

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY



FACULTY OF COMPUTER SCIENCE AND ENGINEERING  
COURSE: COMPUTER ARCHITECTURE LAB (CO2008)

---

## Assignment

LỘC VÀ DỰ ĐOÁN TÍN HIỆU  
BẰNG BỘ LỘC WIENER

---

Ho Chi Minh City, September 2025



## Mục lục

<b>1</b>	<b>Kết quả đạt được</b>	<b>2</b>
<b>2</b>	<b>Giới thiệu</b>	<b>2</b>
<b>3</b>	<b>Yêu cầu</b>	<b>5</b>
3.1	Dữ liệu đầu vào . . . . .	5
3.2	Dữ liệu đầu ra . . . . .	5
3.3	Biến định trước . . . . .	6
3.4	Trường hợp kiểm thử . . . . .	6
3.5	Báo cáo . . . . .	6
<b>4</b>	<b>Nộp bài</b>	<b>7</b>
<b>5</b>	<b>Đạo văn</b>	<b>7</b>
<b>6</b>	<b>Tiêu chí đánh giá</b>	<b>7</b>

## 1 Kết quả đạt được

Sau khi hoàn thành bài tập này, sinh viên có thể sử dụng thành thạo:

- Trình mô phỏng MARS MIPS.
- Các lệnh số học và truyền dữ liệu.
- Các lệnh rẽ nhánh có điều kiện và nhảy không điều kiện.
- Thủ tục (procedures).

## 2 Giới thiệu

Bộ lọc Wiener là một kỹ thuật lọc thích nghi cổ điển được sử dụng trong xử lý tín hiệu để ước lượng tín hiệu mong muốn từ một quan sát bị nhiễu hoặc bị nhiễu loạn, nhằm tối thiểu hóa sai số bình phương trung bình (Mean Square Error - MSE). Trong nhiều ứng dụng thực tế, chúng ta được cung cấp một tín hiệu đầu vào  $\{x(n)\}$ , bao gồm tổng của tín hiệu mong muốn  $\{s(n)\}$  và nhiễu hoặc nhiễu loạn không mong muốn  $\{w(n)\}$ , và nhiệm vụ là thiết kế một bộ lọc để triệt tiêu thành phần nhiễu không mong muốn. Trong trường hợp này, mục tiêu là thiết kế một hệ thống lọc ra nhiễu cộng thêm trong khi vẫn giữ nguyên đặc trưng của tín hiệu mong muốn  $\{s(n)\}$ .

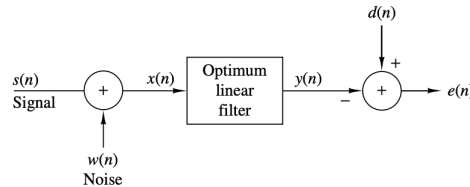
Chúng ta xét bài toán ước lượng tín hiệu trong điều kiện có nhiễu cộng thêm. Bộ ước lượng bị ràng buộc là một bộ lọc tuyến tính với đáp ứng xung  $\{h(n)\}$ , được thiết kế sao cho đầu ra của nó gần với tín hiệu mong muốn  $\{d(n)\}$  nhất.

Chuỗi tín hiệu đầu vào của bộ lọc là  $x(n) = s(n) + w(n)$ , và chuỗi đầu ra là  $y(n)$ . Hiệu giữa tín hiệu mong muốn và đầu ra của bộ lọc là chuỗi sai số  $e(n) = d(n) - y(n)$ .

Ta phân biệt ba trường hợp đặc biệt:

- Nếu  $d(n) = s(n)$ , bài toán ước lượng tuyến tính được gọi là lọc tín hiệu.
- Nếu  $d(n) = s(n + D)$ , bài toán ước lượng tuyến tính được gọi là dự đoán tín hiệu. Lưu ý rằng đây là một vấn đề khác với bài toán dự đoán đã đề cập trước đó trong chương này, nơi  $d(n) = x(n + D)$ .
- Nếu  $d(n) = s(n - D)$ , bài toán ước lượng tuyến tính được gọi là làm mượt tín hiệu.

Tiêu chí được lựa chọn để tối ưu đáp ứng xung của bộ lọc  $\{h(n)\}$  là tối thiểu hóa **sai số bình phương trung bình**. Tiêu chí này có ưu điểm là đơn giản và dễ xử lý về mặt toán học.



Hình 1: Ví dụ về Bộ lọc Tuyến tính Tối ưu Wiener

Trong Figure 1, bộ lọc tuyến tính tối ưu theo nghĩa tối thiểu hóa sai số bình phương trung bình (MMSE) được gọi là *bộ lọc Wiener*. Mục tiêu của bộ lọc tối ưu là cung cấp ước lượng của đầu ra mong muốn “gần đúng nhất có thể”.

Bạn cần tập trung vào các thuật toán và sử dụng đạo hàm để tìm MMSE, điều này liên quan đến *Nguyên lý trực giao*. Ngoài ra, bạn có thể sử dụng *phương trình Wiener-Hopf* để xác định vector trọng số tối ưu  $w$ .

Dưới đây là một số tiêu chí thống kê để tối ưu hóa:

- Giá trị trung bình bình phương của sai số
- Kỳ vọng của giá trị tuyệt đối của sai số
- Kỳ vọng của lũy thừa bậc ba hoặc cao hơn của giá trị tuyệt đối của sai số
- Tiêu chí đầu tiên là được ưu tiên nhất vì nó dẫn đến một lời giải dễ xử lý toán học
- Tiêu chí Sai số Bình phương Trung bình Tối thiểu (MMSE)
- Thiết kế bộ lọc rời rạc tuyến tính sao cho giá trị trung bình bình phương của sai số ước lượng là nhỏ nhất

Giả sử bộ lọc bị ràng buộc có độ dài  $M$  với các hệ số  $\{h_k, 0 \leq k \leq M-1\}$ . Đầu ra  $y(n)$  phụ thuộc vào dãy dữ liệu hữu hạn  $x(n), x(n-1), \dots, x(n-M+1)$ , được biểu diễn như sau:

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

Giá trị trung bình bình phương của sai số giữa đầu ra mong muốn  $d(n)$  và đầu ra thực tế  $y(n)$  là:

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

Vì đây là một hàm bậc hai của các **hệ số bộ lọc**, việc tối thiểu hóa  $E_M$  dẫn đến một **hệ phương trình tuyến tính**:

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

trong đó  $\gamma_{xx}(k)$  là tự tương quan của chuỗi đầu vào  $\{x(n)\}$ , và  $\gamma_{dx}(k) = \mathbb{E}[d(n)x^*(n-k)]$  là hàm tương quan chéo giữa chuỗi mong muốn  $\{d(n)\}$  và chuỗi đầu vào  $\{x(n), 0 \leq n \leq M-1\}$ . Các phương trình tuyến tính này được gọi là phương trình Wiener-Hopf, hay còn được gọi là phương trình chuẩn trong bối cảnh dự đoán tuyến tính một bước.

Phương trình (7.3) có thể được biểu diễn dưới dạng ma trận như sau:

$$R_M \mathbf{h}_M = \boldsymbol{\gamma}_d \quad (4)$$

trong đó  $R_M$  là ma trận Hermitian Toeplitz kích thước  $M \times M$  với các phần tử  $R_{lk} = \gamma_{xx}(l-k)$ , và  $\boldsymbol{\gamma}_d$  là vector tương quan chéo kích thước  $M \times 1$  với các phần tử  $\gamma_{dx}(l)$ ,  $l = 0, 1, \dots, M-1$ .

**Lời giải cho các hệ số bộ lọc tối ưu** là:

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

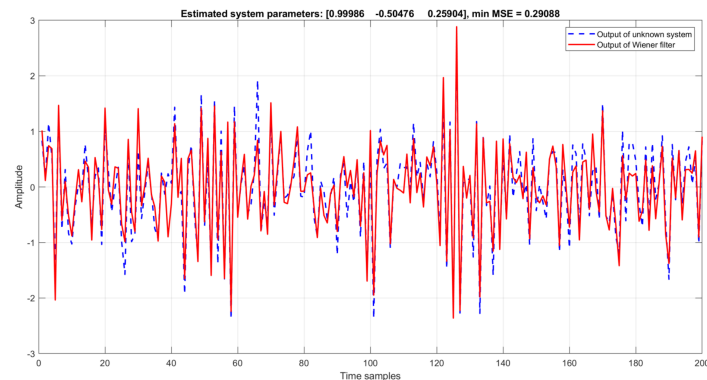
Sai số bình phương trung bình tối thiểu (MMSE) đạt được bởi bộ lọc Wiener là:

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

hoặc, tương đương:

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

trong đó  $\sigma_d^2 = \mathbb{E}[|d(n)|^2]$  là phương sai của tín hiệu mong muốn.



Hình 2: Ví dụ về ứng dụng của bộ lọc Wiener

Trong Figure 2, việc áp dụng bộ lọc Wiener cho tín hiệu thời gian thực được minh họa.

### 3 Yêu cầu

Trong bài tập này, nhiệm vụ của bạn là chứng minh giá trị MMSE và tín hiệu đầu ra 2 chiều sau khi thực hiện bộ lọc Wiener bằng cách sử dụng hợp ngữ MIPS. Các yêu cầu được liệt kê trong các mục sau.

#### 3.1 Dữ liệu đầu vào

Chương trình phải có khả năng nhận dữ liệu đầu vào từ một tập tin bên ngoài có tên là **input.txt**, đã được kết hợp từ hai tập tin ban đầu là **desired.txt** (tín hiệu mong muốn) và **noise.txt** (nhiều). *Khuyến nghị sử dụng nhiều trắng trong bài tập này.* Mỗi tập tin sẽ chứa 500 chuỗi số ngẫu nhiên. Do đó, nội dung tín hiệu sẽ được chuyển sang miền rời rạc.

Tất cả các giá trị đều là số thực dấu phẩy động, được làm tròn đến 1 chữ số thập phân. Tuy nhiên, tất cả các số sẽ được trình bày dưới dạng số thực với đúng 1 chữ số sau dấu phẩy.

#### 3.2 Dữ liệu đầu ra

Tập tin văn bản có tên là **output.txt** sẽ chứa 500 chuỗi số gần đúng với giá trị trong tập tin gốc. Ngoài ra, tính giá trị MMSE (là giá trị sai số giữa đầu ra và dữ liệu gốc). Đầu ra nên tham khảo theo Figure 3.



```
Reset: reset completed.  
Filtered output:  
0 0 -1 -1 -1 -1 0 0  
-- program is finished running --
```

Hình 3: Tham khảo đầu ra

### 3.3 Biến định trước

Một số biến bắt buộc phải được định nghĩa như sau để phục vụ cho việc chấm điểm:

- **desired signal** (Chuỗi): Nơi lưu tín hiệu mong muốn.
- **noise** (Chuỗi): Yêu cầu ít nhất hai loại nguồn nhiễu khác nhau cho các trường hợp kiểm thử.
- **input signal** (Chuỗi): Nơi lưu tín hiệu đầu vào.
- **optimize coefficient** (ma trận): Nơi lưu các hệ số bộ lọc Wiener.
- **MMSE** (số): Nơi lưu giá trị MMSE giữa tín hiệu mong muốn và đầu ra.
- **out** (chuỗi): Nơi lưu giá trị đầu ra.

### 3.4 Trường hợp kiểm thử

Tạo một tập hợp các trường hợp kiểm thử mẫu cùng với tập tin này. Khuyến nghị sử dụng 1 nguồn tín hiệu gốc và 2 loại nguồn nhiễu khác nhau cho các trường hợp kiểm thử. **Nhiều trắng là một trong các loại nhiễu bắt buộc, cần tìm thêm một loại nhiễu khác cho kiểm thử.**

Cần lưu ý rằng kết quả của các trường hợp kiểm thử sẽ không được cung cấp. Khuyến khích phát triển một chương trình riêng bằng các ngôn ngữ lập trình cấp cao mà bạn đã học để tự kiểm thử chương trình MIPS của mình.

Lưu ý rằng nếu kích thước của các tín hiệu không giống nhau, tập tin đầu ra phải chỉ chứa dòng: **"Error: size not match".**

### 3.5 Báo cáo

Một bản báo cáo về bài làm là yêu cầu bắt buộc trong bài tập này. Báo cáo cần bao gồm hình ảnh của các lần chạy kiểm thử, cũng như mô tả chức



năng và logic chương trình. Sơ đồ khối và các hình ảnh minh họa khác về ý tưởng của bạn cũng được khuyến khích. Báo cáo phải bao gồm các thông tin cơ bản như họ tên, mã số sinh viên, lớp và tên môn học.

## 4 Nộp bài

Sinh viên được yêu cầu nộp chương trình MIPS (các tập tin mã nguồn .asm) và báo cáo bài tập lên hệ thống BK E-learning (LMS) ngay sau buổi thực hành cuối cùng của lớp học. Bài tập phải được thực hiện cá nhân.

Sinh viên không nộp bài đúng hạn sẽ bị **0 điểm** cho bài tập này.

**Báo cáo không được chứa mã nguồn.** Thay vào đó, sinh viên cần trình bày thuật toán cũng như ý tưởng thực hiện của mình.

**Không có gia hạn thời hạn nộp bài dưới bất kỳ hình thức nào.**

## 5 Đạo văn

Mức độ giống nhau dưới 30% trong mã MIPS là được chấp nhận. Nói cách khác, bạn sẽ bị **0 điểm** cho bài tập nếu nội dung bài làm của bạn giống với bài của sinh viên khác trên **30%**.

Lưu ý rằng công cụ MOSS do Đại học Stanford phát triển để kiểm tra mức độ giống nhau giữa các bài làm (<https://theory.stanford.edu/~aiken/moss/>).

## 6 Tiêu chí đánh giá

Điểm của bạn sẽ dựa trên số lượng trường hợp kiểm thử (test case) mà bạn thực hiện đúng.

Bạn sẽ bị **0 điểm** nếu không nộp đúng hạn mã nguồn và báo cáo.