VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECHTURE – CO2008

Assignment (Semester: 251)

FILTERING AND PREDICTION SIGNAL WITH WIENER FILTER

Advisor: Nguyễn Thành Lộc

Student(s): Võ Thiện Toàn - 2453265 (CC05)

Ho Chi Minh City, October 2025

Introduction

The Wiener Filter is s recognized as a classic adaptive filtering technique within the field of signal processing. The fundamental objective of this filter is to estimate a desired signal $s(n)$, from an observed input signal $x(n)$, that has been contaminated by noise or interference $w(n)$. Generally, the input signal $x(n)$ is the summation of the desired signal $s(n)$ and the undesired noise $w(n)$. The system is designed to suppress the additive interference while simultaneously preserving the essential characteristics of the desired signal $s(n)$.

The core principle used for optimizing the filter impulse response $h(n)$, is the minimization of the mean-square error (MMSE). This criterion is favored due to its simplicity and mathematical tractability. The optimum linear filter, defined by the criterion of MMSE is referred to as a Wiener Filter. The theoretical derivation of the MMSE involves the use of derivatives and relates directly to the *Principle of orthogonality.* The set of linear equations utilized to determine the optimized coefficients $h(k)$, are known as the *Wiener-Hopf equations*.

The general linear estimation problem distinguishes three special cases:

- $d(n) = s(n)$
- $d(n) = s(n + D)$ (signal predication)
- $d(n) = s(n - D)$ (signal smoothing)

The final MMSE achieved by the Wiener Filter is expressed mathematically in relation to the variance of the desired signal σ_d^2 and the optimized filter coefficients $h_{opt}(k)$.

In real-world applications, the input signal $x(n)$ is the sum of desired signal $s(n)$ and an interference or noise signal $w(n)$. The Wiener Filter is tasked with finding the optimal set of filter coefficients h_{opt} such that the output signal $y(n)$ approximates the desired signal $d(n)$ as closely as possible.

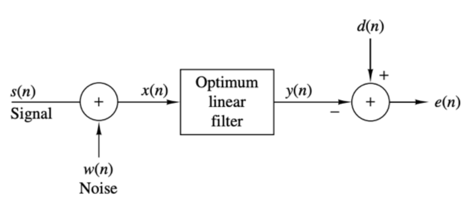

Fig.1: An example of Wiener Filter Linear Optimum Filtering

Theoretical background

Principle of orthogonality: In statistics and signal processing, the orthogonality principle is a necessary and sufficient condition for the optimality of a Bayesian estimator. Loosely stated, the orthogonality principle says that the error vector of the optimal estimator (in a mean square error sense) is orthogonal to any possible estimator. The orthogonality principle is most commonly stated for linear estimators, but more general formulations are possible. Since the principle is a necessary and sufficient condition for optimality, it can be used to find the minimum mean square error estimator.

Gaussian Elimination (row reduction): a mathematical algorithm for solving systems of linear equations by using a sequence of elementary row operations on the augmented matrix to transform it into upper triangular matrix form.

- **The core objective:** the goal is to find the unique set of optimal filter coefficients $\boldsymbol{h} = [h_0, h_1, \dots, h_{M-1}]^T$, that satisfies the linear system derived from minimizing the MMSE:
$$A \cdot x = B$$
Where $A = R_M$ (Autocorrelation Matrix) and $B = \gamma_d$ (Cross-correlation Vector)

- **Forward elimination:** the objective of this phase is to transform the system matrix A into an upper triangular matrix. This mean all elements below the main diagonal are eliminated (set to zero). his process is executed iteratively, pivoting on the diagonal elements $A[k][k]$.
 - **Process:** For every pivot row k (from 0 to $M-2$), the algorithm iterates through all rows i below it ($i = k+1\ to\ M-1$).
 - **Factor:** A multiplier is calculated for each row i: $Factor = \frac{A[i][k]}{A[k][k]}$
 - **Row operation:** Row k is used to eliminate the non-zero element $A[i][k]$. This operation is applied to the entire row i and to the corresponding element $B[i]$:
$$New\ A[i][j] = A[i][j] - Factor \times A[k][j]$$
$$New\ B[i] = B[i] - Factor \times B[k]$$

- **Back substitution:** Once the matrix A is in upper triangular form, the system is much easier to solve. We can start from the bottom equation (which has only one unknown) and substitute the solution upwards.
$$h_i = \frac{B[i] - \sum_{j=i+1}^{M-1}(A[i][j] \cdot h_j)}{A[i][i]}$$

System Definition and Output (Formula 1): The filter operates on an input signal $x(n)$ of finite length N and use a sets of coefficients $\boldsymbol{h}$, where M is the filter order.

- **Output signal:** The output $Y(n)$ is a linear convolution of the input signal and the filter coefficients.

$$y(n) = \sum_{k=0}^{M-1} h_k x(n-k) \quad (Formula\ 1)$$

- $y(n)$: $Output\ signal\ that\ the\ filter\ calculated$

- h_k: $Optimal\ filter\ coefficients$

- M: $The\ length\ of\ the\ filter$

- k: $Time\ lag, ranging\ from\ 0\ to\ M-1\ (0\ \leq k \leq M-1)$

- $x(n-k)$: $The\ value\ of\ the\ input\ signal\ at\ a\ delay\ of\ k\ and\ current\ time$

- We can express the output $y(n)$ as a linear combination of the input sequence $h_0x(n) + h_1x(n-1) + h_2x(n-2) + \ldots + h_{M-1}x(n-M+1)$. This expasion explicitly demostrates the memory structure of the filter.
- These coefficients acts as optimal weights basd on statistical mechanisms, effectively amplifying highly correlated signal components while suppressing uncorrelated random noise. Consequently, $y(n)$ represents the optimal estimate of the desired signal, minimizing the estimation error.

Mean-square value of the error (Formula 2): it represents the cost function that the Wiener Filter algorithm must minimize. It is the core mathematical expression that defines what it means for the filter to be "optimal".

$$E_M = \mathbb{E}[|e(n)|^2] = \mathbb{E}\left|D(n) - \sum_{k=0}^{M-1} h_k x(n-k)\right|^2 \ (Formula\ 2)$$

- $E_M(MSE)$: The single numerical quantity that the entire Wiener filter calculation aims to minimize.

- $e(n)$: Instantaneous Error at time n $(e(n) = d(n) - y(n))$

- $\mathbb{E}$: Expectation (Mean or Average), this operator takes the average of the squared error over time (or across all available data samples N). It transforms the instantaneous error into a single, reliable measure of filter performance.

- The goal of the Wiener filter is to find the set of filter coefficients $\boldsymbol{h} = [h_0, h_1, \dots, h_{M-1}]$ that drives the value of E_M to its minimum possible point. Mathematically, this minimization is achieved by performing partial differentiation of E_M with respect to each filter coefficient h_k and setting the result to zero:

$$\frac{\partial E_M}{\partial h_k} = 0, \qquad for\ k = 0, 1, 2, \dots, M-1$$

  This process is based on the *Principle of Orthogonality*, which states that the residual error must be uncorrelated (orthogonal) to the input data used to compute it.

- Formula (2) is not directly calculated in MIPS program (since the goal is to find the minimum point, not the MSE itself). Instead, it serves as the starting point for the entire theoretical derivation:

  - Substitute (1) into (2): The right-hand side of Formula (2) is obtained by substituting the definition of the output $y(n)$ (Formula 1) into the error term.

  - Differentiate: When the resulting quadratic function of the coefficients is differentiated and set to zero, the final algebraic result is the Wiener-Hopf Equation (Formula 4).

$$Differentiating\ (2)\ \rightarrow Result\ in\ R_M\boldsymbol{h} = \gamma_d$$

- Thus, Formula (2) is the cost function that mathematically justifies the subsequent requirement to solve Formula (4).
- Although, in theoretical continuous math, $\mathbb{E}$ is an abstract operator. In MIPS implemenation (a discrete, finite system), $\mathbb{E}$ must be calculated using a summation and a division.

$$E[\text{Error}^2] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \text{Error}^2(n)$$

  - The need for summation and division arises from the fundamental difference between Probability Theory (where E lives) and Digital Signal Processing (where MIPS code lives).
  - n probability and statistics, the Expectation of a function A is defined as the theoretical mean over an infinite set of possibilities or an infinite timeline:

$$E[A] = \int_{-\infty}^{\infty} A(x)p(x)dx \quad \text{(For continuous probability)}$$

$$E[A] = \lim_{N\to\infty} \frac{1}{N} \int_{-N/2}^{N/2} A(t)dt \quad \text{(Over infinite time)}$$

- Because we cannot measure infinite data, we use the **Law of Large Numbers** to approximate the theoretical expectation with a simple average calculated over the available data points N.

Wiener-Hopf Summation (Formula 4): is the compact and definitive representation of the system you must solve to find the optimal filter. It translates the statistical relationships into a standard linear algebra problem.

The equation is concisely stated as:

$$R_M \boldsymbol{h} = \gamma_d$$

When expanded to show the constituent vectors and matrix (based on M, the filter order), the relationship becomes clear:

$$\underbrace{\begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(1) & \gamma_{xx}(2) & \cdots & \gamma_{xx}(M-1) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(1) & \cdots & \gamma_{xx}(M-2) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \cdots & \gamma_{xx}(M-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{xx}(M-1) & \gamma_{xx}(M-2) & \gamma_{xx}(M-3) & \cdots & \gamma_{xx}(0) \end{bmatrix}}_{\text{Autocorrelation Matrix } R_M} \cdot \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{M-1} \end{bmatrix}}_{\text{Coefficient Vector } \mathbf{h}_M} = \underbrace{\begin{bmatrix} \gamma_{dx}(0) \\ \gamma_{dx}(1) \\ \gamma_{dx}(2) \\ \vdots \\ \gamma_{dx}(M-1) \end{bmatrix}}_{\text{Cross-correlation Vector } \gamma_d}$$

- **Matrix R_M**

  - Autocorrelation Matrix (of the input signal $x(n)$).

  - Size $M \times M$.

  - Consists of the autocorrelation value $\gamma_{xx}(k)$ calculated in Formula (a).

  - This is the matrix A in the Gaussian Elimination solver.

- **Vector $\boldsymbol{h}_M$**

  - Filter Coefficient Vector.

  - Size $M \times 1$.

  - The unkown values h_0 through h_{M-1} that defined the optimal filter.

  - This is the vector x in the linear system solver.

- **Vector γ_d**

  - Cross-correlation Vector.

  - Size $M \times 1$.

  - Consists of the cross-correlation values $\gamma_d(l)$ calculated in Formula (b).

- This vector links the desired behavior (d) to the input signal statistics (x), providing the necessary target for the system to aim for.

- This is the vector B in Gaussian Elimination solver.

Autocorrelation Function (Formula a)

$$\gamma_{xx}(k) = E[x(n)x(n-k)]$$

- $Autocorrelation$ (γ_{xx}): this function measure the correlation of the input signal with time-delayed version of itself.

- $Index\ k$: represents the time delay, or "lag," between the two copies of the signal being multiplied $(0 \le k \le M-1)$.

- Autocorrelation characterizes the power spectrum and structure of the input signal. It tells the filter how predictable the noise and signal components are based on their past values.

Cross-correlation function (Formula b)

The cross-correlation function γ_{dx} is the link between the desired outcome and the input data. It is essential because it provides the statistical intelligence needed to determine the filter coefficients.

$$\gamma_{dx}(k) = \mathbb{E}[d(n)x(n-k)]$$

- $k = 0$ $(\gamma_{dx}(0))$: measure the correlation between $d(n)$ and $x(n)$ at the same time instant.

- $k > 0$ $(\gamma_{dx}(k))$: measure how well past values of the input $x(n-k)$ correlate with the current desired value $d(n)$.

- Cross-correlation measures the degree of similarity between two different signals (desired outcome signal and inuput signal), as a function of time delay k.
- The filter needs to know how a shift in the noisy input $x(n-k)$ correlates with the ideal, clean output $d(n)$.
- The value $\gamma_{dx}(k)$ form the cross-correlation vector γ_d on the right side of Wiener-Hopf equation.

The optimal Solution (Formula 5)

$$\boldsymbol{h}_{opt} = R_M^{-1}\gamma_d$$

Formula (5) represents the explicit mathematical solution to the Wiener-Hopf equation (Formula 4). It directly expresses the optimal filter coefficients $\boldsymbol{h}_{opt}$ as the product of the inverse of the autocorrelation matrix R_M^{-1} and the cross-correlation vector γ_d. This formula tells us exactly how to calculate the filter coefficients if we know the statistical properties of the signals. This is the standard algebraic solution for the linear system $R_M\boldsymbol{h} = \gamma_d$ when R_M is invertible.

Although mathematically elegant, Formula 5 presents significant problems for real-world computing, especially in low-level environments like MIPS Assembly.

Issue	Description	Impact on MIPS
Computational Cost	Calculating the inverse of a matrix R_M^{-1} required $O(M^3)$ operation (e.g., using LU decomposition or Gauss-Jordan method). For large matrices, this is very slow.	MIPS FPU operations are sequential and slow.A 10×10 matrix is manageable, but larger matrices would take longer time execution.
Numerical Stability	Matrix inversion is highly sensitive to the condition number of the matrix. If R_M is ill-conditioned (determinant is close to zero, which often happens with noisy data), the calculated inverse matrix will contain massive floating-point errors.	The resulting filter coefficients would be completely wrong, leading to an incorrect output signal $y(n)$ and a high MMSE.
Porgram Complexity	Writing a robust matrix inversion routine (which handles pivoting and checks for singularity) in Assembly language is extremely difficult and long.	MIPS is a single-issue, single-cycle processor architecture in simulation (MARS). Each floating-point multiplication (mul.s) and division (div.s) instruction takes multiple clock cycles (latency).

In summary, avoiding Formula (5) and using the Gaussian Elimination method is the correct and necessary engineering choice. It ensures the mathematical stability required to produce an accurate MMSE value, thereby meeting the stringent performance requirements for the assignment.

MMSE Calculation (Formula 6)

Formula (6) provides the theoretical minimum value that the Mean-Square Error can achieve after the optimal filter coefficients $\boldsymbol{h}_{opt}$ have been applied.

$$\mathrm{MMSE_M} = \sigma_\mathrm{d}^2 - \sum_{\mathrm{k=0}}^{\mathrm{M-1}} h_{opt}(k)\gamma_{dx}^*(k)$$

- **Total signal power (the upper bound) σ_d^2**
  - This is the variant of the desired signal, often interpreted as the total power or energy of the desired signal $d(n)$.
  - $\sigma_d^2 = E[|D(n)|^2]$.
  - This term represents the maximum possible mean-square error if the filter did nothing ($\boldsymbol{h} = 0$). It is the "baseline" energy contained in the target signal.

- **Energy remove by the filter** $\sum_{k=0}^{M-1} h_{opt}(k)\gamma_{dx}^{*}(k)$
  - This is the inner dot product of the optimal coefficient vector h_{opt} and the cross-correlation vector γ_{dx}.
  - This term represent the maximum amount of signal energy that the optimal filter is able to capture and utilize from the noisy input $x(n)$. Essentially, this is the amoint of noise and uncertainly that the filter successfully eliminated.
  - Since the input data are real numbers, the complex conjugate γ_{dx}^{*} is equal to the real value γ_{dx}.
- **Final calculation of MMSE**

$$MMSE = Total\ Energy - Energy\ Removed$$

The final result is obtained by subtracting the "Energy Removed" (Term 2) from the "Total Power" (Term 1) and then dividing by N (the number of samples) to get the final mean value, thus honoring the $\mathbb{E}$ operator from the original definition.

$$\text{MMSE} = \frac{1}{N}\left(\sum D(n)^2 - \sum h_{opt}(k)\gamma_{dx}(k)\right)$$

This confirms that the MMSE calculation in MIPS is a rigorous implementation of the theoretical Formula 6.