Digital Signal Processing Lab

Experiment 5 – Adaptive Line Enhancer

Utkarsh Patel (18EC30048)

---

1 Objective

To detect low level sine wave of unknown frequency in presence of noise

2 Theory

An adaptive line enhancer (ALE) is used to detect a low-level sine wave of unknown frequency in presence of noise. If the input frequency changes, the filter adapts itself to be a bandpass filter centred at the input frequency. The ALE is usually realised by using the so-called adaptive filter. In a general adaptive filter, the filter coefficients are updated in time by an adaptation algorithm during an initial training phase, so that filter output $y(n)$ becomes a better and better estimate of the desired response $d(n)$ given during this phase.

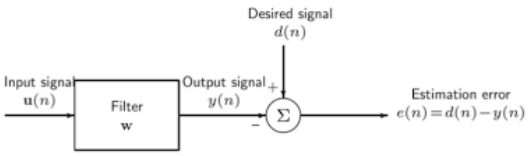

Fig. 1. ***LMS Algorithm:*** $d(n)$ is the desired response, $e(n)$ is the error. The transfer function is given as $W(z) = \sum_{i=0}^{p} w_i z^{-i}$

The filter coefficients are updated in time in the LMS algorithm, following a steepest descent along the negative direction of the gradient of the mean squared error. The ideal steepest descent procedure leads to

$$w(n+1) = w(n) - \frac{\mu}{2}\nabla_w \varepsilon^2$$

where $\varepsilon^2 = E\left(e^2(n)\right)$, $\nabla_w \varepsilon^2 = \left[\frac{\partial \varepsilon^2}{\partial w_0}\ \frac{\partial \varepsilon^2}{\partial w_1} \ldots \frac{\partial \varepsilon^2}{\partial w_p}\right]^T$ and μ is a constant controlling the convergence rate. The above weight update equation can be re-written as

$$w(n+1) = w(n) + \mu\left(\boldsymbol{p} - \boldsymbol{R}w(n)\right)$$

where $\boldsymbol{R} = E[x(n)x(n)^T]$, $\boldsymbol{p} = E[x(n)d(n)]$, $x(n) = [u(n)\ u(n-1) \ldots\ u(n-p)]^T$.

However, in practice $\boldsymbol{R}$ and $\boldsymbol{p}$ are not known in advance and are estimated from the input signal online. In LMS algorithm, $\boldsymbol{R}$ and $\boldsymbol{p}$ are replaced by the estimates $x(n)x(n)^T$ and $x(n)d(n)$ respectively, which modifies the weight update equation to

$$w(n+1) = w(n) + \mu\, x(n)e(n)$$

This method converges for $0 < \mu < \frac{2}{tr(\boldsymbol{R})}$.

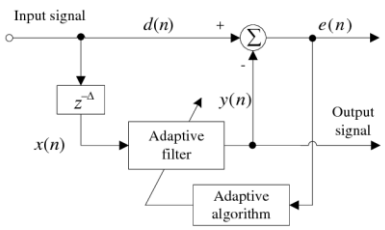

Fig. 2. ***Adaptive Line Enhancer:*** The desired response $d(n)$ is simply the input $x(n)$

3 Procedure

- Take a sinusoidal message waveform $m(t) = Asin(2\pi F_0 t)$. Take $A = 2$ and $F_0 = 1$ KHz.
- Add white Gaussian noise $n(t)$ of zero mean and unity variance to $m(t)$ and obtain $x(t)$.
- Sample $x(t)$ to obtain $x(n)$.
- Pass $x(n)$ to the system shown in Fig. 2.
- Adapt the coefficients of the filter using $\mu = 10^{-4}$.
- Continue the iteration till the relative change (norm wise) in $w(n)$ is less than δ (10^{-3}).
- Repeat the experiment for $F_0 = 2, 3, 10$ KHz.

4 Results

During the simulation, the adaptive filter of order 20 is used.

4.1 Sinusoid with 1 KHz frequency component

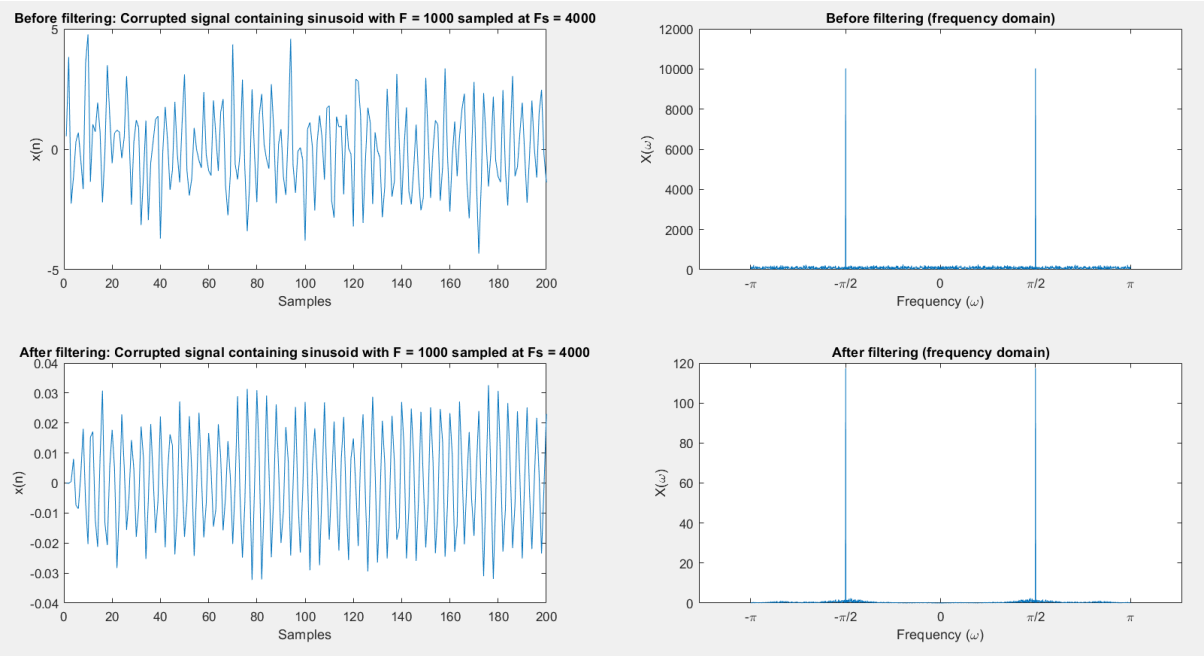

Fig. 3. ***Time and frequency domain representation of corrupted signal before and after filtering:*** It can be observed that the adaptive filter is able to learn the bandpass characteristics with centre frequency at 1 KHz with quite good performance.

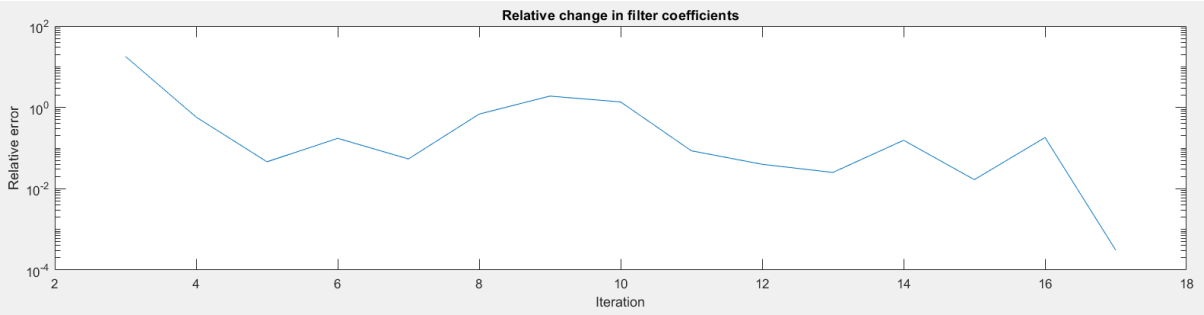

Fig. 4. ***Relative change in filter coefficients over the iterations:*** The change is not monotonic decreasing in nature, however it converges as μ satisfies the convergence criterion.

4.2 Sinusoid with 2 KHz frequency component

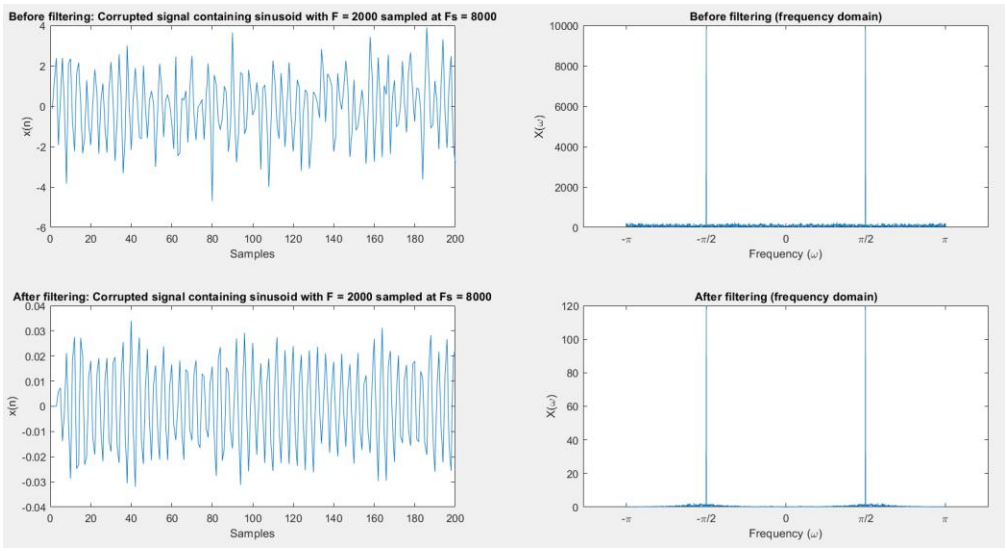

Fig. 5. ***Time and frequency domain representation of corrupted signal before and after filtering:*** It can be observed that the adaptive filter is able to learn the bandpass characteristics with centre frequency at 2 KHz with quite good performance.

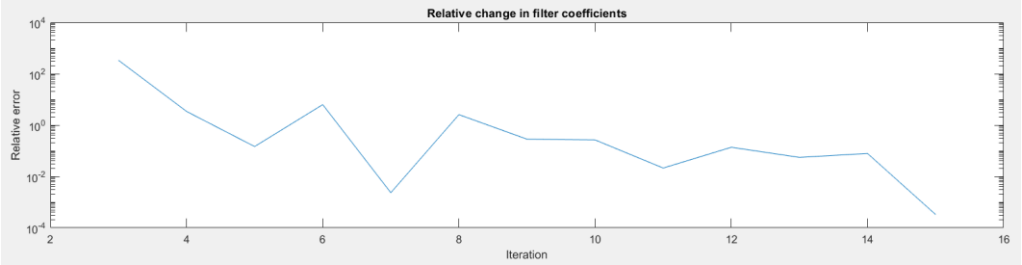

Fig. 6. ***Relative change in filter coefficients over the iterations:*** The change is not monotonic decreasing in nature, however it converges as μ satisfies the convergence criterion.

4.3 Sinusoid with 3 KHz frequency component

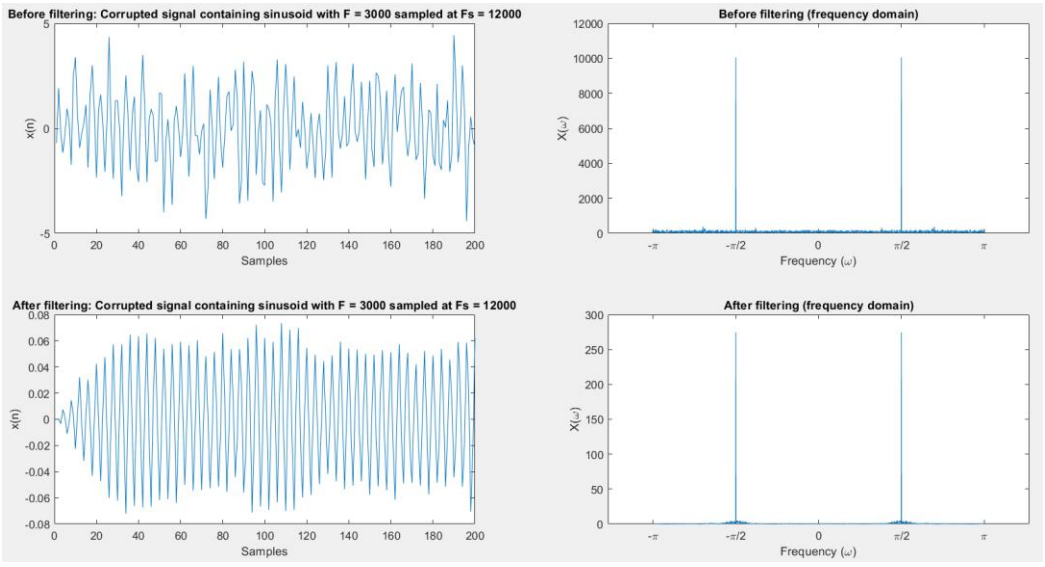

Fig. 7. ***Time and frequency domain representation of corrupted signal before and after filtering:*** It can be observed that the adaptive filter is able to learn the bandpass characteristics with centre frequency at 3 KHz with quite good performance.

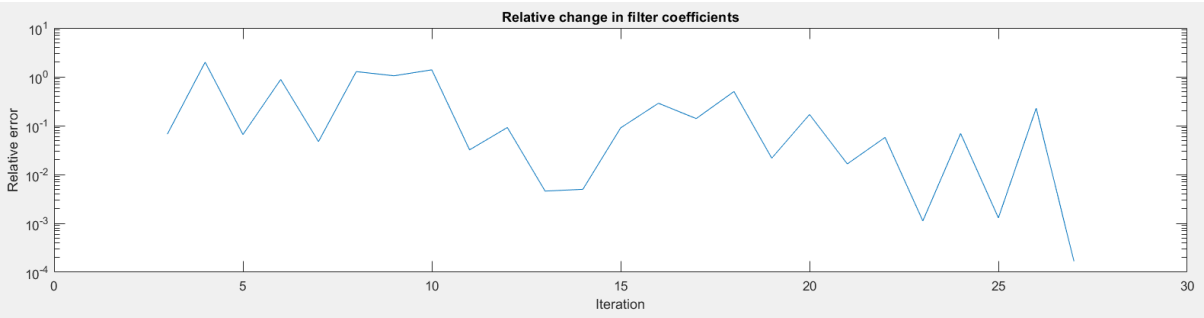

Fig. 8. ***Relative change in filter coefficients over the iterations:*** The change is not monotonic decreasing in nature, however it converges as μ satisfies the convergence criterion.

4.4 Sinusoid with 10 KHz frequency component

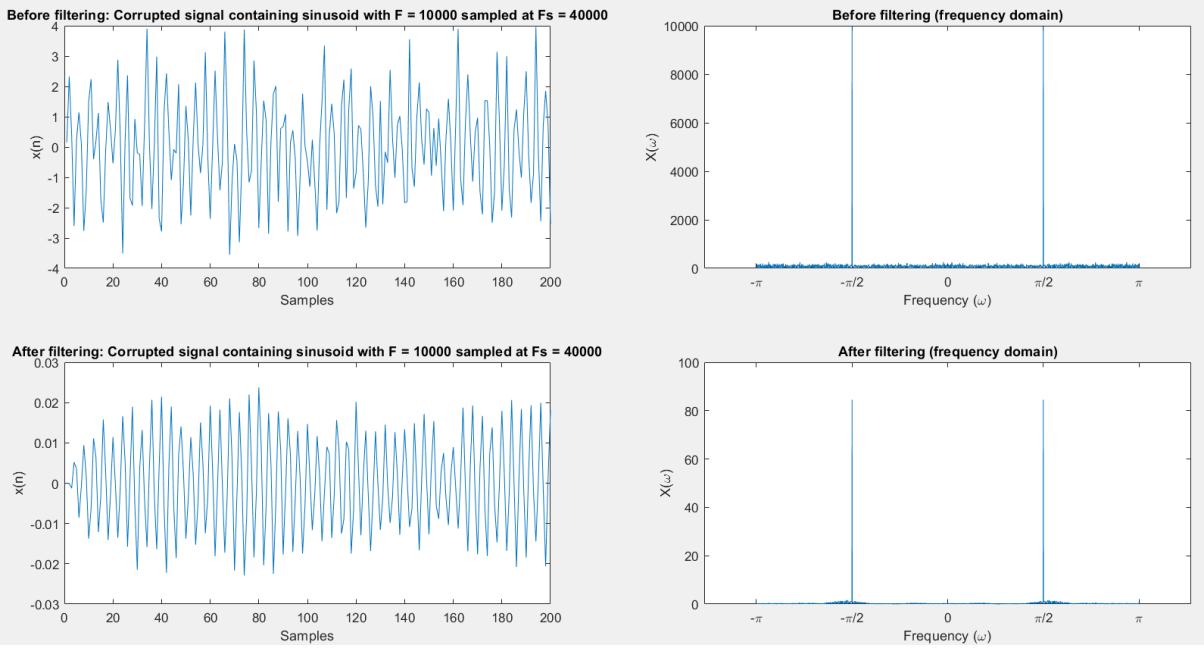

Fig. 9. ***Time and frequency domain representation of corrupted signal before and after filtering:*** It can be observed that the adaptive filter is able to learn the bandpass characteristics with centre frequency at 10 KHz with quite good performance.

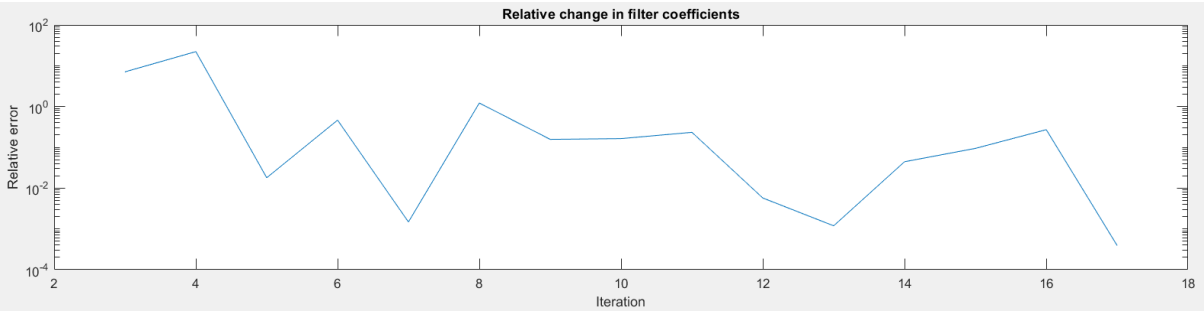

Fig. 8. ***Relative change in filter coefficients over the iterations:*** The change is not monotonic decreasing in nature, however it converges as μ satisfies the convergence criterion.

5 Discussion

- In this experiment, the objective is to detect low-level sine wave of unknown frequency in presence of noise.

- For this purpose, the adaptive line enhancer is used. In this framework, the filter adapts itself to be a bandpass filter centred at the frequency of the sinusoid.

- Adaptive line enhancer is realised by using adaptive filter in which the filter coefficients are updated in time so that the output signal could mimic the desired signal.

- This filter is implemented using LMS algorithm. In this algorithm, the filter coefficients are updated following the steepest descent along the negative direction of the gradient of the mean squared error.

- As the parameters used in LMS algorithm are not known *a-priori*, they are estimated from the input signal itself.

- The convergence of this algorithm is guaranteed under given condition.

- For this experiment, the corrupted signal is realised by summing a sinusoidal signal with white Gaussian noise of unity variance.

- The corrupted signal is fed to the system (Fig. 2) and iterations continue until the relative change is below a given threshold.

- The experiment is performed with sinusoid of frequency 1, 2, 3 and 10 KHz.

- It is observed that the adaptive filter performs quite good in all the four cases, learning the bandpass characteristics by itself. (Fig. 3, 5, 7 and 9)

- In all the four cases, the convergence is achieved within at most 30 iterations. However, the relative change in filter coefficients was not monotonically decreasing with increasing iterations. (See Fig 4, 6, 8 and 10)

6 MATLAB codes

```
N = 20;    % order of adaptive filter
f = 10000; % change frequency here

Plot(f, N);

function Plot(f, N)
    % plots characteristics for filter order N
    % with signal having sinusoid with frequency f
    Fs = 4 * f;
    t = (0: 1: 10000) * (1 / Fs);
    noise = normrnd(0, 1, 1, length(t)); % noise
    a = 2 * sin(2 * pi * f * t); % sinusoid signal
    x = a + noise; % corrupted signal

    w = zeros(1,N+1); % adaptive filter

    y = zeros(1,N+1); % dummy variable
    z = zeros(1,N+1); % dummy variable
    i = 1;

    rerr = zeros(1, 20000);

    u = 0.0001; % learning rate
    eps = 0.001; % epsilon

    while(1)
        nz = zeros(1, N + 1);
        nz(1) = x(i);
        for j = 2: N + 1
            nz(j) = z(j - 1);
        end
        z = nz;
        y = sum(z.*w);
        e = y - x(i) ;
        w1 = w; % saving the previous value of w to w1
        w = w + u * z * e; % gradient descent
        w(1) = 0;
        rerr(i) = (norm(w - w1) / norm(w1));
        rerr(i) = rerr(i) * rerr(i);

        if rerr(i) < eps
            break;
        end

        i = i + 1;
    end
    w_ = -1 * pi: 0.001 * pi: pi;

    disp(rerr(1:i));

    subplot(2, 2, 1);
    plot(x(1:200));
    title('Before filtering: Corrupted signal containing sinusoid with F = ' + string(f) + ' sampled at Fs = ' + string(Fs));
    xlabel('Samples');
    ylabel('x(n)');

    subplot(2, 2, 2);
    plot(w_, abs(freqz(x, 1, w_)));
    xticks([-pi -pi/2 0 pi/2 pi]);
    xticklabels({'-\pi','-\pi/2','0','\pi/2','\pi'});
    title('Before filtering (frequency domain)');
    xlabel('Frequency (\omega)');
    ylabel('X(\omega)');
    x_f = conv(w, x);
    subplot(2, 2, 3);
    plot(x_f(1:200));
    title('After filtering: Corrupted signal containing sinusoid with F = ' + string(f) + ' sampled at Fs = ' + string(Fs));
    xlabel('Samples');
    ylabel('x(n)');

    subplot(2, 2, 4);
    plot(w_, abs(freqz(conv(w, x), 1, w_)));
    xticks([-pi -pi/2 0 pi/2 pi]);
    xticklabels({'-\pi','-\pi/2','0','\pi/2','\pi'});
    title('After filtering (frequency domain)');
    xlabel('Frequency (\omega)');
    ylabel('X(\omega)');

    figure;
    subplot(2, 1, 1);
    plot(rerr(1:i));
    title('Relative change in filter coefficients');
    xlabel('Iteration');
    ylabel('Relative error');
    set(gca, 'YScale', 'log');
end
```