ESE 531: Digital Signal Processing

Week 14

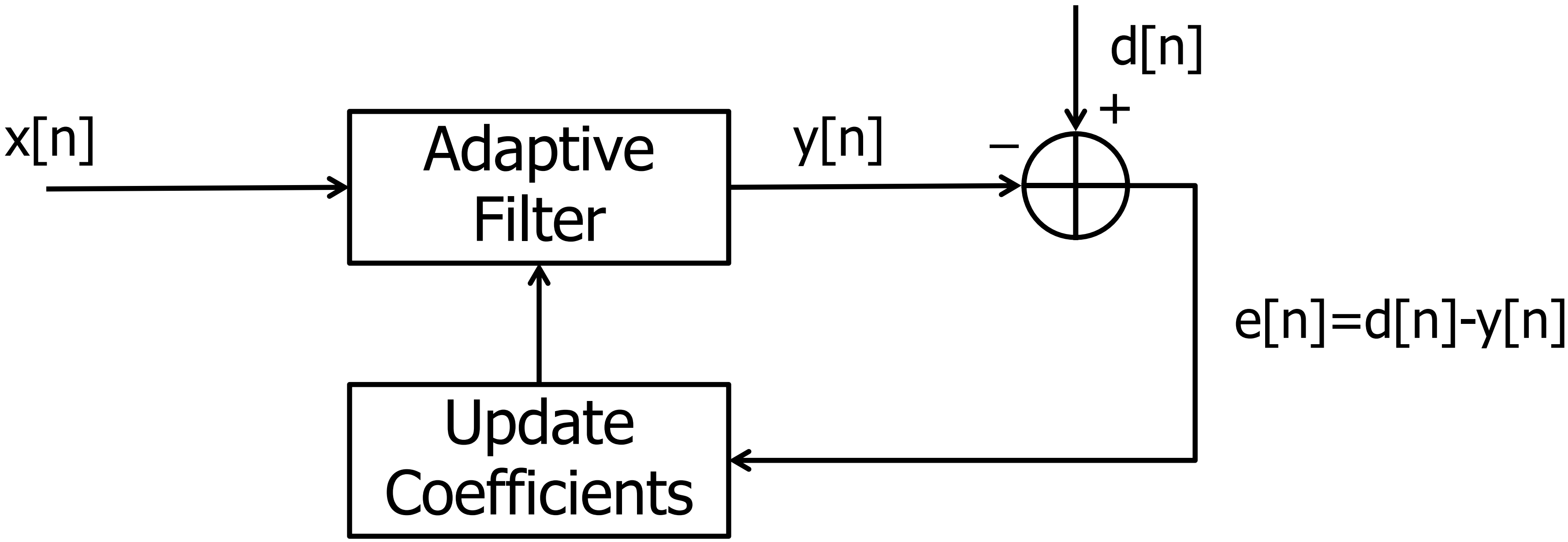
Lecture 25: April 18, 2021

Adaptive Filters

Adaptive Filters

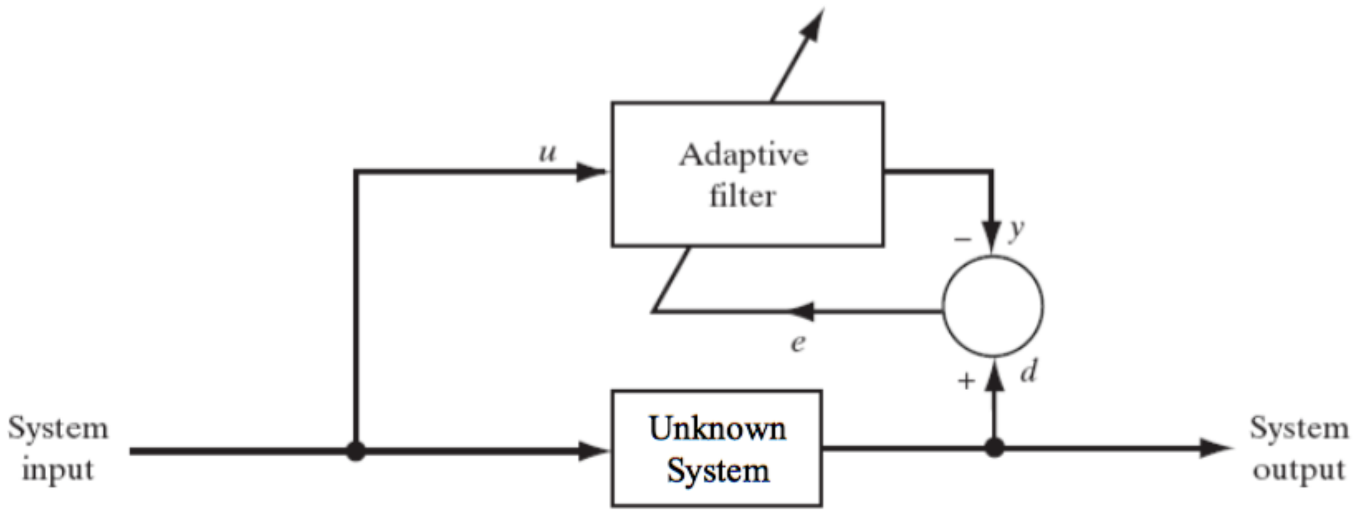
Adaptive Filters

- An adaptive filter is an adjustable filter that processes in time
  - It adapts…



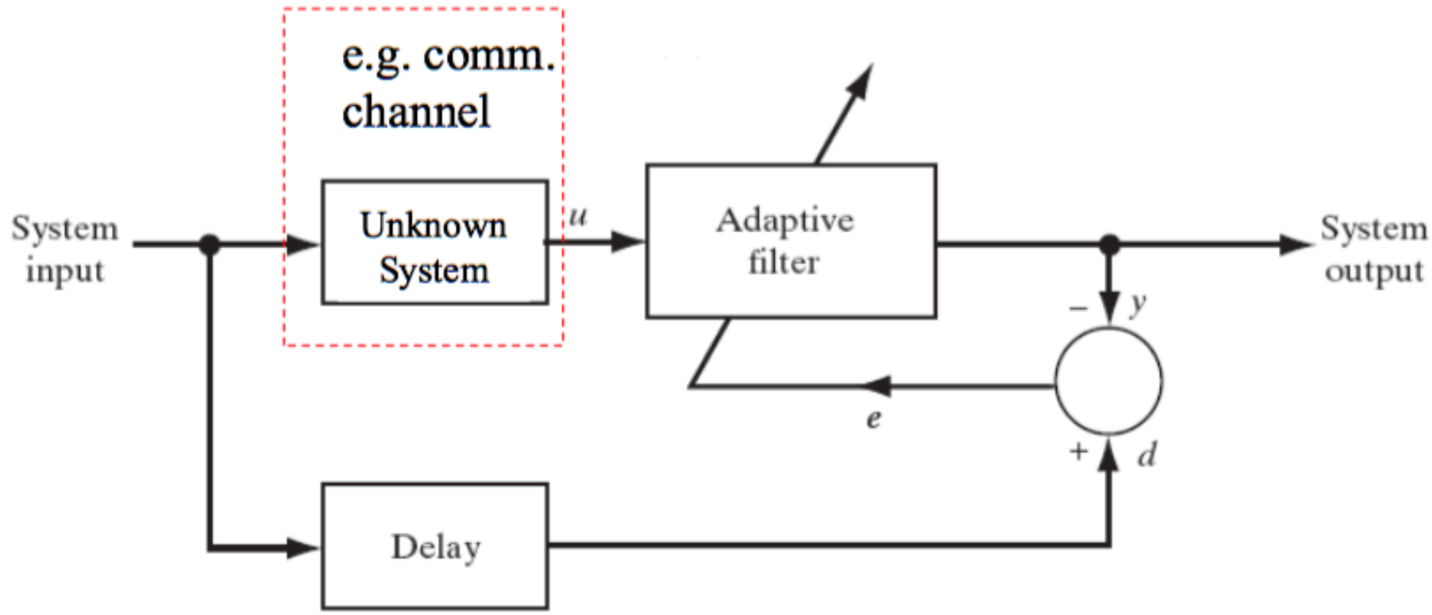
Adaptive Filter Applications

- System Identification



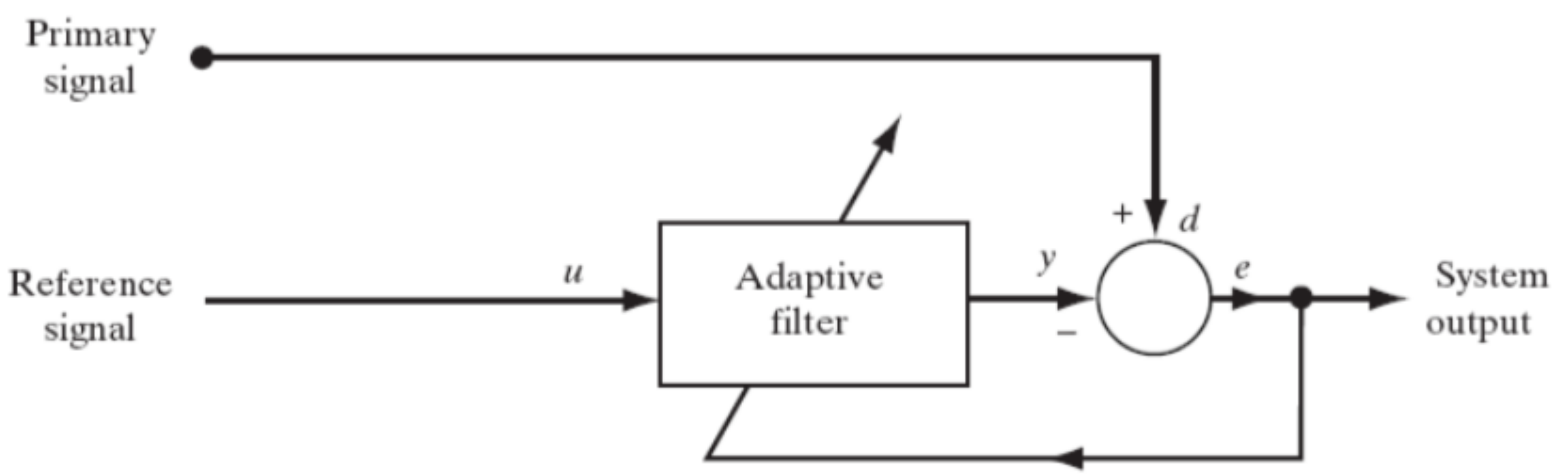
Adaptive Filter Applications

- Identification of inverse system



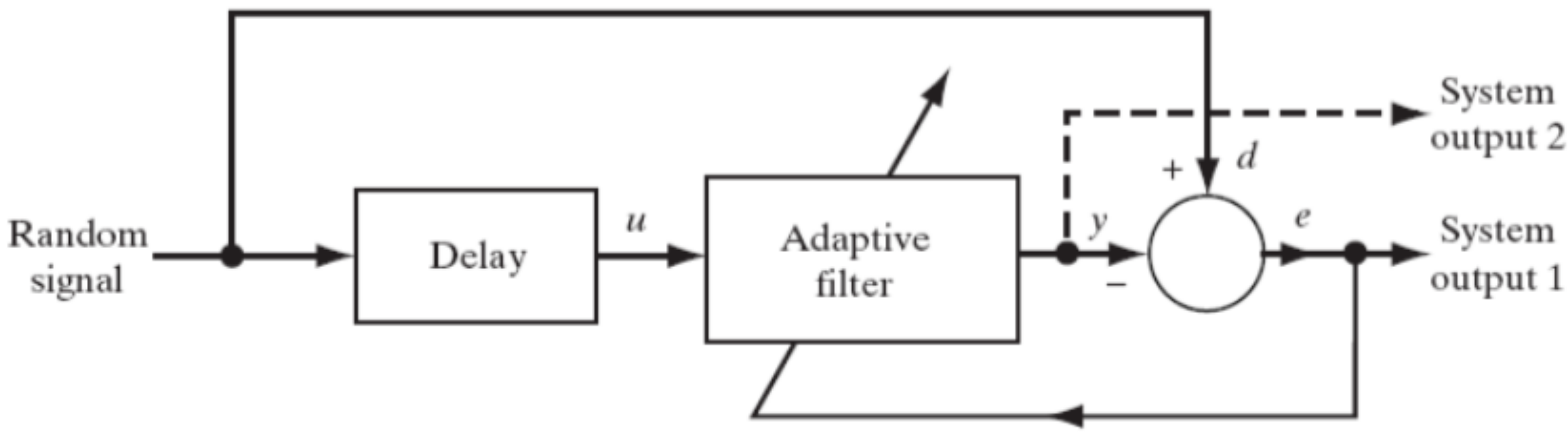
Adaptive Filter Applications

- Adaptive Interference Cancellation



Adaptive Filter Applications

- Adaptive Prediction

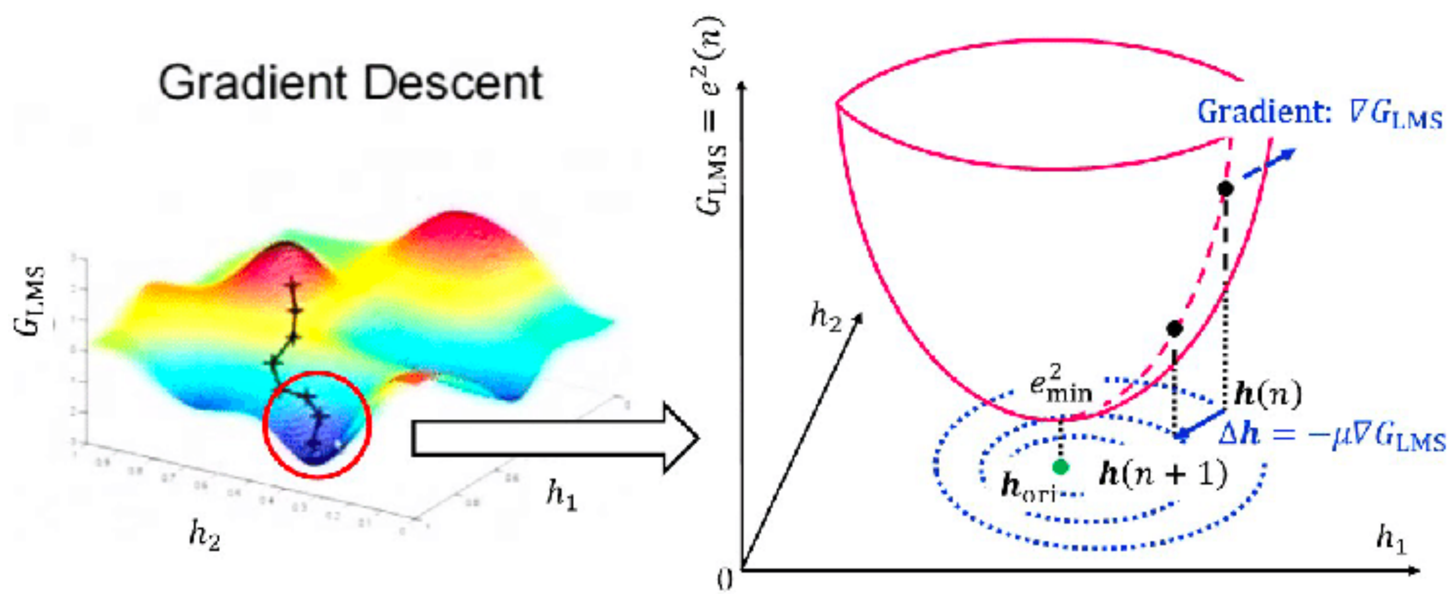


Stochastic Gradient Approach

- Most commonly used type of Adaptive Filters
- Define cost function as mean-squared error
    - Difference between filter output and desired response
- Based on the method of steepest descent
    - Move towards the minimum on the error surface to get to minimum
    - Requires the gradient of the error surface to be known

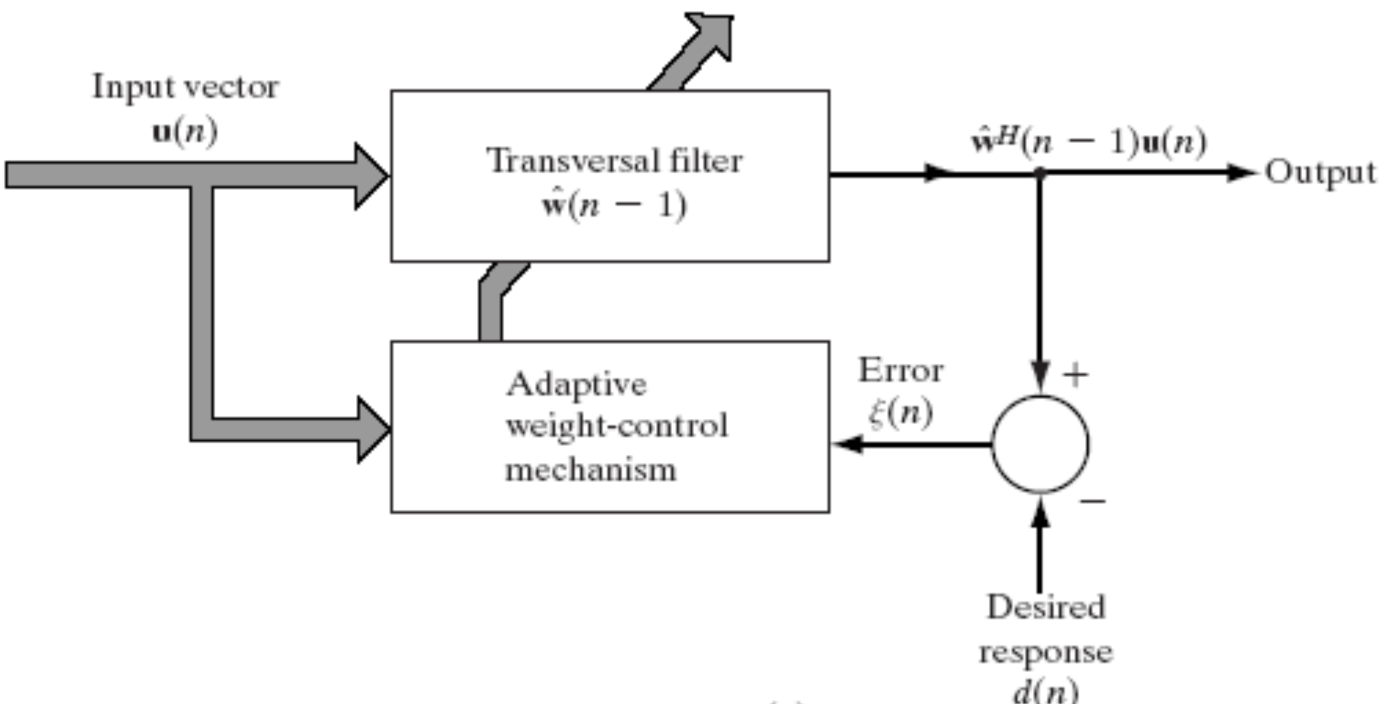
Stochastic Gradient Approach

- Most commonly used type of Adaptive Filters
- Define cost function as mean-squared error
    - Difference between filter output and desired response
- Based on the method of steepest descent
    - Move towards the minimum on the error surface to get to minimum
    - Requires the gradient of the error surface to be known



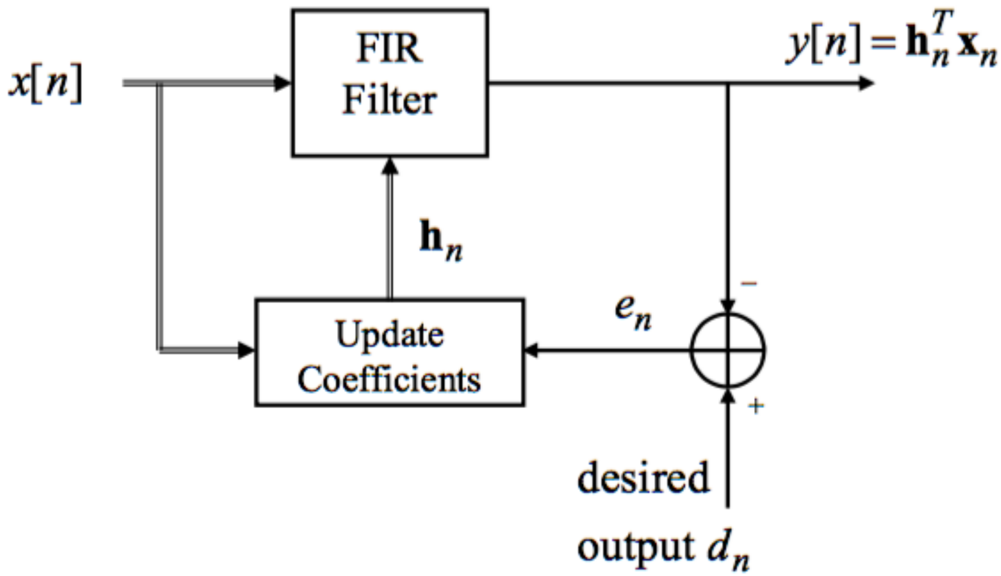
Least-Mean-Square (LMS) Algorithm

- The LMS Algorithm consists of two basic processes
  - Filtering process
    - Calculate the output of FIR filter by convolving input and taps
    - Calculate estimation error by comparing the output to desired signal
  - Adaptation process
    - Adjust tap weights based on the estimation error



(a)

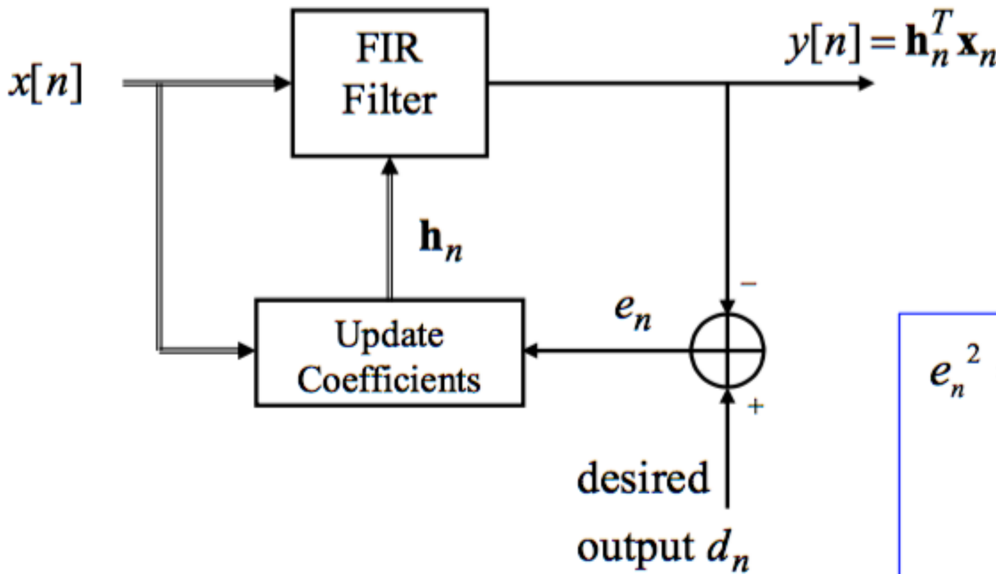
Adaptive FIR Filter: LMS



$$\mathbf{x}_n = (x[n], x[n-1]\ldots, x[n-M])^T$$

$$\mathbf{h}_n = (h_n[0], h_n[1], \ldots, h_n[M])^T$$

Adaptive FIR Filter: LMS

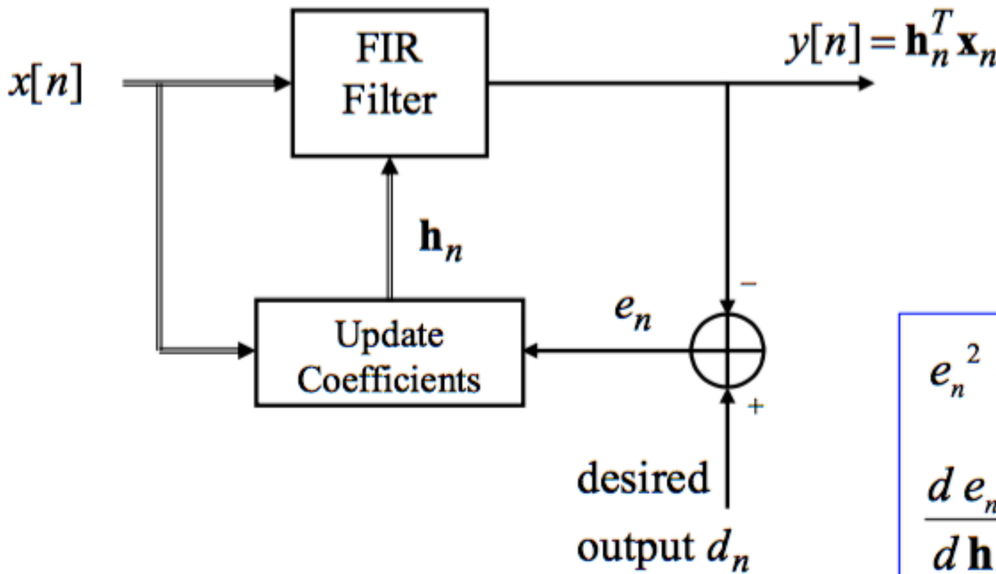


$$\mathbf{x}_n = (x[n], x[n-1]\ldots, x[n-M])^T$$

$$\mathbf{h}_n = (h_n[0], h_n[1], \ldots, h_n[M])^T$$

$$e_n^{\ 2} = (d[n] - y[n])^2 = (d[n] - \mathbf{h}_n^T \mathbf{x}_n)^2$$

Adaptive FIR Filter: LMS



$$\mathbf{x}_n = (x[n], x[n-1] \ldots, x[n-M])^T$$

$$\mathbf{h}_n = (h_n[0], h_n[1], \ldots, h_n[M])^T$$

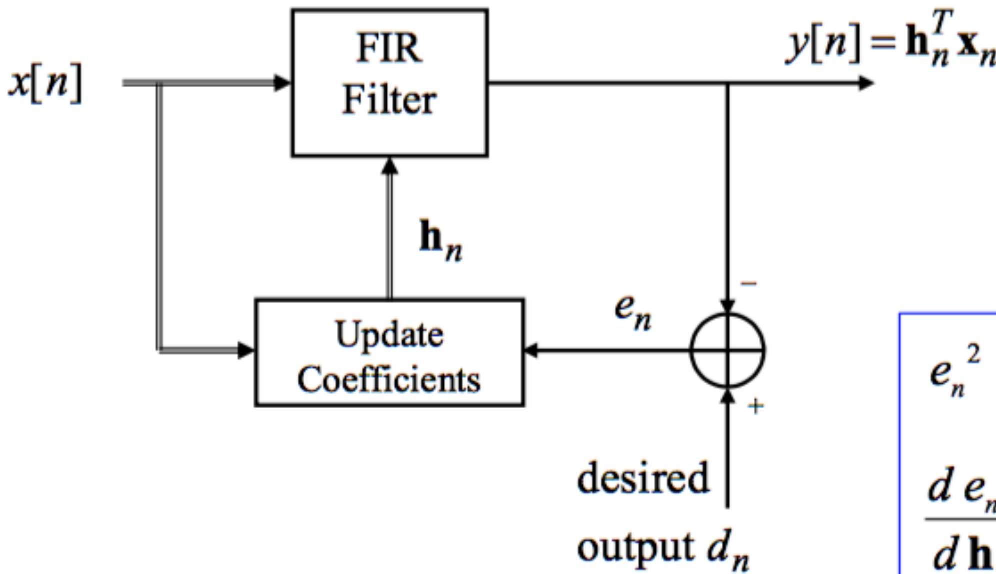
$$e_n^{\,2} = (d[n] - y[n])^2 = (d[n] - \mathbf{h}_n^T \mathbf{x}_n)^2$$

$$\frac{d\, e_n^{\,2}}{d\,\mathbf{h}_n} = -2(d[n] - \mathbf{h}_n^T \mathbf{x}_n)\mathbf{x}_n = -2e_n \mathbf{x}_n$$

$$\bar{a}^T \bar{b} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4$$

$$\frac{\partial\left(a^T b\right)}{\partial \bar{a}} = \begin{bmatrix} \frac{\partial\left(a^T b\right)}{\partial a_1} & \frac{\partial\left(a^T b\right)}{\partial a_2} & \frac{\partial\left(a^T b\right)}{\partial a_3} & \frac{\partial\left(a^T b\right)}{\partial a_4} \end{bmatrix}^T = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \bar{b}$$

Adaptive FIR Filter: LMS



$$\mathbf{x}_n = (x[n], x[n-1] \ldots, x[n-M])^T$$

$$\mathbf{h}_n = (h_n[0], h_n[1], \ldots, h_n[M])^T$$

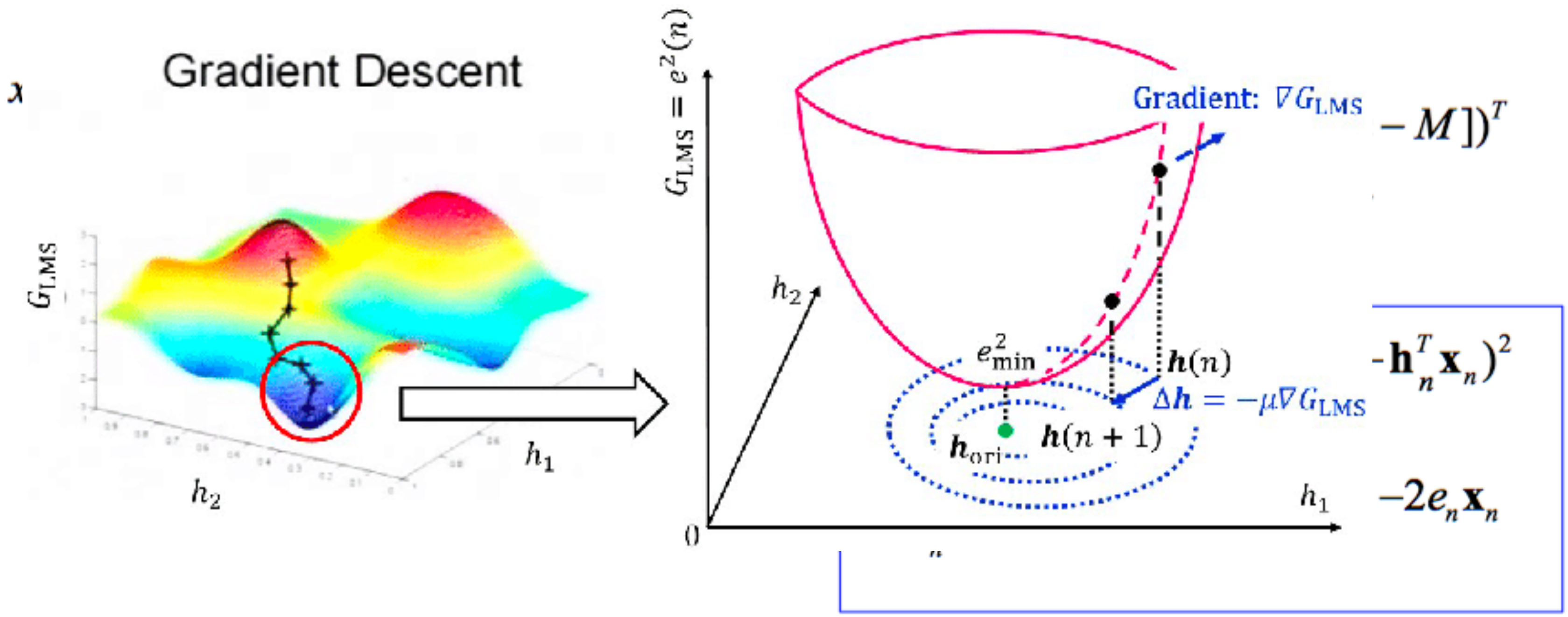
$$e_n{}^2 = (d[n] - y[n])^2 = (d[n] - \mathbf{h}_n^T \mathbf{x}_n)^2$$

$$\frac{d\, e_n{}^2}{d\,\mathbf{h}_n} = -2(d[n] - \mathbf{h}_n^T \mathbf{x}_n)\mathbf{x}_n = -2e_n \mathbf{x}_n$$

Coefficient Update: Move in direction *opposite* to sign of gradient, proportional to magnitude of gradient $\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + 2\mu e_n \mathbf{x}_n}$

Stochastic Gradient Algorithm

Adaptive FIR Filter: LMS

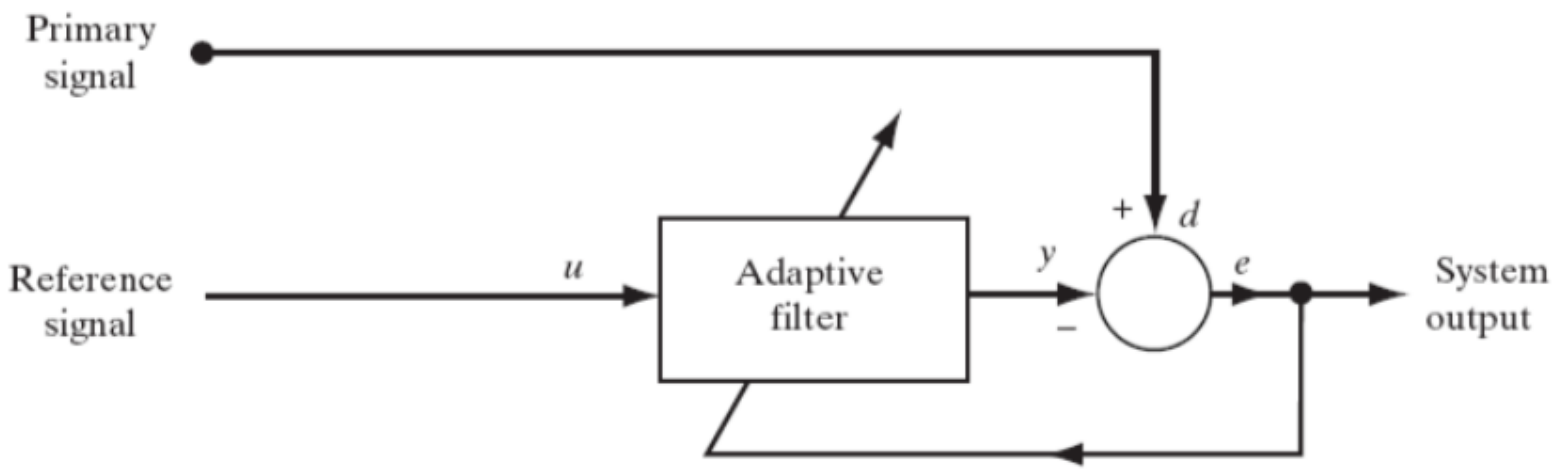


Coefficient Update: Move in direction *opposite* to sign of gradient, proportional to magnitude of gradient $\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + 2\mu e_n \mathbf{x}_n}$

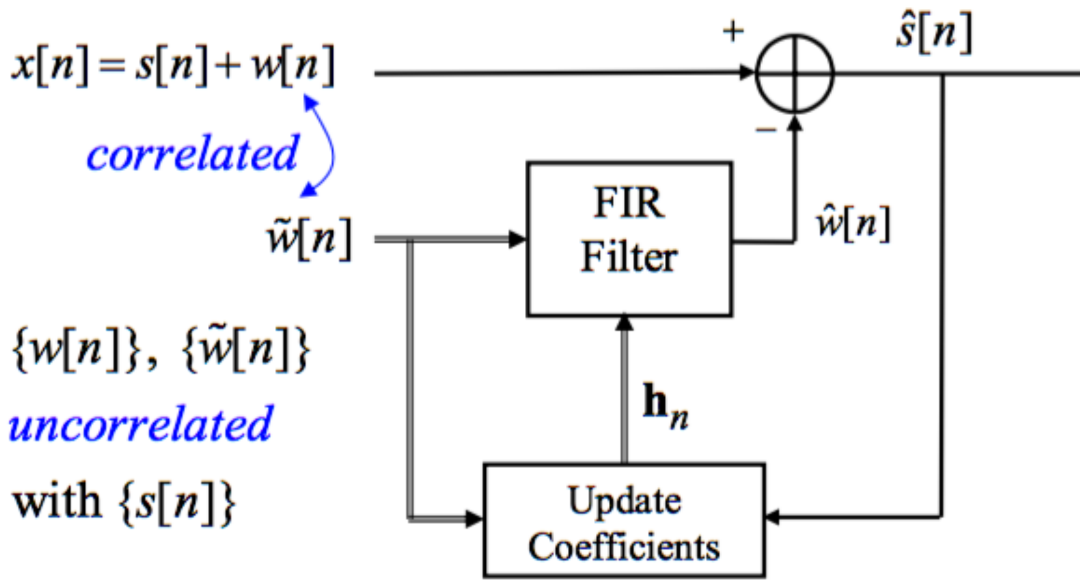
Stochastic Gradient Algorithm

Adaptive Filter Applications

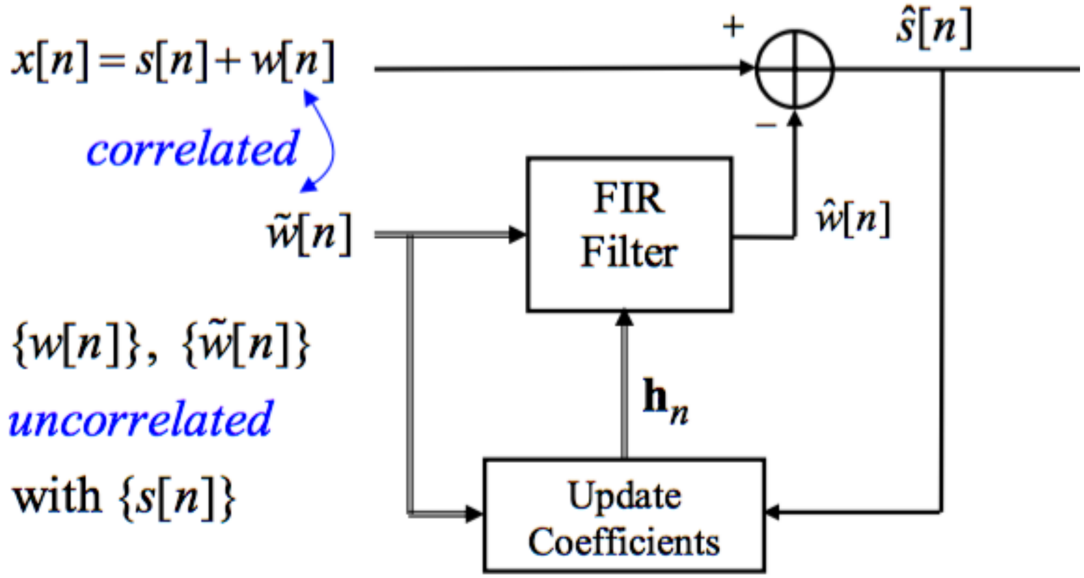
- Adaptive Interference Cancellation



Adaptive Interference Cancellation

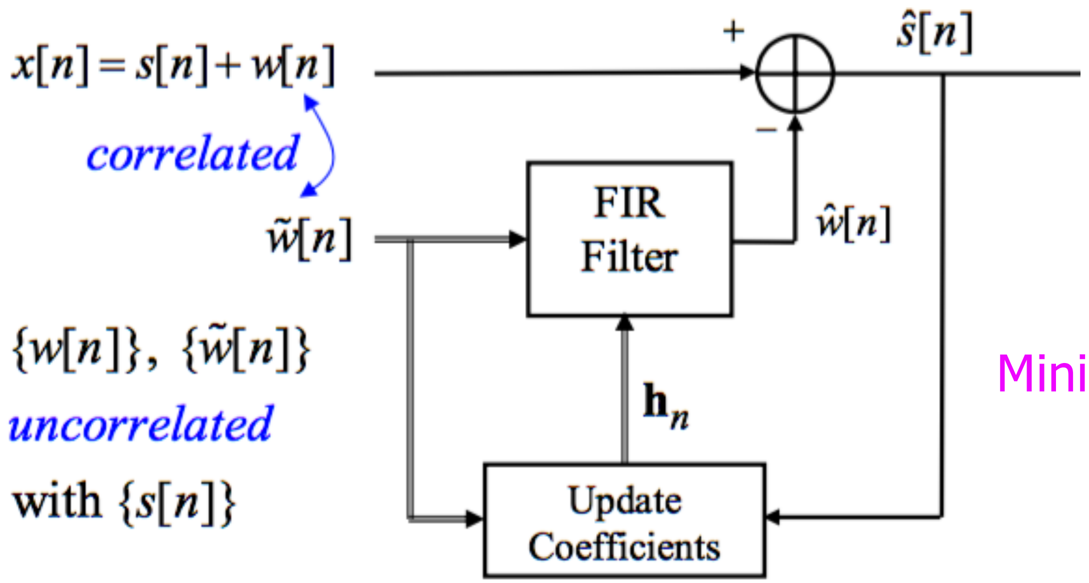


Adaptive Interference Cancellation



$$\hat{s}[n] = s[n] + w[n] - \hat{w}[n]$$
$$= s[n] + w[n] - h_n^T \tilde{w}_n$$

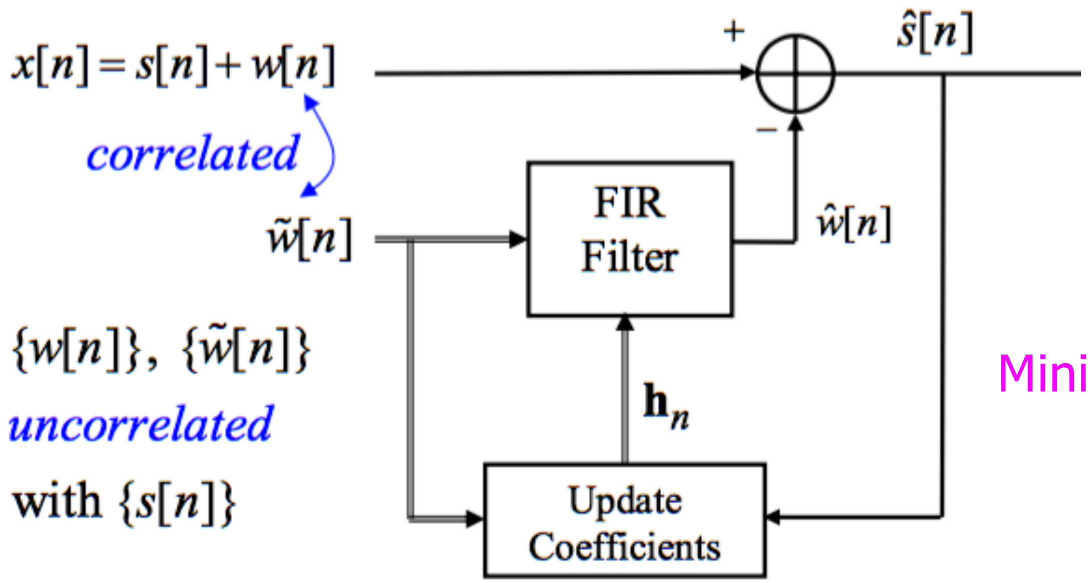
Adaptive Interference Cancellation



$$\hat{s}[n] = s[n] + w[n] - \hat{w}[n]$$
$$= s[n] + w[n] - h_n^T \tilde{w}_n$$

Minimizing $(\hat{s}[n])^2$ removes noise w[n]

Adaptive Interference Cancellation



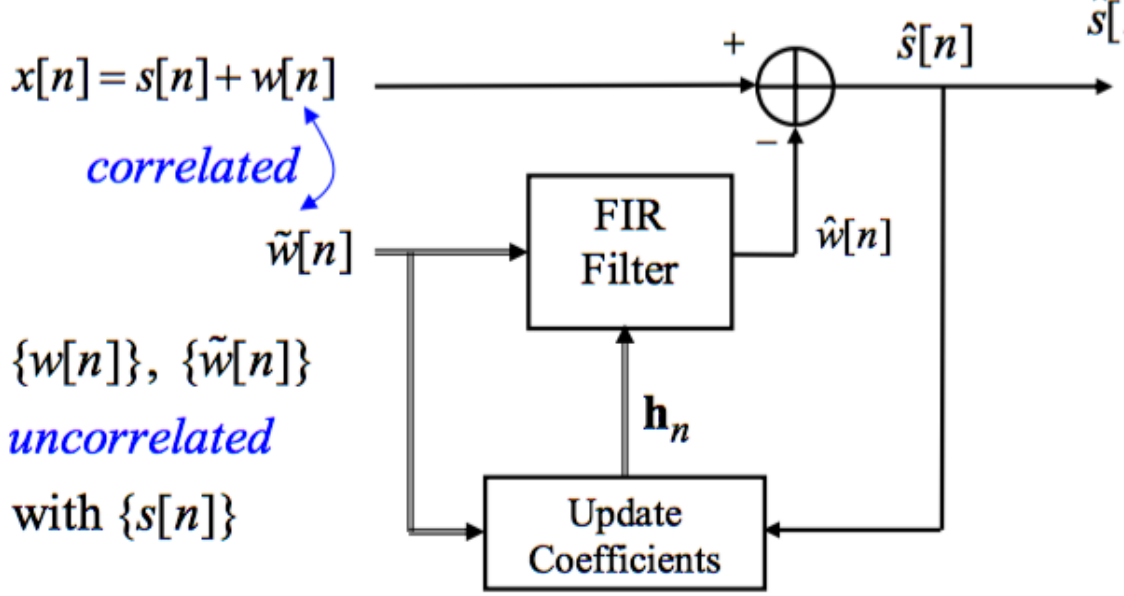
$$\hat{s}[n] = s[n] + w[n] - \hat{w}[n]$$
$$= s[n] + w[n] - h_n^T \tilde{w}_n$$

Minimizing $(\hat{s}[n])^2$ removes noise w[n]

$$\left(\hat{s}[n]\right)^2 = \left(s[n] + w[n] - h_n^T \tilde{w}_n\right)^2$$

$$\frac{\partial \hat{s}^2[n]}{\partial h_n} = 2\left(s[n] + w[n] - h_n^T \tilde{w}_n\right)(-\tilde{w}_n)$$
$$= 2\hat{s}[n](-\tilde{w}_n) = -2\hat{s}[n]\tilde{w}_n$$

Adaptive Interference Cancellation



$\{w[n]\}, \{\tilde{w}[n]\}$
uncorrelated
with $\{s[n]\}$

$$\hat{s}[n] = x[n] - \mathbf{h}_n^T \tilde{\mathbf{w}}_n$$
$$= s[n] + \{w[n] - \mathbf{h}_n^T \tilde{\mathbf{w}}_n\}$$

uncorrelated

Minimize *instantaneous power* $\hat{s}^2[n]$ for best interference cancellation

$$\frac{d\left(\hat{s}[n]\right)^2}{d\,\mathbf{h}_n} = -2\hat{s}[n]\tilde{\mathbf{w}}_n$$

$$\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + 2\mu\,\hat{s}[n]\tilde{\mathbf{w}}_n}$$

Stability of LMS

- The LMS algorithm is convergent in the mean square if and only if the step-size parameter satisfy

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

- Here λ_{max} is the largest eigenvalue of the correlation matrix of the input data
- More practical test for stability is

$$0 < \mu < \frac{2}{\text{input signal power}}$$

- Larger values for step size
  - Increases adaptation rate (faster adaptation)
  - Increases residual mean-squared error

Big Ideas

- Adaptive Filters
  - Use LMS algorithm to update filter coefficients
  - Applications like system ID, channel equalization, and signal prediction

Admin

- Project 2
  - Out now
  - Due 4/30
- Final Exam – 5/5
  - Administered in Canvas
    - 2hr timed exam in 12hr window
  - Open notes
  - Covers lec 1-24*
    - Doesn't include lecture 13