

高等学校教材·电子信息

数字信号处理及 MATLAB 实现

余成波 杨 菁 杨如民 周登义 编著

清华大学出版社
北 京

内 容 提 要

本书系统地介绍了数字信号处理的基本理论、算法及 MATLAB 的实现。全书共分 7 章,内容包括离散时间信号与系统、Z 变换、离散傅里叶变换、快速傅里叶变换、数字滤波器的基本结构、无限冲击响应滤波器(IIR)的设计、有限冲激响应滤波器(FIR)的设计。在每章后面分别结合数字信号处理的典型例题说明应用 MATLAB 进行数字信号处理的编程方法与技巧。

本书可作为高等理工院校电子信息、通信工程、信号与信息处理专业的本科生教材,也可供研究生以及有关科技人员学习参考。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

数字信号处理及 MATLAB 实现 / 余成波等编著. —北京:清华大学出版社, 2005.1

(高等学校教材·电子信息)

ISBN 7-302-10156-6

I. 数... II. 余... III. 数字信号—信号处理—计算机辅助计算—软件包, MATLAB—高等学校—教材
IV. TN911.72

中国版本图书馆 CIP 数据核字(2004)第 135500 号

出 版 者:清华大学出版社

地 址:北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

客 户 服 务:010-62776969

责任编辑:魏江江

封面设计:杨 兮

印 刷 者:

装 订 者:

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:12.25 字数:295 千字

版 次:2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

书 号:ISBN 7-302-10156-6/TN·226

印 数:1~4000

定 价:19.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010) 62770175-3103 或 (010) 62795704

前 言

随着计算机技术和信息科学的迅速发展，数字信号处理的理论、算法及实现手段也获得了迅速的发展，已经逐渐发展成为一门独立的学科并成为信息科学的重要组成部分，在语音处理、图像处理、雷达、航空航天、地质勘探、通信、生物医学工程等众多领域得到了广泛的应用。

数字信号处理的内容广泛，理论复杂，而且由于概念抽象，课时数较少，学生在学习这部分内容时甚感困难。为此，本书编写过程中，以素质培养和能力培养为目标，以加强基础、拓宽专业知识面为原则，同时注重理论和应用相结合，特别注意应用 MATLAB 来解决算法的实现问题，以便帮助读者更直观、形象地理解理论。

全书共分 7 章，包括离散时间信号与系统、Z 变换、离散傅里叶变换、快速傅里叶变换、数字滤波器的基本结构、无限冲击响应滤波器(IIR)的设计、有限冲激响应滤波器(FIR)的设计。建议课堂教学 54 学时。由于课时的原因，本书略去了量化效应与有限长效应、功率谱估计以及小波分析在信号处理中的应用等内容，不同的读者可根据各自的实际要求进行补充。

本书由余成波统稿。其中，第 1、2、3 章由杨菁、徐霞、胡顺仁等编写，第 4 章由余成波、陶红艳等编写，第 5 章由周登义、包明等编写，第 6、7 章由杨如民、余成波等编写。本书在编写过程中得到许多兄弟院校同行们的大力支持和帮助，提出了许多宝贵意见，在此，一并表示衷心的感谢。

全书所给出 MATLAB 实现的实例程序，均已通过作者的调试，读者可直接引用。

本书可作为高等理工院校电子信息、通信工程、信号与信息处理专业的本科生教材，也可供研究生以及有关科技人员学习参考。

限于作者的水平，不妥及错误之处在所难免，恳切希望读者给予批评指正。

编 者

目 录

第 1 章 离散时间信号与系统.....	1
1.1 离散时间信号——序列.....	1
1.2 线性移不变系统.....	7
1.3 常系数线性差分方程.....	11
1.4 模拟信号的数字处理.....	12
1.5 用 MATLAB 表示和实现离散时间信号与系统.....	17
第 2 章 Z 变换.....	27
2.1 Z 变换的定义及收敛域.....	27
2.2 Z 变换的性质.....	31
2.3 Z 反变换.....	35
2.4 Z 变换与连续信号的拉普拉斯变换、傅里叶变换的关系.....	38
2.5 序列的傅里叶变换及其特性.....	40
2.6 离散系统的系统函数、系统的频率响应.....	43
2.7 MATLAB 在 Z 变换、傅里叶变换中的应用.....	45
第 3 章 离散傅里叶变换.....	50
3.1 傅里叶变换的几种形式及应用.....	50
3.2 周期序列的离散傅里叶级数.....	53
3.3 离散傅里叶变换.....	57
3.4 离散傅里叶变换的性质.....	58
3.5 离散傅里叶变换的应用.....	67
3.6 MATLAB 在 DFT 中的应用.....	72
第 4 章 快速傅里叶变换.....	83
4.1 直接计算 DFT 的问题及改进的途径.....	83
4.2 时间抽取基-2 FFT 算法.....	84
4.3 频率抽取基-2 FFT 算法.....	91
4.4 离散傅里叶反变换的快速算法.....	95
4.5 N 为复合数的 FFT 算法——混合基算法.....	96
4.6 线性调频 Z 变换.....	98
4.7 有关的 MATLAB 实现.....	102

第 5 章 数字滤波器的基本结构.....	106
5.1 数字滤波器的结构特点与表示方法.....	106
5.2 无限长单位冲激响应滤波器的基本结构.....	108
5.3 有限长单位冲激响应滤波器的结构.....	111
5.4 数字滤波器的 MATLAB 实现.....	114
第 6 章 无限冲激响应滤波器的设计.....	128
6.1 通过模拟滤波器设计 IIR 数字滤波器.....	128
6.2 冲激响应不变法.....	133
6.3 双线性变换法.....	135
6.4 从原型低通滤波器到其他各型数字滤波器的频带变换法.....	137
6.5 应用 MATLAB 设计 IIR 数字滤波器.....	140
第 7 章 有限冲激响应滤波器的设计.....	165
7.1 线性相位 FIR 滤波器的特点.....	165
7.2 窗函数设计法.....	170
7.3 频率抽样设计法.....	174
7.4 应用 MATLAB 设计 FIR 数字滤波器.....	177
参考书目.....	184

第 1 章 离散时间信号与系统

1.1 离散时间信号——序列

信号可定义为一个承载信息的函数。根据信号的特点，可表示成一个或几个独立变量的函数，例如图像信号就是空间位置（二维变量）的色彩和亮度函数，一维信号的变量可以是时间，也可以是频率、空间或其他物理量，习惯上看成是时间。根据信号的幅值和时间取连续值还是取离散值，可以分为下面几种信号。

连续时间信号：在连续时间范围内定义的信号，信号幅值可以是连续的也可以是离散的。在幅值连续的特定情况下又常称为模拟信号，实际上连续时间信号与模拟信号常常通用，用来说明同一信号。

离散时间信号：时间为离散变量，而幅值是连续变化的信号。

数字信号：时间和幅值都是离散化的信号。

1.1.1 序列的定义

离散时间信号可由连续时间信号 $x(t)$ 通过抽样获得，设抽样时间间隔为 T ，故用 $x(nT)$ 表示此离散时间信号在 nT 点上的值， n 为整数。由于离散时间信号处理常常是非实时的，可以先记录数据后分析或存放在存储器中以供随时取用，因此 $x(nT)$ 可以看做是按照一定顺序排列的一组数据，可以直接用 $x(n)$ 表示第 n 个离散时间点的序列值，并用 $\{x(n)\}$ 表示离散时间信号——序列，为方便起见，通常情况下直接用 $x(n)$ 表示离散序列。

离散时间信号——序列可以用图形来描述，如图 1.1 所示，纵轴线段的长短代表各序列值的大小，横轴代表离散时间点。注意，横轴虽然为连续直线，但是 $x(n)$ 只有在 n 为整数时才有意义， n 不是整数时没有定义，但不能认为 $x(n)$ 的值为零。

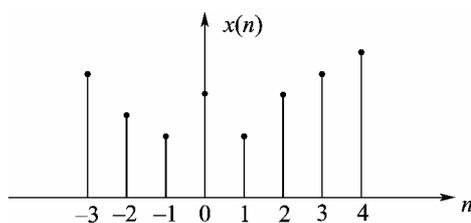


图 1.1 离散时间信号的图形表示

1.1.2 序列的基本运算

1. 序列的相加

序列 $x(n)$ 与序列 $y(n)$ 之和是指两个序列同序号的数值逐项对应相加而构成一个新的序列 $z(n)$ ，表示为 $z(n) = x(n) + y(n)$ 。

例 1.1 已知 $x(n) = \begin{cases} 3^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$, $y(n) = \begin{cases} \left(\frac{1}{2}\right)^n, & n \geq -1 \\ n+1, & n < -1 \end{cases}$, 求 $x(n) + y(n)$ 。

解：

$$z(n) = x(n) + y(n) = \begin{cases} n+1, & n < -1 \\ 2, & n = -1 \\ 3^n + \left(\frac{1}{2}\right)^n, & n > -1 \end{cases}$$

2. 序列的相乘

序列 $x(n)$ 与序列 $y(n)$ 相乘是指两个序列同序号的数值逐项对应相乘而构成的一个新序列 $z(n)$, 表示为 $z(n) = x(n)y(n)$ 。

3. 序列的移位

若序列为 $x(n)$, 则 $x(n-m)$ 是指原序列 $x(n)$ 逐项依次延时 (右移) m 位而构成的一个新序列, 而 $x(n+m)$ 是指原序列 $x(n)$ 逐项依次超前 (左移) m 位, 如图 1.2 所示。

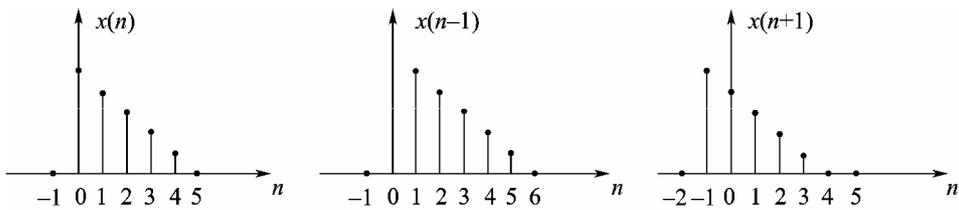


图 1.2 序列的移位

4. 序列的翻褶

序列的翻褶又称为转置或反褶, 若序列为 $x(n)$, 则 $x(-n)$ 就是以 $n=0$ 为对称轴将序列 $x(n)$ 加以翻褶, 如图 1.3 所示。

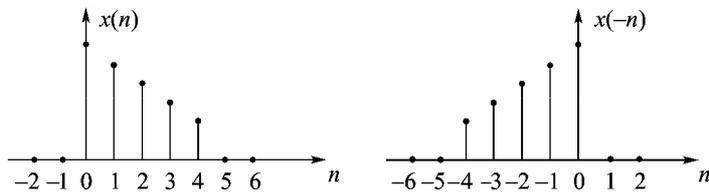


图 1.3 序列的翻褶

5. 序列的累加

若序列为 $x(n)$, 则 $x(n)$ 的累加序列 $y(n)$ 定义为

$$y(n) = \sum_{k=-\infty}^n x(k) \quad (1.1)$$

它表示 $y(n)$ 在某个 n_0 点的值等于这个 n_0 点上的 $x(n_0)$ 以及以前所有 n 值上的 $x(n)$ 值之和。

6. 序列的卷积和

卷积积分是求连续线性时不变系统输出响应的主要方法，同样求离散线性移不变系统输出响应的主要方法是卷积和。

设两个序列为 $x(n)$ 和 $h(n)$ ， $x(n)$ 和 $h(n)$ 的卷积和定义为

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) = x(n) * h(n) \quad (1.2)$$

其中，用“*”代表卷积和运算。卷积和的运算在图形上可以分成四步：翻褶、移位、相乘、相加，以 $x(n) = \begin{cases} 3-n, & 0 \leq n \leq 2 \\ 0, & \text{其他}n \end{cases}$ 和 $h(n) = \begin{cases} 1, & 0 \leq n \leq 3 \\ 0, & \text{其他}n \end{cases}$ 为例，图解如图 1.4 所示，卷积和的图解法计算步骤如下。

(1) 翻褶：先将 $x(n)$ 和 $h(n)$ 的变量置换为 m ，得到 $x(m)$ 和 $h(m)$ ，将 $h(m)$ 以 $m=0$ 的垂直轴为对称轴翻褶为 $h(-m)$ 。

(2) 移位：将 $h(-m)$ 沿 m 轴平移 n 得到 $h(n-m)$ ，当 $n>0$ 时，右移 n 位，当 $n<0$ 时，左移 $|n|$ 位。

(3) 相乘：对给定的某个 n 值，将 $h(n-m)$ 和 $x(m)$ 相同 m 值的对应点相乘。

(4) 相加：再将以上所有对应点的乘积累加，就可以得到给定的某 n 值时的 $y(n)$ 。

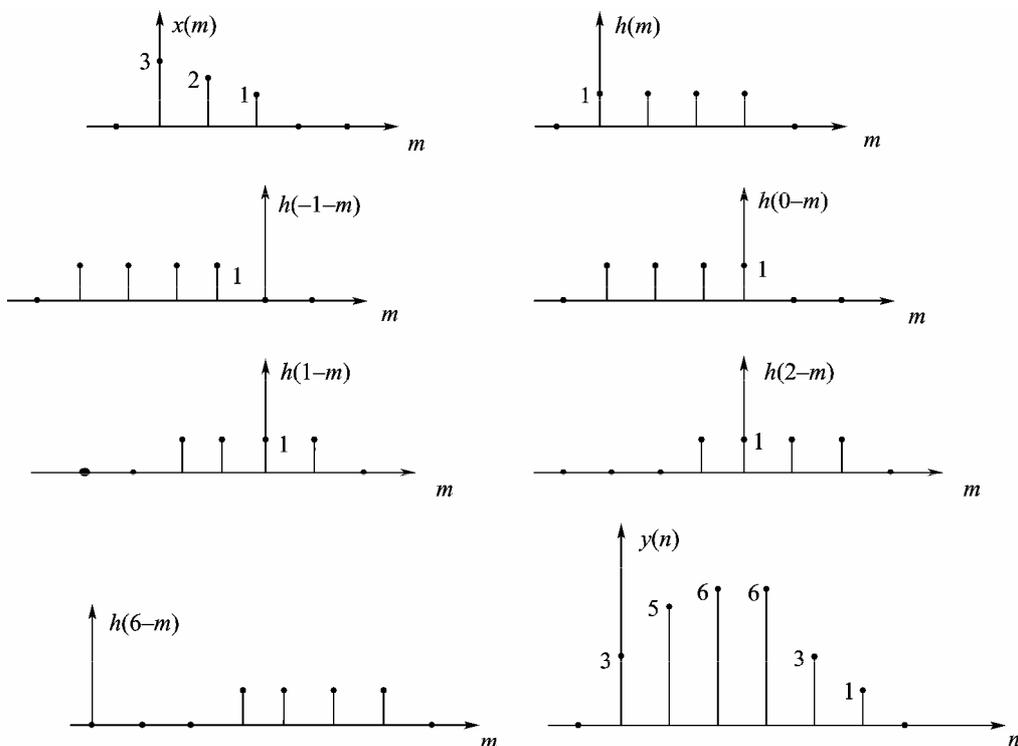


图 1.4 $x(n)$ 和 $h(n)$ 的卷积和图解

由于信号处理中，通常遇到的都是有始信号或有限时间信号，因此在实际计算卷积和时，式(1.2)中的求和是在有限范围内进行的，在计算过程中的上下限的选取和所得结果的分布区间取决于参与卷积的两个序列的分布情况，下面就解析表达式的情况进行讨论。

(1) 两个从 $n=0$ 开始的序列 $x(n)u(n) = \begin{cases} x(n), & n \geq 0 \\ 0, & n < 0 \end{cases}$ 和 $h(n)u(n) = \begin{cases} h(n), & n \geq 0 \\ 0, & n < 0 \end{cases}$ 的卷积和为

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)u(m)h(n-m)u(n-m) = \left[\sum_{m=0}^n x(m)h(n-m) \right] u(n)$$

其中因子 $u(m)$ 、 $u(n-m)$ 和 $u(n)$ 是后面将要介绍的常用典型序列中的单位阶跃序列，而计算结果显示，两个从 $n=0$ 开始的序列的卷积和也是一个从 $n=0$ 开始的序列。

(2) 从 $n = n_1$ 开始的序列 $x(n)u(n-n_1) = \begin{cases} x(n), & n \geq n_1 \\ 0, & n < n_1 \end{cases}$ 和从 $n = n_2$ 开始的序列

$h(n)u(n-n_2) = \begin{cases} h(n), & n \geq n_2 \\ 0, & n < n_2 \end{cases}$ 的卷积和为

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)u(m-n_1)h(n-m)u(n-m-n_2) = \left[\sum_{m=n_1}^{n-n_2} x(m)h(n-m) \right] u(n-n_2-n_1)$$

其中 n_1 和 n_2 可以是任意整数，而计算结果显示卷积和是一个从 $n = n_1 + n_2$ 开始的序列。

(3) 从 $n = 0$ 开始的序列 $x(n)u(n) = \begin{cases} x(n), & n \geq 0 \\ 0, & n < 0 \end{cases}$ 和从 $n = n_1$ 开始的有限长时间序列

$h(n) = \begin{cases} h(n), & n_1 \leq n \leq n_1 + N - 1 \\ 0, & n \text{ 为其他值} \end{cases}$ 的卷积和为

$$\begin{aligned} y(n) &= \sum_{m=-\infty}^{\infty} x(m)u(m)h(n-m)[u(n-m-n_1) - u(n-m-n_1-N+1)] \\ &= \left[\sum_{m=0}^{n-n_1} x(m)h(n-m) \right] u(n-n_1) - \left[\sum_{m=0}^{n-n_1-N+1} x(m)h(n-m) \right] u(n-n_1-N+1) \end{aligned}$$

(4) 从 $n=n_1$ 开始的长度为 N_1 的有限长时间序列 $x(n) = \begin{cases} x(n), & n_1 \leq n \leq n_1 + N_1 - 1 \\ 0, & n \text{ 为其他值} \end{cases}$ 和从

$n=n_2$ 开始的长度为 N_2 的有限长时间序列 $h(n) = \begin{cases} h(n), & n_2 \leq n \leq n_2 + N_2 - 1 \\ 0, & n \text{ 为其他值} \end{cases}$ 的卷积和也是一

个有限长序列，从 $n=n_1+n_2$ 开始，长度为 N_1+N_2-1 。

线性卷积运算服从交换律，结合律和分配律，分别用公式表示如下：

交换律

$$x(n) * h(n) = h(n) * x(n)$$

结合律

$$x(n) * [h_1(n) * h_2(n)] = [x(n) * h_1(n)] * h_2(n) = [x(n) * h_2(n)] * h_1(n)$$

分配律

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$$

7. 差分运算

前向差分

$$\Delta x(n) = x(n+1) - x(n)$$

后向差分

$$\nabla x(n) = x(n) - x(n-1)$$

由此得出

$$\nabla x(n) = \Delta x(n-1)$$

8. 序列的比例 (时间尺度) 变换

对序列 $x(n)$, 其比例变换序列为 $x(mn)$ 或 $x(n/m)$, 其中 m 为正整数。

例如, $x(4n)$ 是以低 4 倍的频率从 $x(n)$ 中每隔 4 个值取 1 个值, 若 $x(n)$ 是连续时间信号 $x(t)$ 的抽样, 那么 $x(4n)$ 相当于将 $x(n)$ 的抽样间隔从 T 增加到 $4T$ 。这种运算称为抽取, 将 $x(4n)$ 称为 $x(n)$ 的抽取序列。

同样的道理, $x(n/4)$ 表示将 $x(n)$ 的抽样间隔从 T 减少到 $T/4$, 将 $x(n/4)$ 称为 $x(n)$ 的插值序列。

1.1.3 常用典型序列

1. 单位抽样序列 $\delta(n)$

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (1.3)$$

$\delta(n)$ 在离散序列处理中的作用类似于连续时间信号处理中的冲激函数 $\delta(t)$, 但是 $\delta(t)$ 是 $t=0$ 时脉宽趋于 0, 幅值趋于无限大, 面积为 1 的信号, 是极限概念的信号, 并不是一个现实的信号, 而 $\delta(n)$ 在 $n=0$ 时取值为 1, 既简单又易计算。单位抽样序列如图 1.5 (a) 所示。因为

$$\delta(n-m) = \begin{cases} 1, & n = m \\ 0, & n \neq m \end{cases}$$

所以

$$x(n)\delta(n-m) = \begin{cases} x(m), & n = m \\ 0, & n \neq m \end{cases}$$

由此可以得到序列的另一种表达形式, 即任何序列都可以表示为单位抽样序列的加权移位和, 即

$$x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n-m) \quad (1.4)$$

2. 单位阶跃序列 $u(n)$

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (1.5)$$

如图 1.5 (b) 所示, 它类似于连续时间信号中的 $u(t)$ 。

$\delta(n)$ 和 $u(n)$ 的关系为

$$\delta(n) = u(n) - u(n-1) \quad (1.6)$$

$$u(n) = \sum_{m=0}^{\infty} \delta(n-m) = \delta(n) + \delta(n-1) + \delta(n-2) + \dots \quad (1.7)$$

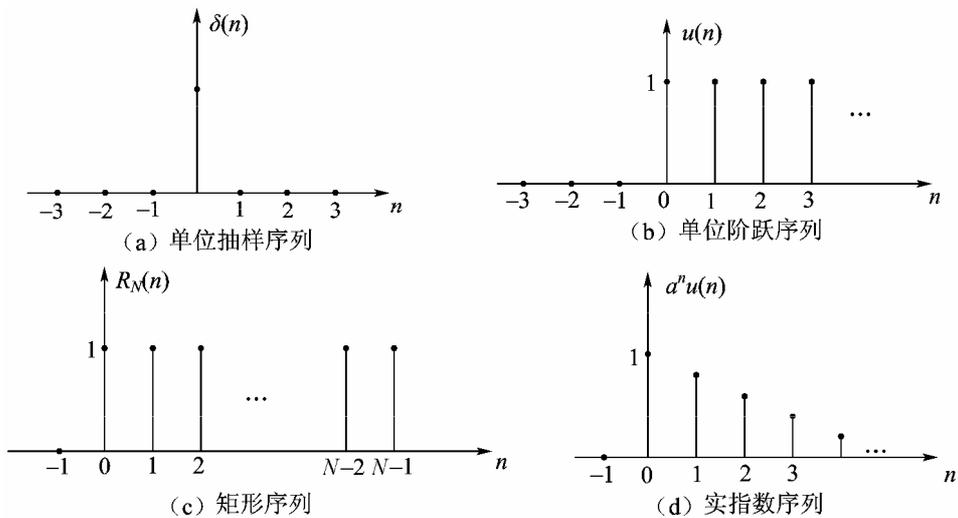


图 1.5 几种常用序列

3. 矩形序列 $R_N(n)$

$$R_N(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & n \text{ 为其他值} \end{cases} \quad (1.8)$$

如图 1.5 (c) 所示, 它类似于连续时间信号中的矩形脉冲。

$R_N(n)$ 与 $\delta(n)$ 、 $u(n)$ 的关系为

$$R_N(n) = u(n) - u(n-N) \quad (1.9)$$

$$R_N(n) = \sum_{m=0}^{N-1} \delta(n-m) \quad (1.10)$$

4. 实指数序列

$$x(n) = a^n u(n) \quad (1.11)$$

其中 a 为实数, 当 $|a| < 1$ 时, 序列是收敛的, 而当 $|a| > 1$ 时, 序列是发散的, 当 $0 < a < 1$ 时, 如图 1.5 (d) 所示。

5. 复指数序列

$$x(n) = e^{n(\sigma + j\omega_0)} u(n) \quad (1.12)$$

也可以用其实部和虚部表示为

$$x(n) = e^{n\sigma} (\cos \omega_0 n + j \sin \omega_0 n) = e^{n\sigma} \cos \omega_0 n + j e^{n\sigma} \sin \omega_0 n$$

或用极坐标表示为

$$x(n) = |x(n)| e^{j \arg[x(n)]} = e^{\sigma n} e^{j \omega_0 n}$$

其中 $|x(n)| = e^{\sigma n}$, $\arg[x(n)] = \omega_0 n$ 。

6. 正弦型序列

$$x(n) = A \sin(n\omega_0 + \phi) \quad (1.13)$$

其中 A 为幅度, ω_0 为数字频率, ϕ 为起始相位。

1.2 线性移不变系统

将输入序列 $x(n)$ 映射成输出序列 $y(n)$ 的惟一性变换或运算定义为时域离散系统, 记为

$$y(n) = T[x(n)] \quad (1.14)$$

式中, $T[\cdot]$ 用来表示这种变换关系。如果变换关系 $T[\cdot]$ 加上各种约束条件就定义了各类时域离散系统。本书研究的是“线性移不变”的离散时间系统。

1.2.1 线性系统

凡是满足均匀性和叠加性的系统称为线性系统, 也就是说, 若 $y_1(n)$ 和 $y_2(n)$ 分别为输入 $x_1(n)$ 和 $x_2(n)$ 的输出响应, 即

$$y_1(n) = T[x_1(n)], \quad y_2(n) = T[x_2(n)]$$

那么当且仅当

$$y(n) = T[ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)] = ay_1(n) + by_2(n) \quad (1.15)$$

时, 该系统称为线性系统, 其中 a, b 为任意常数。

对线性系统若写成 N 个输入的一般表达式, 则为

$$\sum_{i=1}^N a_i y_i(n) = T\left[\sum_{i=1}^N a_i x_i(n)\right] \quad (1.16)$$

1.2.2 移不变系统

如果系统的输出响应随着输入位移而位移, 那么该系统就称为移不变系统, 即若输入 $x(n)$ 产生输出为 $y(n)$, 则输入 $x(n-m)$ 产生输出为 $y(n-m)$, 也就是输入移动任意位, 其输出也移动相等位数, 且幅值保持不变。对移不变系统, 若

$$y(n) = T[x(n)]$$

则

$$y(n-m) = T[x(n-m)] \quad (1.17)$$

其中 m 为任意整数。

1.2.3 线性移不变系统的输入、输出关系

设线性移不变系统输出 $y(n)$ 的初始状态为零，当输入 $x(n) = \delta(n)$ 时，其输出定义为系统的单位抽样响应，用 $h(n)$ 表示，即

$$h(n) = T[\delta(n)] \quad (1.18)$$

$h(n)$ 类似于连续时间系统中的单位冲激响应 $h(t)$ ，完全表征了系统的时域特性。

设线性移不变系统的输入序列为 $x(n)$ ，输出序列为 $y(n)$ ，由式 1.4 可知

$$x(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n-m)$$

所以相应的系统输出为

$$y(n) = T[x(n)] = T\left[\sum_{m=-\infty}^{\infty} x(m)\delta(n-m)\right]$$

根据线性系统的叠加原理，有

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)T[\delta(n-m)]$$

又根据移不变特性，可得

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)h(n-m) = x(n) * h(n) \quad (1.19)$$

式中，符号“*”代表卷积和运算，因此一个线性移不变系统可以用图 1.6 表示，系统的输入输出关系满足卷积关系，而且系统可以用单位抽样响应 $h(n)$ 完全描述。

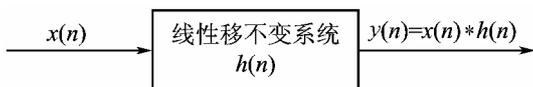


图 1.6 线性移不变系统框图

1.2.4 因果系统

因果系统是指其输出变化不会发生在输入变化之前的一种系统，也就是说，因果系统的 n 时刻的输出只取决于 n 时刻及 n 时刻以前的输入序列，而和 n 时刻以后的输入序列无关，因此系统的因果性是指系统的可实现性，如果现在的输出和未来的输入有关，这在时间上违背了因果性，而且系统也无法实时实现，这样的系统就称为非因果系统。

线性移不变系统具有因果性的充分必要条件是

$$h(n) = 0, \quad n < 0 \quad (1.20)$$

证明：

充分条件。若 $n < 0$ 时， $h(n) = 0$ ，根据卷积和公式有

$$y(n_0) = \sum_{m=-\infty}^{\infty} h(m)x(n_0 - m) = \sum_{m=0}^{\infty} h(m)x(n_0 - m)$$

因为式中 $m \geq 0$ ，所以 $n_0 - m \leq n_0$ ，这就证明了 $y(n_0)$ 的值只取决于 $x(n)$ 在 $n \leq n_0$ 时的值，因此系统是因果的。

必要条件。根据卷积和公式有

$$y(n_0) = \sum_{m=-\infty}^{\infty} h(m)x(n_0 - m) = \sum_{m=-\infty}^{-1} h(m)x(n_0 - m) + \sum_{m=0}^{\infty} h(m)x(n_0 - m)$$

若当 $m < 0$ 时， $h(m) \neq 0$ ，则上式第一项求和式中 $n_0 - m > n_0$ ，即系统在 n_0 时的输出 $y(n_0)$ 与输入 $x(n)$ 在 $n > n_0$ 时的值有关，也就是 $y(n_0)$ 值与 n_0 以后的 $x(n)$ 有关，所以该系统不是因果系统，可见要使 $y(n_0)$ 与 $n > n_0$ 时的 $x(n)$ 无关，则必须使

$$n < 0, \quad h(n) = 0$$

频率特性为矩形的理想低通滤波器以及理想微分器等都是非因果的不可实现系统，但是如果不是实时处理，或虽实时但是允许一定的延时，那么对于一个输出 $y(n)$ 来说，已有大量未来输入数据 $x(n+1)$ ， $x(n+2)$... 存储在存储器中供使用，因此可用具有很大延时的因果系统去逼近非因果系统，这也是数字系统优于模拟系统的特点之一，因而数字系统可以比模拟系统更易获得接近理想的特性。

1.2.5 稳定系统

对每一个有限的输入信号，产生有限输出信号的系统称为稳定系统，一个线性移不变系统是稳定系统的充要条件是系统的单位抽样响应绝对可和，即

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \quad (1.21)$$

证明：

充分条件。若系统满足条件 $\sum_{n=-\infty}^{\infty} |h(n)| < \infty$ ，且输入 $x(n)$ 有界， $|x(n)| \leq M$ ，对所有 n ，

其中 M 是一个任意大的有限数，此时系统的输出为

$$y(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m)$$

两边取绝对值，得

$$|y(n)| = \left| \sum_{m=-\infty}^{\infty} h(m)x(n-m) \right| \leq \sum_{m=-\infty}^{\infty} |h(m)||x(n-m)| \leq M \sum_{m=-\infty}^{\infty} |h(m)| < \infty$$

即输出 $y(n)$ 有界，故系统是稳定的。

必要条件。利用反证法，已知系统稳定，假设 $\sum_{n=-\infty}^{\infty} |h(n)| = \infty$ ，可以找到一个有界的输入

$$x(n) = \begin{cases} 1, & h(n) \geq 0 \\ -1, & h(n) < 0 \end{cases}$$

则

$$y(0) = \sum_{m=-\infty}^{\infty} x(m)h(0-m) = \sum_{m=-\infty}^{\infty} |h(-m)| = \sum_{m=-\infty}^{\infty} |h(m)| = \infty$$

即输出无界，这不符合稳定的假设，因而假设不成立，所以 $\sum_{n=-\infty}^{\infty} |h(n)| < \infty$ 是稳定的必要条件。

例 1.2 设系统输入输出关系为

$$T[x(n)] = x(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

判断其线性、移不变性、因果性和稳定性。

解：

取任意输入 $x_1(n)$ ， $x_2(n)$ ，得输出

$$y_1(n) = T[x_1(n)] = x_1(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

$$y_2(n) = T[x_2(n)] = x_2(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

$$(1) T[ax_1(n) + bx_2(n)] = [ax_1(n) + bx_2(n)] \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

$$= ax_1(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right) + bx_2(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

$$= ay_1(n) + by_2(n)$$

所以此系统为线性系统。

(2) 由于

$$T[x(n-m)] = x(n-m) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

且

$$y(n-m) = x(n-m) \sin\left[\frac{\pi}{5}(n-m) + \frac{2\pi}{3}\right]$$

因此

$$T[x(n-m)] \neq y(n-m)$$

所以此系统不是移不变系统，即系统是移变的。

(3) 若 $x(n)$ 有界，即 $|x(n)| \leq M$ ，则

$$|y(n)| = |T[x(n)]| = \left| x(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right) \right|$$

$$\leq |x(n)| \left| \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right) \right|$$

$$\leq M \left| \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right) \right|$$

而

$$\left| \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right) \right| \leq 1$$

所以 $|y(n)| \leq M < \infty$ ，即有界的输入产生有界的输出，因此系统是稳定的。

(4) 因为

$$y(n) = T[x(n)] = x(n) \sin\left(\frac{\pi}{5}n + \frac{2\pi}{3}\right)$$

只与 $x(n)$ 的当前值有关，而与未来值无关，所以系统是因果的。

1.3 常系数线性差分方程

连续线性时不变系统的输入输出关系常用常系数线性微分方程表示，而离散线性移不变系统的输入输出关系常用常系数线性差分方程表示，即

$$y(n) = \sum_{j=0}^M b_j x(n-j) - \sum_{i=1}^N a_i y(n-i) \quad (1.22)$$

或者

$$\sum_{i=0}^N a_i y(n-i) = \sum_{j=0}^M b_j x(n-j), \quad a_0 = 1 \quad (1.23)$$

常系数是指决定系统特征的系数 $a_1, a_2, \dots, a_N, b_1, b_2, \dots, b_M$ 是常数，若系数中含有 n ，则称为“变系数”。差分方程的阶数等于 $y(n)$ 的变量序号的最高值与最低值之差，例如上式就是 N 阶差分方程。线性是指各 $y(n-i)$ 项和各 $x(n-j)$ 项都只有一次幂而且不存在它们的相乘项，否则就是非线性。

求解差分方程有如下几种方法：递推法、时域经典法、卷积法、变换域法等，时域经典法和微分方程的解法类似，比较麻烦，实际应用中很少采用。卷积法则必须知道系统的单位抽样响应，这样利用卷积和就能得到任意输入时的输出响应。变换域法是利用 Z 变换的方法求解差分方程，将在第 2 章中讨论。这里简单介绍离散时域的递推（迭代）算法。

递推解法比较简单，适合计算机求解，但是只能得到数值解，不易直接得到闭合形式（公式）解答。

在本书讨论的范围内，系统的初始状态为零，因而，单位抽样序列 $\delta(n)$ 作用下产生的系统响应（也就是单位抽样响应） $h(n)$ 就能完全代表系统，那么对于线性移不变系统，任意输入下的系统输出就可以利用卷积和求得。差分方程在给定输入和初值条件下，可用迭代的方法求系统的响应，当输入为 $\delta(n)$ 时，输出（响应）就是单位抽样响应 $h(n)$ 。

例 1.3 常系数差分方程为

$$y(n) = x(n) + \frac{1}{2}y(n-1)$$

(1) 初始条件为 $n < 0$ 时， $y(n) = 0$ ，求其单位抽样响应。

(2) 初始条件为 $n \geq 0$ 时， $y(n) = 0$ ，求其单位抽样响应。

解：(1) 设 $x(n) = \delta(n)$ ，且 $y(-1) = h(-1) = 0$ ，必有

$$y(n) = h(n) = 0, \quad n < 0$$

依次迭代

$$y(0) = h(0) = 1 + \frac{1}{2}h(-1) = 1 + 0 = 1$$

$$y(1) = h(1) = 0 + \frac{1}{2}h(0) = 0 + \frac{1}{2} = \frac{1}{2}$$

$$y(2) = h(2) = 0 + \frac{1}{2}h(1) = \frac{1}{2} \times \frac{1}{2} = \left(\frac{1}{2}\right)^2$$

N

$$y(n) = h(n) = 0 + \frac{1}{2}h(n-1) = 0 + \left(\frac{1}{2}\right)^n = \left(\frac{1}{2}\right)^n$$

所以归纳得单位抽样响应为

$$h(n) = \left(\frac{1}{2}\right)^n u(n) = \begin{cases} \left(\frac{1}{2}\right)^n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

(2) 设 $x(n] = \delta(n)$, 有初始条件知道必有

$$y(n) = h(n) = 0, \quad n \geq 0$$

将原式该写为另一种递推关系

$$y(n-1) = 2[y(n) - x(n)]$$

则

$$y(-1) = h(-1) = 2 \times (0 - 1) = -2$$

$$y(-2) = h(-2) = 2 \times (-2 - 0) = -2^2$$

$$y(-3) = h(-3) = 2 \times (-2^2 - 0) = -2^3$$

N

$$y(n) = h(n) = -2^{-n} = -\left(\frac{1}{2}\right)^n$$

所以单位抽样响应为

$$h(n) = -\left(\frac{1}{2}\right)^n u(-n-1) = \begin{cases} -\left(\frac{1}{2}\right)^n & n < 0 \\ 0 & n \geq 0 \end{cases}$$

由例 1.3 可见, 对于同一个差分方程, 其初始条件不同, 其解也不相同。

1.4 模拟信号的数字处理

在现实世界里, 声音、图像等各种信号多为模拟信号, 要对它进行数字化处理, 首先要将模拟信号经过采样、量化、编码, 变成数字信号, 即进行模/数 (A/D) 转换, 然后用数字技术进行数字信号处理, 最后经过数/模 (D/A) 转换成为模拟信号, 这一处理过程称为模拟信号的数字信号处理, 如图 1.7 所示。



图 1.7 模拟信号数字处理框图

1.4.1 信号的采样

对模拟信号 $x_a(t)$ 进行等间隔采样，其物理意义是将模拟信号 $x_a(t)$ 送入一电子开关，如图 1.8 所示，该开关每隔 T 秒闭合一次，从而获得采样信号 $x_s(t) = x_a(nT)$ ，相当于将 $x_a(t)$ 乘以以 T 为周期的冲激函数 $f_p(t)$ ，即

$$x_s(t) = x_a(t)\delta_p(t) = x_a(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x_a(t)\delta(t - nT) \quad (1.24)$$

由于 $\delta(t - nT)$ 仅在 $t = nT$ 时不为零，故

$$x_s(t) = \sum_{n=-\infty}^{\infty} x_a(nT)\delta(t - nT) \quad (1.25)$$

显然，采样信号 $x_s(t)$ 仅在 $t = 0, \pm T, \pm 2T$ 等处有值，形成离散信号——序列，即

$$x(n) = x_s(t)|_{t=nT} = x_a(nT) \quad (1.26)$$

其中 T 为采样周期，其倒数 $1/T = f_s$ ，称为采样频率。

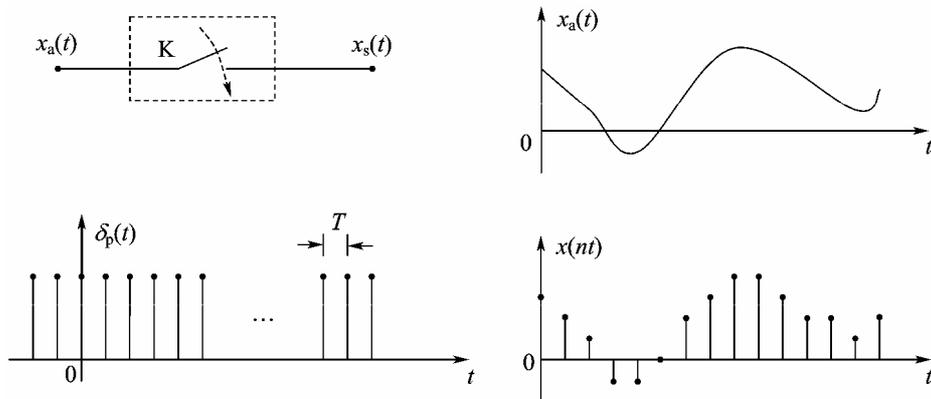


图 1.8 模拟信号的采样

那么采样得到的序列是否能代表原始信号呢？为此须研究它们对应的频谱。令 $X_s(j\Omega)$ 、 $X_a(j\Omega)$ 和 $\Delta_p(j\Omega)$ 分别代表 $x_s(t)$ 、 $x_a(t)$ 和 $f_p(t)$ 的频谱，即

$$X_s(j\Omega) = \int_{-\infty}^{\infty} x_s(t)e^{-j\Omega t} dt \quad (1.27)$$

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t)e^{-j\Omega t} dt \quad (1.28)$$

$$\Delta_p(j\Omega) = \Omega_s \sum_{k=-\infty}^{\infty} \delta[j(\Omega - k\Omega_s)] \quad (1.29)$$

式中 $\Omega_s = 2\pi/T = 2\pi f_s$ ，为采样角频率。

因为 $x_s(t) = x_a(t)\delta_p(t)$ ，所以

$$\begin{aligned}
 X_s(j\Omega) &= \frac{1}{2\pi} X_a(j\Omega) * \Delta_p(j\Omega) \\
 &= \frac{1}{2\pi} X_a(j\Omega) * \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(j\Omega - jk\Omega_s) \\
 &= \frac{1}{T} \int_{-\infty}^{\infty} X_a(j\theta) \sum_{k=-\infty}^{\infty} \delta(j\Omega - jk\Omega_s - \theta) d\theta \\
 &= \frac{1}{T} \sum_{k=-\infty}^{\infty} \int_{-\infty}^{\infty} X_a(j\theta) \delta(j\Omega - jk\Omega_s - \theta) d\theta \\
 &= \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(j\Omega - jk\Omega_s) \tag{1.30}
 \end{aligned}$$

式 (1.30) 表明, 采样信号的频谱 $X_s(j\Omega)$ 是原信号频谱 $X_a(j\Omega)$ 的周期性延拓, 延拓周期为采样频率 Ω_s , 但其幅度有 $1/T$ 加权。对于最高截止频率为 Ω_c 的带限信号, 波形如图 1.9 所示, 图 1.9 (a) 为基带频谱, 图 1.9 (b) 为 $f_p(t)$ 的离散谱, 图 1.9 (c) 表明当 $\Omega_s > 2\Omega_c$ 时, 基带频谱产生以 Ω_s 为周期的周期延拓, 各延拓分量与基带频谱互不重叠, 可以用一个带宽为 Ω_s 的理想低通滤波器滤出基带频谱, 从而不失真的恢复出原模拟信号。图 1.9 (d) 则表明当 $\Omega_s \leq 2\Omega_c$ 时, $X_a(j\Omega)$ 周期延拓后, 前一周期的高频成分和后一周期的低频成分彼此重叠, 这时再用理想低通滤波器滤波就不能分离出原来的基带频谱, 使信息丢失, 形成频谱“混迭”。

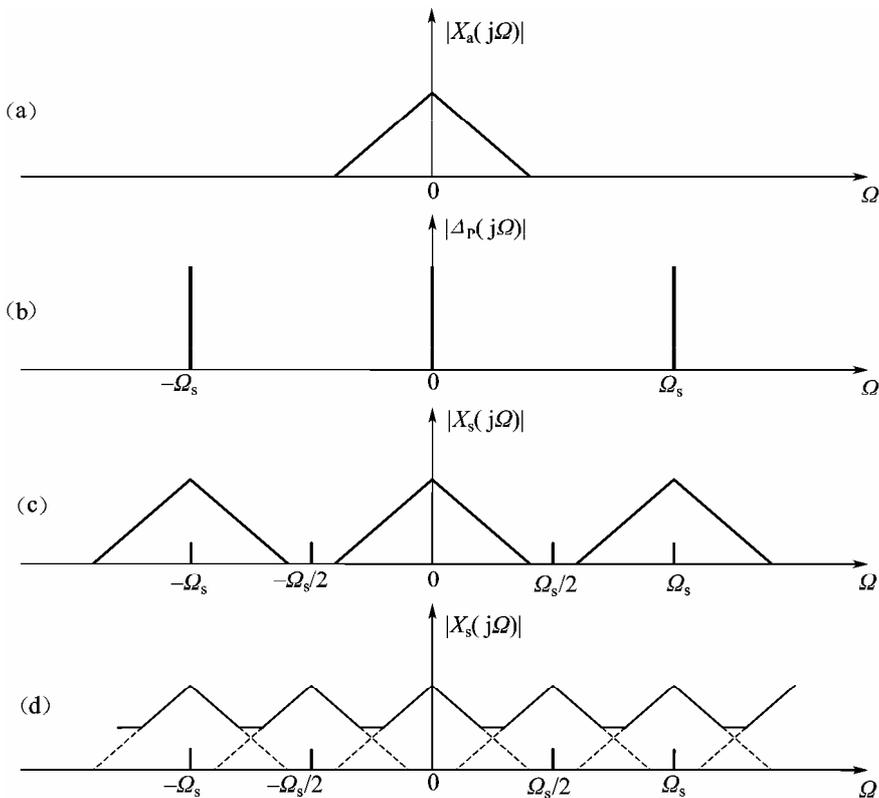


图 1.9 采样信号的频谱

由此可见,对于一个 $\Omega \leq \Omega_C$ 的带限信号,只要采样频率高于带限信号最高频率的两倍,即 $\Omega_s > 2\Omega_C$ 时,可以由其采样信号惟一正确地重建原始信号,这就是采样定理或奈奎斯特(Nyquist)定理。 $2\Omega_C$ 称为奈奎斯特频率, $\Omega_s/2$ 称为折迭频率,信号频率超过它时会折迭回来,形成频谱混迭。

在实际工作中,为避免频谱混迭,采样频率往往选得比 $2\Omega_C$ 更高些,一般为 $\Omega_s = (3 \sim 5)\Omega_C$ 。另外为避免高于 Ω_C 的杂散频率造成频谱混迭,通常在采样之前加入保护性前置低通滤波器——抗混迭滤波器,其截止频率为 Ω_C ,以阻止高于 Ω_C 的频率分量进入采样器。

采样信号是时域离散信号,还需要经过保持、量化、编码才能成为数字信号,在工程上有现成的集成电路芯片可以将模拟信号转换成数字信号,这样的芯片称为模/数转换器(ADC),现在有各种不同频率、不同字长的ADC芯片可以选用。

1.4.2 信号的恢复

前面已经指出,在满足采样定理时,采样信号的频谱不会发生混迭,它只要通过频率特性为

$$H(j\Omega) = \begin{cases} T, & |\Omega| < \Omega_s/2 \\ 0, & |\Omega| \geq \Omega_s/2 \end{cases} \quad (1.31)$$

的理想低通滤波器就可以恢复出原始信号 $x_a(t)$,这是因为当符合采样定理时

$$\begin{aligned} X_s(j\Omega) &= \frac{1}{T} X_a(j\Omega), \quad |\Omega| < \Omega_s/2 = \pi/T \\ X_a(j\Omega) &= X_s(j\Omega) \cdot H(j\Omega) \end{aligned} \quad (1.32)$$

再对 $X_a(j\Omega)$ 进行傅里叶反变换就可以恢复 $x_a(t)$ 。

信号的恢复也可以在时域直接进行。由式 1.32 可得

$$x_a(t) = x_s(t) * h(t)$$

因为

$$\begin{aligned} h(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\Omega) e^{j\Omega t} d\Omega \\ &= \frac{T}{2\pi} \int_{-\Omega_s/2}^{\Omega_s/2} e^{j\Omega t} d\Omega \\ &= \sin c \frac{\pi t}{T} \end{aligned}$$

所以

$$\begin{aligned} x_a(t) &= \sum_{n=-\infty}^{\infty} x_a(nT) \delta(t-nT) * \sin c(\pi t/T) \\ &= \sum_{n=-\infty}^{\infty} x_a(nT) \sin c[\pi(t-nT)/T] \\ &= \sum_{n=-\infty}^{\infty} x_a(nT) \phi_n(t) \end{aligned} \quad (1.33)$$

其中

$$\phi_n(t) = \sin c \frac{\pi(t-nT)}{T} = \frac{\sin[\pi(t-nT)/T]}{\pi(t-nT)/T} \quad (1.34)$$

定义为时域内插函数，如图 1.10 所示。在抽样点 nT 上的函数值为 1，在其余抽样点上函数值为零。式 (1.33) 称为抽样内插公式，它表明了模拟信号 $x_a(t)$ 如何由它的抽样值 $x_a(nT)$ 来表达。在每一个抽样点上，由于只有该抽样值所对应的内插函数不为零，抽样内插公式保证了各采样点上信号值不变，而采样点之间的信号值则是由各抽样值对应的内插函数的波形延伸叠加而成，如图 1.11 所示。

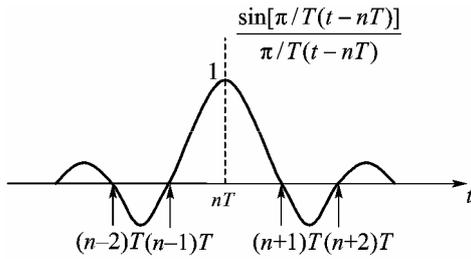


图 1.10 内插函数

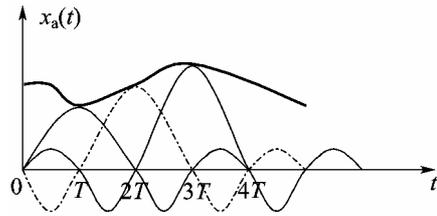


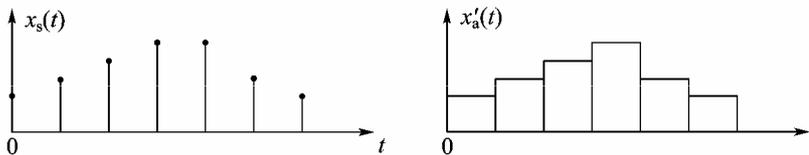
图 1.11 抽样的内插恢复

上述的信号恢复是一种没有失真的恢复，但是由于使用的是理想低通滤波器，其 $h(t)$ 是非因果的，因此在物理上是不可实现的，实际使用的将数字信号转换成模拟信号的数/模转换器 (DAC) 组成如图 1.12 (a) 所示。其中“解码”的作用是将数字信号转换为时域离散信号，即采样值 $x_s(t)$ ，“零阶保持”的作用是将前一个取样值保持到下一个取样时刻，因此相当于常数内插，在时域上的表现则呈阶梯状，其输入、输出波形如图 1.12 (b) 所示。它的单位冲激响应 $h(t)$ 、幅度响应 $|H(j\Omega)|$ 、相位响应 $\arg[H(j\Omega)]$ 分别如图 1.13 (a)、图 1.13 (b)、图 1.13 (c) 所示，频率响应 $H(j\Omega)$ 为

$$H(j\Omega) = \int_{-\infty}^{\infty} h(t)e^{j\Omega t} dt = \int_0^T e^{j\Omega t} dt = T \frac{\sin \frac{\Omega T}{2}}{\frac{\Omega T}{2}} e^{-j\frac{\Omega T}{2}}$$



(a) 数模变换原理框图



(b) 零阶保持器输入、输出波形图

错误！

图 1.12 数模变换原理框图及信号波形图

零阶保持器是一个低通滤波器，起到将采样信号恢复成模拟信号的作用，但是它的幅度特性和理想低通滤波器相比，在 $|\Omega| > \pi/T$ 区域有较多的高频分量通过，为此在零阶保持

器后再接一模拟低通滤波器，滤除不必要的高频分量，对波形起到平滑的作用，因此该滤波器也称为平滑滤波器。

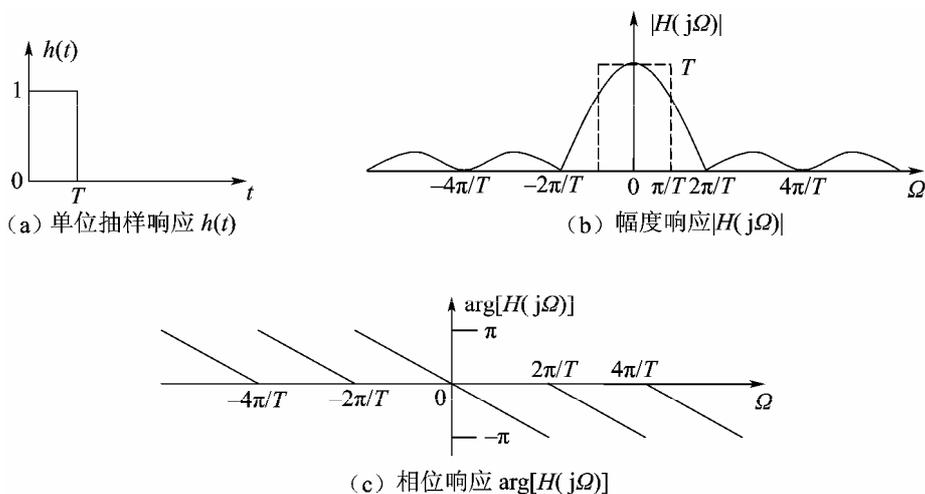


图 1.13 零阶保持器的特性

1.5 用 MATLAB 表示和实现离散时间信号与系统

由于 MATLAB 数值计算的特点，用它来分析离散时间信号与系统是很方便的。在 MATLAB 中，可以用一个列向量来表示一个有限长度的序列，但是这种表示方法没有包含采样位置的信息，要完全表示 $x(n)$ ，要用 x 和 n 两个向量，例如

$$x(n) = \{2, 1, 0, 2, 3, -1, 2, 3\}$$

(箭头表示第 0 个采样点的位置)

在 MATLAB 中表示为

$$n = [-4, -3, -2, -1, 0, 1, 2, 3]; x = [2, 1, 0, 2, 3, -1, 2, 3];$$

当不需要采样位置信息（例如序列从 $n=0$ 开始）时，可以只用 x 来表示。由于内存有限，MATLAB 无法表示无限序列。

例 1.4 编写程序来产生下列基本脉冲序列。

- (1) 单位抽样序列，起点 $n_s = 0$ ，终点 $n_f = 10$ ，在 $n_0 = 3$ 处有一单位脉冲。
- (2) 单位阶跃序列，起点 $n_s = 0$ ，终点 $n_f = 10$ ，在 $n_0 = 3$ 前为 0，在 $n_0 = 3$ 后为 1。
- (3) 复指数序列， $f = -0.2$ ， $\omega_0 = 0.5$ 。

解：程序清单如下。

```
clear, n0=3; ns=0; nf=10;
n1=[ns:nf]; x1=[zeros(1, n0-ns), 1, zeros(1, nf-n0)]; %单位抽样序列的产生
n2=[ns:nf]; x2=[zeros(1, n0-ns), ones(1, nf-n0+1)]; %单位阶跃序列的产生
n3=[ns:nf]; x3=exp((-0.2+0.5j)*n3); %复指数序列的产生
```

```

subplot(2,2,1),stem(n1,x1);title('单位脉冲序列  $f(n-3)$ '); %画图表示单位抽样序列
subplot(2,2,2),stem(n2,x2);title('单位阶跃序列  $u(n-3)$ '); %画图表示单位阶跃序列
subplot(2,2,3),stem(n3, real(x3)); line([0, 10],[0, 0])
title('复指数序列');ylabel('实部'); %画图表示复指数序列的实部
subplot(2,2,4),stem(n3, imag(x3)); line([0, 10],[0, 0])
title('复指数序列');ylabel('虚部');

```

程序运行结果如图 1.14 所示。

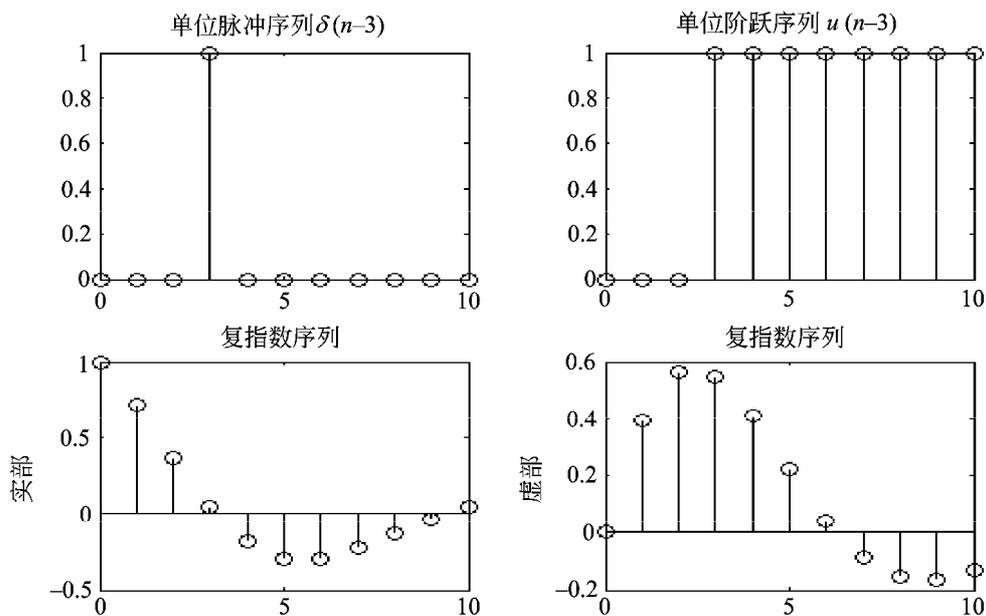


图 1.14 几种基本脉冲序列

在数字信号处理的基本理论和 MATLAB 信号处理工具箱函数的基础上，可以自己编写一些子程序以便调用。

(1) 单位抽样序列 $\delta(n-n_0)$ 的生成函数 impseq.m

```

function [x,n]=impseq(n0,ns,nf)
n=[ns:nf];x=[(n-n0)==0]; %序列的起点为 ns,终点为 nf,在 n=n0 点处生成一个单位脉冲。

```

(2) 单位阶跃序列 $u(n-n_0)$ 的生成函数 stepseq.m

```

function [x,n]=stepseq(n0,ns,nf)
n=[ns:nf];x=[(n-n0)>=0]; %序列的起点为 ns,终点为 nf,在 n=n0 点处生成单位阶跃。

```

(3) 两个信号相加的生成函数 sigadd.m

```

function [y,n]=sigadd(x1,n1,x2,n2)
n=min(min(n1),min(n2)):max(max(n1),max(n2))
y1=zeros(1,length(n));y2=y1;
y1=(find((n>=min(n1))&(n<=max(n1))==1))=x1;

```

```
y2=(find((n>=min(n2))&(n<=max(n2))==1))==x2;
y=y1+y2;
```

(4) 两个信号相乘的生成函数 sigmult.m

```
function [y,n]=sigmult(x1,n1,x2,n2)
n=min(min(n1),min(n2)):max(max(n1),max(n2))
y1=zeros(1,length(n));y2=y1;
y1=(find((n>=min(n1))&(n<=max(n1))==1))==x1;
y2=(find((n>=min(n2))&(n<=max(n2))==1))==x2;
y=y1.*y2;
```

(5) 序列移位 $y(n) = x(n - n_0)$ 的生成函数 sigshift.m

```
function [y,n]=sigshift(x,m,n0)
n=m+n0;y=x
```

(6) 序列翻褶 $y(n) = x(-n)$ 的生成函数 sigfold.m

```
function [y,n]=sigfold(x,n)
y=fliplr(x);n=-fliplr(n)
```

(7) 奇偶综合函数 evenodd.m

此函数可以将任一给定的序列 $x(n)$ 分解为 $x_e(n)$ 和 $x_o(n)$ 两部分。

```
function [xe,xo,m]=evenodd(x,n)
if any (imag(x)=0)
    error('x is not a real sequence')
m=-fliplr(n);m1=min([m,n]);m2=max([m,n]);m=m1:m2
nm=n(1)-m(1);n1=1:length(n);
x1=zeros(1,length(m));
x1(n1+nm)=x;x=x1;
xe=0.5*(x+fliplr(x)); xo=0.5*(x-fliplr(x));
```

(8) 求卷积和

求卷积和直接采用 MATLAB 中的函数 conv，即 $y = \text{conv}(x, h)$ ，它默认序列从 $n = 0$ 开始。但是如果序列是从一负值开始，即

$$\{x(n) : nx1 \leq n \leq nx2\}$$

$$\{h(n) : nh1 \leq n \leq nh2\}$$

其中 $nx1 < 0$ 或 $nh1 < 0$ ，或两者同时为负，这样就不能直接采用 conv 函数。其卷积结果序列为 $\{y(n) : nx1 + nh1 \leq n \leq nx2 + nh2\}$ ，这样就可构成一个新的卷积函数 conv_m，求出带下标的序列卷积。

```
function [y,ny]=conv_m(x,nx,h,nh)
ny1=nx(1)+nh(1);ny2=nx(length(x))+nh(length(h));
ny=[ny1:ny2];
```

$y=\text{conv}(x,h)$

例 1.5 产生系列序列，并绘出离散图。

$$(1) x_1(n)=3f(nf-2)-(n+4) \quad -5 \leq n \leq 5$$

$$(2) x_2(n)=n[u(n)-u(n-5)]+10e^{-0.2(n-5)}[u(n-10)-u(n-20)] \quad 0 \leq n \leq 20$$

$$(3) x_3(n)=\cos(0.04\pi n)+0.2w(n) \quad 0 \leq n \leq 50$$

其中 $w(n)$ 是均值为 0，方差为 1 的白噪声序列。

$$(4) x_4(n)=\tilde{x}(n)=[x_1, x_1, x_1, x_1], \text{其中 } x_1=[1 \ 0 \ 1 \ 2 \ 3]$$

解：程序清单如下。

```
n=[-5:5];x1=3*impseq(2,-5,5)-impseq(-4,-5,5);
subplot(2,2,1);stem(n,x1);xlabel('n');ylabel('x1(n)');axis([-5,5,-2,4]);
```

```
n=[0:20];x21=n.*(stepseq(0,0,20)-stepseq(10,0,20));
x22=10*exp(-0.2*(n-5)).*(stepseq(10,0,20)-stepseq(20,0,20));x2=x21+x22;
subplot(2,2,2);stem(n,x2);xlabel('n');ylabel('x2(n)');axis([0,20,-1,11])
```

```
n=[0:50];x3=cos(0.04*pi*n)+0.2*randn(size(n));
subplot(2,2,3);stem(n,x3);xlabel('n');ylabel('x3(n)');axis([0,50,-2,2])
```

```
n=[-10:9];x4=[1,0,1,2,3];xTilde=x4'*ones(1,4);xtilde=(xtilde(:))';
subplot(2,2,4);stem(n,xtilde);xlabel('n');ylabel('x4(n)');axis([-10,9,-1,4])
```

程序运行结果如图 1.15 所示。

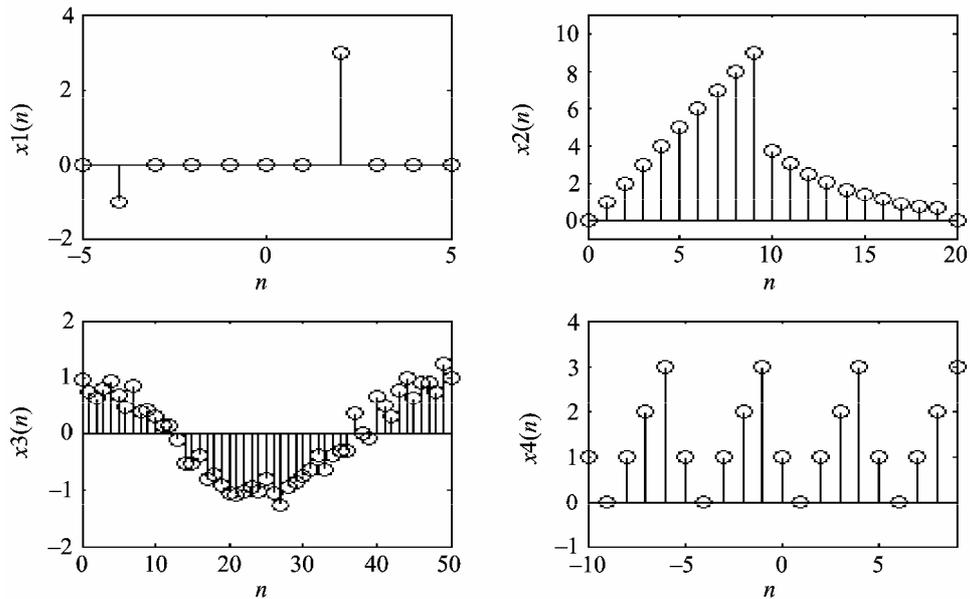


图 1.15 产生系列序列的示意图

例 1.6 已知 $x(n)=u(n)-u(n-10)$ ，要求将序列分解为奇偶序列。

解：程序清单如下。

```
n=[0:10];x=stepseq(0,0,10)- stepseq(10,0,10);
[xe,xo,m]=evenodd(x,n);
subplot(2,2,1);stem(n,x);ylabel('x(n)');title('矩形序列');axis([-10,10,
-1.2,1.2])
subplot(2,2,2);stem(m,xe);ylabel('xe(n)');title('奇序列');axis([-10,10,
-1.2,1.2])
subplot(2,2,4);stem(m,xo);ylabel('xo(n)');title('偶序列');axis([-10,10,
-1.2,1.2])
```

程序运行结果如图 1.16 所示。

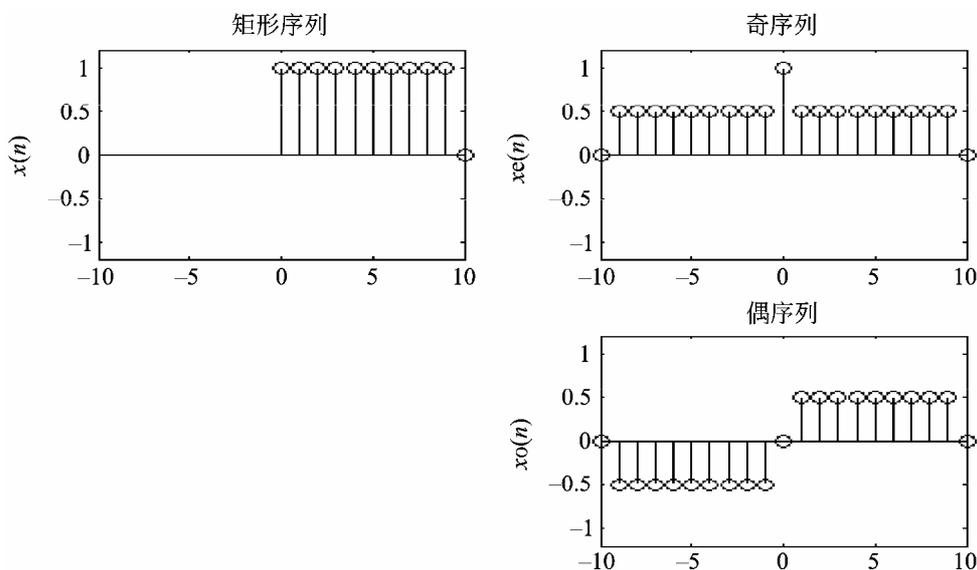


图 1.16 序列分解为奇偶序列

例 1.7 设线性移不变系统的抽样响应 $h(n) = (0.9)^n u(n)$ ，输入序列 $x(n) = u(n) - u(n-10)$ ，求系统的输出 $y(n)$ 。

解：系统的输出 $y(n)$ 为输入 $x(n)$ 与系统的单位抽样响应 $h(n)$ 的卷积，即 $y(n) = x(n) * h(n)$ ，可直接采用 `conv_m` 函数求输出序列。

程序清单如下。

```
n=-5:50;x=stepseq(0,-5,50)-stepseq(10,-5,50);h=((0.9).^n).* stepseq(0,-5,50);
subplot(3,1,1);stem(n,x);axis([-5,50,0,2]);ylabel('x(n)');
subplot(3,1,2);stem(n,h);axis([-5,50,0,2]);ylabel('h(n)');
[y,ny]=conv_m(x,n,h,n);subplot(3,1,3);stem(ny,y);axis([-5,50,0,8]);
xlabel('n');ylabel('y(n)');
```

程序运行结果如图 1.17 所示。

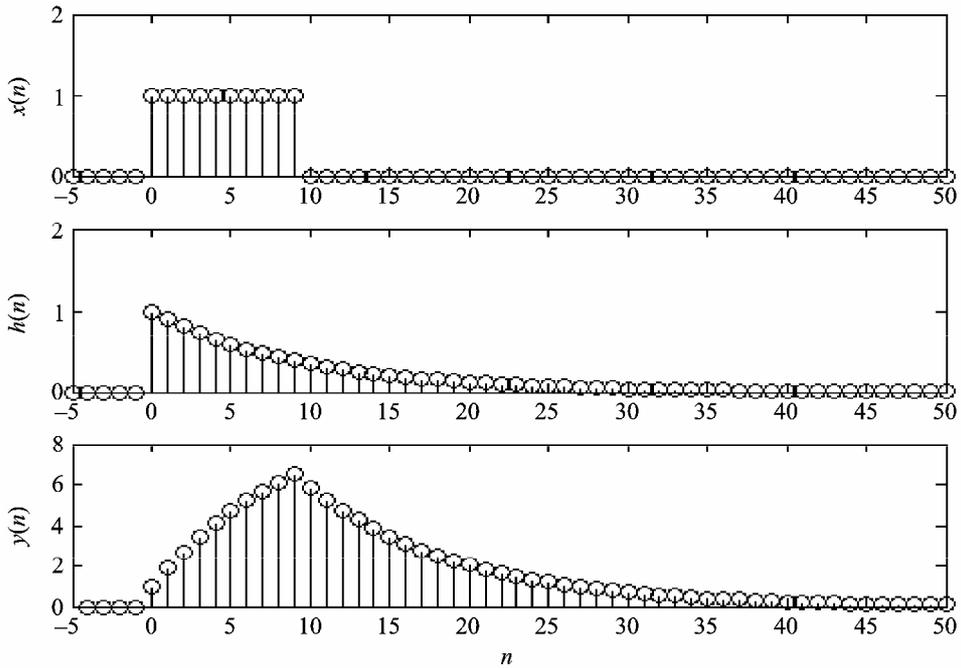


图 1.17 线性移不变系统的输出

例 1.8 给定因果稳定线性移不变系统的差分方程

$$\sum_{k=0}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k)$$

对下列输入序列 $x(n)$ ，求出输出序列 $y(n)$ 。

(1) $x(n) = f(n)$; (2) $x(n) = u(n)$; (3) $x(n) = R_{30}(n)$; (4) $x(n) = e^{j\frac{\pi}{4}n} u(n)$

解：程序清单如下。

```
N=100;n=0:N-1;
b=1;
a=[1,-1,0.9];

x1=[n==0];y1=filter(b,a,x1);
subplot(3,2,1);stem(n,y1,'. ');axis([0,N,-1,2]);ylabel('y1(n)');

x2=[n>=0];y2=filter(b,a,x2);
subplot(3,2,2);stem(n,y2,'. ');axis([0,N,-1,2]);ylabel('y2(n)');

x3=[(n>=0)&(n<30)];y3=filter(b,a,x3);
subplot(3,2,3);stem(n,y3,'. ');axis([0,N,-1,2]);ylabel('y3(n)');

x4=exp(j*pi*n/8);y4=filter(b,a,x4);
```

```
subplot(3,2,5);stem(n,real(y4),'.');axis([0,N,-2,2]);ylabel('Re[y4(n)]');
subplot(3,2,6);stem(n,imag(y4),'.');axis([0,N,-2,2]);ylabel('Im[y4(n)]');
```

程序运行结果如图 1.18 所示。

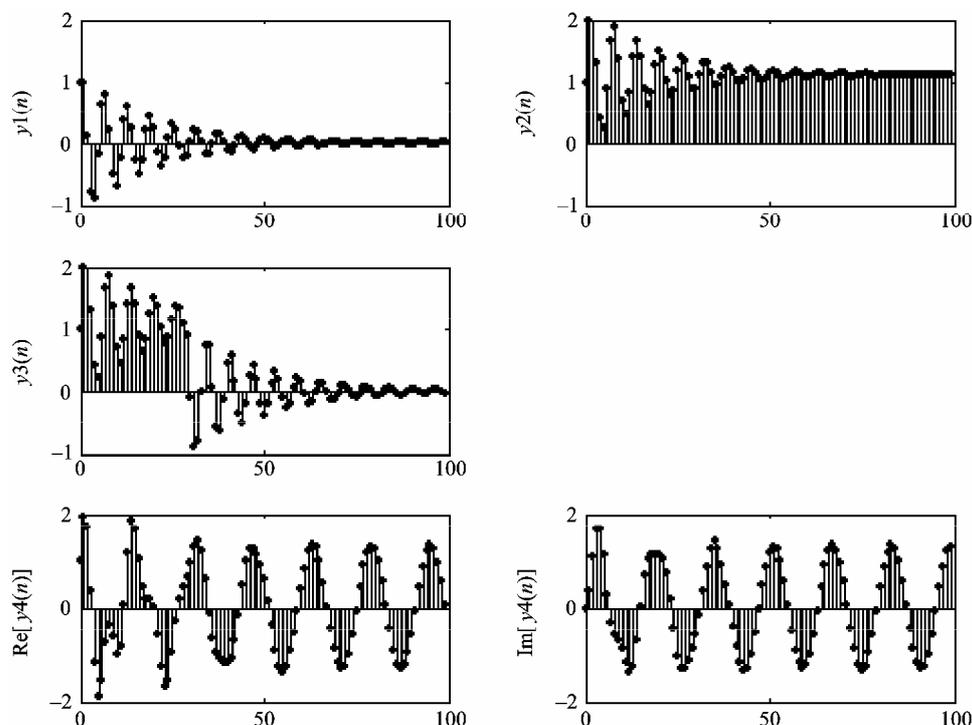


图 1.18 例 1.7 中的数字滤波器对四种信号的响应图

例 1.9 $x_a(t) = e^{-1000|t|}$ ，求其傅里叶变换 $X_a(j\omega)$ 。

解：严格说，在 MATLAB 中不使用 symbolic 工具箱是不能分析模拟信号的，但是当采样时间间隔充分小的时候，可产生平滑的图形，当时间足够长，可显示出所有的模型，也就是可以近似的分析。此例中， $x_a(t)$ 为 $f_h = 2000\text{Hz}$ 的带限信号，因此取

$$T = 5 \times 10^{-5} \ll \frac{1}{2 \times 2000} = 2.5 \times 10^{-4} \text{ s}$$

程序清单如下。

```
Dt=0.00005;t=-0.005:Dt:0.005; %模拟信号
xa=exp(-1000*abs(t));

Wmax=2*pi*2000;K=500;k=0:1:K;W=k*Wmax/K; %连续时间傅里叶变换
Xa=xa*exp(-j*t'*W)*Dt;Xa=real(Xa);
W=[-fliplr(W),W(2:501)];
Xa=[fliplr(Xa),Xa(2:501)];

figure(1)
subplot(2,1,1);plot(t*1000,xa,'.');xlabel('t in msec');ylabel('xa(t)');
```

```
gtext('模拟信号');
subplot(2,1,2);plot(W/(2*pi*1000),Xa*1000,'. ');
xlabel('Frequency in KHz');ylabel('Xa(jw)*1000');gtext('连续时间傅里叶变换');
```

程序运行结果如图 1.19 所示。

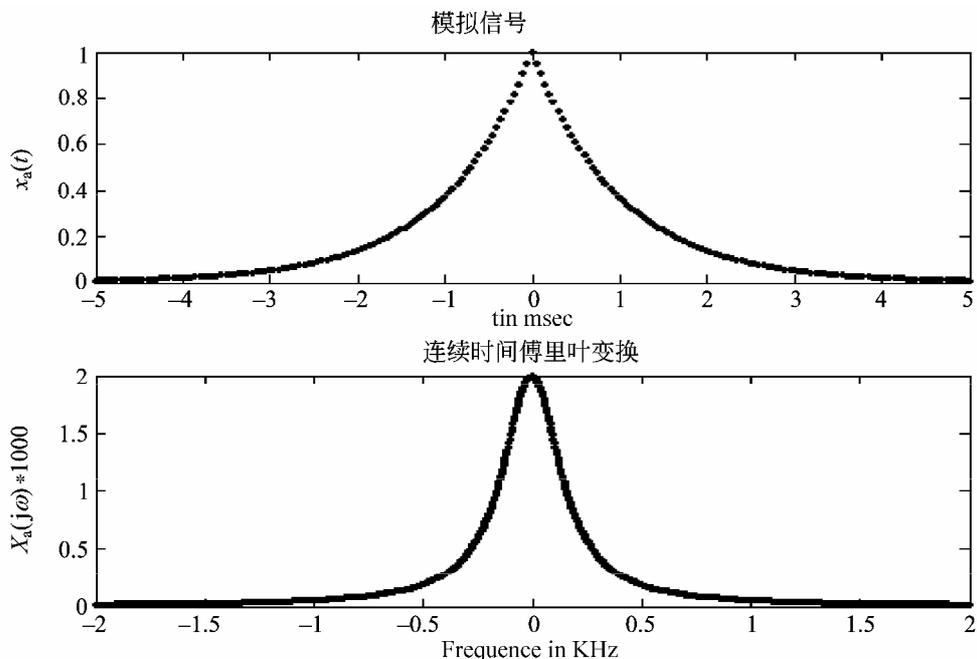


图 1.19 模拟信号及傅里叶变换曲线

例 1.10 以例 1.8 中的 $x_a(t)$ 说明采样频率对频域特性的影响，分别取采样频率为 $f_s = 5000\text{Hz}$ 和 $f_s = 1000\text{Hz}$ ，绘出 $X_a(j\omega)$ 曲线。

解：程序清单如下。

```
Dt=0.00005;t=-0.005:Dt:0.005; %模拟信号
xa=exp(-1000*abs(t));

Ts=0.0002;n=-25:1:25; %离散时间信号
x=exp(-1000*abs(n*Ts));

K=500;k=0:1:K;w=pi*k/K; %离散时间傅里叶变换
X=x*exp(-j*n'*w);X=real(X);
w=[-fliplr(w),w(2:501)];
X=[fliplr(X),X(2:501)];

figure
subplot(2,1,1);plot(t*1000,xa,'. '); ylabel('xa1(t)');
gtext('离散信号');hold on
stem(n*Ts*1000,x);hold off
```

```
subplot(2,1,2);plot(w/pi,X,'. ');ylabel('X1(jw)');
gtext('离散时间傅里叶变换'); gtext('Ts=0.2msec')
```

程序运行结果如图 1.20 所示。

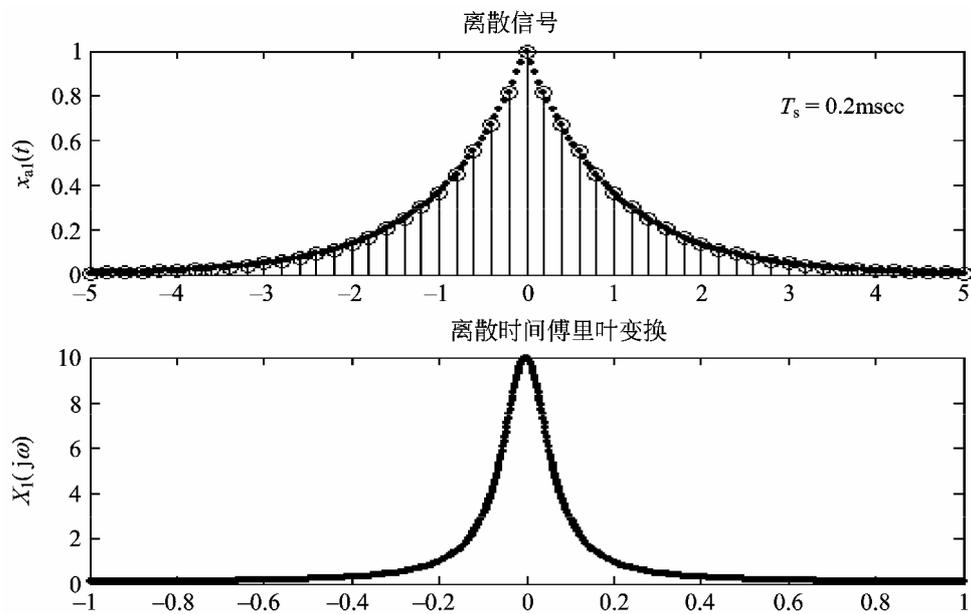


图 1.20 $f_s = 5000\text{Hz}$ 时的离散信号及傅里叶变换曲线

对上面的 MATLAB 程序稍加修改，就可以得到 $f_s = 1000\text{Hz}$ 时的 $X_2(j\omega)$ ，程序运行结果如图 1.21 所示。可以看出，当 $f_s = 5000\text{Hz}$ 时， $X_1(j\omega)$ 的曲线和例 1.9 中的 $X_a(j\omega)$ 曲线完全一致，没有发生混叠现象，而当 $f_s = 1000\text{Hz}$ 时， $X_2(j\omega)$ 的曲线与例 1.9 中的 $X_a(j\omega)$ 曲线不同，产生了混叠的现象。

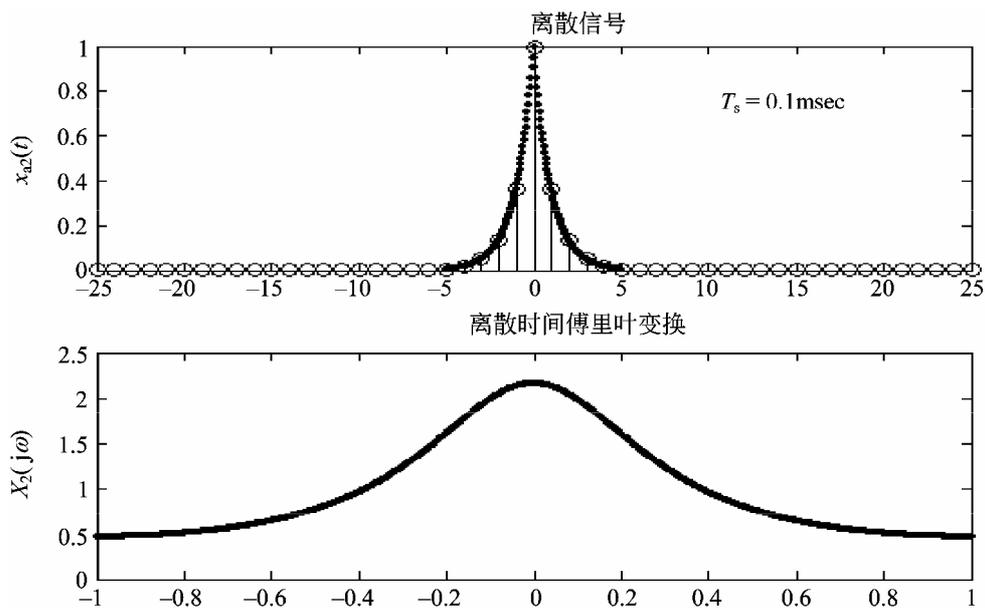


图 1.21 $f_s = 1000\text{Hz}$ 时的离散信号及傅里叶变换曲线

例 1.11 对例 1.10 中用不同采样频率产生的离散序列 $x_1(n)$ 和 $x_2(n)$ ，采用 sinc 函数进行内插重构。

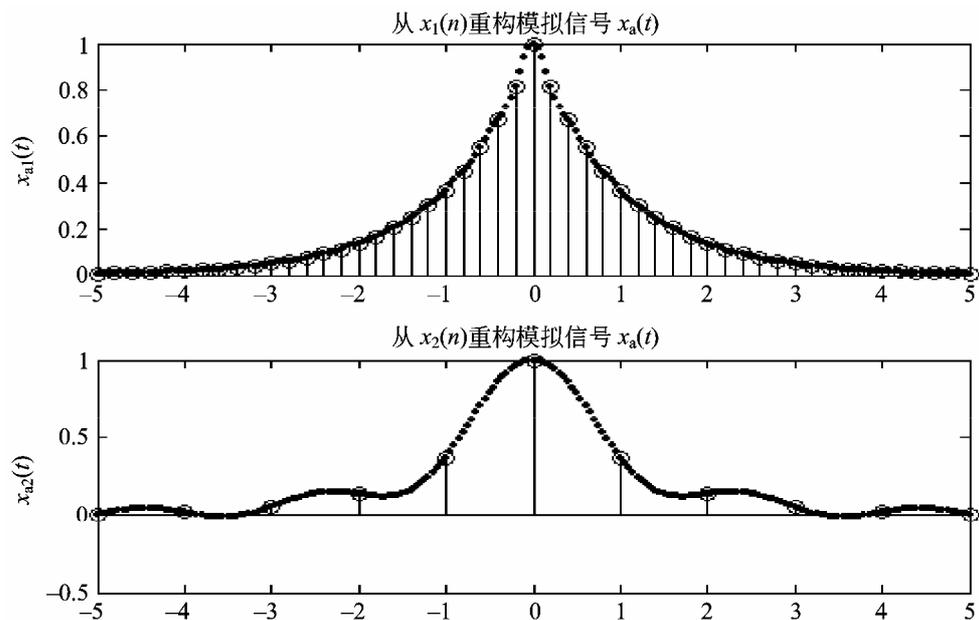
解：程序清单如下。

```
Ts1=0.0002;Fs1=1/Ts1;n1=-25:1:25;nTs1=n1*Ts1;           %离散时间信号
x1=exp(-1000*abs(nTs1));
Ts2=0.001;Fs2=1/Ts2;n2=-5:1:5;nTs2=n2*Ts2;
x2=exp(-1000*abs(nTs2));

Dt=0.00005;t=-0.005:Dt:0.005;                           %模拟信号重构
xa1=x1*sinc(Fs1*(ones(length(nTs1),1)*t-nTs1'*ones(1,length(t))));
xa2=x2*sinc(Fs2*(ones(length(nTs2),1)*t-nTs2'*ones(1,length(t))));

subplot(2,1,1);plot(t*1000,xa1,'.');ylabel('xa1(t)');
title('从 x1(n)重构模拟信号 xa(t)');hold on
stem(n1*Ts1*1000,x1);hold off
subplot(2,1,2);plot(t*1000,xa2,'.');ylabel('xa2(t)');
title('从 x2(n)重构模拟信号 xa(t)');hold on
stem(n2*Ts2*1000,x2);hold off
```

程序运行结果如图 1.22 所示。

图 1.22 从 $x_1(n)$ 和 $x_2(n)$ 分别重构模拟信号

将图 1.22 中的重构后的模拟信号曲线与图 1.19 中的原始模拟信号曲线进行比较，可以看出：用离散信号 $x_1(n)$ 重构出的模拟信号与原始信号误差很小，而用离散信号 $x_2(n)$ 重构出的模拟信号误差较大，这是因为 $x_2(n)$ 的采样频率为 $f_s = 1000\text{Hz}$ ，不满足采样定理 $f_s > 2f_h$ ，因此采样信号的频谱发生混叠造成取样信号不能无失真的恢复出原始信号。

第 2 章 Z 变换

信号与系统的分析方法有时域分析法和变换域分析法，在连续时间系统中，其变换域方法是拉普拉斯变换和傅里叶变换，在离散时间系统中，其变换域方法是 Z 变换和傅里叶变换。对分析离散时间系统而言，Z 变换是个极重要的数学工具，它可以将描述离散系统的差分方程转化为简单的代数方程，从而使求解大大简化。本章直接给出 Z 变换的定义，最后再推导 Z 变换和拉普拉斯变换的关系。

2.1 Z 变换的定义及收敛域

2.1.1 Z 变换的定义

对于一个序列 $x(n)$ ，它的 Z 变换定义为

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (2.1)$$

其中 Z 为一个复变量，式 (2.1) 定义的 Z 变换称为双边 Z 变换或标准 Z 变换，当求和符号中的 n 取 $0 \sim \infty$ 时，则为单边 Z 变换。

2.1.2 Z 变换的收敛域 (ROC)

由于 $x(n)$ 的 Z 变换是一个无穷级数，就必然存在收敛和发散的问题，仅当级数收敛时才可将其表示成一个闭合形式，按照级数理论，级数收敛的充要条件是满足绝对可和的条件，即

$$\sum_{n=-\infty}^{\infty} |x(n)z^{-n}| = M < \infty \quad (2.2)$$

因此，使上式成立的所有 Z 值的集合称为 $X(z)$ 的收敛域，一般用 $R_{x-} < |z| < R_{x+}$ 表示，如图 2.1 所示， R_{x-} 和 R_{x+} 称为收敛半径，当然， R_{x-} 可以小到 0， R_{x+} 可以大到 ∞ 。不同形式的序列其收敛域不同，讨论如下。

1. 有限长序列

序列只在有限长度内取值，即

$$x(n) = \begin{cases} x(n), & n_1 < n < n_2 \\ 0, & n \text{ 为其他值} \end{cases}$$

有限长序列的 Z 变换为

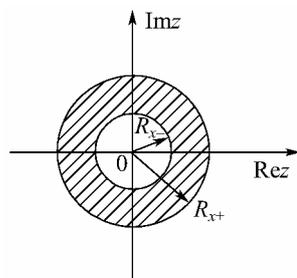


图 2.1 Z 变换的收敛域

$$X(z) = \sum_{n=n_1}^{n_2} x(n)z^{-n} \quad (2.3)$$

因为 $x(n)$ 是有界序列, 故为有限项求和, 显然在 $0 < |z| < \infty$ 上都满足收敛条件, 也就是说收敛域至少是有限 z 平面 $(0, \infty)$, 在 n_1 和 n_2 的特殊取值情况下, 收敛域可扩大为

$$\begin{aligned} 0 < |z| \leq \infty, & \quad n_1 \geq 0 \\ 0 \leq |z| < \infty, & \quad n_2 \leq 0 \end{aligned}$$

收敛域如图 2.2(a) 所示。

例 2.1 矩形序列是一个有限长序列, $x(n) = R_N(n)$, 求其 $X(z)$ 。

解:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} = \sum_{n=0}^{N-1} z^{-n} = \frac{1-z^{-N}}{1-z^{-1}}$$

收敛域为 $0 < |z| \leq \infty$ 。从上式的分母可知在 $z=1$ 处有一个极点, 但是从分子处看出 $z=1$ 处有一个零点, 零极点刚好对消。

2. 右边序列

右边序列只有在 $n \geq n_1$ 时, 序列值不全为零, n 为其他值时, 序列值全为零, 即

$$x(n) = \begin{cases} x(n), & n \geq n_1 \\ 0, & n < n_1 \end{cases}$$

右边序列的 Z 变换为

$$X(z) = \sum_{n=n_1}^{\infty} x(n)z^{-n} \quad (2.4)$$

由级数理论可知, 若一般项为 $|a_n|$, $\lim_{n \rightarrow \infty} \sqrt[n]{|a_n|} = \rho$, 则 $\rho < 1$ 时级数收敛, $\rho > 1$ 时级数发散, $\rho = 1$ 时级数可能收敛也可能发散, 所以对式 (2.4), 若满足

$$\lim_{n \rightarrow \infty} \sqrt[n]{|x(n)z^{-n}|} < 1$$

即

$$|z| > \lim_{n \rightarrow \infty} \sqrt[n]{|x(n)|} = R_{x-}$$

时, 级数收敛, 所以右边序列的收敛域为半径为 R_{x-} 的圆外部区域, 如图 2.2(b) 所示。当 $n_1 \geq 0$ 时, 收敛域为 $R_{x-} < |z| \leq \infty$; 当 $n_1 < 0$ 时, 收敛域为 $R_{x-} < |z| < \infty$, 不包括 $|z| = \infty$ 处; 当 $n_1 = 0$ 时, 序列为因果序列, 因此在 $|z| = \infty$ 处 Z 变换收敛是因果序列的特征。

例 2.2 求指数序列 $x(n) = a^n u(n)$ 的 Z 变换。

解:

$$X(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n = \frac{1}{1-az^{-1}} = \frac{z}{z-a}, \quad |z| > |a|$$

3. 左边序列

左边序列只有在 $n \leq n_2$ 时, 序列值不全为零, n 为其他值时, 序列值全为零, 即

$$x(n) = \begin{cases} x(n), & n \leq n_2 \\ 0, & n > n_2 \end{cases}$$

左边序列的 Z 变换为

$$X(z) = \sum_{n=-\infty}^{n_2} x(n)z^{-n} = \sum_{n=-n_2}^{\infty} x(-n)z^n \quad (2.5)$$

若

$$\lim_{n \rightarrow \infty} \sqrt[n]{|x(-n)z^n|} < 1$$

则

$$|z| < \frac{1}{\lim_{n \rightarrow \infty} \sqrt[n]{|x(-n)|}} = R_2$$

时, 级数收敛, 所以左边序列的收敛域是半径为 R_{x+} 的圆内部区域, 如图 2.2 (c) 所示。当 $n_2 > 0$ 时, 收敛域不包括 $z = 0$, 即 $0 < |z| < R_{x+}$; 当 $n_2 < 0$ 时, 收敛域包括 $z = 0$, 即 $|z| < R_{x+}$ 。

例 2.3 求序列 $x(n) = -b^n u(-n-1)$ 的 Z 变换。

解:

$$X(z) = \sum_{n=-\infty}^{-1} -b^n z^{-n} = \sum_{n=1}^{\infty} -b^{-n} z^n = -\frac{b^{-1}z}{1-b^{-1}z} = \frac{1}{1-bz^{-1}}, \quad |z| < |b|$$

由例 2.2 和例 2.3 可以看出, 如果 $a = b$, 则一个左边序列和一个右边序列的 Z 变换的表达式完全一样, 所以只给出 Z 变换的闭合表达式是不够的, 不能正确得到原序列, 必须同时给出收敛域, 才能惟一确定一个序列, 这表明了研究收敛域的重要性。

4. 双边序列

一个双边序列可以看做一个左边序列和一个右边序列之和, 因此双边序列 Z 变换的收敛域就是这两个序列 Z 变换的公共收敛区间。

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} = \sum_{n=-\infty}^{-1} x(n)z^{-n} + \sum_{n=0}^{\infty} x(n)z^{-n} \quad (2.6)$$

第一项的收敛域为 $|z| < R_{x+}$, 第二项的收敛域为 $|z| > R_{x-}$, 若满足 $R_{x-} < R_{x+}$, 则存在公共收敛域, 即双边序列的收敛域为 $R_{x-} < |z| < R_{x+}$, 如图 2.2 (d) 所示。

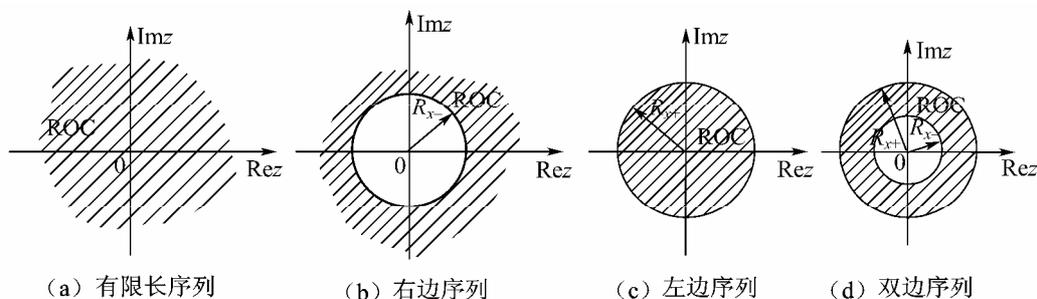


图 2.2 各种序列对应的收敛域

例 2.4 求序列 $x(n) = a^{|n|}$ 的 Z 变换, 其中 $|a| < 1$ 。

解：

$$X(z) = \sum_{n=-\infty}^{\infty} a^{|n|} z^{-n} = \sum_{n=-\infty}^{-1} a^{-n} z^{-n} + \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=1}^{\infty} a^n z^n + \sum_{n=0}^{\infty} a^n z^{-n}$$

第一部分的收敛域为 $|az| < 1$ ，即 $|z| < \frac{1}{|a|}$ ；第二部分的收敛域为 $|az^{-1}| < 1$ ，即 $|z| > |a|$ ，

而且已知 $|a| < 1$ ，所以

$$X(z) = \frac{az}{1-az} + \frac{1}{1-az^{-1}} = \frac{1-a^2}{(1-az)(1-az^{-1})}, \quad |a| < |z| < \frac{1}{|a|}$$

一些常见序列的 Z 变换如表 2.1 所示。

表 2.1 常见序列的 Z 变换

序 列	Z 变 换	收 敛 域
$\delta(n)$	1	$0 \leq z \leq \infty$
$u(n)$	$\frac{z}{z-1} = \frac{1}{1-z^{-1}}$	$ z > 1$
$u(-n-1)$	$-\frac{z}{z-1} = \frac{-1}{1-z^{-1}}$	$ z < 1$
$a^n u(n)$	$\frac{z}{z-a} = \frac{1}{1-az^{-1}}$	$ z > a $
$a^n u(-n-1)$	$-\frac{z}{z-a} = \frac{-1}{1-az^{-1}}$	$ z < a $
$R_n(n)$	$\frac{z^N - 1}{z^{N-1}(z-1)} = \frac{1-z^{-N}}{1-z^{-1}}$	$ z > 0$
$nu(n)$	$\frac{az}{(z-1)^2} = \frac{z^{-1}}{(1-z^{-1})^2}$	$ z > 1$
$na^n u(n)$	$\frac{az}{(z-a)^2} = \frac{az^{-1}}{(1-az^{-1})^2}$	$ z > a $
$e^{-jn\omega_0} u(n)$	$\frac{z}{z-e^{-j\omega_0}} = \frac{1}{1-e^{-j\omega_0} z^{-1}}$	$ z > 1$
$\sin n\omega_0 u(n)$	$\frac{z \sin \omega_0}{z^2 - 2z \cos \omega_0 + 1} = \frac{z^{-1} \sin \omega_0}{1 - 2z^{-1} \cos \omega_0 + z^{-2}}$	$ z > 1$
$\cos n\omega_0 u(n)$	$\frac{z^2 - z \cos \omega_0}{z^2 - 2z \cos \omega_0 + 1} = \frac{1 - z^{-1} \cos \omega_0}{1 - 2z^{-1} \cos \omega_0 + z^{-2}}$	$ z > 1$
$\sin(n\omega_0 + \theta) u(n)$	$\frac{z^2 \sin \theta + z \sin(\omega_0 - \theta)}{z^2 - 2z \cos \omega_0 + 1} = \frac{\sin \theta + z^{-1} \sin(\omega_0 - \theta)}{1 - 2z^{-1} \cos \omega_0 + z^{-2}}$	$ z > 1$
$(n+1)a^n u(n)$	$\frac{z^2}{(z-a)^2} = \frac{1}{(1-az^{-1})^2}$	$ z > a $

2.2 Z变换的性质

Z变换有很多重要性质，熟练掌握这些性质对简化分析过程十分重要。

2.2.1 线性性质

若

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

$$Z[y(n)] = Y(z), \quad R_{y-} < |z| < R_{y+}$$

则对于任意常数 a, b , Z变换满足等式

$$Z[ax(n) + by(n)] = aX(z) + bY(z), \quad R_- < |z| < R_+ \quad (2.7)$$

其中 $R_- = \max[R_{x-}, R_{y-}]$, $R_+ = \min[R_{x+}, R_{y+}]$, 即线性组合后的收敛域为各个序列 Z变换的公共收敛域, 如果这些组合中某些零点和极点相互抵消, 则收敛域可能扩大。

2.2.2 序列的移位

若序列 $x(n)$ 的 Z变换为

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

则有

$$Z[x(n-m)] = z^{-m} X(z), \quad R_{x-} < |z| < R_{x+} \quad (2.8)$$

其中 m 为任意整数, m 为正, 则为延迟; m 为负, 则为超前。

序列移位后收敛域是相同的, 只是对单边序列在 $z=0$ 或 $z=\infty$ 处可能有例外, 而对于双边序列, 其收敛域是环状区域, 已不包括 $z=0$ 和 $z=\infty$, 故序列移位后收敛域不会变化。

例如, $Z[f(n)] = 1$, 在 z 平面处处收敛, 但是 $Z[f(n-1)] = z^{-1}$, 在 $z=0$ 处不收敛, 而 $Z[f(n+1)] = z$, 在 $z=\infty$ 处不收敛。

2.2.3 序列的线性加权

若序列 $x(n)$ 的 Z变换为

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

则有

$$Z[nx(n)] = -z \frac{dX(z)}{dz}, \quad R_{x-} < |z| < R_{x+} \quad (2.9)$$

证明: 由于 Z变换在其收敛域中处处解析

$$\frac{dX(z)}{dz} = \frac{d}{dz} \left[\sum_{n=-\infty}^{\infty} x(n) z^{-n} \right] = \sum_{n=-\infty}^{\infty} x(n) \frac{dz^{-n}}{dz}$$

$$\begin{aligned}
 &= -\sum_{n=-\infty}^{\infty} nx(n)z^{-n-1} = -z^{-1} \sum_{n=-\infty}^{\infty} nx(n)z^{-n} \\
 &= -z^{-1}Z[nx(n)]
 \end{aligned}$$

所以

$$Z[nx(n)] = -z \frac{dX(z)}{dz}, \quad R_{x-} < |z| < R_{x+}$$

可见，序列的线性加权等效于 z 域求导再乘以 $(-z)$ ，通过递推可以证明

$$Z[n^m x(n)] = (-z \frac{d}{dz})^m X(z) \quad (2.10)$$

式中

$$\left(-z \frac{d}{dz}\right)^m X(z) = -z \frac{d}{dz} \left\{ -z \frac{d}{dz} \left[L \left(-z \frac{d}{dz} X(z) \right) \right] \right\}$$

2.2.4 乘以指数序列

若

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

则

$$Z[a^n x(n)] = \sum_{n=-\infty}^{\infty} a^n x(n)z^{-n} = \sum_{n=-\infty}^{\infty} x(n)(a^{-1}z)^n = X(a^{-1}z) \quad (2.11)$$

收敛域为 $R_{x-} < |a^{-1}z| < R_{x+}$ 或 $|a| R_{x-} < |z| < |a| R_{x+}$ ， a 可以是复数。

此性质表明如果 $X(z)$ 在 $z = z_1$ 处为极点，则 $X(a^{-1}z)$ 将在 $a^{-1}z = z_1$ ，即 $z = az_1$ 处为极点。如果 a 为正实数，则表示 z 平面缩小或扩大，零极点在 z 平面沿径向移动；若 a 为复数，则在 z 平面上，零极点既有沿径向的伸缩，又有角度旋转，因此此性质是一种 z 域尺度变换。

2.2.5 序列的倒置

若

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

则

$$Z[x(-n)] = X(z^{-1}), \quad 1/R_{x+} < |z| < 1/R_{x-} \quad (2.12)$$

2.2.6 时域卷积定理

若

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

$$Z[h(n)] = H(z), \quad R_{h-} < |z| < R_{h+}$$

则

$$Z[x(n)^* h(n)] = X(z)H(z), \quad \max[R_{x-}, R_{h-}] < |z| < \min[R_{x+}, R_{h+}] \quad (2.13)$$

证明：

$$\begin{aligned} Z[x(n)*h(n)] &= \sum_{n=-\infty}^{\infty} [x(n)*h(n)]z^{-n} = \sum_{n=-\infty}^{\infty} \left\{ \sum_{m=-\infty}^{\infty} [x(m)h(n-m)] \right\} z^{-n} \\ &= \sum_{m=-\infty}^{\infty} x(m) \left\{ \sum_{n=-\infty}^{\infty} h(n-m)z^{-(n-m)} \right\} z^{-m} = \sum_{m=-\infty}^{\infty} x(m)z^{-m}H(z) = X(z)H(z) \end{aligned}$$

在线性移不变系统中，输出 $y(n)$ 等于输入 $x(n)$ 和系统冲激响应 $h(n)$ 的卷积和。利用时域卷积定理，可以将时域卷积运算转换为 z 域的乘积运算，通过求 $X(z)H(z)$ 的 Z 反变换而求出 $y(n)$ ，可以大大简化对系统的分析。

例 2.5 已知 $x(n) = a^n u(n)$ ， $h(n) = b^n u(-n)$ ， $|a| < |b|$ ，求 $y(n) = x(n) * h(n)$ 。

解：由时域卷积定理计算：

$$\begin{aligned} X(z) &= \frac{z}{z-a}, \quad |z| > |a| \\ H(z) &= \frac{-b}{z-b}, \quad |z| < |b| \end{aligned}$$

则

$$Y(z) = X(z)H(z) = \frac{-bz}{(z-a)(z-b)} = \frac{b}{b-a} \frac{z}{z-a} - \frac{b}{b-a} \frac{z}{z-b}, \quad |a| < |z| < |b|$$

因为 $Y(z)$ 的收敛域为环形区域，故 $y(n)$ 是双边序列

$$y(n) = Z^{-1}[Y(z)] = \frac{b}{b-a} a^n u(n) + \frac{b}{b-a} b^n u(-n-1)$$

2.2.7 复卷积定理

若

$$\begin{aligned} Z[x(n)] &= X(z), \quad R_{x-} < |z| < R_{x+} \\ Z[h(n)] &= H(z), \quad R_{h-} < |z| < R_{h+} \end{aligned}$$

则
$$Z[x(n) \cdot h(n)] = \frac{1}{2\pi j} \oint_C X(v)H\left(\frac{z}{v}\right)v^{-1}dv, \quad R_{x-}R_{h-} < |z| < R_{x+}R_{h+} \quad (2.14)$$

其中 C 是哑变量 v 平面上 $X(v)$ 与 $H\left(\frac{z}{v}\right)$ 的公共收敛域内环绕原点的一条反时针旋转的单闭合围线。

2.2.8 初值定理

如果 $x(n)$ 是因果序列，则有

$$x(0) = \lim_{n \rightarrow \infty} X(z) \quad (2.15)$$

证明：因为 $x(n)$ 是因果序列，有

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} = x(0) + x(1)z^{-1} + x(2)z^{-2} + \dots$$

所以

$$\lim_{n \rightarrow \infty} X(z) = x(0)$$

2.2.9 终值定理

如果 $x(n)$ 是因果序列, 其 Z 变换除可以有一个极点在 $z = 1$ 处, 其他极点均在单位圆内, 则有

$$\lim_{n \rightarrow \infty} x(n) = \lim_{z \rightarrow 1} (z-1)X(z) \quad (2.16)$$

证明:

$$(z-1)X(z) = \sum_{n=-\infty}^{\infty} [x(n+1) - x(n)]z^{-n}$$

由于 $x(n)$ 是因果序列, 则

$$(z-1)X(z) = \lim_{n \rightarrow \infty} \sum_{m=-1}^n [x(m+1) - x(m)]z^{-m}$$

又因为 $(z-1)X(z)$ 在单位圆上无极点, 上式两端对 $z = 1$ 取极限, 即得

$$\begin{aligned} \lim_{z \rightarrow 1} (z-1)X(z) &= \lim_{n \rightarrow \infty} \sum_{m=-1}^n [x(m+1) - x(m)] \\ &= \lim_{n \rightarrow \infty} \{ [x(0) - 0] + [x(1) - x(0)] + [x(2) - x(1)] + \dots + [x(n+1) - x(n)] \} \\ &= \lim_{n \rightarrow \infty} [x(n+1)] \\ &= \lim_{n \rightarrow \infty} x(n) \end{aligned}$$

2.2.10 复序列的共轭

若

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

则

$$Z[x^*(n)] = X^*(z^*), \quad R_{x-} < |z| < R_{x+} \quad (2.17)$$

且有

$$Z\{\operatorname{Re}[x(n)]\} = \frac{1}{2}[X(z) + X^*(z^*)], \quad R_{x-} < |z| < R_{x+} \quad (2.18)$$

$$Z\{\operatorname{Im}[x(n)]\} = \frac{1}{2j}[X(z) - X^*(z^*)], \quad R_{x-} < |z| < R_{x+} \quad (2.19)$$

2.2.11 帕斯瓦尔 (Parseval) 定理

若

$$Z[x(n)] = X(z), \quad R_{x-} < |z| < R_{x+}$$

$$Z[h(n)] = H(z), \quad R_{h-} < |z| < R_{h+}$$

且 $R_{x-}R_{h-} < 1 < R_{x+}R_{h+}$, 则

$$\sum_{n=-\infty}^{\infty} x(n)h^*(n) = \frac{1}{2\pi j} \oint_C X(v)H^*\left(\frac{z}{v^*}\right)v^{-1}dv \quad (2.20)$$

其中 C 是在公共收敛域 $\max[R_{x-}, \frac{1}{R_{h+}}] < |v| < \min[R_{x+}, \frac{1}{R_{h-}}]$ 内的一条闭合围线。

2.3 Z反变换

如果 $X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$, $R_{x-} < |z| < R_{x+}$, 则 $X(z)$ 的反变换为

$$x(n) = \frac{1}{2\pi j} \oint_C X(z)z^{n-1}dz \quad (2.21)$$

积分路径 C 是在收敛域内, 环绕原点的一条反时针旋转的闭合单围线, 如图 2.3 所示。

证明: 在 $X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$ 的两端同乘 z^{m-1} 得

$$X(z)z^{m-1} = \left[\sum_{n=-\infty}^{\infty} x(n)z^{-n} \right] z^{m-1}$$

在收敛域内任意选一条环绕原点的反时针方向的单围线 C , 沿 C 进行积分

$$\oint_C X(z)z^{m-1}dz = \int_C \left[\sum_{n=-\infty}^{\infty} x(n)z^{-n} \right] z^{m-1}dz$$

交换积分和求和次序, 得

$$\oint_C X(z)z^{m-1}dz = \sum_{n=-\infty}^{\infty} x(n) \int_C z^{m-n-1}dz \quad (2.22)$$

根据复变函数中的柯西定理得

$$\oint_C z^{k-1}dz = \begin{cases} 2\pi j, & k=0 \\ 0, & k \neq 0 \end{cases}$$

所以式 (2.22) 右端只剩下 $m=n$ 的项

$$\oint_C X(z)z^{m-1}dz = 2\pi j x(m)$$

即

$$x(n) = \frac{1}{2\pi j} \oint_C X(z)z^{n-1}dz$$

直接计算围线积分比较困难, 求 Z 反变换通常有三种方法: 围线积分法 (留数法)、部分分式展开法和长除法 (幂级数展开法)。

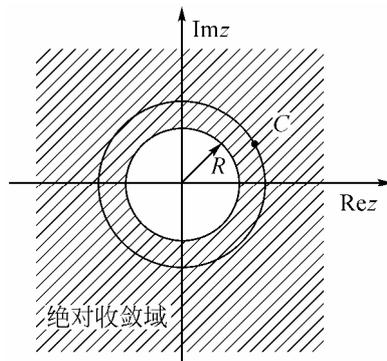


图 2.3 $F(z)$ 的收敛域

续表

序号	序列	傅里叶变换
(9)	$x^*(-n)$	$X^*(e^{j\omega})$
(10)	$\text{Re}[x(n)]$	$X_e(e^{j\omega})$ ($X(e^{j\omega})$ 的共轭对称部分)
(11)	$\text{jIm}[x(n)]$	$X_o(e^{j\omega})$ ($X(e^{j\omega})$ 的共轭反对称部分)
(12)	$x_e(n)$	$\text{Re}[X(e^{j\omega})]$
(13)	$x_o(n)$	$\text{jIm}[X(e^{j\omega})]$
$x(n)$ 为实序列		
(14)	任意实序列	$\begin{cases} X(e^{j\omega}) = X^*(e^{-j\omega}) \\ \text{Re}[X(e^{j\omega})] = \text{Re}[X(e^{-j\omega})] \\ \text{Im}[X(e^{j\omega})] = -\text{Im}[X(e^{-j\omega})] \\ X(e^{j\omega}) = X(e^{-j\omega}) \\ \text{Arg}[X(e^{j\omega})] = -\text{Arg}[X(e^{-j\omega})] \end{cases}$
(15)	$x_e(n)$	$\text{Re}[X(e^{j\omega})]$
(16)	$x_o(n)$	$\text{jIm}[X(e^{j\omega})]$
(17)	$r_{xy}(n)$	$R_{xy}(e^{j\omega}) = X^*(e^{j\omega})Y(e^{j\omega})$

任一序列 $x(n)$ 都能表示成一个共轭对称序列与一个共轭反对称序列之和 (对于实序列就是偶对称序列和奇对称序列之和), 同样, 一个序列 $x(n)$ 的傅里叶变换 $X(e^{j\omega})$ 也可以分解为共轭对称分量 $X_e(e^{j\omega})$ 和共轭反对称分量 $X_o(e^{j\omega})$ 之和, 即

$$X(e^{j\omega}) = X_e(e^{j\omega}) + X_o(e^{j\omega}) \quad (2.46)$$

式中

$$X_e(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) + X^*(e^{-j\omega})]$$

$$X_o(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) - X^*(e^{-j\omega})]$$

$X_e(e^{j\omega})$ 是共轭对称的, 满足 $X_e(e^{j\omega}) = X_e^*(e^{-j\omega})$, $X_o(e^{j\omega})$ 是共轭反对称的, 满足 $X_o(e^{j\omega}) = -X_o^*(e^{-j\omega})$ 。若傅里叶变换 $X(e^{j\omega})$ 是实函数, 且满足共轭对称, 则它为频率的偶函数, 即 $X(e^{j\omega}) = X(e^{-j\omega})$, 与此相似, 若 $X(e^{j\omega})$ 是满足共轭反对称的实函数, 则它为频率的奇函数, 即 $X(e^{j\omega}) = -X(e^{-j\omega})$ 。

序列的傅里叶变换有一些很有用的对称性质, 所谓的对称性质有两方面的含义, 其一是序列或其变换本身所具有的对偶性, 如表 2.3 中的性质 (10)、性质 (11)、性质 (14) 所形成的傅里叶变换本身的对称性; 其二是指频域和时域的对偶关系, 即变换及其反变换之间的对偶关系, 如表 2.3 中的性质 (7) 对应性质 (9), 性质 (10) 对应于性质 (12), 性质 (11) 对应于性质 (13)。

2.6 离散系统的系统函数、系统的频率响应

2.6.1 系统函数的定义

一个线性移不变离散系统可以用它的单位抽样响应 $h(n)$ 来表示其输入输出关系, 即

$$y(n) = x(n) * h(n)$$

对等式两边做 Z 变换

$$Y(z) = X(z)H(z)$$

则

$$H(z) = \frac{Y(z)}{X(z)} \quad (2.47)$$

将 $H(z)$ 定义为线性移不变系统的系统函数, 是单位抽样响应 $h(n)$ 的 Z 变换, 即

$$H(z) = Z[h(n)] = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \quad (2.48)$$

2.6.2 因果稳定系统

前一章已经说过, 一个线性移不变系统稳定的充要条件是 $h(n)$ 必须满足绝对可和条件, 即

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

而 Z 变换的收敛域由满足 $\sum_{n=-\infty}^{\infty} |h(n)z^{-n}| < \infty$ 的那些 z 值确定, 所以如果系统函数的收敛域包含单位圆 $|z| = 1$, 则系统是稳定的。

因果系统的单位抽样响应为因果序列, 其收敛域为 $R_x^- < |z| \leq \infty$ 。

总和上述两点, 一个因果稳定的线性移不变系统的系统函数 $H(z)$ 必须在从单位圆到 ∞ 的整个 z 域内收敛, 即

$$1 \leq |z| \leq \infty$$

也就是说系统函数的全部极点必须在单位圆内。

2.6.3 系统函数和差分方程的关系

一个线性移不变系统可以用差分方程来描述, 其一般形式为

$$y(n) + \sum_{i=1}^N a_i y(n-i) = \sum_{m=0}^M b_m x(n-m) \quad (2.49)$$

若系统的起始状态为零, 直接对上式取 Z 变换 (利用移位特性)

$$Y(z) + \sum_{i=1}^N a_i z^{-i} Y(z) = \sum_{m=0}^M b_m z^{-m} X(z) \quad (2.50)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^M b_m z^{-m}}{1 + \sum_{i=1}^N a_i z^{-i}} \quad (2.51)$$

式 (2.51) 是两个多项式之比, 分别对分子、分母进行因式分解, 可得

$$H(z) = K \frac{\prod_{m=1}^M (1 - c_m z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad (2.52)$$

式中, $z = c_m$ 是 $H(z)$ 的零点, $z = d_k$ 是 $H(z)$ 的极点, 是由差分方程的系数 a_k 和 b_k 决定, 除了比例常数 K , 系统函数完全由它的零点和极点来确定。

式 (2.51) 式 (2.52) 没有给出给定 $H(z)$ 的收敛域, 因而可以代表不同的系统, 这和前面讲的差分方程并不惟一确定一个线性移不变系统的单位抽样响应是一致的。因而要根据 $H(z)$ 惟一确定 $h(n)$, 必须同时确定系统的收敛域。例如对于稳定系统, 其收敛域必须包含单位圆。

例 2.7 已知一线性移不变的因果系统差分方程为 $y(n-1) - \frac{5}{2}y(n) + y(n+1) = x(n)$, 求系统的单位抽样响应 $h(n)$ 并判断, 该系统是否稳定。

解:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{z^{-1} - \frac{5}{2} + z} = \frac{z}{(z-2)(z-\frac{1}{2})} = \frac{2}{3} \left[\frac{z}{z-2} - \frac{z}{z-\frac{1}{2}} \right]$$

由题意知, 系统是因果系统, 因此 $h(n)$ 为因果序列, $H(z)$ 的收敛域为圆外部区域, 即

$$|z| > 2$$

所以

$$h(n) = \frac{2}{3} (2^n - 2^{-n}) u(n)$$

因为系统是因果的, 收敛域为 $|z| > 2$, 不包含单位圆 $|z| = 1$, 因此系统是不稳定的。

2.6.4 系统的频率响应

设系统的输入序列是频率为 ω 的复指数序列 $x(n) = e^{j\omega n}$, $-\infty < n < \infty$ 线性移不变系统的单位抽样响应为 $h(n)$, 利用卷积和, 得到输出为

$$y(n) = h(n) * x(n) = \sum_{m=-\infty}^{\infty} h(m) e^{j\omega(n-m)} = e^{j\omega n} \sum_{m=-\infty}^{\infty} h(m) e^{-j\omega m} = e^{j\omega n} H(e^{j\omega}) \quad (2.53)$$

其中

$$H(e^{j\omega}) = \sum_{m=-\infty}^{\infty} h(m)e^{-j\omega m}$$

由此可知, $H(e^{j\omega})$ 是 $h(n)$ 的傅里叶变换, 称为系统的频率响应, 描述的是复指数序列经过线性移不变系统后, 复振幅 (包括幅度和相位) 的变化。 $h(n)$ 绝对可和, 则系统稳定, 就意味着系统频率响应 $H(e^{j\omega})$ 存在且连续。

系统的频率响应 $H(e^{j\omega})$ 正是系统函数 $H(z)$ 在单位圆上的值, 即

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} \quad (2.54)$$

2.7 MATLAB 在 Z 变换、傅里叶变换中的应用

2.7.1 MATLAB 几个信号处理工具箱函数

1. residuez: 求极点留数分解

前面已经介绍, 离散系统的输出

$$Y(z) = X(z)H(z) = \frac{B(z)}{A(z)}$$

此式必然是 z 的有理分式

$$\begin{aligned} Y(z) &= \frac{B(z)}{A(z)} = \frac{B(1) + B(2)z^{-1} + \text{L} + B(N)z^{-(M-1)} + B(N+1)z^{-M}}{A(1) + A(2)z^{-1} + \text{L} + A(N)z^{-(N-1)} + A(N+1)z^{-N}} \\ &= \frac{r(1)}{1 - p(1)z^{-1}} + \frac{r(2)}{1 - p(2)z^{-1}} + \text{L} + \frac{r(N)}{1 - p(N)z^{-1}} + k(1) + k(2)z^{-1} + \text{L} \end{aligned}$$

从而得出其时域信号为

$$y(n) = r(1)p(1)^n u(n) + r(2)p(2)^n u(n) + \text{L} + r(N)p(N)^n u(n) + k(1)\delta(n) + k(2)\delta(n-1) + \text{L}$$

其中 k 是当 $M \geq N$ 时的直接项, 即有限序列, 而其余的都是无限序列。其中的 r 、 p 、 k 向量可以由下列语句求得。

```
[r,p,k]=residuez(B,A)
```

2. impz: 求 $H(z)$ 的 Z 反变换 $h(n)$

```
[h,T]=impz(B,A,N)
```

h 为存放 $h(n)$ 的列向量, 时间变量 N 存放在列向量 T 中, 当 N 为标量时, 表示 $T = [0 : N-1]'$, 计算 $h(n)$, $n = 0, 1, 2, \text{L}, N-1$; 当 N 为向量时, $T = N$, 仅计算 N 指定的整数点上的 $h(n)$ 。

3. freqz: 求数字滤波器 $H(z)$ 的频率响应函数

```
H=freqz[B,A,w]
```

计算由向量 w 指定的数字频率点上数字滤波器 $H(z)$ 的频率响应, 结果存在 H 向量中。

```
[H,w]=freqz[B,A,M]
```

计算出 M 个频率点上的频率响应, 存放在 H 向量中, M 个频率存放在 w 向量中, `freqz` 函数自动将这 M 个频率点均匀设置在频率范围 $[0, \pi]$ 之间。若缺省 w 和 M 时, 函数自动选取 512 个频率点计算。不带输出向量的 `freqz` 函数将自动绘制幅频和相频曲线。

4. `zplane`: 绘制 $H(z)$ 的零极点图

```
zplane(z,p)
```

绘制出列向量 z 中的零点 (以符号 “o” 表示) 和列向量 p 中的极点 (以符号 “×” 表示) 以及参考单位圆, 并在多阶零点和极点的右上角标出其阶数。如果 z 和 p 为矩阵, 则会以不同颜色绘出 z 和 p 各列中的零点和极点。

```
zplane(B,A)
```

绘制出系统函数 $H(z)$ 的零极点图。

2.7.2 MATLAB 函数应用

例 2.8 设 $X_1(z) = z + 2 + 3z^{-1}$, $X_2(z) = 2z^2 + 4z + 3 + 5z^{-1}$, 求 $Y(z) = X_1(z) X_2(z)$ 。

解: 程序清单如下。

```
x1=[1,2,3];n1=-1:1;
x2=[2,4,3,5];n1=-2:1;
[y,n]=conv_m(x1,n1,x2,n2);
```

程序运行结果:

```
x3=
     2     8    17    23    19    15
n3=
    -3    -2    -1     0     1     2
```

因此 $Y(z) = 2z^3 + 8z^2 + 17z + 23 + 19z^{-1} + 15z^{-2}$ 。

例 2.9 计算 $X(z) = ((1 - 0.9z^{-1})^2(1 + 0.9z^{-1}))^{-1}$, $|z| > 0.9$ 的 Z 反变换。

解: 程序清单如下。

```
b=1;a=poly([0.9 0.9 -0.9]);
[r,p,k]=residuez(b,a)
```

程序运行结果:

```
r =
    0.2500
```

```

0.5000
0.2500
p =
0.9000
0.9000
-0.9000
k =
[]

```

因此得到

$$X(z) = \frac{0.25}{1-0.9z^{-1}} + \frac{0.5}{(1-0.9z^{-1})^2} + \frac{0.25}{1+0.9z^{-1}}, \quad |z| > 0.9$$

相应地

$$x(n) = 0.25(0.9)^n u(n) + \frac{5}{9}(n+1)(0.9)^{n+1} u(n+1) + 0.25(-0.9)^n u(n)$$

例 2.10 求一因果线性移不变系统 $y(n] = 0.81y[n-2] + x[n] - x[n-2]$ 的单位抽样响应 $h(n)$, 单位阶跃响应 $g(n)$, 并绘出 $H(e^{j\omega})$ 的幅频和相频特性。

解：程序清单如下。

```

b=[1,0,-1];a=[1,0,-0.81];
figure(1)
subplot(2,1,1);dimpulse(b,a,50);ylabel('h(n)');
subplot(2,1,2);dstep(b,a,50);ylabel('g(n)');
figure(2)
w=[0:1:500]*pi/500;
freqz(b,a,w)

```

程序运行结果如图 2.4 所示。

例 2.11 梳状滤波器零极点和幅频特性。

梳状滤波器系统函数有如下两种类型。

FIR 型： $H_1(z) = 1 - z^{-N}$

IIR 型： $H_2(z) = \frac{1 - z^{-N}}{1 - a^N z^{-N}}$

分别令 $N = 8$, $a = 0.8, 0.9, 0.98$, 计算并图示 $H_1(z)$ 和 $H_2(z)$ 的零、极点及幅频特性 , 说明极点位置的影响。

解：程序清单如下。

```

b=[1,0,0,0,0,0,0,0,-1];           %H1(z)和H2(z)的分子多项式系数向量
a0=1;                               %H1(z)分母多项式系数向量
a1=[1,0,0,0,0,0,0,0,-(0.8)^8];     %H2(z)分母多项式系数向量(a=0.8)
a2=[1,0,0,0,0,0,0,0,-(0.9)^8];     %H2(z)分母多项式系数向量(a=0.9)
a3=[1,0,0,0,0,0,0,0,-(0.98)^8];    %H2(z)分母多项式系数向量(a=0.98)

```

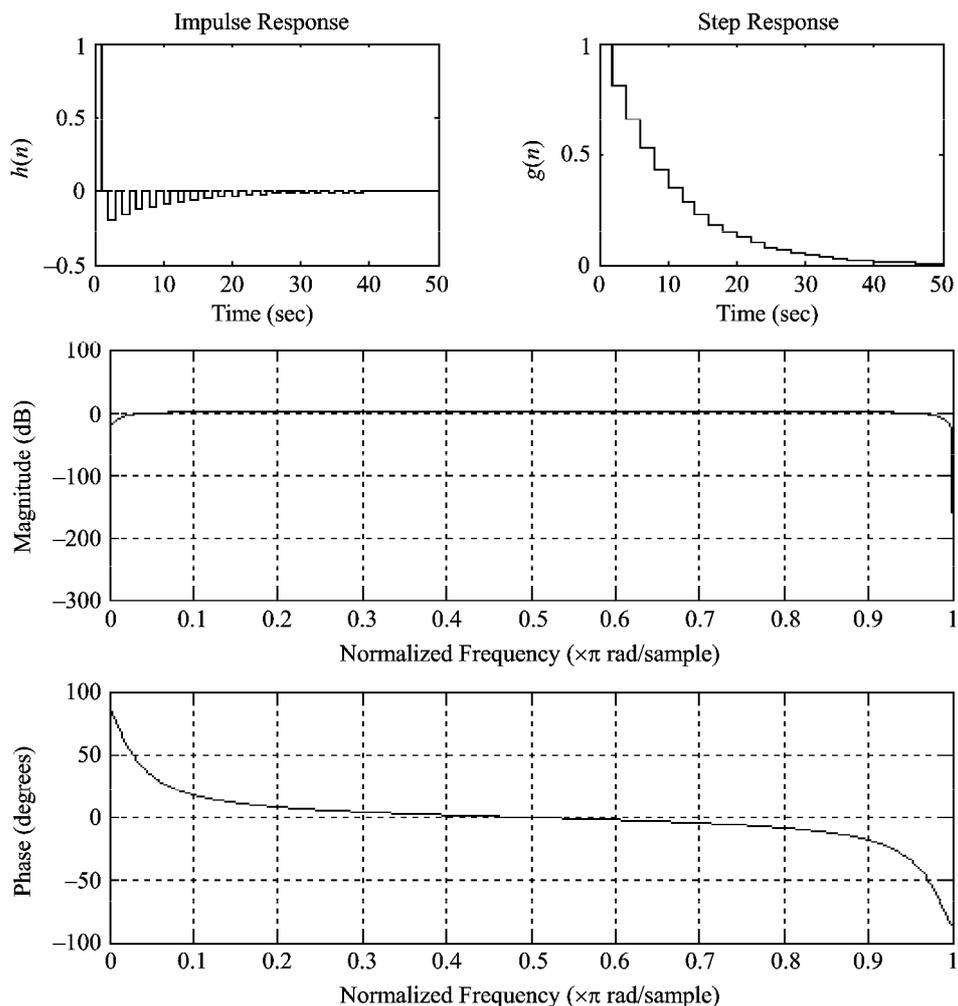


图 2.4 系统的抽样响应、阶跃响应及幅频、相频曲线

```

[H,w]=freqz(b,a0); %H1(z)的频响函数
[H1,w1]=freqz(b,a1);
[H2,w2]=freqz(b,a2);
[H3,w3]=freqz(b,a3);
subplot(4,2,1);zplane(b,a0); xlabel('实部');ylabel('虚部');title('FIR 梳状
滤波器零点图')
subplot(4,2,2);zplane(b,a1);xlabel('实部');ylabel('虚部');
title('IIR 梳状滤波器零极点图 a=0.8')
subplot(4,2,3);plot(w/pi,abs(H));title('FIR 梳状滤波器幅频响应曲线')
subplot(4,2,4); plot(w/pi,abs(H1));title('IIR 梳状滤波器幅频响应曲线 a=0.8')
...(作图部分的程序省略,可参考前面的程序)

```

程序运行结果如图 2.5 所示。

由图 2.5 可以看出,阶数相同的情况下,IIR 滤波器具有更平坦的通带特性和更窄的过渡带,极点距离单位圆越近,这个特性就越明显。

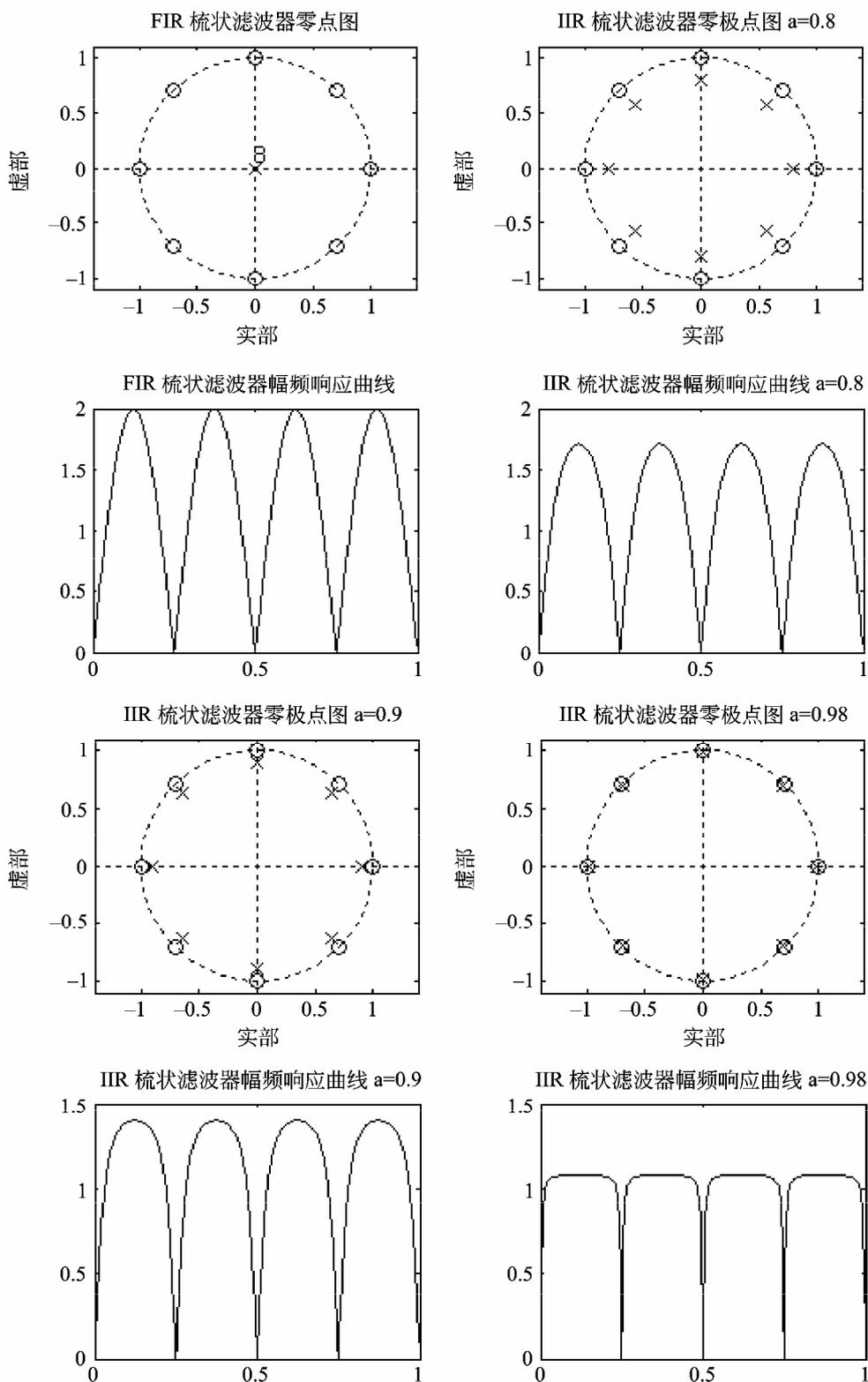


图 2.5 八阶梳状滤波器零、极点图及幅频响应曲线

第 3 章 离散傅里叶变换

前面讨论了序列的傅里叶变换和 Z 变换，其中序列的傅里叶变换在频域是频率的连续函数。有限长序列在数字信号处理中是很重要的一种序列，可以导出反映了“有限长”特点的是离散傅里叶变换 (DFT)，DFT 本身也是一个序列，而不是频率的连续函数，不仅在理论上有着重要的意义，而且因为存在着快速算法——快速傅里叶变换 (FFT)，在实现各种数字信号处理的算法中还起着十分重要的作用。

有限长序列的离散傅里叶变换 (DFT) 和周期序列的离散傅里叶级数 (DFS) 本质上是一样的，为了更好地理解 DFT，需要先讨论 DFS。

3.1 傅里叶变换的几种形式及应用

由于傅里叶变换就是以时间为自变量的“信号”与以频率为自变量的“频谱”函数之间的一种变换关系，当自变量“时间”和“频率”取连续值或离散值时，就形成不同的形式的傅里叶变换对。

3.1.1 非周期的连续时间、连续频率——傅里叶变换

非周期连续时间信号 $x(t)$ 和它的频谱密度函数 $X(j\Omega)$ 构成的傅里叶变换对为正变换

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt \quad (3.1)$$

反变换

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t} d\Omega \quad (3.2)$$

为了说明其性质，以连续时间矩形脉冲为例，它的时间函数曲线和由傅里叶变换得到的幅度频谱密度如图 3.1 所示。可以看出，时域连续函数造成频谱的非周期，而时域的非周期性造成频域是连续的频谱密度函数。

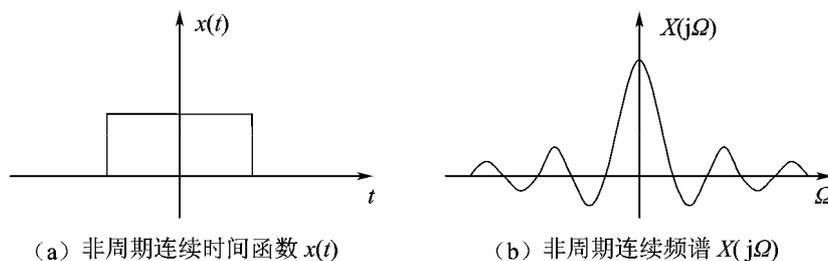


图 3.1 非周期连续时间函数及其傅里叶变换

3.1.2 周期的连续时间、离散频率——傅里叶级数

周期为 T_0 的连续时间信号 $x(t)$ 的傅里叶级数展开的系数为 $X(jk\Omega_0)$ ，构成的傅里叶变换对为

正变换

$$X(jk\Omega_0) = \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} x(t)e^{-jk\Omega_0 t} dt \quad (3.3)$$

反变换

$$x(t) = \sum_{k=-\infty}^{\infty} X(jk\Omega_0)e^{jk\Omega_0 t} \quad (3.4)$$

式 (3.3) 中 $X(jk\Omega_0)$ 是以角频率 Ω_0 为间隔的离散函数形成频域的离散频谱， Ω_0 与时间信号的周期之间的关系为 $\Omega_0 = 2\pi F = 2\pi/T_0$ 。傅里叶级数展开将连续时间周期函数分解为无穷多个角频率为 Ω_0 整数倍的谐波， k 为各次谐波序号。

以连续时间周期矩形脉冲为例，它的时间函数曲线和傅里叶级数展开系数构成的频谱如图 3.2 所示。结果表明，时域连续函数在频域形成非周期的频谱，而时域的周期性对应于频域的离散性。

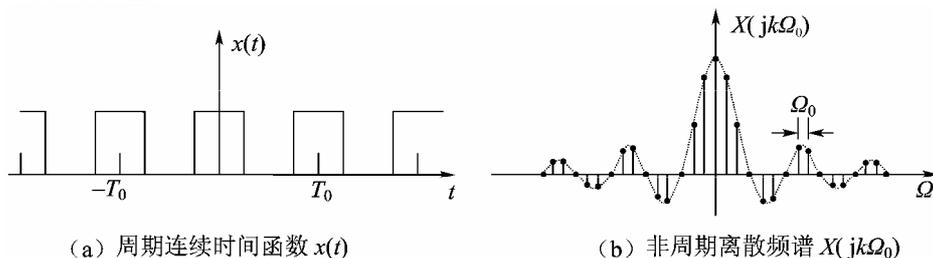


图 3.2 周期连续时间函数及其傅里叶变换

3.1.3 非周期的离散时间、连续频率——序列的傅里叶变换

非周期离散时间信号的傅里叶变换就是前面所说的序列的傅里叶变换，其变换对为正变换

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (3.5)$$

反变换

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n} d\omega \quad (3.6)$$

式中， ω 是数字频率。

如果序列 $x(n)$ 是模拟信号 $x(t)$ 经过抽样得到，抽样时间间隔为 T_s ，抽样频率为 $f_s = 1/T_s$ ，抽样角频率为 $\Omega_s = 2\pi/T_s$ ，由于数字频率 ω 与模拟角频率 Ω 之间的关系为 $\omega = \Omega T_s$ ，因此抽样数字频率 $\omega_s = \Omega_s T_s = 2\pi$ ，则上面的变换对也可写成（代入 $x(n) = x(nT)$ ， $\omega = \Omega T$ ）

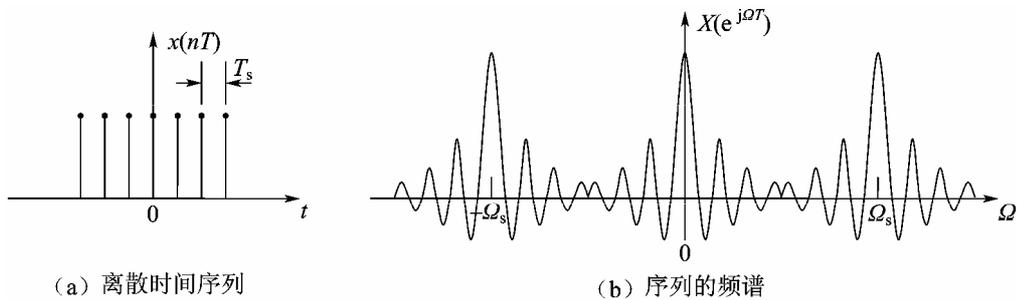
正变换

$$X(e^{j\Omega T}) = \sum_{n=-\infty}^{\infty} x(nT)e^{-jn\Omega T} \quad (3.7)$$

反变换

$$x(nT) = \frac{1}{\Omega_s} \int_{-\frac{\Omega_s}{2}}^{\frac{\Omega_s}{2}} X(e^{j\Omega T}) e^{jn\Omega T} d\Omega \quad (3.8)$$

仍以连续时间矩形脉冲为例。图 3.3 (a) 中的信号是对图 3.1 (a) 中的信号以 T_s 为周期进行抽样得到，得到的频谱如图 3.3 (b) 所示，显然它是图 3.1 (b) 以 Ω_s 为周期进行延拓得到。结果表明，时域的离散造成频域的周期延拓，而时域的非周期性对应与频域的连续性。



错误！

图 3.3 离散时间序列及其傅里叶变换

3.1.4 离散时间、离散频率——离散傅里叶变换

由于数字信号处理是希望在计算机上实现各种运算和变换，其所涉及的变量和运算都是离散的，而前面所讨论的三种傅里叶变换对中，时域或频域中至少有一个域是连续的，所以都不可能在计算机上进行运算和实现。因此，对于数字信号处理，应该找到在时域和频域都是离散的傅里叶变换，即离散傅里叶变换。

前面的讨论已经得出结论：时域的周期性导致频域的离散性，时域的非周期性对应于频域的连续函数形式。那么对于时域和频域都是离散的离散傅里叶变换，应该形成时域和频域都具有周期性的函数。离散傅里叶变换的引入和性质将在后面讨论，这里仅就离散性和周期性的对应关系说明离散傅里叶变换的基本特征。

假如序列 $x(n)$ 是模拟信号 $x(t)$ 经过抽样得到，抽样时间间隔为 T_s ，则频率函数的周期为 $\Omega_s = 2\pi/T_s$ ；如果频率函数也是离散的，其抽样间隔为 Ω_0 ，则时间函数的周期为 $T_0 = 2\pi/\Omega_0$ 。当时间函数序列一个周期内的抽样点数为 N 时，有

$$N = \frac{T_0}{T_s} = \frac{\Omega_s}{\Omega_0} \quad (3.9)$$

上式表明在频域中频谱函数的一个周期内的抽样点数也为 N ，即离散傅里叶变换的时

间序列和频率序列的周期都是 N ，可以得到表示于一个周期内的常用的离散傅里叶变换对为正变换

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{N} nk} \quad (3.10)$$

反变换

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi}{N} nk} \quad (3.11)$$

对图 3.2 (a) 中的周期矩形脉冲信号以 T_s 为周期进行抽样后，它的离散时间函数序列如图 3.4 (a) 所示，由傅里叶变换得到的频谱如图 3.4 (b) 所示，显然这是图 3.2 (b) 以 Ω_s 为周期的延拓。结果表明，时域和频域都是离散的和周期的，一个域的离散必然导致另一个域的周期延拓。

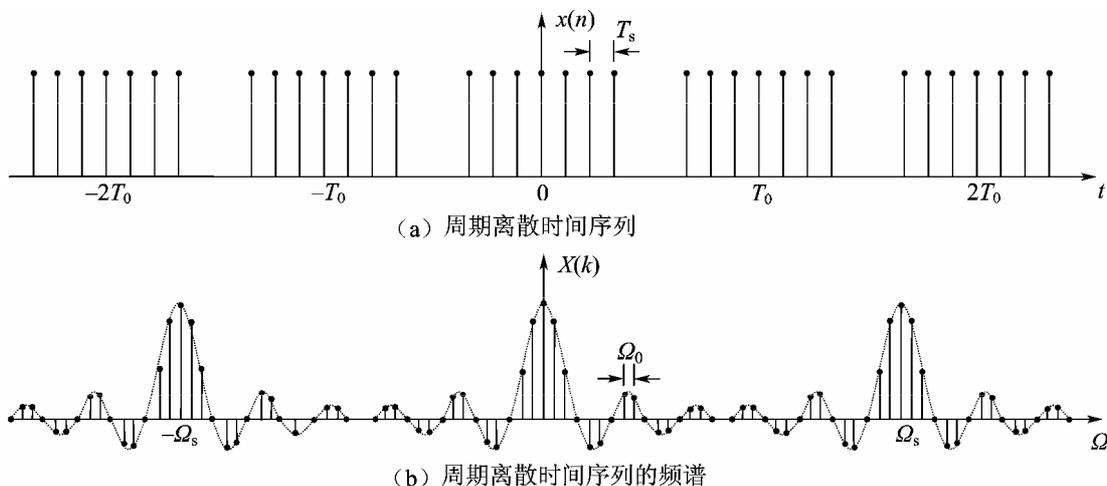


图 3.4 周期离散时间序列及其傅里叶变换

表 3.1 对四种傅里叶变换形式的离散性和周期性的对应关系做了简单总结。

表 3.1 四种傅里叶变换形式的归纳

时 域	频 域
连续性和非周期性	非周期性和连续性
连续性和周期性 (T_0)	非周期性和离散性 ($\Omega_0 = 2\pi T_0$)
离散性 (T_s) 和非周期性	周期性 ($\Omega_s = 2\pi T_s$) 和连续性
离散性 (T_s) 和周期性 (T_0)	周期性 ($\Omega_s = 2\pi T_s$) 和离散性 ($\Omega_0 = 2\pi T_0$)

3.2 周期序列的离散傅里叶级数

3.2.1 周期序列

一个周期为 N 的周期序列 $x(n)$ ，对于所有 n 满足

$$x(n) = x(n + kN), \quad k \text{ 为整数}$$

式中 N 为正整数。周期序列 $x(n)$ 在整个 $-\infty \sim \infty$ 范围内周而复始永不衰减, 但仅有 N 个样本值是独立的, 取其中一个周期内的 N 个样本值足以表征整个序列的特征, 因此可以定义 $n = 0$ 到 $N-1$ 的周期区间为 $x(n)$ 的主值区间, 主值区间内的 N 个样本值组成的有限长序列称为 $x(n)$ 的主值序列, 即

$$x(n) = x(n)R_N(n)$$

这一过程称为取主值序列, 因此对于一个有限长序列

$$x(n) = \begin{cases} x(n), & 0 \leq n \leq N-1 \\ 0, & n \text{ 为其他值} \end{cases}$$

如将其以 N 为周期进行周期性延拓, 则可得

$$x(n) = \sum_{r=-\infty}^{\infty} x(n+rN) = x((n))_N = x(n[\text{mod } N])$$

式中, $((n))_N$ 和 $n[\text{mod } N]$ 均表示模 N 运算, 即取余数运算。

由于周期序列不是绝对可和的, 无论 z 取任何值, 其 Z 变换都是不收敛的, 即

$$\sum_{n=-\infty}^{\infty} |x(n)| |z^{-n}| \rightarrow \infty$$

因此周期序列不能用 Z 变换法或傅里叶变换来进行讨论。

3.2.2 离散傅里叶级数 (DFS)

连续时间周期信号可以展开为傅里叶级数, 同样, 离散时间周期序列也可以用离散傅里叶级数来表示, 如果时间序列的周期为 N , 则离散傅里叶级数表示为周期也是 N 的复指数序列 (代表正弦型序列)。

由于 $x(n)$ 是周期为 N 的周期序列, 可以展开为傅里叶级数为

$$x(n) = \sum_{k=-\infty}^{\infty} a_k e^{j\omega_k n} = \sum_{k=-\infty}^{\infty} a_k e^{jk\omega_0 n} = \sum_{k=-\infty}^{\infty} a_k e^{j\frac{2\pi}{N}kn} \quad (3.12)$$

式中, k 为任意整数; $\omega_k = k\omega_0$ 为 k 次谐波的角频率; $\omega_0 = 2\pi/N$ 为基频角频率; a_k 为 k 次谐波的幅值。

考虑到 $e^{j\frac{2\pi}{N}kn}$ 也是一个周期为 N 的周期序列, 式 (3.12) 可改写为

$$x(n) = \sum_{k=0}^{N-1} a_k e^{j\frac{2\pi}{N}kn} \quad (3.13)$$

将式 (3.12) 两边乘以 $e^{-j\frac{2\pi}{N}kn}$, 并对 n 在一个周期内求和, 有

$$\begin{aligned} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} &= \sum_{n=0}^{N-1} \left[\sum_{k=0}^{N-1} a_k e^{j\frac{2\pi}{N}kn} \right] e^{-j\frac{2\pi}{N}kn} \\ &= \sum_{k=0}^{N-1} a_k \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k-r)n} \end{aligned} \quad (3.14)$$

因为

$$\sum_{k=0}^{N-1} e^{j\frac{2\pi}{N}(k-r)n} = \begin{cases} N, & k-r = mN \\ 0, & k-r \neq mN \end{cases} \quad (3.15)$$

所以式(3.14)变为

$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \quad (3.16)$$

显然 a_k 也是一个周期为 N 的周期序列, 即 $a_k = a_{k+mN}$ 。令 $X(k) = Na_k$, 并代入上式, 得到

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \quad (3.17)$$

$X(k)$ 也是一个以 N 为周期的周期序列, 称为周期序列 $x(n)$ 的离散傅里叶级数系数。

将 $a_k = X(k)/N$ 代入式(3.13), 得

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \quad (3.18)$$

式(3.17)和式(3.18)就构成离散傅里叶级数变换对, 并分别称为离散傅里叶级数的正变换和反变换。

为方便起见, 通常令 $W_N = e^{-j\frac{2\pi}{N}}$, 则 DFS 变换对可写成

$$\text{正变换} \quad X(k) = \text{DFS}[x(n)] = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad (3.19)$$

$$\text{反变换} \quad x(n) = \text{IDFS}[X(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \quad (3.20)$$

其中 $\text{DFS}[\cdot]$ 表示离散傅里叶级数正变换, $\text{IDFS}[\cdot]$ 表示离散傅里叶级数反变换。

离散傅里叶级数表明 $X(k)$ 是以 N 为周期的周期序列, 其基波成分为 $e^{j\frac{2\pi}{N}n}$, k 次谐波成分为 $e^{j\frac{2\pi}{N}nk}$, $X(k)$ 为 DFS 的 k 次谐波分量的复系数。由于 $X(k)$ 的周期性, 当已知 $0 \sim N-1$ 次谐波成分后, 就可以确定其余的谐波分量, 因此, 无论时域还是频域中都只有 N 个序列值是独立的。

3.2.4 离散傅里叶级数的性质

假定 $x_1(n)$ 和 $x_2(n)$ 是周期均为 N 的两个离散周期序列, 它们的 DFS 为

$$X_1(k) = \text{DFS}[x_1(n)]$$

$$X_2(k) = \text{DFS}[x_2(n)]$$

1. 线性特性

$$\text{DFS}[ax_1(n) + bx_2(n)] = aX_1(k) + bX_2(k) \quad (3.21)$$

式中 a, b 为任意常数, 可见由两个离散周期序列 $x_1(n)$ 和 $x_2(n)$ 线性组合成的新周期序列

$a\varphi(n) + b\psi(n)$ 的 DFS 也是周期为 N 的离散周期序列。

2. 移位特性

时域移位

$$\text{DFS}[\varphi(n+m)] = W_N^{-mk} \tilde{\varphi}(k) \quad (3.22)$$

频域移位

$$\text{IDFS}[\tilde{\varphi}(k+l)] = W_N^{ln} \varphi(n) \quad (3.23)$$

如果 $m, l \geq N$, 那么

$$\text{DFS}[\varphi(n+m)] = W_N^{-m'k} \tilde{\varphi}(k), \quad m' = m(\text{mod } N) \quad (3.24)$$

$$\text{IDFS}[\tilde{\varphi}(k+l)] = W_N^{l'n} \varphi(n), \quad l' = l(\text{mod } N) \quad (3.25)$$

上述的两个特性都可以利用 DFS 的定义证明, 这里不再具体证明。

3. 时域卷积特性

两个周期都为 N 的周期序列 $\varphi(n)$ 和 $\psi(n)$, 它们卷积的结果 $\varphi\psi(n)$ 也是周期为 N 的周期序列, 即

$$\varphi\psi(n) = \sum_{m=0}^{N-1} \varphi(m)\psi(n-m) \quad (3.26)$$

这种卷积只限于一个周期内, 即 m 的取值为 $0 \sim N-1$, 因此称为周期卷积, 表示所做的是 N 点圆周卷积和。图 3.5 说明了两个周期序列 (周期 $N=6$) 的周期卷积形成过程。

周期卷积与 DFS 的关系如下:

设

$$\tilde{\varphi}_1(k) = \text{DFS}[\varphi(n)]$$

$$\tilde{\varphi}_2(k) = \text{DFS}[\psi(n)]$$

$$\tilde{\varphi}\psi(k) = \text{DFS}[\varphi\psi(n)]$$

若

$$\varphi\psi(n) = \sum_{m=0}^{N-1} \varphi(m)\psi(n-m)$$

则有

$$\tilde{\varphi}\psi(k) = \tilde{\varphi}_1(k) \cdot \tilde{\varphi}_2(k) \quad (3.27)$$

这就是时域卷积定理。

证明:

$$\begin{aligned}
 \mathcal{Y}(k) &= \text{DFS}[\mathcal{Y}(n)] = \sum_{n=0}^{N-1} \mathcal{Y}(n) W_N^{nk} \\
 &= \sum_{n=0}^{N-1} \left[\sum_{m=0}^{N-1} \mathcal{X}_1(m) \mathcal{X}_2(n-m) \right] W_N^{nk} \\
 &= \sum_{m=0}^{N-1} \mathcal{X}_1(m) \sum_{n=0}^{N-1} \mathcal{X}_2(n-m) W_N^{(n-m)k} W_N^{mk} \\
 &= \sum_{m=0}^{N-1} \mathcal{X}_1(m) W_N^{mk} \sum_{m'=-m}^{N-1-m} \mathcal{X}_2(m') W_N^{m'k} \\
 &= \mathcal{X}_1^0(k) \mathcal{X}_2^0(k)
 \end{aligned}$$

4. 频域卷积特性

对于时域周期序列的乘积，同样对应于频域的周期卷积。
若

$$\mathcal{Y}(n) = \mathcal{X}_1(n) \mathcal{X}_2(n)$$

则

$$\mathcal{Y}(k) = \text{DFS}[\mathcal{Y}(n)] = \frac{1}{N} \sum_{l=0}^{N-1} \mathcal{X}_1^0(l) \mathcal{X}_2^0(k-l) \quad (3.28)$$

根据 DFS 和 IDFS 变换的对称性不难证明上式。

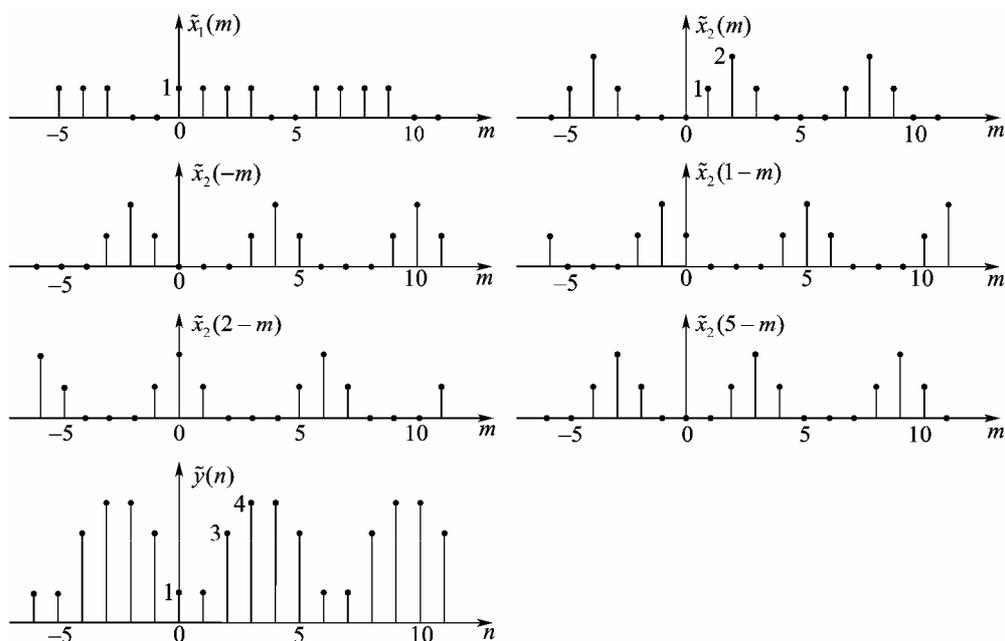


图 3.5 两个周期序列($N=6$)的周期卷积过程

3.3 离散傅里叶变换

离散傅里叶级数 DFS 是周期序列，只有 N 个独立的复值，只要知道一个周期的内容，其他的内容也就知道了。DFS 和 IDFS 表达式都只取 N 点序列值正表明了这一点，也说明了它和有限长序列的本质联系。

由于长度为 N 的有限长序列可以看做是周期为 N 的周期序列的一个周期，因此利用 DFS 计算周期序列的一个周期，就可以得到有限长序列的离散傅里叶变换(DFT)。

设 $x(n)$ 是长度为 N 的有限长序列，可以把它看做是周期为 N 的周期序列 $\mathcal{X}(n)$ 的一个主周期，而将 $\mathcal{X}(n)$ 看做是 $x(n)$ 以 N 为周期进行周期延拓得到，即

$$\begin{aligned} x(n) &= \begin{cases} \mathcal{X}(n), & 0 \leq n \leq N-1 \\ 0, & n \text{ 为其他值} \end{cases} \\ &= \mathcal{X}(n)R_N(n) \end{aligned} \quad (3.29)$$

同理，频域的周期函数 $\mathcal{X}(k)$ 也可以看做是有限长序列 $X(k)$ 的周期延拓，而有限长序列 $X(k)$ 看做是周期序列 $\mathcal{X}(k)$ 的主值序列，即

$$\begin{aligned} \mathcal{X}(k) &= X((k))_N \\ X(k) &= \mathcal{X}(k)R_N(k) \end{aligned}$$

因此有限长序列 $X(k)$ 的离散傅里叶变换为

$$\begin{aligned} X(k) &= \mathcal{X}(k)R_N(k) = \{\text{DFS}[\mathcal{X}(n)]\}R_N(k) \\ &= \left[\sum_{n=0}^{N-1} x((n))_N W_N^{nk} \right] R_N(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad 0 \leq k \leq N-1 \end{aligned} \quad (3.30)$$

同理

$$\begin{aligned} x(n) &= \mathcal{X}(n)R_N(n) = \{\text{IDFS}[\mathcal{X}(k)]\}R_N(n) = \left[\frac{1}{N} \sum_{k=0}^{N-1} X((k))_N W_N^{-nk} \right] R_N(n) \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad 0 \leq n \leq N-1 \end{aligned} \quad (3.31)$$

式 (3.30) 和式 (3.31) 构成了离散傅里叶变换对，即

$$\text{正变换} \quad X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad 0 \leq k \leq N-1$$

$$\text{反变换} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad 0 \leq n \leq N-1$$

离散傅里叶变换 (DFT) 实际上来自于离散傅里叶级数，只不过仅在时域和频域对周期序列 $\mathcal{X}(n)$ 和 $\mathcal{X}(k)$ 各取一个周期而已。 $\mathcal{X}(n)$ 和 $\mathcal{X}(k)$ 都有 N 个独立的复值，故其信息量是相等的。凡是在用到 DFT 的地方，有限长序列都是作为周期序列的一个周期来表示的，都隐含有周期性因素。

3.4 离散傅里叶变换的性质

在以下的讨论中,假定 $x_1(n)$ 和 $x_2(n)$ 都是 N 点的有限长序列,它们的离散傅里叶变换分别为

$$\begin{aligned} X_1(k) &= \text{DFT}[x_1(n)] \\ X_2(k) &= \text{DFT}[x_2(n)] \end{aligned}$$

3.4.1 线性特性

若两个有限长序列 $x_1(n)$ 和 $x_2(n)$ 的线性组合为 $x_3(n) = ax_1(n) + bx_2(n)$, 则有

$$\text{DFT}[ax_1(n) + bx_2(n)] = aX_1(k) + bX_2(k) \quad (3.32)$$

式中 a, b 为任意常数。

说明:

(1) 若 $x_1(n)$ 和 $x_2(n)$ 的长度均为 N , 则 $x_3(n)$ 的长度为 N 。

(2) 若 $x_1(n)$ 和 $x_2(n)$ 的长度不等, $x_1(n)$ 的长度为 N_1 , $x_2(n)$ 的长度为 N_2 , 则 $x_3(n)$ 的长度为 $N = \max[N_1, N_2]$, 离散傅里叶变换的长度必须按 N 来计算。例如, 如果 $N_1 < N_2$, 则取 $N = N_2$, 将 $x_1(n)$ 补上 $N_2 - N_1$ 个零值后变成长度为 N_2 的序列, 然后都做 N_2 点的 DFT, 即

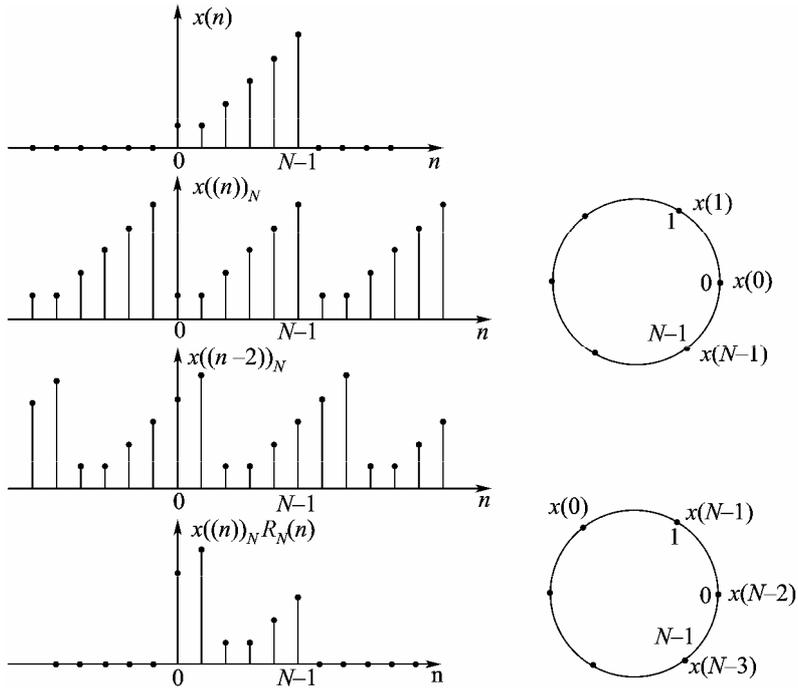
$$\begin{aligned} X_1(k) &= \sum_{n=0}^{N_2-1} x_1(n)W_{N_2}^{nk}, \quad 0 \leq k \leq N_2 - 1 \\ X_2(k) &= \sum_{n=0}^{N_2-1} x_2(n)W_{N_2}^{nk}, \quad 0 \leq k \leq N_2 - 1 \end{aligned}$$

3.4.2 序列的圆周移位

有限长序列 $x(n)$ 的圆周移位是以它的长度 N 为周期, 将其延拓成周期序列 $\mathcal{X}(n)$, 并将周期序列进行移位, 然后取主值区间 ($n = 0$ 到 $N - 1$) 上的序列值。因而一个有限长序列 $x(n)$ 的右圆周移位定义为

$$x((n-m))_N R_N(n) = \mathcal{X}(n-m)R_N(n) \quad (3.33)$$

式中 $x((n-m))_N$ 表示 $x(n)$ 的周期延拓序列 $\mathcal{X}(n)$ 的右移位。而 $x((n-m))_N R_N(n)$ 得到的是周期延拓并移位后的周期序列的主值序列, 因而仍是长度为 N 的有限长序列。移位的过程如图 3.6 所示, 当只观察 $n = 0$ 到 $N-1$ 这一区间时, 某样点值从此区间的一端移出, 而与它

图 3.6 序列的周期移位($N=6$)

相同值的样点值又从这个区间的另一端移进来,因此可以看成 $x(n)$ 排列在一个 N 等分的圆周上,序列 $x(n)$ 的圆周移位就相当于 $x(n)$ 在此圆周上旋转,故称为圆周移位或循环移位。

1. 时域移位定理

设 $X(k)$ 表示周期序列 $x(n)$ 的离散傅里叶级数, $X(k)$ 为有限长序列 $x(n)$ 的离散傅里叶变换,则序列 $x(n)$ 圆周移位后所得到的序列 $x(n-m)$ 的 DFT 为

$$\text{DFT}[x(n-m)] = \text{DFT}[x((n-m))_N R_N(n)] = W_N^{mk} X(k) \quad (3.34)$$

证明:由周期序列的时域移位性质有

$$\text{DFS}[x(n-m)] = W_N^{mk} X(k)$$

由于有限长序列的 DFT 就是周期序列 DFS 在频域中的主值序列,可得序列 $x(n)$ 圆周移位后所得到的序列 $x(n-m)$ 的 DFT 为

$$\text{DFT}[x(n-m)] = \text{DFS}[x(n-m)] R_N(k) = W_N^{mk} X(k) R_N(k)$$

2. 频域移位定理

对于频域,有限长序列 $X(k)$ 也可以看成是分布在一个 N 等分的圆周上,若频域的有限长序列 $X(k)$ 为 $x(n)$ 的傅里叶变换,即

$$X(k) = \text{DFT}[x(n)]$$

则

$$\text{IDFT}[X((k-l))_N R_N(n)] = W_N^{-nl} x(n) \quad (3.35)$$

式 (3.35) 称为频率移位定理,也称为调制定理,此定理说明时域序列的调制等效于

频域的圆周移位。

3.4.3 共轭对称性

在序列的傅里叶变换中讨论了一些对称性质，且定义了共轭对称序列与共轭反对称序列的概念，指出任一序列都可以表示成共轭对称分量和共轭反对称分量之和。

因为有限长序列及它的离散傅里叶变换都隐含有周期性，周期等于序列长度 N ，但是按照前面所讲的共轭对称分量 $x_e(n)$ 和共轭反对称分量 $x_o(n)$ 的定义计算出的都是 $(2N-1)$ 点，因此这里从周期序列的共轭对称分量 $\mathcal{X}_0(n)$ 和共轭反对称分量 $\mathcal{X}_0^*(n)$ 入手，因为它们都是周期性的，周期仍为 N ，取出其主值序列就得到了有限长序列的相应的分量，分别称为圆周共轭对称分量 $x_{ep}(n)$ 和圆周共轭反对称分量 $x_{op}(n)$ ，公式推导如下：

设有限长序列 $x(n)$ 的长度为 N ，以 N 为周期的周期延拓序列为

$$\mathcal{X}(n) = x((n))_N$$

周期序列 $\mathcal{X}(n)$ 的共轭对称分量 $\mathcal{X}_0(n)$ 和共轭反对称分量 $\mathcal{X}_0^*(n)$ 分别为

$$\mathcal{X}_0(n) = \frac{1}{2}[\mathcal{X}(n) + \mathcal{X}_0^*(-n)] = \frac{1}{2}[x((n))_N + x^*((N-n))_N] \quad (3.36)$$

$$\mathcal{X}_0^*(n) = \frac{1}{2}[\mathcal{X}(n) - \mathcal{X}_0^*(-n)] = \frac{1}{2}[x((n))_N - x^*((N-n))_N] \quad (3.37)$$

同样可以证明，它们满足

$$\mathcal{X}_0(n) = \mathcal{X}_0^*(-n) \quad (3.38)$$

$$\mathcal{X}_0^*(n) = -\mathcal{X}_0^*(-n) \quad (3.39)$$

则有限长序列 $x(n)$ 的圆周共轭对称分量 $x_{ep}(n)$ 和圆周共轭反对称分量 $x_{op}(n)$ 分别定义为

$$x_{ep}(n) = \mathcal{X}_0(n)R_N(n) = \frac{1}{2}[x((n))_N + x^*((N-n))_N]R_N(n) \quad (3.40)$$

$$x_{op}(n) = \mathcal{X}_0^*(n)R_N(n) = \frac{1}{2}[x((n))_N - x^*((N-n))_N]R_N(n) \quad (3.41)$$

由于满足 $\mathcal{X}(n) = \mathcal{X}_0(n) + \mathcal{X}_0^*(n)$ ，故

$$\begin{aligned} x(n) &= \mathcal{X}(n)R_N(n) = [\mathcal{X}_0(n) + \mathcal{X}_0^*(n)]R_N(n) \\ &= x_{ep}(n) + x_{op}(n) \end{aligned} \quad (3.42)$$

显然长度为 N 的有限长序列 $x(n)$ 可以分解为圆周共轭对称分量 $x_{ep}(n)$ 和圆周共轭反对称分量 $x_{op}(n)$ 之和， $x_{ep}(n)$ 和 $x_{op}(n)$ 的长度皆为 N 。利用有限长序列与周期序列共轭对称分量和反对称分量的关系式 (3.40) 和式 (3.41)，以及式 (3.42)，可以导出 DFT 的一系列的对称性质。

$$(1) \quad \text{DFT}[x^*(n)] = X^*(-k) = X^*(N-k) \quad (3.43)$$

式中 $x^*(n)$ 表示 $x(n)$ 的共轭复序列。

证明：

$$\text{DFT}[x^*(n)] = \sum_{n=0}^{N-1} x^*(n)W_N^{nk} = \left[\sum_{n=0}^{N-1} x(n)W_N^{-nk} \right]^* = X^*(-k)$$

又因为

$$W_N^{nN} = e^{-j(2\pi/N)nN} = e^{-j2\pi n} = 1$$

所以

$$\text{DFT}[x^*(n)] = \left[\sum_{n=0}^{N-1} x(n)W_N^{(N-k)n} \right]^* = X^*(N-k)$$

$$(2) \quad \text{DFT}[x^*(-n)] = X^*(k) \quad (3.44)$$

证明：

$$\begin{aligned} \text{DFT}[x^*(-n)] &= \sum_{n=0}^{N-1} x^*(-n)W_N^{nk} = \left[\sum_{n=0}^{N-1} x(-n)W_N^{-nk} \right]^* \\ &= \left[\sum_{n=-(N-1)}^0 x(n)W_N^{nk} \right]^* \stackrel{\text{利用DFT的周期性}}{=} \left[\sum_{n=0}^{N-1} x(n)W_N^{nk} \right]^* = X^*(k) \end{aligned}$$

(3) 复序列实部的 DFT 等于序列 DFT 的圆周共轭对称部分，即

$$\text{DFT}\{\text{Re}[x(n)]\} = X_{\text{ep}}(k) = \frac{1}{2}[X(k) + X^*(N-k)] \quad (3.45)$$

证明：

$$\begin{aligned} \text{DFT}\{\text{Re}[x(n)]\} &= \text{DFT}\left\{\frac{1}{2}[x(n) + x^*(n)]\right\} = \frac{1}{2}\{\text{DFT}[x(n)] + \text{DFT}[x^*(n)]\} \\ &= \frac{1}{2}[X(k) + X^*(N-k)] = X_{\text{ep}}(k) \end{aligned}$$

(4) 复序列虚部乘 j 的 DFT 等于序列 DFT 的圆周共轭反对称部分，即

$$\text{DFT}\{j\text{Im}[x(n)]\} = X_{\text{op}}(k) = \frac{1}{2}[X(k) - X^*(N-k)] \quad (3.46)$$

证明：

$$\begin{aligned} \text{DFT}\{j\text{Im}[x(n)]\} &= \text{DFT}\left\{\frac{1}{2}[x(n) - x^*(n)]\right\} = \frac{1}{2}\{\text{DFT}[x(n)] - \text{DFT}[x^*(n)]\} \\ &= \frac{1}{2}[X(k) - X^*(N-k)] = X_{\text{op}}(k) \end{aligned}$$

(5) 若 $x(n)$ 是实序列，则 $X(k)$ 只有圆周共轭对称部分，即满足

$$X(k) = X^*(N-k) \quad (3.47)$$

(6) 若 $x(n)$ 是纯虚数序列，则 $X(k)$ 只有圆周共轭反对称部分，即满足

$$X(k) = -X^*(N-k) \quad (3.48)$$

$$(7) \quad \text{DFT}[x_{\text{ep}}(n)] = \text{Re}[X(k)] \quad (3.49)$$

$$\text{DFT}[x_{\text{op}}(n)] = j\text{Im}[X(k)] \quad (3.50)$$

利用这些共轭对称特性，可以用一次 DFT 运算来计算两个实序列的 DFT，从而减少计算量。

例 3.1 设 $x_1(n)$ 和 $x_2(n)$ 都是实数序列，试求 $X_1(k)$ 和 $X_2(k)$ 。

解：先利用这两个实数序列构成一个复序列，即

$$y(n) = x_1(n) + jx_2(n)$$

$$\text{DFT}[y(n)] = Y(k) = \text{DFT}[x_1(n)] + j\text{DFT}[x_2(n)] = X_1(k) + jX_2(k)$$

又

$$x_1(n) = \operatorname{Re}[y(n)]$$

故

$$X_1(k) = \operatorname{DFT}\{\operatorname{Re}[y(n)]\} = Y_{\text{ep}}(k) = \frac{1}{2}[Y(k) + Y^*(N-k)]$$

同样

$$x_2(n) = \operatorname{Im}[y(n)]$$

故

$$X_2(k) = \frac{1}{j}Y_{\text{op}}(k) = \frac{1}{2j}[Y(k) - Y^*(N-k)]$$

所以用一次 DFT 计算出 $Y(k)$ ，然后用以上公式就可以求出 $X_1(k)$ 和 $X_2(k)$ 。

例 3.2 试利用 DFT 的对称特性求 $\cos \omega_0 n$ 和 $\sin \omega_0 n$ 的 DFT。

解：设

$$x(n) = \cos \omega_0 n + j \sin \omega_0 n = e^{j\omega_0 n}$$

则

$$\operatorname{DFT}[x(n)] = X(k) = \sum_{n=0}^{N-1} e^{j\omega_0 n} W_N^{nk} = \frac{1 - e^{j\omega_0 N} W_N^{Nk}}{1 - e^{j\omega_0} W_N^k} = \frac{1 - e^{j\omega_0 N}}{1 - e^{j\omega_0} W_N^k}$$

因为

$$\cos \omega_0 n = \operatorname{Re}[x(n)]$$

所以

$$\begin{aligned} \operatorname{DFT}[\cos \omega_0 n] &= \operatorname{DFT}\{\operatorname{Re}[x(n)]\} = X_{\text{ep}}(k) = \frac{X(k) + X^*(N-k)}{2} \\ &= \frac{1}{2} \left[\frac{1 - e^{j\omega_0 N}}{1 - e^{j\omega_0} W_N^k} + \frac{1 - e^{-j\omega_0 N}}{1 - e^{-j\omega_0} W_N^k} \right] \\ &= \frac{1 - \cos \omega_0 N - W_N^k \cos \omega_0 + W_N^k \cos(N-1)\omega_0}{1 - 2W_N^k \cos \omega_0 + W_N^{2k}} \end{aligned}$$

因为

$$\sin \omega_0 n = \operatorname{Im}[x(n)]$$

所以

$$\begin{aligned} \operatorname{DFT}[\sin \omega_0 n] &= \operatorname{DFT}\{\operatorname{Im}[x(n)]\} = \frac{1}{j} X_{\text{op}}(k) = \frac{X(k) - X^*(N-k)}{2j} \\ &= \left[\frac{1 - e^{j\omega_0 N}}{1 - e^{j\omega_0} W_N^k} - \frac{1 - e^{-j\omega_0 N}}{1 - e^{-j\omega_0} W_N^k} \right] \frac{1}{2j} \\ &= \frac{W_N^k \sin \omega_0 - \sin \omega_0 N + W_N^k \sin(N-1)\omega_0}{1 - 2W_N^k \cos \omega_0 + W_N^{2k}} \end{aligned}$$

3.4.4 圆周卷积

1. 时域圆周卷积

设 $x_1(n)$ 和 $x_2(n)$ 都是 N 点的有限长序列，它们的离散傅里叶变换分别为

$$X_1(k) = \text{DFT}[x_1(n)]$$

$$X_2(k) = \text{DFT}[x_2(n)]$$

若

$$Y(k) = X_1(k)X_2(k)$$

则

$$\begin{aligned} y(n) = \text{IDFT}[Y(k)] &= \left[\sum_{m=0}^{N-1} x_1(m)x_2((n-m))_N \right] R_N(n) \\ &= \left[\sum_{m=0}^{N-1} x_2(m)x_1((n-m))_N \right] R_N(n) \end{aligned} \quad (3.51)$$

证明：这个卷积和公式相当于周期序列 $\mathcal{X}_1(n)$ 和 $\mathcal{X}_2(n)$ 做周期卷积后再取主值序列。先将 $Y(k)$ 周期延拓，即

$$\mathcal{Y}(k) = \mathcal{X}_1(k)\mathcal{X}_2(k)$$

按照 DFS 的周期卷积和公式

$$\begin{aligned} \mathcal{Y}(n) = \text{IDFT}[\mathcal{Y}(k)] &= \sum_{m=0}^{N-1} \mathcal{X}_1(m)\mathcal{X}_2(n-m) \\ &= \sum_{m=0}^{N-1} x_1((m))_N x_2((n-m))_N \end{aligned}$$

由于 $0 \leq m \leq N-1$ ，为主值区间，故

$$x_1((m))_N = x_1(m)$$

因此

$$y(n) = \mathcal{Y}(n)R_N(n) = \left[\sum_{m=0}^{N-1} x_1(m)x_2((n-m))_N \right] R_N(n)$$

将 $\mathcal{Y}(n)$ 式经过简单的变量换元后，可以得到

$$y(n) = \mathcal{Y}(n)R_N(n) = \left[\sum_{m=0}^{N-1} x_2(m)x_1((n-m))_N \right] R_N(n)$$

此卷积过程如图 3.7 所示。当 $N=6$ 时，可以看出和图 3.5 所示的周期卷积和过程是一致的，只不过这里要取结果的主值序列。公式中的 $x_2((n-m))_N$ 只在 $0 \leq m \leq N-1$ 范围内取值，因而是圆周移位，因此这个卷积和称为圆周卷积和。

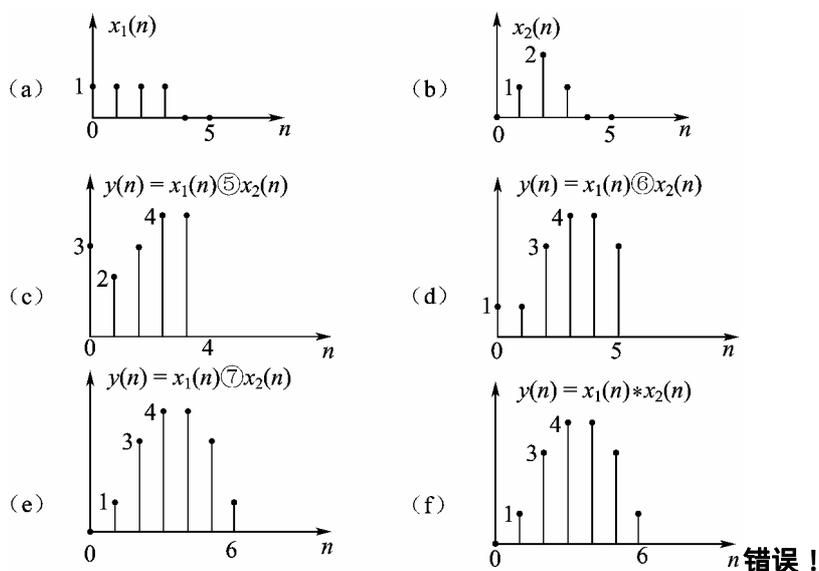


图 3.7 两个有限长序列的圆周卷积和线性卷积

将式(3-51)展开，并利用循环移位特性，可将其写成循环矩阵的形式

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} x_2(0) & x_2(N-1) & x_2(N-2) & \dots & x_2(1) \\ x_2(1) & x_2(0) & x_2(N-1) & \dots & x_2(2) \\ x_2(2) & x_2(1) & x_2(0) & \dots & x_2(3) \\ x_2(3) & x_2(2) & x_2(1) & \dots & x_2(4) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M & M & M & M & M \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_2(N-1) & x_2(N-2) & x_2(N-3) & \dots & x_2(0) \end{bmatrix} \cdot \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \\ \vdots \\ M \\ x_1(N-1) \end{bmatrix}$$

例 3.3 已知两个有限长序列 $x(n)$ 和 $h(n)$ 分别为

$$\begin{array}{rcccccc}
 n = 0 & 1 & 2 & 3 & 4 & 5 \\
 x(n) = & 1 & 4 & 1 & 2 & 0 & 3 \\
 h(n) = & 2 & 3 & 4 & 0 & 1 & 2
 \end{array}$$

试求其圆周卷积 $y(n)$ 。

解：

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \end{bmatrix} = \begin{bmatrix} x(0) & x(5) & x(4) & x(3) & x(2) & x(1) \\ x(1) & x(0) & x(5) & x(4) & x(3) & x(2) \\ x(2) & x(1) & x(0) & x(5) & x(4) & x(3) \\ x(3) & x(2) & x(1) & x(0) & x(5) & x(4) \\ x(4) & x(3) & x(2) & x(1) & x(0) & x(5) \\ x(5) & x(4) & x(3) & x(2) & x(1) & x(0) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ h(3) \\ h(4) \\ h(5) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 3 & 0 & 2 & 1 & 4 \\ 4 & 1 & 3 & 0 & 2 & 1 \\ 1 & 4 & 1 & 3 & 0 & 2 \\ 2 & 1 & 4 & 1 & 3 & 0 \\ 0 & 2 & 1 & 4 & 1 & 3 \\ 3 & 0 & 2 & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 20 \\ 27 \\ 22 \\ 26 \\ 17 \\ 20 \end{bmatrix}$$

2. 频域圆周卷积

利用时域和频域的对称性，可以得到频域圆周卷积定理。

若

$$y(n) = x_1(n)x_2(n)$$

则

$$\begin{aligned} Y(k) = \text{DFT}[y(n)] &= \frac{1}{N} \left[\sum_{l=0}^{N-1} X_1(l)X_2((k-l))_N \right] R_N(k) \\ &= \frac{1}{N} \left[\sum_{l=0}^{N-1} X_2(l)X_1((k-l))_N \right] R_N(k) \end{aligned} \quad (3.52)$$

3. 圆周相关定理

若

$$R_{xy}(k) = X(k)Y^*(k)$$

则

$$\begin{aligned} r_{xy}(m) = \text{IDFT}[R_{xy}(k)] &= \sum_{n=0}^{N-1} x(n)y^*((n-m))_N R_N(m) \\ &= \sum_{n=0}^{N-1} y^*(n)x((n+m))_N R_N(m) \end{aligned} \quad (3.53)$$

其证明类同于时域圆周卷积定理，这里不再详细证明。

4. 用圆周卷积求线性卷积

时域的圆周卷积在频域上相当于两序列的 DFT 的相乘，而 DFT 又有快速算法，因此与线性卷积相比，计算速度可以大大提高。但是一般实际问题都是线性卷积运算，如果信号 $x(n)$ 和单位抽样响应 $h(n)$ 都是有限长序列，那么是否能用圆周卷积的运算来代替线性卷积运算呢？下面就这个问题加以讨论：设 $x_1(n)$ 是 N_1 点的有限长序列， $x_2(n)$ 是 N_2 点的有

限长序列。

(1) $x_1(n)$ 和 $x_2(n)$ 的线性卷积

$$y_l(n) = \sum_{m=-\infty}^{\infty} x_1(m)x_2(n-m) = \sum_{m=0}^{N_1-1} x_1(m)x_2(n-m) \quad (3.54)$$

$x_1(m)$ 的非零区间为 $0 \leq m \leq N_1-1$ ， $x_2(n-m)$ 的非零区间为 $0 \leq n-m \leq N_2-1$ ，将两个不等式相加，得到

$$0 \leq n \leq N_1+N_2-2$$

在上述的区间外，很明显 $y_l(n)=0$ ，因为 $y_l(n)$ 是长度为 (N_1+N_2-1) 的有限长序列，等于参与卷积的两个序列的长度之和减一。

(2) $x_1(n)$ 和 $x_2(n)$ 的圆周卷积

先假设 $x_1(n)$ 和 $x_2(n)$ 进行 L 圆周卷积， $L > \max(N_1, N_2)$ ，再讨论 L 等于何值时，圆周卷积才能代表线性卷积。将两个序列都补零为长度为 L 的序列，即

$$x_1(n) = \begin{cases} x_1(n), & 0 \leq n \leq N_1-1 \\ 0, & N_1 \leq n \leq L-1 \end{cases}$$

$$x_2(n) = \begin{cases} x_2(n), & 0 \leq n \leq N_2-1 \\ 0, & N_2 \leq n \leq L-1 \end{cases}$$

则

$$y(n) = \left[\sum_{m=0}^{L-1} x_1(m)x_2((n-m))_L \right] R_L(n) \quad (3.55)$$

将任一序列（这里采用 $x_2(n)$ ）变成 L 点周期延拓序列，即

$$\mathcal{X}_2(n) = x_2((n))_N = \sum_{r=-\infty}^{\infty} x_2(n+rL) \quad (3.56)$$

把上式代入 $y(n)$ ，得到

$$\begin{aligned} y(n) &= \left[\sum_{m=0}^{L-1} x_1(m)x_2((n-m))_L \right] R_L(n) = \left[\sum_{m=0}^{L-1} x_1(m) \sum_{r=-\infty}^{\infty} x_2(n+rL-m) \right] R_L(n) \\ &= \left[\sum_{r=-\infty}^{\infty} \sum_{m=0}^{L-1} x_1(m)x_2(n+rL-m) \right] R_L(n) = \left[\sum_{r=-\infty}^{\infty} y_l(n+rL) \right] R_L(n) \end{aligned} \quad (3.57)$$

因此 L 点的圆周卷积 $y(n)$ 是线性卷积 $y_l(n)$ 以 L 为周期的周期延拓序列的主值序列。因为 $y_l(n)$ 有 N_1+N_2-1 个非零值，所以延拓周期必须满足 $L \geq N_1+N_2-1$ ，此时各延拓周期才不会交叠。如图 3.7 所示，反映了圆周卷积和线性卷积的关系， $x_1(n)$ 是长度为 $N_1=4$ 的有限长序列， $x_2(n)$ 是长度为 $N_2=4$ 的有限长序列，在图 3.7 (e) 和图 3.7 (f) 中可以看出，当 $L = N_1+N_2-1 = 7$ ，此时圆周卷积和线性卷积的结果相同。

结论：若 $L \geq N_1+N_2-1$ ，则 L 点圆周卷积能代表线性卷积。

3.4.5 帕斯瓦尔 (Parseval) 定理

若 $X_1(k) = \text{DFT}[x(n)]$, 则

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)Y^*(k) \quad (3.58)$$

证明 :

$$\begin{aligned} \sum_{n=0}^{N-1} x(n)y^*(n) &= \sum_{n=0}^{N-1} x(n) \left[\frac{1}{N} \sum_{k=0}^{N-1} Y(k)W_N^{-nk} \right]^* \\ &= \frac{1}{N} \sum_{k=0}^{N-1} Y^*(k) \sum_{n=0}^{N-1} x(n)W_N^{nk} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)Y^*(k) \end{aligned}$$

若令 $y(n) = x(n)$, 则有

$$\sum_{n=0}^{N-1} x(n)x^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)X^*(k)$$

即

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \quad (3.59)$$

3.5 离散傅里叶变换的应用

离散傅里叶变换 DFT 在数字通信、语音处理、图像处理、谱估计、仿真、系统分析、雷达、光学、医学影像、地震等各个领域得到广泛的应用,但是这都是以卷积和相关运算、对连续信号和序列进行谱分析为基础的。

3.5.1 离散傅里叶变换的应用

1. 用 DFT 计算线性卷积

由时域圆周卷积定理可见,圆周卷积可在时域直接计算,也可以按图 3.8 (a) 所示的框图在频域中计算。将 $x_1(n)$ 和 $x_2(n)$ 分别补零,成为长度为 L 点的序列,就可以用 DFT 来计算它们的线性卷积,如图 3.8 (b) 所示。

2. 用 DFT 对模拟信号进行谱分析

在工程实际中经常遇到的是模拟信号 $x_a(t)$, 其频谱函数 $X_a(j\Omega)$ 也是连续函数,为了利用 DFT 对 $x_a(t)$ 进行谱分析,对 $x_a(t)$ 进行时域采样得到 $x(n) = x_a(nT)$, 再对 $x(n)$ 进行 DFT, 得到 $X(k)$ 则是 $x(n)$ 的傅里叶变换 $X(e^{j\omega})$ 在频率区间 $[0, 2\pi]$ 上的 N 点等间隔采样,

这里 $x(n)$ 和 $X(k)$ 都是有限长序列。

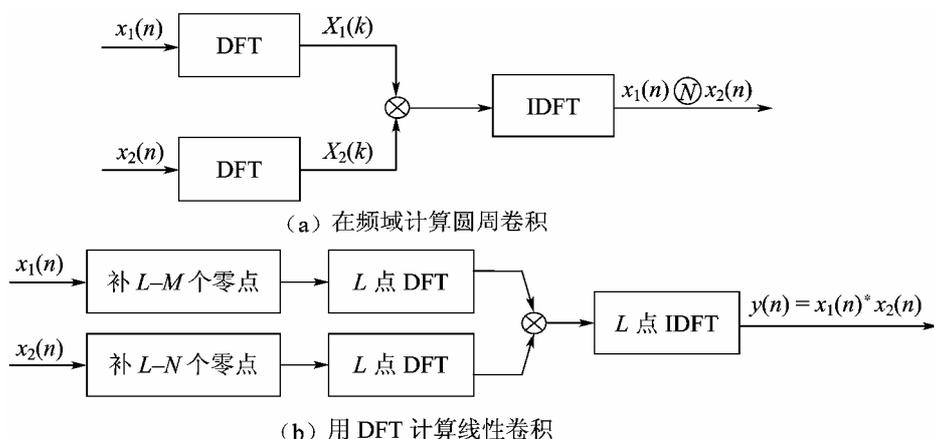


图 3.8 用 DFT 计算线性卷积框图

然而傅里叶变换理论证明，时间有限长的信号其频谱是无限宽的，反之，若信号的频谱是有限宽的则其持续时间将为无限长，因此，按采样定理采样时，采样序列应为无限长，这不满足 DFT 的条件。实际中，对于频谱很宽的信号，为防止时域采样后产生“频谱混迭”，一般用前置滤波器滤除幅度较小的高频成分，使信号的带宽小于折叠频率；同样，对于持续时间很长的信号，采样点数太多也会导致存储和计算困难，一般截取有限点进行计算。上述分析可以看出，用 DFT 对模拟信号进行谱分析，只能是近似的，其近似程度取决于信号带宽、采样频率和截取长度。

模拟信号 $x_a(t)$ 的傅里叶变换对为

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt \quad (3.60)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)e^{j\Omega t} d\Omega \quad (3.61)$$

用 DFT 方法计算这对变换对的方法：

(1) 对 $x_a(t)$ 以 T 为间隔进行采样，即 $x_a(t)|_{t=nT} = x_a(nT) = x(n)$ ，由于

$$t \rightarrow nT, \quad dt \rightarrow T, \quad \int_{-\infty}^{\infty} dt \rightarrow \sum_{n=-\infty}^{\infty} T$$

因此得到

$$X(j\Omega) \approx T \sum_{n=-\infty}^{\infty} x(nT)e^{-j\Omega nT} \quad (3.62)$$

$$x(nT) \approx \frac{1}{2\pi} \int_0^{\Omega_s} X(j\Omega)e^{j\Omega nT} d\Omega \quad (3.63)$$

(2) 将序列 $x(n) = x(nT)$ 截断成包含有 N 个抽样点的有限长序列，因此式 (3.62) 成为

$$X(j\Omega) \approx T \sum_{n=0}^{N-1} x(nT)e^{-j\Omega nT} \quad (3.64)$$

由于时域抽样，抽样频率为 $f_s = 1/T$ ，则频域产生以 f_s 为周期的周期延拓。如果频域是带限信号，则有可能不产生频谱混迭，而成为连续周期频谱序列，频谱的周期为 $f_s = 1/T$ 。

(3) 为了数值计算, 频域上也要抽样, 即在频域的一个周期中取 N 个样点, $f_s = NF_0$, 每个样点间隔为 F_0 。频域抽样使频域的积分式变成求和式, 而在时域就得到原来已经截断的离散时间序列的周期延拓, 时域周期为 $T_0 = 1/F_0$ 。因此有

$$\Omega \rightarrow k\Omega_0, \quad d\Omega \rightarrow \Omega_0, \quad \int_{-\infty}^{\infty} d\Omega \rightarrow \sum_{k=0}^{N-1} \Omega_0$$

并得到一些参量关系为

$$\begin{aligned} T_0 &= \frac{1}{F_0} = \frac{N}{f_s} = NT \\ \Omega_0 &= 2\pi F_0 \\ \Omega_0 T &= \frac{\Omega_0}{f_s} = \Omega_0 \cdot \frac{2\pi}{\Omega_s} = 2\pi \frac{F_0}{f_s} = 2\pi \frac{T}{T_0} = \frac{2\pi}{N} \end{aligned}$$

由式 (3.64) 得到

$$X(jk\Omega_0) = X(j\Omega)|_{\Omega=\Omega_0} \approx T \sum_{n=0}^{N-1} x(nT)e^{-jk\Omega_0 nT} = T \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}nk} = T \text{DFT}[x(n)] \quad (3.65)$$

$$\begin{aligned} x(nT) &\approx \frac{\Omega_0}{2\pi} \sum_{k=0}^{N-1} X(jk\Omega_0)e^{jk\Omega_0 nT} = F_0 \sum_{k=0}^{N-1} X(jk\Omega_0)e^{j\frac{2\pi}{N}nk} \\ &= F_0 N \frac{1}{N} \sum_{k=0}^{N-1} X(jk\Omega_0)e^{j\frac{2\pi}{N}nk} = f_s \frac{1}{N} \sum_{k=0}^{N-1} X(jk\Omega_0)e^{j\frac{2\pi}{N}nk} \\ &= \frac{1}{T} \text{IDFT}[X(jk\Omega_0)] \end{aligned} \quad (3.66)$$

这就是用离散傅里叶变换法求连续非周期信号的傅里叶变换的抽样值的方法。

3.5.2 应用中需要注意的若干问题

1. 时域和频域混迭

根据采样定理, 只有当采样频率 f_s 大于信号的最高频率 f_c 两倍时, 才能避免频域混迭。即

$$f_s > 2f_c$$

也就是抽样间隔为 T 满足 $T = 1/f_s < 1/2f_c$ 。

工程应用中为确保避免频域混迭, 一般将 f_s 取的更高一些, 但是也不宜过高, 因为高的 f_s 意味着采样点数多, 内存消耗大, 运算时间过长, 通常以取 $f_s \geq (4 \sim 10)f_c$ 为宜。

实际信号的持续时间都是有限的, 因而从理论上来说, 其频谱宽度是无限的, 无论取多大的采样频率也不能满足采样定理。但是超过一定范围的高频分量对信号已没有多大的影响, 因而在工程上总是对信号先进行低通滤波——预滤波或抗混迭滤波, 限制高于 f_c 的频率分量出现。

另一方面, DFT 得到的频率函数也是离散的, 其频域抽样间隔为 F_0 , 即频率分辨率。 $T_0 = 1/F_0$ 也就为最短信号记录长度。为了对全部信号进行采样, 必须使抽样点数 N 满足条件

$$N = \frac{T_0}{T} = \frac{f_s}{F_0}$$

从以上 T 和 T_0 两个公式来看, 信号的最高频率分量 f_c 和频率分辨力 F_0 有着矛盾关系, 要想使 f_c 增加, 则时域抽样间隔 T 就一定要减小, 而 f_s 就要增加。若在抽样点数 N 固定的情况下, 必然使 F_0 增加, 即频率分辨力下降。反之, 要提高频率分辨力 (减小 F_0) 就要增加 T_0 , 当 N 给定时, 必然导致 T 增加, 因而就减小了高频容量 f_c 。因此, 若既要保证高频容量 f_c 又要有高的频率分辨力, 惟一有效的方法就是增加在记录长度内的点数 N , 在 f_c 和 F_0 给定的条件下, N 必须满足

$$N > \frac{2f_c}{F_0}$$

2. 截断效应

在实际中遇到的序列 $x(n)$, 其长度往往是很长的, 甚至是无限长的, 用 DFT 对其进行谱分析时, 必须将其截断为长度为 N 的有限长序列, 即

$$y(n) = x(n)R_N(n)$$

根据频率卷积定理有

$$Y(e^{j\omega}) = \frac{1}{2\pi} X(e^{j\omega}) * H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\theta}) R_N(e^{j(\omega-\theta)}) d\theta \quad (3.67)$$

式中

$$R_N(e^{j\omega}) = \text{DFT}[R_N(n)] = e^{-j\omega \frac{N-1}{2}} \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

其幅频曲线如图 3.9 (a) 所示, 其中 $|\omega| < 2\pi/N$ 部分称为主瓣, 其余部分称为旁瓣。

假设 $x(n) = \cos(\omega_0 n)$, $\omega_0 = \pi/4$, 则

$$X(e^{j\omega}) = \pi \sum_{l=-\infty}^{\infty} [\delta(\omega - \frac{\pi}{4} - 2\pi l) + \delta(\omega + \frac{\pi}{4} - 2\pi l)]$$

$x(n)$ 的频谱 $X(e^{j\omega})$ 如图 3.9 (b) 所示, $y(n)$ 对应的幅频曲线如图 3.9 (c) 所示。由图 3.9 可见, 序列截断后的频谱 $Y(e^{j\omega})$ 与原序列频谱 $X(e^{j\omega})$ 有着明显的差别, 这种差别对谱分析带来两方面的影响:

(1) 频谱泄露

原序列 $x(n)$ 的频谱是离散谱线, 经截断后使每根谱线都带上一个辛格谱, 就好像使谱线向两边延伸, 通常将这种因时域上的截断导致频谱展宽称之为“泄露”, 显然泄露使频谱变得模糊, 分辨率降低。

(2) 谱间干扰

因截断使在主谱线两边形成许多旁瓣, 引起不同分量间的干扰, 称之为谱间干扰, 这不仅影响频谱分辨率, 严重时强信号的旁瓣可能湮灭弱信号的主谱线, 或者将强信号谱的旁瓣误认为是另一信号的谱线, 从而形成假信号, 使谱分析产生较大的偏差。

截断效应是无法完全消除的, 只能根据要求折中选择有关参量。首先可以取更长的数

据，也就是使截断窗加宽，当然数据太长也必然会导致存储量和运算量增加；其次数据不要突然截断，也就是不要加矩形窗，而是缓慢截断，即加各种缓变的窗（如三角窗、升余弦窗等），使得窗谱的旁瓣能量更小，卷积后造成的泄露减小。还有一个好的方法就是用近代谱分析的方法，但是这已经超出本书的范围了。

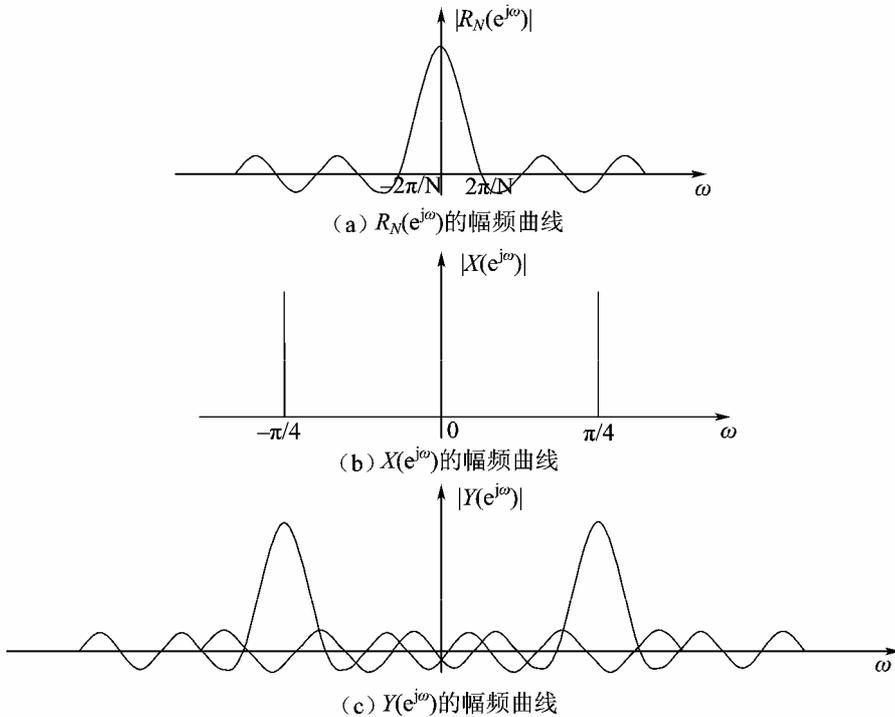


图 3.9 截断效应

3. 栅栏效应

N 点 DFT 是在频率区间 $[0, 2\pi]$ 上对信号的频谱进行 N 点等间隔采样，得到的是若干个离散点 $X(k)$ ，且它们只限制为基频 F_0 的整数倍，这就好像在栅栏的一边通过缝隙看另一边的景象，只能在离散点的地方看到真实的景象，其余部分频谱成分被遮挡，所以称为栅栏效应。减小栅栏效应，可以在时域数据末端增加一些零值点，使一个周期内的点数增加，但是不改变原有的记录数据，即增加频域抽样点数 N ，频域抽样为 $\frac{2\pi}{N}k$ ，这样必然使谱线更密，这样原来看不到的谱分量就可能看到了。

4. 信号长度的选择

在时域内对信号长度的选择会影响 DFT 运算的正确性。实际的信号往往是随机的，没有确定的周期，因此在实际中，应尽可能地估计出几个典型的、带有一定周期性的信号区域进行频谱分析，然后再取其平均值，从而得到合理的结果。

3.6 MATLAB 在 DFT 中的应用

3.6.1 MATLAB 信号处理工具箱提供一些函数

1. fft 和 ifft : 一维快速正傅里叶变换和逆傅里叶变换

$$X = \text{fft}(x, N)$$

采用 FFT 算法计算序列向量 X 的 N 点 DFT, 缺省 N 时, fft 函数自动按 X 的长度计算 DFT。当 N 为 2 的整数次幂时, fft 按基 2 算法计算, 否则用混合基算法。 ifft 的调用格式类似。

2. fft2 和 ifft2 : 二维快速正傅里叶变换和逆傅里叶变换

$$(1) Y = \text{fft2}(X)$$

数据二维傅里中变换参数 X 是向量 $\text{fft2}(X)$ 相当于 $\text{fft}(\text{fft}(X))'$, 即先对 X 的列做一维傅里叶变换, 然后再对变换结果的行做一维傅里叶变换; 若 X 是向量, 则此傅里叶变换即变成一维傅里叶变换 fft ; 若 X 是矩阵, 则是计算该矩阵的二维傅里中变换。

$$(2) Y = \text{fft2}(X, M, N)$$

通过对 X 进行补零或截断, 使得 X 成为 $(M * N)$ 的矩阵。

函数 ifft2 的参数应用与函数 fft2 完全相同。

3. czt : 线性调频 Z 变换

$$y = \text{czt}(x, m, w, s)$$

此函数计算由 $z = a * w.^{-(0:m-1)}$ 定义的 z 平面螺旋线上各点的 Z 变换, a 规定了起点, w 规定了相邻点的比例, m 规定了变换的长度, 后三个变量默认值为 $a = 1$, $w = \exp(j * 2 * \pi / m)$ 及 $m = \text{length}(x)$, 因此 $y = \text{czt}(x)$ 就等于 $y = \text{fft}(x)$ 。

4. dct 和 idct : 离散余弦正变换和离散余弦逆变换

$$y = \text{dct}(x, N)$$

完成如下变换, N 的默认值为 $\text{length}(x)$ 。

$$Y(K) = \sum_{n=1}^N 2x(n) \cos\left(\frac{\pi}{2n} k(2n+1)\right) \quad k = 0, 1, \dots, N-1$$

5. fftsift

$$Y = \text{fftsift}(X)$$

用来重新排列 $X = \text{fft}(x)$ 的输出, 当 X 为向量时, 把 X 的左右两半进行交换, 从而将零

频分量移至频谱的中心；如果 X 为二维傅里叶变换的结果，它同时将 X 的左右和上下部分进行交换。

6. fftfilt

```
y=fftfilt(b,x)
```

采用重叠相加法 FFT 对信号向量 x 快速滤波，得到输出序列向量 y ，向量 b 为 FIR 滤波器的单位脉冲响应， $h(n) = b(n+1)$ ， $n = 0, 1, \dots, \text{length}(b)-1$ 。

```
y=fftfilt(b,x,N)
```

自动选取 FFT 长度 $NF = 2^{\text{nextpow2}(N)}$ ，输入数据 x 分段长度 $M = NF - \text{length}(b) + 1$ ，其中 $\text{nextpow2}(N)$ 函数求得一个整数，满足

$$2^{(\text{nextpow2}(N)-1)} \leq N \leq 2^{\text{nextpow2}(N)}$$

N 缺省时，fftfilt 自动选择合适的 FFT 长度 NF 和对 x 的分段长度 M 。

3.6.2 MATLAB 在 DFT 中的应用

例 3.4 序列的移位和周期延拓运算。

已知 $x(n) = 0.8^n R_8(n)$ ，利用 MATLAB 生成并图示 $x(n)$ ， $x(n-m)$ ， $x((n))_8 R_N(n)$ 和 $x((n-m))_8 R_N(n)$ ，其中 $N = 24$ ， $0 < m < N$ ， $x((n))_8$ 表示 $x(n)$ 以 8 为周期的延拓。

解：程序清单如下。

```
N=24;M=8;m=3; %设移位值为 3
n=0:N-1;
x1=0.8.^n;x2=[(n>=0)&(n<M)];
xn=x1.*x2; %产生 x(n)
[xm,nm]=sigshift(xn,n,m); %产生 x(n-m)
xc=xn(mod(n,8)+1); %产生 x(n)的周期延拓,求余后加 1 是因为 MATLAB 向
量下标从开始
xcm=xn(mod(n-m,8)+1); %产生 x(n)移位后的周期延拓
subplot(2,2,1);stem(n,xn,'.');axis([0,length(n),0,1]);title('x(n)')
subplot(2,2,2);stem(nm,xm,'.');axis([0,length(nm),0,1]);title('x(n-m)')
subplot(2,2,3);stem(n,xc,'.');axis([0,length(n),0,1]);title('x((n)的周期
延拓)')
subplot(2,2,4);stem(n,xcm,'.');axis([0,length(n),0,1]);title('x(n)的循环
移位')
```

程序运行结果如图 3.10 所示。

例 3.5 基本序列的离散傅里叶变换计算。

复正弦序列 $x_1(n) = e^{j\frac{\pi}{8}n} R_N(n)$

余弦序列 $x_2(n) = \cos(\frac{\pi}{8}n) R_N(n)$

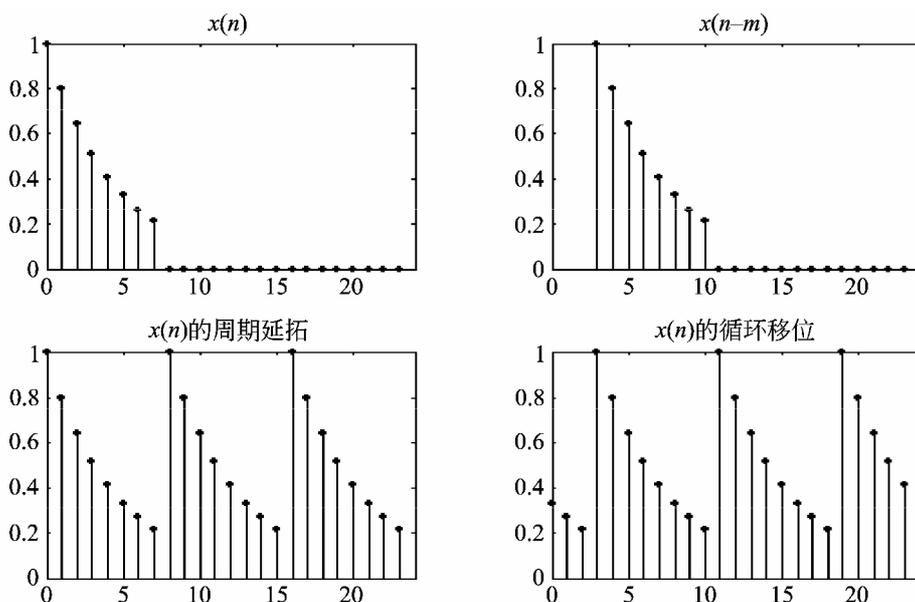


图 3.10 序列的移位和周期延拓

分别对以上序列求当 $N = 16$ 和 $N = 8$ 时的 DFT，并绘出幅频特性曲线，并分析两种 N 值下 DFT 是否有差别，及产生差别的原因。

解：程序清单如下。

```

N=16;N1=8;
n=0:N-1;k=0:N1-1;
x1n=exp(j*pi*n/8);           %产生 x1(n)
X1k=fft(x1n,N);             %计算 N 点 DFT[x1(n)]
X2k=fft(x1n,N1);           %计算 N1 点 DFT[x1(n)]
x2n=cos(pi*n/8);           %产生 x2(n)
X3k=fft(x2n,N);            %计算 N 点 DFT[x2(n)]
X4k=fft(x2n,N1);           %计算 N1 点 DFT[x2(n)]
Subplot(2,2,1);stem(n,abs(X1k),'.');axis([0,20,0,20]);ylabel('|X1(k)|')
title('16 点的 DFT[x1(n)]')
Subplot(2,2,2);stem(n,abs(X3k),'.');axis([0,20,0,20]);ylabel('|X2(k)|')
title('16 点的 DFT[x2(n)]')
Subplot(2,2,3);stem(k,abs(X2k),'.');axis([0,20,0,20]);ylabel('|X1(k)|')
title('8 点的 DFT[x1(n)]')
Subplot(2,2,4);stem(k,abs(X4k),'.');axis([0,20,0,20]);ylabel('|X2(k)|')
title('8 点的 DFT[x2(n)]')

```

程序运行结果如图 3.11 所示。

N 点离散傅里叶变换的一种物理解释就是， $X(k)$ 是 $x(n)$ 以 N 为周期的周期延拓序列的离散傅里叶级数系数 $X(k)$ 的主值区间序列，即 $X(k) = X(k)R_N(k)$ 。当 $N = 16$ 时， $x_1(n)$ 和 $x_2(n)$ 正好分别是 $e^{j\frac{\pi}{8}n}$ 、 $\cos(\frac{\pi}{8}n)$ 的一个周期，所以 $x_1(n)$ 和 $x_2(n)$ 的周期延拓序列就是这两

个单一频率的正弦序列,其离散傅里叶级数的系数分别如图 3.11 中的“16 点的 DFT $[x_1(n)]$ ”和图 3.11 中的“16 点的 DFT $[x_2(n)]$ ”所示。而当 $N=8$ 时, $x_1(n)$ 和 $x_2(n)$ 正好分别是 $e^{j\frac{\pi}{8}n}$ 、 $\cos(\frac{\pi}{8}n)$ 的半个周期,所以 $x_1(n)$ 和 $x_2(n)$ 的周期延拓序列就不再是单一频率的正弦序列,而是含有丰富的谐波成分,其离散傅里叶级数的系数与 $N=16$ 时的差别很大,因此对信号进行谱分析的时候,一定要截取整个周期,否则得到错误的频谱。

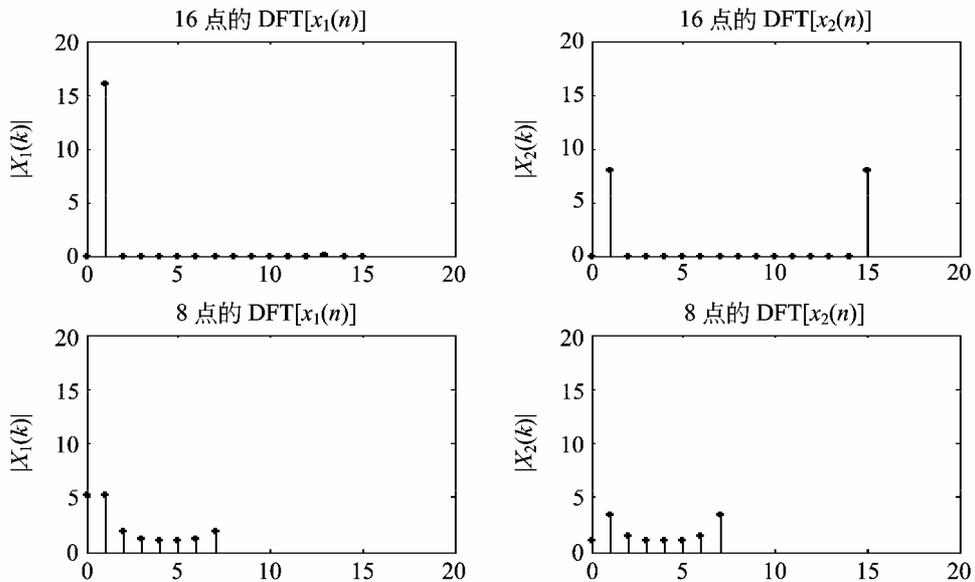


图 3.11 基本序列的离散傅里叶变换

例 3.6 验证 N 点 DFT 的物理意义。

已知 $x(n)=R_4(n)$, $X(e^{j\omega})=\text{DFT}[x(n)]=\frac{1-e^{-j4\omega}}{1-e^{-j\omega}}$, 绘制相应的幅频和相频曲线,并计算

图示 $N=8$ 和 $N=16$ 时的 DFT。

解:程序清单如下。

```
N1=8;N2=16; % 两种 FFT 的变换长度
n=0:N1-1;k1=0:N1-1; k2=0:N2-1;
w=2*pi*(0:2047)/2048;
Xw=(1-exp(-j*4*w))./(1-exp(-j*w));
%对 x(n)的频谱函数采样 2048 个点可以近似地看做是连续的频谱
xn=[(n>=0)&(n<4)]; %产生 x(n)
X1k=fft(xn,N1); %计算 N1=8 点的 X1(k)
X2k=fft(xn,N2); %计算 N2=16 点的 X2(k)
subplot(3,2,1);plot(w/pi,abs(Xw));xlabel('w/π')
subplot(3,2,2);plot(w/pi,angle(Xw));axis([0,2,-pi,pi]);line([0,2],[0,0]);
xlabel('w/π')
subplot(3,2,3);stem(k1,abs(X1k),'.');
```

```

xlabel('k(ω=2πk/N1)');ylabel('|X1(k)|');hold on
plot(N1/2*w/pi,abs(Xw)) %图形上叠加连续频谱的幅度曲线
subplot(3,2,4);stem(k1,angle(X1k));
axis([0,N1,-pi,pi]);line([0,N1],[0,0]);
xlabel('k(ω=2πk/N1)');ylabel('Arg[X1(k)]');hold on
plot(N1/2*w/pi,angle(Xw)) %图形上叠加连续频谱的相位曲线
subplot(3,2,5);stem(k2,abs(X2k),'.'');
xlabel('k(ω=2πk/N2)');ylabel('|X2(k)|');hold on
plot(N2/2*w/pi,abs(Xw))
subplot(3,2,6);stem(k2,angle(X2k),'.'');
axis([0,N2,-pi,pi]);line([0,N2],[0,0]);
xlabel('k(ω=2πk/N2)');ylabel('Arg[X2(k)]');hold on
plot(N2/2*w/pi,angle(Xw))

```

程序运行结果如图 3.12 所示。

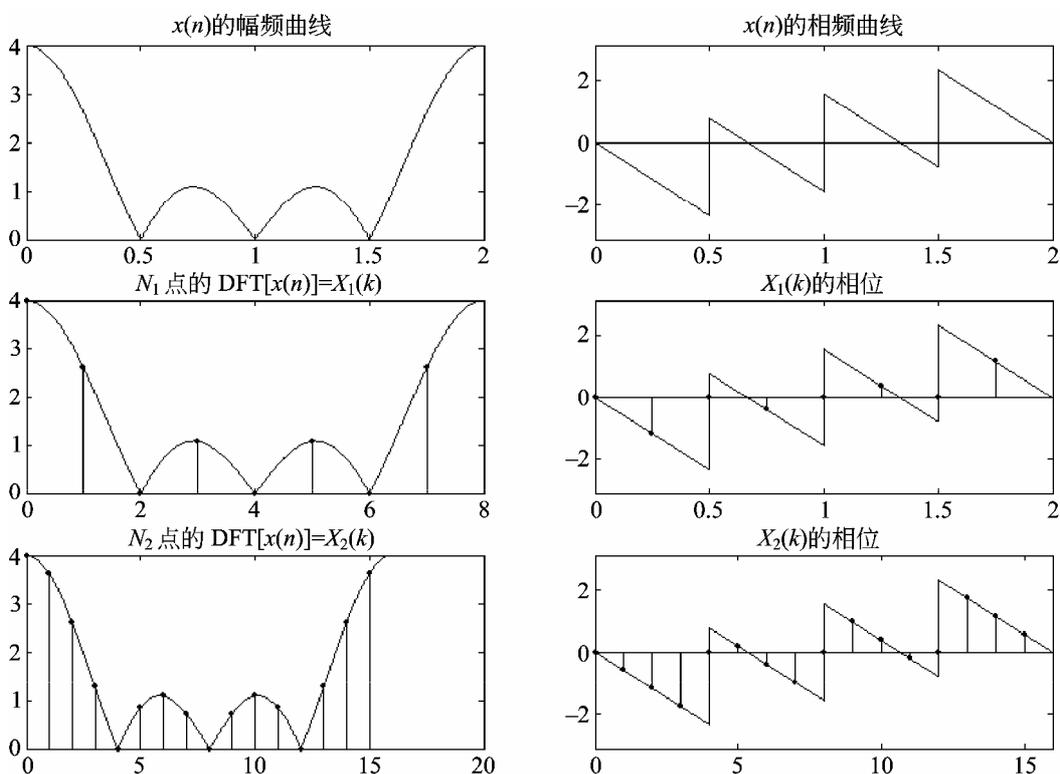


图 3.12 离散傅里叶变换和傅里叶变换的采样关系

通过前面的理论已经知道，序列 $x(n)$ 的 N 点 DFT 的物理意义是对 $X(e^{j\omega})$ 在 $[0, 2\pi]$ 上进行 N 点的等间隔采样。图 3.12 可以直观的看到 $X(k)$ 与 $X(e^{j\omega})$ 之间的采样关系。

例 3.7 验证频域采样和时域采样的对偶性。

- (1) 产生一个三角波序列 $x(n)$ ，长度为 $M = 40$ 。
- (2) 计算 $N = 64$ 时的 $X(k) = \text{DFT}[x(n)]$ ，并图示 $x(n)$ 和 $X(k)$ 。

(3) 对 $X(k)$ 在 $[0, 2\pi]$ 上进行 32 点抽样, 得到 $X_1(k) = X(2k), k=0, 1, \dots, 31$ 。

(4) 求 $X_1(k)$ 的 32 点 IDFT, 即 $x_1(n) = \text{IDFT}[X_1(k)]$ 。

(5) 绘制 $x_1((n))_{32}$ 的波形图, 观察 $x_1((n))_{32}$ 和 $x(n)$ 的关系, 并加以说明。

解: 程序清单如下。

```
M=40;N=64;n=0:M;
xa=[0:floor(M/2)];xb=ceil(M/2)-1:-1:0;
xn=[xa,xb] %产生长度为 M=40 的三角波序列 x(n)
Xk=fft(xn,64); %计算 X(k)=DFT[x(n)]
X1k=Xk(1:2:N) %对 X(k) 隔点抽取得到 X1(k)
x1n=ifft(X1k,32); %计算 x1(n)=IDFT[X1(k)]
nc=0:4*N/2; %取 129 点进行观察
xc=x1n(mod(nc,N/2)+1); %将 x1(n) 以 N/2 为周期进行延拓
subplot(3,2,1);stem(n,xn,'. ');ylabel('x(n)');title('40 点的三角波序列
x(n)')
subplot(3,2,2);k1=0:N-1;stem(k1,abs(Xk),' ');ylabel('|X(k)|');title('64
点的 DFT[x(n)]')
subplot(3,2,3);k2=0:N/2-1;stem(k2,abs(X1k),' ');ylabel('|x1(k)|');
title('隔点抽取 X(k) 得到 X1(k)')
subplot(3,2,4);n1=0:N/2-1;stem(n1,x1n,'. ');ylabel('x1(n)');title('32 点的
IDFT[X2(k)]=x1(n)')
subplot(3,2,5);stem(nc,xc,'. ');ylabel('x1((n))32');title('x1(n) 的周期延拓
序列')
```

程序运行结果如图 3.13 所示。

由图 3.13 看出, 在频域 $[0, 2\pi]$ 上采样点数 $N=32$ 小于离散信号 $x(n)$ 的长度 $M=40$, 所以产生时域混叠现象, 不能由 $X_1(k)$ 恢复出原序列 $x(n)$ 。只有当 $N \geq M$ 时, 才能由频域采样 $X_1(k)$ 不失真的恢复出原序列 $x(n)$, 即 $x(n) = \text{DFT}[X_1(k)]$ 。

例 3.8 用快速卷积法计算下面两个序列的卷积。

$$x(n) = \sin(0.4n)R_{15}(n)$$

$$h(n) = 0.9^n R_{20}(n)$$

快速卷积计算框图如图 3.14 所示。

解: 程序清单如下。

```
M=15;N=20;nx=1:15;nh=1:20;
xn=sin(0.4*nx);hn=0.9.^nh;
L=pow2(nextpow2(M+N-1));
Xk=fft(xn,L);
Hk=fft(hn,L);
Yk=Xk.*Hk;
yn=ifft(Yk,L);ny=1:L
subplot(3,1,1);stem(nx,xn,'. ');title('x(n)');
subplot(3,1,2);stem(nh,hn,'. ');title('h(n)');
subplot(3,1,3);stem(ny,real(yn),' ');title('y(n)');
```

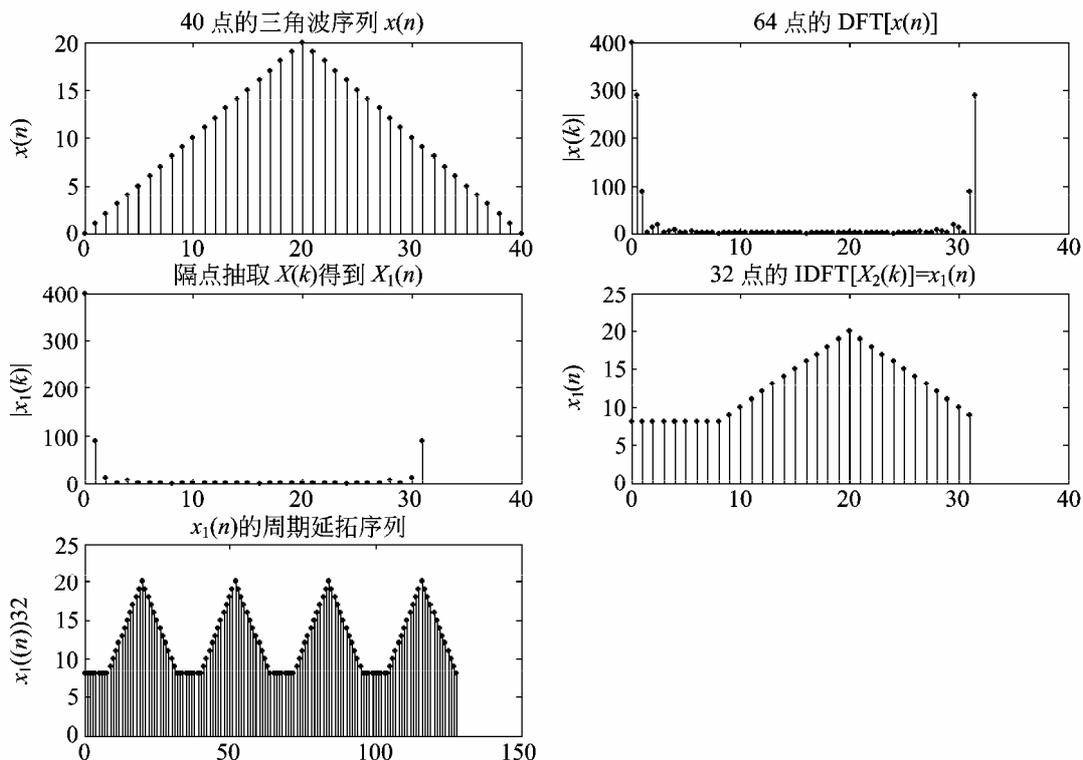
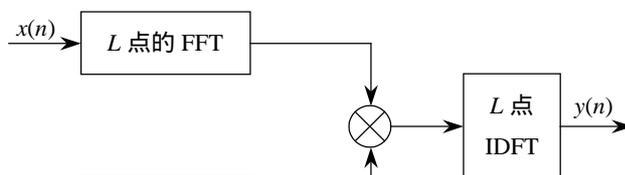


图 3.13 频域采样和时域采样的对偶性



错误！

图 3.14 快速卷积框图

程序运行结果如图 3.15 所示。

FFT 的变换长度 L 必须满足 $L \geq N+M-1$ ，输出 $y(n)$ 才等于 $x(n)$ 和 $h(n)$ 的线性卷积。计算两个序列的卷积时，也可以直接调用函数 conv 来计算，因为 MATLAB 中的计时比较粗糙，所以只有当 N 和 M 较大的时候，才能比较两种方法的执行时间快慢。

例 3.9 用 DFT 对连续信号作谱分析。

设 $x_a(t) = \cos(200\pi t) + \sin(100\pi t) + \cos(50\pi t)$ ，用 DFT 分析 $x_a(t)$ 的频谱结构，选择不同的截取长度 T_p ，观察截断效应，试用加窗的方法减少谱间干扰。

选取的参数：

- (1) 频率 $f_s = 400\text{Hz}$ ， $T = 1/f_s$ 。

(2) 采样信号序列 $x(n) = x_a(nT)w(n)$, $w(n)$ 是窗函数, 选取两种窗函数: 矩形窗函数 $w(n) = R_N(n)$ 和 Hamming 窗, 后者在程序中调用函数 `hamming` 产生宽度为 N 的 Hamming 窗函数列向量 $w(n)$ 。

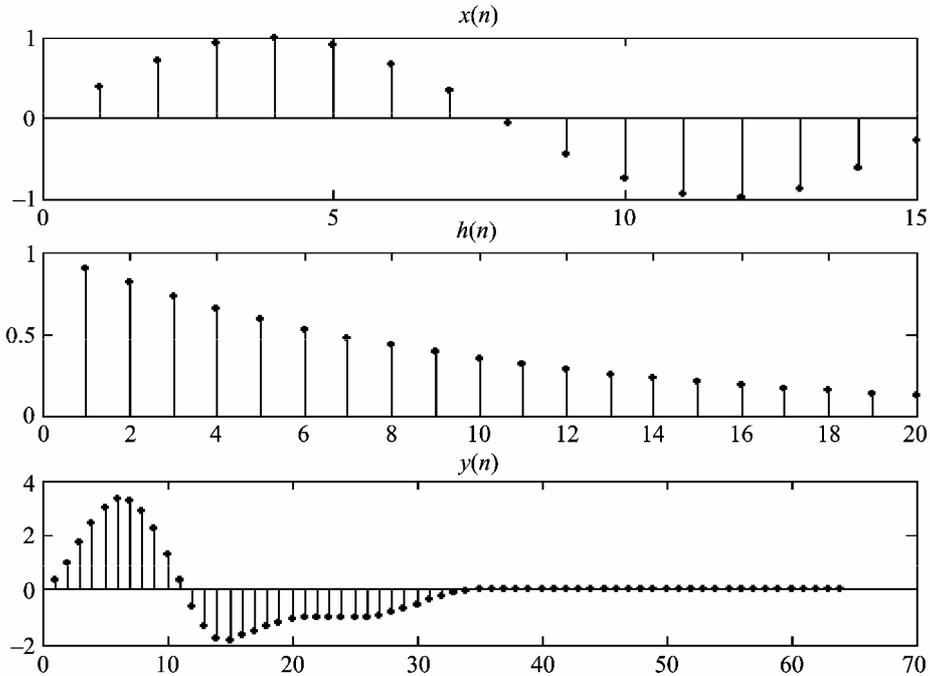


图 3.15 $x(n)$ 、 $h(n)$ 及其线性卷积波形

(3) 对 $x(n)$ 做 2048 点 DFT, 作为 $x_a(t)$ 的近似连续频谱 $X_a(jf)$ 。其中 N 为采样点数, $N = f_s T_p$, T_p 为截取时间长度, 取三种长度 0.04s 、 $4 \times 0.04\text{s}$ 、 $8 \times 0.04\text{s}$ 。

解: 程序清单如下。

```
clear;close all
fs=400;T=1/fs; %采样频率和采样间隔
Tp=0.04;N=Tp*fs; %采样点数 N
N1=[N,4*N,8*N]; %设定三种截取长度
for m=1:3
    n=1:N1(m);
    xn=cos(200*pi*n*T)+ sin(100*pi*n*T)+ cos(50*pi*n*T);
    Xk=fft(xn,4096);
    fk=[0:4095]/4096/T;
    subplot(3,2,2*m-1);plot(fk,abs(Xk)/max(abs(Xk)));
    if m==1 title('矩形窗截取');end
end
%hamming 窗截断
for m=1:3
    n=1:N1(m);
```

```

wn=hamming(N1(m));
xn=cos(200*pi*n*T)+ sin(100*pi*n*T)+ cos(50*pi*n*T).*wn';
Xk=fft(xn,4096);
fk=[0:4095]/4096/T;
subplot(3,2,2*m);plot(fk,abs(Xk)/max(abs(Xk)));
if m==1 title('hamming 窗截取');end
end

```

程序运行结果如图 3.16 所示。

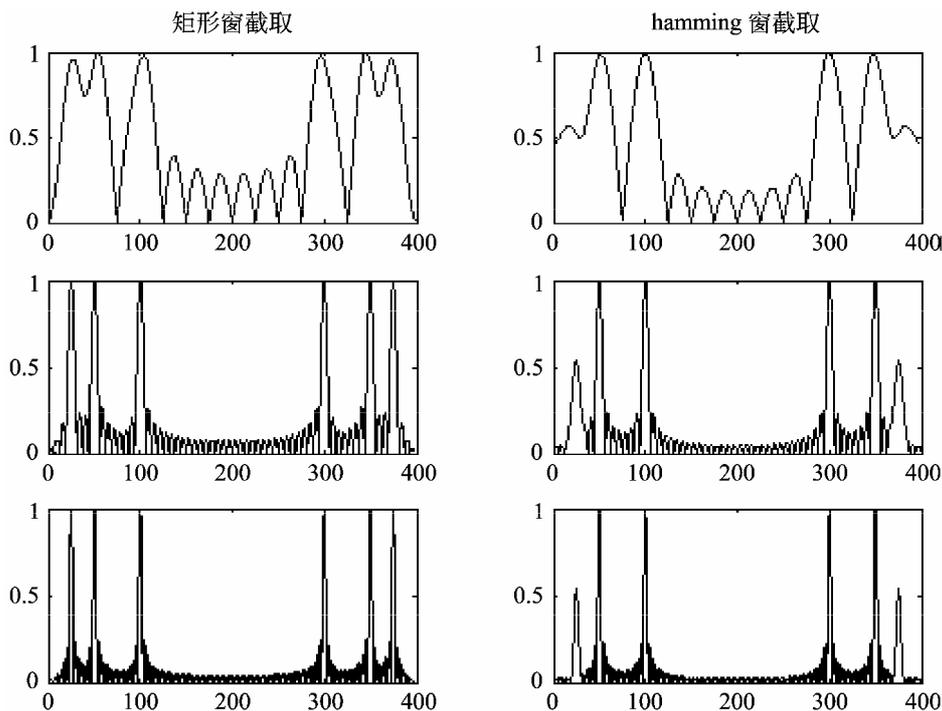


图 3.16 DFT 对连续信号作谱分析

图 3.16 从上到下载取的长度依次分别是 N 、 $4N$ 、 $8N$ ，由于截断使原频谱中的单频谱线展宽（也成为“泄露”），截取的长度越长，泄露越少，频率分辨越高。当截取长度为 N (T_p 为 0.04s) 时， 25Hz 和 50Hz 两根谱线已经分辨不清楚了。另外，在本来应该为零的频段上出现了一些参差不齐的小谱包，成为谱间干扰，其大小取决于加窗的类型。

比较矩形窗和 hamming 窗的谱分析结果可见，用矩形窗比用 hamming 窗的频率分辨率高（泄露小），但是谱间干扰大，因此 hamming 窗是以牺牲分辨率来换取谱间干扰的降低。更详细的分析以及各种窗函数性能可以查阅相关资料。

例 3.10 理解高密度频谱和高分辨频谱的概念。

设 $x(n) = \cos(0.48\pi n) + \cos(0.52\pi n)$

- (1) 取 $0 \leq n \leq 9$ ，求 $X_1(k)$ 。
- (2) 将(1)中的 $x(n)$ 补零加长到 $0 \leq n \leq 99$ ，求 $X_2(k)$ 。

(3) 增加取样值的个数, 取 $0 \leq n \leq 99$, 求 $X_3(k)$ 。

解: 程序清单如下。

```
N1=10;n1=0:N1-1;x1=cos(0.48*pi*n1)+cos(0.52*pi*n1);
Xk1=fft(x1,10);
k1=0:N1-1;w1=2*pi/10*k1;
subplot(3,2,1);stem(n1,x1,'. ');axis([0,10,-2.5,2.5]);title('信号 x(n),0<=
n<=9')
subplot(3,2,2);stem(w1/pi,abs(Xk1),'. ');axis([0,1,0,10]);title('DFT[x(n)]')
N2=100;n2=0:N2-1;x2=[x1(1:1:10) zeros(1,90)];
Xk2=fft(x2,N2)
k2=0:N2-1;w2=2*pi/100*k2;
subplot(3,2,3);stem(n2,x2,'. ');axis([0,100,-2.5,2.5]);title('信号 x(n)补零
到长度为 N=100')
subplot(3,2,4);plot(w2/pi,abs(Xk2));axis([0,1,0,10]);title('DFT[x(n)]')
N3=100;n3=0:N3-1;x3=cos(0.48*pi*n3)+cos(0.52*pi*n3);
Xk3=fft(x3,N3)
k3=0:N3-1;w3=2*pi/100*k3;
subplot(3,2,5);stem(n3,x3,'. ');axis([0,100,-2.5,2.5]);title('信号 x(n),
0<=n<=99')
subplot(3,2,6);plot(w3/pi,abs(Xk3),'. ');axis([0,1,0,60]);title('DFT[x(n)]')
```

程序运行结果如图 3.17 所示。

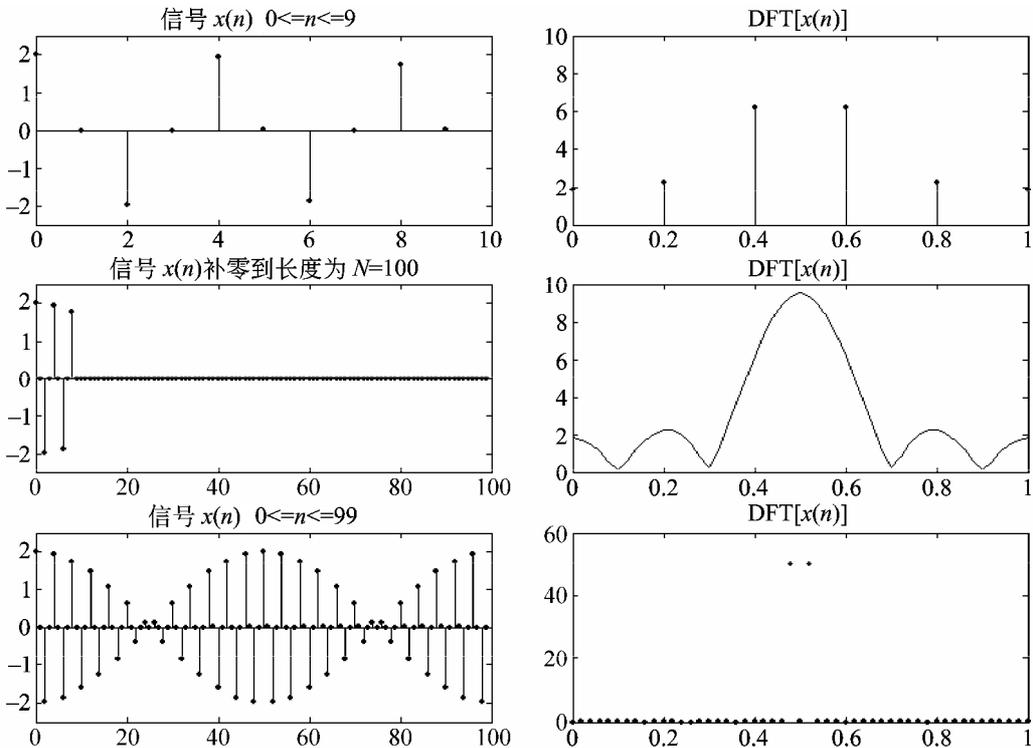


图 3.17 高密度频谱与高分辨频谱

从图 3.17 可以看出, 当 $x(n)$ 取 $0 \leq n \leq 9$ 时, 从相应的 $X(k)$ 图中几乎无法看出有关信号频谱的信息; 将 $x(n)$ 补 90 个零后做 $N = 100$ 点的 DFT, 从相应的 $X(k)$ 图中可以看出, 这时的谱线相当密, 故称为高密度谱线图, 但是从中很难看出信号的频谱部分; 对 $x(n)$ 加长取样数据, 得到长度为 $N = 100$ 的序列, 此时相应的 $X(k)$ 图中可以清晰地看到信号的频谱成分, 这称为高分辨频谱。

第 4 章 快速傅里叶变换

快速傅里叶变换(简称 FFT)并不是与 DFT 不同的另外一种变换,而是为减少比 DFT 计算次数少的一种快速有效的算法。

在信号处理中,DFT 的计算具有举足轻重的地位,信号的相关、滤波、谱估计等都要通过 DFT 来实现。然而,当 N 很大的时候,求一个 N 点的 DFT 要完成 $N \times N$ 次复数乘法和 $N(N-1)$ 次复数加法,其计算量相当大。1965 年库利(J.W.Cooley)和图基(J.W.Tukey)巧妙地利用 W_N 因子的周期性和对称性,构造了一个 DFT 快速算法,即快速傅里叶变换(FFT),从而使得 DFT 的运算真正得到广泛应用。

4.1 直接计算 DFT 的问题及改进的途径

4.1.1 DFT 的运算特点

由前面可知,对于一个有限列长为 N 的序列 $x(n)$,其 DFT 变换为

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k=0, 1, \dots, N-1 \quad (4.1)$$

其逆变换(IDFT)为

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad n=0, 1, \dots, N-1 \quad (4.2)$$

二者的差别只在于 W_N 的指数符号不同,以及差一个比例因子 $1/N$,故式(4.1)或(4.2)运算量是相同的。下面只讨论 DFT 正变换式(4.1)的运算量。

一般来说, $x(n)$ 和 W_N^{nk} 都是复数, $X(k)$ 也是复数,因此每计算一个 $X(k)$ 值,需要 N 次 $x(n)W_N^{nk}$ 形式的复数乘法和 $N-1$ 次复数加法的运算。而 $X(k)$ 共有 N 个点(k 从 $0 \sim N-1$),所以完成全部 DFT 运算总共需要 N^2 次复数乘法和 $N(N-1)$ 次复数加法。由于复数 $x(n)$ 和 W_N^{nk} 可用实数表示成如下形式

$$x(n) = \text{Re}[x(n)] + j\text{Im}[x(n)] \quad (4.3)$$

$$W_N^{nk} = \text{Re}[W_N^{nk}] + j\text{Im}[W_N^{nk}] \quad (4.4)$$

故式(4.1)变为

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n=0}^{N-1} \left\{ \left[\text{Re}[x(n)]\text{Re}[W_N^{nk}] - \text{Im}[x(n)]\text{Im}[W_N^{nk}] \right] + j \left[\text{Re}[x(n)]\text{Im}[W_N^{nk}] + \text{Im}[x(n)]\text{Re}[W_N^{nk}] \right] \right\} \quad (4.5)$$

由式(4.5)可知,一次复数乘法需要用四次实数乘法和二次实数加法(实部、虚部分

别相加)来实现,而一次复数加法则需要二次实数加法。因而每运算一个 $X(k)$ 需要 $4N$ 次复数乘法和 $2N + 2(N-1) = 2(2N-1)$ 次实数加法,要完成整个 DFT 运算共需要 $4N^2$ 次实数相乘和 $N \times 2(2N-1) = 2N(2N-1)$ 次实数加法,由此可见,直接计算 DFT 时,乘法次数与加法次数都与 N^2 成正比, N 愈大,运算工作量将显著增加。为此,需要对 DFT 的计算方法进行改进,以减少总的运算次数。

4.1.2 改善 DFT 运算效率的基本途径

由于系数 W_N^{nk} 具有以下特性:

对称性

$$(W_N^{nk})^* = W_N^{-nk}$$

周期性

$$W_N^{nk} = W_N^{(n+N)k} = W_N^{n(k+N)}$$

则有

$$W_N^{n(N-k)} = W_N^{(N-n)k} = W_N^{-nk}$$

$$W_N^{\frac{N}{2}} = e^{-j\frac{2\pi}{N}\frac{N}{2}} = -1$$

$$W_N^{\left(nk + \frac{N}{2}\right)} = -W_N^{nk}$$

因此,利用 W_N^{nk} 的特性可对 DFT 的运算进行如下改进:

(1) 利用 W_N^{nk} 的对称性使 DFT 运算中有些项合并

式(4.5)中的对称项可合并为

$$\operatorname{Re}[x(n)]\operatorname{Re}\left[W_N^{nk}\right] + \operatorname{Re}[x(N-n)]\operatorname{Re}\left[W_N^{(N-n)k}\right] = \{\operatorname{Re}[x(n)] + \operatorname{Re}[x(N-n)]\}\operatorname{Re}\left[W_N^{nk}\right]$$

$$\operatorname{Im}[x(n)]\operatorname{Im}\left[W_N^{nk}\right] - \operatorname{Im}[x(N-n)]\operatorname{Im}\left[W_N^{(N-n)k}\right] = -\{\operatorname{Im}[x(n)] - \operatorname{Im}[x(N-n)]\}\operatorname{Im}\left[W_N^{nk}\right]$$

式中其他各项也可以找到类似的归并方法。这样,乘法次数可以减少大约一半。

(2) 利用 W_N^{nk} 的周期性和对称性使长序列的 DFT 分解为更小点数的 DFT

由于 DFT 的运算量与 N^2 成正比的, N 越小,DFT 的运算量越小。所以,如果一个长序列的 DFT 能分解为短序列的 DFT 的组合,则显然可取得减少运算工作量的效果。快速傅里叶变换算法正是基于这一基本思想发展起来的。快速傅里叶变换算法形式很多,但基本上可以分成两大类,即按时间抽取(Decimation-In-Time,简称 DIT)法和按频率抽取(Decimation-In-Frequency,简称 DIF)法。

4.2 时间抽取基-2 FFT 算法

4.2.1 算法原理

设序列 $x(n)$ 的长度 N 是 2 的整数幂次方,即

$$N = 2^M \quad (4.6)$$

其中 M 为正整数。如果 N 不满足此关系, 可以通过人为增加若干零值点来到这一要求。这种 N 为 2 的整数幂的 FFT 称为时间抽取 (DIT) 基 -2 FFT。

首先将输入序列 $x(n)$ 按 n 的奇偶分成两个子序列, 即

$$\begin{cases} x(2r) = x_1(r) \\ x(2r+1) = x_2(r) \end{cases} \quad r = 0, 1, \dots, \frac{N}{2}-1 \quad (4.7)$$

则式 (4.1) 变为

$$\begin{aligned} X(k) &= \text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{\substack{n=0 \\ n \text{ 为偶数}}}^{N-1} x(n)W_N^{nk} + \sum_{\substack{n=0 \\ n \text{ 为奇数}}}^{N-1} x(n)W_N^{nk} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x_1(r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x_2(r)(W_N^2)^{rk} \end{aligned}$$

由于 $W_N^2 = e^{-j\frac{2\pi}{N} \cdot 2} = e^{-j\frac{2\pi}{\frac{N}{2}}} = W_{\frac{N}{2}}$, 故上式可表示为

$$X(k) = \sum_{r=0}^{\frac{N}{2}-1} x_1(r)W_{\frac{N}{2}}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x_2(r)W_{\frac{N}{2}}^{rk} = X_1(k) + W_N^k X_2(k) \quad (4.8)$$

式中 $X_1(k)$ 及 $X_2(k)$ 分别是 $x_1(r)$ 及 $x_2(r)$ 的 $N/2$ 点的 DFT。

$$\begin{aligned} X_1(k) &= \sum_{r=0}^{\frac{N}{2}-1} x_1(r)W_{\frac{N}{2}}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_{\frac{N}{2}}^{rk} \\ X_2(k) &= \sum_{r=0}^{\frac{N}{2}-1} x_2(r)W_{\frac{N}{2}}^{rk} = \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_{\frac{N}{2}}^{rk} \end{aligned} \quad (4.9)$$

该式表明: 一个 N 点的 DFT 被分解为两个 $N/2$ 点的 DFT, 这两个 $N/2$ 点的 DFT 按照式 (4.8) 可合成为一个 N 点的 DFT。但 $x_1(r)$ 、 $x_2(r)$ 及 $X_1(k)$ 、 $X_2(k)$ 的列长都是 $N/2$, 即 r 、 k 满足 $r, k = 0, 1, \dots, N/2-1$, 而 $X(k)$ 的列长为 N , 所以按式 (4.8) 计算得到的只是 $X(k)(k = 0, 1, \dots, N-1)$ 的前一半项数的结果, 要用 $X_1(k)$ 、 $X_2(k)$ 来表示全部的 $X(k)$ 值, 还必须应用系数 W 的周期性, 即

$$W_{\frac{N}{2}}^{rk} = W_{\frac{N}{2}}^{r\left(k+\frac{N}{2}\right)}$$

则

$$X\left(\frac{N}{2}+k\right) = \sum_{r=0}^{\frac{N}{2}-1} x_1(r)W_{\frac{N}{2}}^{r\left(\frac{N}{2}+k\right)} = \sum_{r=0}^{\frac{N}{2}-1} x_1(r)W_{\frac{N}{2}}^{rk}$$

即

$$X_1\left(\frac{N}{2}+k\right)=X_1(k) \quad (4.10)$$

同理可得

$$X_2\left(\frac{N}{2}+k\right)=X_2(k) \quad (4.11)$$

式 (4.10) 及式 (4.11) 表明: 后半部分 k 值 ($N/2 \leq k \leq N-1$) 所对应的 $X_1(k)$ 、 $X_2(k)$ 完全重复了前半部分 k 值 ($0 \leq k \leq N/2-1$) 所对应的 $X_1(k)$ 、 $X_2(k)$ 的值。

考虑到 W_N^k 的对称性, 即

$$W_N^{\left(\frac{N}{2}+k\right)}=W_N^{\frac{N}{2}} \cdot W_N^k=-W_N^k \quad (4.12)$$

将式 (4.10) 式 (4.11) 及式 (4.12) 代入式 (4.8) 中, 就可以将 $X(k)$ 的表达式分为前后两部分。

前半部分

$$X(k)=X_1(k)+W_N^k X_2(k), \quad k=0, 1, \dots, \frac{N}{2}-1 \quad (4.13)$$

后半部分

$$\begin{aligned} X\left(\frac{N}{2}+k\right) &= X_1\left(\frac{N}{2}+k\right)+W_N^{\left(\frac{N}{2}+k\right)} X_2\left(\frac{N}{2}+k\right) \\ &= X_1(k)-W_N^k X_2(k), \quad k=0, 1, \dots, \frac{N}{2}-1 \end{aligned} \quad (4.14)$$

由此可见, 只要求出区间 $[0, N/2-1]$ 内各个整数 k 值所对应的 $X_1(k)$ 、 $X_2(k)$ 值, 即可求出 $[0, N-1]$ 区间内的全部 $X(k)$ 值, 从而大大节省了运算 (这一点也正是 FFT 能大量节省计算的关键所在)。

式 (4.13) 和式 (4.14) 的运算过程可用图 4.1 所示信号流图 (蝶形运算流图) 表示。其中图 4.1 (b) 是图 4.1 (a) 的简化形式。图中左边为输入; 中间以一个小圆圈表示加、减运算; 右上支路为相加后的输出, 右下支路为相减后的输出。如果在某一支路上需要进行相乘运算, 则在该支路上标以箭头, 将相乘的系数标在箭头旁边。当支路上没有标出箭头及系数时, 则该支路的传输系数为 1。由此图可知, 每个蝶形运算, 需要一次复数乘法 $[X_2(k)W_N^k]$ 及两次复数加 (减) 法 $[X_1(k)+X_2(k)W_N^k, X_1(k)-X_2(k)W_N^k]$ 。 $N=8$ 时, 采用这种方法分解过程如图 4.2 所示。其输入序列 $x(n)$ 按偶数点的和奇数点进行第一次分解后成为

$$x(2r)=x_1(r), \quad x(2r+1)=x_2(r), \quad r=0, 1, 2, \dots, \frac{N}{2}-1$$

偶序列

$$x_1(0)=x(0), \quad x_1(1)=x(2), \quad x_1(2)=x(4), \quad x_1(3)=x(6)$$

奇序列

$$x_1(0)=x(1), \quad x_1(1)=x(3), \quad x_1(2)=x(5), \quad x_1(3)=x(7)$$

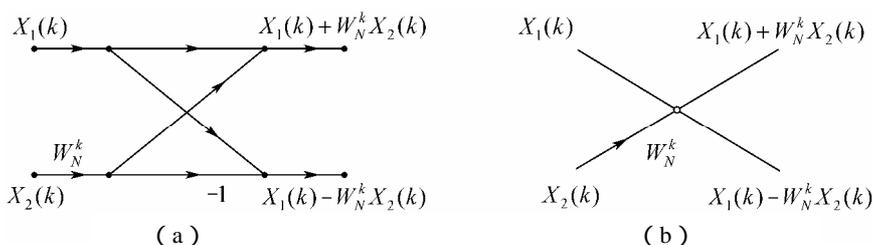


图 4.1 蝶形运算流程图

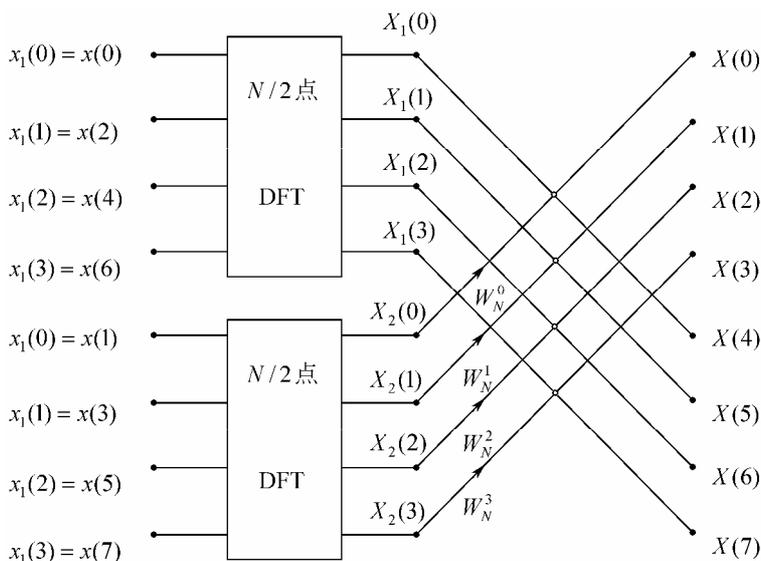


图 4.2 按时间抽取将一个 N 点 DFT 分解为 2 个 $N/2$ 点 DFT ($N=8$)

进一步把每个 $N/2$ 点子序列分解为 $N/4$ 点子序列，对于上面已经形成的偶序列 $x_1(r)$ ，有

$$x_1(2l) = x_3(l), \quad x_1(2l+1) = x_4(l), \quad l=0, 1, 2, \dots, \frac{N}{4}-1$$

偶序列中的偶序列

$$x_3(0) = x_1(0) = x(0), \quad x_3(1) = x_1(2) = x(4)$$

偶序列中的奇序列

$$x_4(0) = x_1(1) = x(2), \quad x_4(1) = x_1(3) = x(6)$$

对上面已经形成的奇数序列 $x_2(r)$ ，有

$$x_2(2l) = x_3(l), \quad x_2(2l+1) = x_6(l), \quad l=0, 1, 2, \dots, \frac{N}{4}-1$$

奇序列中的偶序列

$$x_5(0) = x_2(0) = x(1), \quad x_5(1) = x_2(2) = x(5)$$

奇序列中的奇序列

$$x_6(0) = x_2(1) = x(3), \quad x_6(1) = x_2(3) = x(7)$$

各第一次分解相同，将序列按奇偶两部分进行第二次分解后，可得

$$\begin{aligned} X_1(k) &= \sum_{l=0}^{\frac{N}{4}-1} x_1(2l)W_N^{2lk} + \sum_{l=0}^{\frac{N}{4}-1} x_1(2l+1)W_N^{(2l+1)k} = \sum_{l=0}^{\frac{N}{4}-1} x_3(l)W_{\frac{N}{4}}^{lk} + W_N^k \sum_{l=0}^{\frac{N}{4}-1} x_4(l)W_{\frac{N}{4}}^{lk} \\ &= X_3(k) + W_N^k X_4(k), \quad k=0, 1, \dots, \frac{N}{4}-1 \end{aligned}$$

而

$$X_1\left(\frac{N}{4} + k\right) = X_3(k) - W_N^k X_4(k), \quad k=0, 1, \dots, \frac{N}{4}-1$$

式中

$$X_3(k) = \sum_{l=0}^{\frac{N}{4}-1} x_3(l)W_{\frac{N}{4}}^{lk}, \quad X_4(k) = \sum_{l=0}^{\frac{N}{4}-1} x_4(l)W_{\frac{N}{4}}^{lk}$$

图 4.3 所示为 $N=8$ 时，由两个 $N/4$ 点的 DFT 组合成一个 $N/2$ 点的 DFT 的流图。

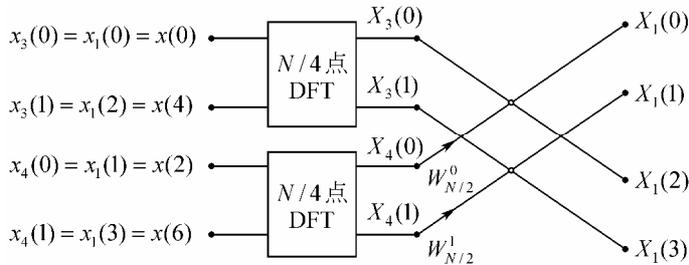


图 4.3 两个 $N/4$ 点的 DFT 组合成一个 $N/2$ 点的 DFT 的流图

$X_2(k)$ 也可以进行相同的分解

$$X_2(k) = X_5(k) + W_N^k X_6(k), \quad k=0, 1, \dots, \frac{N}{4}-1$$

而

$$X_2\left(\frac{N}{4} + k\right) = X_5(k) - W_N^k X_6(k), \quad k=0, 1, \dots, \frac{N}{4}-1$$

式中

$$X_5(k) = \sum_{l=0}^{\frac{N}{4}-1} x_2(2l)W_N^{lk} = \sum_{l=0}^{\frac{N}{4}-1} x_5(l)W_{\frac{N}{4}}^{lk}, \quad X_6(k) = \sum_{l=0}^{\frac{N}{4}-1} x_2(2l+1)W_N^{lk} = \sum_{l=0}^{\frac{N}{4}-1} x_6(l)W_{\frac{N}{4}}^{lk}$$

将系数统一为 $W_N^k = W_N^{2k}$ ，则一个 $N=8$ 点的 DFT 就可以分解为 4 个 $N/4$ 点的 DFT，先做 $N/4$ 点的 DFT 计算，再由相应的 2 个 $N/4$ 点的 DFT 结果合成 $N/2$ 点的 DFT，从而得到 $X_1(k)$ 和 $X_2(k)$ 。最后按式 (4.13) 和式 (4.14) 组合成 N 点的 DFT，其流图如图 4.4 所示。由此可得出一个按时间抽取的完整的 8 点 FFT 流图 (如图 4.5 所示)。

以上算法是将时间 n 按奇、偶分开，故称为时间抽取算法 (decimation in time, DIT)。

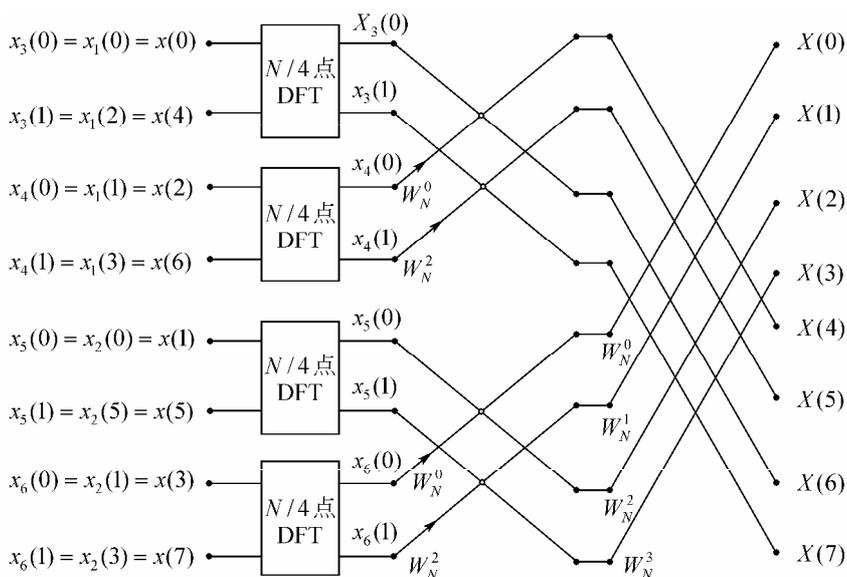


图 4.4 用 4 个 $N/4$ 点的 DFT 及两级蝶形组合来计算 N 点 DFT

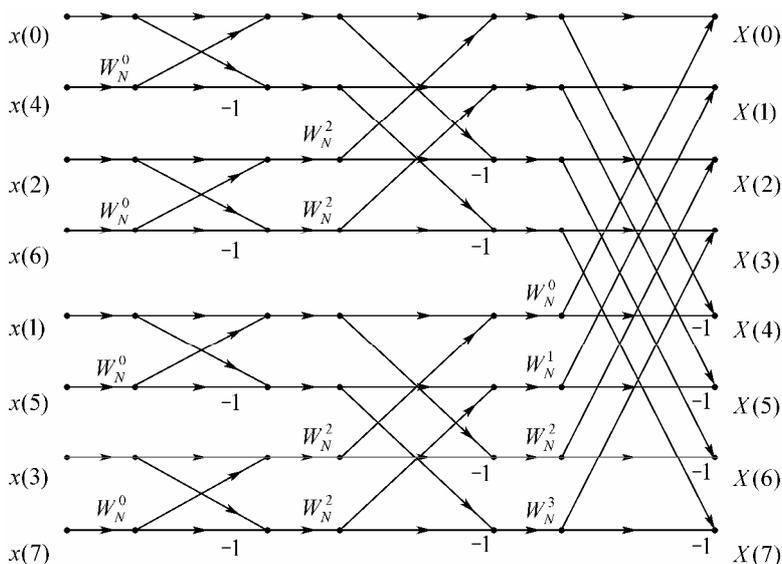


图 4.5 $N=8$ 按时间抽取 FFT 算法信号流图

4.2.2 算法的讨论

下面对 4.2.1 节所述的推导过程进行详细讨论，以找到 FFT 算法的一般规律。

1. “级”的概念

上面推导过程，将 N 点 DFT 先分成 2 个 $N/2$ 点 DFT，再分成 4 个 $N/4$ 点 DFT，进而

分成 8 个 $N/8$ 点的 DFT，直至分成 $N/2$ 个 2 点 DFT。每分一次，称为一“级”运算。因为 $M = \lg N$ ，所以 N 点 DFT 可分成 M 级（如图 4.5 所示），图中 $N=8$ ， $M=4$ ，从左至右依次为 $m=0$ 级， $m=1$ 级， $m=2$ 级。

2. 蝶形单元

在图 4.5 中有大量如图 4.1 的运算结构，由于该运算结构的几何形状像蝴蝶，故称为“蝶形运算单元”，在第 m 级，有

$$\left. \begin{aligned} X_{m+1}(p) &= X_m(p) + W_N^r X_m(q) \\ X_{m+1}(q) &= X_m(p) - W_N^r X_m(q) \end{aligned} \right\}$$

p 、 q 是参与本蝶形单元运算的上、下节点的序号。很明显，第 m 级序号为 p 、 q 的两点只参与这一个蝶形单元的运算，其输出在第 $m+1$ 级，且这一蝶形单元也不再涉及别的点。由于这一特点，在计算机编程时，可将蝶形单元的输出仍放在输入数组中，故称为“同址运算”。

由于每一级都含有 $N/2$ 个蝶形单元，每一个蝶形单元又只需要一次复数乘、两次复数加，因此，完成 $M = \lg N$ 级共需要的复数乘法数 M_f 和复数加法数 M_a 分别是

$$M_f = \frac{N}{2} \lg N = \frac{MN}{2}$$

$$M_a = N \lg N = MN$$

那么，在图 4.5 中，在第 m 级，上、下节点 p 、 q 之间的距离为

$$q - p = 2^m$$

3. “组”的概念

由图 4.5 可见，每一级的 $N/2$ 个蝶形单元可以分成若干组，每一组有着相同的结构及 W^r 因子分布。如 $m=0$ 级分成了四组， $m=1$ 级分成了两组， $m=M-1$ 级分成了一组，因此，第 m 级的组数是 $N/2^{m+1}$ （其中， $m=0, 1, \dots, M-1$ ）。

4. W^r 因子的分布

由式 (4.2) 至式 (4.14) 可见，第一次将 N 点 DFT 分成两个 $N/2$ 点 DFT 时，相当于图 4.5 的最右边一级，这时出现的 W^r 因子是 W_N^r ，而 $r=0, 1, \dots, N/2-1$ ，再往下分时，依次是 $W_{\frac{N}{2}}^r$ 、 $W_{\frac{N}{4}}^r$ ， \dots ，故每一级 W^r 因子分布的规律如下：

$$m=0 \text{ 级, } W_2^r, r=0$$

$$m=1 \text{ 级, } W_4^r, r=0, 1$$

$$m=2 \text{ 级, } W_8^r, r=0, 1, 2, 3$$

\dots

$$m=M-1 \text{ 级, } W_N^r, r=0, 1, \dots, \frac{N}{2}-1$$

因此， W^r 因子分布的一般规律为

第 m 级, $W_{2^{m+1}}^r$, $r=0, 1, \dots, 2^m-1$

5. 码位倒置

由图 4.5 可知, 变换后的输出序列 $X(k)$ 依照正序排列, 但输入序列 $x(n)$ 的次序不再是原来的自然顺序, 这正是由于将 $x(n)$ 按奇、偶分开, 得到两组 $N/2$ 点 DFT, $x(n)$ 的序号是

$$0, 2, 4, 6 \mid 1, 3, 5, 7$$

对每一组再按奇、偶分开, 这时应将每一组仍按自然顺序排列, 故抽取后得四组, 每组的序号是

$$0, 4 \mid 2, 6 \mid 1, 5 \mid 3, 7$$

这一顺序正是图 4.5 输入端序列 $x(n)$ 的排列次序。掌握这一规律, 对 N 为 2 的更高次幂, 都可得到正确的抽取次序。

如果将 $x(n)$ 的序号 $n=0, 1, \dots, N-1$ 写成二进制, 如 $N=8$, 那么 $x(0), \dots, x(7)$ 对应为

$$x(000), x(001), x(010), x(011), x(100), x(101), x(110), x(111)$$

将二进制数码翻转得

$$x(000), x(100), x(010), x(110), x(001), x(101), x(011), x(111)$$

它们对应的十进制序号分别是

$$x(0), x(4), x(2), x(6), x(1), x(3), x(5), x(7)$$

这也正是按奇、偶抽取所得到的顺序。掌握了这一规律, 就可以正确编程, FFT 的软件已是通用程序, 读者只要了解规律即可。

4.3 频率抽取基-2 FFT 算法

在前面, 已经讨论了按时间抽取 FFT 算法, 是把输入序列 $x(n)$ 按其顺序的偶、奇分解为越来越短的序列, 与 DIT 算法相对应, 可把输出序列 $X(k)$ (也是 N 点序列) 按其顺序的偶、奇分解为越来越短的序列, 这种算法称为按频率抽取 (DIF) 的算法。

4.3.1 算法原理

为了讨论方便, 仍假设输入序列 $x(n)$ 的长度为 $N=2^M$, M 为整数。先将 $x(n)$ 按 n 的顺序分成前后两半 (不是按奇偶分成两半)。

前半子序列

$$x(n)$$

后半子序列

$$x\left(n + \frac{N}{2}\right), 0 \leq n \leq \frac{N}{2} - 1$$

则

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{nk} \\
 &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x\left(n + \frac{N}{2}\right)W_N^{\left(n+\frac{N}{2}\right)k} \\
 &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n + \frac{N}{2}\right)W_N^{\frac{N}{2}k} \right] W_N^{nk} \quad k=0, 1, \dots, N-1
 \end{aligned} \tag{4.15}$$

由于 $W_N = e^{-j\frac{2\pi}{N}}$ $\neq W_{\frac{N}{2}}$, 所以式 4.15 中的两个和式并不代表 $N/2$ 点的 DFT。又因

$W_N^{\frac{N}{2}} = -1, W_N^{\frac{N}{2}k} = (-1)^k$, 故上式变为

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{nk} \quad k=0, 1, \dots, N-1 \tag{4.16}$$

当 k 为偶数时, $(-1)^k = 1$, k 为奇数时, $(-1)^k = -1$, 因此, 按 k 的奇偶可将 $X(k)$ 分为两部分。

令 $\begin{cases} k=2r \\ k=2r+1 \end{cases} \quad r=0, 1, \dots, \frac{N}{2}-1$, 于是得

$$X(2r) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^{2rm} = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{\frac{N}{2}}^m, \quad r=0, 1, \dots, \frac{N}{2}-1 \tag{4.17}$$

$$\begin{aligned}
 X(2r+1) &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^{(2r+1)n} \\
 &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n W_{\frac{N}{2}}^m, \quad r=0, 1, \dots, \frac{N}{2}-1
 \end{aligned} \tag{4.18}$$

其中, 式 (4.17) 为输入序列前一半与后一半之和的 $N/2$ 点的离散傅里叶变换, 式 (4.18) 为输入序列前一半与后一半之差与 W_N^n 之积的 $N/2$ 点的离散傅里叶变换。

令

$$g(n) = x(n) + x\left(n + \frac{N}{2}\right), \quad n=0, 1, \dots, \frac{N}{2}-1 \tag{4.19}$$

$$h(n) = \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^n, \quad n=0, 1, \dots, \frac{N}{2}-1 \tag{4.20}$$

则

$$X(2r) = \sum_{n=0}^{\frac{N}{2}-1} g(n)W_{\frac{N}{2}}^m, \quad r=0, 1, \dots, \frac{N}{2}-1 \tag{4.21}$$

$$X(2r+1) = \sum_{n=0}^{\frac{N}{2}-1} h(n)W_{\frac{N}{2}}^m, \quad r=0, 1, \dots, \frac{N}{2}-1 \quad (4.22)$$

式 (4.21) 式 (4.22) 的运算关系可用图 4.6 (a) 所示蝶形运算来表示。这样, 就将一个 N 点的 DFT 按频率 k 的奇偶分解为两个 $N/2$ 点的 DFT。图 4.6 (b) 所示为 $N=8$ 时的分解过程图。

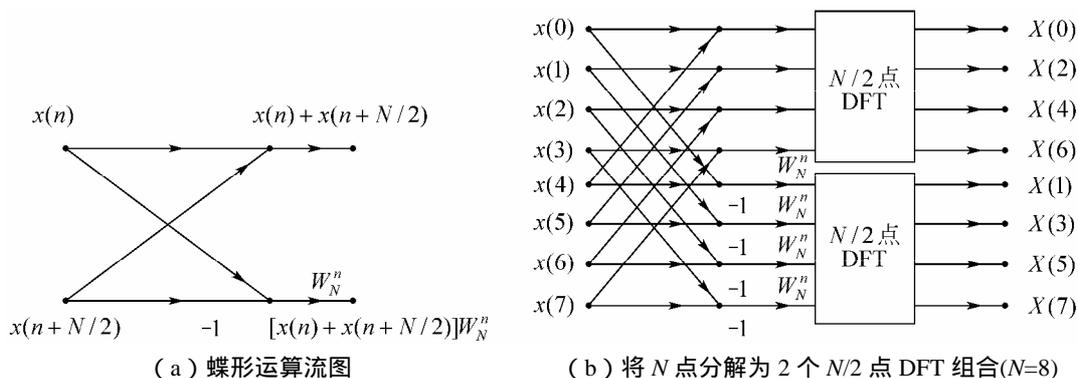


图 4.6 按频率抽取法的 FFT 算法

与时间抽取法的推导过程一样, 由于 $N=2^M$, $N/2$ 仍是一个偶数, 因此可以将 $N/2$ 点的 DFT 的输出再分解为偶数组与奇数组, 这就将 $N/2$ 点的 DFT 进一步分解为 2 个 $N/4$ 点的 DFT。这 2 个 $N/4$ 点 DFT 的输入也是将 $N/2$ 点 DFT 的输入上下对半分, 通过蝶形运算而形成, 其分解过程如图 4.7 所示。这样的分解过程可以一直进行下去, 直到分解 M 步后变成了 $N/2$ 个 2 点的 DFT 为止。而这 $N/2$ 个 2 点 DFT 的计算结果 (共 N 个值) 就是 $x(n)$ 的 N 点 DFT 的结果 $X(k)$ 。2 点 DFT 实际上只有加减法运算, 为了统一表示运算结果, 仍然采用系数为 W_n^0 蝶形运算来表示。一个 $N=8$ 时的完整的按频率抽取的 DFT 结构如图 4.8 所示。

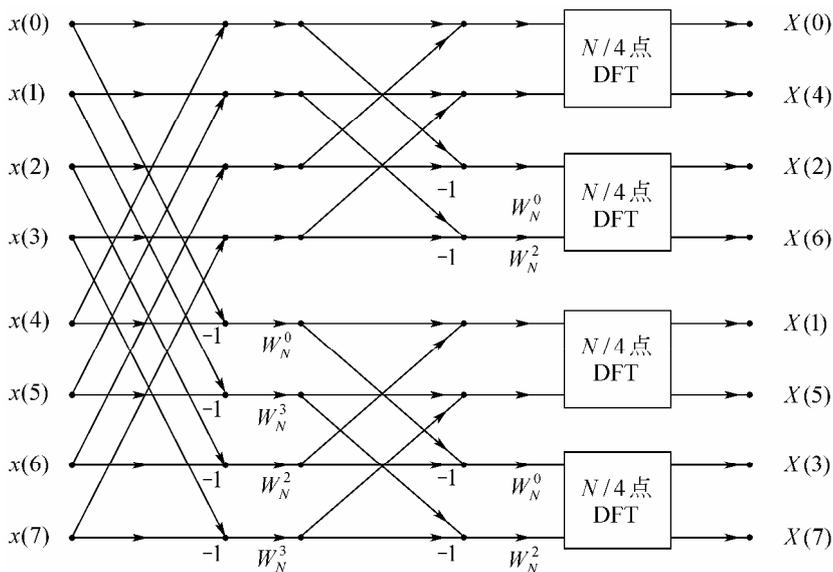
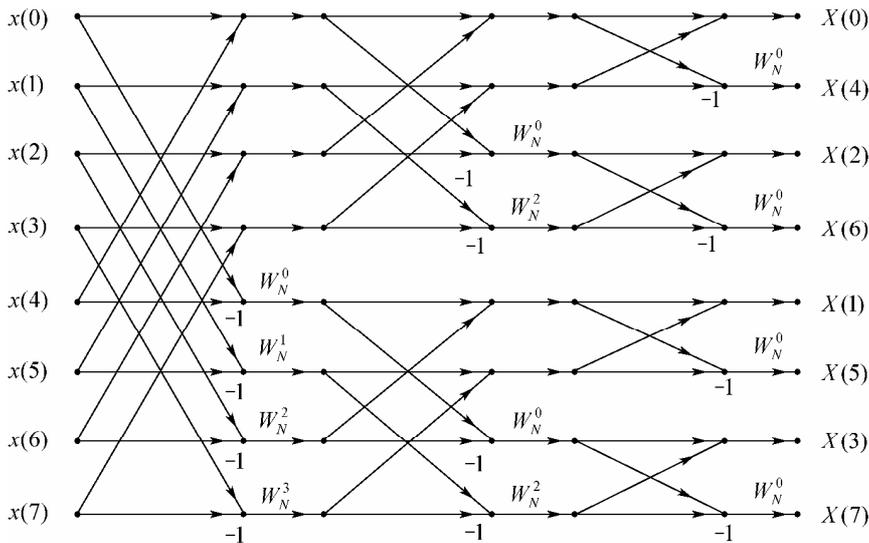


图 4.7 按频率抽取将 N 点分解为 $N/4$ 点的 FFT 算法

图 4.8 $N=8$ 按频率抽取 FFT 算法

4.3.2 DIT 算法与 DIF 算法的比较

1. DIT 算法与 DIF 算法的区别

由图 4.5 与图 4.8 可知，两种算法的差别在于：

- DIF 的输入是自然顺序，输出是倒位序的，DIT 情况正好相反。由于 DIF 输出是倒位序的，所以运算完成后，要需经过重排变为自然顺序输出，其重排的规律与按时间抽取法（DIT）相同。
- DIF 的蝶形运算与 DIT 的蝶形运算略有不同，其差别仅在于 DIF 中复数乘法出现于减法运算之后。

2. DIT 算法与 DIF 算法的相同之处

- 计算量和时间抽取法相同。由图 4.8 可见，频率抽取法共有 M 列运算，每列运算也需要 $N/2$ 个蝶形运算来完成。因此也需要 $m_F = \frac{MN}{2}$ 次复乘和 $a_F = NM$ 次复加。
- 两种运算均可作为原位运算。从图 4.8 可见，DIF 仍是一个原位运算。比较图 4.5 与图 4.8 发现二者互为转置。将图 4.5 反一个面并倒转信号流图的方向，交换输入与输出，即可从图 4.5 得到图 4.8。同理，也可通过转置从图 4.8 得到图 4.5。因此，对于每一种按时间抽取的 FFT 算法都存在一种按频率抽取的算法，二者互为转置。

4.4 离散傅里叶反变换的快速算法

按频率抽取的 FFT 算法同样可以用于离散傅里叶反变换 (IDFT), 称为快速傅里叶反变换 (简记为 IFFT)。IDFT 计算为

$$x(n) = \text{IDFT}[X(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk} \quad (4.23)$$

由式 (4.1) 和式 (4.23) 对比可知, 只要把 DFT 运算中的每一个系数 W_N^{nk} 换成 W_N^{-nk} , 最后再乘以常数 $1/N$, 则以上所有的按时间抽取或按频率抽取的 DFT 算法可以直接用来运算 IDFT。例如, 按照上述原则, 可以直接由按频率抽取的流图, 即图 4.8 出发, 把 W_N^{nk} 换成 W_N^{-nk} , 并在每列 (级) 运算中都乘因子 $\frac{1}{2}$ (因为乘 $\frac{1}{N}$ 等效于乘 $\frac{1}{N} = \frac{1}{2^M} = \left(\frac{1}{2}\right)^M$, 相当于每列都乘 $\frac{1}{2}$ 因子), 这样就得到图 4.9 所示的 IDFT 流图。当把时间抽取的 FFT 算法用于 IFFT 时, 由于输入变量由时间序列 $x(n)$ 变成了频率序列 $X(k)$, 因此原来按 $x(n)$ 的奇偶次序分组的时间抽取法 FFT, 现在变为按 $X(k)$ 的奇偶次序抽取的 IFFT, 故称之为频率抽取的 IFFT。同样, 频率抽取的 FFT 算法用于 IFFT 时, 称之为时间抽取的 IFFT。相应地, 当把时间抽取的 FFT 流图用于 IDFT 时, 得到的流图称为频率抽取的 IFFT 流图; 频率抽取的 FFT 流图用于 IDFT 时, 得到的流图称为时间抽取的 IDFT 流图。

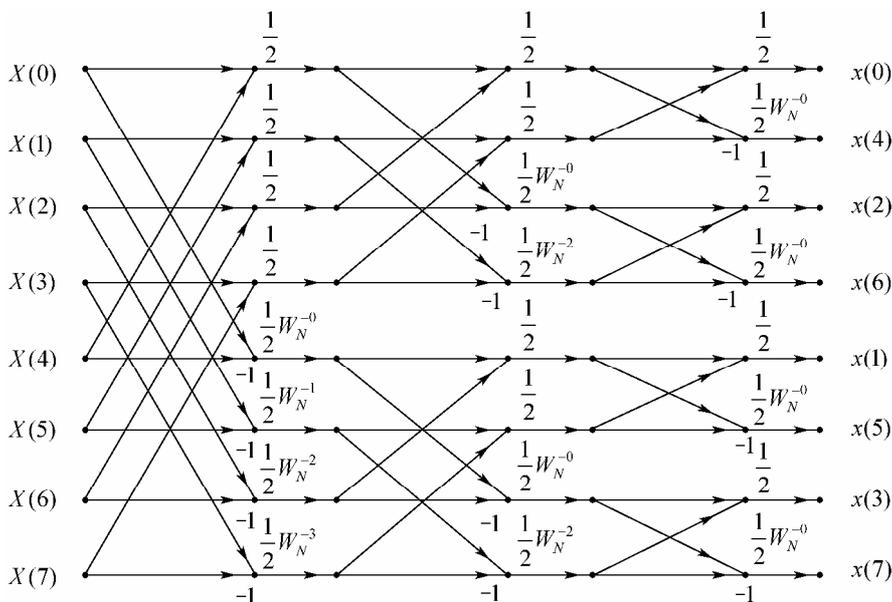


图 4.9 $N=8$ IFFT 信号流图

上面所叙述的 IFFT 算法, 虽然编程方便, 但需要变动 FFT 的程序和参数才能实现。然而, 利用 DFT 的性质, 可以直接用 FFT 程序来实现 IDFT 的运算。对式 (4.23) 取共

则有

$$x^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X^*(k) W_N^{nk}$$

故

$$x(n) = \frac{1}{N} \left[\sum_{k=0}^{N-1} X^*(k) W_N^{nk} \right]^* = \frac{1}{N} \{ \text{DFT} [X^*(k)] \}^* \quad (4.24)$$

此式表明：只要先将 $X(k)$ 取共轭，即可直接利用 FFT 程序，只需将运算结果取一次共轭，并乘以因子 $1/N$ ，即可得到 $x(n)$ 。

4.5 N 为复合数的 FFT 算法——混合基算法

对于长度为 $N = 2^M$ 的输入序列，可采用 DIT 基 2 的 FFT 算法或 DIF 基 2 FFT 算法。若输入序列的长度 $N \neq 2^M$ ，一般采用以下几种方式来处理：

(1) 在序列 $x(n)$ 后端补一些零值点使长度 N 增加到最邻近的一个 2^M 数值。例如当 $N = 28$ 时，可以在序列 $x(n)$ 后端补入 $x(28) = x(29) = x(30) = x(31) = 0$ ，共 4 个零值点，使 $N = 2^5 = 32$ 。由 DFT 的性质可知，有限长序列补零之后，并不影响其频谱 $X(e^{j\omega})$ ，只是频谱的样点也增加了，即由 28 点增加到 32 点。这种方法所造成的不利结果是增加了计算量，如果计算量增加太大，则会造成很大的浪费，故此方法不是最恰当的计算方法。

(2) 如果要求准确的 N 点 DFT，而 N 又是不能再分解成因子的素数，则只能采用直接的 DFT 算法，或者用后面将要介绍的 CZT 方法。

(3) 若 N 是一个复合数，即它可以分解成一些因子的复合数，则可以用 FFT 的一般算法，即任意基 FFT 算法，实际上基 2 算法只是这种算法的特例。

下面只讨论 N 为复合数的快速傅里叶变换算法。

4.5.1 算法原理

由式 (4.1) 可知， N 点的 $x(n)$ 的 DFT 为

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = 0, 1, \dots, N-1$$

设 N 是一个复合数，可表示为 $N = pq$ ，将序号 n 表示为

$$n = pn_1 + n_0, \quad n_1 = 0, 1, \dots, q-1, \quad n_0 = 0, 1, \dots, p-1 \quad (4.25)$$

即将原来信号的序号 n 表示为矩阵的形式，其中 n_0 为列序号， n_1 为行序列号， p 为总列数， q 为总行数。例如 $N = 12$ ， $p = 4$ ， $q = 3$ ，则可以把 12 个取样值的序号 n 排列成 3 行 4 列的矩阵。其中 $n = 6 = p + 2$ 的样点，位于矩阵中的第 1 行 ($n_1 = 1$) 第 2 列 ($n_0 = 2$) 处。

那么该序列中序号 n 可写为

$$n = 4n_1 + n_0, \quad n_1 = 0, 1, 2, \quad n_0 = 0, 1, 2, 3 \quad (4.26)$$

同理，对应的输出频谱序列的 k 值，也可用矩阵排列的形式表达为

$$k = pk_1 + k_0 \quad k_1 = 0, 1, \dots, p-1 \quad k_0 = 0, 1, \dots, q-1 \quad (4.27)$$

k_1 为变换后的列变量， k_0 为变换后的行变量。当按式 (4.27) 组合这些变量时，就得到单一的变量。同样，对于 $N=12$ ， $p=4$ ， $q=3$ 的情况有

$$k = 3k_1 + k_0 \quad k_1 = 0, 1, 2, 3 \quad k_0 = 0, 1, 2$$

这样通过 n_0, n_1, k_0, k_1 可组合成 n 及 k 的从 $0, 1, 2, \dots$ ，直到 $N-1$ 的各个数。

将式 (4.25) 式 (4.27) 代入式 (4.1) 可得

$$\begin{aligned} X(k) &= X(qk_1 + k_0) = X(k_1, k_0) \\ &= \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{n_0=0}^{p-1} \sum_{n_1=0}^{q-1} x(pn_1 + n_0)W_N^{(pn_1+n_0)(qk_1+k_0)} \\ &= \sum_{n_0=0}^{p-1} \sum_{n_1=0}^{q-1} x(n_1, n_0)W_N^{pn_1qk_1}W_N^{pn_1k_0}W_N^{n_0qk_1}W_N^{n_0k_0} \\ &= \sum_{n_0=0}^{p-1} \sum_{n_1=0}^{q-1} x(n_1, n_0)W_N^{pn_1k_0}W_N^{n_0qk_1}W_N^{n_0k_0} \end{aligned} \quad (4.28)$$

上面推导过程中应用了 $W_N^{pm_1qk_1} = W_N^{pm_1k_1} = 1$ 。为了计算出用 n_1, n_0 表示的 n 所有位，式 (4.28) 中的单个求和现在必须用对 n_1, n_0 两个求和号来代替。 $x(n_1, n_0)$ 表示 $x(n)$ 在 n 按 n_1, n_0 顺序排列时的情况， $X(k_1, k_0)$ 则表示 $X(k)$ 在 k 按 k_1, k_0 顺序排列时的情况。

式 (4.28) 可进一步表示为

$$\begin{aligned} X(k) &= X(k_1, k_0) = \sum_{n_0=0}^{p-1} \left[\sum_{n_1=0}^{q-1} x(n_1, n_0)W_N^{pn_1k_0} \right] W_N^{n_0qk_1}W_N^{n_0k_0} \\ &= \sum_{n_0=0}^{p-1} \left\{ \left[\sum_{n_1=0}^{q-1} x(n_1, n_0)W_q^{n_1k_0} \right] W_N^{n_0k_0} \right\} W_p^{n_0k_1} = \sum_{n_0=0}^{p-1} \{ X_1(k_0, n_0)W_N^{n_0k_0} \} W_p^{n_0k_1} \\ &= \sum_{n_0=0}^{p-1} X_1'(k_0, n_0)W_p^{n_0k_1} \\ &= X_2(k_0, k_1) \end{aligned} \quad (4.29)$$

式中

$$X_1(k_0, n_0) = \sum_{n_1=0}^{q-1} x(n_1, n_0)W_q^{n_1k_0}, \quad k_0 = 0, 1, 2, \dots, q-1$$

$$X_1'(k_0, n_0) = X_1(k_0, n_0)W_N^{n_0k_0}$$

$$X_2(k_0, k_1) = \sum_{n_0=0}^{p-1} X_1'(k_0, n_0)W_p^{n_0k_1}$$

式 (4.29) 表明一个 $N = pq$ 的序列的 DFT 可以用 p 组 q 点的 DFT 组成。

4.5.2 信号流图

以 $N=6=3\times 2$ 为例, 信号流图如图 4.10 所示。如果分解后的 p 和 q 仍是合数, 则仍可以继续分解, 直到全部分解为质因数为止, 这样将最大限度地减少计算量。

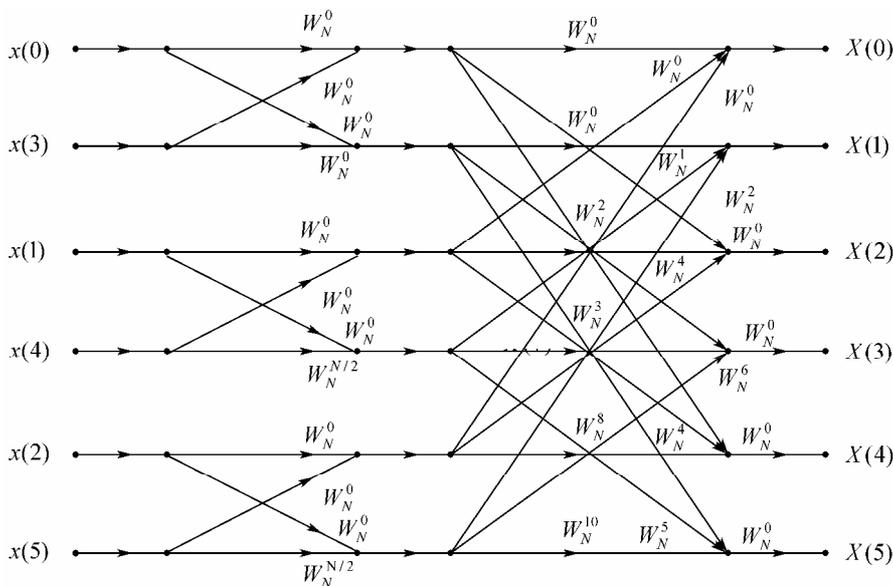


图 4.10 $N=6, p=3, q=2$ 的分解流图

4.6 线性调频 Z 变换

离散傅里叶变换可以看做信号在 Z 域上沿单位圆的均匀采样。但在实际应用中, 并非整个单位圆上的频谱都有意义。一些情况下, 如对于窄带信号, 只希望分析信号所在的一段频带, 采样点的轨迹是一条弧线或圆周。这种需求就导致了线性调频 Z 变换 (Chirp Z 变换——CZT) 的出现。

Chirp Z 变换是离散信号和系统分析与综合的重要工具, 其地位和作用犹如拉普拉斯变换对于连续信号和连续系统。与 DFT 计算整个频谱的算法不同, 它是一种更为灵活的计算频谱的算法, 可以用来计算单位圆上任一段弧线的 Z 变换, 做 DFT 时输入的点数 N 和输出的点数 M 可以不相等, 从而达到频域“细化”的目的。

4.6.1 CZT 的定义

设 $x(n)$ 为已知的时间信号, 由第 2 章中 Z 变换定义可得

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$

式中

$$z = e^{sT_s} = e^{(\alpha + j\Omega)T_s} = e^{\alpha T_s} e^{j\Omega T_s} = A e^{j\omega}$$

s 为拉普拉斯变量, $A = e^{\alpha T_s}$ 为实数, 圆频率 $\omega = \Omega T_s$ 为一角度。现对上式的 z 作一变换, 令

$$z_k = A W^{-k}$$

式中

$$A = A_0 e^{j\theta_0}, \quad W = W_0 e^{-j\varphi_0}$$

则

$$z_k = A_0 e^{j\theta_0} W_0^{-k} e^{j\varphi_0 k} \quad (4.30)$$

A_0, W_0 为任意的正实数, 给定 $A_0, W_0, \theta_0, \varphi_0$, 当 $k=0, 1, \dots, \infty$ 时, 便可得到 z 平面上的一个个点 $z_0, z_1, \dots, z_\infty$, 取这些点上的 Z 变换, 有

$$X(z_k) = \text{CZT}[x(n)] = \sum_{n=0}^{\infty} x(n) z_k^{-n} = \sum_{n=0}^{\infty} x(n) A^{-n} W^{nk} \quad (4.31)$$

这正是 CZT 的定义, 下面将对 $A_0, W_0, \theta_0, \varphi_0$ 含义进行解释。

由式 (4.30) 可知, 当 $k=0$ 时, $z_0 = A_0 e^{j\theta_0}$, 该点 (如图 4.11 中的 P 点) 在 z 平面上的幅度为 A_0 , 幅角为 θ_0 , 是 CZT 的起点。当 $k=1$ 时, $z_1 = A_0 W_0^{-1} e^{j(\theta_0 + \varphi_0)}$, z_1 点的幅度变为 $A_0 W_0^{-1}$, 角度在 θ_0 的基础上有增量 φ_0 。不难想象, 当随着 k 的变化, 点 z_0, z_1, z_2, \dots 构成了 CZT 变换的路径。因此, 对第 $M-1$ 点, 即 $Q = z_{M-1}$ 点, 该点的极坐标应是

$$Q = A_0 e^{j\theta_0} W_0^{-(M-1)} e^{j(M-1)\varphi_0} \quad (4.32)$$

如图 4.11 所示。这样, CZT 在 z 平面上的变换路径是一条螺旋线, 显然:

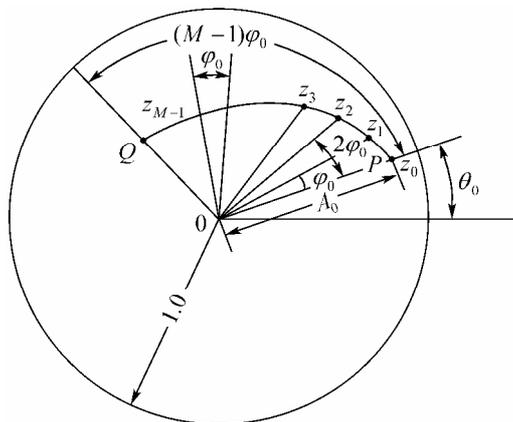


图 4.11 CZT 的变换路径

- (1) 当 $A_0 > 1$ 时, 螺旋线在单位圆外, 反之, 在单位圆内。
- (2) 当 $W_0 > 1$ 时, $A_0 W_0^{-1} < A_0$, 螺旋线内旋; 反之, 螺旋线外旋。
- (3) 当 $A_0 = W_0 = 1$ 时, CZT 的变换路径为单位圆上的一段圆弧, 起于 P 点, 终于 Q 点, P, Q 之间的分点 M 不一定等于数据的点数 N 。

(4) 当 $A_0 = W_0 = 1$, $\theta_0 = 0$, $M = N$ 时, CZT 变成了普通的 DFT。

为了得到的信号的频谱分析, 故应在单位圆上去实现 CZT, 而 A_0 、 W_0 都应取为 1。 $x(n)$ 的长度假定为 $n = 0, 1, L, N-1$, 变换的长度 $k = 0, 1, L, M-1$, 则有

$$X(z_k) = \sum_{n=0}^{N-1} x(n) A^{-n} W^{nk} \quad (4.33)$$

由于

$$nk = \frac{1}{2} [k^2 + n^2 - (k-n)^2]$$

所以式 (4.33) 可改写为

$$X(z_k) = \sum_{n=0}^{N-1} x(n) A^{-n} W^{\frac{k^2}{2}} W^{\frac{n^2}{2}} W^{-\frac{(k-n)^2}{2}} \quad (4.34)$$

令

$$\begin{cases} g(n) = x(n) A^{-n} W^{\frac{n^2}{2}} \\ h(n) = W^{-\frac{n^2}{2}} \end{cases} \quad (4.35)$$

故有

$$X(z_k) = W^{\frac{k^2}{2}} \sum_{n=0}^{N-1} g(n) h(k-n) = W^{\frac{k^2}{2}} [g(k) * h(k)] = W^{\frac{k^2}{2}} y(k) \quad (4.36)$$

式中

$$y(k) = g(k) * h(k) = \sum_{n=0}^{N-1} g(n) W^{-\frac{(k-n)^2}{2}}, \quad k = 0, 1, L, M-1 \quad (4.37)$$

这样就将式 (4.33) 转换成卷积形式, 从而可采用 FFT 算法, 就大大提高了运算速度。式 (4.36) 的计算可用图 4.12 所示的步骤来实现。

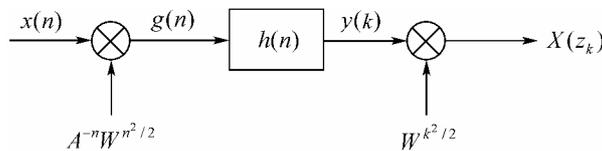


图 4.12 Chirp Z 变换运算流程

4.6.2 CZT 的计算方法

计算出单位圆上 M 点 $X(z_k)$ 的关键是实现式 (4.36) 中 $g(n)$ 和 $h(n)$ 的线性卷积。由式 (4.35) 可知, $A = e^{j\theta_0}$, $W = e^{-j\theta_0}$, 所以 $h(n) = W^{-\frac{n^2}{2}}$ 应是一无穷长的序列, 且是以 $n=0$ 为

偶对称的。同理, $A^{-n}W^{\frac{n^2}{2}}$ 也应是无穷长序列。但因为 $x(n)$ 是 N 点序列, 所以由式 (4.35) 可知, $g(n)$ 也应是 N 点序列, 即 $n=0, 1, L, N-1$ 。在式 (4.36) 中, 由于 n 的取值和 k 的取值范围的影响, 则 $h(n)$ 的取值是由 $n=-(N-1)$ 到 $n=M-1$, 其长度为 $N+M-1$ 。因此, $g(n)^*h(n)$ 的长度应为 $2N+M-2$ 。为了能用 FFT 的圆周卷积计算这个线性卷积且不产生混叠失真, 要求圆周卷积的长度应大于或等于 $2N+M-2$ 。但是, 由于只需要前 M 个 $X(z_k)$ 值 ($k=0, 1, L, M-1$), 其后的其他值是否有混叠失真对结果并无影响, 于是就可将圆周卷积的长度缩减到最小为 $N+M-1$ 。不过, 如果要进行基 2FFT 运算, 圆周卷积的长度应取 $L \geq N+M-1$, 同时又满足 $L=2^M$ 的最小 L 。为此, 可将 $h(n)$ 先补零值点到长度等于 L , 然后将此序列以 L 为周期进行周期延拓, 再取主值序列; 再将 $g(n)$ 补上零值点, 使之成为 L 点序列, 即可用 FFT 计算圆周卷积。

综上所述, 实现 CZT 运算的步骤归纳如下:

(1) 选择一个最小的整数 L , 使其满足 $L \geq N+M-1$, 同时又满足 $L=2^M$, 以便能用基 2FFT 算法求 $g(n)$ 和 $h(n)$ 的卷积。

(2) 根据已知的 $A_0, W_0, \theta_0, \phi_0$, 由式 $A=A_0e^{j\theta_0}, W=W_0e^{-j\phi_0}$ 求出 A, W , 则可算出 $A^{-n}W^{\frac{n^2}{2}}$, ($0 \leq n \leq N-1$) 的各值, 将这些数值与信号 $x(n)$, ($0 \leq n \leq N-1$) 相乘, 即得 $g(n)$, ($0 \leq n \leq N-1$)。将 $g(n)$ 补上零值点, 使其变为长度为 L 的序列。

(3) 利用 FFT 法求序列 $g(n)$ 的 L 点 DFT 结果 $G(r)$ 。

(4) 形成长度为 L 的序列 $h(n)$ 。在 $n=1$ 到 $M-1$ 一段取 $h(n)=W^{-\frac{n^2}{2}}$, 在 $n=M$ 到 $L-N$ 一段取为 $h(n)$ 任意值 (一般取为零), 在 $n=L-N+1$ 到 $L-1$ 一段取 $h(n)$ 为 $W^{\frac{n^2}{2}}$ 的周期延拓序列 $W^{\frac{(L-n)^2}{2}}$, 即为

$$h(n) = \begin{cases} W^{-\frac{n^2}{2}}, & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq L-M \\ W^{\frac{(L-n)^2}{2}}, & L-N+1 \leq n \leq L-1 \end{cases}$$

(5) 用 FFT 法求序列 $h(n)$ 的 L 点 DFT 结果 $H(r)$ 。

(6) 计算 $Q(r)=G(r)H(r)$, 再对序列 $Q(r)$ 做离散傅里叶反变换 (IDFF), 得 $q(k)$ 。

(7) 将 $q(k)$ 代入式 (4.36), 即得到所要求的 M 个 $X(z_k)$ 值。

上述计算过程如图 4.13 所示。

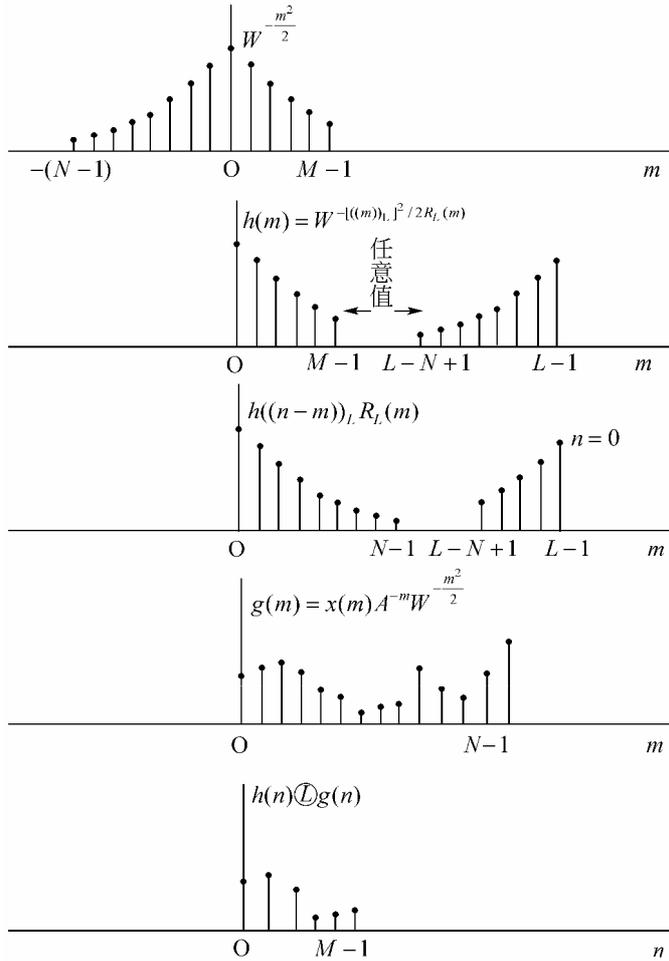


图 4.13 Chirp Z 变换计算

4.7 有关的 MATLAB 实现

4.7.1 与本章内容有关的 MATLAB 文件

MATLAB 为计算数据的离散快速傅里叶变换, 提供了一系列丰富的数学函数, 主要有 fft、ifft、fft2、ifft2 和 czt 等。当所处理的数据的长度为 2 的幂次时, 采用基 2 算法进行计算, 计算速度会显著增加。所以, 要尽可能使所要处理的数据长度为 2 幂次或者用添零的方法来添补数据使长度成为 2 的幂次。

1. fft 和 ifft 函数

调用格式是:

$$(1) Y = \text{fft}(X)$$

如果 X 是向量, 则采用傅里叶变换来求解 X 的离散傅里叶变换; 如果 X 是矩阵, 则计算该矩阵每一列的离散傅里叶变换; 如果 X 是 $N \times D$ 维数组, 则是对第一个非单元素的维进行离散傅里叶变换。

$$(2) Y = \text{fft}(X, N)$$

N 是进行离散傅里叶变换的 X 的数据长度, 可以通过对 X 进行补零或截取来实现。

$$(3) Y = \text{fft}(X, [], \text{dim}) \text{ 或 } Y = \text{fft}(X, N, \text{dim})$$

在参数 dim 指定的维上进行离散傅里叶变换; 当 X 为矩阵时, dim 用来指定变换的实施方向: $\text{dim}=1$, 表明变换按列进行; $\text{dim}=2$, 表明变换按行进行。

函数 ifft 的参数应用与函数 fft 完全相同。

2. fft2 和 ifft2 函数

调用格式是:

$$(1) Y = \text{fft2}(X)$$

如果 X 是向量, 则此傅里叶变换即变成一维傅里叶变换 fft ; 如果 X 是矩阵, 则是计算该矩阵的二维快速傅里叶变换; 数据二维傅里叶变换 $\text{fft2}(X)$ 相当于 $\text{fft}(\text{fft}(X)')$, 即先对 X 的列做一维傅里叶变换, 然后再对变换结果的行做一维傅里叶变换。

$$(2) Y = \text{fft2}(X, M, N)$$

通过对 X 进行补零或截断, 使得 X 成为 $M \times N$ 的矩阵。

函数 ifft2 的参数使用方法与函数 fft2 完全相同。

fftn 、 ifftn 是对数据进行多维快速傅立变换, 其应用与 fft2 、 ifft2 类似, 在此不再叙述。

3. czt 函数

调用格式是:

$$X = \text{czt}(X, M, W, A)$$

式中 X 是待变换的时域信号 $x(n)$, 其长度设为 N , M 是变换的长度, W 确定变换的步长, A 确定变换的起点。若 $M=N$, $A=1$, 则 CZT 变成 DFT。

4.7.2 与本章有关的 MATLAB 应用

例 4.1 设 $x(n)$ 是由两个正弦信号及白噪声的叠加, 试用 fft 函数对其做频谱分析。

解: 程序如下。

```
% 产生两个正弦加白噪声;
N=256;
f1=.1;f2=.2;fs=1;
a1=5;a2=3;
w=2*pi/fs;
x=a1*sin(w*f1*(0:N-1))+a2*sin(w*f2*(0:N-1))+randn(1,N);
% 应用 FFT 求频谱;
```

```

subplot(2,2,1);
plot(x(1:N/4));
title('原始信号');
f=-0.5:1/N:0.5-1/N;
X=fft(x);
y=ifft(X);
subplot(2,2,2);
plot(f,fftshift(abs(X)));
title('频域信号');
subplot(2,2,3);
plot(real(x(1:N/4)));
title('时域信号');

```

运行结果如图 4.14 所示，该程序同时完成了傅里叶变换与傅里叶反变换。

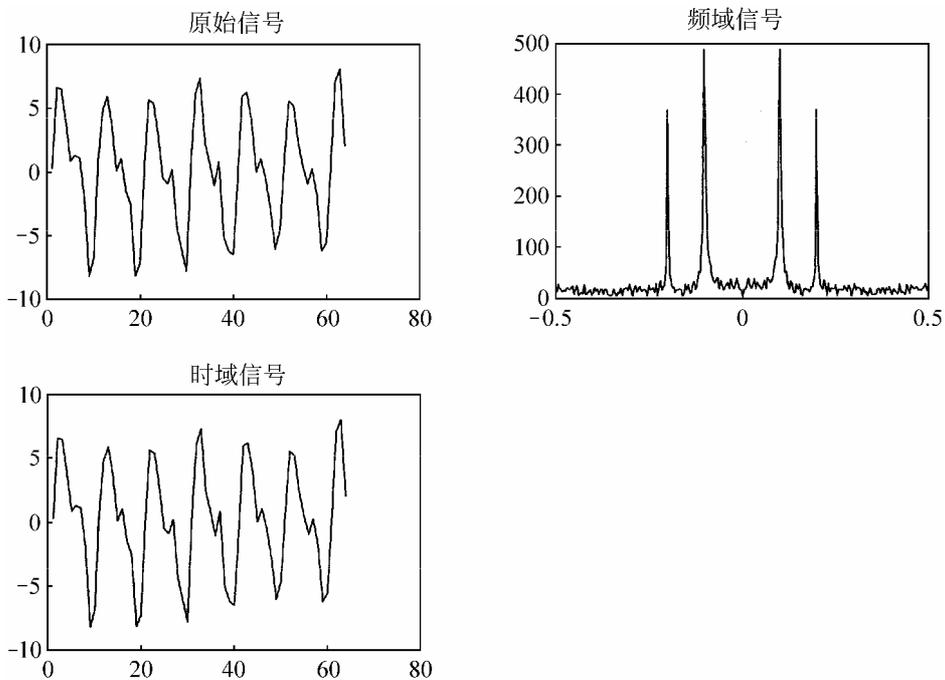


图 4.14 fft 应用实例

例 4.2 利用 Chirp Z 变换计算滤波器 h 在 100Hz~200Hz 的频率特性，并比较 CZT 和 FFT 函数。

解：程序如下。

```

h=fir1(30,125/500,boxcar(31));
Fs=1000;
f1=100;
f2=200;
m=1024;

```

```
w=exp(-j*2*pi*(f2-1)/(m*Fs));  
a=exp(j*2*pi*f1/Fs);  
y=fft(h,m);  
z=czt(h,m,w,a);  
fy=(0:length(y)-1)*Fs/length(y);  
fz=(0:length(z)-1)*(f2-f1)/length(z)+f1;  
subplot(2,1,1)  
plot(fy(1:500),abs(y(1:500)));  
title('fft');  
subplot(2,1,2)  
plot(fz,abs(z));  
title('czt');
```

运行结果如图 4.15 所示。

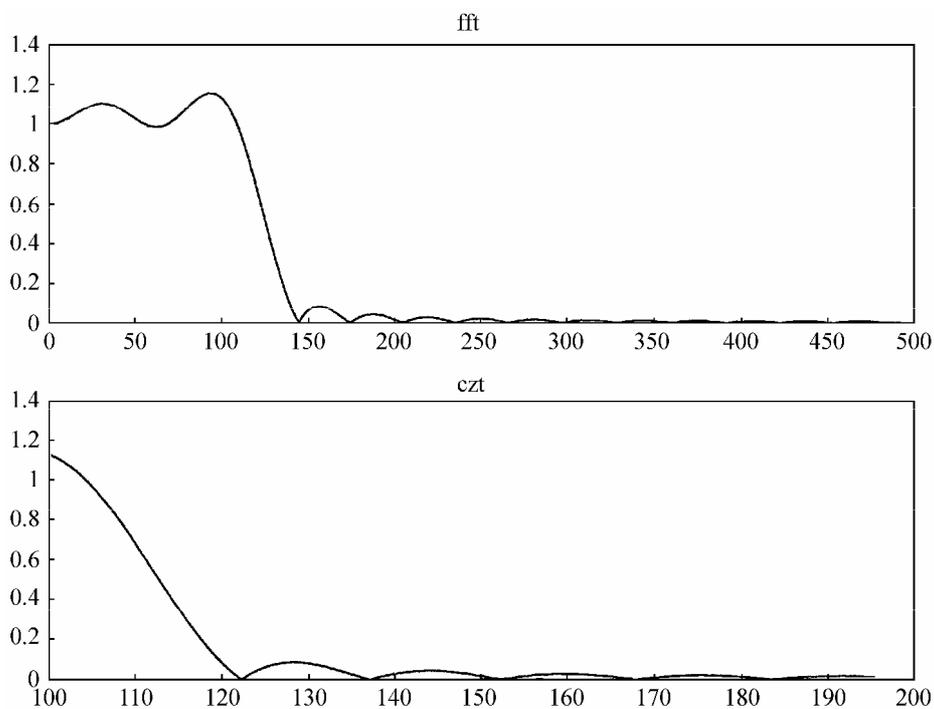


图 4.15 FFT 和 CZT 计算频率响应的结果图

第 5 章 数字滤波器的基本结构

在信号处理过程中，所处理的信号往往混有噪声，从接收到的信号中消除或减弱噪声是信号传输和处理中十分重要的问题。根据有用信号和噪声的不同特性，消除或减弱噪声，提取有用信号的过程称为滤波，实现滤波功能的系统称为滤波器。滤波器可以用各种标准来分类，按照信号的种类可分为模拟滤波器和数字滤波器，按照频带来分可分为低通滤波器、高通滤波器和带通滤波器及带阻滤波器等。在本章集中讨论数字滤波器的基本结构及其表示法，然后分别介绍无限冲击响应和有限冲击响应的网络结构。

5.1 数字滤波器的结构特点与表示方法

数字滤波器是数字信号处理的一个重要组成部分，数字滤波器实际上是一种运算过程。它是指完成频率选择或频率分辨任务的线性时不变系统的通用名称。因此，离散时间 LTI 系统也称为数字滤波器。其功能是将一组输入的数字序列通过一定的运算后转变为另一组输出的数字序列，因此它本身就是一台数字式的处理设备。与模拟滤波器类似，数字滤波器按频率特性划分也有低通、高通、带通、带阻、全通等类型。由于频率响应的周期性，频率变量以数字频率 ω 来表示， $\omega = \Omega T = \Omega / f_s$ ， f_s 为模拟角频率， T 为抽样时间间隔， Ω 为抽样频率，所以数字滤波器设计中必须给出抽样频率。

数字滤波器一般可以用两种方法实现：一种是设计专用的数字硬件、专用的数字信号处理器或采用通用的数字信号处理器来实现；另一种是直接利用计算机，将所需要的运算编成程序让计算机来执行，这也就是用软件来实现数字滤波器。

数字滤波器是离散时间系统，所处理的信号是离散时间信号。一般时域离散系统或网络可以用差分方程、单位脉冲响应以及系统函数进行描述。如果系统输入、输出服从 N 阶差分方程

$$y(n) = \sum_{k=0}^M b_k x(n-k) + \sum_{k=1}^N a_k y(n-k) \quad (5.1)$$

则其系统函数，即滤波器的传递函数为

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} = \frac{Y(z)}{X(z)} \quad (5.2)$$

由式 (5.1) 可知，实现一个数字滤波器需要如下几种基本的运算单元。

- 加法器：此元件有两个输入和一个输出，三个或多个信号的相加由相连的两输入加法器实现。

- 乘法器 (增益): 这是一个单入单出元件, 由于乘 1 时易于理解, 因此, 这里明确给出示意图。
- 延迟元件 (移位或记忆): 此元件把通过它的信号延迟一个样本, 其是由移位寄存器实现的。

这些基本的单元可采用两种方法描述——方框图法和信号流图法。对于同一个系统函数 $H(Z)$, 对输入信号的处理可实现的算法有很多种, 每一种算法对应于一种不同的运算结构 (网络结构)。因而一个数字滤波器的运算结构也有这样两种表示法, 如图 5.1 所示。

下面以一个二阶数字滤波器系统为例, 说明表示离散时间系统运算结构的方框图和信号流图方法。设此二阶数字系统的差分方程为

$$y(n] = a_1 y(n-1) + a_2 y(n-2) + b_0 x(n) + b_1 x(n-1) \quad (5.3)$$

则其系统函数为

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1} - a_2 z^{-2}} \quad (5.4)$$

由式 (5.3) 可得如图 5.2 所示的方框图。由图 5.2 可知实现离散时间系统的运算结构。如果系统是在一台通用计算机上用软件实现, 图 5.2 所示的网络结构可作为实现该系统的编程基础; 如果是用硬件实现, 该方框图给出了系统的硬件结构。

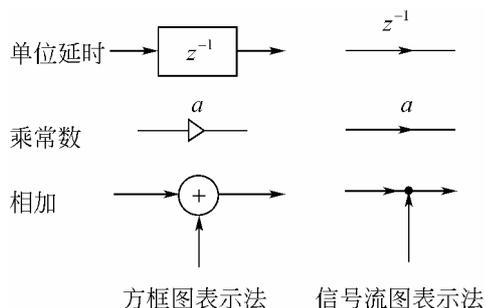


图 5.1 基本运算的方框图表示及流图表示

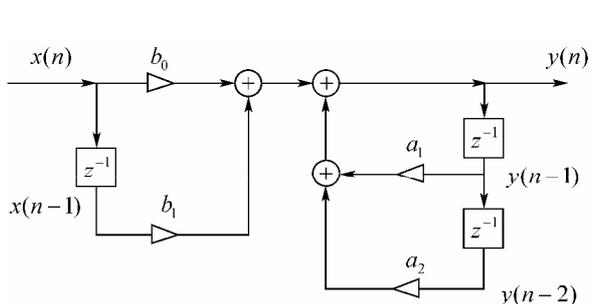


图 5.2 二阶数字滤波器的方框图

图 5.3 给出了式 (5.3) 表示的二阶数字滤波器的信号流图。在信号流图中, 基本运算单元形成的有向支路将节点连接成一个网络, 每一节点可以有几条输入支路和几条输出支路, 与各节点相关的节点变量值等于该节点的所有输入信号之和, 如果一个节点有两个以上的输入支路, 该节点就相当于一个加法器。输入信号 $x(n)$ 处的节点称为源节点, 输出信号 $y(n)$ 处的节点称为汇节点, 源节点没有输入支路, 汇节点只有输入支路。每一支路都有输入信号和输出信号, 输出信号值等于该支路的传输函数与输入信号 Z 变换的乘积。因此, 当支路传输函数为常数时, 输出信号值等于该支路起点处的节点信号值与常数的乘积; 如果支路上没有标明传输系数值, 则其传输系数为 1; 当支路传输函数为单位延迟 z^{-1} 时, 输

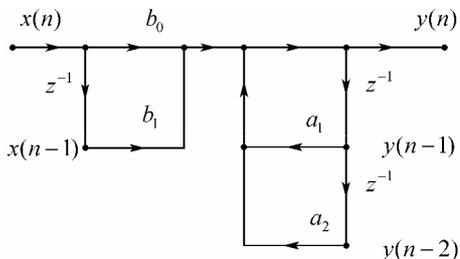


图 5.3 二阶数字滤波器的信号流图

出信号值等于该支路起点处的节点信号值延迟一个样本。

因此，运算结构是非常重要的，不同的结构会影响系统的精度、误差、稳定性、经济性以及运算速度等重要的性能。此外，在有限精度（有限字长）情况下，不同运算结构的误差、稳定性是不同的。数字滤波器有无限长单位脉冲响应（IIR）数字滤波器和有限长单位脉冲响应（FIR）数字滤波器两种。从结构上看，前者采用递归结构，后者采用非递归结构。两者各有不同的特点，下面将分别进行讨论。

5.2 无限长单位冲激响应滤波器的基本结构

如果一个 LTI 系统的脉冲响应具有无限长度，则此系统称为无限长脉冲响应（IIR）滤波器。无限长单位脉冲响应（IIR）滤波器有以下几个特点：

- 系统的单位冲激响应 $h(n)$ 是无限长的；
- 系统函数 $H(z)$ 在有限 z 平面 ($0 < |z| < \infty$) 有极点存在；
- 结构上存在着输出到输入的反馈，也就是结构上是递归型的。

同一种系统函数 $H(z)$ 可以有多种不同的结构，其基本网络结构有直接型、级联型和并联型三种，其中，直接型又可分为直接 I 型和直接 II 型两种。

5.2.1 直接 I 型 IIR 滤波器

一个 N 阶的 IIR 滤波器的输入输出关系可以用式 (5.1) 所示的 N 阶的差分方程来描述。把式 (5.1) 重写如下：

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

这就表示了一种计算方法。右端第一部分 $\sum_{k=0}^M b_k x(n-k)$ 表示将输入及延时后的输入，组成 M

节的延时网络，把每节延时抽头后加权（加权系数 b_k ），

然后把结果相加，就构成一个横向结构网络；第二部分

$\sum_{k=0}^N a_k y(n-k)$ 表示将输出加以延时，组成 N 节的延

时网络，再将每节延时抽头后加权（加权系数是

a_k ），然后把结果相加，最后的输出 $y(n)$ 是把这两个和

式相加的和。由于包含了输出的延时部分，故是一个

有反馈的网络。由式 (5.1) 右端的第一个和式构成了

反馈网络。这种结构称为直接 I 型结构，其结构流图

如图 5.4 所示。由图 5.4 可看出，总的网络是由上面讨

论的两部分网络级联组成，第一个网络实现零点，第

二个网络实现极点。另外还可看出，直接 I 型结构需要 $N+M$ 级延时单元。

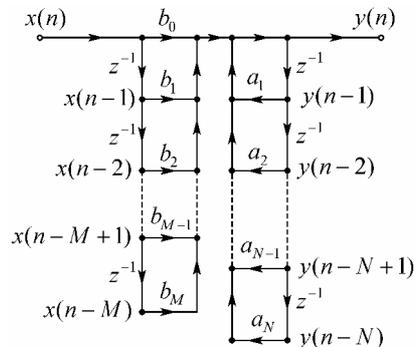


图 5.4 直接 I 型结构 IIR 滤波器

5.2.2 直接 II 型 IIR 滤波器

一个线性时不变系统，若交换其级联子系统的次序，系统函数是不变的，也就是说总的输入输出关系不改变。

直接 II 型 IIR 滤波器结构又称为正准型结构。图 5.4 直接 I 型 IIR 滤波器结构的系统函数 $H(z)$ 也可以看成是独立的系统函数的乘积。输入信号 $x(n)$ 先通过系统 $H_1(z)$ ，得到中间输出变量 $y_1(n)$ ；通过系统 $H_2(z)$ ，得到输出信号 $y(n)$ ，即可得到另一种结构（如图 5.5 所示）。它包含了两个级联子网络，第一个实现系统函数的极点，第二个实现系统函数的零点。两行串行延时支路有相同的输入，因而可以把它们合并，得到图 5.6 的结构，称为直接 II 型结构或典范型结构。

比较图 5.4 和图 5.6 可知：直接 II 型比直接 I 型 IIR 滤波器的结构延时单元少，用硬件实现可以节省寄存器，比直接 I 型经济；若用软件实现则可节省存储单元。但是它们都是直接性的实现方法，其共同的缺点是系数 a_k 、 b_k 对滤波器的性能控制作用不明显，这是因为它们与系统函数的零、极点关系不明显，因而调整困难；此外，这种结构极点对系数的变化过于灵敏，从而使系统频率响应对系数的变化过于灵敏，也就是对有限精度（有限字长）运算过于灵敏，容易出现不稳定或产生较大误差。高阶系统直接型结构都存在调整零、极点困难，对系数量化效应敏感度高缺点。

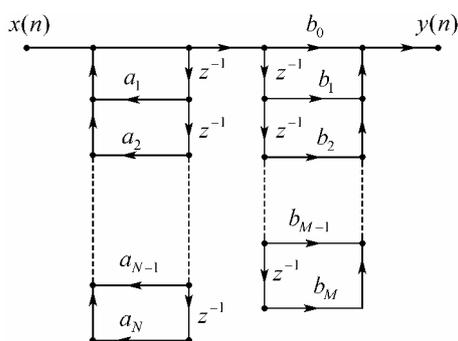


图 5.5 直接 I 型结构滤波器的变型
(网络的零点与极点的级联次序互换)

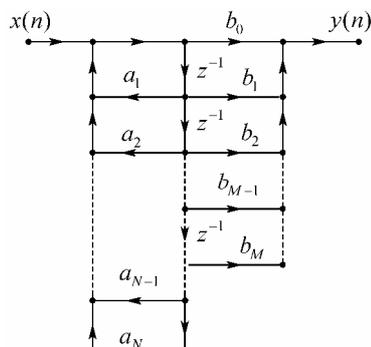


图 5.6 直接 II 型结构典范型结构

5.2.3 级联型结构

若把式 (5.2) 描述的 N 阶 IIR 滤波器的函数将 $H(z)$ 的分子和分母分别进行因式分解，则可表示为

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} = A \frac{\prod_{k=1}^M (1 - C_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad (5.5)$$

式中， A 为常数， C_k 和 d_k 分别表示 $H(z)$ 的零点和极点。由于 $H(z)$ 分子、分母是实系数多

项式，而实系数多项式的根只有实根和共轭复根两种情况。将每一对共轭零点（极点）合并起来构成一个实系数的二阶因子，并把单个的实根因子看成是二次项系数等于零的二阶因子，则可以把 $H(z)$ 表示成多个实系数的二阶数字网络 $H_k(z)$ 的连乘积形式，如式 (5.6) 所示。

$$H(z) = A \prod_k \frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 - \alpha_{1k} z^{-1} - \alpha_{2k} z^{-2}} = A \prod_k H_k(z) \quad (5.6)$$

若每一个实系数的二阶数字网络的系数函数 $H_k(z)$ 的网络结构均采用前面介绍的直接 II 型结构，则可以得到系统函数 $H(z)$ 的级联型结构，如图 5.7 所示。

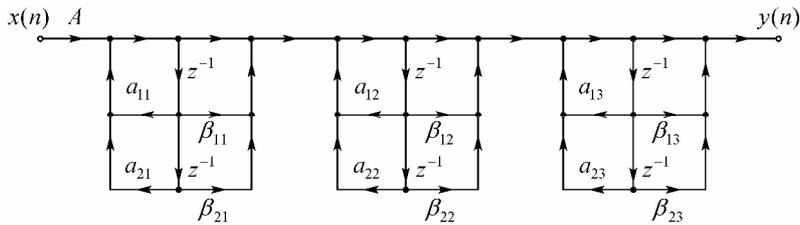


图 5.7 级联型结构 IIR 滤波器

在级联型结构中，每个二阶网络只关系到滤波器的一个零点、一个极点；每个二阶网络只关系到滤波器的一对共轭零点和一对共轭极点。调整系数 β_{1j} 、 β_{2j} 只影响滤波器的第 j 对零点，对其他零点并无影响；同样，调整分母多项式的系数 α_{1j} 、 α_{2j} 也只单独调整了第 j 对极点。所以，这种结构，便于准确实现滤波器频率响应的性能。因此，与直接型结构相比，级联型结构便于准确地实现滤波器零、极点的调整。

此外，级联各节之间要有电平的放大和缩小，以使变量值不会太大或太小。不能太大是为了避免在定点运算中产生溢出现象；不能太小是为了防止信号与噪声的比值太小，这将在以后讨论。另外，级联结构具有最少的存储器。

5.2.4 并联型结构

将传递函数 $H(z)$ 展成部分分式的形式，就得到并联型 IIR 滤波器的基本结构，即表示为若干一阶和二阶基本节网络与一个常数 B_0 之和

$$H(z) = B_0 + \sum_{k=1}^{M_1} \frac{B_k}{1 - p_k z^{-1}} + \sum_{k=1}^{M_2} \frac{b_{0k} + b_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \quad (5.7)$$

其中 $M_1 + 2M_2 = N$ ，同样也可以统一表示为二阶基本节的形式

$$H(z) = B_0 + \sum_{k=1}^N \frac{b_{0k} + b_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} = B_0 + \sum_{k=1}^N H_k(z) \quad (5.8)$$

并联型结构信号如图 5.8 所示，其中二阶基本节网络可以用直接 II 型结构实现。并联型结构也可以单独调整极点位置，但却不能像级联型结构那样直接控制零点的分布，因为并联型结构各二阶基本节网络的零点并不是整个系统函数的零点。因此，当要准确控制零

点时，以采用级联型结构为宜。不过，由于并联型基本节之间互不影响，所以运算误差比级联型的要小一些。

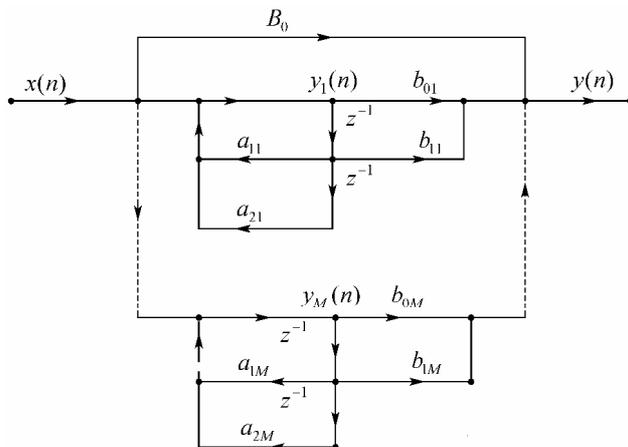


图 5.8 并联型结构 IIR 滤波器

传输函数为 $H(z)$ 的滤波器的频率特性为 $H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$ 。由于实际的滤波器都是稳定系统，其所有极点都在 z 平面的单位圆内，单位抽样响应 $h(n)$ 是一个无限长的衰减序列，所以要截取有限长的一段来做频率特性分析。截取的长度越长，逼近的程度越高。

5.3 有限长单位冲激响应滤波器的结构

如果一个 LTI 系统的单位脉冲响应长度有限，则此系统称为有限长度脉冲响应（或 FIR）滤波器。FIR 数字滤波器的单位脉冲响应 $h(n)$ 是一个有限序列，其滤波器的结构采用非递归结构。

有限长单位冲激响应滤波器有以下几个特点：

- 系统的单位冲激响应 $h(n)$ 在有限个 n 个值处不为零。
- 系统函数 $H(z)$ 在 $|z|>0$ 处收敛， $|z|>0$ 处只有零点，有限 z 平面只有零点，而全部极点都在 $z=0$ 处（因果系统）。
- 结构上主要是非递归结构，没有输出到输入的反馈，但有些结构中（例如频率抽样结构）也包含有反馈的递归部分。

设 FIR 数字滤波器的单位脉冲响应 $h(n)$ 为一个 N 点序列， $0 \leq n \leq N-1$ ，则滤波器系统函数为

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \quad (5.9)$$

就是说，它有 $(N-1)$ 阶极点在 $z=0$ 处，有 $(N-1)$ 个零点位于有限 z 平面的任何位置。

基本网络结构有四种：直接型、级联型、频率采样型和快速卷积型。

5.3.1 直接型

式 (5.9) 系统的差分方程表达式为

$$y(n) = \sum_{m=0}^{N-1} h(m)x(n-m) \quad (5.10)$$

很明显, 这就是线性移不变系统的卷积和公式, 也是 $x(n)$ 的延时链的横向结构 (如图 5.9 所示), 由于该结构利用输入信号 $x(n)$ 和滤波器单位脉冲响应 $h(n)$ 的线性卷积来描述输出信号 $y(n)$, 称为横截型结构或卷积型结构, 也可称为直接型结构。

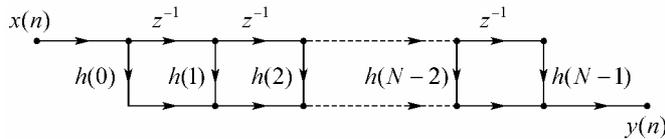


图 5.9 FIR 滤波器的直接型结构

5.3.2 级联型

为了得到 FIR 系统的级联结构, 将传递函数 $H(z)$ 分解成二阶实系数因子的形式:

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} = \prod_{k=1}^{\lfloor \frac{N}{2} \rfloor} (\beta_{0k} + \beta_{1k}z^{-1} + \beta_{2k}z^{-2}) \quad (5.11)$$

由式 (5.11) 可得到 FIR 系统的级联结构 (如图 5.10 所示)。级联结构中每一基本节控制一对零点, 所用的系数乘法次数比直接型多, 运算时间较直接型长。

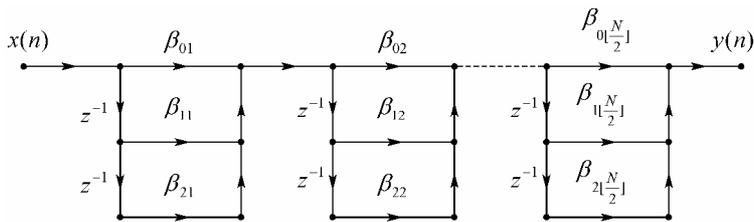


图 5.10 FIR 滤波器的级联型结构 (N 为奇数)

5.3.3 频率采样型

由频率采样定理可知, 对有限长序列 $h(n)$ 的 Z 变换 $H(z)$ 在单位圆上做 N 点的等间隔采样, N 个频率采样值的离散傅里叶反变换所对应的时域信号 $h_N(n)$ 是原序列 $h(n)$ 以采样点数 N 为周期进行周期延拓的结果, 当 N 大于等于原序列 $h(n)$ 长度 M 时, $h_N(n) = h(n)$, 不会发生信号失真, 此时 $H(z)$ 可以用频域采样序列 $H(R)$ 内插得到, 内插公式如下:

$$H(z) = (1 - z^{-N}) \frac{1}{N} \sum_{k=0}^{N-1} \frac{H(k)}{1 - W_N^{-k} z^{-1}} \quad (5.12)$$

则有

$$H(z) = \frac{1}{N} H_c(z) \sum_{k=0}^{N-1} H'_k(z) \quad (5.13)$$

这个公式为 FIR 滤波器提供了另外一种结构, 这种结构由两部分级联组成。显然, $H(z)$ 的第一部分 $H_c(z)$ $H_c(z) = 1 - e^{-N}$, 其频率响应为

$$H_c(e^{j\omega}) = 1 - e^{-j\omega N} = 2 \sin\left(\frac{N}{2}\omega\right) e^{-j\omega \frac{N}{2}}$$

即频率特性是一个由 N 阶延时单元组成的梳状滤波器 (如图 5.11 所示)。它在单位圆上有 N 个等间隔的零点。第二部分由 N 个一阶网络 $H'_k(z)$ 组成的并联结构, 每个一阶网络在单位圆上有一个极点, 因此, $H(z)$ 的第二部分是一个有 N 个极点的谐振网络。这些极点正好与第一部分梳状滤波器的 N 个零点相抵消, 从而使 $H(z)$ 在这些频率上的响应等于 $H(k)$ 。 N 个并联谐振器与梳状滤波器级联后, 就得到图 5.12 所示的频率采样结构。频率采样结构的特点是它的系数 $H(k)$ 就是滤波器在 $\omega = \frac{2\pi}{N}k$ 处的响应, 因此, 控制滤波器的频率响应很方便。

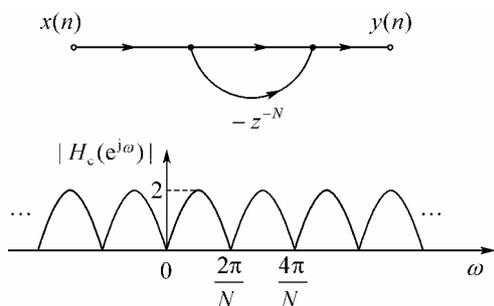


图 5.11 梳状滤波器结构及频率响应幅度

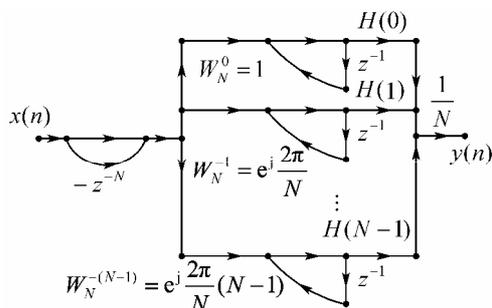


图 5.12 FIR 滤波器的频率采样型结构

FIR 滤波器的频率采样型结构的主要优点: 首先, 它的系数 $H(k)$ 直接就是滤波器在 $\omega = \frac{2\pi}{N}k$ 处的响应值, 因此可以直接控制滤波器的响应。此外, 只要滤波器的 N 阶数相同, 对于任何频响形状, 其梳状滤波器部分的结构完成相同, 只是各支路的增益 $H(k)$ 不同, 因此频率采样型结构便于标准化、模块化。但是该结构也有两个缺点:

- 该滤波器所有的系数 $H(k)$ 和 $-W_N^{-k}$ 一般为复数, 增加了乘法次数和存储量, 复数相乘运算实现起来较麻烦。
- 系统稳定是靠位于单位圆上的 N 个零极点对消来保证的, 如果滤波器的系数稍有误差, 极点就可能移到单位圆外, 造成零极点不能完全对消, 影响系统的稳定性。

5.3.4 快速卷积型

根据圆周卷积和线性卷积的关系可知,两个长度为 N 的序列的线性卷积,可以用这两个序列的 $2N-1$ 点的圆周卷积来实现。由 FIR 滤波器的直接型结构:滤波器的输出信号 $y(n)$ 是输入信号 $x(n)$ 和滤波器单位脉冲响应 $h(n)$ 的线性卷积。所以,对有限长序列 $x(n)$,可以通过补零的方法延长 $x(n)$ 和 $h(n)$ 序列,然后计算它们的圆周卷积,从而得到 FIR 系统的输出 $y(n)$ 。利用圆周卷积定理,采用 FFT 实现有限长序列 $x(n)$ 和 $h(n)$ 的线性卷积,则可得 FIR 滤波器的快速卷积结构,如图 5.13 所示,图中 $L \geq N+M-1$, M 为 $x(n)$ 的长度, N 为 $h(n)$ 的长度。

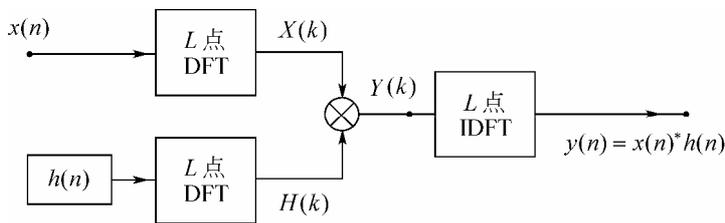


图 5.13 FIR 滤波器的快速卷积结构

对 $x(n)$ 为无限长的一般情况,可用重叠相加法或重叠保留法实现 FIR 滤波器的快速卷积结构。

5.4 数字滤波器的 MATLAB 实现

为了描述 IIR 和 FIR 的各种结构,实现这些结构,需要对多项式进行运算,因此 MATLAB 是一个很方便的工具。

5.4.1 IIR 滤波器的 MATLAB 实现

1. 直接形式

在 MATLAB 中,用 filter 函数实现 IIR 的直接形式。

例 5.1 用直接型实现系统函数为

$$H(z) = \frac{1 - 3z^{-1} + 11z^{-2} + 27z^{-3} + 18z^{-4}}{1 + 16z^{-1} + 12z^{-2} + 2z^{-3} - 4z^{-4} - z^{-5}}$$

的 IIR 数字滤波器,求单位脉冲响应和单位阶跃信号的输出。

解:程序如下。

```

b=[1,-3,11,27,18];a=[16,12,2,-4,-1];
N=25;
h=impz(b,a,N); %直接型单位脉冲响应
x=[ones(1,5),zeros(1,N-5)]; %单位阶跃信号
y=filter(b,a,x);%直接型输出信号
subplot(1,2,1);stem(h);title('直接型 h(n)');
subplot(1,2,2);stem(y);title('直接型 y(n)');

```

运行结果如图 5.14 所示。

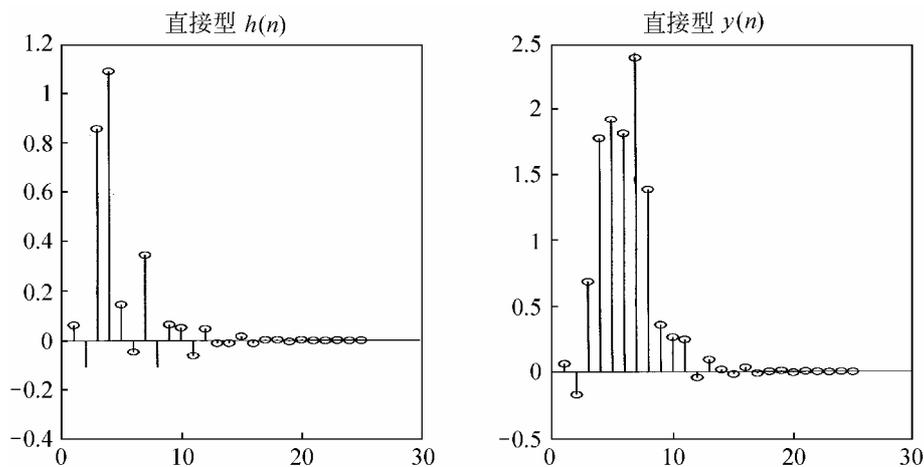


图 5.14 直接型单位脉冲响应和输出信号

2. 级联形式

在 MATLAB 中给定级联型系统函数，由扩展函数 casfilter 实现 IIR 的级联形式。

例 5.2 用级联型结构实现系统函数

$$H(z) = \frac{4(1+z^{-1})(1-1.4142136z^{-1}+z^{-2})}{(1-0.5z^{-1})(1+0.9z^{-1}+0.81z^{-2})}$$

解：程序如下。

```

b0=4;B=[1,1,0;1,-1.4142136,1];A=[1,-0.5,0;1,0.9,0.81];
N=60;
delta=impseq(0,0,N);
h=casfiltr(b0,B,A,delta);
x=[ones(1,5),zeros(1,N-5)];
y=casfilter(b0,B,A,x);
subplot(1,2,1);stem(h);title('直接型 h(n)');
subplot(1,2,2);stem(y);title('直接型 y(n)');

```

运行结果如图 5.15 所示。

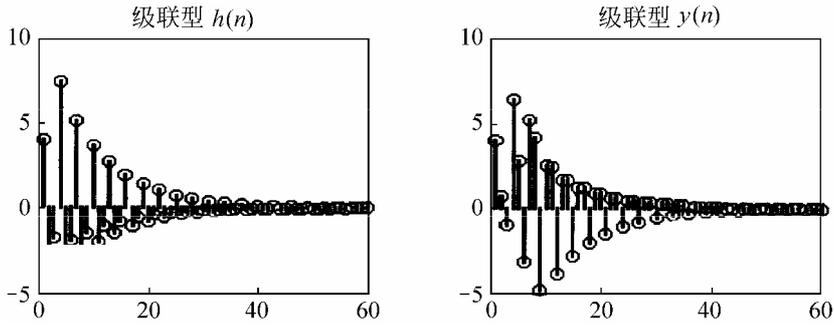


图 5.15 级联型单位脉冲响应和输出信号

若给定直接形式滤波器的系数 a_k 、 b_i ，可由扩展函数 `dir2cas` 计算出 b_0 、 $B_{k,i}$ 、 $A_{k,i}$ ，然后由扩展函数 `casfilter` 实现 IIR 的级联形式。

例 5.3 用级联型实现系统函数为

$$H(z) = \frac{1 - 3z^{-1} + 11z^{-2} + 27z^{-3} + 18z^{-4}}{1 + 16z^{-1} + 12z^{-2} + 2z^{-3} - 4z^{-4} - z^{-5}}$$

的 IIR 数字滤波器，求单位脉冲响应和单位阶跃信号的输出。

解：程序如下。

```
N=25;
Delta=impseq(0,0,N);
b=[1,-3,11,27,18];
a=[16,12,2,-4,-1];
[b0,B,A]=dir2cas(b,a);%直接型转换成级联型
h=casfilttr(b0,B,A,delta); %级联型单位脉冲响应
x=[ones(1,5),zeros(1,N-5)];
y=casfilter(b0,B,A,x);%级联型输出信号
subplot(1,2,1);stem(h);title('直接型 h(n)');
subplot(1,2,2);stem(y);title('直接型 y(n)');
```

运行结果如图 5.16 所示。

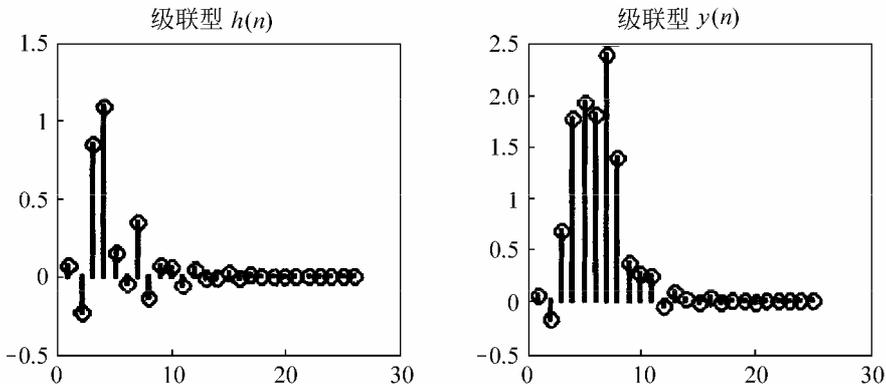


图 5.16 级联型单位脉冲响应和输出信号

同理,若给定系统函数为级联形式,也可由扩展函数 `cas2dir` 将其转换成直接形式,再由 `filter` 函数实现 IIR 的直接形式。

例 5.4 用直接型结构实现系统函数

$$H(z) = \frac{4(1+z^{-1})(1-1.4142136z^{-1}+z^{-2})}{(1-0.5z^{-1})(1+0.9z^{-1}+0.81z^{-2})}$$

的 IIR 数字滤波器,求单位脉冲响应和单位阶跃信号的输出。

解:程序如下。

```
b0=4;B=[1,1,0;1,-1.4142136,1];A=[1,-0.5,0;1,0.9,0.81];
N=60;
delta=impzseq(0,0,N);
x=[ones(1,5),zeros(1,N-5)];
[b,a]=cas2dir(b0,B,A);
h=filter(b,a,delta);%直接型单位脉冲响应
y=filter(b,a,x);%直接型输出信号
subplot(1,2,1);stem(h);title('直接型 h(n)');
subplot(1,2,2);stem(y);title('直接型 y(n)');
```

运行结果如图 5.17 所示。

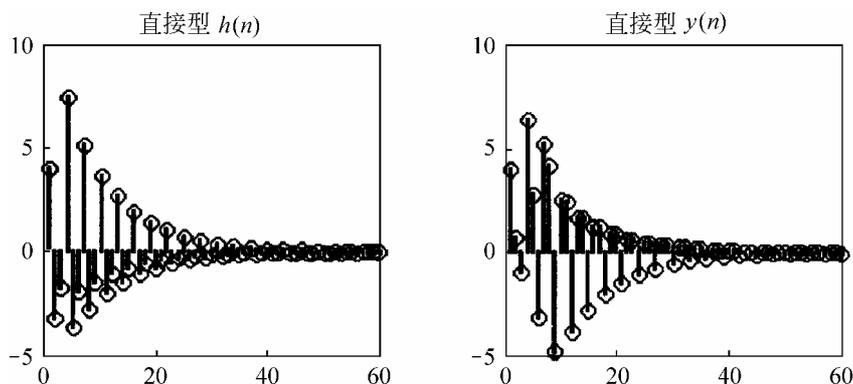


图 5.17 直接型单位脉冲响应和输出信号

3. 并联形式

在 MATLAB 中用给定并联系统函数,由扩展函数 `parfiltr` 实现 IIR 的并联形式。

例 5.5 用并联型实现系统函数为

$$H(z) = \frac{-14.75 - 12.90z^{-1}}{1 - \frac{7}{8}z^{-1} + \frac{3}{32}z^{-2}} + \frac{24.50 + 26.82z^{-1}}{1 - z^{-1} + \frac{1}{2}z^{-2}}$$

的滤波器。

解:程序如下。

```

C=0; B=[-14.75,-12.90;24.50,26.82]; A=[1,-7/8,3/32;1,-1,0.5];
N=60;
delta=impzseq(0,0,N);
h=parfiltr(C,B,A,delta);
x=[ones(1,5),zeros(1,N-5)];
y=parfilter(C,B,A,x);
subplot(1,2,1);stem(h);title('并联型 h(n)');
subplot(1,2,2);stem(y);title('并联型 y(n)');

```

运行结果如图 5.18 所示。

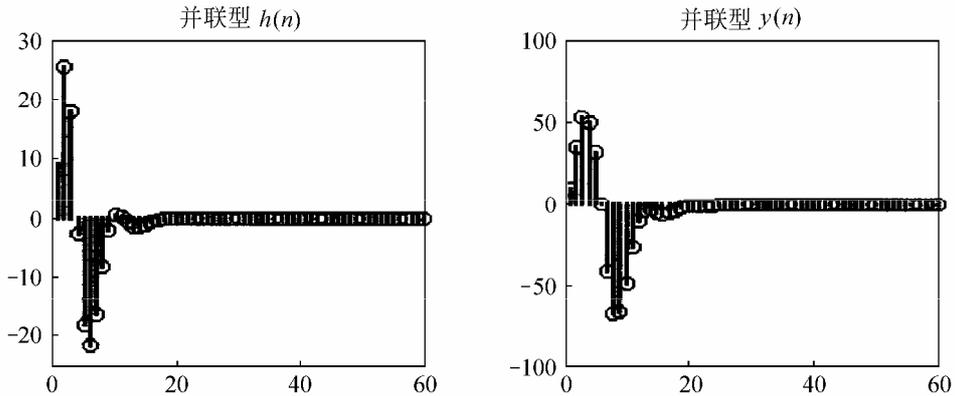


图 5.18 并联型单位脉冲响应和输出信号

若给定直接型滤波器的 a_k 、 b_i ，可由扩展函数 `dir2par` 计算出 b_0 、 $B_{k,i}$ 、 $A_{k,i}$ ，然后由扩展函数 `parfiltr` 实现 IIR 的并联形式。

例 5.6 用并联型实现系统函数为

$$H(z) = \frac{1 - 3z^{-1} + 11z^{-2} + 27z^{-3} + 18z^{-4}}{1 + 16z^{-1} + 12z^{-2} + 2z^{-3} - 4z^{-4} - z^{-5}}$$

的 IIR 数字滤波器，求单位脉冲响应和单位阶跃信号的输出。

解：程序如下。

```

b=[1,-3,11,27,18]; a=[16,12,2,-4,-1];
N=25;
delta=impzseq(0,0,N);
[C,B,A]=dir2par(b,a);
h=parfiltr(C,B,A,delta);
x=[ones(1,5),zeros(1,N-5)];
y=parfilter(C,B,A,x);
subplot(1,2,1);stem(h);title('并联型 h(n)');
subplot(1,2,2);stem(y);title('并联型 y(n)');

```

运行结果如图 5.19 所示。

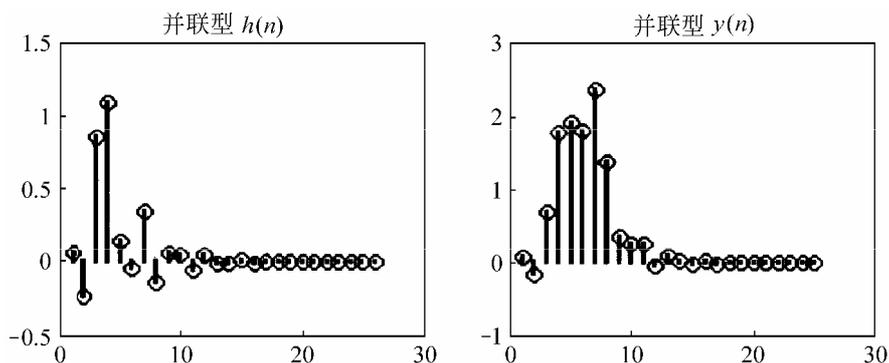


图 5.19 并联型单位脉冲响应和输出信号

若给定系统函数为并联形式时，也可由扩展函数 `par2dir` 将其转换成直接形式，再由 `filter` 直接形式实现。

例 5.7 用直接型实现系统函数为

$$H(z) = \frac{-14.75 - 12.90z^{-1}}{1 - \frac{7}{8}z^{-1} + \frac{3}{32}z^{-2}} + \frac{24.50 + 26.82z^{-1}}{1 - z^{-1} + \frac{1}{2}z^{-2}}$$

的滤波器。

解：程序如下。

```
C=0; B=[-14.75,-12.90;24.50,26.82]; A=[1,-7/8,3/32;1,-1,0.5];
N=60;
delta=impseq(0,0,N);
[b,a]=par2dir(C,B,A);
h=filter(b,a,delta);
x=[ones(1,5),zeros(1,N-5)];
y=filter(b,a,x);
subplot(1,2,1);stem(h);title('直接型 h(n)');
subplot(1,2,2);stem(y);title('直接型 y(n)');
```

运行结果如图 5.20 所示。

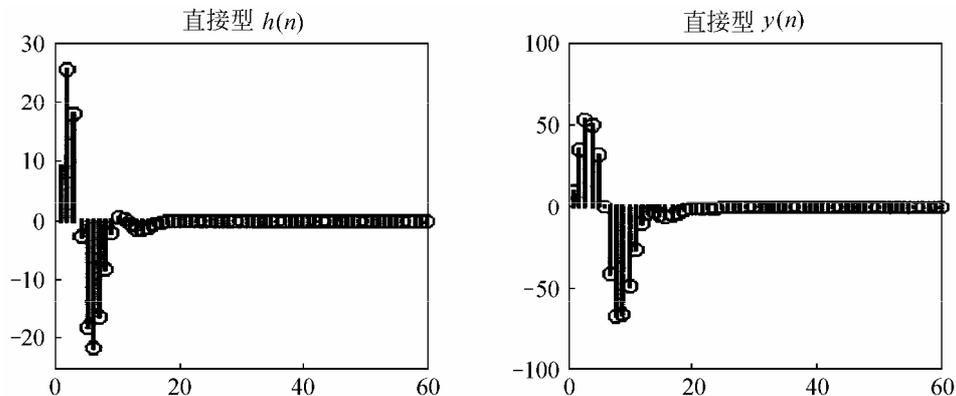


图 5.20 直接型单位脉冲响应和输出信号

下面给出直接型、级联型及并联型之间的互换函数。

直接型转换为级联型：

```
function [ b0,B,A]=dir2cas(b,a);
% 直接型到级联形式的形式转换(复数对型);
% -----
% [b0,B,A]=dir2cas(b,a);
% b=直接型的分子多项式系数
% a=直接型的分母多项式系数
% b0=增益系数
% B=包含各 bk 的 K 乘 3 维实系数矩阵
% A=包含各 ak 的 K 乘 3 维实系数矩阵

% 计算增益系数 b0
b0=b(1);b=b/b0;
a0=a(1);a=a/a0;
b0=b0/a0;
%
M=length(b);N=length(a);
if N > M
    b=[b zeros(1,N-M)];
elseif M > N
    a=[ a zeros(1,M-N)];N=M;
else
    NM=0
end
%
K=floor(N/2);B=zeros(K,3);A=zeros(K,3);
if K*2==N;
    b=[b 0];
    a=[a 0];
end
%
broots=cplxpair(roots(b));
aroots=cplxpair(roots(a));
for i=1:2:2 * K
    Brow=broots(i:1:i+1,:);
    Brow=real(poly(Brow));
    B(fix((i+1)/2),:)=Brow;
    Arow=aroots(i:1:i+1,:);
    Arow=real(poly(Arow));
    A(fix((i+1)/2),:)=Arow;
end
```

上述函数把矢量 b 和 a 转换成 $k \times 3$ 维矩阵 B 和 A 。首先计算 b_0/a_0 ，(假设 $a_0 \neq 1$)，然后通过给较短的矢量填零，使矢量 b 和 a 一样长，这就保证每一个对二阶环节的分子和分

母均不为零；接着计算多项式 $B(z)$ 和 $A(z)$ 的复根，用 `cplxpair` 函数把这些根按共轭复根对的次序排列；然后用 `poly` 函数把每一对根再转换成二阶分子或分母多项式。

滤波器级联型实现：

```
function y=casfiltr(b0,B,A,x);
% IIR 各 FIR 滤波器的级联型的实现
% -----
% y=casfiltr(b0,B,A,x);
% y=输出序列
% b0=级联型的增益系数
% B=包含各 bk 和 K 乘 3 维实系数矩阵
% A=包含各 ak 的 K 乘 3 维实系数矩阵
% x=输入序列
%
[K,L]=size(B);
N=length(x);
w=zeros(K+1,N);
w(1,:)=x;
for i=1:L:K
    w(i+1,:)=filter(B(i,:),A(i,:),w(i,:));
end
y=b0 * w(K+1,:);
```

级联型转换成直接型：

```
function [b,a]=cas2dir(b0,B,A);
% 级联型到直接型的形式转换
% -----
% [b,a]=cas2dir(b0,B,A)
% b=直接型的分子多项式系数
% a=直接型的分母多项式系数
% b0=增益系数
% B=包含各 bk 的 K 乘 3 维实系数矩阵
% A=包含各 ak 的 K 乘 3 维实系数矩阵
%
[K,L]=size(B);
b=[1];
a=[1];
for i=1:L:K
    b=conv(b,B(i,:));
    a=conv(a,A(i,:));
end
b=b*b0;
```

滤波器并联型实现：

```
function y=parfiltr(C,B,A,x);
```

```

% IIR 滤波器的并联型实现
% -----
% [y]=parfiltr(C,B,A,x);
% y=输出序列
% C=当 M>=N 时(FIR)的多项式部分
% B=包含各 bk 的 K 乘 2 维实系数矩阵
% A=包含各 ak 的 K 乘 3 维实系数矩阵
% x=输入序列
%
[K,L]=size(B);
N=length(x);
w=zeros(K+1,N);
w(1,:)=filter(C,1,x);
for i=1:1:K
    w(i+1,:)=filter(B(i,:),A(i,:),x);
end
y=sum(w);

```

直接型转换成并联型：

```

function[C,B,A]=dir2par(b,a)
% 直接型到并联型的转换
% -----
% [C,B,A]=dir2par(b,a)
% C=当 length(b)>=length(a)时的多项式部分
% B=包含各 bk 的 K 乘 2 维实系数矩阵
% A=包含各 ak 的 K 乘 3 维实系数矩阵
% b=直接型的分子多项式系数
% a=直接型的分母多项式系数
%
M=length(b);N=length(a);
[r1,p1,C]=residuez(b,a);
p=cplxpair(p1,10000000*eps);
I=cplxcomp(p1,p);
r=r1(I);
K=floor(N/2);B=zeros(K,2);A=zeros(K,3);
if K*2 == N; % A(z)
for i =1:2:N-2
    Brow=r(i:1:i+1,:);
    Arow=p(i:1:i+1,:);
    [Brow,Arow]=residuez(Brow,Arow,[]);
    B(fix(i+1)/2,:) =real(Brow;ã;
    A(fix(i+1)/2,:) =real(Arow;ã;
end
[Brow,Arow]=residuez(r(N-1),p(N-1),[]);
B(K,:)=real(Brow;ã0;A(K,:)=real(Arow;ã0;

```

```

else
    for i=1:2:N-1
        Brow=r(i:1:i+1,:);
        Arow=p(i:1:i+1,:);
        [Brow,Arow]=residuez(Brow,Arow,[]);
        B(fix((i+1)/2),:)=real(Brow;ã);
        A(fix((i+1)/2),:)=real(Arow;ã);
    end
end
end

```

`dir2par` 函数首先在 z 域里用 `residuez` 函数计算部分分式展开。需要把极点-留数对按复共轭极点-留数对、实极点-留数对的顺序进行排列，用 MATLAB 中的 `cplxpair` 函数可做到这一点，这可把复数数组分类为复共轭对。但由于连续两次调用此函数（一次为极点，一次为留数），不能保证极点和留数的互相对应。因此，开发了一个新的 `cplxcomp` 函数，它把两个混乱的复数数组进行比较，返回一个数组的下标，用它重新给另一个数组排序。

```

function I=cplxcomp(p1,p2)
% I=cplxcomp(p1,p2)
% 比较两个包含同样标量元素但（可能）有不同下标的复数对
% 本程序必须用 CPLXPAIR 程序之后以便重新排序频率极点向量
% 及其相应的留数向量：
%     P2=CPLXPAIR ( P1 )
%
I=[];
for j=1:1:length(p2)
    for i=1:1:length(p1)
        if(abs(p1(i)-p2(j))<0.0001)
            I=(I,i);
        end
    end
end
I=I;ã

```

得到极点-留数对后，`dir2par` 函数用 `residuez` 函数的逆形式计算双二阶环节的分子和分母。

在实现并联形式的 `parfiltr` 函数中，并联形式的系数由 `parfiltr` 函数把每个二阶环节的系数存在矩阵 B 和 A 中，并放在一个循环中，在此循环中，使用 `filter` 函数输入首先经过 FIRC 部分的滤波，其结果存在矩阵 w 的第一行中，随后，计算相同输入下的所有二阶滤波器的输出，并把结果存于矩阵 w 的各行中，最后，把 w 中所有的列相加，产生最终的输出。

并联型转换成直接型：

函数 `par2dir` 可用来从并联形式得到直接形式，它计算每一个双二阶环节的极点和留数，然后把它们组合为系统的极点和留数，再调用 `residuez` 函数，计算分子和分母多项式。

```

function [b,a] =par2dir(C,B,A);
% 并联型到直接型的转换
% -----
% [b,a]=par2dir(C,B,A)
% b= 直接型的分子多项式系数
% a= 直接型的分母多项式系数
% C= 并联型的多项式
% B=包含各 bk 的 K 乘 2 维实系数矩阵
% A=包含各 ak 的 K 乘 3 维实系数矩阵
%
[K,L]=size(A);R=[];p=[];

for i=1:1:K
[r,p,k]=residuez(B(i,:),A(i,:));
R=[R;r];P=[P;p];
end
[b,a]=residuez(R,P,C);
b=b(:);ãa=a(:);ã

```

5.4.2 FIR 滤波器的 MATLAB 实现

1. 直接形式

FIR 直接型系统函数表示为

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_{M-1} z^{-(M-1)}$$

即 FIR 结构的直接形式由包含系数 $\{b_n\}$ 的行向量 \mathbf{b} 描述。在 MATLAB 中，把分母矢量 \mathbf{a} 置为 1，这种结构用 filter 函数实现。

2. 级联形式

FIR 级联型系统函数表示为

$$H(z) = b_0 \prod_{k=1}^K (1 + B_{k,1} z^{-1} + B_{k,2} z^{-2})$$

即级联型 FIR 滤波器可以通过 casfilt 函数实现。但这种形式与 IIR 形式类似，也可以使用 dir2cas 函数，把分母矢量 \mathbf{a} 置为 1，用 cas2dir 函数从级联形式转换为直接形式而获得。

例 5.8 FIR 滤波器的系统函数为

$$H(z) = \begin{cases} 0.2^n, & 0 \leq n \leq 5 \\ 0, & \text{其他} \end{cases}$$

试分别用直接型和级联型实现。

解：程序如下。

```
n=0:5;
```

```

b=0.2.^n;
N=30;
delta=impseq(0,0,N);
h=filter(b,1,delta);
x=[ones(1,5),zeros(1,N-5)];
y=filter(b,1,x);
subplot(2,2,1);stem(h);title('直接型 h(n)');
subplot(2,2,2);stem(y);title('直接型 y(n)');
[b0,B,A]=dir2cas(b,1);
h=casfiltr(b0,B,A,delta);
y=casfiltr(b0,B,A,x);
subplot(2,2,3);stem(h);title('级联型 h(n)');
subplot(2,2,4);stem(y);title('级联型 y(n)');

```

运行结果如图 5.21 所示。

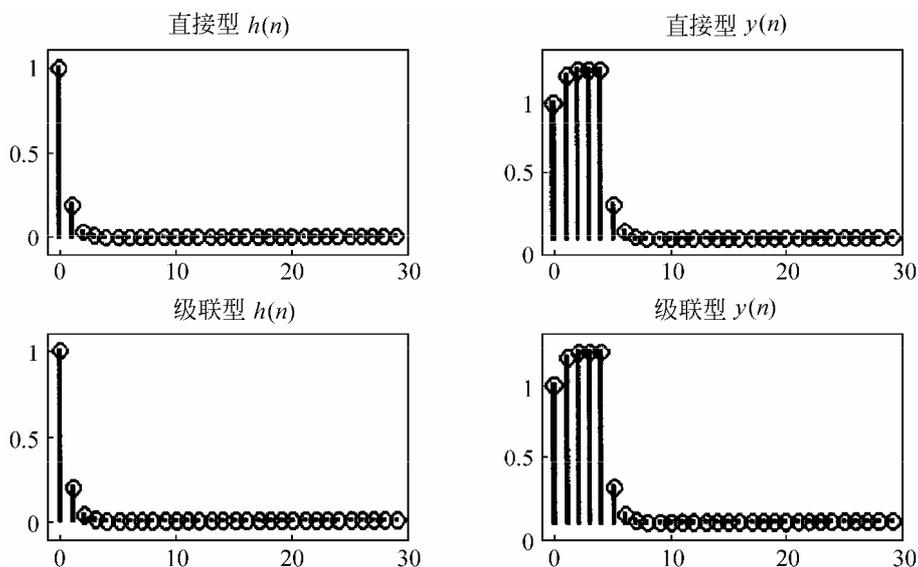


图 5.21 单位脉冲响应和输出信号

3. 频率采样型

由于 FIR 滤波器几乎总是实数，利用 $H(k)$ 的对称性和 W_N^{-k} 因子，可把式 (5.12) 表示为

$$H(z) = \frac{1-z^{-N}}{N} \left\{ \sum_{k=1}^L 2|H(k)|H_k(z) + \frac{H(0)}{1-z^{-1}} + \frac{H\left(\frac{N}{2}\right)}{1+z^{-1}} \right\}$$

其中 N 为奇数时 $L = (M-1)/2$ ；

N 为偶数时 $L = M/2 - 1$ 。

$$H_k(z) = \frac{\cos[\angle H(k)] - z^{-1} \cos\left[\angle H(k) - \frac{2\pi k}{N}\right]}{1 - 2z^{-1} \cos\left(\frac{2\pi k}{N}\right) + z^{-2}}$$

若给定直接型的 $h(n)$ 或 $H(k)$ ，则可以由 `dir2fs` 函数转换为频率采样结构。

例 5.9 令 $h(n) = \{1, 2, 3, 2, 1\}/9$ ，确定并画出 FIR 的频率取样形式结构。

解：程序如下。

```
h=[1,2,3,2,1]/9;
[C,B,A]=dir2fs(h);
```

运行结果为：

```
C=
    0.5818
    0.0849
    1.0000
B=
   -0.8090    0.8090
    0.3090   -0.3090
A=
    1.0000   -0.6180    1.0000
    1.0000    1.6180    1.0000
    1.0000   -1.0000     0
```

FIR 频率采样结构形式为

$$H(z) = \frac{1-z^{-5}}{5} \left[0.5818 \times \frac{-0.809 + 0.809z^{-1}}{1 - 0.618z^{-1} + z^{-2}} + 0.0849 \times \frac{0.309 - 0.309z^{-1}}{1 + 0.618z^{-1} + z^{-2}} + \frac{1}{1 - z^{-1}} \right]$$

直接型到频率采样型转换函数：

```
function[C,B,A]=dir2fs(h)
%C=并联部分增益的行向量
%B=按行排列的分子系数矩阵
%A=按行排列的分母系数矩阵
%h=FIR 滤波器的单位脉冲响应向量
M=length(h);
H=fft(h,M);
magH=abs(H);phaH=angle(H);
if(M==2*floor(M/2))
    L=M/2-1;
    A1=[1,-1,0;1,1,0];
    C1=[real(H(1)),real(H(L+2))];
else
    L=(M-1)/2;
    A1=[1,-1,0];
```

```
C1=[real(H(1))];  
end  
k=[1:L]';  
B=zeros(L,2);A=ones(L,3);  
A(1:L,2)=-2*cos(2*pi*k/M);A=[A;A1];  
B(1:L,1)=cos(phaH(2:L+1));  
B(1:L,2)=cos(phaH(2:L+1)-(2*pi*k/M));  
C=[2*magH(2:L+1),C1]';
```

第 6 章 无限冲激响应滤波器的设计

滤波器在实际的信号处理中起着重要的作用，它是去除信号中噪声的基本手段，因此，滤波器的设计问题是数字信号处理中的基本问题。本章将集中讨论 IIR 滤波器的设计问题，在下一章将集中讨论 FIR 滤波器的设计问题。

6.1 通过模拟滤波器设计 IIR 数字滤波器

6.1.1 模拟低通滤波器原型

1. 滤波器的技术指标

典型的模拟低通滤波器的技术指标如图 6.1 (a) 所示，其中 $|H_\alpha(j\Omega)|$ 为模拟滤波器的幅度响应， Ω_p 、 Ω_{st} 分别为通带频率和阻带频率， δ_p 、 δ_s 分别为通带、阻带的容限。在通带内要求

$$1 - \delta_p \leq |H_\alpha(j\Omega)| \leq 1$$

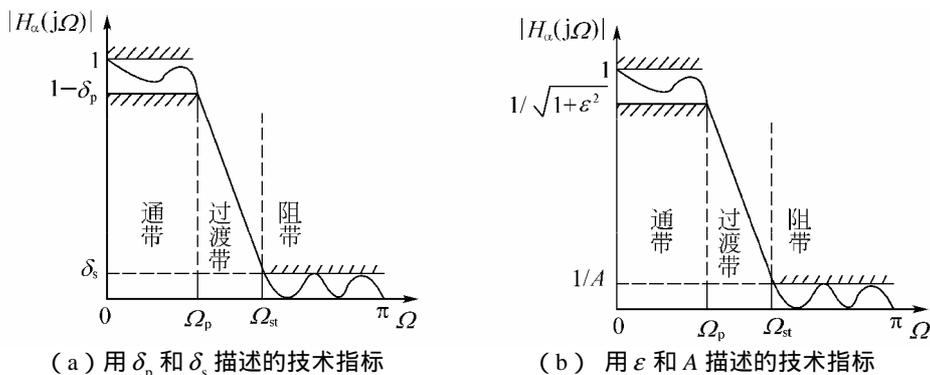


图 6.1 模拟低通滤波器的技术指标

有时，具体的技术指标由通带允许的最大衰减 α_p 和阻带应达到的最小衰减 α_{st} 给出。通带及阻带的衰减 α_p 、 α_{st} 分别定义为

$$\alpha_p = -20\lg(1 - \delta_p)$$

$$\alpha_{st} = -20\lg(\delta_s)$$

第二种常用的指标是用参数 ϵ 和 A 描述通带和阻带的要求，如图 6.1 (b) 所示。

δ_p 、 δ_s 与 ϵ 、 A 的关系为

$$\epsilon = \left[(1 - \delta_p)^{-2} - 1 \right]^{1/2}$$

$$\delta_s = \frac{1}{A}$$

根据 δ_p 、 δ_s 或 ε 、 A ，及 Ω_p 、 Ω_{st} 可以导出另外两个辅助参数 d 、 k 。

$$d = \left[\frac{(1 - \delta_p)^{-2} - 1}{\delta_s^{-2} - 1} \right]^{1/2} = \frac{\varepsilon}{\sqrt{A^2 - 1}}$$

$$k = \frac{\Omega_p}{\Omega_{st}}$$

d 、 k 分别叫判别因子和选择性因子。

模拟滤波器幅度响应也可以用幅度平方函数 $|H_\alpha(j\Omega)|^2$ 来表示，即

$$|H_\alpha(j\Omega)|^2 = H_\alpha(j\Omega)H_\alpha^*(j\Omega)$$

由于滤波器冲激响应函数 $h(t)$ 是实函数，因而， $H(j\Omega)$ 满足

$$H^*(j\Omega) = H(-j\Omega)$$

所以

$$|H_\alpha(j\Omega)|^2 = H_\alpha(j\Omega)H_\alpha(-j\Omega) = H_\alpha(s)H_\alpha(-s) \Big|_{s=j\Omega} \quad (6.1)$$

其中 $H_\alpha(s)$ 是模拟滤波器的系统函数，它是 s 的有理函数， $H_\alpha(j\Omega)$ 是滤波器频率响应。

2. 巴特沃兹 (Butterworth) 低通滤波器的设计

巴特沃兹 (Butterworth) 低通滤波器幅度平方函数定义为

$$|H_\alpha(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}} \quad (6.2)$$

式中 N 为正整数，代表滤波器的阶次， Ω_c 称为截止频率。当 $\Omega = \Omega_c$ 时，有

$$|H_\alpha(j\Omega)|^2 = \frac{1}{2}$$

即

$$|H_\alpha(j\Omega)| = \frac{1}{\sqrt{2}}, \quad \delta_1 = 20 \lg \left| \frac{H_\alpha(j0)}{H_\alpha(j\Omega_c)} \right| = 3 \text{dB}$$

所以又称 Ω_c 为巴特沃兹低通滤波器的 3dB 带宽。

巴特沃兹低通滤波器的特点：

- 当 $\Omega = 0$ 时， $|H_\alpha(j0)|^2 = 1$ ，即在 $\Omega = 0$ 处无衰减。
- 当 $\Omega = \Omega_c$ 时， $|H_\alpha(j\Omega_c)|^2 = \frac{1}{2}$ ， $|H_\alpha(j\Omega_c)| = 0.707$ ，或 $\delta_1 = -20 \lg |H_\alpha(j\Omega_c)| = 3 \text{dB}$ ， δ_1 为通带最大衰减。即不管 N 为多少，所有的曲线都通过 -3dB 点，或者说衰减 3dB，这就是 3dB 不变性。
- 在 $\Omega < \Omega_c$ 的通带内 $|H_\alpha(j\Omega)|^2$ 有最大平坦的幅度特性，即 N 阶巴特沃兹低通滤波器在 $\Omega = 0$ 处， $|H_\alpha(j\Omega)|^2_{\Omega=0}$ 的前 $(2N-1)$ 阶导数为零，因而巴特沃兹滤波器又称为最平幅度特性滤波器。随着 Ω 由 0 变到 Ω_c ， $|H_\alpha(j\Omega)|^2$ 单调减小， N 越大，减小得越慢，通带内特性越平坦。

- 当 $\Omega > \Omega_c$, 即在过渡带及阻带中, $|H_\alpha(j\Omega)|^2$ 也随 Ω 增加而单调减小, 但是 $\Omega/\Omega_c > 1$, 故比通带内衰减的速度要快得多, N 越大, 衰减速度越大。当 $\Omega = \Omega_{st}$, 即频率为阻带截止频率时, 衰减为 $\delta_2 = -20\lg |H_\alpha(j\Omega_{st})|$ 。

巴特沃兹低通滤波器的幅度特性如图 6.2 所示。由式 (6.2) 得

$$H_\alpha(s)H_\alpha(-s) = |H_\alpha(j\Omega)|^2 \Big|_{\Omega=\frac{s}{j}} = \frac{1}{1 + \left(\frac{s}{j\Omega_c}\right)^{2N}} \quad (6.3)$$

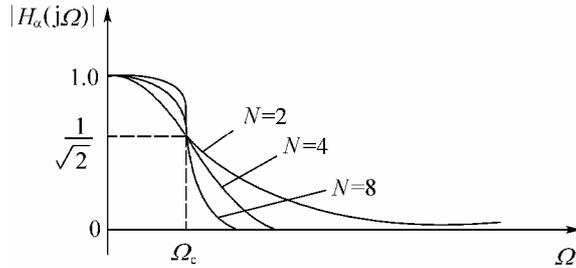


图 6.2 巴特沃兹低通滤波器的幅度特性

所以巴特沃兹低通滤波器的零点全部在 $s = \infty$ 处, 在有限 z 平面内, 只有极点, 属于“全极点滤波器”, 其幅度平方函数 $H_\alpha(s)H_\alpha(-s)$ 的 $2N$ 个极点为

$$s_k = (-1)^{2N} (j\Omega_c) = \Omega_c e^{j\left(\frac{1+2k-1}{2+2N}\right)\pi}, \quad k=1, 2, \dots, 2N \quad (6.4)$$

$H_\alpha(s)H_\alpha(-s)$ 在左半平面的极点即为 $H_\alpha(s)$ 的极点, 因而

$$H_\alpha(s) = \frac{\Omega_c^N}{\prod_{k=1}^N (s - s_k)} \quad (6.5)$$

在一般设计中, 把上式中的 Ω_c 选为 1 rad/s, 即将频率归一化, 归一化后巴特沃兹低通滤波器的系统函数有以下形式:

$$H_\alpha(s) = \frac{1}{s^N + a_1 s^{N-1} + \dots + a_{N-1} s + a_N} \quad (6.6)$$

表 6.1 中列出了式 (6.6) 中分母多项式的系数。设计一个巴特沃兹滤波器的步骤如下:

- (1) 根据滤波器技术指标, 求选择性因子 k 和判别因子 d 。
- (2) 确定满足技术要求所需的滤波器阶数 N 。

$$N \geq \frac{\lg d}{\lg k}$$

- (3) 设 3dB 截止频率为 Ω_c , Ω_c 可以是以下区间的任一个数值。

$$\Omega_p [(1 - \delta_p)^{-2} - 1]^{-\frac{1}{2N}} \leq \Omega_c \leq \Omega_s [\delta_s^{-2} - 1]^{-\frac{1}{2N}}$$

- (4) 由式 (6.6) 并查表求出归一化的巴特沃兹滤波器的系统函数。

例 6.1 设计一个满足以下技术指标的低通巴特沃兹滤波器。

$$f_p = 6\text{kHz}, \quad f_s = 10\text{kHz}, \quad \delta_p = \delta_s = 0.1$$

表 6-1 阶数 $1 \leq N \leq 8$ 归一化巴特沃兹滤波器系统函数的系数

N	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	1.0000							
2	1.4142	1.0000						
3	2.0000	2.0000	1.0000					
4	2.6131	3.4142	2.6131	1.0000				
5	3.2361	5.2361	5.2361	3.2361	1.0000			
6	3.8637	7.4641	9.1416	7.4641	3.8637	1.0000		
7	4.4940	10.0978	14.5918	14.5918	10.0978	4.4940	1.0000	
8	5.1258	13.1371	21.8462	25.6884	21.8642	13.1372	5.1258	1.0000

解：先计算判别因子和选择性因子。

$$d = \left[\frac{(1 - \delta_p)^{-2} - 1}{\delta_s^{-2} - 1} \right]^{1/2} = 0.0487, \quad k = \frac{\Omega_p}{\Omega_s} = \frac{f_p}{f_s} = 0.6$$

由于
$$N \geq \frac{\lg d}{\lg k} = 5.92$$

得到最小的滤波器阶数是 $N=6$ ，而

$$f_p [(1 - \delta_p)^{-2} - 1]^{-\frac{1}{2N}} = 6770, \quad f_s [\delta_s^{-2} - 1]^{-\frac{1}{2N}} = 6819$$

中心频率可以是以下区间的任何一个数。

$$6770 \leq f_c \leq 6819$$

根据式 (6.6) 并查表求出归一化的巴特沃兹滤波器

$$H_\alpha(s) = \frac{1}{s^6 + 3.8637s^5 + 7.4641s^4 + 9.1416s^3 + 7.4641s^2 + 3.8637s + 1}$$

若用 $s = s/\Omega_c$ 代入上式，即可得截止频率为 Ω_c 而非 1 的巴特沃兹滤波器。

3. 切比雪夫 (Chebyshev) I 型滤波器的设计

切比雪夫 I 型滤波器的幅度平方函数为

$$|H_\alpha(j\Omega)|^2 = \frac{1}{1 + \varepsilon^2 T_N^2\left(\frac{\Omega}{\Omega_c}\right)} \quad (6.7)$$

其中 $T_N^2\left(\frac{\Omega}{\Omega_c}\right)$ 为切比雪夫多项式，它的定义为

$$T_N(x) = \begin{cases} \cos(N \cos^{-1} x), & |x| \leq 1 \\ \cosh(N \cosh^{-1} x), & |x| > 1 \end{cases} \quad (6.8)$$

这些多项式可以通过迭代产生

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x), \quad k \geq 1 \\ T_0(x) &= 1, \quad T_1(x) = x \end{aligned}$$

切比雪夫 I 型滤波器是一个全极点滤波器，在通带内等波纹，在阻带内单调下降。

切比雪夫 I 型滤波器的系统函数为

$$H_{\alpha}(s) = H_{\alpha}(0) \prod_{k=0}^{N-1} \frac{-s_k}{s - s_k}$$

其中 N 为偶数时, $H_{\alpha}(0) = (1 - \varepsilon^2)^{-1/2}$, N 为奇数时, $H_{\alpha}(0) = 1$ 。

给定通带和阻带的截止频率 Ω_p 和 Ω_s , 通带和阻带波动 δ_p 和 δ_s (或参数 ε 和 A), 设计一个切比雪夫 I 型滤波器的步骤如下:

- (1) 求选择性因子 k 和判别因子 d 。
- (2) 用下面的公式确定滤波器阶数。

$$N \geq \frac{\cosh^{-1}\left(\frac{1}{d}\right)}{\cosh^{-1}\left(\frac{1}{k}\right)}$$

- (3) 组成有理函数

$$G_a(s) = H_a(s)H_a(-s) = \frac{1}{1 + \varepsilon^2 T_N^2\left(\frac{s}{j\Omega_p}\right)}$$

其中 $\varepsilon = [(1 - \delta_p)^{-2} - 1]^{1/2}$, 取 N 个 $G_a(s)$ 在 s 左半平面的极点组成系统函数 $H_a(s)$ 。

例 6.2 设计一个切比雪夫 I 型低通滤波器, 满足例 6.1 中给出的技术指标。

解: 求得 $d = 0.0487$, $k = 0.6$ 所需的滤波器阶数为

$$N \geq \frac{\cosh^{-1}\left(\frac{1}{d}\right)}{\cosh^{-1}\left(\frac{1}{k}\right)} = 3.38$$

取 $N=4$, 所以

$$\varepsilon = [(1 - \delta_p)^{-2} - 1]^{1/2} = 0.4843$$

$$T_4(x) = 4x^3 - 4x$$

$$T_4^2\left(\frac{\Omega}{\Omega_p}\right) = 16\left(\frac{\Omega}{\Omega_p}\right)^2 \left[\left(\frac{\Omega}{\Omega_p}\right)^2 - 1 \right]^2$$

这样

$$|H_{\alpha}(j\Omega)|^2 = \frac{1}{1 + 3.7527\left(\frac{\Omega}{\Omega_p}\right)^2 \left[\left(\frac{\Omega}{\Omega_p}\right)^2 - 1 \right]^2}$$

6.1.2 由模拟滤波器设计 IIR 数字滤波器

在各种原型模拟滤波器的设计基础上, 如果能建立模拟滤波器与数字滤波器的映射关系, 则可以很好地利用模拟滤波器的研究成果来设计数字滤波器。

要建立模拟滤波器与数字滤波器的映射关系,就是要建立 s 平面与 z 平面的映射关系,即把 s 平面映射到 z 平面,使模拟系统函数 $H_a(s)$ 变换成所需的数字滤波器的系统函数 $H(z)$ 。这种由复平面 s 到复平面 z 的映射(变换)关系,必须满足两个基本要求:

- $H(z)$ 的频率响应要能模仿 $H_a(s)$ 的频率响应,即 s 平面的虚轴 $j\Omega$ 必须映射到 z 平面的单位圆 $e^{j\omega}$ 上。
- 变换前 $H_a(s)$ 是因果稳定的,变换后 $H(z)$ 也必须是因果稳定的。也就是说, s 轴的左半平面必须映射到 z 平面的单位圆内,如图 6.3 所示。

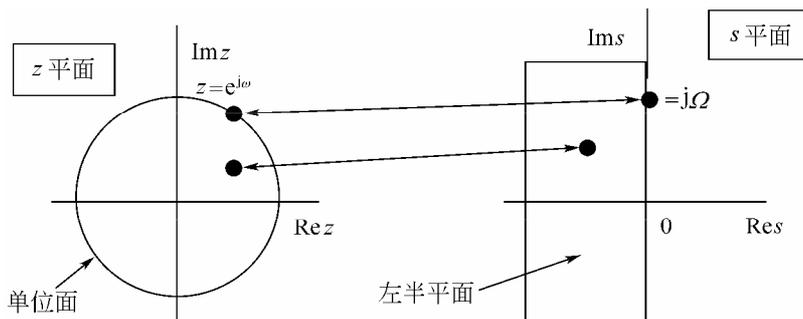


图 6.3 s 平面到 z 平面的映射

把模拟滤波器映射成数字滤波器后,就能使数字滤波器“模仿”模拟滤波器的特性,从而达到由模拟滤波器设计数字滤波器的目的。

通常,有以下几种映射方法:冲激响应不变法、双线性变换法等。

6.2 冲激响应不变法

6.2.1 变换原理

冲激响应不变法是使数字滤波器的单位冲激响应序列 $h(n)$ 模仿模拟滤波器的单位冲激响应 $h_a(t)$,将模拟滤波器的单位冲激响应加以等间隔抽样,使 $h(n)$ 正好等于 $h_a(t)$ 的抽样值,即满足

$$h(n) = h_a(nT) \quad (6.9)$$

其中 T 是抽样周期。

抽样序列的 z 变换与模拟信号的拉普拉斯变换之间的关系为

$$H(z) \Big|_{z=e^{sT}} = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a \left(s - j \frac{2\pi}{T} k \right) \quad (6.10)$$

可以看出,冲激响应不变法将模拟滤波器的 s 平面变换到数字滤波器的 z 平面,从 s 到 z 的变换关系为 $z = e^{sT}$,其映射关系如图 6.4 所示。

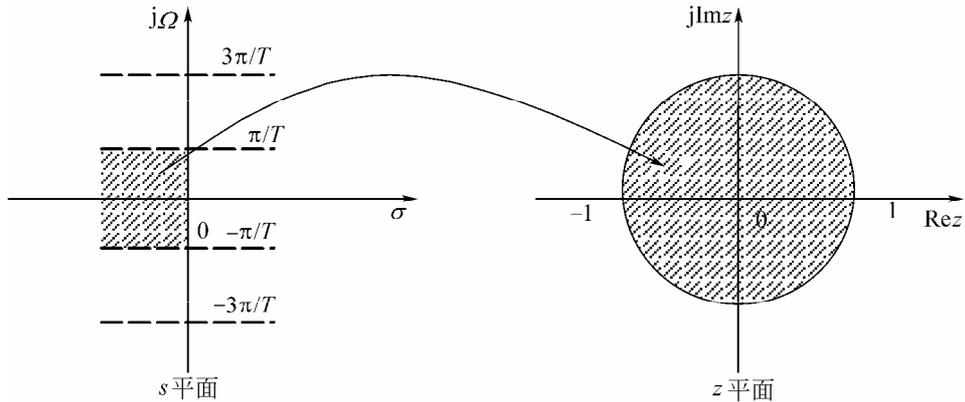


图 6.4 冲激响应不变法映射关系

图 6.4 中, s 平面每一条宽度为 $2\pi T$ 的横条都将重叠地映射到整个 z 平面上, 而每一横条的左半边映射到 z 平面单位圆以内, 右半边映射到单位圆以外, 而 s 平面虚轴 ($j\Omega$ 轴) 映射到单位圆上, 虚轴上每一段长为 $2\pi T$ 的线段都映射到 z 平面单位圆上一周。

由此可知, 冲激响应不变法把稳定的 $H_a(s)$ 转换为稳定的 $H(z)$ 。由此方法可得到一阶系统的最基本的转换关系式为

$$\frac{1}{s+a} \Rightarrow \frac{1}{1-e^{-aT}s^{-1}} \quad (6.11)$$

即 s 平面的单极点 $s = -a$ 映射到时 z 平面的单极点 $z = e^{-aT}$ 。

6.2.2 混叠失真

由式 (6.10) 可知, 数字滤波器的频率响应与模拟滤波器的频率响应间的关系为

$$H(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(j\frac{\Omega-2\pi k}{T}\right) \quad (6.12)$$

即数字滤波器的频率响应是模拟滤波器频率响应的周期延拓。根据奈奎斯特抽样定理, 只有当模拟滤波器的频率响应是严格限带的, 且带限于折叠频率 $[-\Omega_s/2, \Omega_s/2]$ 以内时, 才能使数字滤波器的频率响应在折叠频率以内重现模拟滤波器的频率响应而不产生混叠失真。但是, 任何一个实际的模拟滤波器频率响应都不是严格限带的, 变换后都会产生周期延拓分量的频谱交叠, 即产生频率响应的混叠失真, 因而模拟滤波器的频率响应在折叠频率以上衰减越大、越快, 变换后频率响应混叠失真就越小。

例 6.3 设模拟滤波器的系统函数为

$$H_a(s) = \frac{2}{s^2 + 4s + 3} = \frac{1}{s+1} - \frac{1}{s+3}$$

试利用冲激响应不变法, 设计 IIR 数字滤波器。

解: 由于模拟滤波器 $H_a(s)$ 有两个一阶极点 $s_1 = -1$, $s_2 = -3$, 根据极点映射的特点, 直接利用式 (6.11) 可得到数字滤波器的系统函数为

$$H(z) = \frac{1}{1-z^{-1}e^{-T}} - \frac{1}{1-z^{-1}e^{-3T}} = \frac{z^{-1}(e^{-T} - e^{-3T})}{1-z^{-1}(e^{-T} + e^{-3T}) + z^{-2}e^{-4T}}$$

设 $T=1$ ，则有

$$H(z) = \frac{0.318z^{-1}}{1 - 0.4177z^{-1} + 0.01831z^{-2}}$$

模拟滤波器 $H_\alpha(j\Omega)$ 的频率响应以及数字滤波器 $H(e^{j\omega})$ 的频率响应分别为

$$H_\alpha(j\Omega) = \frac{2}{(3 - \Omega^2) + j4\Omega}$$

$$H(e^{j\omega}) = \frac{0.318e^{-j\omega}}{1 - 0.4177e^{-j\omega} + 0.01831e^{-j2\omega}}$$

$|H_\alpha(j\Omega)|$ 和 $|H(e^{j\omega})|$ 画在图 6.5 上, 由该图可看出, 由于 $H_\alpha(j\Omega)$ 不是充分限带, 所以 $H(e^{j\omega})$ 产生了很大的频谱混叠失真。

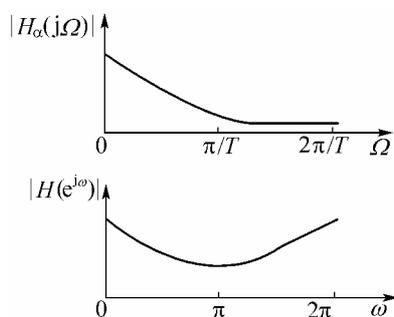


图 6.5 冲激响应不变法的频率响应幅度

6.2.3 优缺点

- 冲激响应不变法使数字滤波器的冲激响应完全模仿模拟滤波器的冲激响应, 即在时域逼近良好。
- 模拟频率和数字频率之间呈线性关系, 即 $\omega = \Omega T_s$, 因而一个线性相位滤波器可以映射成一个线性相位的数字滤波器。
- 由于有混叠效应, 所以只适用于带限模拟滤波器, 即只适用于低通滤波器和带通滤波器。对于高通和带阻滤波器不宜采用冲激响应不变法。

6.3 双线性变换法

6.3.1 变换原理

IIR 滤波器设计的另一简单、有效的方法就是双线性变换法。

该方法与前述冲激响应不变法的基本思路一样, 不直接设计数字滤波器, 而是先设计一个模拟 IIR 滤波器, 然后映射成一个等效的数字滤波器。其变换原理如图 6.6 所示。

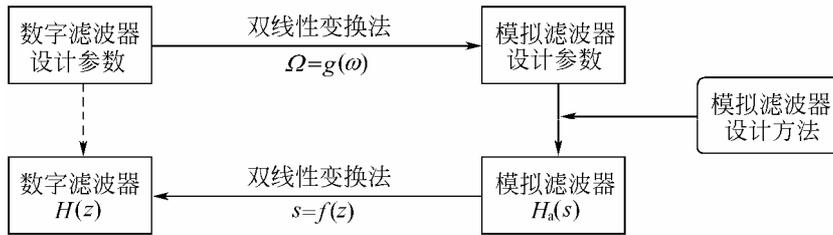


图 6.6 双线性变换法

这样，就可以把 z 平面的数字滤波器的设计转化为 s 平面的等效模拟滤波器的设计。 s 平面和 z 平面的映射关系为

$$s = f(z) = \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (6.13)$$

将 $s = j\Omega$ 及 $z = e^{j\omega}$ 代入式 (6.13)，得到数字频率与等效的模拟频率之间的映射关系为

$$\Omega = g(\omega) = \tan\left(\frac{\omega}{2}\right) \quad (6.14)$$

由于数字频率与模拟频率之间的变换关系不是线性关系，所以式 (6.14) 被称为频率预畸变换法。

双线性变换法的设计步骤如下：

- (1) 给定数字滤波器的幅度响应参数。
- (2) 用频率预畸公式 (6.14) 将数字滤波器参数变换为相应的等效模拟滤波器的参数。
- (3) 采用模拟滤波器设计方法设计等效模拟滤波器—— $H_a(s)$ 。
- (4) 采用双线性变换法公式 (6.13) 把等效模拟滤波器逆映射为所期望的数字滤波器，即

$$H(z) = H_a(s) \Big|_{s=f(z)} = H_a(f(z)) \quad (6.15)$$

$$H(\omega) = H_a(\Omega) \Big|_{\Omega=g(\omega)} = H_a(g(\omega)) \quad (6.16)$$

6.3.2 优缺点

- 避免了频率响应的混叠现象。
- 模拟频率与数字频率不再是线性关系，所以一个线性相位模拟滤波器经双线性变换后所得到的数字滤波器不再保持原有的线性相位了。

例 6.4 试用双线性变换法设计一低通滤波器，给定技术指标是 $f_p = 100\text{Hz}$ ， $f_{st} = 300\text{Hz}$ ， $\alpha_p = 3\text{dB}$ ， $\alpha_s = 20\text{dB}$ ，抽样频率为 $F_s = 100\text{Hz}$ 。

解：首先求数字频率 ω ，因为 2π 对应于 F_s ，所以 $\omega_p = 2\pi f_p / F_s = 0.2\pi$ ， $\omega_{st} = 2\pi f_{st} / F_s = 0.6\pi$ 。

- (1) 用频率预畸公式 (6.14) 将数字滤波器的技术要求转换为模拟滤波器的技术要求。

$$\Omega_p = \tan\left(\frac{\omega_p}{2}\right) = 0.3249, \quad \Omega_{st} = \tan\left(\frac{\omega_{st}}{2}\right) = 1.37638$$

(2) 设计低通滤波器 $G(s)$ 。

先计算判别因子和选择性因子。

因为

$$\alpha_p = -20\lg(1 - \delta_p), \quad \alpha_{st} = -20\lg(\delta_s)$$

所以

$$1 - \delta_p = 10^{-\frac{3}{20}} = 0.70795, \quad \delta_s = 0.1$$

$$d = \left[\frac{(1 - \delta_p)^{-2} - 1}{\delta_s^{-2} - 1} \right]^{\frac{1}{2}} = 0.10027, \quad k = \frac{\Omega_p}{\Omega_s} = \frac{f_p}{f_{st}} = \frac{1}{3}$$

由

$$N \geq \frac{\lg d}{\lg k} = 2.09$$

得滤波器阶数 $N=2$ ，查表 6.1 得 $a_1 = 1.4142$ ， $a_2 = 1$ ，所以得归一化的 2 阶低通巴特沃兹滤波器 $G(p)$ 为

$$G(p) = \frac{1}{p^2 + 1.4142p + 1}$$

$$G(s) = G(p) \Big|_{p=\frac{s}{\Omega_p}} = \frac{0.3249^2}{s^2 + 0.4595s + 0.3249^2}$$

(3) 采用双线性变换法公式 (6.15) 求 $H(z)$ 。

$$H(z) = G(s) \Big|_{s=\frac{z-1}{z+1}} = \frac{0.06745 + 0.1349z^{-1} + 0.06745z^{-2}}{1 - 1.143z^{-1} + 0.4208z^{-2}}$$

6.4 从原型低通滤波器到其他各型数字滤波器的频带变换法

前面已经详细介绍了模拟低通滤波器的设计方法；基于冲激响应不变法的数字低通滤波器的设计方法；基于双线性 Z 变换的数字低通滤波器的设计方法。在此基础上很容易得到数字高通、数字带通、数字带阻滤波器的设计方法。

若给定数字滤波器的低通原型系统函数 $H_L(z)$ ，同样可以通过一定的变换，来设计其他各种不同类型的数字滤波器系统函数 $H_d(Z)$ ，这种变换是将 $H_L(z)$ 的 z 平面映射变换到 $H_d(Z)$ 的 Z 平面，该映射关系为

$$z^{-1} = G(Z^{-1}) \quad (6.17)$$

则有

$$H_d(Z) = H_L(z) \Big|_{z^{-1}=G(Z^{-1})} \quad (6.18)$$

对变换函数 $G(Z^{-1})$ 的要求如下：

- z 平面的单位圆必须映射到 Z 平面的单位圆上；

- z 平面的单位圆内部必须映射到 Z 平面的单位圆内部；
- $G(Z^{-1})$ 必须是 Z^{-1} 的有理函数。

只有满足上述要求，才能使一个因果稳定的数字低通有理系统函数 $H_L(z)$ 变换成因果稳定的有理系统函数 $H_d(Z)$ 。

6.4.1 数字低通—数字低通

此时 $H_L(e^{j\theta})$ 和 $H_d(e^{j\omega})$ 都是低通系统函数，只不过截止频率不相同，因而 θ 从 0 变到 π 时，相应的 ω 也从 0 变到 π ，变换函数为

$$z^{-1} = G(Z^{-1}) = \frac{Z^{-1} - \alpha}{1 - \alpha Z^{-1}} \quad (6.19)$$

变换关系式 (6.19) 的边界条件为

$$\begin{aligned} z = 1 &\rightarrow Z = 1 \\ z = e^{j\theta_c} &\rightarrow Z = e^{j\omega_c} \end{aligned} \quad (6.20)$$

将 $z = e^{j\theta_c}$ ， $Z = e^{j\omega_c}$ 代入式 (6.19) 可得

$$\alpha = \frac{\sin\left(\frac{\theta_c - \omega_c}{2}\right)}{\sin\left(\frac{\theta_c + \omega_c}{2}\right)} \quad (6.21)$$

这样就确定了整个变换函数。因此，由式 (6.19) 及式 (6.21)，就可由已有的数字低通滤波器系统函数 $H_L(z)$ (截止频率为 θ_c) 得到新的数字低通滤波器系统函数 $H_d(Z)$ (截止频率为 ω_c)，即

$$H_d(Z) = H_L(z) \Big|_{z^{-1} = \frac{Z^{-1} - \alpha}{1 - \alpha Z^{-1}}} \quad (6.22)$$

6.4.2 数字低通—数字高通

低通变换为高通只需将低通频率响应在单位圆上旋转 180° ，也就是将 Z 变换成 $-Z$ 即可，这就是旋转变换。只需将式 (6.18) 中的 Z^{-1} 用 $-Z^{-1}$ 代替，就完成了数字低通到数字高通 (Z) 的变换，即

$$z^{-1} = \frac{-Z^{-1} - \alpha}{1 + \alpha Z^{-1}} \quad (6.23)$$

变换关系式 (6.23) 的边界条件为

$$\begin{aligned} z = 1 &\rightarrow Z = -1 \\ z = e^{j\theta_c} &\rightarrow Z = e^{-j\omega_c} \end{aligned} \quad (6.24)$$

将 $z = e^{j\theta_c}$ ， $Z = e^{-j\omega_c}$ 代入式 (6.23) 可得

$$\alpha = \frac{\cos\left(\frac{\theta_c + \omega_c}{2}\right)}{\cos\left(\frac{\theta_c - \omega_c}{2}\right)} \quad (6.25)$$

6.4.3 数字低通—数字带通

对于低通到带通的变换，变换后 ω 在 $(-\pi, \pi)$ 之间或 $(0, 2\pi)$ 之间形成两个通带，变换函数为

$$z^{-1} = -\frac{Z^{-2} + d_1 Z^{-1} + d_2}{d_2 Z^{-2} + d_1 Z^{-1} + 1} \quad (6.26)$$

变换关系式 (6.26) 的边界条件为

$$\begin{aligned} z = e^{-j\theta_c} &\rightarrow Z = e^{j\omega_1} \\ z = e^{j\theta_c} &\rightarrow Z = e^{j\omega_2} \\ z = 1 &\rightarrow Z = e^{\pm j\omega_1} \end{aligned} \quad (6.27)$$

将 $z = e^{-j\theta_c} \rightarrow Z = e^{j\omega_1}$ ， $z = e^{j\theta_c} \rightarrow Z = e^{j\omega_2}$ 代入式 (6.27) 可得

$$\begin{aligned} d_1 &= \frac{-2 \cos\left(\frac{\omega_2 + \omega_1}{2}\right) / \cos\left(\frac{\omega_2 - \omega_1}{2}\right)}{1 + \cot\left(\frac{\theta_c}{2}\right) \tan\left(\frac{\omega_2 - \omega_1}{2}\right)} \\ d_2 &= \frac{1 - \cot\left(\frac{\theta_c}{2}\right) \tan\left(\frac{\omega_2 - \omega_1}{2}\right)}{1 + \cot\left(\frac{\theta_c}{2}\right) \tan\left(\frac{\omega_2 - \omega_1}{2}\right)} \end{aligned} \quad (6.28)$$

其中 ω_2 ， ω_1 分别为带通滤波器要求的上、下截止频率，中心频率为

$$\cos \omega_0 = \frac{\cos\left(\frac{\omega_2 + \omega_1}{2}\right)}{\cos\left(\frac{\omega_2 - \omega_1}{2}\right)} \quad (6.29)$$

6.4.4 数字低通—数字带阻

对于低通到带阻的变换，变换后 ω 在 $(-\pi, \pi)$ 之间或 $(0, 2\pi)$ 之间形成两个阻带，变换函数为

$$z^{-1} = \frac{Z^{-2} + d_1 Z^{-1} + d_2}{d_2 Z^{-2} + d_1 Z^{-1} + 1} \quad (6.30)$$

变换关系式 (6.29) 的边界条件为

$$\begin{aligned} z &= e^{-j\theta_c} \rightarrow Z = e^{j\omega_2} \\ z &= e^{j\theta_c} \rightarrow Z = e^{j\omega_1} \\ z &= 1 \rightarrow Z = \pm 1 \end{aligned} \quad (6.31)$$

与导出数字带通滤波器相同方法可得

$$\begin{aligned} d_1 &= \frac{-2 \cos\left(\frac{\omega_2 + \omega_1}{2}\right) / \cos\left(\frac{\omega_2 - \omega_1}{2}\right)}{1 + \tan\left(\frac{\theta_c}{2}\right) \tan\left(\frac{\omega_2 - \omega_1}{2}\right)} \\ d_2 &= \frac{1 - \tan\left(\frac{\theta_c}{2}\right) \tan\left(\frac{\omega_2 - \omega_1}{2}\right)}{1 + \tan\left(\frac{\theta_c}{2}\right) \tan\left(\frac{\omega_2 - \omega_1}{2}\right)} \end{aligned} \quad (6.32)$$

$$\cos \omega_0 = \frac{\cos\left(\frac{\omega_2 + \omega_1}{2}\right)}{\cos\left(\frac{\omega_2 - \omega_1}{2}\right)} \quad (6.33)$$

其中 ω_2, ω_1 分别为带阻滤波器阻带边沿的上、下两个通带截止频率， ω_0 为阻带中心频率。

6.5 应用 MATLAB 设计 IIR 数字滤波器

目前，设计 IIR 数字滤波器的通用方法是先设计相应的低通滤波器，然后再通过双线性变换法和频率变换得到所需要的数字滤波器。模拟滤波器从功能上有低通、高通、带通及带阻四种，从类型上有巴特沃兹（Butterworth）滤波器、切比雪夫（Chebyshev）I 型滤波器、切比雪夫 II 型滤波器、椭圆（Elliptic）滤波器以及贝塞尔（Bessel）滤波器等。

6.5.1 应用 MATLAB 设计 IIR 数字滤波器相关文件

下面给出与 IIR 数字滤波器设计有关的 MATLAB 文件。

1. buttord.m

用来确定数字低通或模拟低通滤波器的阶次，其调用格式分别是

- (1) $[N, Wn] = \text{buttord}(Wp, Ws, Rp, Rs)$
- (2) $[N, Wn] = \text{buttord}(Wp, Ws, Rp, Rs, 's')$

格式 (1) 对应数字滤波器，式中 Wp, Ws 分别是通带和阻带的截止频率，实际上它们是归一化频率，其值在 $0 \sim 1$ 之间，1 对应抽样频率的一半。对低通和高通滤波器， Wp, Ws 都是标量，对带通和带阻滤波器， Wp, Ws 都是 1×2 的向量。 Rp, Rs 分别是通带和阻带的衰减，单位为 dB。 N 是求出的相应低通滤波器的阶次， Wn 是求出的 3dB 频率，它和

W_p 稍有不同。

格式 (2) 对应模拟滤波器, 式中各个变量有含义和格式 (1) 相同, 但 W_p , W_s 及 W_n 的单位为 rad/s, 因此, 它们实际上是频率。

2. buttap.m

用来设计模拟低通原型滤波器 $G(p)$, 其调用格式是

$$[z, p, k]=\text{buttap}(N)$$

N 是欲设计的低通原型滤波器的阶次, z 、 p 和 k 分别是设计出的 $G(p)$ 的极点、零点及增益。

3. lp2lp.m

4. lp2hp.m

5. lp2bp.m

6. lp2bs.m

以上 4 个文件的功能是将模拟低通原型滤波器 $G(p)$ 分别转换为实际的低通、高通、带通及带阻滤波器。其调用格式分别是

$$(1) [B, A]=\text{lp2lp}(b, a, W_o) \quad \text{或} \quad [B, A]=\text{lp2hp}(b, a, W_o)$$

$$(2) [B, A]=\text{lp2bp}(b, a, W_o, B_w) \quad \text{或} \quad [B, A]=\text{lp2bs}(b, a, W_o, B_w)$$

式中 b, a 分别是模拟低通原型滤波器 $G(p)$ 的分子、分母多项式的系数向量, B, A 分别是转换后的 $H(s)$ 的分子、分母多项式的系数向量; 在格式 (1) 中, W_o 是低通或高通滤波器的截止频率; 在格式 (2) 中, W_o 是带通或带阻滤波器的中心频率, B_w 是其带宽。

7. bilinear.m

实现双线性变换, 即由模拟滤波器 $H(s)$ 得到数字滤波器 $H(z)$ 。其调用格式是

$$[B_z, A_z]=\text{bilinear}(B, A, F_s)$$

式中 B, A 分别是 $H(s)$ 的分子、分母多项式的系数向量, B_z, A_z 分别是 $H(z)$ 的分子、分母多项式的系数向量, F_s 是抽样频率。

8. butter.m

用来直接设计巴特沃兹数字滤波器, 实际上它把 `buttord.m`, `buttap.m`, `lp2lp.m` 及 `bilinear.m` 等文件都包含进去, 从而使设计过程更简捷。其调用格式是

$$(1) [B, A]=\text{butter}(N, W_n)$$

$$(2) [B, A]=\text{butter}(N, W_n, \text{'high'})$$

$$(3) [B, A]=\text{butter}(N, W_n, \text{'stop'})$$

$$(4) [B, A]=\text{butter}(N, W_n, \text{'s'})$$

格式 (1) ~ (3) 用来设计数字滤波器, B, A 分别是 $H(z)$ 的分子、分母多项式的系

数向量， W_n 是通带截止频率，范围在 0~1 之间，1 对应抽样频率的一半。若 W_n 是标量，则格式 (1) 用来设计低通数字滤波器，若 W_n 是 1×2 的向量，则格式 (1) 用来设计数字带通滤波器；格式 (2) 用来设计数字高通滤波器；格式 (3) 用来设计数字带阻滤波器，显然，这时的 W_n 是 1×2 的向量；格式 (4) 用来设计模拟滤波器。

9 . cheb1ord.m

求切比雪夫 I 型滤波器的阶次。

10 . cheb1ap.m

用来设计原型切比雪夫 I 型模拟滤波器。

11 . cheby1.m

直接设计切比雪夫 I 型滤波器。

以上 3 个文件的调用格式和对应的巴特沃兹滤波器的文件类似。

12 . cheb2ord.m

13 . ellipord.m

14 . cheb2ap.m

15 . ellipap.m

16 . besslap.m

17 . cheby2.m

18 . ellip.m

19 . besself.m

以上分别为切比雪夫 II 型滤波器、椭圆及贝塞尔滤波器设计函数。其格式与切比雪夫 I 型滤波器和巴特沃兹滤波器类同。

此外，与本章内容有关的 MATLAB 文件还有：

20 . impinvar.m

用冲激响应不变法实现 ω 到 Ω 及 s 到 z 的转换。

21 . maxflat.m

设计广义巴特沃兹低通滤波器。

22 . yulewalk.m

利用最小平方方法设计 Yule-Walker 滤波器。

6.5.2 应用 MATLAB 设计 IIR 数字滤波器实例

例 6.5 用上述的 MATLAB 文件设计例 6.4 的数字低通滤波器。

解：相应的 MATLAB 程序如下。

```
fp=100;fst=300;Fs=1000;
rp=3;rs=20;
wp=2*pi*fp/Fs;
ws=2*pi*fst/Fs;
Fs=Fs/Fs; % let Fs=1
wap=tan(wp/2);was=tan(ws/2);
[n,wn]=buttord(wap,was,rp,rs,'s')
[z,p,k]=buttap(n);
[bp,ap]=zp2tf(z,p,k)
[bs,as]=lp2lp(bp,ap,wap)
[bz,az]=bilinear(bs,as,Fs/2)
[h,w]=freqz(bz,az,256,Fs*1000);
plot(w,abs(h));grid on;
```

调试运行该程序，运行结果如下：

```
bp=[1,0,0],ap=[1,1.4142,1],bs=[0.1056,0,0],as=[1,0.4595,0.1056]
bz=[0.0675,0.1349,0.06745],az=[1,-1.143,0.4128]
```

与例 6.4 的结果一致。

例 6.6 一个数字系统的抽样频率 $F_s = 2000\text{Hz}$ ，试设计一个为此系统使用的带通数字滤波器 $H_{\text{dtp}}(z)$ ，希望采用巴特沃兹滤波器。要求：(1) 通带范围为 $300 \sim 400\text{Hz}$ ，在带边频率处的衰减不大于 3dB 。(2) 在 200Hz 以下和 500Hz 以上衰减不小于 18dB 。

解：相应的 MATLAB 的程序如下。

```
clear all;
fp=[300 400];fs=[200 500];
rp=3;rs=18;
Fs=2000;
wp=fp*2*pi/Fs;
ws=fs*2*pi/Fs;
%
% Firstly to finish frequency prewarping;
wap=2*Fs*tan(wp./2)
was=2*Fs*tan(ws./2);
[n,wn]=buttord(wap,was,rp,rs,'s');
% Note: 's'!
```

```

[z,p,k]=buttap(n);
[bp,ap]=zp2tf(z,p,k)
%
bw=wap(2)-wap(1)
w0=sqrt(wap(1)*wap(2))
[bs,as]=lp2bp(bp,ap,w0,bw)
%
[h1,w1]=freqs(bp,ap);
figure(1)
plot(w1,abs(h1));grid;
ylabel(' lowpass G(p)')
%
w2=[0:Fs/2-1]*2*pi;
h2=freqs(bs,as,w2);
% Note: z=(2/Ts)(z-1)/(z+1);
[bz1,az1]=bilinear(bs,as,Fs)
[h3,w3]=freqz(bz1,az1,1000,Fs);
figure(2)
plot(w2/2/pi,20*log10(abs(h2)),w3,20*log10(abs(h3)));grid;
ylabel('Bandpass AF and DF')
xlabel(' Hz')
%

```

运行结果如下（其原型和幅频响应如图 6.7 所示）。

```

wap =
  1.0e+003 *
    2.0381    2.9062
bp =
    0    0    1
ap =
    1.0000    1.4142    1.0000
bw =
    868.0683
w0 =
  2.4337e+003
bs =
  1.0e+005 *
    7.5354   -0.0000    0.0000
as =
  1.0e+013 *
    0.0000    0.0000    0.0000    0.0007    3.5083
bz1 =
    0.0201    0.0000   -0.0402    0.0000    0.0201
az1 =
    1.0000   -1.6368    2.2376   -1.3071    0.6414

```

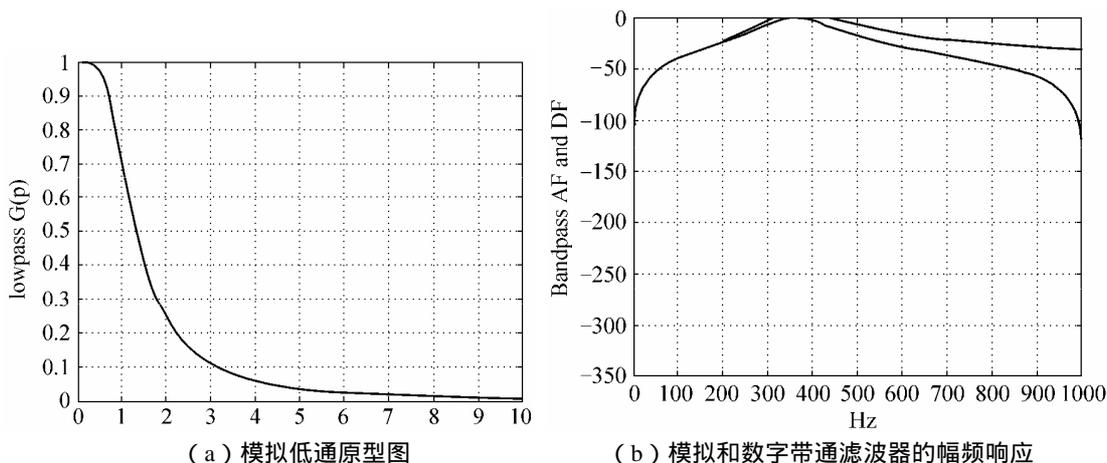


图 6.7 巴特沃兹滤波器

例 6.7 试用 MATLAB 设计一低通数字滤波器，给定技术指标是 $f_p = 100\text{Hz}$, $f_s = 300\text{Hz}$, $\alpha_p = 3\text{dB}$, $\alpha_s = 20\text{dB}$, 抽样频率 $F_s = 1000\text{Hz}$ 。

解：程序如下。

```
clear all;
fp=100;fs=300;Fs=1000;
rp=3;rs=20;
%
wp=2*pi*fp/Fs;
ws=2*pi*fs/Fs;
Fs=Fs/Fs; % let Fs=1
% Firstly to finish frequency prewarping ;
wap=tan(wp/2);was=tan(ws/2); %
[n,wn]=buttord(wap,was,rp,rs,'s') %
% Note: 's'!
[z,p,k]=buttap(n);
[bp,ap]=zp2tf(z,p,k)
[bs,as]=lp2lp(bp,ap,wap)
% Note: s=(2/Ts)(z-1)/(z+1);Ts=1,that is 2fs=1,fs=0.5;
[bz,az]=bilinear(bs,as,Fs/2)
[h,w]=freqz(bz,az,256,Fs*1000);
plot(w,abs(h));grid on;
```

运行结果是：

```
n =
    2
wn =
    0.4363
bp =
```

```

    0    0    1
ap =
    1.0000    1.4142    1.0000
bs =
    0.1056
as =
    1.0000    0.4595    0.1056
bz =
    0.0675    0.1349    0.0675
az =
    1.0000   -1.1430    0.4128

```

幅频响应曲线如图 6.8 所示。

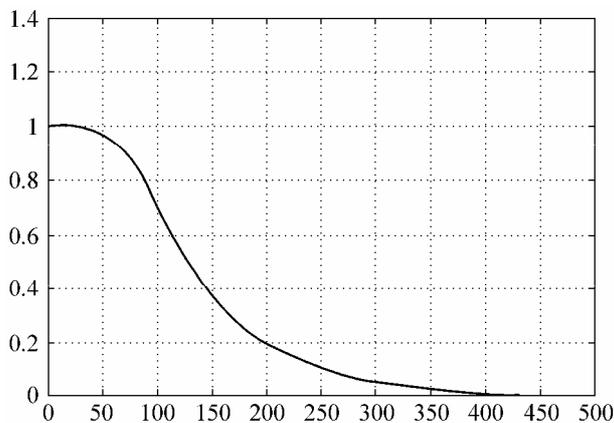


图 6.8 幅频响应曲线

1. 巴特沃兹模拟滤波器的 MATLAB 实现

例 6.8 设计一个模拟巴特沃兹低通滤波器，它在 30rad/s 处具有 1dB 或更好的波动，在 50rad/s 处具有至少 30dB 的衰减。求出级联形式的系统函数，画出滤波器的幅度响应、对数幅度响应、相位响应和脉冲响应图。

解：程序如下。

```

Wp=30;Ws=50;Rp=1;As=30;           %技术指标
Ripple=10^(-Rp/20);
Attn=10^(-As/20);
[b,a]=afd_butt(Wp,Ws,Rp,As)        %巴特沃兹低通滤波器
[C,B,A]=sdir2cas(b,a)              %计算二阶节系数，级联型实现
[db,mag,pha,w]=freqs_m(b,a,50);    %计算幅频响应
[ha,x,t]=impz(b,a);                %计算模拟滤波器的单位脉冲响应
figure(1);clf;
subplot(2,2,1);plot(w,mag);title('Magnitude Response');
xlabel('Analog frequency in rad/s');

```

```

ylabel('H');

axis([0,50,0,1.1])
set(gca,'XTickMode','manual','XTick',[0,30,40,50]);
set(gca,'YTickMode','manual','YTick',[0,Attn,Ripple,1]);
grid
subplot(2,2,2);plot(w,db);title('Magnitude in dB');
xlabel('Analog frequency in rad/s');
ylabel('decibels');

axis([0,50,-40,5])
set(gca,'XTickMode','manual','XTick',[0,30,40,50]);
set(gca,'YTickMode','manual','YTick',[-40,-As,-Rp,0]);
grid
subplot(2,2,3);plot(w,pha/pi);title('Phase Response');
xlabel('Analog frequency in rad/s');
ylabel('radians');

axis([0,50,-1.1,1.1])
set(gca,'XTickMode','manual','XTick',[0,30,40,50]);
set(gca,'YTickMode','manual','YTick',[-1,-0.5,0,0.5,1]);
grid
subplot(2,2,4);plot(t,ha);title('Impulse Response');
xlabel('time in seconds');
ylabel('ha(t)');
axis([0,max(t)+0.05,min(ha),max(ha)+0.025]);
set(gca,'XTickMode','manual','XTick',[0,0.1,max(t)]);
set(gca,'YTickMode','manual','YTick',[0,0.1,max(ha)]);
grid
%巴特沃兹模拟滤波器的设计子程序
function[b,a]=afd_butt(Wp,Ws,Rp,As);
if Wp<=0
    error('Passband edge must be larger than 0')
end
if Ws<=Wp
    error('Stopband edge must be larger than Passed edge')
end
if (Rp<=0)|(As<0)
    error('PB ripple and /or SB attenuation must be larger than 0')
end
N=ceil((log10((10^(Rp/10)-1)/(10^(As/10)-1)))/(2*log10(Wp/Ws)));
fprintf('\n ***Butterworth Filter Order=%2.0f\n',N)
OmegaC=Wp/((10^(Rp/10)-1)^(1/(2*N)));

```

```

[b,a]=u_buttap(N,OmegaC);
%设计非归一化巴特沃兹模拟低通滤波器原型子程序
function [b,a]=u_buttap(N,OmegaC);
[z,p,k]=buttap(N);
p=p*OmegaC;
k=k*OmegaC^N;
B=real(poly(z));
b0=k;
b=k*B;
a=real(poly(p));

%计算系统函数的幅度响应和相位响应子程序
function [db,mag,pha,w]=freqs_m(b,a,wmax);
w=[0:1:500]*wmax/500;
H=freqs(b,a,w);
mag=abs(H);
db=20*log10((mag+eps)/max(mag));
pha=angle(H);

%直接形式转换成级联形式子程序
function [C,B,A]=sdir2cas(b,a);
Na=length(a)-1;Nb=length(b)-1;
b0=b(1);b=b/b0;
a0=a(1);a=a/a0;
C=b0/a0;
p=cplxpair(roots(a));K=floor(Na/2);
if K*2==Na
    A=zeros(K,3);
    for n=1:2:Na
        Arow=p(n:1:n+1,:);Arow=poly(Arow);
        A(fix((n+1)/2),:)=real(Arow);
    end
elseif Na==1
    A=[0 real(poly(p))];
else
    A=zeros(K+1,3);
    for n=1:2:2*K
        Arow=p(n:1:n+1,:);Arow=poly(Arow);
        A(fix((n+1)/2),:)=real(Arow);
    end
    A(K+1,:)= [0 real(poly(p(Na)))];
end
z=cplxpair(roots(b));K=floor(Nb/2);

```

```

if Nb==0
    B=[0 0 poly(z)];
elseif K*2==Nb
    B=zeros(K,3);
    for n=1:2:Nb
        Brow=z(n:1:n+1,:);Brow=poly(Brow);
        B(fix((n+1)/2),:)=real(Brow);
    end
elseif Nb==1
    B=[0 real(poly(z))];
else
    B=zeros(K+1,3);
    for n=1:2:2*K
        Brow=z(n:1:n+1,:);Brow=poly(Brow);
        B(fix((n+1)/2),:)=real(Brow);
    end
    B(K+1,:)= [0 real(poly(z(Nb)))];
end
end

```

运行结果如下 (如图 6.9 所示):

```

***Butterworth Filter Order=11f
b =
    3.8682e+013
a =
    1.0e+013 *

Columns 1 through 6
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001
Columns 7 through 10
    0.0036    0.0613    0.6888    3.8682

C =
    3.8682e+013
B =
    0    0    1
A =
    1.0e+003 *

    0.0010    0.0608    1.0458
    0.0010    0.0495    1.0458
    0.0010    0.0323    1.0458
    0.0010    0.0112    1.0458
    0    0.0010    0.0323

```

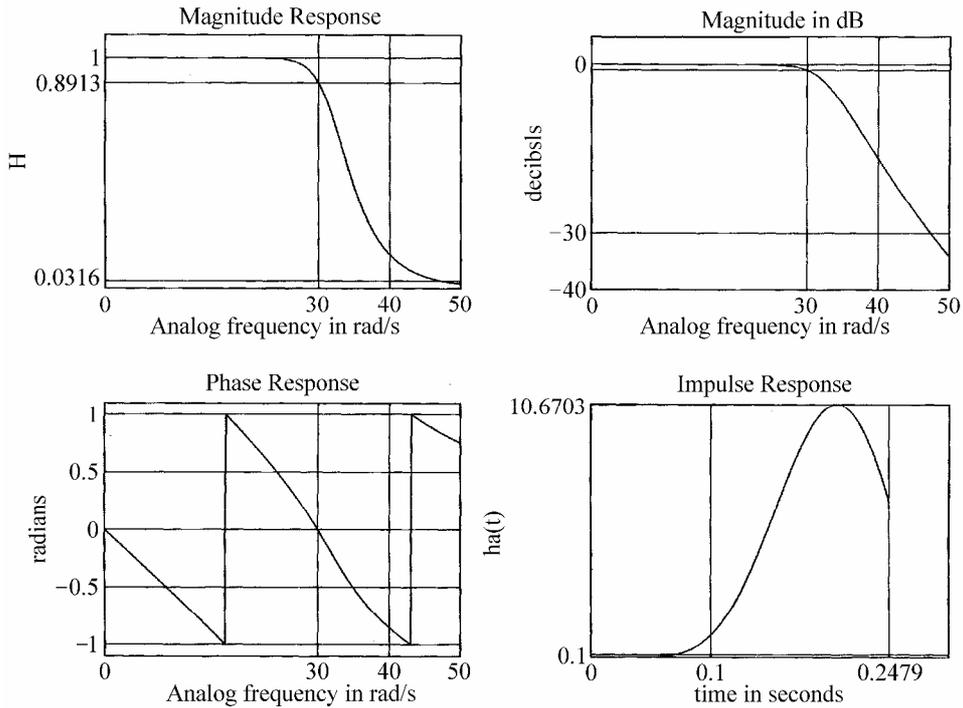


图 6.9 巴特沃兹模拟低通滤波器例

2. 切比雪夫模拟滤波器 MATLAB 的实现

(1) 切比雪夫 I 型模拟滤波器 MATLAB 的实现

例 6.9 设计一个低通切比雪夫 I 型模拟滤波器，满足：通带截止频率 $\Omega_p = 0.2\pi \text{rad/s}$ ，通带波动 $\delta = 1\text{dB}$ ；阻带截止频率 $\Omega_s = 0.3\pi \text{rad/s}$ ，阻带衰减 $A_s = 16\text{dB}$ 。

解：程序如下。

```

Omegap=0.2*pi;Omegar=0.3*pi;Dt=1;Ar=16; %技术指标
[b,a]=afd_chb1(Omegap,Omegar,Dt,Ar); %切比雪夫 I 型模拟低通滤波器
[C,B,A]=sdir2cas(b,a) %级联形式
[db,mag,pha,w]=freqs_m(b,a,pi); %计算幅频响应
[ha,x,t]=impz(b,a); %计算模拟滤波器的单位脉冲响应
subplot(221);plot(w/pi,mag);title('幅度响应 |Ha(j\Omega)|');
subplot(222);plot(w/pi,db);title('幅度响应(dB)');
subplot(223);plot(w/pi,pha/pi);title('相位响应');axis([0,1,-1,1]);
subplot(224);plot(t,ha);title('单位脉冲响应 ha(t)');
axis([0,max(t),min(ha),max(ha)]);

```

%切比雪夫 I 型模拟滤波器的设计子程序

```

function[b,a]=afd_chb1(Omegap,Omegar,Dt,Ar);
if Omegap<=0

```

```

    error('通带边缘必须大于 0')
end
if Omegar<=Omegap
    error('阻带边缘必须大于通带边缘')
end
if (Dt<=0)|(Ar<0)
    error('通带波动或阻带衰减必须大于 0')
end
ep=sqrt(10^(Dt/10)-1);
A=10^(Ar/20);
OmegaC=Omegap;
OmegaR=Omegar/Omegap;
g=sqrt(A*A-1)/ep;
N=ceil(log10(g+sqrt(g*g-1))/log10(OmegaR+sqrt(OmegaR*OmegaR-1)));
fprintf('\n***切比雪 I 型模拟低通滤波器阶次=%2.0f\n',N);
[b,a]=u_chblap(N,Dt,OmegaC);

```

%设计非归一化切比雪夫 I 型模拟低通滤波器原型子程序

```

function [b,a]=u_chblap(N,Dt,OmegaC);
[z,p,k]=cheblap(N,Dt);
a=real(poly(p));
aNn=a(N+1);
p=p*OmegaC;
a=real(poly(p));
aNu=a(N+1);
k=k*aNu/aNn;
b0=k;
B=real(poly(z));
b=k*B;

```

另外，计算系统函数的幅度响应和相位响应子程序和直接形式转换成级联形式子程序如例 6.8 所示。

运行结果如下（如图 6.10 所示）：

```

***切比雪 I 型模拟低通滤波器阶次= 4
C =
    0.0383
B =
    0    0    1
A =
    1.0000    0.4233    0.1103
    1.0000    0.1753    0.3895

```

(2) 切比雪夫 II 型模拟滤波器 MATLAB 的实现

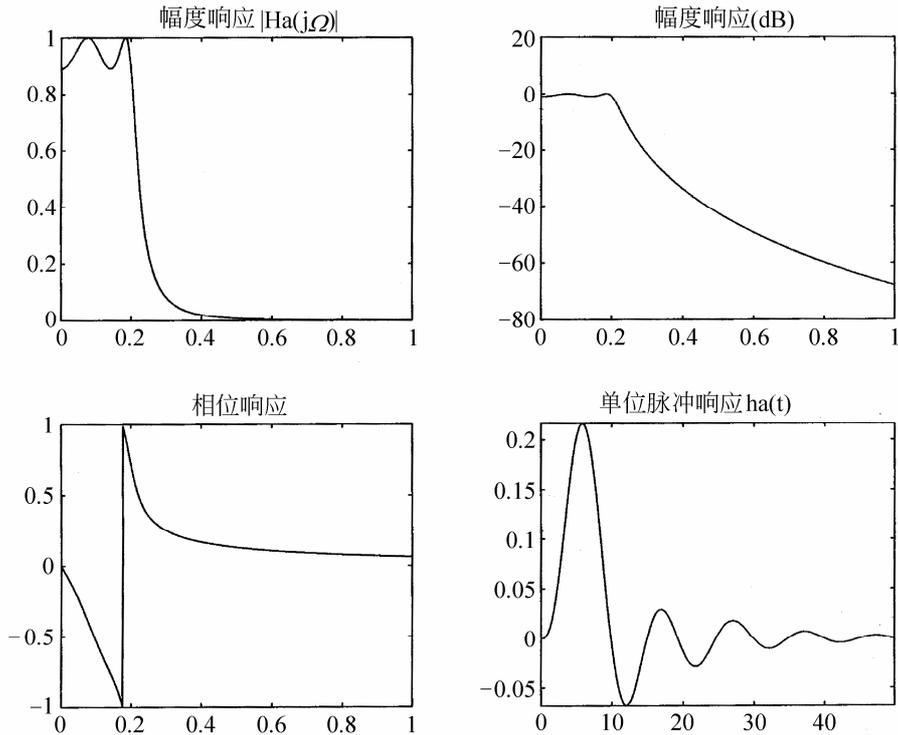


图 6.10 切比雪夫 I 型模拟滤波器例

例 6.10 设计一个低通切比雪夫 II 型模拟滤波器, 满足 通带截止频率 $\Omega_p = 0.2\pi \text{rad/s}$, 通带波动 $\delta = 1\text{dB}$; 阻带截止频率 $\Omega_s = 0.2\pi \text{rad/s}$, 阻带衰减 $A_s = 16\text{dB}$ 。

解: 程序如下。

```

Omegap=0.2*pi;Omegar=0.3*pi;Dt=1;Ar=16;
[b,a]=afd_chb2(Omegap,Omegar,Dt,Ar);
[C,B,A]=sdir2cas(b,a)
[db,mag,pha,w]=freqs_m(b,a,pi);
[ha,x,t]=impulse(b,a);
subplot(221);plot(w/pi,mag);title('幅度响应 |Ha(j\Omega)|');
subplot(222);plot(w/pi,db);title('幅度响应(dB)');
subplot(223);plot(w/pi,pha/pi);title('相位响应');axis([0,1,-1,1]);
subplot(224);plot(t,ha);title('单位脉冲响应 ha(t)');
axis([0,max(t),min(ha),max(ha)]);

```

%切比雪夫 II 型模拟滤波器的设计子程序

```

function[b,a]=afd_chb2(Omegap,Omegar,Dt,Ar);
if Omegap<=0
    error('通带边缘必须大于 0')
end
if Omegar<=Omegap

```

```

    error('阻带边缘必须大于通带边缘')
end
if (Dt<=0)|(Ar<0)
    error('通带波动或阻带衰减必须大于0')
end
ep=sqrt(10^(Dt/10)-1);
A=10^(Ar/20);
OmegaC=Omegap;
OmegaR=Omegar/Omegap;
g=sqrt(A*A-1)/ep;
N=ceil(log10(g+sqrt(g*g-1))/log10(OmegaR+sqrt(OmegaR*OmegaR-1)));
fprintf('\n***切比雪夫 II 型模拟低通滤波器阶次=%2.0f\n',N);
[b,a]=u_chb2ap(N,Ar,OmegaC);

```

%设计非归一化切比雪夫 II 型模拟低通滤波器原型子程序

```

function [b,a]=u_chb2ap(N,Ar,OmegaC);
[z,p,k]=cheb2ap(N,Ar);
a=real(poly(p));
aNn=a(N+1);
p=p*OmegaC;
a=real(poly(p));
aNu=a(N+1);
k=k*aNu/aNn;
b0=k;
B=real(poly(z));
b=k*B;

```

另外，计算系统函数的幅度响应和相位响应子程序和直接形式转换成级联形式子程序如例 6.1 所示。

运行结果如下（如图 6.11 所示）。

```

***切比雪夫 II 型模拟低通滤波器阶次= 4
C =
    0.0247
B =
    1.0000         0    6.8284
    1.0000         0    1.1716
A =
    1.0000    1.3014    0.6554
    1.0000    0.2480    0.3015

```

3. 模拟到数字滤波器的设计 MATLAB 的实现

(1) 脉冲响应不变法的 MATLAB 的实现

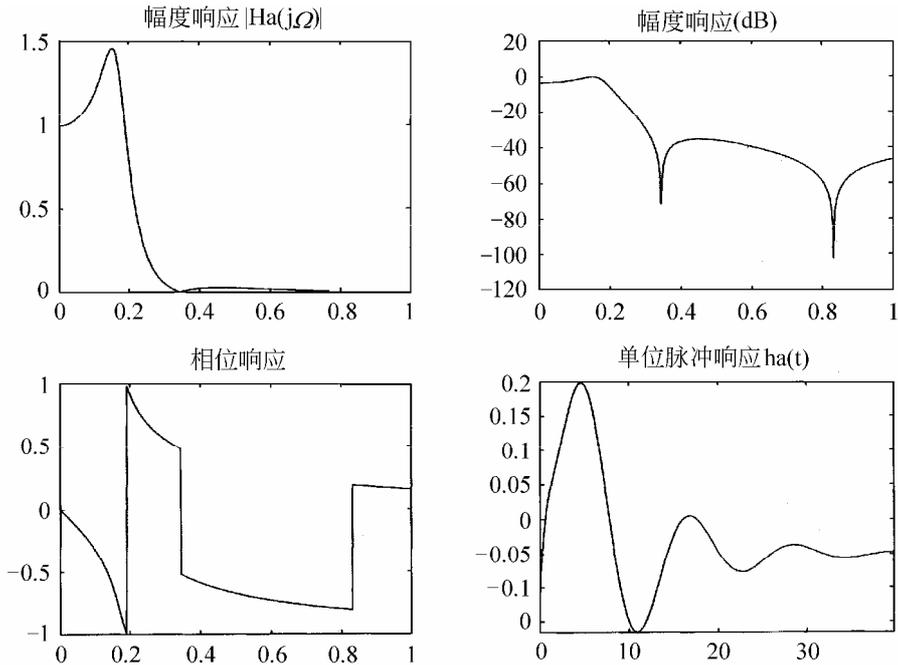


图 6.11 切比雪夫 II 型模拟滤波器例

例 6.11 设计低通数字滤波器，要求在通带内频率低于 $0.2\pi\text{rad}$ 时，允许幅度误差在 1dB 以内，在频率 $0.3\pi\text{rad} \sim \pi\text{rad}$ 之间的阻带衰减大于 15dB。用脉冲响应不变法设计数字滤波器， $T=1$ ，模拟滤波器采用切比雪夫 I 型滤波器原型。

解：模拟滤波器采用切比雪夫 I 型滤波器原型程序如下。

```

Wp=0.2*pi;Wr=0.3*pi;Ap=1;Ar=15;T=1;
Omegap=Wp/T;Omegar=Wr/T;
[cs,ds]=afd_chb1(Omegap,Omegar,Ap,Ar)
[C,B,A]=sdir2cas(cs,ds);
[db,mag,pha,Omega]=freqs_m(cs,ds,pi);
subplot(234);plot(Omega/pi,mag);title('模拟滤波器幅度响应|Ha(j\Omega)|');
[b,a]=imp_invr(cs,ds,T);
[h,n]=impz(b,a);
[C,B,A]=dir2par(b,a)
[db,mag,pha,grd,w]=freqz_m(b,a);
subplot(231);plot(w/pi,mag);title('数字滤波器幅度响应|Ha(j\Omega)|');
subplot(232);plot(w/pi,db);title('数字滤波器幅度响应(dB)');
subplot(233);plot(w/pi,pha/pi);title('数字滤波器相位响应');
subplot(235);plot(n,h);title('脉冲响应');

```

%脉冲响应不变法子程序

```

function [b,a]=imp_invr(c,d,T)
[R,p,k]=residue(c,d);
p=exp(p*T);

```

```
[b,a]=residuez(R,p,k);
b=real(b).*T;
a=real(a);
```

%数字滤波器响应子程序

```
function [db,mag,pha,grd,w]=freqz_m(b,a);
[H,w]=freqz(b,a,1000,'whole');
H=(H(1:501))';w=(w(1:501))';
mag=abs(H);
db=20*log10((mag+eps)/max(mag));
pha=angle(H);
grd=grpdelay(b,a,w);
```

%直接型转换成并联型子程序

```
function [C,B,A]=dir2par(b,a);
M=length(b);
N=length(a);
[r1,p1,C]=residuez(b,a);
p=cplxpair(p1,10000000*eps);
I=cplxcomp(p1,p);
r=r1(I);
K=floor(N/2);B=zeros(K,2);A=zeros(K,3);
if K*2==N;
    for i=1:2:(N-2)
        Brow=r(i:1:(i+1),:);
        Arow=p(i:1:(i+1),:);
        [Brow,Arow]=residuez(Brow,Arow,[]);
        B(fix((i+1)/2),:)=real(Brow);
        A(fix((i+1)/2),:)=real(Arow);
    end
    [Brow,Arow]=residuez(r(N-1),p(N-1),[]);
    B(K,:)=[real(Brow) 0];A(K,:)=real(Arow) 0];
else
    for i=1:2:(N-1)
        Brow=r(i:1:(i+1),:);
        Arow=p(i:1:(i+1),:);
        [Brow,Arow]=residuez(Brow,Arow,[]);
        B(fix((i+1)/2),:)=real(Brow);
        A(fix((i+1)/2),:)=real(Arow);
    end
end
end
```

%比较两个含同样标量元素但(可能)有不同下标的复数对其相应留数向量子程序

```
function I=cplxcomp(p1,p2);
I=[];
```

```

for j=1:length(p2)
    for i=1:length(p1)
        if (abs(p1(i)-p2(j))<0.0001)
            I=[I,i];
        end
    end
end
end
end

```

另外，其他一些子程序见前面例子所示。

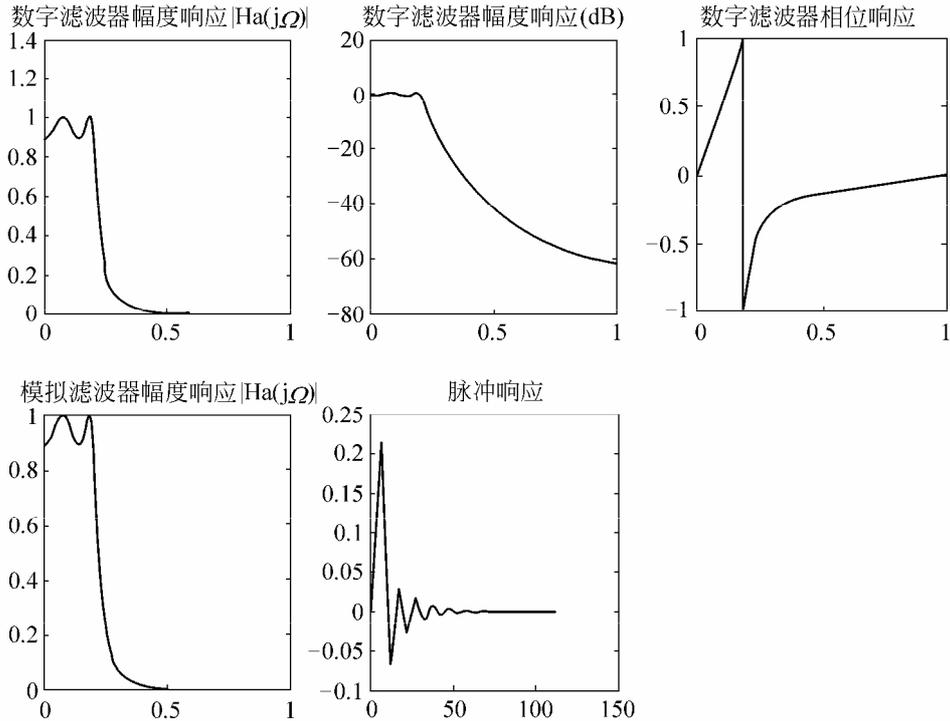


图 6.12 脉冲响应不变法设计的切比雪夫 I 型数字滤波器

运行结果如下所示（如图 6.12 所示）：

```

***切比雪 I 型模拟低通滤波器阶次= 4
cs =
    0.0383
ds =
    1.0000    0.5987    0.5740    0.1842    0.0430
C =
    []
B =
   -0.0833   -0.0246
    0.0833    0.0239
A =

```

```

1.0000 -1.4934 0.8392
1.0000 -1.5658 0.6549

```

(2) 双线性变换法 MATLAB 的实现

例 6.12 设计低通数字滤波器，要求在通带内频率低于 $0.2\pi\text{rad}$ 时，允许幅度误差在 1dB 以内，在频率 $0.3\pi\text{rad} \sim \pi\text{rad}$ 之间的阻带衰减大于 15dB。用脉冲响应不变法设计数字滤波器， $T=1$ ，模拟滤波器采用巴特沃兹滤波器原型。

解：程序如下。

```

Wp=0.2*pi;Wr=0.3*pi;Ap=1;Ar=15;T=1;
Omegap=(2/T)*tan(Wp/2);Omegar=(2/T)*tan(Wr/2);
[cs,ds]=afd_butt(Omegap,Omegar,Ap,Ar)
[C,B,A]=sdir2cas(cs,ds);
[db,mag,pha,Omega]=freqs_m(cs,ds,pi);
subplot(234);plot(Omega/pi,mag);title('模拟滤波器幅度响应|Ha(j\Omega)|');
[b,a]=bilinear(cs,ds,T); % 双线性变换法设计
[h,n]=impz(b,a);
[C,B,A]=dir2cas(b,a)
[db,mag,pha,grd,w]=freqz_m(b,a);
subplot(231);plot(w/pi,mag);title('数字滤波器幅度响应|Ha(j\Omega)|');
subplot(232);plot(w/pi,db);title('数字滤波器幅度响应(dB)');
subplot(233);plot(w/pi,pha/pi);title('数字滤波器相位响应');
subplot(235);plot(n,h);title('脉冲响应');
delta_w=2*pi/1000;
Ap=-(min(db(1:1:Wp/delta_w+1)))
Ar=-round(max(db(Wr/delta_w+1:1:501)))

```

运行结果如下（如图 6.13 所示）：

```

***Butterworth Filter Order= 6f
cs =
    0.1480
ds =
    Columns 1 through 5
    1.0000    2.8100    3.9482    3.5168    2.0884
    Columns 6 through 7
    0.7862    0.1480
C =
    5.7969e-004
B =
    1.0000    2.0284    1.0287
    1.0000    1.9997    0.9999
    1.0000    1.9719    0.9722
A =
    1.0000   -0.9459    0.2342
    1.0000   -1.0541    0.3753

```

```

1.0000   -1.3143    0.7149
Ap =
1.0000
Ar =
18

```

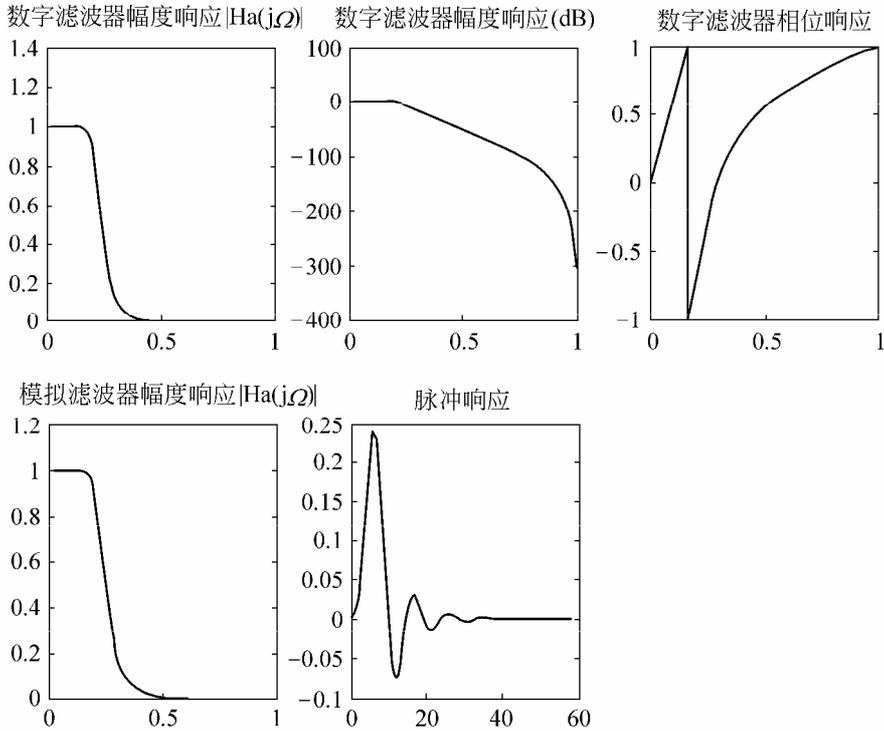


图 6.13 双线性变换法设计的巴特沃兹型数字滤波器

(3) 数字滤波器的原型变换法 MATLAB 的实现

模拟低通滤波器变换成数字高通滤波器

例 6.13 设计一个巴特沃兹高通滤波器，要求通带截止频率为 0.6π ，通带内衰减不大于 1 dB，阻带起始频率为 0.4π ，阻带内衰减不小于 15 dB， $T=1$ 。

解：程序如下。

```

Wp=0.6*pi;Wr=0.4*pi;Ap=1;Ar=15;T=1;
[N,Wn]=buttord(Wp/pi,Wr/pi,Ap,Ar) %计算巴特沃兹滤波器阶次和截止频率
[b,a]=butter(N,Wn,'high');% 频率变换法设计巴特沃兹高通滤波器
[C,B,A]=dir2cas(b,a)
[db,mag,pha,grd,w]=freqz_m(b,a);
subplot(211);plot(w/pi,mag);title('数字巴特沃兹高通滤波器幅度响应 |Ha(j\Omega)|');
subplot(222);plot(w/pi,db);title('数字巴特沃兹高通滤波器幅度响应 (dB)');

```

运行结果如下（如图 6.14 所示）：

```

N =
    4
Wn =
    0.5344
C =
    0.0751
B =
    1.0000  -2.0000  1.0000
    1.0000  -2.0000  1.0000
A =
    1.0000  0.1562  0.4488
    1.0000  0.1124  0.0425

```

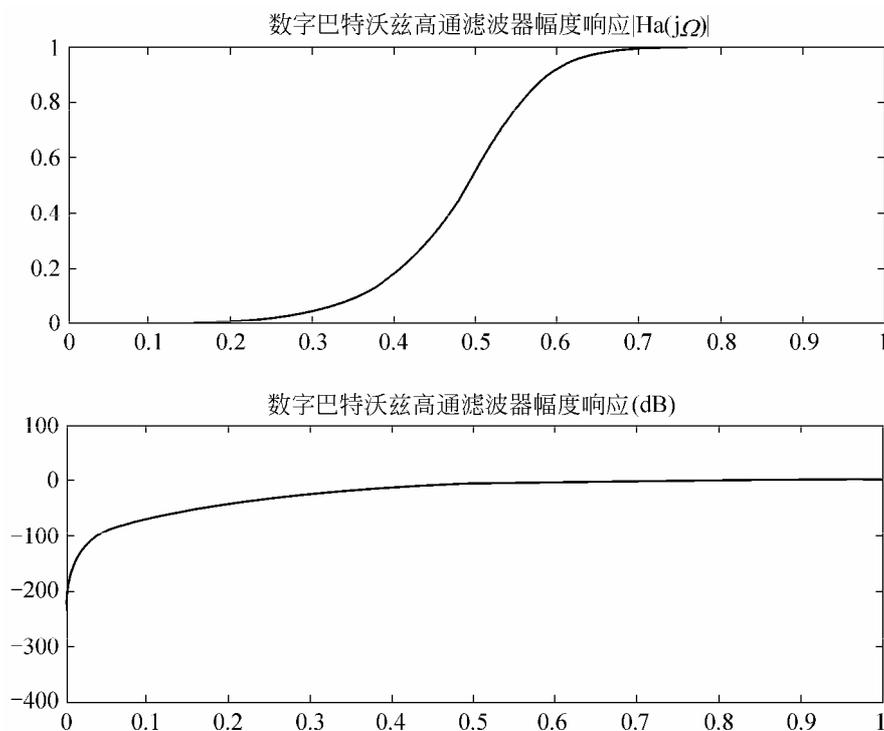


图 6.14 巴特沃兹型数字高通滤波器

模拟低通滤波器变换数字带通滤波器

例 6.14 设计一个巴特沃兹带通滤波器，要求通带上下截止频率为 0.4π 、 0.3π ，通带内衰减不大于 3dB，阻带上下起始频率为 0.5π 、 0.2π ，阻带内衰减不小于 18dB。

解：程序如下。

```

Wp=[0.3*pi,0.4*pi];Wr=[0.2*pi,0.5*pi];Ap=3;Ar=18;
[N,Wn]=buttord(Wp/pi,Wr/pi,Ap,Ar); %计算巴特沃兹滤波器阶次和截止频率
[b,a]=butter(N,Wn,'bandpass') % 频率变换法设计巴特沃兹带通滤波器
[C,B,A]=dir2cas(b,a)

```

```
[db,mag,pha,grd,w]=freqz_m(b,a);
subplot(211);plot(w/pi,mag);title('数字巴特沃兹带通滤波器幅度响应|Ha(j\Omega)|');
subplot(212);plot(w/pi,db);title('数字巴特沃兹带通滤波器幅度响应(dB)');
```

运行结果如下所示 (如图 6.15 所示):

```
b =
    0.0213         0   -0.0426         0    0.0213
a =
    1.0000   -1.6303    2.2183   -1.2919    0.6320

C =
    0.0213
B =
    1.0000   -2.0000    1.0000
    1.0000    2.0000    1.0000
A =
    1.0000   -0.6249    0.7840
    1.0000   -1.0053    0.8061
```

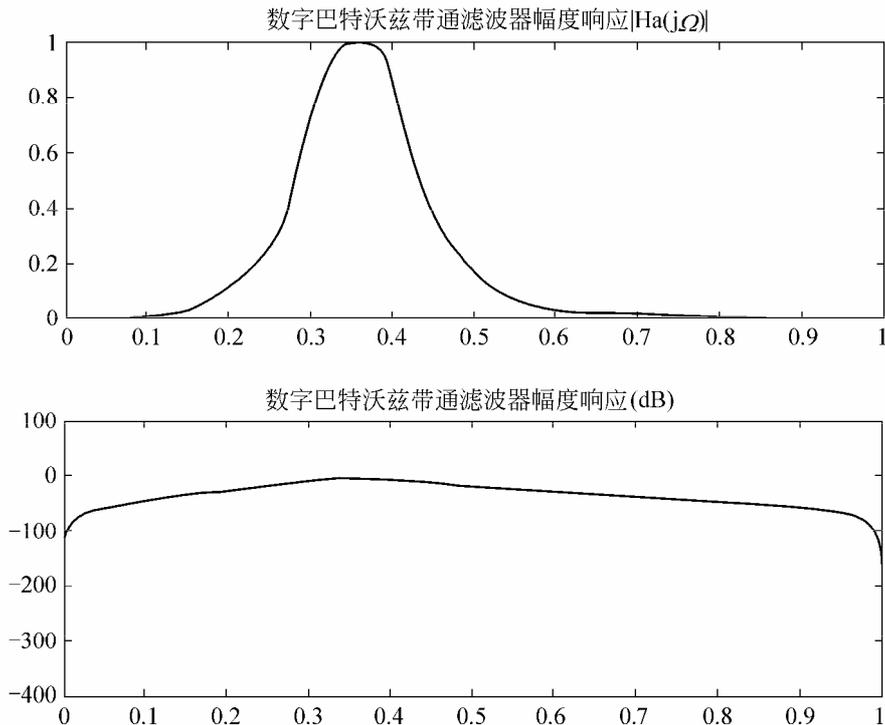


图 6.15 巴特沃兹型数字带通滤波器

模拟低通滤波器变换成数字带阻滤波器

例 6.15 设计一个巴特沃兹带阻滤波器, 要求通带上下截止频率为 0.8π 、 0.2π , 通带内衰减不大于 1dB, 阻带上下起始频率为 0.7π 、 0.4π , 阻带内衰减不小于 30dB。

解：程序如下。

```
Wp=[0.2*pi,0.8*pi];Wr=[0.4*pi,0.7*pi];Ap=1;Ar=30;
[N,Wn]=buttord(Wp/pi,Wr/pi,Ap,Ar); %计算巴特沃兹滤波器阶次和截止频率
[b,a]=butter(N,Wn,'stop'); % 频率变换法设计巴特沃兹带阻滤波器
[C,B,A]=dir2cas(b,a)
[db,mag,pha,grd,w]=freqz_m(b,a);
subplot(211);plot(w/pi,mag);title('数字巴特沃兹带阻滤波器幅度响应|Ha(j\Omega)|');
subplot(212);plot(w/pi,db);title('数字巴特沃兹带阻滤波器幅度响应(dB)');
```

运行结果如下所示（如图 6.16 所示）：

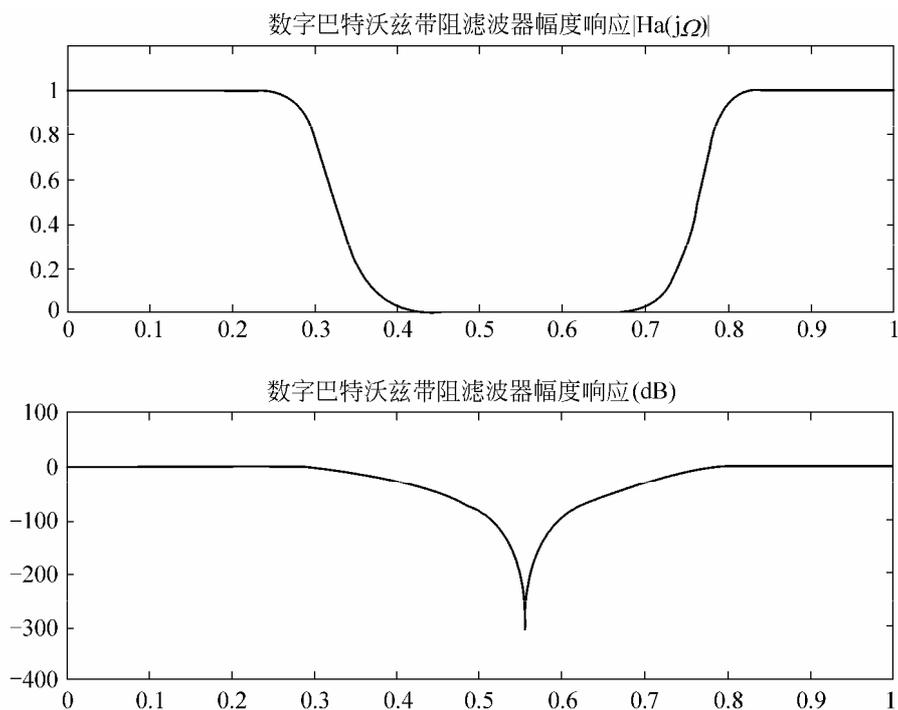


图 6.16 巴特沃兹型数字带阻滤波器

```
C =
    0.0394
B =
    1.0000    0.3564    1.0023
    1.0000    0.3549    0.9969
    1.0000    0.3526    1.0054
    1.0000    0.3497    0.9946
    1.0000    0.3473    1.0032
    1.0000    0.3458    0.9977
    1.0000    1.3568    0.7928
    1.0000    1.0330    0.4633
```

```

1.0000    0.6180    0.1775
1.0000   -0.2492    0.1113
1.0000   -0.6616    0.3755
1.0000   -0.9782    0.7446

```

数字低通到数字滤波器的设计

为了从低通数字滤波原型得到新数字滤波器的有理函数，必须实现有理代换，通常采用 `zmapping` 函数实现。该函数如下。

```

function [bz,az]=zmapping(bZ,aZ,Nz,Dz);
bzord=(length(bZ)-1)*(length(Nz)-1);
azord=(length(aZ)-1)*(length(Dz)-1);
bz=zeros(1,bzord+1);
for k=0:bzord
    pln=[1];
    for l=0:k-1
        pln=conv(pln,Nz);
    end
    pld=[1];
    for l=0:azord-k-1
        pld=conv(pld,Dz);
    end
    bz=bz+bZ(k+1)*conv(pln,pld);
end
az=zeros(1,azord+1);
for k=0:azord
    pln=[1];
    for l=0:k-1
        pln=conv(pln,Nz);
    end
    pld=[1];
    for l=0:azord-k-1
        pld=conv(pld,Dz);
    end
    az=az+aZ(k+1)*conv(pln,pld);
end
az1=az(1);az=az/az1;bz=bz/az1;

```

例 6.16 试用 `zmapping` 函数实现例 6.12 中的高通滤波器。

解：程序如下。

```

wplp=0.2*pi;%低通数字通带频率
wrlp=0.3*pi;%低通数字阻带频率
Ap=1;%通带波动
Ar=15;%阻带波动
T=1;
Omegap=(2/T)*tan(wplp/2);%低通原型通带频率

```

```

OmeGAR=(2/T)*tan(wrlp/2);%低通原型阻带频率
[cs,ds]=afd_chb1(OmeGAp,OmeGAR,Ap,Ar);%模拟切比雪夫 I 型低通滤波器
[blp,alp]=bilinear(cs,ds,T);%双线性变换求低通数字滤波器
wphp=0.6*pi;%高通数字通带频率
alpha=-((cos((wplp+wphp)/2))/(cos((wplp-wphp)/2))) %低通到高通系数
Nz=-[alpha,1];Dz=[1,alpha];%变换关系
[bhp,ahp]=zmapping(blp,alp,Nz,Dz);%数字高通滤波器系统函数
[C,B,A]=dir2cas(bhp,ahp)%级联实现
[db,mag,pha,grd,w]=freqz_m(blp,alp);%低通频率响应
[dbl,mag1,pha1,grd1,w1]=freqz_m(bhp,ahp);%高通频率响应
subplot(221);plot(w/pi,mag);title('低通滤波器幅度响应|Ha(j\Omega)|');
subplot(222);plot(w/pi,db);title('低通滤波器幅度响应(dB)');
subplot(223);plot(w1/pi,mag1);title('高通滤波器幅度响应|Ha(j\Omega)|');
subplot(224);plot(w1/pi,dbl);title('高通滤波器幅度响应(dB)');

```

运行结果如下所示(如图 6.17 所示):

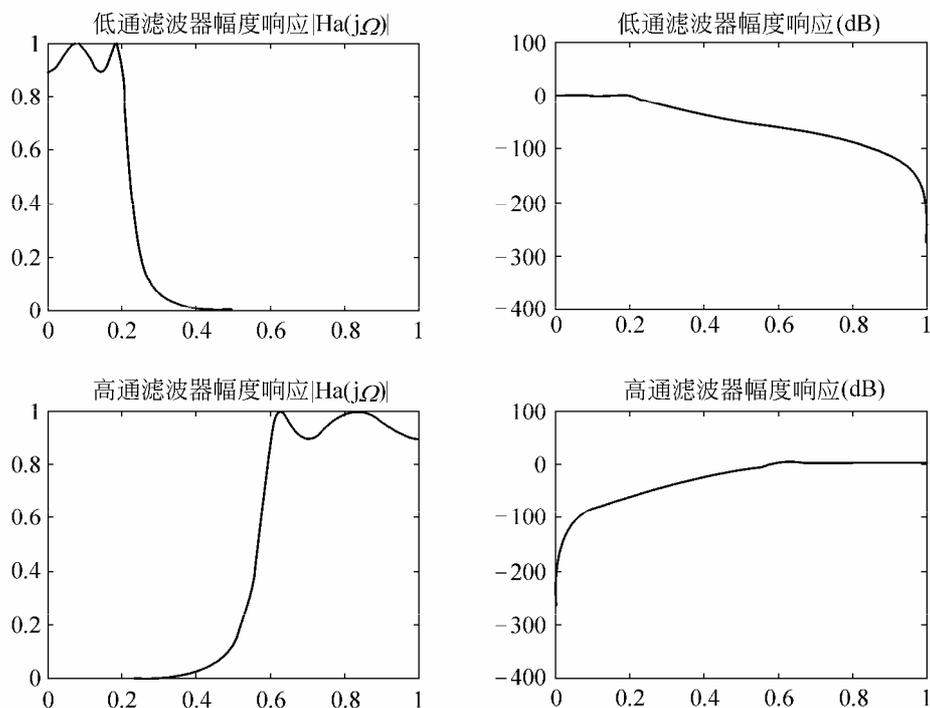


图 6.17 数字域变换法设计高通滤波器

```

***切比雪 I 型模拟低通滤波器阶次= 4
alpha =
    -0.3820
C =
    0.0243
B =
    1.0000   -2.0000    1.0000

```

```

1.0000 -2.0000 1.0000
A =
1.0000 1.0416 0.4019
1.0000 0.5561 0.7647

```

4. IIR 滤波器的应用实例

例 6.17 一个含有高频噪声的信号保存在数据文件 noisbloc 中，试用脉冲响应不变法设计低通滤波器对其进行除噪。

解：程序如下。

```

Wp=0.2*pi;Wr=0.4*pi;Ap=0.25;Ar=50;
T=1;Omegap=Wp/T;Omegar=Wr/T;
[cs,ds]=afd_chb1(Omegap,Omegar,Ap,Ar);
[b,a]=imp_invr(cs,ds,T);
[db,mag,pha,grd,w]=freqz_m(b,a);
load noisbloc;indx=1:1000;
x=noisbloc(indx);
subplot(211);plot(x);title('含噪声的输入信号');
y=filter(b,a,x);
subplot(212);plot(y);title('除噪后的输出信号');

```

运行结果如图 6.18 所示。

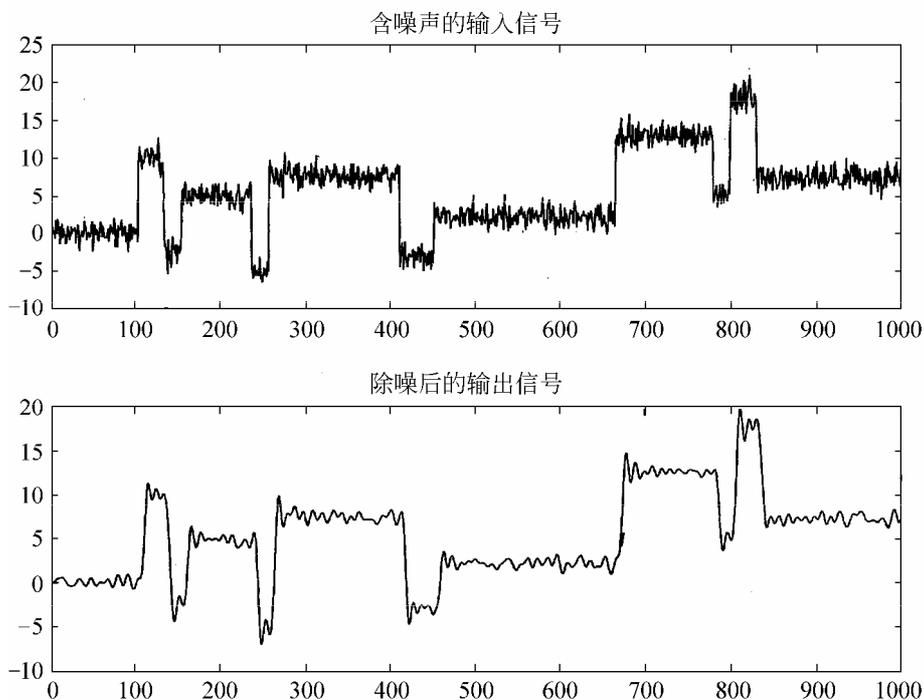


图 6.18 原信号及除噪后的结果

第 7 章 有限冲激响应滤波器的设计

无限长单位冲激响应 (IIR) 数字滤波器的优点是可以利用模拟滤波器设计的结果, 而模拟滤波器的设计有大量图表可查, 方便简单。但是它也有明显的缺点, 就是相位的非线性, 在图像处理以及数据传输等要求信道具有线性相位特性的场合, IIR 滤波器就不太适用了。有限单位冲激响应 (FIR) 数字滤波器则可以做成具有严格的线性相位, 同时又可以具有任意的幅度特性。此外, FIR 滤波器的单位抽样响应是有限长的, 因而 FIR 滤波器一定是稳定的。再有, 只要经过一定的延时, 任何非因果有限长序列都能变成因果的有限长序列, 因而总能用因果系统来实现。最后, FIR 滤波器由于单位冲激响应是有限长的, 因而可以用快速傅里叶变换 (FFT) 算法来过滤信号, 从而可大大提高运算效率。但是, 要取得很好的衰减特性, FIR 滤波器 $H(z)$ 的阶次比 IIR 滤波器的要高。

IIR 滤波器设计中的各种变换法对 FIR 滤波器设计是不适用的, 这是因为那里是利用有理分式的系统函数, 而 FIR 滤波器的系统函数只是 z^{-1} 的多项式。

因为最感兴趣的是具有线性相位的 FIR 滤波器, 对非线性相位的 FIR 滤波器, 一般可以用 IIR 滤波器来代替, 故本章主要讨论线性相位滤波器的设计。

7.1 线性相位 FIR 滤波器的特点

7.1.1 线性相位条件

如果一个线性移不变系统的频率响应有如下形式

$$H(e^{j\omega}) = H(\omega)e^{j\theta(\omega)} = |H(e^{j\omega})|e^{-j\alpha\omega} \quad (7.1)$$

则其具有线性相位。这里 α 是一个实数。因而, 线性相位系统有一个恒定的群延时

$$\tau = \alpha \quad (7.2)$$

在实际应用中, 有两类准确的线性相位, 分别要求满足

$$\theta(\omega) = -\tau\omega \quad (7.3)$$

$$\theta(\omega) = \beta - \tau\omega \quad (7.4)$$

FIR 滤波器具有式 (7.3) 的线性相位的充分必要条件是, 单位抽样响应 $h(n)$ 关于群延时 τ 偶对称, 即满足

$$\tau = \frac{N-1}{2} \quad (7.5)$$

$$h(n) = h(N-1-n), \quad 0 \leq n \leq N-1 \quad (7.6)$$

由于 $h(n)$ 的点数 N 又分为奇数与偶数两种情况, 因此把满足式 (7.5) 和式 (7.6) 的偶对称条件的 FIR 滤波器分别称为 I 型线性相位滤波器和 II 型线性相位滤波器, 如图 7.1 (a) 图 7.1 (b) 所示。

FIR 滤波器具有式 (7.4) 的线性相位的充分必要条件是, 单位抽样响应 $h(n)$ 关于群延时 τ 奇对称, 即满足

$$\tau = \frac{N-1}{2} \quad (7.7)$$

$$\beta = \pm \frac{\pi}{2} \quad (7.8)$$

$$h(n) = -h(N-1-n), \quad 0 \leq n \leq N-1 \quad (7.9)$$

同理, 把满足式 (7.7) 式 (7.8) 和式 (7.9) 的奇对称条件的 FIR 滤波器分别称为 I 型线性相位滤波器和 II 型线性相位滤波器, 如图 7.1 (c) 图 7.1 (d) 所示。

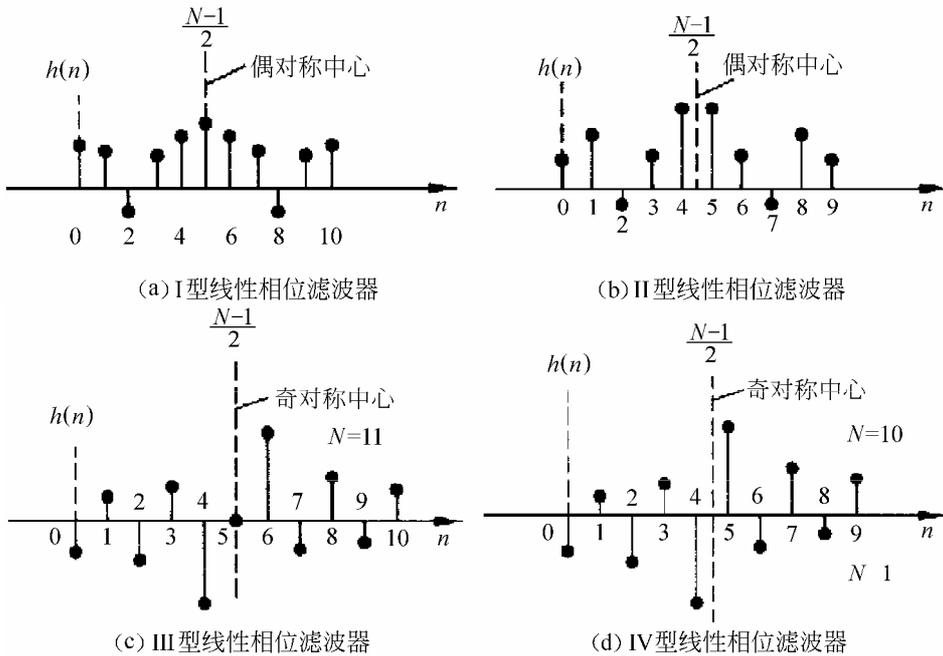


图 7.1 线性相位滤波器单位抽样响应的对称性

7.1.2 线性相位滤波器频率响应的特点

1. I 型线性相位滤波器

一个 I 型线性相位滤波器的, 由于偶对称性, 其频率响应可以表示为

$$H(e^{j\omega}) = e^{-j\frac{(N-1)\omega}{2}} \sum_{k=0}^{\frac{N-1}{2}} a(k) \cos(k\omega) \quad (7.10)$$

这里

$$a(0) = h\left(\frac{N-1}{2}\right)$$

$$a(k) = 2h\left(\frac{N-1}{2} - k\right), \quad k=1, 2, \dots, \frac{N-1}{2}$$

与式(7.1)对比,可得幅度函数为

$$H(\omega) = \sum_{k=0}^{\frac{N-1}{2}} a(k) \cos(k\omega) \quad (7.11)$$

相位函数为

$$\theta(\omega) = \frac{-(N-1)\omega}{2} \quad (7.12)$$

从式(7.10)式(7.11)可以看出, I 型线性相位滤波器的幅度函数 $H(\omega)$ 也对 $\tau = \frac{N-1}{2}$ 偶对称,同时对 $\omega=0, \pi, 2\pi$ 也呈偶对称。从式(7.12)可以看出,相位函数为准确的线性相位。

2. 型线性相位滤波器

一个 型线性相位滤波器,由于 N 是偶数,所以, $h(n)$ 的对称中心在半整数点 $\tau = \frac{N-1}{2}$ 。其频率响应可以表示为

$$H(e^{j\omega}) = e^{-j\frac{(N-1)\omega}{2}} \sum_{k=0}^{\frac{N}{2}} b(k) \cos\left[\left(k - \frac{1}{2}\right)\omega\right] \quad (7.13)$$

这里

$$b(k) = 2h\left(\frac{N}{2} - k\right) \quad k=1, 2, \dots, \frac{N}{2}$$

幅度函数为

$$H(\omega) = \sum_{k=0}^{\frac{N}{2}} b(k) \cos\left[\left(k - \frac{1}{2}\right)\omega\right] \quad (7.14)$$

相位函数为

$$\theta(\omega) = \frac{-(N-1)\omega}{2} \quad (7.15)$$

从式(7.13)和式(7.14)可以看出, 型线性相位滤波器的幅度函数 $H(\omega)$ 有下列特点:

- (1) 当 $\omega = \pi$ 时, $H(\pi) = 0$, 也就是说 $H(z)$ 在 $z = -1$ 处必然有一个零点。
- (2) $H(\omega)$ 对 $\omega = \pi$ 呈奇对称, 对 $\omega = 0, 2\pi$ 呈偶对称。

型线性相位滤波器的相位函数的特点与 I 型线性相位滤波器相同。

3. 型线性相位滤波器

由于 型线性相位滤波器关于 $\tau = \frac{N-1}{2}$ 奇对称,且 τ 为整数,所以, 型线性相位滤波器的频率响应可以表示为

$$H(e^{j\omega}) = j e^{-j\frac{(N-1)\omega}{2}} \sum_{k=1}^{\frac{N-1}{2}} c(k) \sin(k\omega) \quad (7.16)$$

这里

$$c(k) = 2h \left(\frac{N-1}{2} - k \right), \quad k=1, 2, \dots, \frac{N-1}{2}$$

幅度函数为

$$H(\omega) = \sum_{k=1}^{\frac{N-1}{2}} c(k) \sin(k\omega) \quad (7.17)$$

相位函数为

$$\theta(\omega) = \frac{-(N-1)\omega}{2} + \frac{\pi}{2} \quad (7.18)$$

从式 (7.17) 式 (7.18) 可以看出, 型线性相位滤波器的幅度函数 $H(\omega)$ 有下列特点:

- (1) $H(\omega)$ 在 $\omega=0, \pi, 2\pi$ 处必为零, 也就是说 $H(z)$ 在 $z = \pm 1$ 处都为零点。
- (2) $H(\omega)$ 对 $\omega=0, \pi, 2\pi$ 均呈奇对称。

同时, 由式 (7.18) 可见, 相位函数既是准确的线性相位, 又包括 $\pi/2$ 的相移, 所以又称为 90° 移相器, 或称为正交变换网络, 它和理想低通滤波器、理想微分器一样, 有着极重要的理论和实践意义。

4. 型线性相位滤波器

一个 型线性相位滤波器关于 $\tau = \frac{N-1}{2}$ 奇对称, 且 N 为偶数, 所以 τ 为非整数。型线性相位滤波器的频率响应可以表示为

$$H(e^{j\omega}) = j e^{-j \frac{(N-1)}{2} \omega} \sum_{k=1}^{\frac{N}{2}} d(k) \sin \left[\left(k - \frac{1}{2} \right) \omega \right] \quad (7.19)$$

这里

$$d(k) = 2h \left(\frac{N}{2} - k \right) \quad k=1, 2, \dots, \frac{N}{2}$$

幅度函数为

$$H(\omega) = \sum_{k=1}^{\frac{N}{2}} d(k) \sin \left[\left(k - \frac{1}{2} \right) \omega \right] \quad (7.20)$$

相位函数为

$$\theta(\omega) = \frac{-(N-1)\omega}{2} + \frac{\pi}{2} \quad (7.21)$$

从式 (7.20) 和式 (7.21) 可以看出, 型线性相位滤波器的幅度函数 $H(\omega)$ 有下列特点:

- (1) $H(\omega)$ 在 $\omega=0, 2\pi$ 处必为零, 也就是说 $H(z)$ 在 $z=1$ 处为零点。
- (2) $H(\omega)$ 在 $\omega=0, 2\pi$ 处呈奇对称, 在 $\omega=\pi$ 处呈偶对称。

型线性相位滤波器的相位函数的特点与 型线性相位滤波器相同。

7.1.3 零点位置

一个线性相位系统, 其单位抽样响应的对称性强加于系统函数一些约束。对于 I 型或

型线性相位滤波器， $h(n) = h(N-1-n)$ ，这意味着

$$H(z) = z^{-(N-1)} H(z^{-1})$$

类似地，对于 Ⅱ 型线性相位滤波器， $h(n) = -h(N-1-n)$ ，这意味着

$$H(z) = -z^{-(N-1)} H(z^{-1})$$

在这两种情况下，如果 $H(z)$ 在 $z = z_0$ 处等于零，则在 $z = 1/z_0$ 处也一定等于零，所以 $H(z)$ 的零点呈倒数对出现。另外，若 $h(n)$ 是实值的，则复零点呈共轭倒数对出现，或者说共轭镜像的。因此，一个线性相位滤波器零点的约束条件如下：

(1) 零点 $z = z_i$ 既不在实轴上，也不在单位圆上，即 $z_i = r_i e^{j\theta_i}$ ， $r_i \neq 1$ ， $\theta_i \neq 0$ ，零点是一组互为倒数的共轭对，如图 7.2 (a) 所示，因而它们的基本因子为

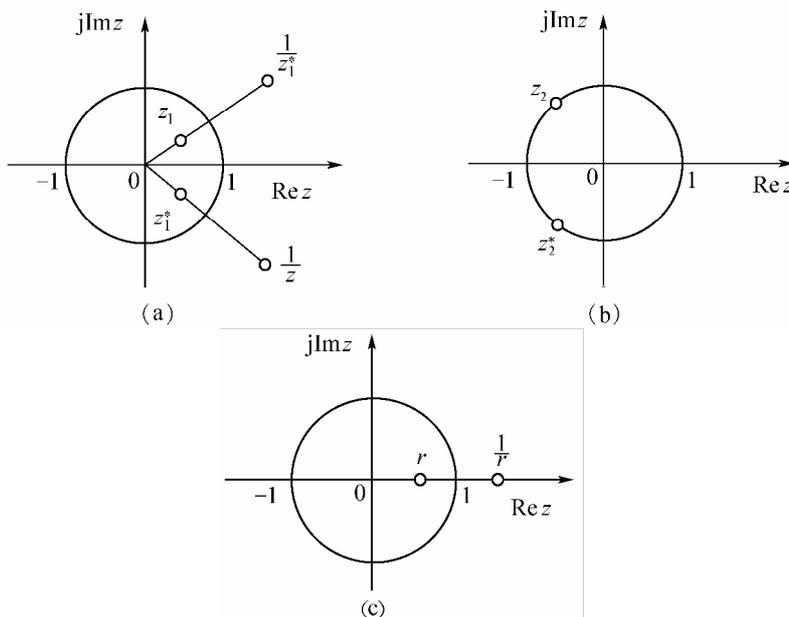


图 7.2 线性相位 FIR 滤波器的零点位置图

$$\begin{aligned}
 H_i(z) &= (1 - z^{-1} r_i e^{j\theta_i})(1 - z^{-1} r_i e^{-j\theta_i}) \left(1 - z^{-1} \frac{1}{r_i} e^{j\theta_i}\right) \left(1 - z^{-1} \frac{1}{r_i} e^{-j\theta_i}\right) \\
 &= \frac{1}{r_i^2} [1 - 2r_i(\cos \theta_i) z^{-1} + r_i^2 z^{-2}] [1 - 2r_i(\cos \theta_i) z^{-1} + z^{-2}]
 \end{aligned}
 \tag{7.22}$$

式 (7.22) 可用线性相位滤波器的级联结构实现，如图 7.3 所示。在此情况下， $N = 5$ ， $\tau = (N-1)/2 = 2$ 。

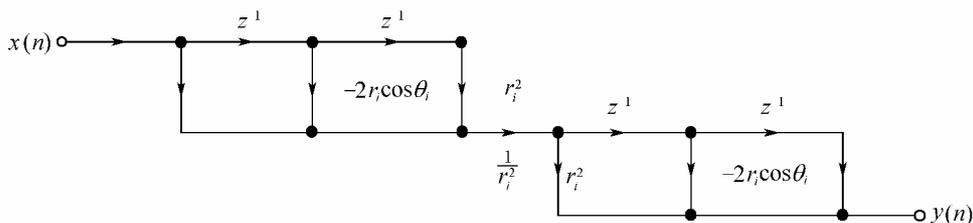


图 7.3 线性相位 FIR 滤波器的级联结构

(2) 零点 $z = z_i$ 在单位圆上, 但不在实轴上, 即 $r_i = 1$, $\theta_i \neq 0$, $\theta_i \neq \pi$, 此时零点的共轭值就是它的倒数, 如图 7.2 (b) 所示, 因而它们的基本因子为

$$H_i(z) = (1 - z^{-1}e^{j\theta_i})(1 - z^{-1}e^{-j\theta_i}) = 1 - 2(\cos\theta_i)z^{-1} + z^{-2} \quad (7.23)$$

在此情况下, $N = 3$, $\tau = (N - 1)/2 = 1$ 。

(3) 零点 $z = z_i$ 在实轴上, 但不在单位圆上, 即 $r_i \neq 1$, $\theta_i = 0$ 或 π , 此时零点是实数, 它没有复共轭部分, 只有倒数, 倒数也在实轴上, 如图 7.2 (c) 所示, 因而它们的基本因子为

$$H_i(z) = (1 \pm r_i z^{-1}) \left(1 \pm \frac{1}{r_i} z^{-1} \right) = 1 \pm \left(r_i + \frac{1}{r_i} \right) z^{-1} + z^{-2} \quad (7.24)$$

式中“+”号相当于 $\theta_i = \pi$, 零点在负实轴上, “-”相当于 $\theta_i = 0$, 零点在正实轴上。此时, $N = 3$, $\tau = (N - 1)/2 = 1$ 。

(4) 零点 $z = z_i$ 既在实轴上, 又在单位圆上, 即 $r_i = 1$, $\theta_i = 0$ 或 π , 此时零点只有两种情况, 即 $z = 1$ 或 $z = -1$, 这时零点既是自己的复共轭, 又是倒数, 其基本因子为

$$H_i(z) = 1 \pm z^{-1} \quad (7.25)$$

式中“+”表示零点在 $z = -1$ 处, “-”表示表示零点在 $z = 1$ 处。此时, $N = 2$, $\tau = \frac{N-1}{2} = 1/2$, 即有半个抽样的延时。

线性相位滤波器 $H(z)$ 只能由以上这几种因子的组合构成。

7.2 窗函数设计法

7.2.1 设计方法

一般是先给所要求的理想低通滤波器频率响应 $H_d(e^{j\omega})$, 要求设计一个 FIR 滤波器频率响应 $H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}$ 来逼近 $H_d(e^{j\omega})$ 。由于设计是在时域进行的, 因而先由 $H_d(e^{j\omega})$ 的傅里叶反变换导出 $h_d(n)$, 即

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \frac{\sin(\omega_c n)}{\pi n} \quad (7.26)$$

由于 $H_d(e^{j\omega})$ 是矩形频率特性, 故 $h_d(n)$ 一定是无限长的序列, 而且是非因果的, 而要设计的是 FIR 滤波器, $h(n)$ 必然是有限长的, 所以要用有限长的 $h(n)$ 来逼近无限长的 $h_d(n)$, 最有效的方法是截断 $h_d(n)$, 即用一个有限长度的窗函数序列 $\omega(n)$ 来截取 $h_d(n)$, 并将截短后的 $h_d(n)$ 移位, 得

$$h(n) = \omega \left(n - \frac{N-1}{2} \right) h_d \left(n - \frac{N-1}{2} \right) \quad (7.27)$$

由于 $\omega(n)$ 的长度为 N , 所以 $h(n)$ 是因果的。因此窗函数序列 $\omega(n)$ 的形状及长度的选择就很关键。

现举例说明上述方法的应用及存在的问题。

例 7.1 设计一低通滤波器, 所希望的频率响应截止频率 $H_d(e^{j\omega})$ 在 $0 \leq \omega \leq 0.25\pi$ 之间为 1, 在 $0.25\pi \leq \omega \leq \pi$ 之间为 0, 分别取 $N=11, 21, 41$, 观察其频谱响应的特点。

解:

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{(N-1)}{2}\omega}, & 0 \leq \omega \leq 0.25\pi \\ 0, & 0.25\pi \leq \omega \leq \pi \end{cases}$$

取矩形窗

$$\omega(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases},$$

则根据式 (7.27), 可知

$$h(n) = h_d\left(n - \frac{N-1}{2}\right) = \frac{\sin\left[0.25\pi \times \left(n - \frac{N-1}{2}\right)\right]}{\pi\left(n - \frac{N-1}{2}\right)}$$

当 $N=11$ 时, 求得

$$h(0) = h(10) = -0.045, h(1) = h(9) = 0, h(2) = h(8) = 0.075$$

$$h(3) = h(7) = 0.1592, h(4) = h(6) = 0.2251, h(5) = 0.25$$

显然 $\tau = (N-1)/2 = 5$, 满足对称关系, 如图 7.4 (a) 所示。

根据序列 $h(n)$, 分别求得 $N=11, 21, 41$ 时的幅频特性 $|H(e^{j\omega})|$, 如图 7.4 (b) 所示。

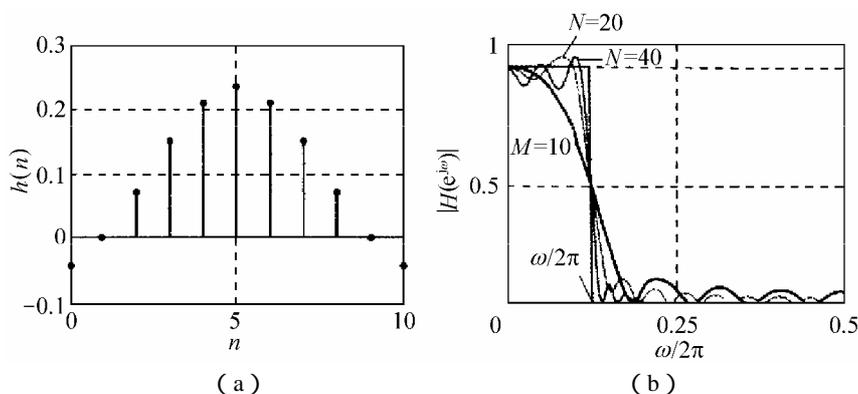


图 7.4 例 7.1 的设计结果

由图 7.4 (b) 可以看出, 当 N 取值太小时, 通频带过窄, 且阻带内波纹较大, 过渡带较宽, 当 N 增大时, $H(e^{j\omega})$ 与 $H_d(e^{j\omega})$ 的近似程度越来越好。但当 N 增大时, 通带内出现了波纹, 而且随着 N 的继续增大, 这些波纹并不消失, 只是最大的尖峰越来越接近于间断点, 这种现象称为吉布斯现象。

吉布斯现象的产生是对 $h_d(n)$ 突然截短的结果。在时域加窗, 等于频域 $H_d(e^{j\omega})$ 与矩形窗频谱的卷积, 由于矩形窗频谱 $D(e^{j\omega})$ 为 $\text{sinc}(\omega)$ 函数, 它有着较大的旁瓣, 正是这些旁瓣在和 $H_d(e^{j\omega})$ 卷积时产生了吉布斯现象。

为了减弱吉布斯现象, 应选取旁瓣较小的窗函数。

7.2.2 各种窗函数

常用的窗函数有以下几种。

1. 矩形窗

窗函数为

$$\omega(n) = R_N(n) \quad (7.28)$$

幅度函数为

$$W_R(\omega) = |W_R(e^{j\omega})| = \frac{\sin\left(\frac{N\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \quad (7.29)$$

主瓣宽度为 $4\pi/N = 2 \times 2\pi/N$ ，过渡带宽 $\Delta\omega = 0.9 \times 2\pi/N$ 。

2. 汉宁 (Hanning) 窗 (又称为升余弦窗)

窗函数为

$$\omega(n) = \left[0.5 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) \right] R_N(n) \quad (7.30)$$

幅度函数为

$$W(\omega) = 0.5W_R(\omega) + 0.25 \left[W_R\left(\omega - \frac{2\pi n}{N-1}\right) + W_R\left(\omega + \frac{2\pi n}{N-1}\right) \right] \quad (7.31)$$

主瓣宽度为 $4 \times 2\pi/N = 8\pi/N$ ，过渡带宽 $\Delta\omega = 3.1 \times 2\pi/N$ 。

3. 汉明 (Hamming) 窗 (又称为改进的升余弦窗)

窗函数

$$\omega(n) = \left[0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \right] R_N(n) \quad (7.32)$$

幅度函数为

$$W(\omega) = 0.54W_R(\omega) + 0.23 \left[W_R\left(\omega - \frac{2\pi n}{N-1}\right) + W_R\left(\omega + \frac{2\pi n}{N-1}\right) \right] \quad (7.33)$$

主瓣宽度为 $4 \times 2\pi/N = 8\pi/N$ ，过渡带宽 $\Delta\omega = 3.3 \times 2\pi/N$ 。

4. 凯泽 (Kaiser) 窗

窗函数为

$$\omega(n) = \frac{I_0\left(\beta \sqrt{1 - \left(1 - \frac{2n}{N-1}\right)^2}\right)}{I_0(\beta)}, \quad 0 \leq n \leq N-1 \quad (7.34)$$

其中 $I_0(\cdot)$ 为第一类变形零阶贝塞尔函数， β 是一个可自由选择的参数，它可以同时调

整主瓣宽度, β 越大, 则 $\omega(n)$ 窗越窄, 而频谱的旁瓣越小, 但主瓣宽度相应增加。因而改变 β 值就可对主瓣宽度与旁瓣衰减进行选择, 一般选择 $4 < \beta < 9$ 。过渡带宽 $\Delta\omega = 5 \times 2\pi/N$ 。

以上 4 种窗函数的主要性能列于表 7.1 中, 供设计 FIR 滤波器时参考。

表 7.1 几种窗函数的基本参数比较

窗函数	窗谱性能指标		加窗后滤波器性能指标	
	旁瓣峰值 (dB)	主瓣宽度 ($\times 2\pi/N$)	过渡带宽 ($\times 2\pi/N$)	阻带最小衰减 (dB)
矩形窗	-13	2	0.9	-21
汉宁窗	-31	4	3.1	-44
汉明窗	-41	4	3.3	-53
凯泽窗	-57		5	-80

最小阻带衰减只由窗形决定, 不受 N 的影响, 而过渡带宽则随 N 的增加而减小。

7.2.3 其他各型 FIR 滤波器的设计方法

前面介绍了低通 FIR 数字滤波器的设计方法, 若希望设计高通、带通、带阻数字滤波器, 只需改变式 (7.26) 积分的上、下限, 现分别予以介绍。

1. 高通数字滤波器的设计

令

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{(N-1)}{2}\omega}, & \omega_c \leq \omega \leq \pi \\ 0, & 0 \leq \omega \leq \omega_c \end{cases} \quad (7.35)$$

则

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\omega_c} e^{j\omega\left(n-\frac{N-1}{2}\right)} d\omega + \frac{1}{2\pi} \int_{\omega_c}^{\pi} e^{j\omega\left(n-\frac{N-1}{2}\right)} d\omega$$

求得

$$h_d(n) = \frac{\sin\left[\pi \times \left(n - \frac{N-1}{2}\right)\right] - \sin\left[\left(n - \frac{N-1}{2}\right) \times \omega_c\right]}{\pi\left(n - \frac{N-1}{2}\right)} \quad (7.36)$$

2. 带通数字滤波器的设计

令

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{(N-1)}{2}\omega}, & \omega_l \leq \omega \leq \omega_h \\ 0, & \text{其他} \end{cases} \quad (7.37)$$

则

$$h_d(n) = \frac{1}{2\pi} \int_{-\omega_h}^{-\omega_l} e^{j\omega\left(n-\frac{N-1}{2}\right)} d\omega + \frac{1}{2\pi} \int_{\omega_l}^{\omega_h} e^{j\omega\left(n-\frac{N-1}{2}\right)} d\omega$$

求得

$$h_d(n) = \frac{\sin\left[\left(n - \frac{N-1}{2}\right)\omega_h\right] - \sin\left[\left(n - \frac{N-1}{2}\right)\omega_l\right]}{\pi\left(n - \frac{N-1}{2}\right)} \quad (7.38)$$

3. 带阻数字滤波器的设计

令

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{(N-1)\omega}{2}}, & |\omega| \leq \omega_l, |\omega| \geq \omega_h \\ 0, & \text{其他} \end{cases} \quad (7.39)$$

则

$$h_d(n) = \frac{1}{2\pi} \int_{-\omega_h}^{-\pi} e^{j\omega\left(n - \frac{N-1}{2}\right)} d\omega + \frac{1}{2\pi} \int_{-\omega_l}^{\omega_l} e^{j\omega\left(n - \frac{N-1}{2}\right)} d\omega + \frac{1}{2\pi} \int_{\omega_h}^{\pi} e^{j\omega\left(n - \frac{N-1}{2}\right)} d\omega$$

求得

$$h_d(n) = \frac{\sin\left[\left(n - \frac{N-1}{2}\right)\omega_l\right] + \sin\left[\left(n - \frac{N-1}{2}\right)\pi\right] - \sin\left[\left(n - \frac{N-1}{2}\right)\omega_h\right]}{\pi\left(n - \frac{N-1}{2}\right)} \quad (7.40)$$

比较式 (7.36) 式 (7.38) 式 (7.40) 可知, 一个高通滤波器相当于用一个全通滤波器减去一个低通滤波器; 一个带通滤波器相当于两个低通滤波器相减, 其中一个截止频率为 ω_h , 另一个为 ω_l ; 一个带阻滤波器相当于一个低通滤波器加上一个高通滤波器, 低通滤波器的截止频率为 ω_l , 高通滤波器为 ω_h 。

选取一个满意的窗函数, 令

$$h(n) = \omega(n)h_d(n), \quad n=0, 1, \dots, N-1 \quad (7.41)$$

则 $h(n)$ 即为要设计的滤波器的单位抽样响应。

按上述方法设计的滤波器, 由于满足了 $h(n) = \pm h(N-1-n)$ 的对称关系, 因此都具有线性相位。

7.3 频率抽样设计法

窗函数法是从时域出发, 把理想的 $h_d(n)$ 用一定形状的窗函数截取成有限长的 $h(n)$, 以此 $h(n)$ 来近似理想的 $h_d(n)$, 这样得到的频率响应 $H(e^{j\omega})$ 逼近于所要求的理想的频率响应 $H_d(e^{j\omega})$ 。

频率抽样法则是从频域出发, 把给定的理想频率响应 $H_d(e^{j\omega})$ 加以等间隔抽样, 即

$$H_d(e^{j\omega}) \Big|_{\omega = \frac{2\pi m}{k}} = H_d(k)$$

然后以此 $H_d(k)$ 作为实际 FIR 数字滤波器的频率特性的抽样值 $H(k)$, 即令

$$H(k) = H_d(k) = H_d(e^{j\omega}) \Big|_{\omega = \frac{2\pi k}{N}}, \quad k=0, 1, \dots, N-1 \quad (7.42)$$

知道 $H(k)$ 后, 由 DFT 定义, 可以用频域的这些 N 个抽样值 $H(k)$ 来惟一确定有限长序列 $h(n)$, 即

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} H_d(k) e^{j\frac{2\pi}{N}nk}, \quad k=0, 1, \dots, N-1 \quad (7.43)$$

利用式 (7.43) 可以求得滤波器的系统函数, 即

$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h(n)z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} H_d(k) e^{j\frac{2\pi}{N}nk} \right] z^{-n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} H_d(k) \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}nk} z^{-n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} H_d(k) \frac{1 - z^{-N}}{1 - e^{j\frac{2\pi}{N}nk} z^{-1}} \end{aligned} \quad (7.44)$$

该系统的频率响应为

$$H(e^{j\omega}) = H(z) \Big|_{z=e^{j\omega}} = \frac{1}{N} \sum_{k=0}^{N-1} H_d(k) \frac{1 - e^{-j\omega N}}{1 - e^{j\frac{2\pi}{N}nk} e^{-j\omega}} \quad (7.45)$$

经过推导, 有

$$H(e^{j\omega}) = e^{-j\frac{(N-1)}{2}\omega} \sum_{k=0}^{N-1} H_d(k) e^{j\frac{(N-1)}{2}k\pi} \frac{\sin \left[\frac{N \left(\omega - \frac{2\pi k}{N} \right)}{2} \right]}{N \sin \left[\frac{\left(\omega - \frac{2\pi k}{N} \right)}{2} \right]} \quad (7.46)$$

由式 (7.46) 可知, $H(e^{j\omega})$ 是由内插函数

$$\Phi(\omega) = e^{-j\frac{(N-1)}{2}\omega} \frac{\sin \left(\frac{\omega N}{2} \right)}{N \sin \left(\frac{\omega}{2} \right)}$$

的插值所决定的, 即

$$H(e^{j\omega}) = \sum_{k=0}^{N-1} H_d(k) \Phi \left(\omega - \frac{2\pi}{N}k \right) \quad (7.47)$$

由内插公式 (7.47) 可知, 在各频率抽样点上, 滤波器的实际频率响应严格地和理想频率响应值相等。但在抽样点响应数值相等, 即 $H(e^{j\frac{2\pi}{N}k}) = H(k) = H_d(k) = H_d(e^{j\frac{2\pi}{N}k})$ 。但是

在抽样点之间的频率响应则是由 N 个离散值 $H_d(k)$ 作为权重和插值函数 $\Phi(\omega)$ 线性组合的结果。显然 $H_d(e^{j\omega})$ 抽样点 N 取值越大, $H(e^{j\omega})$ 对 $H_d(e^{j\omega})$ 的近似程度越好, N 的选取要视 $H(e^{j\omega})$ 在通带和阻带内的技术要求而定。

为由式 (7.43) 求得 $h(n)$, 首先要指定 $H_d(k)$, $H_d(k)$ 的指定原则如下:

- 在通带内可令 $|H_d(k)|=1$, 阻带内 $|H_d(k)|=0$, 且在通带内赋给 $H_d(k)$ 一相位函数。
- 指定的 $H_d(k)$ 应保证由式 (7.43) 求出的 $h(n)$ 是实序列。
- 由 $h(n)$ 求出的 $H(e^{j\omega})$ 应具有线性相位。

现根据以上三个原则来讨论 $H_d(k)$ 的确定。

由式 (7.46) 知, 若保证

$$H_d(k)e^{j\frac{(N-1)k\pi}{N}} = \text{实数}$$

那么, $H(e^{j\omega})$ 就具有线性相位。

$$\theta(\omega) = -\frac{N-1}{2}\omega$$

满足上式并考虑 $|H_d(k)|=1$, 等效地指定

$$H_d(k) = e^{-j\frac{(N-1)k\pi}{N}}, \quad k=0, 1, \dots, N-1 \quad (7.48)$$

根据 DFT 的性质可知, 为保证 $h(n)$ 是实序列, $H_d(k)$ 应满足下列对称关系

$$H_d^*(k) = H_d(-k) = H_d(N-k) \quad (7.49)$$

结合式 (7.48) 式 (7.49), 由于

$$H_d(N-k) = e^{-j\frac{(N-1)(N-k)\pi}{N}} = e^{-j(N-1)\pi} e^{j\frac{(N-1)k\pi}{N}} = e^{-j(N-1)\pi} H_d^*(k) \quad (7.50)$$

当 N 为偶数时, $e^{-j(N-1)\pi} = -1$; 当 N 为奇数时, $e^{-j(N-1)\pi} = 1$ 。这样当 N 为偶数时, 若按式 (7.48) 对 $H_d(k)$ 赋值, 就不能满足式 (7.49) 的对应关系。为此, 按如下原则对 $H_d(k)$ 赋值。

N 为偶数时

$$H_d(k) = \begin{cases} e^{-j\frac{(N-1)k\pi}{N}}, & k=0, 1, \dots, \frac{N}{2}-1 \\ 0, & k=\frac{N}{2} \\ -e^{-j\frac{(N-1)k\pi}{N}}, & k=\frac{N}{2}+1, \dots, N-1 \end{cases} \quad (7.51)$$

N 为奇数时

$$H_d(k) = e^{-j\frac{(N-1)k\pi}{N}}, \quad k=0, 1, \dots, N-1 \quad (7.52)$$

根据上述原则, 现在可以归纳出用频率抽样法设计 FIR 数字滤波器的步骤如下:

(1) 根据所设计的滤波器的通带与阻带的要求, 根据 N 为偶数还是奇数, 按式 (7.51) 式 (7.52) 指定 $H_d(k)$, 在阻带内, $H_d(k)=0$ 。

(2) 由指定的 $H_d(k)$ 构成所设计的滤波器的转移函数, 也可由式 (7.44) 求得频率响应 $H(e^{j\omega})$ 。

例 7.2 用频率抽样法设计一个低通滤波器, 其截止频率是抽样频率的 $1/10$, 取 $N=20$ 。

解: 此处 N 为偶数, 且在通带内对 $H_d(e^{j\omega})$ 抽样时, 仅得两个点, 由式 (7.51) 有

$$H_d(0) = 1$$

$$H_d(1) = e^{-j\frac{19\pi}{20}}$$

$$H_d(19) = H_d(20-1) = H_d^*(1) = e^{j\frac{19\pi}{20}}$$

在其他 k 点处, $H_d(k) = 0$ 。

将 $H_d(k)$ 代入式 (7.43), 得 $h(n)$ 序列如下:

$$h(0) = h(19) = -0.04877, \quad h(1) = h(18) = -0.0391$$

$$h(2) = h(17) = -0.0207, \quad h(3) = h(16) = 0.0046$$

$$h(4) = h(15) = 0.03436, \quad h(5) = h(14) = 0.0656$$

$$h(6) = h(13) = 0.0954, \quad h(7) = h(12) = 0.12071$$

$$h(8) = h(11) = 0.1391, \quad h(9) = h(10) = 0.14877$$

7.4 应用 MATLAB 设计 FIR 数字滤波器

7.4.1 与本章内容有关的 MATLAB 文件

与 FIR 数字滤波器设计有关的 MATLAB 文件主要分为两类。一类是用于产生各种窗函数, 另一类是用于设计 FIR 数字滤波器。

1. 窗函数

用于产生窗函数的 MATLAB 文件有 8 个。

- (1) bartlett.m (三角窗)
- (2) blackman.m (布莱克曼窗)
- (3) boxcar.m (矩形窗)
- (4) hamming.m (汉明窗)
- (5) hanning.m (汉宁窗)
- (6) triang.m (三角窗)
- (7) chebwin.m (切比雪夫窗)

(8) kaiser.m (凯泽窗)

2. FIR 数字滤波器的文件

以下 MATLAB 文件是用于 FIR 数字滤波器的文件。

(1) fir1.m

本文件采用窗函数法设计 FIR 数字滤波器，其调用格式是

- $b = \text{fir1}(N, W_n)$
- $b = \text{fir1}(N, W_n, \text{'high'})$
- $b = \text{fir1}(N, W_n, \text{'stop'})$

式中 N 为滤波器的阶次，因此滤波器的长度为 $N+1$ ； W_n 是通带截止频率，其值在 $0 \sim 1$ 之间， 1 对应抽样频率的一半； b 是设计好的滤波器系数 $h(n)$ 。

对于第一种格式，若 W_n 是一标量，则可用来设计低通滤波器；若 W_n 是 1×2 的向量，则用来设计带通滤波器；若 W_n 是 $1 \times L$ 的向量，则可用来设计 L 带滤波器，此时，格式将变为

$b = \text{fir1}(N, W_n, \text{'DC-1'})$ 或 $b = \text{fir1}(N, W_n, \text{'DC-0'})$

其中，前者保证第一个带为通带，后者保证第一个带为阻带。

第二种格式用来设计高通滤波器，第三种格式用来设计带阻滤波器。

值得注意的是，在上述所有格式中，若不指定窗函数的类型，则 `fir1` 自动选择汉明窗。

(2) fir2.m

本文件采用窗函数法设计具有任意幅频特性的 FIR 数字滤波器。其调用格式是

$b = \text{fir1}(N, F, M)$

其中 F 是频率向量，其值在 $0 \sim 1$ 之间， M 是与 F 相对应的所希望的幅频响应。不指定窗函数的类型时，则自动选择汉明窗。

(3) remez.m

本文件用来设计采用切比雪夫最佳一致逼近 FIR 数字滤波器。同时，还可以用来设计希尔伯特变换器和差分器。其调用格式是

- $b = \text{remez}(N, F, A)$
- $b = \text{remez}(N, F, A, W)$
- $b = \text{remez}(N, F, A, W, \text{'hilbert'})$
- $b = \text{remez}(N, F, A, W, \text{'differentiator'})$

其中， N 是给定的滤波器的阶次； b 是设计的滤波器的系数，其长度为 $N+1$ ； F 是频率向量，其值在 $0 \sim 1$ 之间； A 是对应 F 的各频段上的理想幅频响应； W 是各频段上的加权向量。

值得注意的是，若 b 的长度为偶数，设计高通和带阻滤波器时有可能出现错误，因此最好保证 b 的长度为奇数，即 N 应为偶数。

(4) remexord.m

本文件采用切比雪夫一致逼近设计 FIR 数字滤波器时所需要的滤波器阶次。其调用格式是

$[N, F_o, A_o, W] = \text{remexord}(F, A, \text{DEV}, F_s)$

式中, F 、 A 的含义同文件 `remez.m`, 是通带和阻带上的偏差; 该文件输出的是符合要求的滤波器阶次 N 、频率向量 F_0 、幅度向量 A_0 和加权向量 W 。若设计者事先不能确定自己要设计的滤波器的阶次, 那么调用 `remexord` 后, 就可利用这一族参数再调用 `remez`, 即 $b = \text{remez}(N, F_0, A_0, W)$, 从而设计出所需要的滤波器。因此, 通常 `remez` 和 `remexord` 结合使用。

值得说明是, `remexord` 给出的阶次 N 有可能偏低, 这时适当增加 N 即可; 另外, 若 N 为奇数, 就可令其加 1, 使其变为偶数, 这样 b 的长度为奇数。

(5) `sgolay.m`

本文件用来设计 Savitzky-Golay 平滑滤波器。其调用格式是

$b = \text{sgolay}(k, f)$

式中, k 是多项式的阶次, f 是拟合的双边点数。要求 $k < f$, 且 f 为奇数。

(6) `firls.m`

本文件用最小二乘法设计线性相位 FIR 数字滤波器。可设计任意给定的理想幅频特性。

(7) `fircls.m`

本文件用带约束的最小二乘法设计线性相位 FIR 数字滤波器。可设计任意给定的理想幅频特性。

(8) `fircls1.m`

本文件用带约束的最小二乘法设计线性相位 FIR 低通和高通滤波器。可设计任意给定的理想幅频特性。

(9) `firrcos.m`

本文件用来设计低通线性相位 FIR 数字滤波器, 其过渡带为余弦函数形状。

7.4.2 应用 MATLAB 设计 FIR 数字滤波器

例 7.3 令 $N=10$, 分别用矩形窗和汉明窗重复例 7.1。

解: 根据要求编制 MATLAB 程序如下。

```
clear all;
N=10;
b1=fir1(N,0.25,boxcar(N+1)); % 用矩形窗作为冲激响应的窗函数
b2=fir1(N,0.25,hamming(N+1)); % 用汉明窗作为冲激响应的窗函数
%
M=128;
h1=freqz(b1,1,M);
h2=freqz(b2,1,M);
% 分别求两个滤波器的频率响应
t=0:10;
subplot(221)
stem(t,b2,'.');hold on;
```

```

plot(t,zeros(1,11));grid;
f=0:0.5/M:0.5-0.5/M;
M1=M/4;
for k=1:M1
    hd(k)=1;
    hd(k+M1)=0;
    hd(k+2*M1)=0;
    hd(k+3*M1)=0;
end
subplot(222)
plot(f,abs(h1),'b-',f,abs(h2),'g-',f,hd,'-');grid;

```

运行结果如图 7.5 所示。

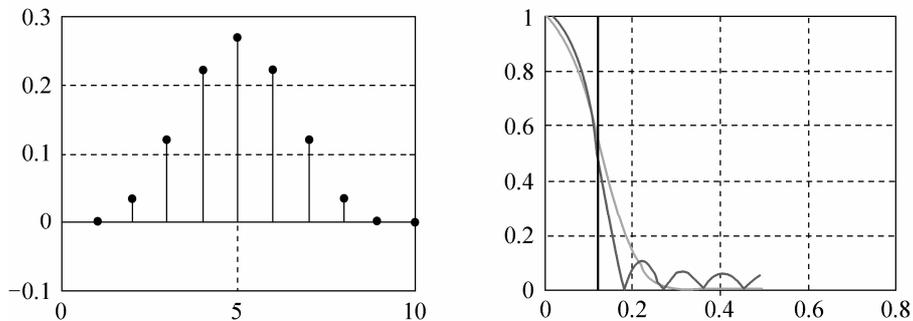


图 7.5 运行结果

例 7.4 设计一多带滤波器，要求理想幅频响应在归一化频率 0.2~0.3，0.6~0.8 之间为 1，其余均为 0。

解：程序如下：

```

clear all;
f=[0 0.19 0.2 0.3 0.31 0.59 0.6 0.8 0.81 1];
% 给定频率轴分点
m=[0 0 1 1 0 0 1 1 0 0];
% 给定在这些频率分点上理想的幅频响应
N1=30;
N2=90;
% 取两种不同的滤波器长度
b1=fir2(N1,f,m);
b2=fir2(N2,f,m);
% 得到两个滤波器

subplot(311);
stem(b1,'.');grid;
subplot(312);

```

```

stem(b2,'. ');grid;

M=128;
[h1,w]=freqz(b1,1,M,1);
[h2,w]=freqz(b2,1,M,1);
subplot(313);
plot(w,abs(h1),'b-',w,abs(h2),'g-');grid;

```

运行结果如图 7.6 所示。

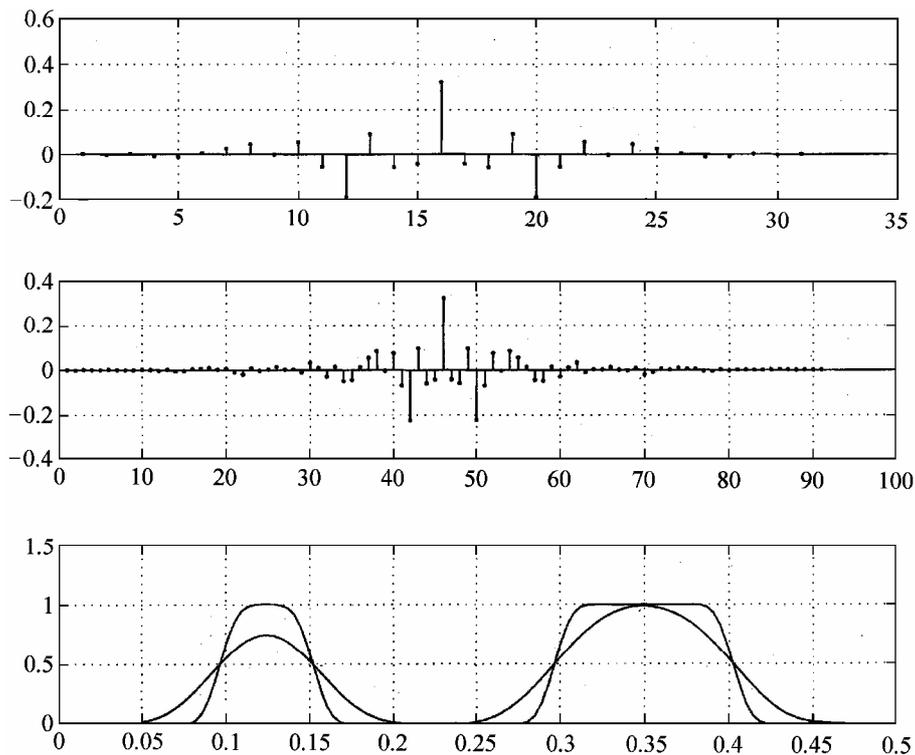


图 7.6 滤波器单位抽样响应及其幅频响应曲线

例 7.5 利用切比雪夫最佳一致逼近法设计一低通滤波器,要求通带边缘频率 $\omega_p=0.6\pi$, 阻带边缘频率 $\omega_s=0.7\pi$ 。

解:程序如下。

```

clear all;
f=[0 .6 .7 1];
% 给定频率轴分点
A=[1 1 0 0];
% 给定在这些频率分点上理想的幅频响应
weigh=[1 10];
% 给定在这些频率分点上的加权

```

```

b=remez(32,f,A,weigh);
% 设计出切比雪夫最佳一致逼近滤波器
[h,w]=freqz(b,1,256,1);
h=abs(h);
h=20*log10(h);
figure(1)
stem(b,'. ');grid;
figure(2)
plot(w,h);grid;

```

运行结果如图 7.7 所示。

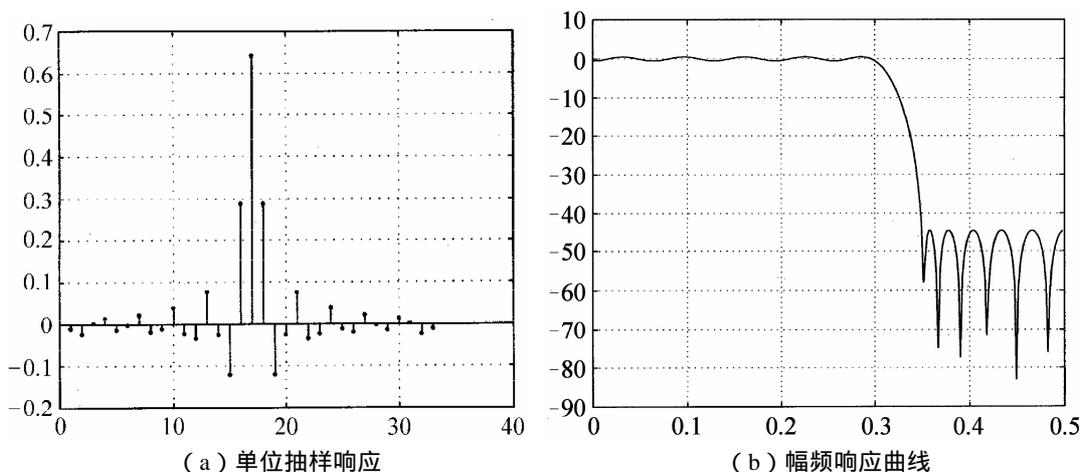


图 7.7 低通滤波器单位抽样响应及幅频响应曲线

例 7.6 利用切比雪夫最佳一致逼近法设计一个多阻带陷波器，数字系统的抽样频率为 500Hz，去掉工频信号（50Hz）及二次、三次谐波的干扰。

解：程序如下。

```

clear all;
% 用切比雪夫最佳一致逼近设计线性相位多带 FIR 滤波器
f=[0 .14 .18 .22 .26 .34 .38 .42 .46 .54 .58 .62 .66 1];
A=[1 1 0 0 1 1 0 0 1 1 0 0 1 1];
weigh=[8 1 8 1 8 1 8];
b=remez(64,f,A,weigh);
%
[h,w]=freqz(b,1,256,1);
hr=abs(h);
h=abs(h);
h=20*log10(h);
figure(1)
stem(b,'. ');grid;
figure(2)

```

```
plot(w,h);grid;
```

运行结果如图 7.8 所示。

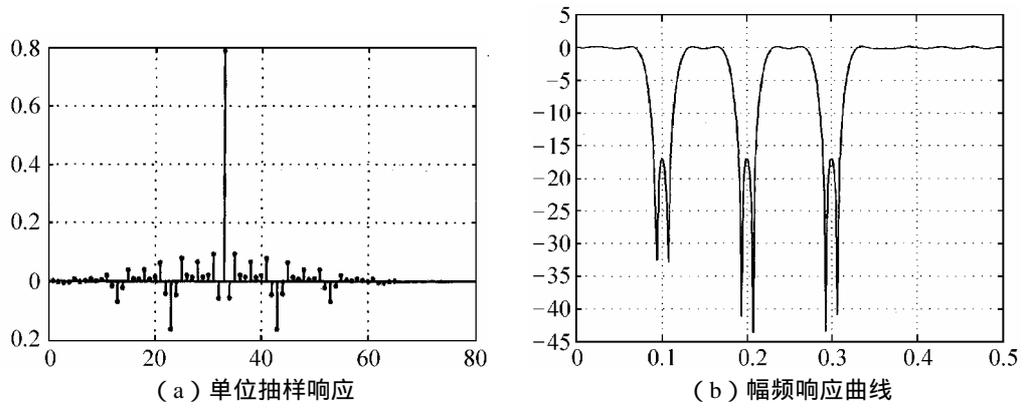


图 7.8 陷波器单位抽样响应和幅频响应曲线

参 考 书 目

- 1 余成波, 张兢. 信号分析基础. 重庆: 重庆大学出版社, 2000
- 2 曾禹村, 张宝俊. 信号与系统. 北京: 北京理工大学出版社, 1986
- 3 燕庆明. 信号与系统教程. 北京: 高等教育出版社, 2004
- 4 郑君里, 应启珩, 杨为理. 信号与系统(第二版). 北京: 高等教育出版社, 2000
- 5 赵红怡, 张常年. 数字信号处理及其 MATLAB 实现. 北京: 化学工业出版社, 2002
- 6 李行一. 数字信号处理. 重庆: 重庆大学出版社, 2002
- 7 胡广书. 数字信号处理理论、算法与实现(第二版). 北京: 清华大学出版社, 2003
- 8 张小虹. 数字信号处理习题详解. 南京: 东南大学出版社, 2002
- 9 薛年喜. MATLAB 在数字信号处理中的应用. 北京: 清华大学出版社, 2003

参考文献

- [1] 余成波, 张兢编著. 信号分析基础. 重庆: 重庆大学出版社, 2000
- [2] 曾禹村, 张宝俊. 信号与系统. 北京: 北京理工大学出版社, 1986
- [3] 燕庆明. 信号与系统教程. 北京: 高等教育出版社, 2004
- [4] 郑君里, 应启珩, 杨为理. 信号与系统 (第二版). 北京: 高等教育出版社, 2000
- [5] Simon Haykin, Barry Van Veen. Signals and systems (second Edition). printed in the United States of America, 2002
- [6] 赵红怡, 张常年编著. 数字信号处理及其 MATLAB 实现. 北京: 化学工业出版社, 2002
- [7] 李行一主编. 数字信号处理. 重庆: 重庆大学出版社, 2002
- [8] 胡广书编著. 数字信号处理理论、算法与实现 (第二版). 北京: 清华大学出版社, 2003
- [9] 张小虹编著. 数字信号处理习题详解. 南京: 东南大学出版社, 2002
- [10] 薛年喜主编. MATLAB 在数字信号处理中的应用. 北京: 清华大学出版社, 2003