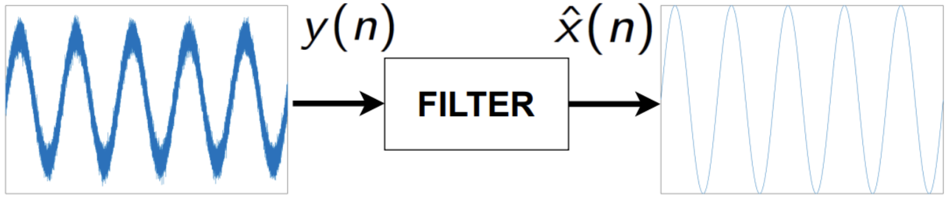
Introduction to Adaptive Filtering, Wiener-Hopf Equations, and Normal Equations

Jesper Rindom Jensen, jrj@es.aau.dk

Advanced Signal Processing, Electronic Systems, Aalborg University

Introduction to Adaptive Filtering

- **Filter:** system that is designed to extract information about a prescribed quantity of interest from noisy data
  - $y(n) = x(n) + v(n)$, where $v(n) \sim \mathcal{N}$ is noise



  - Filters, $f(\cdot)$ can be linear or non-linear: $\hat{x}(n) = f(y(n))$
  - *Example 1:* the (linear) Wiener filter is optimum in the mean-squared error sense
  - *Example 2:* the (linear) Kalman filter is more appropriate to deal with non-stationary signals.

Introduction to Adaptive Filtering

- **Adaptive filter**: filter relying on a recursive algorithm that makes it possible to perform well under non-stationary conditions (tracking)
  - Non-linear systems (data dependent)

- That said, an adaptive filter $f(\cdot)$ is linear if it follows the principle of superposition whenever its parameters are fixed:
  - Additivity: $f(y_1(n) + y_2(n)) = f(y_1(n)) + f(y_2(n))$
  - Homogeneity: $f(\alpha y(n)) = \alpha f(y(n))$

Introduction to Adaptive Filtering

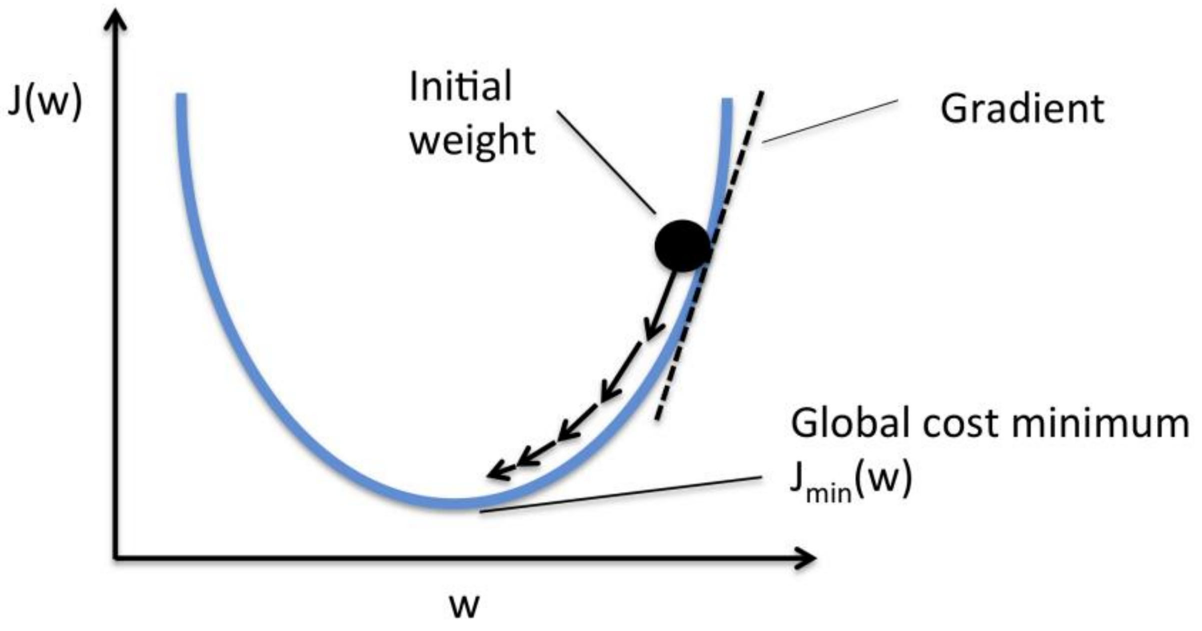
- A reminder…
  - FIR: $\hat{x}(n) = \sum_{i=0}^{P} b_i y(n-i)$
  - IIR: $\hat{x}(n) = \frac{1}{a_0}\left(\sum_{i=0}^{P} b_i y(n-i) - \sum_{j=1}^{Q} a_j \hat{x}(n-j)\right)$
- Unlike IIR filters, FIR filters are inherently stable ⇒ FIR filters as the structural basis for the design of linear adaptive filters.

Introduction to Adaptive Filtering

Two main approaches to derive recursive algorithms for linear adaptive filters:

- **Stochastic Gradient Descent (SGD):**

Least Mean Square (LMS) algorithm

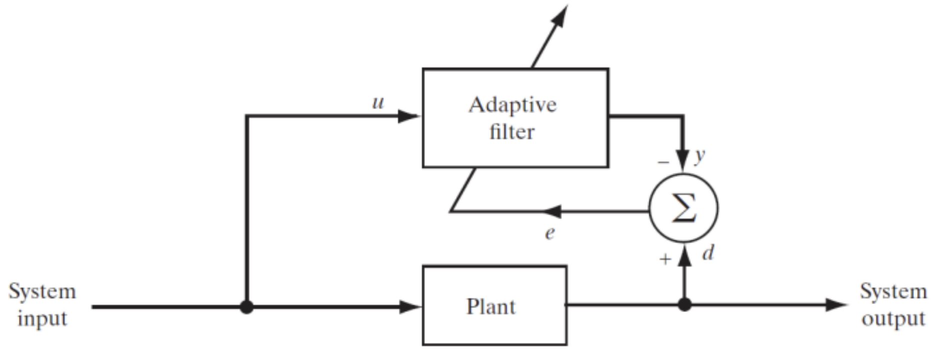


Introduction to Adaptive Filtering

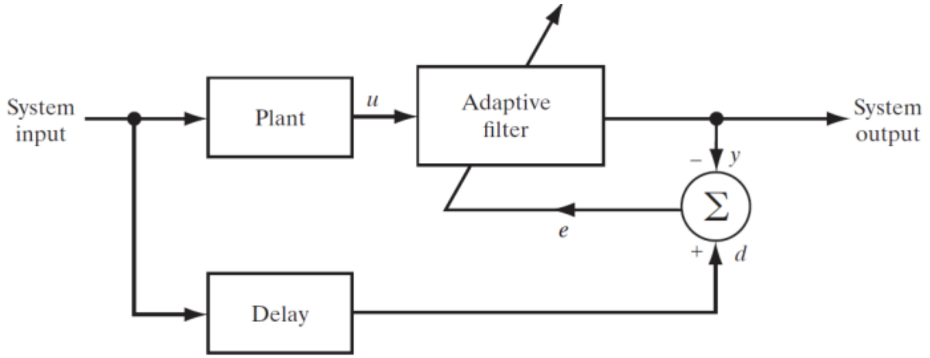
- **Least Squares:** mean squared error (MSE) minimization is carried out using algebraic matrix manipulations unlike SGD.
  - Recursive least-squares (RLS) algorithm
- RLS converges faster and has a higher computational complexity than LMS.

Introduction to Adaptive Filtering

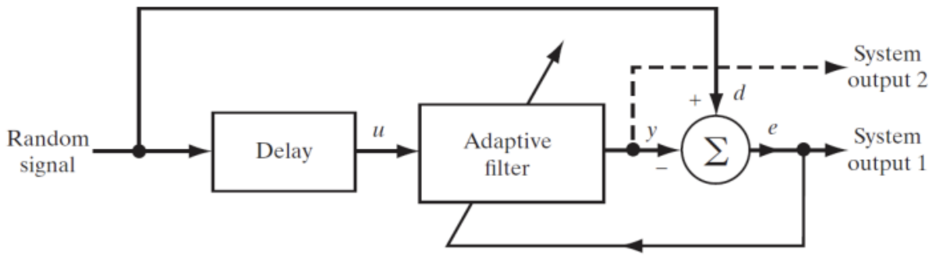
Applications



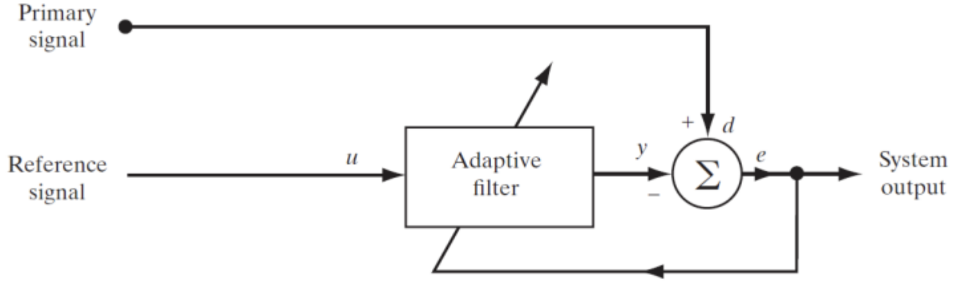
System identification



Inverse modeling (equalization)



Prediction



Noise cancellation

Stochastic Processes: The Basics

- **Stochastic process**: random process or statistical phenomenon evolving over time (e.g., speech signal)
- A stochastic process can be either continuous or discrete, e.g., $u(n)$
- **Definitions:**
  - Mean: $\mu(n) = E[U(n)]$
  - Autocorrelation: $r(n, n-k) = E[u(n)u^*(n-k)]$
  - Autocovariance: $c(n, n-k) = E[(u(n) - \mu(n))(u(n-k) - \mu(n-k))^*]$
    $c(n, n-k) = r(n, n-k) - \mu(n)\mu^*(n-k)$
- A stochastic process is a wide-sense stationary (WSS) process if…
  - $\mu(n) = \mu \forall n$
  - $r(n, n-k) = r(k) \wedge c(n, n-k) = c(k)$

Stochastic Processes: The Basics

$$\mathbf{u}(n) = \begin{bmatrix} u(n) & u(n-1) & \cdots & u(n-M+1) \end{bmatrix}^T$$

Correlation matrix:

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)] = \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r(-1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M+1) & r(-M+2) & \cdots & r(0) \end{bmatrix}$$

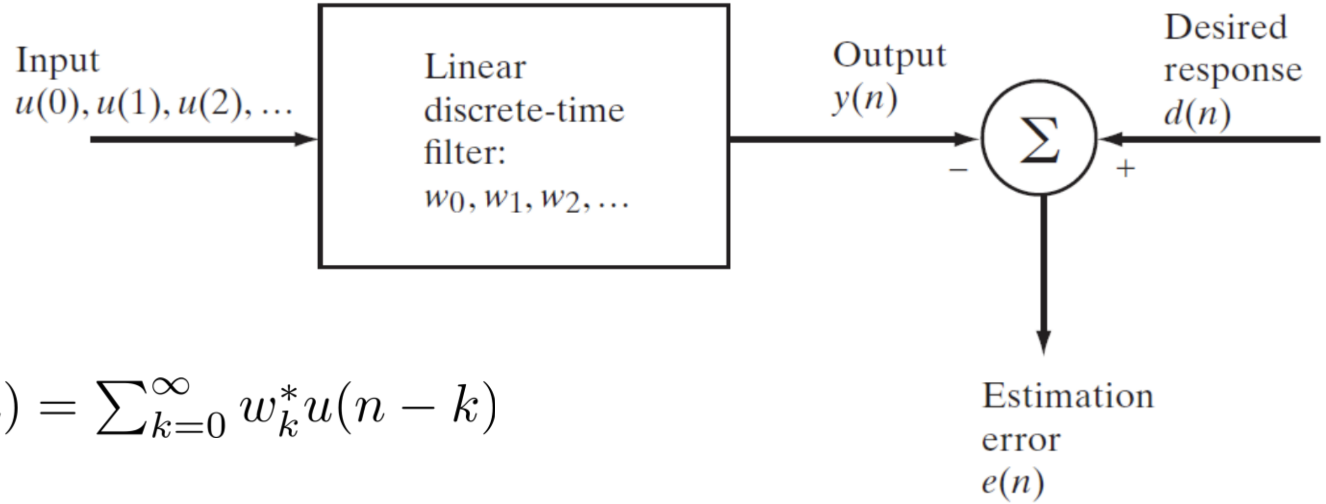
The correlation matrix is Hermitian, i.e., $\mathbf{R} = \mathbf{R}^H$, so $r(-k) = r^*(k)$

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)] = \begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \cdots & r(0) \end{bmatrix}$$

Toeplitz

Wiener Filtering

- **Wiener filters:** a class of linear optimum discrete-time filters



$$y(n) = \sum_{k=0}^{\infty} w_k^* u(n-k)$$

- $u(n)$ and $d(n)$ are **WSS stochastic processes** with zero mean
- The goal is to design a filter minimizing the MSE:

$$J = E[e(n)e^*(n)] = E[|e(n)|^2] = E[|d(n) - y(n)|^2]$$

Wiener Filtering

- **Wiener optimal solution:**

$$\nabla_k J = \nabla_k E[e(n)e^*(n)] = -2E[u(n-k)e^*(n)]$$

$$\nabla_k J = E[u(n-k)e_o^*(n)] = 0$$

- **Principle of orthogonality:** when the filter operates optimally (i.e., J is minimum), $e_o(n)$ is orthogonal to each input sample $u(n-k)$ contributing to the estimation of the desired response
- *Corollary to the principle of orthogonality*:

$$E[y_o(n)e_o^*(n)] = 0$$

Wiener-Hopf Equations

- We can further develop the Wiener optimal solution:

$$E[u(n-k)e_o^*(n)] = E\left[u(n-k)\left(d^*(n) - \sum_{i=0}^{\infty} w_{oi} u^*(n-i)\right)\right] = 0$$

$$\sum_{i=0}^{\infty} w_{oi} E[u(n-k)u^*(n-i)] = E[u(n-k)d^*(n)]$$

- Notice that…
  - $E[u(n-k)u^*(n-i)] = r(i-k)$
  - $E[u(n-k)d^*(n)] = p(-k)$

- **Wiener-Hopf equations:** $\sum_{i=0}^{\infty} w_{oi} r(i-k) = p(-k)$

Wiener-Hopf Equations

- For a FIR filter with *M* weights ⇒ $\sum_{i=0}^{M-1} w_{oi} r(i-k) = p(-k)$

- **In matrix form:** $\mathbf{R}\mathbf{w}_o = \mathbf{p}$
- Notice that…
  - $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$
  - $\mathbf{w}_o = \begin{bmatrix} w_{o,0} & w_{o,1} & \cdots & w_{o,M-1} \end{bmatrix}^T$
  - $\mathbf{p} = E[\mathbf{u}(n)d^*(n)] = \begin{bmatrix} p(0) & p(-1) & \cdots & p(1-M) \end{bmatrix}^T$
- Then, the optimal filter weights are simply calculated as

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$$

Least-Squares Filtering

- Wiener filtering (*ensemble averages*, WSS) *versus* least squares (*batch-processing approach*)
- Multiple linear regression model:

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

Measurement error

- **Linear least-squares filter:** set of weights minimizing

$$\sum_{i=i_1}^{i_2} |e(i)|^2$$

- The covariance method makes no assumptions about the data ($u(1), u(2), \ldots, u(N)$) outside the interval $[1, N]$, i.e., $i_1 = M$ and $i_2 = N$.

Least-Squares Filtering

- As for Wiener filtering…

$$\mathcal{E} = \sum_{i=M}^{N} e(i)e^*(i) = \sum_{i=M}^{N} |e(i)|^2$$

$$\nabla_k \mathcal{E} = 0$$

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

- **Principle of orthogonality**: when the FIR filter operates optimally (i.e., in its least-squares condition), the minimum-error time series $e_{\min}(i)$ is orthogonal to the time series $u(i - k)$

- *Corollary to the principle of orthogonality:* $\sum_{i=M}^{N} y_{\min}(i)e^*_{\min}(i) = 0$

Normal Equations

- **Normal equations**: an alternative to the principle of orthogonality to describe the least-squares condition of a FIR filter

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

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

- Notice that...
  - $\sum_{i=M}^{N} u(i-k)u^*(i-t) = \phi(t,k)$
  - $\sum_{i=M}^{N} u(i-k)d^*(i) = z(-k)$
- **Normal equations**: $\sum_{t=0}^{M-1} \hat{w}_t \phi(t,k) = z(-k)$

Normal Equations

$$\mathbf{\Phi} = \begin{bmatrix} \phi(0,0) & \phi(1,0) & \cdots & \phi(M-1,0) \\ \phi(0,1) & \phi(1,1) & \cdots & \phi(M-1,1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(0,M-1) & \phi(1,M-1) & \cdots & \phi(M-1,M-1) \end{bmatrix}$$

$$\mathbf{z} = \begin{bmatrix} z(0) & z(-1) & \cdots & z(-M+1) \end{bmatrix}^T$$

$$\hat{\mathbf{w}} = \begin{bmatrix} \hat{w}_0 & \hat{w}_1 & \cdots & \hat{w}_{M+1} \end{bmatrix}^T$$

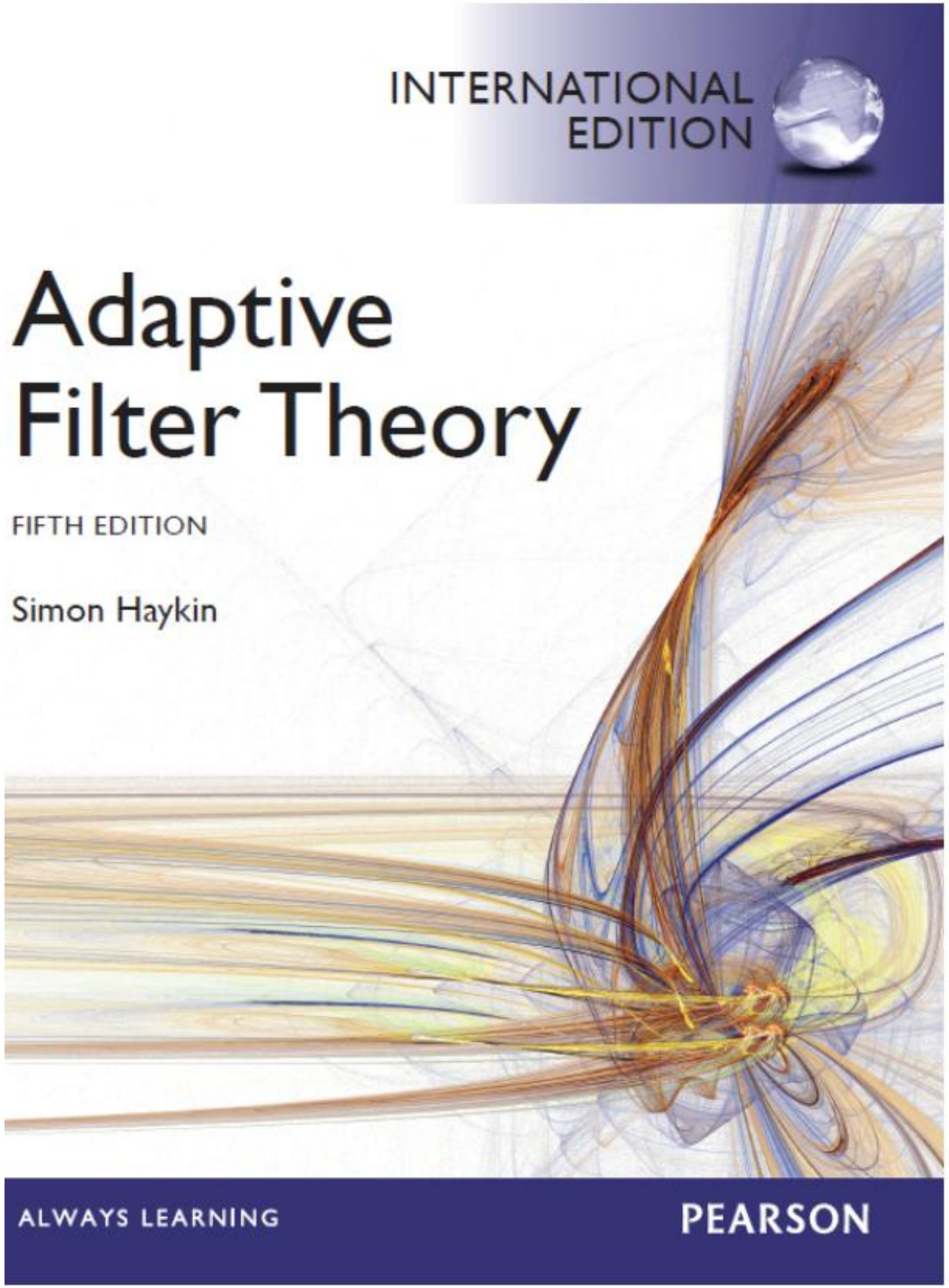
- **Normal equations:** $\mathbf{\Phi}\hat{\mathbf{w}} = \mathbf{z}$
- Then, the optimal set of weights can be calculated as

$$\hat{\mathbf{w}} = \mathbf{\Phi}^{-1}\mathbf{z}$$

Bibliography

- Simon Haykin, “Adaptive Filter Theory (5th Edition)”. Pearson, 2014

  - Introduction to Adaptive Filtering: Background and Preview
  - Stochastic Processes: The Basics: 1.1 and 1.3
  - Wiener Filtering: 2.1 and 2.2
  - Wiener-Hopf Equations: 2.4
  - Least-Squares Filtering: 9.1, 9.2 and 9.3
  - Normal Equations: 9.5



Assignment: Wiener Filtering

Consider the autoregressive process described by the difference equation

$$d(n) = 0.75d(n-1) + v(n)$$

and the noisy process

$$u(n) = d(n) + w(n) + 0.5w(n-1)$$

where both $v(n)$ and $w(n)$ are white noise with zero mean and unit standard deviation, and $E[v(n)w(n)] = 0$. We are interested in a **first-order Wiener filter** estimating $d(n)$ from $u(n)$.

- Draw the block diagram of the system
- Write the error signal $e(n)$ and the cost function to be minimized
- Calculate, **by hand**, expressions for the optimal set of weights as well as obtain the values of the optimal weights (**explain, step by step, how you do it!**)

Assignment: Wiener Filtering

- Obtain a numerical approximation of the optimal weights. For that, using, e.g., MatLab or Python, generate 10,000 samples of $d(n)$ and $u(n)$, where $d(0) = w(0) = 0$, and apply the expressions calculated in 3). Compare the result with that from 3)
- Using the estimated Wiener filter, apply it to u(n) to calculate the estimate $\hat{d}(n)$. Both by hand and using numerical approximation, calculate the error variances

$$\sigma_{du}^2 = E[(d(n) - u(n))^2]$$

- and

$$\sigma_{d\hat{d}}^2 = E\left[\left(d(n) - \hat{d}(n)\right)^2\right]$$

- and compare them (when by **hand, explain, step by step, how you do it!**)