MỤC LỤC

LỜI MỞ ĐẦU

Đđánh dấu cho cuộc cách mạng khoa học công nghệ hiện nay đó là sự ra đời và phát triển ồ ạt của các máy tính cũng nh- các ph- ơng tiện xử lý thông tin. Đặc biệt là các hệ thống xử lý song song với tốc độ ngày càng cao. Cùng với sự phát triển các công cụ tín hiệu số đòi hỏi sự phát triển đồng bộ các ph- ơng pháp xử lý số hiện đại. Một trong những công cụ chính của kỹ thuật xử lý số đó là bộ lọc.

Bộ lọc là một hệ thống có thể ứng dụng rất nhiều trong lĩnh vực cuộc sống. Khi công nghệ ngày càng phát triển thì việc lọc nhiễu để đạt đ- ợc những tín hiệu tốt hơn ngày càng trở nên quan trọng.

Về lịch sử phát triển, bộ lọc đ- ợc nghiên cứu nhiều nhất trong xử lý tín hiệu số. Và đã dành đ- ợc sự quan tâm, đầu t- nghiên cứu của các nhà khoa học, các trung tâm nghiên cứu lớn trên thế giới. Hiện nay, bộ lọc liên tục phát triển tạo ra các kỹ thuật quan trọng ảnh h- ởng trực tiếp đến lĩnh vực điện tử, thông tin liên lạc, phát thanh truyền hình, các ngành công nghệ khác …

Trong thông tin liên lạc, tín hiệu âm thanh đ- ợc truyền đi ở những khoảng cách rất xa, nên không tránh khỏi bị tác động nhiễu của môi tr- ờng, đ- ờng truyền, tần số, hay trong chính hệ thống của nó ... Nh- ng khi qua bộ lọc nhiễu, âm thanh sẽ trở nên rõ ràng và chính xác hơn. Trong các thiết bị điện tử th- ờng gặp nh- loa đài, máy phát, máy thu … ngày càng có chất l- ợng âm thanh tốt hơn là do bộ lọc ngày càng đ- ợc tối - u hơn.

Vì những ứng dụng quan trọng trong thực tế nh- vậy, nên vấn đề đặt ra là làm thế nào để thu đ- ợc âm thanh có chất l- ợng tốt hơn. Đó cũng chính là mục tiêu mà đồ án của em h- ớng tới. Trong đề tài này em nghiên cứu một số ph- ơng pháp lọc, và mô phỏng việc lọc âm thanh qua phần mền Matlap.

Với mục tiêu xác định nh- trên, đồ án đ- ợc chia ra làm 3 phần với nội dung cơ bản nh- sau:

Ch- ơng 1: Lý thuyết chung về xử lý tín hiệu số.

Ch- ơng 2: Ước l- ợng tuyến tính và những bộ lọc tuyến tính tối - u.

Ch- ơng 3: Mô phỏng

Trong quá trình làm đồ án em đã nhận đ- ợc sự giúp đỡ rất nhiệt tình của các thầy, các cô và các bạn trong lớp. Đặc biệt là của thạc sỹ Nguyễn Văn D- ơng ng- ời đã trực tiếp h- ớng dẫn em hoàn thành đồ án này.

Em xin chân thành cảm ơn thạc sỹ Nguyễn Văn D- ơng, các thầy cô giáo trong tổ bộ môn điện tử viên thông và các bạn trong lớp ĐT901 đã giúp tôi hoàn thành tốt nhiệm vụ đồ án nhà tr- ờng và tổ bộ môn giao cho.

Hải Phòng, tháng 8 năm 2009

Sinh viên thực hiện

Trần Thu Huyền

CHƯƠNG 1:

LÝ THUYẾT CHUNG VỀ XỬ LÝ TÍN HIỆU SỐ

1.1. TÍN HIỆU VÀ HỆ THỐNG RỜI RẠC THEO THỜI GIAN

Trong hầu hết các lĩnh vực có liên quan đến xử lý tin tức hoặc thông tin đều bắt đầu với việc biểu diễn tín hiệu nh- một dạng mẫu thay đổi liên tục. Từ các mẫu tín hiệu, để thuận tiện, ng- ời ta dùng các hàm toán học để biểu diễn chúng, nh- các hàm biến đổi theo thời gian t. Ở đây chúng ta sẽ dùng dạng biểu diễn $x_a(t)$ để biểu diễn các dạng sóng thời gian thay đổi liên tục (tín hiệu analog). Ngoài ra tín hiệu còn có thể biểu diễn nh- một dãy rời rạc các giá trị và ta dùng dạng biểu diễn x(n) để biểu thị. Nếu tín hiệu đ- ợc lấy mẫu từ tín hiệu t- ơng tự với chu kỳ lấy mẫu T, khi đó chúng ta có dạng biểu diễn $x_a(nT)$.

Trong các hệ thống xử lý số tín hiệu, chúng ta th- ờng dùng đến các dãy đặc biệt, nh- :

Mẫu đơn vị hoặc dãy xung đơn vị đ- ợc định nghĩa:

$$\delta(n) = \begin{cases} 1 \text{ với } & n = 0 \\ 0 \text{ với n còn lại} \end{cases} \tag{1.1.1}$$

Dãy nhảy bậc đơn vị

$$u(n) = \begin{cases} 1 & \text{với } n \geq 0 \\ 0 & \text{với các n còn lại} \end{cases} \tag{1.1.2}$$

Dãy hàm mũ

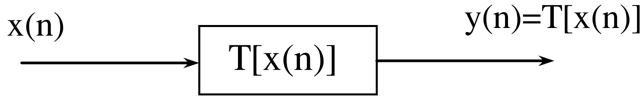
$$x(n) = a^n \tag{1.1.3}$$

Nếu a là số phức nh-

$$a = r.e^{j\omega_0 n} = r^n(\cos\omega_0 n + j\sin\omega_0 n) \tag{1.1.4}$$

Nếu $r \geq 1, \omega_0 \neq 0$, thì x(n) có dạng sin phức; nếu ω_0=0, x(n) là thực; và r<1, $\omega_0 \neq 0$, x(n) là một dãy thay đổi, suy giảm theo luật hàm mũ. Dãy kiểu này xuất hiện đặc biệt trong biểu diễn các hệ thống tuyến tính và trong mô hình dạng sóng tiếng nói.

Trong xử lý tín hiệu, chúng ta phải chuyển đổi tín hiệu về dạng mẫu nh- ta mong muốn. Nên ta phải quan tâm đến các hệ thống rời rạc, hoặc t- ơng đ- ơng với sự chuyển đổi của một dãy tín hiệu vào để đ- ợc một dãy tín hiệu ra. Ta miêu tả sự chuyển đổi này bằng một khối nh- ở hình 1.1.



Hình 1.1. Mô phỏng hệ thống

Những hệ thống nh- trên hoàn toàn có thể đ- ợc xác định bằng đáp ứng xung của nó đối với mẫu xung đơn vị đ- a vào. Đối với những hệ thống này, đầu ra có thể đ- ợc tính khi ta đ- a vào dãy x(n) và đáp ứng xung đơn vị h(n), dùng tổng chập để tính

$$y(n)=\sum_{k=-\infty}^{\infty}x(k)h(n-k)=x(n)*h(n) \qquad (1.1.5a)$$

Dấu * ở đây dùng cho tổng chập. T- ơng tự ta cũng có

$$y(n)=\sum_{k=-\infty}^{\infty}h(k)x(n-k)=h(n)*x(n) \qquad (1.1.5b)$$

1.2. BIỂU DIỄN SỰ BIẾN ĐỔI CỦA TÍN HIỆU VÀ HỆ THỐNG

Phân tích và thiết kế của các hệ thống tuyến tính sẽ rất đơn giản nếu chúng ta sử dụng trong miền Z và miền tần số cho cả hệ thống và tín hiệu, khi đó chúng ta cần thiết phải xét đến sự biểu diễn Fourier, miền Z của hệ thống và tín hiệu rời rạc theo thời gian.

1.2.1 Biến đổi sang miền Z

Sự biến đổi sang miền Z của một dãy đ- ợc định nghĩa bằng hai ph- ơng trình sau:

$$X(Z)=\sum_{n=-\infty}^{\infty}x(n)Z^{-n} \qquad (1.2.1a)$$

$$x(n)=\frac{1}{2\pi j}\oint_C X(Z)Z^{n-1}dZ \qquad (1.2.1b)$$

Từ một dãy x(n) để biến đổi sang miền Z (biến đổi thuận), ta dùng công thức (1.2.1a). Ta có thể thấy dãy X(Z) là một dãy luỹ thừa đối với biến Z^{-1}, giá trị của dãy x(n) biểu diễn bộ các hệ số trong dãy luỹ thừa. Một cách chung nhất, điều kiện đủ để biến đổi sang miền Z là dãy luỹ thừa phải hội tụ tại một giá trị giới hạn.

$$\sum_{n=-\infty}^{\infty} \left| x(n) Z^{-n} \right| < \infty \qquad (1.2.2)$$

Một bộ các giá trị cho các dãy hội tụ đ- ợc định nghĩa bằng một vùng trong mặt phẳng Z. Nói chung miền này có dạng:

$$R_1 < |Z| < R_2 \qquad (1.2.3)$$

Bảng 1.1. Các tính chất của phép biến đổi Z ng- ợc

Các tính chất	Dãy miền n	Biến đổi Z
1. Tính tuyến tính	$ax_1(n)+bx_2(n)$	$aX_1(Z)+bX_2(Z)$
2. Tính dịch chuyển theo thời gian	$x(n+n_0)$	$Z^{n_0}X(Z)$
3. Thay đổi thang tỉ lệ (nhân với dãy hàm mũ a^n)	$a^n x(n)$	$X(a^{-1}Z)$
4. Vi phân của X(Z) theo Z	$nx(n)$	$-Z\frac{dX(Z)}{dZ}$
5. Đảo trục thời gian	$X(-n)$	$X(Z^{-1})$
6. Tích chập của hai dãy	$x(n)*h(n)$	$X(Z).H(Z)$
7. Tích của hai dãy	$x(n).w(n)$	$\frac{1}{2\pi j}\oint_C X(V)W(Z/V)V^{-1}dV$

Phép biến đổi Z ng- ợc đ- ợc đ- a ra bởi tích phân đ- ờng trong ph- ơng trình (1.2.1b), trong đó C là đ- ờng cong kín bao quanh gốc toạ độ trong mặt phẳng Z, nằm trong miền hội tụ của X(Z).

1.2.2. Biến đổi Fourier

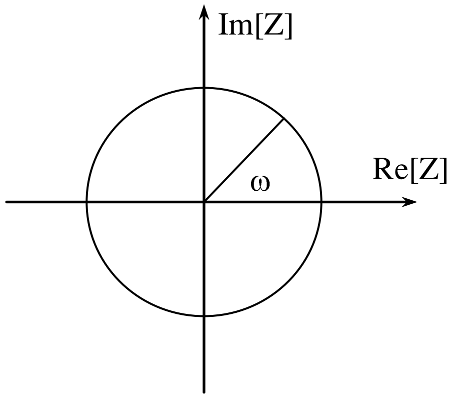
Phép biến đổi Fourier của tín hiệu rời rạc theo thời gian đ- ợc biểu diễn bằng công thức sau:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \qquad (1.2.4a)$$

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega \qquad (1.2.4b)$$

Ngoài ra biểu diễn Fourier có thể đạt đ- ợc bằng cách giới hạn phép biến đổi Z (Z – Transform) vào vòng tròn đơn vị của mặt phẳng Z, nh- thay $Z = e^{j\omega}$, nh- trong hình 1.2, biến số ω có thể biểu diễn bằng góc trong mặt phẳng Z. Điều kiện đủ để tồn tại biến đổi Fourier có thể tính bằng cách gán $|Z| = 1$ trong ph- ơng trình (1.2.2), ta có:

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \qquad (1.2.5)$$



Hình 1.2. Vòng tròn đơn vị trong mặt phẳng Z

Một đặc điểm quan trọng của biến đổi Fourier $X(e^{j\omega})$ là một hàm tuần hoàn của ω, tuần hoàn với chu kỳ là 2π, điều này có thể dễ nhận ra bằng cách thay thế ω+2π vào ph- ơng trình (1.2.4a). Một cách khác, bởi vì $X(e^{j\omega})$ đ- ợc tính bằng X(Z) trên vòng tròn đơn vị, nên chúng ta có thể thấy rằng $X(e^{j\omega})$ phải lặp lại mỗi lần khi ω quay hết một vòng quanh vòng tròn đơn vị (t- ơng ứng với một góc là 2π Radian).

Bằng cách thay Z= $e^{j\omega}$ vào mỗi công thức trong bảng (1.1), chúng ta có thể đạt đ- ợc các công thức cho biến đổi Fourier. Tất nhiên kết quả này chỉ đúng với biến đổi Fourier khi phép biến đổi đã tồn tại.

1.3. BỘ LỌC SỐ

Bộ lọc số là hệ thống tuyến tính bất biến theo thời gian. Thông số vào và ra của hệ thống quan hệ với nhau bằng tổng chập trong ph- ơng trình (1.1.5), quan hệ trong miền Z đ- ợc đ- a ra trong bảng (1.1).

$$Y(Z)=H(Z).X(Z) \qquad (1.3.1)$$

Chuyển đổi miền Z của đáp ứng xung đơn vị H(Z) đ- ợc gọi là hàm hệ thống. Biến đổi Fourier của đáp ứng xung đơn vị $H(e^{j\omega})$ là một hàm phức của ω, biểu diễn theo phần thực và phần ảo là

$H(e^{j\omega})=Hr(e^{j\omega})+jHi(e^{j\omega})$ (1.3.2)

Hoặc biểu diễn d- ới dạng góc pha:

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{j\arg|H(e^{j\omega})|} \quad (1.3.3)$$

Một hệ thống tuyến tính bất biến nhân quả là dạng có h(n)=0 với n<0. Một hệ thống ổn định là dạng với tất cả các thông số đ- a vào hữu hạn sẽ có thông số ra hữu hạn.

Điều kiện cần và đủ cho một hệ thống tuyến tính bất biến ổn định là:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \quad (1.3.4)$$

Điều kiện này giống với công thức (1.2.5). Thêm vào đó, tất cả các hệ thống tuyến tính bất biến có các thông số vào và ra nh- các bộ lọc thoả mãn ph- ơng trình sai phân có dạng:

$$y(n) - \sum_{k=1}^{N} a_k y(n-k) = \sum_{r=0}^{M} b_r x(n-r) \quad (1.3.5)$$

Chuyển đổi sang miền Z cả hai vế của ph- ơng trình ta đ- ợc:

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\sum_{r=0}^{M} b_r Z^{-r}}{1 - \sum_{k=1}^{N} a_k Z^{-k}} \quad (1.3.6)$$

So sánh hai ph- ơng trình trên, từ ph- ơng trình sai phân (1.3.3) ta có thể đạt đ- ợc H(Z) trực tiếp bằng cách đồng nhất các hệ số của phần tử vào trễ trong (1.3.5) với các luỹ thừa t- ơng ứng Z^{-1}.

Hàm hệ thống H(Z) là một hàm hữu tỉ của Z^{-1}. Nó có thể đ- ợc biểu diễn bằng dạng điểm cực và điểm không trong mặt phẳng Z. Nh- vậy H(Z) có thể viết dạng:

$$H(Z) = \frac{A\prod_{r=1}^{M} (1 - c_r Z^{-1})}{\prod_{k=1}^{N} (1 - d_k Z^{-1})} \quad (1.3.7)$$

Nh- chúng ta đã xét trong miền Z, hệ thống nhân quả sẽ có miền hội tụ dạng $|Z| < R_1$. Nếu hệ thống cũng là ổn định thì R_1 phải nhỏ hơn giá trị đơn vị, do đó miền hội tụ bao gồm là vòng tròn đơn vị. Nh- vậy trong hệ thống bất biến, nhân quả thì tất cả các điểm cực của H(Z) phải nằm trong vòng tròn đơn vị. Để thuận tiện, ta phân thành các lớp hệ thống, những lớp này bao gồm hệ thống đáp ứng xung hữu hạn (Finit duration Impulse Response_FIR), và hệ thống đáp ứng xung vô hạn (Infinit duration Impulse Response_IIR).

1.3.1. Hệ thống FIR

Nếu các hệ số a_k trong ph- ơng trình (1.3.5) bằng không, khi đó ph- ơng trình sai phân sẽ là:

$$y(n) = \sum_{r=0}^{M} b_r x(n-r) \tag{1.3.8}$$

So sánh (1.3.8) với (1.1.5b) chúng ta thấy rằng:

$$h(n) = \begin{cases} b_n & 0 \le n \le M \\ 0 & \text{với các n còn lại} \end{cases} \tag{1.3.9}$$

Hệ thống FIR có rất nhiều thuộc tính quan trọng, tr- ớc tiên chúng ta chú ý rằng H(Z) chỉ có điểm không là một đa thức của Z^{-1} và tất cả các điểm cực của H(Z) đều bằng không, tức là H(Z) chỉ có điểm không. Thêm nữa, hệ thống FIR có thể có chính xác pha tuyến tính. Nếu h(n) xác định theo công thức sau

$$h(n) = \pm h(M-n) \tag{1.3.10}$$

thì $H(e^{j\omega})$ có dạng

$$H(e^{j\omega}) = A(e^{j\omega}).e^{-j\omega(M/Z)} \tag{1.3.11}$$

$H(e^{j\omega})$ chỉ có phần thực hoặc phần ảo tuỳ thuộc vào ph- ơng trình (1.3.10) lấy dấu (+) hay dấu (-).

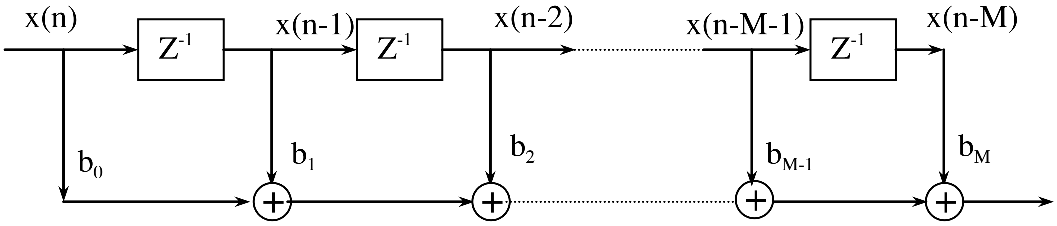
Dạng pha tuyến tính chính xác th- ờng rất hữu ích trong các ứng dụng xử lý âm thanh, khi mà xác định thứ tự thời gian là cần thiết. Các thuộc tính này của bộ lọc FIR cũng có thể đơn giản hoá vấn đề xấp xỉ, nó chỉ xét đến khi đáp ứng độ lớn cần thiết. Khoảng sai số mà đ- ợc bù để thiết kế các bộ lọc với đáp ứng xung pha tuyến tính chính xác là phần mà một khoảng thời gian tồn tại đáp ứng xung phù hợp đ- ợc yêu cầu để xấp xỉ phần nhọn bộ lọc bị cắt đi.

Dựa trên những thuộc tính chung với bộ lọc FIR pha tuyến tính, ng- ời ta đã phát triển ba ph- ơng pháp thiết kế xấp xỉ. Những ph- ơng pháp này là:

- Thiết kế cửa sổ
- Thiết kế mẫu tần số
- Thiết kế tối - u

Chỉ có ph- ơng pháp đầu tiên là ph- ơng pháp phân tích, thiết kế khối khép kín tạo bởi các ph- ơng trình có thể giải để nhận đ- ợc các hệ số bộ lọc.

Ph- ơng pháp thứ hai và ph- ơng pháp thứ ba là ph- ơng pháp tối - u hoá, nó sử dụng ph- ơng pháp lặp liên tiếp để đ- ợc thiết kế bộ lọc



Hình 1.3. Mạng số cho hệ thống FIR

Bộ lọc số th- ờng đ- ợc biểu diễn dạng biểu đồ khối, nh- hình (1.3) ta biểu diễn ph- ơng trình sai phân (1.3.8). Sơ đồ nh- vậy th- ờng đ- ợc gọi là một cấu trúc bộ lọc số. Trên sơ đồ, biểu diễn các toán tử yêu cầu tính giá trị mỗi dãy ra từ giá trị của dãy đ- a vào. Những phần tử cơ bản của sơ đồ biểu diễn ý nghĩa phép cộng, nhân các giá trị của dãy với hằng số (các hằng số trên nhánh hàm ý phép nhân), và chứa các giá trị tr- ớc của dãy vào. Vì vậy biểu đồ khối đ- a ra chỉ dẫn rõ ràng về tính phức tạp của hệ thống.

1.3.2. Hệ thống IIR

Nếu hàm hệ thống của ph- ơng trình (1.3.7) có các điểm cực cũng nh- điểm không, thì ph- ơng trình sai phân (1.3.5) có thể viết:

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{r=0}^{M} b_r x(n-r) \qquad (1.3.12)$$

Ph- ơng trình này là công thức truy hồi, nó có thể đ- ợc sử dụng để tính giá trị của dãy ra từ các giá trị tr- ớc đó của thông số ra và giá trị hiện tại, tr- ớc đó của dãy đầu vào. Nếu M<N trong ph- ơng trình (1.3.7), thì H(Z) có thể biến đổi về dạng:

$$H(Z) = \sum_{k=1}^{N} \frac{A_k}{1 - d_k Z^{-1}} \qquad (1.3.13)$$

Cho hệ thống nhân quả, ta dễ dàng biểu diễn

$$h(n) = \sum_{k=1}^{N} A_k (d_k)^n u(n) \qquad (1.3.14)$$

Ta có thể thấy rằng dãy h(n) có chiều dài vô hạn. Tuy nhiên, vì công thức truy hồi (1.3.12) th- ờng dùng để thực hiện bộ lọc IIR, nó sử dụng ít phép tính hơn là đối với bộ lọc FIR. Điều này đặc biệt đúng cho các bộ lọc lựa chọn tần số cắt nhọn.

Có nhiều ph- ơng pháp thiết kế sẵn có cho bộ lọc IIR. Những ph- ơng pháp thiết cho bộ lọc lựa chọn tần số (thông thấp, thông dải, ...) một cách chung nhất là dựa trên những biến đổi của thiết kế t- ơng tự.

- Các thiết kế Butterword
- Các thiết kế Bessel
- Các thiết kế Chebyshev
- Các thiết kế Elliptic

Tất cả những ph- ơng pháp trên dùng phép phân tích tự nhiên và đ- ợc ứng dụng rộng rãi để thiết kế các bộ lọc IIR. Thêm vào đó các ph- ơng pháp tối - u hoá IIR đã đ- ợc phát triển cho thiết kế xấp xỉ liệt kê, điều này không dễ thích nghi với một trong các ph- ơng pháp xấp xỉ trên.

Sự khác nhau chính giữa FIR và IIR là IIR không thể thiết kế để có pha tuyến tính chính xác, khi mà FIR có những thuộc tính này, còn bộ lọc IIR hiệu quả hơn trong thực hiện lọc cắt nhọn hơn là FIR.

Mạng bao hàm ph- ơng trình (1.3.12) đ- ợc biểu diễn trong hình 1.4a cho tr- ờng hợp N=M=3, nó th- ờng đ- ợc gọi là dạng biểu diễn trực tiếp. Ph- ơng trình sai phân (1.3.12) có thể đ- ợc chuyển sang dạng t- ơng đ- ơng. Đặc biệt bộ ph- ơng trình sau th- ờng đ- ợc sử dụng:

$$\begin{aligned} w(n) &= \sum_{k=1}^{N} a_k w(n-k) + x(n) \\ y(n) &= \sum_{r=0}^{M} b_r w(n-r) \end{aligned} \qquad (1.3.15)$$

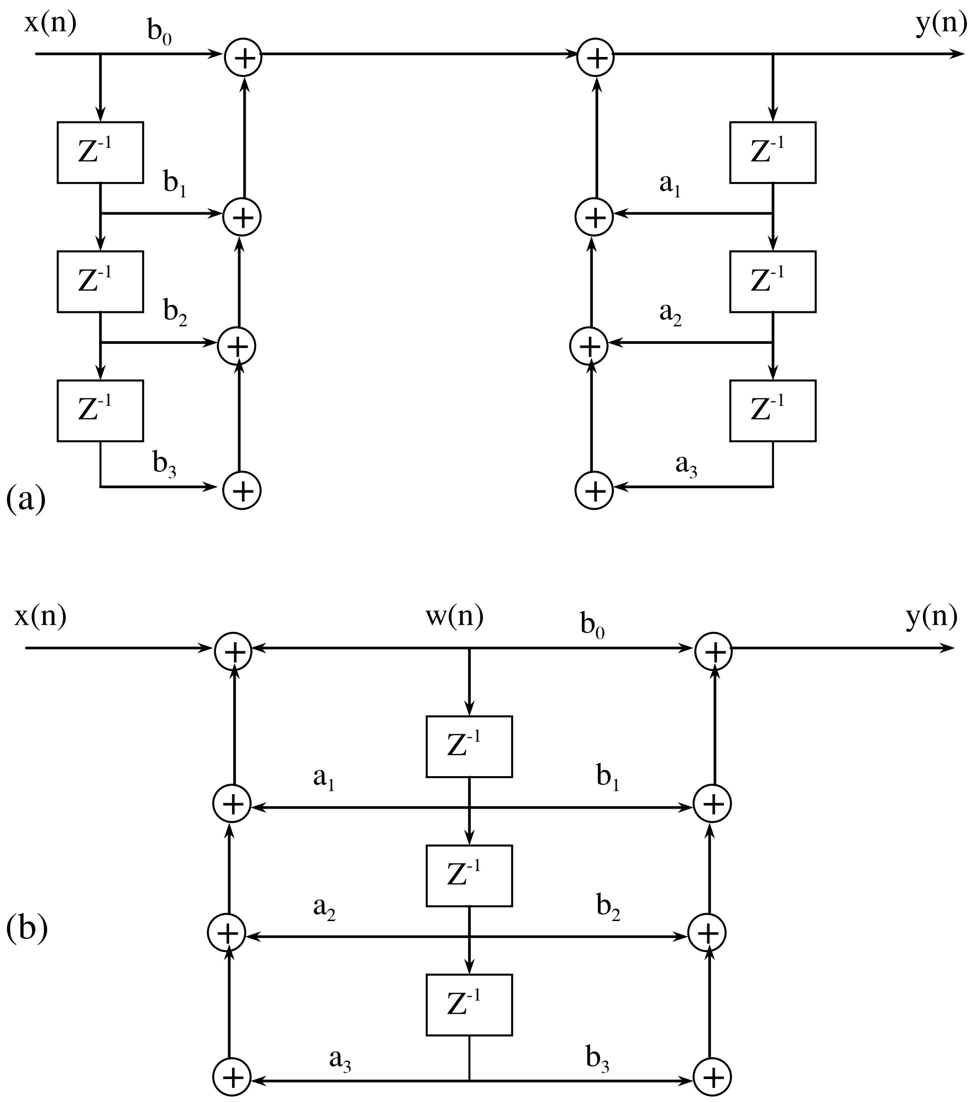
Bộ ph- ơng trình này có thể biểu diễn nh- trong hình 1.4b, với bộ nhớ để l- u giữ đ- ợc yêu cầu và chứa các giá trị dãy trễ.

Ph- ơng trình (1.3.7) chỉ ra rằng H(Z) có thể biểu diễn nh- một tích các điểm cực. Những điểm cực và điểm không này là các cặp liên hiệp phức, vì các hệ số a_k và b_k là thực.

Bằng những nhóm liên hiệp phức điểm cực và điểm không trong cặp liên hợp phức, nó cũng có thể biểu diễn H(Z) nh- tích của các hàm hệ thống cơ bản cấp hai dạng:

$$H(Z) = A\prod_{k=1}^{K}\left[\frac{1+b_{1k}Z^{-1}+b_{2k}Z^{-2}}{1-a_{1k}Z^{-1}-a_{2k}Z^{-2}}\right] \qquad (1.3.16)$$

K là phần nguyên của (N+1)/2. Hệ thống cấp hai này đ- ợc biểu diễn nh- trong hình 1.5a cho tr- ờng hợp N=M=4.

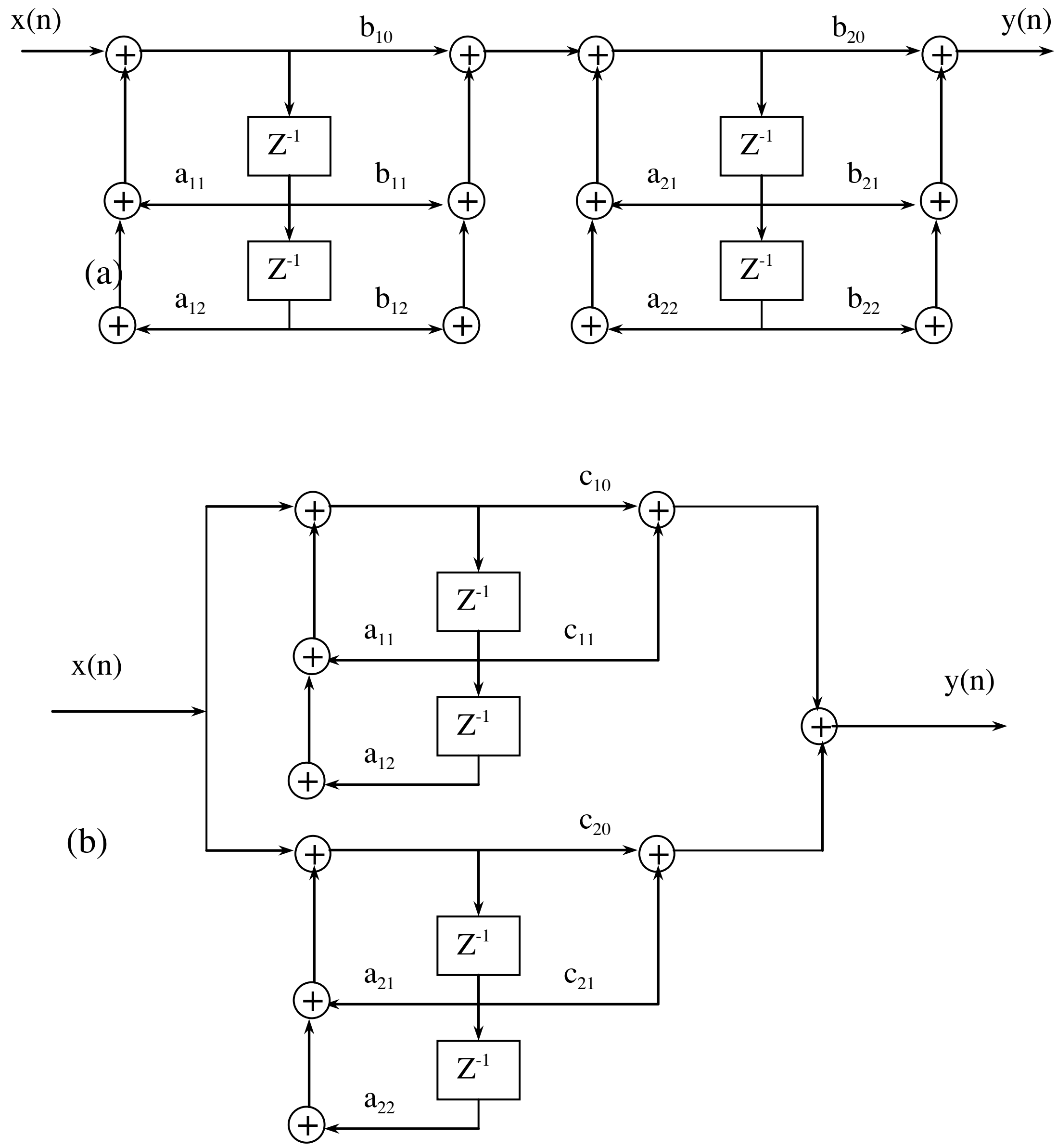


Hình 1.4. (a) Cấu trúc dạng trực tiếp;
(b) Cấu trúc dạng trực tiếp tối giản

Tiếp tục, một cấp độ cao hơn đ- ợc xét đến. Dạng phân số mở rộng của ph- ơng trình (1.3.13) cho ta h- ớng khác để biểu diễn. Bằng cách kết hợp những phần liên quan đến cực liên hợp phức, H(Z) có thể viết dạng:

$$H(Z) = \sum_{k=1}^{K} \frac{c_{0k} + c_{1k}Z^{-1}}{1 - a_{1k}Z^{-1} - a_{2k}Z^{-2}} \qquad (1.3.17)$$

Điều này gợi ý một dạng sơ đồ song song biểu diễn nh- hình 1.5b cho N=4.



Hình 1.5. (a) Dạng tầng;

(b) Dạng song song

Trong những ứng dụng lọc tuyến tính, dạng song song đ- a ra những đặc tính cao hơn về ph- ơng diện làm tròn giảm tiếng ồn, các sai số hệ số, và tính ổn định.

1.4. LẤY MẪU

Để sử dụng các ph- ơng pháp xử lý số tín hiệu đối với tín hiệu t- ơng tự, chúng ta cần biểu diễn tín hiệu nh- một dãy các giá trị. Để thực hiện biến đổi, thông th- ờng ng- ời ta dùng ph- ơng pháp lấy mẫu tín hiệu t- ơng tự. Từ $x_a(t)$, lấy các giá trị cách đều nhau ta đ- ợc:

$$x(n)=x_a(nT) \qquad -\infty<n<\infty \tag{1.4.1}$$

trong đó n là số nguyên.

Định lý lấy mẫu

Các điều kiện mà dãy các mẫu là biểu diễn duy nhất của tín hiệu t- ơng tự đ- ợc xác định nh- sau:

Nếu một tín hiệu $x_a(t)$ có biến đổi Fourier dải giới hạn $X_a(j\Omega)$, tức là $X_a(j\Omega)=0$ với $\Omega\geq 2\pi F_N$, thì $x_a(t)$ có thể tạo lại một cách duy nhất từ các mẫu cách đều nhau $x_a(nT)$, $-\infty<n<\infty$, nếu $1/T>2F_N$.

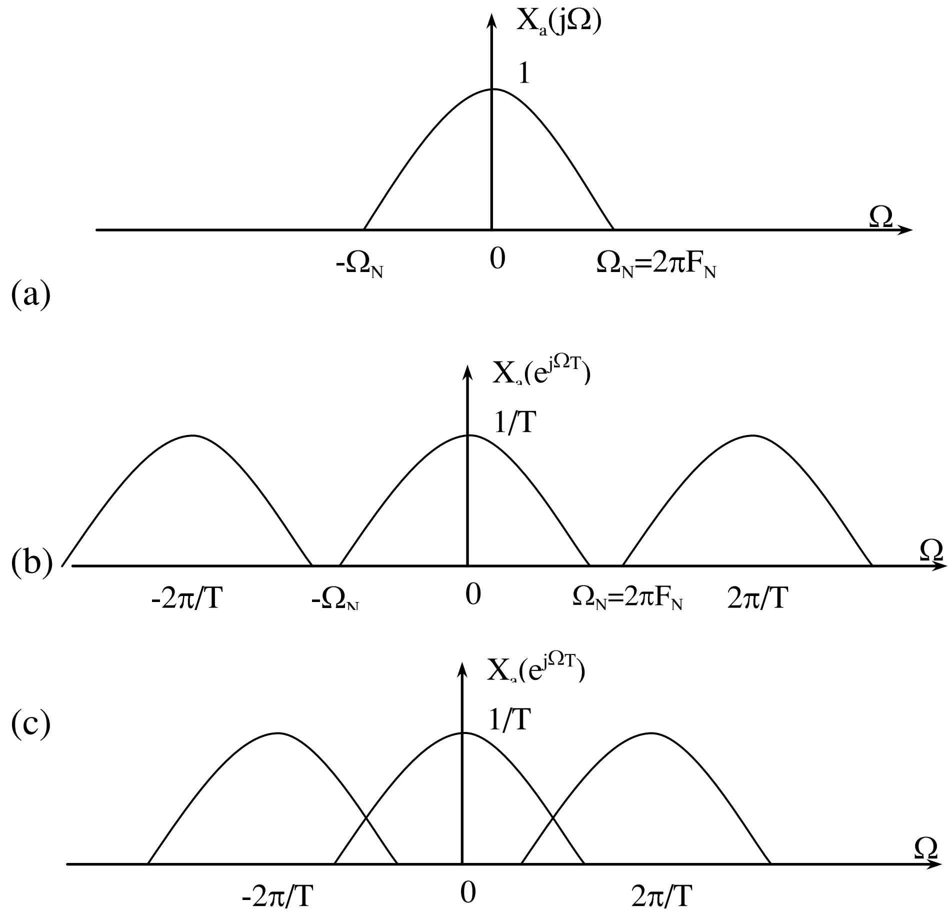
Định lý trên xuất phát từ thực tế là nếu biến đổi Fourier của $x_a(t)$ đ- ợc định nghĩa

$$X_a(j\Omega)=\int_{-\infty}^{\infty} x_a(t)e^{-j\Omega t}dt \tag{1.4.2}$$

và biến đổi Fourier của dãy x(n) đ- ợc định nghĩa nh- trong ph- ơng trình (1.2.4a) thì nếu $X(e^{j\omega})$ đ- ợc tính cho tần số $\omega=\Omega T$, thì $X(e^{j\Omega T})$ quan hệ với $X(j\Omega)$ bằng ph- ơng trình:

$$X(e^{j\Omega T})=\frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(j\Omega+j\frac{2\pi}{T}k\right) \tag{1.4.3}$$

Để thấy đ- ợc mối quan hệ trong ph- ơng trình (1.4.3), ta hãy giả thiết rằng $X_a(j\Omega)$ đ- ợc biểu diễn nh- hình 1.6a, nh- vậy $X_a(j\Omega)=0$ với $|\Omega|>\Omega_N=2\pi F_N$, tần số F_N gọi là tần số Nyquist. Theo nh- ph- ơng trình (1.4.3), $X(e^{j\Omega T})$ là tổng của một số vô hạn các bản sao của $X_a(j\Omega)$, với mỗi trung tâm là bội số nguyên của $2\pi/T$. Hình 1.6b biểu diễn tr- ờng hợp $1/T>2F_N$. Hình 1.6c biểu diễn tr- ờng hợp $1/T<2F_N$, trong tr- ờng hợp này trung tâm của ảnh tại $2\pi/T$ gối lên dải cơ bản. Điều kiện này, nơi mà một tần số cao có vẻ đảm nhiệm giống nh- là tần số thấp, đ- ợc gọi là trùm phổ. Rõ ràng rằng hiện t- ợng trùm phổ chỉ tránh đ- ợc khi biến đổi Fourier có dải giới hạn và tần số lấy mẫu lớn hơn hoặc bằng hai lần tần số lấy mẫu ($1/T>2F_N$).



Hình 1.6. Minh hoạ lấy mẫu tần số

Với điều kiện $1/T>2F_N$, rõ ràng rằng biến đổi Fourier của dãy các mẫu t- ơng ứng với biến đổi Fourier của tín hiệu t- ơng tự trong dải cơ bản nh- ,

$$X\left(e^{j\Omega T}\right) = \frac{1}{T} X_a(j\Omega) \qquad |\Omega| < \frac{\pi}{T} \tag{1.4.4}$$

Sử dụng kết quả này chúng ta có thể thiết lập mối quan hệ giữa tín hiệu t- ơng tự cơ bản và dãy các mẫu theo công thức nội suy:

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT)\left[\frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T}\right] \tag{1.4.5}$$

Nh- vậy với tần số lấy mẫu lớn hơn hoặc bằng hai lần tần số Nyqiust thì ta có thể khôi phục lại tín hiệu t- ơng tự cơ bản bằng ph- ơng trình (1.4.5).

1.5. DFT VÀ FFT

1.5.1 DFT

Khi tín hiệu t- ơng tự là một tín hiệu tuần hoàn với chu kỳ N, tức là:

$$\tilde{x}(n) = \tilde{x}(n+N) \qquad -\infty < n < \infty \tag{1.5.1}$$

Nh- vậy $\tilde{x}(n)$ có thể biểu diễn bằng tổng rời rạc, không cần biểu diễn bằng tích phân nh- trong ph- ơng trình (1.2.4b). Biểu diễn Fourier của một dãy tuần hoàn là:

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j\frac{2\pi}{N}kn} \tag{1.5.2a}$$

$$\tilde{x}(n) = \frac{1}{N}\sum_{N=0}^{N-1} \tilde{X}(k) e^{j\frac{2\pi}{N}kn} \tag{1.5.2b}$$

Đây là sự biểu diễn chính xác của dãy tuần hoàn. Bây giờ ta xét đến dãy có độ dài hữu hạn, tức là các giá trị nằm ngoài khoảng 0 ≤ n ≤ N-1 đều bằng không, biến đổi Z của dãy đó sẽ là:

$$X(Z) = \sum_{n=0}^{N-1} x(n) Z^{-n} \tag{1.5.3}$$

Nếu tính X(Z) tại N điểm cách đều nhau trên vòng tròn đơn vị, tức là $Z_k = e^{j\frac{2\pi}{N}k}$, $k = 0, 1, \ldots, N-1$, ta sẽ đ- ợc:

$$X\left(e^{j\frac{2\pi}{N}k}\right) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}, \qquad k = 0, 1, \ldots, N-1 \tag{1.5.4}$$

Nếu ta cấu trúc một dãy thành vô hạn, bằng cách lặp lại dãy x(n) nh- sau:

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n+rN) \tag{1.5.5}$$

Ta dễ dàng thấy rằng tính $X\left(e^{j\frac{2\pi}{N}k}\right)$ bằng ph- ơng trình (1.5.2a). Nh- vậy một dãy có độ dài hữu hạn có thể sử dụng biến dổi Fourier rời rạc (Discrete Fourier Transform_DFT) theo công thức:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \qquad \text{k=0, 1, ..., N-1} \tag{1.5.6a}$$

$$x(n) = \frac{1}{N}\sum_{N=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \qquad \text{n=0, 1, ..., N-1} \tag{1.5.6b}$$

Rõ ràng rằng ph- ơng trình (1.5.6) và (1.5.2) chỉ khác nhau là bỏ kí hiệu ~ (kí hiệu chỉ tính tuần hoàn) và hạn chế trong khoảng 0≤k≤N-1, 0≤n≤N-1. Tuy nhiên một điều quan trong khi sử dụng biểu diễn DFT là tất cả các dãy đ- ợc xét đến nh- là tuần hoàn. Tức là DFT thực sự là sự biểu diễn của dãy tuần hoàn đ- a ra trong ph- ơng trình (1.5.5). Một điểm khác là khi biểu diễn DFT đ- ợc sử dụng thì các chỉ số dãy phải đ- ợc thể hiện phần d- cuả N (mod). Điều này xuất phát từ thực tế là nếu x(n) có độ dài N thì

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n + rN) = x(n \bmod N) = x((n))_N \qquad (1.5.7)$$

Kí hiệu dấu ngoặc đơn kép ở trên để chỉ tính chu kỳ lặp lại của biểu diễn DFT. Một đặc điểm hiển nhiên nhất là dãy dịch chuyển đ- ợc dịch đi phần d- của N.

Biểu diễn DFT có những - u điểm sau

- DFT, X(k) có thể đ- ợc xem nh- cấp độ lấy mẫu của biến đổi Z (hoặc biến đổi Fourier) của dãy h- u hạn.
- DFT có các thuộc tính rất giống với nhiều thuộc tính hữu ích của biến đổi Z và biến đổi Fourier.
- Giá trị N của X(k) có thể tính rất hiệu quả bằng cách sử dụng các thuật toán nh- FFT (Fast Fourier Transform).

Sau đây là một số tính chất quan trong của biến đổi DFT

Bảng 1.2 Các dãy và DFT của nó

Các tính chất	Dãy miền n	DFT N điểm
1. Tính tuyến tính	$ax_1(n)+bx_2(n)$	$aX_1(k)+bX_2(k)$
2. Tính dịch chuyển theo thời gian	$x((n+n_0))_N$	$e^{j\frac{2\pi}{N}kn_0} X(k)$
3. Đảo trục thời gian	$x((-n))_N$	X*(k)
4. Tích chập của hai dãy	$\sum_{m=0}^{N-1} x(m) h((n-m))_N$	X(k).H(k)
5. Tích của hai dãy	x(n).w(n)	$\frac{1}{N}\sum_{r=0}^{N-1} X(r) W((k-r))_N$

1.5.2. FFT

Ở trên chúng ta đã biết biến đổi Fourier rời rạc (DFT). Nh- ng trong tính toán, để tăng tốc độ tính, ng- ời ta đã tìm ra thuật toán tính DFT một cách nhanh chónh và hiệu quả đ- ợc gọi là phép biến đổi nhanh Fourier.

Nh- chúng ta đã biết, DFT của dãy x(n) là:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0, 1, ..., N-1 \tag{1.5.8}$$

trong đó

$$W_N^{kn} = e^{-j\frac{2\pi}{N}kn} = W^{kn} = \cos\left(\frac{2\pi}{N}kn\right) - j\sin\left(\frac{2\pi}{N}kn\right)$$

Biến đổi Fourier rời rạc ng- ợc (IDFT) của X(k) là:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad n = 0, 1, ..., N-1 \tag{1.5.9}$$

Trong công thức (1.5.8) và (1.5.9) , cả x(n) và X(k) đều có thể là số phức

x(n)=a(n)+jb(n)

X(k)=A(k)+jB(k)

Do đó

$$A(k) + jB(k) = \sum_{n=0}^{N-1} [a(n) + jb(n)]\left[\cos\left(\frac{2\pi}{N}kn\right) - \sin\left(\frac{2\pi}{N}kn\right)\right] \tag{1.5.10}$$

hoặc

$$A(k) = \sum_{n=0}^{N-1}\left[a(n)\cos\left(\frac{2\pi}{N}kn\right) + b(n)\sin\left(\frac{2\pi}{N}kn\right)\right] \tag{1.5.11}$$

$$B(k) = \sum_{n=0}^{N-1}\left[b(n)\cos\left(\frac{2\pi}{N}kn\right) + a(n)\sin\left(\frac{2\pi}{N}kn\right)\right] \tag{1.5.12}$$

Các biểu thức (1.5.8) và (1.5.9) chỉ khác nhau về dấu của số mũ của W và ở hệ số tỉ lên 1/N. vì vậy mọi lý luận về cách tính biểu thức (1.5.8) đều đ- ợc áp dụng cho biểu thức (1.5.9) với một vài thay đổi nhỏ về dấu và hệ số tỉ lệ. Tr- ớc hết chúng ta xem xét qua cách tính trực tiếp DFT với một số nhận xét và l- u ý sau:

- Một phép nhân số phức t- ơng đ- ơng với bốn phép nhân số thực

- Số l- ợng phép tính chỉ là t- ơng đối, ví dụ nh- phép nhân với W=1 trong thực tế không cần thực hiện nh- ng ta vẫn tính, vì n lớn nên các phép tính kiểu này sẽ không đáng kể.

- Thời gian làm một phép nhân (tn), trong máy tính vạn năng lớn hơn rất nhiều thời gian làm một phép cộng (tc). Vì vậy chúng ta phải quan tâm làm giảm nhỏ phép nhân là chính. Thời gian phụ (tp) làm các công việc khác nh- truyền số liệu, đọc các hệ số sẽ có thể tạm bỏ qua. Do vậy độ phức tạp tính toán trên ph- ơng diện thời gian sẽ tỉ lệ với số phép tính số học (số phép tính nhân là chính và số phép tính cộng).

Việc tính X(k) t- ơng đ- ơng với việc tính phần thực A(k) và phần ảo B(k). Ta thấy rằng đối với mỗi giá trị của k, việc tính toán trực tiếp X(k) cần 4N phép nhân số thực và (4N-2) phép cộng số thực. Vì X(k) phải tính cho các giá trị khác nhau của k, cho nên cách tính trực tiếp DFT của một dãy x(n) cần có $4N^2$ phép tính nhân thực và N(4N-2) phép cộng số thực. Hay nói cách khác cần có N^2 phép nhân số phức và N(N-1) phép cộng số phức. Do số lần tính toán và do đó thời gian tính toán tỉ lệ gần đúng với N^2 nên rõ ràng rằng số phép toán số học cần có để tính trực tiếp DFT sẽ trở lên rất lớn khi N tăng. Do vậy mọi thuật toán đều cố gắng tìm mọi cách làm giảm số phép tính, đặc biệt là số phép nhân.

Chúng ta sẽ xét một vài thuật toán FFT cơ bản nhất và hiệu quả, các thuật toán này có số phép tính tỉ lệ với $N.\log_2(N)$. Nguyên tắc cơ bản của tất cả các thuật toán là dựa trên việc phân tích cách tính DFT của một dãy N điểm (gọi tắt là DFT N điểm) thành các phép tính DFT của các dãy nhỏ hơn. Nguyên tắc này đã dẫn đến các thuật toán khác nhau và tất cả đều giảm đáng kể thời gian tính toán. Trong phần này chúng ta sẽ xét đến hai lớp cơ bản nhất của thuật toán FFT: Thuật toán FFT phân chia theo thời gian và phân chia theo tần số.

1.5.2.1. Thuật toán FFT phân chia theo thời gian

Nguyên tắc chung

Nguyên tắc cơ bản nhất của tất cả các thuật toán FFT là dựa trên việc phân tách DFT N điểm thành DFT nhỏ hơn (tức là số điểm tính DFT nhỏ hơn). Theo cách này chúng ta sẽ khai thác cả tính tuần hoàn và tính đối xứng của W.

* Tính đối xứng $W^{k(N-n)} = (W^{kn})^*$

* Tính tuần hoàn $W^{kn} = W^{k(n+N)} = W^{(k+N)n} = W^{(k+N)(n+N)}$

Thuật toán phân chia dựa trên việc phân chia dãy x(n) thành các dãy nhỏ hơn gọi là thuật toán phân chia theo thời gian, vì chỉ số n th- ờng đ- ợc gắn với thời gian. Nguyên tắc của thuật toán này đ- ợc minh hoạ rõ rệt nhất khi ta xem sét tr- ờng hợp N lấy các giá trị đặc biệt: N là luỹ thừa của 2, (do đó nó còn có tên là FFT cơ số 2), tức là $N=2^M$.

Do N là một số chẵn nên ta có thể tính X(k) bằng cách tách x(n) thành hai dãy, mỗi dãy có N/2 điểm, một dãy chứa điểm lẻ của x(n) và một dãy chứa điểm chẵn của x(n). Cụ thể từ công thức tính X(k) ta có:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, ..., N-1$$

Sau khi tách dãy x(n) thành các dãy đánh số chẵn và số lẻ, ta có:

$$X(k) = \sum_{n=chẵn}^{N-1} x(n) W_N^{kn} + \sum_{n=lẻ}^{N-1} x(n) W_N^{kn}$$

hoặc bằng cách thay thế biến n=2r đối với N chẵn và n=2r+1 đối với N là lẻ.

$$\begin{aligned} X(k) &= \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r) (W^2)_N^{rk} + W^k . \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) (W^2)_N^{rk} \end{aligned} \qquad (1.5.13)$$

Bởi vì $W^2 = W_{\frac{N}{2}}$, $W^2 = e^{-j2\frac{2\pi}{N}} = e^{-j\frac{2\pi}{N/2}} = W_{\frac{N}{2}}$ nên biểu thức (1.5.13) có thể viết lại thành:

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) (W)_{N/2}^{rk} + W^k . \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) (W)_{N/2}^{rk}$$

Đặt $X_0(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) (W)_{N/2}^{rk}$ (X_0 t- ơng ứng với r chẵn)

và $X_1(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) (W)_{N/2}^{rk}$ (X_1 t- ơng ứng với r lẻ)

ta có

$$X(k)=X_0(k)+W^k.X_1(k) \qquad (1.5.14)$$

Có thể thấy ngay $X_0(k)$ và $X_1(k)$ chính là DFT của N/2 điểm, trong đó $X_0(k)$ là DFT N/2 điểm của các điểm đánh số chẵn của dãy x(n) ban đầu, còn $X_1(k)$ là DFT N/2 điểm đánh số lẻ của dãy ban đầu. Mặc dù chỉ số k của dãy X(k) chạy qua N giá trị: k=0, 1, ..., N-1 nh- ng ta chỉ cần tính $X_0(k)$ và $X_1(k)$ với k chạy từ 0 đến N/2 -1, do $X_0(k)$ và $X_1(k)$ tuần hoàn với chu kỳ N/2. Sau khi hai DFT $X_0(k)$ và $X_1(k)$ t- ơng ứng đ- ợc tính, chúng sẽ đ- ợc kết hợp với nhau để tạo ra DFT N điểm là X(k).

Bây giờ ta có thể sơ bộ tính số phép nhân và cộng cần có cho cách tính DFT kiểu này. Ta biết rằng một DFT N điểm nếu tính trực tiếp thì cần N^2 phép nhân phức và khoảng N^2 (chính xác là N(N-1)) phép cộng phức. Sau khi phân tách thành 2 DFT N/2 điểm ta cần $2(N/2)^2$ phép nhân phức và khoảng $2(N/2)^2$ phép cộng phức để thực hiện $X_0(k)$ và $X_1(k)$. Sau đó ta mất thêm N phép nhân phức để thực hiện nhân giữa W^k và $X_1(k)$ và thêm N phép cộng phức để tính X(k) từ $X_0(k)$ và $W^k.X_1(k)$. Tổng cộng lại ta cần $2N+2(N/2)^2=2N+N^2/2$ phép nhân phức và phép cộng phức để tính tất cả các giá trị X(k). Dễ dàng kiểm tra lại rằng với N>2 thì $2N+N^2/2$ sẽ nhỏ hơn N^2. Nh- vậy với N chẵn ta đã chia nhỏ DFT N điểm thành 2 DFT N/2 điểm với số phép tính và thời gian tính nhỏ hơn. Với N/2 là một số chẵn thì lại hoàn toàn t- ơng tự, ta lại có thể chia DFT N/2 điểm thành các DFT N/4 điểm. Nếu số N có dạng $N=2^M$ thì ta có thể chia đôi nh- vậy M lần, cho đến khi số điểm tính DFT là bằng 2. Do việc liên tục chia 2 nên ng- ời ta còn gọi FFT cơ số 2 để phân biệt FFT cơ số 4 nếu $N=4^M$. Cụ thể $X_0(k)$ có thể lại đ- ợc tách nh- sau:

$$X_0(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_{N/2}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} g(r) W_{N/2}^{rk}$$

t- ơng tự nh- tr- ớc, ta đặt l=2r để tách g(r) thành hai dãy chẵn lẻ

$$\begin{aligned}
X_0(k) &= \sum_{l=0}^{\frac{N}{4}-1} g(2l) W_{N/2}^{2lk} + \sum_{l=0}^{\frac{N}{4}-1} g(2l+1) W_{N/2}^{(2l+1)k} \\
&= \sum_{l=0}^{\frac{N}{4}-1} g(2l) W_{N/4}^{lk} + W_{N/2}^{k}.\sum_{l=0}^{\frac{N}{4}-1} g(2r+1) W_{N/4}^{rk} \\
&= X_{00}(k) + W_{N/2}^{k}.X_{01}(k)
\end{aligned}$$

Nh- vậy $X_0(k)$ lại đ- ợc tách thành 2 DFT là $X_{00}(k)$ và $X_{01}(k)$. Với $X_{00}(k)$ là DFT của dãy g(r) có chỉ số chẵn và $X_{01}(k)$ là DFT của dãy g(r) có chỉ số lẻ. Công việc đ- ợc làm hoàn toàn t- ơng tự cho $X_1(k)$.

Cuối cùng việc phân tách nh- vậy dẫn đến các DFT 2 điểm, khi đó các hệ số W thực sự mang giá trị đặc biệt là 1 và -1 nên trong thực tế không phải làm phép nhân nữa và việc phân chia cũng dừng lại ở đây.

Với $N=2^M$, số lần phân chia là M lần. Số phép tính nhân và cộng phức cần thực hiện sau $M=\log_2 N$ phân chia có thể tính nh- sau: t- ơng ứng với mỗi lần phân chia ta cần N phép nhân phức để nhân các kết quả của DFT của tầng tr- ớc với hệ số W t- ơng ứng và N phép cộng phức để nhóm kết quả lại với nhau. Tổng cộng lại, ta chỉ cần $N.\log_2 N$ phép nhân phức và $N\log_2 N$ phép cộng phức để thực hiện FFT.

1.5.2.2. Thuật toán FFT cơ số 2 phân chia theo tần số

Nguyên tắc chung

ở trên chúng ta đã trình bày thuật toán FFT dựa trên việc phân chia nhỏ dãy vào x(n) để phân tách việc tính DFT N điểm thành các DFT nhỏ hơn. Trong phần này chúng ta sẽ xem xét thuật toán FFT dựa trên việc phân tách dãy ra X(k) thành các dãy nhỏ hơn theo cùng một cách phân tách dãy x(n). Do chỉ số k của dãy X(k) gắn liền với thang tần số nên các thuật toán này đ- ợc gọi là các thuật toán FFT phân chia theo tần số.

Với giả thiết $N=2^M$, ta có thể chia dãy vào thành hai nửa, một nửa chứa N/2 mẫu đầu, x(n) với n=0, 1, ..., N/2 -1, nửa sau ch- a N/2 mẫu còn lại, ta có:

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{kn}+\sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{kn}$$

hoặc

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{kn}+W^{\frac{N}{2}k}.\sum_{n=0}^{\frac{N}{2}-1} x\left(n+\frac{N}{2}\right)W_N^{kn}$$

Với $W_N^{N/2}=-1$ và kết hợp tổng lại ta có:

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}\left[x(n)+(-1)^k x\left(n+\frac{N}{2}\right)\right]W_N^{kn}$$

xét k=2r (k chẵn) và k=2r+1 (k lẻ) ta nhận đ- ợc X(2r) và X(2r+1) t- ơng ứng với dãy ra chỉ số chẵn và dãy ra chỉ số lẻ:

$$X(2r) = \sum_{n=0}^{\frac{N}{2}-1}\left[x(n) + x\left(n+\frac{N}{2}\right)\right]W_N^{2rn}$$

với r=0, 1, ..., (N/2-1)

$$X(2r+1) = \sum_{n=0}^{\frac{N}{2}-1}\left[x(n) - x\left(n+\frac{N}{2}\right)\right]W_N^{n}W_N^{2rn}$$

Do $W_N^{2rn} = W_{N/2}^{rn}$ nên ta có thể thấy ngay X(2r) chính là DFT N/2 điểm của dãy g(n)=x(n)+x(n+N/2); g(n) là tổng của nửa đầu của dãy x(n) với nửa sau dãy x(n). Còn X(2r+1) là DFT N/2 điểm của tích W với dãy h(n)=x(n)-x(n+N/2); h(n) là hiệu của nửa đầu dãy x(n) với nửa sau của dãy x(n). Nh- vậy DFT N điểm của dãy x(n) có thể đ- ợc tính nh- sau:

Tr- ớc hết tạo ra hai dãy h(n) và g(n), sau đó thực hiện W.h(n). Cuối cùng thực hiện DFT của hai dãy này, ta sẽ có các điểm ra X(k) chỉ số chẵn và X(k) chỉ số lẻ.

Với mỗi DFT N/2 điểm ta lại tiến hành hoàn toàn t- ơng tự nh- đã làm ở trên để tách mỗi DFT N/2 điểm thành 2 DFT N/4 điểm. Cứ thế cho đến khi DFT cuối cùng là các DFT hai điểm. Qua quá trình nh- vậy tại mỗi lần phân tách, ta cần N/2 phép nhân và tất cả có M=$\log_2$N lần phân tách. Số phép nhân tổng cộng là $\frac{N}{2}\log_2 N$, bằng với phép nhân trong cách tính theo ph- ơng pháp phân chia theo thời gian, số phép cộng cũng nh- vậy.

CHƯƠNG 2 :

ƯỚC LƯỢNG TUYẾN TÍNH VÀ CÁC BỘ LỌC TUYẾN TÍNH TỐI ƯU

2.1. BIỂU DIỄN QUÁ TRÌNH NGẪU NHIÊN ỔN ĐỊNH

Trong phần này chúng ta minh họa một quá trình ngẫu nhiên ổn định với độ nhạy cao có thể biểu diễn nh- đầu ra của một hệ thống tuyến tính nhân quả và hệ thống tuyến tính khả đảo nhân quả bị tác động bởi nhiễu trắng. Điều kiện hệ thống khả đảo cũng cho phép biểu diễn quá trình ngẫu nhiên ổn định độ nhạy cao bằng đầu ra của hệ thống ng- ợc, nó là một quá trình nhiễu trắng.

Xét quá trình ổn định độ nhạy cao x(n) với chuỗi tự t- ơng quan γ_{xx}(m) và mật độ phổ công suất $\Gamma_{xx}(f)$, $|f| \le \frac{1}{2}$. Biến đổi z của chuỗi tự t- ơng quan γ_{xx}(m) là:

$$\Gamma_{xx}(z) = \sum_{m=-\infty}^{\infty} \gamma_{xx}(m) z^{-m} \tag{2.1.1}$$

từ công thức chúng ta đạt đ- ợc mật độ phổ công suất Γ_{xx}(z), xét trong vòng tròn đơn vị (bằng việc thay thế z=exp$(j2\pi f)$).

Bây giờ, giả sử logΓ_{xx}(z) đ- ợc phân tích (xử lý đạo hàm của tất cả các bậc) trong miền vành khuyên của mặt phẳng z có chứa vòng tròn đơn vị (r_1<|Z|< r_2, trong đó r_1< 1 và r_2>1). Nh- vậy, log Γ_{xx}(z) có thể khai triển thành chuỗi Laruent theo công thức

$$\log\Gamma_{xx}(z) = \sum_{m=-\infty}^{\infty} v(m) z^{-m} \tag{2.1.2}$$

ở đây, v(m) là những hệ số trong chỗi mở rộng. Chúng ta có thể quan niệm v(m) nh- là chuỗi biến đổi Z, V(z) = logΓ_{xx}(z). Chúng ta có thể tính logΓ_{xx}(z) trên vòng tròn đơn vị

$$\log\Gamma_{xx}(f) = \sum_{m=-\infty}^{\infty} v(m) e^{-j2\Pi fm} \tag{2.1.3}$$

Vì vậy, v(m) là những hệ số Fourier trong chuỗi Fourier mở rộng của hàm tuần hoàn log $\Gamma_{xx}(f)$. Do đó

$$v(m) = \int_{-\frac{1}{2}}^{\frac{1}{2}} [\log \Gamma_{xx}(f)] e^{j2\Pi fm} df \qquad m = 0, \pm 1, \ldots \qquad (2.1.4)$$

Chúng ta quan sát thấy v(m)= v(-m), khi $\Gamma_{xx}(f)$ là thực và là hàm chẵn của f.

Từ (2.1.2) ở trên ta có

$$\log \Gamma_{xx}(z) = \exp\left[\sum_{m=1}^{\infty} v(m) z^{-m}\right] \qquad (2.1.5)$$

$$= \sigma_w^2 H(z) H(z^{-1})$$

ở đây, bằng cách định nghĩa σ_w^2 = exp[v(0)] và

$$H(z) = \exp\left[\sum_{m=1}^{\infty} v(m) z^{-m}\right] \qquad |z| > r_1 \qquad (2.1.6)$$

Nếu (2.1.5) đ- ợc tính trên vòng tròn đơn vị, chúng ta có mật độ phổ là:

$$\Gamma_{xx}(f) = \sigma_w^2 |H(f)|^2 \qquad (2.1.7)$$

Chú ý rằng

$$\log \Gamma_{xx}(f) = \log \sigma_w^2 + \log H(f) + \log H^*(f)$$

$$= \sum_{m=-\infty}^{\infty} v(m) e^{-j2\Pi fm}$$

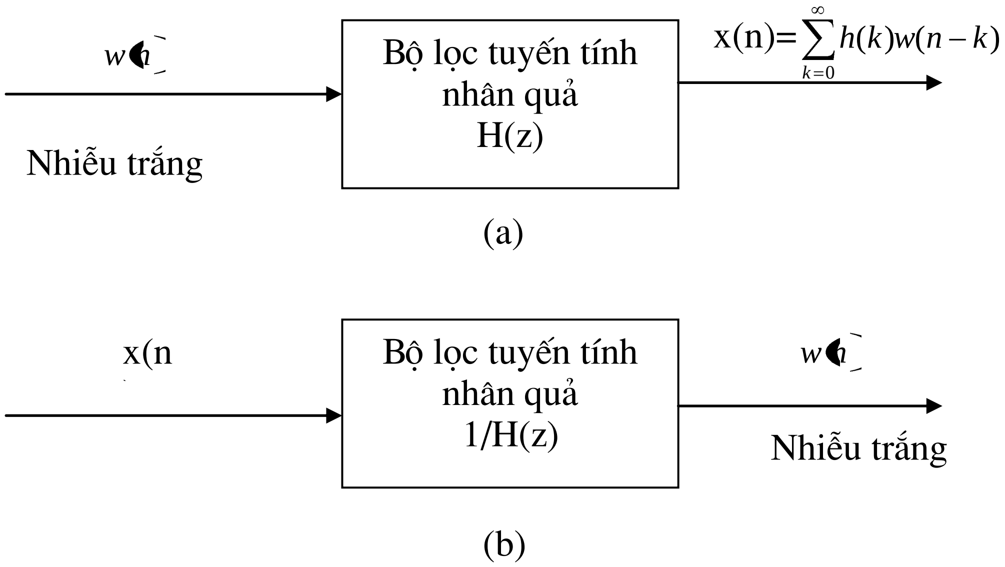
Từ định nghĩa H(z) trong công thức (2.1.6), nó đ- ợc hiểu rằng là một phần nhân quả của chuỗi Fourier trong (2.1.3) đ- ợc kết hợp với H(z) và một phần khác đ- ợc kết hợp với $H(z^{-1})$.

Bộ lọc với hàm hệ thống đ- ợc đ- a ra bởi (2.1.6) đ- ợc xét trong miền $|z| > r_1 < 1$, do đó trong miền này nó là chuỗi Taylor mở rộng nh- một hệ thống nhân quả có dạng:

$$H(z) = \sum_{m=0}^{\infty} h(n) z^{-n} \qquad (2.1.8)$$

Đầu ra của bộ lọc này h- ớng tới chuỗi đầu vào nhiễu trắng $w(n)$ với mật độ phổ công suất σ_w^2 là quá trình ngẫu nhiên ổn định x(n), với mật độ phổ đầu vào $\Gamma_{xx}(f)$ có thể biến đổi thành quá trình nhiễu trắng bằng cách đ- a x(n) qua bộ lọc tuyến tính với hàm hệ thống 1/H(z). Chúng ta gọi bộ lọc này *là bộ lọc nhiễu trắng*. Ở đầu ra, $w(n)$ đ- ợc gọi *là quá trình biến đổi*. Kết hợp với quá trình ngẫu nhiên ổn định x(n). Hai mối quan hệ này đ- ợc chứng minh trong hình (2.1).

Kết quả của quá trình ngẫu nhiên ổn định x(n) khi đầu ra của bộ lọc IIR với hàm hệ thống H(z) đ- a ra bởi (2.1.8) và kích thích bởi chuỗi nhiễu trắng $w(n)$ đ- ợc gọi là biểu diễn *Wold*.



Hình 2.1: (a) Bộ lọc sinh ra quá trình ngẫu nhiên x(n) từ chuỗi nhiễu trắng

(b) Bộ lọc ng- ợc

2.1.1 Công suất phổ tỉ lệ

Bây giờ, chúng ta xét tr- ờng hợp mật độ phổ công suất của quá trình ngẫu nhiên ổn định x(n) là hàm hữu tỉ, đ- ợc biểu diễn

$$\Gamma_{xx}(z) = \sigma_w^2 \frac{B(z)B(z^{-1})}{A(z)A(z^{-1})} \qquad r_1 < |z| < r_2 \qquad (2.1.9)$$

ở đây đa thức A(z) và B(z) có nghiệm, nghiệm này nằm trong vòng tròn đơn vị trong mặt phẳng z. Bộ lọc tuyến tính H(z) sinh ra quá trình ngẫu nhiên x(n) từ chuỗi nhiễu trắng $w(n)$ cũng là hữu tỉ và đ- ợc biểu diễn

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum b_k z^{-k}}{1 + \sum a_k z^{-k}} \qquad |z| > r_1 \qquad (2.1.10)$$

ở đây b_k và a_k là những hệ số bộ lọc, nó xác định vị trí của các điểm không và điểm các cực tách biệt của H(z). Do đó H(z) là nhân quả, ổn định và pha tối thiểu. Nghịch đảo 1/H(z) cũng là nhân quả, ổn định và là hệ thống tuyến tính pha tối thiểu. Do vậy, quá trình ngẫu nhiên x(n) là kết quả duy nhất về đặc tính đã thống kê của quá trình biến đổi $w(n)$ và ng- ợc lại.

Để hệ thống tuyến tính cùng với hàm của hệ thống ngẫu nhiên H(z) đ- ợc đ- a ra bởi (2.1.10), đầu ra x(n) có quan hệ với đầu vào $w(n)$ bằng các ph- ơng trình sai phân

$$x(n) + \sum a_k x(n-k) = \sum b_k \omega(n-k) \qquad (2.1.11)$$

Chúng ta sẽ phân biệt trong 3 tr- ờng hợp cụ thể:

Quá trình tự hồi qui (AR): $b_0 = 1$, $b_k = 0$, $k > 0$

Trong tr- ờng hợp này, bộ lọc tuyến tính H(z) = 1/A(z) là bộ lọc toàn điểm cực và ph- ơng trình sai phân cho mối quan hệ đầu vào_đầu ra là

$$x(n) + \sum a_k x(n-k) = \omega(n) \qquad (2.1.12)$$

bộ lọc nhiễu trắng tạo ra quá trình biến đổi là bộ lọc toàn điểm không.

Quá trình dịch chuyển trung bình(MA) : $a_k = 0$, $k \geq 1$

Trong tr- ờng hợp này bộ lọc tuyến tính H(z)= B(z) là bộ lọc toàn điểm không và ph- ơng trình sai phân cho mối quan hệ đầu vào_đầu ra là

$$x(n) = \sum_{k=0}^{a} b_k \omega(n-k) \qquad (2.1.13)$$

Bộ lọc nhiễu trắng cho quá trình MA là bộ lọc toàn điểm cực.

Quá trình dịch chuyển trung bình tự hồi qui (ARMA)

Trong tr- ờng hợp này bộ lọc tuyến tính H(z)=B(z)/A(z) có hữu hạn cả điểm cực và điểm không trong mặt phẳng z và t- ơng ứng với ph- ơng trình khác đ- a ra bởi (2.1.11). Hệ thống ng- ợc tạo ra quá trình biến đổi từ x(n) cũng là hệ thống điểm không - điểm cực của công thức

1/H(z)= A(z)/B(z)

2.1.2. Mối quan hệ giữa các thông số bộ lọc và chuỗi tự t- ơng quan

Khi mật độ phổ công suất của quá trình ngẫu nhiên ổn định là hàm hữu tỷ, tồn tại mối quan hệ cơ bản giữa chuỗi tự t- ơng quan $y_{xx}(m)$ và thông số a_k, b_k của bộ lọc tuyến tính H(z), bộ lọc đ- ợc tạo ra bởi quá trình lọc chuỗi nhiễu trắng w(n). Mối quan hệ này có thể đạt đ- ợc bằng cách nhân ph- ơng trình sai phân trong (2.1.11) với $x^*(n-m)$, ta đ- ợc kết quả mong muốn ở hai vế của ph- ơng trình. Do đó chúng ta có:

$$E[x(n)x^*(n-m)] = -\sum_{k=1}^{p} a_k E[x(n-k)x^*(n-m)]$$

$$+\sum_{k=0}^{q} b_k E[w(n-k)x^*(n-m)] \qquad (2.1.14)$$

Vì vậy

$$y_{xx}(n) = -\sum_{k=1}^{p} a_k y_{xx}(n-k) + \sum_{k=0}^{q} b_k y_{wx}(n-k) \qquad (2.1.15)$$

ở đây $y_{wx}(n)$ là chuỗi t- ơng quan chéo giữa $w(n)$ và $x(n)$

T- ơng quan chéo $y_{wx}(n)$ có quan hệ với đáp ứng xung của bộ lọc. Đó là,

$$\begin{aligned} y_{wx}(n) &= E[x^*(n)w(n+m)] \\ &= E\left[\sum h(k)\omega^*(n-k)w(n+m)\right] \qquad (2.1.16) \\ &= \sigma_w^2 h(-m) \end{aligned}$$

trong đó, ở b- ớc tr- ớc, chúng ta sử dụng chuỗi $w(n)$ là trắng. Do đó

$$Y_{wx}(n) = \begin{cases} 0, & m > 0 \\ \sigma_w^2 h(-m), & m \le 0 \end{cases} \qquad (2.1.17)$$

Bằng cách kết hợp (2.1.17) với (2.1.14) chúng ta đạt đ- ợc mối quan hệ mong muốn

$$y_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k y_{xx}(m-k), & m > q \\ -\sum_{k=1}^{p} a_k y_{xx}(m-k) + \sigma_\omega^2 \sum_{k=0}^{q-m} h(k) b_{k+m}, & 0 \le m \le q \\ y_{xx}^*(-m), & m \le 0 \end{cases} \qquad (2.1.18)$$

Kết quả này là mối quan hệ không tuyến tính giữa $y_{xx}(m)$ và thông số a_k và b_k.

Mối quan hệ trong (2.1.18) thông th- ờng dùng trong quá trình ARMA. Đối với quá trình AR (2.1.18) đơn giản hơn:

$$y_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k y_{xx}(m-k) & m>0 \\ -\sum_{k=1}^{p} a_k y_{xx}(m-k) + \sigma_\omega^2, & m=0 \\ y_{xx}^*(-m) & m<0 \end{cases} \tag{2.1.19}$$

Do đó chúng ta có mối quan hệ tuyến tính giữa $y_{xx}(m)$ và thông số a_k. Ph- ơng trình này đ- ợc gọi là *ph- ơng trình Yule_Walker* và có thể biểu diễn trong ma trận

$$\begin{bmatrix} y_{xx}(0) & y_{xx}(-1) & y_{xx}(-2) & \cdots & y_{xx}(-p) \\ y_{xx}(1) & y_{xx}(0) & y_{xx}(-1) & \cdots & y_{xx}(-p+1) \\ \vdots & \vdots & \vdots & & \vdots \\ y_{xx}(p) & y_{xx}(p-1) & y_{xx}(p-2) & \cdots & y_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \begin{bmatrix} \sigma_\omega^2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.1.20}$$

Ma trận t- ơng quan là Toeplitz và do đó nó có thể đ- ợc đảo ng- ợc bằng cách dùng các thuật toán đ- ợc miêu tả trong phần 2.3.

Cuối cùng, bằng cách đặt $a_k = 0$, $1 \le k \le p$ và $h(k) = b_k$, $0 \le k \le q$, trong (2.1.18), chúng ta đạt đ- ợc mối quan hệ cho chuỗi tự t- ơng quan trong tr- ờng hợp của quá trình MA, cụ thể là,

$$y_{xx}(m) = \begin{cases} \sigma_\omega^2 \sum_{k=0}^{q} b_k b_{k+m}, & 0 \le m \le q \\ 0 & m>0 \\ y_{xx}^*(-m) & m<0 \end{cases} \tag{2.1.21}$$

2.2 □ ỚC L□ ỢNG TUYẾN TÍNH TIẾN VÀ LÙI

Ước l- ợng tuyến tính là đề tài quan trọng trong xử lý tín hiệu số - ớc l- ợng có rất nhiều ứng dụng trong thực tế. Trong phần này chúng ta xem xét vấn đề giá trị - ớc l- ợng tuyến tính của quá trình ngẫu nhiên ổn định tiến hoặc lùi về mặt thời gian. Biến đổi công thức dẫn tới cấu trúc bộ lọc l- ới và một vài vấn đề liên quan tới tham số của các mẫu tín hiệu.

2.2.1 Ước l- ợng tuyến tính tiến

Hãy bắt đầu với vấn đề - ớc l- ợng giá trị tr- ớc của quá trình ngẫu nhiên ổn định từ các giá trị nhận đ- ợc tr- ớc đó. Đặc biệt, chúng ta xét *- ớc l- ợng tuyến tính một b- ớc*, thực hiện - ớc l- ợng giá trị x(n) bằng tổ hợp tuyến tính có trọng số của giá trị cũ x(n-1), x(n-2) ... x(n-p). Do đó giá trị - ớc l- ợng tuyến tính của x(n) là

$$\hat{x}(n) = -\sum_{k=1}^{p} a_p(k)\, x(n-k) \qquad (2.2.1)$$

ở đây, $-a_p(k)$ đại diện cho trọng số trong tổ hợp tuyến tính. Trọng số này đ- ợc gọi là *hệ số - ớc l- ợng* của - ớc l- ợng tuyến tính tiến một b- ớc của *bậc p*. Dấu âm trong định nghĩa x(n) để phù hợp trong toán học và thuận tiện trong thực hiện.

Sự chênh lệch giữa giá trị x(n) và giá trị - ớc l- ợng $\hat{x}(n)$ đ- ợc gọi là *lỗi - ớc l- ợng tiến*, đ- ợc biểu diễn nh- $f_p(n)$

$$f_p(n) = x(n) - \hat{x}(n)$$

$$= x(n) + \sum_{k=1}^{p} a_p(k)\, x(n-k) \qquad (2.2.2)$$

Chúng ta xem - ớc l- ợng tuyến tính t- ơng đ- ơng tới việc lọc tuyến tính, ở đây giá trị - ớc l- ợng là tạp nhiễu trong bộ lọc tuyến tính nh- trong hình (2.2). Đây đ- ợc gọi *bộ lọc - ớc l- ợng lỗi* với chuỗi đầu vào x(n) và chuỗi đầu ra $f_p(n)$. Sơ đồ thực hiện cho bộ lọc - ớc l- ợng lỗi thể hiện trong hình (2.3). Sơ đồ thực hiện này là bộ lọc FIR dạng trực tiếp. Với hàm của hệ thống.

$$A_p(z) = \sum_{k=0}^{p} a_p(k)\, z^{-k} \qquad (2.2.3)$$

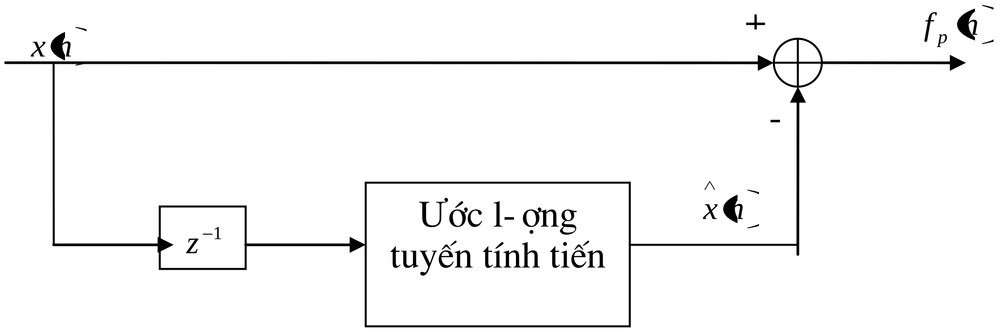
ở đây, định nghĩa $a_p(0) = 1$.

Ta có thể sử dụng dạng sơ đồ khác của bộ lọc lỗi - ớc l- ợng, nó có dạng cấu trúc hình thang. Miêu tả cấu trúc này và quan hệ của nó tới cấu trúc bộ lọc FIR dạng trực tiếp. Hãy bắt đầu với giá trị - ớc l- ợng của bậc p=1. Đầu ra của bộ lọc là

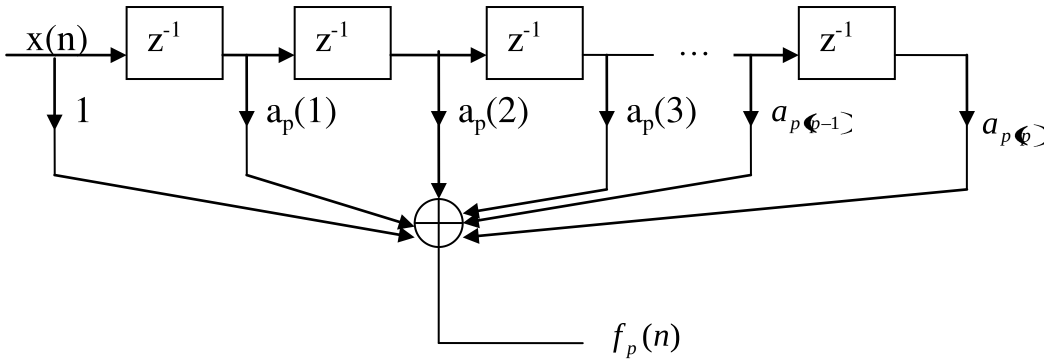
$$f_1(n) = x(n) + a_1(1)\, x(n-1) \qquad (2.2.4)$$

Đầu ra này có thể đạt đ- ợc từ bộ lọc l- ới đơn tầng minh hoạ trong (2.4) bằng cách kích thích cả hai đầu vào bởi (2.2.4) và lấy đầu ra trên nhánh trên.

Vì vậy đầu ra chính xác theo công thức (2.2.4) nếu chúng ta chọn $K_1 = a_1(1)$. Thông số K_1 trong bộ lọc l- ới gọi là *hệ số phản xạ*.



Hình 2.2 : Ước l- ợng tuyến tính tiến



Hình 2.3 : Bộ lọc - ớc l- ợng lỗi

Tiếp theo, xét đến - ớc l- ợng của bậc p=2. Trong tr- ờng hợp đầu ra của bộ lọc FIR dạng trực tiếp là:

$$f_2(n) = x(n) + a_2(1)x(n-1) + a_2(2)x(n-2) \tag{2.2.5}$$

Bằng cách kết hợp hai tầng l- ới nh- trong hình 2.5. Nó có khả năng đạt đ- ợc giống nh- đầu ra (2.2.5). Thực vậy, hai đầu ra từ tầng đầu là

$$\begin{aligned} f_1(n) &= x(n) + K_1 x(n-1) \\ g_1(n) &= K_1^* x(n) + x(n-1) \end{aligned} \tag{2.2.6}$$

Hai đầu ra từ tầng thứ 2 là

$$\begin{aligned} f_2(n) &= f_1(n) + K_2 g_1 x(n-1) \\ g_2(n) &= K_2^* f_1(n) + g_1(n-1) \end{aligned} \tag{2.2.7}$$

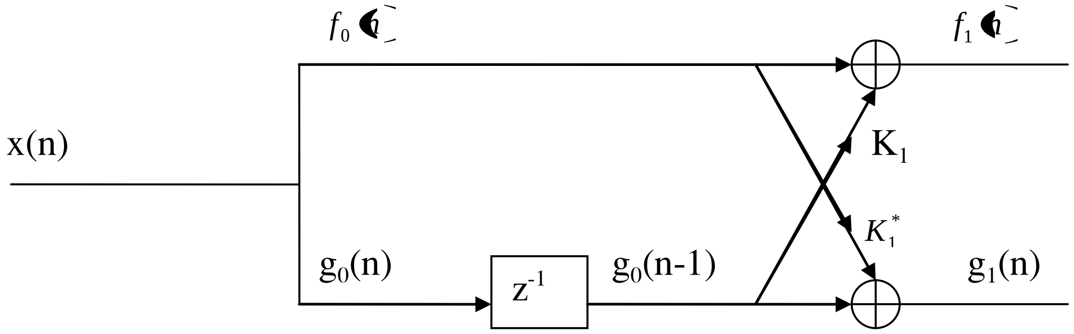
Nếu chúng ta tập trung chú ý vào $f_2(n)$ và thay thế $f_1(n)$ và $g_{1(n-1)}$ từ (2.2.6) thành (2.2.7). Chúng ta đạt đ- ợc

$$f_2(n) = x(n) + K_1 x(n-1) + K_2\left[K_1^* x(n-1) + x(n-2)\right] \tag{2.2.8}$$

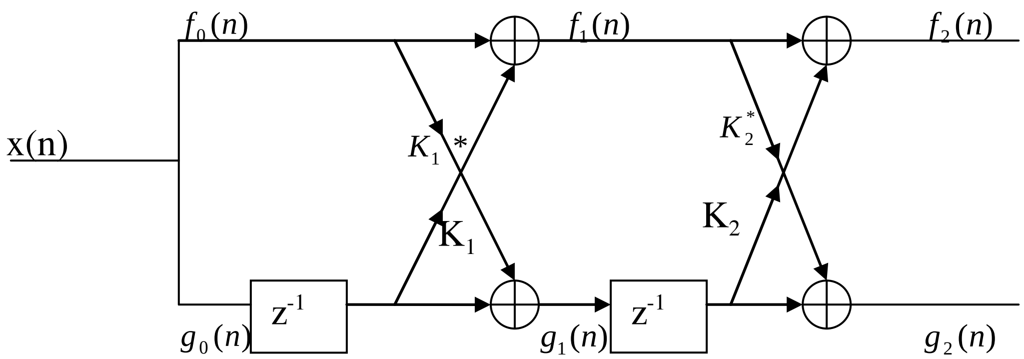
$$= x(n) + (K_1 + K_1^* K_2)x(n-1) + K_2 x(n-2)$$

Bây giờ (2.2.8) giống với đầu ra của bộ lọc FIR dạng trực tiếp đ- a ra bởi (2.2.5) nếu chúng ta cân bằng các hệ số. Do đó

$a_2(2) = K_2$, $\qquad a_2(1) = K_1 + K_1^* K_2$ (2.2.9)



Hình 2.4 : Bộ lọc l- ới đơn tầng



Hình 2.5 : Bộ lọc l- ới hai tầng

hoặc t- ơng đ- ơng,

$K_2 = a_2(2) \qquad K_1 = a_1(1)$ (2.2.10)

Bằng cách tiếp tục một quá trình này có thể chứng minh bằng ph- ơng pháp qui nạp t- ơng đ- ơng giữa bộ lọc mẫu trực tiếp FIR loại mth và bộ lọc l- ới mth hoặc loại m. Bộ lọc l- ới nói chung đ- ợc miêu tả đặt theo sau *những ph- ơng trình bậc đệ qui:*

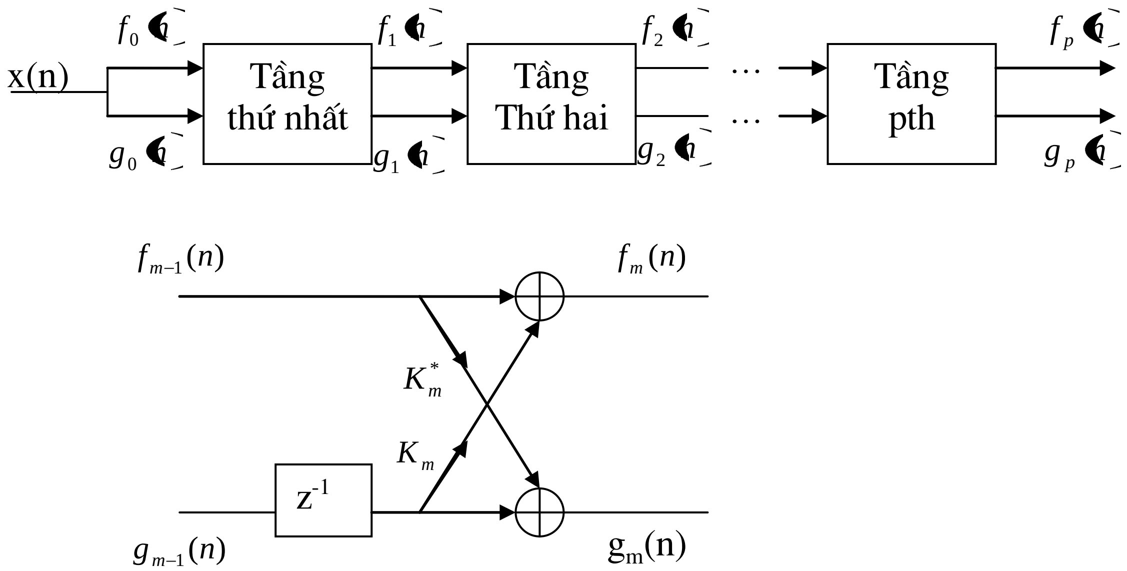
$$f_0(n) = g_0(n) = x(n)$$

$$\begin{aligned} f_m(n) &= f_{m-1}(n) + K_m g_{m-1}(n-1) \quad m = 1,2,\ldots,p \\ g_m(n) &= K_m^* f_{m-1}(n) + g_{m-1}(n-1) \quad m = 1,2,\ldots,p \end{aligned} \qquad (2.2.11)$$

Sau đó đầu ra của bộ lọc l- ới tầng p giống với đầu ra của bộ lọc mẫu trực tiếp bậc p. Hình 2.6 minh hoạ bộ lọc l- ới tầng p dạng sơ đồ khối nh- biểu diễn trong công thức (2.2.11).

Kết quả của sự t- ơng đ- ơng giữa bộ lọc -ớc l-ợng lỗi FIR dạng trực tiếp và bộ lọc FIR dạng l- ới, đầu ra của bộ lọc l- ới tầng p chính xác là

$$f_p(n) = \sum_{k=0}^{p} a_p(k)\,x(n-k) \qquad a_p(0) = 1 \tag{2.2.12}$$



Hình 2.6: Bộ lọc l- ới tầng-p

Từ (2.2.12) là tổng chập, mối quan hệ biến đổi z là

$$F_p(z) = A_p(z)\,X(z) \tag{2.2.13}$$

hoặc t- ơng đ- ơng

$$A_p(z) = \frac{F_p(z)}{X(z)} = \frac{F_p(z)}{F_0(z)} \tag{2.2.14}$$

Giá trị trung bình bình ph- ơng của lỗi -ớc l-ợng tuyến tính tiến $f_p(n)$ là

$$\xi_p^f = E\left[\left|f_p(n)\right|^2\right] \tag{2.2.15}$$

$$= y_{xx}(0) + 2\,\mathrm{RE}\left[\sum a_p^*(k)\,y_{xx}(k)\right] + \sum_{k=1}^{p}\sum a_p^*(l)\,a_p(k)\,y_{xx}(l-k)$$

ξ_p^f là hàm bậc hai của hệ số - ớc l- ợng và hàm cực tiểu h- ớng đến tập hợp của những ph- ơng trình tuyến tính.

$$y_{xx}(l) = -\sum_{k=1}^{p} a_p(k)\, y_{xx}(l-k), \qquad l = 1,2,\ldots,p \qquad (1.2.16)$$

Đây đ- ợc gọi là *những ph- ơng trình trung bình* cho những hệ số - ớc l- ợng tuyến tính. Cực tiểu trung bình bình ph- ơng của - ớc l- ợng lỗi là tuyệt đối.

$$\min\left[\xi_p^f\right] \equiv E_p^f = y_{xx}(0) + \sum_{k=1}^{p} a_p(k)\, y_{xx}(-k) \qquad (2.2.17)$$

Trong phần tiếp theo chúng ta h- ớng tới sự phát triển cao hơn về vấn đề giá trị - ớc l- ợng của chuỗi thời gian trong h- ớng đối nghịch, khoảng thời gian lùi.

2.2.2 Ước l- ợng tuyến tính lùi

Giả sử chúng ta có chuỗi dữ liệu x(n), x(n-1) . . . x(n-p+1) từ quá trình ngẫu nhiên ổn định và chúng ta cần - ớc l- ợng giá trị x(n-p) của quá trình. Trong tr- ờng hợp này chúng ta tận dụng *- ớc l- ợng tuyến tính lùi đơn b- ớc* của p. Do đó

$$\hat{x}(n-p) = -\sum_{k=0}^{p-1} b_p(k)\, x(n-k) \qquad (2.2.18)$$

Sự chênh lệch giữa giá trị x(n-p) và giá trị - ớc l- ợng $\hat{x}(n-p)$ đ- ợc gọi là *lỗi - ớc l- ợng lùi* kí hiệu $g_p(n)$

$$g_p(n) = x(n-p) + \sum_{k=0}^{p-1} b_p(k)\, x(n-k)$$

$$= \sum_{k=0}^{p} b_p(k)\, x(n-k) \qquad b_p(p) = 1 \qquad (2.2.19)$$

Ước l- ợng tuyến tính lùi có thể thực hiện bằng cấu trúc bộ lọc FIR dạng trực tiếp t- ơng tự cấu trúc biểu diễn trong hình (2.2) hoặc cấu trúc l- ới. Cấu trúc l- ới đ- ợc chỉ ra trong hình (2.6) chỉ ra - ớc l- ợng tuyến tính lùi cũng đảm bảo tốt nh- là - ớc l- ợng tuyến tính tiến. Chứng minh quan điểm này, hãy xét đầu ra của bộ lọc l- ới này từ nhánh thấp nhất. Đầu ra này đ- a ra:

$$g_1(n) = K_1^* x(n) + x(n-1) \qquad (2.2.20)$$

Do đó trọng số hệ số - ớc l- ợng lùi là $b_1(0) = K_1^*$

Trong tầng l- ới hai trong hình (2.5), đầu ra tầng thứ hai từ nhánh cơ bản là

$$g_2(n) = K_2^* f_1(n) + g_1(n-1) \tag{2.2.21}$$

Nếu chúng ta thay thế từ (2.2.6) cho $f_1(n)$ và $g_1(n-1)$, chúng ta đạt đ- ợc

$$g_2(n) = K_2^* x(n) + (K_1^* + K_1 K_2^*) x(n-1) + x(n-2) \tag{2.2.22}$$

Dó đó, trọng số của các hệ số trong - ớc l- ợng tuyến tính lùi là đồng nhất tới các hệ số cho - ớc l- ợng tuyến tính tiến, nh- ng chúng xuất hiện theo thứ tự ng- ợc lại. Do đó chúng ta có

$$b_p(k) = a_p^*(p-k) \qquad k = 0,1, \ldots ,p \tag{2.2.23}$$

Trong miền z, tổng chập trong (2.2.19) trở thành

$$G_p(z) = B_p(z) X(z) \tag{2.2.24}$$

hoặc t- ơng đ- ơng

$$B_p(z) = \frac{G_p(z)}{X(z)} = \frac{G_p(z)}{G_0(z)} \tag{2.2.25}$$

ở đây $B_p(z)$ là kết quả hàm hệ thống của bộ lọc FIR với các hệ số $b_p(k)$.

Từ đó $b_p(k) = a_p^*(p-k)$, $G_p(z)$ có quan hệ tới $A_p(z)$ nh- sau

$$B_p(z) = \sum_{k=0}^{p} b_p(k) z^{-k}$$

$$= \sum_{k=0}^{p} a_p^*(p-k) z^{-k}$$

$$= z^{-p} \sum_{k=0}^{p} a_p^*(k) z^{k}$$

$$= z^{-p} A_p^*(z^{-1}) \tag{2.2.26}$$

Mối quan hệ trong (2.2.26) xác định các điểm không của bộ lọc FIR với hàm hệ thống $B_p(z)$ là nghịch đảo liên hiệp phức của những điểm không của $A_p(z)$. Do đó $B_p(z)$ đ- ợc gọi là nhịch đảo hoặc *đa thức đảo* của $A_p(z)$.

Bây giờ chúng ta đã thiết lập mối quan hệ giữa bộ lọc FIR dạng trực tiếp FIR và bộ lọc FIR dạng l- ới, hãy quay trở lại ph- ơng trình l- ới đệ qui trong (2.2.11) và biến đổi chúng sang miền z. Do vậy chúng ta có

$$F_0(z) = G_0(z) = X(z)$$

$$F_m(z) = F_{m-1}(z) + K_m z^{-1} G_{m-1}(z), \qquad m = 1, 2, \ldots ,p \tag{2.2.27}$$

$G_m(z) = K_m^* F_{m-1}(z) + z^{-1} G_{m-1}(z)$, m = 1, 2, . . . ,p

Nếu chúng ta chia mỗi ph- ơng trình bởi X(z), chúng ta đạt đ- ợc kết quả mong đợi trong công thức

$$A_0(z) = B_0(z) = 1$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \qquad m = 1, 2, \ldots, p \qquad (2.2.28)$$

$$B_m(z) = K_m^* A_{m-1}(z) + z^{-1} B_{m-1}(z), \qquad m = 1, 2, \ldots, p$$

Do đó bộ lọc l- ới đ- ợc miêu tả trong miền z bởi ph- ơng trình ma trận

$$\begin{bmatrix} A_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_m z^{-1} \\ K_m^* & z^{-1} \end{bmatrix} \begin{bmatrix} A_{m-1}(z) \\ B_{m-1}(z) \end{bmatrix} \qquad (2.2.29)$$

Mối quan hệ trong (2.2.28) để $A_m(z)$ và $B_m(z)$ cho phép chúng ta đạt đ- ợc bộ lọc FIR dạng trực tiếp hệ số $a_m(k)$ từ hệ số phản xạ K_m.

Công thức để xác định hệ số bộ lọc $a_p(k)$ đệ qui có thể dễ dàng nhận ra từ mối quan hệ đa thức (2.2.28).Chúng ta có

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \qquad (2.2.30)$$

$$\sum_{k=0}^{m} a_m(k) z^{-k} = \sum_{k=0}^{m-1} a_{m-1}(k) z^{-k} + K_m \sum_{k=0}^{m-1} a_{m-1}^*(m-1-k) z^{-(k+1)}$$

Bằng cách tính hệ số của ph- ơng trình công suất z^{-1} và lấy lại $a_m(0)=1$ cho m=1, 2, . . . , p, chúng ta đạt đ- ợc ph- ơng trình đệ qui mong muốn cho hệ số bộ lọc trong công thức

$$a_m(0) = 1$$

$$a_m(m) = K_m$$

. .

. .

. .

$$a_m(k) = a_{m-1}(k) + K_m a_{m-1}^*(m-k)$$

$$= a_{m-1}(k) + a_m(m) a_{m-1}^*(m-k) \qquad (2.2.31)$$

$$1 \le k \le m-1 \qquad m = 1, 2, \ldots, p$$

Chuyển đổi công thức từ bộ lọc FIR dạng trực tiếp hệ số $a_p(k)$ sang hệ số phản xạ l- ới K_i cũng rất đơn giản. Đối với tầng p chúng ta lập tức đạt đ- ợc hệ số phản xạ $K_p = a_p(p)$. Để tính $K_{p-1} \ldots K_1$, chúng ta cần đa thức $A_m(z)$ cho m=p-1, . . . , 1, từ (2.2.29) chúng ta đ- ợc

$$A_{m-1}(z) = \frac{A_m(z) - K_m B_m(z)}{1-|K_m|^2} \qquad m = p,....,1 \qquad (2.2.32)$$

với đa thức đệ qui lùi đơn b- ớc. Nhờ đó, chúng ta tính toán đ- ợc tất cả đa thức bậc thấp $A_m(z)$ bắt đầu với $A_{p-1}(z)$ và đ- ợc hệ số phản xạ l- ới mong muốn từ mối quan hệ $K_m = a_m(m)$. Chúng ta nhận thấy những thủ tục bậc lớn hơn nh- $|K_m| \neq 1$ cho $m = 1,2,..., p-1$. Từ hồi quy giảm bậc cho đa thức, chúng ta dễ dàng đạt đ- ợc công thức cho mối quan hệ giữa cách tính toán theo hồi quy và trực tiếp K_m, $m = p-1,...,1$. Cho $m = p-1,...,1$ chúng ta có

$$K_m = a_m(m)$$

$$a_{m-1}(k) = \frac{a_m(k) - K_m b_m(k)}{1-|K_m|^2}$$

$$= \frac{a_m(k) - a_m(m)a_m^*(m-k)}{1-|a_m(m)|^2} \qquad (2.2.33)$$

ph- ơng trình này chỉ ra hồi qui trong phần kiểm tra sự ổn định Schur _ Cohn cho đa thức $A_m(z)$.

Nh- ở trên đã chỉ ra, ph- ơng trình hồi qui trong (2.2.33) sẽ bị phá vỡ nếu bất cứ thông số l- ới nào $|K_m| = 1$. Trong tr- ờng hợp này đa thức $A_{m-1}(z)$ có nghiệm nằm trên vòng tròn đơn vị. Nh- vậy nghiệm có thể đ- ợc đánh hệ số ngoài $A_{m-1}(z)$ và quá trình lặp trong (2.2.33) có thể dẫn tới hệ thống có số bậc giảm.

Cuối cùng chúng ta xem xét đến việc giảm đến mức cực tiểu trung bình bình ph- ơng lỗi trong - ớc l- ợng tuyến tính lùi. Lỗi - ớc l- ợng tuyến tính lùi là

$$g_p(n) = x(n-p) + \sum_{k=1}^{p} b_p x(n-k) \qquad (2.2.34)$$

và giá trị trung bình bình ph- ơng của nó là

$$\xi_p^b = E\left[|g_p(n)|^2\right] \qquad (2.2.35)$$

Giá trị tối thiểu của ξ_p^b đối với hệ số - ớc l- ợng sinh ra giống nh- tập hợp ph- ơng trình tuyến tính trong (2.2.16). Do đó cực tiểu trung bình bình ph- ơng lỗi là

$$\min\left[\xi_p^b\right] = E_p^b = E_p^f \qquad (2.2.36)$$

với ph- ơng trình đ- a ra bởi (2.2.17)

2.2.3 Hệ số phản xạ tối - u cho - ớc l- ợng l- ới tiến và lùi

Trong phần (2.2.1) và (2.2.2) chúng ta hiểu đ- ợc tập hợp của những ph- ơng trình tuyến tính, ph- ơng trình này cung cấp hệ số - ớc l- ợng mà tối thiểu hoá giá trị trung bình bình ph- ơng của lỗi - ớc l- ợng. Trong phần này chúng ta xem xét vấn đề của hệ số phản xạ tối - u trong - ớc l- ợng l- ới.

Lỗi - ớc l- ợng tiến trong bộ lọc l- ới đ- ợc biểu diễn là

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1) \tag{2.2.37}$$

Giá trị tối thiểu của $E\left[|f_m(n)|^2\right]$ đối với hệ số phản xạ K_m mang lại kết quả

$$K_m = \frac{-E\left[f_{m-1}(n)g^*_{m-1}(n-1)\right]}{E\left[|g_{m-1}(n-1)|^2\right]} \tag{2.2.38}$$

hoặc t- ơng đ- ơng

$$K_m = \frac{-E\left[f_{m-1}(n)g^*_{m-1}(n-1)\right]}{\sqrt{E^f_{m-1}E^b_{m-1}}} \tag{2.2.39}$$

ở đây

$$E^f_{m-1} = E^b_{m-1} = E\left[|g_{m-1}(n-1)|^2\right] = E\left[|f_{m-1}(n)|^2\right].$$

Chúng ta quan sát lựa chọn tối - u của các hệ số phản xạ trong - ớc l- ợng l- ới là âm của hệ số t- ơng quan chéo giữa lỗi tiến và lùi trong bộ lọc. Vì vậy nó thể hiện từ (2.2.38) mà $|K_m| \le 1$, theo sau đó giá trị trung bình bình ph- ơng cực tiểu của lỗi - ớc l- ợng, lỗi mà có thể biểu diễn bằng đệ qui

$$E^f_m = \left(1-|K_m|^2\right)E^f_{m-1} \tag{2.2.40}$$

là chuỗi giảm bớt tính đơn điệu

2.2.4 Mối quan hệ của quá trình AR tới - ớc l- ợng tuyến tính

Hệ số của quá trình AR(p) có quan hệ mật thiết tới - ớc l- ợng bậc p cho quá trình t- ơng đ- ơng. Xét mối quan hệ này, chúng ta có quá trình AR(p), chuỗi t- ơng quan $\gamma_{xx}(m)$ có mối quan hệ tới hệ số a_k bởi ph- ơng trình Yule _ Walker đ- a ra trong (2.1.19) hoặc (2.1.20). Các ph- ơng trình t- ơng đ- ơng cho - ớc l- ợng bậc p đ- ợc đ- a ra bởi (2.2.16) và (2.2.17).

So sánh trực tiếp hai tập hợp của mối quan hệ này chúng ta có mối quan hệ t- ơng ứng tỉ lệ một – một giữa hệ số a_k của quá trình AR(p) và hệ số - ớc l- ợng $a_p(k)$ của - ớc l- ợng bậc thứ p. Trong thực tế, nếu sau quá trình x(n) là AR(p), hệ số - ớc l- ợng của - ớc l- ợng bậc thứ p sẽ đồng nhất tới σ_w^2, ph- ơng sai của quá trình nhiễu trắng. Trong tr- ờng hợp này, bộ lọc - ớc l- ợng lỗi là bộ lọc nhiễu trắng, bộ lọc mà sinh ra chuỗi nhiễu trắng $w(n)$

2.3 GIẢI CÁC PH□ ƠNG TRÌNH CHUẨN TẮC

Trong phần tr- ớc chúng ta có đ- ợc tối thiểu hoá giá trị trung bình bình ph- ơng của kết quả lỗi - ớc l- ợng tiến trong tập hợp ph- ơng trình tuyến tính cho hệ số - ớc l- ợng đ- a ra bởi (2.2.16), ph- ơng trình này gọi là *ph- ơng trình chuẩn tắc*, có thể biểu diễn rõ ràng trong công thức:

$$\sum_{k=0}^{p} a_p(k)\gamma_{xx}(l-k) = 0, \qquad l = 1,2,..., p. \qquad a_p(0) = 1 \tag{2.3.1}$$

Kết quả tối thiểu MSE (MMSE) đ- a ra bởi (2.1.17). Nếu chúng ta có thêm yếu tố (2.2.17) tới ph- ơng trình chuẩn tắc đ- a ra bởi (2.3.1), chúng ta đạt đ- ợc tập hợp của *ph- ơng trình chuẩn tắc gia tố*, ph- ơng trình này có thể biểu diễn là

$$\sum_{k=0}^{p} a_p(k)\gamma_{xx}(l-k) = \begin{cases} E_p^f & l = 0 \\ 0 & l = 1,2,..., p \end{cases} \tag{2.3.2}$$

Chúng ta cũng chú ý rằng nếu xử lý ngẫu nhiên là xử lý AR(p) thì MMSE là $E_p^f = \sigma_w^2$

Trong phần này chúng ta miêu tả hai thuật toán tính toán hiệu quả cho cách giải ph- ơng trình chuẩn tắc. Thuật toán thứ nhất, có nguồn gốc từ Levinson _ Durbin. Thuật toán này phù hợp cho xử lý chuỗi và có tính toán phức tạp của $0(p^2)$. Thuật toán thứ hai, có nguồn gốc từ Schur (1917) cũng tính toán hệ số phản xạ trong $0(p^2)$ nh- ng với xử lý song song việc tính toán có thể thực hiện đ- ợc trong thời gian 0(p). Khai thác cả hai thuật toán Toeplits vốn có tính chất đối xứng trong ma trận t- ơng quan. Chúng ta hãy bắt đầu miêu tả thuật toán levinson _ Durbin.

2.3.1 Thật toán Levinson _ Durbin

Thuật toán levinson _ Durbin là thuật toán tính toán hiệu quả cho kết quả ph- ơng trình chuẩn tắc trong (2.3.1) cho hệ số - ớc l- ợng. Khai thác thuật toán cho đặc tính đối xứng trong ma trận t- ơng quan

$$\Gamma_p = \begin{bmatrix} y_{xx}(0) & y_{xx}^*(1) & \cdots & y_{xx}^*(p-1) \\ y_{xx}(1) & y_{xx}(0) & \cdots & y_{xx}^*(p-2) \\ \vdots & \vdots & & \\ y_{xx}(p-1) & y_{xx}(p-2) & \cdots & y_{xx}(0) \end{bmatrix} \tag{2.3.3}$$

Chú ý rằng $\Gamma_p(i,j) = \Gamma_p(i-j)$, cho nên ma trận t- ơng quan là ma trận Toeplitz. Do đó $\Gamma_p(i,j) = \Gamma_p^*(j,i)$, ma trận cũng là ma trận Hermitian.

Chìa khoá để giải đáp cho ph- ơng pháp Levinson _ Durbin, ph- ơng pháp mà khai thác đ- ợc tính chất Toeplitz của ma trận là xuất phát từ đệ qui. Bắt đầu với - ớc l- ợng của loại m=1 (hệ số 1) và tăng bậc đệ qui lên, sử dụng kết quả của bậc thấp để đạt đ- ợc kết quả của bậc tiếp theo. Do đó kết quả - ớc l- ợng bậc đầu tiên đạt đ- ợc bởi kết quả (2.3.1) là

$$a_1(1) = -\frac{y_{xx}(1)}{y_{xx}(0)} \tag{2.3.4}$$

và kết quả MMSE là

$$\begin{aligned} E_1^f &= y_{xx}(0) + a_1(1)\, y_{xx}(-1) \\ &= y_{xx}(0)\left[1 - |a_1(1)|^2\right] \end{aligned} \tag{2.3.5}$$

Chú ý $a_1(1) = K_1$, là hệ số phản xạ đầu tiên của bộ lọc l- ới

B- ớc tiếp theo là kết quả cho hệ số $a_2(1)$ và $a_2(2)$ của - ớc l- ợng loại hai và biểu diễn kết quả trong giới hạn của $a_1(1)$. Hai ph- ơng trình đạt đ- ợc từ (2.3.1) là

$$\begin{aligned} a_2(1)\, y_{xx}(0) + a_2(2)\, y_{xx}^*(1) &= -y_{xx}(1) \\ a_2(1)\, y_{xx}(1) + a_2(2)\, y_{xx}(0) &= -y_{xx}(2) \end{aligned} \tag{2.3.6}$$

Bằng cách sử dụng kết quả trong (2.3.4) rút gọn $y_{xx}(1)$, chúng ta đạt đ- ợc kết quả

$$\begin{aligned} a_2(2) &= -\frac{y_{xx}(2) + a_1(1)\, y_{xx}(1)}{y_{xx}(0)\left[1 - |a_1(1)|^2\right]} \\ &= -\frac{y_{xx}(2) + a_1(1)\, y_{xx}(1)}{E_1^f} \end{aligned} \tag{2.3.7}$$

$a_2(1) = a_1(1) + a_2(2)a_1^*(1)$

Do đó chúng ta đạt đ- ợc hệ số của - ớc l- ợng bậc hai. Lại lần nữa chúng ta chú ý rằng $a_2(2) = K_2$, là hệ số phản xạ thứ hai trong bộ lọc l- ới.

Tiếp tục thực hiện, chúng ta có thể đạt đ- ợc hệ số của bậc thứ m trong giới hạn của hệ số - ớc l- ợng bậc (m-1). Do đó chúng ta có thể viết hệ số vector a_m nh- tổng của hai vector, cụ thể là

$$a_m = \begin{bmatrix} a_m(1) \\ a_m(2) \\ \vdots \\ a_m(m) \end{bmatrix} = \begin{bmatrix} a_{m-1} \\ \\ \ldots \\ 0 \end{bmatrix} + \begin{bmatrix} d_{m-1} \\ \\ \ldots \\ K_m \end{bmatrix} \tag{2.3.8}$$

ở đây a_{m-1} là hệ số - ớc l- ợng vector của - ớc l- ợng bậc (m-1), vector d_{m-1} và giá trị vô h- ớng K_m đã đ- ợc xác định. Hệ số $m \times n$ của ma trận t- ơng quan Γ_{xx} là

$$\Gamma_m = \begin{bmatrix} \Gamma_{m-1} & y_{m-1}^{b*} \\ y_{m-1}^{bt} & y_{xx}(0) \end{bmatrix} \tag{2.3.9}$$

ở đây $y_{m-1}^{bt} = [y_{xx}(m-1) \quad y_{xx}(m-2) \quad \cdots \quad y_{xx}(1)] = (y_{m-1}^{b})^t$ dấu hoa thị (*) biểu thị hàm liên hợp phức và y_m^t biểu thị sự hoán vị của y_m. Chữ b ở bên trên y_{m-1} biểu thị vector $y_{m-1}^t = [y_{xx}(1) \quad y_{xx}(2) \quad \cdots \quad y_{xx}(m-1)]$ với thành phần lấy trong bậc đảo ng- ợc.

Kết quả ph- ơng trình $\Gamma_m a_m = -y_m$ có thể biểu diễn nh-

$$\begin{bmatrix} \Gamma_{m-1} & y_{m-1}^{*} \\ y_{m-1}^{bt} & y_{xx}(0) \end{bmatrix} \left(\begin{bmatrix} a_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} d_{m-1} \\ K_m \end{bmatrix} \right) = - \begin{bmatrix} y_{m-1} \\ y_{xx}(m) \end{bmatrix} \tag{2.3.10}$$

Đây là chìa khoá cho thuật toán Levinson _ Durbin. Từ (2.3.10) chúng ta đạt đ- ợc hai ph- ơng trình

$$\Gamma_{m-1}^{'} a_{m-1} + \Gamma_{m-1} d_{m-1} + K_m y_{m-1}^{b*} = y_{m-1} \tag{2.3.11}$$

$$y_{m-1}^{bt} a_{m-1} + y_{m-1}^{bt} d_{m-1} + K_m y_{xx}(0) = -y_{xx}(m) \tag{2.3.12}$$

Do đó $r_{m-1} a_{m-1} = -y_{m-1}$, (2.3.11) sinh ra kết quả

$$d_{m-1} = -K_m r_{m-1}^{-1} y_{m-1}^{b*} \tag{2.3.13}$$

Nh- ng y_{m-1}^{b*} chỉ là y_{m-1} với các thành phần lấy trong bậc đảo ng- ợc và liên hợp phức. Bởi vậy, kết quả trong (2.5.13) đơn giản là

$$d_{m-1} = K_m a_{m-1}^{b*} = K_m \begin{bmatrix} a_{m-1}^{*}(n-1) \\ a_{m-1}^{*}(n-2) \\ \vdots \\ a_{m-1}^{*}(1) \end{bmatrix} \qquad (2.3.14)$$

Ph- ơng trình vô h- ớng (2.3.12) hiện tại có thể dùng để giải K_m. Nếu rút gọn d_{m-1} trong (2.3.12) bằng cách dùng (2.3.14) ta đ- ợc

$$K_m\left[y_{xx}(0) + y_{m-1}^{bt} a_{m-1}^{b*}\right] + y_{m-1}^{bt} a_{m-1} = -y_{xx}(n)$$

do đó

$$K_m = -\frac{y_{xx}(n) + y_{m-1}^{bt} a_{m-1}}{y_{xx}(0) + y_{m-1}^{bt} a_{m-1}^{b*}} \qquad (2.3.15)$$

Vì vậy, bằng cách thay thế kết quả trong (2.3.14) và (2.3.15) thành (2.3.8), chúng ta đạt đ- ợc yêu cầu đệ qui cho hệ số - ớc l- ợng trong thuật toán Levinson _ Durbin là

$$a_m(n) = K_m = -\frac{y_{xx}(n) + y_{m-1}^{bt} a_{m-1}}{y_{xx}(0) + y_{m-1}^{bt} a_{m-1}^{b*}} = -\frac{y_{xx}(n) + y_{m-1}^{bt} a_{m-1}}{E_m^f} \qquad (2.3.16)$$

$$\begin{aligned} a_m(k) &= a_{m-1}(k) + K_m a_{m-1}^{*}(n-k) \\ &= a_{m-1}(k) + a_m a_{m-1}^{*}(n-k) \qquad (2.3.17) \\ k &= 1,2,\ldots,m-1 \\ m &= 1,2,\ldots,p \end{aligned}$$

L- u ý, mối quan hệ đệ qui trong (2.3.17) là đồng nhất với mối quan hệ đệ qui trong (2.2.31) cho hệ số - ớc l- ợng, hệ số mà đạt đ- ợc từ đa thức $A_m(z)$ và $B_m(z)$. Hơn nữa, K_m là hệ số phản xạ trong bậc thứ m của - ớc l- ợng l- ới. Sự triển khai này chứng minh rõ ràng rằng thuật toán Levinson_ Durbin sinh ra hệ số phản xạ cho - ớc l- ợng l- ới tối - u nh- sự - ớc l- ợng FIR dạng trực tiếp.

Cuối cùng, hãy xác định biểu thức cho MMSE - ớc l- ợng bậc thứ m, chúng ta có:

$$\begin{aligned} E_m^f &= y_{xx}(0) + \sum_{k=1}^{m} a_m(k) y_{xx}(-k) \\ &= y_{xx}(0) + \sum_{k=1}^{m}\left[a_{m-1}(k) + a_m(n) a_{m-1}^{*}(n-k)\right] y_{xx}(-k) \qquad (2.3.18) \\ &= E_{m-1}^f\left[1 - |a_m(n)|^2\right] = E_{m-1}^f\left(1 - |K_m|^2\right), \qquad m = 1,2,\ldots,p \end{aligned}$$

Ở đây $E_0^f = y_{xx}(0)$. Vì vậy những hệ số phản xạ phù hợp với thuộc tính mà $|K_m| \le 1$, MMSE cho chuỗi của - ớc l- ợng thoả mãn điều kiện

$$E_0^f \ge E_1^f \ge E_2^f \ge .... \ge E_p^f \quad (2.3.19)$$

Kết luận này bắt nguồn từ thuật toán Levinson _ Durbin kết quả ph- ơng trình tuyến tính $\Gamma_m a_m = -y_m$ (m=0,1,…, p). Chúng ta quan sát ph- ơng trình tuyến tính có thuộc tính đặc biệt là vector ở phía bên phải xuất hiện nh- một vector trong Γ_m. Trong các tr- ờng hợp thông th- ờng khác vector phía bên phải là một vài vector khác, gọi là C_m, tập hợp ph- ơng trình tuyến tính có thể giải hồi qui bằng cách tạo ph- ơng trình đệ qui thứ hai tới kết quả ph- ơng trình tuyến tính chung $\Gamma_m b_m = C_m$. Kết qủa là thuật toán Levinson _ Durbin tổng quát.

Đệ qui Levinson _ Durbin đ- a ra bởi (2.3.17) yêu cầu O(m) tăng lên và thêm vào từ tầng m tới tầng (m+1). Vì vậy, để cho tầng p, sẽ phải tính qua các bậc 1+2+3+….+p = (p+1)/2 hoặc thuật toán $O(p^2)$ giải hệ số bộ lọc - ớc l- ợng hoặc hệ số phản xạ, so sánh với thuật toán $O(p^3)$ nếu chúng không khai thác tính chất Toeplitz của ma trận t- ơng quan.

Nếu thuật toán levinson _ Durbin đ- ợc thực hiện trên chuỗi nối tiếp hoặc bộ xử lý tín hiệu nối tiếp, đòi hỏi thời gian tính toán trên bậc của $O(p^2)$ đơn vị thời gian. Theo h- ớng khác, nếu quá trình xử lý đ- ợc thực hiện song song sử dụng bằng nhiều bộ xử lý cần thiết khai thác hết sự t- ơng đ- ơng trong thật toán, phép nhân cũng nh- là phép cộng khi yêu cầu tính (2.3.17). Vì thế, tính toán có thể thực hiện trong O(p) đơn vị thời gian. Tuy nhiên việc tính toán trong (2.3.16) cho hệ số phản xạ tốn thêm thời gian. Dĩ nhiên, tích vô h- ớng này bao gồm vector a_{m-1} và y_{m-1}^b có thể tính toán đồng thời bởi việc xử lý song song. Tuy nhiên phép cộng này không thể làm đồng thời nh- ng thay vào đó, yêu cầu O(log p) đơn vị thời gian. Do đó các tính toán trong thuật toán Levinson _ Durbin, khi thực hiện bằng p bộ xử lý song song có thể hoàn thành trong thời gian O(p log p).

2.3.2. Thuật toán Schur

Thuật toán Schur đ- ợc liên hệ với việc kiểm tra đệ quy cho xác định phân tích d- ơng của ma trận t- ơng quan. Cụ thể hãy xem xét ma trận t- ơng đ- ơng Γ_{p+1} liên kết thêm với ph- ơng trình chuẩn tắc đ- a ra bởi (2.3.2). Từ các thành phần của ma trận này chúng ta tạo hàm:

$$R_0(Z) = \frac{Y_{xx}(1).Z^{-1} + Y_{xx}(2).Z^{-2} + ... + Y_{xx}(p).Z^{-p}}{Y_{xx}(0) + Y_{xx}(1).Z^{-1} + ... + Y_{xx}(p).Z^{-p}} \qquad (2.3.20)$$

Và chuỗi của hàm $R_m(Z)$ đ- ợc định nghĩa đệ quy là:

$$R_m(Z) = \frac{R_{m-1}(Z) - R_{m-1}(\infty)}{Z^{-1}\left[1 - R^*_{m-1}(\infty).R_{m-1}.(Z)\right]} \qquad m= 1, 2,\ldots \qquad (2.3.21)$$

Phát biểu định lý Schur's, điều kiện cần và đủ của định lý cho ma trận t- ơng quan xác định d- ơng là $|R_m(\infty)|<1$ cho m= 1,2,…,p

Hãy chứng minh rằng điều kiện cho xác định d- ơng của ma trận tự t- ơng quan Γ_{p+1} là t- ơng đ- ơng với điều kiện hệ số phản xạ trong bộ lọc l- ới t- ơng đ- ơng thoả mãn điều kiện $|K_m|<1$, m=1,2,…,p.

Đầu tiên chúng ta chú ý rằng $R_0(\infty) = 0$. Sau đó từ (2.3.21) chúng ta có

$$R_1(Z) = \frac{Y_{xx}(1) + Y_{xx}(2).Z^{-1} + ... + Y_{xx}(p).Z^{-p+1}}{Y_{xx}(0) + Y_{xx}(1).Z^{-1} + ... + Y_{xx}(p).Z^{-p}} \qquad (2.3.22)$$

Do đó $R_1(\infty) = Y_{xx}(1)/Y_{xx}(0)$ ta đ- ợc $R_1(\infty) = -K_1$

Thứ hai, ta tính toán $R_2(Z)$ phụ thuộc vào (2.3.21) và đánh giá kết quả tại $Z=\infty$. Do đó ta đ- ợc

$$R_2(\infty) = \frac{Y_{xx}(2) + K_1 Y_{xx}(1)}{Y_{xx}(0).(1-|K_1|^2)}$$

Mặt khác, ta lại có $R_2(\infty) = -K_2$. Bằng cách tiếp tục khai triển, chúng ta tìm thấy $R_m(\infty) = -K_m$ cho m= 1,2,…,p. Dó đó điều kiện $|R_m(\infty)|<1$ cho m=1,2,…,p là đồng nhất với điều kiện $|K_m|<1$ cho m= 1,2,…,p và đảm bảo định nghĩa rõ ràng của ma trận t- ơng đ- ơng Γ_{p+1}.

Do hệ số phản xạ có thể tính đ- ợc từ chuỗi của hàm $R_m(Z)$, m=1,2,…p, chúng ta có cách khác để tìm lời giải cho ph- ơng trình chính tắc. Chúng ta gọi cách này là thuật toán Schur.

Thuật toán Schur: đầu tiên hãy viết lại $R_m(Z)$

$$R_m(Z) = \frac{P_m(Z)}{Q_m(Z)} \qquad m= 1,2,\ldots p \qquad (2.3.23)$$

ở đây: $P_0(z) = Y_{xx}(1)Z^{-1} + Y_{xx}(2)Z^{-2} + \ldots + Y_{xx}(p)Z^{-p}$

$$Q_0(z) = Y_{xx}(0) + Y_{xx}(1)Z^{-1} + \ldots + Y_{xx}(p)Z^{-p} \qquad (2.3.24)$$

Do đó: $K_0 = 0$ và $K_m = -R_m(\infty)$ cho m= 1,2,…p, phương trình đệ quy (2.3.21) đ- a đến những ph- ơng trình đệ quy tiếp theo cho những đa thức $P_m(z)$ và $Q_m(z)$

$$\begin{bmatrix} P_m(z) \\ Q_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_{m-1} \\ K^*{}_{m-1}Z^{-1} & Z^{-1} \end{bmatrix} \begin{bmatrix} P_{m-1}(z) \\ Q_{m-1}(z) \end{bmatrix}, \text{ m= 1,2,…,p} \qquad (2.3.25)$$

Do đó chúng ta có:

$$P_1(z) = P_0(z) = Y_{xx}(1)Z^{-1} + Y_{xx}(2)Z^{-2} + \ldots + Y_{xx}(p)Z^{-p}$$

$$Q_1(z) = Z^{-1}Q_0(z) = Y_{xx}(0)Z^{-1} + Y_{xx}(1)Z^{-2} + \ldots + Y_{xx}(p-1)Z^{-p}$$

và

$$K_1 = -\left.\frac{P_1(z)}{Q_1(z)}\right|_{z=\infty} = -\frac{Y_{xx}(1)}{Y_{xx}(0)}$$

Tiếp theo hệ số phản xạ K_2 tính đ- ợc bởi việc xác định $P_2(z)$ và $Q_2(z)$ từ (2.3.25), chia $P_2(z)$ bởi $Q_2(z)$ và đánh giá kết quả tại $z=\infty$. Vì vậy chúng ta tìm đ- ợc

$$P_2 = P_1(z) + K_1Q_1(z) = [Y_{xx}(2) + K_1Y_{xx}(1)]Z^{-2} + \ldots + [Y_{xx}(p) + K_1Y_{xx}(p-1)]Z^{-p}$$

$$Q_2(z) = Z^{-1}[Q_1(z) + K_1^*P_1(z)]$$

$$= [Y_{xx}(0) + K_1^*Y_{xx}(1)]Z^{-2} + \ldots + [Y_{xx}(p-2) + K_1^*Y_{xx}(p-1)]Z^{-p}$$

Do đó, chúng ta thấy rằng ph- ơng trình đệ quy trong (2.3.25) t- ơng đ- ơng tới (2.3.21).

Căn cứ vào những mối quan hệ này, thuật toán Schur đ- ợc miêu tả bởi ph- ơng trình đệ qui sau

Bắt đầu Tạo ma trận sinh $2 \times (p+1)$

$$G_0 = \begin{bmatrix} 0 & y_{xx}(1) & y_{xx}(2) & \cdots & y_{xx}(p) \\ y_{xx}(0) & y_{xx}(1) & y_{xx}(2) & \cdots & y_{xx}(p) \end{bmatrix} \qquad (2.3.29)$$

ở đây các thành phần của hàng đầu tiên là những hệ số của $P_0(z)$ và những thành phần của hàng thứ hai là hệ số của $Q_0(z)$.

B- ớc 1. Dịch hàng thứ hai của ma trận sinh về bên phải 1 vị trí, bỏ thành phần cuối của hàng này, thêm số 0 vào vị trí khuyết ở đầu hàng. Do đó chúng đạt đ- ợc ma trận sinh mới

$$G_1 = \begin{bmatrix} 0 & y_{xx}(1) & y_{xx}(2) & \cdots & y_{xx}(p) \\ 0 & y_{xx}(0) & y_{xx}(1) & \cdots & y_{xx}(p-1) \end{bmatrix} \qquad (2.3.30)$$

(Nghịch đảo) tỷ số của các thành phần trong cột thứ hai sinh ra hệ số phản xạ

$$K_1 = -y_{xx}(1)/y_{xx}(0)$$

B- ớc2. Nhân ma trận sinh với ma trận 2×2

$$V_1 = \begin{bmatrix} 1 & K_1 \\ K_1^* & 1 \end{bmatrix}$$

ta đ- ợc

$$V_1G_1 = \begin{bmatrix} 0 & 0 & y_{xx}(2)+K_1y_{xx}(1) & \cdots & y_{xx}(p)+K_1y_{xx}(p-1) \\ 0 & y_{xx}(0)+K_1^*y_{xx}(1) & \ldots\ldots\ldots\ldots\ldots\ldots & \cdots & y_{xx}(p-1)+K_1^*y_{xx}(p) \end{bmatrix} \qquad (2.3.32)$$

B- ớc 3. Dịch hàng thứ hai của G_1V_1 một vị trí về bên phải và do đó tạo đ- ợc ma trận sinh mới.

$$G_2 = \begin{bmatrix} 0 & 0 & y_{xx}(2)+K_1y_{xx}(1) & \cdots & y_{xx}(p)+K_1y_{xx}(p-1) \\ 0 & 0 & y_{xx}(0)+K_1^*y_{xx}(1) & \ldots & y_{xx}(p-2)+K_1^*y_{xx}(p-1) \end{bmatrix} \qquad (2.3.33)$$

Tỉ lệ nghịch của các thành phần trong cột thứ ba của G_2 sinh ra K_2.

B- ớc thứ 2 và thứ 3 lặp lại tr- ớc khi chúng ta tính mọi hệ số phản xạ p. Nhìn chung, ma trận 2×2 trong b- ớc m là

$$V_m = \begin{bmatrix} 1 & K_1 \\ K_1^* & 1 \end{bmatrix}$$

và nhân V_m với G_m sinh ra V_mG_m. Trong b- ớc ba chúng ta dịch hàng thứ hai của V_mG_m một vị trí về bên phải đ- ợc ma trận sinh mới G_{m+1}

Chúng ta thấy rằng phép toán dịch hàng thứ 2 trong vòng lặp t- ơng đ- ơng tới việc nhân bởi hoạt động trễ z^{-1} trong ph- ơng trình đệ qui thứ hai trong (2.3.25). Chúng ta cũng chú ý rằng phép chia của đa thức $P_m(z)$ bởi đa thức $Q_m(z)$ và - ớc l- ợng th- ơng số tại $z = \infty$ là t- ơng đ- ơng với phép chia các thành phần trong cột (m+1) của G_m. Sự tính toán hệ số phản xạ p có thể hoàn thành bằng cách dùng xử lý song song trong đơn vị thời gian 0(p). Sau đó chúng ta miêu tả kiến trúc đ- ờng ống cho việc thực hiện tính toán này.

Một cách minh họa khác mối quan hệ của thuật toán Schur với thuật toán Levinson - Durbin và - ớc l- ợng l- ới t- ơng ứng là xác định rõ đầu ra của bộ lọc l- ới đạt đ- ợc khi chuỗi đầu vào là chuỗi t- ơng quan $\{y_{xx}(m), m = 0,1,\ldots\}$.

Vì đầu vào đầu tiên tới bộ lọc l- ới là $y_{xx}(0)$, đầu vào thứ hai là $y_{xx}(1)$, và t- ơng tự các đầu vào tiếp theo [e..... $f_0(n) = y_{xx}(n)$]. Sau khi trễ trong tầng đầu chúng ta có $g_0(n-1) = y_{xx}(n-1)$, do đó cho n=1, tỉ số $f_0(1)/g_0(0) = y_{xx}(1)/y_{xx}(0)$ tỉ số này là nghịch đảo của hệ số phản xạ K_1. Cách khác chúng ta có thể biểu diễn mối quan hệ này là

$$f_0(1) + K_1 g_1(0) = y_{xx}(1) + K_1 y_{xx}(0) = 0$$

Hơn nữa, $g_0(0) = y_{xx}(0) = E_0^f$. Tại thời điểm n=2, đầu ra tầng thứ hai, theo (2.2.11),

$$f_1(2) = f_0(2) + K_1 g_0(1) = y_{xx}(2) + y_{xx}(1)$$

và sau một đơn vị của trễ trong tầng thứ hai, chúng ta có

$$g_1(1) = K_1^* f_0(1) + g_0(0) = K_1^* y_{xx}(1) + y_{xx}(0)$$

Bây giờ tỉ số $f_1(2)/g_1(1)$ là

$$\frac{f_1(2)}{g_1(1)} = \frac{y_{xx}(2) + K_1 y_{xx}(1)}{y_{xx}(0) + K_1^* y_{xx}(1)} = \frac{y_{xx}(2) + K_1 y_{xx}(1)}{E_1^f} = -K_2$$

Do đó $f_1(2) + K_2 g_1(1) = 0$

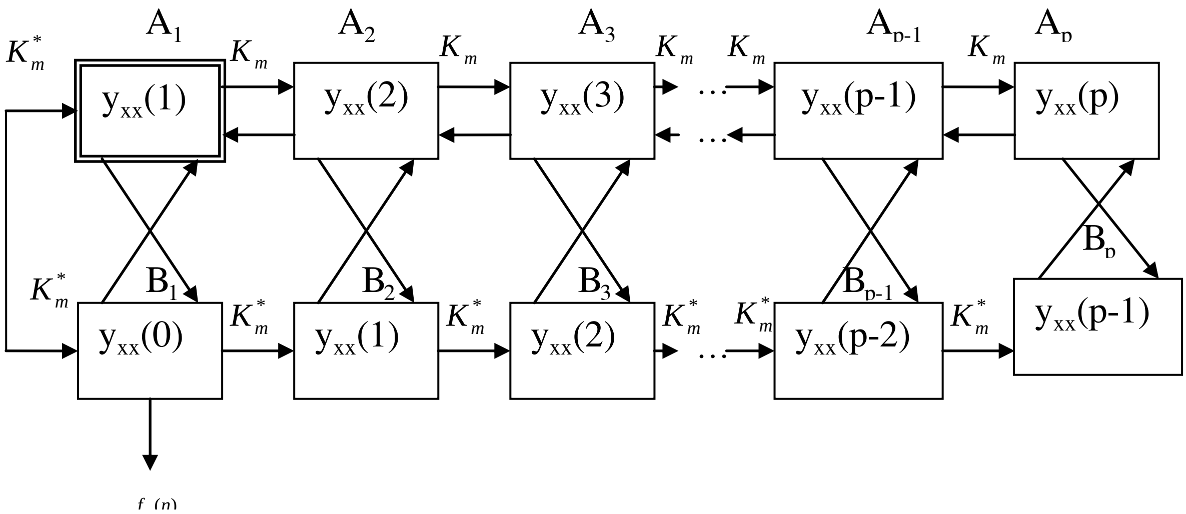
$$g_1(1) = E_1^f$$

Tiếp tục tính theo cách này, chúng ta thấy rằng tại đầu vào của tầng l- ới thứ m, tỉ số $f_{m-1}(m)/g_{m-1}(m-1) = -K_m$ và $g_{m-1}(m-1) = E_{m-1}^f$. Do đó, hệ số bộ lọc l- ới đạt đ- ợc từ thuật toán Levinson là chính xác tới hệ số đạt đ- ợc trong thuật toán Schur. Hơn nữa, cấu trúc bộ lọc l- ới cung cấp một cách thức tính toán hệ số phản xạ trong - ớc l- ợng l- ới.

Kiến trúc đ- ờng ống cho việc thực hiện thuật toán Schur. Kung và Hu (1983) phát triển bộ xử lý dạng l- ới đ- ờng ống cho việc thực hiện thuật toán Schur. Xử lý bao gồm một giai đoạn của các tầng kiểu l- ới p, ở đó mỗi tầng gồm hai thành phần xử lý (PEs), PEs trên, bao hàm A_1, A_2, . . . , A_p và PEs d- ới bao hàm B_1, B_2, . . . , B_p. Nh- nhìn trong hình (2.7). PE chỉ rõ A_1 đ- ợc phân chia nhiệm vụ cho việc thực hiện những phép chia, PEs còn lại thực hiện một phép nhân và một phép cộng cho mỗi lần lặp (một chu kỳ đo).

Ban đầu, PEs trên tải các thành phần của hàng đầu của ma trận sinh ra G_0, nh- chứng minh trong hình (2.7). PEs d- ới tải các thành phần của hàng thứ hai của ma trận sinh ra G_0. Việc xử lý tính toán bắt đầu với phép chia PE,

A_1, phép chia này tính toán đ- ợc hệ số phản xạ đầu tiên là $K_1 = -y_{xx}(1)/y_{xx}(0)$. Giá trị của K_1 đ- ợc gửi đồng thời tới mọi PEs trong nhánh trên và nhánh d- ới.



Hình 2.7 : Xử lý song song đ- ờng ống cho tính toán hệ số phản xạ

B- ớc thứ hai trong việc tính toán cập nhập nội dung của tất cả phần tử xử lý cùng một lúc. Nội dung của PEs thấp và cao đ- ợc cập nhật nh- sau:

PE A_m : $A_m \leftarrow A_m + K_1 B_{m'}$ m = 2, 3, . . . ,p

PE B_m : $B_m \leftarrow B_m + K_1^* A_{m'}$ m = 1, 2, . . . ,p

B- ớc ba bao gồm dịch nội dung của PEs trên một vị trí về bên trái. Do đó chúng ta có

PE A_m : $A_{m-1} \leftarrow A_{m'}$ m = 2, 3, . . . ,p

Tại điểm PE này A_1 bao gồm $y_{xx}(2) + K_1^* y_{xx}(1)$ trong khi PE B_1 bao gồm $y_{xx}(0) + K_1^* y_{xx}(1)$. Do đó quá trình A_1 sẵn sàng bắt đầu qui trình thứ hai bằng cách tính toán hệ số phản xạ thứ hai với phép chia A_1/B_1 đ- ợc lặp lại trong khi mọi hệ số phản xạ p đ- ợc tính. Chú ý rằng PE B_1 cung cấp lỗi trung bình bình ph- ơng cực tiểu E_m^f cho mỗi b- ớc lặp.

Nếu τ_d bao hàm thời gian cho PE A_1 thực hiện phép chia (hoàn thành) và τ_{ma} là thời gian yêu cầu cho việc thực hiện một phép nhân (phức) và phép cộng. Thời gian yêu cầu cho việc tính toán mọi hệ số phản xạ p là $p(\tau_d + \tau_{ma})$ cho thuật toán Schur.

2.4 CÁC THUỘC TÍNH CỦA BỘ LỌC LỖI □ ỚC L□ ỢNG TUYẾN TÍNH

Những bộ lọc - ớc l- ợng tuyến tính có nhiều thuộc tính quan trọng mà chúng ta sẽ đề cập đến sau đây, ban đầu là với chứng minh rằng bộ lọc lỗi - ớc l- ợng tiến là pha cực tiểu.

Thuộc tính pha cực tiểu của bộ lọc lỗi □ớc l□ợng tiến.

Chúng ta đã chứng minh những hệ số phản xạ K_1 là những hệ số t- ơng quan, và do đó $|K_1| \le 1$ với mọi i. Điều kiện này và mối quan hệ $E_m^f = (1-|K_m|^2)E_{m-1}^f$ có thể sử dụng để xem những điểm không của bộ lọc lỗi - ớc l- ợng nằm hoàn toàn bên trong vòng tròn đơn vị hay là chúng ở bên trên vòng tròn đơn vị.

Đầu tiên, chúng ta xét nếu $E_p^f > 0$, các điểm không $|z_1| < 1$ với mọi i. Chứng minh bằng ph- ơng pháp qui nạp. Rõ ràng rằng, cho p =1, hàm hệ thống cho bộ lọc lỗi - ớc l- ợng là

$A_1(z) = 1+K_1z^{-1}$

Do đó $z_1 = -K_1$ và $E_1^f = (1-|K_1|^2)E_0^f > 0$. Bây giờ giả sử rằng giả thiết là đúng cho p-1. Sau đó nếu z_1 là nghiệm của $A_p(z)$ chúng ta có từ (2.2.26) và (2.2.28)

$$A_p(z_1) = A_{p-1}(z_1) + K_p z_1^{-1} B_{p-1}(z_1)$$

$$= A_{p-1}(z_1) + K_p z_1^{-p} A_{p-1}^*\left(\frac{1}{z_1}\right) = 0$$

Do đó

$$\frac{1}{K_p} = \frac{z_1^{-p} A_{p-1}^*\left(\frac{1}{z_1}\right)}{A_{p-1}(z_1)} = Q(z_1)$$

Chúng ta chú ý rằng hàm Q(z) là thông tất. Thông th- ờng, hàm thông tất có công thức

$$P(z) = \prod_{k=1}^{N} \frac{z z_{p-1}^* + 1}{z + z_k} \qquad |z_k| < 1$$

thoả mãn tính chất $|P(z)|>1$ cho $|z|<1$, $|P(z)|=1$ cho $|z|=1$ và $|P(z)|<1$ cho $|z|>1$. Do đó Q(z) = - P(z)/z, tiếp theo $|z_1|<1$ nếu $|Q(z)|>1$. Rõ ràng rằng, đây là tr- ờng hợp $Q(z_1)=1/K_p$ và $E_p^f>0$.

Cách khác, giả sử $E_p^f>0$ và do $E_p^f=0$. Trong tr- ờng hợp này $|K_p|=1$ và $|Q(z_1)|=1$. Do đó MMSE là 0, qui trình ngẫu nhiên x(n) đ- ợc gọi là có khả năng - ớc l- ợng hoặc là xác định tr- ớc. Cụ thể, quá trình ngẫu nhiên hoàn toàn hàm sin của công thức

$$x(n)=\sum_{k=1}^{M}\alpha_k e^{j(\omega k+\theta_k)} \tag{2.4.6}$$

ở đây pha θ_k đã đ- ợc thống kê độc lập và phân bố đều trên $(0,2\pi)$, có t- ơng quan

$$y_{xx}(n)=\sum_{k=1}^{M}\alpha_k^2 e^{jm\omega k}$$

và mật độ phổ đầu vào

$$\Gamma_{xx}(f)=\sum_{k=1}^{M}\alpha_k^2\delta(f-f_k), \qquad f_k=\frac{\omega_k}{2\pi} \tag{2.4.7}$$

Quá trình này có thể - ớc l- ợng tr- ớc với giá trị - ớc l- ợng của bậc $p\geq M$.

Để chứng minh tính hợp lý của quá trình trên, xét giá trị này từ đầu đến cuối của bộ lọc - ớc l- ợng lỗi bậc $p\geq M$. MSE tại đầu ra của bộ lọc này là

$$\xi_p^f=\int_{-1/2}^{1/2}\Gamma_{xx}(f)|A_p(f)|^2 df$$

$$=\int_{-1/2}^{1/2}\left[\sum_{k=1}^{M}\alpha_k^2\delta(f-f_k)\right]|A_p(f)|^2 df$$

$$=\sum_{k=1}^{M}\alpha_k^2|A_p(f_k)|^2 \tag{2.4.8}$$

Bằng cách lựa chọn M của các điểm không p của bộ lọc lỗi - ớc l- ợng đồng nhất với tần số f_k, MSE ξ_p^f có thể ép bằng 0. Còn lại p – M các điểm không có thể lựa chọn tuỳ ý ở bất kỳ chỗ nào bên trong vòng tròn đơn vị.

Thuộc tính pha cực đại của bộ lọc lỗi □ớc l□ợng lùi.

Hàm hệ thống cho bộ lọc lỗi - ớc l- ợng lùi bậc p là

$$B_p(z) = z^{-p} A_p^*(z^{-1}) \tag{2.4.9}$$

Từ đó, các nghiệm của $B_p(z)$ là nghịch đảo nghiệm của bộ lọc lỗi - ớc l- ợng tiến với hàm hệ thống $A_p(z)$. Do vậy, nếu $A_p(z)$ là pha cực tiểu $B_p(z)$ là pha cực đại. Tuy nhiên, nếu quá trình x(n) là - ớc l- ợng, tất cả các nghiệm của $B_p(z)$ nằm trong vòng tròn đơn vị.

Thuộc tính nhiễu trắng.Giả sử rằng quá trình ngẫu nhiên x(n) là quá trình ngẫu nhiên ổn định AR(p) đ- ợc tạo ra bởi cho nhiễu trắng với sự thay đổi σ_w^2 qua bộ lọc toàn điểm cực với hàm hệ thống

$$H_k = \frac{1}{1+\sum_{k=1}^{p} a_k z^{-1}} \tag{2.4.10}$$

Sau đó bộ lọc lỗi - ớc l- ợng của loại p có hàm hệ thống

$$A_p(z) = 1+\sum_{k=1}^{p}\alpha_p(k) z^{-k}$$

Ở đây, những hệ số - ớc l- ợng $a_p(k) = a_k$. T- ơng ứng của bộ lọc lỗi - ớc l- ợng là chuỗi nhiễu trắng $w(n)$. Trong tr- ờng hợp này bộ lọc - ớc l- ợng lỗi hoá trắng quá trình ngẫu nhiên đầu vào x(n) và đ- ợc gọi là bộ lọc trắng, đ- ợc chỉ ra trong phần (2.2).

Hơn thế nữa, thậm chí nếu quá trình đầu vào x(n) không phải là quá trình AR, bộ lọc lỗi - ớc l- ợng cố gắng loại bỏ sự t- ơng quan trong các mẫu tín hiệu mẫu của quá trình đầu vào. Khi bậc của - ớc l- ợng tăng lên đầu ra của - ớc l- ợng $\hat{x}(n)$ sẽ trở nên gần xấp xỉ tới x(n) và do đó sự chênh lệch $f(n) = \hat{x}(n) - x(n)$ gần giống chuỗi nhiễu trắng.

Tính trực giao của các lỗi ước lượng lùi. Lỗi - ớc l- ợng lùi $g_m(k)$ từ các tầng khác nhau trong bộ lọc l- ới FIR là trực giao. Đó là

$$E[g_m(n) g_l^*(n)] = \begin{cases} 0, & 0 \le l \le m-1 \\ E_m^b & l = m \end{cases} \tag{2.4.12}$$

Tính chất này đ- ợc chứng minh dễ dàng bằng cách thay thế $g_m(n)$ và $g_l^*(n)$ vào (2.4.12) và đ- ợc kết quả mong muốn. Do đó

$$E[g_m(n) g_l^*(n)] = \sum_{k=0}^{m} b_m(k) \sum_{j=0}^{l} b_l^*(j) E[x(n-k) x^*(n-j)]$$

$$= \sum_{j=0}^{l} b_1^*(j) \sum_{k=0}^{m} b_m(k)\,\gamma_{xx}(j-k) \tag{2.4.13}$$

Nh- ng những ph- ơng trình chuẩn tắc cho - ớc l- ợng tuyến tính lùi yêu cầu rằng

$$\sum_{k=0}^{m} b_m(k)\,\gamma_{xx}(j-k) = \begin{cases} 0, & j = 1,2,...,m-1 \\ E_m^b & j = m \end{cases}$$

do đó

$$E\left[g_m(n)\,g_1^*(n)\right] = \begin{cases} E_m^b = E_m^f, & m = l \\ 0, & 0 \le l \le m-1 \end{cases}$$

Những thuộc tính khác: đây là một nhóm những thuộc tính khác về - ớc l- ợng lỗi tiến và lùi trong bộ lọc l- ới FIR. Những thuộc tính này đ- ợc đ- a ra d- ới đây với các tín hiệu có giá trị thực.

(a) $E\left[f_m(n)\,x(n-i)\right] = 0, \quad 1 \le i \le m$

(b) $E\left[g_m(n)\,x(n-i)\right] = 0, \quad 0 \le i \le m-1$

(c) $E\left[f_m(n)\,x(n)\right] = E\left[g_m(n)\,x(n-m)\right] = E_m$

(d) $E\left[f_1(n)\,f_1(n)\right] = E_{\max(i,j)}$

(e) $E\left[f_i(n)\,f_j(n-t)\right] = 0, \qquad cho \begin{cases} 1 \le t \le i-j, & i > j \\ -1 \ge t \ge i-j, & i < j \end{cases}$

(f) $E\left[g_i(n)\,g_j(n-t)\right] = 0, \qquad cho \begin{cases} 0 \le t \le i-j, & i > j \\ 0 \ge t \ge i-j+1, & i < j \end{cases}$

(g) $E\left[f_i(n+i)\,f_j(n+j)\right] = \begin{cases} E_0, & i = j \\ 0, & i \ne j \end{cases}$

(h) $E\left[g_1(n+i)\,g_1(n+j)\right] = E_{\max(i,j)}$

(i) $E\left[f_i(n)\,g_j(n)\right] = \begin{cases} K_j E_i, & i \ge j, \quad i,j \ge 0, \quad K_0 = 1 \\ 0, & i < j \end{cases}$

(j) $E\left[f_i(n)\,g_i(n-1)\right] = -K_{i+1}E_i$

(k) $E\left[g_i(n-1)\,x(n)\right] = E\left[f_i(n+1)\,x(n-i)\right] = -K_{i+1}E_i$

(l) $E\left[f_1(n)\,g_1(n-1)\right] = \begin{cases} 0, & i > j \\ -K_{j+1}E_i & i \le j \end{cases}$

2.5 BỘ LỌC L□ ỚI AR VÀ BỘ LỌC L□ ỚI HÌNH THANG ARMA

Trong phần 2.4.2 chúng ta đã trình bày cấu trúc l- ới FIR toàn điểm không và đ- a ra mối quan hệ với - ớc l- ợng tuyến tính. Ước l- ợng tuyến tính với hàm truyền

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k) z^{-k} \tag{2.5.1}$$

khi bị kích thích bởi quá trình ngẫu nhiên đầu vào x(n) và đ- ợc đầu ra gần giống chuỗi nhiễu trắng khi $p \to \infty$. Mặt khác, nếu quá trình đầu vào là AR(z), đầu ra của $A_p(z)$ là trắng. Do đó $A_p(z)$ sinh ra MA(p) khi bị kích thích với chuỗi nhiễu trắng, bộ lọc l- ới toàn điểm không đôi khi đ- ợc gọi là l- ới MA. Sau đó, chúng ta phát triển cấu trúc l- ới cho bộ lọc ng- ợc $1/A_p(z)$ bộ lọc mà chúng ta gọi là l- ới AR và cấu trúc thang l- ới cho xử lý ARMA.

2.5.1 Cấu trúc l- ới AR

Hãy xét hệ thống toàn điểm cực với hàm hệ thống

$$H(z) = \frac{1}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} \tag{2.5.2}$$

Ph- ơng trình khác cho hệ thống IIR là

$$y(n) = -\sum_{k=1}^{p} a_p(k) y(n-k) + x(n) \tag{2.5.3}$$

Bây giờ giả sử rằng chúng ta thay đổi vai trò của đầu vào và đầu ra [nh- thay đổi x(n) với y(n) trong (2.5.3)]. Do đó chúng ta đạt đ- ợc ph- ơng trình khác

$$x(n) = -\sum_{k=1}^{p} a_p(k) x(n-k) + y(n)$$

hoặc t- ơng đ- ơng

$$y(n) = x(n) + \sum_{k=1}^{p} a_p(k) x(n-k) \tag{2.5.3}$$

Chúng ta thấy rằng (2.5.4) là ph- ơng trình khác cho hệ thống FIR với hàm chức năng $A_p(z)$. Do đó hệ thống toàn điểm cực IIR có thể thay đổi tới hệ thống toàn điểm không bằng cách thay đổi vai trò đầu vào và đầu ra.

Căn cứ vào quan sát này, chúng ta có thể đạt đ- ợc cấu trúc l- ới AR(p) từ l- ới MA(p) bằng cách thay thế đầu vào với đầu ra. Do đó l- ới MA(p) có $y(n) = f_p(n)$ khi nó là đầu ra và $x(n) = f_0(n)$ là đầu vào, chúng ta có

$$\begin{aligned} x(n) &= f_p(n) \\ y(n) &= f_0(n) \end{aligned} \qquad (2.5.5)$$

Những định nghĩa này chỉ ra rằng ph- ơng trình $f_m(n)$ đ- ợc tính toán trong tầng d- ới. Sự tính toán này có thể hoàn thành bằng cách sắp xếp ph- ơng trình đệ qui cho $f_m(n)$ trong (2.2.11) và kết quả cho $f_{m-1}(n)$ trong giới hạn của $f_m(n)$. Do đó chúng ta đạt đ- ợc

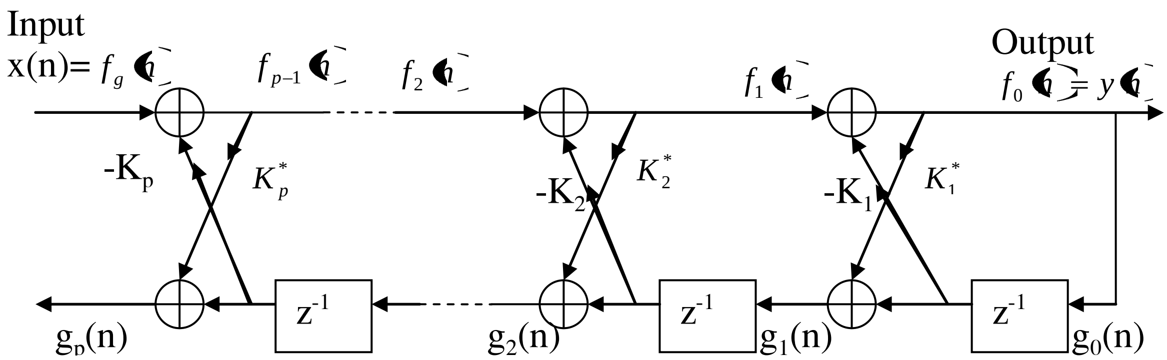
$$f_{m-1}(n) = f_m(n) - K_m g_{m-1}(n-1) \qquad m = p, p-1, \ldots, 1$$

Ph- ơng trình cho $g_m(n)$ còn lại không bị thay đổi. Kết quả của sự thay đổi này là tập hợp các ph- ơng trình

$$\begin{aligned} x(n) &= f_p(n) \\ f_{m-1}(n) &= f_m(n) - K_m g_{m-1}(n-1) \\ g_m(n) &= K_m^* f_{m-1}(n) + g_{m-1}(n-1) \\ y(n) &= f_0(n) = g_0(n) \end{aligned} \qquad (2.5.6)$$

Cấu trúc t- ơng ứng cho l- ới AR(p) đ- a ra trong hình (2.8). Chú ý rằng cấu trúc l- ới toàn điểm cực có một h- ớng toàn điểm không với đầu vào $g_0(n)$ và đầu ra $g_p(n)$ nó giống với đ- ờng toàn điểm không trong cấu trúc l- ới MA(p). Vì vậy ph- ơng trình cho $g_m(n)$ là giống nhau trong hai cấu trúc l- ới.

Chúng ta cũng quan sát thấy rằng cấu trúc l- ới AR(p) và MA(p) đ- ợc đặc tr- ng bởi các hệ số, nói rõ hơn, các hệ số phản xạ K_1. Kết quả ph- ơng trình đ- a ra trong (2.2.31) và (2.2.33) cho sự chuyển đổi giữa các thông số hệ thống $a_p(k)$ trong sự thực hiện dạng trực tiếp của hệ thống toàn điểm không $A_p(z)$ và các hệ số l- ới, K_1, của cấu trúc MA(p), xét đến giống với cấu trúc toàn điểm cực.



Hình 2.8 : Cấu trúc l- ới cho hệ thống toàn điểm cực (AR(p))

2.5.2 Quá trình ARMA và bộ lọc l- ới hình thang

L- ới toàn điểm không cung cấp khối xây dựng cơ bản cho cấu trúc kiểu l- ới mà minh hoạ hệ thống IIR có chứa cả điểm cực và điểm không. Để xây dựng cấu trúc thích hợp, chúng ta hãy xét một hệ thống IIR với hàm hệ thống

$$H(z) = \frac{\sum_{k=0}^{q} c_q(k) z^{-k}}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} = \frac{C_q(z)}{A_q(z)} \qquad (2.5.7)$$

Bỏ qua suy giảm thông th- ờng chúng ta giả sử là $p \geq q$.

Hệ thống này đ- ợc miêu tả bởi những ph- ơng trình sai phân

$$v(n) = -\sum_{k=1}^{p} a_p(k) v(n-k) + x(n)$$

$$y(n) = \sum_{k=1}^{p} c_q(k) v(n-k) \qquad (2.5.8)$$

ph- ơng trình này đạt đ- ợc bằng cách xem hệ thống nh- một tầng của hệ thống toàn điểm cực sinh ra bởi hệ thống toàn điểm không. Từ (2.5.8) chúng ta thấy rằng tại đầu ra y(n) chỉ đơn giản là sự kết hợp của các đầu ra trễ từ hệ thống toàn điểm cực.

Vì mọi điểm không sẽ là kết quả từ công thức tổ hợp tuyến tính của đầu ra tr- ớc. Chúng ta có thể mang sự quan sát này tới cấu trúc hệ thống điểm không và điểm cực bằng cách sử dụng cấu trúc l- ới toàn điểm cực nh- khối xây dựng cơ bản. Chúng ta thấy rằng $g_m(n)$ trong l- ới toàn điểm cực có thể biểu diễn nh- là tổ hợp tuyến tính của những đầu ra ở hiện tại và quá khứ. Trên thực tế, hệ thống

$$H_b(z) = \frac{G_m(z)}{Y_z} = B_m(z) \tag{2.5.9}$$

trong hệ thống toàn điểm không. Do đó, bất kỳ sự kết hợp tuyến tính nào của $g_m(n)$ cũng là bộ lọc toàn điểm không.

Hãy bắt đầu với bộ lọc l- ới toàn điểm cực với hệ số $K_m,\ \ 1 \le m \le p$ và thêm vào phần thang bằng cách lấy đầu sự tổ hợp tuyến tính có trọng số của $g_m(n)$. Kết quả là bộ lọc điểm không và điểm cực có cấu trúc thang_l- ới nh- trong hình 2.9. Đầu ra là

$$y(n) = \sum_{k=0}^{p} \beta_k g_k(n) \tag{2.5.10}$$

Ở đây, β_k là thông số xác định các điểm không của hệ thống. Hàm hệ thống t- ơng ứng (2.5.10) là

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{q} \beta_k \frac{G_k(z)}{X(z)} \tag{2.5.11}$$

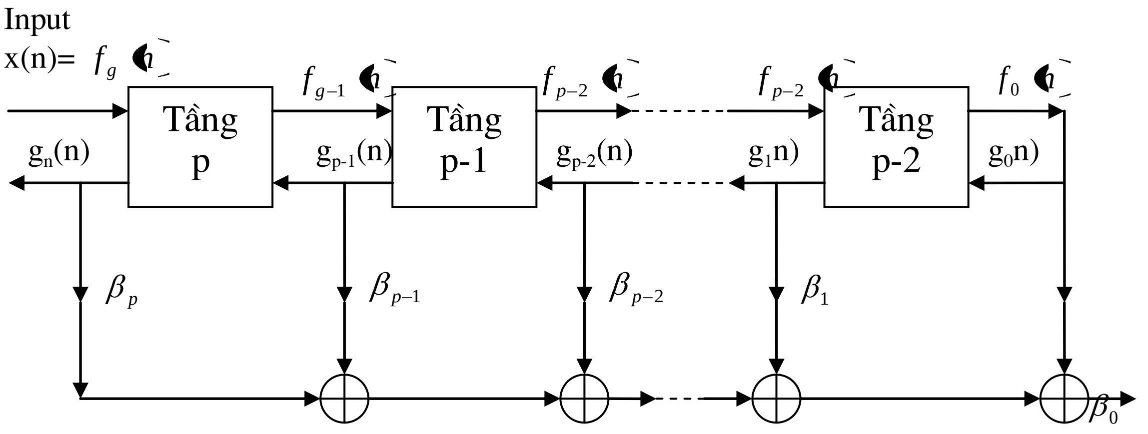
Từ $X(z) = F_p(z)$ và $F_0(z) = G_0(z)$, (2.5.11) có thể biểu diễn

$$H(z) = \sum_{k=0}^{q} \beta_k \frac{G_k(z)F_0(z)}{G_0(z)F_p(z)} \tag{2.5.12}$$

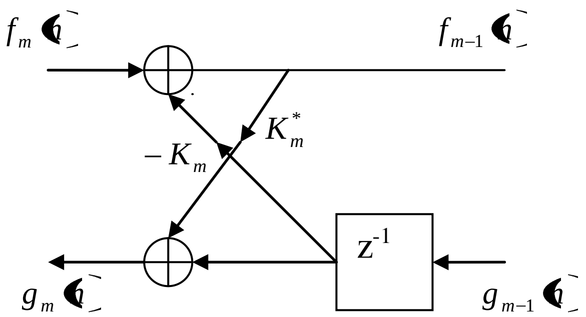
$$= \frac{1}{A_p(z)} \sum_{k=0}^{q} \beta_k B_k(z)$$

do đó

$$C_q(z) = \sum_{k=0}^{q} \beta_k B_k(z) \tag{2.5.13}$$



(a) Hệ thống điểm không_điểm cực



(b) Tầng thứ m của l- ới

Hình 2.9 : Cấu trúc l- ới thang cho hệ thống điểm cực_điểm không

Đây là mối quan hệ mong muốn mà có thể sử dụng để xác định hệ số trọng số β_k

Đ- a ra đa thức $C_q(z)$ và $A_p(z)$, trong đó $p \ge q$, hệ số phản xạ K_1 đ- ợc xác định đầu tiên từ hệ số $a_p(k)$. Bằng giá trị trung bình của mối quan hệ đệ qui lùi đơn b- ớc đ- a ra bởi (2.2.32) chúng ta cũng đạt đ- ợc đa thức $B_k(z)$, k = 1,2 . . . ,p. Sau đó những hệ số thang có thể đạt đ- ợc từ (2.5.13), hệ số mà có thể biểu diễn nh-

$$C_m(z) = \sum_{k=1}^{p} \beta_k B_k(z) + \beta_m B_m(z)$$

$$= C_{m-1}(z) + \beta_m B_m(z) \qquad (2.5.14)$$

hoặc t- ơng đ- ơng

$$C_{m-1}(z) = C_m(z) - \beta_m B_m(z), \qquad m = p, p-1, \ldots, 1 \qquad (2.5.15)$$

Bằng cách tiếp tục thực hiện mối quan hệ đệ qui lùi này, chúng có thể sinh ra mọi đa thức bậc thấp, $C_m(z)$, $m = p-1,\ldots,1$. Do đó $b_m(m) = 1$, thông số β_m đ- ợc xác định từ (2.5.15) bằng cách sắp đặt

$$\beta_m = c_m(m), \qquad m = p, p-1, \ldots, 1, 0 \qquad (2.5.16)$$

Cấu trúc bộ lọc l- ới này, khi bị kích thích bởi chuỗi nhiễu trắng, sinh ra quá trình ARMA(p,q) quá trình này có mật độ phổ đầu vào

$$\Gamma_{xx}(f) = \sigma_w^2 \frac{\left|C_q(f)\right|^2}{\left|A_p(f)\right|^2} \qquad (2.5.17)$$

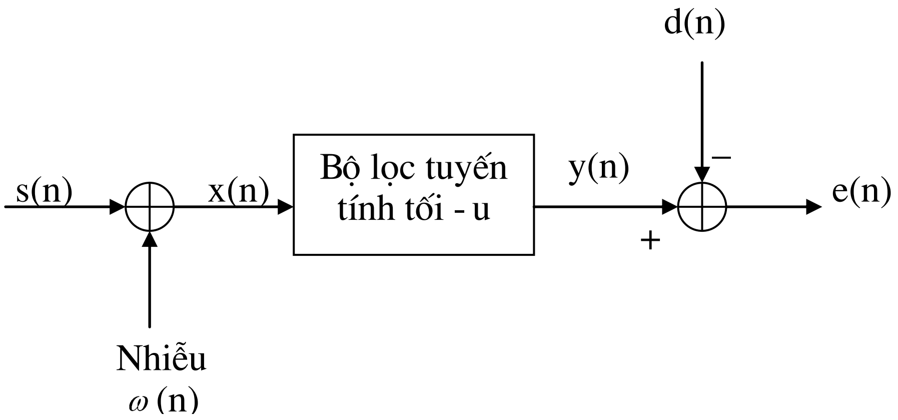
và hàm tự t- ơng quan mà thoả mãn (2.1.18),trong đó σ_w^2 trong sự biến đổi của chuỗi nhiễu trắng đầu vào.

2.6 BỘ LỌC WIENER SỬ DỤNG LỌC VÀ □ ỚC L□ ỢNG

Trong những ứng dụng thực tế chúng ta đ- a ra tín hiệu đầu vào, x(n), tín hiệu mà bao gồm tổng của các tín hiệu mong muốn, s(n), và tiếng ồn không mong muốn hoặc nhiễu w(n), và chúng ta thiết kế bộ lọc, bộ lọc mà sẽ triệt tiêu đ- ợc những thành phần không mong muốn. Trong tr- ờng hợp nh- vậy mục tiêu là thiết kế hệ thống mà lọc đi nhiễu thêm vào trong khi phải đảm bảo những đặc tính của tín hiệu mong muốn, s(n).

Trong phần này, chúng ta giải quyết vấn đề - ớc l- ợng tín hiệu trong sự có mặt của những tạp âm thêm vào. Bộ - ớc l- ợng giới hạn về bộ lọc tuyến tính với đáp ứng xung h(n), nó đ- ợc thiết kế để đầu ra xấp xỉ một vài chuỗi tín hiệu mong muốn theo lý thuyết d(n). Hình (2.10) minh hoạ vấn đề - ớc l- ợng tuyến tính.

Chuỗi đầu vào tới bộ lọc là x(n) = s(n)+w(n), và chuỗi đầu ra là y(n). Sự khác nhau giữa tín hiệu mong muốn và đầu ra của bộ lọc là chuỗi lỗi e(n) = d(n) - y(n). Chúng ta phân biệt ba tr- ờng hợp đặc biệt sau:



Hình 2.10 : Mô hình cho vấn đề - ớc l- ợng tuyến tính

1. Nếu d(n) = s(n), vấn đề - ớc l- ợng tuyến tính có liên quan tới *việc lọc*

2. Nếu d(n) = s(n+D), ở đây D > 0, vấn đề - ớc l- ợng tuyến tính có liên quan tới *- ớc l- ợng* tín hiệu. Chú ý rằng vấn đề này là sự khác với sự - ớc l- ợng đề cập trong phần tr- ớc. Ở đây d(n) = x(n+D), D $\geq$ 0.

3. Nếu d(n) = s(n-D), ở đây D > 0, vấn đề - ớc l- ợng tuyến tính liên quan tới tín hiệu *san bằng.*

Việc nghiên cứu sẽ tập trung ở việc lọc và - ớc l- ợng.

Tiêu chuẩn lựa chọn cho việc tối - u đáp ứng xung của bộ lọc h(n) là cực tiểu của lỗi trung bình bình ph- ơng. Tiêu chuẩn này có thuận lợi là dễ dàng và dễ dùng trong toán học. Giả định cơ bản là những chuỗi s(n), w(n) và d(n) là trung bình 0 và ổn định có độ nhạy cao. Bộ lọc tuyến tính sẽ đ- ợc cho là FIR hoặc là IIR. Nếu nó là IIR, chúng ta giả sử dữ liệu đầu vào x(n) tồn tại giá trị trên khoảng hữu hạn thời điểm tr- ớc. Chúng ta bắt đầu h- ớng tới thiết kế bộ lọc FIR tối - u. Bộ lọc tuyến tính tối - u, trong độ nhạy của lỗi trung bình bình ph- ơng tối thiểu (MMSE), đ- ợc gọi là *bộ lọc Wiener*.

2.6.1 Bộ lọc Wiener FIR

Giả sử là bộ lọc bị giới hạn độ dài về M với các hệ số h(k) $0 \le k \le M-1$. Do đó đầu ra y(n) phụ thuộc vào dữ liệu hữu hạn x(n), x(n-1), . . . , x(n-M+1)

$$y(n) = \sum_{k=0}^{M-1} h(k)\,x(n-k) \qquad (2.6.1)$$

Giá trị trung bình bình ph- ơng của lỗi đầu ra mong muốn $d(n)$ và $y(n)$ là

$$\xi_M = E\left[|e(n)|^2\right] \qquad (2.6.2)$$

$$= \mathrm{E}\left[\left|d(n) - \sum_{k=0}^{M-1} h(k)\,x(n-k)\right|^2\right]$$

Do đó đây là hàm bậc hai của các hệ số bộ lọc, cực tiểu của ξ_M đạt đ- ợc tập các ph- ơng trình tuyến tính

$$\sum_{k=0}^{M-1} h(k)\,y_{xx}(l-k) = y_{dx}(l) \qquad l = 0, 1, . . . ,M-1 \qquad (2.6.3)$$

ở đây $y_{ss}(k)$ là tự t- ơng quan của chuỗi đầu vào x(n) và $y_{dx}(k) = E[d(n)x^*(n-k)]$ là t- ơng quan chéo giữa chuỗi mong muốn d(n) và chuỗi đầu vào, x(n), $0 \le n \le M-1$. Tập các ph- ơng trình tuyến tính chỉ rõ bộ lọc tối - u đ- ợc gọi là *ph- ơng trình Wiener _ Hopf*. Những ph- ơng trình này cũng đ- ợc gọi là những ph- ơng trình chuẩn tắc.

Thông th- ờng, ph- ơng trình trong (2.6.3) có thể biểu diễn dạng ma trận

$$\Gamma_M h_M = y_d \qquad (2.6.4)$$

Ở đây Γ_M là ma trận Toeplitz (Hermitian) với các thành phần $\Gamma_M = y_{xx}(l-k)$ và y_d là vector t- ơng quan chéo $M\times 1$ với các thành phần $y_{dx}(l)$, l = 0, 1, . . . , M-1. Kết quả cho hệ số bộ lọc tối - u là

$$h_{opt} = \Gamma_M^{-1} y_d \qquad (2.6.5)$$

Và kết quả cực tiểu MSE đạt đ- ợc bởi bộ lọc Wiener là

$$MMSE_M = \min_{h_M} \xi_M = \sigma_d^2 - \sum_{k=0}^{M-1} h_{opt}(k) y_{dx}^*(k) \qquad (2.6.6)$$

hoặc t- ơng đ- ơng

$$MMSE_M = \sigma_d^2 - y_d^{*t} \Gamma_M^{-1} y_d$$

ở đây $\sigma_d^2 = E\left[|d(n)|^2\right]$.

Hãy xét một vài tr- ờng hợp đặc biệt của (2.6.3). Nếu chúng có quan hệ với bộ lọc, d(n) = s(n). Hơn nữa, nếu s(n) và w(n) là những chuỗi ngẫu nhiên không t- ơng quan, nh- th- ờng thấy trong thực tế,

$$y_{xx}(k) = y_{ss}(k) + y_{ww}(k)$$

$$y_{dx}(k) = y_{ss}(k)$$

và những ph- ơng trình chuẩn tắc trong (2.6.3) trở thành

$$\sum_{k=0}^{M-1} h(k)\left[y_{ss}(l-k) + y_{\omega\omega}(l-k)\right] = y_{ss}(l), \qquad l = 0, 1, . . . ,M-1 \qquad (2.6.9)$$

Nếu chúng ta xét về sự - ớc l- ợng thì d(n) = y_{ss}(k+D) ở đây D > 0. Gả sử là s(n) và w(n) là những chuỗi ngẫu nhiên không t- ơng quan, chúng ta có

$$y_{dx}(k) = y_{ss}(l+D) \qquad (2.6.10)$$

Do đó những ph- ơng trình cho bộ lọc - ớc l- ợng Wiener trở thành

$$\sum_{k=0}^{M-1} h(k)\left[y_{ss}(l-k) + y_{\omega\omega}(l-k)\right] = y_{ss}(l+D), \qquad l = 0, 1, . . . ,M-1 \qquad (2.6.10)$$

Trong tất cả những tr- ờng hợp này, ma trận t- ơng quan đ- ợc nghịch đảo là Toeplitz. Do đó thuật toán Levinson _ Durbin có thể sử dụng để tính các hệ số bộ lọc tối - u.

2.6.2 Nguyên tắc trực giao trong - ớc l- ợng trung bình bình ph- ơng tuyến tính

Ph- ơng trình chuẩn tắc cho hệ số bộ lọc tối - u đ- ợc đ- a ra trong (2.6.3) có thể đạt đ- ợc trực tiếp bằng cách áp dụng nguyên tắc trực giao trong - ớc l- ợng trung bình bình ph- ơng tuyến tính. Lỗi trung bình bình ph- ơng ξ_M

trong (2.6.2) là nhỏ nhất nếu hệ số bộ lọc h(k) đ- ợc lựa chọn giống nh- lỗi là trực giao cho mọi điểm dữ liệu trong - ớc l- ợng.

$$E\left[e(n)x^*(n-l)\right] = 0, \qquad l = 0, 1, \ldots, M-1 \tag{2.6.12}$$

trong đó

$$e(n) = d(n) - \sum_{k=0}^{M-1} h(k)x(n-k) \tag{2.6.13}$$

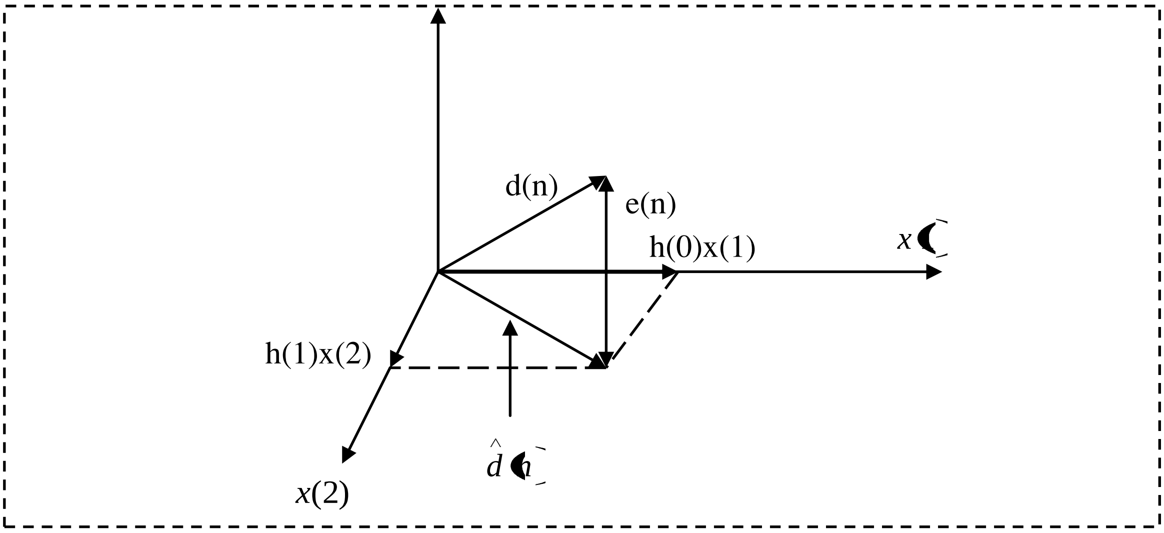
Ng- ợc lại, nếu hệ số bộ lọc thoả mãn (2.6.12), kết quả MSE là cực tiểu.

Khi xét ở ph- ơng diện hình học, đầu ra của bộ lọc, đ- ợc - ớc l- ợng

$$\bar{d}(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{2.6.14}$$

là vector trong không gian con đ- ợc mở rộng bởi dữ liệu x(k), $0 \le k \le M-1$. Lỗi e(n) là vector từ d(n) tới $\bar{d}(n)$ [i.e. $d(n) = e(n) + \bar{d}(n)$], nh- đ- a ra trong hình (2.11). Những trạng thái trực giao cơ bản có độ dài $\xi_M = E\left[|e(n)|^2\right]$ là nhỏ nhất khi e(n) là đ- ờng vuông góc với không gian dữ liệu (nh- e(n) là trực giao tới mọi điểm dữ liệu x(k), $0 \le k \le M-1$).

Chúng ta chú ý rằng kết quả đạt đ- ợc từ ph- ơng trình chuẩn tắc (2.6.3) là duy nhất nếu dữ liệu x(n) trong - ớc l- ợng d(n) là *tuyến tính độc lập*. Trong tr- ờng hợp này ma trận t- ơng quan Γ_M là không duy nhất. Mặt khác, nếu dữ liệu là tuyến tính độc lập, vị trí của Γ_M nhỏ hơn M và do đó kết quả không phải là duy nhất. Trong tr- ờng hợp này - ớc l- ợng $\bar{d}(n)$ có thể biểu diễn nh- tổ hợp tuyến tính của tập rút gọn của ph- ơng trình các điểm dữ liệu tuyến tính độc lập tới vị trí Γ_M.



Hình 2.11 : Biểu diễn hình học của vấn đề tuyến tính MSE

Do đó MSE đ- ợc tối thiểu hóa bằng cách lựa chọn các hệ số của bộ lọc thoả mãn nguyên lý trực giao, mức tối thiểu thặng d- MSE là

$$MMSE_n = E\left[e(n)\,d^*(n)\right] \qquad (2.6.15)$$

từ đó đạt đ- ợc kết quả đ- a ra trong (2.6.6)

2.6.3 Bộ lọc Wiener IIR

Trong phần tr- ớc chúng ta giới hạn bộ lọc trở thành FIR và đạt đ- ợc tập hợp của những ph- ơng trình tuyến tính M cho hệ số bộ lọc tối - u. Trong phần này chúng ta cho phép bộ lọc có độ dài vô hạn trong khoảng không gian (IIR) và chuỗi dữ liệu cũng sẽ vô hạn. Do đó đầu ra bộ lọc

$$y(n) = \sum_{k=0}^{\infty} h(k)\,x(n-k) \qquad (2.6.16)$$

Hệ số của bộ lọc đ- ợc lựa chọn để tối thiểu lỗi trung bình bình ph- ơng giữa đầu ra mong muốn d(n) và y(n)

$$\xi_M = E\left[|e(n)|^2\right]$$
$$= \mathrm{E}\left[\left|d(n) - \sum_{k=0}^{M-1} h(k)\,x(n-k)\right|^2\right] \qquad (2.6.17)$$

Ứng dụng của nguyên lý trực giao dẫn đến ph- ơng trình Wiener_Hopf

$$\sum_{k=0}^{\infty} h(k)\,\gamma_{xx}(l-k) = \gamma_{dx}(l) \qquad l \geq 0 \qquad (2.6.18)$$

Phần d- MMSE đơn giản đạt đ- ợc bằng cách ứng dụng điều kiện đ- a ra trong (2.6.15). Do đó chúng ta đạt đ- ợc

$$MMSE_{\infty} = \min_{h} \xi_M = \sigma_d^2 - \sum_{k=0}^{\infty} h_{opt}(k) y_{dx}^*(k) \quad (2.6.19)$$

Ph- ơng trình Wiener _ Hopf đ- a ra bởi (2.6.18) không thể giải trực tiếp với kỹ thuật biến đổi sang miền z bởi vì ph- ơng trình chỉ có ý nghĩa với $l \geq 0$. Chúng ta sẽ giải bộ lọc Wiener IIR tối - u dựa trên sự biểu diễn t- ơng ứng của quá trình ngẫu nhiên ổn định x(n).

Ta đã có quá trình ngẫu nhiên ổn định x(n) với chuỗi tự t- ơng quan $y_{xx}(l)$ và mật độ phổ công suất $\Gamma_{xx}(f)$ có thể biểu diễn bằng quá trình t- ơng đ- ơng i(n) bằng cách đ- a x(n) qua bộ lọc nhiễu trắng với hàm hệ thống 1/G(z), ở đây G(z) là phần pha tối thiểu đạt đ- ợc từ hệ số phổ của $\Gamma_{xx}(f)$

$$\Gamma_{xx}(z) = \sigma_i^2 G(z) G(z^{-1}) \quad (2.6.20)$$

Vì vậy G(z) đ- ợc phân tích trong miền $|z| > r_1$, ở đây $r_1>1$

Bây giờ, bộ lọc tối - u Wiener có thể xem nh- một tầng của bộ lọc nhiễu trắng 1/G(z) với bộ lọc thứ hai, gọi là Q(z), mà đầu ra của nó y(n) là giống với đầu ra của bộ lọc Wiener tối - u. Từ đó

$$y(n) = \sum_{k=0}^{\infty} q(k) i(n-k) \quad (2.6.21)$$

và e(n) = d(n) – y(n), ứng dụng nguyên lý trực giao ta đ- ợc ph- ơng trình Wiener _ Hopf mới nh-

$$\sum_{k=0}^{\infty} q(k) y_{ii}(l-k) = y_{dx}(l) \qquad l \geq 0 \quad (2.6.22)$$

Nh- ng vì i(n) là trắng, nên $y_{ii}(l-k) = 0$ với $l \neq k$. Do đó chúng ta đạt đ- ợc kết quả là

$$q(l) = \frac{y_{di}(l)}{y_{ii}(0)} = \frac{y_{di}(l)}{\sigma_i^2}, \qquad l \geq 0 \quad (2.6.23)$$

Biến đổi z của chuỗi q(l) là

$$Q(z) = \sum_{k=0}^{\infty} q(k) z^{-k}$$

$$= \frac{1}{\sigma_i^2} \sum_{k=0}^{\infty} y_{di}(k) z^{-k} \quad (2.6.24)$$

Nếu chúng ta kí hiệu biến đổi z hai phía của dãy t- ơng quan chéo $y_{di}(l)$ bởi $\Gamma_{di}(z)$

$$\Gamma_{di}(z) = \sum_{k=-\infty}^{\infty} y_{di}(k) z^{-k} \qquad (2.6.25)$$

và định nghĩa $[\Gamma_{di}(z)]_+$ nh-

$$[\Gamma_{di}(z)]_+ = \sum_{k=-\infty}^{\infty} y_{di}(k) z^{-k} \qquad (2.6.26)$$

sau đó

$$Q(z) = \frac{1}{\sigma_i^2}[\Gamma_{di}(z)]_+ \qquad (1.6.27)$$

Để xác định $[\Gamma_{di}(z)]_+$, chúng ta bắt đầu với đầu ra của bộ lọc nhiễu trắng, bộ lọc mà có thể biểu diễn nh- là

$$i(n) = \sum_{k=0}^{\infty} v(k) x(n-k) \qquad (2.6.28)$$

ở đây v(k), k $\geq$ 0, là đáp ứng xung t- ơng ứng của bộ lọc nhiễu trắng.

$$\frac{1}{G(z)} = V(z) = \sum_{k=0}^{\infty} v(k) z^{-k} \qquad (2.6.29)$$

sau đó

$$\begin{aligned} y_{di}(k) &= E[d(n) i^*(n-k)] \\ &= \sum_{k=0}^{\infty} v(m) E[d(n) x^*(n-m-k)] \\ &= \sum_{k=0}^{\infty} v(m) y_{dx}(k+m) \qquad (2.6.30) \end{aligned}$$

Biến đổi z của t- ơng quan chéo $y_{di}(k)$ là

$$\begin{aligned} \Gamma_{di}(z) &= \sum_{k=-\infty}^{\infty}\left[\sum_{m=0}^{\infty} v(m) y_{dx}(k+m)\right] z^{-k} \\ &= \sum_{m=0}^{\infty} v(m) \sum_{k=-\infty}^{\infty} y_{dx}(k+m) z^{-k} \\ &= \sum_{m=0}^{\infty} v(m) z^{m} \sum_{k=-\infty}^{\infty} y_{dx} z^{-k} \\ &= V(z^{-1}) \Gamma_{dx}(z) = \frac{\Gamma_{dx}(z)}{G(z^{-1})} \qquad (2.6.31) \end{aligned}$$

Vì vậy

$$Q(z) = \frac{1}{\sigma_i^2}\left[\frac{\Gamma_{dx}(z)}{G(z^{-1})}\right]_+ \qquad (2.6.32)$$

Cuối cùng, bộ lọc Wiener IIR tối - u có hàm chức năng

$$H_{opr}(z) = \frac{Q(z)}{G(z)}$$

$$= \frac{1}{\sigma_i^2 G(z)} \left[\frac{\Gamma_{dx}(z)}{G(z^{-1})} \right]_+ \qquad (2.6.33)$$

Tóm lại, giải pháp cho bộ lọc IIR Wiener yêu cầu chúng ta thực hiện tìm thừa số phổ của $\Gamma_{ii}(z)$ để đạt đ- ợc G(z), G(z) là thành phần pha cực tiểu, và sau đó chúng ta giải phần nhân quả của $\Gamma_{di}(z)/G(z^{-1})$.

Với giá trị tối thiểu MSE đ- a ra bởi (2.6.19) trong giới hạn miền tần số đặc tr- ng cho bộ lọc. Đầu tiên chúng ta chú rằng $\sigma_d^2 = E\left(|d(n)|^2\right)$ là giá trị tuyệt đối của chuỗi tự t- ơng quan y_{dd}(k). Do đó

$$y_{dd}(k) = \frac{1}{2\pi j} \oint_C \Gamma_{dd}(z) z^{k-1} dz \qquad (2.6.34)$$

theo đó

$$\sigma_d^2 = y_{dd}(0) = \frac{1}{2\pi j} \oint_C \frac{\Gamma_{dd}(z)}{z} dz \qquad (2.6.35)$$

ở đây tích phân đ- ờng đ- ợc đánh giá dọc theo vòng khép kín theo h- ớng bao quanh gốc trong miền hội tụ của $\Gamma_{dd}(z)$.

Phần thứ hai trong (2.6.19) cũng biến đổi dễ dàng tới miền tần số bằng cách ứng dụng thuật toán Parseval's. Do đó $h_{opt}(k) = 0$ cho k < 0, chúng ta có

$$\sum_{k=-\infty}^{\infty} h_{opt}(k) y_{dx}^*(k) = \frac{1}{2\pi j} \oint_C H_{opt}(z) \Gamma_{dx}(z^{-1}) z^{-1} dz \qquad (2.6.36)$$

ở đây C là vòng khép kín theo h- ớng quanh gốc, h- ớng mà thông th- ờng nằm bên trong miền hội tụ của $H_{opt}(z)\Gamma_{dx}(z^{-1})$.

Bằng cách kết hợp (2.6.35) với (2.6.36), chúng ta đạt đ- ợc kết quả mong muốn cho MMSE trong công thức

$$MMSE_\infty = \frac{1}{2\pi j} \oint_C \left[\Gamma_{dd}(z) - H_{opt}(z)\Gamma_{dd}(z^{-1})\right] z^{-1} dx \qquad (2.6.37)$$

2.6.4 Bộ lọc Wiener không nhân quả

Trong phần tr- ớc chúng ta giới hạn bộ lọc Wiener tối - u là nhân quả $(e,\ldots, h_{opt}(n) = 0 \quad for \quad n < 0)$. Trong phần này chúng ta bỏ điều kiện này và cho bộ lọc bao gồm cả vô hạn tr- ớc và vô hạn sau

$$y(n) = \sum_{k=-\infty}^{\infty} h(k) x(n-k) \qquad (2.6.38)$$

Kết quả của bộ lọc là không thể thực hiện đ- ợc về mặt vật lý. Nó cũng có thể xem nh- *bộ lọc san bằng*, bộ lọc mà giá trị tín hiệu không giới hạn sau đ- ợc dùng để san bằng - ớc l- ợng $\hat{d}$ (n) =y(n) của tín hiệu mong muốn d(n)

Ứng dụng của nguyên lý trực giao đạt đ- ợc ph- ơng trình Wiener_Hopf cho bộ lọc không nhân quả trong công thức

$$\sum h(k)\gamma_{xx}(l-k)=\gamma_{dx}(l) \qquad -\infty < l < \infty \tag{2.6.39}$$

và kết quả $MMSE_{xx}$ là

$$MMSE_{nc}=\sigma_d^2-\sum_{k=-\infty}^{+\infty} h(k)\gamma_{dx}^*(k) \tag{2.6.40}$$

Từ (2.6.39) cho $-\infty < l < \infty$, ph- ơng trình này có thể biến đổi trực tiếp để đạt đ- ợc bộ lọc Wiener không nhân quả tối - u là

$$H_{nc}(z)=\frac{\Gamma_{dx}(z)}{\Gamma_{xx}(z)} \tag{2.6.41}$$

$MMSE_{nc}$ cũng có thể biểu diễn đơn giản trong miền z là

$$MMSE_{nc}=\frac{1}{2\pi f}\oint\left[\Gamma_{dd}(z)-H_{nc}(z)\Gamma_{dx}(z^{-1})\right]z^{-1}dx \tag{2.6.42}$$

CH ƠNG 3 :

MÔ PHỎNG BỘ LỌC TUYẾN TÍNH TỐI U

3.1 GIỚI THIỆU VỀ SIMULINK

Simulik là một phần mềm dùng để mô hình hoá, mô phỏng và phân tích một hệ thống tự động. Simulik cho phép mô tả hệ thống tuyến tính, hệ phi tuyến, các mô hình trong thời gian liên tục gián đoạn hay một hệ kết hợp cả liên tục và gián đoạn. Để mô hình hoá, Simulik cung cấp một giao diện đồ hoạ để xây dựng mô hình nh- là một sơ đồ khối sử dụng thao tác "nhấn và kéo" chuột. Với giao diện này bạn có thể xây dựng mô hình nh- xây dựng trên giấy. Đây là sự khác xa các phần mềm mô phỏng tr-ớc nó mà ở đó ng-ời sử dụng phải đ-a vào các ph-ơng trình vi phân và các ph-ơng trình sai phân bằng một ngôn ngữ lập trình.

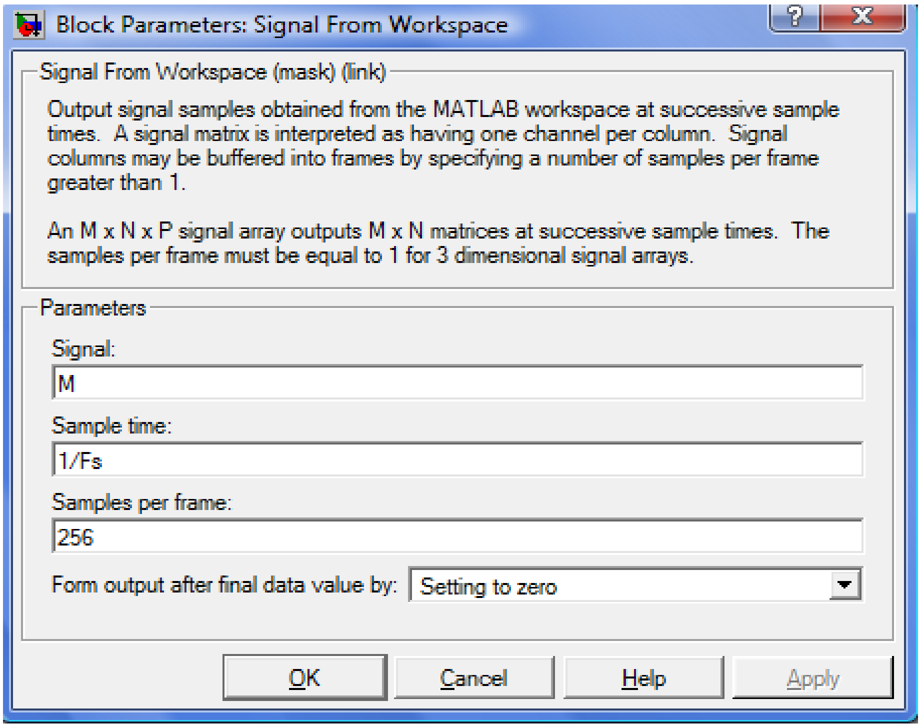
Việc lập trình trên Simulik sử dụng các đối t-ợng đồ hoạ gọi là Graphic Programming Unit. Loại hình lập trình này có xu thế đ-ợc sử dụng nhiều trong kỹ thuật bởi -u điểm lớn nhất của nó là tính trực quan.

Th- viện của Simulik cũng bao gồm toàn bộ th- viện các khối nh-: khối nhận tín hiệu, các khối nguồn tín hiệu, các phần tử tuyến tính và phi tuyến, các đầu nối chuẩn. Ng-ời sử dụng có thể quan sát hệ thống ở mức tổng quát, vừa có thể đạt đ-ợc mức độ cụ thể bằng cách nháy kép vào từng khối xác định xem xét chi tiết mô hình của từng khối. Với cách xây dựng kiểu này, ng-ời sử dụng có thể hiểu đ-ợc sâu sắc tổ chức của một mô hình và những tác động qua lại của các phần tử trong mô hình nh- thế nào.

Sau khi tạo lập ra đ-ợc một mô hình, ng-ời sử dụng có thể mô phỏng nó trong Simulik bằng cách nhập lệnh trong các của sổ lệnh của Matlab hay sử dụng các Menu có sẵn. Hơn nữa ng-ời sử dụng có thể thay đổi thông số một cách trực tiếp và nhận biết đ-ợc các ảnh h-ởng đến mô hình.

3.2 CÁC KHỐI SIMULINK DÙNG TRONG BỘ LỌC

3.2.1 Khối Signal From Workspace



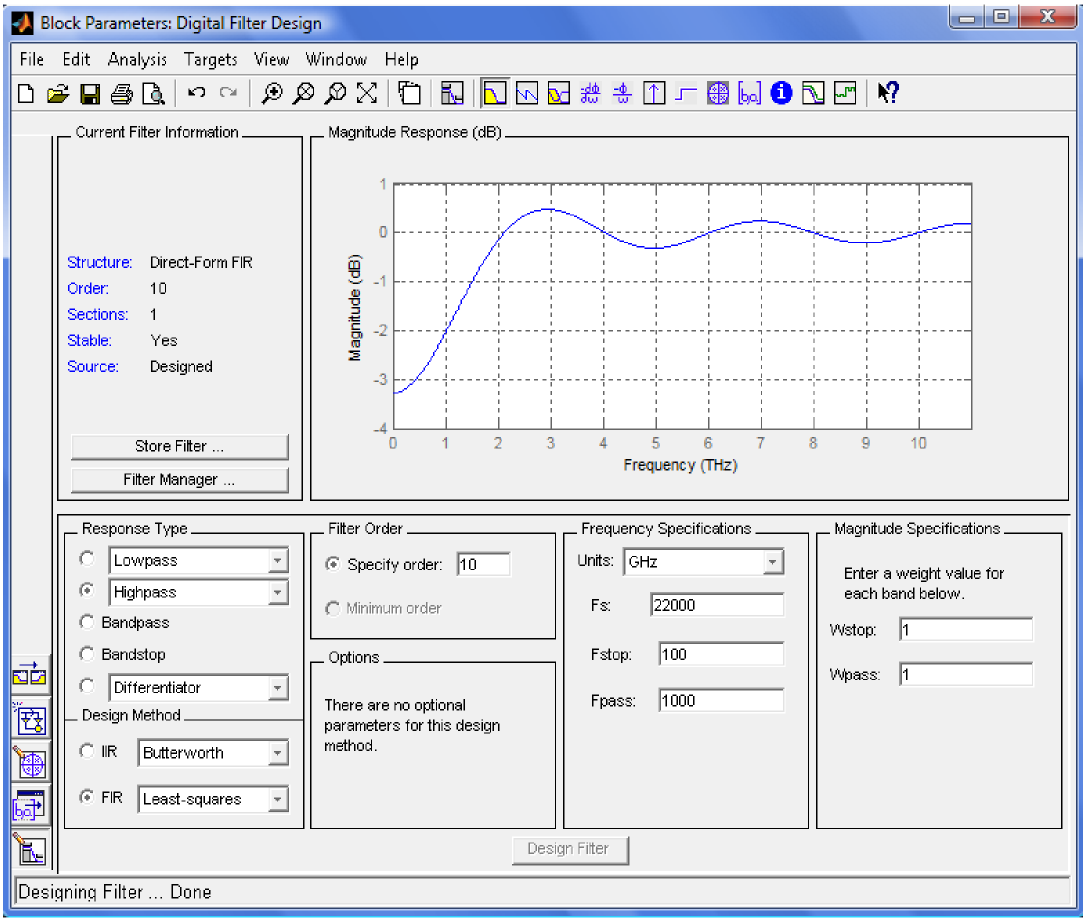
Các thông số của khối:

- Tín hiệu đ- a vào hệ thống (Signal)

- Chu kỳ lấy mẫu (Sample time)

- Số mẫu lấy cho mỗi khung (Samples per frame)

3.2.2 Khối Digital Signal design



Đây là khối thiết kế bộ lọc số, khối này bao gồm nhiều phần nhỏ để thiết kế bộ lọc.

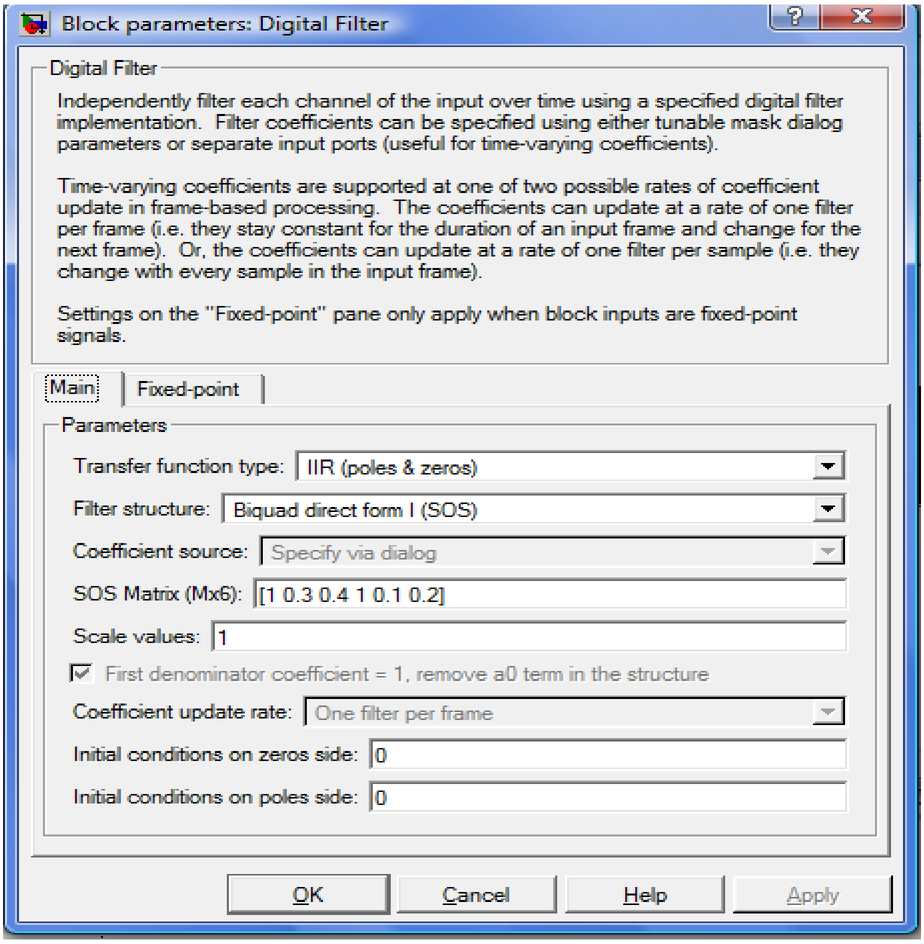
- Các kiểu bộ lọc: có thể lựa chọn bộ lọc thông thấp, bộ lọc thông cao, bộ lọc chắn dải, bộ lọc thông dải. Ph- ơng pháp thiết kế: có thể thiết kế giống bộ lọc IIR hoặc FIR.

- Bậc của bộ lọc (Filter order): lựa chọn bậc.

- Thông số của tần số (Ferquency Specification): đơn vị (Hz), tần số, dải tần tín hiệu. . .

- Thông số biên độ (Magnitude Specification): đơn vị(dB), dải tần biên độ . . .

3.2.3 Khối Digital filter



Các thông số của bộ lọc số

- Các kiểu chuyển đổi của bộ lọc (Transfer function type)

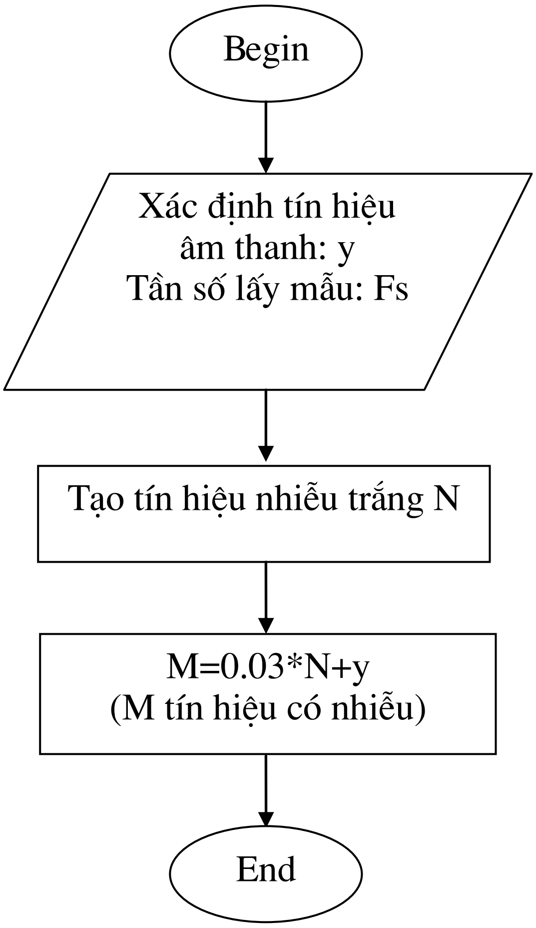
- Cấu trúc bộ lọc (Filter structure)

- Hệ số nguồn (Coeficient source)

- Mức giá trị (Scale value)

3.2.4 Ch- ơng trình tạo tín hiệu nhiễu trong Khối Signal From Workspace

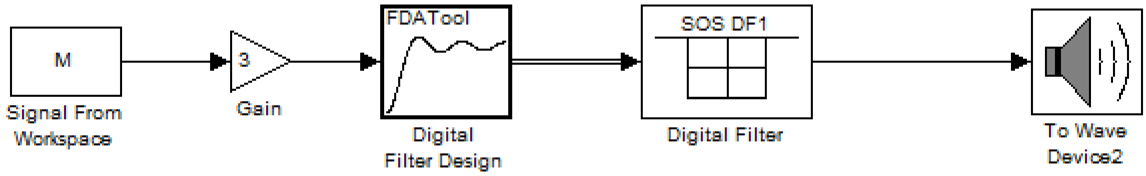
3.2.4.1 L□u đồ thuật toán



3.2.4.2 Chương trình chạy

```
function [M,Fs]=loc()
 [y,Fs,N]=wavread('c:/speech_dft.wav');
 sound(y,Fs);
 length(y)
 N=WGN(length(y),1,0);
 M=0.01*N+y;
 M=M;
 sound(M,Fs);
```

3.3 THỰC HIỆN VIỆC MÔ PHỎNG



Hình 3.1: Mô phỏng hệ thống lọc âm thanh

Tín hiệu có nhiễu đ- ợc lấy ra từ Singnal From Workspace, với tần số lấy mẫu Fs=22050 đ- ợc khuếch đại với hệ số khuếch đại K=3 đ- a vào khối thiết kế bộ lọc số (Digital Filter Design). Khi thiết kế ta chọn bộ lọc thông thấp (Lowpass) với tần số lấy mẫu Fs=22050Hz, dải tần tín hiệu (500 ÷ 11000)Hz. Ph- ơng pháp thiết kế, chọn bộ lọc FIR trong bộ lọc này chọn bình ph- ơng tối thiểu (least-squares). Bậc của bộ lọc (filter Order) chọn bằng 10. Sau đó, tín hiệu đ- ợc đ- a qua bộ lọc số (Digital Filter) ta có thể chọn các thông số bất kỳ nh- trong kiểu hàm chuyển đổi (Transfer function type) chọn FIR(all zeros-bộ lọc mọi điểm 0). Cấu trúc của bộ lọc có thể chọn từ trực tiếp (Direct form). Hệ số nguồn (Coefficient source) chọn Specify via dialog. Sau khi chọn các thông số thích hợp đ- a ra khối nguồn nghe lại âm thanh đã đ- ợc lọc nhiễu. Các thông số của các khối có thể thay đổi để đạt đ- ợc âm thanh có chất l- ợng tốt hơn.

KẾT LUẬN

Sau thời gian ba tháng với sự nỗ lực cố gắng tìm tòi, nghiên cứu, tham khảo các tài liệu và đ- ợc sự giúp đỡ tận tình của các thầy cô và các bạn. Đặc biệt là Th.S Nguyễn Văn D- ơng em đã hoàn thành xong nhiệm vụ đồ án của mình.

Với mục đích của đề tài là nghiên cứu bộ lọc tuyến tính tối - u, nên trong nội dung của đề tài em đã trình bày đ- ợc: cách biểu diễn quá trình ngẫu nhiên ổn định, - ớc l- ợng tuyến tính tiến và lùi, các thuật toán giải ph- ơng trình chuẩn tắc, đ- a ra một số bộ lọc nh- : bộ lọc l- ới AR, bộ lọc l- ới hình thang ARMA. Đặc biệt em đi sâu vào bộ lọc Wiener, với mục tiêu là thiết kế bộ lọc triệt tiêu đ- ợc những thành phần không mong muốn, lọc đi nhiễu thêm vào trong khi phải đảm bảo những đặc tính của tín hiệu mong muốn.

Tuy nhiên trong giới hạn của đề tài này ch- a trình bày đ- ợc những ứng dụng cụ thể của bộ lọc tuyến tính, ch- a thiết kế đ- ợc bộ lọc tuyến tính tối - u. Đây cũng là hạn chế và đồng thời cũng là h- ớng phát triển của đề tài.

Trong thời gian thực hiện làm đồ án tốt nghiệp, em đã cố gắng hết sức tìm hiểu, học hỏi về lĩnh vực này. Mặc dù đã cố gắng song do trình độ bản thân cũng nh- thời gian còn nhiều hạn chế nên đồ án này chắc chắn sẽ còn nhiều sai sót. Em rất mong đ- ợc sự góp ý, chỉ bảo của các thầy cô và các bạn để cho đồ án tốt nghiệp của em đ- ợc hoàn chỉnh hơn.

Em xin gửi lời cảm ơn chân thành đến các thầy cô trong ngành Điện tử _ Viễn thông, đặc biệt một lần nữa em xin gửi lời cảm ơn sâu sắc tới Th.S Nguyễn Văn D- ơng đã tận tình giúp đỡ em hoàn thành đồ án này.

TÀI LIỆU THAM KHẢO

1. Nguyễn Quốc Trung (2001), *Xử lý tín hiệu và lọc số* (tập 1, 2), Nhà xuất bản khoa học và kĩ thuật.

2. Quách Tuấn Ngọc, Xử lý tín hiệu số, Nhà xuất bản Giáo dục(1997)

3. Nguyễn Hữu Tình, Lê Tấn Dũng, Phạm Thị Ngọc Yến, Nguyễn Thị Lan H- ơng (1999), *Cơ sở matlab và ứng dụng,* Nhà xuất bản khoa học và kĩ thuật.

4. Jackson, L.B., Digital Filters and Signal Processing, Second Edition, Kluwer Academic Publishers, 1989. pp. 255-257.

5. John G.Proakis, Charles M. Rader, Fuyun Ling, Chrysostomos L.Nikias, Advanced Digital Signal Processing – Macmollan Publishing Company, Republic of Singapore (1992)