Chapter 5  Least-mean-square Adaptive Filters

5.2 Least-mean-square Adaptation Algorithm

According to Eq. (4.8), the gradient vector at iteration \( n \), is

\[
\nabla J(n) = -2 \hat{p} + 2 \hat{K} \hat{w}(n).
\]

The simplest gradient vector estimate is the instantaneous estimates for \( \hat{K} \) and \( \hat{p} \) based on the sample values of the tap-input vector and desired response, defined by

\[
\hat{K}(n) = \hat{u}(n) \hat{u}^H(n)
\]

\[
\hat{p}(n) = \hat{u}(n) d^*(n)
\]

The instantaneous estimate of the gradient vector is

\[
\hat{\nabla} J(n) = -2 \hat{u}(n), d^*(n) + 2 \hat{u}(n) \hat{u}^H(n) \hat{w}(n)
\]

Therefore, the recursive update of the weight vector is given by
\[ \hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n) [ \mathbf{d}^*(n) - \frac{\mathbf{u}^H(n) \hat{\mathbf{w}}(n)}{y(n)} ] \]

\[ = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n) \mathbf{e}^*(n), \]

where \( \mathbf{e}(n) = \mathbf{d}(n) - y(n) \),

and \( 0 < \mu < \frac{2}{M S_{\text{max}}} \), \( (M \) is the filter length, \( S_{\text{max}} \) is the maximum value of the input power spectral density).

This LMS algorithm can be applied for both deterministic and random signals \( \mathbf{u}(n) \) and \( \mathbf{d}(n) \). It can track the non-stationary variations as well.

Read 5.3 Applications Yourself!
5.4 Statistical LMS Theory

As discussed in Chapter 4, the weight-error vector in the LMS filter is given by
\[ \hat{E}(n) = \hat{\omega}_o - \hat{\omega}(n), \]
where \( \hat{\omega}_o \) is the optimal tap-weight vector.

According to Eq. (5.5),
\[ \hat{E}(n+1) = \left[ I - \mu \hat{u}(n) \hat{u}^H(n) \right] \hat{E}(n) \]
\[ - \mu \hat{u}(n) e_o^*(n), \]
where \( I \) is the identity matrix and
\[ e_o(n) = d(n) - \hat{\omega}_o^H \hat{u}(n) \]
is the estimation error produced by the optimum filter.

Equation (5.56) is a random difference equation. To study the convergence behavior of such a random algorithm in an average sense, we may invoke the direct-averaging method.
According to this method, the solution of Eq. (5.56), with a small step-size $\mu$, is approximated by the ensemble average,

$$E\{ \hat{\mathbf{I}} - \mu \hat{\mathbf{u}}(n) \hat{\mathbf{u}}^H(n) \} = \hat{\mathbf{I}} - \mu \hat{\mathbf{R}}.$$

Hence,

$$\hat{\mathbf{E}}_0(n+1) = (\hat{\mathbf{I}} - \mu \hat{\mathbf{R}}) \hat{\mathbf{E}}_0(n) - \mu \hat{\mathbf{u}}(n) \hat{\mathbf{e}}_0^*(n),$$

where $\hat{\mathbf{E}}_0(n) \approx \hat{\mathbf{e}}(n)$ for small $\mu$.

\textbf{Butterweck's Iterative Procedure}

The error vector in Eq. (5.56) can be written as a sum of iterative partial error vectors such that

$$\hat{\mathbf{e}}(n) = \hat{\mathbf{e}}_0(n) + \hat{\mathbf{e}}_1(n) + \hat{\mathbf{e}}_2(n) + \ldots,$$

where $\hat{\mathbf{e}}_0(n)$ is the error vector in Eq. (5.58) for $\mu \rightarrow 0$. $\hat{\mathbf{e}}_1(n), \hat{\mathbf{e}}_2(n), \ldots$ are higher-order corrections for $\mu > 0$.

The zero-mean difference matrix is defined as

$$\hat{\mathbf{P}}(n) = \hat{\mathbf{u}}(n) \hat{\mathbf{u}}^H(n) - \hat{\mathbf{R}}.$$
From Eq. (5.59), (5.60) and (5.56), we obtain

\[ \hat{E}_0(n+1) + \hat{E}_1(n+1) + \hat{E}_2(n+2) + \cdots \]

\[ = (I - \mu \hat{R}) \left[ \hat{E}_0(n) + \hat{E}_1(n) + \hat{E}_2(n) + \cdots \right] \]

\[ - \mu \hat{P}(n) \left[ \hat{E}_0(n) + \hat{E}_1(n) + \hat{E}_2(n) + \cdots \right] \]

\[ - \mu \hat{u}(n) E_0^*(n) \]

Hence we can readily deduce the set of coupled difference equations:

\[ \hat{E}_i(n+1) = \left( I - \mu \hat{R} \right) \hat{E}_i(n) + \hat{f}_i(n), \quad i = 0, 1, 2, \ldots \]

where the driving force \( \hat{f}_i(n) \) is defined as

\[ \hat{f}_i(n) = \begin{cases} -\mu \hat{u}(n) E_0^*(n) & \text{for } i = 0 \\ -\mu \hat{P}(n) \hat{E}_{i-1}(n) & \text{for } i = 1, 2, \ldots \end{cases} \]

Eq. (5.61) states that the analysis of the LMS filter is reduced to a study of transmission of a stationary random process through a low-pass filter with an extremely low cut-off frequency as \( \mu \to 0 \).
The correlation matrix of the weight-error vector \( \tilde{E}(n) \) is defined as
\[
\tilde{K}(n) = E\{ \tilde{E}(n) \tilde{E}^H(n) \} = \sum_i \sum_k E\{ \tilde{E}_i(n) \tilde{E}_k^H(n) \}, \quad (i,k) = 0,1,2, \ldots
\]

\[
\tilde{K}(n) = \tilde{K}_0(n) + \mu \tilde{K}_1(n) + \mu^2 \tilde{K}_2(n) + \ldots
\]

where \( \mu \tilde{K}_j(n) = \left\{ \begin{array}{ll}
E[\tilde{E}_0(n) \tilde{E}_0^H(n)], & \text{for } j = 0 \\
\sum_i \sum_k E[\tilde{E}_i(n) \tilde{E}_k^H(n)], & \text{for } (i,k) \geq 0
\end{array} \right.

such that \( i+k = 2j-1, 2j, \ldots \)

**Small-step-size Statistical Theory**

**Assumption I.** The step-size \( \mu \) is small, so the LMS filter acts as a low-pass filter with a low cut-off frequency.

From Eq. (5.58), the \( k \)-th element in \( \tilde{V}^H \tilde{E}(n) \) can be written as
\[
\tilde{E}_{0,k}(n+1) = (1 - \mu \tilde{p}_k) \tilde{E}_{0,k}(n) + \tilde{p}_k \tilde{f}(n)
\]

where \( \tilde{V} \) is the eigenvector matrix of \( \tilde{K} \).
The zero of the low-pass filter specified in the above difference equation is 

\[ z = (1 - \mu \sigma_w^2) \to 1 \quad \text{as} \quad \mu \to 0 \]

Assumption II. The physical mechanism for generating the observable data, i.e., the desired response is described by a linear multiple regression model that is modeled exactly by the Wiener filter, that is 

\[ d(n) = \mathbf{w}^H u(n) + e_0(n) \]

where the irreducible estimation error \( e_0(n) \) is a white process independent of the input \( u(n) \). 

\[
E\left\{ e_0(n) e_0^*(n-k) \right\} = \begin{cases} T_{min} & \text{for } k=0 \\ 0 & \text{for } k \neq 0 \end{cases}
\]

Assumption III. The input vector \( \mathbf{u}(n) \) and the desired response \( d(n) \) are jointly Gaussian.
Natural Modes of the LMS filter

Under Assumption I, Butterweck's iterative procedure reduces to the following equations:

\[ \hat{e}_0(n+1) = (\hat{I} - \mu \hat{R}) \hat{e}_0(n) + \hat{f}_0(n) \]

\[ \hat{f}_0(n) = -\mu u(n) e_0^*(n) \]

(one-zero model of error vector)

Suppose \( \hat{R} = \hat{\Theta} \hat{\Theta}^H \) (eigen-decomposition)

We define \( \hat{\nu}(n) = \hat{\Theta}^H \hat{e}_0(n) \);

then \( \hat{\nu}(n+1) = (\hat{I} - \mu \hat{\Theta}) \hat{\nu}(n) + \hat{\varphi}(n) \),

where \( \hat{\varphi}(n) = \hat{\Theta}^H \hat{f}_0(n) \).

We can easily verify that:

1. The mean of the random force vector \( \hat{\varphi}(n) \) is zero: \( E[\hat{\varphi}(n)] = 0 \), \( \forall n \)

2. The correlation matrix of the random force vector \( \hat{\varphi}(n) \) is diagonal matrix.
\[ E \left[ \hat{\phi}(n) \hat{\phi}^H(n) \right] = \mu^2 \text{J}_{\text{min}} \tilde{\Lambda} \]

where \( \text{J}_{\text{min}} \) is the minimum mean-square error produced by the Wiener filter.

Proof:

1. From Eq. (5.69), \( \hat{\phi}(n) = \hat{\alpha}^H \hat{f}_0(n) \)
   \[ \hat{\alpha}^H = -\mu \hat{\alpha}^H \hat{u}(n) \hat{e}_0^*(n) \]
   \[ E \left[ \hat{\phi}(n) \right] = -\mu \hat{\alpha}^H E \left[ \hat{u}(n) \hat{e}_0^*(n) \right] \]
   \[ = \frac{-\hat{\alpha}^H \hat{\beta}}{\sigma} \]
   due to the principle of the orthogonality.

2. \[ E \left[ \hat{\phi}(n) \hat{\phi}^H(n) \right] = \mu^2 \hat{\alpha}^H E \left[ \hat{u}(n) \hat{e}_0^*(n) \hat{e}_0(n) \hat{u}(n) \right] \]
   \[ = \mu^2 \hat{\alpha}^H E \left[ |\hat{e}_0(n)|^2 \right] E \left[ \hat{u}(n) \hat{u}^H(n) \right] \]
   \[ = \mu^2 \hat{\alpha}^H \text{J}_{\text{min}} \tilde{\Lambda} \hat{\alpha} \]
   \[ = \mu^2 \text{J}_{\text{min}} \tilde{\Lambda} \]
According to Eq. (5.93), the \( k \)th natural mode (the \( k \)th element in \( \vec{v}(n) \)) is written as

\[
U_k(n+1) = (1 - \mu \lambda_k) U_k(n) + \phi_k(n), \quad k = 1, 2, \ldots, M.
\]

The iterative change in the natural mode \( U_k \) is

\[
\Delta U_k(n) = U_k(n+1) - U_k(n) = -\mu \lambda_k U_k(n) + \phi_k(n), \quad k = 1, 2, \ldots, M.
\]

From Eq. (5.98),

\[
U_k(n) = (1 - \mu \lambda_k)^n U_k(0) + \sum_{i=0}^{n-1} (1 - \mu \lambda_k)^{n-1-i} \phi_k(i).
\]

Properties of the natural mode \( U_k(n) \),

\( k = 1, 2, \ldots, M \)

1. Mean value:

\[
E \left\{ U_k(n) \right\} = U_k(0) (1 - \mu \lambda_k)^n.
\]

2. The mean-square value:

\[
E \left\{ |U_k(n)|^2 \right\} = \frac{\mu T_{\text{MIN}}}{2 - \mu \lambda_k} + (1 - \mu \lambda_k)^{2n} (|U_k(0)|^2).
\]
X. Learning Curves

Two kinds of learning curves regarding LMS algorithm can be identified:

1. The mean-square error (MSE) learning curve, which is based on ensemble averaging of the squared estimation error \( |e(n)|^2 \). This learning curve is thus a plot of the MSE,

   \[ J(n) = E \{ |e(n)|^2 \} \]

2. The mean-square deviation (MSD) learning curve, which is based on ensemble averaging of the squared error deviation \( \| \tilde{e}(n) \|_2^2 \). This learning curve is thus a plot of the MSD,

   \[ D(n) = E \{ \| \tilde{e}(n) \|_2^2 \} \]
The estimation error is
\[ e(n) = d(n) - \frac{\mathbf{w}^H(n)}{\mathbf{w}(n)} \mathbf{u}(n) \]
\[ = d(n) - \hat{\mathbf{w}}_0^H \hat{\mathbf{u}}(n) + \hat{\mathbf{w}}_0^H \mathbf{e}(n) \mathbf{u}(n) \]
\[ = e_0(n) + \hat{\mathbf{w}}_0^H \mathbf{e}(n) \mathbf{u}(n) \]
\[ \approx e_0(n) + \hat{\mathbf{w}}_0^H \mathbf{e}(n) \mathbf{u}(n) \quad \text{for small } \mathbf{u}. \]

Hence,
\[ J(n) = E \left\{ | e(n) |^2 \right\} \]
\[ = E \left\{ (e_0(n) + \hat{\mathbf{w}}_0^H \mathbf{e}(n) \mathbf{u}(n)) (e_0^*(n) + \hat{\mathbf{w}}_0^H \mathbf{e}(n) \mathbf{u}(n)) \right\} \]
\[ = J_{\text{min}} + 2 \Re \left\{ E \left[ e_0^*(n) \hat{\mathbf{w}}_0^H \mathbf{e}(n) \hat{\mathbf{u}}(n) \mathbf{u}(n) \right] \right\} + E \left\{ \hat{\mathbf{w}}_0^H \mathbf{e}(n) \mathbf{u}(n) \hat{\mathbf{u}}(n) \mathbf{e}(n) \right\}. \]

Under Assumption II, \( e_0(n) \) is statistically independent of \( \mathbf{u}(n) \) and \( \hat{\mathbf{w}}_0(n) \), so
\[ E \left[ e_0^*(n) \hat{\mathbf{w}}_0^H(n) \mathbf{u}(n) \right] = 0. \]
Under Assumption I, the variations of the vector $\tilde{e}_0(n)$ are slow compared with those of the input $\tilde{u}(n)$.

Hence

\[ E \left[ \tilde{e}_0^H(n) \tilde{u}(n) \tilde{u}^H(n) \tilde{e}_0(n) \right] \]

\[ \approx E \left\{ \tilde{e}_0^H(n) E \left[ \tilde{u}(n) \tilde{u}^H(n) \right] \tilde{e}_0(n) \right\} \]

\[ = E \left[ \tilde{e}_0^H(n) \tilde{R} \tilde{e}_0(n) \right] \]

\[ = \text{tr} \left\{ E \left[ \tilde{e}_0^H(n) \tilde{R} \tilde{e}_0(n) \right] \right\} \]

\[ = E \left\{ \text{tr} \left[ \tilde{R} \tilde{e}_0(n) \tilde{e}_0^H(n) \right] \right\} \]

\[ = \text{tr} \left\{ \tilde{R} E \left[ \tilde{e}_0(n) \tilde{e}_0^H(n) \right] \right\} \]

\[ = \text{tr} \left[ \tilde{R} \tilde{K}_0(n) \right] \tilde{K}_0(n) \]

Therefore, $J(n) = J_{\text{min}} + \text{tr} \left[ \tilde{R} \tilde{K}_0(n) \right]$

The excess mean-square error can be written as

\[ J_{\text{ex}}(n) = J(n) - J_{\text{min}} \]

\[ \approx \text{tr} \left[ \tilde{R} \tilde{K}_0(n) \right] \]
\[ J_{\text{ex}}(n) \approx \text{tr} \{ \mathbf{R} E [ \hat{\mathbf{E}}_e(n) \hat{\mathbf{E}}_e^H(n) ] \} \]
\[ = \text{tr} \{ \mathbf{R} E [ \hat{\mathbf{V}}(n) \hat{\mathbf{V}}^H(n) \hat{\mathbf{V}}^H(n) ] \} \]
\[ = E \{ \text{tr} [ \mathbf{R} \hat{\mathbf{V}}(n) \hat{\mathbf{V}}^H(n) ] \} \]
\[ = E \{ \text{tr} [ \hat{\mathbf{V}}^H(n) \hat{\mathbf{V}}^H(n) ] \} \]
\[ = E \{ \text{tr} [ \hat{\mathbf{V}}(n) \hat{\mathbf{V}}(n) ] \} \]
\[ = \sum_{k=1}^{M} \lambda_k E [ |U_k(n)|^2 ] \]

According to Assumption I, Eqs. (5.71) and (5.84)
\[ D(n) \approx E [ \| \hat{\mathbf{E}}_e(n) \|^2 ] \]
\[ = E [ \| \hat{\mathbf{V}}(n) \|^2 ] \]
\[ = \sum_{k=1}^{M} E [ |U_k(n)|^2 ] \]

If we denote \( \lambda_{\text{min}} \) and \( \lambda_{\text{max}} \) as the minimum and maximum eigenvalues of \( \mathbf{R} \),
\[ \lambda_{\text{min}} \leq \lambda_k \leq \lambda_{\text{max}}, \ k=1,2, \ldots, M \]

From Eqs. (5.91) and (5.92), the MSD can be bounded by
\[ \lambda_{\text{min}} D(n) \leq J_{\text{ex}}(n) \leq \lambda_{\text{max}} D(n), \forall n \]
\[ \frac{J_{\text{eq}}(n)}{\lambda_{\text{min}}} \geq D(n) \geq \frac{J_{\text{eq}}(n)}{\lambda_{\text{max}}}, \quad \forall n. \]

\section*{X: Transient Behavior and Convergence Considerations}

According to Eqs. (5.82), (5.90), (5.91), we have

\[ J(n) = J_{\text{min}} + \mu J_{\text{min}} \sum_{k=1}^{M} \frac{\lambda_k}{2-\mu \lambda_k} \]

\[ + \sum_{k=1}^{M} \lambda_k \left( 1 |v_k(0)|^2 - \frac{\mu J_{\text{min}}}{2-\mu \lambda_k} \right) (1-\mu \lambda_k)^n \]

\[ \approx J_{\text{min}} + \frac{\mu J_{\text{min}}}{2} \sum_{k=1}^{M} \lambda_k \]

\[ + \sum_{k=1}^{M} \lambda_k \left( 1 |v_k(0)|^2 - \frac{\mu J_{\text{min}}}{2} \right) (1-\mu \lambda_k)^n, \]

for small \( \mu \).

\[ \lim_{n \to \infty} J(n) = J(\infty) = J_{\text{min}} + \mu J_{\text{min}} \sum_{k=1}^{M} \frac{\lambda_k}{2-\mu \lambda_k} \]

\[ \approx J_{\text{min}} + \frac{\mu J_{\text{min}}}{2} \sum_{k=1}^{M} \lambda_k, \quad \text{for small } \mu. \]

\[ + J_{\text{min}} \]
A new parameter, the misadjustment, can be defined as

$$M = \frac{J_{\text{ex}}(\infty)}{J_{\text{min}}}$$

$$= \frac{J(\infty) - J_{\text{min}}}{J_{\text{min}}}$$

$$= \frac{\mu}{2} \sum_{k=1}^{M} \lambda_k$$

where $M$ is dimensionless and it provides a measure of how close the LMS algorithm to the optimality in the mean square sense.

$$M = \frac{\mu}{2} \text{tr} \left[ \hat{R} \right]$$

$$= \frac{\mu}{2} \text{tr} [ R(0) ]$$

$$= \frac{\mu}{2} \mathbb{E} \left[ \| \hat{u}(n) \|^2 \right]$$

$$= \frac{\mu}{2} \times (\text{total tap-weight power})$$
The average eigenvalue of \( \mathbf{R} \) is defined as 

\[
\lambda_{av} = \frac{1}{M} \sum_{k=1}^{M} \lambda_k .
\]

Suppose that the ensemble-average learning curve of LMS algorithm is approximated by a single exponential with time constant \( T_{mse, av} \). According to Eq. (4.32), the average time constant for LMS algorithm is defined as 

\[
T_{mse, av} = \frac{1}{2 \mu \lambda_{av}} .
\]

According to Eq. (5.103), (5.103) & (5.104), the misadjustment can be approximated by

\[
M = \frac{\mu M T_{av}}{2} = \frac{M}{4 T_{mse, av}} .
\]
Remarks:

1. The misadjustment $M$ increases linearly with the filter length $M$ for a fixed time $\tau$.

2. The settling time of the LMS algorithm (i.e., the time taken for the transients to die out) is proportional to $T_{\text{settle}}$ and is inversely proportional to $M$. Therefore, there is a trade-off, in that if $M$ is reduced so as to reduce the misadjustment, then the settling time (time to converge) of the LMS algorithm is increased. Conversely, if $M$ is increased so as to reduce the settling time (faster convergence), then the misadjustment is increased.

Read 5.6 - 5.9 yourselves!!