BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CNTT-1

★ ★ ★

BÀI TẬP LỚN XỬ LÝ TÍN HIỆU SỐ

BÀI 2: THIẾT KẾ BỘ LỌC FIR THÔNG CAO

Giảng viên: Trần Tuấn Anh

Sinh viên thực hiện: Trần Đình Hào (B22DCCN278)

Hà Nội, 11/2023

MỤC LỤC

I/ Giới thiệu nội dung

Bộ lọc FIR (Finite Impulse Response), còn được gọi là bộ lọc đáp ứng hồi quy hữu hạn, là một trong những loại bộ lọc kỹ thuật số quan trọng trong lĩnh vực xử lý tín hiệu và hệ thống. Bộ lọc FIR được thiết kế để thực hiện việc loại bỏ hoặc tạo ra một phạm vi tần số cụ thể trong tín hiệu số.

Là một thành phần quan trọng của xử lý tín hiệu kỹ thuật số, bộ lọc FIR có những đặc điểm quan trọng sau:

1. Hữu hạn đáp ứng xung: Bộ lọc FIR có đáp ứng xung có độ dài hữu hạn. Điều này có nghĩa là sau khi đầu vào tín hiệu kết thúc, đầu ra của bộ lọc sẽ dừng ngay lập tức mà không có đáp ứng xung kéo dài vô hạn theo thời gian.

2. Dễ thiết kế: Thiết kế bộ lọc FIR thường dễ dàng hơn so với bộ lọc IIR (Infinite Impulse Response), vì nó không bao gồm đáp ứng xung vô hạn. Bộ lọc FIR thường được sử dụng trong các ứng dụng yêu cầu đáp ứng tần số tuyệt đối và phản hồi tốt.

3. Linh hoạt: Bộ lọc FIR có thể được thiết kế để đáp ứng một loạt yêu cầu tần số khác nhau, từ loại bỏ nhiễu đến cắt băng thông, tạo ra băng chuyền, và nhiều ứng dụng khác. Các bộ lọc FIR có thể thực hiện cả bộ lọc tuyến tính và phi tuyến tính.

4. Ứng dụng đa dạng: Bộ lọc FIR được sử dụng rộng rãi trong các lĩnh vực như xử lý tín hiệu âm thanh, xử lý ảnh số, truyền thông số, xử lý tín hiệu điện tử, và nhiều ứng dụng khác.

5. Công cụ thiết kế: Có nhiều phương pháp để thiết kế bộ lọc FIR, bao gồm sử dụng *Cửa sổ, Lấy mẫu tần số, và Bình phương tối thiểu.* Các công cụ tính toán và phần mềm như MATLAB, Python (với numpy và scipy), và các phần mềm chuyên dụng giúp thiết kế bộ lọc FIR một cách hiệu quả.

Tóm lại, bộ lọc FIR là một công cụ quan trọng trong xử lý tín hiệu số, cho phép chúng ta điều chỉnh và xử lý tín hiệu số một cách linh hoạt để đáp ứng các yêu cầu cụ thể trong nhiều ứng dụng khác nhau.

Trong nội dung của bài báo cáo ngày hôm nay, chúng em sẽ trình bày phương pháp Thiết kế ***bộ lọc FIR thông cao*** sử dụng các phương pháp: ***Cửa sổ (Window)***, ***Lấy mẫu tần số (Frequency Sampling)*** và ***Bình phương tối thiểu (Least Square Error)***.

II/ Cơ sở lý thuyết 1: Bộ lọc số lý tưởng (Ideal Filter)

Có bốn loại bộ lọc số lý tưởng chính, chia thành hai loại chính: bộ lọc thông (pass) và bộ lọc chắn (stop), và mỗi loại lại chia thành thông thấp (low-pass) và thông cao (high-pass). Dưới đây là mô tả cho mỗi loại:

1. Bộ lọc thông thấp (Low-Pass Filter): Bộ lọc thông thấp cho phép các tần số thấp đi qua và loại bỏ hoặc làm suy giảm đáng kể các tần số cao hơn. Điều này thường được sử dụng để loại bỏ nhiễu tần số cao hoặc để giữ lại các thành phần tần số thấp của tín hiệu.

2. Bộ lọc thông cao (High-Pass Filter): Bộ lọc thông cao cho phép các tần số cao đi qua và loại bỏ hoặc làm suy giảm đáng kể các tần số thấp hơn. Điều này thường được sử dụng để loại bỏ thành phần tần số thấp không mong muốn hoặc để tách tín hiệu tần số cao.

3. Bộ lọc thông dải (Band-Pass Filter): Bộ lọc thông dải cho phép các tần số nằm trong một khoảng tần số cụ thể được truyền qua và loại bỏ tần số ngoài khoảng đó. Bộ lọc này thường được sử dụng để tách lọc tín hiệu trong một khoảng tần số cụ thể hoặc để loại bỏ nhiễu tần số không mong muốn.

4. Bộ lọc chắn dải (Band-Stop Filter) hoặc Bộ lọc chắn tần số (Notch Filter): Bộ lọc chắn dải chặn hoặc làm suy giảm đáng kể các tần số nằm trong một khoảng tần số cụ thể và cho phép các tần số ngoài khoảng đó đi qua. Bộ lọc này thường được sử dụng để loại bỏ nhiễu tần số cụ thể hoặc để ngăn cản các tần số không mong muốn.

Mỗi loại bộ lọc có ứng dụng và mục tiêu thiết kế riêng, và chúng có thể được sử dụng để xử lý tín hiệu trong các tình huống khác nhau tùy thuộc vào yêu cầu cụ thể của ứng dụng. Ta phân loại bộ lọc lý tưởng dựa trên đáp ứng tần số.

Đối với Bộ lọc thông cao:_____

- Đáp ứng tần số: $$\left|H(e^{j\omega})\right| = \begin{cases} 1 & \begin{cases} -\pi \le \omega \le \omega_c \\ \omega_c \le \omega \le \pi \end{cases} \\ 0 & \omega \text{ còn lại} \end{cases}$$

- Đáp ứng xung:

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega n})d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j\omega n}d\omega - \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}d\omega = \delta(n) - \frac{\sin(\omega_c n)}{\pi n}$$

- Tần số cắt: f_c Dải thông: $f \in [f_c, \infty]$ Dải chặn: $f \in [0, f_c]$

- Tín hiệu số có phổ nằm trong dải tần $f > f_c$ đi qua .

- Không cho tín hiệu trong dải tần $f < f_c$ đi qua.

III/ Cơ sở lý thuyết 2: Bộ lọc số FIR (Finite Impulse Response)

1, Đáp ứng tần số của bộ lọc FIR pha tuyến tính:

Các bộ lọc số FIR có đặc tính xung h(n) hữu hạn thỏa mãn:

$$L[h(n)] = [0, N-1] = N$$

→ Ta có hàm hệ thống: $H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$

→ Đặc tính tần số của bộ lọc FIR: $H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n}$

Trong phạm vi của bài tập, ta chỉ xét các bộ lọc FIR có pha tuyến tính:

$$\theta(\omega) = \beta - \alpha\omega \quad \text{(trong đó } \alpha \text{ và } \beta \text{ là các hằng số)}$$

Với dạng xung h(n) đối xứng:

$$h(n) = h(M-n-1) \begin{cases} n = 0, 1, \dots, (M-1)/2 (odd) \\ n = 0, 1, \dots, \frac{M}{2} - 1 (even) \end{cases}$$

$$\alpha = (M-1)/2$$

Với dạng xung h(n) phản đối xứng:

$$h(n) = -h(M-n-1) \begin{cases} n = 0, 1, \dots, (M-1)/2 (odd) \\ n = 0, 1, \dots, \frac{M}{2} - 1 (even) \end{cases}$$

$$\alpha = \frac{M-1}{2} \text{ và } \beta = \frac{\pi}{2}$$

Trường hợp h(n)=h(M-1-n)

$$H(\omega) = H_r(\omega) e^{-\frac{j\omega(M-1)}{2}}$$

$$H_r(\omega) = \frac{h(M-1)}{2} + 2\sum_{n=0}^{\frac{M-3}{2}} h(n) cos\omega\left(\frac{M-1}{2} - n\right), \quad M\ odd$$

$$H_r(\omega) = 2\sum_{n=0}^{\frac{M-3}{2}} h(n) cos\omega\left(\frac{M-1}{2} - n\right), \quad M\ even$$

$$\theta(\omega) = \begin{cases} -\omega\left(\frac{M-1}{2}\right), & if H_r(\omega) > 0 \\ -\omega\left(\frac{M-1}{2}\right) + \pi, & if H_r(\omega) < 0 \end{cases}$$

Trường hợp h(n)= -h(M-1-n)

$$H(\omega) = H_r(\omega)e^{j\left[\frac{\omega(M-1)}{2}+\frac{\pi}{2}\right]} H(\omega) = H_r(\omega)We^{j\left[\frac{\omega(M-1)}{2}+\frac{\pi}{2}\right]}$$

$$H_r(\omega) = 2\sum_{n=0}^{\frac{M-3}{2}} h(n)sin\omega\left(\frac{M-1}{2}-n\right), \quad M\ odd$$

$$H_r(\omega) = 2\sum_{n=0}^{\frac{M}{2}-1} h(n)sin\omega\left(\frac{M-1}{2}-n\right), \quad M\ even$$

$$\theta(\omega) = \begin{cases} \frac{\pi}{2} - \omega\left(\frac{M-1}{2}\right), & if H_r(\omega) > 0 \\ \frac{3\pi}{2} - \omega\left(\frac{M-1}{2}\right) + \pi, & if H_r(\omega) < 0 \end{cases}$$

Như vậy, có bốn loại bộ lọc số FIR pha tuyến tính

- Loại 1: $\beta = 0$, N lẻ, đặc tính xung h(n) đối xứng

- Loại 2: $\beta = 0$, N chẵn, đặc tính xung h(n) đối xứng

- Loại 3: $\beta = \pm\frac{\pi}{2}$, N lẻ, đặc tính xung h(n) phản đối xứng.

- Loại 4: $\beta = \pm\frac{\pi}{2}$, N chẵn, đặc tính xung h(n) phản đối xứng

Type	LPF	HPF	BPF	SBF	Hilbert	Differentiator
FIR Type 1	✓	✓	✓	✓		
FIR Type 2	✓		✓			
FIR Type 3			✓		✓	✓
FIR Type 4		✓	✓		✓	✓

2, Thiết kế bộ lọc FIR tuyến tính:

Các khái niệm về tâm đối xứng, tâm phản đối xứng, chiều dài bộ lọc số FIR, N chẵn hay lẻ sẽ hình thành nên các đặc điểm của bộ lọc số. Căn cứ vào các đặc điểm của bộ lọc, chúng ta sẽ đi tổng hợp các bộ lọc số FIR. Trong nội dung của bài tập, chúng em sẽ trình bày 3 phương pháp:

➞ Phương pháp Cửa sổ (Windowing method)
➞ Phương pháp Lấy mẫu tần số (Frequency sampling method)
➞ Phương pháp Bình phương tối thiểu (Least Square Error)

IV/ Các phương pháp thiết kế bộ lọc số FIR

1, Phương pháp cửa sổ (Windowing method):

Nguyên lý: Phương pháp Cửa sổ (Windowing method) là một kỹ thuật quan trọng trong thiết kế bộ lọc FIR (Finite Impulse Response filter). Bộ lọc FIR được sử dụng để xử lý tín hiệu số bằng cách kết hợp các mẫu tín hiệu đầu vào với các hệ số cố định. Phương pháp cửa sổ giúp cải thiện hiệu suất của bộ lọc FIR bằng cách giảm độ nhạy của bộ lọc đối với các nhiễu tần số cao và thấp, tạo ra một hàm phản ứng tần số mong muốn.

Dưới đây là một số bước cơ bản trong việc thiết kế bộ lọc FIR bằng phương pháp cửa sổ:

1. Xác định yêu cầu của bộ lọc: Đầu tiên, bạn cần xác định yêu cầu cụ thể cho bộ lọc, chẳng hạn như dải tần số mong muốn, độ dốc tần số, độ trễ thời gian, v.v.

2. Chọn hàm cửa sổ: Bước này là quyết định quan trọng vì bạn sẽ chọn một hàm cửa sổ cụ thể để áp dụng cho các hệ số của bộ lọc. Các hàm cửa sổ phổ biến bao gồm cửa sổ Hamming, cửa sổ Hanning, cửa sổ Blackman, cửa sổ rectangular, v.v. Mỗi loại cửa sổ có các đặc điểm khác nhau và sẽ ảnh hưởng đến hiệu suất của bộ lọc. Độ gợn của dải (ripple) sẽ quyết định loại cửa sổ nên dùng:

Window's name	Mainlobe	Mainlobe/sidelobe	Peak $20\log_{10}\delta$
Rectangular	$4\pi/\ M$	-13dB	-21dB
Hanning	$8\pi/M$	-32dB	-44dB
Hamming	$8\pi/M$	-43dB	-53dB
Blackman	$12\pi/M$	-58dB	-74dB

3. Thiết kế bộ lọc ban đầu: Bắt đầu với một hàm cửa sổ chọn lựa, bạn tính toán bộ lọc FIR ban đầu bằng cách sử dụng phương trình chuyển đổi Fourier ngược (inverse Fourier transform) từ hàm phản ứng tần số mong muốn.

4. Điều chỉnh bộ lọc: Sau đó, bạn có thể điều chỉnh các tham số của bộ lọc, chẳng hạn như số mẫu hoặc chiều dài cửa sổ, để đáp ứng yêu cầu cụ thể của bạn.

5. Chuyển đổi hàm cửa sổ và hệ số bộ lọc: Cuối cùng, bạn tính toán các hệ số thực tế của bộ lọc bằng cách kết hợp hàm cửa sổ với bộ lọc ban đầu.

Các dạng cửa sổ thông dụng:

a, Cửa sổ Hình chữ nhật (Rectangular): Trong miền n, cửa sổ hình chữ nhật được định nghĩa:

$$w(n) = rect_N(n) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & n \text{ còn lại} \end{cases}$$

Nhận xét: $w(n) = rect_N(n)$

Trong miền tần số: $W_R(e^{j\omega}) = \left(\frac{\sin\left(\frac{\omega M}{\omega}\right)}{\sin\left(\frac{\omega}{2}\right)}\right) e^{-j\omega\frac{M-1}{2}}$

***b, Cửa sổ Bartlett (Triangular)*:** Trong miền n, cửa sổ Bartlett được định nghĩa:

$$w_T(n)_N = \begin{cases} \frac{2n}{N-1} & 0 \le n \le \frac{N-1}{2} \\ 2 - \frac{2n}{N-1} & \frac{N-1}{2} \le n \le N-1 \\ 0 & n \ne \end{cases}$$

➞ Thiết kế giống cửa sổ hình chữ nhật nhưng dạng hàm khác nhau.

c, Cửa sổ Hanning: Trong miền n, cửa sổ Hanning được định nghĩa:

$$\omega(n) = \begin{cases} 0.5 - 0.5\cos\frac{2\pi n}{M-1} & 0 \le n \le M-1 \\ 0 & n \text{ còn lại} \end{cases}$$

- Ta có tham số của bộ lọc Hanning:

$$\Delta\omega_{Han} = \frac{8\pi}{N}$$

$$\lambda_{Han}(n)_N \approx -32dB$$

d, Cửa sổ Hamming: Trong miền n, cửa sổ Hamming được định nghĩa:

$$\omega(n) = \begin{cases} 0.54 - 0.46\cos\frac{2\pi n}{M-1} & 0 \le n \le M-1 \\ 0 & n \text{ còn lại} \end{cases}$$

- Ta có tham số của bộ lọc Hamming:

$$\Delta\omega_{Han} = \frac{8\pi}{N}$$

$$\lambda_{Han}(n)_N \approx -43dB$$

e, Cửa sổ Blackman: Trong miền n, cửa sổ Blackman được định nghĩa:

$$w_B = \begin{cases} 0,42 - 0,5cos\dfrac{2\pi n}{N-1} + 0,08\,cos\dfrac{4\pi n}{N-1} & 0 \le n \le N-1 \\ 0 & n\ còn\ lại \end{cases}$$

Với điều kiện: $\sum_{m=0}^{\frac{N-1}{2}} a_m = 1$

f, Cửa sổ Kaiser: Trong miền n, cửa sổ Kaiser được định nghĩa như sau:

$$w_k(n)_N = \begin{cases} \dfrac{I_0(\beta\left(\frac{n-1}{2}\right)\cdot\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2})}{I_0(\beta\left(\frac{n-1}{2}\right))} & 0 \le n \le N-1 \\ 0 & n \ne \end{cases}$$

Để đạt hiệu quả cao trong thiết kế, người ta thường lấy $4 \le \beta \le 9$

2, Phương pháp lấy mẫu tần số (Frequency sampling method):

Nguyên lý: Phương pháp Lấy mẫu tần số (Frequency sampling method) là một trong những phương pháp thông thường để thiết kế bộ lọc FIR (Finite Impulse Response filter). Phương pháp này cho phép bạn thiết kế bộ lọc FIR bằng cách xác định trực tiếp các hệ số bộ lọc dựa trên phản ứng tần số mong muốn của bộ lọc. Dưới đây là các bước cơ bản trong phương pháp lấy mẫu tần số:

1. Xác định hàm phản ứng tần số mong muốn (desired frequency response): Trước hết, bạn cần xác định phản ứng tần số mà bạn muốn từ bộ lọc FIR. Điều này bao gồm việc xác định các giá trị của phản ứng tần số ở các tần số cụ thể trong dải tần số mong muốn.

2. Xác định số lượng mẫu tần số (frequency samples): Bạn cần quyết định số lượng mẫu tần số (sampling points) mà bạn sẽ sử dụng để đại diện cho phản ứng tần số mong muốn. Số lượng mẫu này thường phụ thuộc vào độ phân giải tần số mà bạn cần và đặc tính của bộ lọc.

3. Tính toán phản ứng tần số mẫu (sampled frequency response): Sử dụng các tần số mẫu đã chọn, bạn tính toán giá trị của phản ứng tần số mong muốn tại các tần số đó.

4. Sử dụng biến đổi Fourier ngược (inverse Fourier transform): Biến đổi Fourier ngược được sử dụng để chuyển đổi phản ứng tần số mẫu thành hệ số bộ lọc FIR. Bạn có thể sử dụng biến đổi Fourier ngược Discrete Fourier Transform (DFT) hoặc Fast Fourier Transform (FFT) để thực hiện việc này.

5. Trong tính toán hệ số bộ lọc: Hệ số bộ lọc FIR được tính toán dựa trên kết quả từ bước trước. Hệ số này sẽ xác định cách bộ lọc ảnh hưởng đến tín hiệu đầu vào tại các tần số cụ thể.

Phương pháp lấy mẫu tần số cho phép bạn thiết kế bộ lọc FIR dựa trên yêu cầu tần số cụ thể mà bạn muốn đạt được, và nó thường được ưa chuộng trong các ứng dụng yêu cầu kiểm soát nghiêm ngặt về tần số. Tuy nhiên, nhược điểm là có thể dẫn đến số lượng hệ số lớn nếu bạn cần đạt được độ phân giải tần số cao, điều này có thể gây ra chi phí tính toán và lưu trữ tăng lên.

3, Phương pháp Bình phương tối thiểu (Least Square Error):

Nguyên lý: Phương pháp bình phương tối thiểu (Least Squares Method) trong thiết kế bộ lọc FIR (Finite Impulse Response filter) tập trung vào việc tối thiểu hóa tổng bình phương sai số giữa phản ứng tần số thực tế của bộ lọc và phản ứng tần số mong muốn. Đây là một trong những phương pháp phổ biến để thiết kế bộ lọc FIR, đặc biệt trong các ứng dụng yêu cầu đáp ứng tần số chính xác.

Dưới đây là các bước cơ bản để thiết kế bộ lọc FIR bằng phương pháp bình phương tối thiểu:

1. Xác định đáp ứng tần số mong muốn (desired frequency response): Đầu tiên, bạn cần xác định đáp ứng tần số mong muốn của bộ lọc FIR. Điều này bao gồm xác định giá trị mong muốn của đáp ứng tần số tại các tần số cụ thể trong dải tần số.

2. Xác định đáp ứng tần số của bộ lọc (filter frequency response): Bạn cần xác định đáp ứng tần số của bộ lọc FIR dự kiến, dựa trên các hệ số của bộ lọc. Phản ứng tần số này thường được tính toán bằng cách sử dụng biến đổi Fourier hoặc các phương pháp khác.

3. Xác định hàm sai số (error function): Hàm sai số là sự khác biệt giữa phản ứng tần số mong muốn và phản ứng tần số của bộ lọc. Hàm sai số thường được tính toán bằng cách tính bình phương của sự khác biệt tại các tần số cụ thể.

4. Tối ưu hóa hệ số bộ lọc: Bằng cách tối ưu hóa hàm sai số, bạn cố gắng điều chỉnh các hệ số của bộ lọc để giảm thiểu sai số. Phương pháp tối ưu hóa này có thể sử dụng các phương pháp tối ưu hóa như phương pháp gradient descent, phương pháp RLS (Recursive Least Squares), hoặc phương pháp giải phương trình bình phương tối ưu.

5. Xác định hệ số của bộ lọc: Cuối cùng, sau khi đã tối ưu hóa hàm sai số, bạn sẽ có các hệ số cụ thể của bộ lọc FIR mà bạn có thể sử dụng để thực hiện xử lý tín hiệu.

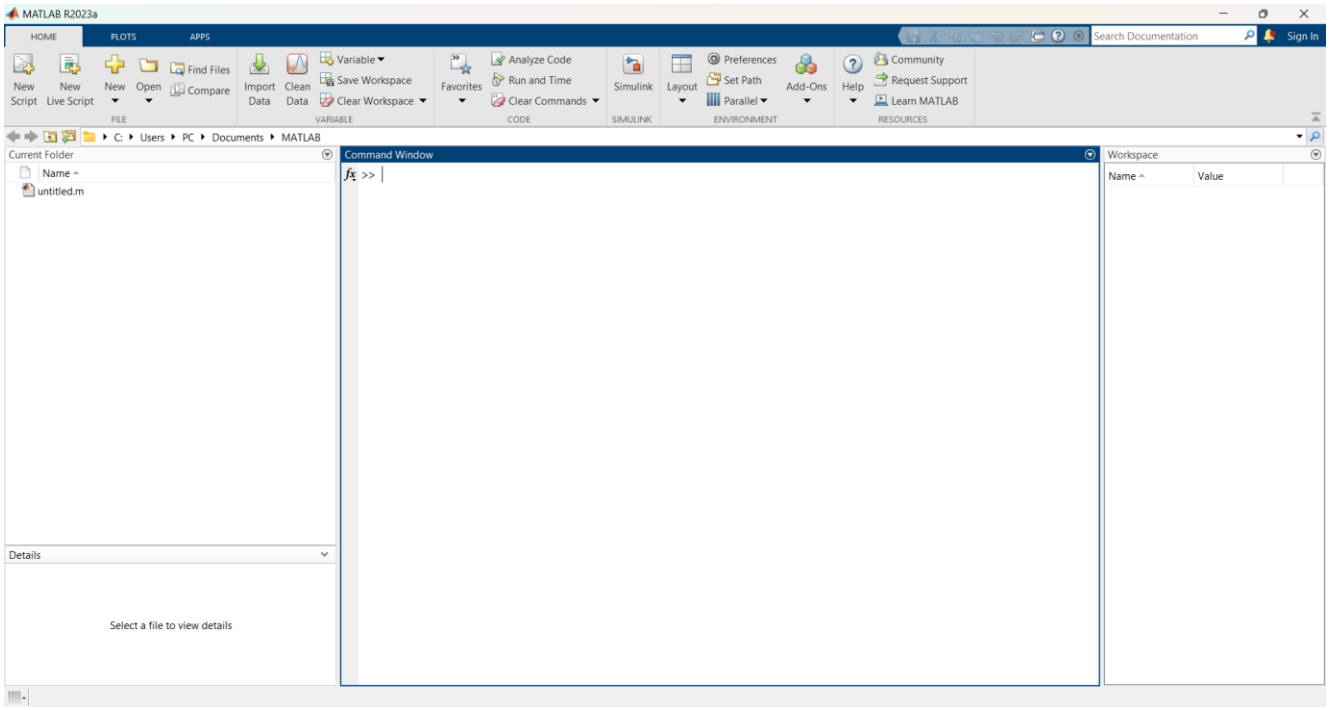
Phương pháp bình phương tối thiểu cho phép bạn điều chỉnh bộ lọc sao cho sai số giữa phản ứng tần số thực tế của bộ lọc và phản ứng tần số mong muốn là nhỏ nhất

dưới dạng bình phương. Điều này giúp đạt được ước tính tốt nhất của phản ứng tần số mong muốn trong bối cảnh của hàm bình phương của sai số.

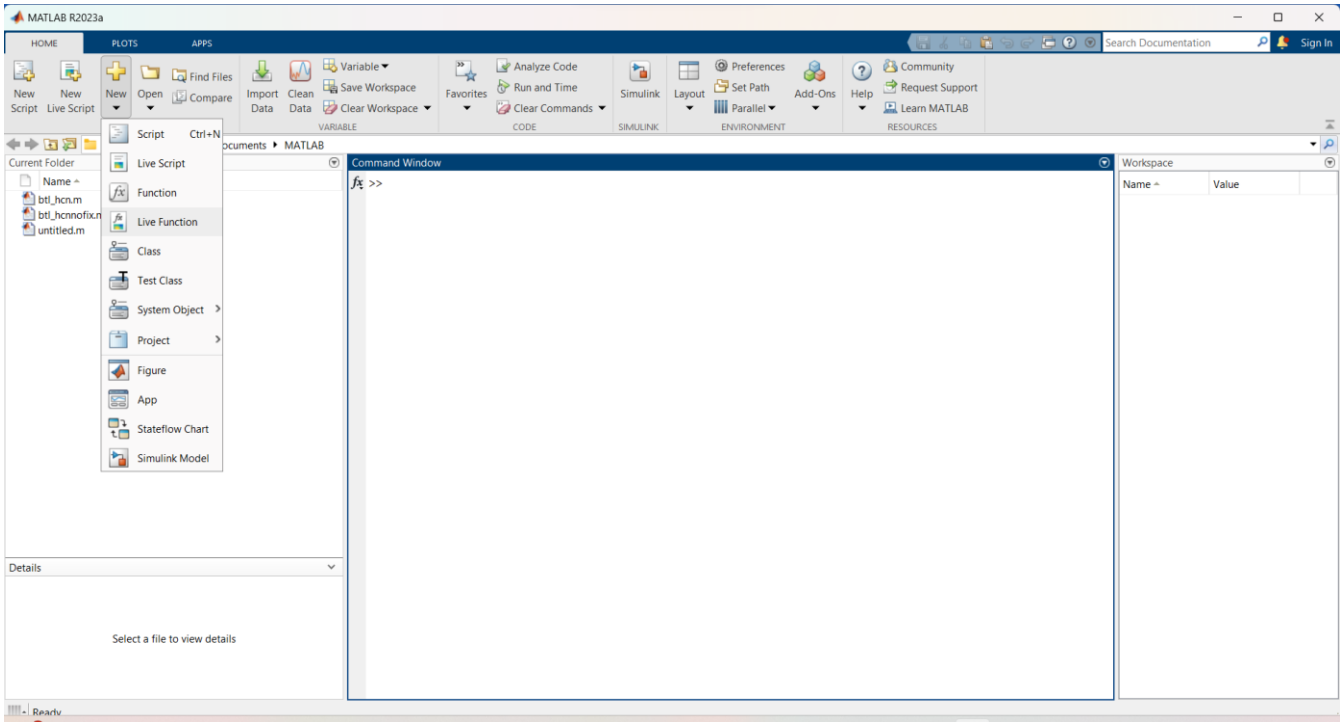
V/ Quy trình thiết kế (Ứng dụng trên MATLAB)

1, Thao tác thực hiện:

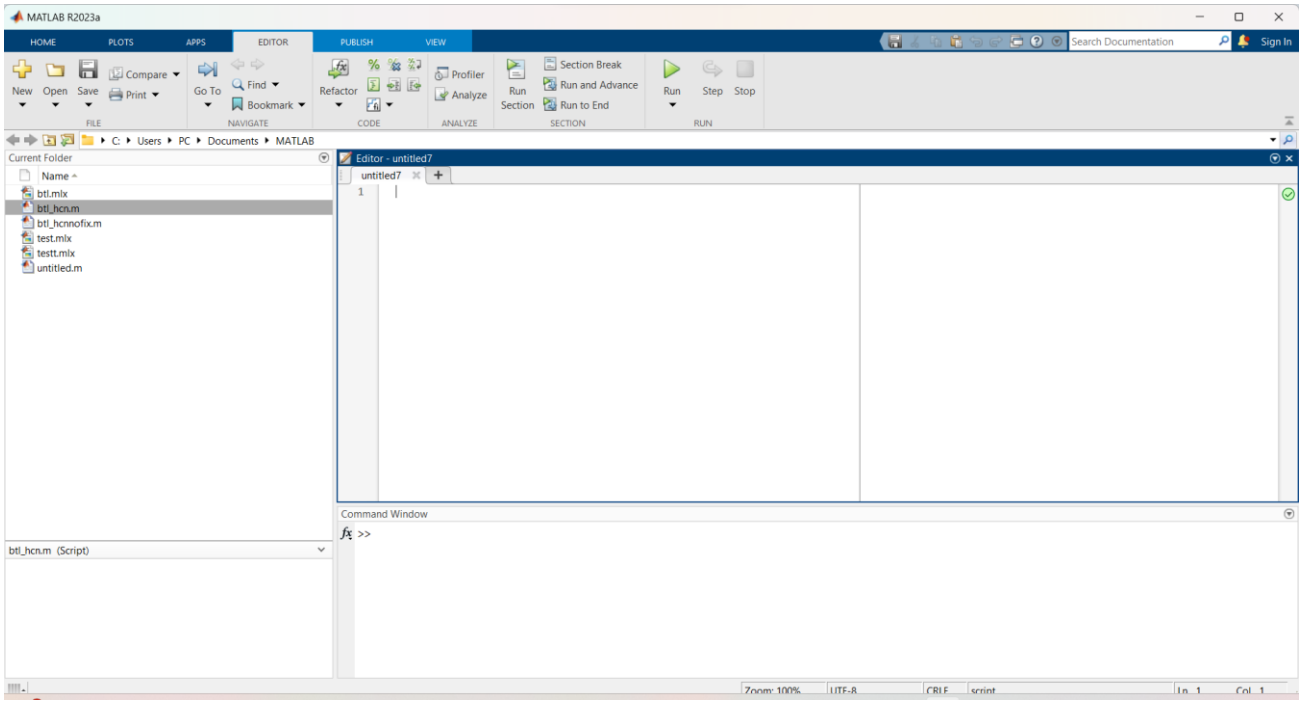
B1: Mở ứng dụng MATLAB:



B2: Trong mục New, chọn Script:



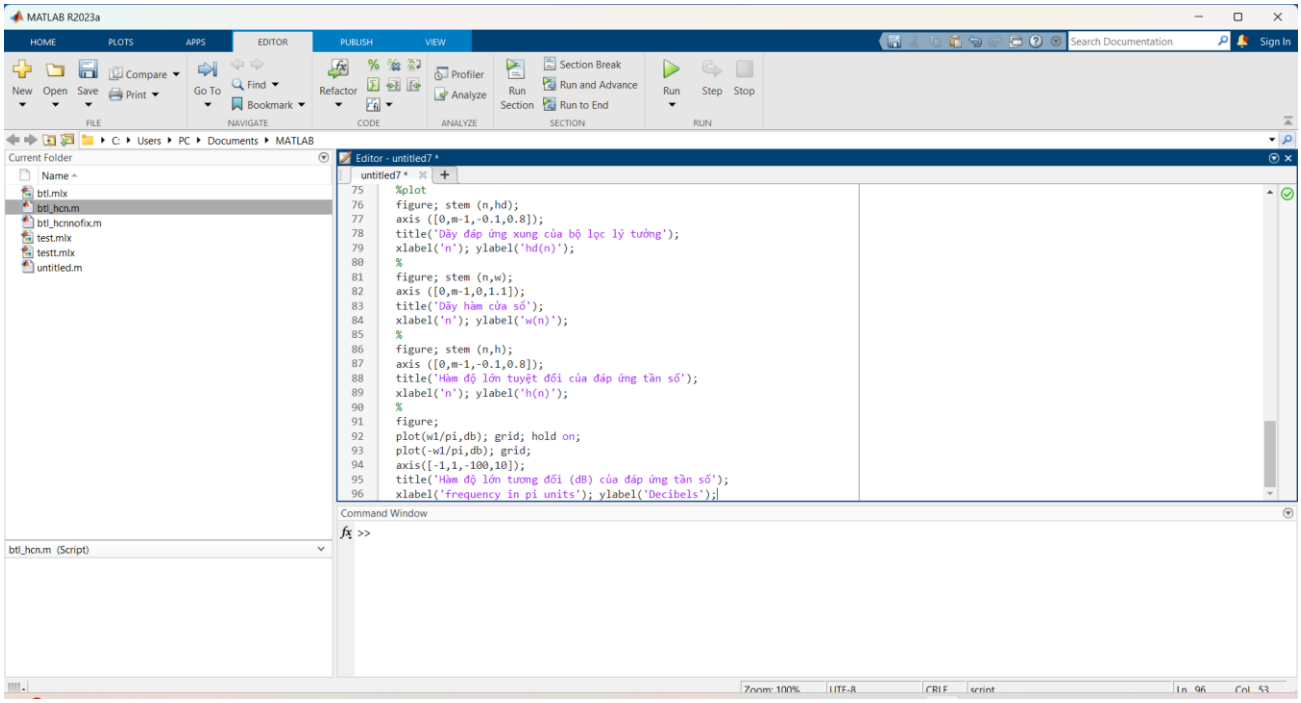
B3: Cửa sổ Live Editor hiện ra:



2, Thực hành:

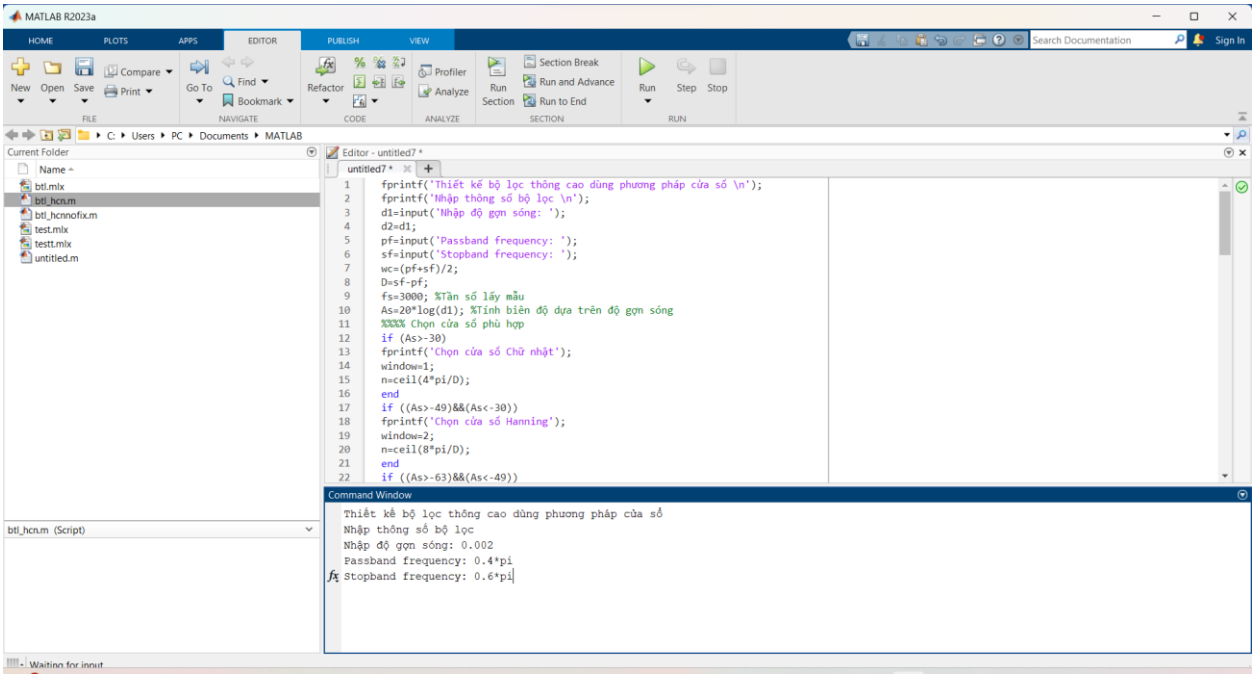
1, Thiết kế bộ lọc FIR thông cao sử dụng phương pháp Cửa sổ:

- Nhập phần code https://ideone.com/ZG52uD vào khung:

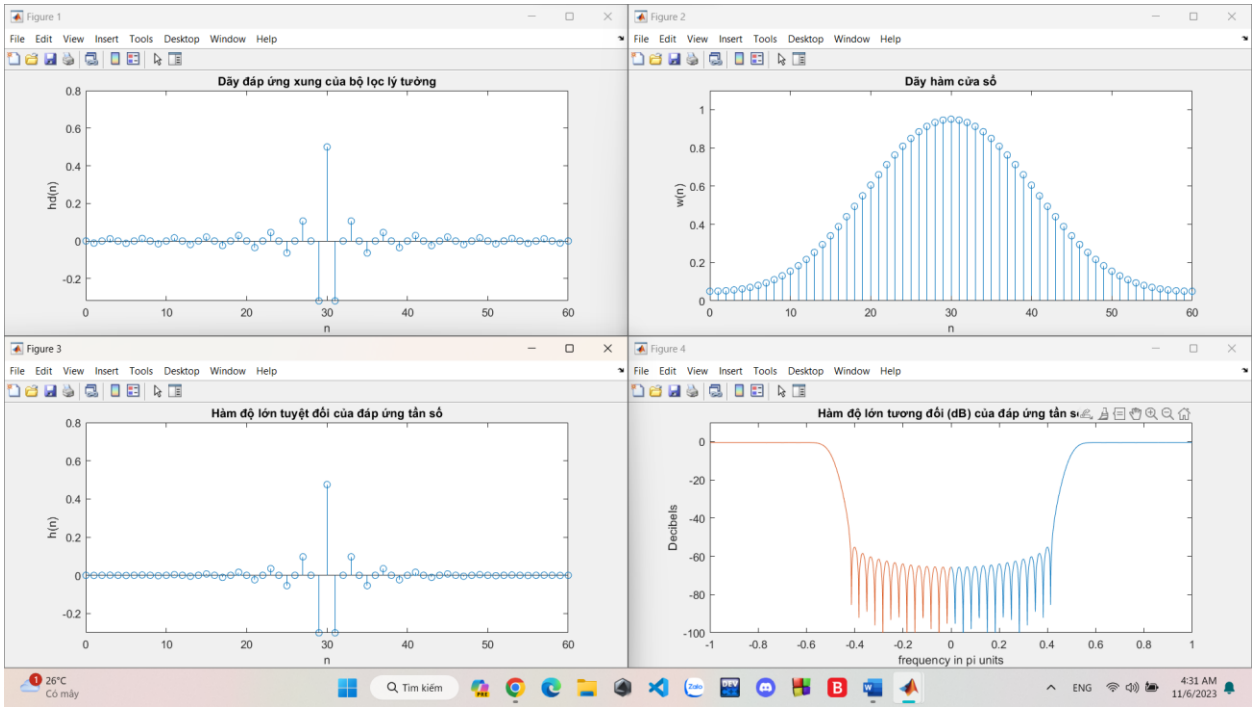


- Nhấn Run Section và nhập vào các giá trị của Độ gợn, Passband frequency, Stopband frequency và xem kết quả.

VD: Độ gợn = 0.002, Passband frequency = 0.4*pi, Stopband frequency = 0.6*pi.

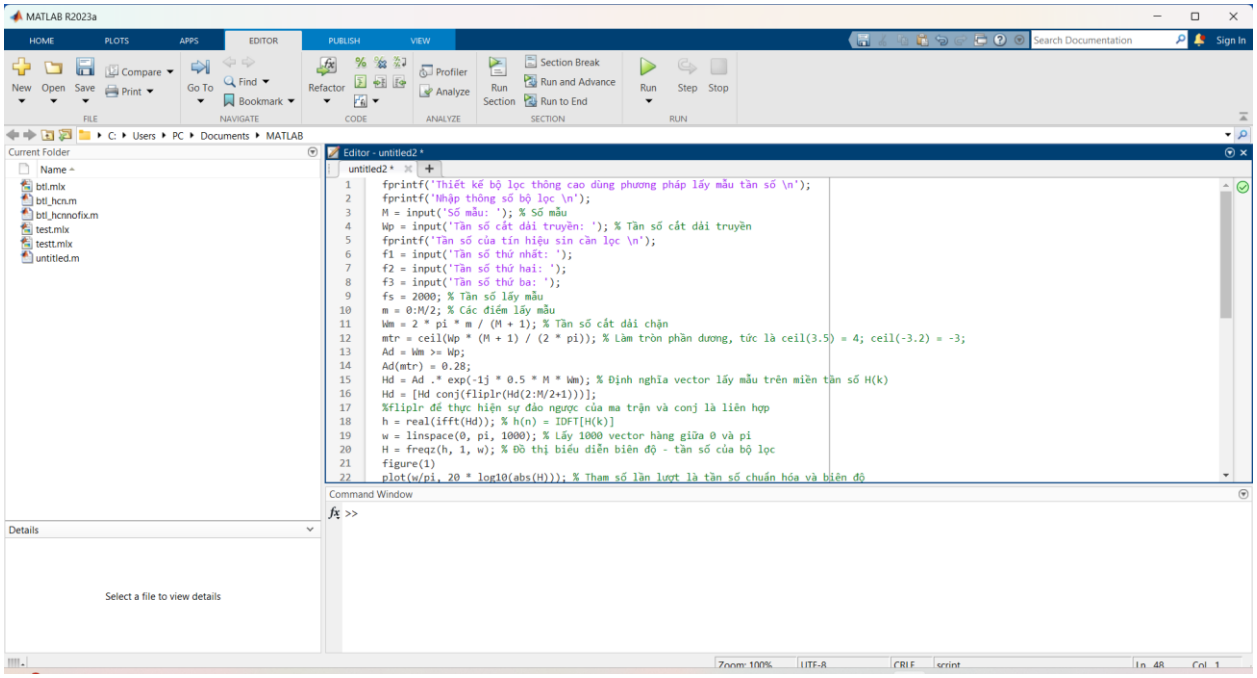


➞ Kết quả như hình:



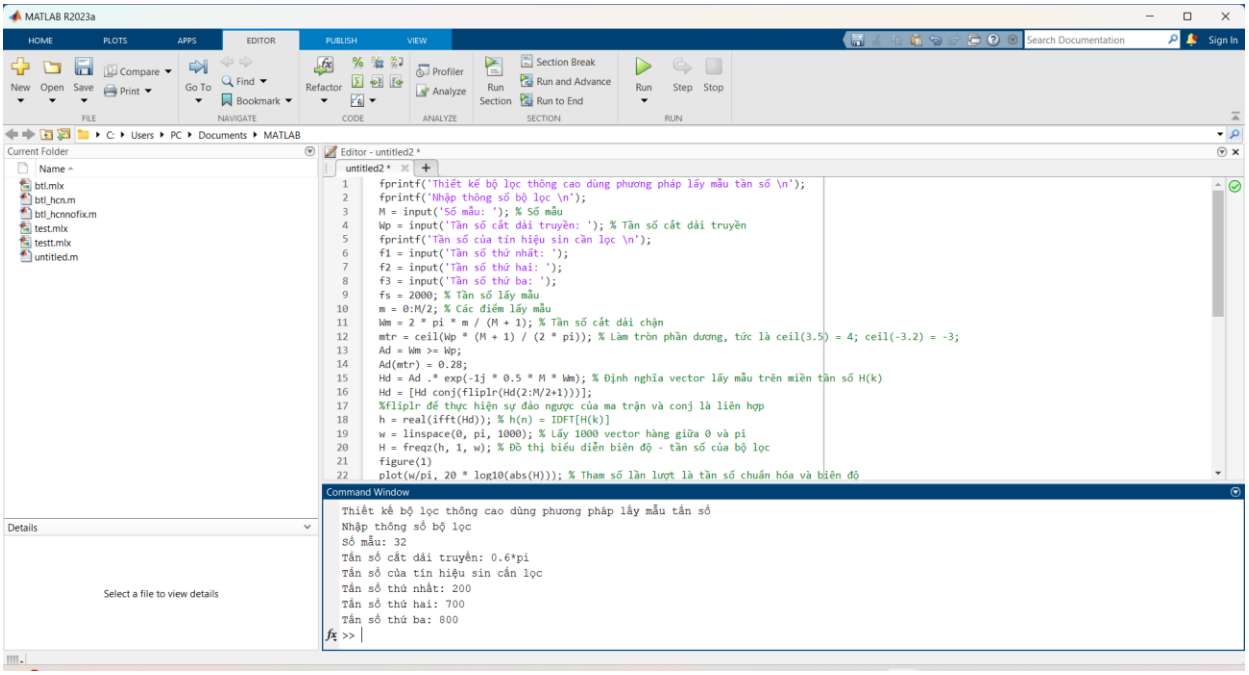
2, Thiết kế bộ lọc FIR thông cao sử dụng phương pháp Lấy mẫu tần số:

- Nhập phần code https://ideone.com/8yCyhH vào khung:

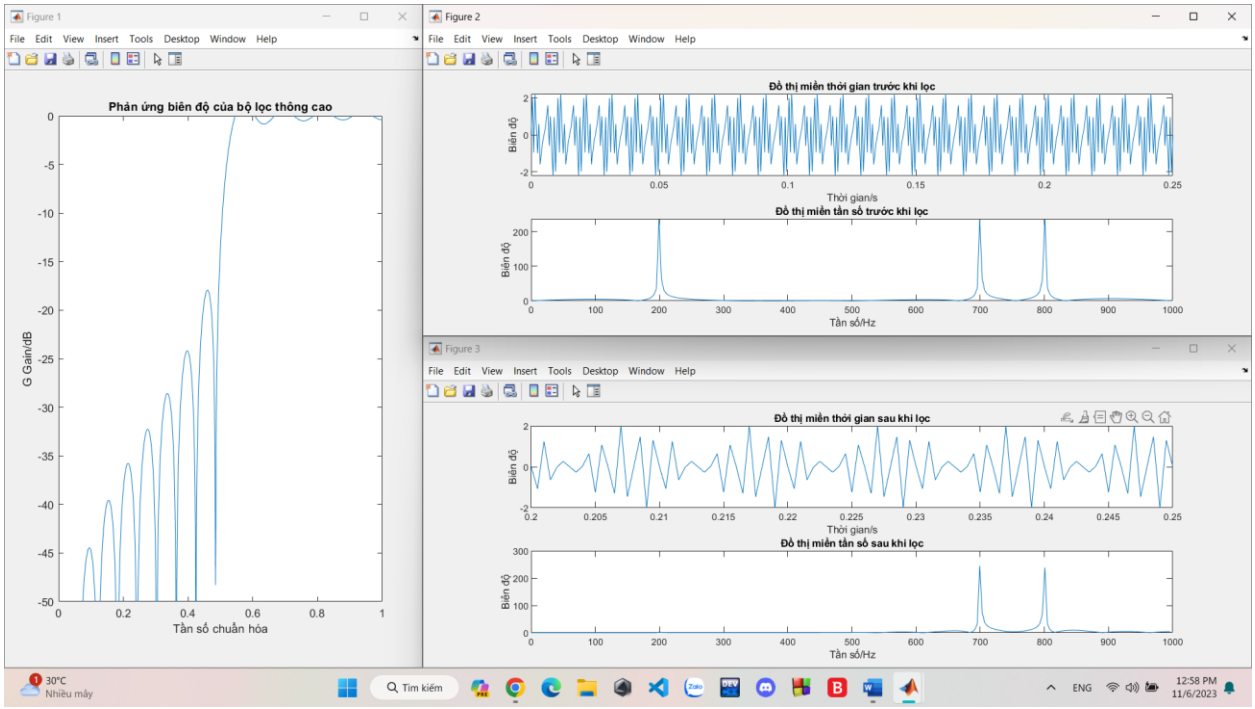


- Nhấn Run Section và nhập vào các giá trị của Số mẫu, Tần số cắt dải truyền, Tần số của tín hiệu sin cần lọc và xem kết quả.

VD: Số mẫu = 32, Tần số cắt dải truyền = 0.6*pi, Tần số của tín hiệu sin cần lọc lần lượt là 200, 700, 800

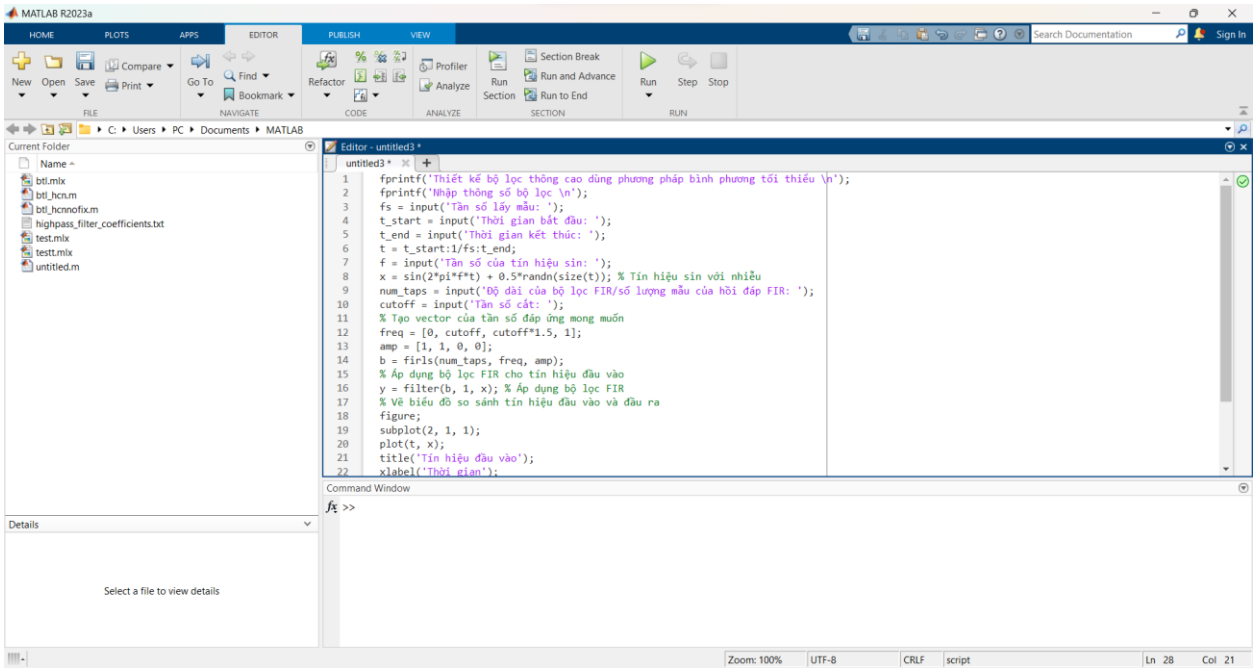


➞ Kết quả như hình:



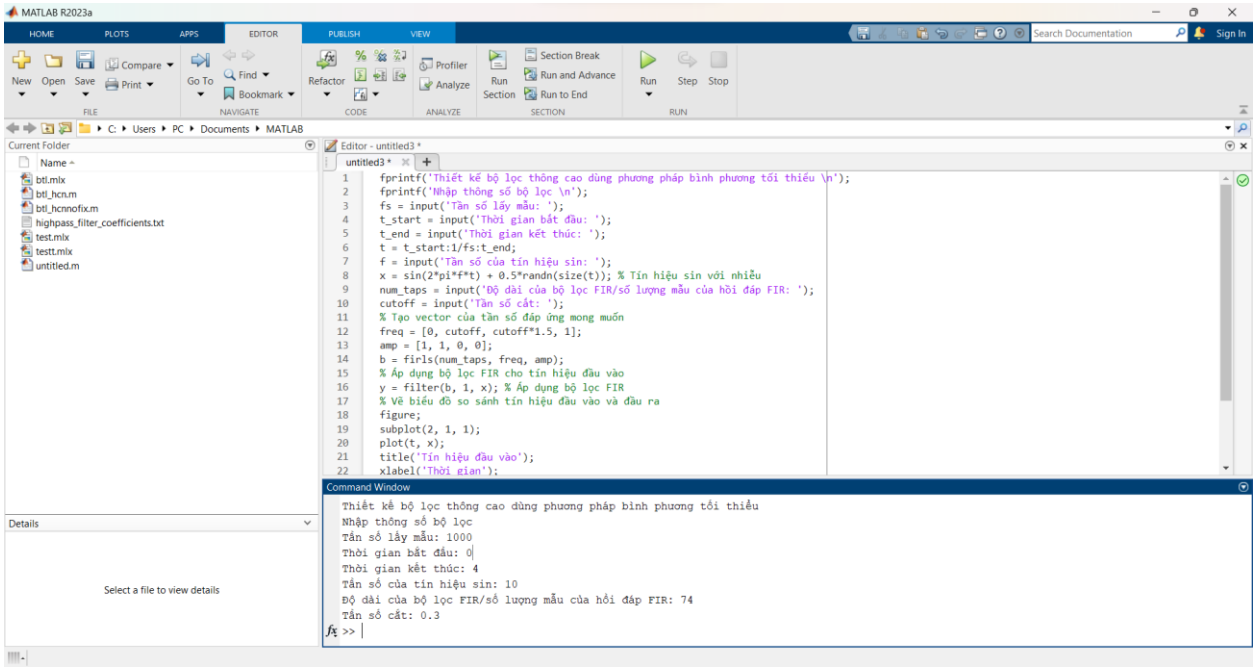
3, Thiết kế bộ lọc FIR thông cao sử dụng phương pháp Bình phương tối thiểu:

- Nhập phần code https://ideone.com/YiCmS6 vào khung:

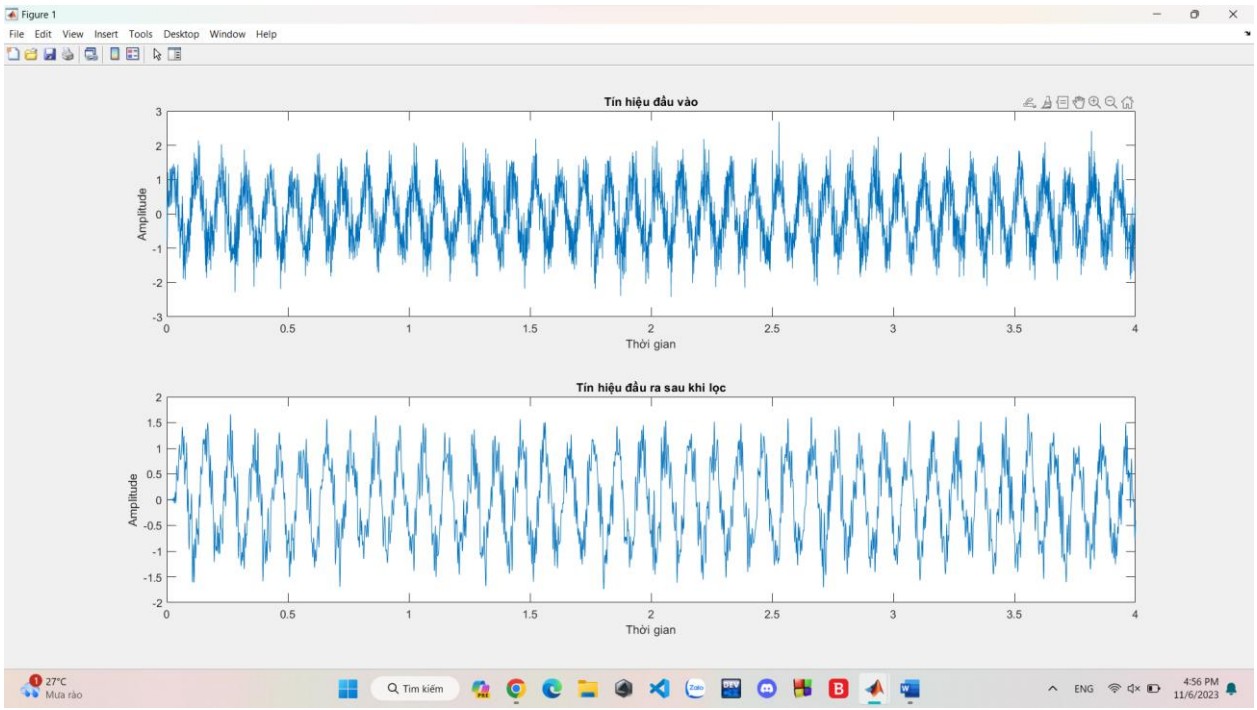


- Nhấn Run Section và nhập vào các giá trị của Tần số lấy mẫu, Thời gian bắt đầu, Thời gian kết thúc, Tần số của tín hiệu sin, Độ dài của bộ lọc FIR/số lượng mẫu của hồi đáp FIR, Tần số cắt và xem kết quả.

VD: Tần số lấy mẫu = 1000, Thời gian bắt đầu = 0, Thời gian kết thúc = 4, Tần số của tín hiệu sin = 10, Độ dài của bộ lọc FIR/số lượng mẫu của hồi đáp FIR = 74, Tần số cắt = 0.3



➞ Kết quả như hình:



So sánh tín hiệu đầu vào và tín hiệu đầu ra:

Đối với phương pháp Cửa sổ:

Dưới đây là sự so sánh giữa tín hiệu đầu vào và ra khi áp dụng phương pháp này:

- Tín hiệu đầu vào:
  - Có thể là tín hiệu nguồn từ môi trường thực tế, ví dụ như tín hiệu âm thanh, hình ảnh hoặc dữ liệu từ các cảm biến.
  - Thường chứa thông tin tần số ở các phạm vi khác nhau, bao gồm các thành phần tần số cao và thấp.
- Tín hiệu đầu ra:
  - Là kết quả của việc áp dụng bộ lọc FIR thông cao vào tín hiệu đầu vào.
  - Sau khi qua bộ lọc FIR thông cao sẽ bao gồm các thành phần tần số cao và loại bỏ các thành phần tần số thấp.

Đối với phương pháp Lấy mẫu tần số:

- Tín hiệu đầu vào:
  - Thường là tín hiệu vùng tần số thấp hoặc bất kỳ tín hiệu nào cần được lọc để loại bỏ các thành phần tần số thấp và chỉ giữ lại các thành phần tần số cao.
- Tín hiệu đầu ra:
  - Tín hiệu đầu ra của phương pháp thiết kế FIR thông cao sử dụng phương pháp lấy mẫu tần số là tín hiệu đã được lọc thông qua bộ lọc FIR. Tín hiệu này có các đặc điểm sau:
    + Nó chỉ chứa các thành phần tần số cao (được xác định bởi các tần số mẫu bạn chọn).
    + Nó đã loại bỏ các thành phần tần số thấp hoặc tần số nằm ngoài khoảng tần số lấy mẫu đã chọn.
    + Nó có phản hồi biên độ bằng phẳng trong khoảng tần số lấy mẫu, nghĩa là mức độ tương ứng của các tần số trong khoảng này được duy trì mà không bị biến đổi.

Đối với phương pháp Bình phương tối thiểu:

- Tín hiệu đầu vào:
  - Có thể là một tín hiệu đầu vào có dạng chuỗi số học hoặc một tín hiệu liên tục, tùy thuộc vào ứng dụng cụ thể.
  - Có thể chứa nhiễu (noise) hoặc thành phần tín hiệu mục tiêu mà bạn muốn lọc ra.
- Tín hiệu đầu ra:
  - Sau khi đi qua bộ lọc FIR thông cao sẽ không còn chứa các thành phần tần số thấp và sẽ tập trung vào các thành phần tần số cao hơn.

- Có thể bị mất đi một phần năng lượng của tín hiệu ban đầu, nhất là nếu bộ lọc được thiết kế có bậc cao. Điều này có thể dẫn đến hiện tượng mất thông tin (information loss) trong tín hiệu đầu ra.

→ Qua đây, ta cũng thấy được một cách tổng quát nhất về bộ lọc FIR thông cao. Dưới đây là 1 số ưu điểm và nhược điểm của bộ lọc FIR thông cao:

Nhận xét ưu điểm:

- ***Tính ổn định:*** Bộ lọc FIR thông cao thường dễ thiết kế và ổn định. Bộ lọc này không có phản hồi pha vòng (phase wrap) và không gây nguy cơ dao động hay bất ổn trong xử lý tín hiệu.
- ***Dễ hiểu và triển khai:*** Các hệ số của bộ lọc FIR có ý nghĩa và dễ hiểu, giúp người lập trình dễ dàng triển khai nó trên các hệ thống thời gian thực hoặc vi điều khiển.
- ***Khả năng đáp ứng chính xác tần số cắt:*** Bạn có khả năng kiểm soát chính xác tần số cắt (cutoff frequency) của bộ lọc FIR thông cao, cho phép tùy chỉnh để nổi bật hoặc loại bỏ các thành phần tần số cao theo yêu cầu.
- ***Khả năng loại bỏ nhiễu tần số thấp:*** Bộ lọc FIR thông cao rất hiệu quả trong việc loại bỏ nhiễu tần số thấp từ tín hiệu đầu vào, giúp cải thiện chất lượng tín hiệu và giảm nhiễu.

Nhận xét nhược điểm:

- ***Kéo dài thời gian chờ (latency):*** Bộ lọc FIR thông cao có thể tạo ra độ trễ (latency) trong xử lý tín hiệu. Điều này có thể không phù hợp cho các ứng dụng yêu cầu thời gian thực hoặc cần giảm thiểu độ trễ.
- ***Tốn tài nguyên tính toán:*** Thiết kế bộ lọc FIR thông cao có thể đòi hỏi nhiều tài nguyên tính toán hơn so với các loại bộ lọc khác như bộ lọc IIR (Infinite Impulse Response).
- ***Không phù hợp cho việc thay đổi tần số cắt trong thời gian thực:*** Bộ lọc FIR thông cao thường cố định về tần số cắt sau khi được thiết kế. Nếu bạn cần thay đổi tần số cắt trong thời gian thực, điều này có thể khó khăn và đòi hỏi việc thiết kế lại bộ lọc.

→ Tùy thuộc vào ứng dụng cụ thể và yêu cầu của bạn, bộ lọc FIR thông cao có thể là một lựa chọn tốt hoặc không phù hợp.