

	<b>VIETTEL AI RACE</b>	TD888
	<b>HƯỚNG DẪN MÔ PHỎNG HỆ THỐNG LỌC SỐ VÀ THIẾT KẾ BỘ LỌC SỐ</b>	Lần ban hành: 1

## 1. GIỚI THIỆU VỀ LỌC SỐ SỐ

Lọc số (Digital Filtering) là kỹ thuật xử lý tín hiệu rời rạc nhằm triệt tiêu hoặc giảm thiểu thành phần nhiễu, tăng cường thành phần mong muốn.

Trong lĩnh vực xử lý tín hiệu, thiết kế bộ lọc số chiếm vai trò then chốt với các ứng dụng từ âm thanh, hình ảnh đến thông tin vô tuyến và điều khiển tự động.

MATLAB cung cấp nhiều công cụ mạnh mẽ như Signal Processing Toolbox để xây dựng, mô phỏng và đánh giá hiệu năng của các bộ lọc FIR và IIR.

## 2. NỘI DUNG CHI TIẾT HƯỚNG DẪN MÔ PHỎNG

### 2.1 Yêu cầu trước khi thực hành

Một số yêu cầu bao gồm:

- Nắm bắt kiến thức cơ bản về tín hiệu rời rạc và hệ thống LTI.
- Hiểu khái niệm biến đổi  $z$  và biến đổi Fourier rời rạc.
- Sử dụng thành thạo MATLAB và Signal Processing Toolbox.
- Có kinh nghiệm đọc đồ thị đáp ứng tần số và nhóm pha.

### 2.2 Mục đích của phần thực hành

- Xây dựng và so sánh bộ lọc FIR và IIR cho bài toán loại bỏ nhiễu.
- Thiết kế cửa sổ Hamming, Kaiser và thuật toán Parks–McClellan.
- Đánh giá đáp ứng tần số, đáp ứng bước và độ ổn định của bộ lọc.
- Phân tích ảnh hưởng của tham số thiết kế lên đặc tính lọc.

### 2.3 Tóm tắt lý thuyết

Bộ lọc số được chia thành hai họ chính: FIR (Finite Impulse Response) và IIR (Infinite Impulse Response).

FIR có đáp ứng xung hữu hạn, dễ đạt pha tuyến tính; IIR có đáp ứng xung vô hạn, đạt đặc tính biên dải sắc với bậc thấp nhưng pha thường không tuyến tính.

Quy trình thiết kế tiêu chuẩn gồm xác định yêu cầu phổ, lựa chọn loại bộ lọc (thấp, cao, băng, dừng), xác lập tần số cắt và gợn rêu, sau đó chọn phương pháp thiết kế phù hợp và kiểm chứng ổn định, thực thi.

	VIETTEL AI RACE	TD888
	HƯỚNG DẪN MÔ PHỎNG HỆ THỐNG LỌC SỐ VÀ THIẾT KẾ BỘ LỌC SỐ	Lần ban hành: 1

## 2.4 Định nghĩa các hệ thống lọc cơ bản

- FIR (đáp ứng hữu hạn):

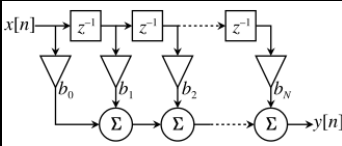
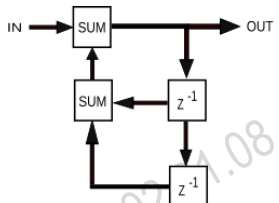
[CT1]	$y[n] = \sum_{k=0}^M b_k \cdot x[n-k]$
-------	--

- IIR (đáp ứng vô hạn):

[CT2]	$y[n] = \sum_{k=0}^M b_k \cdot x[n-k] - \sum_{k=1}^N a_k \cdot y[n-k]$
-------	--

- Bộ lọc thấp (Low-pass): Cho phép tần số  $\leq f_c$ .
- Bộ lọc cao (High-pass): Cho phép tần số  $\geq f_c$ .
- Bộ lọc băng (Band-pass): Cho phép  $f_1 \leq f \leq f_2$ .
- Bộ lọc dừng (Band-stop): Chặn  $f_1 \leq f \leq f_2$ .

## 2.5 Phân loại hệ thống lọc số

Loại lọc	Đặc điểm chính	Minh họa	Ứng dụng chính	Tính nhân quả	Tính ổn định
FIR	Pha tuyến tính; đáp ứng nhanh		Xử lý âm thanh, đo lường	Có	Luôn ổn định
IIR	Tiết kiệm bậc; pha không tuyến tính		Điều khiển, liên lạc	Có hoặc không	Tùy hệ số
Thấp (Low-pass)	Chặn thành phần tần số cao		Làm mượt tín hiệu, khử nhiễu	Có	Luôn ổn định

	<b>VIETTEL AI RACE</b>	TD888
	<b>HƯỚNG DẪN MÔ PHỎNG HỆ THỐNG LỌC SỐ VÀ THIẾT KẾ BỘ LỌC SỐ</b>	Lần ban hành: 1

Cao (High-pass)	Chặn thành phần tần số thấp		Phát hiện cạnh, phân tích	Có	Tùy thiết kế
Băng thông (Band-pass)	Cho phép dải tần xác định		Xử lý vô tuyến, phân tích	Có	Tùy thiết kế
Băng dừng (Band-stop)	Chặn dải tần xác định		Loại bỏ tạp âm cụ thể	Có	Tùy thiết kế

## 2.6 Phương pháp thiết kế bộ lọc số

Trong MATLAB, thường sử dụng ba nhóm phương pháp:

- Cửa sổ (Window Method): Hamming, Hanning, Kaiser.
- Equiripple (Parks–McClellan): Tối ưu theo Chebyshev.
- Thiết kế IIR cổ điển: Butterworth, Chebyshev I/II, Elliptic.

Phương pháp thiết kế	Băng thông cắt	Độ gợn râu trong (dB)	Độ gợn râu dừng (dB)	Độ dốc (dB/oct)	Độ phức tạp tính toán	Ghi chú
Window Hamming	Xác định thủ công	~0.02	~50	Trung bình	Thấp	Dễ cài đặt, pha gần tuyến tính
Window Kaiser	Điều chỉnh $\beta$	Tùy $\beta$	Tùy $\beta$	Cao	Trung bình	Linh hoạt, cân bằng gợn râu–độ dốc
Parks–McClellan	Tùy yêu cầu	Thiết lập rõ ràng	Thiết lập rõ ràng	Cao	Cao	Equiripple, tối ưu chặt chẽ
Butterworth (IIR)	Tự nhiên	-3	~20	Thấp	Thấp	Pha không tuyến tính
Chebyshev I (IIR)	Tự nhiên	0	~65			

## 2.7 Quy trình thiết kế chi tiết

	VIETTEL AI RACE	TD888
	HƯỚNG DẪN MÔ PHỎNG HỆ THỐNG LỌC SỐ VÀ THIẾT KẾ BỘ LỌC SỐ	Lần ban hành: 1

#### 2.7.1 Đặc tả yêu cầu phổ:

- **Loại:** Thấp, cao, băng thông, băng dừng.
- **Tần số:** Chỉ rõ  $\omega_p, \omega_s$  (hoặc  $f_p, f_s$ ).
- **Độ gợn râu:** Đặt  $A_p$  (dB) và  $A_s$  (dB).
- **Ràng buộc pha/độ trễ:** Xác định yêu cầu tuyến tính pha nếu có.

#### 2.7.2 Chọn họ bộ lọc:

- **FIR** nếu cần pha tuyến tính, ổn định vô điều kiện, chấp nhận bậc cao.
- **IIR** nếu cần biên sắc với bậc thấp, chấp nhận méo pha.

#### 2.7.3 Chọn phương pháp thiết kế:

- **FIR–Window:**
  - **Ưu tiên:** Triển khai nhanh, dễ điều chỉnh; dùng Hamming/Hann cho gợn râu thấp, Kaiser khi cần tinh chỉnh.
  - **Bước tính:** Ước lượng bậc từ suy hao dải dừng và vùng chuyển tiếp; tiền xử lý chuẩn hóa tần số; tạo bộ hệ số.
- **FIR–Equiripple:**
  - **Ưu tiên:** Tối ưu hóa cao, bậc tối thiểu; chỉ định trọng số từng dải để cân bằng lỗi.
- **IIR–Prototype:**
  - **Butterworth/Chebyshev/Elliptic:** Xác định bậc từ  $(A_p, A_s, \omega_p, \omega_s)$ ; tiền biến dạng tần số; dùng biến đổi song tuyến tính; phân rã thành biquad.

#### 2.7.4 Kiểm chứng miền tần số:

- **Biên độ:** Đáp ứng dải thông trong  $[-A_p, +A_p]$  dB; dải dừng dưới  $-A_s$  dB.
- **Pha/độ trễ nhóm:** Với FIR tuyến tính pha, xác minh  $\tau_g$  gần hằng trên dải thông.

#### 2.7.5 Đánh giá miền thời gian:

- **Đáp ứng bước/xung:** Kiểm tra vượt đỉnh, gợn sóng, thời gian xác lập.
- **Tín hiệu thực tế:** Áp dụng trên dữ liệu để đánh giá nhiễu tồn dư, méo biên dạng.

#### 2.7.6 Ràng buộc thực thi:

- **Độ dài/độ trễ:** FIR dài gây trễ; cân nhắc decimation hoặc cấu trúc phân pha (polyphase).

	<b>VIETTEL AI RACE</b>	TD888
	<b>HƯỚNG DẪN MÔ PHỎNG HỆ THỐNG LỌC SỐ VÀ THIẾT KẾ BỘ LỌC SỐ</b>	Lần ban hành: 1

- **Số học cố định:** Dùng biquad, chuẩn hóa thang, chống tràn; lượng tử hóa hệ số.

#### 2.7.7 Tinh chỉnh tham số:

- **FIR:** Điều chỉnh bậc, loại cửa sổ, trọng số equiripple.
- **IIR:** Điều chỉnh bậc, gọn râu, tần số cắt; chọn cấu trúc trực tiếp II–transposed/biquad để ổn định số học.

2025-10-19 02.31.08\_AI Race

2025-10-19 02.31.08\_AI Race

2025-10-19 0