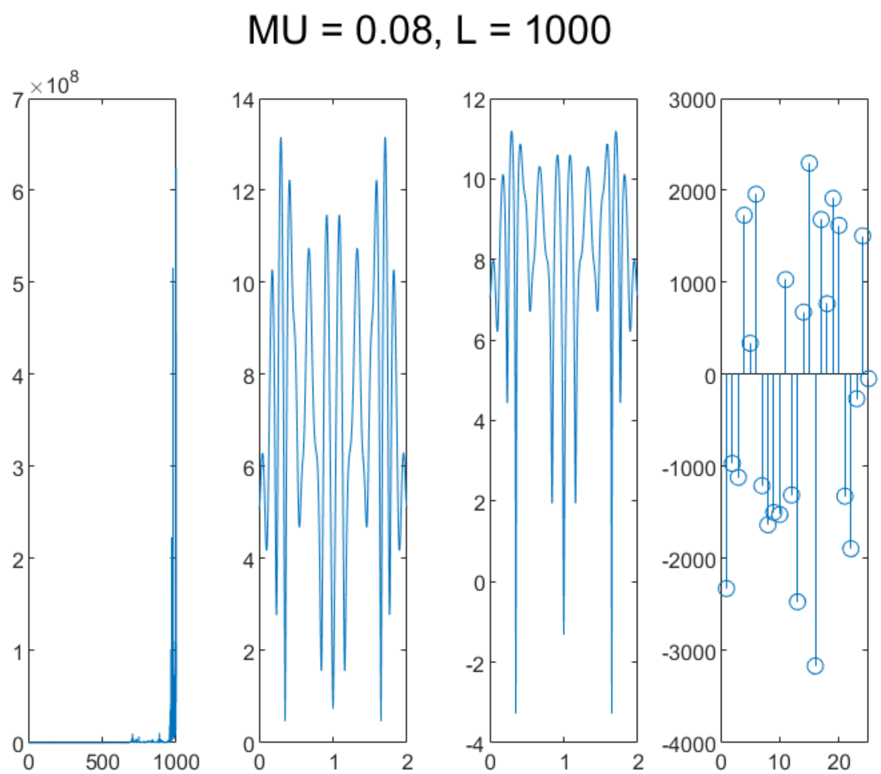

```
y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.05;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end
```

```
    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 4, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 4, 2);
plot(W/pi, abs(H));
subplot(1, 4, 3);
plot(W/pi, 10*log10(abs(H)));
subplot(1, 4, 4);
stem(w);
sgtitle("MU = 0.05, L = 1000");
```



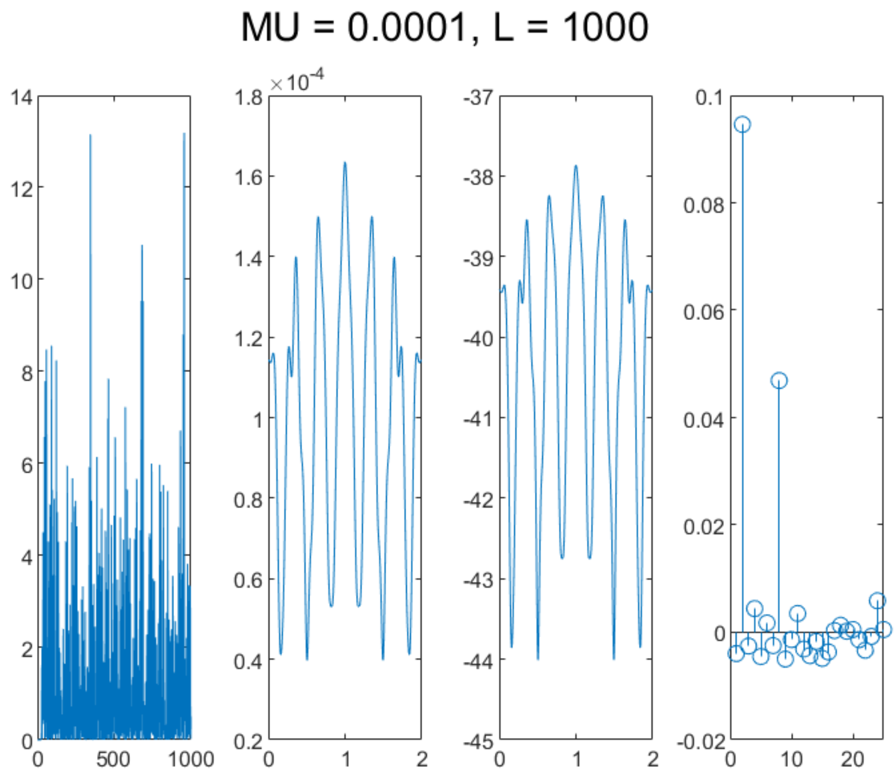

```
y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);
```

```
mu = 0.08;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 4, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 4, 2);
plot(W/pi, abs(H));
subplot(1, 4, 3);
plot(W/pi, 10*log10(abs(H)));
subplot(1, 4, 4);
stem(w);
sgtitle("MU = 0.08, L = 1000");
```

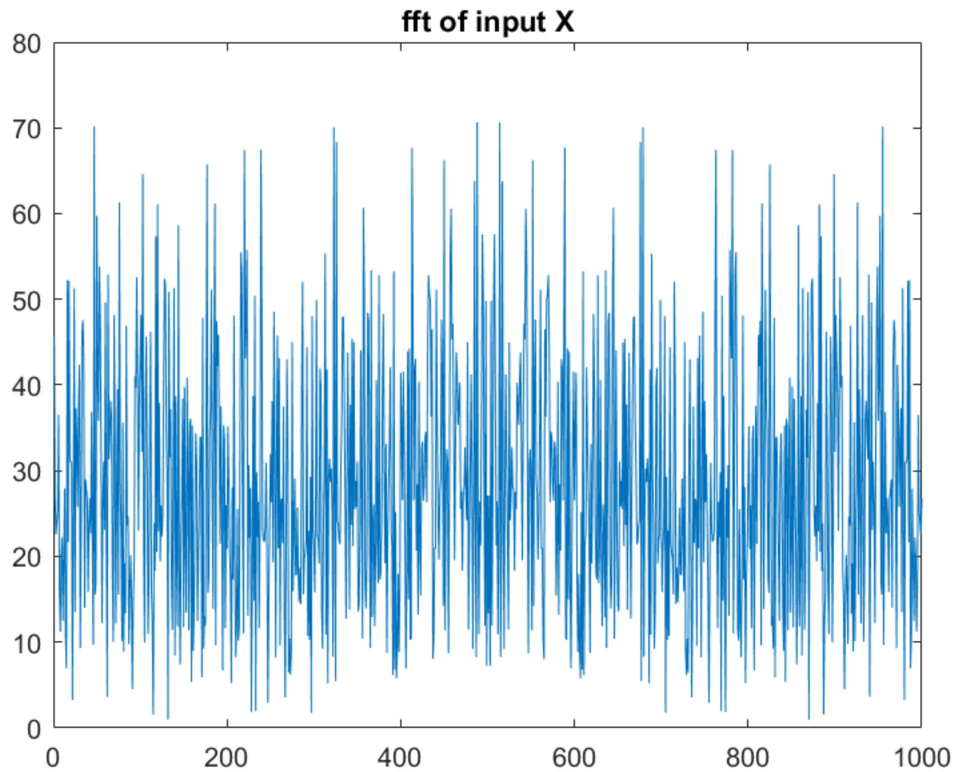



```
y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.0001;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 4, 1);
```

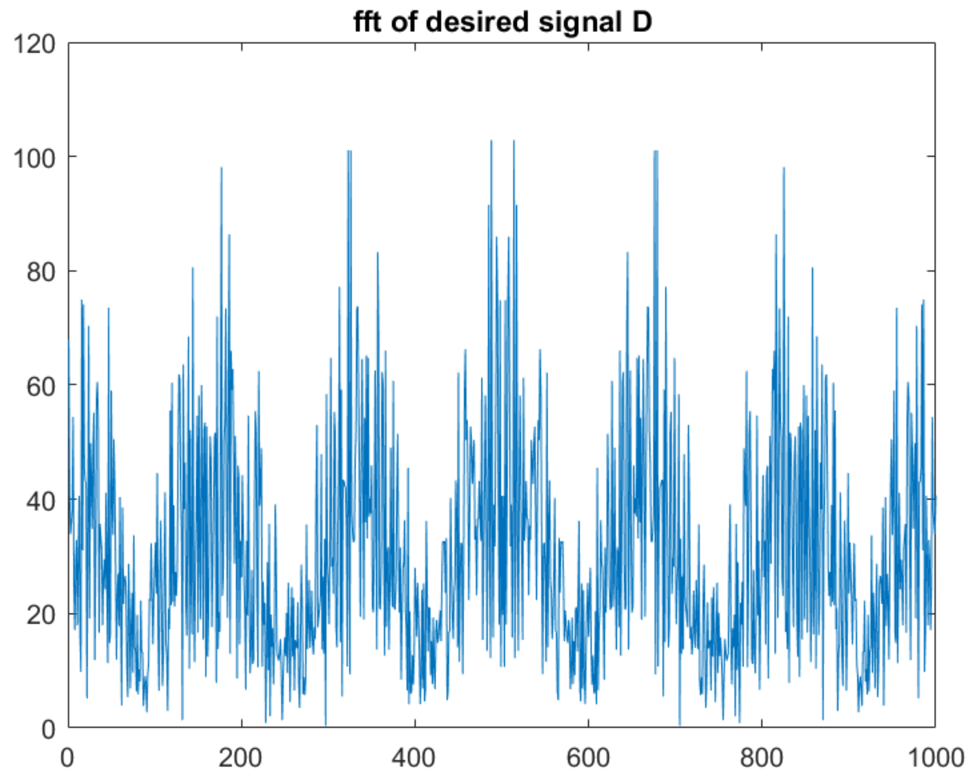
```
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 4, 2);
plot(W/pi, abs(H));
subplot(1, 4, 3);
plot(W/pi, 10*log10(abs(H)));
subplot(1, 4, 4);
stem(w);
sgtitle("MU = 0.0001, L = 1000");
```




```
y_out = zeros(L, 1);
err_out = zeros(L, 1);

w = zeros(M, 1);

mu = 0.01;

d = x_unk(length(h):end);
figure;
plot(abs(X));
title("fft of input X");
```

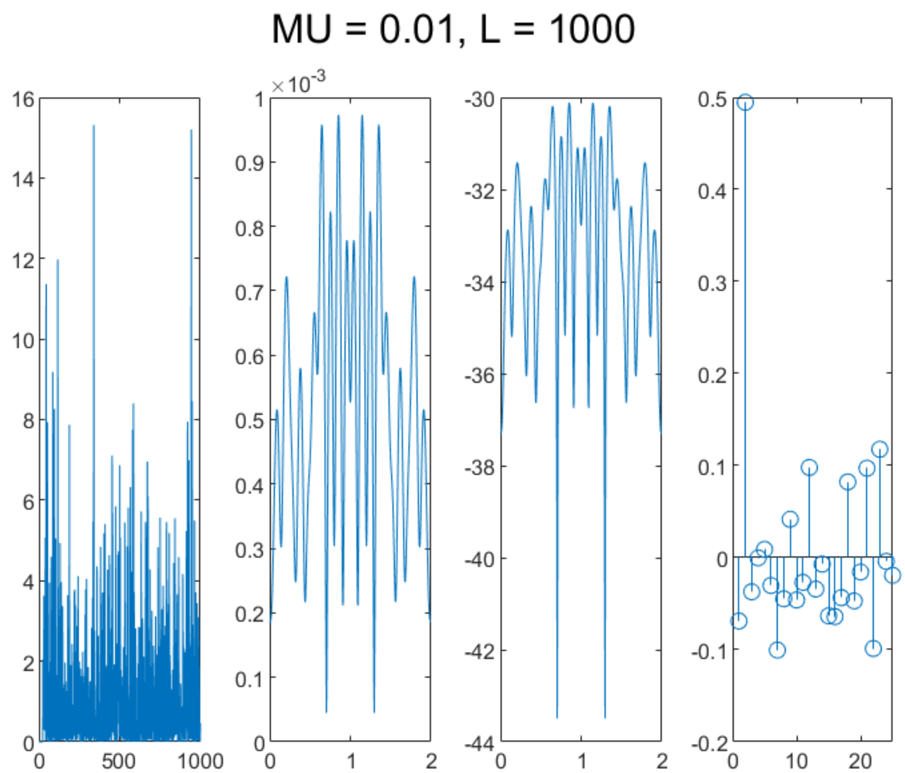



```
figure;
plot(abs(D));
title("fft of desired signal D");
```



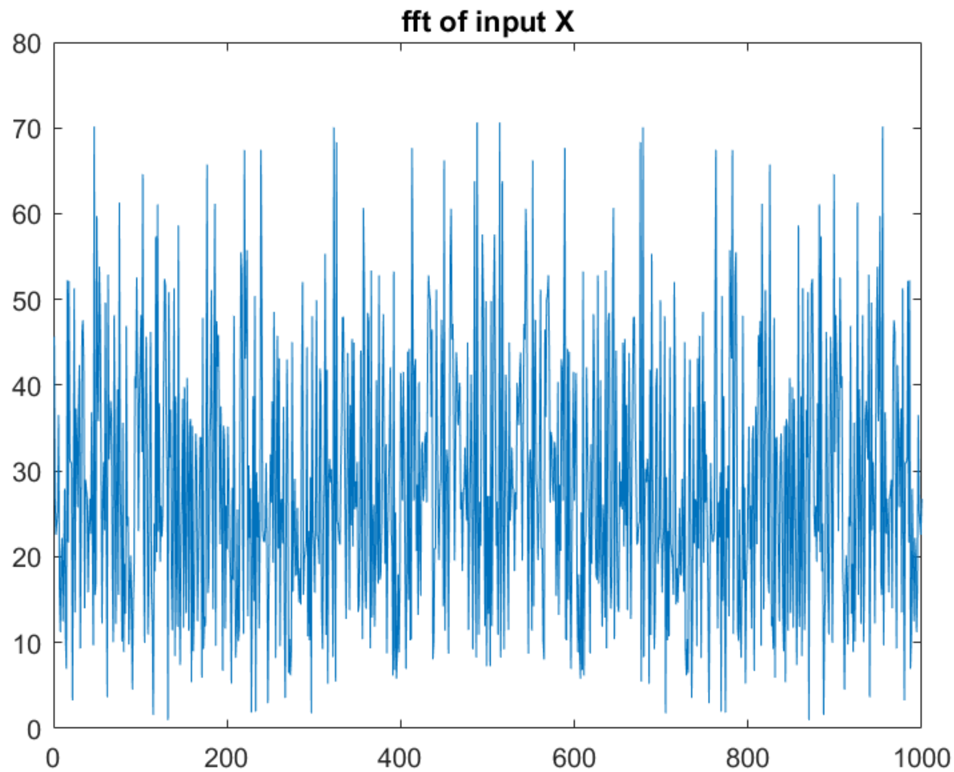

```
for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 4, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 4, 2);
plot(W/pi, abs(H));
subplot(1, 4, 3);
plot(W/pi, 10*log10(abs(H)));
```

```
subplot(1, 4, 4);
stem(w);
sgtitle("MU = 0.01, L = 1000");
```

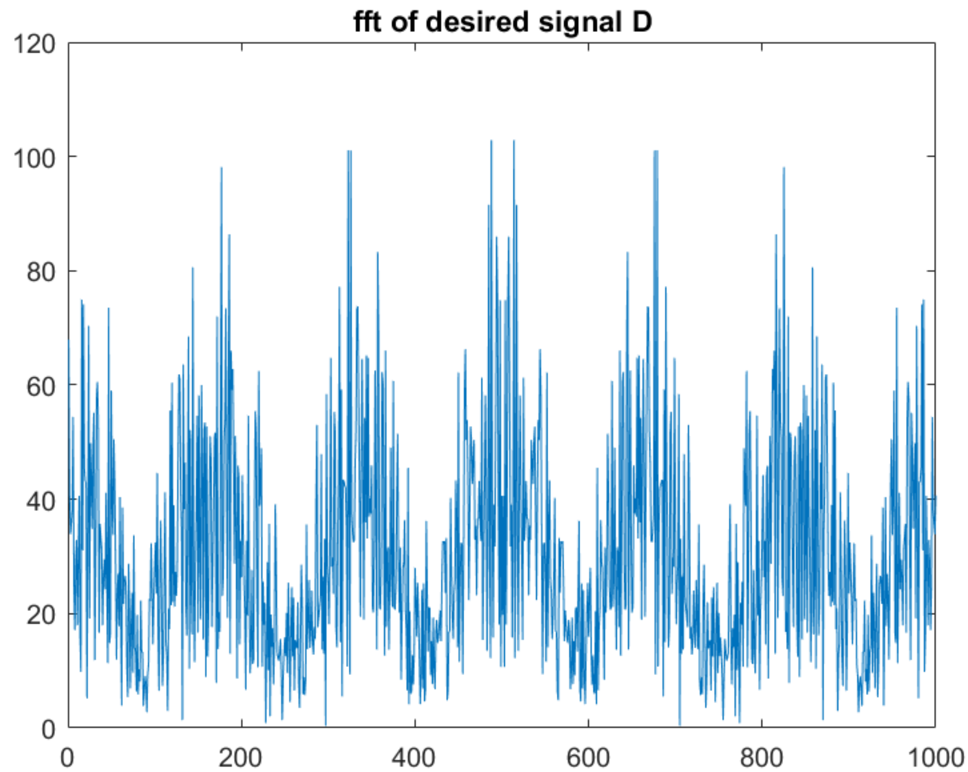


Step 4

```
M = 25;
L = 1000;
f_pm = [0, 1200, 1600, 2400, 2800, 4000]/4000;
a_pm = [0,0,1,1,0,0];
n_pm = 64;
b_pm = firpm(n_pm, f_pm, a_pm);
x = randn(1,L);
x_unk = conv(b_pm, x);
d = x_unk(1: end- length(b_pm)+1);

figure;
plot(abs(X));
title("fft of input X");
```

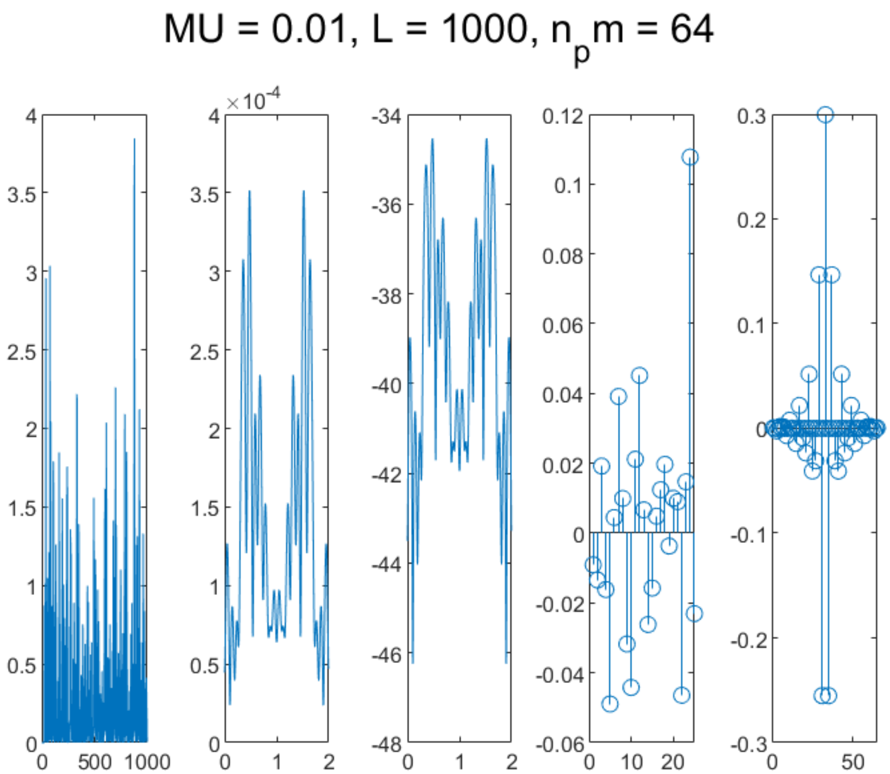



```
figure;
plot(abs(D));
title("fft of desired signal D");
```

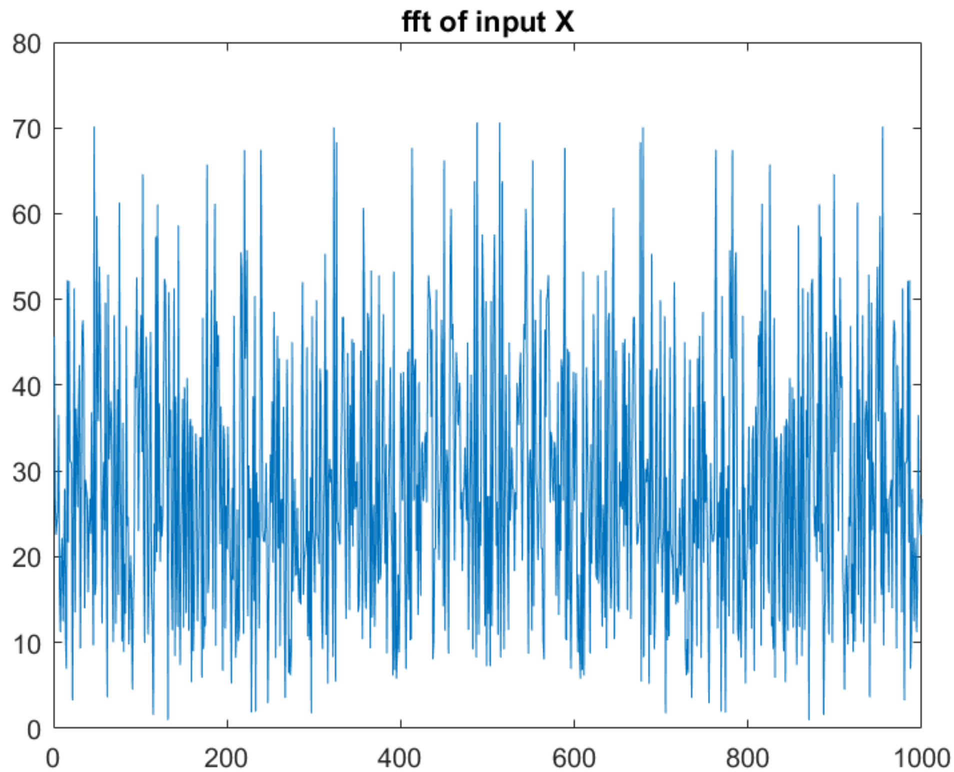



```
mu = 0.01;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 5, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 5, 2);
plot(W/pi, abs(H));
```

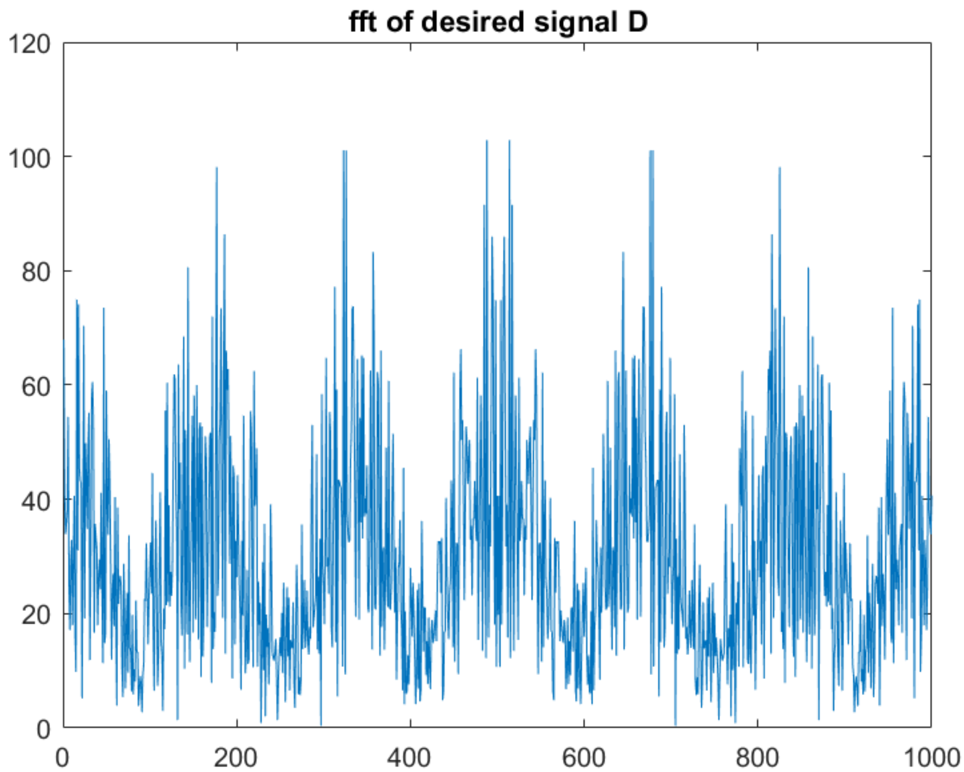
```
subplot(1, 5, 3);
plot(W/pi, 10*log10(abs(H)));
subplot(1, 5, 4);
stem(w);
subplot(1, 5, 5);
stem(b_pm);
sgtitle("MU = 0.01, L = 1000, n_pm = 64");
```




```

M = 25;
L = 1000;
f_pm = [0, 1200, 1600, 2400, 2800, 4000]/4000;
a_pm = [0,0,1,1,0,0];
n_pm = 32;
b_pm = firpm(n_pm, f_pm, a_pm);
x = randn(1,L);
x_unk = conv(b_pm, x);
d = x_unk(1: end- length(b_pm)+1);

figure;
plot(abs(X));
title("fft of input X");
```




```
figure;
plot(abs(D));
title("fft of desired signal D");
```




```
mu = 0.01;

for n = M:L

    xn_index = 1;

    for j = n:-1:n-M+1
        xn(xn_index) = x(mod(j, L) + 1);
        xn_index = xn_index + 1;
    end

    yn = dot(transpose(w), xn);

    y_out(n) = yn;
    en = d(n) - yn;
    err_out(n) = en;

    w = w + mu*en*transpose(xn);
end

figure;
subplot(1, 5, 1);
plot(err_out.^2);

[H, W] = freqz(w, 1024, 'whole');
subplot(1, 5, 2);
plot(W/pi, abs(H));
```

```
subplot(1, 5, 3);
plot(W/pi, 10*log10(abs(H)));
subplot(1, 5, 4);
stem(w);
subplot(1, 5, 5);
stem(b_pm);


sgtitle("MU = 0.01, L = 1000, n pm = 32");
```