Stochastic gradient based adaptation: Least Mean Square (LMS) Algorithm

LMS algorithm derivation based on the Steepest descent (SD) algorithm

Steepest descent search algorithm (from last lecture)

Given
- the autocorrelation matrix $R = E\underline{u}(n)\underline{u}^T(n)$
- the cross-correlation vector $\underline{p}(n) = E\underline{u}(n)d(n)$

Initialize the algorithm with an arbitrary parameter vector $\underline{w}(0)$.
Iterate for $n = 0, 1, 2, 3, \ldots, n_{max}$

$$\underline{w}(n+1) = \underline{w}(n) + \mu[\underline{p} - R\underline{w}(n)] \qquad \text{(Equation } \mathbf{SD} - \underline{p}, R\text{)}$$

We have shown that adaptation equation ($\mathbf{SD} - \underline{p}, R$) can be written in an equivalent form as (see also the Figure with the implementation of SD algorithm)

$$\underline{w}(n+1) = \underline{w}(n) + \mu[Ee(n)\underline{u}(n)] \qquad \text{(Equation } \mathbf{SD} - \underline{u}, e\text{)}$$

In order to simplify the algorithm, instead the true gradient of the criterion

$$\nabla_{\underline{w}(n)} J(n) = -2E\underline{u}(n)e(n)$$

LMS algorithm will use an immediately available approximation

$$\hat{\nabla}_{\underline{w}(n)} J(n) = -2\underline{u}(n)e(n)$$

Using **the noisy gradient**, the adaptation will carry on the equation

$$\underline{w}(n+1) = \underline{w}(n) - \frac{1}{2}\mu\hat{\nabla}_{\underline{w}(n)}J(n) = \underline{w}(n) + \mu\underline{u}(n)e(n)$$

In order to gain new information at each time instant about the gradient estimate, the procedure will go through all data set $\{(d(1), u(1)), (d(2), u(2)), \ldots\}$, many times if needed.

LMS algorithm

Given
- the (correlated) input signal samples $\{u(1), u(2), u(3), \ldots\}$, generated randomly;
- the desired signal samples $\{d(1), d(2), d(3), \ldots\}$ correlated with $\{u(1), u(2), u(3), \ldots\}$

1 Initialize the algorithm with an arbitrary parameter vector $\underline{w}(0)$, for example $\underline{w}(0) = 0$.

2 Iterate for $n = 0, 1, 2, 3, \ldots, n_{max}$

2.0	Read /generate a new data pair,	$(\underline{u}(n), d(n))$
2.1	(Filter output)	$y(n) = \underline{w}(n)^T\underline{u}(n) = \sum_{i=0}^{M-1} w_i(n)u(n-i)$
2.2	(Output error)	$e(n) = d(n) - y(n)$
2.3	(Parameter adaptation)	$\underline{w}(n+1) = \underline{w}(n) + \mu\underline{u}(n)e(n)$

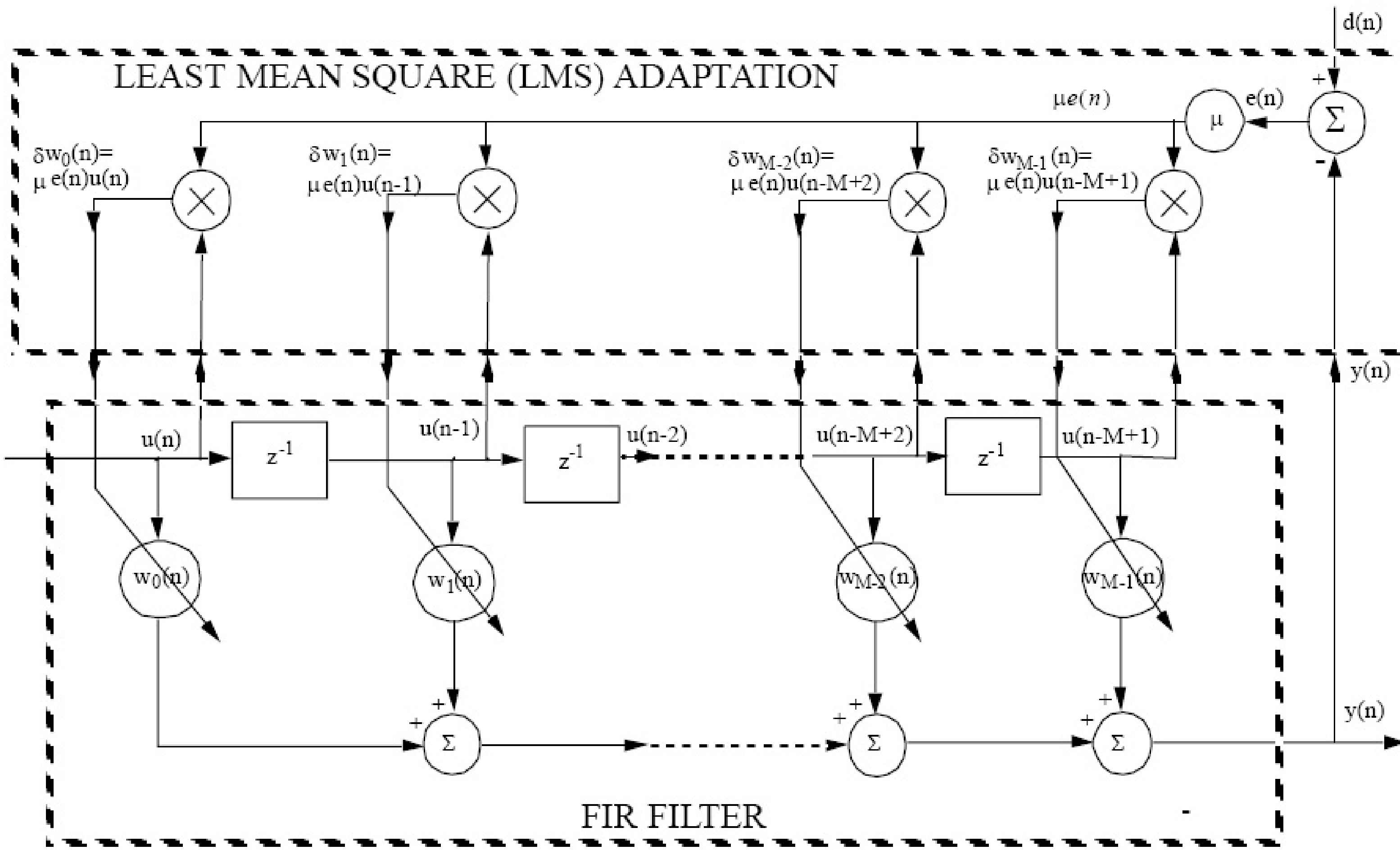
or componentwise

$$\begin{bmatrix} w_0(n+1) \\ w_1(n+1) \\ . \\ . \\ . \\ w_{M-1}(n+1) \end{bmatrix} = \begin{bmatrix} w_0(n) \\ w_1(n) \\ . \\ . \\ . \\ w_{M-1}(n) \end{bmatrix} + \mu e(n) \begin{bmatrix} u(n) \\ u(n-1) \\ . \\ . \\ . \\ u(n-M+1) \end{bmatrix}$$

□

The complexity of the algorithm is $2M + 1$ multiplications and $2M$ additions per iteration.

Schematic view of LMS algorithm



Stability analysis of LMS algorithm

SD algorithm is guaranteed to converge to Wiener optimal filter if the value of μ is selected properly (see last Lecture)

$$\underline{w}(n) \to \underline{w}_o$$

$$J(\underline{w}(n)) \to J(\underline{w}_o)$$

The iterations are deterministic : starting from a given $\underline{w}(0)$, all the iterations $\underline{w}(n)$ are perfectly determined.

LMS iterations are not deterministic: the values $w(n)$ depends on the realization of the data $d(1), \ldots, d(n)$ and $u(1), \ldots, u(n)$. *Thus, $\underline{w}(n)$ is now a random variable.*

The convergence of LMS can be analyzed from following perspectives:

- Convergence of parameters $\underline{w}(n)$ in the mean:

$$E\underline{w}(n) \to \underline{w}_o$$

- Convergence of the criterion $J(\underline{w}(n))$ (in the mean square of the error)

$$\underline{J}(\underline{w}(n)) \to \underline{J}(\underline{w}_\infty)$$

Assumptions (needed for mathematical tractability) = Independence theory

1. The input vectors $\underline{u}(1), \underline{u}(2), \ldots, \underline{u}(n)$ are statistically independent vectors (very strong requirement: even white noise sequences don't obey this property);

2. the vector $\underline{u}(n)$ is statistically independent of all $d(1), d(2), \ldots, d(n-1)$
3. The desired response $d(n)$ is dependent on $\underline{u}(n)$ but independent on $d(1), \ldots, d(n-1)$.
4. The input vector $\underline{u}(n)$ and desired response $d(n)$ consists of mutually Gaussian-distributed random variables.

Two implications are important:

* $\underline{w}(n+1)$ is statistically independent of $d(n+1)$ and $\underline{u}(n+1)$

* The Gaussion distribution assumption (Assumption 4) combines with the *independence* assumptions 1 and 2 to give *uncorrelated*-ness statements

$$E\underline{u}(n)\underline{u}(k)^T = 0, \quad k = 0, 1, 2, \ldots, n-1$$

$$E\underline{u}(n)d(k) = 0, \quad k = 0, 1, 2, \ldots, n-1$$

Convergence of average parameter vector $E\underline{w}(n)$

We will subtract from the adaptation equation

$$\underline{w}(n+1) = \underline{w}(n) + \mu\underline{u}(n)e(n) = \underline{w}(n) + \mu\underline{u}(n)(d(n) - \underline{w}(n)^T\underline{u}(n))$$

the vector $\underline{w}_o$ and we will denote $\underline{\varepsilon}(n) = \underline{w}(n) - \underline{w}_o$

$$\begin{aligned}
&\underline{w}(n+1) - \underline{w}_o = \underline{w}(n) - \underline{w}_o + \mu\underline{u}(n)(d(n) - \underline{w}(n)^T\underline{u}(n)) \\
&\underline{\varepsilon}(n+1) = \underline{\varepsilon}(n) + \mu\underline{u}(n)(d(n) - \underline{w}_o^T\underline{u}(n)) + \mu\underline{u}(n)(\underline{u}(n)^T\underline{w}_o - \underline{u}(n)^T\underline{w}(n)) \\
&\quad = \underline{\varepsilon}(n) + \mu\underline{u}(n)e_o(n) - \mu\underline{u}(n)\underline{u}(n)^T\underline{\varepsilon}(n) = (I - \mu\underline{u}(n)\underline{u}(n)^T)\underline{\varepsilon}(n) + \mu\underline{u}(n)e_o(n)
\end{aligned}$$

Taking the expectation of $\underline{\varepsilon}(n+1)$ using the last equality we obtain

$$E\underline{\varepsilon}(n+1) \;=\; E(I-\mu\underline{u}(n)\underline{u}(n)^T)\underline{\varepsilon}(n) + E\mu\underline{u}(n)e_o(n)$$

and now using the statistical independence of $\underline{u}(n)$ and $\underline{w}(n)$, which implies the statistical independence of $\underline{u}(n)$ and $\underline{\varepsilon}(n)$,

$$E\underline{\varepsilon}(n+1) \;=\; (I-\mu E[\underline{u}(n)\underline{u}(n)^T])E[\underline{\varepsilon}(n)] + \mu E[\underline{u}(n)e_o(n)]$$

Using the principle of orthogonality which states that $E[\underline{u}(n)e_o(n)] = 0$, the last equation becomes

$$E[\underline{\varepsilon}(n+1)] \;=\; (I-\mu E[\underline{u}(n)\underline{u}(n)^T])E[\underline{\varepsilon}(n)] = (I-\mu R)E[\underline{\varepsilon}(n)]$$

Reminding the equation

$$\underline{c}(n+1) = (I-\mu R)\underline{c}(n) \tag{1}$$

which was used in the analysis of SD algorithm stability, and identifying now $\underline{c}(n)$ with $E\underline{\varepsilon}(n)$, we have the following result:

> The mean $E\underline{\varepsilon}(n)$ converges to zero, and consequently $E\underline{w}(n)$ converges to $\underline{w}_o$
>
> iff
>
> $0 < \mu < \dfrac{2}{\lambda_{max}}$ $\quad (STABILITY CONDITION!)$ where λ_{max} is the largest eigenvalue of the matrix $R = E[\underline{u}(n)\underline{u}(n)^T]$.

Stated in words, LMS is convergent in mean, iff the stability condition is met.

The convergence property explains the behavior of the first order characterization of $\underline{\varepsilon}(n) = \underline{w}(n) - \underline{w}_o$.

Now we start studying **the second order characterization** of $\underline{\varepsilon}(n)$.

$$\underline{\varepsilon}(n+1) \;=\; \underline{\varepsilon}(n) + \mu \underline{u}(n) e(n) = \underline{\varepsilon}(n) - \frac{1}{2}\mu \hat{\nabla} J(n)$$

Now we split $\hat{\nabla} J(n)$ in two terms: $\hat{\nabla} J(n) = \nabla J(n) + 2\underline{N}(n)$ where $\underline{N}(n)$ is the gradient noise. Obviously $E[\underline{N}(n)] = 0$

$$\begin{aligned}
\underline{u}(n)e(n) &= -\frac{1}{2}\hat{\nabla} J(n) = -\frac{1}{2}\nabla J(n) - \underline{N}(n) = -(R\underline{w}(n) - \underline{p}) - \underline{N}(n) \\
&= -R(\underline{w}(n) - \underline{w}_o) - \underline{N}(n) = -R\underline{\varepsilon}(n) - \underline{N}(n)
\end{aligned}$$

$$\begin{aligned}
\underline{\varepsilon}(n+1) &= \underline{\varepsilon}(n) + \mu \underline{u}(n)e(n) = \underline{\varepsilon}(n) - \mu R \underline{\varepsilon}(n) - \mu \underline{N}(n) \\
&= (I - \mu R)\underline{\varepsilon}(n) - \mu \underline{N}(n) = (I - Q\Lambda Q^H)\underline{\varepsilon}(n) - \mu \underline{N}(n) = Q(I - \mu\Lambda)Q^H \varepsilon(n) - \mu \underline{N}(n)
\end{aligned}$$

We denote $\underline{\varepsilon}'(n) = Q^H \underline{\varepsilon}(n)$ and $\underline{N}'(n) = Q^H \underline{N}(n)$ the rotated vectors (remember, Q is the matrix formed by the eigenvectors of matrix R) and we thus obtain

$$\underline{\varepsilon}'(n+1) \;=\; (I - \mu\Lambda)\underline{\varepsilon}'(n) - \mu \underline{N}'(n)$$

or written componentwise

$$\varepsilon'_j(n+1) \;=\; (1 - \mu\lambda_j)\varepsilon'_j(n) - \mu N'_j(n)$$

Taking the modulus and then taking the expectation in both members:

$$E|\varepsilon_j'(n+1)|^2 \;=\; (1-\mu\lambda_j)^2 E|\varepsilon_j'(n)|^2 - 2\mu(1-\mu\lambda_j)E[N_j'(n)\varepsilon_j'(n)] + \mu^2 E[|N_j'(n)|]^2$$

Making the assumption: $E[N_j'(n)\varepsilon_j'(n)] = 0$ and denoting

$$\gamma_j(n) = E[|\varepsilon_j'(n)|]^2$$

we obtain the recursion showing how $\gamma_j(n)$ propagates through time.

$$\gamma_j(n+1) \;=\; (1-\mu\lambda_j)^2\gamma_j(n) + \mu^2 E[|N_j'(n)|]^2$$

More information can be obtained if we assume that the algorithm is in the steady-state, and therefore $\nabla J(n)$ is close to 0. Then

$$e(n)\underline{u}(n) = -\underline{N}(n)$$

i.e. the adaptation vector used in LMS is only noise. Then

$$E[\underline{N}(n)\underline{N}(n)^T] = E[e^2(n)\underline{u}(n)\underline{u}(n)^T] \approx E[e^2(n)]E[\underline{u}(n)\underline{u}(n)^T] = J_o R = J_o Q\Lambda Q^H$$

and therefore

$$E[\underline{N}'(n)\underline{N}'(n)^H] = J_o\Lambda$$

or componentwise

$$E[|N_j'(n)|]^2 = J_o\lambda_j$$

and finally

$$\gamma_j(n+1) \;=\; (1-\mu\lambda_j)^2\gamma_j(n) + \mu^2 J_o\lambda_j$$

We can iterate this to obtain

$$\gamma_j(n+1) \;=\; (1-\mu\lambda_j)^{2n}\gamma_j(0) + \mu^2 \sum_{i=0}^{n/2} (1-\mu\lambda_j)^{2i} J_o \lambda_j$$

Using the assumption $|1-\mu\lambda_j| < 1$(which is also required for convergence in mean) at the limit

$$\boxed{\lim_{n\to\infty} \gamma_j(n) \;=\; \mu^2 J_o \lambda_j \frac{1}{1-(1-\mu\lambda_j)^2} = \frac{\mu J_o}{2-\mu\lambda_j}}$$

This relation gives an estimate of the variance of the elements of $Q^H(\underline{w}(n)-\underline{w}_o)$ vector. Since this variance converges to a nonzero value, it results that the parameter vector $\underline{w}(n)$ continues to fluctuate around the optimal vector $\underline{w}_o$. In Lecture 2 we obtained the canonical form of the quadratic form which expresses the mean square error:

$$J(n) \;=\; J_o + \sum_{i=1}^{M} \lambda_i |\nu_i|^2$$

where $\underline{\nu}$ was defined as

$$\underline{\nu}(n) = Q^H \underline{c}(n) = Q^H(\underline{w}(n)-\underline{w}_o) \tag{2}$$

Similarly it can be shown that in the case of LMS adaptation, and using the independence assumption,

$$J(n) \;=\; J_o + \sum_{i=1}^{M} \lambda_i E|\varepsilon_i'|^2 = J_o + \sum_{i=1}^{M} \lambda_i \gamma_i(n)$$

and defining the criterion J_∞ as the value of criterion $J(\underline{w}(n)) = J(n)$ when $n \to \infty$ we obtain

$$J_\infty \;=\; J_o + \sum_{i=1}^{M} \lambda_i \frac{\mu J_o}{2 - \mu\lambda_i}$$

For $\mu\lambda_i \ll 2$

$$J_\infty \;=\; J_o + \mu J_o \sum_{i=1}^{M} \lambda_i/2 = J_o(1 + \mu \sum_{i=1}^{M} \lambda_i/2) = J_o(1 + \mu tr(R)/2) = \tag{3}$$

$$\boxed{J_\infty \;=\; J_o(1 + \mu M r(0)/2) = J_o(1 + \frac{\mu M}{2} \cdot \text{Power of the input}) \;\; (4)}$$

The steady state mean square error J_∞ is close to optimal mean square error if μ is small enough.

In [Haykin 1991] there was a more complex analysis, involving the transient analysis of $J(n)$. It showed that convergence in mean square sense can be obtained if

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

or in another, simplified, form

$$\mu < \frac{2}{\text{Power of the input}}$$

Small Step Size Statistical Theory (Haykin 2002)

- **Assumption 1** The step size parameter μ is small, so the LMS acts as a low pass filter with a low cutoff frequency.

  It allows to approximate the equation

  $$\underline{\varepsilon}(n+1) = (I - \mu\underline{u}(n)\underline{u}(n)^T)\underline{\varepsilon}(n) - \mu\underline{u}(n)e_o(n) \tag{5}$$

  by an approximation:

  $$\underline{\varepsilon}_o(n+1) = (I - \mu R)\underline{\varepsilon}_o(n) - \mu\underline{u}(n)e_o(n) \tag{6}$$

- **Assumption 2** The physical mecanism for generating the desired response $d(n)$ has the same form as the adaptive filter $d(n) = \underline{w}_o^T\underline{u}(n) + e_o(n)$ where $e_o(n)$ is a white noise, statistically independent of $\underline{u}(n)$.

- **Assumption 3** The input vector $u(n)$ and the desired response $d(n)$ are jointly Gaussian.

Learning curves

The statistical performance of adaptive filters is studied using learning curves, averaged over many realizations, or ensemble-averaged.

- **The mean square error MSE learning curve** Take an ensemble average of the squared estimation error $e(n)^2$

$$J(n) = Ee^2(n) \tag{7}$$

- **The mean-square deviation (MSD) learning curve** Take an ensemble average of the squared error deviation $||\varepsilon(n)||^2$

$$\mathcal{D}(n) = E||\varepsilon(n)||^2 \tag{8}$$

- **The excess mean-square-error**

$$J_{ex}(n) = J(n) - J_{min} \tag{9}$$

where J_{min} is the MSE error of the optimal Wiener filter.

Results of the small step size theory

The statistical performance of adaptive filters is studied using learning curves, averaged over many realizations, or ensemble-averaged.

- **Connection between MSE and MSD**

$$\lambda_{min}\mathcal{D}(n) \leq J_{ex}(n) \leq \lambda_{max}\mathcal{D}(n) \tag{10}$$

$$J_{ex}(n)/\lambda_{min} \leq \mathcal{D}(n) \leq J_{ex}(n)/\lambda_{max} \tag{11}$$

It is therefore enough to study the transient behavior of $J_{ex}(n)$, since $\mathcal{D}(n)$ follows its evolutions.

- **The condition for stability**

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{12}$$

- **The excess mean-square-error** converges to

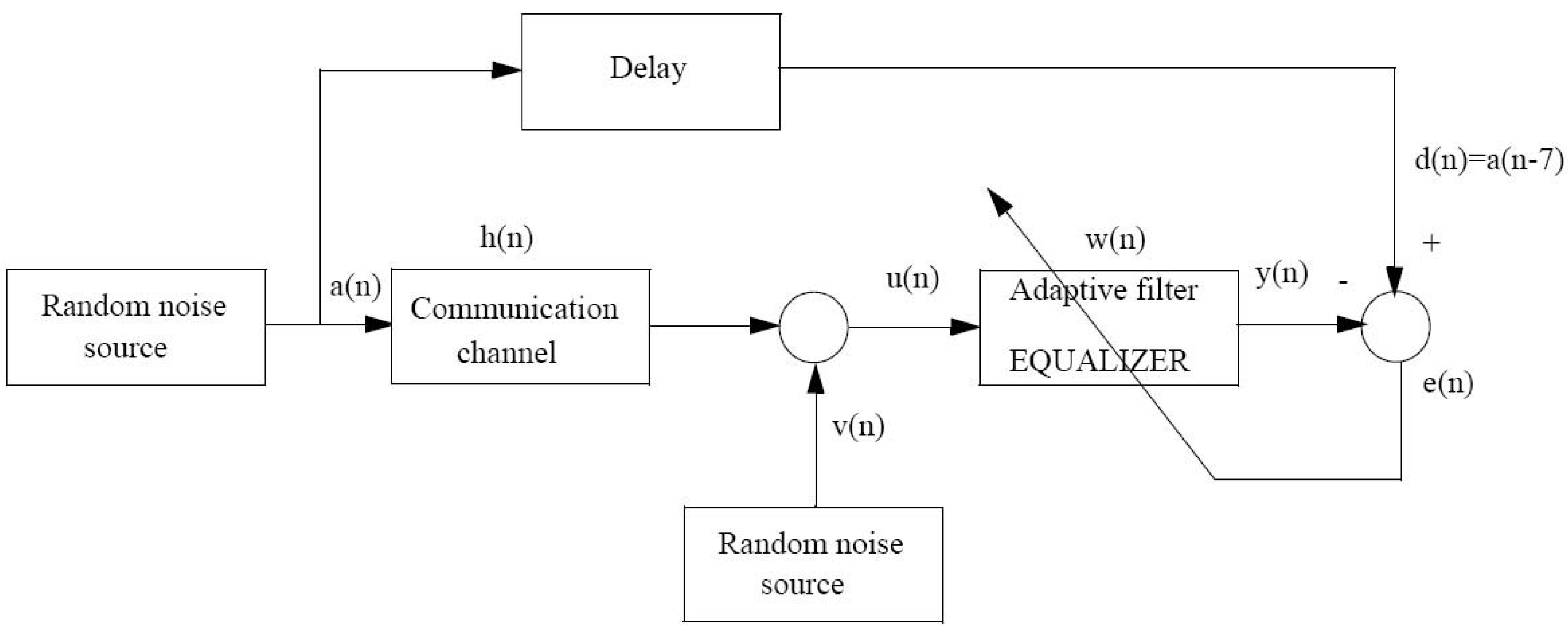
$$J_{ex}(\infty) = \frac{\mu J_{min}}{2}\sum_{k=1}^{M}\lambda_k \tag{13}$$

- **The misadjustment**

$$\mathcal{M} = \frac{J_{ex}(\infty)}{J_{min}} = \frac{\mu}{2}\sum_{k=1}^{M}\lambda_k = \frac{\mu}{2}tr(R) = \frac{\mu}{2}Mr(0) \tag{14}$$

Application of LMS algorithm : Adaptive Equalization

Block diagram of adaptive equalizer experiment



Modelling the communication channel

We assume the impulse response of the channel in the form

$$h(n) = \begin{cases} \frac{1}{2}\left[1 + cos(\frac{2\pi}{W}(n-2))\right], & n = 1, 2, 3 \\ 0, & otherwise \end{cases}$$

The filter input signal will be

$$u(n) = (h * a)(n) = \sum_{k=1}^{3} h(k)a(n-k) + v(n)$$

where $\sigma_v^2 = 0.001$

Selecting the filter structure

The filter has $M = 11$ delays units (taps).

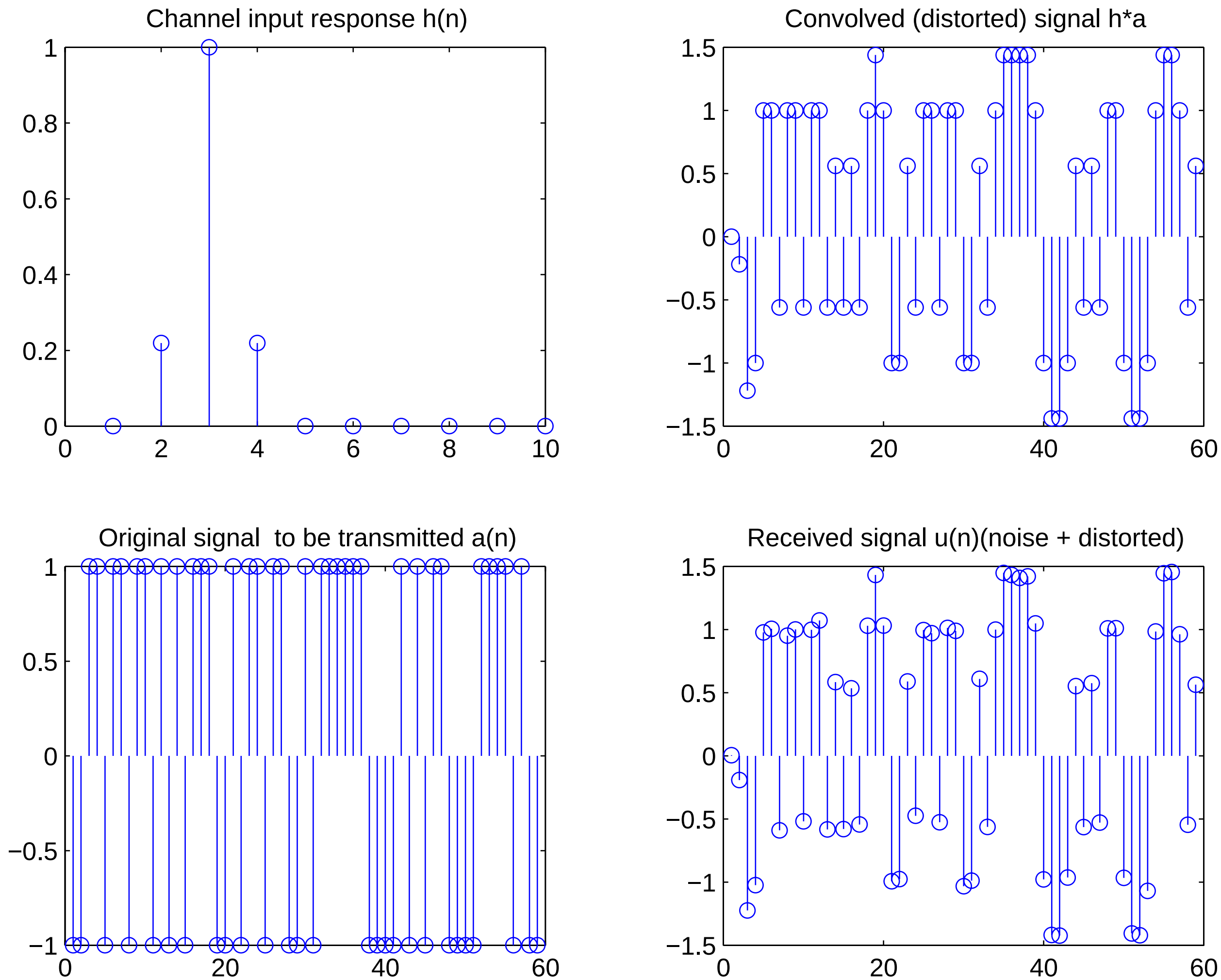
The weights (parameter) of the filter are symmetric with respect to the middle tap ($n = 5$).

The channel input is delayed 7 units to provide the desired response to the equalizer.

Correlation matrix of the Equalizer input

Since $u(n) = \sum_{k=1}^{3} h(k)a(n-k) + v(n)$ is a MA process, the correlation function will be

$$\begin{aligned} r(0) &= h(1)^2 + h(2)^2 + h(3)^2 \\ r(1) &= h(1)h(2) + h(2)h(3) \\ r(2) &= h(1)h(3) \\ r(3) = r(4) &= \ldots = 0 \end{aligned}$$



Sample signals in adaptive equalizer experiment

Effect of the parameter W on the eigenvalue spread

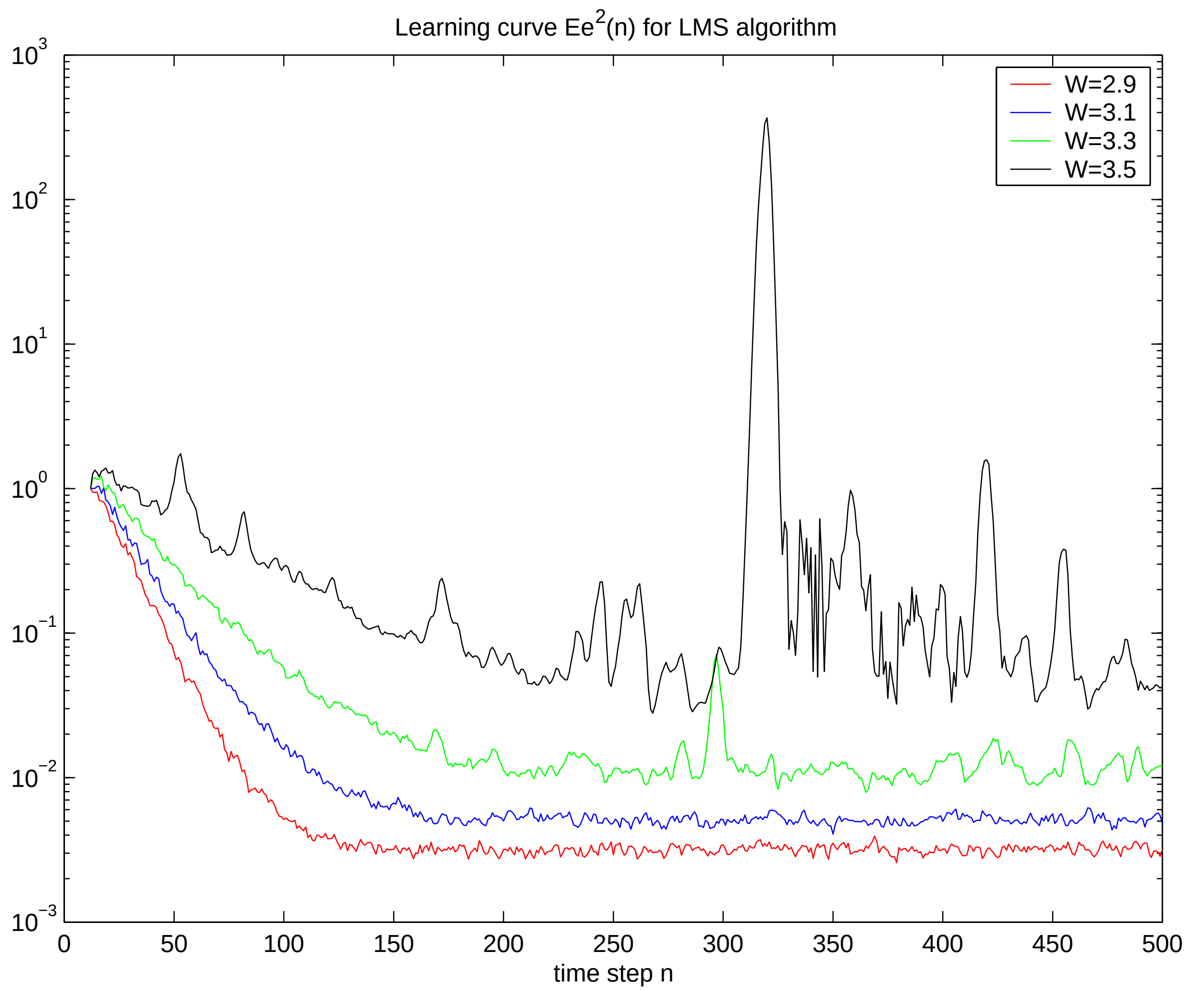
We define the eigenvalue spread $\chi(R)$ of a matrix as the ratio of the maximum eigenvalue over the minimum eigenvalue

W	2.9	3.1	3.3	3.5
$r(0)$	1.0973	1.1576	1.2274	1.3032
$r(1)$	0.4388	0.5596	0.6729	0.7775
$r(2)$	0.0481	0.0783	0.1132	0.1511
λ_{min}	0.3339	0.2136	0.1256	0.0656
λ_{max}	2.0295	2.3761	2.7263	3.0707
$\chi(R) = \lambda_{max}/\lambda_{min}$	6.0782	11.1238	21.7132	46.8216

Experiment 1: Effect of eigenvalue spread

- The step size was kept constant at $\mu = 0.075$
- The eigenvalue spread were taken $[6.0782 \;\; 11.1238 \;\; 21.7132 \;\; 46.8216]$(see previous table), thus varying in a wide range
- for small eigenvalue spread, $\chi(R) = 6.07$, the convergence is the fastest, and the best steady state average squared error. The convergence time is about 80 iterations. The steady state average squared error is about 0.003.
- for small eigenvalue spread, $\chi(R) = 46.8$, the convergence is the slowest, and the worst steady state average squared error. The convergence time is about 200 iterations. The steady state average squared error is about 0.04.

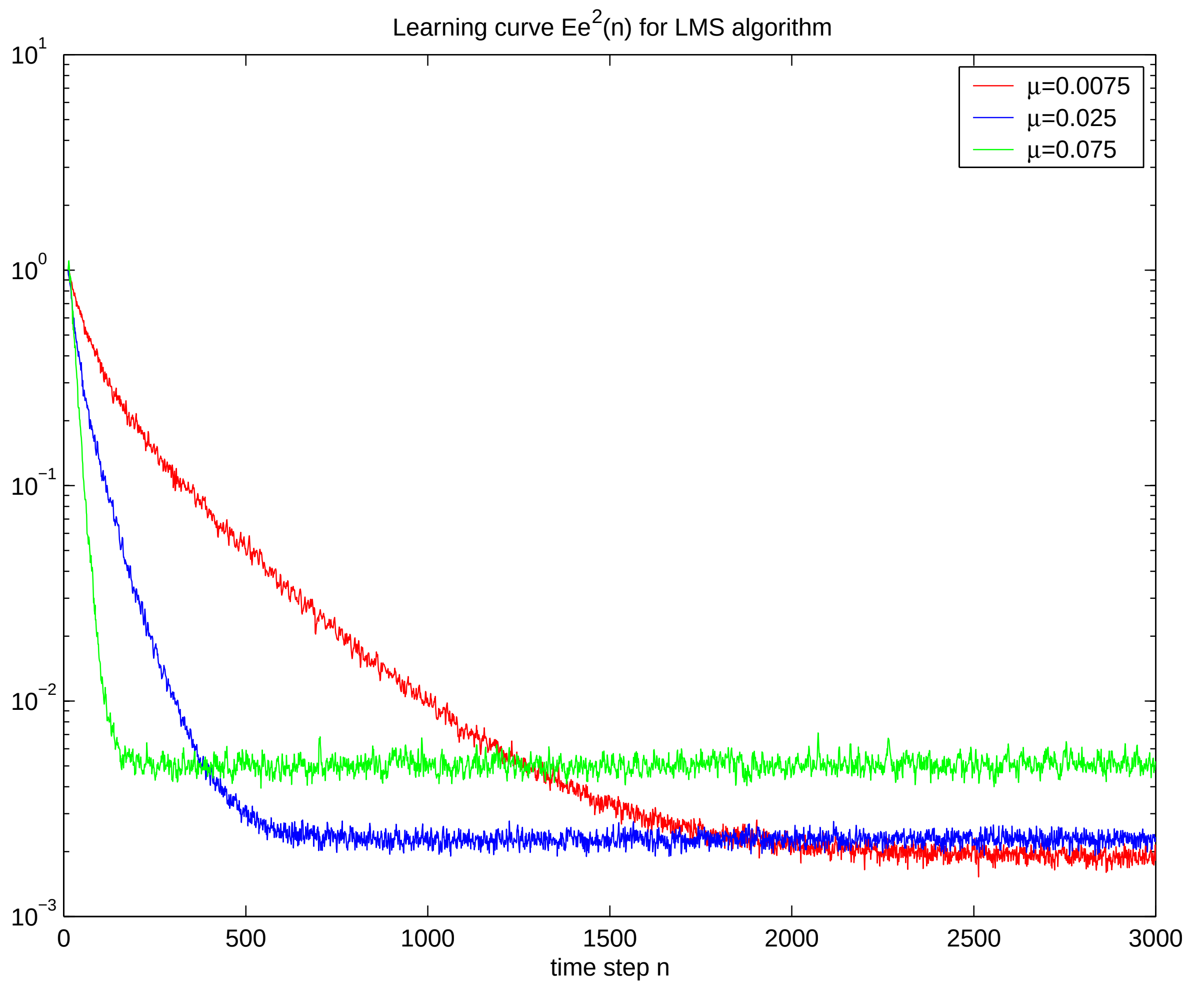
Learning curves for $\mu = 0.075, W = [2.9\ 3.1\ 3.3\ 3.5]$



Experiment 2: Effect of step size

- The eigenvalue spread was kept constant at $\chi = 11.12$
- The step size were taken [0.0075 0.025 0.075], thus varying in a range1:10
- for smallest step sizes, $\mu = 0.0075$, the convergence is the slowest, and the best steady state average squared error. The convergence time is about 2300 iterations. The steady state average squared error is about 0.001.
- for large step size, $\mu = 0.075$, the convergence is the fastest, and the worst steady state average squared error. The convergence time is about 100 iterations. The steady state average squared error is about 0.005.

Learning curves for $\mu = [0.0075\ 0.025\ 0.075]; W = 3.1$



```
% Adaptive equalization

      % Simulate some (useful) signal to be transmitted
 a= (randn(500,1)>0) *2-1;    %   Random bipolar (-1,1) sequence;

      % CHANNEL MODEL
 W=2.9;
 h= [ 0,    0.5 * (1+cos(2*pi/W*(-1:1)))  ];
 ah=conv(h,a);
 v= sqrt(0.001)*randn(500,1); %   Gaussian noise with variance 0.001;
 u=ah(1:500)+v;
 subplot(221) , stem(impz(h,1,10)), title('Channel input response h(n)')
 subplot(222) , stem(ah(1:59)), title('Convolved (distorted) signal h*a')
 subplot(223) , stem(a(1:59)), title('Original signal  to be transmitted')
 subplot(224) , stem(u(1:59)), title('Received signal (noise + distortion)')


     % Deterministic design of equalizer (known h(n))
```

```
H=diag(h(1)*ones(9,1),0) +diag(h(2)*ones(8,1),-1) + diag(h(3)*ones(7,1),-2);
H=[H ; zeros(1,7) h(2) h(3); zeros(1,8) h(3) ]
b=zeros(11,1); b(6)=1;
c=H\b

   % Independent trials N=200
  average_J= zeros(500,1);N=2
  for trial=1:N
  v= sqrt(0.001)*randn(500,1); %   Gaussian noise with variance 0.001;
  u=ah(1:500)+v;

   % Statistical design of equalizer (unkown h(n))
      mu= 0.075;
      w=zeros(11,1);
      for i=12:500
         y(i)=w'*u((i-11):(i-1));
         e(i)=a(i-7)-y(i);
         w=w+mu*e(i)*u((i-11):(i-1));
         J(i)=e(i)^2;
```

```
        average_J(i)=average_J(i)+J(i);
      end
end
average_J=average_J/N;
semilogy(average_J), title('Learning curve Ee^2(n) for LMS algorithm'),
xlabel('time step n')
```

Summary

- LMS is simple to implement.
- LMS does not require preliminary modelling.
- Main disadvantage: slow rate of convergence.
- Convergence speed is affected by two factors: the step size and the eigenvalue spread ofthe correlation matrix.
- **The condition for stability** is

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

- For LMS filters with filter length M moderate to large, the convergence condition on the step size is

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

  where S_{max} is the maximum value of the power spectral density of the tap inputs.