第5章

第5章 离散时间系统的相位、结构

5.1 离散时间系统的相频响应

5.2 FIR 系统的线性相位特性

5.3 线性相位FIR系统零点分布

5.4 全通系统与最小相位系统

5.5 谱分解、反卷积及系统辨识

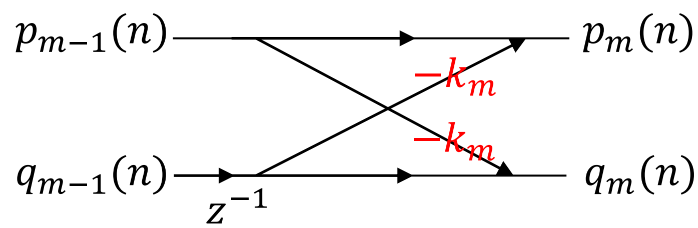
5.6 系统的信号流图与结构

5.7 离散时间系统的 Lattice 结构

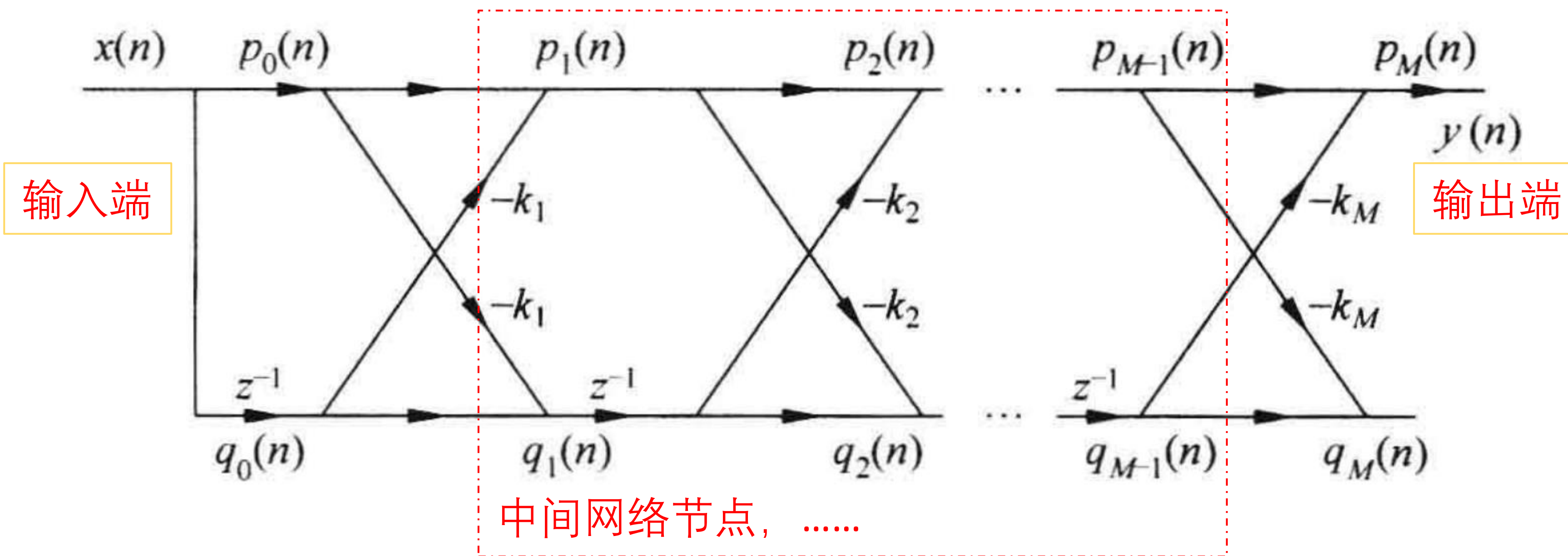
5.8 离散时间系统的状态变量描述

5.7 离散时间系统的 Lattice 结构

Lattice 结构，又称“格形”结构，是一种非常有特色的结构，在基于模型的功率谱估计、语音信号处理、自适应滤波方面有着重要的应用。

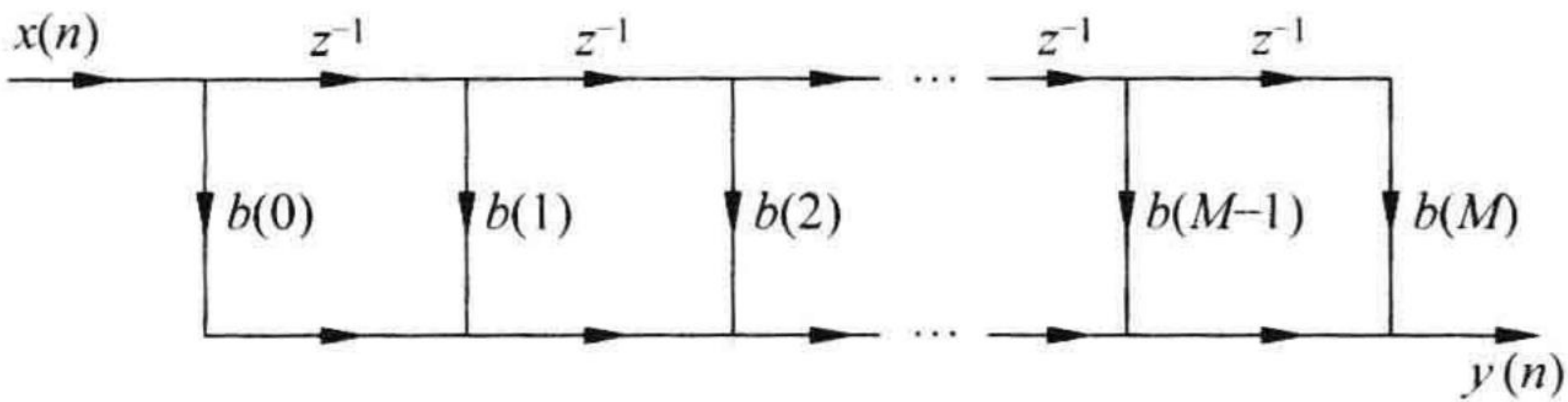


一、全零点系统(FIR)的Lattice结构



对照FIR直接型结构理解

$$H(z)=B(z)=\sum_{i=0}^{M} b(i) z^{-i}=1+\sum_{i=1}^{M} b_M^{(i)} z^{-i}$$



M阶FIR直接型实现方式中，设输入信号$x(n)$为$\delta(n)$，输出$y(n)$为单位脉冲响应$h(n)$，对于每个时刻（$n=0,1,\cdots,M$），有几个非零路径?

滤波器$h(n)$长度为N，$n=0,1,\cdots,N-1$，$M=N-1$。

1. FIR Lattice结构特点

（1）M个参数k_1、k_2、$\cdots$、k_M，$2M$次乘法，M次延迟。

直接型M个参数是$b(1)$、$b(2)$、$\cdots$、$b(M)$

由参数b求参数k

（2）FIR系统

设输入信号$x(n)$为$\delta(n)$，则各个时刻上端支路的输出$y(n)$对应单位脉冲响应序列$h(n)$的各个点。

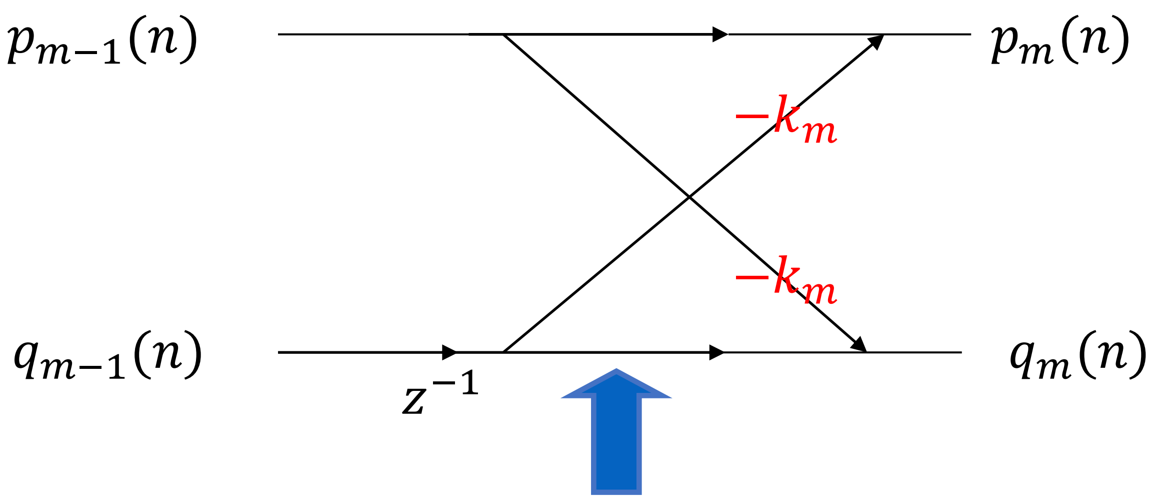
图中的系数k_m与系统函数中的系数$b(i)$的关系不明确，涉及多个路径，关系复杂（k_M除外，即对于z^{-M}项的系数有$k_M=-b_M^{(M)}$）。$b_M^{(i)}=b(i), M$阶。

试推导$h(1)=b(1)=f(k_1,k_2,\cdots,k_M)=?$

由参数k求参数b

（3）基本单元

由参数b求参数k



k_m：反射系数

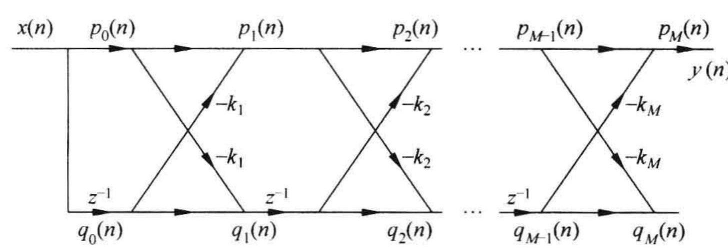
Lattice 结构的基本单元

$$p_m(n) = p_{m-1}(n) - k_m q_{m-1}(n-1)$$

$$q_m(n) = -k_m p_{m-1}(n) + q_{m-1}(n-1)$$

差分方程

$$p_0 = q_0 = x(n)$$
$$y(n) = p_M(n)$$



2. 参数求解方法

如何实现滤波器系数$b(i)$和k_m的相互转换?

从基本单元出发

$$\begin{cases} p_m(n) = p_{m-1}(n) - k_m q_{m-1}(n-1) \\ q_m(n) = -k_m p_{m-1}(n) + q_{m-1}(n-1) \end{cases}$$

Lattice结构
基本单元
ZT关系

$$\begin{cases} P_m(z) = P_{m-1}(z) - k_m z^{-1} Q_{m-1}(z) \\ Q_m(z) = -k_m P_{m-1}(z) + z^{-1} Q_{m-1}(z) \\ \begin{bmatrix} P_m(z) \\ Q_m(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m z^{-1} \\ -k_m & z^{-1} \end{bmatrix} \begin{bmatrix} P_{m-1}(z) \\ Q_{m-1}(z) \end{bmatrix} \end{cases}$$

再引入上下两条路径上网络节点转移函数

定义：
$$\begin{cases} B_m(z) = P_m(z)/P_0(z) = 1 + \sum_{i=1}^{m} b_m^{(i)} z^{-i} \\ \tilde{B}_m(z) = Q_m(z)/Q_0(z) \end{cases} \quad m = 1,2,\cdots,M$$

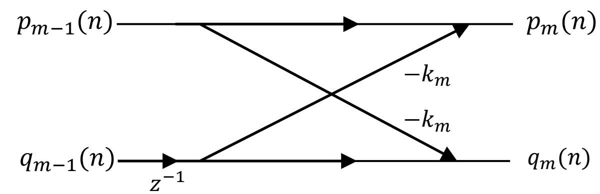
$$P_0(z) = Q_0(z)$$

$B_m(z), \tilde{B}_m(z)$ 是Lattice 结构中第 m 个上、下结点相对输入端的转移函数。

$$m = M$$

$$H(z) = B(z) = \sum_{i=0}^{M} b(i) z^{-i} = B_M(z) = 1 + \sum_{i=1}^{M} b_M^{(i)} z^{-i}$$

$$\begin{bmatrix} P_m(z) \\ Q_m(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m z^{-1} \\ -k_m & z^{-1} \end{bmatrix} \begin{bmatrix} P_{m-1}(z) \\ Q_{m-1}(z) \end{bmatrix}$$



然后，对Lattice结构基本单元的ZT关系式两式，分别除以$P_0(z)$、$Q_0(z)$，得到：

由低阶到高阶，或由高到低的混合递推关系。

$$\Rightarrow \begin{bmatrix} B_m(z \\ \tilde{B}_m(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m z^{-1} \\ -k_m & z^{-1} \end{bmatrix} \begin{bmatrix} B_{m-1}(z) \\ \tilde{B}_{m-1}(z) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} B_{m-1}(z \\ \tilde{B}_{m-1}(z) \end{bmatrix} = \left[\begin{pmatrix} 1 & k_m \\ z k_m & z \end{pmatrix} \right] \begin{bmatrix} B_m(z) \\ \tilde{B}_m(z) \end{bmatrix} / (1 - k_m^2)$$

上端网络节点转移函数是想要的，而下端网络节点的转移函数是中间变量；便于分析。

消除中间变量

掌握消除中间变量$\tilde{B}_m(z)$的递推关系的推导!

$$\begin{bmatrix} B_m(z) \\ \tilde{B}_m(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m z^{-1} \\ -k_m & z^{-1} \end{bmatrix} \begin{bmatrix} B_{m-1}(z) \\ \tilde{B}_{m-1}(z) \end{bmatrix}$$

递推初值:

$$B_0(z) = P_0(z)/P_0(z) = 1$$
$$\tilde{B}_0(z) = Q_0(z)/Q_0(z) = 1$$

$$b_M^{(0)} = 1$$

$m = 1$:

$$B_1(z) = B_0(z) - k_1 z^{-1} \tilde{B}_0(z) = 1 - k_1 z^{-1}$$
$$\tilde{B}_1(z) = -k_1 B_0(z) + z^{-1} \tilde{B}_0(z) = -k_1 + z^{-1}$$

$$\tilde{B}_1(z) = z^{-1} B_1(z^{-1})$$

$m = 2,3,\cdots,M$:

$$\tilde{B}_m(z) = z^{-m} B_m(z^{-1})$$

对比之前学过的:镜像对称多项式,互为镜像多项式,倒序多项式

消除中间变量$\tilde{B}_m(z)$，得到：

$$B_m(z) = B_{m-1}(z) - k_m z^{-m} B_{m-1}(z^{-1})$$

$$B_{m-1}(z) = [B_m(z) + k_m z^{-m} B_m(z^{-1})]/(1 - k_m^2)$$

低阶到高阶

高阶到低阶

在事先给定FIR系统函数$H(z) = B(z) = B_M(z)$后，可采用：高阶到低阶的递推。

高阶到低阶系数关系：两边分别展开，对应项的系数相等。

定义：

$$B_{m-1}(z) = 1 + \sum_{i=1}^{m-1} b_{m-1}^{(i)} z^{-i} \qquad B_m(z) = 1 + \sum_{i=1}^{m} b_m^{(i)} z^{-i}$$

关系：

$$B_{m-1}(z) = [B_m(z) + k_m z^{-m} B_m(z^{-1})]/(1 - k_m^2)$$

$$1+\sum_{i=1}^{m-1} b_{m-1}^{(i)} z^{-i}=\left\{1+\sum_{i=1}^{m} b_m^{(i)} z^{-i}+k_m z^{-m}\left[1+\sum_{i=1}^{m} b_m^{(i)} z^{i}\right]\right\}/\left(1-k_m^2\right)$$

$$=\left\{1+k_m b_m^{(m)}+\sum_{i=1}^{m-1}\left[b_m^{(i)}+k_m b_m^{(m-i)}\right] z^{-i}+\left[b_m^{(m)}+k_m\right] z^{-m}\right\}/\left(1-k_m^2\right)$$

对应关系

$$b_m^{(m)}+k_m=0 \rightarrow k_m=-b_m^{(m)}$$

$$\left(1+k_m b_m^{(m)}\right)/\left(1-k_m^2\right)=1 \rightarrow k_m=-b_m^{(m)}$$

$$b_{m-1}^{(i)}=\left(b_m^{(i)}+k_m b_m^{(m-i)}\right)/\left(1-k_m^2\right)$$

$$i=1,2\cdots,(m-1), m=1,2,\cdots M$$

得到时域递推关系

$$\left.\begin{array}{l} b_m^{(m)} = -k_m \\ b_m^{(i)} = b_{m-1}^{(i)} - k_m b_{m-1}^{(m-i)} \end{array}\right\} \text{低到高阶}$$

$$\left.\begin{array}{l} k_m = -b_m^{(m)} \\ b_{m-1}^{(i)} = [b_m^{(i)} + k_m b_m^{(m-i)}]/(1 - k_m^2) \end{array}\right\} \text{高到低阶}$$

$$H(z) = B(z) = \sum_{i=0}^{M} b(i) z^{-i} = 1 + \sum_{i=1}^{M} b_M^{(i)} z^{-i}$$

MATLAB中有相应的 m 文件。

例 $H(z)=B(z)=1-1.7z^{-1}+1.5z^{-2}-0.648z^{-3}$

$$b_3^{(1)}=-1.7,\quad b_3^{(2)}=1.5,\quad b_3^{(3)}=-0.648$$

$$\boxed{k_3=-b_3^{(3)}=0.648}$$

$$b_2^{(1)}=[b_3^{(1)}+k_3b_3^{(2)}]/(1-k_3^2)=-1.221453$$
$$b_2^{(2)}=[b_3^{(2)}+k_3b_3^{(1)}]/(1-k_3^2)=0.738498$$

$$B_2(z)=1-1.221453z^{-1}+0.738498z^{-2}$$

$$\boxed{k_2=-b_2^{(2)}=-0.738498}$$

$$b_1^{(1)}=[b_2^{(1)}+k_2b_2^{(1)}]/(1-k_2^2)=-0.70259$$

$$B_1(z)=1-0.70259z^{-1}$$

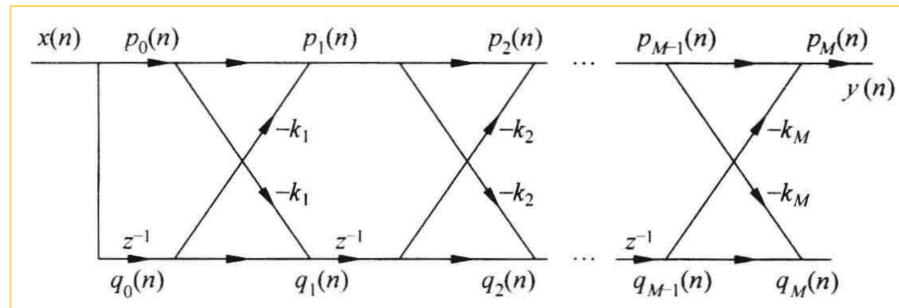
$$\boxed{k_1=-b_1^{(1)}=0.70259}$$

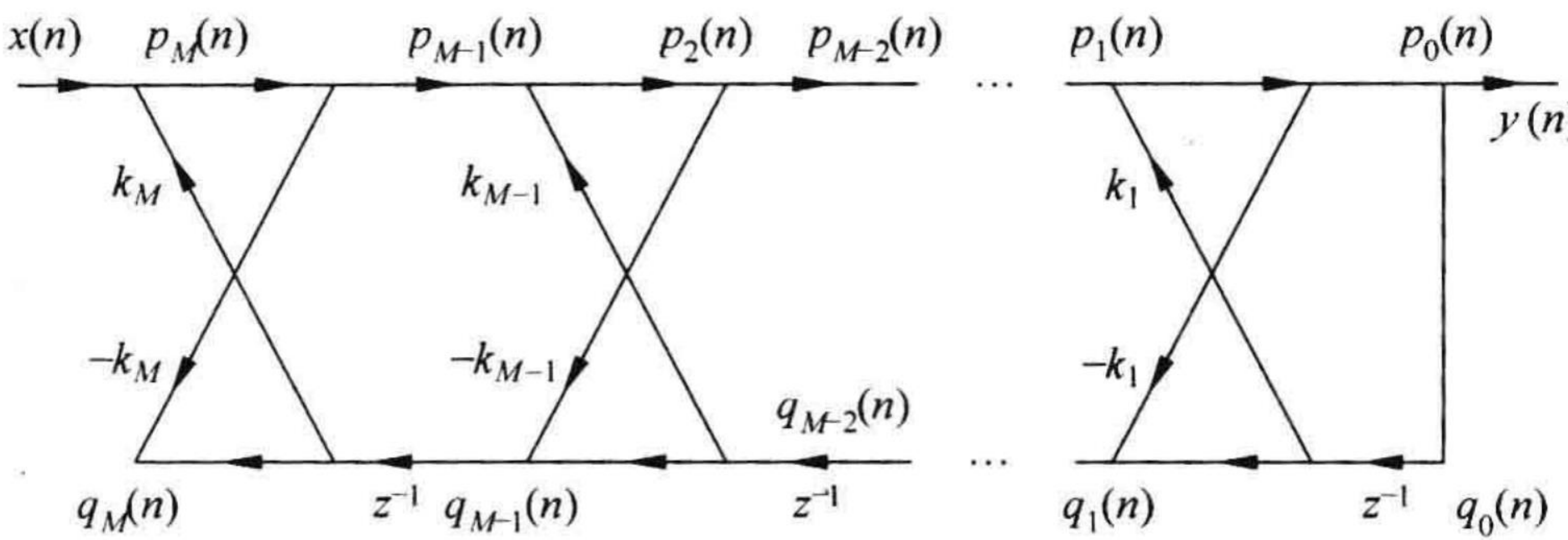
试验证$h(1)=b(1)=f(k_1,k_2,\cdots,k_M)=?$

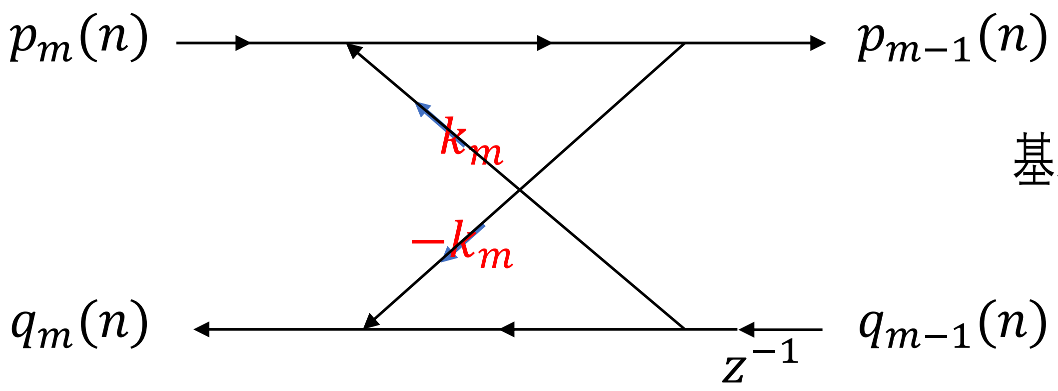
二、全极点系统(IIR)的Lattice结构

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

看作FIR系统的**逆**形式







基本单元逆形式

$$p_{m-1}(n) = p_m(n) + k_m q_{m-1}(n-1)$$

$$q_m(n) = -k_m p_{m-1}(n) + q_{m-1}(n-1)$$

或改写于FIR系统Lattice结构基本单元的关系式:

$$p_m(n) = p_{m-1}(n) - k_m q_{m-1}(n-1)$$

$$q_m(n) = -k_m p_{m-1}(n) + q_{m-1}(n-1)$$

推导k参数的求解方法，基本思路:

一阶Lattice结构，$M=1$，…；二阶Lattice结构，$M=2$，…；高阶Lattice结构。

最后分析表明：上述基本结构构成的全极点的IIR系统的Lattice结构是全零点Lattice结构的逆过程，二者的基本结构的差分方程是一样的，故可以采用前面的系数求解方法。

$M = 1$

$$p_{m-1}(n) = p_m(n) + k_m q_{m-1}(n-1)$$
$$q_m(n) = -k_m p_{m-1}(n) + q_{m-1}(n-1)$$

根据一阶Lattice结构，有

$$p_0(n) = p_1(n) + k_1 q_0(n-1)$$
$$q_1(n) = -k_1 p_0(n) + q_0(n-1)$$

因为： $p_0(n) = q_0(n) = y(n)$， $p_1(n) = x(n)$ $\quad p_M(n) = x(n)$， $M = 1$

所以： $y(n) = p_1(n) + k_1 y(n-1) = x(n) + k_1 y(n-1)$ $\boxed{\text{一阶IIR}}$

$q_1(n) = -k_1 y(n) + y(n-1)$ $\boxed{\text{一阶FIR}}$

令 $\dfrac{Y(z)}{P_1(z)} = \dfrac{1}{1-k_1 z^{-1}} = \dfrac{1}{A_1(z)}$， $A_1(z) = 1 - k_1 z^{-1}$ $\boxed{\text{注意输入输出关系}}$

$$\frac{Q_1(z)}{Y(z)} = -k_1 + z^{-1} = z^{-1}(1 - k_1 z) = z^{-1} A_1(z^{-1}) = \tilde{A}_1(z)$$

$M = 2$

$$\frac{Y(z)}{P_2(z)} = \frac{1}{A_2(z)} \qquad \frac{Q_2(z)}{Y(z)} = \tilde{A}_2(z)$$

$$A_2(z) = 1 - k_1(1 - k_2)z^{-1} - k_2 z^{-2}$$

$$\tilde{A}_2(z) = -k_2 - k_1(1 - k_2)z^{-1} + z^{-2}$$

$$\tilde{A}_2(z) = z^{-2} A_2(z^{-1})$$

倒序多项式

镜像多项式

高阶，类推；定义：

$$\frac{1}{A_m(z)} = \frac{Y(z)}{P_m(z)} \qquad \tilde{A}_m(z) = \frac{Q_m(z)}{Y(z)}$$

有：

$$\tilde{A}_m(z) = z^{-m} A_m(z^{-1})$$

$$H(z) = \frac{Y(z)}{P_M(z)} = \frac{1}{A_M(z)} = \frac{1}{1 + \sum_{i=1}^{M} a_M^{(i)} z^{-i}}$$

系数$k_1, k_2, \cdots, k_M$

及$a_m^{(i)}, i = 1,2,\cdots m, m = 1,2,\cdots, M$

的求解方式同FIR系统Lattice结构的计算方法，只是将多项式的系数$b_m^{(i)}$换成$a_m^{(i)}$

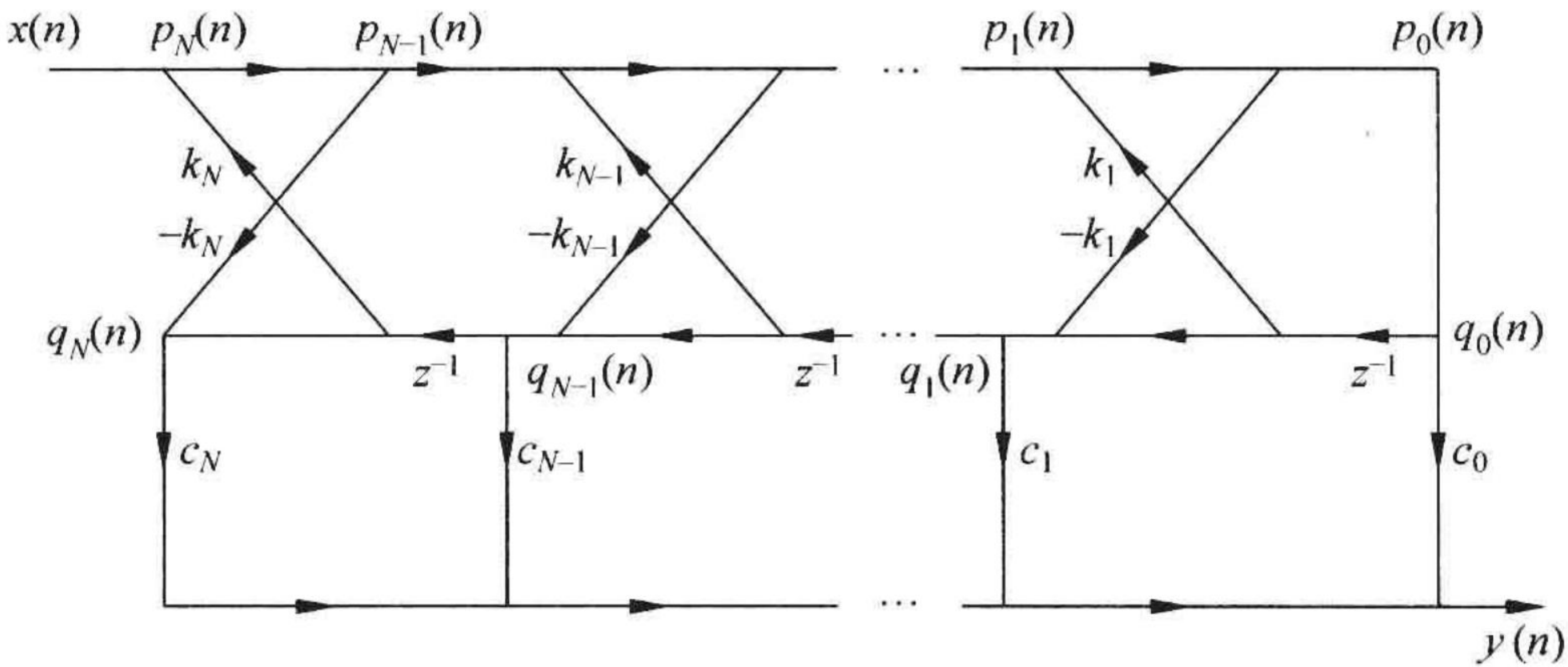
注意：在递推求解的过程中，反射系数

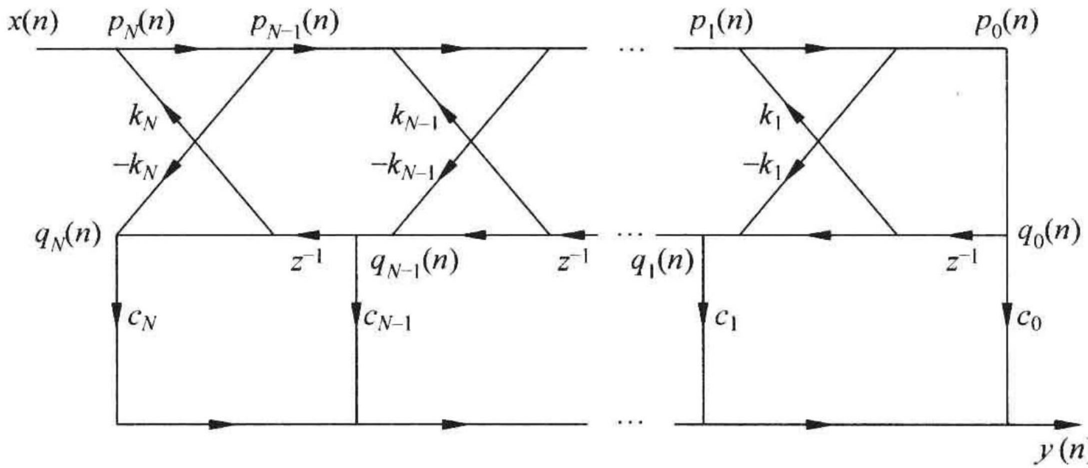
$$k_m \neq 1, \quad k = 1,2,\cdots, M$$

有关反射系数的更多讨论见信号建模。

三、极－零系统的Lattice结构

$$H(z)=\frac{\sum_{r=0}^{N}b_k z^{-k}}{1+\sum_{k=1}^{N}a_k z^{-k}}=\frac{B(z)}{A(z)}$$





两组Lattice系数

$k_1, k_2, \cdots, k_N$

$c_0, c_1, \cdots, c_N$

上半部对应全极点系统 $1/A(z)$ | an all-poles lattice with coefficient k_m

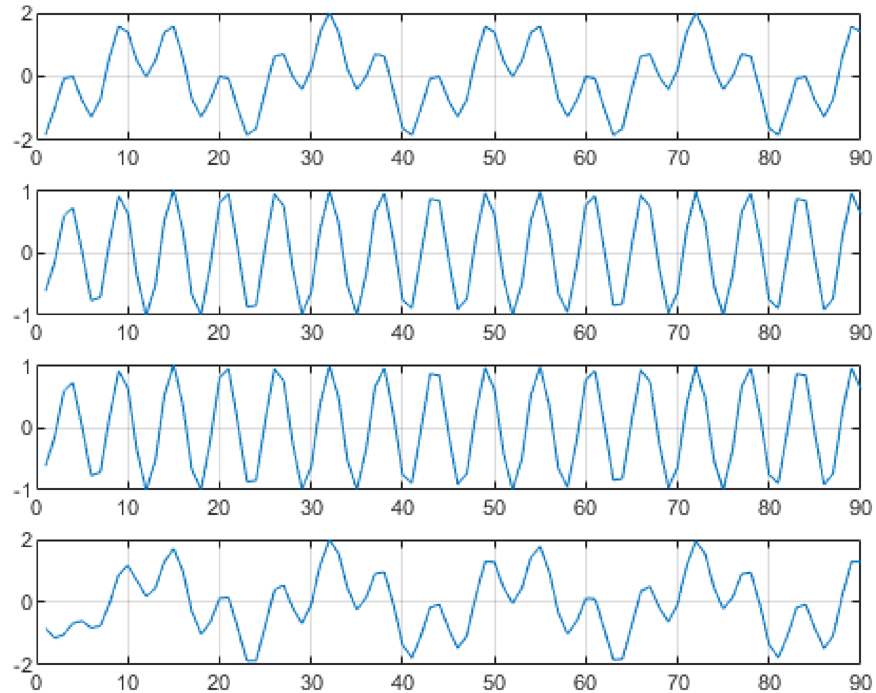
下半部对应全零点系统 $B(z)$ | a ladder part, ladder coefficients c_m

先求：$k_1, k_2, \cdots, k_N$　　　求解方法，同全极点系统

再求： $c_k = b_k - \sum_{m=k+1}^{N} c_m a_m^{(m-k)}$　　　$k = 0,1,\cdots, N$

递推求解

```
clear;
% 给定 IIR滤波器；
B=[0.0201 0 -0.0402 0 0.0201];
A=[1 -1.637 2.237 -1.307 0.641];

% 产生信号 x；
w1=0.1*pi;w2=0.35*pi;
N=100;n=0:N-1;
x=cos(w1*n)+cos(w2*n);

y1=filter(B,A,x);   % 直接滤波；

[k,c]=tf2latc(B,A) % 求lattice系数；

[y2,g]=latcfilt(k,c,x); % lattice滤波；

subplot(411);plot(x(10:N-1));grid on;
subplot(412);plot(y1(10:N-1));grid on;
subplot(413);plot(y2(10:N-1));grid on;
subplot(414);plot(g(10:N-1));grid on;
```




```
k =
   -0.4817
    0.9519
   -0.4374
    0.6410
```

```
c =
    0.0329
   -0.0567
   -0.0405
    0.0329
    0.0201
```

更多分析

5.8 离散时间系统的状态变量描述

为什么要引入状态变量描述法?

LSI系统的输入输出描述法是一种“黑盒子”方法

基本输入输出关系是时域卷积、频域乘积、复频域乘积

基于单位脉冲响应的系统外部特性的描述方法

实际情况中需要了解系统的内部结构

系统的研究包括系统分析、系统综合

系统包括单输入单输出系统、多输入多输出系统;

系统分析，要借助矩阵、线性代数等更多的数学工具;

系统综合，主要指系统实现，包括软硬件，需要了解系统的外部特性，**需要分析系统的内部特性**

→ 引入状态变量法

LSI系统动态特性之差分方程描述

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

将$y(n_0-1), y(n_0-2),\ldots,y(n_0-N)$作为系统在$n = n_0$时刻的一组初始条件，可以递归求解出$n$时刻以及后面时刻的系统的输出。

它们也可以作为系统内部的“状态”。

后面**从系统的直接实现出发来讨论状态变量法**。

状态变量分析法

状态变量分析法用状态方程和输出方程两个矩阵方程描述系统。**状态方程**，把系统内部的状态变量和输入信号联系起来；**输出方程**，则把输出信号和状态变量联系起来。

状态变量只取信号流图中的**少量节点变量**，要求它们之间必须**线性无关**，即任何一个状态变量不能由其它状态变量以线性组合的方式构成。

在滤波器的三个基本组成元件中，倍率器和相加器都属于线性运算，只有**延迟器**不属于线性组合运算。

系统的结构图中，**状态变量数目**也就等于**延迟器数目**。可以把每个**迟延环节的输出变量**作为**状态变量**。

状态变量分析法

以每个状态变量为主体可以列写出一个状态方程，若有N个状态变量，就可以列出N个状态方程，构成N阶状态空间方程组。

输出变量既然是状态变量的线性组合，所以一定能用状态变量的线性方程来表示，称为输出方程。

习惯上**状态变量**是用$x_i(i=1,2,\ldots,N)$表示，$u(n)$为**输入变量**，$y(n)$是**输出变量**。

以**四阶IIR直接型滤波器为例**说明它的状态方程列写方法。

直接II型实现

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

$$Y(z) = \frac{\sum_{r=0}^{M} b_r z^{-r}}{1 + \sum_{k=1}^{N} a_k z^{-k}} X(z)$$

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

$$\boxed{W(z) = \frac{X(z)}{1 + \sum_{k=1}^{N} a_k z^{-k}}}$$

则:

$$Y(z) = W(z) \sum_{r=0}^{M} b_r z^{-r}$$

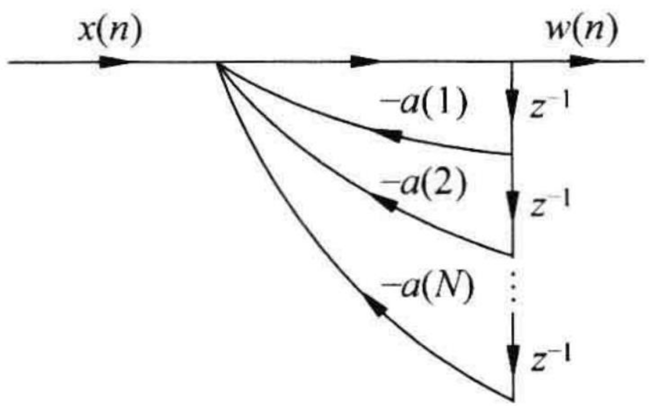
及

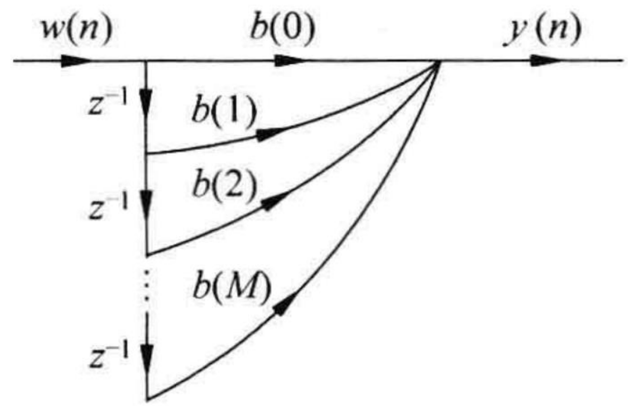
$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n)$$

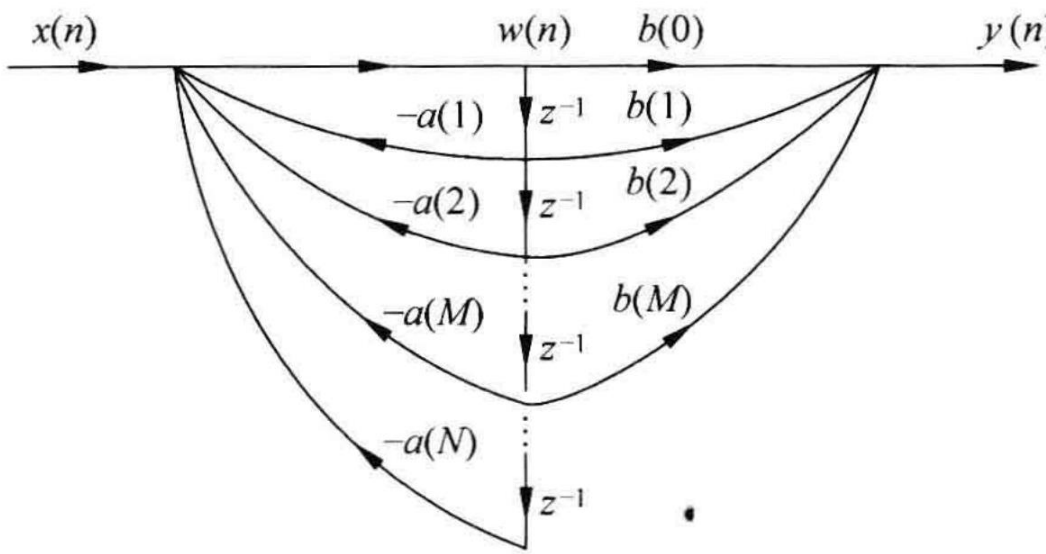
$$y(n) = \sum_{r=0}^{M} b_r w(n-r)$$

直接II型实现

信号流图





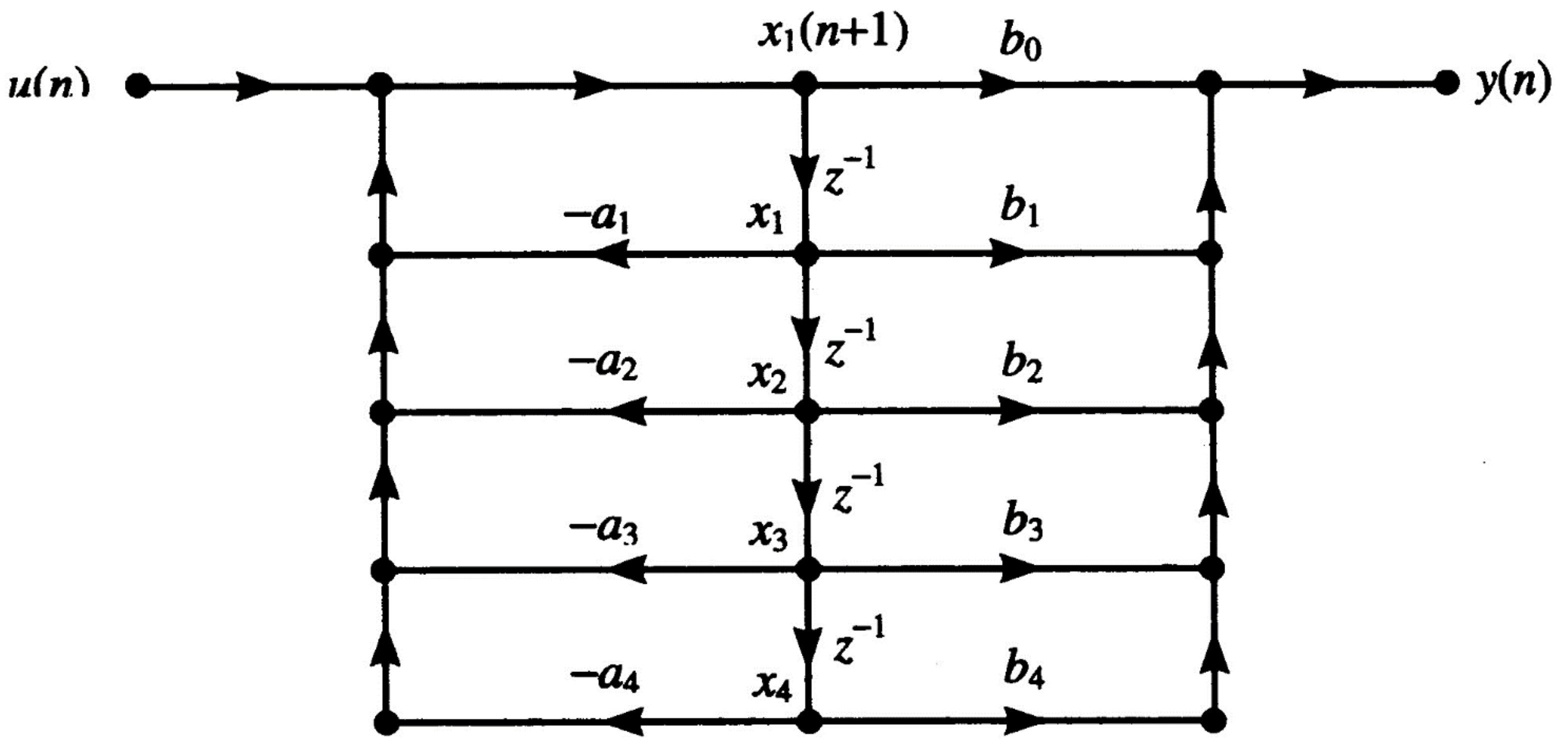


状态变量分析法

四阶IIR直接型滤波器其差分方程为

$$\begin{aligned} y(n) = &-a_1 y(n-1) - a_2 y(n-2) - a_3 y(n-3) \\ &-a_4 y(n-4) + b_0 u(n) + b_1 u(n-1) \\ &+b_2 u(n-2) + b_3 u(n-3) + b_4 u(n-4) \end{aligned}$$

x_i

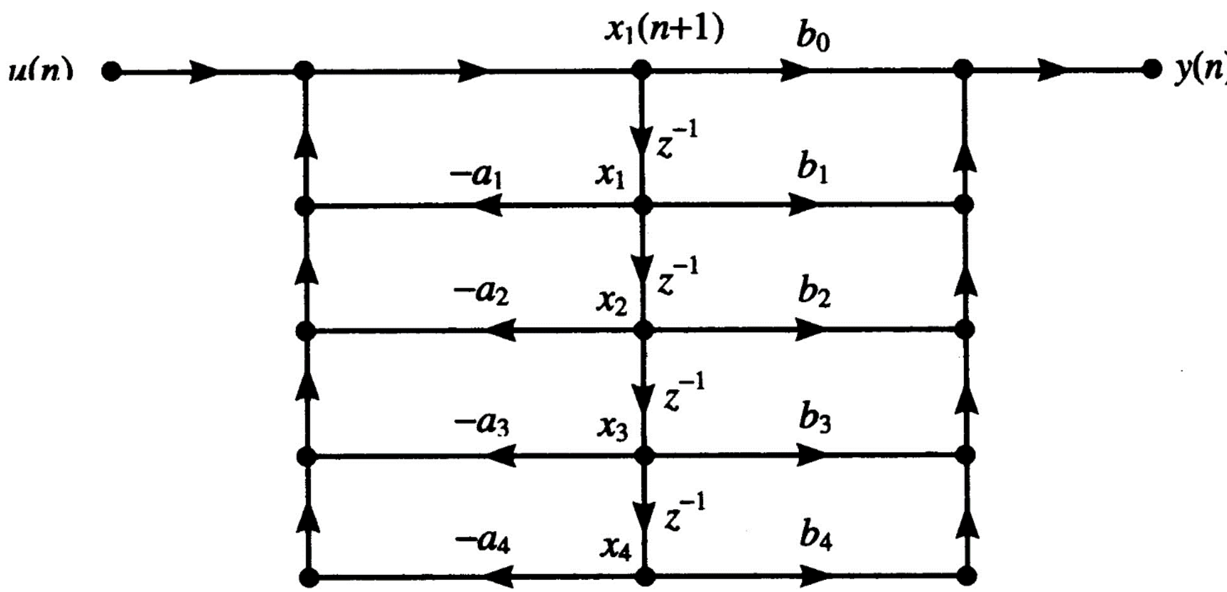


状态变量分析法

系统的结构图用直接II型表示。其中在四个迟延器后的变量分别为$x_1(n)$、$x_2(n)$、$x_3(n)$和$x_4(n)$，选它们为此系统的四个状态变量。

在各个迟延器前的变量是$x_1(n+1)$、$x_2(n+1)$、$x_3(n+1)$和$x_4(n+1)$ 。根据信号流图节点方程，它们可以用状态变量的线性组合来表示。得到状态方程：

$$x_1(n+1) = u(n) - a_1x_1(n) - a_2x_2(n) - a_3x_3(n) - a_4x_4(n)$$
$$x_2(n+1) = x_1(n)$$
$$x_3(n+1) = x_2(n)$$
$$x_4(n+1) = x_3(n)$$



状态变量分析法

输出方程也可从图中得到，把$x(n+1)$代换掉，整理成状态变量的线性组合，可得：

$$y(n) = b_0 x(n+1) + b_1 x_1(n) + b_2 x_2(n) + b_3 x_3(n) + b_4 x_4(n)$$

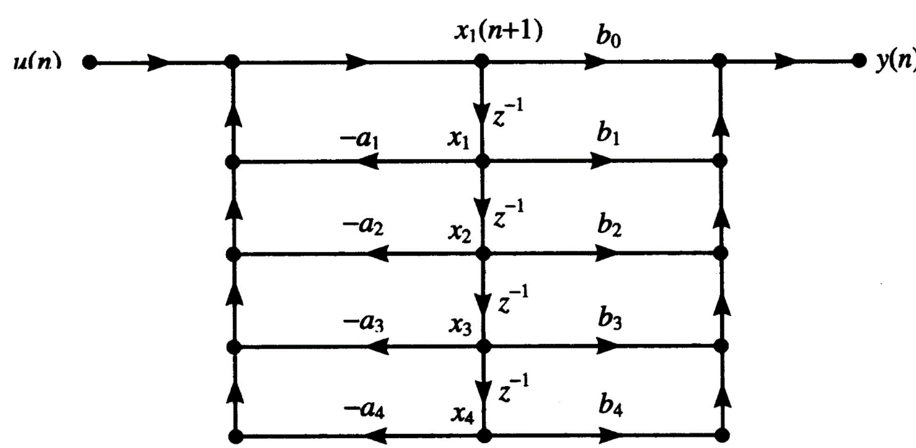
$$= b_0[u(n) - a_1 x_1(n) - a_2 x_2(n) - a_3 x_3(n) - a_4 x_4(n)]$$

$$+ b_1 x_1(n) + b_2 x_2(n) + b_3 x_3(n) - b_4 x_4(n)$$

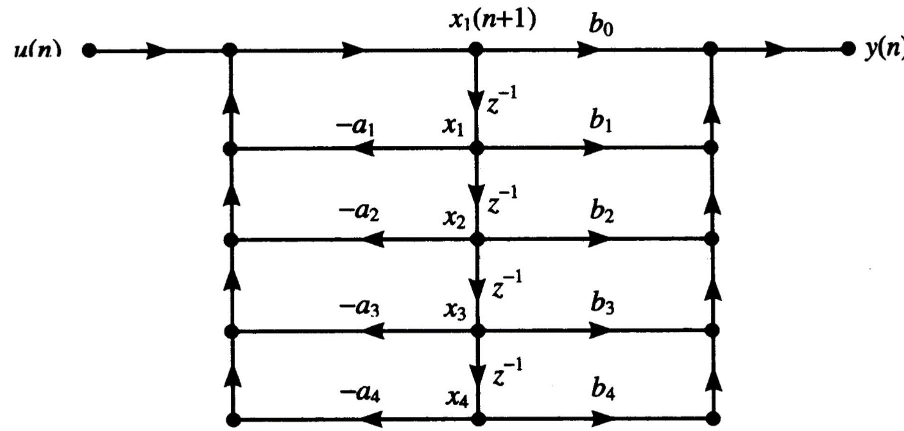
得到输出方程的最后形式：

$$y(n) = (b_1 - b_0 a_1) x_1(n) + (b_2 - b_0 a_2) x_2(n)$$

$$+ (b_3 - b_0 a_3) x_3(n) + (b_4 - b_0 a_4) x_4(n) + b_0 u(n)$$



状态变量分析法



系统的状态方程和输出方程用矩阵表示为

$$X(n+1)=\begin{bmatrix}x_1(n+1)\\x_2(n+1)\\x_3(n+1)\\x_4(n+1)\end{bmatrix}=\begin{bmatrix}-a_1 & -a_2 & -a_3 & -a_4\\1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\cdot\begin{bmatrix}x_1(n)\\x_2(n)\\x_3(n)\\x_4(n)\end{bmatrix}+\begin{bmatrix}b_0\\0\\0\\0\end{bmatrix}\cdot u(n)$$

$$y(n)=[(b_1-a_1)\quad(b_2-a_2)\quad(b_3-a_3)\quad(b_4-a_4)]\begin{bmatrix}x_1(n)\\x_2(n)\\x_3(n)\\x_4(n)\end{bmatrix}+b_0u(n)$$

这就符合差分状态方程组的标准形式:

状态变量分析法

状态方程　　$X(n+1) = AX(n) + BU(n)$

输出方程　　$Y(n) = CX(n) + DU(n)$

A、B、C、D是系统状态方程组的基本系数矩阵。知道这四个系数矩阵，系统的性能就唯一地确定了。

如果系统有N个延迟环节，因而有N个状态变量，就称为N维的。

若有K个输出变量，L个输入变量，则变量X是$N\times1$向量，Y是$K\times1$向量，U是$L\times1$向量，而系数矩阵A是$N\times N$阶，B是$N\times L$阶，C是$K\times N$阶，D是$K\times L$阶。

状态变量分析法

由状态方程的四个参数矩阵很容易求出等价的其它形式结构的参数。

首先推导**状态空间的输入输出关系式**，将它变换为**传递函数**的公式。

用z变换算子表示状态方程组，可以写出

$$X(n+1)=zX(n)=AX(n)+BU(n),$$

移项得到： $(zI-A)X(n)=BU(n)$

对方程两端变量作z变换，移项得：

$$\frac{X(z)}{U(z)}=(zI-A)^{-1}B$$

状态变量分析法

因为传递函数$H(z)$是输入输出的z变换之比，将输出方程代入，可以得到：

$$H(z)=\frac{Y(z)}{U(z)}=C\frac{X(z)}{U(z)}+D=C(zI-A)^{-1}B+D$$

得到$H(z)$以后，当然就很容易得到**零极增益**或**极点留数**等表示式。

由传递函数转换为状态空间参数矩阵的逆运算不是唯一的。因为同一个系统，选择的状态变量不同，就会得出不同的状态方程，因而具有不同的系数矩阵。

线性系统系数变换表

	传递函数 *b*,*a*	状态空间 *A*,*B*,*C*,*D*	零极增益 *z*,*p*,*k*	部分分式 *r*,*p*,*h*	级联结构 sos	格形结构 *k*
传递函数 *b*,*a*		tf2ss	tf2zpk tf2zp roots	residue residuez	tf2sos	tf2latc
状态空间 *A*,*B*,*C*,*D*	ss2tf	ss2ss	ss2zp			
零极增益 *z*,*p*,*k*	zp2tf poly	zp2ss			zp2sos	
部分分式 *r*,*p*,*h*	residue residuez					
级联结构 sos	sos2tf	sos2ss	sos2zp			
格形结构 *k*	latc2tf					

MATLAB提供了多种形式的系统参数之间的变换关系，归纳在一起，方便查看滤波器参数的变换。

要特别**注意**：这个表概括了连续系统和离散系统的变换，所以输入变元的意义在两种情况下可能不同。

例如，传递函数的系数向量a和b在连续系统是用正幂排列，而在离散系统则用负幂排列。

与本章内容有关的MATLAB文件

1. fiftfilt.m，实现零相位滤波。调用格式：

y=filtfilt(B, A, x)

其中B是$H(z)$的分子多项式，A是分母多项式，x是待滤波信号，y是滤波后的信号。

2. grpdelay.m，求系统的群延迟。调用格式：

[gd w]=grpdelay(B, A, N) , 或

[gd F]=grpdelay(B, A, N, FS)

其中B和A仍是 $H(z)$ 的分子、分母多项式，gd是群延迟，w、F是频率分点，二者的维数均为N；FS为抽样频率，单位为Hz。

3．tf2latc.m 和latc2tf.m

实现转移函数和Lattice 系数之间的相互转换。tf2latc的调用格式是:

(1) k=tf2latc(b)

(2) k=tf2latc(1,a)

(3) [k, c]=tf2latc(b,a)

其中（1）对应全零系统，（2）对应全极系统，（3）对应极－零系统。

latc2tf的调用格式和tf2latc正好相反。

需要说明的是，tf2latc求出的Lattice系数k和本书求出的k差一个负号，这是由于我们在图中用的是$-k$。

●**tf2latc** Convert transfer function filter parameters to lattice filter form

●Syntax

```
[k,v] = tf2latc(b,a)
k = tf2latc(1,a)
[k,v] = tf2latc(1,a)
k = tf2latc(b)
k = tf2latc(b,'phase')
```

●Description

`[k,v] = tf2latc(b,a)` finds the lattice parameters k and the ladder parameters v for an IIR (ARMA) lattice-ladder filter, normalized by a(1). Note that an error is generated if one or more of the lattice parameters are exactly equal to 1.

`k = tf2latc(1,a)` finds the lattice parameters k for an IIR all-pole (AR) lattice filter.

`[k,v] = tf2latc(1,a)` returns the scalar ladder coefficient at the correct position in vector v. All other elements of v are zero.

`k = tf2latc(b)` finds the lattice parameters k for an FIR (MA) lattice filter, normalized by b(1).

`k = tf2latc(b,'phase')` specifies the type of FIR (MA) lattice filter, where '*phase*' is

- 'max', for a maximum phase filter.
- 'min', for a minimum phase filter.

- **latc2tf** Convert lattice filter parameters to transfer function form
- **Syntax**

```
[num,den] = latc2tf(k,v)
[num,den] = latc2tf(k,'iiroption')
num = latc2tf(k,'firoption')
```

- **Description**

`[num,den] = latc2tf(k,v)` finds the transfer function numerator `num` and denominator `den` from the IIR lattice coefficients `k` and ladder coefficients `v`.

`[num,den] = latc2tf(k,'`*`iiroption`*`')` produces an IIR filter transfer function according to the value of `'`*`iiroption`*`'`:

- `'allpole'`: Produces an all-pole filter transfer function from the associated all-pole IIR lattice filter coefficients `k`.
- `'allpass'`: Produces an allpass filter transfer function from the associated allpass IIR lattice filter coefficients `k`.

`num = latc2tf(k,'`*`firoption`*`')` produces an FIR filter according to the value of `'`*`firoption`*`'`:

- `'min'`: Produces a minimum-phase FIR filter numerator from the associated minimum-phase FIR lattice filter coefficients `k`.
- `'max'`: Produces a maximum-phase FIR filter numerator from the associated maximum-phase FIR lattice filter coefficients `k`.
- `'FIR'`: Produces a general FIR filter numerator from the lattice filter coefficients `k` (this is equivalent to not specifying `'`*`iiroption`*`'` or `'`*`firoption`*`'`).

4. latcfilt.m

用来实现Lattice 结构下的信号滤波。调用格式是:

(1) [y, g]=latcfilt(k, x): 对应全零系统

(2) [y, g]=latcfilt(k, 1, x): 对应全极系统

(3) [y, g]=latcfilt(k, c, x): 对应极－零系统

x是待滤波的信号，y是用Lattice 结构作正向滤波的输出，g是作反向滤波的输出。

若输入x是$\delta(n)$，则输出y是$H(z)$的系数，g 是$z^{-(N-1)}H(z^{-1})$的系数。

- **latcfilt** Lattice and lattice-ladder filter implementation
- **Syntax**

```
[f,g] = latcfilt(k,x)
[f,g] = latcfilt(k,v,x)
[f,g] = latcfilt(k,1,x)
[f,g,zf] = latcfilt(___,"ic",zi)
[f,g,zf] = latcfilt(___,dim)
```

- **Description**

`[f,g] = latcfilt(k,x)` filters input signal x with the FIR lattice coefficients specified by k and returns the forward lattice filter result f and backward filter result g.

`[f,g] = latcfilt(k,v,x)` filters input signal x with the IIR lattice coefficients specified by k and ladder coefficients specified by v. Both k and v must be vectors, while x can be a matrix.

`[f,g] = latcfilt(k,1,x)` filters input signal x with the IIR lattice coefficients specified by k and returns the all-pole lattice filter result f and the allpass filter result g.

`[f,g,zf] = latcfilt(___,"ic",zi)` specifies the initial conditions of the lattice states zi and returns the final conditions of the lattice states zf.

`[f,g,zf] = latcfilt(___,dim)` filters x along the dimension dim.

更多分析　　参考分析代码

5. tf2ss.m 和 ss2tf.m

实现转移函数和相应状态变量之间的转换。二者的调用格式分别是:

[A, B, C, D]=tf2ss(b, a), [b, a]=ss2tf(A, B, C, D)

其中b, a 分别是$H(z)$分子、分母多项式的系数向量，A, B, C及D的定义见书。

6. sos2ss.m

实现由转移函数的二阶级联形式转换为状态变量表示。调用格式:

[A, B, C, D]=sos2ss(sos, g)

A, B, C, D的定义见书。

有关sos和g的说明见2.8节关于tf2sos.m的说明。