

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN & ĐIỆN TỬ



EE2015

BÀI TẬP LỚN

Đề Tài
Thiết kế bộ lọc thông thấp IIR
bằng phương pháp Butterworth , Chebyshev Type II

GVBM: NGUYỄN XUÂN XINH

NHÓM THỰC HIỆN: Nhóm 9, DT01
Nguyễn Hữu Lãm 2013594

Tp. Hồ Chí Minh, Tháng 4/2023

Mục lục

1	Cơ sở lý thuyết và bài toán thiết kế	2
1.1	Giới thiệu chung về bộ lọc IIR	2
1.1.1	Giới thiệu	2
1.1.2	Hai cách tiếp cận	3
1.2	Thiết kế bộ lọc IIR	3
1.3	Các đặc điểm sơ bộ	3
1.3.1	Tỉ lệ tuyến tính tương đối:	3
1.3.2	Các tính chất của $ H_a(j\Omega) ^2$	4
2	Phương pháp thiết kế bộ lọc IIR	5
2.1	Các đặc trưng của bộ lọc Analog điển hình	5
2.1.1	Bộ lọc thông thấp Butterworth	5
2.1.1.a	Các đặc trưng:	5
2.1.1.b	Các phương trình thiết kế:	6
2.1.2	Bộ lọc Chebyshev I:	7
2.1.2.a	Bộ lọc Chebyshev I:	8
2.1.2.b	Bộ lọc Chebyshev II:	9
2.1.3	Bộ lọc thông thấp Ellipic:	10
2.1.4	Các đáp ứng pha của bộ lọc điển hình:	10
2.2	Phép biến đổi bộ lọc tương tự thành bộ lọc số	11
2.2.1	Biến đổi bất biến xung:	11
2.2.2	Biến đổi song tuyến tính:	12
3	Chương trình thiết kế	13
3.1	Tính toán thiết kế	13
3.2	Thuật toán giải quyết	14
3.3	Thiết kế bộ lọc bằng python:	14
3.3.1	Thiết lập bộ lọc IIR bằng phương pháp Butterworth	14
3.3.2	Kết quả thực hiện việc lọc tín hiệu bằng bộ lọc IIR bằng phương pháp Butterworth	16
4	Đánh giá kết quả ,kết luận	19
4.1	Chỉ tiêu kỹ thuật	19
4.2	Chỉ tiêu kỹ thuật:	20
4.3	Kết luận:	20

Giới thiệu đề tài

Xử lý số tín hiệu (Digital Signal Processing-DSP) đã trở thành môn học cơ sở cho nhiều ngành khoa học, kỹ thuật như: Điện, Điện Tử, Tin Học, Viễn thông, Tự động hóa.... Xử lý số tín hiệu được ứng dụng rộng rãi trong nhiều lĩnh vực và thiết bị như: CD, DVD, VCD, camera, scanner, y khoa..., trong các hệ thống truyền hình số, thông tin địa lý, bản đồ số, viễn thông...v.v..

Phương pháp xử lý cơ bản nhất của DSP là lọc, và các hệ thống được đề cập nhiều nhất trong xử lý tín hiệu số là bộ lọc số (Digital Filter). Nếu xét về đáp ứng xung có thể chia bộ lọc số thành 2 loại chính là bộ lọc có đáp ứng xung hữu hạn FIR (Finite Impulse Response) còn gọi là lọc không đệ quy, và bộ lọc có đáp ứng xung vô hạn IIR (Infinte Impulse Response) còn gọi là lọc đệ quy. Xét về đáp ứng tần số biên độ có thể chia các bộ lọc, FIR hay IIR, thành 4 loại cơ bản: thông thấp, thông cao, thông dải và chặn dải. Các bộ lọc này có thể được thiết kế bằng những phương pháp khác nhau, mỗi phương pháp đều có những ưu điểm và khuyết điểm riêng. Trong khuôn khổ của bài tiểu luận môn học này, em xin phép trình bày nội dung đề tài: Thiết kế bộ lọc thông thấp FIR và IIR bằng phương pháp Butterworth, Chebyshev Type II

Nội dung tiểu luận được chia thành 4 phần:

- Phần I: Bộ lọc IIR và bài toán thiết kế
- Phần II: Phương pháp thiết kế bộ lọc IIR
- Phần III: Chương trình thiết kế
- Phần IV: Đánh giá kết quả - kết luận

1 Cơ sở lý thuyết và bài toán thiết kế

1.1 Giới thiệu chung về bộ lọc IIR

1.1.1 Giới thiệu

Bộ lọc IIR có đáp ứng xung vô hạn, vì vậy chúng có thể khớp với các bộ lọc analog, mà nói chung đều có đáp ứng xung dài vô hạn.

Kỹ thuật cơ bản để thiết kế lọc IIR là biến đổi các bộ lọc analog điển hình (well-known) thành các bộ lọc digital sử dụng các ánh xạ giá trị-phức.

Sự thuận tiện của kỹ thuật này là chỗ có sẵn các bảng thiết kế lọc analog (AFD) và các ánh xạ được mở rộng trong thư viện.

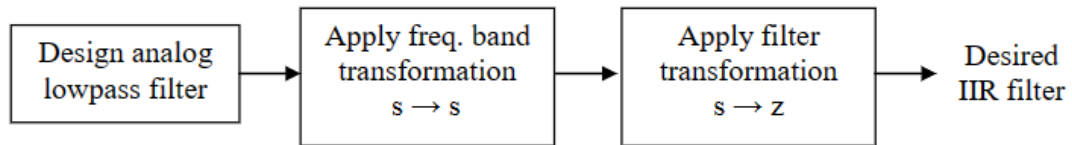
Các kỹ thuật cơ bản được gọi là các phép biến đổi lọc A/D.

Tuy nhiên, các bảng AFD chỉ dùng cho các bộ lọc thông thấp. Trong khi ta cần thiết kế các bộ lọc chọn tần khác (thông cao, thông dải, chặn dải, v.v...)

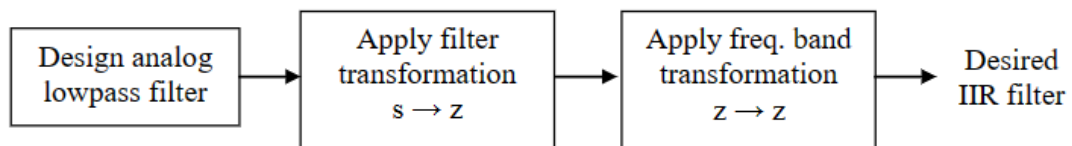
Cần áp dụng các phép biến đổi băng tần đối với các bộ lọc thông thấp. Các phép biến đổi này cũng được gọi là ánh xạ giá trị-phức, và chúng cũng có sẵn trong thư viện.

1.1.2 Hai cách tiếp cận

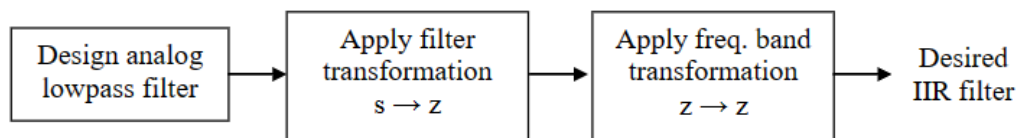
2.1. Cách 1, được sử dụng trong Matlab



2.2. Cách 2, được sử dụng để học tập, nghiên cứu



1.2 Thiết kế bộ lọc IIR



* Thiết kế bộ lọc thông thấp analog.

*Nghiên cứu và áp dụng các phép biến đổi bộ lọc để thu được bộ lọc số thông thấp.

*Nghiên cứu và áp dụng các phép biến đổi băng tần để thu được các bộ lọc số khác từ bộ lọc số thông thấp.

Các vấn đề tồn tại:

*Không điều khiển được đặc tính pha của bộ lọc IIR.

*Các thiết kế lọc IIR chỉ xử lý như các thiết kế về biên độ.

1.3 Các đặc điểm sơ bộ

1.3.1 Tỷ lệ tuyến tính tương đối:

➤ $H_a(j\Omega)$ là đáp ứng tần số của bộ lọc tương tự.

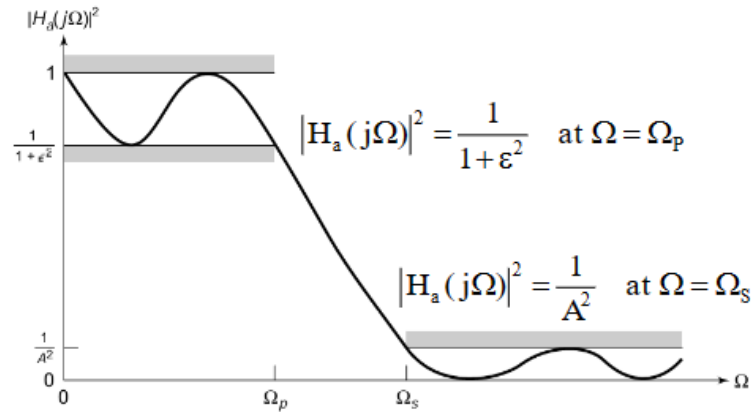
➤ Các đặc trưng bộ lọc thông thấp trên đáp ứng bình phương biên độ được cho bởi:

$$\frac{1}{1+\epsilon^2} \leq |H_a(j\Omega)|^2 \leq 1, \quad |\Omega| \leq \Omega_p$$

$$0 \leq |H_a(j\Omega)|^2 \leq \frac{1}{A^2}, \quad \Omega_s \leq |\Omega|$$

Trong đó:

- ε là **thông số gợn sóng** dài thông.
- Ω_p là **tần số cắt dài thông** (rad/s).
- A là **tham số suy hao dài chặn**.
- Ω_s là **tần số cắt của dài chặn** (rad/s).



Các thông số kỹ thuật của bộ lọc thông dải Analog

➤ **Các hệ thức giữa $\varepsilon, A, R_p, A_s, \delta_1$ và δ_2**

$$R_p = -10 \log_{10} \frac{1}{1+\varepsilon^2} \Rightarrow \varepsilon = \sqrt{10^{R_p/10} - 1}$$

$$A_s = -10 \log_{10} \frac{1}{A^2} \Rightarrow A = 10^{A_s/20}$$

$$\frac{1-\delta_1}{1+\delta_1} = \sqrt{\frac{1}{1+\varepsilon^2}} \Rightarrow \varepsilon = \frac{2\sqrt{\delta_1}}{1-\delta_1}$$

$$\frac{\delta_2}{1+\delta_1} = \frac{1}{A} \Rightarrow A = \frac{1+\delta_1}{\delta_2}$$

1.3.2 Các tính chất của $|H_a(j\Omega)|^2$

Các đặc trưng của bộ lọc Analog được cho theo các hệ số của đáp ứng bình phương độ lớn, không bao hàm thông tin về pha. Do đó để đánh giá hàm truyền hệ thống $H_a(s)$ trong miền-s ta xét:

$$H_a(j\Omega) = H_a(s)|_{s=j\Omega}$$

Sau đó ta có: $|H_a(j\Omega)|^2 = H_a(j\Omega)H_a^*(j\Omega) = H_a(j\Omega)H_a(-j\Omega) = H_a(s)H_a(-s)|_{s=j\Omega}$

Hay: $H_a(s)H_a(-s) = |H_a(j\Omega)|^2|_{\Omega=s/j}$

Vì vậy các điểm cực và điểm không của hàm bình phương biên độ được phân bố theo đối xứng ảnh-gương xét theo trục $j\Omega$.

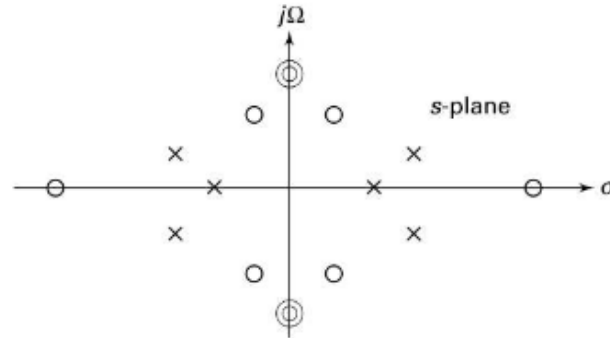
Đối với các bộ lọc thực, các điểm cực và điểm không xuất hiện theo cặp liên hợp phức (hoặc đối

xứng ảnh-gương theo trục thực).

Từ các mẫu này chúng ta có thể xây dựng $H_a(s)$, là hàm truyền hệ thống của bộ lọc analog.

Ta muốn $H_a(s)$ để biểu diễn một bộ lọc nhân quả và ổn định. Khi đó tất cả các điểm cực của $H_a(s)$ trong nửa mặt phẳng bên trái. Như vậy ta gán tất cả các điểm cực nửa-trái của $H_a(s)H_a(-s)$ lên $H_a(s)$. Hoặc chúng ta sẽ chọn các điểm không của $H_a(s)H_a(-s)$ nằm bên cạnh hoặc trên trục $j\Omega$ như các điểm không của $H_a(s)$.

Bộ lọc kết quả được gọi là một bộ lọc pha-tối thiểu.



2 Phương pháp thiết kế bộ lọc IIR

Như đã nói ở phần trước, các kỹ thuật thiết kế lọc IIR dựa trên bộ lọc Analog đã có để thu được các bộ lọc số. Chúng ta thiết kế các bộ lọc Analog này theo các bộ lọc điển hình.

ở phần này, ta sẽ tìm hiểu các phần chính sau:

1. Các đặc trưng và phương pháp thiết kế các bộ lọc thông thấp Analog điển hình.
2. Các phép biến đổi bộ lọc để thu được bộ lọc số thông thấp từ bộ lọc Analog.
3. Các phép biến đổi băng tần để thu được các bộ lọc số khác từ bộ lọc số thông thấp

2.1 Các đặc trưng của bộ lọc Analog điển hình

Có ba kiểu bộ lọc Analog điển hình được sử dụng rộng rãi trong thực tế:

- Thông thấp Butterworth.
- Thông thấp Chebyshev (Kiểu I và II).
- Thông thấp Elliptic.

2.1.1 Bộ lọc thông thấp Butterworth

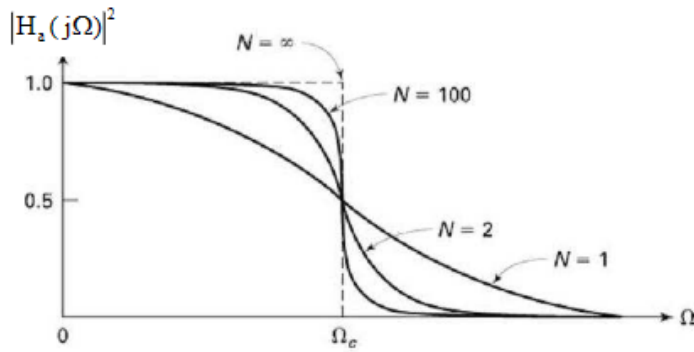
2.1.1.a Các đặc trưng:

Bộ lọc này được đặc trưng bởi tính chất đáp ứng biên độ là bằng phẳng trong cả dải thông và dải chặn

Đáp ứng bình phương-biên độ của bộ lọc thông thấp bậc-N:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_C}\right)^{2N}} \quad \Omega_C \text{ là tần số cắt (rad/s)}$$

Đồ thị đáp ứng bình phương-biên độ:



Từ đồ thị trên ta có nhận xét:

-

$$|H_a(0)|^2 = 1$$

với mọi N.

- $|H_a(j\Omega_c)|^2 = 0.5$ với mọi N (hệ số suy giảm 3dB ở Ω_c)
- $|H_a(j\Omega_c)|^2$ đơn điệu giảm theo Ω
- Tiến đến bộ lọc lý tưởng $N \rightarrow \infty$

Xác định hàm truyền hệ thống $H_a(s)$

$$H_2(s)H_2(-s) = |H_2(j\Omega)|^2 \Big|_{\Omega=s/j} = \frac{1}{1 + \left(\frac{s}{j\Omega_c}\right)^{2N}} = \frac{(j\Omega_c)^{2N}}{s^{2N} + (j\Omega_c)^{2N}} = \frac{(j\Omega_c)^{2N}}{\prod_{k=1}^{2N} (s - p_k)}$$

Các điểm cực: $p_k = (-1)^{\frac{k+1}{2N}} (j\Omega_c) = \Omega_c e^{j\frac{\pi}{2N}(2k+1)}$, $k = 0, 1, \dots, 2N-1$

Nhận xét về các điểm cực của $H_a(s)H_a(-s)$

- Có $2N$ điểm cực được phân bố đều đặn trên đường tròn bán kính Ω_c với khoảng cách góc
- Với N lẻ, $p_k = \Omega_c e^{j\frac{k\pi}{N}}$, $k=0, 1, \dots, 2N-1$
- Với N chẵn, $p_k = \Omega_c e^{j\frac{k\pi}{2N} + \frac{k\pi}{N}}$, $k=0, 1, \dots, 2N-1$
- Đối xứng theo trục ảo.
- Một điểm cực không bao giờ rơi vào trục ảo, và rơi vào trục thực chỉ nếu N là lẻ.
- Một bộ lọc ổn định và nhân quả $H_a(s)$ có thể được xác định bằng cách chọn các điểm cực trong nửa mặt phẳng trái, và $H_a(s)$ có thể được viết dưới dạng:

$$H_a(s) = \frac{\Omega_c^N}{\prod_{LHP \text{ poles}} (s - p_k)}$$

Thi hành trên Matlab:

Hàm `[z,p,k]=buttap(N)`

- Để thiết kế một bộ lọc Analog Butterworth chưa chuẩn hóa với $\Omega_c = 1$ bậc N.

- z:zeros;p:poles;k:gain value.

Hàm `[a,b]=u-buttap(N,Omegac)`

- Để thiết kế một bộ lọc Analog Butterworth chưa chuẩn hoá với Ω_c tùy ý, bậc N

- Cung cấp một cấu trúc dạng trực tiếp với: b là tử thức, a là mẫu thức.

Hàm `[C,B,A]=sdir2cas(b,a)`

- Chuyển đổi dạng trực tiếp thành dạng ghép tầng.

2.1.1.b Các phương trình thiết kế:

Bộ lọc thông thấp analog được đặc trưng bởi các thông số Ω_p, R_p, Ω_s và A_s . Vì vậy ưu điểm của thiết kế trong trường hợp bộ lọc Butterworth là thu được bậc N và tần số cắt Ω_c .

Chúng ta muốn:

- Tại $\Omega = \Omega_p$, $-10 \log_{10} |H_a(j\Omega)|^2 = R_p$ hay: $-10 \log_{10} \left(\frac{1}{\left(1 + \frac{\Omega_p}{\Omega_c}\right)^{2N}} \right) = R_p$
- Tại $\Omega = \Omega_s$, $-10 \log_{10} |H_a(j\Omega)|^2 = A_s$ hay: $-10 \log_{10} \left(\frac{1}{\left(1 + \frac{\Omega_s}{\Omega_c}\right)^{2N}} \right) = A_s$

Giải 2 phương trình trên ta thu được:

- Bậc $N = \left\lceil \frac{\log_{10} \left[\frac{(10^{R_p/10} - 1)}{(10^{A_s/10} - 1)} \right]}{2 \log_{10} (\Omega_p / \Omega_s)} \right\rceil$
- Tần số cắt Ω_c :
 - Để đáp ứng thông số kỹ thuật tại Ω_p : $\Omega_c = \frac{\Omega_p}{\sqrt[2N]{(10^{R_p/10} - 1)}}$
 - Để đáp ứng thông số kỹ thuật tại Ω_s : $\Omega_c = \frac{\Omega_s}{\sqrt[2N]{(10^{A_s/10} - 1)}}$

Thi hành trên Matlab:

Hàm `[b,a]=afd-butt(Wp,Ws,Rp,As)`

Để thiết kế bộ lọc thông thấp analog Butterworth, cho bởi các chỉ tiêu của nó

Hàm `[db,mag,pha,w] = freqs-m(b,a,wmax)`

-Đáp ứng biên độ tuyệt đối cũng như tương đối theo thang dB và đáp ứng pha.

Hàm `[ha,x,t] = impulse(b,a)`

-Đáp ứng xung $h_a(t)$ của bộ lọc Analog.

Phần tiếp theo sẽ giới thiệu thêm về các bộ lọc thông thấp điển hình khác: Chebyshev, Ellipic, nhưng do giới hạn yêu cầu của báo cáo này nên sẽ không đi sâu như đối với bộ lọc Butterworth.

2.1.2 Bộ lọc Chebyshev I:

- Các bộ lọc Chebyshev-I: Có đáp ứng cân bằng gợn sóng trong dải thông.
- Các bộ lọc Chebyshev-II: Có đáp ứng cân bằng gợn sóng trong dải chặn
- Các bộ lọc Butterworth: Có đáp ứng đơn điệu trong cả hai dải
- Lưu ý rằng chọn một bộ lọc cân bằng gợn sóng thay vì bộ lọc đơn điệu, ta thu được một bộ lọc có bậc-thấp.
- Vì vậy các bộ lọc Chebyshev cho bậc thấp hơn so với các bộ lọc Butterworth có cùng chỉ tiêu.

2.1.2.a Bộ lọc Chebyshev I:

- Đáp ứng bình phương biên độ:

$$|H_s(j\Omega)|^2 = \frac{1}{1 + \varepsilon^2 T_N^2\left(\frac{\Omega}{\Omega_c}\right)}$$

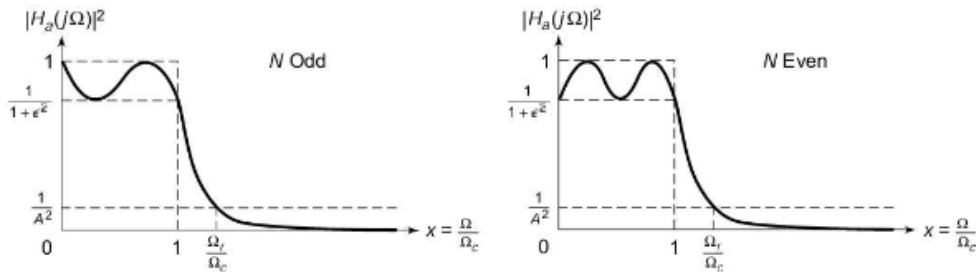
Trong đó: N là bậc bộ lọc

ε là hệ số gợn sóng dải thông

- Đa thức Chebyshev bậc N

$$T_N(x) = \begin{cases} \cos(N \cos^{-1}(x)), & 0 \leq x \leq 1 \\ \cosh(\cosh^{-1}(x)), & 1 < x < \infty \end{cases} \quad \text{khi } x = \frac{\Omega}{\Omega_c}$$

- Với $0 < x < 1$, $T_N(x)$ dao động giữa -1 và 1
- Với $1 < x < \infty$, $T_N(x)$ tăng đơn điệu đến vô cùng



Hai dạng đồ thị của đáp ứng bình phương-biên độ (N lẻ và N chẵn)

-Nhận xét:

-Tại $x=0$ (hoặc $\Omega=0$)

$$|H_a(j0)|^2 = 1 \quad (N \text{ odd})$$

$$|H_a(j0)|^2 = \frac{1}{1+\varepsilon^2} \quad (N \text{ even})$$

-Tại $x=1$ (hoặc $\Omega = \Omega_c$)

$$|H_a(j1)|^2 = \frac{1}{1+\varepsilon^2} \quad (\text{all } N)$$

-Tại $x > 1$ (hoặc $\Omega > \Omega_c$), $|H_a(jX)|^2$ giảm đơn điệu về 0.

-Tại

$$x = \Omega_r, |H_a(jX)|^2 = \frac{1}{A^2}$$

$H_a(s)$ nhân quả và ổn định:

Để xác định một hàm $H_a(s)$ nhân quả và ổn định, ta phải tìm các điểm cực của $H_a(s)H_a(-s)$ thu được bằng cách tìm nghiệm của:

$$1 + \varepsilon^2 T_N^2\left(\frac{s}{j\Omega_c}\right)$$

Có thể chỉ ra rằng nếu

$$p_k = \sigma_k + j\Omega_k$$

, $k=0,1,\dots,N-1$ là nghiệm (nửa mặt phẳng trái)
của đa thức trên thì:

$$\sigma_k = (a\Omega_c) \cos\left[\frac{\pi}{2} + \frac{(2k+1)\pi}{2N}\right]$$

$$k=0,\dots,N-1$$

$$\Omega_k = (b\Omega_c) \sin\left[\frac{\pi}{2} + \frac{(2k+1)\pi}{2N}\right]$$

Trong đó

$$a = 1/2(\sqrt[N]{\alpha} - \sqrt[N]{\frac{1}{\alpha}}), b = 1/2(\sqrt[N]{\alpha} + \sqrt[N]{\frac{1}{\alpha}})$$

Và

$$\alpha = \frac{1}{\varepsilon} + \sqrt{1 + \frac{1}{\varepsilon^2}}$$

Các điểm cực rơi trên một ellipse với trục chính $b\Omega_c$ và trục phụ $a\Omega_c$.
Hàm hệ thống là:

$$H_z(s) = \frac{K}{\prod_k (s + p_k)}$$

Với K là một hệ số chuẩn hóa được chọn để:

$$H_z(j0) = \begin{cases} 1, & N \text{ odd} \\ \frac{1}{\sqrt{1+\varepsilon^2}}, & N \text{ even} \end{cases}$$

Thi hành trên matlab

Hàm $[z,p,k]=\text{chelap}(N,R_p)$

- Để thiết kế một bộ lọc analog chuẩn hoá Chebyshev-I analog có bậc N và gợn sóng dải thông R_p
- z mảng các điểm không
- p mảng các điểm cực trong p
- Giá trị độ lợi k

Hàm $[b,a]=\text{u-chblap}(N,R_p,\Omega_c)$

- Trả lại $H_a(s)$ theo dạng trực tiếp.

Các phương trình thiết kế:

Cho Ω_p , Ω_s , R_p và A_s , ba tham số được yêu cầu để xác định một bộ lọc Chebyshev-I

Ta có:

$$\varepsilon = \sqrt{10^{0.1R_p} - 1} \quad \text{and} \quad A = 10^{A_s/20}$$

$$\Omega_c = \Omega_p \quad \text{and} \quad \Omega_r = \frac{\Omega_s}{\Omega_c}$$

$$g = \sqrt{(A^2 - 1) / \varepsilon^2}$$

$$N = \left\lceil \frac{\log_{10} \left[g + \sqrt{g^2 - 1} \right]}{\log_{10} \left[\Omega_r + \sqrt{\Omega_r^2 - 1} \right]} \right\rceil$$

Thi hành trên Matlab dùng hàm $[b,a] = \text{afd_chbl}(W_p, W_s, R_p, A_s)$

2.1.2.b Bộ lọc Chebyshev II:

Liên quan đến bộ lọc Chebyshev-I thông qua một phép biến đổi đơn giản.

Nó có dải thông đơn điệu và dải chặn cân bằng gợn sóng, nghĩa là bộ lọc này có cả các điểm cực và các điểm không trong mặt phẳng- s .

Vì vậy các đặc trưng trễ nhóm là tốt hơn (và đáp ứng pha tuyến tính hơn) trong dải thông so với bộ lọc Chebyshev-I prototype.

Đáp ứng bình phương biên độ:

$$|H_a(j\Omega)|^2 = \frac{1}{1 + (\varepsilon^2 T_N^2(\frac{\Omega_c}{\Omega}))^{-1}}$$

2.1.3 Bô lọc thông thấp Elliptic:

Các bộ lọc này thường cân bằng gợn sóng ở dải thông cũng như dải chắn. Chúng có các đặc trưng đáp ứng biên độ tương tự như các bộ lọc FIR cân bằng gợn sóng.

Vì vậy các bộ lọc elliptic là các bộ lọc tối ưu trong đó đạt được bậc tối thiểu N đối với các chỉ tiêu đã cho

Các bộ lọc này, vì nhiều lý do đã xét trước đây, là rất khó để phân tích và thiết kế.

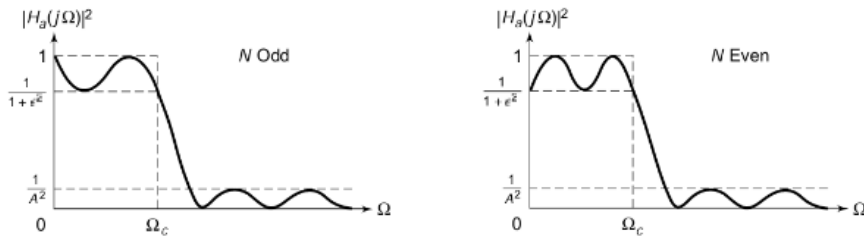
Không thể thiết kế chúng bằng các công cụ đơn giản, và thường phải dùng các chương trình hoặc bảng để thiết kế.

Đáp ứng bình phương biên độ:

-N :bậc

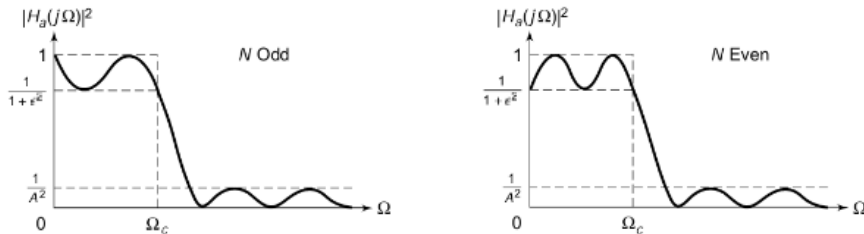
ε :gợn sóng dải thông

$U_N()$ là hàm Jacobian elliptic bậc-N



Hai dạng đồ thị của đáp ứng bình phương-biên độ (N lẻ và N chẵn)

Tính toán cho bộ lọc bậc N:



Hai dạng đồ thị của đáp ứng bình phương-biên độ (N lẻ và N chẵn)

2.1.4 Các đáp ứng pha của bộ lọc điển hình:

Bộ lọc Elliptic cho tính năng tối ưu về đáp ứng bình phương-biên độ nhưng có đáp ứng pha phi tuyến hơn trong dải thông (không thích hợp cho nhiều ứng dụng).

Ngay cả khi chúng ta quyết định không lo lắng gì đến đáp ứng pha trong thiết kế, pha vẫn giữ vai trò quan trọng trên toàn hệ thống.

Đối với các bộ lọc Butterworth, có đáp ứng biên độ bằng phẳng tối đa và đòi hỏi bậc N cao hơn (nhiều điểm cực hơn) để đạt được cùng một chỉ tiêu dải chắn. Tuy nhiên chúng có một đáp ứng pha không tuyến tính trong dải thông.

Các bộ lọc Chebyshev có các đặc tính pha nằm ở giữa.

Vì vậy trong các ứng dụng thực tế chúng ta xem xét các bộ lọc Butterworth cũng như Chebyshev, cộng thêm các bộ lọc elliptic.

Việc lựa chọn phụ thuộc vào cả bậc của bộ lọc (thường ảnh hưởng đến tốc độ xử lý và độ phức tạp thi hành) và các đặc tính pha (để điều khiển méo).

2.2 Phép biến đổi bộ lọc tương tự thành bộ lọc số

Sau khi khảo sát các tiếp cận khác nhau để thiết kế các bộ lọc tương tự, chúng ta sẵn sàng biến đổi chúng thành bộ lọc số. Các phép biến đổi này đạt được bằng cách bảo toàn các aspects khác nhau của các bộ lọc tương tự và lọc số.

Biến đổi bất biến xung

Bảo toàn hình dạng của đáp ứng xung từ lọc tương tự thành lọc số

- Kỹ thuật xấp xỉ sai phân hữu hạn

Chuyển đổi biểu diễn một phương trình vi phân thành một phương trình sai phân tương ứng.

- Bất biến bước nhảy

Bảo toàn hình dạng của đáp ứng bước nhảy

- Biến đổi song tuyến tính

Bảo toàn biểu diễn hàm hệ thống từ miền tương tự sang miền số

Trong phần này chỉ đề cập đến biến đổi bất biến xung và biến đổi song tuyến tính.

2.2.1 Biến đổi bất biến xung:

Trong phương pháp này chúng ta muốn đáp ứng xung của bộ lọc số trông tương tự như đáp ứng xung của bộ lọc chọn tần analog.

Lấy mẫu $h_a(t)$ ở các chu kỳ lấy mẫu T ta thu được $h(n)$:

$$h(n) = h_a(nT)$$

T được chọn sao hình dạng của $h_a(t)$ được giữ bởi mẫu, lúc này:

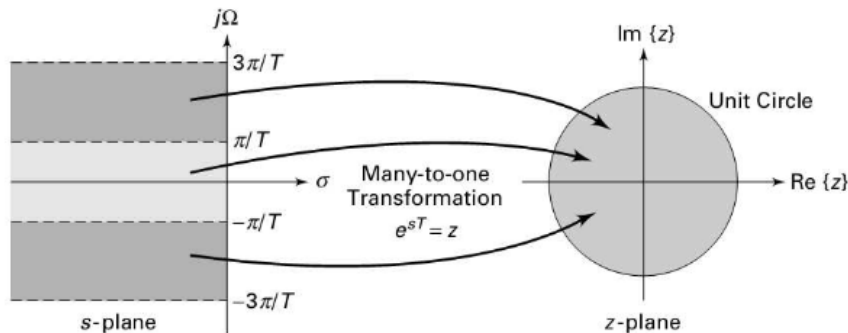
$$\omega = \Omega T \text{ or } e^{j\omega} = e^{j\Omega T}$$

Do $z = e^{j\omega}$ trên đường tròn đơn vị và $s = j\Omega$ trên trục ảo, ta có phép biến đổi sau đây từ mặt phẳng-s sang mặt phẳng-z:

$$z = e^{sT}$$

Quan hệ giữa hàm hệ thống $H(z)$ và $H_a(s)$ trong miền tần số:

$$H(z) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(s - j\frac{2\pi}{T}k\right)$$



Ảnh xạ mặt phẳng phức trong phép biến đổi bất biến xung

Thủ tục thiết kế

Với các chỉ tiêu đã cho w_p , w_s , R_p , A_s , chúng ta muốn xác định $H(z)$ bằng cách thiết kế trước tiên một bộ lọc analog tương đương và sau đó ánh xạ chúng thành bộ lọc số mong muốn. Các bước như sau:

1. Chọn T và xác định các tần số analog: $\Omega_p = \frac{\omega_p}{T}$ and $\Omega_s = \frac{\omega_s}{T}$
2. Thiết kế một bộ lọc analog $H_a(s)$ sử dụng các đặc tính của một trong ba bộ lọc điển hình trong phần trước.
3. Sử dụng phép khai triển riêng phần, khai triển $H_a(s)$ thành: $H_a(s) = \sum_{k=1}^N \frac{R_k}{s - p_k}$
4. Biến đổi các điểm cực analog $\{p_k\}$ thành các điểm cực số $\{e^{p_k T}\}$ để thu được bộ lọc số:

$$H(z) = \sum_{k=1}^N \frac{R_k}{1 - e^{p_k T} z^{-1}}$$

Thi hành matlab

Hàm[b,a]=impzinvr(c,d,T)

b= các hệ số ở tử thức của H(Z)

a = các hệ số ở mẫu thức của H(Z)

c = các hệ số ở tử thức của $H_a(S)$

d = các hệ số ở mẫu thức của $H_a(S)$

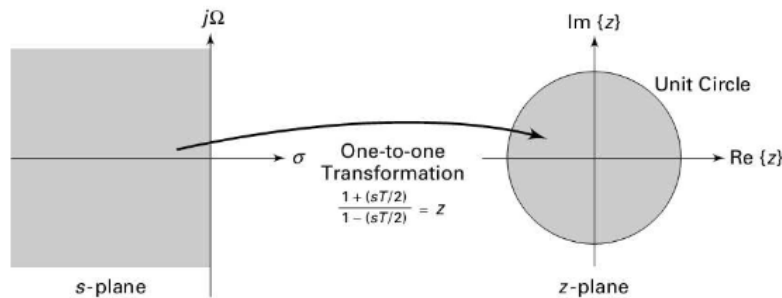
T = chu kỳ lấy mẫu

2.2.2 Biến đổi song tuyến tính:

Ảnh xạ này là phương pháp biến đổi tốt nhất.

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \Rightarrow z = \frac{1 + sT/2}{1 - sT/2}$$

Ở đây T là một tham số. Một tên gọi khác của phép biến đổi này là *Biến đổi phân tuyến tính* (linear fractional) vì ta có: $\frac{T}{2}sz + \frac{T}{2}s - z + 1 = 0$ là tuyến tính với mỗi biến (s hoặc z) nếu biến còn lại được cố định, hoặc song tuyến tính với s và z.



Ảnh xạ mặt phẳng phức trong phép biến đổi song tuyến tính

Thủ tục thiết kế

Với các chỉ tiêu đã cho của bộ lọc số w_p , w_s , R_p , A_s , chúng ta cần xác định $H(z)$. Các bước như sau:

1. Chọn một giá trị T tùy ý, và có thể đặt $T = 1$.
2. Chuyển đổi các tần số cắt ω_p và ω_s , nghĩa là tính toán Ω_p và Ω_s sử dụng:

$$\Omega_p = \frac{2}{T} \tan\left(\frac{\omega_p}{2}\right), \quad \Omega_s = \frac{2}{T} \tan\left(\frac{\omega_s}{2}\right)$$

3. Thiết kế một bộ lọc thông thấp $H_a(s)$ phù hợp các chỉ tiêu này.
4. Cuối cùng lấy: $H(z) = H_a\left(\frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}\right)$. Và nhận được $H(z)$ là một hàm hữu tỉ theo z^{-1}

Thi hành matlab:

Hàm $[b,a] = \text{bilinear}(c,d,Fs)$
b = các hệ số ở tử thức của $H(Z)$
a = các hệ số ở mẫu thức của $H(Z)$
c = các hệ số ở tử thức của $H_a(S)$
d = các hệ số ở mẫu thức của $H_a(S)$
Fs = tần số lấy mẫu

3 Chương trình thiết kế

Thiết kế dựa vào bộ lọc Butterworth và phép biến đổi song tuyến tính.

Có hai hướng để thiết kế:

- Hướng thứ nhất: sử dụng trực tiếp các thông số kỹ thuật yêu cầu để thiết kế bộ lọc số thông thấp Analog Butterworth, sau đó biến đổi song tuyến tính để thu được bộ lọc số thông thấp.
- Hướng thứ 2: Sử dụng một bộ lọc số thông thấp đã thiết kế trước (dùng bộ lọc Butterworth và biến đổi song tuyến tính), sau đó thực hiện phép chuyển đổi băng tần để thu được bộ lọc thông thấp số có thông số kỹ thuật theo yêu cầu. .

3.1 Tính toán thiết kế

Cho bộ lọc thông thấp có thông số như sau:

$$\omega_p = 0.3\pi, \omega_s = 0.6\pi, A_s = 50dB, R_p = 0.5dB$$

- Chọn chu kỳ lấy mẫu $T = 1$
- Chuyển đổi ω_p, ω_s (tính toán Ω_p, Ω_s)

$$\Omega_p = \frac{2}{T} \tan \frac{\omega_p}{2} = \frac{2}{1} \tan \frac{0.3\pi}{2} = 1.019(\text{rad/s})$$

$$\Omega_s = \frac{2}{T} \tan \frac{\omega_s}{2} = \frac{2}{1} \tan \frac{0.6\pi}{2} = 2.753(\text{rad/s})$$

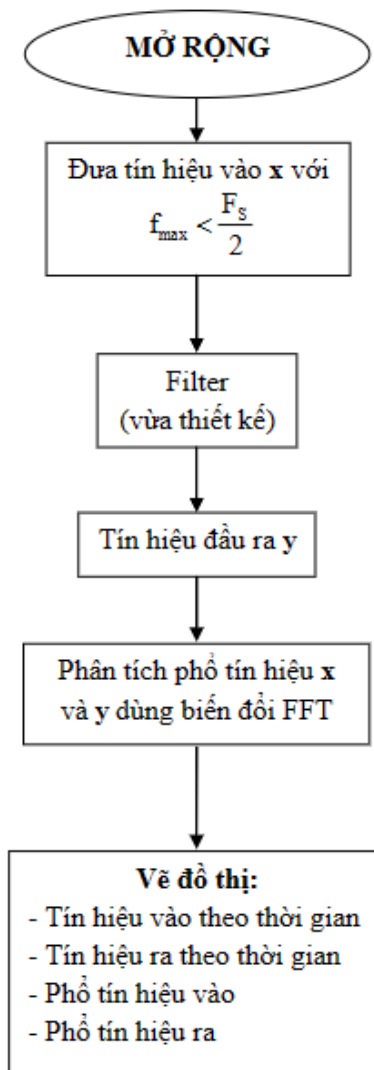
$$N=7$$

$$\Omega_c = \frac{\Omega_p}{\sqrt[2N]{10^{R_p/10} - 1}} = \frac{1.019}{\sqrt[14]{10^{0.5/10} - 1}} = 1.184(\text{rad/s})$$

$$\Omega_c = \frac{\Omega_s}{\sqrt[2N]{10^{A_s/10} - 1}} = \frac{1.019}{\sqrt[14]{10^{50/10} - 1}} = 1.21(\text{rad/s})$$

Ta có thể chọn Ω_c ở giữa hai giá trị trên, chọn $\Omega_c = 1.2(\text{rad/s})$

3.2 Thuật toán giải quyết



3.3 Thiết kế bộ lọc bằng python:

3.3.1 Thiết lập bộ lọc IIR bằng phương pháp Butterworth

- Khai báo các thư viện cần dùng. Các thư viện này phải được tải sẵn trong máy.

```

# import required modules
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
import math
from scipy.signal import iirfilter, freqz, butter, tf2zpk, dimpulse, dlti, buttord, sosfreqz, sosfiltfilt, sos2zpk, unit_impulse, sosfilt
from scipy.io.wavfile import write
  
```

Tạo mạch lọc thông thấp IIR bằng phương pháp Butterworth (có bậc thấp nhất)

```
# sampling frequency
fs = 1000
# pass band frequency
wp = 150

# stop band frequency
ws = 300

# pass band ripple
gpass = 0.5

# stop band attenuation
gstop = 50

# Bậc của bộ lọc
# Hàm tìm bậc bé nhất của bộ lọc:
ord, wn = buttord(wp, ws, gpass, gstop, False, fs)
print('Bac cua bo loc la ', ord)
print('Tan so cat ', wn)

# Thiết kế bộ lọc IIR, xuất ra dạng theo dạng ma trận sos
sos = iirfilter(ord, wn, rp=None, rs=None, btype='low', analog=False, ftype='butter', output='sos', fs=fs)
# có thể dùng hàm butter có sẵn của python
b1, a1 = butter(ord, wn, btype='low', output='ba', analog=False, fs=fs)
print('So phép tính nhan la ', Len(b1) + Len(a1))
```

- Xuất số phép tính nhân dựa vào tổng hệ số của tử và mẫu của đáp ứng tần số.

```
b1, a1 = butter(ord, wn, btype='low', output='ba', analog=False, fs=fs)
print('So phép tính nhan la ', Len(b1) + Len(a1))
```

- Xuất đáp ứng biên độ của bộ lọc

```
# Vẽ đáp ứng biên độ
w, h = sosfreqz(sos, fs=fs)
#plt.subplot(2, 2, 2)
plt.plot(w, 20 * np.log10(abs(h)))
plt.xlim(0, fs/2)
plt.ylim(-150, 50)
plt.title('IIR Butterworth filter frequency response')
plt.xlabel('Frequency [Hz]')
plt.ylabel('Amplitude [dB]')
plt.margins(0, 0.1)
plt.grid(which='both', axis='both')
plt.axhline(-50, color='red', ls='--', label = '-50 dB')
plt.legend(loc='best')
plt.axvline(150, color='orange', label = '150 Hz')
plt.legend(loc='best')
plt.axvline(300, color='blue', label = '300 Hz')
plt.legend(loc='best')
plt.show()
```

- Xuất giản đồ cực và zero của bộ lọc

```
# Vẽ giản đồ cực và zero theo dạng ma trận sos
z, p, k = sos2zpk(sos)
#plt.subplot(2, 2, 1)
plt.plot(np.real(z), np.imag(z), 'ob', markerfacecolor='none')
plt.plot(np.real(p), np.imag(p), 'xr')
plt.legend(['Zeros', 'Poles'], loc=2)
plt.title('Pole / Zero Plot')
plt.xlabel('Real')
plt.ylabel('Imaginary')
plt.grid()
plt.show()
```

- Xuất đáp ứng xung của bộ lọc (100 trọng số đầu tiên)


```
# Vẽ đáp ứng xung của 100 trọng số đầu tiên
x1= unit_impulse(100)
y_sos = sosfilt(sos, x1)
tt = np.arange(0,100)
plt.plot(tt, y_sos, linewidth=1)
markerline, stemlines, baseline = plt.stem(tt, y_sos)
plt.setp(stemlines, 'linewidth', 1)
plt.grid()
plt.xlabel('Impulse response [100]')
plt.ylabel('Amplitude')
plt.show()
```

-Tạo tín hiệu nhiễu

```
#cac tin hieu vao
f1 = 180 # Frequency of 1st signal
f2 = 400 # Frequency of 2nd signal

# Generate the time vector of 1 sec duration
t = np.linspace(0, 1,1000) # Generate 1000 samples in 1 sec

# Generate the signal containing f1 and f2
sig = np.sin(2 * np.pi * f1 * t) + np.sin(2 * np.pi * f2 * t)
```

- Áp bộ lọc vào tín hiệu nhiễu và vẽ tín hiệu đầu ra đầu vào

```
# Display the signal
fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True)
ax1.plot(t, sig)
ax1.set_title('Tín hiệu trước khi lọc')
ax1.axis([0, 1, -2, 2])

# Áp bộ lọc vào tín hiệu

# Lọc theo một chiều (one-dimension)
#y = sosfilt(sos, sig, 0)

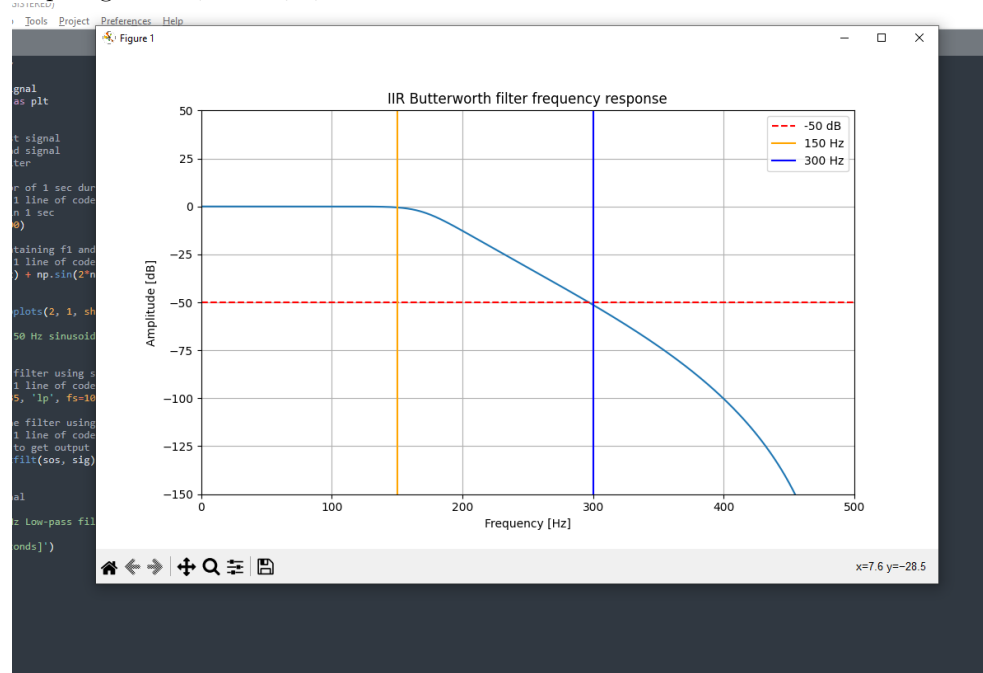
# Lọc tiến lùi (Forward-backward filtering)
y = sosfiltfilt(sos, sig, 0)
# Display the output signal
ax2.plot(t, y)
ax2.set_title('Sau khi lọc')
ax2.axis([0, 1, -2, 2])
ax2.set_xlabel('Time [seconds]')
plt.tight_layout()
plt.show()
```

3.3.2 Kết quả thực hiện việc lọc tín hiệu bằng bộ lọc IIR bằng phương pháp Butterworth

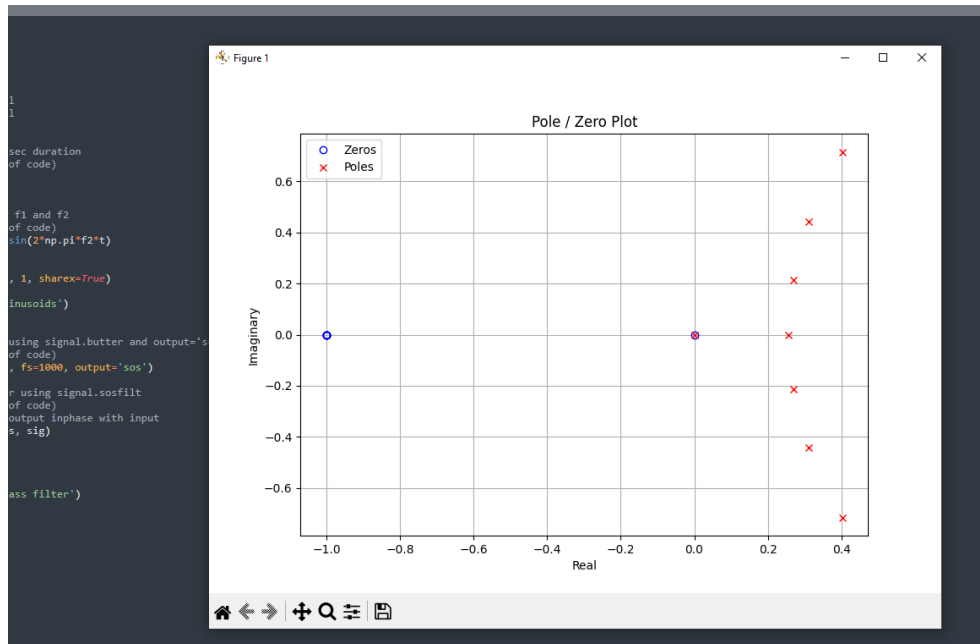
- Ta có bậc của bộ lọc là 7
- Tần số cắt là 170 Hz
- Số phép tính nhân cần thực hiện với mỗi mẫu đầu vào là 16.

```
Bậc của bộ lọc là 7  
Tan số cắt 170.1738172018734  
Số phép tính nhân là 16  
[Finished in 467.8s]
```

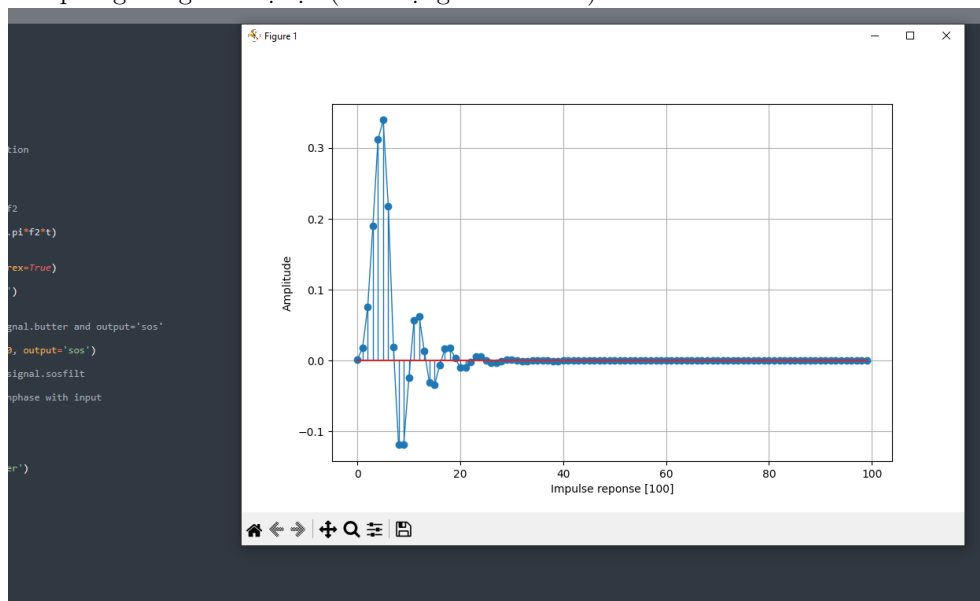
Đáp ứng biên độ của bộ lọc.



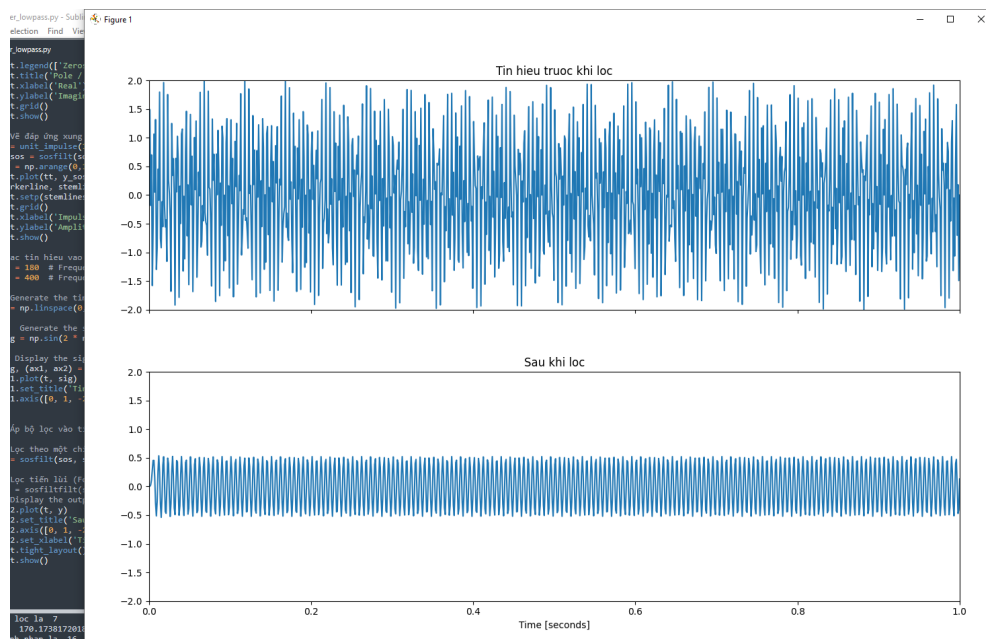
- Giảm đồ cực và zero của bộ lọc.



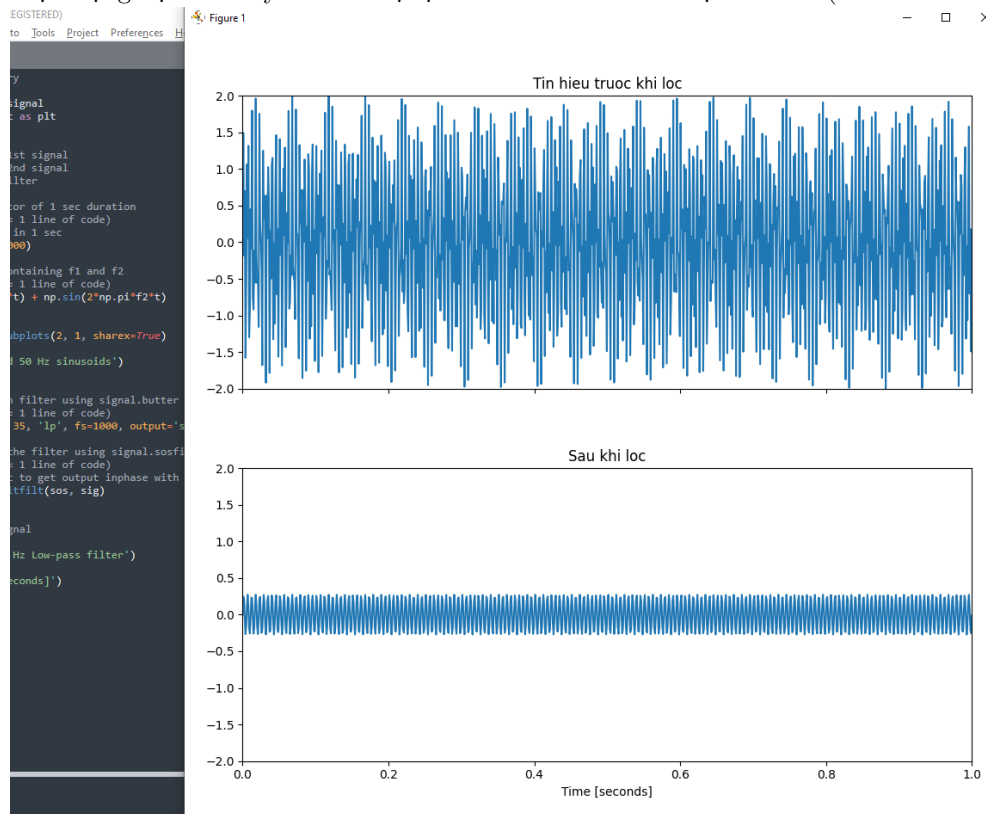
- Đáp ứng xung của bộ lọc (100 trọng số đầu tiên).



- Tín hiệu trước và sau khi được lọc
Lọc theo một chiều (one-dimension)



-Tận dụng bộ nhớ máy tính và bộ lọc có sẵn để tiến hành lọc tiến-lùi (Forward-backward filtering).



4 Đánh giá kết quả ,kết luận

4.1 Chỉ tiêu kỹ thuật

Đáp ứng được các thông số kỹ thuật như yêu cầu (wp,ws,As,Ps)

Đáp ứng biên độ bằng phẳng tối đa và đòi hỏi bậc N cao hơn (nhiều điểm cực hơn) để đạt được cùng một chỉ tiêu dải chắn.

Nhờ áp dụng phương pháp biến đổi song tuyến tính (phương pháp biến đổi tốt nhất) nên bộ lọc số

thu được:

- Đáp ứng tốt các chỉ tiêu kỹ thuật
- Không có aliasing (sai số lấy mẫu) trong miền tần số
- Bộ lọc thiết kế là ổn định (các điểm cực đều nằm trong đường tròn đơn vị z-plane)

4.2 Chỉ tiêu kỹ thuật:

-Quan sát quá trình mô phỏng với tín hiệu vào ở trên, ta nhận thấy bộ lọc thiết kế hoạt động tương đối tốt.

-Đáp ứng biên độ của bộ lọc IIR bằng phương pháp Butterworth có không có độ gợn trong cả dải dừng và dải dẫn nhưng dải chuyển tiếp khá lớn, ảnh hưởng tới việc lọc nhiễu

-Bộ lọc có thể lọc được tốt với một chiều lọc và lọc tiến-lùi.

4.3 Kết luận:

Tiểu luận đã tóm tắt các vấn đề lý thuyết cơ bản về thiết kế bộ lọc số IIR. Nêu các đặc tính bộ lọc số cần xác định và các phương pháp khi thiết kế bộ lọc: Sử dụng 4 kiểu Butterworth, Chebyshev 1-2, Ellipic. Tiểu luận đã đi sâu nghiên cứu phương pháp thiết kế bộ lọc IIR thông thấp sử dụng bộ lọc Butterworth và phép biến đổi song tuyến tính.

Sau khi tìm hiểu lý thuyết cơ bản của phương pháp này, tiểu luận đã vận dụng Python để minh họa lý thuyết đồng thời nêu ra bài toán thiết kế và cách giải quyết, so sánh kết quả đạt được.

Xong tiểu luận vẫn còn nhiều sai sót rất mong sự góp ý của thầy để nội dung được hoàn chỉnh hơn.

TÀI LIỆU THAM KHẢO

- [1] Lê Tiến Trường. Xử lý số tín hiệu và Wavelets. Nhà xuất bản đại học quốc gia Tp.Hồ Chí Minh.
[2] Sagnikmukherjee2. https://www.geeksforgeeks.org/digital-low-pass-butterworth-filter-in-python/?fbclid=IwAR3GiPd0Z_nseR4m4Bn69bPa5f2qazcL9uopQxC11lDIe1CW7bt6hE3X2uY.