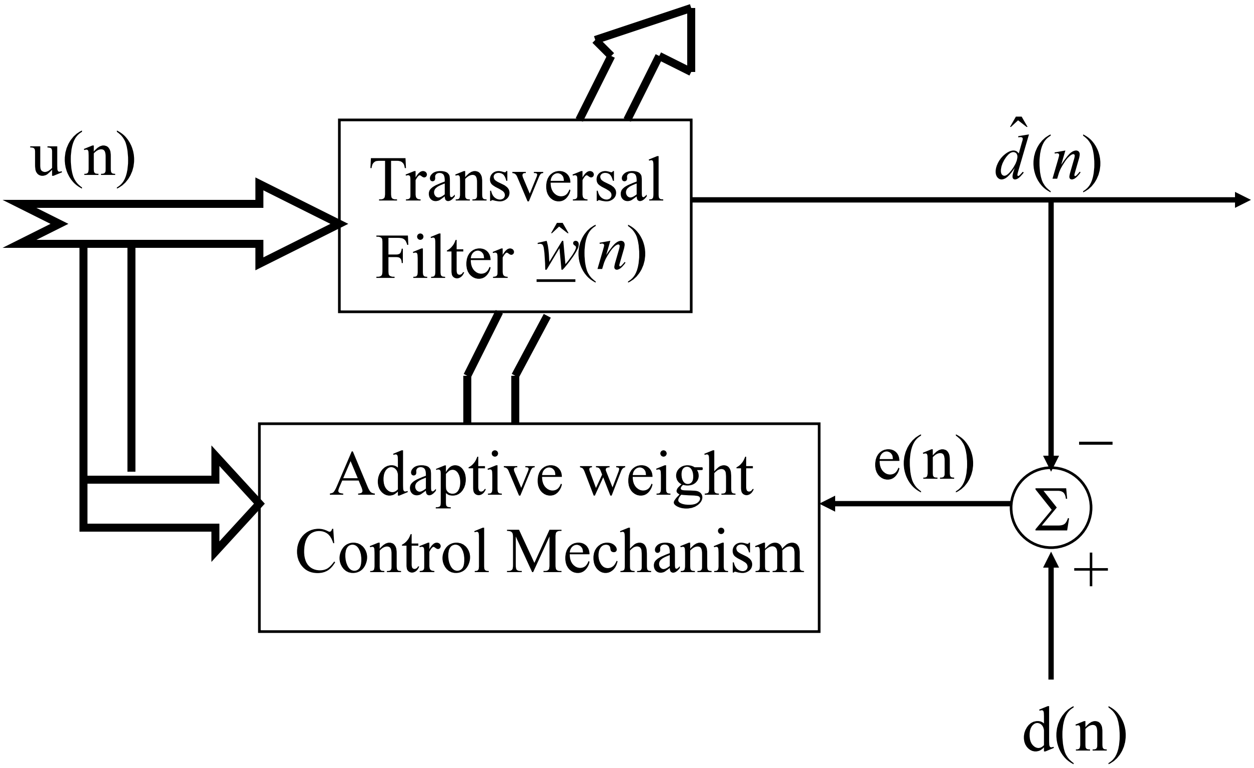
LINEAR FIR ADAPTIVE FILTERING (III)

Normalized LMS Adaptive Filters

Dr. Yogananda Isukapalli

- Summary of the LMS Linear Adaptive Transversal Filter (FIR)



Parameters : M = number of taps

μ = Step-size parameter

$$0 < \mu < \frac{2}{\text{total input power}(\,Mr(0)\,)}$$

$$\text{provided } \mu \ll 2/\lambda_{\max}$$

Initial Conditions : $\underline{\hat{w}}(0) = \underline{0}$

(a) Given $\underline{\mathrm{u}}(n) = M - by - 1$ tap - input

at time n

$d(n) =$ desired response at time n

(b) To be computed : $\underline{\hat{w}}(n+1)$ estimate of $\underline{\hat{w}}$

at n + 1

$$\left.\begin{array}{r} \vdots \\ e(n) = d(n) - \underline{\hat{w}}^{H}(n)\,\underline{\mathrm{u}}(n) \\ \underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) + \mu \underline{\mathrm{u}}(n) e^{*}(n) \end{array}\right] \qquad n = 0,1,2,.$$

- Normalized LMS Algorithm

Motivation : The correction term $\mu \underline{u}(n)e^*(n)$ in the LMS algorithm :

$$\hat{\underline{w}}(n+1) = \hat{\underline{w}}(n) + \mu \underline{u}(n)e^*(n)$$

applied to the tap-weight vector $\hat{\underline{w}}(n)$ at time n+1 is directly proportional to $\underline{u}(n)$. When $\underline{u}(n)$ is large, the LMS experiences a gradient noise amplification.

Solution : $\hat{\underline{w}}(n+1) = \hat{\underline{w}}(n) + \dfrac{\bar{\mu}}{\|u(n)\|^2}\underline{u}(n)e^*(n)$

Given the new input data (at time n) represented by the tap-weight vector $\underline{u}(n)$ and the desired response, $d(n)$, the normalized LMS algorithm updates the tap-weight vector in such a way that $\hat{\underline{w}}(n+1)$ exhibits the minimum change with respect to $\hat{\underline{w}}(n)$ at time n.

- The Development of the Normalized LMS Algorithm :

Constrained Optimization Problem :

Problem Statement :

Given $\underline{u}(n)$ and $d(n)$, determine $\underline{\hat{w}}(n+1)$ so as to minimize the squared Euclidean norm of the change

$$\delta\underline{\hat{w}}(n+1)=\underline{\hat{w}}(n+1)-\underline{\hat{w}}(n)$$

in the tap-weight vector $\underline{\hat{w}}(n+1)$ with respect to its old value $\underline{\hat{w}}(n)$, subject to the constraint

$$\underline{\hat{w}}^{H}(n+1)\underline{u}(n)=d(n)$$

Start from

$$\left\|\delta\hat{w}(n+1)\right\|^2 = \delta\underline{\hat{w}}^H(n+1)\delta\underline{\hat{w}}(n+1)$$

$$= [\underline{\hat{w}}(n+1) - \underline{\hat{w}}(n)]^H [\underline{\hat{w}}(n+1) - \underline{\hat{w}}(n)]$$

$$= \sum_{k=0}^{M-1} \left|\hat{w}_k(n+1) - \hat{w}_k(n)\right|^2$$

By defining real and imaginary components for :

$$\hat{w}_k(n) = a_k(n) + jb_k(n) \qquad k = 0,1,....M-1$$

$$d(n) = d_1(n) + jd_2(n)$$

$$u(n-k) = u_1(n-k) + ju_2(n-k)$$

We rewrite :

$$\left|\delta\hat{w}(n+1)\right|^2$$

$$= \sum_{k=0}^{M-1} ([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2)$$

and we can rewrite $\underline{\hat{w}}^H(n+1)\underline{u}(n)=d(n)$

$as:$

$$\sum_{k=0}^{M-1}(a_k(n+1)u_1(n-k)+b_k(n+1)u_2(n-k)=d_1(n)$$

$$\sum_{k=0}^{M-1}(a_k(n+1)u_2(n-k)-b_k(n+1)u_1(n-k)=d_2(n)$$

Using the method of Lagrange multipliers, we can formulate the constrained optimization problem :

$$J(n)$$

$$=\sum_{k=0}^{M-1}([a_k(n+1)-a_k(n)]^2+[b_k(n+1)-b_k(n)]^2)$$

$$=\lambda_1\left[\begin{array}{l} d_1(n)- \\ \quad \sum_{k=0}^{M-1}(a_k(n+1)u_1(n-k)+b_k(n+1)u_2(n-k)) \end{array}\right]$$

$$+\lambda_2\left[\begin{array}{l} d_2(n) \\ \quad -\sum_{k=0}^{M-1}(a_k(n+1)u_2(n-k)-b_k(n+1)u_1(n-k)) \end{array}\right]$$

where λ_1 and λ_2 are lagrange multipliers :

To find the optimum values of $a_k(n+1)$ and $b_k(n+1)$; we do :

$$\frac{\partial J(n)}{\partial a_k(n+1)} = 0$$

$$\frac{\partial J(n)}{\partial b_k(n+1)} = 0$$

Giving :

$$2[a_k(n+1) - a_k(n)] - \lambda_1 u_1(n-k) - \lambda_2 u_2(n-k)] = 0$$

$$2[b_k(n+1) - b_k(n)] - \lambda_1 u_2(n-k) + \lambda_2 u_1(n-k)] = 0$$

We can combine them back to complex form :

$$2[\mathrm{w}_{\mathrm{k}}^*(n+1) - \hat{w}_k(n)] = \lambda^* u(n-k) \quad k = 0,1,. \ . \ . \ . \ .M$$

where $\lambda = \lambda_1 + j\lambda_2$

By multiplying by $u^*(n-k)$ and then sum from k = 0 to M - 1;

$$\lambda^* = \frac{2}{\sum_{k=0}^{M} |u(n-k)|^2} \left[\sum_{k=0}^{M-1} \hat{w}_k(n+1) u^*(n-k) - \sum_{k=0}^{M-1} \hat{w}_k(n) u^*(n-k) \right]$$

$$= \frac{2}{\|\underline{u}(n)\|^2} \left[\hat{w}^T(n+1) u^*(n) - \underline{\hat{w}}^T(n) \underline{u}^*(n) \right]$$

Rewrite :

$$\lambda^* = \frac{2}{\|\underline{u}(n)\|^2} \left[d^*(n) - \underline{\hat{w}}^T(n) \underline{u}^*(n) \right]$$

Since $e(n) = d(n) - \underline{\hat{w}}^H(n) \underline{u}(n)$

$$\lambda^* = \frac{2}{\|\underline{u}(n)\|^2} e^*(n)$$

Finally

$$\delta \hat{\mathrm{w}}_{\mathrm{k}}(n+1) = \hat{w}_k(n+1) - \hat{w}_k(n)$$

$$= \frac{1}{\|\underline{u}(n)\|^2} u(n-k)e^*(n) \quad k = 0,1\ldots.M-1$$

Equivalently :

$$\delta \underline{\hat{\mathrm{w}}}(n+1) = \underline{\hat{w}}(n+1) - \underline{\hat{w}}(n)$$

$$= \frac{1}{\|\underline{u}(n)\|^2} \underline{u}(n)e^*(n)$$

Introduce $\overline{\mu}$ to control over the change in tap-weight vector :

$$\delta \underline{\hat{\mathrm{w}}}(n+1) = \frac{\overline{\mu}}{\|\underline{u}(n)\|^2} \underline{u}(n)e^*(n)$$

or

$$\boxed{\underline{\hat{\mathrm{w}}}(n+1) = \underline{\hat{w}}(n) + \frac{\overline{\mu}}{\|\underline{u}(n)\|^2} \underline{u}(n)e^*(n)}$$

By setting

$$\mu(\mathrm{n}) = \frac{\bar{\mu}}{\left\| \underline{u}(n) \right\|^2}$$

the normalized algorithm is viewed as an LMS algorithm with a time-varying step-size parameter.

$\bar{\mu}$ must satisfy : $0 < \bar{\mu} < 2$

then the normalized LMS is convergent in the mean square sense.

To avoid the division by a small number when $\underline{u}(n)$ is small,

$$\underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) + \frac{\bar{\mu}}{a + \left\| \underline{u}(n) \right\|^2} \underline{u}(n) e^*(n)$$

$a > 0$

SUMMARY OF THE NORMALIZED LMS ALGORITHM

parameters : M = number of steps

$\bar{\mu}$ = adaptation constant

$0 < \bar{\mu} < 2$

a = positive constant

Initial condition : $\underline{\hat{w}}(0) = \underline{0}$

Data

(a) Given $\underline{u}(n)$: M by 1 input vector at time n

$d(n)$: desired response at n

(b) To be computed : $\hat{w}(n+1)$

Computation : $n = 0,1,2,\ldots\ldots$

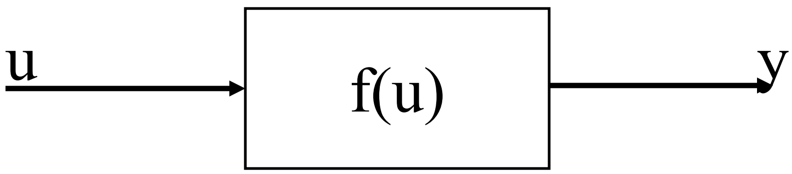
$$e(n) = d(n) - \underline{\hat{w}}^H(n)\underline{u}(n)$$

$$\underline{\hat{w}}(n+1) = \underline{\hat{w}}(n) + \frac{\bar{\mu}}{a + \|\underline{u}(n)\|^2}\underline{u}(n)e^*(n)$$

- Method of Least Squares

Let $u(1), u(2), \ldots u(N)$ represent measurements at $t_1, t_2, \ldots t_N$, the problem then is to fit a curve by using these points in some optimum fashion. Let $f(t_i)$ represent this curve. The method of least squares finds the " best " fit by minimizing the sum of difference between $f(t_i)$ and $u(i), i = 1,2, \ldots N$. Unlike in Weiner filter theory where ensemble averages are used, the method of Least Squares uses time averages. As a result, no asumption on statistics are assumed.

- Linear Regression Example



Consider now : $y = f(u)$

For discrete values : $y_i = f(u_i), \quad i = 1, . . . M$

For linear regression, assume :

$$f_a(u) = w_o + w_1 u$$

where w_o and w_1 are coefficients to be determined that produce the least square solution.

Let $e_i = f(u_i) - f_a(u_i) \quad i = 1, ... M$

Choose w_o and w_1 to minimize

$$S = \sum_{i=1}^{M} e_i^2$$

the sum of the squares of the deviations.

Now :

$$S = \sum_{i=1}^{M} [f(u_i) - f_a(u_i)]^2$$

$$= \sum_{i=1}^{M} [y_i - (w_o + w_1 u_i)]^2$$

then : $$\frac{\partial S}{\partial w_o} = \sum_{i=1}^{M} 2[y_i - (w_o + w_1 u_i)]\,(-1) = 0$$

$$\frac{\partial S}{\partial w_1} = \sum_{i=1}^{M} 2[y_i - (w_o + w_1 u_i)]\,(-u_i) = 0$$

Rewrite :

$$\sum_{i=1}^{M} y_i = M w_o + \left(\sum_{i=1}^{M} u_i\right) w_1 \qquad (1)$$

$$\sum_{i=1}^{M} u_i y_i = \left(\sum_{i=1}^{M} u_i\right) w_o + \left(\sum_{i=1}^{M} u_i^2\right) w_1 \qquad (2)$$

The solution for w_o and w_1 :

$$w_o = \frac{\left(\sum_{i=1}^{M} y_i\right)\left(\sum_{i=1}^{M} u_i^2\right) - \left(\sum_{i=1}^{M} u_i\right)\left(\sum_{i=1}^{M} u_i y_i\right)}{\Delta}$$

$$w_1 = \frac{M\left(\sum_{i=1}^{M} u_i y_i\right) - \left(\sum_{i=1}^{M} u_i\right)\left(\sum_{i=1}^{M} y_i\right)}{\Delta}$$

where $\Delta = M\left(\sum_{i=1}^{M} u_i^2\right) - \left(\sum_{i=1}^{M} u_i\right)^2$

- Solution Using Optimization in Hilbert space

  From the data :

$$w_1 + w_2 u_1 = y_1$$

$$w_1 + w_2 u_2 = y_2$$

$$\cdot$$

$$\cdot$$

$$w_1 + w_2 u_M = y_M$$

In matrix notation

$$\underset{\sim}{A}\,\underline{x} = \underline{y}$$

$$\underset{\sim}{A} = \begin{pmatrix} 1 & u_1 \\ . & . \\ . & . \\ . & . \\ 1 & u_M \end{pmatrix} \qquad \underline{x} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \qquad \underline{y} = \begin{pmatrix} y_1 \\ . \\ . \\ . \\ y_M \end{pmatrix}$$

that is : $\underline{y} = \underline{x}_1 w_1 + \underline{x}_2 w_2$

where $\underline{x}_1 = \begin{pmatrix} 1 \\ . \\ 1 \end{pmatrix}$ and $\underline{x}_2 = \begin{pmatrix} u_1 \\ . \\ u_M \end{pmatrix}$

Now the approximation is given by :

$$\underline{y}_a = \underset{\sim}{A}\,\underline{x}_a = \hat{w}_1 \underline{x}_1 + \hat{w}_2 \underline{x}_2$$

Using the orthogonality priciple :

$$(\underline{y} - \underline{y}_a, \underline{x}_l) = 0 \qquad l = 1,2$$

$$(\underline{y} - (\underline{x}_1 \hat{w}_1 + \underline{x}_2 \hat{w}_2, \underline{x}_l)) = 0 \qquad l = 1,2$$

Or

$$\begin{pmatrix} (\underline{x}_1,\underline{x}_1) & (\underline{x}_2,\underline{x}_1) \\ (\underline{x}_1,\underline{x}_2) & (\underline{x}_2,\underline{x}_2) \end{pmatrix} \begin{pmatrix} \hat{w}_1 \\ \hat{w}_2 \end{pmatrix} = \begin{pmatrix} \underline{y},\underline{x}_1 \\ \underline{y},\underline{x}_2 \end{pmatrix}$$

Follows :

$$(\underline{x}_1,\underline{x}_1) = \sum_{i=1}^{M} 1^2 = M$$

$$(\underline{x}_2,\underline{x}_2) = \sum_{i=1}^{M} u_i^2$$

$$(\underline{x}_1,\underline{x}_2) = (\underline{x}_2,\underline{x}_1) = \sum_{i=1}^{M} u_i$$

$$(\underline{y},\underline{x}_1) = \sum_{i=1}^{M} y_i$$

$$(\underline{y},\underline{x}_2) = \sum_{i=1}^{M} u_i y_i$$

Normal Equations :

$$\begin{pmatrix} M & \sum_{i=1}^{M} u_i \\ \sum_{i=1}^{M} u_i & \sum_{i=1}^{M} u_i^2 \end{pmatrix} \begin{pmatrix} \hat{w}_1 \\ \hat{w}_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{M} y_i \\ \sum_{i=1}^{M} u_i y_i \end{pmatrix}$$

$\hat{w}$ and $\hat{w}_2$ will give the same solution as directly done for w_o and w_1. They are equivalent approaches to solve least-squares problems.

- Multiple linear Regression Problem

Given : $\{d(i)\}$ and $\{u(i)\}$

$\{d(i)\}$ is observed at time i in response to input variables $u(i), u(i-1), \ldots u(i-M+1)$.

$d(i) = f(u(i))$ and assumed to be linear.

$$d(i) = \sum_{k=0}^{M-1} w_{ok}^{*} u(i-k) + e_o(i)$$

where $e_o(i)$ is error.

Assume the measurement error is white with zero mean and variance σ^2.

$$E[e_o(i)] = 0 \quad all\ i$$

$$E[e_o(i)e_o^*(k)] = \begin{cases} \sigma^2 & i = k \\ 0 & i \neq k \end{cases}$$

Follows that :

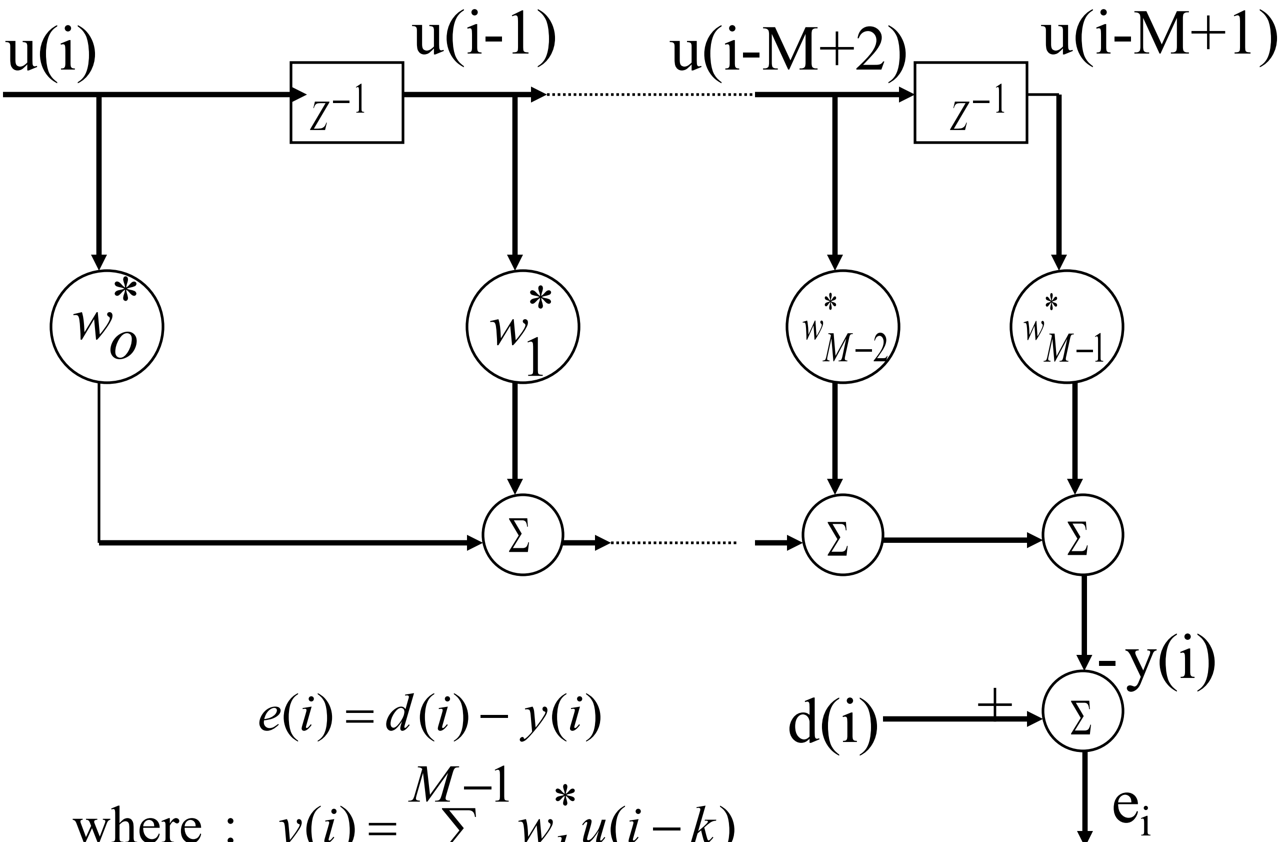
$$E[d(i)] = \sum_{k=0}^{M-1} w_{ok}^* u(i-k)$$

<u>Problem</u> : Estimate the unknown parameter of the multiple linear regression model. Estimate w_{ok}, given the two observable sets : $\{u(i)\}$ and $= \{d(i)\}$,

$i = 1,2,. . . . . N.$

- Linear least-squares filter

Assume :



$$e(i) = d(i) - y(i)$$

where : $y(i) = \sum_{k=0}^{M-1} w_k^* u(i-k)$

then : $e(i) = d(i) - \sum_{k=0}^{M-1} w_k^* u(i-k)$

Minimize the cost function : the sum of error squares :

$$\xi(w_o, \ldots \ldots w_{M-1}) = \sum_{i=i_1}^{i_2} |e(i)|^2$$

where tap - weight filter weights $w_o, \ldots \ldots w_{M-1}$ are held constant over $i_1 \le i \le i_2$

Data Windowing : Since the input data $\{u(i)\}$ $i=1,2,\ldots N$, the rectangular matrix constructed for the Mth order transversal filter may vary based on the method of windowing the input data :

a. Covariance method : Set $i_1 = M$ and $i_2 = N$ implying that no assumptions are made outside the window [1, N]
the input data matrix :

$$\begin{pmatrix} u(M) & u(M+1) & . & u(N) \\ u(M-1) & u(M) & . & u(N-1) \\ . & . & . & . \\ u(1) & u(2) & . & u(N-M+1) \end{pmatrix}$$

b. <u>Autocorrelation Method</u> :

Data prior to $i=1$ and the data after time $i=N$ are zero. Set $i_1=1$ and $i_2=N+M-1$ the input data matrix then is :

$$\begin{pmatrix} u(1) & u(2) & . & u(M) & u(M+1) & . & . & u(N) & 0 & . & 0 \\ 0 & u(1) & . & u(M-1) & u(M) & . & . & u(N-1) & u(N) & . & 0 \\ . & . & . & . & . & . & & . & . & . & . \\ . & . & . & . & . & . & & . & . & . & . \\ . & . & . & . & . & . & & . & . & . & . \\ . & . & . & . & . & . & & . & . & . & . \\ 0 & 0 & . & u(1) & u(2) & . & . & u(N-M+1) & u(N-M) & . & u(N) \end{pmatrix}$$

c. Prewindowing Method :

the input data prior to $i=1$ are zero , but makes no assumption after $i=N$. $i_1=1$ and $i_2=N$

d. Post windowing method : no assumption prior to time $i=1$ but the data after $i=N$ are zero. $i_1=M$ and $i_2=N+M-1$

Covariance Method :

Consider the cost function

$$\xi(w_o,\dots w_{M-1}) = \sum_{i=M}^{N} |e(i)|^2$$

the limit assures that for each value of i, all the M tap inputs of the transversal filter have non-zero values.

Rewrite :

$$\xi(w_o,\dots w_{M-1}) = \sum_{i=M}^{N} e(i)e^*(i)$$

By writing $w_k = a_k + jb_k \quad k = 0,\dots .M-1,$

and :
$$e(i) = d(i) - \sum_{k=0}^{M-1}(a_k - jb_k)u(i-k)$$

then the gradient vector :

$$\nabla_{\mathrm{k}}(\xi) = \frac{\partial\xi}{\partial a_k} + j\frac{\partial\xi}{\partial b_k}$$

the minimization of the cost function with respect to tap weights, $w_0, w_1, \ldots w_{M-1}$ leads to :

$$\nabla_k(\xi) = 0 \quad k = 0,1,\ldots M-1$$

where $\nabla_k(\xi) = -2\sum_{i=M}^{N} u(i-1)e^*(i)$

then :

$$\boxed{\sum_{i=M}^{N} u(i-k)e^*_{min}(i) = 0}$$

$$k = 0,1.\ .\ M-1$$

where $e_{min}(i)$ is the minimum value. This is simply the principle of orthogonality.

Implies : $\{e_{min}(i)\}$ is orthogonal to the time series $\{u(i-k)\}$ applied to tap k of a transversal filter of length M for $k = 0,1.\ .\ M-1$ when the filter is operating in its least square condition.

- We can also show that

$$\sum_{i=M}^{N} \hat{d}(i) e^{*}_{\min}(i) = 0$$

the corollary to the principle of orthogonality.

- Minimum Sum of Error Squares

Start : $d(i) = \hat{d}(i) + e_{\min}(i)$

desired *estimate of desired* *estimation error*

Evaluate the energy of the time series $\{d(i)\}$ $i = [\mathrm{M}, \mathrm{N}]$, we can show

$$\xi_d = \xi_{est} + \xi_{\min}$$

$$\xi_d = \sum_{i=M}^{N} |d(i)|^2$$

$$\xi_{est} = \sum_{i=M}^{N} |\hat{d}(i)|^2$$

$$\xi_{\min} = \sum_{i=M}^{N} |e_{\min}(i)|^2$$

- Linear Least - Squares Filters :
  Normal Equations

Start From :

$$e(i) = d(i) - \sum_{k=0}^{M-1} w_k^* u(i-k)$$

This for least - square solution can be written :

$$e_{\min}(i) = d(i) - \sum_{t=0}^{M-1} w_t^* u(i-t)$$

t is the dummy index :

Substitute this in :

$$\sum_{i=M}^{N} u(i-k) e_{\min}^*(i) = 0$$

By rearranging :

$$\sum_{t=0}^{M-1} \hat{w}_t \sum_{i=M}^{N} u(i-k) u^*(i-t) = \sum_{i=M}^{N} u(i-k) d^*(i)$$

$$k = 0, ... M-1$$

Define now :

$$\phi(t,k)=\sum_{i=M}^{N}u(i-k)u^{*}(i-t) \quad 0\le t$$

$$k\le M-1$$

the time averaged autocorrelation function of the tap inputs

$$\theta(\text{-k})=\sum_{i=M}^{N}u(i-k)d^{*}(i) \qquad 0\le k\le M-1$$

Cross - correlation between the tap inputs and the desired response.

Then : System of M simultaneous equations

$$\sum_{t=0}^{M-1}\hat{w}_{t}\phi(t,k)=\theta(\text{-k}) \quad k=0,1,. \quad . \quad . \quad . \quad .M-1$$

the expanded system of the normal equations for a linear - least square filter.

- Matrix Representation

$$\underset{\sim}{\Phi}\,\underline{\hat{\mathrm{w}}} = \underline{\theta}$$

where

$$\underset{\sim}{\Phi} = \begin{pmatrix} \phi(0,0) & \phi(1,0) & . & \phi(M-1,0) \\ \phi(0,1) & \phi(1,1) & . & \phi(M-1,1) \\ . & . & . & . \\ \phi(0,M-1) & \phi(1,M-1) & . & \phi(M-1,M-1) \end{pmatrix}$$

$$\underline{\theta} = [\theta(0) \quad \theta(-1) \quad . \quad . \quad \theta(-M+1)]^T$$

$$\underline{\hat{\mathrm{w}}} = \left[\hat{w}_o \quad \hat{w}_1 \quad . \quad . \quad \hat{w}_{M-1}\right]^T$$

Then : the solution to the normal equations :

$$\underline{\hat{\mathrm{w}}} = \underset{\sim}{\Phi}^{-1}\,\underline{\theta}$$

when $\underset{\sim}{\Phi}^{-1}$ exists .

Note that $\underset{\sim}{\Phi}$ is the time-averaged correlation matrix of the tap inputs and $\underline{\theta}$ is the time-averaged cross-correlation vector. In this sense, this is the linear-least-square filter which is counter part to the Weiner filter.

- Minimum Sum of Error Squares

We can rewrite earlier results in matrix form :

$$\xi_{\text{est}} = \underline{\hat{w}}^H \underset{\sim}{\Phi} \underline{\hat{w}}$$

$$= \underline{\hat{w}}^H \underline{\theta} = \underline{\theta}^H \underline{\hat{w}}$$

and

$$\xi_{\min} = \xi_d - \underline{\theta}^H \underline{\hat{w}}$$

$$= \xi_d - \underline{\theta}^H \Phi^{-1} \underline{\theta}$$

Properties of $\underset{\sim}{\Phi}$

Rewrite : $\underset{\sim}{\Phi} = \sum_{i=M}^{N} \underline{u}(i)\underline{u}^H(i)$

where $\underline{u}(i) = [u(i) \quad u(i-1) \quad . \quad . \quad u(i-M+1]^T$

$a.$ $\underset{\sim}{\Phi}$ is Hermitian

$$\underset{\sim}{\Phi}^H = \underset{\sim}{\Phi}$$

b. $\underset{\sim}{\Phi}$ is nonnegative definite :

$$\underline{x}^H \underset{\sim}{\Phi} \underline{x} \geq 0$$

for any M by 1 vector $\underline{x}$

c. Eigenvalues of $\underset{\sim}{\Phi}$ are real and nonnegative

d. $\underset{\sim}{\Phi}$ is the product of two rectangular Toeplitz matrices that are the Hermitian transpose of each other

$$\underset{\sim}{\Phi} = \underset{\sim}{A}^H \underset{\sim}{A}$$

where $\underset{\sim}{A}^H = [\underline{u}(M), \quad \underline{u}(M+1) \quad . \quad . \quad \underline{u}(N)]$

$$A^H = \begin{pmatrix} u(M) & u(M+1) & . & . & u(N) \\ u(M-1) & u(M) & . & . & u(N-1) \\ . & . & . & . & . \\ . & . & . & . & . \\ u(1) & u(2) & . & . & u(N-M+1) \end{pmatrix}$$

- Normal Equations In Terms Of Data Matrices

Define : $\underline{d}^H = [d(M), \quad d(M+1) \quad . \quad . \quad d(N)]$

Follows :

$$\underline{\theta} = \underset{\sim}{A}^H \underline{d}$$

$$A^H \underset{\sim}{A} \underline{\hat{w}} = \underset{\sim}{A}^H \underline{d}$$

$$\boxed{\underline{\hat{w}} = (\underset{\sim}{A}^H \underset{\sim}{A})^{-1} \underset{\sim}{A}^H \underline{d}}$$

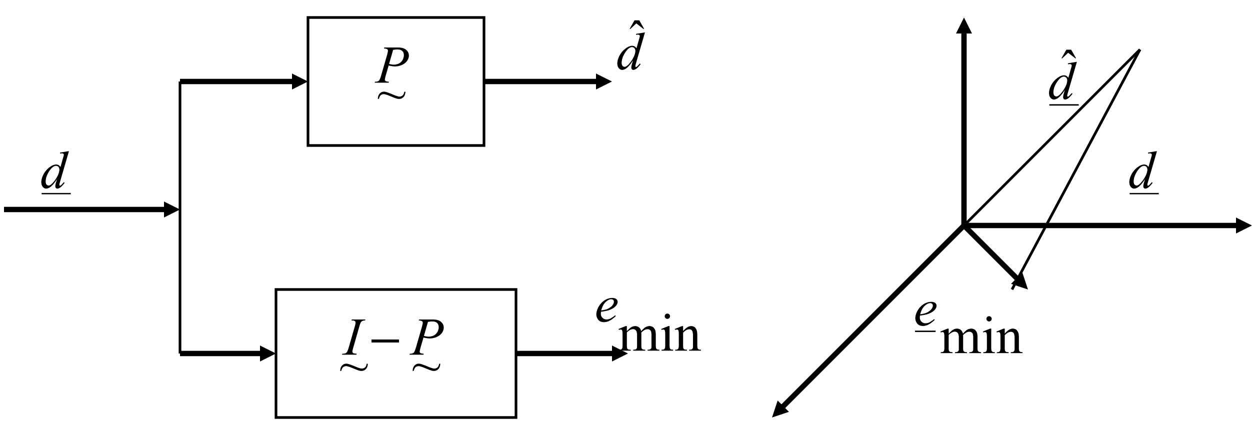
Also $\quad \xi_{\min} = \underline{d}^H \underline{d} - \underline{d}^H \underset{\sim}{A}(\underset{\sim}{A}^H \underset{\sim}{A})^{-1} \underset{\sim}{A}^H \underline{d}$

- Projection Operator Interpretation

Suppose we estimate $\hat{\underline{d}}$ from $\hat{\underline{w}}$ as :

$$\hat{\underline{d}} = A\hat{\underline{w}}$$
$$= \underset{\sim}{A}(\underset{\sim}{A}^H \underset{\sim}{A})^{-1} \underset{\sim}{A}^H \underline{d}$$

Then $\underset{\sim}{A}(\underset{\sim}{A}^H \underset{\sim}{A})^{-1} \underset{\sim}{A}^H$ is defined as a projection operator and $\underset{\sim}{I} - \underset{\sim}{A}(\underset{\sim}{A}^H \underset{\sim}{A})^{-1} \underset{\sim}{A}^H$ is known as the orthogonal complement projector. Let $\underset{\sim}{P} = \underset{\sim}{A}(\underset{\sim}{A}^H \underset{\sim}{A})^{-1} \underset{\sim}{A}^H$ and the following inter - pretation is useful :



- Uniqueness of Least - Square Estimate :

The least - squares estimate $\underline{\hat{w}}$ is unique when the data matrix $\underset{\sim}{A}$ has linearly independent columns. Implies that $\underset{\sim}{A}$ has at least as many rows and columns: $(N-M+1)\geq M$. Also means $\underset{\sim}{A}\,\underline{\hat{w}}=\underline{d}$ used in the minimization is overdetermined, meaning more equations than unknowns. Thus the least - squares estimate has the unique value :

$$\underline{\hat{w}}=(\underset{\sim}{A}^{H}\,\underset{\sim}{A})^{-1}\,\underset{\sim}{A}^{H}\,\underline{d}$$

provided $\underset{\sim}{A}$ has linearly independent columns, and M x M matrix $\underset{\sim}{A}^{H}\,\underset{\sim}{A}$ is non - singular.

Properties of Least-Squares solutions :

- $\hat{\underline{w}}$ is unbiased, provided that $\{e_o(i)\}$ has zero mean

$$E[\hat{\underline{w}}] = \underline{w}_o$$

- When $\{e_o(i)\}$ is white with zero-mean and variance σ^2, $\text{cov}[\hat{\underline{w}}] = \sigma^2 \underset{\sim}{I}^{-1}$

- When $\{e_o(i)\}$ is white and zero mean, $\hat{\underline{w}}$ is the best linear unbiased estimate.

- When $\{e_o(i)\}$ is white, Gaussian and has a zero mean, $\hat{\underline{w}}$ achieves the Cramer-Rao lower bound for unbiased estimates.

• Application of Least - Squares Method To AR Spectrum Estimation

Given the time series $\{u(i)\}\quad 1 \le i \le N$, the Forward-Backward Linear Prediction Algorithm (FBLP) is used to compute the tap-weight vector $\underline{\hat{w}}$ of a forward predictor or the tap-weight vector $\underline{\hat{a}}$ of the prediction error filter. The vector $\underline{\hat{a}}$ represents as estimate of AR model used to fit the time series $\{u(i)\}$. $\xi_{\min}$ represents as estimate of the white noise variance σ^2 in the AR model. The estimate of the AR spectrum is given by

$$\hat{S}_{AR}(w) = \frac{\xi_{\min}}{\left|1 + \sum_{k=1}^{M} \hat{a}_k^* e^{-jwk}\right|^2}$$

$M \approx N/3$ for best performance

Application: MVDR Spectrum Estimation

- Independent sensors placed at different points in space, "listen" to the received signal and try to distinguish between the spatial properties of signal and noise.
- Beamformer places nulls in the directions of the sources of interference in order to increase the output SINR.
- The goal is to minimize the variance (average power) of the beamformer output while a distortionless response is maintained along the direction of a target signal of interest.

- Output of linear transversal filter in response to tap inputs:

$$y(i)=\sum_{t=0}^{M}a_t^{*}u(i-t)$$

- The requirement is to minimize the output energy:

$$\xi_{out}=\sum_{i=M+1}^{N}\left|y(i)\right|^{2}$$

- Instead of a desired response we now have a constraint:

$$\sum_{k=0}^{M}a_k^{*}e^{-jk\omega_0}=1$$

- To solve the constrained minimization problem , a constrained cost function is defined:

$$\xi = \underbrace{\sum_{i=M+1}^{N} \left| y(i) \right|^2}_{output\ \ energy} + \underbrace{\lambda\left(\sum_{k=0}^{M} a_k^* e^{-jk\omega_0} - 1\right)}_{linear\ \ constarints}$$

Where, λ is a complex Lagrange multiplier.

- The minimization involves equating the gradient to zero:

$$\sum_{k=0}^{M} \hat{a}_t \phi(t,k) = -\frac{1}{2}\lambda^* e^{-jk\omega_0}, \ \ k = 0,1,\ldots,M.$$

Where, is ϕ autocorrelation function of tap inputs.

- Solving for λ subjecting to the constraint and Substituting it in the equation for optimum tap weights, gives the **MVDR** formula as follows:

$$\hat{\mathbf{a}} = \frac{\Phi^{-1}\mathbf{s}(\omega_0)}{\mathbf{s}^H(\omega_0)\Phi^{-1}\mathbf{s}(\omega_0)}$$

- The minimum value of output energy:

$$\mathbf{S}_{MVDR}(\omega_0) = \frac{1}{\mathbf{s}^H(\omega)\Phi^{-1}\mathbf{s}(\omega)}.$$

- The above equation is referred as the MVDR Spectrum estimate, at any ω the power due to other frequencies is minimized. Hence the Spectrum exhibits relatively sharp peaks.