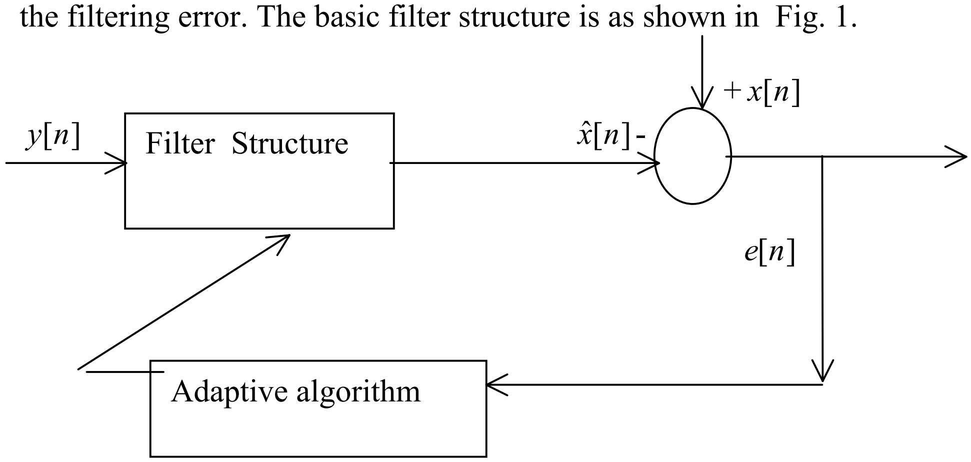
Adaptive Filtering

Wiener filter is a linear time invariant filter.
In practical situation, the signal is nonsationary and demands the optimal filter to be time varying.
How to do this?

- Assume statinarity within certain data length. Buffering of data is required and may work in some applications.
- The time-duration over which stationarity is a valid assumption, may be short so that accurate estimation of the model parameters is difficult.

One solution is *adaptive filtering*. Here the filter coeffients are updated as a function of the filtering error. The basic filter structure is as shown in Fig. 1.



The filter structure is FIR of known tap-length, because the adaptation algorithm updates each filter coefficient individually.

Method of Steepest Descent

Consider the FIR Wiener filter of length M. We want to compute the filter coefficients iteratively.

Let us denote the time-varying filter parameters by

$$h_i[n], \mathrm{i} = 0,1,\ldots \mathrm{M} - 1$$

and define the filter parameter vector by

$$\mathbf{h}[n] = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{M-1} \end{bmatrix}$$

We want to find the filter coefficients so as to minimize the mean-square eror $Ee^2[n]$ where

$$\begin{aligned} e[n] &= x[n] - \hat{x}[n] \\ &= \mathrm{x[n]} - \sum_{i=0}^{M-1} h_i[n] y[n-i] \\ &= \mathrm{x[n]} - \mathbf{h}'[n]\mathbf{y}[n] \\ &= \mathrm{x[n]} - \mathbf{y}'[n]\mathbf{h}[n] \end{aligned}$$

$$\text{where } \mathbf{y}[n] = \begin{bmatrix} y[n] \\ y[n-1] \\ \vdots \\ y[n-M+1] \end{bmatrix}$$

Therefore

$$\begin{aligned} Ee^2[n] &= E(x[n] - \mathbf{h}'[n]\mathbf{y}[n])^2 \\ &= R_{XX}[0] - 2\mathbf{h}'[n]\mathbf{r}_{xy} + \mathbf{h}'[n]\mathbf{R}_{\mathbf{YY}}\mathbf{h}[n] \end{aligned}$$

$$\text{where } \mathbf{r}_{XY} = \begin{bmatrix} R_{XY}[0] \\ R_{XY}[1] \\ \vdots \\ R_{XY}[M-1] \end{bmatrix}$$

The cost function represented by $Ee^2[n]$ is a quadratic in $\mathbf{h[n]}$ with a unique minimum obtained by setting the gradient of $Ee^2[n]$ to zero.

The optimal set of filter parameters are given by

$$\mathbf{h}_{\mathbf{opt}} = \mathbf{R}_{\mathbf{XY}}^{-1}\mathbf{r}_{\mathbf{XY}}$$

which is the FIR Wiener filter.

Many of the adaptive filter algorithms are obtained by simple modifications of the algorithms for deterministic optimization. Most of the popular adaptation algorithms are based on gradient-based optimization techniques, particularly the steepest descent technique.

The optimal Wiener filter can be obtained iteratively by the method of steepest descent. The optimum is found by updating the filter parameters by the rule

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \frac{\mu}{2}(-\nabla Ee^2[n])$$

where

$$\nabla Ee^2[n] = \begin{bmatrix} \frac{\partial Ee^2[n]}{\partial h} \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \frac{\partial Ee^2[n]}{\partial h_{M-1}} \end{bmatrix}$$

$$= -2\mathbf{r}_{XY} + 2\mathbf{R}_{\mathbf{YY}}\mathbf{h}[n]$$

and μ is the step - size parameter.

So the steepest descent rule will now give

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \mu(\mathbf{r}_{XY} - \mathbf{R}_{YY}\mathbf{h}[n])$$

Convergence of the steepest descent method

We have

$$\begin{aligned} \mathbf{h}[n+1] &= \mathbf{h}[n] + \mu(\mathbf{r}_{\mathbf{XY}} - \mathbf{R}_{\mathbf{YY}}\mathbf{h}[n]) \\ &= \mathbf{h}[n] - \mu\mathbf{R}_{\mathbf{YY}}\mathbf{h}[n] + \mu\mathbf{r}_{XY} \\ &= (\mathbf{I} - \mu\mathbf{R}_{\mathbf{YY}})\mathbf{h}[n] + \mu\mathbf{r}_{XY} \end{aligned}$$

where $\mathbf{I}$ is the MXM identity matrix.

This is a coupled set of linear difference equations.

Can we break it into simpler equations?

$\mathbf{R}_{\mathbf{yy}}$ can be diagonalised (KL transform) by the following relation

$$\mathbf{R}_{\mathbf{YY}} = \mathbf{Q\Lambda Q'}$$

where $\mathbf{Q}$ the orthogonal matrix of the eigen vector of $\mathbf{R}_{YY}$.

$\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigen values as the diagonal elements.

Also $\mathbf{I} = \mathbf{QQ'} = \mathbf{Q'Q}$

Therefore

$$\mathbf{h}[n+1] = (\mathbf{QQ'} - \mu\mathbf{Q}\Lambda\mathbf{Q'})\mathbf{h}[n] + \boldsymbol{\mu}\cdot\mathbf{r}_{XY}$$

Multiply by $\mathbf{Q'}$

$$\mathbf{Q'h}[n+1] = (\mathbf{I} - \mu\cdot\mathbf{\Lambda})\mathbf{Q'h}[n] + \mu\cdot\mathbf{Q'}\cdot\mathbf{r}_{XY}$$

Define a new variable

$\overline{\mathbf{h}}[n] = \mathbf{Q'h}[n]$ and $\overline{\mathbf{r}}_{XY} = \mathbf{Q'r}_{XY}$

Then

$$\overline{\mathbf{h}}[n+1] = (\mathbf{I} - \mu\mathbf{\Lambda})\overline{\mathbf{h}}[n] + \mu\overline{\mathbf{r}}_{XY}$$

$$
= \begin{bmatrix} 1-\mu\lambda_1 & 0\cdots\cdots & \cdots & .0 \\ 0 & & & \\ \vdots & & & \\ \vdots & & & \\ 0 & \ldots\ldots & \ldots & 1-\mu\lambda_M \end{bmatrix} \overline{\mathbf{h}}[n] + \mu\overline{\mathbf{r}}_{\mathbf{xy}}
$$

This is a decoupled set of linear difference equations

$$\overline{h}_i[n+1] = (1-\mu\lambda_i).\overline{h}_i[n] + \mu.\overline{R}_{xy}[i] \qquad i = 0,1,...M-1$$

and can be easily solved for stability. The stability condition is given by

$$\left|1-\mu\lambda_i\right| < 1$$

$$\Rightarrow -1 < 1-\mu\lambda_i < 1$$

$$\Rightarrow 0 < \mu < 2/\lambda_i, i = 1,.......M$$

Let $\lambda_{\max}$ is the maximum eigen value

$$
\begin{aligned}
\lambda_{\max} &< \lambda_1 + \lambda_2 + ...\lambda_M \\
&= \text{Trace}(\mathbf{R}_{\mathbf{yy}})
\end{aligned}
$$

$$
\begin{aligned}
\therefore 0 < \mu &< \frac{2}{Trace(\mathbf{R}_{\mathbf{yy}})} \\
&= \frac{2}{\text{M.R}_{\text{YY}}[0]}
\end{aligned}
$$

The steepest decent algorithm converges to the corresponding Wiener filter

$$\lim_{n\to\infty} \mathbf{h}[n] = \mathbf{R}_{\mathbf{yy}}^{-1}\mathbf{r}_{\mathbf{xy}}$$

if the stepsize μ is within the range of specified by the above relation.

LMS algorithm (Least – Mean –Square) algorithm

Consider the steepest descent relation

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \frac{\mu}{2}\nabla \mathbf{E}e^2[n]$$

Where

$$\nabla \mathbf{E}e^2[n] = \begin{bmatrix} \frac{\partial Ee^2[n]}{\partial h_0} \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \frac{\partial Ee^2[n]}{\partial h_{M-1}} \end{bmatrix}$$

In the LMS algorithm $Ee^2[n]$ is approximated by $e^2[n]$ to achieve a computationally simple algorithm.

$$\nabla \mathbf{E}e^2[n] \cong 2.e[n].\begin{bmatrix} \frac{\partial e[n]}{\partial h_0} \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \frac{\partial e[n]}{\partial h_{M-1}} \end{bmatrix}$$

Now consider

$$e[n] = x[n] - \sum_{i=0}^{M-1} h_i[n]y[n-i]$$

$$\frac{\partial e[n].}{\partial h_j} = -y[n-j], j = 0,1,\ldots\ldots.M-1$$

$$\therefore \begin{bmatrix} \frac{\partial e[n]}{\partial h_0} \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \\ \frac{\partial e[n]}{\partial h_{M-1}} \end{bmatrix} = -\begin{bmatrix} y[n] \\ y[n-1] \\ \ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots \\ y[n-M+1] \end{bmatrix} = -\mathbf{y}[n]$$

$\therefore \nabla \mathbf{E}\mathbf{e}^{\mathbf{2}}[n] = -\mathbf{2}e[n]\mathbf{y}[n]$

The steepest descent updation now becomes

$$\mathbf{h[n+1]} = \mathbf{h[n]} + \mu e[n]\mathbf{y[n]}$$

This modification is due to Widrow and Hoff and the corresponding adaptive filter is known as the LMS filter.

Hence the LMS algorithm is as follows

Given the input signal $y[n]$, reference signal $x[n]$ and step size μ

Initialization $h_i[0] = 0, i = 0,1,2.....M-1$

For $n > 0$

1. Filter output

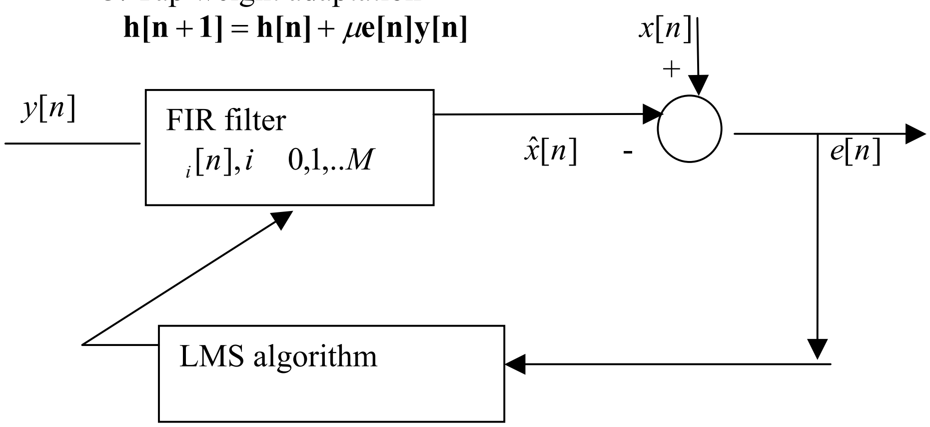
   $\hat{x}[n] = \mathbf{h}'[n]\mathbf{y}[n]]$

2. Estimation of the error

   $e[n] = x[n] - \hat{x}[n]$

3. Tap weight adaptation

   $\mathbf{h[n+1]} = \mathbf{h[n]} + \mu\mathbf{e[n]y[n]}$



Convergence of the LMS algorithm.

As there is a feedback loop in the adaptive algorithm, convergence is generally not assured. The convergence of the algorithm depends on the step size parameter μ.

- The LMS algorithm is convergent in the mean if the step size parameter μ satisfies the condition.

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

$$\mathbf{h[n+1]} = \mathbf{h[n]} + \mu e[n]\mathbf{y[n]}$$

$$\begin{aligned}\therefore E\mathbf{h[n+1]} &= E\mathbf{h[n]} + \mu E e[n]\mathbf{y[n]} \\ &= E\mathbf{h[n]} + \mu E(x[n] - \mathbf{y'[n]}h[\mathbf{n}])\mathbf{y[n]} \\ &= E\mathbf{h[n]} + \mu \mathbf{r_{XY}} - \mu E\mathbf{y}[\mathbf{n}]\mathbf{y'[n]h}[\mathbf{n}]\end{aligned}$$

Assuming the coefficient to be independent of data, (Independence Assumption) we get

$$\begin{aligned}E\mathbf{h[n+1]} &= E\mathbf{h[n]} + \mu \mathbf{r_{XY}} - \mu E\mathbf{y}[\mathbf{n}]\mathbf{y'[n]}E\mathbf{h}[\mathbf{n}] \\ &= E\mathbf{h[n]} + \mu \mathbf{r_{XY}} - \mu \mathbf{R_{XY}} E\mathbf{h}[\mathbf{n}]\end{aligned}$$

Hence the mean value of the filter coefficients satisfy the steepest descent iterative relation so that the same stability condition applies to the mean of the filter coefficients.

- In the practical situation , knowledge of $\lambda_{\max}$ is not available and Trace $\mathbf{R}_{yy}$ can be taken as the conservative estimate of $\lambda_{\max}$ so that for convergence
- $0 < \mu < \dfrac{2}{Trace(\mathbf{R}_{yy})}$
- Also note that trace, $Trace(\mathbf{R}_{yy}) = MR_{yy}[0]$ = Tape input power of the LMS filter.

Generally, a too small value of μ results in slower convergence where as big values of μ will result in larger fluctuations from the mean. Choosing a proper value of μ is very important for the performance of the LMS algorthm.

The average of each filter tap –weight converges to the corresponding optimal filter tap-weight. But this does not ensure that the coefficients converge to the optimal values.

EXCESS MEAN SQUARE ERROR:

$$\mathbf{h[n+1]} = \mathbf{h[n]} + \mu e[n]\mathbf{y[n]}$$

Hence the mean of LMS coefficient converges to the steepest descent solution .But this does not guarantee that the mean square error of the LMS estimator will converges to the mean square error corresponding to the wiener solution . There is a fluctuation of the LMS coefficient from the wiener filter coefficient .

Let $\mathbf{h}_{\mathbf{opt}}$ = optimal wiener filter impulse response.

The instantaneous destination of the LMS coefficient from h opt is

$$\Delta\mathbf{h} = \mathbf{h}[n] - \mathbf{h}_{\mathbf{opt}}$$

$$\begin{aligned}
\varepsilon[n] = Ee^2[n] &= E\{x[n] - \mathbf{h}'_{opt}\mathbf{y}[n] - \Delta\mathbf{h}'\mathbf{y}[n]\}^2 \\
&= E\{x[n] - \mathbf{h}'_{\mathbf{opt}}y[n]\}^2 + E\Delta\mathbf{h}'[n]\mathbf{y[n]y[n]}'\Delta\mathbf{h}[n] - 2E(e_{opt}[n]\Delta\mathbf{h}'\mathbf{[n]y[n]} \\
&= \varepsilon_{\min} + E\Delta\mathbf{h}'[n]\mathbf{y[n]y'[n]}\Delta\mathbf{h}[n] - 2E(e_{opt}[n]\Delta\mathbf{h}'\mathbf{[n]y[n]} \\
&= \varepsilon_{\min} + E\Delta\mathbf{h}'[n]\mathbf{y[n]y'[n]}\Delta\mathbf{h}[n]
\end{aligned}$$

assuming the independence of derivative with respect to data.

Therefore

$$\varepsilon_{excess} = E\Delta\mathbf{h}'[n]\mathbf{y[n]y'[n]}\Delta\mathbf{h}[n]$$

An exact analysis of the excess mean square error is quite complicated and its approximate value is given by

$$\varepsilon_{excess} = \varepsilon_{\min}\frac{\sum_{i=1}^{M}\frac{\mu\lambda_i}{2-\mu\lambda_i}}{1-\sum_{i=1}^{M}\frac{\mu\lambda_i}{2-\mu\lambda_i}}$$

The LMS algorithm is said to converge in the mean square provided the step-length parameter satisfies the relation

$$\mu\sum_{i=1}^{M}\frac{2\lambda}{2-\mu\lambda_i} < 1$$

and $0 < \mu < \frac{2}{\lambda_{\max}}$

If $\sum_{i=1}^{M}\frac{\mu\lambda_i}{2-\mu\lambda_i} << 1$

$$\varepsilon_{excess} = \varepsilon_{\min}\sum_{i=1}^{M}\frac{\mu\lambda_i}{2-\mu\lambda_i}$$

The factor $\frac{\varepsilon_{excess}}{\varepsilon_{\min}} = \sum_{i=1}^{M}\frac{\mu\lambda_i}{2-\mu\lambda_i}$ is called the misadjustment factor.

Leaky LMS Algorithm

Minimizes $e^2[n] + \alpha \|h[n]\|^2$

where $\|h[n]\|$ is the modulus of the LMS weight vector and α a positive quantity.
The corresponding algorithm is given by

$\mathbf{h[n+1]} = (1 - \mu\alpha)\mathbf{h[n]} + \mu\alpha\mathbf{e[n]y[n]}$
where $\mu\alpha$ is chosen to be less than 1. In such a situation the pole will be inside the unit circle, instability problem will not be there and the algorithm will converge.