

# Thiết kế bộ lọc FIR

## 1 IIR, FIR và cấu trúc hệ thống

Xét phương trình biểu diễn một hệ thống tổng quát:

$$\begin{aligned} \sum_{k=0}^M a_k y[n-k] &= \sum_{k=0}^N b_k x[n-k] \\ \Leftrightarrow a_0 y[n] + \sum_{k=1}^M a_k y[n-k] &= \sum_{k=0}^N b_k x[n-k] \\ \Leftrightarrow y[n] + \sum_{k=1}^M a_k y[n-k] &= \sum_{k=0}^N b_k x[n-k] \\ \Leftrightarrow \mathcal{Z}(y[n] + \sum_{k=1}^M a_k y[n-k]) &= \mathcal{Z}(\sum_{k=0}^N b_k x[n-k]) \\ \Leftrightarrow Y(z) \left(1 + \sum_{k=1}^M a_k z^{-k}\right) &= X(z) \left(\sum_{k=0}^N b_k z^{-k}\right) \\ \Leftrightarrow H(z) &= \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^M a_k z^{-k}} \end{aligned}$$

Đến đây, ta xây dựng 2 khái niệm về bộ lọc IIR và FIR, lúc này ta có 2 trường hợp

- $a_k = 0$  với  $\forall k$ , tức là mẫu số bằng 1, đây là bộ lọc FIR.
- $a_k \neq 0$  và  $\sum_{k=0}^N b_k z^{-k} = \text{const}$  với  $\forall k$ , đây là bộ lọc IIR.

Đối với bộ lọc FIR, ta có

$$H(z) = \sum_{k=0}^N b_k z^{-k} \Leftrightarrow Y(z) = \left(\sum_{k=0}^N b_k z^{-k}\right) X(z) \Leftrightarrow y[n] = \sum_{k=0}^n b_k x[n-k] \Leftrightarrow Y(z) = \left(\sum_{k=0}^N b_k z^{-k}\right) X(z)$$

Và ta lấy biến đổi  $Z$  ngược của hàm truyền  $H(z)$ , ta thu được:

$$h[n] = \mathcal{Z}^{-1}(H(z)) = \sum_{k=0}^N b_k \delta[n-k]$$

Do  $N$  hữu hạn (số bộ nhớ ta cần thiết kế), nên hiển nhiên  $h[n]$  hữu hạn, do đó bộ lọc FIR được gọi là bộ lọc **đáp ứng xung hữu hạn**. Đối với bộ lọc IIR, ta có:

$$H(z) = \frac{1}{1 + \sum_{k=1}^M a_k z^{-k}} \Leftrightarrow Y(z) = X(z) + \sum_{k=1}^M a_k Y(z) z^{-k}$$

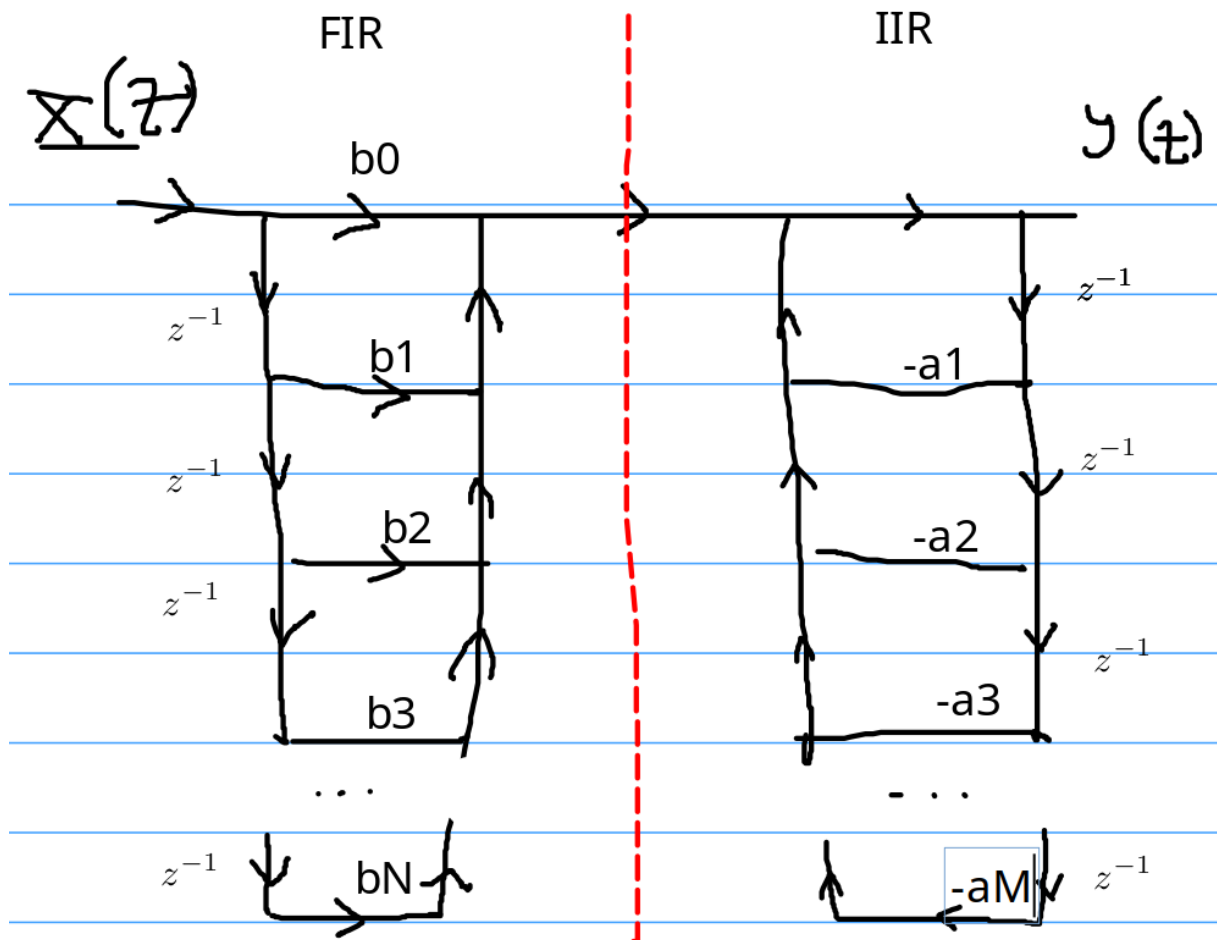
Mẫu số là một đa thức bậc  $M$ , hiển nhiên nó sẽ có  $M$  nghiệm (kể cả nghiệm bội và nghiệm phức), vậy nên với bất kì hàm  $H(z)$  nào ta luôn có thể phân tích nó thành tổng các phân thức con như sau:

$$H(z) = \frac{1}{1 + \sum_{k=1}^M a_k z^{-k}} = \sum_{i=1} \frac{A_i}{\lambda_i - z^{-1}}$$

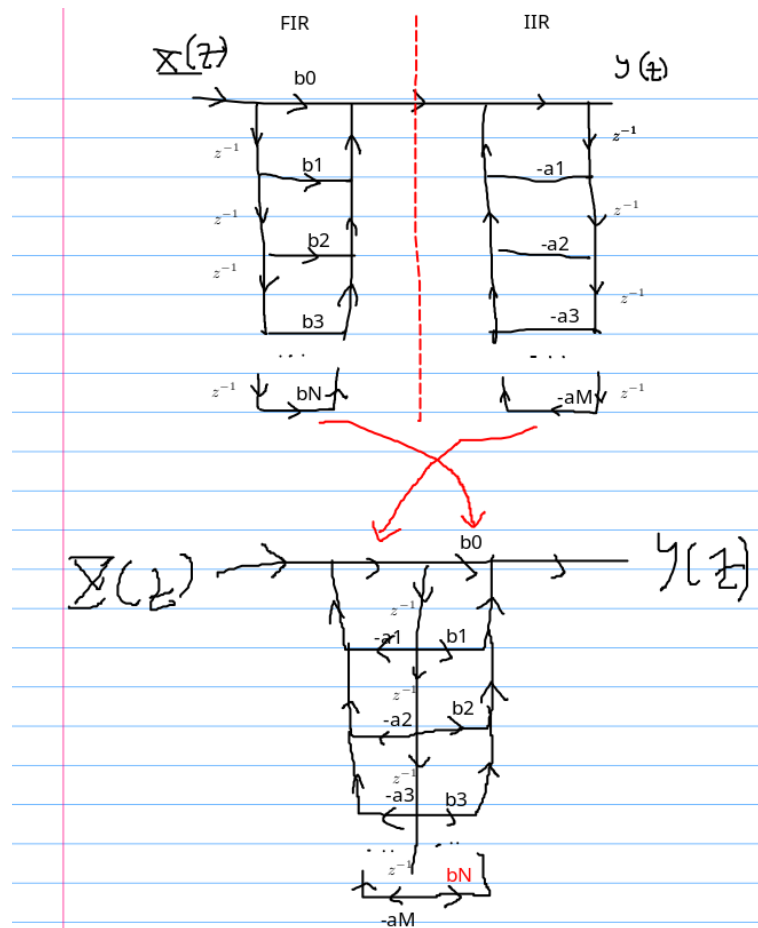
Và ta lấy biến đổi  $\mathcal{Z}$  ngược của hàm truyền  $H(z)$ , ta thu được:

$$h[n] = \mathcal{Z}^{-1}(H(z)) = \sum_{i=1} A_i e^{-\lambda_i n} u[n]$$

Do độ dài của một đáp ứng xung mũ như phương trình trên là vô hạn, do đó bộ lọc IIR được gọi là bộ lọc **đáp ứng xung vô hạn**. Ta xây dựng cấu trúc hệ thống tổng quát của cả hai dạng bộ lọc IIR và FIR như sau:



Bộ lọc FIR là bộ lọc không hồi quy (không có phản hồi ngược lại), còn bộ lọc IIR là bộ lọc tự hồi quy (có phản hồi ngược lại từ đầu ra để feedback đầu vào). Để tối ưu hóa tài nguyên, ta xây dựng kiến trúc hệ thống loại II, bằng cách **chuyển toàn bộ phần IIR sang bên trái và FIR sang bên phải** như sau:



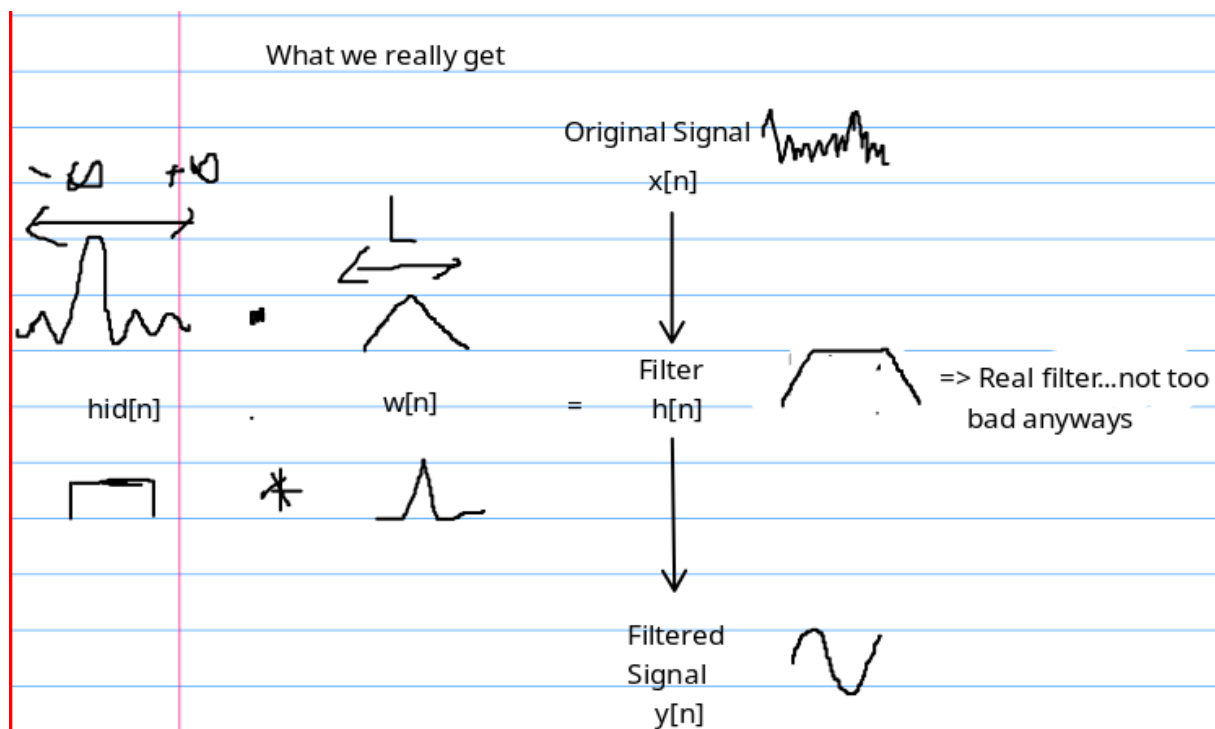
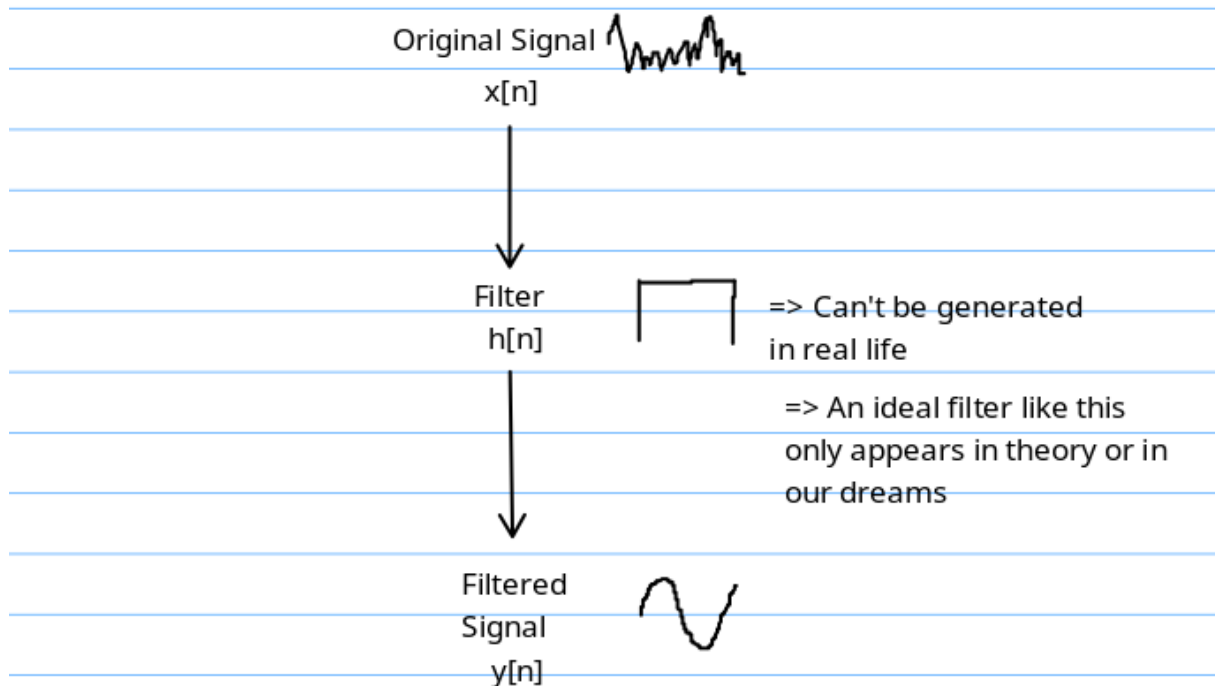
Vậy khi gặp cấu trúc hệ thống loại II, tốt nhất ta nên chuyển nó về loại I trước cho dễ nhìn, sau đó xây dựng phương trình sai phân biểu diễn hệ thống rồi làm gì thì làm. Cách chuyển đã được trình bày trên hình vẽ, chỉ cần "dịch" toàn bộ nửa bên trái sang bên phải và nửa bên phải sang bên trái là xong.

## 2 Thiết kế bộ lọc FIR

### 2.1 Ý tưởng thiết kế bộ lọc FIR

Ý tưởng của việc thiết kế bộ lọc FIR cực kì đơn giản như sau:

What we expect:



## 2.2 Thuật toán thiết kế bộ lọc LP:

- Tính tần số cắt  $\omega_c$  hoặc  $\nu_c$
- Tính đáp ứng xung lý tưởng  $h_{id}[n]$
- Đọc thông số đề bài, chú ý đến  $A_s$  và chọn cửa sổ, tính L

- Xây dựng hàm  $h[n] = w[n]h_{id}[n]$
- Nhân quả hóa hàm  $h[n]$

Ví dụ: Xác định đáp ứng xung của một bộ lọc LP lý tưởng có  $\omega_c = \frac{\pi}{4}$  và thiết kế bộ lọc FIR có hệ số 125 bằng cửa sổ Blackman.

$$h_{id}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \frac{\sin(\omega_c n)}{n\pi}$$

Thay  $\omega_c = \frac{\pi}{4}$ , xong:

$$h_{id}[n] = \frac{\sin(\frac{\pi}{4}n)}{n\pi}$$

Hệ số 125, tức là  $L = 125$ . Ta suy ra đáp ứng xung của bộ lọc dùng cửa sổ Blackman với hệ số  $L = 125$  là:

$$h[n] = h_{id}[n]w[n] \quad |n| \leq \frac{L-1}{2}$$

Cuối cùng, ta chuyển bộ lọc về bộ lọc nhân quả bằng cách dịch bộ lọc đi.

$$h_{NQ}[n] = h \left[ n - \frac{L-1}{2} \right] = h_{id} \left[ n - \frac{L-1}{2} \right] w \left[ n - \frac{L-1}{2} \right]$$

## 2.3 Thuật toán thiết kế bộ lọc HP:

Dựa trên phổ của bộ lọc LP và HP, ta có nhận xét sau:

$$H_{HP}(\omega) = 1 - H_{LP}(\omega) \quad (1)$$

$$H_{HP}(\omega) = H_{LP}(\omega - \pi) \quad (2)$$

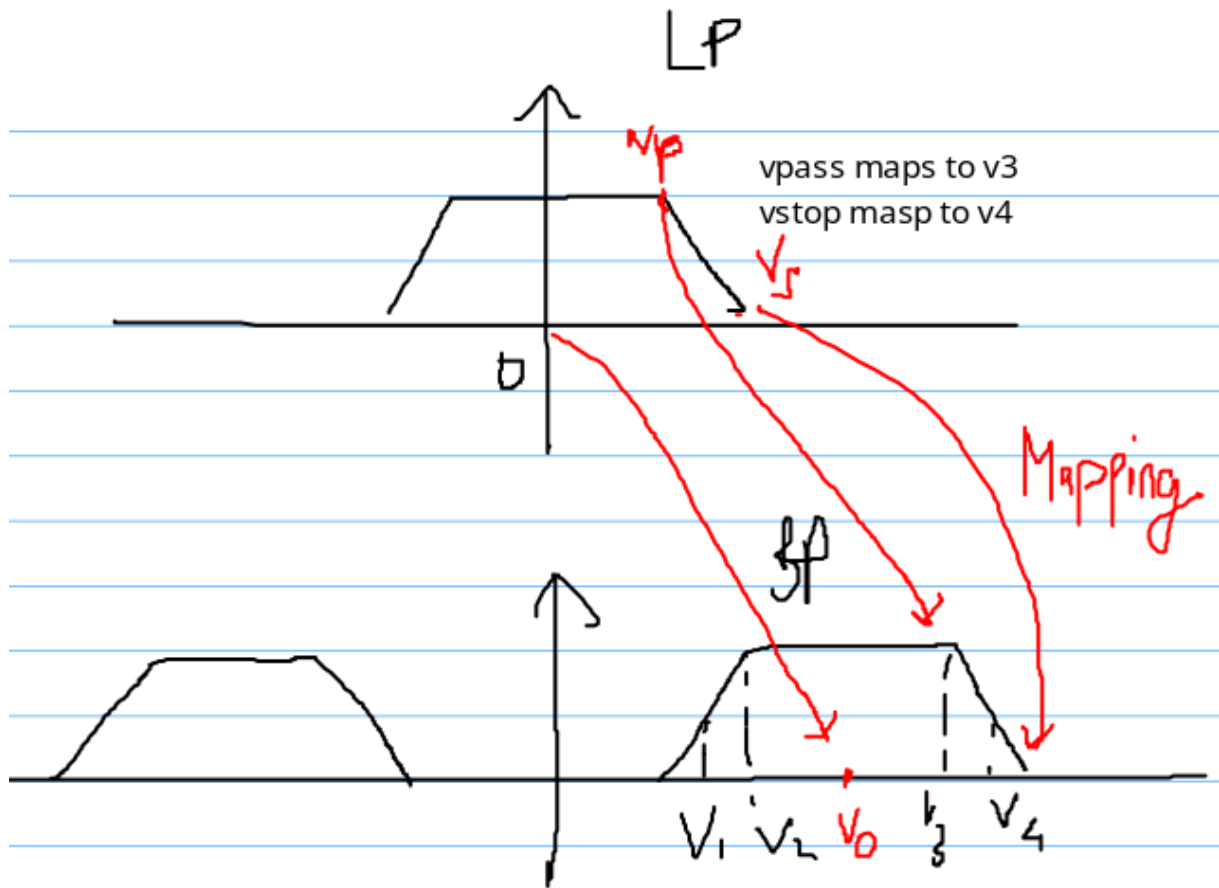
Với hai cách biểu diễn phổ của bộ lọc HP như trên, ta có 2 thuật toán tương ứng để chuyển từ bộ lọc LP sang HP tương ứng.

1. Thiết kế bộ lọc  $h_{LP}[n]$  với tần số cắt  $\omega_c = (\omega_s + \omega_p)/2$ , sau đó chuyển về bộ lọc HP qua phương trình:  $h_{HP}[n] = \delta(n) - h_{LP}[n]$ , sau đó dịch về nhân quả.
2. Thiết kế bộ lọc  $h_{LP}[n]$  với tần số cắt  $\omega_c = \pi - [(\omega_s + \omega_p)/2]$ , sau đó chuyển về bộ lọc HP qua phương trình:  $h_{HP}[n] = (-1)^n h_{LP}[n]$ , sau đó dịch về nhân quả.

Trong thực tế thường người ta sẽ hay dùng cách (2) hơn vì tạo ra xung  $\delta(n)$  khá khó.

## 2.4 Thuật toán thiết kế bộ lọc BP:

Dựa trên phổ của bộ lọc LP, ta sẽ dịch phổ của bộ lọc LP về hai phía là thu được bộ lọc BP dễ dàng.



Thuật toán thiết kế như sau:

- Đổi tất cả các tần số về tần số số
- Tính các tần số sau:

$$v_0 = \frac{v_2 + v_3}{2} = \frac{v_1 + v_4}{2}$$

$$v_s = v_3 - v_0$$

$$v_p = v_4 - v_0$$

$$v_c = \frac{v_s + v_p}{2} = \frac{v_3 + v_4}{2} - v_0$$

- Thiết kế bộ lọc  $h_{LP}[n]$  tương ứng với các thông số đã cho
- Dịch bộ lọc  $h_{LP}[n]$  bằng phương trình:

$$h_{BP}[n] = h_{LP}[n] 2 \cos(2v_0 n)$$

## 2.5 Thuật toán thiết kế bộ lọc BS

Dựa trên phổ của bộ lọc BP, ta chỉ lấy phần bù, tức là:

$$H_{BS}(\omega) = 1 - H_{BP}(\omega)$$

là xong.