Experiment 06

Code:

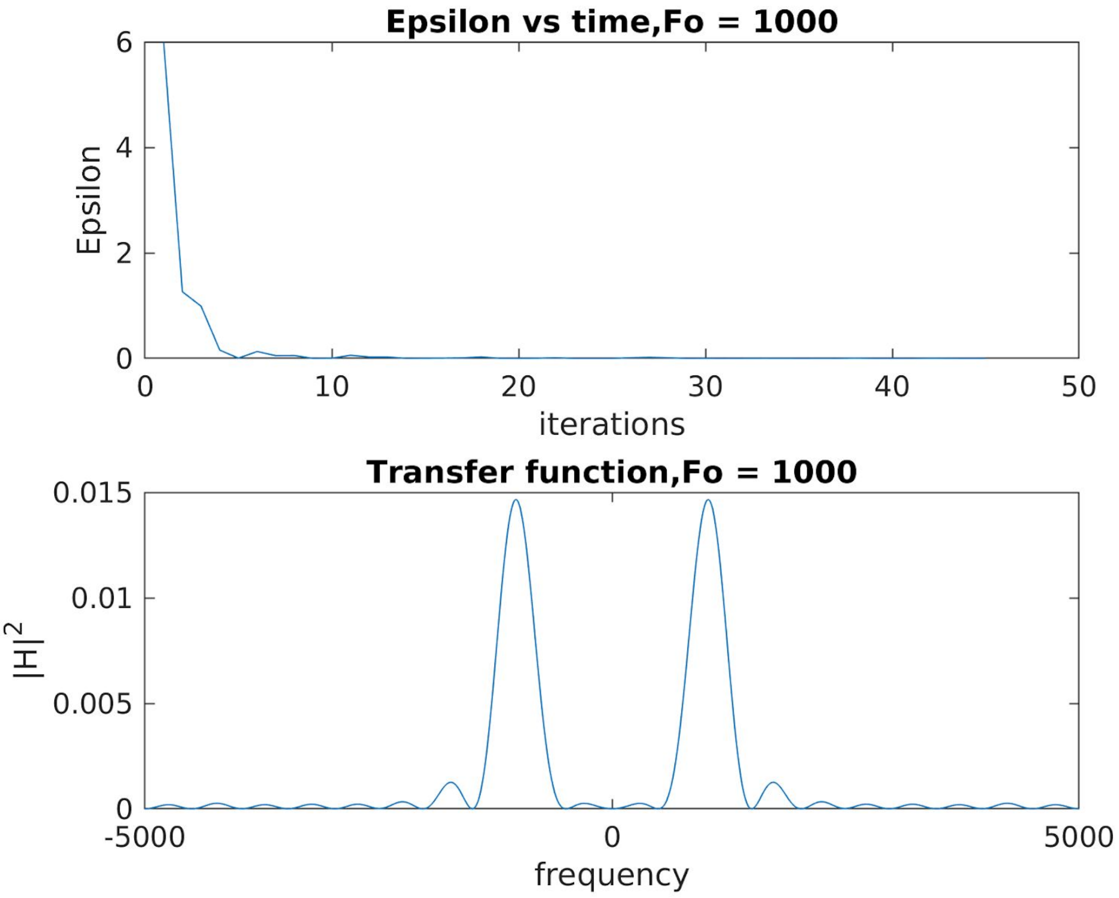
```
clc
clear all
close all

A = 2;
Fo = 1000;          %Fo = 1KHz
Fs = 10000;
t = 0:1/Fs:0.1;
m = A*sin(2*pi*Fo*t);

noise = normrnd(0,1,1,length(m));
x = m + noise;

N=20;
mu = 0.0001;
w = zeros(1,N);
epsilon = 1;
error = [];
i = 1;
a = [];

while epsilon > 0.00001
   w_prev = w;
   error(i,1) = x(1,i+N) - w*(x(N+i-1:-1:i))';
   w = w +mu*x(N+i-1:-1:i)*error(i,1);
   epsilon = ((norm(w-w_prev))/(norm(w_prev)))^2;
   a(i,1) = epsilon;
   i = i+1;
end

epsilon_vec = a(2:length(a),1);
subplot(2,1,1);
plot(epsilon_vec);
xlabel('iterations')
ylabel('Epsilon');
title('Epsilon vs time,Fo = 1000');

subplot(2,1,2);
```

```
[h,omega] = freqz(w,1,linspace(-pi,pi,1000));
plot(omega*Fs*0.5/pi, abs(h).^2);
xlabel('frequency');
ylabel('|H|^2');
title('Transfer function,Fo = 1000');
```

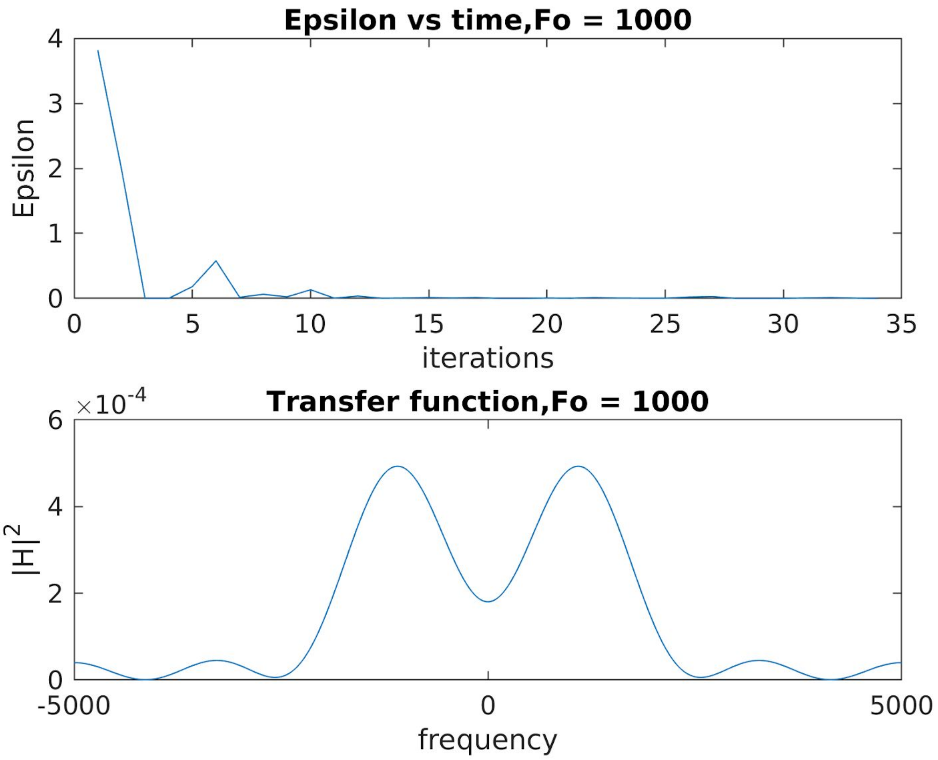
Plots :

For all plots, mu(learning rate) = 10^{-4} and epsilon = 10^{-5}

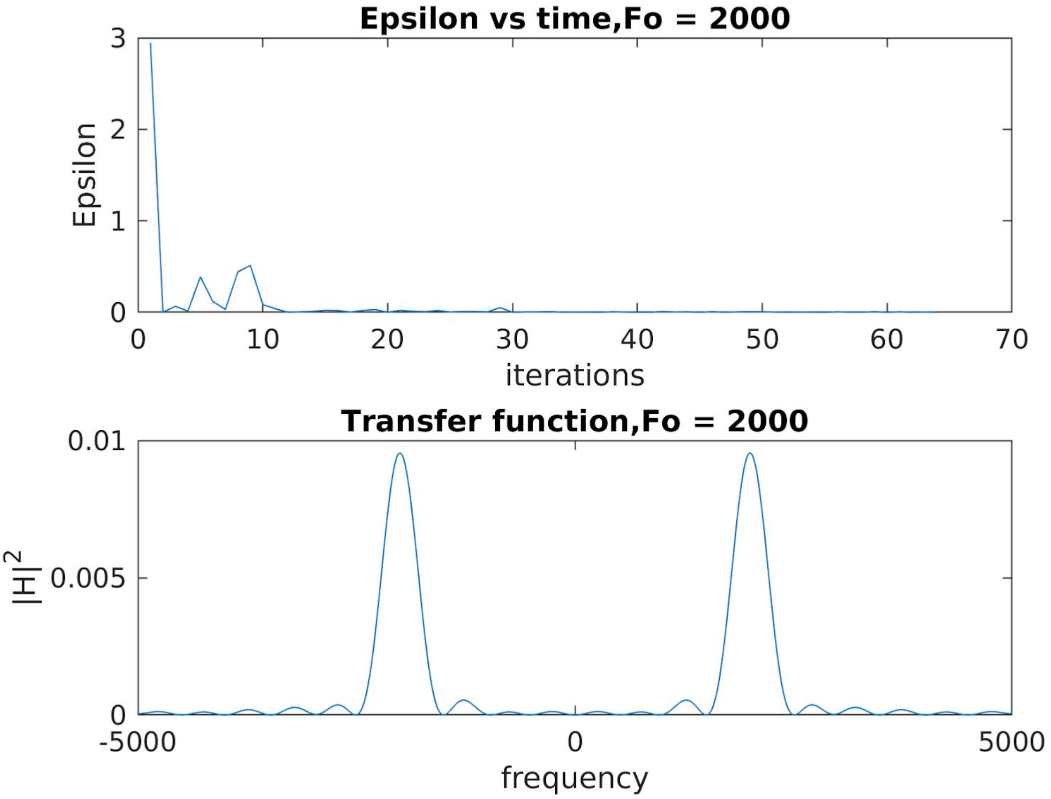
- N = 20, Fo = 1000Hz:



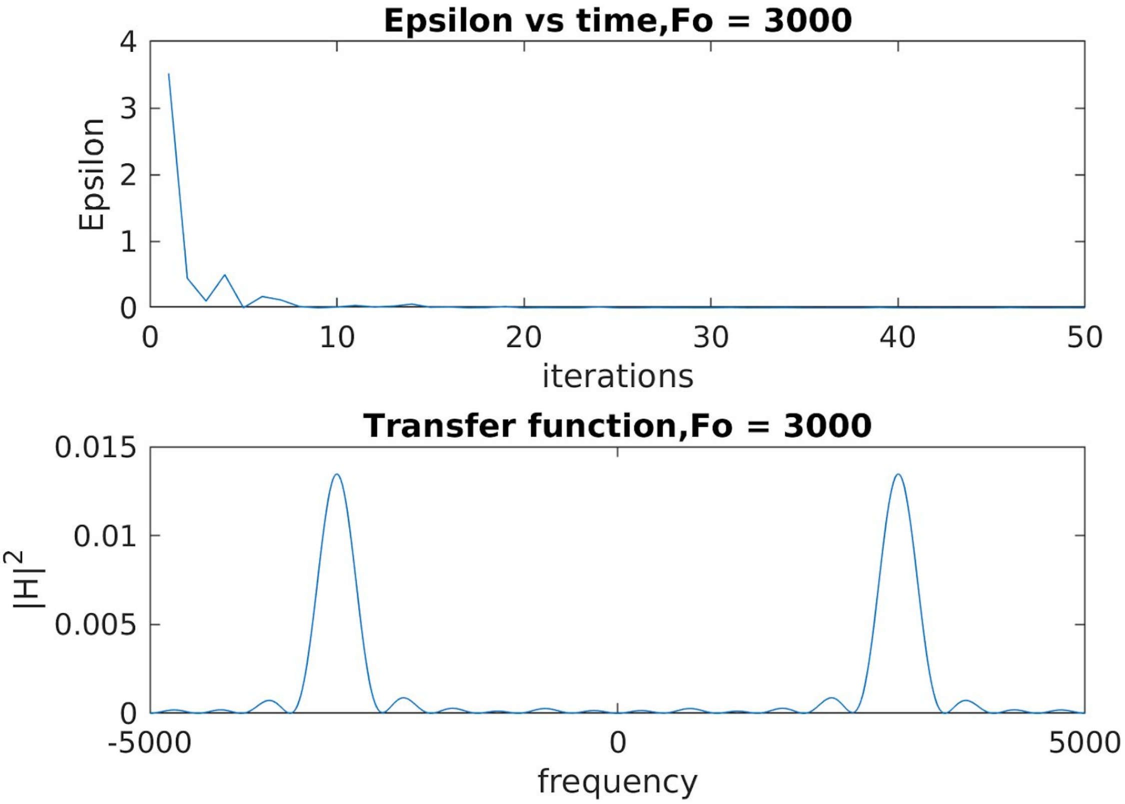
- For N = 6, Fo = 1000 Hz, we get peaks at the desired values (1000Hz), but it is less steeper.



- N = 20, F0 = 2000



- N = 20, F0 = 3000



- The larger value of Mu , the quicker will be the convergence of the weights but if mu is very high , it might miss the minimum value.
- Here we started the updating the values of weight matrix from n = 20 ; because we needed previous n-20 values of signal to calculate the error.
- Instead we could have interpolated the signal till n = -20 and started the weight updation at n = 0 itself.