Ho Chi Minh City University of Technology

Faculty of Computer Science and Engineering

Course: Computer Architecture Lab (CO2008)

Assignment

FILTERING AND PREDICTION SIGNAL WITH WIENER FILTER

Ho Chi Minh City, September 2025

Contents

1 Outcomes **2**

2 Introduction **2**

3 Requirements **5**

3.1 Input 5
3.2 Output 5
3.3 Pre-determined variables 5
3.4 Test cases 6
3.5 Report 6

4 Submission **6**

5 Plagiarism **6**

6 Rubric for evaluation **6**

1 Outcomes

After finishing this assignment, students can proficiently use:

- MARS MIPS simulator.
- Arithmetic & data transfer instructions.
- Conditional branch and unconditional jump instructions.
- Procedures.

2 Introduction

The Wiener filter is a classic adaptive filtering technique used in signal processing to estimate a desired signal from a noisy or jammed observation, minimizing the mean square error (MSE). In many practical applications we are given an input signal x(n), consisting of the sum of a desired signal s(n) and an undesired noise or interference w(n), and we are asked to design a filter that suppresses the undesired interference component. In such a case, the objective is to design a system that filters out the additive interference while preserving the characteristics of the desired signal s(n).

We treat the problem of signal estimation in the presence of an additive noise disturbance. The estimator is constrained to be a linear filter with impulse response h(n), designed so that its output approximates some specified desired signal d(n).

The input sequence to the filter is x(n) = s(n) + w(n), and its output sequence is y(n). The difference between the desired signal and the filter output is the error sequence e(n) = d(n) - y(n).

We distinguish three special cases:

- If d(n) = s(n), the linear estimation problem is referred to as filtering.
- If d(n) = s(n + D), the linear estimation problem is referred to as signal prediction. Note that this problem is different than the prediction considered earlier in this chapter, where d(n) = x(n + D).
- If d(n) = s(n - D), the linear estimation problem is referred to as signal smoothing.

  The criterion selected for optimizing the filter impulse response h(n) is the minimization of the mean-square error. This criterion has the advantages of simplicity and mathematical tractability.

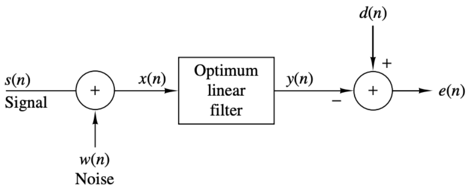

Figure 1: An example of Wiener Filter Linear Optimum Filtering

In Figure 1, The optimum linear filter, in the sense of minimum mean-square error (MMSE), is called a *Wiener filter*. The goal of the optimum filter is to provide an estimate of the desired response that is "as close as possible".

You must focus on the algorithms and use derivative to find the MMSE, which relates to *Principle of orthogonality*. Moreover, you can use Wiener-Hopf equation to determine optimized w.

Here are some Statistical Criteria for Optimization:

- Mean-squared value of the error
- Expectation of the absolute value of the error
- Expectation of third or higher order powers of the absolute value of the error.
- The first choice is most preferred as it leads to a mathematically tractable solution.
- Minimum Mean-Squared Error (MMSE) criteria
- Design the linear discrete time filter such that the mean-squared value of the estimation error is minimum.

Suppose a filter is constrained to have length M with coefficients $\{h_k, 0 \leq k \leq M-1\}$. The output $y(n)$ depends on the finite data record $x(n), x(n-1), \ldots, x(n-M+1)$, expressed as:

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

The mean-square value of the error between the desired output $d(n)$ and the actual output $y(n)$ is:

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

Since this is a quadratic function of the filter coefficients, minimizing E_M yields the set of linear equations:

$$\sum_{k=0}^{M-1} h_k \gamma_{xx}(l-k) = \gamma_{dx}(l), \quad l = 0, 1, \ldots, M-1 \tag{3}$$

where $\gamma_{xx}(k)$ is the autocorrelation of the input sequence $\{x(n)\}$, and $\gamma_{dx}(k) = \mathbb{E}[d(n)x^*(n-k)]$ is the cross-correlation between the desired sequence $\{d(n)\}$ and the input sequence $\{x(n), 0 \leq n \leq M-1\}$. These linear equations are known as the Wiener-Hopf equations, also referred to as the normal equations in the context of linear one-step prediction.

The equations in (7.3) can be expressed in matrix form as:

$$R_M \mathbf{h}_M = \boldsymbol{\gamma}_d \tag{4}$$

where R_M is an $M \times M$ Hermitian Toeplitz matrix with elements $R_{lk} = \gamma_{xx}(l-k)$, and $\boldsymbol{\gamma}_d$ is the $M \times 1$ crosscorrelation vector with elements $\gamma_{dx}(l)$, $l = 0, 1, \ldots, M-1$.

The solution for the optimum filter coefficients is:

$$\mathbf{h}_{\text{opt}} = R_M^{-1} \boldsymbol{\gamma}_d \tag{5}$$

The resulting minimum mean-square error (MMSE) achieved by the Wiener filter is:

$$\text{MMSE}_M = \min_{\mathbf{h}_M} E_M = \sigma_d^2 - \sum_{k=0}^{M-1} h_{\text{opt}}(k) \gamma_{dx}^*(k) \tag{6}$$

or, equivalently:

$$\text{MMSE}_M = \sigma_d^2 - \boldsymbol{\gamma}_d^{*t} R_M^{-1} \boldsymbol{\gamma}_d \tag{7}$$

where $\sigma_d^2 = \mathbb{E}[|d(n)|^2]$ is the variance of the desired signal.

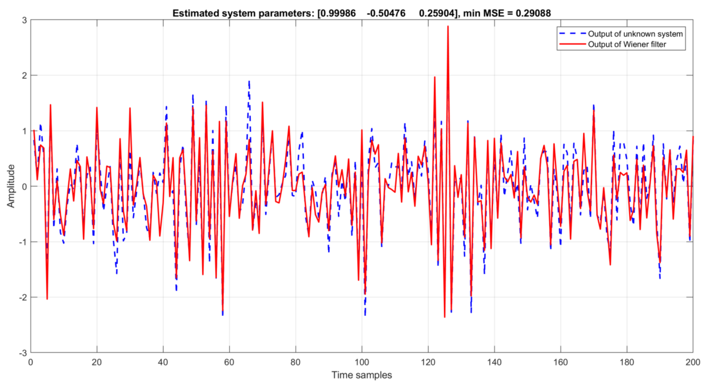

Figure 2: An example of Wiener filter apllication

In Figure 2, applying the Wiener filter for real-time signal.

3 Requirements

In this assignment, your task is to demonstrate the MMSE value and 2-D output signal after implementing the Wiener filter with MIPS assembly. The requirements are listed in the following sections.

3.1 Input

The program should be able to receive input from an external input file. It should be named **input.txt**, which in essence, is the combination of the original file named **desired.txt** and the noise. Each file will contain a sequence of **10** numbers. Thus, the contents of signal will turn into discrete domain.

The content are **floating-point numbers** rounded to 1 decimal point.

3.2 Output

You should print the result onto the terminal including 2 lines. The first line is the output sequence after filtering. The second line is the MMSE value calculated between the desired and output sequences. Please view Figure 3 for reference.

A text file named **output.txt** must be generated with the same content as printed into the terminal Please note that if the signal size is not as same size, the result must only be: "Error: size not match".

```
Filtered output: 0.1 -1.2 0.3 22.4 1.5 6.6 -0.6 1.2 5.5 1024.8
MMSE: 0.2
```

Figure 3: Result example

3.3 Pre-determined variables

Some variables must be defined as follows for grading:

- **desired_signal** (sequence): Where the desired signal is saved.
- **input_signal** (sequence): Where the input value is saved.
- **optimize_coefficient** (matrix): Here the Wiener filter coefficent is saved.
- **mmse** (number): where the MMSE value between desired signal and output is saved.

- **output_signal** (sequence): Where the output value is saved.

3.4 Test cases

A few example test cases will be provided to the LMS. However, you are advised to also develop a separated program in high-level programming languages you have learned so far to test the MIPS program yourself.

3.5 Report

A report of your work should be made for this assignment. Images of test runs as well as description of the functions and logic are anticipated. Flow charts and other visualization of your ideas are also encouraged. It is not advised to use long pure MIPS code in the report. Instead, pseudo-code and short important sections of the MIPS codes should be used. The report must contain basic information such as name, ID, class, and subject.

4 Submission

Students are requested to submit the MIPS program(s)/source code (.asm files) and the Assignment report to BK E-learning system (LMS) before the last lab session of your class. Assignment must be done in groups of 3-4 people.

Students who do not submit on time will get 0 for the assignment.

There will be NO deadline extensions.

5 Plagiarism

Similarity less than 50% in MIPS code is allowed. In other words, you will get 0 for assignment if your answers are similar to another student's more than 50%. Note that, we will use the MOSS tool developed by Stanford for checking similarity (https://theory.stanford.edu/~aiken/moss/).

6 Rubric for evaluation

Your work will be graded as follows:

- Demo (5 points): You are able to provide an output sequence and calculate the MMSE and show them onto the terminal and the **output.txt** file. Your grades will be subtracted if:

- Your MMSE is in the range [0.3, 0.5]: -2 points.
- Your MMSE is greater than 0.5 or fail to calculate the MMSE: -5 points.
- The result not in the correct format: -1 points.
- You are unable to provide the output sequence: -5 points.

- Report (5 points): You are able to provide a thorough report with detailed explanations and clear visualization. The necessary group and project details are present. Your grades will be subtracted if:
  - Your report contains few explanations and/or unclear results: -2 points.
  - Your report has bad structure. The sections and necessary information are either missing or not clear: -2 points.
  - Team workload is not reported or reported incorrectly: -1 points.

You will receive a 0 if you fail to submit on time the source code and report.