第4章 差分方程与滤波

中国科学技术大学

曾凡平

http://staff.ustc.edu.cn/~billzeng

第4章 主要内容
(数字信号处理装置的描述)

- 本章将介绍滤波器，它是数字信号处理中的基本单元之一。数字滤波器可用差分方程来实现。本章内容包括：

1. 介绍基本的滤波概念，包括增益和带宽；比较模拟滤波器和数字滤波器
2. 定义线性、时不变、因果系统
3. 用递归和非递归差分方程实现滤波器
4. 阐述叠加原理
5. 说明差分方程流图
6. 介绍脉冲响应
7. 介绍阶跃响应

4.1 滤波器的基础知识

- 滤波器是以特定方式改变信号的频率特性，从而变换信号的系统。
- 滤波器通常有以下四种：
  1. **低通滤波：** 消除高频信号 如音乐中的背景噪声消除。
  2. **高通滤波：** 滤除低频信号 如声纳系统用于目标的识别保留了高频信号（目标特性）。
  3. **带通滤波：** 允许一定频带内的信号通过，提取感兴趣的信号，如电话机按键信号的解码（双音多频DTMF）。
  4. **带阻滤波：** 允许一定频带以外的信号通过，过滤无用的信号。如消除音乐中的某个单音。

图4.1 双音多频DTMF信号的解码

- 数字电话的每一个按键对应键盘网格上的一对频率。
- 接收端的一组带通滤波器，通过检测信号是7个频率中的哪两个频率来区别每个按键。

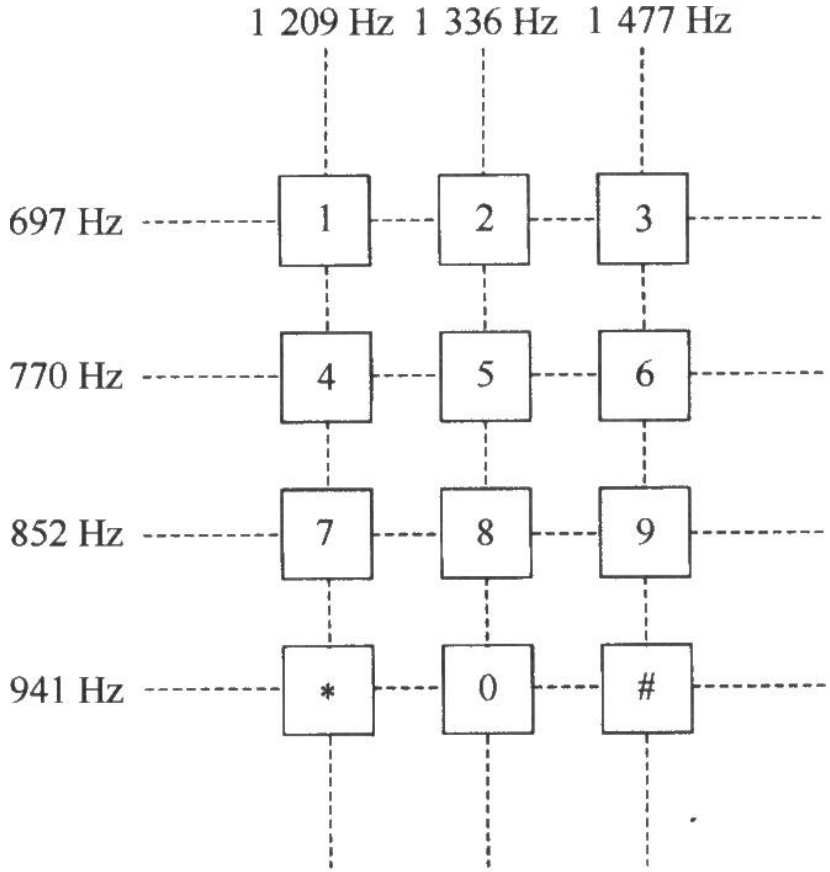


图 4.1　双音多频(DTMF)信号

滤波器的特性通过它的频域形状来描述
(幅度频率特性图-频谱图)

截止频率：
增益为最大值的0.707所对应的频率

带宽(bandwidth)：
信号可以通过的频率范围

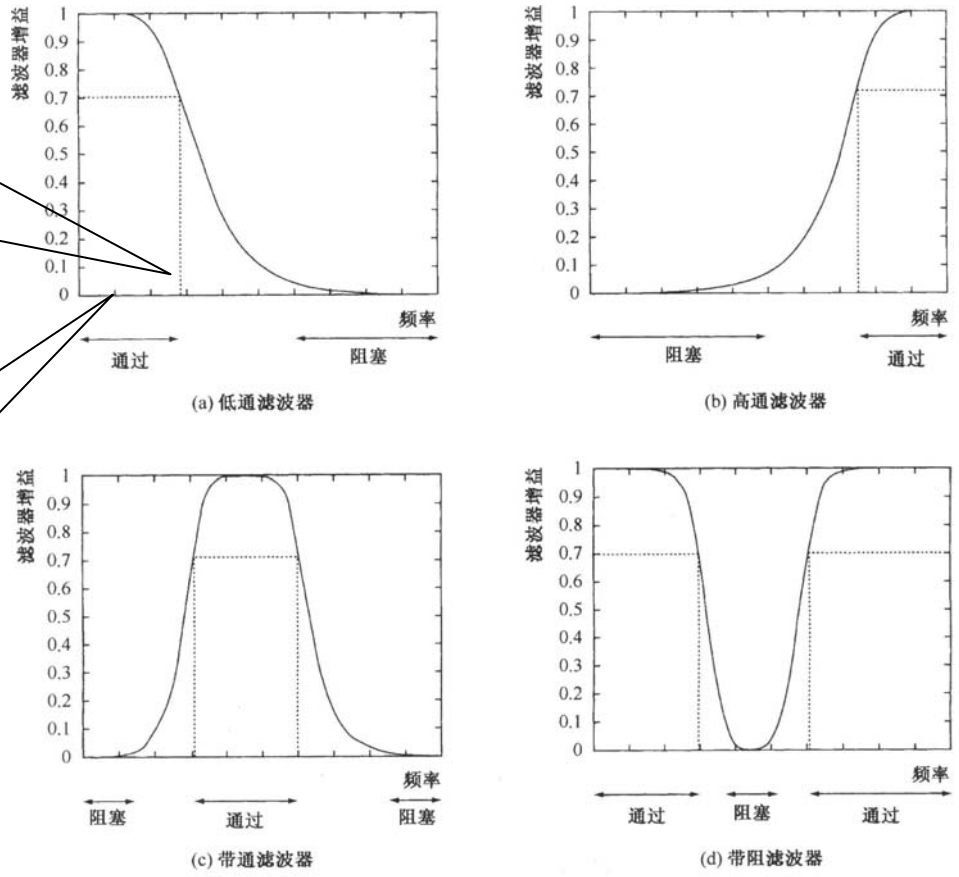


图 4.2　常用滤波器类型

滤波器的频域参数

- 截止频率：增益为最大值的 $\frac{1}{\sqrt{2}}=0.707$ 所对应的频率。

- 增益(dB)=20log(线性增益)：-3dB↔0.707，-3dB的频率↔截止频率。
- 带宽（bandwidth）：信号可以通过的频率范围
  - 从0.707对应的增益作水平线，与幅频特性的交点为带宽的起点或终点。
- 增益：滤波器对输入的该频率正弦信号的放大倍数
  - 例：输入为r(t)=5sin(100πt)通过滤波器，信号频率=50Hz
    若滤波器在50Hz的增益A(50)=0.6，则输出信号为

$$y(t)=5A(50)\sin[100\pi t+\theta(50)]=3\sin[100\pi t+\theta(50)]$$

$\theta(50)$：滤波器对50Hz频率的相移

例 4.1　求出图 4.3 所示带通滤波器的带宽。

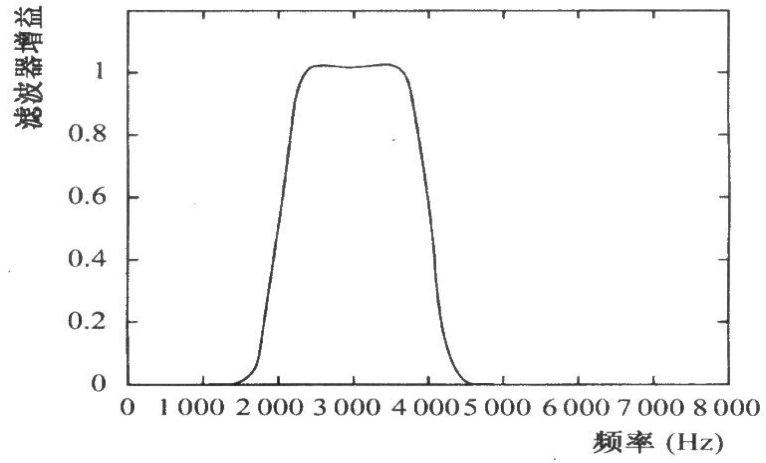


图 4.3　例 4.1 的带通滤波器

解：

通带的边缘频率为 0.707 所对应的频率(如图 4.4 所示)，则滤波器的带宽略小于 2 000 Hz。

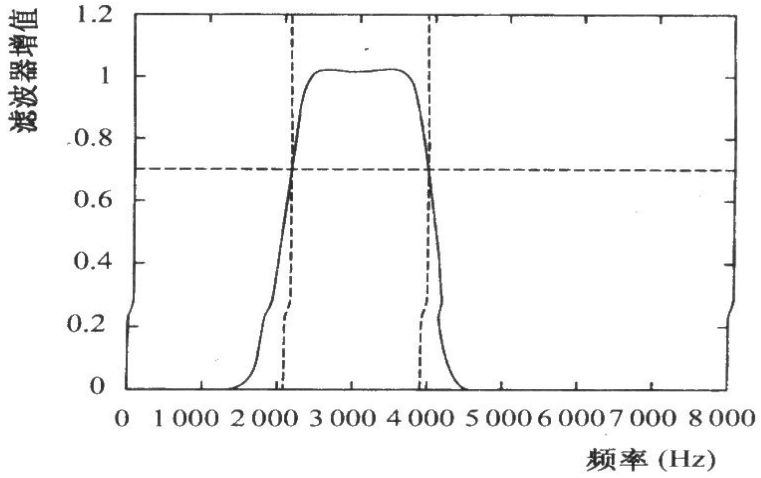


图 4.4　例 4.1 的通带计算

滤波器的作用

- 每一种滤波器对输入信号的作用不同。
- 低通滤波器可以平滑信号的突变；相反，高通滤波器可以强化信号的锐变。

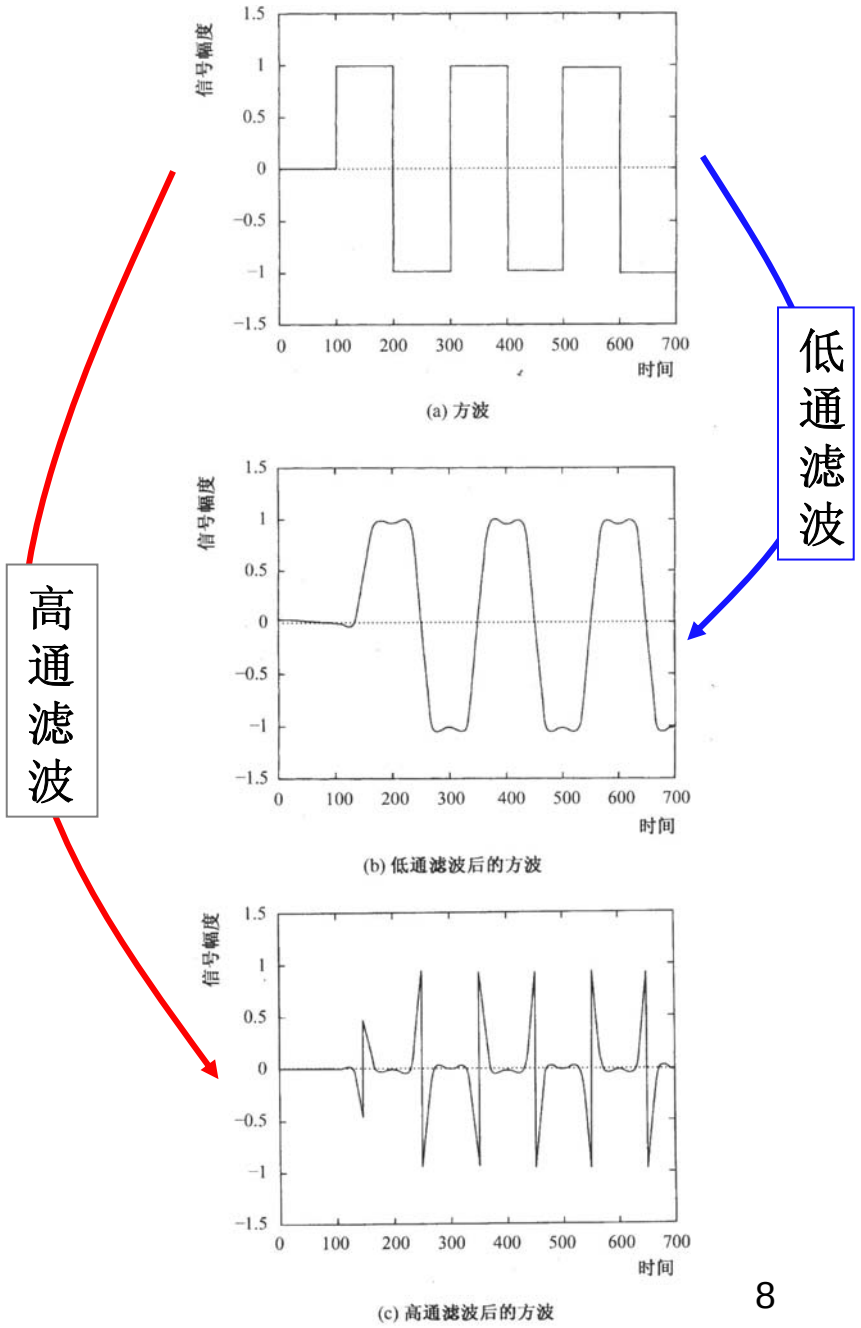
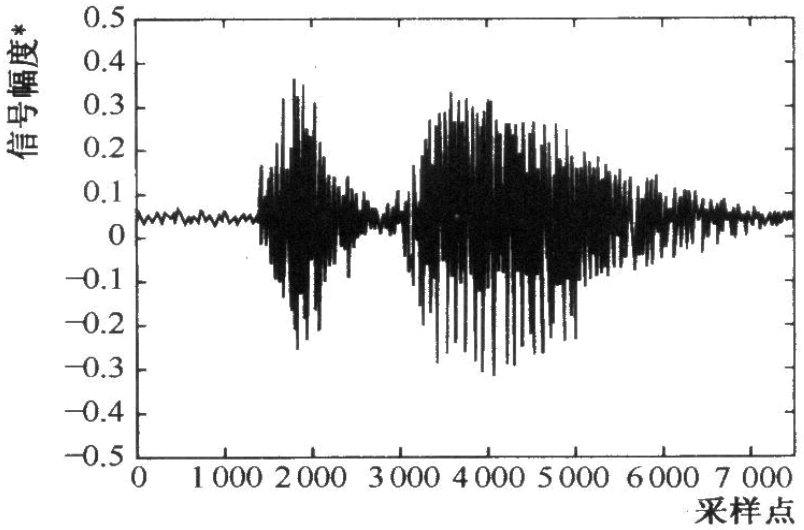
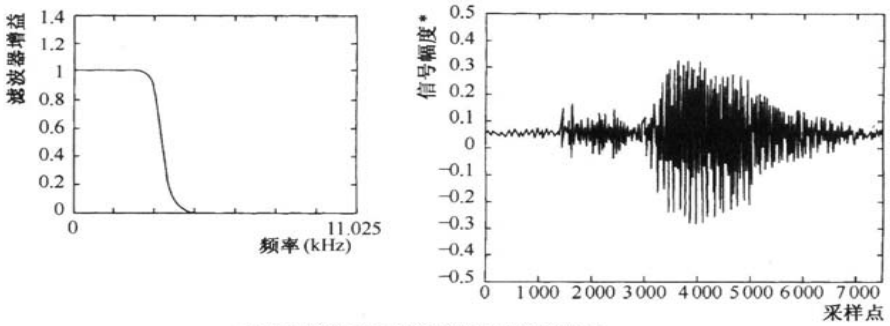


图 4.5 低通滤波器和高通滤波器的作用

- 图4.6说明了不同方式对语音进行滤波可以获得不同的频率分量。单词“two”的发音用22.05 kHz进行采样并用一组10阶巴特沃斯滤波器(将在10.4节中介绍)进行滤波。低通、高通和带通滤波器的频率特性由增益对频率(Hz)画出。
- 图4.6(a)所示原始信号由两个主要部分组成：“t”音爆破时的嘈杂高频和“oo”的低频周期部分，低频周期部分正如采样点3 200和5 000之间的规则间隔的尖峰。低通滤波器选出低频部分，衰减了高频分量，如图4.6(b)所示。

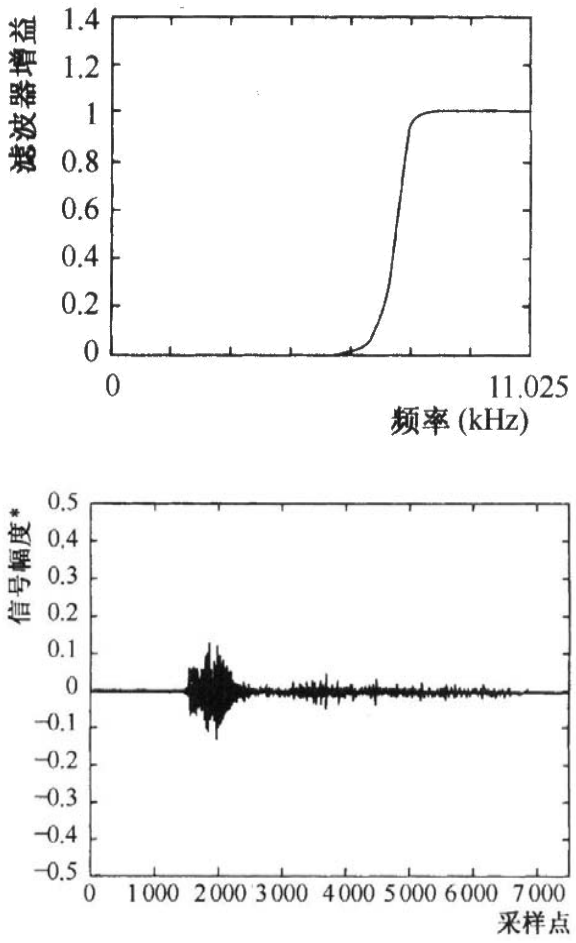


(a) 原语音信号“two”的采样值

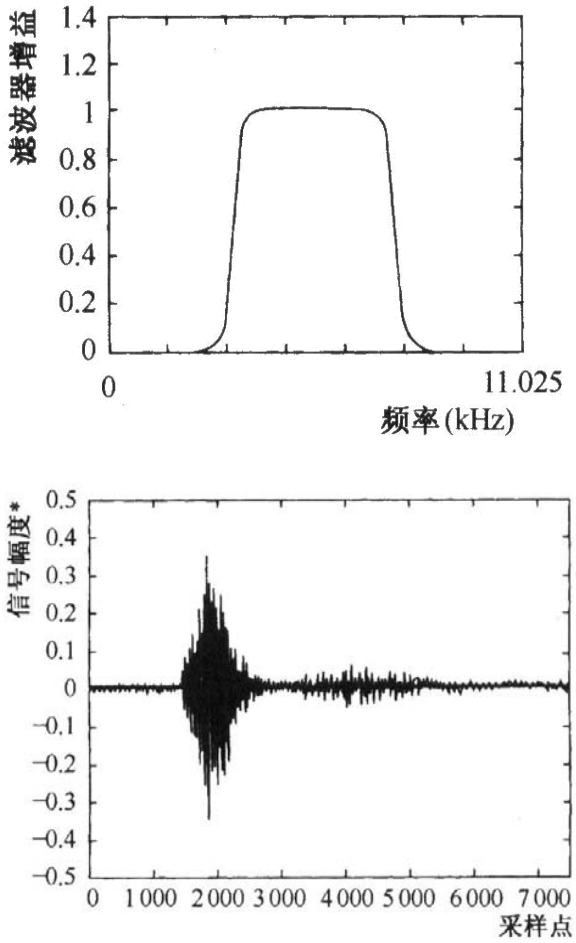


(b) 低通滤波器及低通滤波后的语音信号

- 图4.6(c)表示高通滤波器选出的频率分量，这一部分频率分量在原信号中较少。

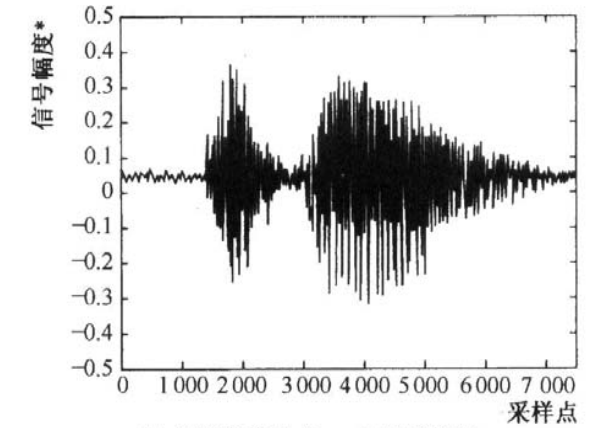


- “t”音的爆破大部分由中频组成，它们由带通滤波器选出，如图4.6(d)所示。

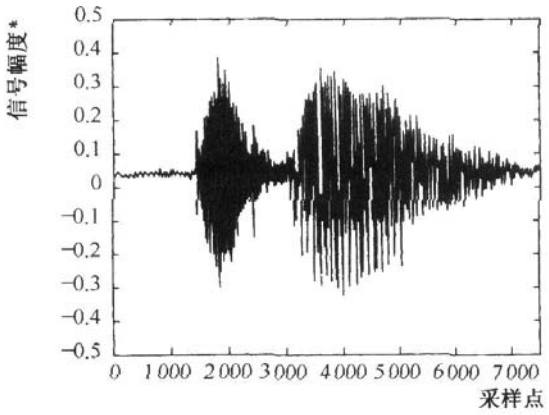


- 图4.6(e)是低通、高通和带通输出信号的和，它非常接近原始信号。

- 由于滤波器不是理想滤波器，所以与原信号相比略有不同。



(a) 原语音信号“two”的采样值

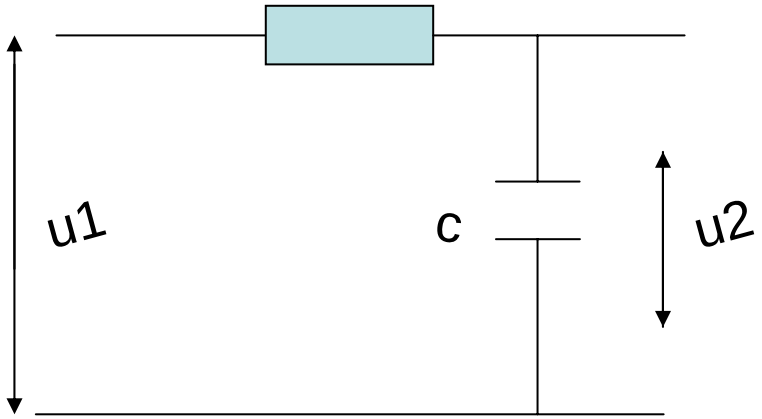


(e) 低通、高通、带通滤波后的和信号

4.2 模拟滤波器和数字滤波器

- 前节所述的信号滤波可以用模拟滤波器或数字滤波器实现。
- 数字滤波器具有模拟滤波器无法比拟的优点，所以在许多情况下宁可选用数字滤波器。
- 模拟滤波器是由电阻、电容和电感等部件构成的电路，这样，滤波器特性对所用部件的值非常敏感，而有些部件的特性随温度变化很大。

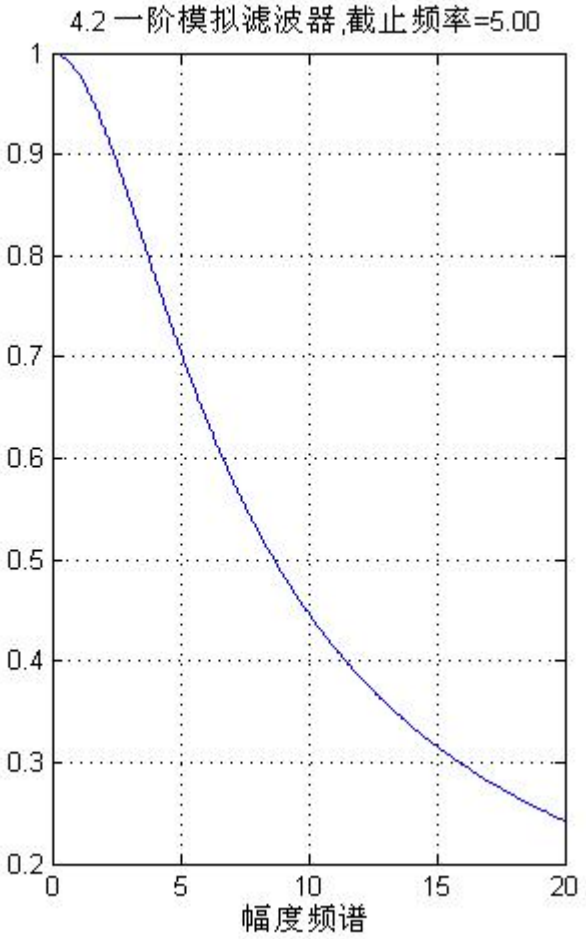
模拟滤波器的例子

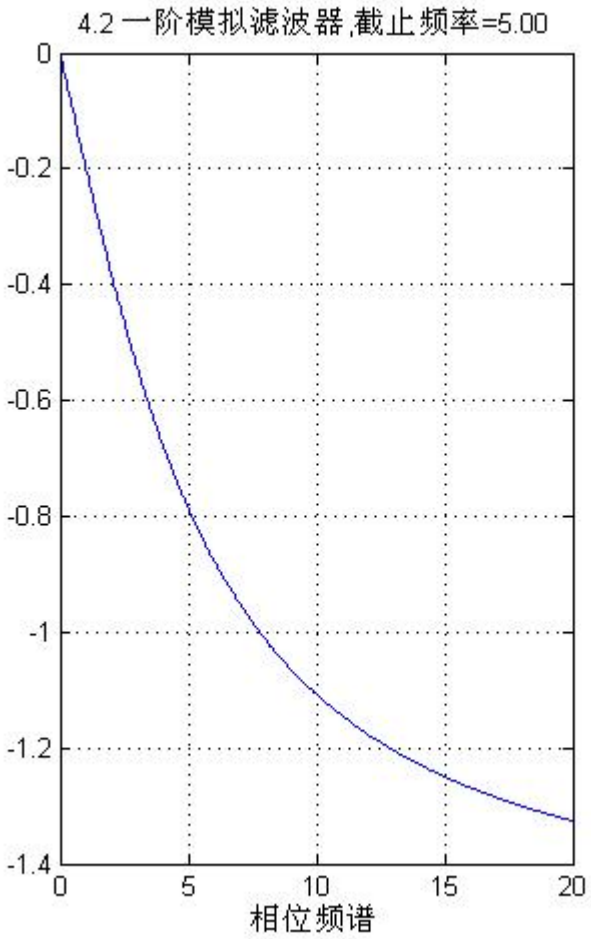


$$\frac{u_2(s)}{u_1(s)} = \frac{1}{RCs+1} = \frac{1}{Ts+1} = G(s)$$

$$\text{频率特性} = |G(j\omega)| = \left|\frac{1}{j\omega T+1}\right| = \frac{1}{\sqrt{\omega^2 T^2+1}}$$

$$\text{截止频率} = 1/\mathrm{T}$$





模拟滤波器和数字滤波器的性能比较

- 数字滤波器：由DSP程序中的系数决定的，对外界的变化不敏感。
- 另外，高阶复杂的模拟滤波器的设计和参数的调整极为困难，相比之下，高阶数字滤波器容易实现，只是增加滤波器系数就行了。
- 当然数字滤波器也不是一点毛病也没有，量化噪声和采样所引起的混叠就会带来噪声，此外，数字运算有时的舍入和溢出引起的误差亦应该引起重视。

模拟和数字滤波器的性能比较(2)

- 但总的来说，数字滤波器的优点比缺点更多，在能用数字滤波器的场合一般选用数字滤波。

- 数字滤波器程序的实现有两种主要的方式。一种是用滤波器的差分方程(见下一节)计算滤波器的输出，另一种是利用卷积过程(见第5章)计算输出。后一种方法需要滤波器的脉冲响应(见4.7节)。

4.3 线性、时不变、因果系统

- 线性系统：满足叠加原理 若 $y(t)=f[x(t)]$

$$f[a\cdot x_1(t)+b\cdot x_2(t)]=a\cdot f[x_1(t)]+b\cdot f[x_2(t)]=a\cdot y_1(t)+b\cdot y_2(t)$$

- 例如模拟运算放大器在其输入范围内是线性的，电容，电阻，杠杆；平方系统不满足线性性质（叠加原理），为非线性系统。
- 又如：电磁铁的磁道量与电流为非线性关系，但可以在静态工作点附近近似成为线性系统，采用微偏线性化方法，取一阶导数项，忽略高阶导数项。

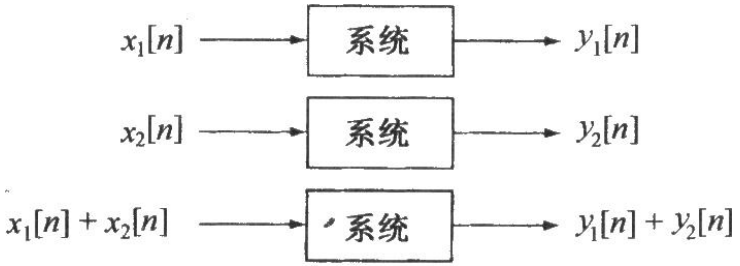


图 4.7　线性系统

时不变、因果系统

- **时不变系统**：输入延迟，输出也有相同的延迟；系统无论什么时间加上输入，输出都是相同的；
- **因果系统**：输出取决于现在和以前的数据，而与将来的数据无关。
- DSP系统为因果系统，本课程只考虑线性、时不变、因果系统。

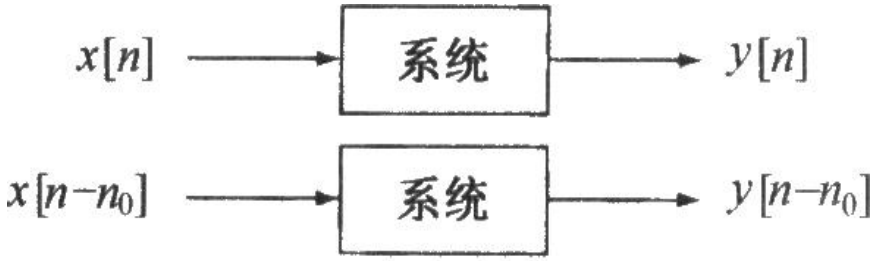


图 4.8　时不变系统

4.4 差分方程

- 差分方程(difference equation)可用来描述线性、时不变、因果数字滤波器。
- 一般来讲，滤波器的输出依赖于现在和以前的输入，也依赖于过去的输出。通常用 x 表示滤波器的输入，用 y 表示滤波器的输出。如果现在的输入为 $x[n]$，则前一输入为 $x[n-1]$，再前一个为 $x[n-2]$，每个值之间有一个采样周期的延迟。类似地，过去的输出为 $y[n-1]$，$y[n-2]$，等等。
- 差分方程的一般表达式为：

$$\begin{aligned} &a_0 \cdot y[n] + a_1 \cdot y[n-1] + \cdots + a_N \cdot y[n-N] \\ &= b_0 \cdot x[n] + b_1 \cdot x[n-1] + \cdots + b_M \cdot x[n-M] \end{aligned} \qquad (4.1)$$

差分方程的各种形式

- 方程式的左侧为输出，右侧为输入。a_k，b_k为**权系数**，决定了每个输入和输出的贡献大小。这些权系数称为**滤波器系数(filter coefficient)**。

- 式(4.1)为差分方程的一般表达式，有N+1个系数a_k和M+1个系数b_k。

- 这个方程还可以写成更紧凑的形式：

$$\sum_{k=0}^{N} a_k \cdot y[n-k] = \sum_{k=0}^{M} b_k \cdot x[n-k] \quad (4.2)$$

  - M是所需以前输入的个数。
  - N为所需过去输出的个数，max(M,N)称为**滤波器的阶数**

- 令a_0=1,则

$$y[n] = -\sum_{k=1}^{N} a_k \cdot y[n-k] + \sum_{k=0}^{M} b_k \cdot x[n-k] \quad (4.3)$$

递归和非递归滤波器

- 此式表明怎样从过去的输出、现在的输入和以前的输入计算滤波器每一个新的输出。
- 当数字系统依赖于输入和过去的输出，称其为**递归滤波器(recursive filter)(有反馈)**。式(4.3)给出了这类滤波器的一般表示式。递归滤波器将在第10章学习。

$$y[n] = -\sum_{k=1}^{N} a_k \cdot y[n-k] + \sum_{k=0}^{M} b_k \cdot x[n-k] \qquad (4.3)$$

非递归滤波器

- 当数字滤波器仅依赖于输入，而不依赖过去的输出，被称为**非递归滤波器(nonrecursive filter)(无反馈)**。式(4.4)给出了这类滤波器的一般式。非递归滤波器将在第9章学习。

$$y[n]=\sum_{k=0}^{M} b_k \cdot x[n-k] \qquad (4.4)$$
$$=b_0 \cdot x[n]+b_1 \cdot x[n-1]+\cdots+b_M \cdot x[n-M]$$

例4.2 递归滤波器的例子

例4.2 一个滤波器的差分方程为：

$$y[n]=0.5y[n-1]+x[n] \quad (4.5)$$

a.确定所有系数ak，bk。

b.它是递归滤波器还是非递归滤波器?

c.如果输入x[n](如图4.9所示)，从n=0开始求出前12个输出。

解：

a.将差分方程重新改写，输出放在左侧，输入放在右侧，则有：

$$y[n]-0.5y[n-1]=x[n]$$

系数的值很易确定，参照式(4.1)滤波器的系数为：a0=1.0，a1=-0.5及b0=1.0，除 a0，a1，b0外其他所有系数为零。

b.由于输出y[n]取决于过去的输出y[n-1]，所以数字滤波器是递归滤波器。

c.输出可以从n=0开始，通过反复计算式(4.5)求出。对于n=0这种情况，计算时需要输出y[-1]。本书中，

- **假定数字滤波器是因果的，这就意味着直到第一个输入不为零时，输出才开始变化。**

例4.2 计算递归滤波器的输出

y[0] = 0.5y[-1]+x[0]
=0.5×(0.0000)+1.0=1.0000
y[1] = 1.5000
y[2] = 1.7500
y[3] = 1.8750
y[4] = 1.9375
y[5] = 1.9688
y[6] = 1.9844
y[7] = 1.9922
y[8] = 1.9961
y[9] = 1.9980
y[10]=1.9990
y[11]=1.9995

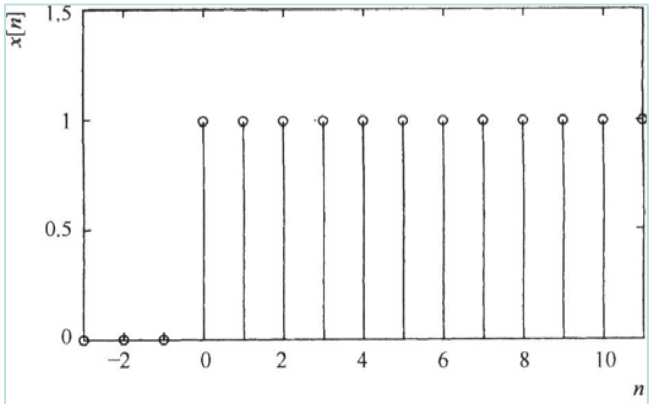


图 4.9　例 4.2 中滤波器的输入

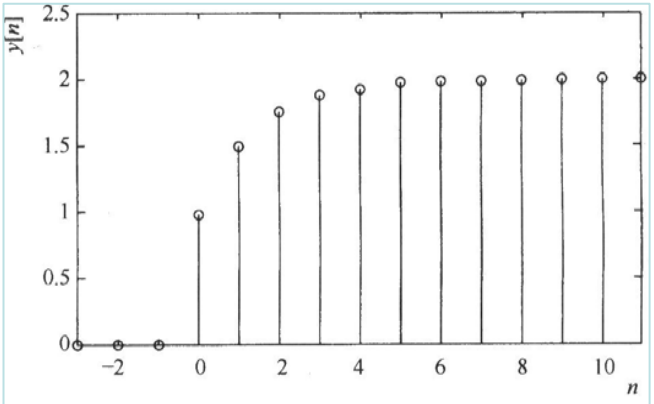


图 4.10　例 4.2 中滤波器的输出

差分方程的Matlab表示及实现

- 在Matlab环境中，用系数向量b和a表示差分方程：
  - b=[b0,b1,b2,……]; %缺的项用0代替，左边的0不省略
  - a=[a0,a1,a2,……]; %缺的项用0代替
- 如果已知输入*x*[*n*]，表示为向量x：
  - *x*=[*x0*,*x1*,*x2*,……];
- 则输出向量y可以用以下M*atlab*命令求出：
  - *y*=*filter*(*b*, *a*, *x*); **（演示）**

例4.3 非递归滤波器

例4.3

$$y[n]=0.5x[n]-0.3x[n-1]$$

a.确定所有系数*ak*，*bk*。

b.它是递归滤波器还是非递归滤波器差分方程?

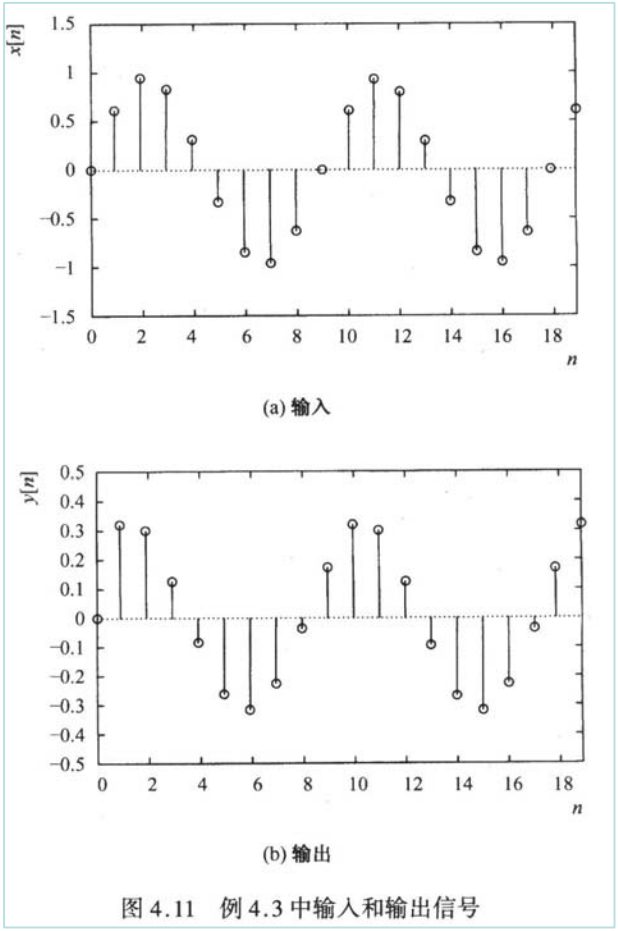
c.输入*x[n]=sin(n2 ／9)u[n]*，求出前20个输出。

解：

a. *a0=1.0*，*b0=0.5*及*b1=-0.3*。

b. 由于输出不取决于过去的输出，所以数字滤波器是非递归滤波器。

c. 由于输入中有*u[n]*，所以n=0以前的输入为零。表4.1给出了输入和输出的前20个值。图4.11给出了输入和输出的图形。注意，虽然两个信号的幅度和相位不同，但它们均具有正弦特性和相同的数字周期。



(a) 输入

(b) 输出

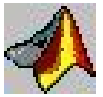
图 4.11　例 4.3 中输入和输出信号

4.5 叠加原理

- 有时，几个输入同时加到滤波器上，此时滤波器的响应要应用叠加原理。当滤波器是线性的，多个输入的情况较易处理。如图4.7，总的输出可用两种方法求取。
  - **第一种方法**是分别计算每一个输入的输出，然后把它们的输出加起来得到总的输出信号。
  - **第二种方法**是先把所有输入加起来，然后求取滤波器对这个**和信号**的响应。
- 计算结果相同，见例4.4。

例 4.4 滤波器用差分方程描述为：

$$y[n]=x[n]+0.5x[n-1]$$



两个输入(如图 4.12 所示)加到滤波器上，它们分别是：

$$x_1[n] = 2u[n]$$

$$x_2[n] = \sin(n\pi/7)u[n]$$

求出两个信号共同产生的前 20 个输出，并画出图。

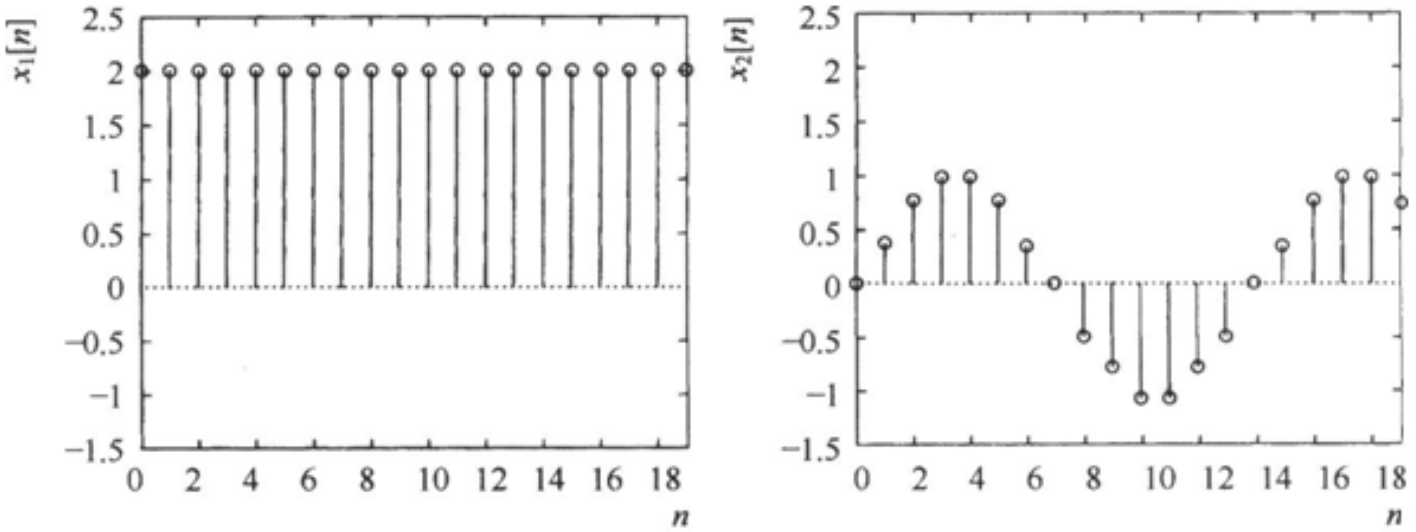


图 4.12 例 4.4 的输入信号

解：输入为两部分：$x_1[n]=2u[n]$，$x_2[n]=\sin(n\pi/7)u[n]$。每个输入所对应的输出 y1[n]，y2[n]可以单独求出，然后把两个输出相加得到总的输出。注意输入在 n=0 之前为零。计算出的前 20 个输出如表 4.2 和图 4.13 所示。比如 n=3 的情况，表 4.2 中用黑体表示。x1[n]的输出 y1[n]由下式给出：

$$y1[n]=x1[n]+0.5x1[n-1]$$

这样，y1[3]=x1[3]+0.5x1[2]=2+0.5*(2)=3。x2[n]的输出 y2[n]为：

$$y2[n]=x2[n]+0.5x2[n-1]$$

这样，y2[3]=x2[3]+0.5x2[2]=0.975+0.5×(0.782)=1.37，两个信号总的输出为 y1[n]+y2[n]，对于 n=3，Y1[3]+Y2[3]=3+1.37=4.37。把两个输入信号加起来同样可以得到相同的结果。在加黑的行上，y[3]=x[3]+0.5x[2]=2.975+0.5×(2.782)=4.37。实际上，输出列 y1[n]+y2[n]可以直接从输入之和列 x1[n]+x2[n]计算得到，而不需要经过中间 y1[n]，y2[n]两列。

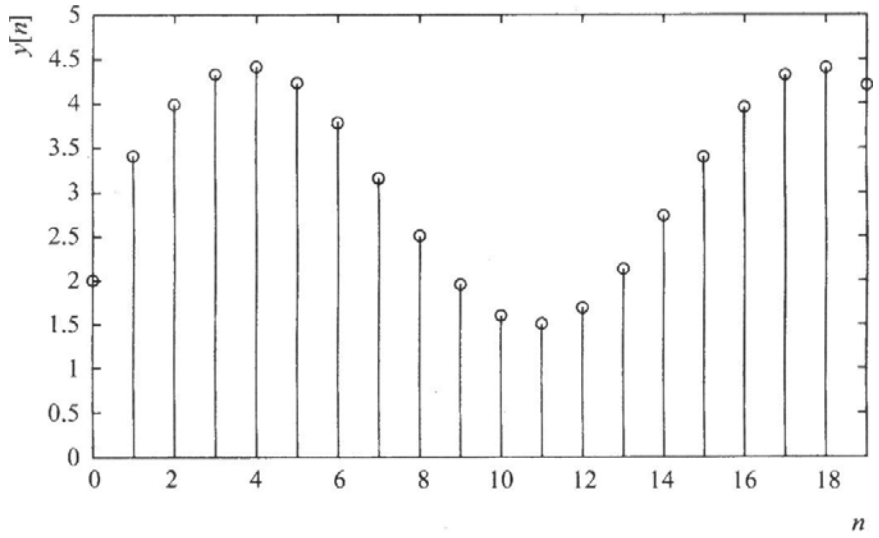


图 4.13 例 4.4 的输出信号

表 4.2　例 4.4 中输出信号的计算

n	$x_1[n]$	$x_2[n]$	$x[n]=x_1[n]+x_2[n]$	$y_1[n]$	$y_2[n]$	$y[n]=y_1[n]+y_2[n]$
0	2	0.00	2.00	2	0.00	2.00
1	2	0.434	2.434	3	0.43	3.43
2	**2**	**0.782**	**2.782**	3	1.00	4.00
3	**2**	**0.975**	**2.975**	**3**	**1.37**	**4.37**
4	2	0.975	2.975	3	1.46	4.46
5	2	0.782	2.782	3	1.27	4.27
6	2	0.434	2.434	3	0.82	3.82
7	2	0.000	2.000	3	0.22	3.22
8	2	−0.434	1.566	3	−0.43	2.57
9	2	−0.782	1.218	3	−1.00	2.00
10	2	−0.975	1.025	3	−1.37	1.63
11	2	−0.975	1.025	3	−1.46	1.54
12	2	−0.782	1.218	3	−1.27	1.73
13	2	−0.434	1.566	3	−0.82	2.18
14	2	0.000	2.000	3	−0.22	2.78
15	2	0.434	2.434	3	0.43	3.43
16	2	0.782	2.782	3	1.00	4.00
17	2	0.975	2.975	3	1.37	4.37
18	2	0.975	2.975	3	1.46	4.46
19	2	0.782	2.782	3	1.27	4.27

4.6 差分方程流图

- 流图的基本单元（4个）见图4.14 P87
  1. 延迟单元：对输入滞后一个采样点，得到前一个数据。
  2. 系数乘法器：对输入信号按比例放大。
  3. 加法器：对输入信号进行叠加（相加）。
  4. 分支点：改变信号的传递路径。

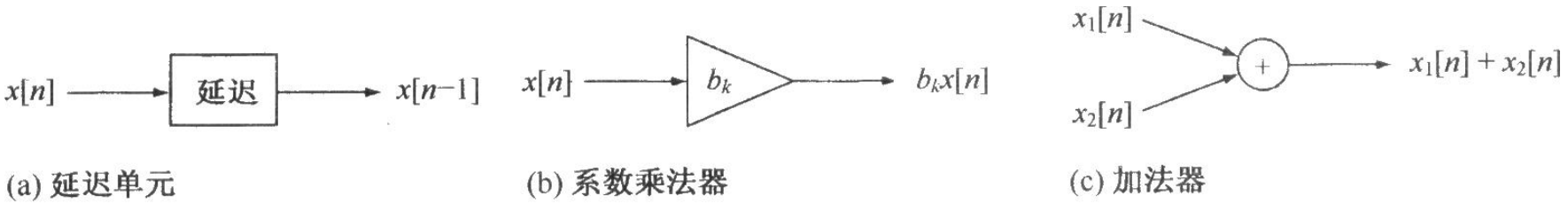


图 4.14　差分方程流图单元

系数的量化和有限字长效应

- 差分方程流图常用做软件实现滤波器的模型，滤波器的系数必须量化才能在DSP中实现。量化会带来误差。

- **系数的量化：**系数转化为DSP的内部数字表示。

- **有限字长效应：**由于处理器有效比特数有限而产生的影响，实现滤波器时应该采取措施减小这些效应。

4.6.1 非递归差分方程的流图

- 用图4.15所示的流图实现，这样的实现容易产生有限字长效应。
- 有限字长效应的根源在于系数的量化会带来误差。因此，如果增大滤波器的系数

$$y[n]=\sum_{k=0}^{M} b_k \cdot x[n-k]$$

$$=b_0 \cdot x[n]+b_1 \cdot [n-1]+\cdots+b_M \cdot x[n-M]$$

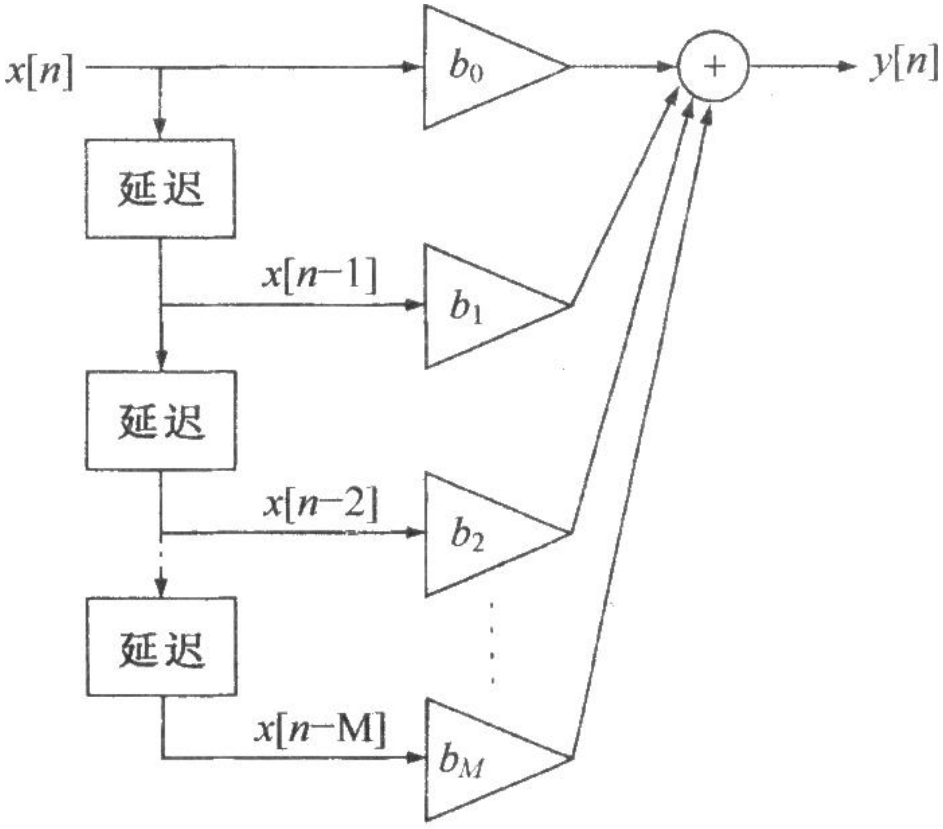


图 4.15　非递归差分方程流图

减小有限字长效应的方法

- 将高阶滤波器分解为(等效为)若干个二阶滤波器串连，这样分解后，每个二阶滤波器的系数比原高阶滤波器的系数大，对量化误差的敏感程度低，从而减小了量化误差对最终输出值的影响，也即减小了有限字长效应。

用级联的方法减小有限字长效应 (系数的量化误差)

- 如图4.16所示。当高阶滤波器用这种方法分解后，每一个二阶滤波器节的系数，平均来说较原滤波器的系数大。这样，对量化误差的敏感程度较低。对于滤波器阶数为奇数的情况，可以在二阶滤波器组里加一个一阶滤波器节。

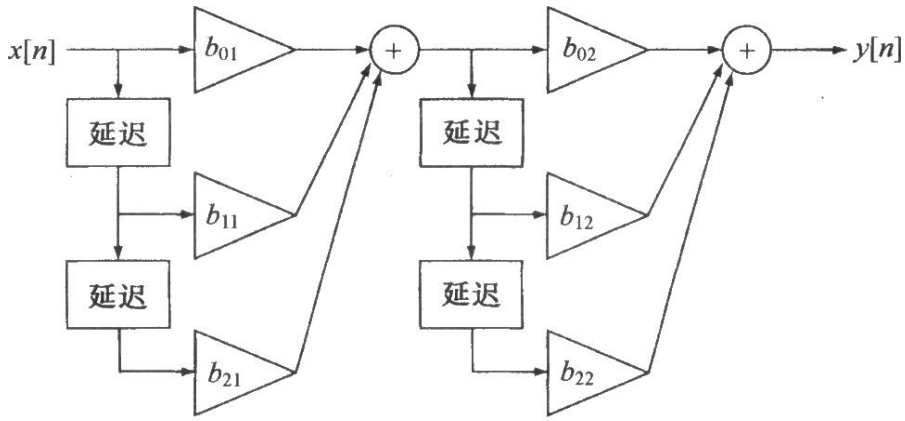


图 4.16　二阶非递归滤波器节的级联

$$y[n]=x[n]-0.1x[n-1]+0.1x[n-2]-0.06x[n-3]+0.02x[n-4]$$

变为
$$\begin{cases} y_1[n]=x[n]-0.5x[n-1]+0.1x[n-2] \\ y[n]=y_1[n]+0.4y_1[n-1]+0.2y_1[n-2] \end{cases}$$

图4.16的量化分析

- 若用8bit来表示系数（-128~127）256个数字，量化单位=1/128
- b = [1,-0.1,0.1,-0.06,0.02];
- b量化后的数字量、误差及(误差/系数)之百分比分别为：

  [128 -13 13 -8 3]

  [0 -0.0016 0.0016 -0.0025 0.0034]

  [0 1.5625 1.5625 4.1667 17.1875]%
- 可见系数越小，量化带来的影响越严重。

级联后的误差分析

- b1=[1.0000　0.4000　0.2000]
- b2=[1.0000　-0.5000　0.1000]
- 量化后的误差及误差与系数之百分比分别为：
  - [0, -0.0016, 0.0031] [0, -0.3906, 1.5625]%
  - [0　　0　0.0016]　[0　　0　1.5625]%
- 可见，最大误差百分数为1.56，小于原来17.18

例4.5 画出下列差分方程的流图：

- $y[n]=0.5x[n]+0.4x[n-1]-0.2x[n-2]$

解：

- 以上差分方程的流图如图4.17所示。

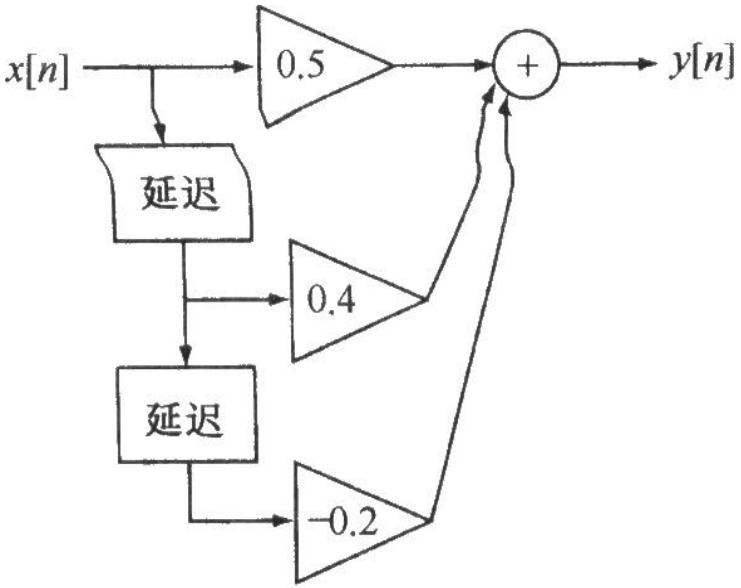


图 4.17　例 4.5 的差分方程流图

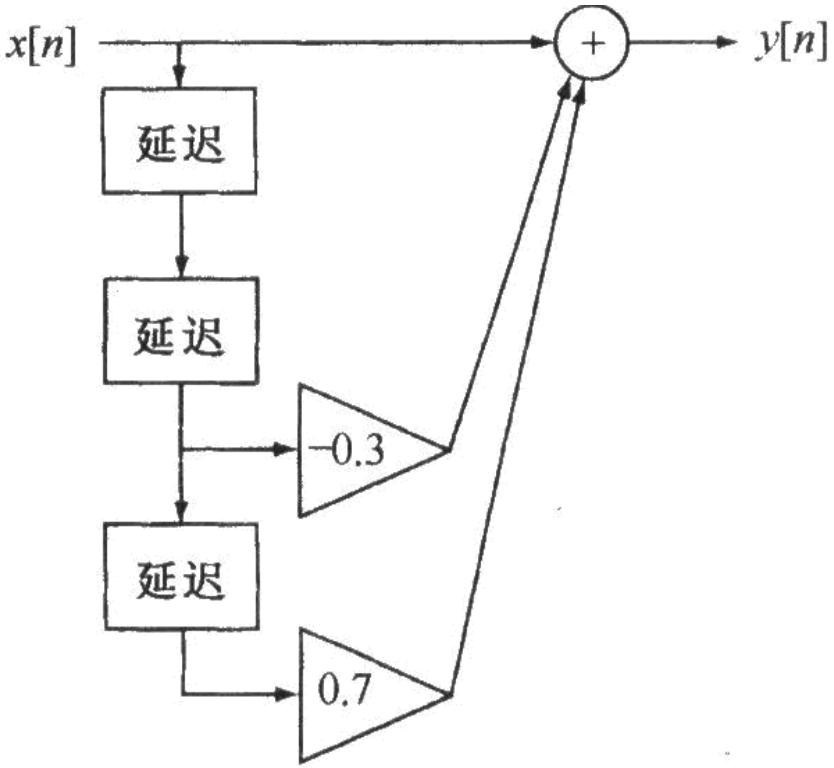


图 4.18　例 4.6 的差分方程流图

例4.6 写出图4.18流图的差分方程。

解：

- $x[n]$的乘数为1，$x[n-1]$项的乘数为0，差分方程是：

$$y[n]=x[n]-0.3x[n-2]+0.7x[n-3]$$

例4.7 写出图4.19级联流图的差分方程。

解：

第一级的差分方程为：

$$y_1[n]=x_1[n]-0.1x_1[n-1]+0.2x_1[n-2]$$

第二级的差分方程为：

$$y_2[n]=x_2[n]+0.3x_2[n-1]+0.1x_2[n-2]$$

第三级的差分方程为：$y_3[n]=x_3[n]-0.4x_3[n-1]$

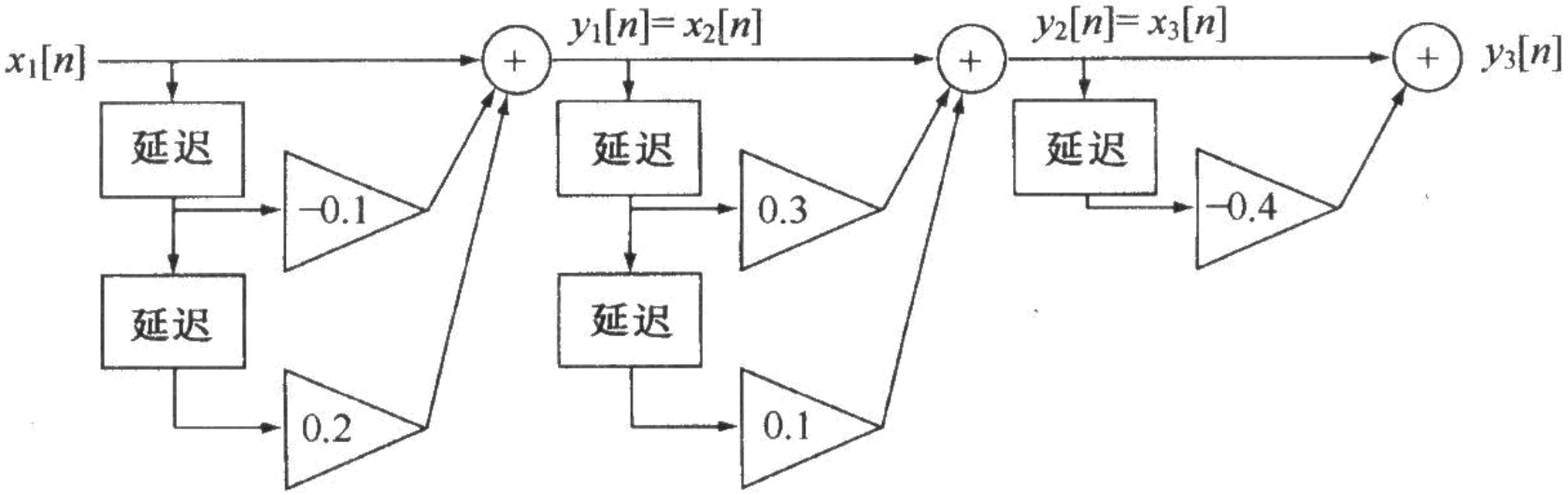


图 4.19 例 4.7 的差分方程流图

4.6.2 递归差分方程

- 4.6.2.1 直接1型实现：
- 计算 $y[n]$需要$M+1$个输入、N个输出、$M+N+1$个系数乘法、$M+N$个加法。

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

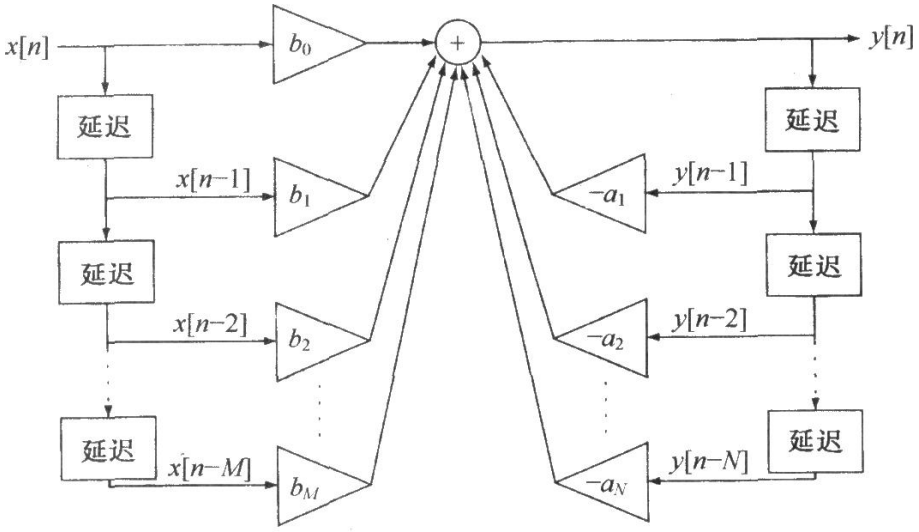


图 4.20　递归差分方程流图(直接 1 型)

- 对有限字长效应非常敏感，同样可以分解为二阶滤波器的串联的方法以减小效应。

例4.8 画出如下差分方程所描述的递归数字滤波器的直接1型流图：

$$y[n]+0.5y[n-2]=0.8x[n]+0.1x[n-1]-0.3x[n-2]$$

解：从方程可知$a0=1.0$，$a1=0.5$，$b0=0.8$，$b1=0.1$和b2=-0.3。

将差分方程重新排列如下：

$$y[n]=-0.5y[n-2]+0.8x[n]+0.1x[n-1]-0.3x[n-2]$$

可以画出流图如图4.21。

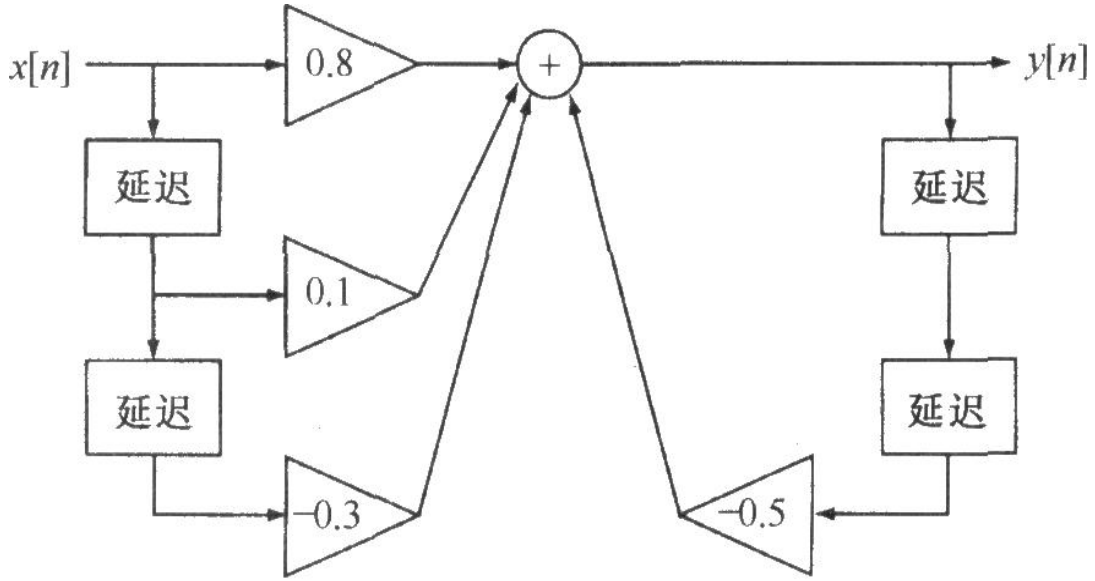


图 4.21　例 4.8 的差分方程流图

例4.9 写出如下流图的差分方程：

解： 差分方程为

$$y[n]=0.1y[n-1]-0.3y[n-2]+0.6y[n-3]-0.8x[n-1]-0.2x[n-3]$$

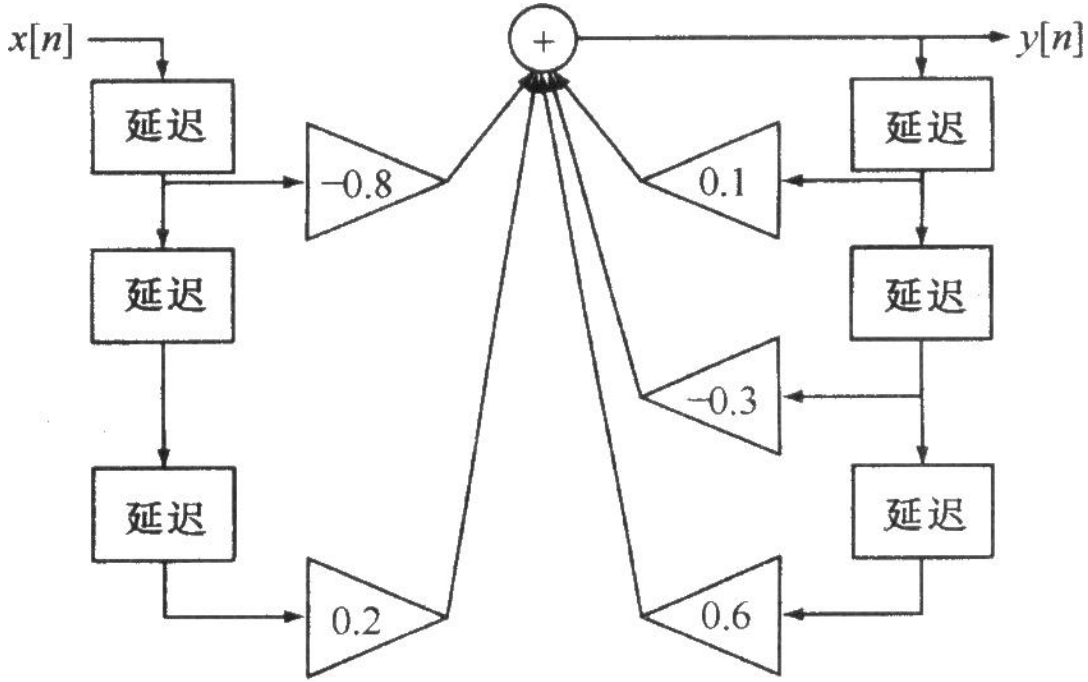


图 4.22　例 4.9 的差分方程流图

4.6.2.2 直接2型实现

- 用4.6.2.1节介绍的直接1型，对于递归差分方程的实现不是最有效的方法。如果采用直接2型(见附录C)，所用的存储器要少得多。这一实现需要中间信号 $w[n]$，它代替过去的输入和输出，记录滤波器历史的重要信息。不需要N个过去的输出和M个过去的输入，仅记忆$max(N, M)$个中间信号的采样值以产生新的滤波器输出。定义直接2型实现的两个方程为：

$$\begin{cases} W[n] = x[n] - \sum_{k=0}^{N} a_k \cdot W[n-k] \\ y[n] = \sum_{k=0}^{N} b_k \cdot W[n-k] \end{cases}$$

直接2型实现的图示

- 用这种实现计算$y[n]$需要$max(N，M)+1$个中间状态、$M+N+1$系数个乘法及$M+N$个加法。由于减少了过去输入和输出状态的存储，直接2型较直接1型节省存储器。

- 直接2型需要两个加法器，在DSP硬件实现时，它有可能引起算术溢出。尽管如此，直接2型(也称为标准型)的存储效率高，在滤波器实现时广泛选用。

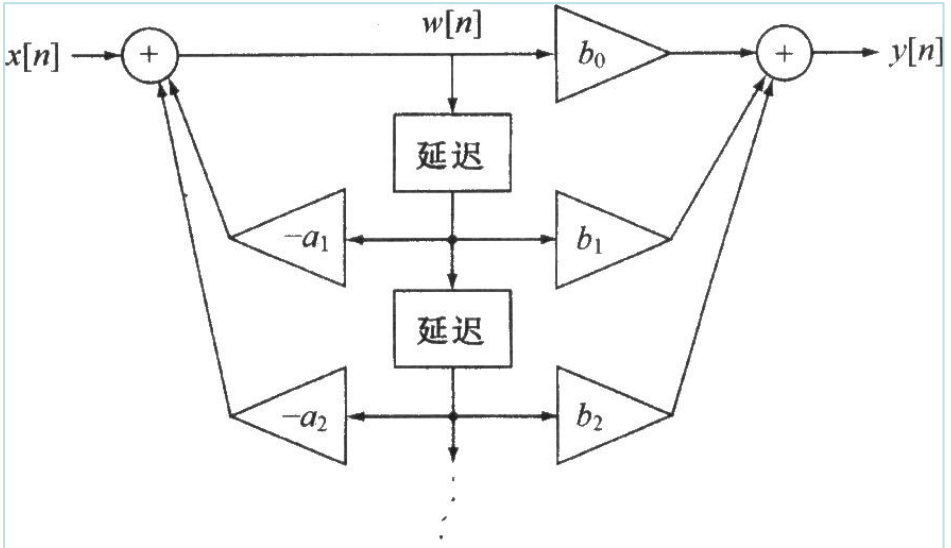


图 4.23　递归差分方程流图(直接 2 型)

转置直接2型实现

- 颠倒图4.23中的信息流向，可得到转置直接2型(如图4.24所示)：
  - 即相同延迟次数的信号相加放在一起。
- 为了减小有限字长效应，通常将直接2型或其转置的二阶块通过级联或并联的方式连接起来，构成高阶滤波器。

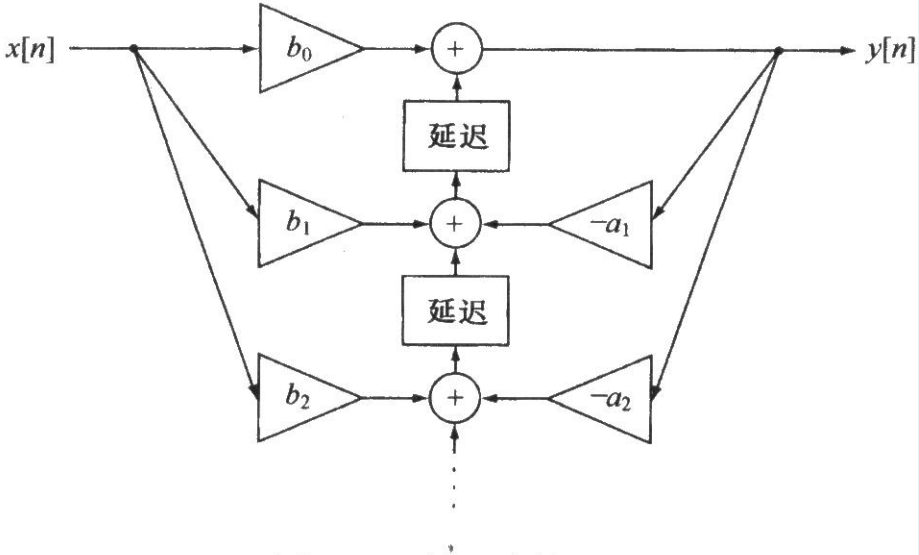


图 4.24　转置直接 2 型

对于递归滤波器，还有其他实现结构。但选择的目标是减少数据的存储、减少乘法和加法的数量，并对计算机中不可避免的有效字长效应尽可能不敏感

例**4.10** 求出图4.25所示流图的滤波器差分方程：

解：

最下面的加法器输出为***0.1x*[*n*]-*0.3y*[*n*]**，延迟一个单位得到

0.1x[n-1]-0.3y[n-1]

中间的加法器输出为

0.1x[n-1]-0.3y[n-1]4-0.2x[n]-0.2y[n]，

再延迟一个单位，加上***0.8x[n]***，最后得到：

y[n]=0.1x[n-2]-0.3y[n-2]+0.2x[n-1]-0.2y[n-1]+0.8x[n]

即得所示滤波器的差分方程。表示成更一般的形式为：

y[n]=-0.2y[n-1]-0.3y[n-2]+0.8x[n]+0.2x[n-1]+0.1x[n-2]

例4.10 的差分方程流图

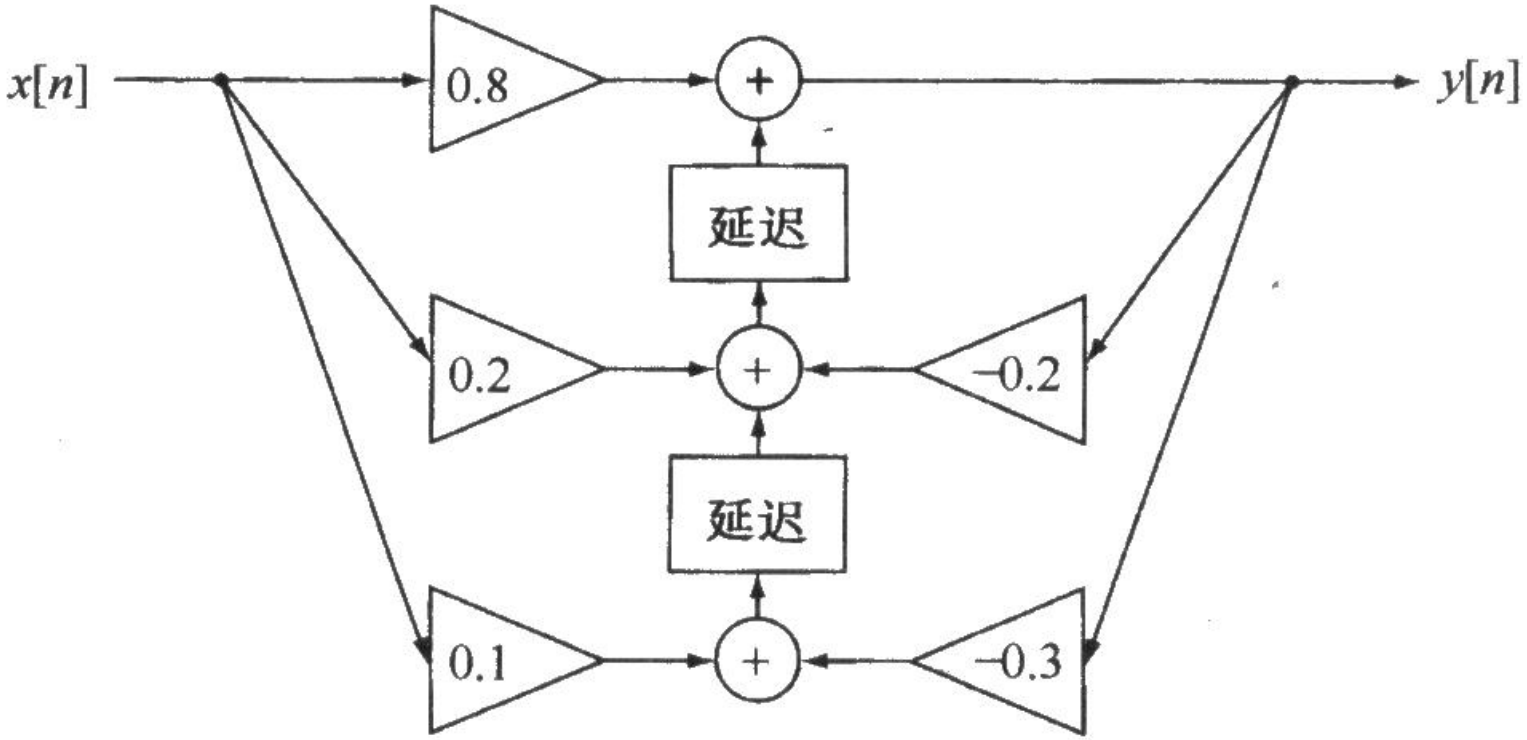


图 4.25　例 4.10 的差分方程流图

4.7 脉冲响应 (*impulse response*)

- 滤波器的脉冲响应(*impulse response*)就是滤波器对脉冲输入的响应；换句话讲，当滤波器的输入为单位脉冲时，滤波器的输出就是单位脉冲响应，如图4.26所示：

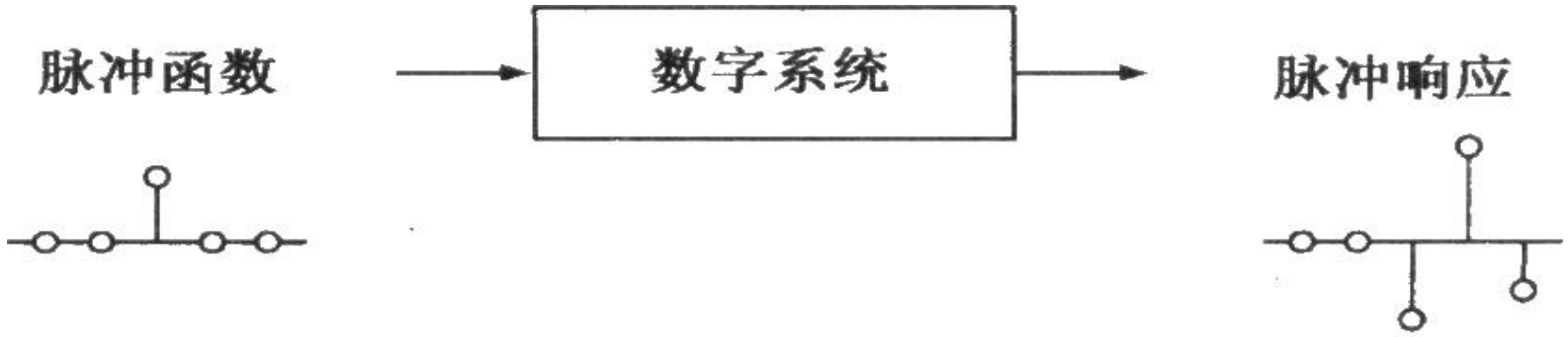


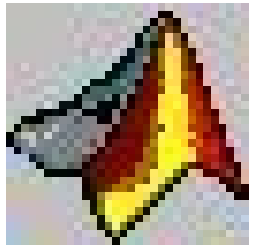
图 4.26　脉冲响应

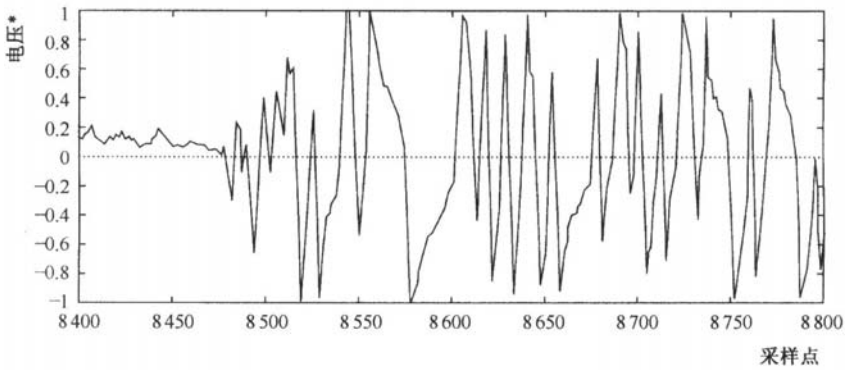
现实生活中的脉冲响应

- 实际的稳定的滤波器的脉冲响应逐渐消失到零，脉冲输入产生持续的输出。
- 应用的场合不同，对滤波器脉冲响应也有不同的要求：
  - 对于乐器：要求脉冲响应具有较长的持续时间以维持音律。
  - 对于汽车(等运输工具)：当汽车行驶时，颠簸对汽车悬架的作用，相当于脉冲的输入，设计时应使脉冲响应迅速的消失，可以使汽车平稳行驶。

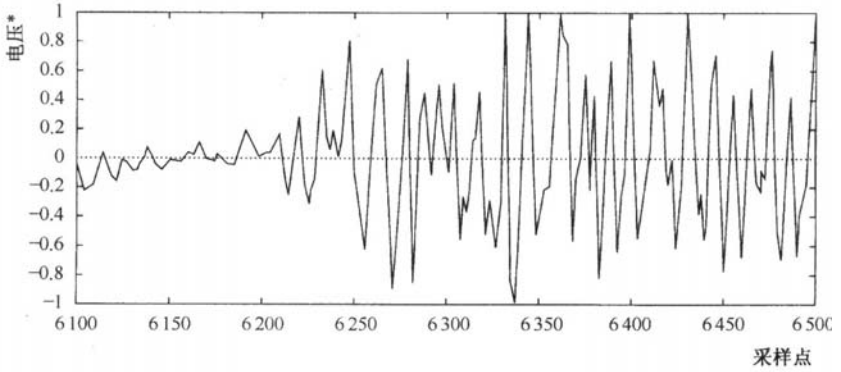
图4.27

- 22.05 kHz采样时钢琴、管风琴、羽管键琴的脉冲响应。

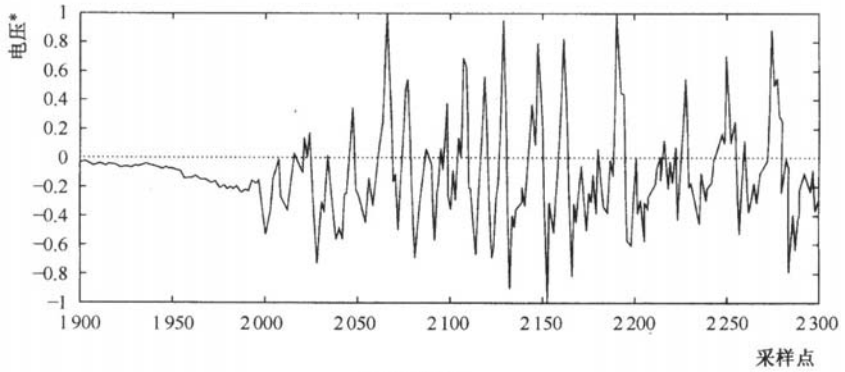




(a) 钢琴



(b) 管风琴



(c) 羽管键琴

由差分方程计算脉冲响应

- 由于输入必须是脉冲函数，则差分方程的输入$x[n]$用 $\delta[n]$代替，而滤波器的脉冲响应通常用$h[n]$表示。如此替代之后，从$n=0$开始逐点计算，一步步地计算$h[n]$。

- 脉冲响应反映了滤波器的基本特性。由于所有的数字信号可以由脉冲函数构成(见3.3.1节)，所以脉冲响应可用来求各种输入条件下的系统输出。

例4.11 对下列差分方程求出脉冲响应的前六个值。

$$y[n]-0.4y[n-1]=x[n]-x[n-1]$$

解： 首先，用 $\delta[n]$ 代替 $x[n]$，$h[n]$ 代替 $y[n]$，有：

$$h[n]-0.4h[n-1]=\delta[n]-\delta[n-1]$$

或

$$h[n]=0.4h[n-1]+\delta[n]-\delta[n-1]$$

从 $n=0$ 开始：

$$h[0]=0.4h[-1]+\delta[0]-\delta[-1]$$

脉冲函数的值已知：n=0时，它为1；在其他n不等于零处，它为零。假定滤波器是因果系统，即脉冲响应在n=0之前为零。因此：

$$h[0]=0.4\times(0.0)+1.0-0.0=1.0$$

注意 $\delta[-1]=0$ 是函数 $\delta[n]$ 当 $n=-1$ 时的值，而非因果性的结果。

随后的脉冲响应为：

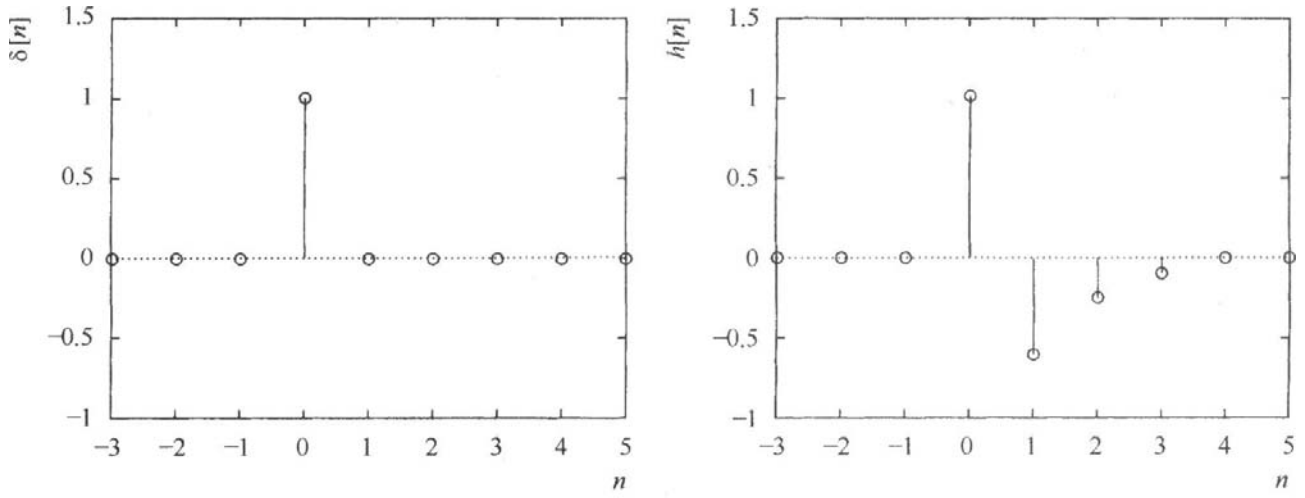
$$h[1]=0.4h[0]+\delta[1]-\delta[0]=0.4\times(1.0)+0.0-1.0=-0.6$$

$$h[2]=0.4h[1]+\delta[2]-\delta[1]=0.4\times(-0.6)+0.0-0.0=-0.24$$

$$h[3]=0.4h[2]+\delta[3]-\delta[2]=0.4\times(-0.24)+0.0-0.0=-0.096$$

$$h[4]=0.4h[3]+\delta[4]-\delta[3]=0.4\times(-0.096)+0.0-0.0=-0.0384$$

$$h[5]=0.4h[4]+\delta[5]-\delta[4]=0.4\times(-0.0384)+0.0-0.0=-0.01536$$



(a) 脉冲函数　　(b) 脉冲响应

图 4.28　例 4.11 的脉冲函数和脉冲响应

单位脉冲响应的Matlab计算

- 方法1：
  - 输入b, a；
  - 构造采样点数n=[-n1:n2];
  - 构造脉冲输入xn=[n==0];
  - 计算输出yn=filter(b,a, xn);
- 方法2：
  - 直接用[h,t]=impz(b,a,n,fs)计算：
  - fs为采样频率

无限脉冲响应
(*infinite impulse response，IIR*)

- 对于例4.11的递归滤波器，因为新的输出取决于过去的输出，所以脉冲响应不会消失，这个响应被称为**无限脉冲响应(*infinite impulse response，IIR*)**。
- 根据式(4.3)的一般递归差分方程，***N***个以前的输出和***M***个以前的输入对每个新的输出结果都有影响。

- **由于在计算每一个新的输出时要用到*N*个过去的输出，所以，即使输入已经为零，也有非零输出。**
- **无限脉冲响应*IIR*：递归滤波器的单位脉冲响应(永不消失)永不下降到零(但可能收敛于零)。**

有限脉冲响应
(finite impulse response，FIR)

- **有限脉冲响应*FIR*：脉冲响应在有限个采样点后下降到零，对应于非递归滤波器。**

  – 非递归滤波器在计算每一个新的输出时，要用到M个过去的输入，而不用过去的输出。脉冲响应变为零所需要的采样点数取决于计算中所用到的过去的输入个数。当输入是脉冲函数时，只有输入中的单个脉冲到达过去的M点之前，非递归滤波器的输出才能受到脉冲输入的影响，此后，滤波器的输出变为零。

例4.12 FIR实例

例4.12 求出下列滤波器脉冲响应的前六个采样值。

$$y[n]=0.25(x[n]+x[n-1]+x[n-2]+x[n-3]) \quad (4.8)$$

解：

用 $\delta[n]$ 代替 $x[n]$，$h[n]$ 代替 $y[n]$，有：

$$h[n]=0.25(\delta[n]+\delta[n-1]+\delta[n-2]+\delta[n-3]) \quad (4.9)$$

这样：

$$h[0]=0.25(\delta[0]+\delta[-1]+\delta[-2]+\delta[-3])=0.25\times(1.0+0.0+0.0+0.0)=0.25$$
$$h[1]=0.25(\delta[1]+\delta[0]+\delta[-1]+\delta[-2])=0.25\times(0.0+1.0+0.0+0.0)=0.25$$
$$h[2]=0.25(\delta[2]+\delta[1]+\delta[0]+\delta[-1])=0.25\times(0.0+0.0+1.0+0.0)=0.25$$
$$h[3]=0.25(\delta[3]+\delta[2]+\delta[1]+\delta[0])=0.25\times(0.0+0.0+0.0+1.0)=0.25$$
$$h[4]=0.25(\delta[4]+\delta[3]+\delta[2]+\delta[1])=0.25\times(0.0+0.0+0.0+0.0)=0.0$$
$$h[5]=0.25(\delta[5]+\delta[4]+\delta[3]+\delta[2])=0.25\times(0.0+0.0+0.0+0.0)=0.0$$

FIR与滤波器的系数

- **对于非递归滤波器，脉冲响应的采样值给出了差分方程的系数**。具有M个非零采样值的脉冲响应$h[n]$，根据

$$h[n] = h[0]\delta[n] + h[1]\delta[n-1] + \cdots + h[M]\delta[n-M] \tag{4.10}$$

可表示成脉冲函数之和，另一方面，非递归差分方程

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M] \tag{4.11}$$

对脉冲响应有：

$$h[n] = b_0\delta[n] + b_1\delta[n-1] + \cdots + b_M\delta[n-M] \tag{4.12}$$

由于式(4.10)与式(4.12)相等，所以对于所有的 k 值，有 $b_k = h[k]$。借助这个结论，将式(4.11)改写为：

$$y[n] = h[0]x[n] + h[1]x[n-1] + \cdots + h[M]x(n-M) \tag{4.13}$$

例4.13

- 例4.13 写出图4.30所示脉冲响应的滤波器差分方程。
- 解：
- 脉冲响应可以写成脉冲函数之和：

$$h[n]=\delta[n]+0.8\delta[n-1]+0.2\delta[n-2]$$

- 这样，差分方程具有类似的结构：

$$y[n]=x[n]+0.8x[n-1]+0.2x[n-2]$$

- 如果已知线性滤波器的脉冲响应，那么，对于任何输入都可以求出其输出。
- 由于每个数字输入都可以表示成脉冲函数之和，所以输出将是脉冲响应之和。

- 例4.14详述了这种联系，此联系使得脉冲响应成为分析滤波器特性的强有力工具。事实上，由于系统的所有输出都是由系统脉冲响应的副本构成的，所以可用脉冲响应对系统进行完全说明。

例4.14

- 例**4.14** 图4.31所示信号$x[n]$加到线性滤波器的输入端，滤波器的脉冲响应$h[n]$如图4.32 所示。将输入信号分成脉冲函数并求每一个的响应，然后求滤波器的输出$y[n]$。

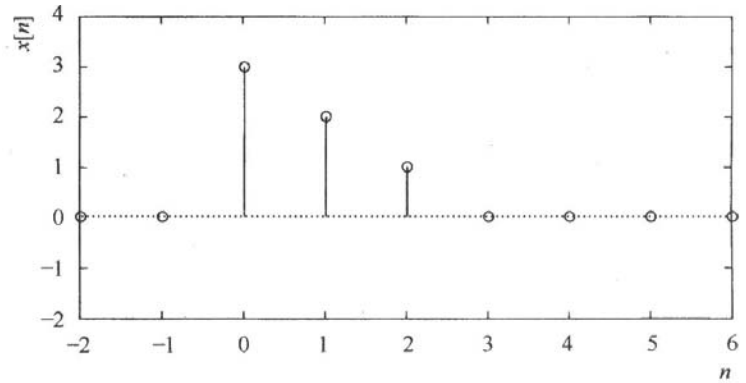


图 4.31　例 4.14 的输入信号

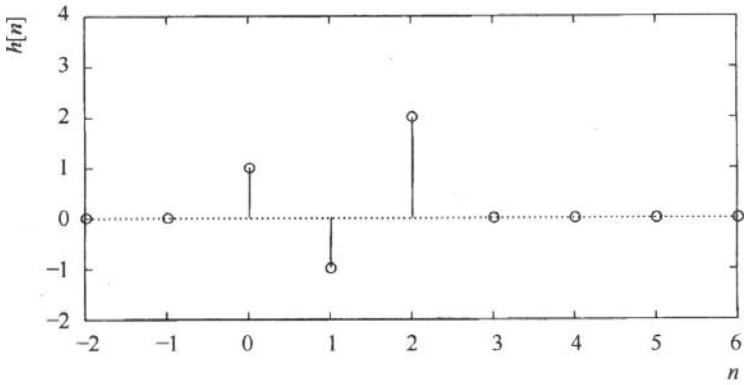
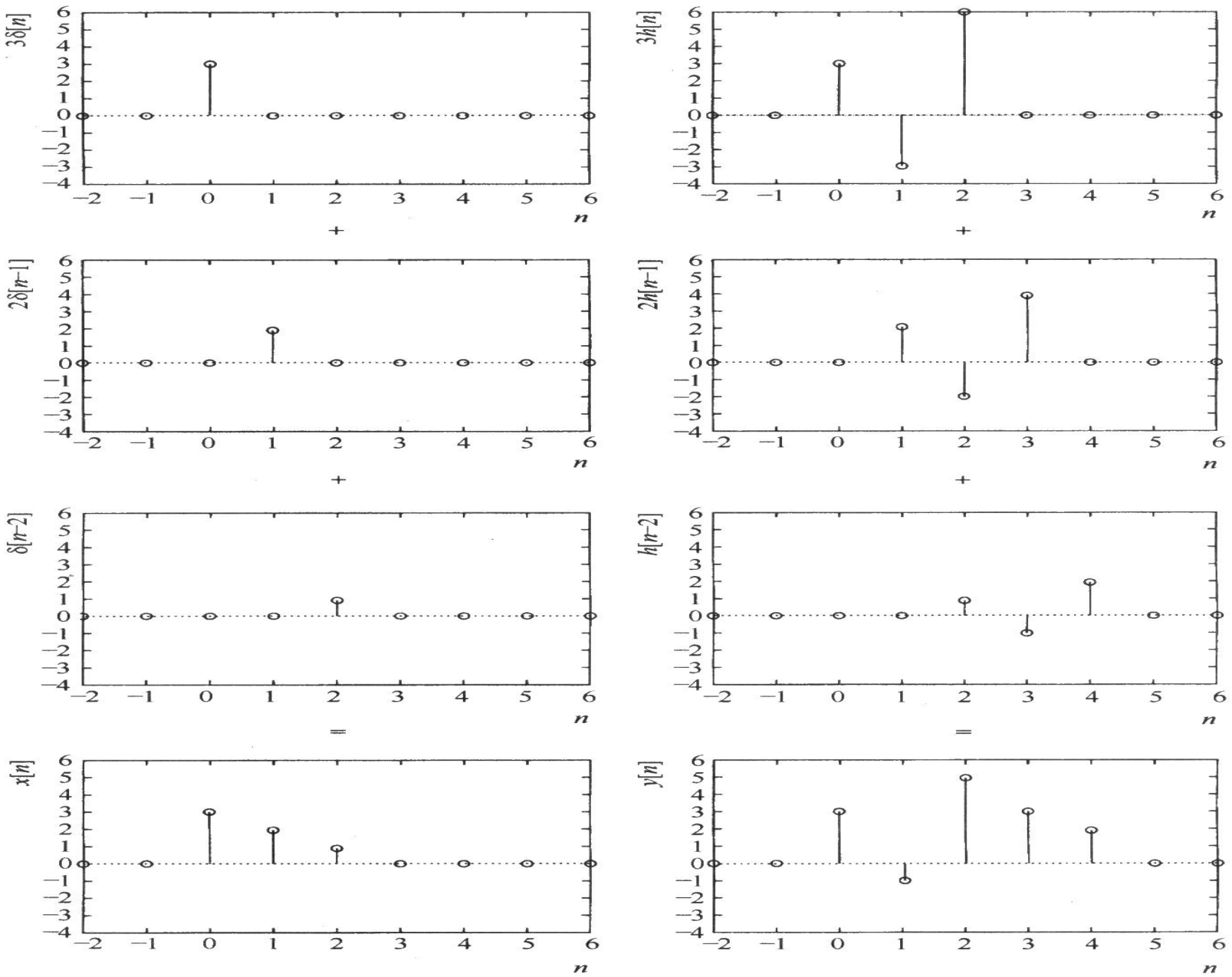


图 4.32　例 4.14 的脉冲响应



4.8 阶跃响应(*step response*)

- 阶跃响应(step response)是滤波器对单位阶跃函数的响应，它给出了系统对输入端电平变化的响应。

- 阶跃函数表示为u[n]，**阶跃响应表示为s[n]**。滤波器阶跃响应的求取有两种方法：
  - 一种方法是直接用差分方程，将u[n]作为输入，这种方法见例4.15。

- 由于已知阶跃函数是移位脉冲函数的无限和，另一种方法则是求脉冲响应的累积和。加到滤波器上的每一个脉冲函数，都将产生脉冲响应。对于线性滤波器，脉冲函数的和作为输入加到滤波器上，产生的输出是脉冲响应的和，见例4.14。
- 例4.16用图的形式说明了累积和的方法，所得结果与例4.15相同。已知滤波器的脉冲响应，通过求累积和来计算阶跃响应最为方便。

阶跃响应和脉冲响应的关系

$$\begin{cases} h[n] = S[n] - S[n-1] \\ S[n] = \sum_{k=0}^{n} h[k] \end{cases}$$

例4.15 求滤波器的阶跃响应：

y[n]-0.2y[n-2]=0.5x[n]+0.3x[n-1]

解：对于图4.34(a)所示的阶跃函数输入x[n]=u[n]，输出y[n]=s[n]如下：

s[n]=0.2s[n-2]+0.5u[n]+0.3u[n-1]， 这样

s[0]=0.2s[0-2]+0.5u[0]+0.3u[0-1]=0.2×(0)+o.5×(1)+o.3×(0)=O.5

s[1]= 0.2s[1-2]+0.5u[1]+0.3u[1-1]=o.2×(0)+o.5×(1)+0.3×(1)=0.8

s[2]=O.2s[0]+0.5u[2]+0.3u[1]=0.2×(0.5)+o.5×(1)+0.3×(1)=0.9

s[3]=O.2s[1]+O.5u[3]+O.3u[2]=O.2×(0.8)+0.5×(1)+0.3×(1)=0.96

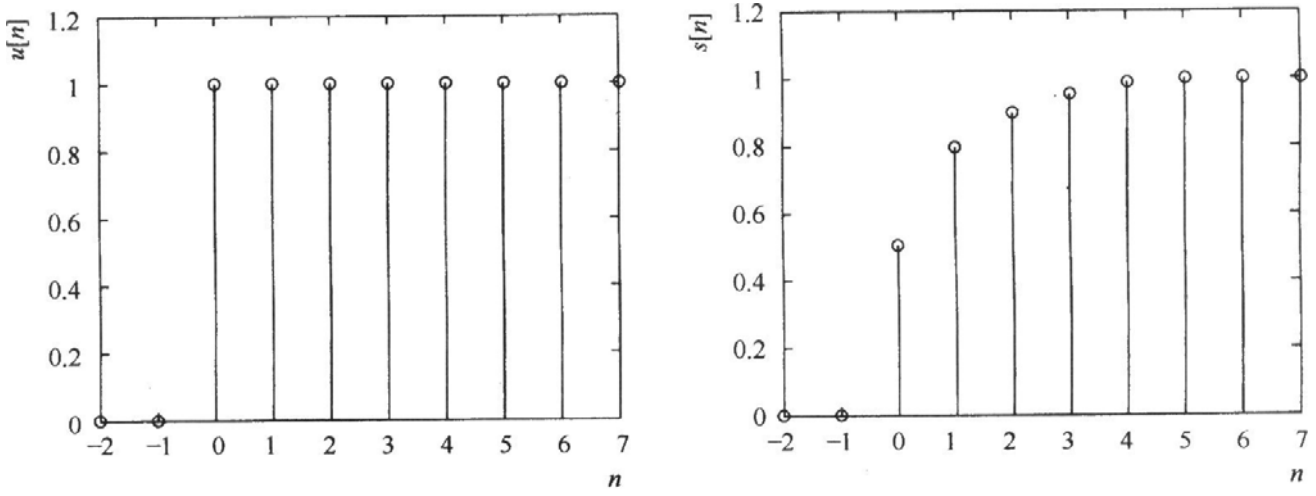
s[4]=0.2s[2]+0.5u[4]+O.3u[3]=0.2×(0.9)+0.5×(1)+O.3×(1)=0.98

s[5]=o.2s[3]+o.5u[5]+0.3u[4]=0.2×(0.96)+o.5×(1)+o.3 x(1)=0.992

s[6]=O.2s[4]+o.5u[6]+0.3u[5]=0.2×(0.98)+o.5×(1)+o.3×(1)=o.996

s[7]=0.2s[5]+0.5u[7]+0.3u[6]=0.2×(0.992)+0.5×(1)+0.3×(1)=o.998 4

滤波器的阶跃响应示于图4.34(b)。



(a) 阶跃函数　　(b) 阶跃响应

单位阶跃响应的Matlab计算

- 方法1：
  - 输入b, a；
  - 构造采样点数n=[-n1:n2];
  - 构造阶跃输入xn=[n>=0];
  - 计算输出sn=filter(b,a, xn);
- 方法2：
  - 直接用[sn,t] = stepz(b,a,n,fs)计算：
  - fs为采样频率

例4.16 滤波器：

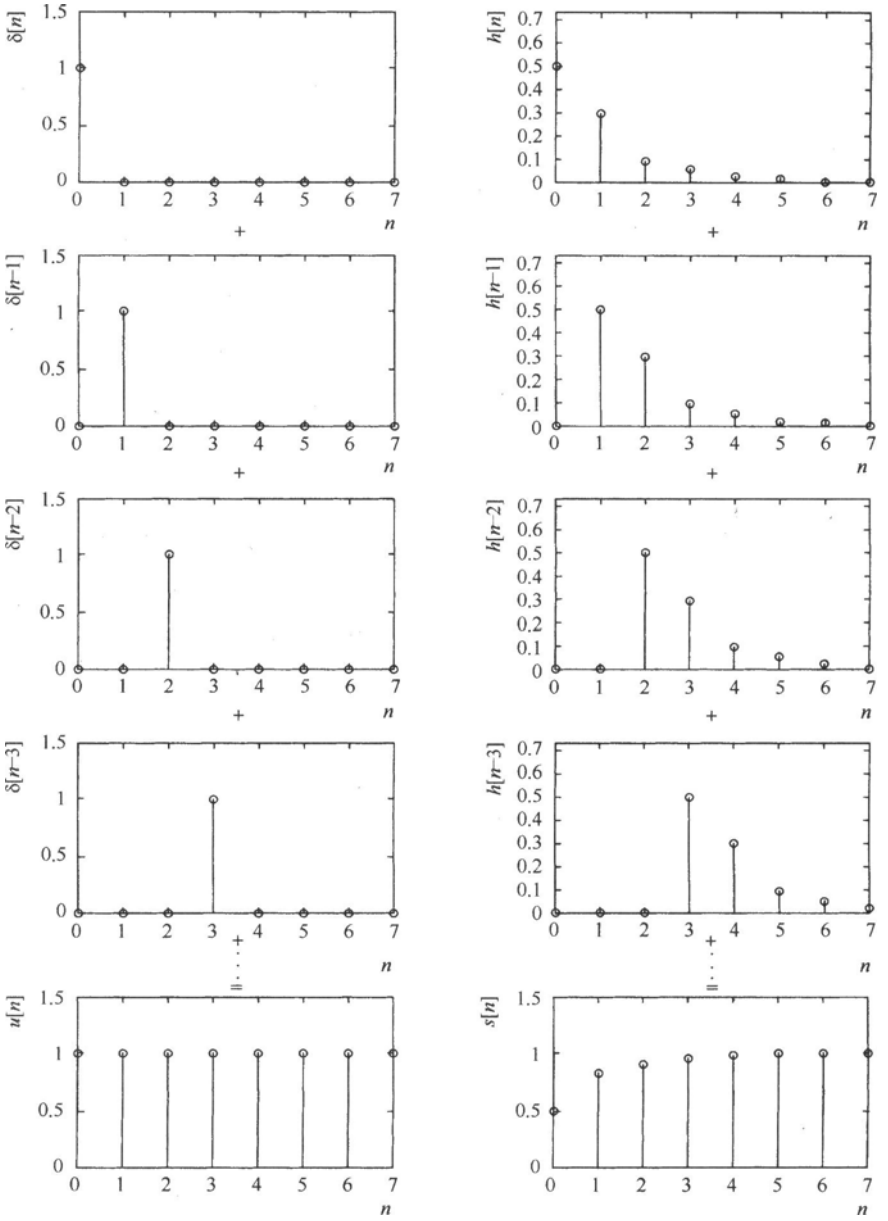
$y[n]-0.2y[n-2]=0.5x[n]+0.3x[n-1]$

的脉冲响应也可以通过：

$h[n]-0.2h[n-2]=0.5\delta[n]+0.3\delta[n-1]$

从例4.15得到。h[n]的值位于表4.3的第二列，通过求累积和，可以从脉冲响应得到阶跃响应，如图4.35所示。即每个阶跃响应的值是所有到n的脉冲响应的和。

例如，s[2]为 $h[0]+h[1]+h[2]=0.5+0.3+0.1=0.9$。



本章小结1

1. 滤波器从信号中滤除频率分量。它在特定频率上的增益决定了对此频率输入信号的放大量。在滤波器增益较高(大于最大值的1 √ 2)的频率上的信号可以通过，在滤波器增益较低的频率上的信号受到衰减。如果滤波器的最大增益为1，滤波器的截止频率用-3 dB增益标定，则也可以由界定滤波器带宽的频点标定。
2. 低通滤波器通过低频信号，高通滤波器通过高频信号，带通滤波器通过中频频带信号，而带阻滤波器阻碍中频频带信号的通过。

本章小结2

3. 数字滤波器较模拟滤波器有许多优点：抗噪声、易重新设计。另外，与低阶滤波器相比，实现高阶滤波器并不困难，而模拟滤波器并非如此。
4. 滤波器是系统的一个实例，实际滤波器是线性、时不变及因果的。
5. 差分方程用来计算滤波器的输出，非递归滤波器有：

本章小结3

$$y[n]=b_0x[n]+b_1x[n-1]+\ldots+b_Mx[n-M]$$

它在计算滤波器的输出y[n]时仅依赖输入。递归滤波器有：

$$y[n]=-a_1y[n-1]-a_2y[n-2]-\ldots-a_Ny[n-N]$$
$$+b_0x[n]+b_1x[n-1]+\ldots+b_Mx[n-M]$$

它在计算滤波器的输出y[n]时依赖输入及过去的输出。

6. 对于线性系统，运用叠加原理，多个信号可同时加到输入端。

本章小结4

6. 差分方程的实现经常表示为差分方程流图。能减小存储需求和有限字长效应的实现是最好的，可以通过低阶滤波器的级联或并联的组合实现高阶滤波器。
7. 脉冲响应是滤波器对脉冲函数输入的响应，对于任意输入，滤波器的输出可表示为移位脉冲响应之和。
8. 阶跃响应是滤波器对阶跃函数输入的响应。

作业

9，15，16，20，25，32