ENGG30201 – Machine Learning for Engineers: Lecture 5: Adaptive Systems

Prof Harris Tsimenidis

Nottingham Trent University
Department of Engineering

March 5, 2025

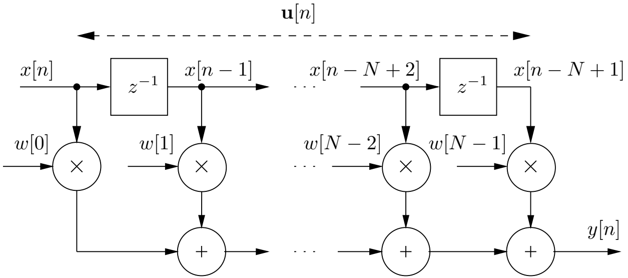
Nottingham Trent University

At the end of this lecture, students should be able to:

- Understand the concept of Adaptive Filters (AF) and their applications in various engineering domains.
- Explain the structure and functioning of Finite Impulse Response (FIR) filters.
- Discuss different applications of adaptive systems including adaptive cancellation, adaptive equalization, and prediction of future values of signals.
- Describe the Wiener filter and its limitations in practical implementations.
- Analyse the Wiener-Hopf equations and their implications in filter design.
- Implement the Least Mean Square (LMS) algorithm for adaptive filtering.
- Apply the LMS algorithm in practical scenarios such as noise cancellation and equalization.

Introduction

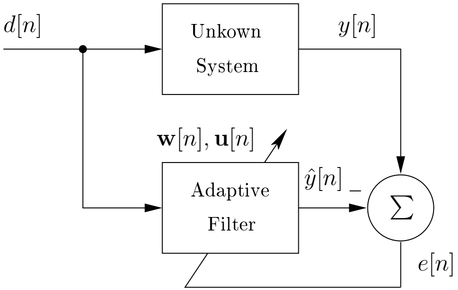
- An AF adjusts its coefficients over time to achieve an optimum response for a particular scenario.
- It is usually a FIR filter with adaptive weights.
- The adaptive algorithm learns from the input data iteratively and continually updates the filter coefficients, such that an error signal is minimised according to some criterion (cost function).
- The learning process in adaptive filters often involves optimization techniques such as gradient descent or recursive least squares.
- AF have many applications in removing noise (adaptive cancellation) or removing distortion (adaptive equalization) from signals, prediction of future values of signals (stock market).



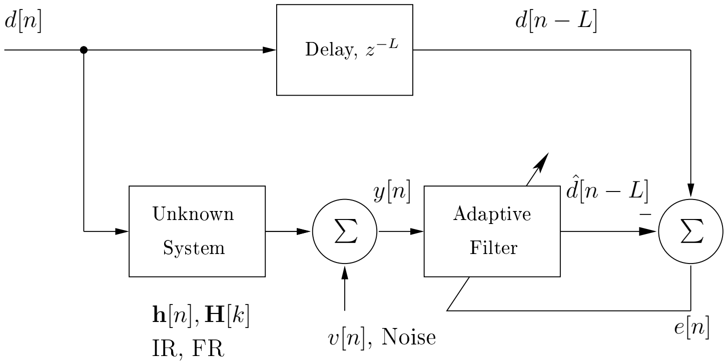
where

- $x[n]$ is the input signal.
- $\mathbf{u}[n] \in \mathbb{R}^{N\times 1}$ is the filter memory
- $\mathbf{w}[n] \in \mathbb{R}^{N\times 1}$ are the filter coefficients.
- $y[n]$ is the output signal computed as:

$$y[n] = \sum_{i=0}^{N-1} w[i]x[n-i] = \mathbf{w}^T[n]\mathbf{u}[n]$$

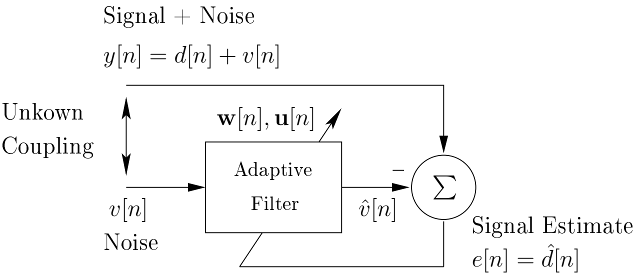


- *Desired* (training) signal, $d[n]$ is input to both the unknown system and to the adaptive filter.
- Access to the unknown system output, $y[n]$, is required.
- If the unknown system is analog, ADCs at its input and output are required.
- Application: echo cancellation, system identification, ECG mains interference cancellation.



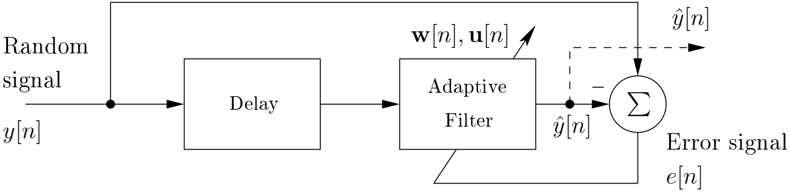
- *Desired* signal is delayed to allow for propagation through the unknown system.
- Delay required needs to be estimated.
- Application: adaptive equalization in comms. systems.

Interference Cancellation



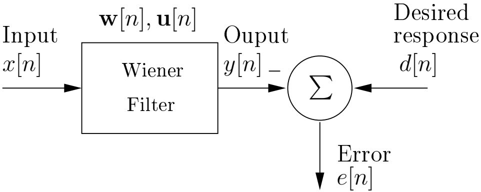
- *Desired* signal corrupted by noise.
- Reference signal (noise) is the input to AF.
- AF estimates coupled noise using reference signal.
- Application: noise cancelling microphones/headphones.

Prediction



- AF provides the best prediction of a future value of a random signal.
- Input to AF is the delayed random signal.
- Application: speech compression, stock market.

Wiener Filter



- Define:

$$\mathbf{u}[n] = \Big[u[n], u[n-1], \ldots, u[n-N+1]\Big]^T, \; \mathbf{u}[n] \in \mathbb{R}^{N\times 1}$$

$$\mathbf{w}[n] = \Big[w[n], w[n-1], \ldots, w[n-N+1]\Big]^T, \; \mathbf{w}[n] \in \mathbb{R}^{N\times 1}$$

- The error signal:

$$e[n] = d[n] - y[n] = d[n] - \sum_{i=0}^{N-1} w[n]u[n-i]$$

$$= d[n] - \mathbf{w}^T[n]\mathbf{u}[n]$$

Wiener Filter Theory (Cont.)

- Instantaneous squared error:

$$\begin{aligned} e^2[n] &= (d[n] - \mathbf{w}^T[n]\mathbf{u}[n])(d[n] - \mathbf{w}^T[n]\mathbf{u}[n]) \\ &= d^2[n] - 2\mathbf{w}^T[n]d[n]\mathbf{u}[n] + \mathbf{w}^T[n]\mathbf{u}[n]\mathbf{u}^T[n]\mathbf{w}[n] \end{aligned}$$

- The mean squared error (MSE), ξ_n is defined by the statistical *expectation* ($\mathbb{E}\{\,.\,\}$) of the squared error:

$$\begin{aligned} \xi_n = \mathbb{E}\{e^2[n]\} &= \mathbb{E}\{d^2[n]\} - 2\mathbf{w}^T[n]\underbrace{\mathbb{E}\{d[n]\mathbf{u}[n]\}}_{\mathbf{r}_{du}} + \mathbf{w}^T[n]\underbrace{\mathbb{E}\{\mathbf{u}[n]\mathbf{u}^T[n]\}}_{\mathbf{R}_{uu}}\mathbf{w}[n] \\ &= \mathbb{E}\{d^2[n]\} - 2\mathbf{w}^T[n]\mathbf{r}_{du} + \mathbf{w}^T[n]\mathbf{R}_{uu}\mathbf{w}[n] \end{aligned}$$

- The gradient is computed as:

$$\nabla \xi[n] = \frac{\partial}{\partial \mathbf{w}^T[n]}\, \xi[n] = -2\mathbf{r}_{du} + 2\mathbf{R}_{uu}\mathbf{w}[n]$$

- The error signal $e[n]$ is minimized if:

$$\frac{\partial}{\partial \mathbf{w}^T[n]}\, \xi[n] = 0$$

- Thus,

$$-2\mathbf{r}_{du} + 2\mathbf{R}_{uu}\mathbf{w}[n] = 0$$

Wiener-Hopf Equations (Cont.)

- Re-arranging yields:

$$\mathbf{R}_{uu}\mathbf{w}[n] = \mathbf{r}_{du}$$

  which are known as the Wiener-Hopf equations in matrix form.
- Solving for $\mathbf{w}[n]$ results in the optimal solution that minimizes $\xi[n]$, i.e.:

$$\mathbf{w}^{\texttt{opt}}[n] = \mathbf{R}_{uu}^{-1}\mathbf{r}_{du}$$

Disadvantages of the Wiener Filter

- Its implementation is impractical as it requires:
  - A matrix inversion, which is computationally costly.
  - A block of N samples, which results in a delay.
  - If the signal statistics change, $\mathbf{w}^{\mathtt{opt}}[n]$ has to be re-computed to track the changing conditions.
- How do we resolve this issue?

Steepest Descent Algorithm

- Assuming $\mathbf{w}[n]$ will not change significantly between subsequent iterations:

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \tilde{\eta}\nabla\xi[n]$$

where $\tilde{\eta}$ is a constant known as the adaptive step size (usually $\tilde{\eta} << 1$).

- The gradient is computed as:

$$\nabla\xi[n] = \mathbb{E}\left\{\frac{\partial e^2[n]}{\partial \mathbf{w}^T[n]}\right\} = \mathbb{E}\left\{2e[n]\frac{\partial(d[n] - \mathbf{w}^T[n]\mathbf{u}[n])}{\partial \mathbf{w}^T[n]}\right\} = -2\mathbb{E}\{e[n]\mathbf{u}[n]\}$$

- The expectation $\mathbb{E}\{\cdot\}$ is a statistical average and can be computed using the M-sample mean:

$$\nabla\xi[n] \approx -\frac{2}{M}\sum_{i=0}^{M-1} e[n-i]\mathbf{u}[n-i]$$

LMS Algorithm

- If we restrict the gradient estimate to the current sample, i.e. $M = 1,\ i = 0$, we obtain:

$$\nabla\xi[n] \approx -2e[n]\mathbf{u}[n]$$

- The steepest decent update equation becomes

$$\mathbf{w}[n+1] = \mathbf{w}[n] - 2\tilde{\eta}e[n]\mathbf{u}[n]$$

which is known as the Least Mean Square (LMS) algorithm.

- The factor of -2 can be absorbed into a new constant, $\eta = -2\tilde{\eta}$, thus, the update equation becomes

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta e[n]\mathbf{u}[n]$$

LMS Algorithm (Real-valued Signals)

Start-up:

1. Initialize μ, $\mathbf{w}[0] = \mathbf{0}$ and $\mathbf{u}[0] = \mathbf{0}$

For each sample in $d[n]$ iterate as:

2. Update $\mathbf{u}[n]$ with a new sample from $x[n]$, i.e.:

$$u[0 \to N-2] = u[1 \to N-1]$$
$$u[N-1] = x[n]$$

3. Compute the output of the LMS filter:

$$y[n] = \mathbf{w}^T[n]\mathbf{u}[n]$$

4. Compute the error signal:

$$e[n] = d[n] - y[n]$$

5. Update the filter weights

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta e[n]\mathbf{u}[n]$$

LMS Algorithm (Complex-valued Signals)

- For complex-valued signals, two modifications are required.
  1. The LMS filter output is computed as:

$$y[n] = \mathbf{w}^H[n]\mathbf{u}[n]$$

  2. The filter weights are computed as:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \eta e^*[n]\mathbf{u}[n]$$

- $(\ )^*$ is the complex conjugate operator with

$$z^* = (a + jb)^* = a - jb$$

- The operator $(\ )^H$ is the Hermittian transpose defined as:

$$\mathbf{x}^H = (\mathbf{x}^*)^T = (\mathbf{x}^T)^*$$

- The filter output, weights and state are complex-valued,i.e.:

$$y[n] \in \mathbb{C}, \mathbf{w}[n], \mathbf{u}[n] \in \mathbb{C}^{N\times 1}$$

LMS Algorithm Step Size

- The LMS step size is chosen as:

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

- λ_{max} is the maximum eigenvalue of the input autocorrelation matrix $\mathbf{R}_{xx}$, i.e.

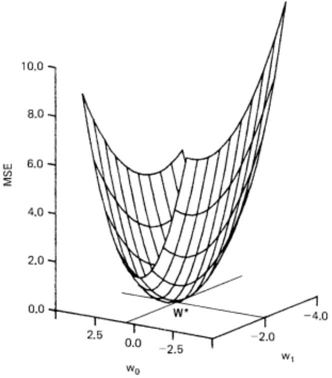
$$\lambda_{max} = \text{trace}[\mathbf{R}_{xx}] = \sum (\text{diagonal elements of } \mathbf{R}_{xx})$$

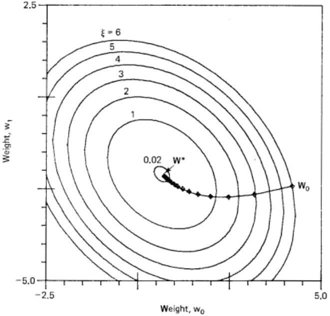
- For the FIR-based LMS filter:

$$0 < \eta < \frac{2}{(N+1)\sigma_x^2}$$

where σ_x^2 is the power of the input signal.

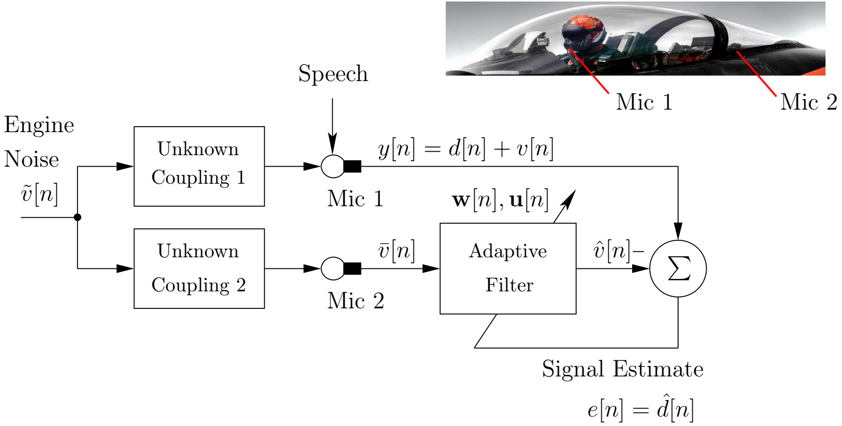
The Performance Surface $\xi[n]$



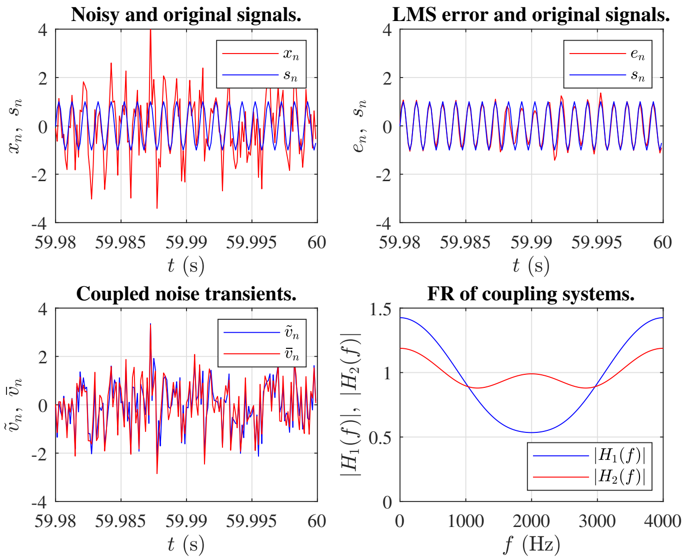


From Widrow & Stearns 1985, Adaptive Signal Processing, pp. 21

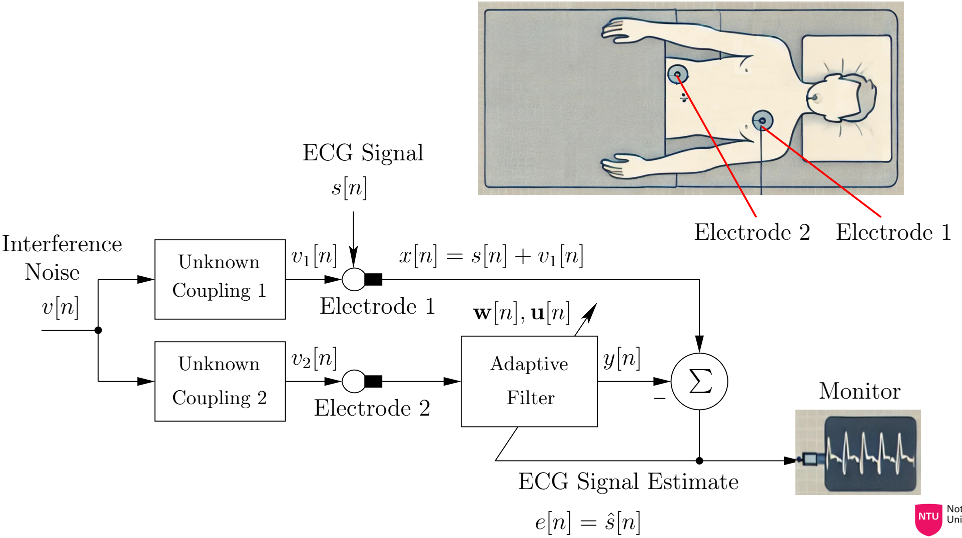
Example 1: LMS Noise Canceller

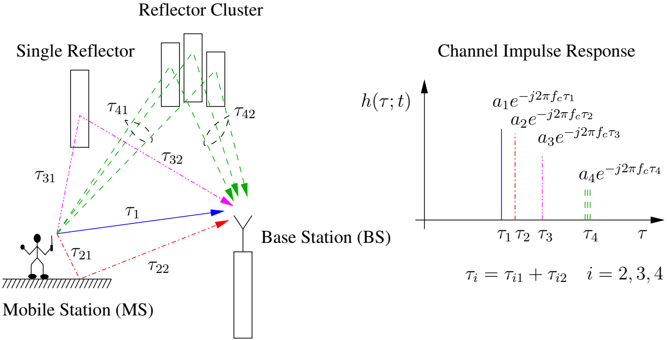


Example 1: LMS Noise Canceller (Cont.)



Example 2: LMS Interference Canceller

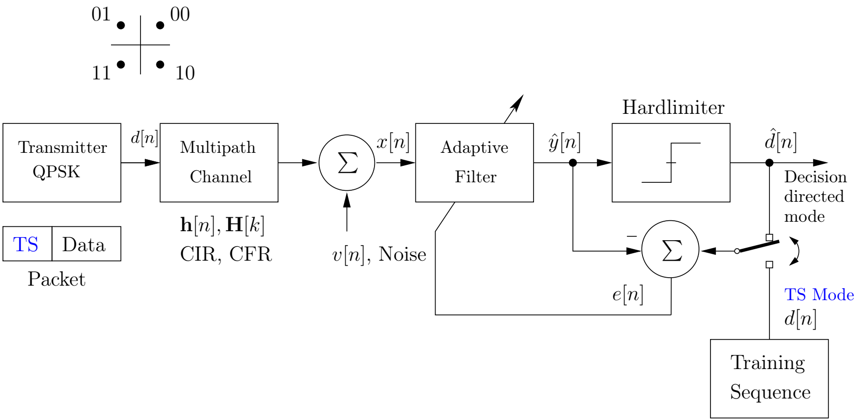




- Static multipath: High system bandwidth is selected to make channel time-invariant compared to movement, i.e. $\tau_l(t) = \tau_l, a_l(t) = a_l, \phi_l(t) = \phi_l$ and the channel impulse response becomes

$$h(\tau) = \sum_{l=1}^{L} a_l e^{j\phi_l} \delta(t - \tau_l)$$

Example 3: Channel Compensation Using the LMS Equalizer

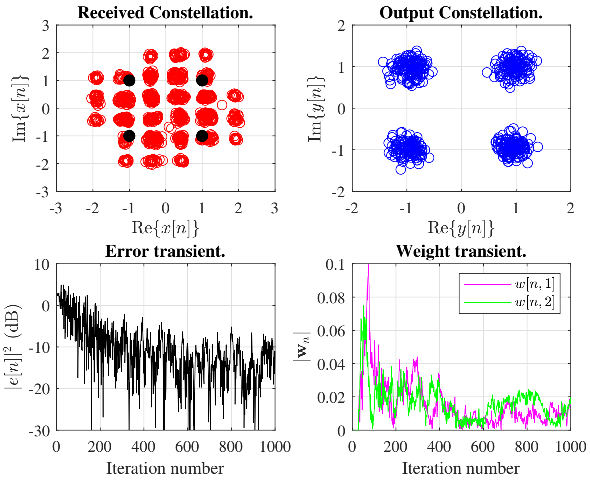


- The received signal is given as

$$y[n] = h[n] * d[n] + v[n]$$

Example 3: LMS Equalizer (Cont.)

QPSK through a multipath channel: $h[n] = e^{j0.5}\delta[n] - 0.2\delta[n-1] + 0.4\delta[n-4]$



Example 3: LMS Equalizer (Cont.)

FR of multipath channel: $H(e^{j\Omega}) = e^{j0.4} - 0.2e^{-j\Omega} + 0.4e^{-j4\Omega}$