DIGITAL SIGNAL PROCESSING LABORATORY
E&ECE DEPARTMENT
INDIAN INSTITUTE OF TECHNOLOGY
KHARAGPUR

Experiment No: 06

ADAPTIVE LINE ENHANCER

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 centered at the input frequency. The ALE is usually realized 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. This is shown in Figure 1. The most widely used adaptation algorithm is the Least Mean Square (LMS) algorithm, which uses the error signal $e(n)$ in a feedback loop (not shown) for coefficient adaptation.

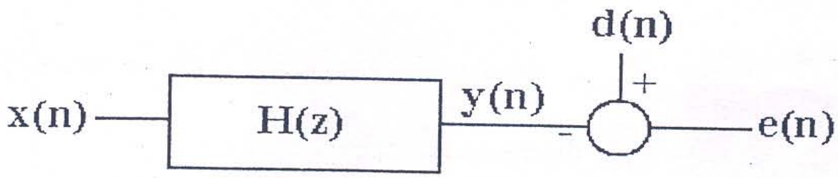

Fig. 1: $d(n)$ is the desired response, $e(n)$ is the error.

The transfer function, $H(z) = w_0 + w_1 z^{-1} + w_2 z^{-2} + \ldots + w_p z^{-p}$

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,

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \nabla_{\mathbf{w}} \varepsilon^2 \Big|_{\mathbf{w}=\mathbf{w}(n)}, \text{where } \varepsilon^2 = E\left[e^2(n)\right],$$

$$\nabla_{\mathbf{w}} \varepsilon^2 = \begin{bmatrix} \dfrac{\partial \varepsilon^2}{\partial w_0} & \dfrac{\partial \varepsilon^2}{\partial w_1} & \cdots & \cdots & \dfrac{\partial \varepsilon^2}{\partial w_p} \end{bmatrix}^T$$ and μ is a constant controlling the convergence rate.

This can be equivalently written as,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(\mathbf{p} - \mathbf{R}\mathbf{w})\big|_{\mathbf{w}=\mathbf{w}(n)}$$

where, $\mathbf{R} = E\left[\mathbf{x}(n)\mathbf{x}(n)^T\right]$, $\mathbf{p} = E\left[\mathbf{x}(n)d(n)\right]$,

$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad \ldots \quad x(n-p)]^T$ and $\mathbf{w}(n) = [w_0 \quad w_1 \quad \ldots \quad \ldots \quad w_p]^T$.

However, in practice, $\mathbf{R}$ and $\mathbf{p}$ are not known *a priori* and are estimated from the data online, thus making the weight update recursion adaptive. In the case the LMS algorithm, $\mathbf{R}$ and $\mathbf{p}$ are replaced by the estimates : $\mathbf{x}(n)\,\mathbf{x}^T(n)$ and $\mathbf{x}(n)\,d(n)$ respectively. This results in the celebrated LMS filter coefficient update formula :

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\,\mathbf{x}(n)\,e(n).$$

It can be shown that the algorithm converges for $0 < \mu < \dfrac{2}{trace(\mathbf{R})}$.

In an adaptive line enhancer (Fig.2) the desired response is simply the input $x(n)$.

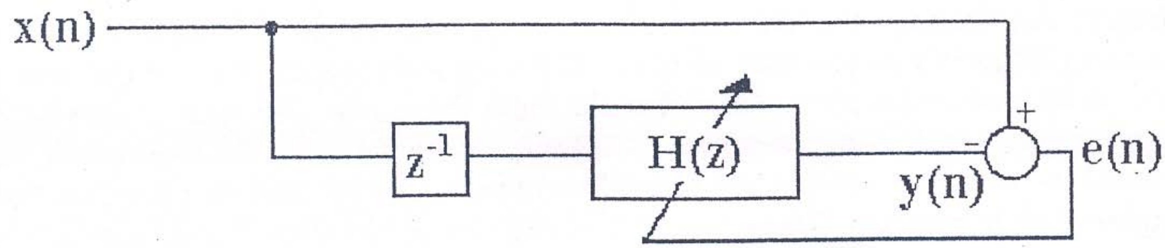

Fig.2: An adaptive line enhancer

Step:

1. Take a sinusoidal message waveform $m(t) = A\sin(2\pi \times F_0 t)$ (Take $A = 2$ and F_0=1 kHz).
2. Add white Gaussian noise $n(t)$ of zero mean and unity variance to $m(t)$ and obtain the input signal $x(t)$.
3. Sample it properly to generate the discrete input signal $x(n)$.
4. Pass $x(n)$ through the system as shown below (Fig.2)
5. Adapt the filter coefficients as
   $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\,\mathbf{x}(n)\,e(n)$
   Take $\mu = 10^{-4}$.
6. Continue the iteration in step 5. until the relative change $\dfrac{\|\mathbf{w}(n+1) - \mathbf{w}(n)\|^2}{\|\mathbf{w}(n)\|^2} < \varepsilon'$

   Take $\varepsilon' = 10^{-3}$.
7. Plot the latest transfer function $\left|H\left(e^{j2\pi f}\right)\right|^2$ of the filter (Fig.3).
8. Repeat the above with F_0=2, 3, 10 kHz and observe the change in $\left|H\left(e^{j2\pi f}\right)\right|^2$.

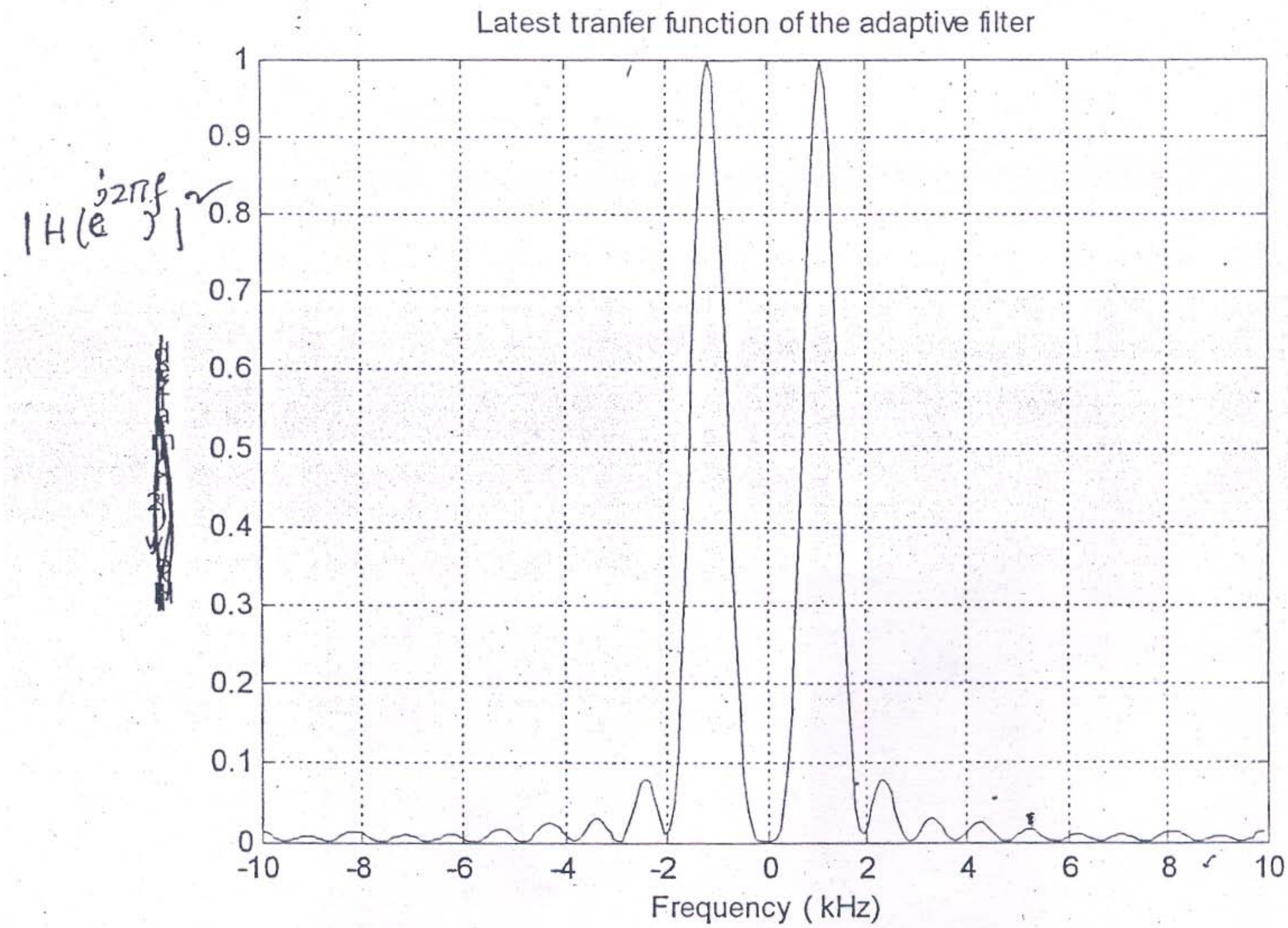

Fig.3:. The transfer function of the adaptive filter

- MATLAB functions that you may require: *randn,freqz.*

Reference:

[1] S.Haykin, *Adaptive Filter Theory*, Prentice-Hall, 1985

[2] J.Makhoul, "Linear prediction: A tutorial review," Proc. IEEE, vol. 63, pp. 649-661, 1975.