AGC

LRA

DSP

Optimum LSE Filtering

Wiener Filter

Adaptive LMS Filtering

Steepest Descent Method

Stochastic Gradient Method

Adaptive Signal Processing

- Problem statement:
  - Equalise the distortion caused by a time-varying communication channel with the aid of a FIR filter.
  - If the channel were fixed (LTI) then a possible solution is the *Wiener filter* approach (MMSE)
    - Requires *time-invariance* and *a priori knowledge* of autocorrelation matrix of the transmitted signal and cross-correlation vector between the input and desired response
- Solution: *adaptive signal processing (ASP)*
  - Used when second-order statistics of a signal are not available
  - Estimates for the autocorrelation matrix and cross-correlation derived from accumulated data
  - Filter coefficients progressively adjusted w.r.t. signal statistics

Adaptive Signal Processing

- Problem is particularly acute when *both* environment *and* signal involved are changing in a quasi-stationary ("slowly" time-varying) manner
- Strategy: follow temporal behaviour of the signals and adapt the correlation parameters as the environment is changing
- Result: a temporally adaptive filter (preferably FIR, to realise stability)
- Criterion for optimal adjustment of filter coefficients
- Intensively studied since 1970s

AGC

LRA

DSP

Adaptive Signal Processing

- Filter coefficients to be adjusted according to some type of "optimality" criterion
  - Criterion should be meaningful, produce "accurate" results and be practically realizable
    - E.g. minimum mean probability of error (MMPE)
      - Positive definite, but ...
      - ... MMPE is nonlinear function of filter coefficients and signal statistics, may get trapped in local minimum, etc.

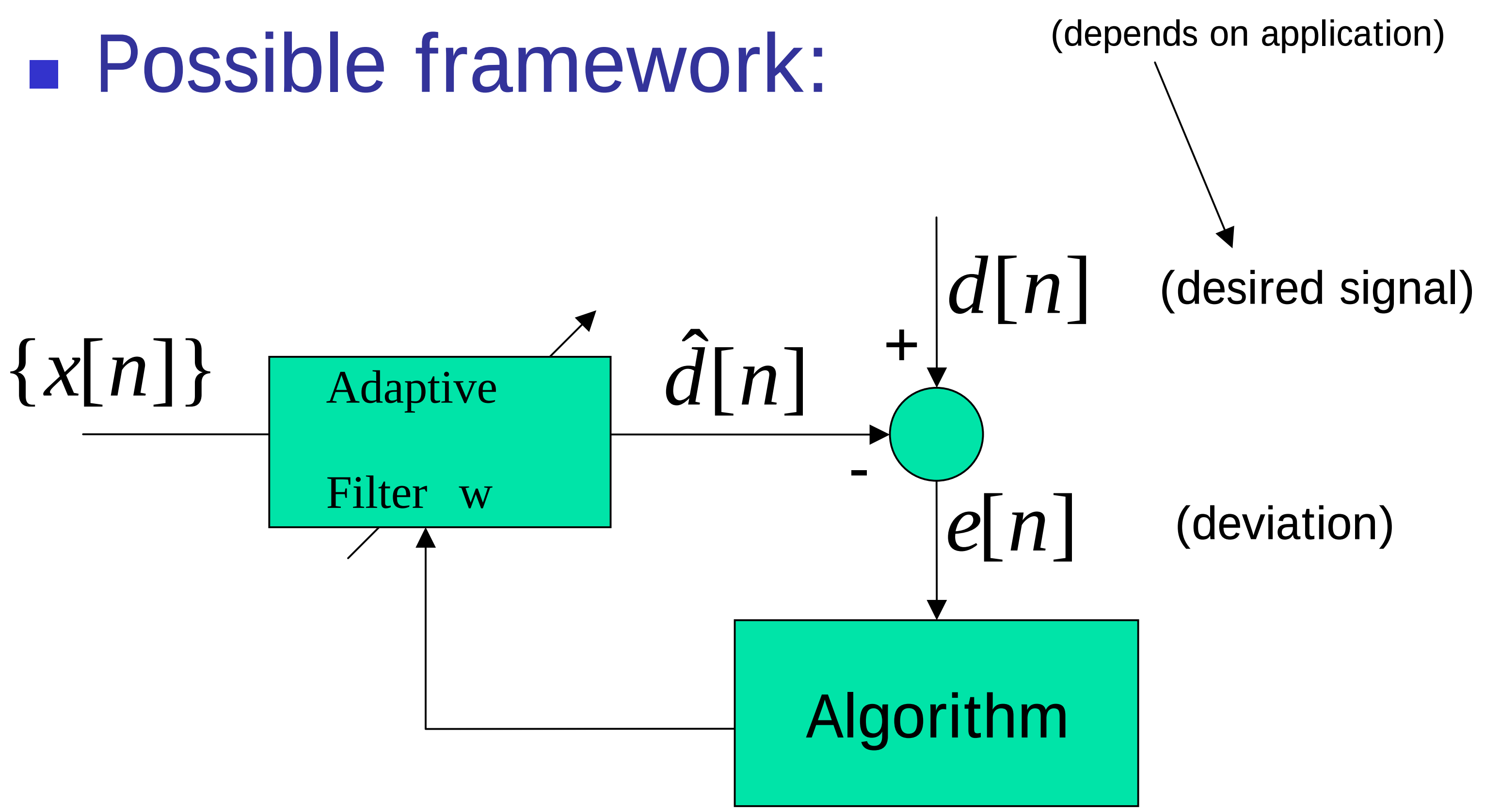
      $\Rightarrow$ MMPE impractical as optimality criterion in ASP
    - Better choice: least square (LSE) or minimum mean square error (MMSE) criterion:
      - practical: *quadratic* function for optimisation
      - single minimum

Adaptive Signal Processing

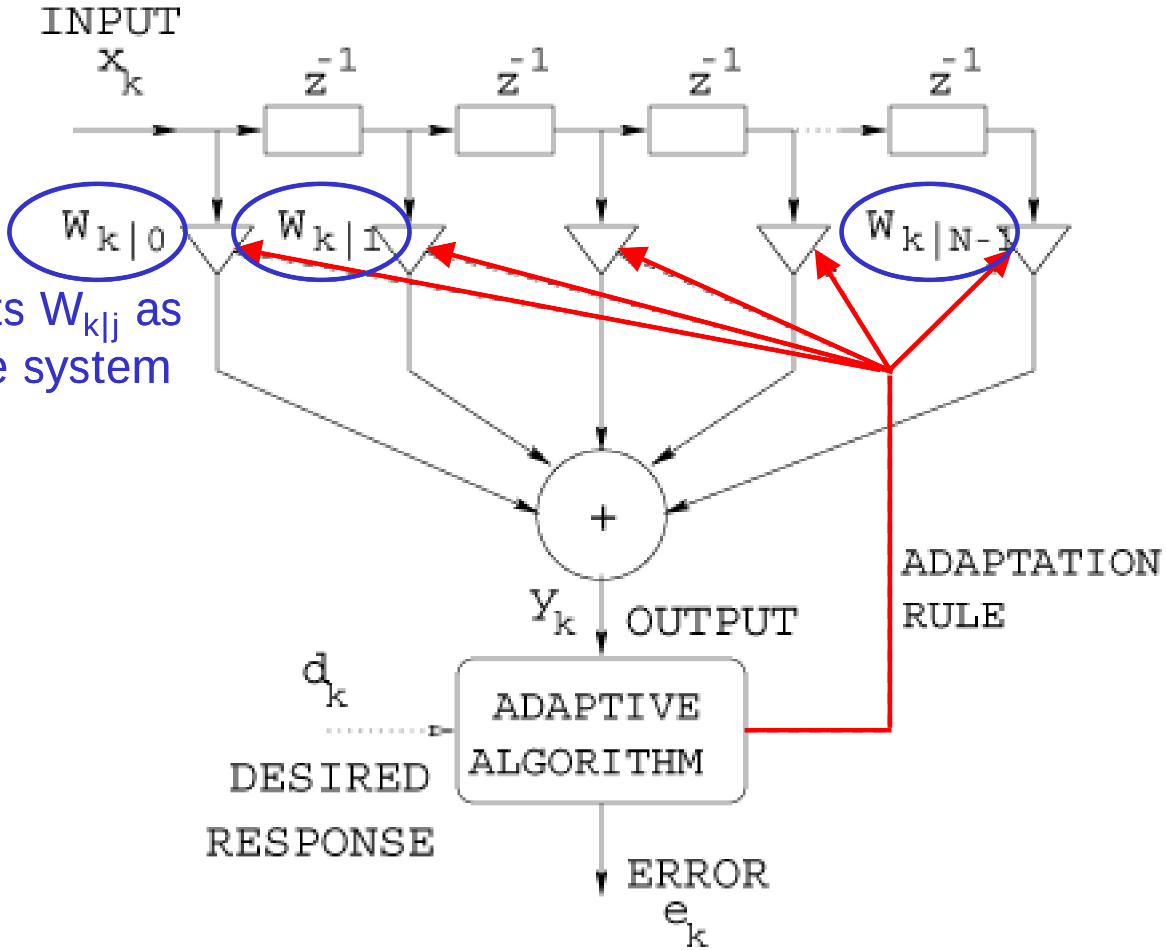
- Applications:
  - Digital communications
  - Channel equalisation of intersymbol interference (ISI)
  - Adaptive echo or noise cancellation
  - System or channel identification
  - Adaptive/smart antenna systems (beam steering)
  - Blind system equalisation
  - Etc.

ASP: General Approach

- Possible framework:



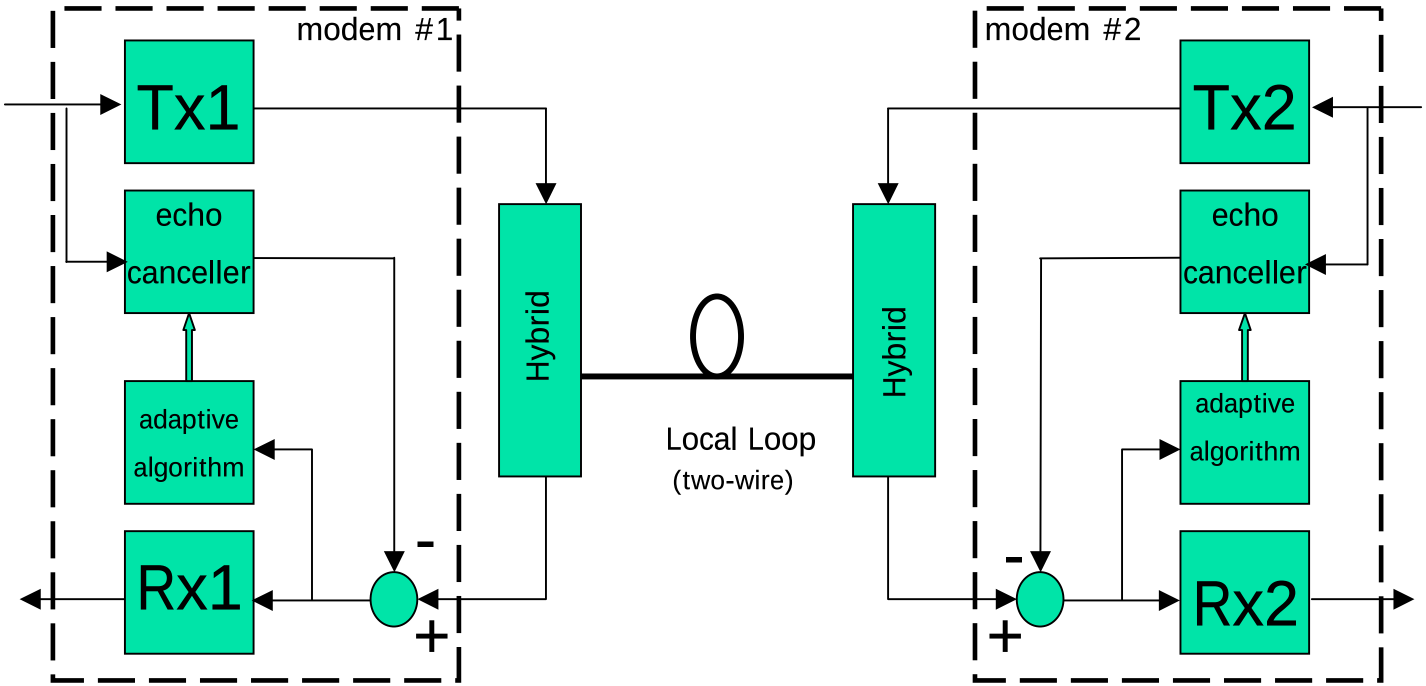
Direct-form Adaptive FIR Filter



adjustable weights $W_{k|j}$ as parameters of the system

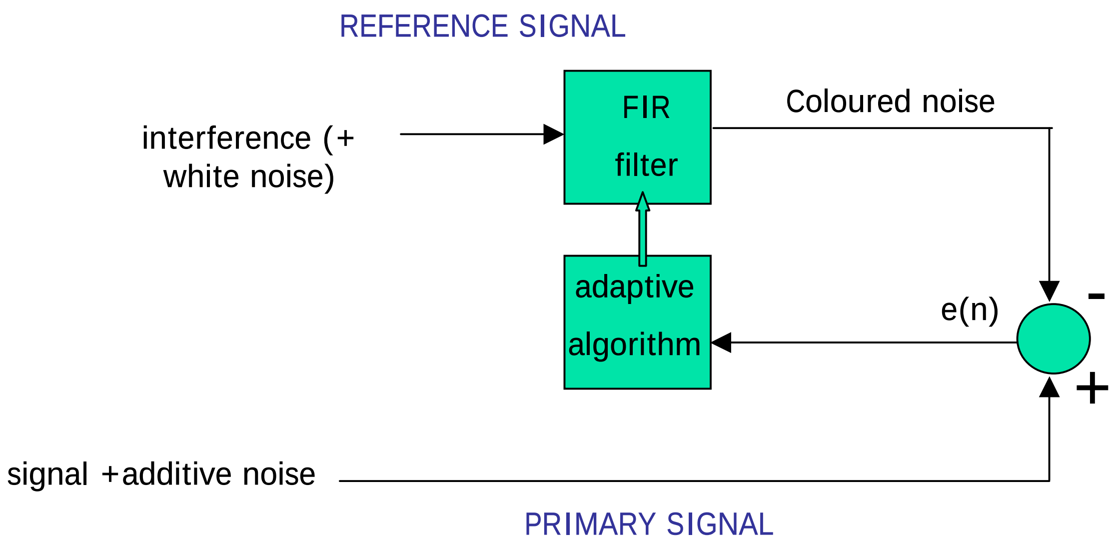
ASP Applications (I)

- Echo Cancellation in Local Loops
  - Adaptive filtering with adjustable cffs., used inside modem to suppress echo in two-wire data transmission



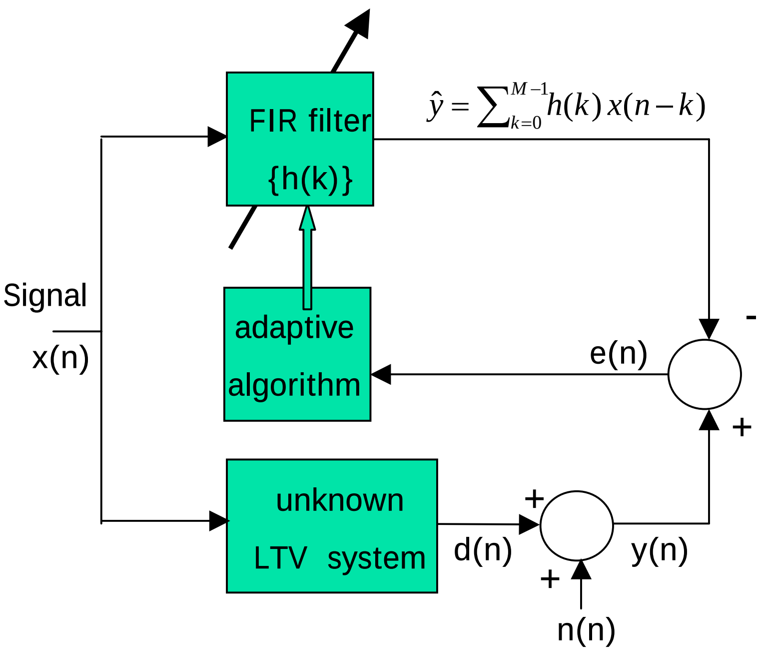
ASP Applications (II)

- Adaptive Noise Cancellation
  - cffs. of FIR filter are adjusted based on e(n) such that generated coloured noise (FIR filtered interference signal) matches unknown additive noise in primary signal



ASP Applications (III)

- System Identification:
  - determining unknown parameters of LTV transfer function modelled as FIR filter with *M* adjustable cffs.



h(*k*) to be adjusted according to MMSE criterion:

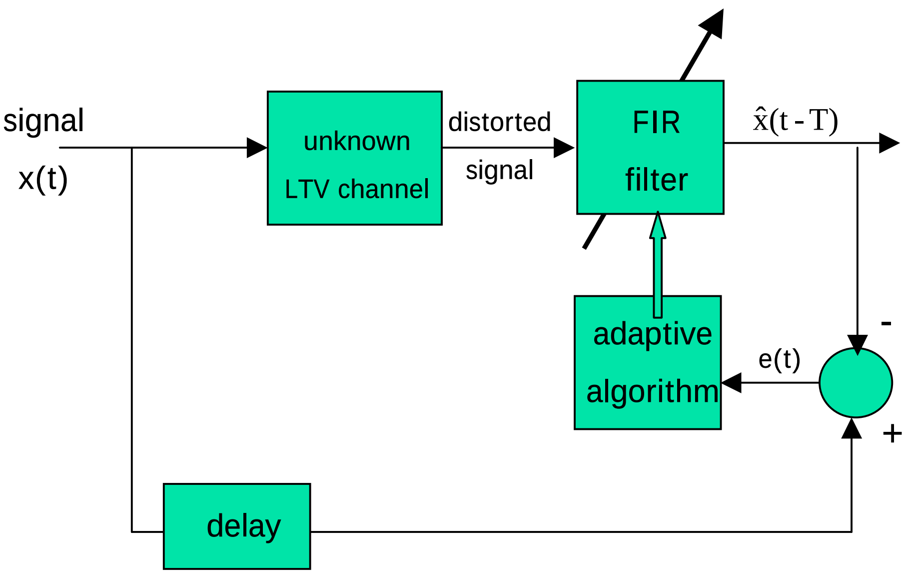
$$\sum_{n=0}^{N} e^2(n) = \sum_{n=0}^{N} \left[y(n) - \sum_{k=0}^{M-1} h(k)\, x(n-k) \right]^2 \min.$$

such that output of FIR filter tracks output of unknown system in time

d(n) and n(n) are not observable

ASP Applications (IV)

- System Equalisation (Annihilation):
  - compensation for distortion caused by LTV channel



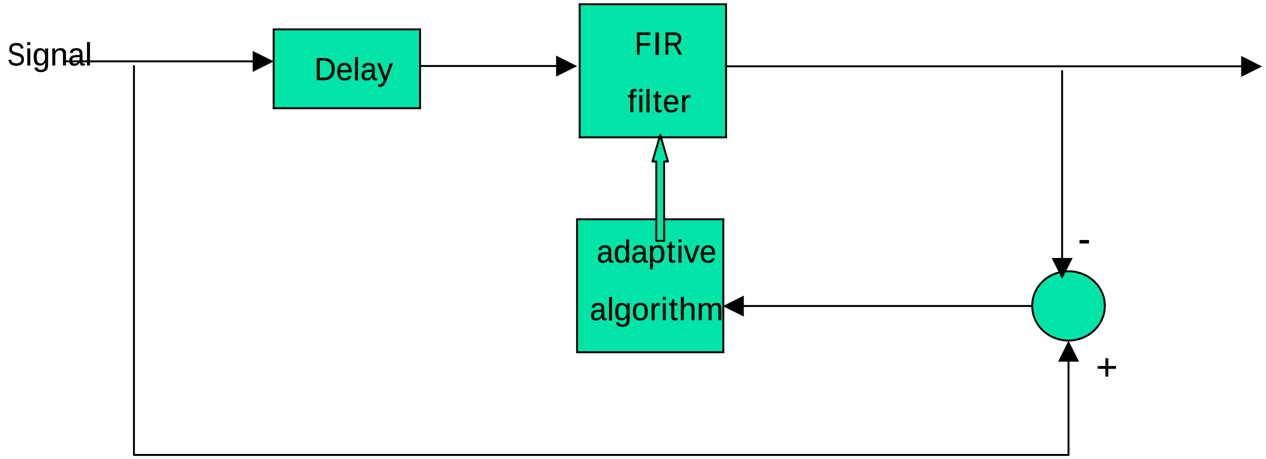
Adjusted distorted signal $\hat{x}(t - T)$ tracks original signal x(t); FIR filter is inverse of unknown LTV system

Used for reduction of ISI by channel-induced distortion

NB: channel is time-varying; equalisation is only effective if filter cffs. convergence sufficiently rapidly, i.e., channel variations must be quasi-static relative to computation speed

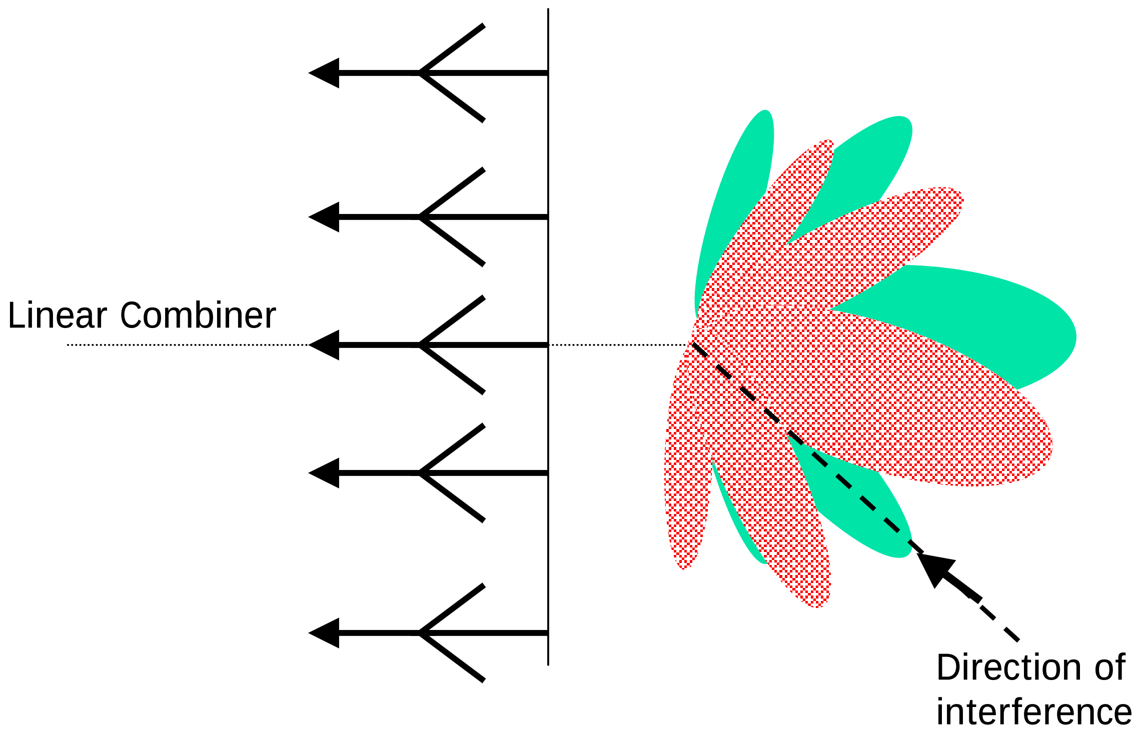
ASP Applications (V)

- Adaptive Predictors:
  - LPC: used with speech signals
    - Constructing model of speech signal to allow for differential coding (DM, (A)DPCM) to reduce bit rate



ASP Applications (VI)

- Adaptive arrays of sensors (multiple signals)



Proper adjustment of all feed signals allows adjusting radiation pattern (transmission or reception)

Original radiation pattern to be steered such that unwanted signal (interference) is directed along a null of the pattern

Adaptive LMS Algorithm

- Examples so far involve adaptively changing filter coefficients (weights of taps for impulse response function)
- Now: discussion of an algorithm to select filter coefficients, viz., least mean squares (LMS)
- Basic principles:
  - 1) Form an object (cost, penalty) function (performance criterion)
  - 2) Find gradient of object function w.r.t. FIR filter weights
  - 3) Several different approaches are then possible
  - 4) Form a differential/difference equation from the gradient

Adaptive LMS: MMSE

- Desired signal: $d[n]$
- Input signal: $x[n]$
- Output signal: $y[n]$
- Form the vectors of *m* elements:

$$\mathbf{x}[n] = \left[x[n] \quad x[n-1] \quad . \quad x[n-m+1]\right]^{\mathrm{T}}$$

$$\mathbf{h} = \left[h[0] \quad h[1] \quad . \quad h[m-1]\right]^{\mathrm{T}}$$

whence $y[n] = \mathbf{x}[n]^{\mathbf{T}} \cdot \mathbf{h}$ is an <u>estimate</u> of $d[n]$:

$$y[n] = \hat{d}[n], \qquad e[n] = d[n] - y[n]$$

AGC
LRA
DSP

Adaptive LMS: MMSE

- Now construct the object function *J* based on predefined criterion, i.e., MSE (estimation error):

$$J(\mathbf{h}) = E\{[d[n] - y[n]]^2\} = \mathrm{MSE} = E(|e[n]|^2)$$

$$J(\mathbf{h}) = \sigma_d^2 - \mathbf{p}^{\mathrm{T}}.\mathbf{h} - \mathbf{h}^{\mathrm{T}}.\mathbf{p} + \mathbf{h}^{\mathrm{T}}.\mathbf{R}.\mathbf{h}$$

with

$\mathbf{R} = E\{\mathbf{x}[n]\mathbf{x}[n]^{\mathrm{T}}\}$: autocorrelation matrix (m×m) (Toeplitz, positive definite → invertible)

$\mathbf{p} = E\{\mathbf{x}[n]d[n]\}$: cross-correlation vector (m×1) (projection of ***x*** onto *d*)

- *J*(**h**) is scalar <u>quadratic</u> function of filter cffs. that has <u>unique</u> minimum (because **R** positive definite)

Adaptive LMS: Steepest Descent Algorithm

- Aim: minimize $J(\mathbf{h})$ at any instance n
  - Assume for the moment: **R** and **p** are known
  - For *Steepest Descent Method* :

$$\mathbf{h}[n+1] = \mathbf{h}[n] - \frac{1}{2}\mu \frac{\partial J(\mathbf{h}[n])}{\partial \mathbf{h}[n]} = \mathbf{h}[n] + \mu E\{\mathbf{x}[n]e[n]\}$$

where $\frac{\partial J(\mathbf{h})}{\partial \mathbf{h}} = -2\mathbf{p} + 2\mathbf{R}\cdot\mathbf{h} = -2E\{\mathbf{x}[n](d[n]\text{-}y[n])$ is the gradient (ascent), i.e., direction vector for n^{th} iteration and μ is a step size to be chosen (cf. infra)

NB: different strategies for choosing the gradient lead to different algorithms (CGM, Fletcher-Powell, etc.) with faster convergence

Adaptive LMS: Wiener Filter

- Hence weights are updated in accordance with

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \mu(\mathbf{p} - \mathbf{R}.\mathbf{h}[n])$$

- Convergence is reached in *m* steps (because *J* is quadratic in *h*)
- This update equation is not practical (iterative form)
- If **R** and **p** were known *a priori* , then the required solution is the Wiener filter:

$$\boxed{\mathbf{h}_{\text{opt}} = \mathbf{R}^{-1}.\mathbf{p}}$$

i.e. closed-form linear solution to MMSE minimization problem for LTI system (optimal solution for Gaussian noise)

Adaptive LMS: Suboptimal

- Associated residual error is $J(\mathbf{h}) = \sigma_d^2 - \mathbf{p}^{\mathrm{T}}.\mathbf{R}.\mathbf{p}$
- However, **R** and **p** are unknown!
- Solution: approximate expressions are obtained by ignoring the expectations in the above forms (approximation of order N=1):

$$\hat{\mathbf{R}}[n] = \mathbf{x}[n]\mathbf{x}[n]^{\mathrm{T}}, \quad \hat{\mathbf{p}}[n] = \mathbf{x}[n]d[n]$$

- This is a crude approximation, particularly near start of process. However, because the update equation accumulates such quantities, the approximation improves progressively, in the mean

LMS Algorithm

- Thus, we have (no expectation values!)

$$\boxed{\mathbf{h}[n+1] = \mathbf{h}[n] + \mu\,\mathbf{x}[n](d[n] - \mathbf{x}[n]^{\mathrm{T}}.\mathbf{h}[n])}$$

with associated estimation error

$$e[n] = d[n] - \mathbf{x}[n]^{\mathrm{T}}.\mathbf{h}[n] = d[n] - y[n]$$

- Hence, filter cffs. are consecutively calculated as

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

- This is also called *stochastic gradient descent* : adaptive filtering based on <u>instantaneous </u>error
- More sophisticated: $\mathbf{x}[n]e[n] \rightarrow \frac{1}{N}\sum_{k=0}^{N-1}\mathbf{x}[n-k]e[n-k]$

Convergence & Step Size

- The step size μ is a parameter to be selected carefully:
  - If μ too small: long time to convergence
  - If μ too large: possible instability (oscillations or divergence of h[n])

Convergence & Step Size

- The error in the weights with respect to their optimal values is given by (using the Wiener solution $\mathbf{h}_{\mathrm{opt}}$ for $\mathbf{p}$ - not *a priori* known):

$$\mathbf{h}[n+1]-\mathbf{h}_{\mathrm{opt}}=\mathbf{h}[n]-\mathbf{h}_{\mathrm{opt}}+\mu(\mathbf{R}.\mathbf{h}_{\mathrm{opt}}-\mathbf{R}.\mathbf{h}[n])$$

or, rewritten,

$$\mathbf{e}_h[n+1]=\mathbf{e}_h[n]-\mu\mathbf{R}.\mathbf{e}_h[n]$$

- Write the autocorrelation matrix in the eigen decomposition form:

$$\mathbf{R}=\mathbf{Q}^{\mathrm{T}}.\mathbf{\Lambda}.\mathbf{Q},$$

Q orthogonal ($Q.Q^T=I$),

$\Lambda=\mathrm{diag}(\lambda_1,\ldots,\lambda_M)$

Convergence

- Substituting: $\mathbf{e}_h[n+1] = (\mathbf{I} - \mu \mathbf{Q}^{\mathrm{T}}.\mathbf{\Lambda}.\mathbf{Q}).\mathbf{e}_h[n]$
- Define $\mathbf{v}[n] = \mathbf{Q}.\mathbf{e}_h[n]$ (orthogonally transformed weight error)
- Hence

$$\begin{aligned}\mathbf{Q}.\mathbf{e}_h[n+1] &= \mathbf{Q}.(1 - \mu \mathbf{Q}^{\mathrm{T}}.\mathbf{\Lambda}.\mathbf{Q}).\mathbf{e}_h[n] \\ &= (\mathbf{Q} - \mu \mathbf{Q}.\mathbf{Q}^{\mathrm{T}}.\mathbf{\Lambda}.\mathbf{Q}).\mathbf{e}_h[n] \\ &= (\mathbf{I} - \mu \mathbf{\Lambda}).\mathbf{Q}.\mathbf{e}_h[n]\end{aligned}$$

i.e.,

$$\boxed{v_k[n+1] = \left[(\mathbf{I} - \mu \mathbf{\Lambda}).\mathbf{v}[n]\right]_k = \left(1 - \mu \lambda_k\right) v_k[n]}$$

Convergence & Step Size

- Significance:
  - each element of **v**[n] depends on the corresponding previous value **v**[n-1] via a *constant* scaling factor (scalar)
  - In the time domain, this corresponds to a linear differential equation with constant cffs. with exponential solution. The largest coefficient $(1-\mu\lambda_k)$ dominates the solution
  - Convergence if $|1-\mu\lambda_{\max}|<1$, i.e., $0<\mu<\dfrac{2}{\lambda_{\max}}$
    - In practice: $\mu << 2/\lambda_{\max}$ chosen
    - μ can be optimized (*normalized LMS algorithm*)

Limiting Forms

- When $n \to \infty$, the update equation yields

$$E\{\mathbf{h}[n+1]\} = E\{\mathbf{h}[n]\}$$

- Taking expectations of both sides (p. 20):

$$E\{\mathbf{h}[n+1]\} = E\{\mathbf{h}[n]\} + \mu E\{\mathbf{x}[n](d[n] - \mathbf{x}[n]^{\mathrm{T}}.\mathbf{h}[n])\}$$

hence

$$0 = \mu E\{\mathbf{x}[n]d[n] - \mathbf{x}[n](\mathbf{x}[n]^{\mathrm{T}}.\mathbf{h}[n])\}$$

This indicates that the solution ultimately converges to the Wiener form and that the estimate is unbiased