

BÀI THỰC TẬP CHUYÊN ĐỀ

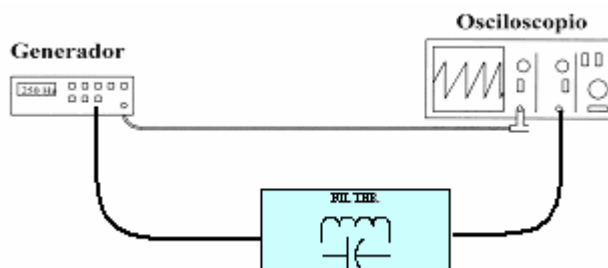
KHẢO SÁT CÁC ĐẶC TÍNH ĐÁP ỨNG BỘ LỌC THÔNG THẤP

1. Mục đích

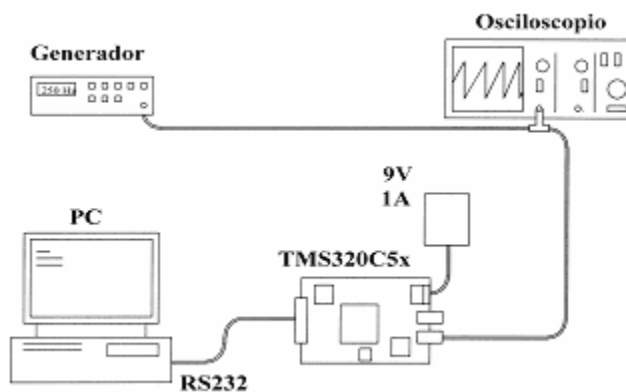
Mục đích của bài thí nghiệm này là sử dụng một bộ lọc tích cực cứng được thiết kế như là bộ lọc thông thấp 5 cực để tạo thành bộ lọc có thể thay đổi được, để có đặc tính Butterworth, hoặc Chebyshev hay là Bessel .. Nó có trở kháng lối vào khá lớn (khoảng 10k) và trở kháng lối ra khá nhỏ và vì thế hoạt động của nó không phụ thuộc nhiều vào lối vào và lối ra (Chú ý : Hãy giữ điện áp lối vào dưới 5 Volt đỉnh-đỉnh để tránh quá tải) , Bên cạnh đó sinh viên dùng chương trình MATLAB Signal Processing Toolbox để thiết kế bộ lọc này và sử dụng bản mạch Startup kit TMS320C50 DSP (Bộ xử lý số tín hiệu) với phần mềm hợp ngữ DSK để tạo ra bộ lọc đáp có các đặc tính trên.

Sơ đồ khối của thí nghiệm được mô tả như sau :

a) Sơ đồ thí nghiệm sử dụng bộ lọc tích cực cứng :



b) Sơ đồ thí nghiệm sử dụng TMS320C50 DSP



2. Dụng cụ thực nghiệm

- i) Máy phát chức năng
- ii) Bộ lọc tích cực với nguồn $\pm 15^{\text{Volt}}$
- iii) Dao động ký
- iv) Volt kế RMS
- v) Máy đo phase
- vi) Máy tính với phần mềm DSK, MatLab
- vii) Startup kit TMS320C50 DSP
- viii) phát chức năng (Function Generator);

A. PHẦN LÝ THUYẾT

1. Các đặc tính đáp ứng bộ lọc thông thấp

Trong phần này sinh viên phải phân tích, đánh giá các đặc tính tần số và đáp ứng truyền dẫn của đường truyền có đặc trưng lọc Butterworth, hoặc Chebyshev hay là Bessel theo các chỉ dẫn dưới đây :

- o Với đáp ứng tần số, phải chú ý đến giá trị suy hao đáng kể tại các tần số nào đó. Các tần số này phải được đo và vẽ theo tần số được chuẩn hoá. Với mỗi một bộ lọc, các tần số được chuẩn hoá được sử dụng để vẽ trên đồ thị sao cho chúng có thể biểu diễn đáng điệu chung nhất của bộ lọc để từ đó xem xét được đáp ứng mỗi khi có một sự thay đổi tần số.
- o Từ đáp ứng tần số, các trễ pha tại các tần số được lựa chọn phải được đo và các đặc tính độ khuếch đại, độ chậm phase, trễ phase và trễ nhóm được tính toán từ các giá trị đo được.
- o Với đáp ứng truyền dẫn, đáp ứng lỗi ra của mỗi bộ lọc khi một bước lỗi vào được sử dụng

a) Giới thiệu :

Mỗi một bộ lọc bậc-5 có một hàm truyền được chuẩn hoá dưới dạng :

$$H(s) = \frac{e}{S^5 + a S^4 + b S^3 + c S^2 + d S^1 + e}$$

Trong đó a, b, c, d, e là các hệ số của bộ lọc

- Butterworth có hàm truyền (TF) được chuẩn hoá tới $\omega_c = 1$ rad/s
- Chebyshev có 1 dB ripple ,

$$\omega_c = \cosh [(1/n) \cosh^{-1}(1/e)]$$

Do bộ lọc bậc-5 có $n=5$ và $e=0.50885$ vì thế $\omega_c \approx 1.0338$ và sao cho ω_r không dưới -3dB thì $\omega_r = (20000/1.0338) = 19346$ rad/s và do vậy TF được đặt lại bằng $s/\omega_r = 20000/19346$.

- Bessel được chuẩn hoá tới thời gian trễ $T_0 = 1$ sec để đảm bảo cho $\omega_c = 20000$ rad/s thì độ trễ phải giảm xuống

$$t_0 = 1.21 \times 10^{-4} \text{ sec}$$

TF được chuẩn hoá bằng cách đặt lại $s = st_0$, trong đó t_0 là độ trễ thiết kế

Độ trễ phase “ tần số zero “ được xác định bằng cách đặt s hay ω bằng zero và vì vậy chỉ còn hệ số bậc-1 trong $H(s)$, xác định độ trễ phase kết quả của biểu thức (bằng cách đặt $s = j\omega$) với $T_p = -f(\omega)/\omega$

Cho bộ lọc có thể chuyển đổi thành các loại bộ lọc Butterworth, hoặc Chebyshev hay là Bessel cùng với các thiết bị thí nghiệm nêu ở trên , sinh viên hãy tiến hành các bước sau đây :

b) Độ trễ phase “ tần số zero “

Các mẫu số được chuẩn hoá theo 3 hàm truyền là

✓ Butterworth $S^5 + 3.236 S^4 + 5.236 S^3 + 5.236 S^2 + 3.236 S^1 + 1$

✓ Chebyshev $S^5 + 0.908 S^4 + 1.601 S^3 + 0.889 S^2 + 0.509 S^1 + 0.104$

✓ Bessel $S^5 + 6.181 S^4 + 17.827 S^3 + 29.381 S^2 + 27.238 S^1 + 11.222$

sử dụng tần số lên tới $\omega_0 = 20\,000$ rad/s ($f_0 = 3.18$ KHz) sao cho đáp ứng tần số mong đợi phù hợp với **TF** với S được thay thế bằng $j\omega/20000$. Hãy sử dụng các thông tin này để tính toán độ trễ phase “ tần số zero ” β/ω đặc tính của mỗi một bộ lọc (bằng cách đặt $\omega \rightarrow 0$ vì thế chỉ còn lại đại lượng bậc nhất)

c) Các tần số tiệm cận

Các tần số của các bộ lọc bậc 5 này được tính toán theo hàm truyền nêu trong mục (a) trên đây

d) Quan sát đáp ứng tần số

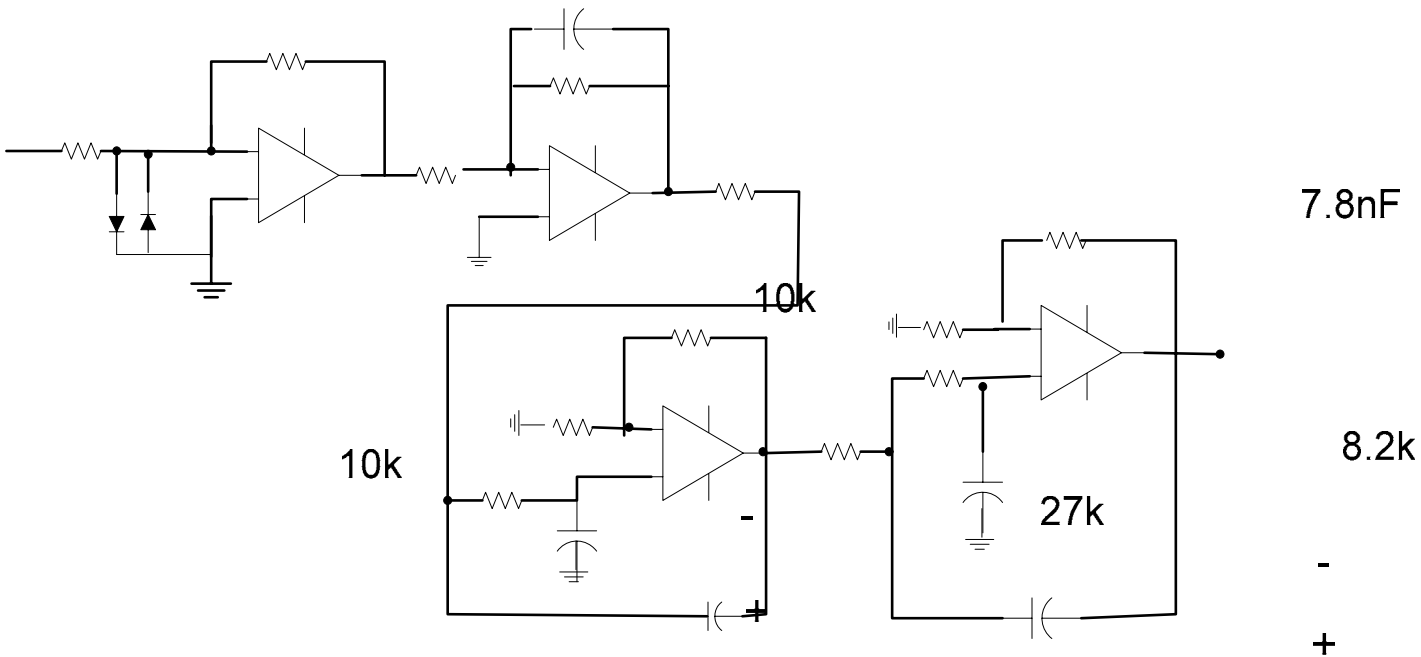
- Biên độ : Với mỗi một bộ lọc, hãy so sánh điện áp lối ra tính theo đơn vị **dB** và ghi lại các tần số mà tại đó có sự suy hao đáng kể về biên độ lối ra, tức là với Butterworth và Bessel, các tần số cho -0.5dB, -1dB, -3dB, -10dB, -20dB, -30dB và -40dB, còn với Chebyshev tại các tần số và giá trị suy hao ứng với giá trị tối đa trong băng dải và giá trị suy hao ứng với giá trị tối thiểu tại gần mép băng dải, các tần số cho cho -0.5dB, -1dB, -3dB, -10dB, -20dB, -30dB và -40dB
Hãy vẽ đáp ứng theo đơn vị **dB** đối lại với thang *logarithm* tần số được chuẩn hoá
- Hãy dùng máy đo phase chính xác và ghi lại sự chậm phase của bộ lọc tại mỗi một trạng thái của nó tại các tần số sau đây
40, 120, 400, 480, 700, 780, 1000, 1080, 1300, 1380, 1600, 1680, 1900, 1980, 2100, 2180, 2300, 2380, 2700, 2780, 3000, 3080, 4000, 4080, 5000, 5080Hz
Độ trễ phase β/ω có thể được xem xét bên cạnh các tần số này và giá trị tương ứng với độ trễ nhóm ($\Delta\beta / \Delta\omega$) có thể được tính theo các tần số 80, 440, 740, 1040, 1340, 1640, 1940, 2140, 2340, 2740, 3040, 4040, 5040Hz
Hãy vẽ sự chậm phase, độ trễ phase và độ trễ nhóm ứng với mỗi một trạng thái bộ lọc khi dùng các thang tần số được chuẩn hoá tuyến tính rồi tính các độ trễ tần số zero

e) Quan sát đáp ứng truyền dẫn

- Cho lối vào bộ lọc một xung điện thế dạng sóng hình vuông, hãy quan sát thời gian cần thiết để cho điện thế lối ra đạt giá trị ổn định sau mỗi một bước của sóng hình vuông
- Với mỗi một bộ lọc, hãy quan sát độ trễ thời gian giữa tín hiệu lối vào và lối ra khi lối ra đạt được 50% giá trị cuối cùng của nó. Chúng có thể được coi như độ trễ hiệu dụng của bộ lọc theo đáp ứng lối vào

f) Mô phỏng

Cho mạch lọc với sơ đồ theo hình vẽ dưới đây :



- Hãy sử dụng phần mềm PSPICE (hay là ELECTRONIC WORKBENCH) để đưa ra đáp ứng tần số của bộ lọc
- Sử dụng các phần mềm MatLab hay Mathematica để tính đáp ứng biên độ và đáp ứng phase của bộ lọc
- Thay thế 1 vài giá trị của một vài phần tử trong bộ lọc trên đây (theo yêu cầu của giáo viên) rồi mô phỏng và thu nhận lại kết quả

2. Thiết kế bộ lọc IIR dùng mẫu Analog.

Một lớp quan trọng thiết kế bộ lọc số IIR là dựa trên biến đổi bộ lọc tương tự cổ điển Butterworth, Chebyshev I, Chebyshev II, và Cauer sang dạng số tương ứng. Phương pháp cổ điển thiết kế các bộ lọc số IIR được thực hiện vì những lý do sau đây:

- Ưu điểm của thiết kế bộ lọc IIR tương tự. Do đó, rất thuận tiện để sử dụng thủ tục thiết kế đã có để phát triển bộ lọc tương tự
- Có rất nhiều phương pháp thiết kế bộ lọc tương tự hữu ích có công thức thiết kế dạng đơn giản, phương pháp thiết kế mạch lọc số dựa trên công thức thiết kế tương tự chuẩn hơn là đơn giản thực hiện nó, sử dụng cả biến đổi tương tự hay bất kỳ dạng nào.

- c) Phương pháp xấp xỉ tiêu chuẩn của bộ lọc tương tự không cho ta công thức thiết kế đơn giản khi các phương pháp này được áp dụng trực tiếp để thiết kế bộ lọc FIR.

Bốn hàm trong MATLAB™ SIGNAL PROCESSING TOOLBOX để thiết kế bộ lọc IIR thông thấp nói trên là :

[b,a]=butter(N,Wn)
[b,a]=cheby1(N,Rp,Wn)
[b,a]=cheby2(N,Rs,Wn)
[b,a]=ellip(N,Rp,Rs,Wn)

Có 4 phương pháp thiết kế bộ lọc thông thấp bậc N có tần số cắt W_n , W_n nằm trong khoảng 0 và 1, với 1.0 là một nửa tần số lấy mẫu (tần số Nyquist). Với cheby1 và ellip, biên R_p là decibel của gợn (ripple) trong dải thông, với cheby2 và ellip dải chặn được đặt là R_s decibel trở xuống từ giá trị đỉnh của dải thông.

Các hàm sau cho ta hệ số bộ lọc có chiều dài N+1 hàng vector b và a. Các hệ số bộ lọc là lũy thừa của z:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n-1)z^{-n}}{a(1) + a(2)z^{-1} + \dots + a(n-1)z^{-n}} \quad (12)$$

Các giá trị nhỏ hơn của gợn R_p thông dải và các giá trị lớn hơn R_s chặn dải đều dẫn đến độ quá độ lớn hơn (đặc trưng cắt ngắn hơn).

Nếu W_m là vector 2 thành phần, $W_n=[w1,w2]$, thiết kế 4 hàm bộ lọc thông dải bậc 2N với dải $w1 < \omega < w2$.

Bộ lọc thông cao được thiết kế bằng cách thêm chuỗi “high” vào sau câu lệnh:

[b,a]=ellip(N,Rp,Rs,Wn,'high')

Bộ lọc chặn dải được thiết kế bằng cách thêm chuỗi “stop” vào cuối câu lệnh:

[b,a]=butter(N,Rp,Rs,Wn,'stop')

là bộ lọc chặn dải bậc 2N nếu $W_n=[w1,w2]$. Dải chặn là $w1 < \omega < w2$.

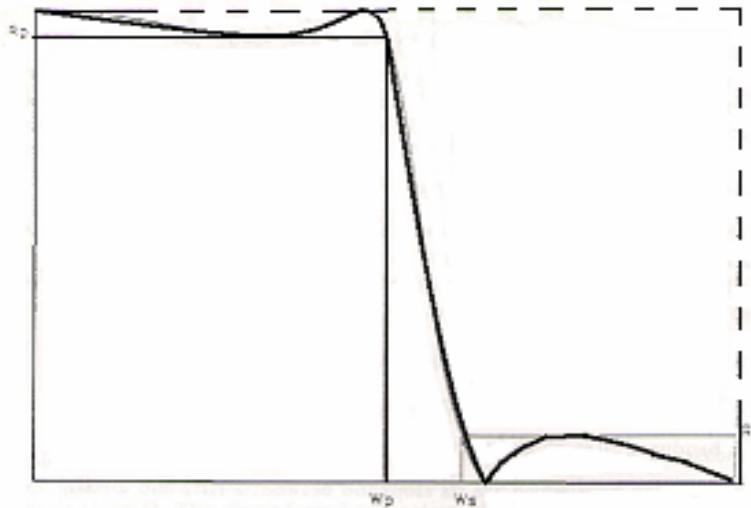


Figure 8.1 Traditional Filter Specifications.

Bậc bộ lọc nhỏ nhất yêu cầu phải đạt được các tính chất thực hiện đặc biệt, được định nghĩa trong hình 8.1, có thể được tính toán sử dụng một hàm chọn bậc bộ lọc:

$[N, Wn] = \text{buttord}(Wp, Ws, Rp, Rs)$
 $[N, Wn] = \text{cheb1ord}(Wp, Ws, Rp, Rs)$
 $[N, Wn] = \text{cheb2ord}(Wp, Ws, Rp, Rs)$
 $[N, Wn] = \text{ellipord}(Wp, Ws, Rp, Rs)$

Các hàm này trả lại N và Wn sử dụng trong butter, cheby1, cheby2, hay ellip. Các hàm chọn bậc này cũng được dùng để thiết kế bộ lọc chặn dải, thông dải và thông cao. Với bộ lọc thông cao, $Wp > Ws$, Với bộ lọc thông dải và chặn dải, đặt Wp và Ws là vector 2 thành phần xác định tần số ở cả 2 lề của bộ lọc, lề đầu trước lề thứ 2. Wn, với bộ lọc thông dải, được trả lại là vector hàng có 2 thành phần để đưa vào hàm thiết kế bộ lọc.

Ex1: Chọn bậc bộ lọc Butterworth, $Rp = 3\text{dB}$, Kết quả là lồi ra $Wn = Wp$. Dữ liệu với tần số lấy mẫu 1000 Hz, bộ lọc thông thấp có độ suy giảm nhỏ hơn 3dB ở 100 Hz và giảm 15 dB với 150 Hz:

$Wp = 100/500;$
 $Ws = 150/500;$
 $Rp = 3;$
 $Rs = 15;$
 $[N, Wn] = \text{buttord}(Wp, Ws, Rp, Rs)$
 $[b, a] = \text{butter}(n, Wn);$

Ex2: Thiết kế bộ lọc thông dải có dải thông từ 100 Hz đến 200 Hz, có độ suy giảm nhỏ hơn 3 dB ở dải thông và dưới 30 dB ở 50 Hz ở cả 2 phía, giả sử tần số lấy mẫu là 1000 Hz:

```
Wp=[100 200]/500;  
Ws=[50 250]/500;  
Rp=3;  
Rs=30;  
[N,Wn]=buttord(Wp,Ws,Rp,Rs)  
[b,a]=butter(n,Wn);
```

Ex3: Chọn bậc bộ lọc Chebyshev I, thủ tục cũng giống như trên Dữ liệu với tần số lấy mẫu 1000 Hz, thiết kế bộ lọc thông thấp có suy giảm nhỏ hơn 3dB ở 100 Hz và giảm 15 dB với 150 Hz:

```
Wp=100/500;  
Ws=150/500;  
Rp=3;  
Rs=15;  
[N,Wn]=cheb1ord(Wp,Ws,Rp,Rs)  
[b,a]=cheby1(n,Rp,Wn);
```

Ex4: Thiết kế bộ lọc thông dải Chebyshev loại I có dải thông từ 100 Hz đến 200 Hz, suy giảm nhỏ hơn 3 dB ở dải thông và dưới 30 dB ở 50 Hz ở cả 2 phía, giả sử tần số lấy mẫu là 1000 Hz:

```
Wp=[100 200]/500;  
Ws=[50 250]/500;  
Rp=3;  
Rs=30;  
[N,Wn]=cheb1ord(Wp,Ws,Rp,Rs)  
[b,a]=cheby1(n,Rp,Wn);
```

Ex5: Chọn bậc bộ lọc Chebyshev loại II, thủ tục cũng giống như trên Dữ liệu với tần số lấy mẫu 1000 Hz, thiết kế bộ lọc thông thấp có suy giảm nhỏ hơn 3dB ở 100 Hz và giảm 15 dB với 150 Hz:

```
Wp=100/500;  
Ws=150/500;  
Rp=3;  
Rs=15;  
[N,Wn]=cheb2ord(Wp,Ws,Rp,Rs)  
[b,a]=cheby2(n,Rs,Wn);
```

Ex6: Thiết kế bộ lọc thông dải Chebyshev loại I có dải thông từ 100 Hz đến 200 Hz, suy giảm nhỏ hơn 3 dB ở dải thông và dưới 30 dB ở 50 Hz ở cả 2 phía, giả sử tần số lấy mẫu là 1000 Hz:

$$\begin{aligned} W_p &= [100 \ 200]/500; \\ W_s &= [50 \ 250]/500; \\ R_p &= 3; \\ R_s &= 30; \\ [N, W_n] &= \text{cheb2ord}(W_p, W_s, R_p, R_s) \\ [b, a] &= \text{cheby2}(n, R_s, W_n); \end{aligned}$$

Ex7: Chọn bậc bộ lọc elliptic, thủ tục cũng giống như trên. Dữ liệu với tần số lấy mẫu 1000 Hz, thiết kế bộ lọc thông thấp có suy giảm nhỏ hơn 3dB ở 100 Hz và giảm 15 dB với 150 Hz:

$$\begin{aligned} W_p &= 100/500; \\ W_s &= 150/500; \\ R_p &= 3; \\ R_s &= 15; \\ [N, W_n] &= \text{ellipord}(W_p, W_s, R_p, R_s) \\ [b, a] &= \text{ellip}(n, R_p, R_s, W_n); \end{aligned}$$

Ex8: Thiết kế bộ lọc thông dải elliptic có dải thông từ 100 Hz đến 200 Hz, suy giảm nhỏ hơn 3 dB ở dải thông và dưới 30 dB ở 50 Hz ở cả 2 phía, giả sử tần số lấy mẫu là 1000 Hz:

$$\begin{aligned} W_p &= [100 \ 200]/500; \\ W_s &= [50 \ 250]/500; \\ R_p &= 3; \\ R_s &= 30; \\ [N, W_n] &= \text{ellipord}(W_p, W_s, R_p, R_s) \\ [b, a] &= \text{ellip}(n, R_p, R_s, W_n); \end{aligned}$$

Thực hiện bộ lọc IIR

Bộ lọc đáp ứng đơn xung vô hạn (IIR) được sử dụng rộng rãi trong các ứng dụng xử lý số tín hiệu. Hàm truyền của bộ lọc IIR được cho bởi :

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} = \frac{Y(z)}{X(z)}$$

Hình dưới đây cho ta giả đồ khối của bộ lọc IIR (kiểu 2) bậc-N Direction-form. Trong miền thời gian n một bộ lọc IIR bậc N được biểu diễn bởi hai biểu thức khác nhau như sau :

Tại khoảng thời gian thứ n

- $x(n)$ là mẫu lối vào hiện hành
- $y(n)$ là mẫu lối ra hiện hành

$$\begin{aligned} \rightarrow d(n) &= x(n) - d(n-1)a_1 - \dots - d(n-N+1)a_{N-1} \\ \rightarrow y(n) &= d(n)b_0 + d(n-1)b_1 + \dots + d(n-N+1)b_{N-1} \end{aligned}$$

Hai biểu thức trên dễ dàng được sử dụng cho TMS320C50 khi dùng các lệnh nhân-accumulator (MAC, MACD, MADS, MADD). Chú ý rằng biểu thức thứ hai cũng đòi hỏi các lệnh di chuyển dữ liệu để nâng cấp (update) trạng thái xâu biến đổi được $d(n)$.

Để thực hiện bộ lọc IIR trình bày ở phương trình (9) trực tiếp. Có 2 cách thực hiện bộ lọc IIR trực tiếp là : Direct-Form I và Direct-Form II. Thực hiện Direct-Form I phương trình sai phân bậc N được chỉ ra trên hình 8.2. Cấu trúc mạng được chỉ ra trên hình 8.2 có phương trình sai phân như sau:

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k] \quad (13)$$

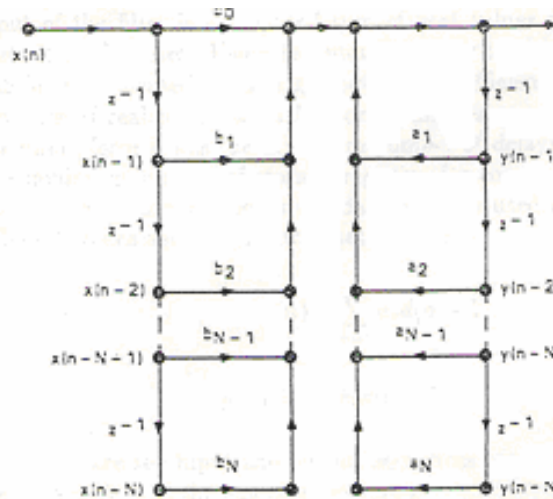
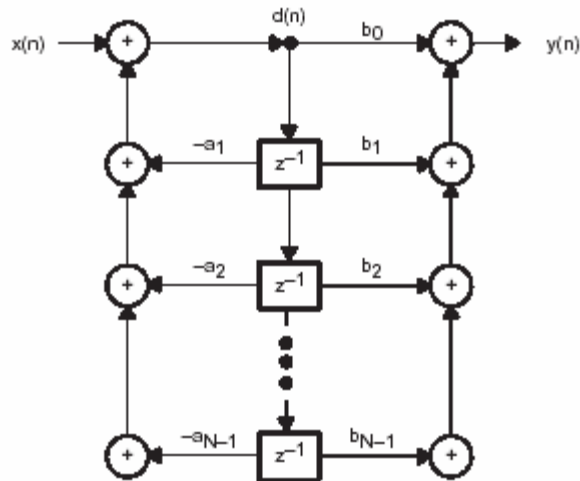


Figure 8.2 Direct-Form I IIR Filter.

với $x[n]$ và $y[n]$ là lối vào và lối ra tương ứng. Phương trình này cho thấy lối ra của bộ lọc là tổng các trọng số của lối vào và lối ra của bộ lọc. Sử dụng các kỹ thuật tương tự với bộ lọc FIR, điều này có thể thực hiện trực tiếp và có hiệu quả trên TMS320C50,



Nth-Order, Direct-Form, Type II, IIR Filter

Thực hiện Direct-Form II của phương trình sai phân bậc N được cho trên hình 8.3. Vì dạng Direct-Form II có trễ nhỏ nhất (các nhánh có nhãn z^{-1}), nó cần ít thanh ghi lưu trữ khi tính toán. Cấu trúc này rất thuận lợi với bộ nhớ dữ liệu nhỏ khi thực hiện bộ lọc IIR. Phương trình sai phân cho mạng này là:

$$d(n) = x(n) + \sum_{k=1}^N a_k d(n-k) \quad (14)$$

$$y(n) = \sum_{k=1}^N b_k d(n-k) \quad (15)$$

với $x(n)$ và $y(n)$ tương ứng là lối vào và lối ra; $d(n)$ là biến trung gian. Khi $N=2$, phương trình trở thành:

$$d(n) = x(n) + a_1 d(n-1) + a_2 d(n-2) \quad (16)$$

$$y(n) = b_0 d(n) + b_1 d(n-1) + b_2 d(n-2) \quad (17)$$

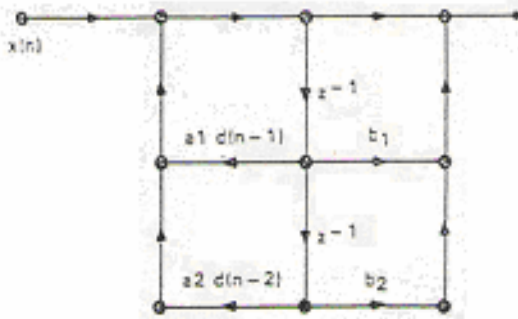
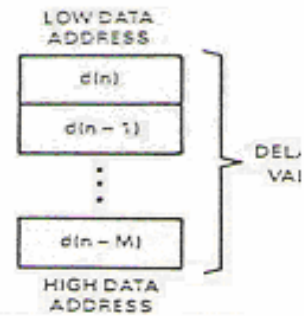


Figure 8.4 Second-Order Direct-Form II IIR

Figure 8.5



Cấu trúc mạng của nó được cho trên hình 8.4. Chú ý rằng sử dụng Direct-Form II này, $d(n)$ là tần số cần lưu trong bộ lưu giá trị trễ. Sắp xếp lưu giá trị delay-node cho Direct-Form II IIR được trình bày trên hình 8.5.

B. PHẦN THỰC NGHIỆM

1. Thủ tục

- 1) Tắt nguồn, nối bản mạch với cổng PC RS232-C trên máy tính. Nối 9 Vac Adapter với bản mạch và cắm ra bên ngoài. Bật máy tính và thay đổi đường dẫn có chứa phần mềm DSK.
- 2) Chạy chương trình DSKL(DSK5L) và kiểm tra là chương trình DSK_COMM được tải chính xác. Bây giờ thực hiện AUTOTEST (lệnh “A”) và xác minh là đã kiểm tra vòng lặp. Nếu DSK được kết nối và chức năng chính xác thì bạn có thể bắt đầu thực hiện. Dừng chương trình DSKL.
- 3) Thiết kế bộ lọc IIR thông thấp. Có các yêu cầu như sau:
 - passband 0-3 KHz
 - passband ripple 0.5db
 - stopband attenuation 45db
 - sampling frequency 15 KHz
 - transition width 450 Hz
 - Xác định các hệ số cho bộ lọc sử dụng Matlab.

- 4) Lượng tử hóa các hệ số và liệt kê các hệ số tỉ lệ sẽ được sử dụng để chống tràn. Liệt kê các hệ số tỉ lệ này và tất cả các hệ số, hệ số lượng tử. Có thể phụ thuộc vào hệ số tỉ lệ thiết kế.
- 5) Trước khi thực hiện bộ lọc, so sánh đáp ứng tần số của hệ số tính toán với đáp ứng của hệ số lượng tử, và chứng minh rằng lượng tử không làm giảm đáp ứng tần số của bộ lọc.
- 6) Tạo ra cấu trúc thực hiện và biểu đồ miêu tả bộ lọc giống như bộ lọc thực hiện dùng DSP.
- 7) Tạo ra mã nguồn để thực hiện bộ lọc. Đưa 1 Vrms vào DSK và chạy chương trình. Quan sát lỗi ra của DSK trên dao động ký và xác nhận rằng đó là xung sin khi tần số lỗi vào nằm trong phần thông dải của bộ lọc. Đo biên độ lỗi ra khi tần số thay đổi trong khoảng 50 Hz đến 3500 Hz. Nếu lỗi ra là sóng sin, dùng máy đo điện áp xoay chiều để đo biên độ. So sánh phổ đo đạc của bộ lọc với giá trị mong muốn.

2. Yêu cầu tiến hành thực tập

Dưới đây là các chương trình ví dụ về bộ lọc IIR. Chương trình ví dụ (*) dưới đây dùng cho bộ lọc IIR thông thấp bậc $N=4$, tần số cắt khoảng 3.4KHz, dùng các lệnh vòng lặp đơn (RPT) và nhân accumulator (MAC, MACD) Do bản chất có thể hồi quy được của một bộ lọc IIR nên việc lượng tử hoá các hệ số của bộ lọc có thể gây ra sự thay đổi lớn so với đáp ứng tần số mong muốn. Để tránh được điều này, hàm chuyển đổi bộ lọc mong muốn có thể được bẻ thành các phần có bậc thấp hơn, liên tiếp nhau.

Các ví dụ (**) và chương trình ví dụ (***) dưới đây cho ta thấy một ứng dụng của bộ lọc IIR bậc-2 N tầng. Các hệ số bộ lọc và các biến số trạng thái được cất giữ trong bộ nhớ dữ liệu. Lưu ý việc sử dụng các chỉ dẫn LTD và MPYA để thực hiện các hoạt động di chuyển dữ liệu và đa tích lũy.

Hãy tận dụng chương trình này để tiến hành các thí nghiệm sau

1. Trong các thí nghiệm sau, tần số lấy mẫu của mạch TMS320C50 cần đặt là 22,321 KHz và tần số cắt của bộ lọc tương tự thông thấp của mạch là 10KHz. Các bộ lọc IIR dưới đây được thiết kế và thực hiện, có độ suy giảm 3 dB ở tần số thông dải và 30 dB ở tần số chặn dải.
2. Bộ lọc thông thấp có tần số cắt 2 KHz;

3. Bộ lọc thông cao có tần số cắt 3.4 KHz;
4. Bộ lọc thông dải có dải thông nằm trong khoảng 300 Hz và 3.4 KHz.
5. Bộ lọc chặn dải có dải chặn nằm giữa 2 KHz và 3 KHz.

Thí nghiệm 2 và 4 là không bắt buộc. Các bộ lọc Butterworth, Chebyshev I,II và elliptic có thể sử dụng. Hiệu suất của các bộ lọc IIR có thể được kiểm tra so sánh với hiệu suất của bộ lọc FIR , ghi lại và báo cáo những gì bạn thấy.

Bài báo cáo phải trình bày thủ tục thiết kế và kết quả thực hiện.

Ví dụ (*). Nth-Order IIR Filter Using RPT and MACD Instructions
 .title "Nth Order IIR Type II Filter"

```

    .mmregs
N      .set 4
N-1   .set 3

      .ds 0f00h
TA      .word 20      ;
RA      .word 20      ; Freq Sampling 8KHz
TAp     .word 1       ; not used
RAp     .word 1       ; not used
TB      .word 31      ; <10>
RB      .word 31      ;
AIC_CTR .word 09h     ;

YN      .word 0
YN-1   .word 0
YNLAST .word 0
POUBY  .word 0

MA      .word 0      ;
XN      .word 0,0,0,0 ; <20>
XNLAST .word 0
POUBX  .word 0      ;

      .ps 0980h
COEFFA .word -5249,19621,-30929,24108
COEFFB .word 186,-107,296,-107,186

```

```

;*****
;
; Set up the ISR vector *
;*****
,
        .ps    080ah        ;
rint:    B    RECEIVE        ;0A    ; Serial port receive interrupt
RINT.
xint:    B    TRANSMIT        ;0C    ; Serial port transmit interrupt
XINT.

```

```

;*****
;
; TMS32C05X INITIALIZATION *
;*****
,

```

```

        .ps    0a00h

        .entry
        SETC    INTM        ; Disable interrupts
        LDP     #0          ; Set data page pointer
        OPL     #0834h,PMST ; OR the status register
        LACC    #0
        SAMM    CWSR        ; Set software wait state to 0
        SAMM    PDWSR        ;
        SETC    SXM
        SPLK    #022h,IMR    ; Using XINT syn TX & RX
        CALL    AICINIT      ; initialize AIC and enable interrupts
        SPLK    #12h,IMR    ; Select serial port receive interrupt
        CLRC    OVM
        SPM 0
        CLRC    INTM        ; Enable interrupts

forever IDLE
        B    forever

;----- end of main program -----;

```

```

;----- RECIEVER ISR -----

```

```

RECEIVE:
        LDP     #XN
        CLRC    INTM
        LAMM    DRR        ; get input in ACC

```

SACL

XN

```

; start IIR
;*****
;* This routine implements an N-th order type II IIR filter.      *
;*   d(n) = x(n) - d(n-1)a1 - d(n-2)a2 -...- d(n-N+1)aN-1      *
;*   y(n) = d(n)b0 + (dn-1)b1 +...+ d(n-N+1)bN-1              *
;* Memory Requirement: State variables (low to high data memory): *
;*   d(n) d(n-1) ... d(n-N+1)                                  *
;* Coefficient (low to high program memory):                    *
;*   -a(N-1) -a(N-2) ... -a(1)  b(N-1) b(N-2) ... b(1) b(0)    *
;* Entry Conditions:                                           *
;*   AR0 -> Input                                              *
;*   AR1 -> d(n-N+1)                                          *
;*   AR2 -> Output                                            *
;*   COEFFA -> -a(N-1)                                        *
;*   COEFFB -> b(N-1)                                        *
;*   ARP = AR0                                               *
;*****
LAR      AR0,#XN
LAR      AR1,#XNLAST
LAR      AR2,#YNLAST

```

IIR_N:

```

ZPR                ; Clear P register
LACC                ; Get Q15 input
RPT                ; For i=1,i<=N-1,++i
MAC                ; Acc+=-a(N-i))*d(n-N+i)
APAC                ; Final accumulation
SACH                ; Save d(n)
ADRK               *,1
LMM                ; AR1 -> d(n-N+1)
LMM                ; Acc -> a(N-1)
ADD                ; Acc -> b(N-1)
SMM                ; BMAR -> b(N-1)
RPTZ               #4
MACD               COEFFB,*-
LTA                ; Final accumulation
SACH                ; ??? Save Yn
LACC               YNLAST
AND                #0fffch ; and ls 2 bits of top 16 bits, for DAC.
SMM                DXR
RETE

```

;----- Transmit ISR -----

TRANSMIT:
RETE

```
.*****  
,  
;* DESCRIPTION: This routine initializes the TLC32040 for      *  
;*                your sample rate and command word          *  
.******  
,  
;* aic initialization data
```

AICINIT:

```
    SPLK      #01h,PRD  
    SPLK      #20h,TCR      ; make a 10 mhz clock signal  
    MAR      *,AR0  
    LACC      #0008h      ; Non continuous mode  
    SACL      SPC      ; FSX as input  
    LACC      #00c8h      ; 16 bit words  
    SACL      SPC  
    LACC      #080h      ; Pulse AIC reset by setting it low  
    SACH      DXR  
    SACL      GREG  
    LAR      AR0,#0FFFFh  
    RPT      #10000      ; and taking it high after 10000 cycles  
    LACC      *,0,AR0      ; (.5ms at 50ns)  
    SACH      GREG  
    LDP      #TA      ;  
    SETC      SXM      ;  
    LACC      TA,9      ; Initialized TA and RA register  
    ADD      RA,2      ;  
    CALL     AIC_2ND      ;  
;-----  
    LDP      #TB  
    LACC      TB,9      ; Initialized TB and RB register  
    ADD      RB,2      ;  
    ADD      #02h      ;  
    CALL     AIC_2ND      ;  
;-----  
    LDP      #AIC_CTR  
    LACC      AIC_CTR,2      ; Initialized control register  
    ADD      #03h      ;  
;LACC      #11100000b  
    CALL     AIC_2ND      ;
```

```

        RET                ;
AIC_2ND:
    LDP                #0
    SACH                DXR                ;
    CLRC                INTM
    IDLE
    ADD                #6h,15                ; 0000 0000 0000 0011 [X..X]16 b
    SACH                DXR                ;
    IDLE
    SACL                DXR                ;
    IDLE
    LACL                #0                ;
    SACL                DXR                ; make sure the word got sent
    IDLE
    SETC                INTM
    RET                ;
.end

```

Ví dụ (**) Nth-Order IIR Filter Using RPT and MACD Instructions

.title "Nth Order IIR Type II Filter"

.mmregs

```

*****
* This routine implements an N-th order type II IIR filter.                *
*  $d(n) = x(n) - d(n-1)a_1 - d(n-2)a_2 - \dots - d(n-N+1)a_{N-1}$                 *
*  $y(n) = d(n)b_0 + d(n-1)b_1 + \dots + d(n-N+1)b_{N-1}$                 *
*                                                                           *
* Memory Requirement:                *
* State variables (low to high data memory):                *
*  $d(n) d(n-1) \dots d(n-N+1)$                 *
*                                                                           *
* Coefficient (low to high program memory):                *
*  $-a(N-1) -a(N-2) \dots -a(1) b(N-1) b(N-2) \dots b(1) b(0)$                 *
*                                                                           *
* Entry Conditions:                *
*   AR0 -> Input                *
*   AR1 ->  $d(n-N+1)$                 *
*   AR2 -> Output                *
*   COEFFA ->  $-a(N-1)$                 *
*   COEFFB ->  $b(N-1)$                 *
*   ARP = AR0                *
*

```



```

* AR3 -> input sample (Q15 number) *
* AR4 -> output sample (Q15 number) *
* DP = 0, PM = 0, ARP = 3 *
*****

```

```

BIQUAD:                                ;Setup variables
    ZPR                                ;Clear P register
    LACC *,15,AR1                       ;Get Q15 input
    SPLK #2,INDX                         ;Setup index register
    SPLK #N-1,BRCR                       ;Setup count

    RPTB ELOOP-1                         ;Begin computation;
    RPTB ELOOP-1                         ;Repeat for N biquads
LOOP:
    LT *-,AR2                            ;T = d(n-2)
    MPYA *+,AR1                          ;Acc = x(n), P = -d(n-2)a2
    LTA *-,AR2                            ;Acc += -d(n-2)a2, T = d(n-1)
    MPY *+                                ;P = -d(n-1)a1
    LTA *+,AR1                            ;Acc += -d(n-1)a1, T = b2
    SACH *0+,1                           ;Save d(n)
    MPY *-                                ;P = d(n-2)b2
    LACL #0                               ;Acc = 0
    LTD *-,AR2                            ;T = d(n-1), d(n-2) = d(n-1)
    MPY *+,AR1                            ;Acc += d(n-2)b2, P = d(n-1)b1
    LTD *-,AR2                            ;T = d(n), d(n-1) = d(n)
    MPY *+,AR1                            ;Acc += d(n-1)b1, P = d(n)b0

ELOOP:
    LTA *,AR4                             ;Final accumulation
    SACH *,1                              ;Save output in Q15 format

```

C. TÀI LIỆU THAM KHẢO

- [1]. TMS320C50 Fixed-Point DSP Assembly Language Tools-User's Guide", Texas Instruments Incorporated, 1990.
- [2] "Second- Generation TMS320- User's Guide", Texas Instruments, 1989.
- [3] The ART Electronic Designed . MIT University , Boston 1996.
- [4] Digital Signal Processing , Sanjit K. Mitra , University California . McGrawHill 2006

MỤC LỤC

A. PHẦN LÝ THUYẾT (120)

3. Các đặc tính đáp ứng bộ lọc thông thấp (120)

4. Thiết kế bộ lọc IIR dùng mẫu Analog. (123)

B. PHẦN THỰC NGHIỆM (130)

1. Thủ tục (130)

2. Yêu cầu tiến hành thực tập (131)

C. TÀI LIỆU THAM KHẢO (139)