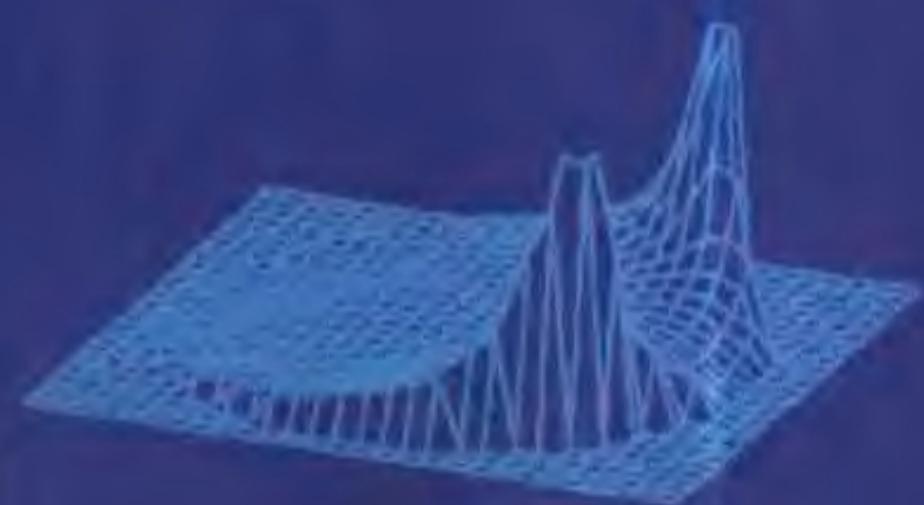
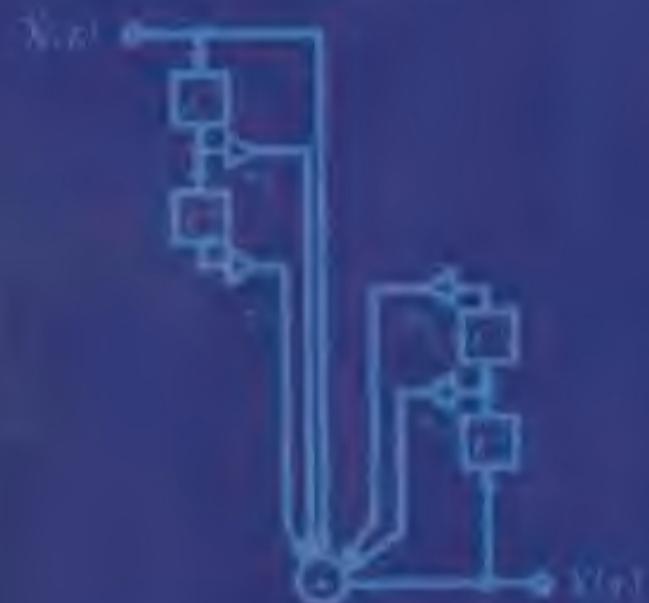
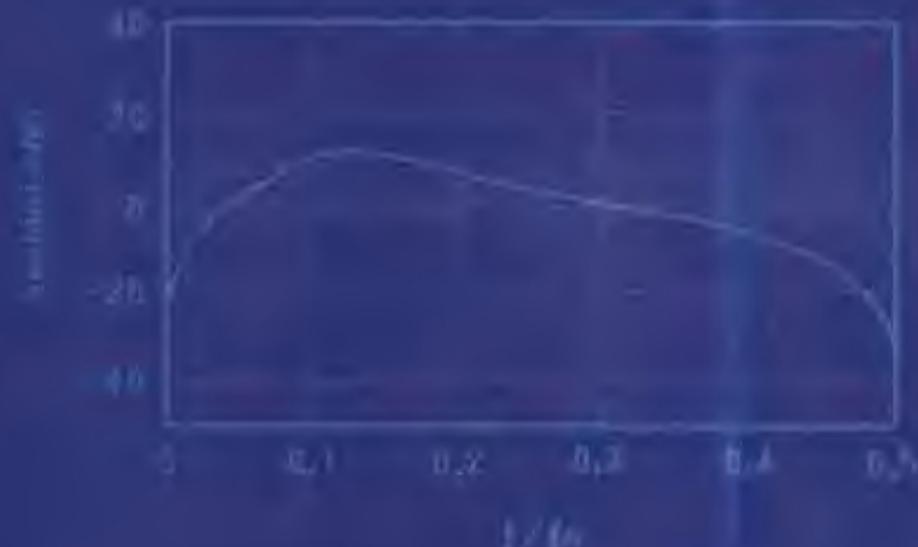


生物医学数字信号处理

[美] Willis J. Tompkins 著

林家瑞 徐邦荃 等译

C-LANGUAGE EXAMPLES
AND LABORATORY
EXPERIMENTS FOR
THE IBM® PC



责任编辑：叶见欣
封面设计：秦 茹

- C语言编程
- UW DigiScope软件可设计各种滤波器并显示处理结果
- ECG信号放大电路
- 详细解释数字信号处理的直观方法
- 检测ECG信号主要特征的实时算法

ISBN 7-5609-2579-0

9 787560 925790 >

定价：38.00元

(附磁盘一片)

生物医学数字信号处理

[美]Willis J. Tompkins 著

林家瑞 徐邦荃 等译

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制



A0967189

华中科技大学出版社

图书在版编目(CIP)数据

生物医学数字信号处理 / [美]Willis J. Tompkins 著
林家瑞 徐邦荃 等译

武汉:华中科技大学出版社, 2001年11月

ISBN 7-5609-2579-0

I. 生…
I. ①林… ②徐…
I. 数字信号-信号处理-应用-生物医学工程
N. R318



© 1993 by PTR Prentice-Hall
A division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

生物医学数字信号处理

[美]Willis J. Tompkins 著
林家瑞 徐邦荃 等译

责任编辑:叶见欣

封面设计:蔡茹

责任校对:张兴田

责任监印:张正林

出版发行:华中科技大学出版社

武昌喻家山 邮编:430074 电话:(027)87545012

经 销:新华书店湖北发行所

印 刷:湖北省新华印务有限公司

开本:787×1092 1/16

印张:16.25

字数:374 000

版次:2001年11月第1版

印次:2001年11月第1次印刷

印数:1—2 000

ISBN 7-5609-2579-0/R·21

定价:38.00元

(本书若有印装质量问题,请向出版社发行部调换)

内 容 简 介

本书译自美国威斯康辛大学电气与计算机工程系教授、前 IEEE 生物医学工程学会主席 Willis J. Tompkins 编著的“Biomedical Digital Signal Processing”一书。主要介绍生物医学数字信号处理中最重要的基础知识，基本处理方法及其在生物医学工程中的具体应用。本书共分 14 章：信号转换技术，数字滤波器基础，有限冲激响应滤波器，无限冲激响应滤波器，整数型滤波器，自适应滤波器，信号平均技术，数据压缩技术，时域和频域分析方法，ECG 的 QRS 复波检测，ECG 分析系统，VLSI 在数字信号处理中的应用等。

全书以生物医学信号中最常见的 ECG 信号作为处理对象，强调实时性的处理方法。紧紧结合课程内容几乎每章有一个实验，可让学生进行实践。

本书可供生物医学工程专业及有关专业作为教材，也可供有关研究人员、技术人员作为数字信号处理的实用参考书。

译者序

当人类正进入信息时代,而数字化又是信息时代发展的方向。因此,数字信号处理技术已成为每一位电子信息科学工作者必须掌握的重要知识。这种知识用之生物医学工程领域就形成了生物医学数字信号处理学科,在国内、外,它已成为生物医学工程专业本科生与研究生教学中的主干课程或学位课程。

《生物医学数字信号处理》一书译自[美]Willis J. Tompkins 教授编著“Biomedical Digital Signal Processing”(PRENTICE HALL, 1993 年版权)。该书作者为[美]威斯康辛大学(麦迪逊)电气与计算机工程系教授,他曾任[美]IEEE 生物医学工程学会主席。他与著名学者 J. G. Webster 教授合作编著的生物医学工程系列教科书,已先后被译成中文为我国广大读者所欢迎。它们是:《带微处理机的医学仪器设计》,吕维雪译,人民卫生出版社,1986 年;《传感器与 IBM PC 接口技术》,林家瑞,罗述谦等译,华中理工大学出版社,1993 年。这样,本书也就成为 W. J. Tompkins 教授编著的第三本中译本。像作者编著的其他教科书一样,本书也是编著者多年教学实践经验的结晶。该书思路明晰,学用结合。内容反映了生物医学数字信号处理中最重要的基础知识、基本处理方法及其在生物医学工程中的具体应用,主要是在 ECG 中的应用。本书共分 14 章,包括信号转换技术,数字滤波器基础,有限冲激响应滤波器,无限冲激响应滤波器,整数型滤波器,自适应滤波器,信号平均技术,数据压缩技术,其他时域和频域分析方法,ECG 的 QRS 波检测,ECG 分析系统,最后简要地介绍了 VLSI 在数字信号处理中的应用。全书以生物医学信号中最常见的 ECG 信号作为处理对象,强调实时性的处理算法。重点突出,举一反三,这是该书重要特点之一。

该书重要特点之二是每章附有大量的复习性、思考性与拓宽性习题,全书共 238 道题,有利于学生课后的复习思考。

该书重要特点之三是外附有一张与之配套的磁盘,为学生提供了一个实用性强的 UW DigiScope 软件。该软件为学生提供能在 PC 机上操作并运行的 9 个实验。从第二章开始,紧紧结合课程内容,几乎每章有一个实验,逐步地让学生设计各种类型数字滤波器,并观察数字滤波器的处理结果。实验中所需的各种 ECG 输入信号均由数据文件读取并加以显示。实验装置除 PC 机外,无需其他的附加的硬件设备,省钱、方便、实用。

译者对 Tompkins 教授为我们提供原版书及其对译书成功的恳切期望,深表诚挚的感谢。

参加本书翻译工作的有林家瑞(第 1、2、3、4、9 章),徐邦荃(第 5、6 章),周云波(第 10 章)、黄春明(第 7 章)、林岚(第 8 章),陈瑞红(第 11 章),原芸(第 12 章),杨振野(第 13 章),朱帆三(第 14 章)以及叶晖、黄敏(附录),全文最后由林家瑞校阅定稿。黄敏及使用本书英文版教材的研究生们对本书的出版问世也做了许多有益工作,在此表示感谢。由于我们的水平有限,译文中会有一些错误,欢迎读者提出宝贵意见。

译者

2001 年 5 月于华中科技大学

前　　言

在生物医学信号处理中,广泛应用数字滤波与模式识别算法。医疗仪器通常应用于临床,例如,危重病房的计算机必须用于采集病人的心电图、识别信号的主要特征、判断其正常与否,并及时地为医生提供这些信息。

在生物医学工程的课程中,学生们通常缺少动手从事设计的机会。数年前,Wisconsin 大学,Madson 实验室将单板机用于实时数字信号处理算法的实验,使学生可以对算法的操作过程进行分析。然而,由于硬件只能采用内置微处理器的汇编语言进行编程,不易通过再编程实现新的算法,因此,学生自行设计和编写信号处理算法的机会受到了一定限制。

目前,在数字信号处理课程中,滤波器设计是非交互式的。学生们缺少机会通过硬件来构成滤波器,并观察其实时性能,这使学生对滤波器的设计和实际特性不能得到深刻理解。然而本书提供了作者开发的一套实验及专用软件,以供对生物医学信号处理感兴趣的读者进行研究。实验除需要一台 IBM PC 机或其兼容机外,不需要任何其他设备。因此,本书提出的概念和方法不仅适用于生物医学工程领域,也适用于电子工程领域的学生,作为信号处理课程的内容。本书还为拥有信号转换硬件的读者提供了动手实践的指南。

本书内容包括用于生物医学数字信号处理的基础知识和用于设计并实现简单数字滤波器的 C 语言程序。所有例子均用 Turbo C (Borland) 语言编写。选用 C 语言的目的是因为以往的学习方法所需要的汇编语言编程知识缺乏灵活性,且很难从概念上将信号处理算法与其汇编语言相联系,而高级 C 语言可使学生能更好地理解滤波器与实现程序间的联系。

本书所提供的实验均可用模/数转换器或虚拟转换器(即软件仿真)来实现。实验可在设备齐全的实验室或任何 IMB PC 机及其兼容机上完成。对 PC 机的唯一要求是至少应有 640KB 的 RAM 随机存储器和 VGA(或 EGA 或单色)图形显示卡。图形显示设备为信号处理结果的可视化提供了高分辨率的环境。在某些应用场合中,尤其是进行频域处理时,数学协处理器可用于加快计算的速度,但并不是非要不可。

本书配套软盘中包括专用软件 UW DigiScope。学生可用该软件进行实验、设计各种数字滤波器,观察数字滤波器的处理结果。除虚拟转换器外,该软件还支持以下两种设备:实时设备公司(Real-Time Devices)的内置式 ADA2100 信号转换卡和通过串行口连接的摩托罗拉公司(Motorola)的外置式 68HC1IEVB 卡。虚拟转换器用软件仿真信号转换,从数据文件中读取输入波形。软盘中附有一些已采样信号样本和用于以 UW DigiScope 软件格式录入用户数据的程序。希望通过数据录入程序及标准文件格式来达到读者之间共享生物医学数据的目的。

本书第一章综述了计算机在医学中的应用,回顾了该领域的发展历史。由于全书将 ECG 作为演示数字信号处理技术的典型生物医学信号,因而在第二章中回顾了 ECG 检测技术。本章的实验通过 ECG 放大电路阐述了模拟信号的提取与预处理问题。第三章介绍了数据转换技术,所附的实验旨在使学生对提取数字信号的技术和可能犯的错误有深刻的理解。

第四、五、六章介绍数字信号处理的基本原理。与使学生颇感困惑的充满数学方程式的传统方法不同,本书强调从概念上进行理解,使学生能尽快地参与设计过程,并减少对基础理论的论证。这三章的两个实验可利用 UW DigiScope 软件使学生有设计和运行数字滤波器的实践机会。

第七章介绍一种特殊的整系数型数字滤波器。该类滤波器不需进行浮点运算，因而在实时数字信号处理中特别有用。但大多数数字信号处理书籍都未包括这类滤波器。本章的实验通过设计练习来帮助学生加深理解。

第八章介绍自适应滤波器。这种滤波器不断学习环境特征，用改变滤波特性的方法来优化所需的功能。第九章综述了信号平均技术及其在医学工程中的应用。

第十章介绍数据压缩技术。在存储、处理及传输数据时，减少数据量是非常重要的。由于越来越多的信号需要进行数字化存储和处理，数据压缩就越来越重要。本章的实验使用书中提供了数据进行压缩及重建的练习。

第十一章总结其他重要信号处理方法，重点强调频域方法，并通过实验演示了 ECG 信号的频域分析。

第十二章展示了检测心电基本特征的大量方法，如本章的实验所示，其重点在于实时算法。第十三章介绍以上技术在实际医疗监护系统中的应用。

第十四章总结了用于数字信号处理的新的超大规模集成电路技术，并展望其未来发展趋势。

附录 A、B、C 介绍了两种信号转换设备以及可用于任何 PC 机的虚拟转换器的接口及其使用。

附录 D 是专用软件 UW DigiScope 的用户手册。附录 E 介绍 UW DigiScope 的信号生成功能，以便学生以所需的噪声和采样频率生成仿真信号，并可存在磁盘文件中供虚拟信号转换器使用。

我要特别感谢 1991 年秋季参加“计算机在医学中的应用”(ECE463)课程的学生，感谢他们帮我从书中和软件中发现了很多小错误(以及少许问题)。我还要感谢第五章的作者 Jesse Olson，感谢他使得 UW DigiScope 成为一个可靠的教学工具。

与大多数数字信号处理课程中的离线数字信号处理技术不同，本书实验中的算法和安排都强调实时数字信号处理的方法。

Willis J. Tompkins

[美]威斯康辛大学(麦迪逊)

电气与计算机工程系

目 录

第一章 计算机在医学中的应用	(1)
1.1 医学数据的特点	(1)
1.2 医学仪器	(1)
1.3 医疗过程的重复性	(2)
1.4 微处理器系统的发展	(3)
1.5 基于微型计算机的医学仪器	(8)
1.6 数字滤波器的软件设计	(9)
复习思考题	(10)
第二章 心电图	(11)
2.1 基本的心电图	(11)
2.2 ECG 导联系统	(17)
2.3 ECG 信号特点	(20)
2.4 实验:模拟滤波器、ECG 放大器及 QRS 检测器	(21)
复习思考题	(26)
第三章 信号转换	(28)
3.1 采样技术基础	(28)
3.2 简单信号转换系统	(30)
3.3 生物医学信号的转换要求	(31)
3.4 信号转换电路	(31)
3.5 实验:信号转换	(38)
复习思考题	(39)
第四章 数字滤波器基础	(41)
4.1 数字滤波器	(41)
4.2 z 变换	(41)
4.3 数字滤波器的组成	(43)
4.4 数字滤波器的类型	(44)
4.5 差分方程的传递函数	(45)
4.6 z 平面的零-极点分布图	(45)
4.7 橡胶膜的概念	(47)
复习思考题	(53)
第五章 有限冲激响应滤波器	(55)
5.1 FIR 滤波器的特性	(55)
5.2 平滑滤波器	(57)
5.3 陷波滤波器	(62)
5.4 求导数	(63)
5.5 窗的设计	(66)
5.6 频率采样	(67)

5.7 最小设计.....	(68)
5.8 实验:FIR 滤波器的设计	(68)
复习思考题	(70)
第六章 无限冲激响应滤波器	(71)
6.1 IIR 滤波器的通用表达式	(71)
6.2 简单单极点实例.....	(72)
6.3 积分器.....	(73)
6.4 双极点滤波器的设计方法.....	(78)
6.5 实验:用于 ECG 分析的 IIR 数字滤波器.....	(82)
复习思考题	(83)
第七章 整数型滤波器.....	(89)
7.1 基本设计概念.....	(89)
7.2 低通整数型滤波器.....	(92)
7.3 高通整数型滤波器.....	(93)
7.4 带通和带阻整数型滤波器.....	(94)
7.5 滤波器级联效应.....	(94)
7.6 其他快速设计方法.....	(95)
7.7 设计举例和设计工具.....	(96)
7.8 实验:用于 ECG 分析的整数型滤波器	(100)
复习思考题.....	(101)
第八章 自适应滤波器	(103)
8.1 基本的噪声消除器模型	(103)
8.2 自适应滤波器消除 60Hz 干扰的正弦波模型	(104)
8.3 自适应滤波器的其他应用	(106)
8.4 实验:60Hz 自适应滤波器	(107)
复习思考题.....	(107)
第九章 信号平均	(109)
9.1 信号平均的基础	(109)
9.2 信号平均器	(112)
9.3 典型的平均器	(112)
9.4 信号平均软件	(112)
9.5 信号平均的局限性	(113)
9.6 实验:ECG 信号平均	(113)
复习思考题.....	(114)
第十章 数据压缩技术	(115)
10.1 转折点算法.....	(115)
10.2 AZTEC 算法	(117)
10.3 Fan 算法	(121)
10.4 Huffman 编码	(123)
10.5 实验:ECG 数据压缩算法	(126)
复习思考题.....	(127)

第十一章 其他时域及频域分析方法	(130)
11.1 傅里叶变换	(130)
11.2 相关性	(134)
11.3 卷积	(137)
11.4 功率谱估算	(139)
11.5 实验:ECG 的频域分析	(142)
复习思考题	(142)
第十二章 ECG 的 QRS 复波的检测	(143)
12.1 ECG 的功率谱	(143)
12.2 带通滤波方法	(144)
12.3 差分法	(146)
12.4 模板匹配法	(147)
12.5 QRS 复波检测算法	(149)
12.6 实验:实时 ECG 处理算法	(157)
复习思考题	(158)
第十三章 ECG 分析系统	(160)
13.1 ECG 分析	(160)
13.2 ST 段分析	(163)
13.3 便携式心律失常监护仪	(164)
复习思考题	(169)
第十四章 VLSI 在数字信号处理中的应用	(170)
14.1 数字信号处理器	(170)
14.2 高性能的大规模集成电路信号处理器	(172)
14.3 VLSI 在医学中的应用	(174)
14.4 用于生物医学信号的传感器	(175)
14.5 VLSI 开发工具	(176)
14.6 设计方案的选择	(176)
复习思考题	(176)
附录 A UW Digiscope 软件的硬件配置	(177)
A.1 在 IBM PC 机中安装实时设备 ADA2100	(178)
A.2 摩托罗拉 68HC11 EVBU 的配置	(179)
A.3 摩托罗拉 68HC11 EVB 的配置	(182)
A.4 虚拟输入/输出设备(数据文件)	(183)
A.5 在二进制文件上加一个文件头:ADDHEAD	(183)
附录 B 数据采集与控制程序	(185)
B.1 数据结构	(185)
B.2 顶层设备程序	(186)
B.3 内部 I/O 设备(RTD ADA2100)	(187)
B.4 外部 I/O 设备(摩托罗拉 68HC11 EVBU)	(188)
B.5 虚拟 I/O 设备(数据文件)	(190)
附录 C 数据采集与控制——若干提示	(193)

C. 1	内部 I/O 装置(RTD ADA2100)	(193)
C. 2	外部 I/O 装置(摩托罗拉 68HC11 EVBU)	(195)
C. 3	虚拟 I/O 装置(数据文件)	(198)
C. 4	自编接口软件	(200)
附录 D	UW DigiScope 程序使用手册	(204)
D. 1	UW DigiScope 程序的简要介绍	(204)
D. 2	各种功能的总结	(205)
附录 E	信号发生器	(212)
E. 1	信号发生方法	(212)
E. 2	信号发生器程序(GENWAVE)	(212)
附录 F	有限长寄存器效应	(215)
F. 1	量化噪声	(215)
F. 2	有限循环	(216)
F. 3	缩放比例	(218)
F. 4	IIR 滤波器中的舍入误差	(218)
F. 5	浮点寄存器滤波器	(219)
F. 6	总结	(219)
F. 7	实验:数字滤波器中的有限长寄存器效应	(219)
附录 G	商用 DSP 系统	(220)
G. 1	数据采集系统	(220)
G. 2	DSP 软件	(221)
英汉对照索引词		(222)
参考文献		(239)

第一章 计算机在医学中的应用

计算机在医学中的应用是十分广泛的,本书仅介绍了其中的一小部分。这里需要强调的是实时信号在医学仪器中的应用是很重要的。本章着重讨论医学数据的特性,医学仪器的通性和有关医学领域。介绍微处理器在当代医学仪器设计中的地位,以及微处理器医学仪器系统的开发历史,并举例说明微处理器在医学仪器中的重要作用及用其开发的新仪器。最后讨论软件设计和个人计算机在医学仪器开发中的重要作用。

1.1 医学数据的特点

图 1.1 所示的是医院医学数据的三个基本部分:文字数字、医学图像及生理信号。其中文字数字信息包括病人的姓名、地址、编号、检验结果和医生的处方,医学图像数据包括来自 X 光片、CT、磁共振和超声波检测的图像信息。生理信号则包括心电图、脑电图和血压等信号。

三种数据由不同系统进行处理,通常由通用的大型计算机来组织和管理。对于文字数字信息,用一般的主计算机就可建立和管理数据库。医学图像数据通常以底片存档,图像存档和交换系统将图像在光盘上以数字形式存储,并通过高速局域网按要求将它们分配给各医院的高清晰度显示器上。

在手术室进行手术过程中,那些生理信号应被实时监视。当这些连续信号出现异常时,临床医师能立即从仪器上得到这些信号。

生理信号是最重要的一类信息。生理信号中常被用来监视病人状态的是心电图,后面的章节中将把它作为典型的信号来举例说明。

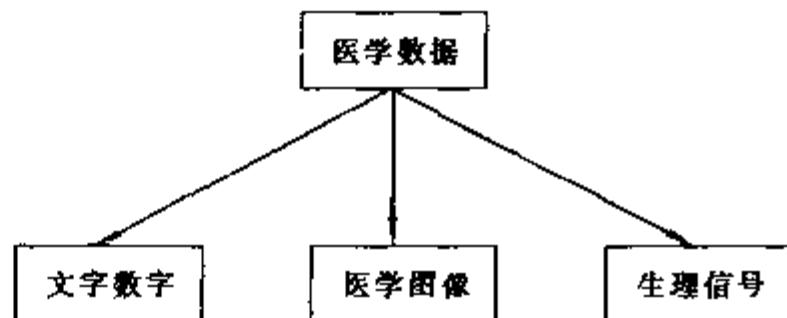


图 1.1 医学数据的类型

1.2 医学仪器

医学仪器有许多不同的类型。图 1.2 所示的是这类仪器系统的基本组成框图。传感器用来测量病人的生理信号,并产生类似于实际信号的电信号(通常为时变的)。

一对电极在人体的表面可以感受不同的电压,如心电压或脑电压。不同类型的传感器可分别将体温、血压转换成电压。由传感器产生的电信号将输到微处理机去进行信号的处理和分析,并通过网络对原始信号或分析结果进行显示、记录和分配。在某些仪器中,处理器执行控制功能。基于信号分析的结果,处理器可命令控制器直接对病人进行治疗(闭环控制)或在对病人的信号进行处理时要有人的介入(开环控制)。

这里列举两种典型的医学仪器,并介绍了它们是怎样按照上面的方框图工作的。第一种是大型检查系统,它可同时监视许多的病人。第二种是心脏起搏器,它体积特别小,必须安装在病

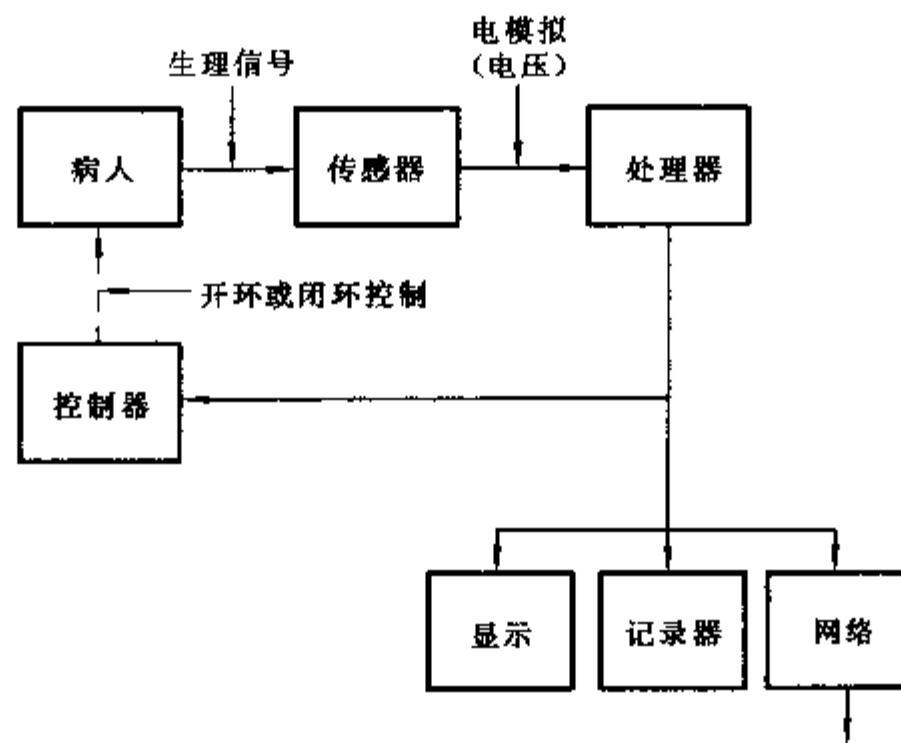


图 1.2 医学仪器系统的基本组成

人体内。

在 ICU 环境中，常常有一些传感器与接受诊断检查的每个病人相连，处理器（实际上常常不只一个）监视和分析所有受检查的人。若处理器发现异常，它就会向医务人员报警。显示器允许医务人员查看原始数据，如查看每个病人的 ECG、心率和血压等数据。在 ICU 系统中，工作网中央控制台控制部分仪器。另一个工作网是将 ICU 系统与医院的其他数据库联系的。一个闭环例子是时时使用的输液泵。传感器监视从病人中收集来的尿液，观察流量损失，然后处理器命令泵输入适量的流体给病人，使之体液达到平衡。

对于安放心脏起搏器的患者，电极被安装在导管上，然后再放置在心脏里，这种处理器通常设计有一种专门的积分电路而且是一种超低功率装置，因而不是普通的处理器。这种处理器监视来自心脏的心电图，分析它并决定心脏是否自身跳动。若很长时间没有看到自身的激励信号，它就将产生电脉冲。这种电脉冲有足够的电流，通过电极和监控器，引起心脏的跳动。这种装置被称为闭环疗法传送系统。最早的起搏器是工作在开环状态下的，无论心脏是否跳动，它总以固定频率简单地激励心脏。这种装置远不如现代智能机器令人满意。通常，微处理器通过体外一种装置用遥测方法与起搏器交换信息，显示和记录它的控制参数。这种装置也能设置新的控制参数，如电流激励和振幅。此外，有些装置还可通过电话网络与中央处理器交换信息。

1.3 医疗过程的重复性

图 1.3 所示的是医疗服务系统的工作过程。数据采集是健康服务的开始，医师要求病人讲述病史，记录心电图，测试血压和其他一些可以诊断病人的测试。当然，医疗器械对数据采集的某些方面有一定的帮助，甚至可做某些数据的准备工作。总之，医师分析采集的数据，然后确定病人的基本问题是什么。医疗过程就按图 1.3 所示的闭环系统的程序不断地采集数据、分析、决策、治疗，直到病人痊愈为止。

图 1.2 所示的医学仪器的功能是医疗服务系统的一种模式。

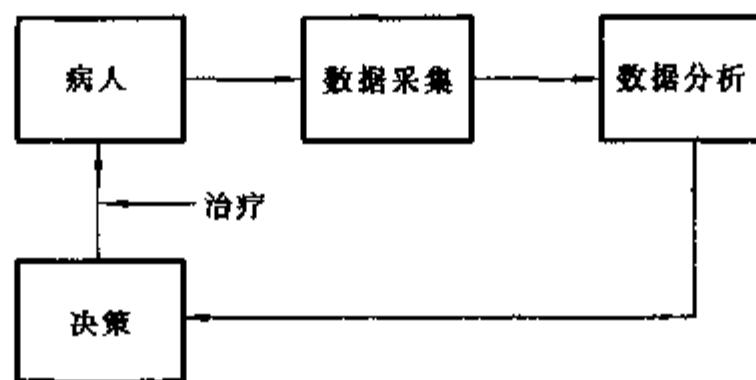


图 1.3 医学管理系统的基本组成

1.4 微处理器系统的发展

最近 10 年来，在生物医学仪器设计中，基于微型计算机的医学仪器有了重大的突破。基于微型计算机的医学仪器的发展趋向于高智能化。越来越多的运算器和存储器集成在尽量小的空间内。带有很强的计算功能的 PC 机的商业化加速了用电池供电、病人可随身携带的仪器的发展。在病人的日常生活中，这种器件像一台个人计算机一样可方便地为病人寻找他的问题。微型计算机以及普通 PC 机已成为生物医学实验的一种重要工具。

基于微型机的医学仪器随着微处理器自身的发展而发展。图 1.4 所示的是 Intel(英特尔)

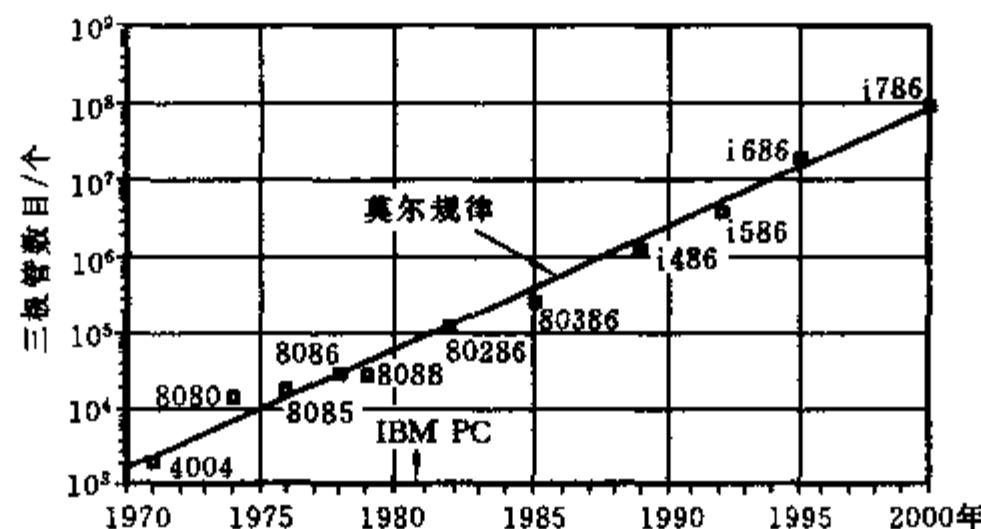


图 1.4 微处理器的发展

微处理器中晶体管数量过去一直按指数增长，这一趋势还将继续下去

公司开发的微处理器中所含的晶体管数量随时间而变化的曲线图。图中所示的是从 1971 年带有 2000 个晶体管的积分电路开始，直到今天含有 1 000 000 个晶体管的中央处理器的演变情况，莫尔(Moore)发现，Intel 公司在 20 多年来生产的微处理器上的晶体管的数量是每年翻了一番。图中实线反映这种 Moore 规律。该公司预言：他们生产的微处理器中的晶体管数目将以指数级数增长。这样，在不到 10 年时间内，开发出一种在每块硅片上有 1 亿个晶体管的并行处理装置，这种微处理器定会成为超级型的。它是最有可能比当今的任何一种高级计算机的功能更为强大的微处理器，并成为台式计算机的一部分，而且有可能由电池供电而使之应用于便携式仪器中。

从 1971 年开始的微处理器到如今功能强大的微处理器，其发展对生物医学仪器的设计产

生了重大影响。在较小的集成电路里压缩更多的运算器和存储器就可得到功能更强的计算工具,使PC机在生物医学计算方面成为一种有效的工具。正因为基于微处理器的系统在计算能力、存储能力上不断提高,而在尺寸、价格、能源消耗上不断降低,以上设想才得以实现。

1.4.1 个人计算机的发展

图1.5所示的是计算机的发展历史,这个历史是从第一批,如19世纪Charles Babbidge发明的机械式计算机开始的,一直到现代的个人计算机,如IBM PC机和Apple Macintosh机。在20世纪前唯一的机械式计算机是在齿轮和机械联动装置基础上工作的。

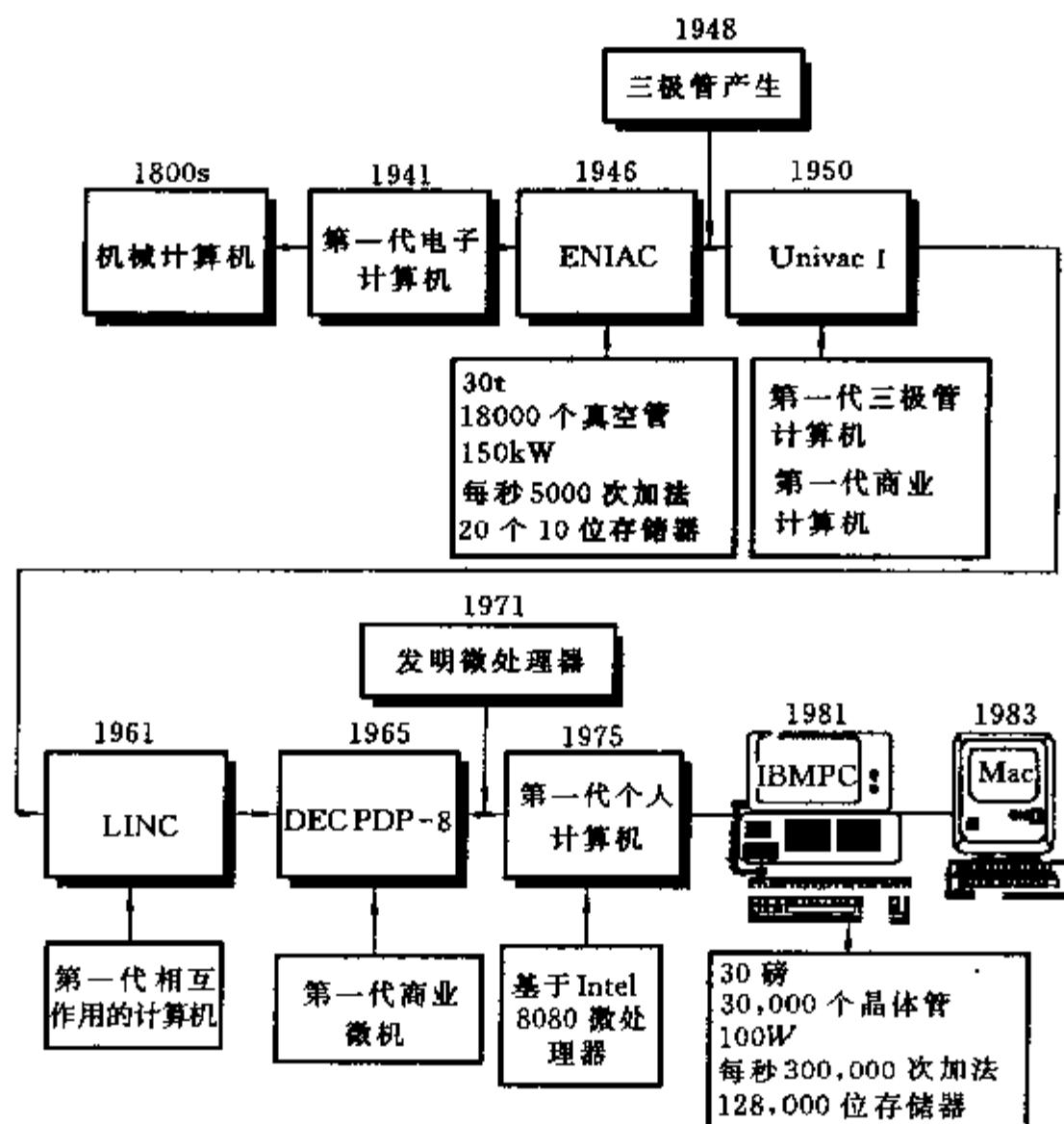


图1.5 个人计算机的发展

1941年,一位名叫Atansoff的研究者创造出第一台电子数字式计算机。这种装置即使和今天四功能的袖珍计算器相比,其功能也是很低的。第一台正式通过验收的数字计算机被称为ENIAC(电子数字积分器和运算器),于1946年在宾夕法尼亚大学电子工程Moore学院研制成功,和现代PC机相比,其功能很简单,这台设备占了Moore学院地下室面积的一大部分,而且还要一台相当大的空调机来冷却其中成千上万的真空管。

晶体管的发明导致了Univac I机的诞生,它是第一台商业计算机,包括IBM公司在内的几个公司终于把晶体管计算机投入了市场。1961年,在麻省理工学院和林肯实验室工作的研究者们使用当时的技术创造了一台与商业计算机迥然不同的新颖的微型计算机,这种由分立元件组成,带有磁心存储器的晶体管微型计算机称为LINC机(实验室仪器计算机),它的诞生是PC机发展中最具历史意义的进步。

设计者的基本设计目标是将一台普通用途的计算机转变成用于生物医学计算方面的实验

设备。正如设计者们所预料的,由于这样的计算机只要改变存储器中存储的程序,其功能就可能被完全改正过来,所以它具有广泛的用途。这样,这种计算机不仅在一般计算意义上可作为一台解决方程式的装置来运行,而且通过再编程(软件),它又能模拟许多其他用途的实验仪器。

LINC 机是应用于生物医学方面最成功的微型计算机,另外,它的设计包含了现代 PC 机中所期望的种种特征。它是世界上第一台对话式计算机。LINC 机有键盘和显示器,而不像它以前的其他计算机那样要用穿孔卡,所以使用者可以坐下来直接编程。它是第一台具有对话式图形显示和在功能上类似现在按键的数字计算机。它还有固有的信号转换装置,仪器接口硬件,有小巧可靠的数字磁带录音机和发音能力,可直接从每一位病人身上获得一份心电图,而且在图形显示器上显示波形。

如果 LINC 机更小一些(它大约有大水箱那么大小),更便宜一些(一套 50 000 美元),它一定会成为一流的个人计算机。

在 LINC 机之后,商业小型计算机问世。它也是由数字装备公司生产的由分立元件组成的 PDP—8 LINC 机。接着,此公司生产了一个新商业版本的带有 PDP—8 的 LINC 机,称为 LINC—8 机,后来又生产了更先进的版本称为 DDP—12 机。由于 LINC—8 机是第一台商业小型计算机,我对它有一种特殊爱好。LINC—8 机能显示图形,不需要使用一些棘手的中间介质,如穿孔卡和穿孔纸带等。在 1960 年后期,在 LINC—8 机上编写的程序之一是计算和显示病人的矢量心电图环(见第二章)。像这样的程序在现代 PC 机上,使用高级语言,如 Pascal 或 C 是很容易实现的。

1971 年发明了第一台微处理器,其中央控制单元的功能较差,而价格又相对较昂贵。直到 20 世纪 70 年代中期,8 位微处理器,如 Intel 8080 的有效使用,才使第一台家用微型计算机得以诞生。Altair 8800 机是以 8080 微处理器为基础开发的。

1.4.2 PC 机的普及

一个具有重要历史意义的标志是 1981 年 IBM PC 机的推出。英文缩写 IBM 代表个人台式计算机。早于 IBM PC 机的流行的计算机使用的是 8 位的 Zilog Z80 微处理器(Intel 8080 的增强产品),带有 CP/M(为微处理器的控制程序)操作系统,但没有统一的标准格式化软盘,所以很难从一个公司的 PC 机上将数据转换到另一个 PC 机上。IBM 公司几乎在一夜之间独立地将 16 位 Intel 8088 微处理器、微软 DOS(磁盘操作系统)标准化了,其产品都使用相同格式化的软盘,因此可将数据从一台计算机上转移到另一台计算机上。许多国际计算机公司都生产与 IBM PC 机兼容的计算机,这为编程软件的发展提供了一个标准的计算平台。由于如此多相似的计算机产生,因此价格低廉,功能强大的应用程序开始发展起来。

图 1.6 所示的是微处理器是如何推动台式计算机发展的,这个图是基于一台 PC 机在任何时间都按 5000 美元的价格销售的假设推导出的。在 1981 年,第一台 IBM PC 机引入 Intel 8088 微处理器后,它的计算速度为 0.1 MIPS,假设每台需 5000 美元,则需 10 台 IBM PC 机或 50000 美元的代价才能提供 1MIPS 的计算能力。随着以 Intel 80286 微处理器为基础的 IBM PC/AT 机的产生,1 台价格 5000 元的台式计算机就能提供 1MIPS 的计算能力。最近的 IBM PC 机使用了 Intel 486 微处理器,它的计算速度是 10MIPS,同样它的价格也是 5000 美元。可见,具有不同计算能力的 IBM PC 机的价格随着微处理器的发展按对数规律减少。

另一个重要的标志是 1983 年 Apple Macintosh 机的诞生。这种计算机使一种简单、直观的

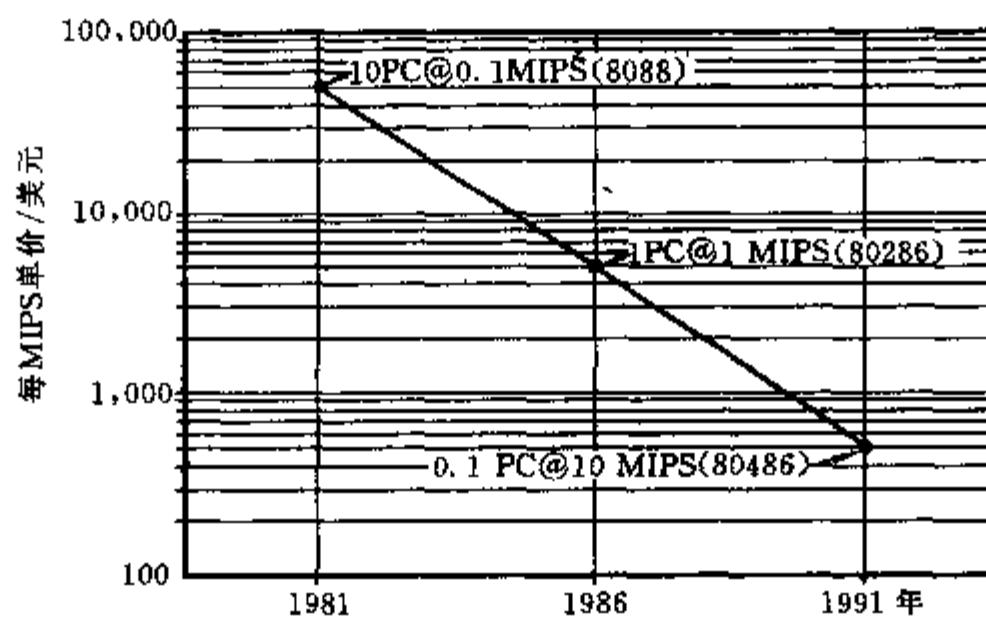


图 1.6 计算机的计算功能与价格之间的反向关系

人机界面普及化。自从那时起,有许多设计者就在为 IBM PC 机实现相似的人机界面而努力。

自 IBM PC 机诞生以来,在工业中发生了许多变化。台式 PC 机随着技术的发展不断地演变和提高,尤其是微处理器本身。现在已生产出折叠式和手掌式的轻便和电池供电的 PC 机,无论走到哪里,都有能量提供给计算机。

图 1.7 所示的是在全功能 PC 机中电子组成部分的数目随着时间增长按指数规律下降的情况。将来还会继续下降。在 1983 年,每个 PC 机大约有 300 个积分电路,其中一半作为微处理器和支持逻辑电路,而另一半则用来组成 256B 的存储器。此后 PC 机中的芯片数又由带有 256B 存储器的 300 个积分电路变为现在 2MB 的 18ICS。直到 20 世纪 90 年代中期,4MB 内存的计算机才由三部分电子元件组装而成,即:中央处理单元,ROM(用于基本的输入输出软件)和 4M B 的动态随机存储器(DRAM,用于用户编程和存储数据)。从而 PC 机朝着功能强大,存储容量大,所占空间小,低成本,低功耗的方向发展。

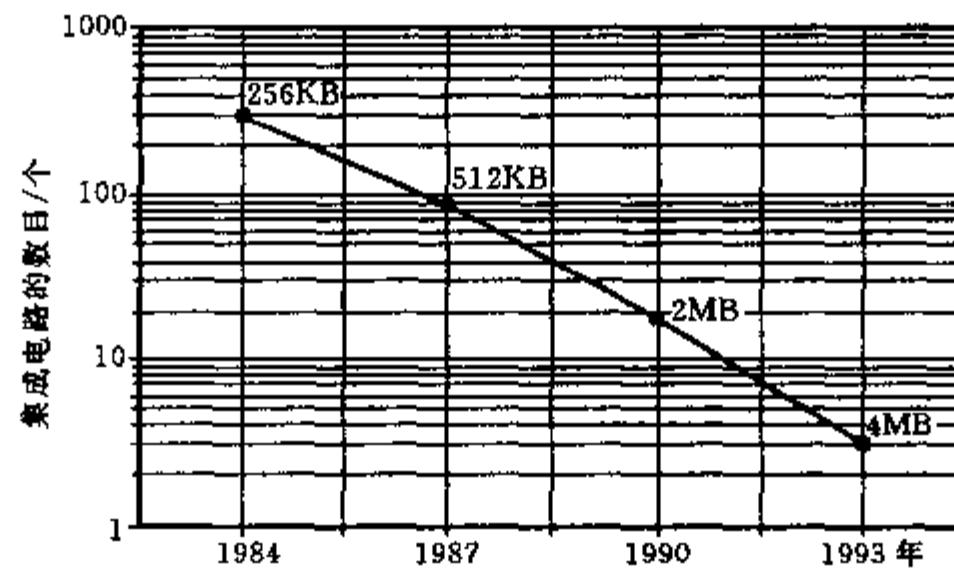


图 1.7 PC 机电子元件的变化

随着计算功能的强大,PC 机的元件数目却逐渐减少

图 1.8 所示的是人脑和现代 PC 机的比较。由图可知,计算机的优点和缺点与人脑间的关系。从这种关系看,个人计算机比人脑有更大的能量消耗和更精确的数量级。PC 机“大脑”中的功能计算元件和记忆功能与人脑少几个数量级。

系统	重量/ 1b	大小/ cm ³	功率/ W	CPU 器件	存储器/位	导电率/ 个脉冲/s	Benchmark/ 次加法/s
IBM PC	30	5	200	10 ⁶ 个 晶体管 (or equiv.)	10 ⁷	10 ⁵	10 ⁶
大脑	3	0.05	10	10 ¹⁰ 个 神经元	10 ²⁰	10 ²	1
PC/大脑	10	100	20	10 ⁴	10 ⁻¹³	10 ³	10 ⁶

图 1.8 PC 机和人脑的特性比较

PC 机有什么好处呢？答案已在表中最后两项的内容中显示。计算机传输速度远远超过大脑的传送速度。这可用一个基准来解释，即叫一个人将一个数加一个数再加一个数等等，在每次相加后报出结果。PC 机作这种计算任务比人脑快一百万倍。如果发挥它的这种优势，PC 机应用于这种任务将大大超过人类。

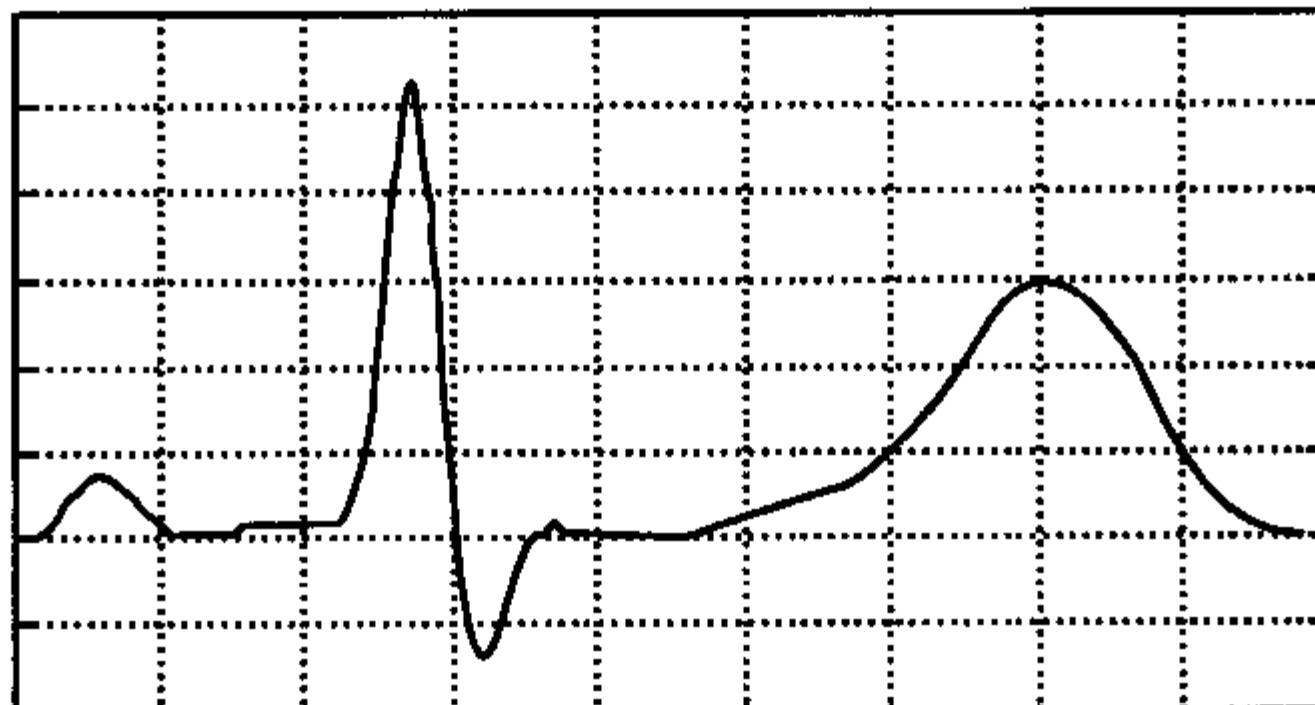
图 1.9(a)所示的是一个数列，所有数据被计算机处理前置于基本格式中。这些数据是图

```

0 0 0 0 2 5 8 10 13 14 14 14 12 11 9 7 5 4 2 1 1 0 0 1 1 1 1 1 2 2 2 3 3 3 3 3 3 3
3 3 6 11 20 33 51 72 91 103 105 96 77 53 27 5-11-23-28-23-17-10-5-1
0 1 2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 2 2 3 3 4 4 5 6 7 8 9 10 10 11 11 12 12
12 13 14 16 18 20 22 24 27 29 31 34 37 39 42 44 47 49 52 54 55 56 57 57 58 58
57 57 56 56 54 52 50 47 43 40 36 33 29 26 23 20 17 14 12 10 8 7 5 3 2 1 1 0 0 0

```

(a)



(b)

图 1.9 心电图的两种视域

- (a) 计算机中表现为随着时间变化的幅值；
- (b) 人看到的是随时间变化的波形

1.9(b)所示的心电图的等距幅值，它们是用模数转换器从每秒 200 次的采样频率采样心电图得到的。心电图的数字表示叫做数字信号。从多年研究这些信号的特征的经验可知，人类眼睛

系统尤其擅长于分析模拟信号波形和判断这样一个信号是正常或不正常。

为了对信号作出判断,计算机需要用软件分析这列数字。这些软件一般包括数字信号处理软件(它为本书的重点),以及为分析生物医学和医学图像的判定逻辑软件。

1.5 基于微型计算机的医学仪器

在过去 10 年中,台式和手提式计算机已取得长足的进展,它给基于 PC 机或专用微型计算机的设备提供了方法,这为解决以前不能解决的医学上的问题打下基础。人们的兴趣之一是设计轻、小,且用电池供电的手提式装置。这种典型的信号处理设备是一个真实的个人计算机,它能将来自于传感器或放置在人身上的电极的信号进行处理。

1.5.1 基于微机便携式的医学仪器

微机式的医学仪器的一个例子是便携式心律失常监护仪,它可检查从胸电极得到的病人的心电图,并实时进行分析,确定是否有心律异常现象。早在 10 年前已设计出这种装置的雏形,由于那时技术的局限性,这种装置与现代可作为商品的便携式心律失常监护仪相比还很原始。技术的发展允许我们去做更深入的扩展。现在,当病人到急诊室就诊时,可用心脏监视器协助医师诊断心脏的异常现象。在仔细设计下,同一装置在手术室外科手术的过程中能配合手术检查病人心脏情况,连续听病人的心跳是否正常。病人住院的整个过程中,这个装置都在观察他,在特别护理、正规诊治,甚至病人在就餐的过道上发生情况时,它都会给医务工作者发出警报。这种装置可放在病人家中,特点是可在做完心脏手术后放在病人家中进行监控。第十三章将详细地讨论便携式心律失常监护仪的原理。

在市场和研究实验室里有许多便携式生物仪器的例子。另一种以微型计算机为基础的装置,称为 CALTRAC,它使用小型加速表检查身体和运动量,然后将这一活动测量值转换成同等数量的热量值,并将累加结果在 LCD 显示器上显示出来。现在有一种起搏器,它用加速表测量病人的活动,以便调节其步伐快慢。

目前已开发了一种便携式仪器,它可监测几个压力通道,这些通道从置于食道的导管上的传感器引出来。它可检测吞咽时压力变化的信号,然后将这些信号记录在半导体存储器上,便于以后传递给 IBM PC 机,对数据进行分析。

另一种便携式仪器,可监测置于鞋中的压力传感器,它提取病人脚下压力分布信号,诸如脚中无知觉的糖尿病患者脚下压力分布信号的变化。

1.5.2 基于 PC 机的医学仪器

大批量生产的经济性使台式 PC 机成为许多生物医学研究用仪器的主机。许多公司使用 PC 机来做诸如采样分析生理信号的工作,在医院临床工程部,用于维护仪器数据库以及生理系统的模拟和仿真。

把 PC 机当做一种通用的实验仪器,使许多生物医学问题的研究更加方便。研究人员可对其编程,运用通常的硬件界面便可适用于不同的应用场合。例如将 PC 机用于康复工程,如可为盲人设计这样一个系统,它可将视觉图像转成触觉图像系统。PC 机从摄像机中获取图像并将它存储在存储器中,程序通过能激压盲人指尖的触觉压力感受器将图像信息一点点传递到盲人的指尖。用这种方法,来研究盲人通过触觉“看”图像的能力。

另一项应用是基于 Apple Macintosh I 计算机的电阻抗成像——EIT。通过电极向人体输入无害高频电流，测量电极阻抗，而不用常见的 X—CT 所采用的有害射线。该方法基于身体内器官的电阻值不同的原理，将电极置于体表面来测得阻抗，使人体的内部器官成像。

计算机控制装置通常装有 32 通道的电流发生器，它将高频(50kHz)的电流输入人体，然后计算机将采样值通过模/数转换器转换成电流，产生体表面电势分布。另外可利用胸腔的有限元模型和边界，测定出各点的值，再经过反复计算使之能满足所测数据。使用计算机的绘图功能，则可得到身体横向面的图像。在该项目中，Apple Macintosh 计算机承担了所有仪器操作任务，包括输入电流的控制，电阻率的测量，解决精算算法以及最后图像的显示。

在医学仪器中，PC 机有许多应用。目前已用 IBM PC 机对心电图的分析开发出信号过程和人工神经网络的算法，还开发出数据压缩技术，以减少保存心电图数据所需的存储空间。

1.6 数字滤波器的软件设计

在实验时，除了要选择好计算机硬件系统外，还必须进行软件选择。对于特种硬件系统，这种选择常常是封闭的和有限的。图 1.10 所示的是硬件和客观世界之间的三个层次的软件（操作系统、支持软件和应用软件）支持。应用软件使得计算机能作为医学仪器运用，软件的选择情况对各种装置有很大影响。

两个大的软件选择是磁盘操作系统的选和语言的选择。虽然采用各种操作系统和语言的不同组合都能适应同一台机器，但这种选择通常还是很重要的，因为某一选择常常会比另外几种要好。

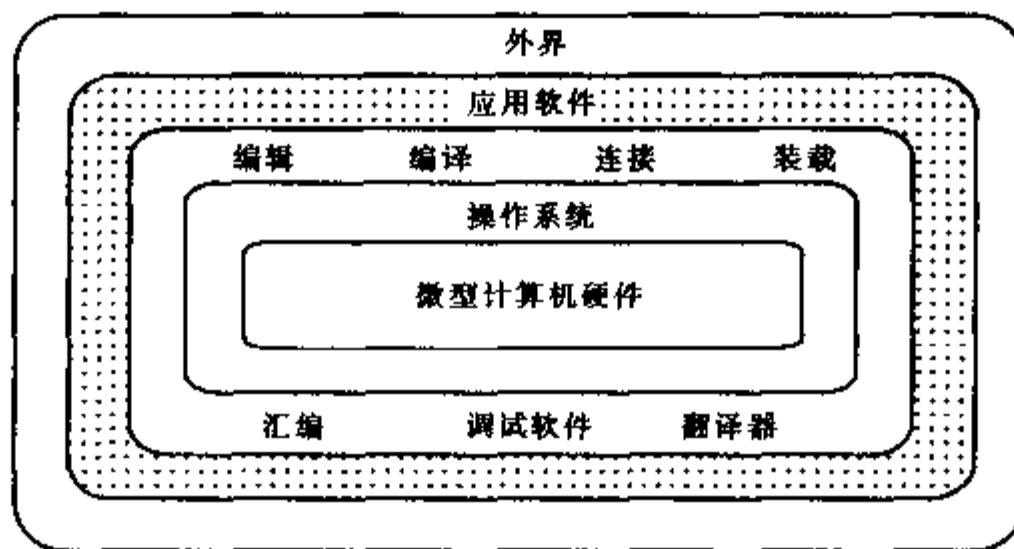


图 1.10 微机硬件系统与客观世界之间存在三种层次的软件

在实际应用中，常需用软件来实现实时信号处理算法，故这里围绕该问题来展开讨论。对不同的人而言，实时计算的概念完全不同。本书的实时计算概念与电子游戏机的实时计算概念相似。微机作为游戏机的中央处理单元时需要在短时间内完成计算并得出结果。如果游戏机每次发射的子弹需要 1~2min 后才能确定轨迹，判断出能否击中敌方的飞船，则这游戏就没有什么乐趣了。

在生物医学上一个典型的实时处理例子是医院护理过程的心电图分析。在这个典型的医学应用中，病人通常与监视器相连，心脏每跳动一次，喇叭就响一声。若这个监视器的微型计算

机需要 1~2min 做这种复杂的模式识别,也就是说,对于每次有效的心跳,都要鸣叫 1min 左右,则这种装置是无用的。

考虑实时的要求,运行的 DOS 应具有强的灵活性和可用性,图 1.11 所示的是一般微型计算机的磁盘操作方案,DOS 能解决许多不同类型的问题,包括实时操作等。

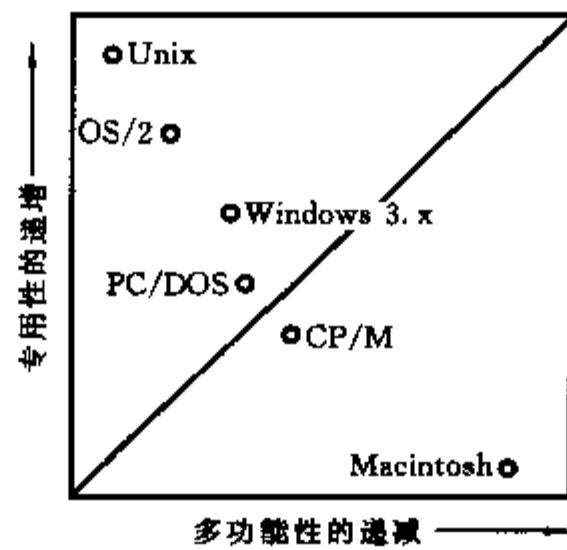


图 1.11 磁盘操作系统

复习思考题

1. 1 比较开发实时程序的各种操作系统,解释各自的优点。
1. 2 解释“阐释”、“编译”和“集成环境编译”三语言的差别,对每种类型进行举例说明。
1. 3 列出用 C 语言进行仪器实时应用开发的两个优点,说明为什么这两个优点很重要。

第二章 心电图

用于诊断心脏病的主要技术之一是心电图(Electrocardiogram,简称ECG)。心电图机通过测量心电在体表的电位来推测心脏的缺陷。心电图机可以探测心率和其他心脏参数。

2.1 基本的心电图

临床 ECG 的测量有三种基本方法。第一种方法是最常用的标准临床 ECG,此方法在病房中进行,有 12 个 ECG 导联的电位差用于记录病人体表电位。第二种方法使用不同的体表电位作为三维心电激动向量模型的输入,称为向量心电图(Vectorcardiogram,简称 VCG)的心脏激动模型。第三种方法用于危重病人的长时间强制观察,只用一个或两个 ECG 导联来监视或记录心脏节律,这种方法称为节律分析。ECG 的三种基本方法如下:

- (1) 标准临床 ECG(12 导联)。
- (2) VCG(3 向量导联)。
- (3) 监护 ECG(1 或 2 导联)。

图 2.1 所示的为 ECG 的基本目标,通过观察

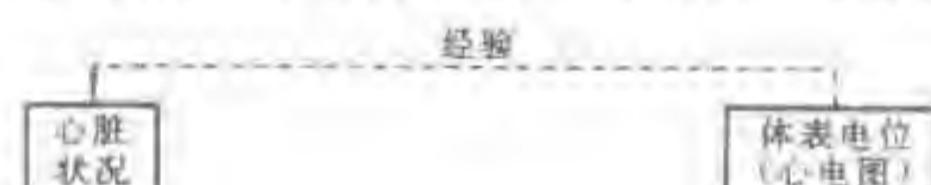


图 2.1 心电图的目的是用无创体表电位
测量方法来推测心脏机电状况

体表的电信号,心电图的专家希望能探测到心脏的功能状态。尽管 ECG 信号是电信号,但心脏的机械状态的改变也会导致相应的体表电位扩散,因而会改变体表 ECG,早期心电图的研究基于对成千上万病人多年的 ECG 信号记录以及不同病人之间的不同波形的关系,因此,临床 ECG 中大量的是经验性的,取决于实验知识。心电图专家从其他专家处获取 ECG 信号中的各部分的含义,而这些专家又从其他专家那里获得有关知识。

图 2.2 所示的为 20 世纪初 Einthoven 用于记录最早的 ECG 的情况,它将一缸盐水作为连接到体表的电连接器,线性电流表作为记录 ECG 的仪表。

2.1.1 电极

随着时代的发展,金属电极已成为了体表的电连接器。一个由盐溶液和胶组成的电极层成为金属电极与皮肤的电接触面。身体内部的电流是由离子运动产生的,而在导线中电流是由电子的运动产生的。电极系统可完成离子电流到电子电流的转换。

(约于 1890)

镀镍的导电金属片可用做 ECG 电极,但存在一些问题。用于获取 ECG 的两个电极与充盐的躯体的作用



图 2.2 Einthoven 的早期 ECG 测量方法

就像一个电池。在电极之间出现的直流漂移电位可以等于或大于 ECG 信号的峰值。电极两面会出现电层(正、负离子以一定距离分开)。病人身体的运动导致的电极变化会改变双面充电层状态并改变直流漂移电位。因为这些小失调电压会随 ECG 放大 1000 倍,因此,小信号的变化也会导致输出信号的基线漂移。这种电极称为极化电极,只能用于对安静病人的检查。

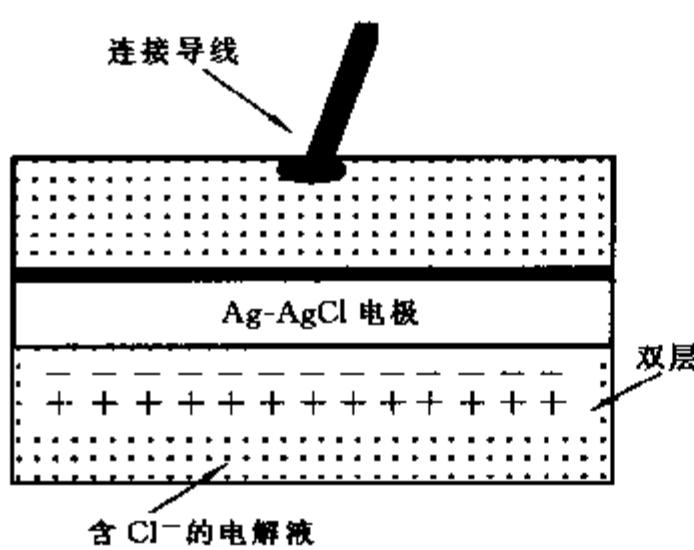


图 2.3 Ag-AgCl ECG 电极

目前最常用的电极是 Ag-AgCl 电极,因为它更接近非极化电极。图 2.3 所示的为这种电极。这种电极漂移电位非常小。它在 Ag 层上镀了一层 AgCl。氯离子将在身体内、电极内,以及在 AgCl 层内运动,在这里转换成在 Ag 中的电子运动并传导到导线中。这种方法把直流漂移电位减小到与峰值相比非常小的程度。因此,这种电极移动导致的基线漂移比其他极化电极要小得多。

2.1.2 心电等效发生器

图 2.4 所示的是心电等效发生器,它用于测量心电变化的物理模型中。最常用的物理模型是双极电流源模型,这在数学上表示为时变矢量,适用于临床矢量心电图(VCG)检测。Einthoven 推导出心电激励可以作为一个矢量模型。他还认识到,四肢就像躯体上点之间的直接连接线一样,因为体内双极源电流主要在躯干内部流动而对四肢没有影响。因此他观察到了电极可以直接连到每一个肩上以及肚脐上。

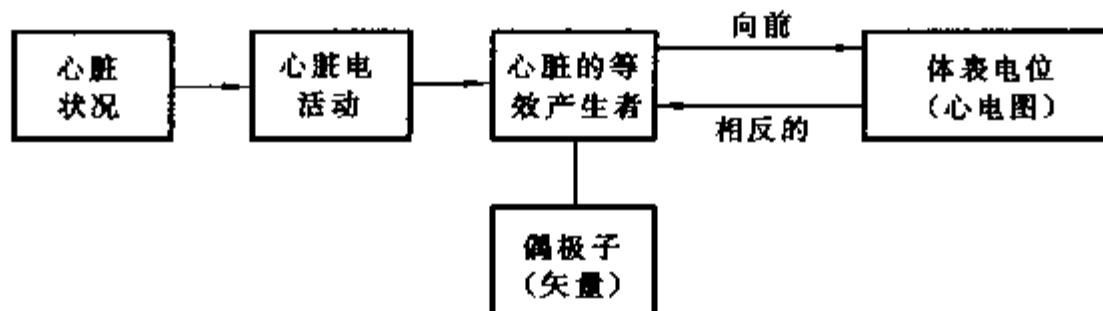


图 2.4 心脏活动模型称为心电等效发生器

Einthoven 画了一个以两肩与腰部作成的三角形,而且是一个等边三角形,这个三角形如图 2.5 所示(RA—右臂,LA—左臂,LL—右腿)这就是著名的 Einthoven 等边三角形。如果代表心电激励扩散的矢量为已知,那么两臂之间所测得的电位差(即三角形的两个边)简单地正比于连接肢干的三角边的矢量的投影。图示表明了 Einthoven 矢量与 3 个前向肢干导联中的每一个关系(导联 I、II、III)。正符号表示每个导联连到仪表放大器的正端。

因为双极的大小和方向(作为心脏的去极化单元)在整个心跳过程中都在改变,因此,有以下的矢量表达式

$$\mathbf{P}(t) = P_x(t)\mathbf{x} + P_y(t)\mathbf{y} + P_z(t)\mathbf{z} \quad (2.1)$$

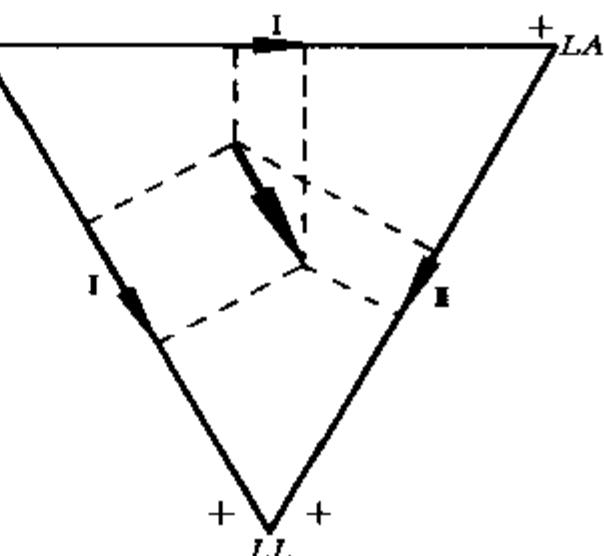


图 2.5 Einthoven 等边三角形 RA 和 LA 分别是右臂和左臂,LL 是左腿

式中, $P(t)$ 是时变心电矢量; $P_i(t)$ 是矢量的正交分量, 也称为比例矢量; x, y, z 是 x, y, z 方向的单位矢量。

20世纪50年代, 一个名叫Frank的权威VCG研究者做了一个如图2.6所示的人体石膏像, 对它做了防水处理, 并将它充以盐水。在躯体模型的心脏部位放置了由两个电极组成的双极源。一个电流源向电极输送电流并且在整个导体区产生电流。利用石膏像内置的电极, Frank测量了许多点的由电流源产生的体表电位分布。

根据测量结果, 他发现了每一个体表电位与双极源有关的几何传递系数。

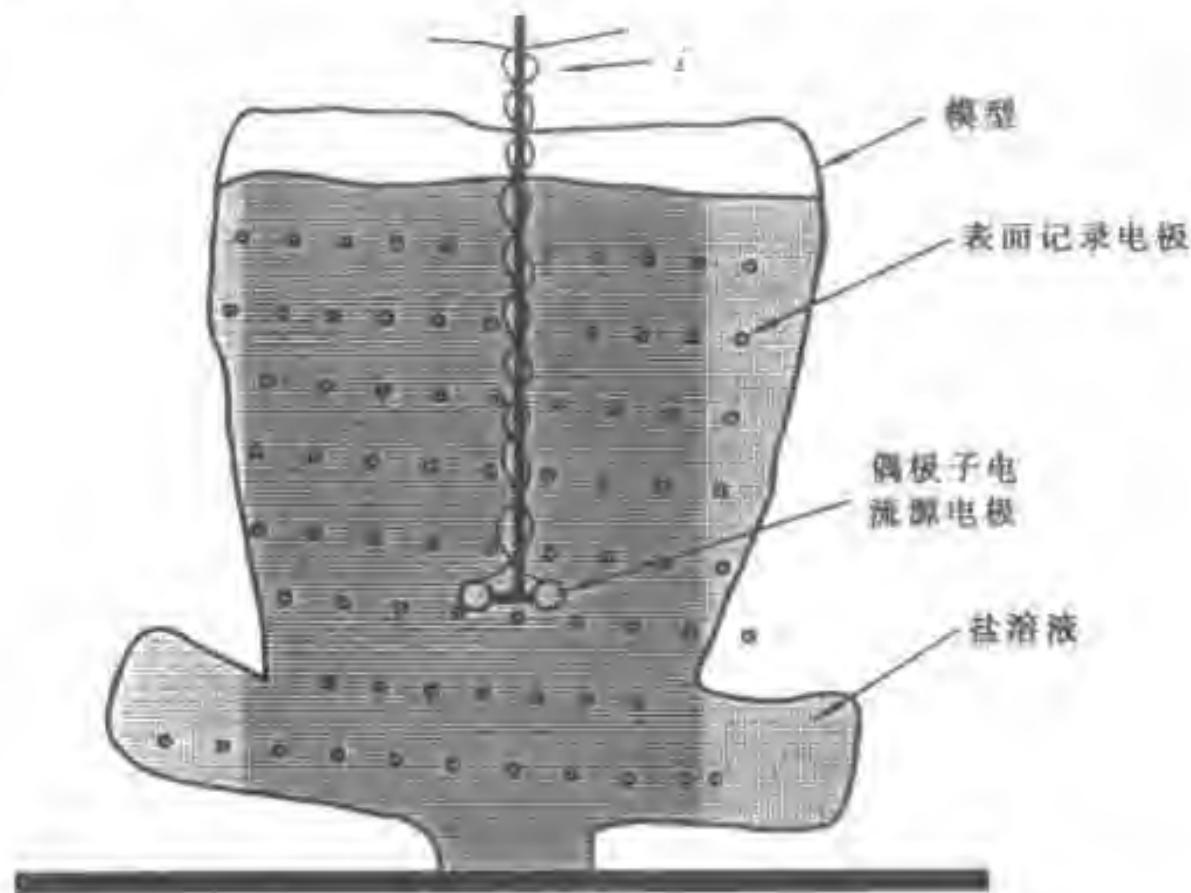


图2.6 矢量心电图Frank导联系统的躯体模型

一旦传递系数已知, 那么对于任何双极源的前向ECG就可以求解。对于一个给定的心电双极, 可以得到体表任意点的电位递推解, 其数学表达式为

$$V_n(t) = t_{nx}P_x(t) + t_{ny}P_y(t) + t_{nz}P_z(t) \quad (2.2)$$

这个递推解表明, 电压 $V_n(t)$ (即ECG)在体表任意点 n 的值可以用一组传递系数 $[t_{ni}]$ 积的线性和来表示, $[t_{ni}]$ 对于该点和相应的正交双极矢量分量 $[P_i(t)]$ 来讲是唯一的, ECG是时变的, 双极分量也是时变的, 而传递系数只取决于结构与非均匀性, 因此, 对于一组 k 个体表电位(即导联), 存在一组 k 个方程, 可以表达为下面的矩阵形式:

$$V = T \times P \quad (2.3)$$

式中, V 是时变电压的 $k \times 1$ 矢量; T 是传递系数的 $k \times 3$ 矩阵, T 对于不同的人来讲是各不相同的; P 是时变的心脏 3×1 矢量。

当然对于某个人来讲, 心脏矢量和传递系数是未知的。但是如果有方法计算心脏矢量, 那么就可以将它用于递推方程的求解, 以及求得体表任意处的ECG。求解这个问题的方法是基于人体躯干的物理模型。该模型提供了许多体表点电位与心脏矢量的相关系数。有了这个信息, 可以选择三个ECG导联来模拟所希望测量的异常ECG内部特性。然后反向求解得到心电双极向量:

$$P = B \times V \quad (2.4)$$

式中, B 是一个 $3 \times k$ 矩阵, 是直接由传递系数矩阵 T 转换推导得出的导联系数矩阵。因此, 对于三个心脏矢量分量, 有三个如下的线性方程:

$$P_x(t) = b_{x1}V_1(t) + b_{x2}V_2(t) + \cdots + b_{xk}V_k(t) \quad (2.5)$$

如果我们选 k 个体表 ECG 导联 $\{V_1(t), V_2(t), \dots, V_k(t)\}$, 而导联系数可以由人体躯干模型得出, 则可以反向求解得出时变心脏矢量。一旦得到这些双极分量, 就可以用式(2.3)求递推解, 进而求出体表任意处的 ECG。

2.1.3 ECG 的起源

图 2.7 所示的是用附于体表的电极和与其相连的仪表放大器来测量 ECG 的示意图。在某

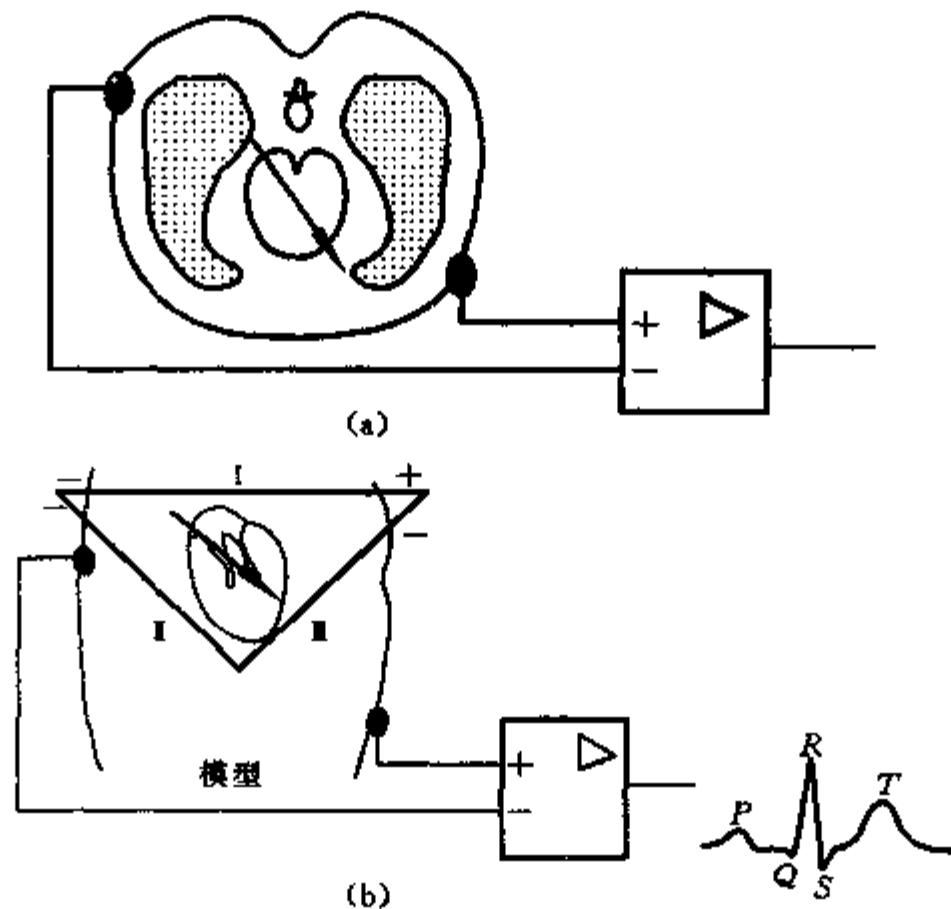


图 2.7 记录 ECG 的基本连接图

(a) 示出了心和肺的身体横截面; (b) 导联 I 的电极位置的前视图

时刻某点, 当矢量指向仪表放大器的正端连接点时, ECG 信号就向正方向变化。如果矢量指向负极, 那么 ECG 就为负。时变的心脏矢量在一个心跳周期内将产生具有 P 波、T 波和 QRS 复合波的 ECG。图 2.8 所示的是导联 I 记录的一次心跳的典型正常 ECG。

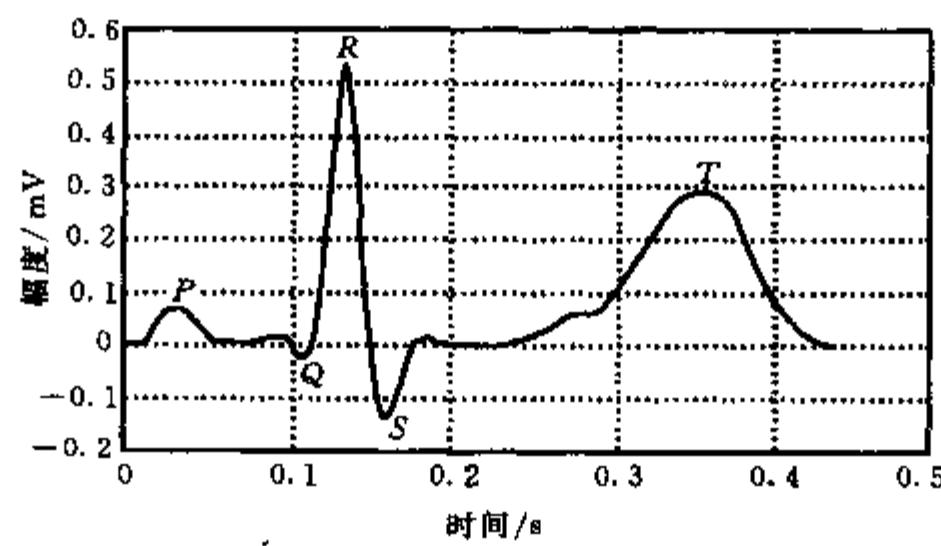


图 2.8 一个典型正常心跳 ECG

图 2.9 所示的是在不同点不同时间用与体表 ECG 起因有关的矢量表示的激励的心电扩

散图,ECG 信号放大图如图 2.8 所示。图 2.9(a)所示的是起于 SA 节点的前庭慢速去极化产生的 P 波,图 2.9(b)所示的是在 AV 节点信号被延迟并导到 P 波后的电隔离区,Purkinje 系统开始对腹部肌肉提供刺激后,Q 波就开始出现了。图 2.9(c)所示的是腹部肌肉的快速去极化图,它被描绘为一个大而快的矢量,并产生 R 波。图 2.9(e)所示的是腹部去极化的最后一个相位出现情况,激励朝腹部基准处(图的顶点)扩散并导致 S 波的出现。

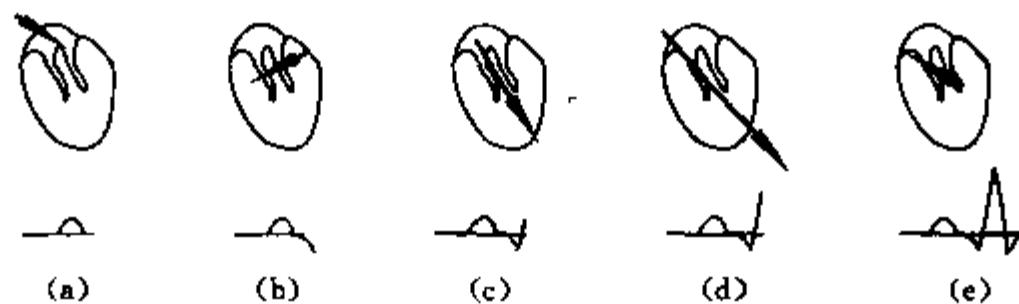


图 2.9 各个时刻心电活动的传播与 ECG 产生过程之间的关系

2.1.4 标准肢体导联

图 2.10 所示的是观察肢体之间的电位差的方法(*RA*—右臂,*LA*—左臂,*LL*—左腿),作为理想电压源可以用具有高输入阻抗的仪表放大器来测量,显然这三个电压形成了闭合的电压环路。根据 Kirchhoff 电压定律,环路电压之和为零,即

$$V_1 - V_2 - V_3 = 0 \quad (2.6)$$

可以写出任一导联与其余两个导联电压的关系式:

$$V_1 = V_2 + V_3 \quad (2.7a)$$

$$V_2 = V_1 - V_3 \quad (2.7b)$$

$$V_3 = V_1 - V_2 \quad (2.7c)$$

显然,从这些公式中可知,有一个电压完全是多余的,因为可以根据其中两个求出第三个。实际上这正是现代 ECG 机所采用的。大多数 ECG 机可以测量导联 I 和 II 电压并计算出导联 III 电压值。因为有一个电压是冗余的,所以不含有导联 I 和 II 之外的任何新信息。要弄清这个问题,可回到图 2.1 中并回顾一下心电图专家怎样根据 ECG 特性与不同异常情况之间的关系来发现心脏病的,以及心脏病 ECG 标准导联电压之间的关系,问题就会很清楚了。

2.1.5 加压肢体导联

早期的仪表放大器的增益不大,还不足以观察到所有的 ECG,如图 2.11 所示的框图可产生更大的信号幅值。在这种情况下,左臂信号,称为加压肢体导联电压 V_{LL} ,它用其他两个肢体的平均电位作为测量基准。

我们可以用电路理论来分析这个图,根据左下环路,有

$$iR + iR - V_1 = 0 \quad (2.8)$$

或者 $iR = \frac{V_1}{2} \quad (2.9)$

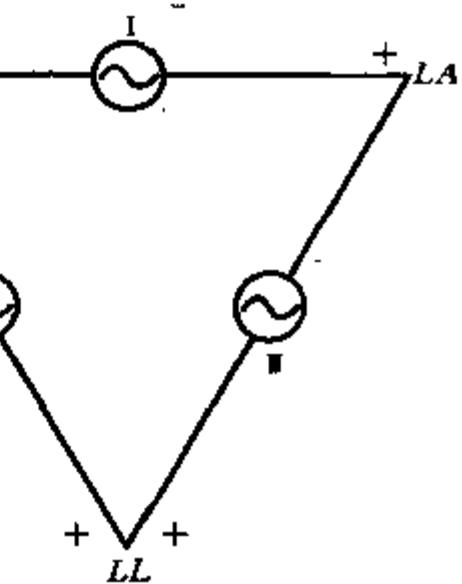


图 2.10 导联 I、II 和 III 是肢体之间的电位差

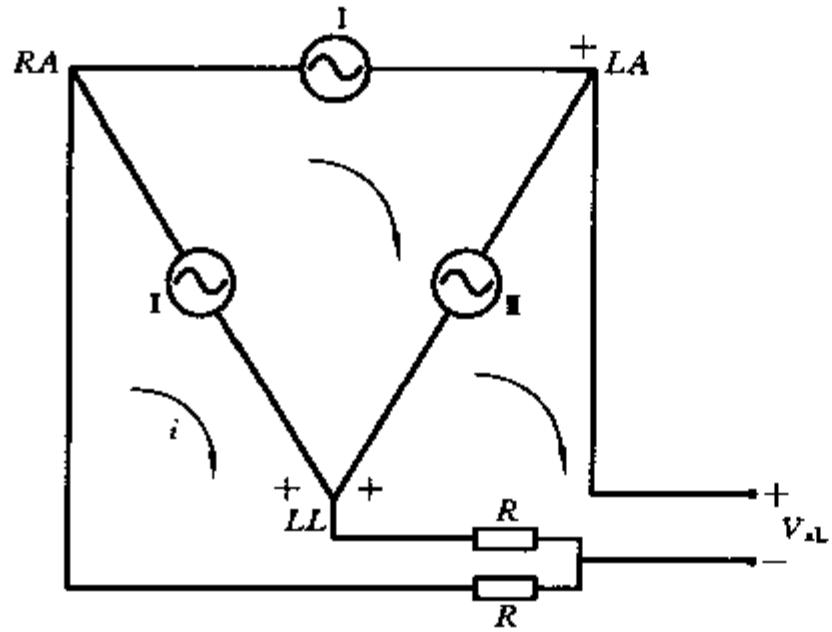


图 2.11 加压肢体导联 V_{AL}

根据右下环路,有

$$-iR + V_1 + V_{AL} = 0 \quad (2.10)$$

$$\text{或者} \quad V_{AL} = iR - V_1 \quad (2.11)$$

联合式(2.9)和式(2.11),可得

$$V_{AL} = \frac{V_1}{2} - V_1 = \frac{V_1 - 2 \times V_1}{2} \quad (2.12)$$

由上中环路,可得

$$V_1 = V_1 + V_1 \quad (2.13)$$

替换得:

$$V_{AL} = \frac{V_1 + V_1 - 2 \times V_1}{2} = \frac{V_1 - V_1}{2} \quad (2.14)$$

这就是以增强的肢体导联电压 V_{AL} 作为两个前肢导联电压平均值的 Thevenin 等效电压。显然 V_{AL} 是一个多余的导联,因为它可以由其他两个导联来得到。其他两个增强导联电压为 V_{AR} 和 V_{AF} ,同样也可以表达成导联 I 和 II 的函数。这样可找出三个多余的导联,它们都可以用前肢导联来计算,所以它们是完全多余的,毫无新的实际信息。然而由于 ECG 的经验性的本质特点,医生仍然要看看这些导联的特征以便更好地诊断。

图 2.12 所示的是用短路理想电压源并从输出端看进去的方法求 Thevenin 等效电阻的步骤。

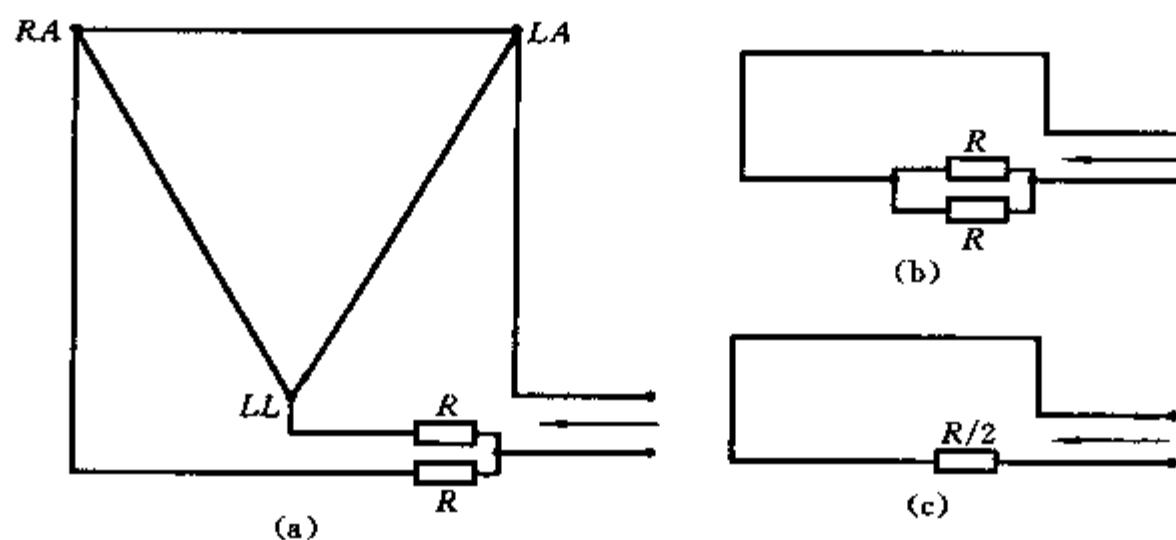


图 2.12 V_{AL} 等效电路的 Thevenin 电阻测量

(a)所有理想电压源均短路;(b)两个相同电阻的并联;

(c)等效的 Thevenin 电阻为 $R/2$

图 2.13 所示的是一个记录系统,包括一个连在仪表放大器正输入端的附加电阻,其值等于 Thevenin 等效电阻。这可以平衡放大器的每个输入的电阻,以获得最优的共模抑制比(CMRR)。

图 2.14 所示的是根据两个标准肢体导联电压来求加压肢体导联电压的向量方法。该肢体导联的矢量朝向它们相应的三角形边的方向并集中于原点。比如求 V_{AL} 时,可用正在测量的两个肢体的导联向量,比如它们连到左臂上。用导联 I 作为求和的一个向量,因为其正向末端连到左臂。我们可以忽略肢体导联 II 的矢量(即旋转 180°),因为其负向末端连接左臂。导联的电压 V_{AL} 是导联 I 和 -II 之电压和的一半(见式(2.14))。

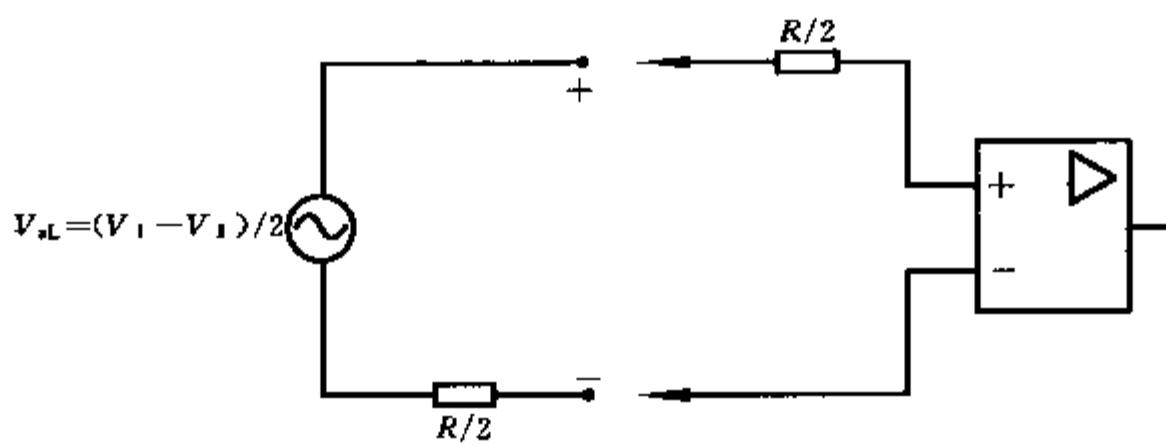


图 2.13 获得最好的共模抑制比
在仪表放大器的正输入端加了一个值为 $R/2$ 的 Thevenin 等效电阻

图 2.15 所示的为前肢导联的整组向量。从图中可以看出，三个加压导联是前向导联的函数。

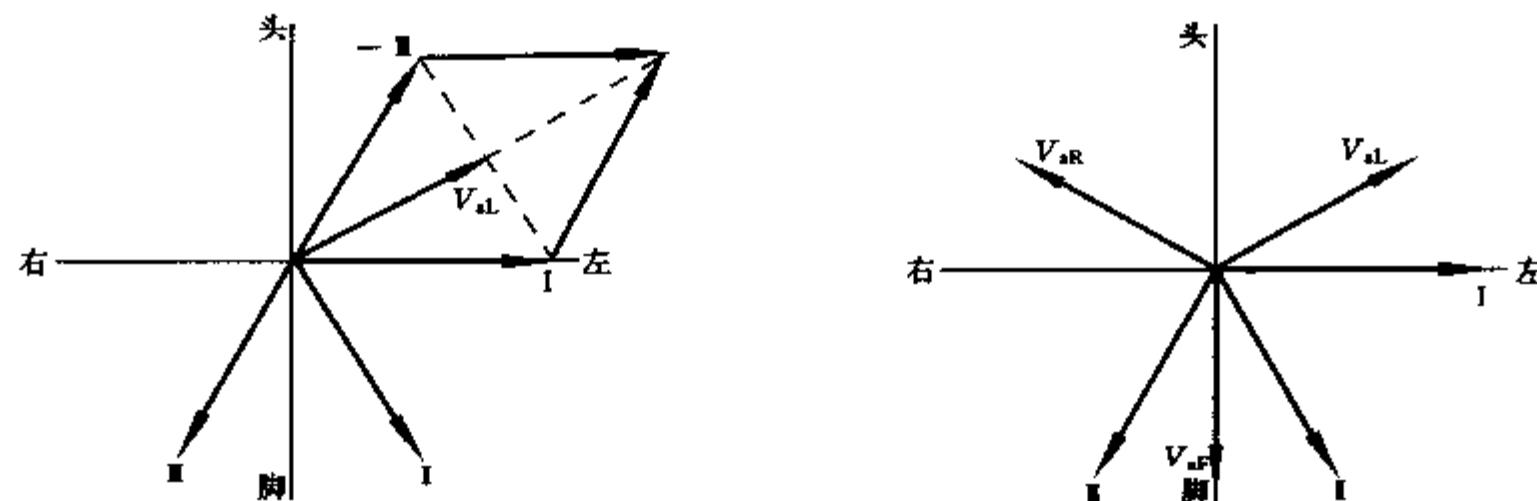
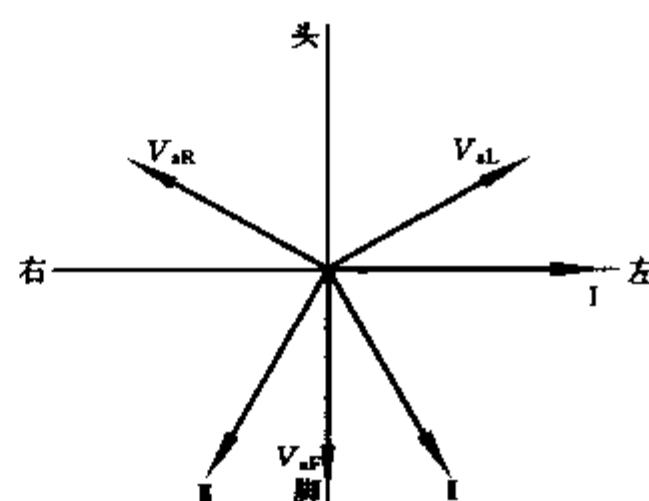


图 2.14 用导联 I 和 II 来表示 V_{aL} 的向量图



2.15 所有前臂面导联之间的矢量关系

2.2 ECG 导联系统

在心电图学中有三种基本导联系统。第一个导联系统具有最普遍的 12 导联，它定义了一组 12 个电位差，用它们来形成标准临床 ECG。第二个导联系统规定了记录 VCG 的电极的位置。第三导联系统为监测系统，典型的只分析一个或两个导联。

2.2.1 12 导联 ECG

图 2.16 所示的是记录 12 导联的标准临床 ECG 的示意图。图 2.17 所示的为普通病人的标准 12 导联 ECG。左边的校验脉冲标定为 1mV，记录速度为 25mm/s。每小格为 1mm，每大格是 5mm。因此对于导联 I，R 波幅值约为 1.1mV，心跳之间的时间间隔几乎为 1s（即心跳速率约 60 次/min）。

仪表放大器是专为心电图记录设计的，就像图 2.13 所示的一样。在现代基于微机的 ECG 机中，有 8 个一样的 ECG 放大器同时记录导联的电压 V_1 、 V_2 和 $V_1 \sim V_6$ ，然后在最后报告中计算导联 V_{aL} 、 V_{aR} 、 V_{af} 和 V_{af} 。

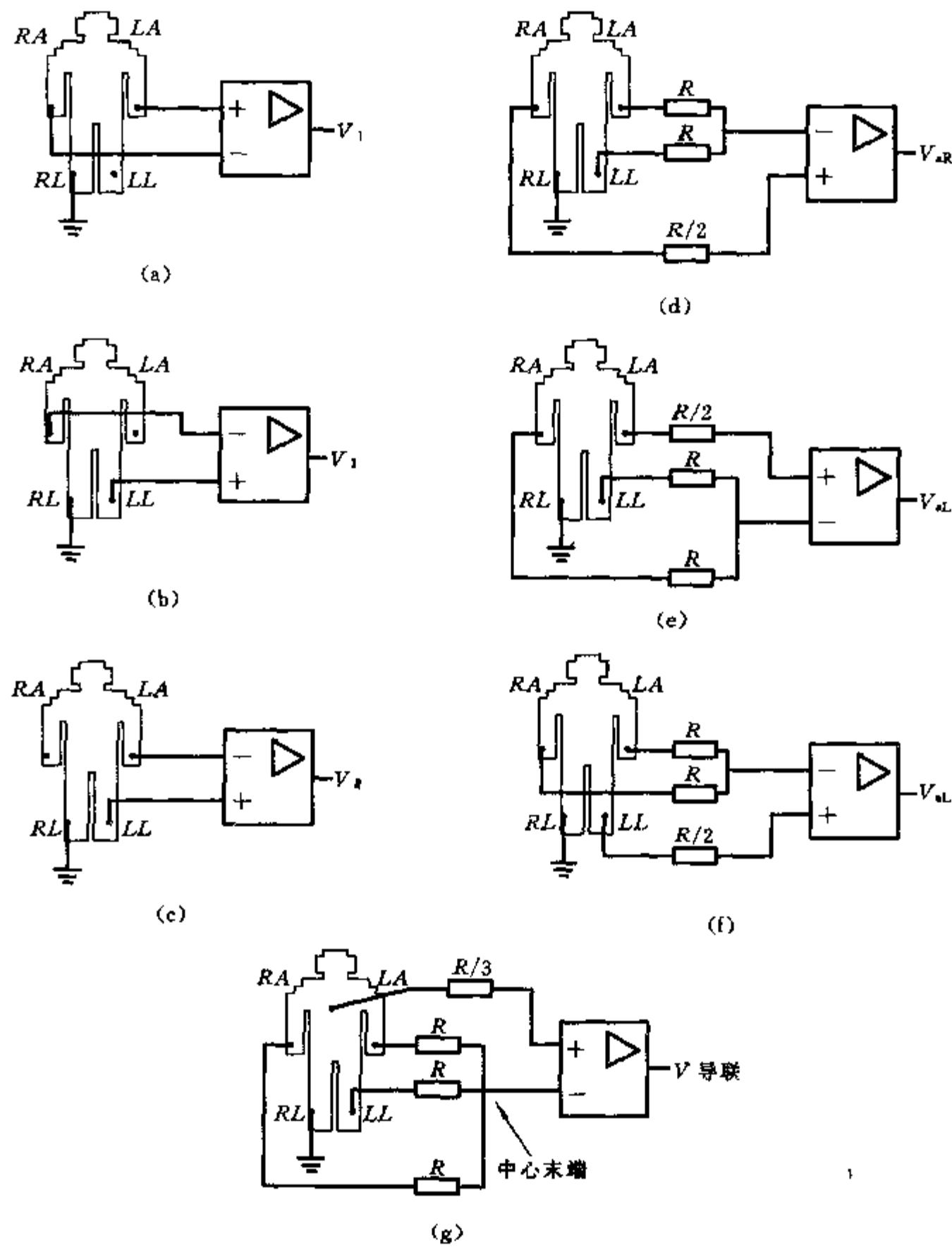


图 2.16 标准临床 12 导联 ECG
 (a) 导联 V_1 ; (b) 导联 V_1 ; (c) 导联 V_1 ; (d) V_{aR} ; (e) V_{aL} ;
 (f) V_F ; (g) 以 wilson 中心端为参考的六个 V 导联

2.2.2 向量心电图

图 2.18 所示的是 FrankVCG 导联系统的电极配置。在国际上这是最通用的 VCG 导联系统。图 2.19 所示的是用电阻网络对电压进行线性组合来计算 Frank 导联系统的三个时变正交分量的方法。图 2.20 所示的是普通病人 VCG 的 IBMPC 屏幕显示图。

图 2.20 中,一次心跳过程中三个时变向量导联,如图(a)所示,图(a)中从上到下依次为 x 、 y 、 z 导联;图(b)的上图所示的为一次心跳过程中向量顶点运动的前视图,图(b)的下图所示的为从此人的头顶上看下去的向量环图;图(c)所示的为从病人左边看过去的左向量图。

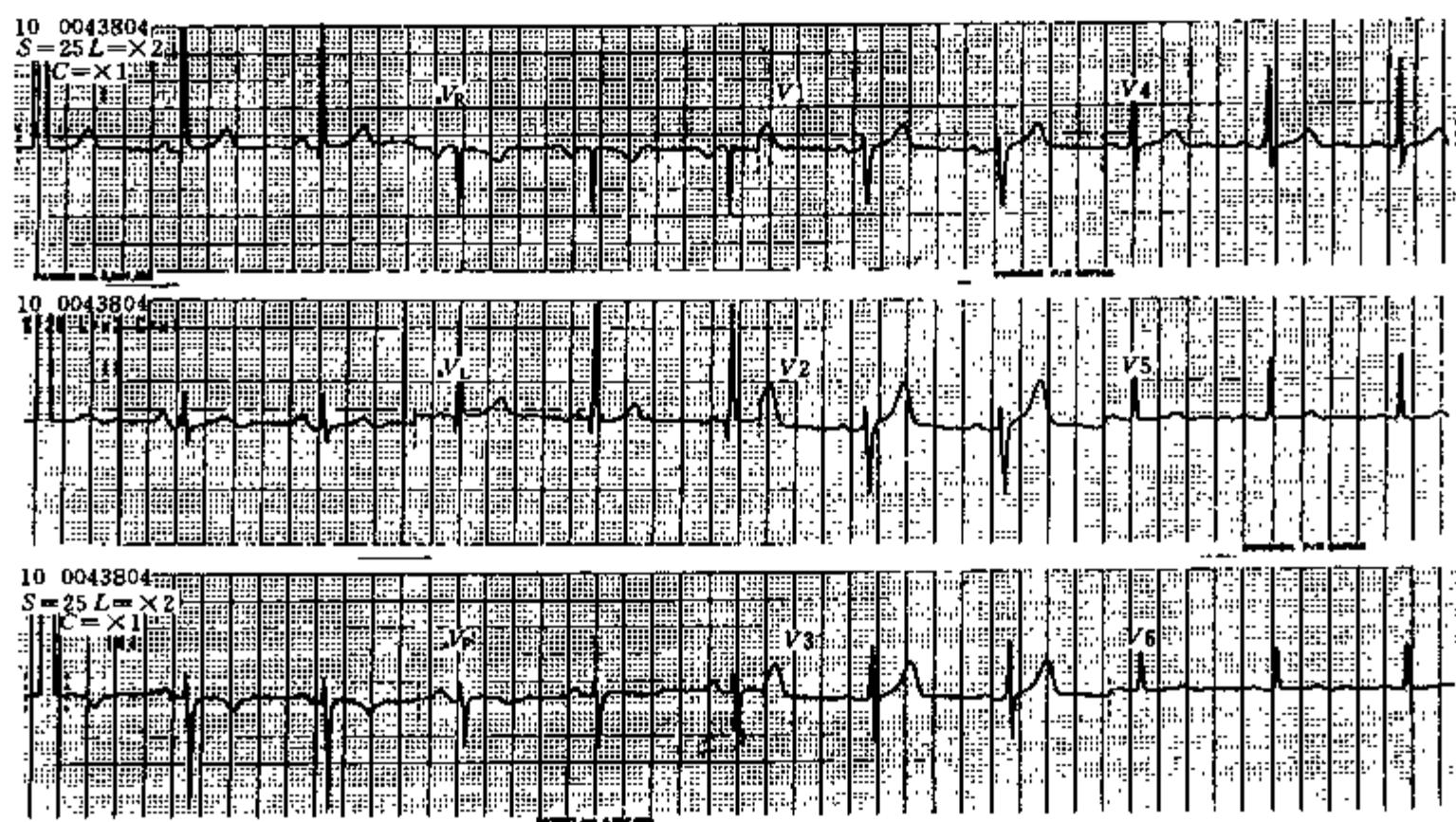


图 2.17 常规男性病人的 12 异联 ECG

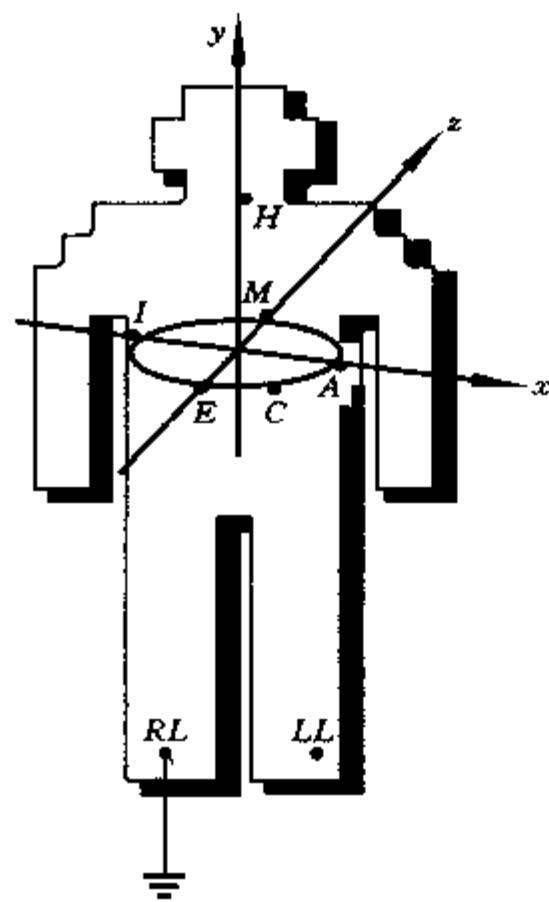


图 2.18 FrankVCG 导联系统的电极配置

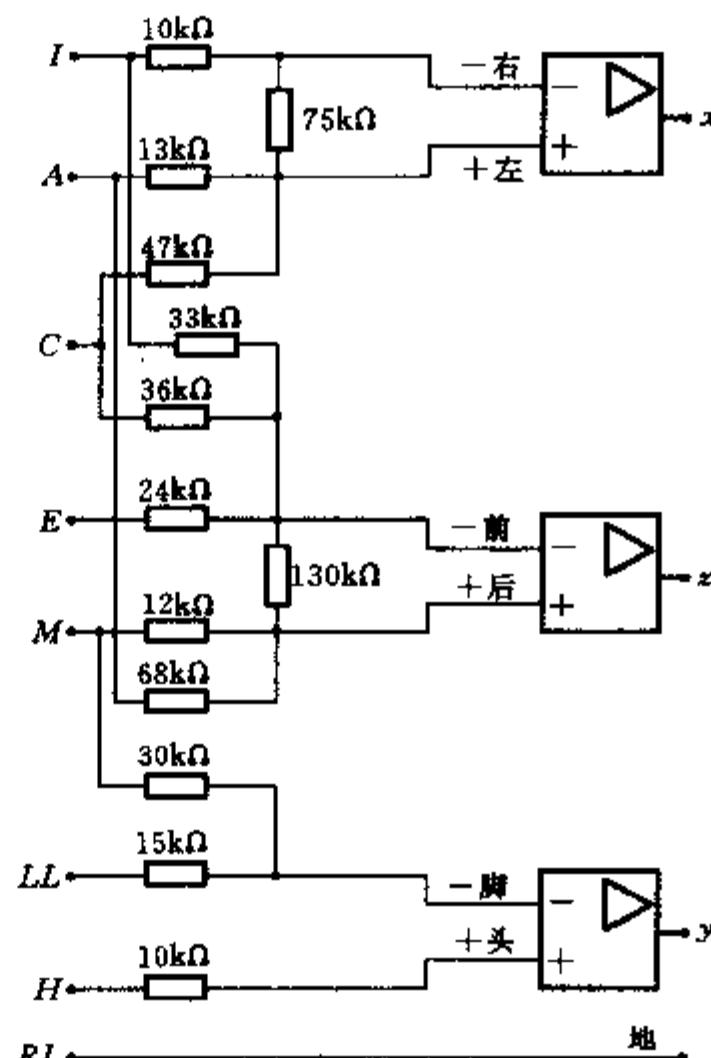


图 2.19 用来连接体表电位的电阻
网络产生 FrankVCG 导联系统的三个时变无向量导联

2.2.3 监测式导联系统

监测式导联系统不用标准电极而使用典型的双导联。因为该系统的主要目标是要可靠地

识别每次心跳并进行节律分析,所以电极的配置应以获得在基本 ECG 中有较大的 R 波为原则。这样可保证心跳探测时有较高信噪比。因为导联 I 对于许多病人来讲其峰值较大,所以此导联往往为多家生产厂家推荐作为第一选择。不同电极位置的第二导联一般作为备份,以防止导联脱落等问题。

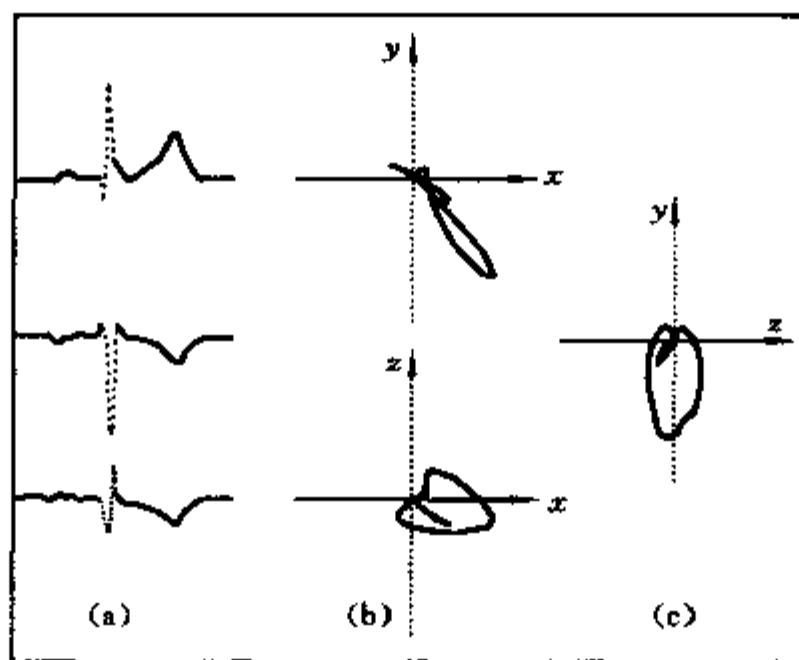


图 2.20 常规男性的向量心电图

2.3 ECG 信号特点

图 2.21 所示的为不同应用的三种心电图带宽,用于记录标准临床 12 导联 ECG 的带宽是 0.05~100Hz。用于监护的,比如强化处理病人或急诊病人时,其带宽限制在 0.5~50Hz 范围

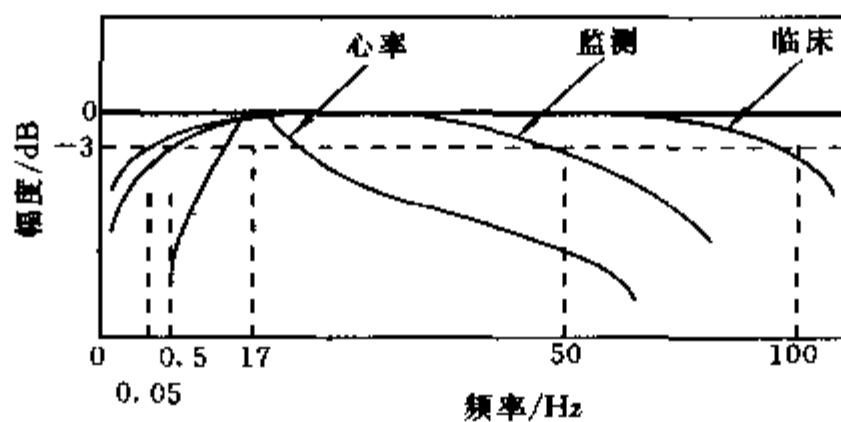


图 2.21 三种不同应用心电图的带宽

内。在这种情况下,对于心律紊乱(心律失常)波形,只要求大致地了解,而不需要了解波形的细节。第三种带宽用于心率测试(心率计),为了探测 QRS 复波,要求信噪比最大。这种滤波器可让 QRS 复波通过而抑制非 QRS 复波,以及抑制 P 波和 T 波的噪声。这样的滤波器有利于探测 QRS 复波,但是会导致 ECG 失真,以至无临床价值。另外一种应用没有在图中标出,其带宽达 500Hz,可用于测量晚电位。这是 ECG 中接在 QRS 复波之后的一些小的高频波。

ECG 信号的峰值大约为 1mV,所以为了达到 1V 的峰值,典型的 ECG 放大器增益约为 1000。

2.4 实验: 模拟滤波器、ECG 放大器及 QRS 检测器

通过本实验, 可学到四种模拟滤波器: 低通、高通、带通和带阻滤波器的特性。可将这些滤波器用于 ECG 放大器中, 然后将学会如何在 QRS 检测电路中应用这些滤波器。QRS 检测电路在每次 QRS 复波出现时都将产生一个脉冲。注意必须亲手制作所有电路。

2.4.1 设备

实验所需要的设备主要有如下几种。

- (1) 双踪示波器。
- (2) 信号发生器。
- (3) ECG 电极。
- (4) 记录仪。
- (5) ECG 放大器和 QRS 探测电路板。
- (6) 模拟滤波器板。

2.4.2 背景知识

1. 低通滤波器/积分器

图 2.22(a)所示的是一个低通滤波器电路。其低频增益为:

$$A_L = -\frac{R_2}{R_1} \quad (2.15)$$

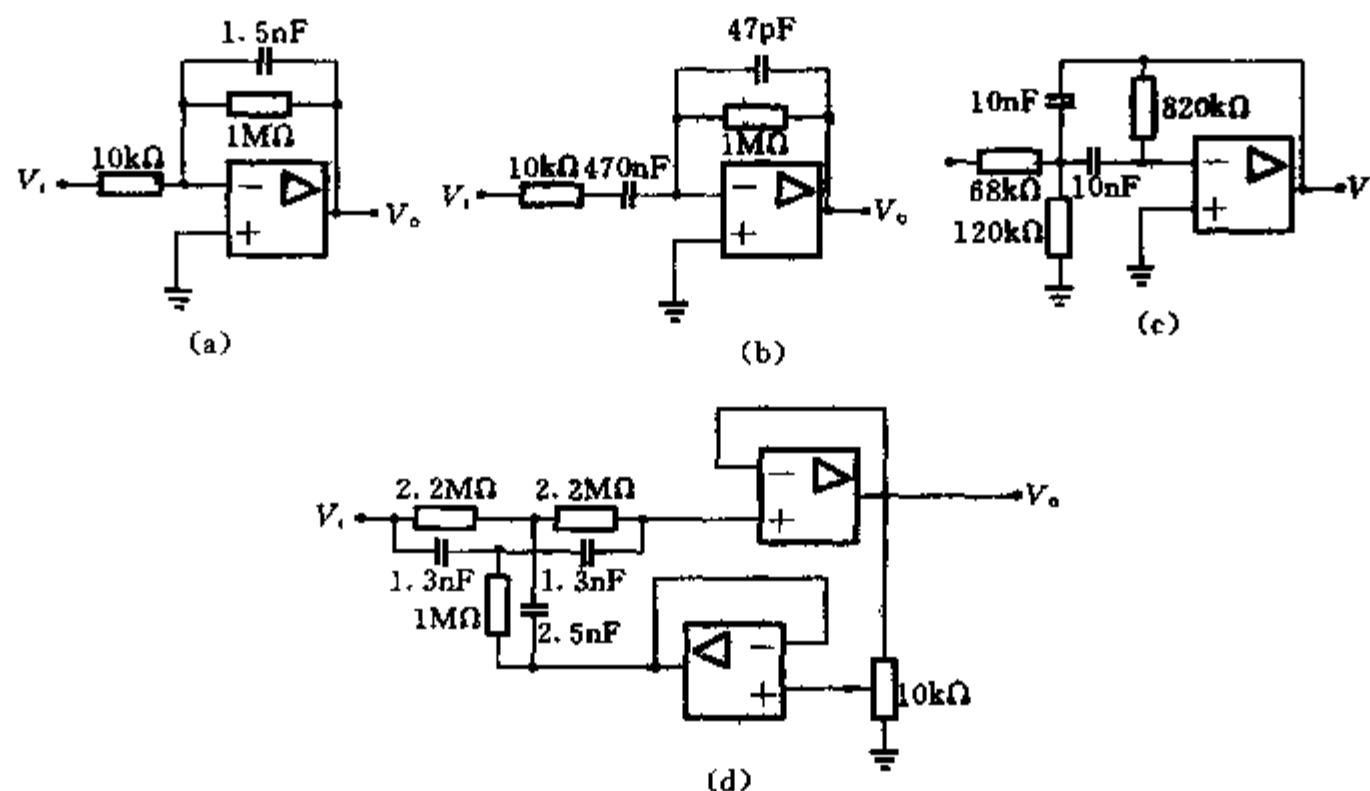


图 2.22 模拟滤波器

(a) 低通滤波器(积分器); (b) 高通滤波器(微分器);
(c) 带通滤波器; (d) 带阻滤波器(陷波器)

因为运算放大器是反向连接的放大器, 所以符号为负。高频段转折频率为:

$$f_h = \frac{1}{2\pi R_2 C_1} \quad (2.16)$$

对于高频而言,低通滤波器的作用就像一个积分器。

积分器的输出为:

$$\begin{aligned} V_o &= -\frac{1}{1+j\omega R_1 C_1} V_i \\ &= -\frac{1}{R_1 C_1} \int V_{in} \end{aligned} \quad (2.17)$$

这可以用观测输出对输入的相移来得到证实。

对于正弦信号,输出相移为 90° ,故

$$\int V \sin \omega dt \geq t = -\frac{V}{\omega} \cos \omega t = \frac{v}{\omega} \sin(\omega t + \frac{\pi}{2}) \quad (2.18)$$

因此,积分器的增益在高频段降低。注意如果积分器不含 R_2 ,则直流增益为无穷大,即在直流时运算放大器直流偏置电流对积分电容 C_1 充电并使运算放大器饱和。

2. 高通滤波器/微分器

与低通滤波器在高频段作用相当于积分器相对应,高通滤波器在低频段的作用相当于微分器。如图 2.22(b)所示,可以得到高频增益 A_h 和低端转折频率 f_l 分别为:

$$A_h = -\frac{R_2}{R_1} \quad (2.19)$$

$$f_l = \frac{1}{2\pi R_1 C_1} \quad (2.20)$$

高通滤波器在低频段的微分作用可以用方程来描述,正如积分器的方程一样。差分器增益随频率增加而增加。

3. 带通滤波器

带通滤波器的电路如图 2.22(c)所示。带通滤波器增益在中心频率处最大,而在其两端均减小。带通滤波器带宽被定义为两个转折频率之差。带通滤波器的 Q 值定义为:

$$Q = \frac{\text{中心频率}}{\text{带宽}} \quad (2.21)$$

4. 带阻滤波器/陷波器

电源线噪声是主要干扰源。有时 60Hz 带阻滤波器(陷波器)用于抑制这种干扰。一般这种滤波器只抑制一个特定频率,而让所有其他频率通过。图 2.22(d)所示的是一个带阻滤波器。对于图 2.23 所示的 60Hz 陷波器,60Hz 抑制系数定义为:

$$60\text{Hz 抑制系数} = \frac{100\text{Hz 处滤波器输出电压}}{60\text{Hz 处滤波器输出电压}} \quad (2.22)$$

注意公式(2.22)中的值是指输入电压值相同时的值。

5. ECG 放大器

在振幅上,ECG 信号通常约为 1 mv,其频率成分为 0.05~100Hz。为了处理 ECG 信号,必须将它放大。图 2.23 给出了 ECG 放大电路。ECG 放大器是高放大倍数(约 1000),频率响应为 0.05~100Hz,高输入阻抗,以及低输出阻抗。其增益和频率响应方程式的推导作为读者的练习。

6. QRS 检测器

图 2.24 和图 2.25 所示的分别是 QRS 探测器的框图和电路图。QRS 探测器包含以下五

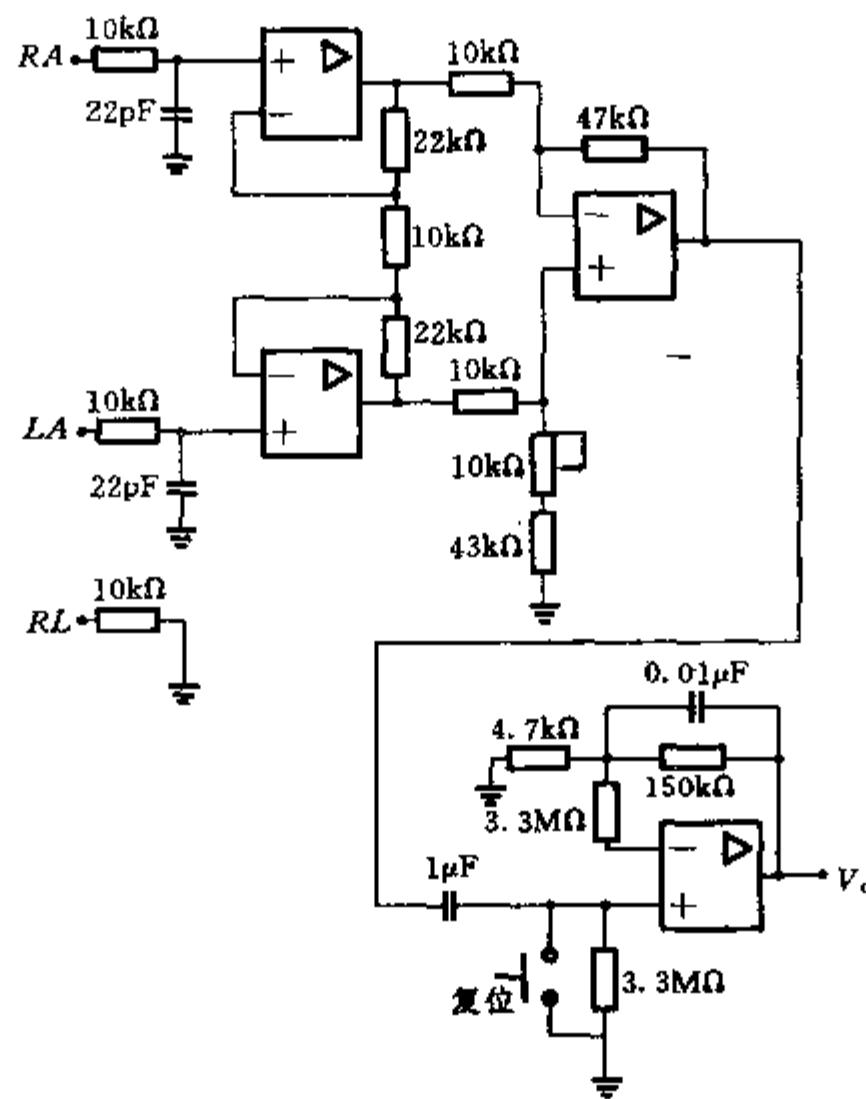


图 2.23 ECG 放大器的电路图

部分：

(1) QRS 滤波器。一般的 ECG 信号的功率谱在 17Hz 处信噪比最高。为了探测 QRS 复波，ECG 要通过一个中心频率为 17Hz、带宽为 6Hz 的带通滤波器。这个滤波器的输出有大量环路振荡。

(2) 半波整流器。滤波后的 QRS 复波是经过半波修正的，所以相应的电压可用探测器电路产生的门限电压进行比较。

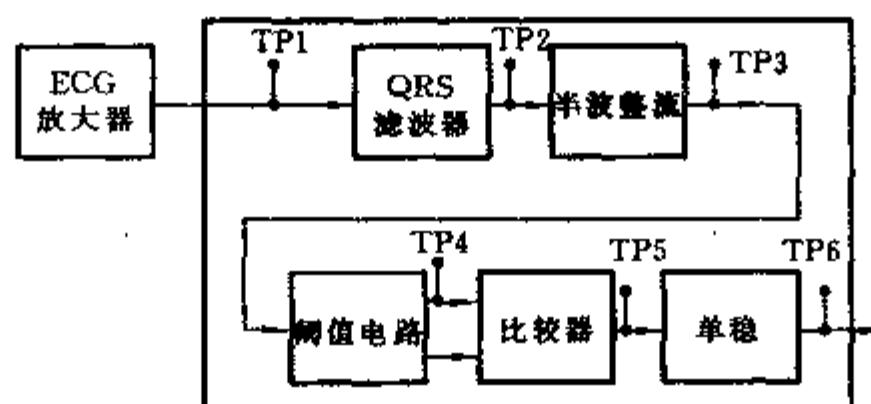


图 2.24 QRS 检测器框图

(3) 门限电压电路。滤波与整流过的峰值电压存储在一个电容中。这个电压的一部分(门限电压)用于与经滤波并校正后的 ECG 输出进行比较。

(4) 比较器。当超过门限电压时比较器可探测到一个 QRS 脉冲。每来一次脉冲，电容就将重新充电到一个新的门限电压。因此每次脉冲后都将产生一个根据过去值产生的新的门限电

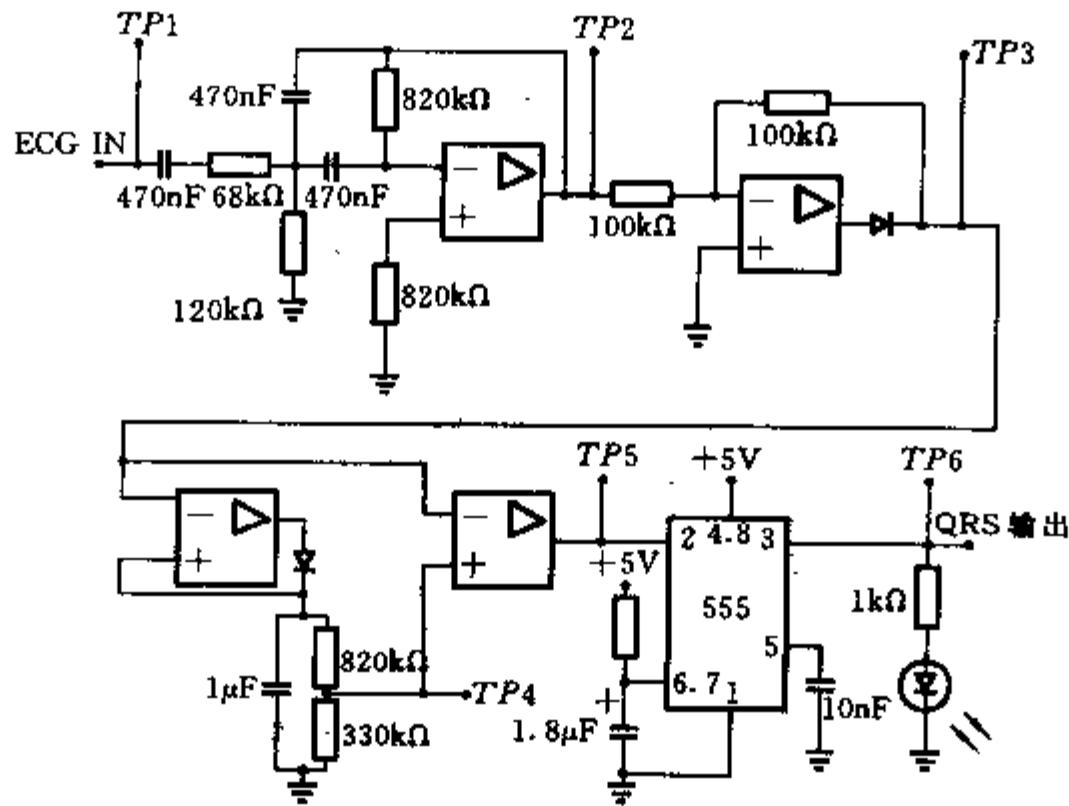


图 2.25 QRS 检测器电路

压。

(5)单稳态电路。单稳态电路每探测到一个 QRS 复波，就会产生一个 200ms 的脉冲。这个脉冲用于驱动 LED。

某些病人安装有心脏起搏器，因为起搏器会导致尖的 QRS 脉冲，所以电路往往包含抑制起搏器送来信号的部分。这种抑制通过限制放大器的速率来实现。

2.4.3 实验步骤

采用集成四个运算放大器的 LM324 可构成上述的所有电路。LM324 电路图如图 2.26 所示。

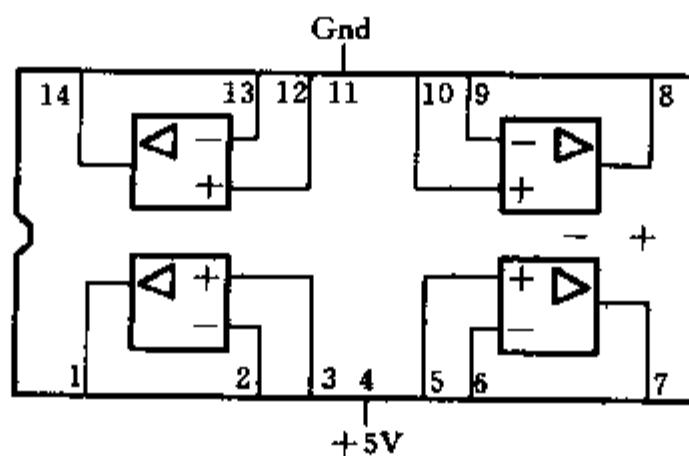


图 2.26 LM324 运算放大器集成电路引脚图

1. 低通滤波器

(1)打开滤波器板上的电源。将信号发生器产生的 10Hz 的正弦波信号以尽可能小的幅值送到积分器的输入端，同时用示波器观察输入和输出。计算增益。

(2)从 10Hz 开始，以 10Hz 为单位逐渐增加频率，直至 200Hz 为止，记录每个频率点的输出。用这些值画出幅-频特性图。然后找到输出值为 10Hz 处输出值的 0.707 倍的频率点。记录这个频率值。

(3)通过观察输入与输出的相移来验证低通滤波器在高频段的积分作用。记录在高端转折频率处的相移。

2. 高通滤波器

(1)将信号发生器产生的最小幅值的 200Hz 正弦信号送到差分放大器输入端, 同时用示波器观察输入和输出。计算增益。

(2)从 200Hz 频率点开始, 以每 20Hz 为单位逐渐减小频率, 直至接近主频率为止, 记录每个频率点的输出, 然后找出在什么频率处的幅值为 200Hz 处幅值的 0.707 倍。这是低端频率的 3dB 点。记下这个值。

(3)通过观察输入、输出的相移, 验证高通滤波器在低频段的差分效果。另一种简单的验证方法, 是在输入端加 10Hz 方波信号, 观察输出端的尖峰信号。

3. 带通滤波器

对于 1V(P-P) 的正弦信号, 从 10Hz 到 150Hz 变化其频率。记录高、低转折频率。找到这个滤波器的中心频率和带宽。

4. 带阻滤波器/陷波器

给本滤波器施加 1V(P-P)、60Hz 正弦波信号, 测量输出电压。对于 100Hz 正弦信号重复此过程。记录结果。

5. ECG 放大器

(1)将放大器的 LA 输入端和 RA 输入端接地, 观察输出。调整 $100k\Omega$ 电位器以调整失调电压。

(2)将 LA 输入和 RA 输入连到信号高端。RL 输入接信号高端(60Hz), RL 输出接信号低端。这是共模工作方式。计算共模增益。

(3)将 LA 输入接信号高端(30Hz), RL 输入接信号低端(通过衰减器以避免饱和)。这是差动工作方式。计算差模增益。

(4)求放大器的频率响应。

(5)将三个电极连到你的身体上。将这些电极接到放大器输入端。观察放大器输出。如果噪声很大, 则可将电极缠绕在一起。当得到清晰的信号时, 用记录仪进行记录。

6. QRS 检测器

(1)使用三个 ECG 电极。将电极连到 ECG 放大器板的输入端。开电源, 用示波器观察 ECG 放大器的输出。如果有额外的噪声, 则按紧电极可消除噪声。

(2)将 ECG 放大器的输出端连到 QRS 探测器板的输入端。用示波器观察下面的信号并用纸带记录仪记录。注意 ECG(TP1)用一个通道。其他每个测试信号(TP1~TP6)占用另一个通道。走纸速度应合适(一般为 25mm/s)。

所要观察的信号为:

测试点	信号
TP1	你的 ECG
TP2	滤波后的输出
TP3	校正过的输出
TP4	比较器输入
TP5	比较器输出

当每次探测到 QRS 时,LED 应该闪烁。

2.4.4 实验报告

(1)用本文介绍的方程式,求低通滤波器的 A_L 和 f_L 。将这些值与实验结果所得的值进行比较,分析其差异。

(2)画出滤波器幅-频特性图。标明本图的-3dB 点。

(3)对于低通滤波器,你得到了什么样的相移值?

(4)用本文介绍的方程式,求高通滤波器的 A_H 和 f_H 。将这些值与实验结果所得的值进行比较,分析其差异。

(5)画出滤波器幅-频特性图。标明本图的-3dB 点。

(6)对于高通滤波器,你得到了什么样的相移值?

(7)对于带通滤波器,列出你所得到的以下值:

- ① 中心频宽。
- ② 通带增益。
- ③ 带宽。
- ④ Q 。

写出计算过程。

(8)对于你所用的带阻滤波器,60Hz 抑制系数是多少?

(9)你的 ECG 放大器的高、低-3dB 点频率各是多少? 如何与理论值进行比较?

(10)你的 ECG 放大器增益多少? 如何与理论值进行比较?

(11)你的 ECG 放大器的 CMRR 是多少?

(12)如何改变本放大器的-3dB 频率?

(13)对记录仪记录的波形进行解释。这些是你所希望得到的吗?

(14)本实验用的 QRS 检测器能用于任何其他人的 ECG 吗? 评价你的答案。

(15)将记录纸与实验报告贴在一起,写出适当的计算过程。

复习思考题

2.1 什么是心电等效发生器? 它与实际心电活动有什么差别? 给出两个例子。

2.2 什么是矢量心电图,它是如何记录的?

2.3 心脏矢量在某时刻的方位如图 2.27 所示。此时,哪一个前向导联(I、II 和 III)是正向活动的?

2.4 某个基于微机的 ECG 机只对导联 I 和 II 进行采样存储。什么样的标准导联可由这两个计算得到?

2.5 显然 ECG 的所有六个前向导联可用其中任意两个来表示。用导联 I 和 II 来表示增加的右臂(即 V_{aR})导联。

2.6 用 V_{RF} 和 V_{RL} 表示导联 III。

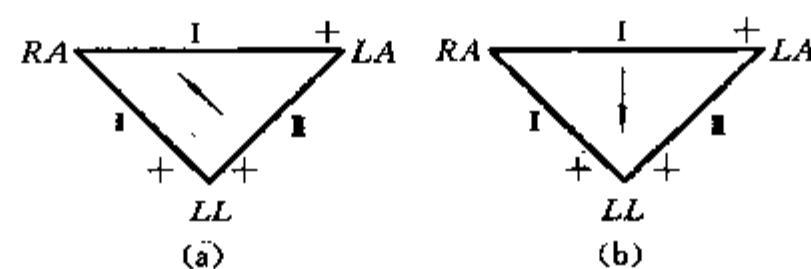


图 2.27 心脏矢量

- 2.7 可以用其他两个导联来表示 V_6 吗？是否有其他方法用更多的导联计算 V_6 ？
- 2.8 在 ECG 中用了四种不同的带宽。描述一下各种带宽的主要的应用。
- 2.9 以下所用的标准 3dB 带宽的频率范围是什么？①临床 ECG，②ECG 监护应用，如强制监护，③为什么临床与监护的带宽不一样？
- 2.10 某位心电专家在机器上记录了病人的 ECG，但对该 ECG 怀疑有错。她注意到病人 ECG 的 QRS 复合波的峰-峰值比记录仪记录的要小。请分析一下，是仪表的什么带宽会导致这样的结果？
- 2.11 某位心电专家注意到一般病人的 ECG 的 T 波扭曲，以致看起来像一个单相正弦波形而不是双极波形。分析一下，是仪表的什么带宽会导致这个结果？
- 2.12 记录正在救护的病人的 ECG 时所用的最好电极是什么？

第三章 信号转换

倘若模拟的生物医学信号不能被精确地捕获，并转换成数字形式，则计算机将无法有效地分析和直观地描述生物医学信号的功能，信号不能得到有效地利用。本章将讨论采样的基本原理和典型信号转换系统所需的基本硬件。其中 3.1 节从理论上讨论了采样的基础知识；3.2 节介绍了几种实现实际信号转换所需的电路。这里分析了一个 ECG 信号转换系统的误差，还复习了数/模转换器(DAC)和模/数转换器(ADC)，以及其他相关电路，包括放大器、采样/保持电路以及模拟多路转换器。

3.1 采样技术基础

把一个连续时间信号通过微处理器转换成可用的离散形式，这种思想是基于这样一个事实：可以用在定期点上及时地采集的瞬时幅值代表连续时间信号，更重要的是，可以通过这些采样点来很好地重建原始信号。这种思想也运用于电影中，即单个画面是连续变化场景的快照，当这些单个画面以足够快的速度回放时，就可得到原始场景的精确描述。

3.1.1 采样定理

采样定理最早是由香农(Shannon)提出的：在一定条件下，采样要保证原始信号可以从采集的信号无失真地重建起来，即对于有限带宽的连续信号，当其频率成分不含高于 f_c 的分量时，若以不低于 $2f_c$ 的频率对其采样，则原始信号就可以完全无失真地恢复。等于最高频率 f_c 的 2 倍的采样频率 f_s 称为奈奎斯特频率。

3.1.2 混淆、重叠现象及其他实际问题

为了进一步讨论采样机制，应在频域中分析一下信号频谱。如图 3.1(c)所示，如果采样频率大于 $2f_c$ ，那么只要在采样器输出端接上一个低通滤波器就可以恢复原始信号。如果采样速率太低，就会产生图 3.1(e)所示的情形。重叠区域的信号互相混合而不能再进行恢复的现象，即较高频率反射入较低频率范围，称为混淆。

如果预先知道信号的最高频率成分，则理论上就可以把采样速率定在 2 倍最高频率以上。然而，实际的信号总耦合有噪声，噪声经常包含有比信号本身更高的频率成分，例如，如果心电极放置在肌肉上，就会同时检测到不期望的肌肉信号。但这个问题通常容易解决，即在放大器输入端加上低通滤波器就可除掉不需要的频率分量。

然而，不理想的滤波器也会妨碍我们很好地重建原始信号。如果输入滤波器不理想，就会如图 3.2(a)所示的那样。实际情况往往如此，高频成分仍能通过滤波器，混淆现象仍会发生。

平时，输出滤波器的效果往往被忽视，从图 3.2 就可以看到，如果输出滤波器不理想，重建的信号就可能不正确。特别是，输出滤波器的截止频率必须大于 f_c ，小于 $f_s - f_c$ ，以便不会把不期望的下个周期的频率成分引进来。

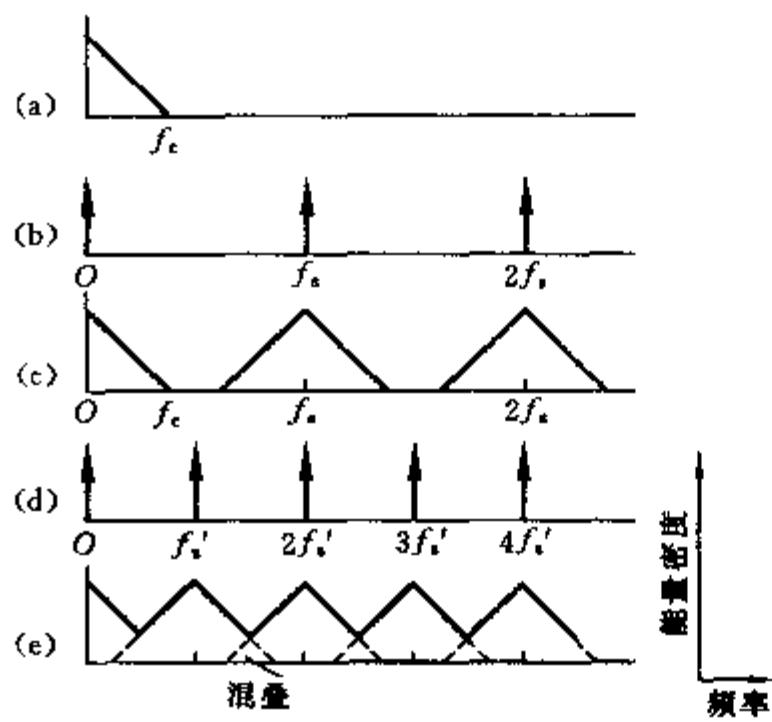


图 3.1 时域采样对频域的影响

- (a) 原始信号的频谱;
- (b) 采样函数的频谱;
- (c) 当 $f_s > 2f_c$ 时采样信号频谱;
- (d) 当 $f_s < 2f_c$ 时采样函数频谱;
- (e) $f_s < 2f_c$ 时采样信号频谱

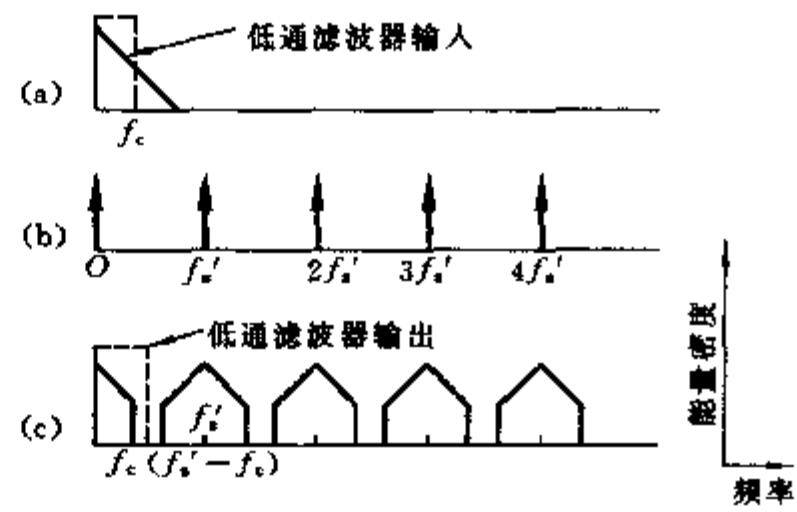


图 3.2 输入和输出端滤波器作用

- (a) 输入端低通滤波器可消除高频信号;
- (b) 采样函数频谱;
- (c) 不提供折叠, 输出端低通滤波器恢复原始信号

另外, 即使信号的最高频率已确实知道, 但实际上仍可能会出现没有把采样频率设计在奈奎斯特频率上的情况。很多生物医学信号处在低频范围内, 采样频率过高, 硬件花费就多, 也需要更大的内存空间。因而, 常允许有一些可接受的误差, 以换取在技术上采用更为实用的采样频率。

3.1.3 生物医学信号举例

图 3.3 所示的是一些人体生理信号的幅值和带宽。

脑电图
频率范围 DC~100Hz(0.5~60Hz)
信号范围: 15~100mV
肌电图(EMG)
频率范围: 10~200Hz
信号范围: 是肌肉活动和电极放置位置的函数
心电图(ECG)
频率范围: 0.05~100Hz
信号范围: 10μV(婴儿), 5mV(成年人)
心率
频率范围: 45~200 次/min
血压
频率范围: DC~200Hz(DC~60Hz)
信号范围: 40~300mmHg(动脉); 0~15mmHg(静脉)
呼吸率
频率范围: 12~40 次/min

图 3.3 生物医学信号、范围及主要诊断范围

3.2 简单信号转换系统

生物医学信号有各种形状和大小。然而，提取和分析这些信号的一般处理过程与所有的信号却是相同的。图 3.4 所示的是一个一般的 ADC(A/D)信号转换系统。

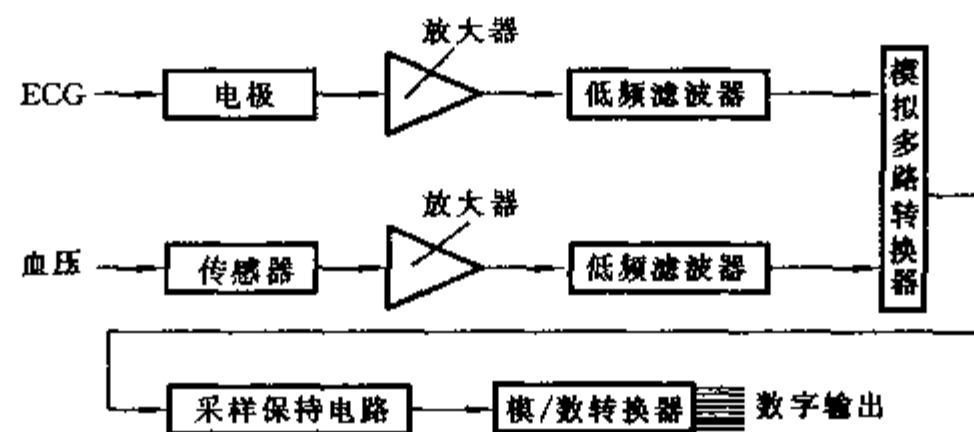


图 3.4 一个典型的 ADC 转换系统
包括传感器、放大器多路传输器、低通滤波器、
采样/保持电路和 ADC

首先是信号的提取，如果其本身为电信号，则可用一个简单的电极把信号从人体传到信号转换系统中。对于其他信号，则需要有一个传感器把生物医学信号转换成电压信号。从电极或传感器得到的信号通常幅值都很小（例如 EEG 信号在 $10\mu\text{V} \sim 5\text{mV}$ 范围）。因而必须通过放大，把信号幅度变大到 ADC 能接收的范围内。对于信号的放大，必须在尽可能靠近信号源的地方进行，以防止信号的衰减，如果有几路输入信号要转换，就需要一个多通道传输器作为各个信号到 ADC 的通道。为了减小混淆，在采样之前常常要用一个低通滤波器对信号限带。在 ADC 输入前，还要有一个采样/保持电路（变化非常慢的信号除外），以便在转换过程中模拟信号能保持常值。最后，ADC 把所存储的模拟电压信号通过采样/保持电路转换成数字形式。

既然已得到生物医学信号的数字形式，那么它可以作何用呢？通常数字信息存储在存储设备中，以便计算机对其进行后期处理。后面各章将详细讨论通用的生物医学信号处理中各种数字信号处理的算法。有些情况下处理能够实时进行。数字信号另一个可能的用途是用于控制系统。在这种情况下，信号由计算机处理后再反馈到待控制的设备。通常控制者需要一个模拟信号，因而就需要有 DAC。图 3.5 所示的是一般的 DAC 系统。该模拟系统可用于控制麻醉机的气体流量或孵卵器内的温度。

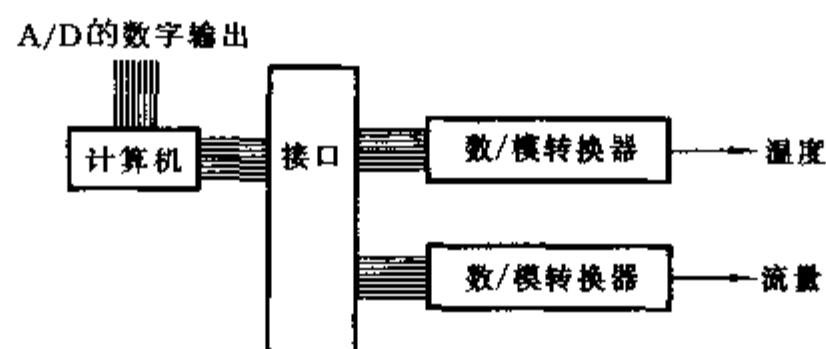


图 3.5 利用计算机对转换前的数字信号进行处理
的典型 DAC 系统

3.3 生物医学信号的转换要求

正如 3.1.3 小节所述,生物医学信号有各种特性。任何转换系统的最终目的都是把生物医学信号转换成数字形式,并尽可能地少丢失信息。转换系统的规格依赖于信号的特性和运用。从 3.1.3 小节可知,生物医学信号都是典型的低频率、低幅值信号。设计一个转换系统时一般都必须考虑下列参数:精度、采样速率、增益、处理速度、功耗及尺寸大小。

3.4 信号转换电路

连续模拟信号的数字形式在时间(由采样速率决定)和幅值(由采样数据的位数决定)上都是离散的。有多种电路结构可用来将信号在模拟与数字域之间转换。其中有一些将在本章中讨论。每一种方法都有其优点和缺点。这里讨论的仅限于那些通用于生物医学信号转换的技术。首先将讨论 DAC 转换,因为它常常可形成 ADC 的一部分。

3.4.1 转换器特性

在详细阐述转换器的硬件之前,了解一些用于表示转换器性能特性的基本术语是很重要的。一个共同的判定 DAC(或 ADC)特性的方法是看其静态和动态特性的好坏,即 Allen 和 Holberg(1987)所提出的那些指标。

1. 静态

为了便于说明,现以 DAC 为例,但是所讨论的静态误差也适用于 ADC。理想的 3 位 DAC 静态特性如图 3.6 所示,所有的数字输入的集合在水平轴上,模拟的输出在垂直轴上。最大的模拟输出信号是参考电压 V_{ref} 满刻度的 Y_8 。每一个单个的数字输入,对应于唯一的模拟输出。图 3.6 所示的任何偏差都是静态转换误差。静态转换误差可划分成积分线性误差、微分线性误差、单调性误差和分辨率误差等。

积分线性误差是转换器输出中以理想的最小值和理想的最大值所画直线之间最大的偏差。积分线性误差常常表示成满刻度的百分数或表示成最小有效位 LSB,积分线性误差可以进一步划分成绝对线性误差、偏移或零误差、满刻度误差、增益误差和单调性误差。绝对线性误差是指零误差和满刻度误差。零误差或偏移误差是指数字输出应是零时的实际输出与零之间的差值。满刻度误差是指当数字是满刻度输出时真实电压和理想电压之间的偏差。增益误差则是当真实输出的斜率不同于理想输出的斜率时存在的误差。图 3.7(a)所示的是偏移误差和增益误差。DAC 中的单调性是指当转换器的数字输入超过满刻度范围时,模拟输出在一个转换步和下一步之间不能减少。换句话说,输出的斜率不能为负。图 3.7(b)所示的是一个非线性转换器的输出。

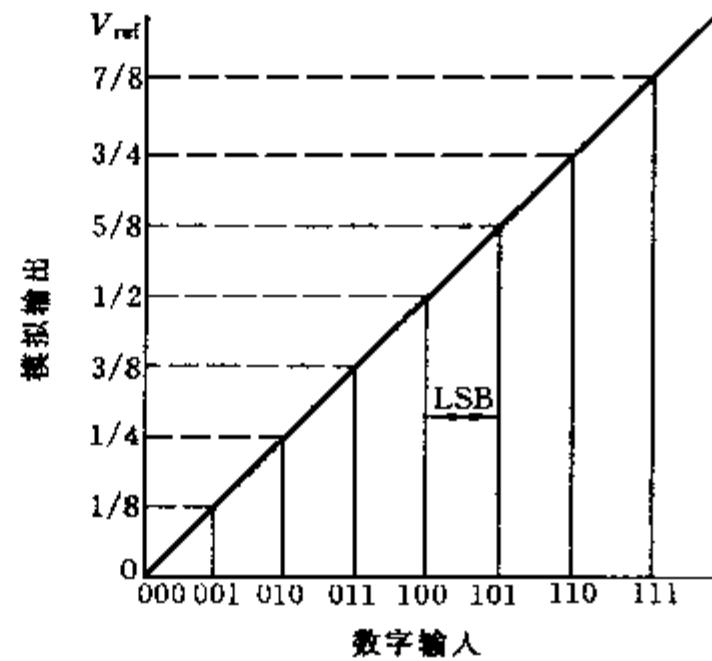


图 3.6 理想的 3 位 DAC 的静态特性, 单个数字输入对应唯一模拟的输出

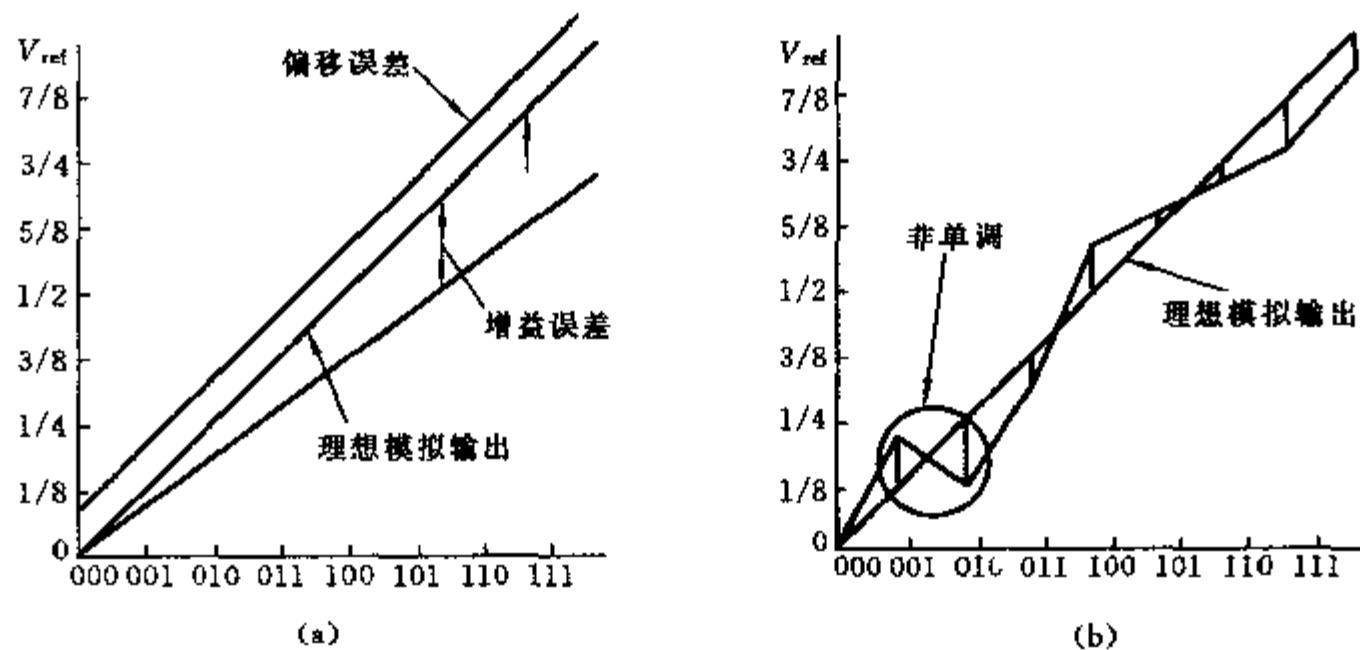
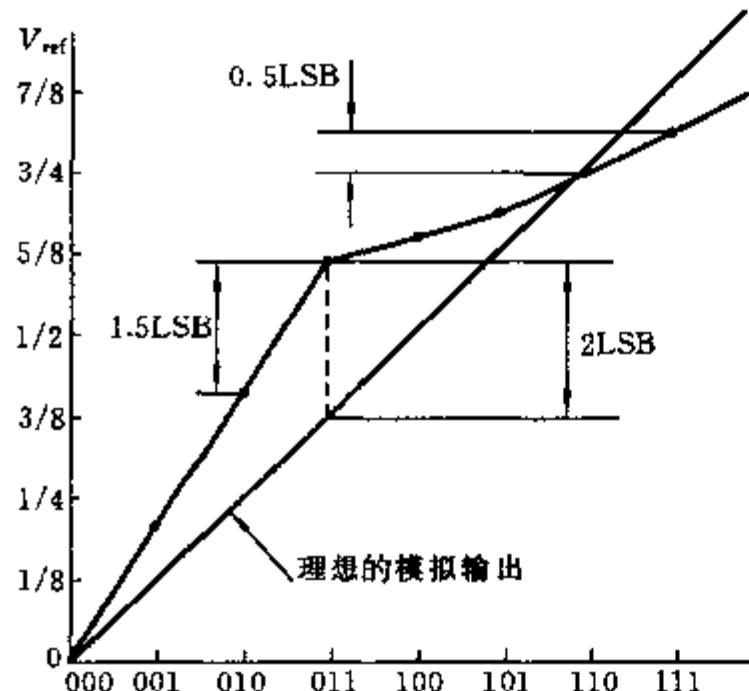


图 3.7 DAC 转换特性

(a) 增益误差和偏移误差; (b) 单调误差

微分线性误差与积分线性误差的区别在于它测量的是邻近水平之间的分隔(Allen 和 Holberg, 1987)。换言之,微分线性误差测量的是理想输出中每 1 个 LSB 上的位-位偏差。图 3.8 所示的是积分线性误差和微分线性误差间的差别。

分辨率定义为产生一个模拟输出电平的最小的输入数字码。理论上, n 位 DAC 的分辨率是 2^n 离散的模拟输出电平。实际上,分辨率误差很少是由噪声和分量漂移引起的。



3.8 土2LSB 积分线性和土0.5LSB
微分线性 DAC

2. 动态

DAC 稳定时间定义为转换器响应一个输入变化所花费的时间。更明确地说,稳定时间就是介于转换器的输入接收到新数字信号的时刻与输出已达到终值的时刻(在一些额定公差中)之间的时间。很多因素都可影响稳定时间,因而如果欲作比较,则必须交待清楚确定稳定时间的条件。需注意的因素包括所作输入变化的量值和输出负载。稳定时间对 DAC 来说很重要,因为它直接关系到转换速度。输出在进一步处理之前必须完全稳定。

3.4.2 DAC

DAC 的功能是将所给的数字输入信号转换成模拟输出信号。DAC 首要的要求就是要有精确的参考电压;其次是参考电压必须标度,以提供对应于可能的数字输入的模拟输出。通常用电压或电荷作标准量;最后,输出可进行内插处理,以提供经光滑处理的模拟输出。

1. 参考电压

一个精确的参考电压对 DAC 的运行是很重要的。模拟输出的大小即源于这个参考电压。共同的参考电压误差要么起因于初值调整,要么是由时间和温度漂移产生。参考电压的获得方法有两种,一种是利用齐纳的二极管的反向击穿电压得到参考电压,另一种是由硅的外推带隙

电压获得参考电压。两种方法都要用到温度补偿。参考电压的标度通常是由无源元件(电压标度用电阻,电荷标度用电容)完成的。

2. 电压标度

参考电压的标度是以串联的电阻接于参考电压和地之间得到的。它可有选择地获得二者之间的离散电压。图 3.9 所示的是一个简单的 3 位 DAC 的电压标度。数字输入经译码后输出相应的电压。电压标度器件结构很规则,它是通过使用一些小量值的电阻对电压标度的。这很适合于积分电路技术。因此当电阻绝对值控制在 50% 时,相对精度可以低至 1% (Allen 和 Holberg, 1987),这就是说,如果在一片硅上制作一套电阻,每个阻值约为 $10\text{k}\Omega$,则实际上每个阻值可达 $15\text{k}\Omega$,它们的误差值都只将是在彼此数值的 1% 以内。

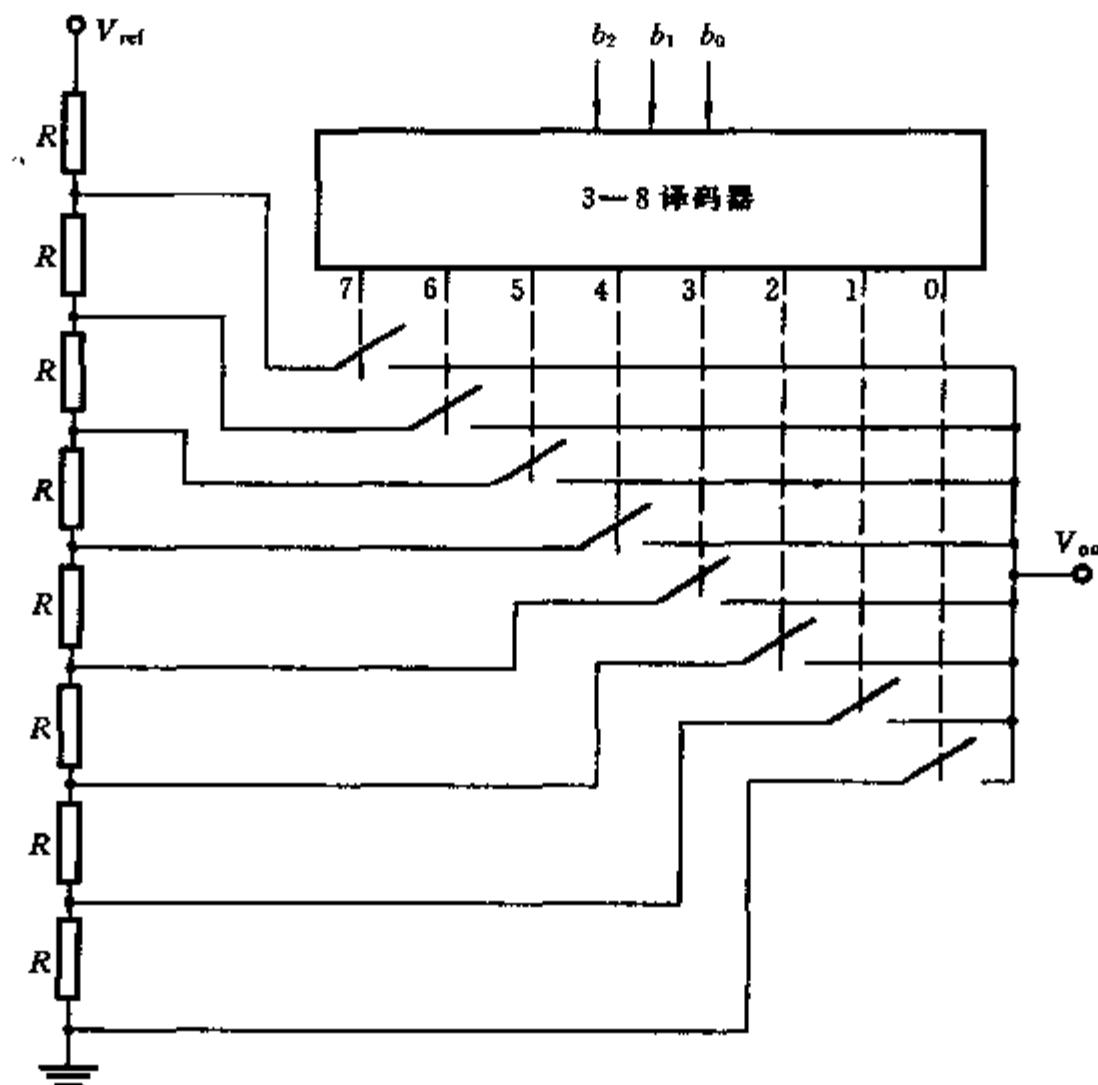


图 3.9 一个简单的电压标度 DAC

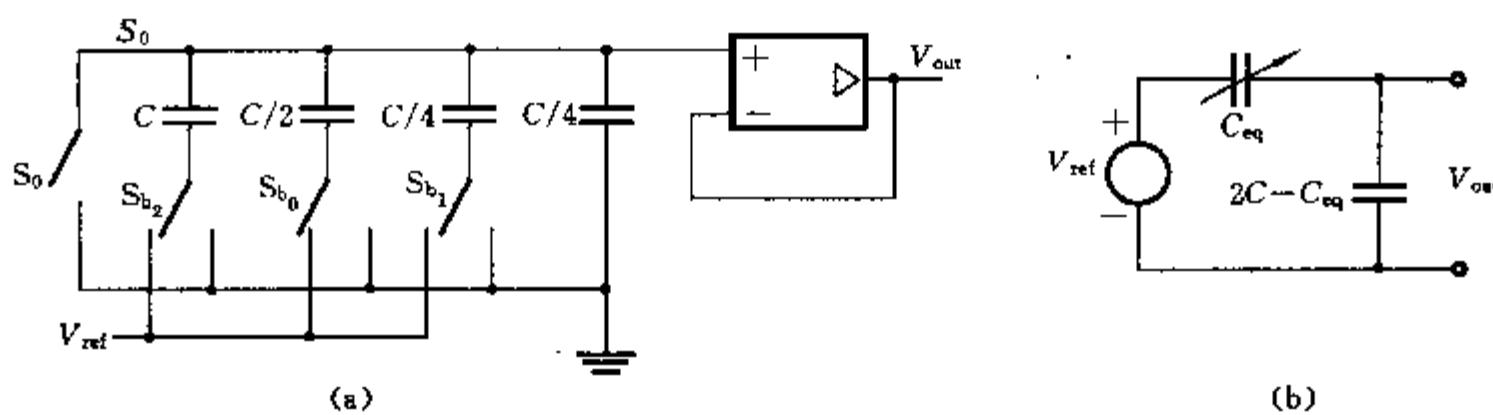
3. 电荷标度

用电荷标度 DAC 时可用电容组对总电荷做二进制分度。图 3.10 所示的是 3 位电荷标度 DAC,其中采用了双相位时钟。在相位 1 期间, S_b 闭合,开关 S_{b_1} 、 S_{b_2} 、 S_{b_3} 也闭合,所有的电容短接地。在相位 2 期间,与表示“1”的字位相关的电容连接到 V_{ref} ,表示“0”的则接地。输出仅在相位 2 期间有效。以下两式解释了这个作用:

$$C_{eq} = b_0 C + b_1 \frac{C}{2} + b_2 \frac{C}{4} \quad (3.1)$$

$$V_{out} = \frac{C_{eq}}{2C} \quad (3.2)$$

注意,不管转换字有多少位,总电容总是 $2C$ 。电荷标度转换器的精度取决于电容比。积分电路电容的误差率通常低于 0.1%。



3.10 3位电荷标度 DAC

(a)二进制数字输入为 101 的电路;(b)任何数字输入等效电路

4. 输出插值法

诸如图 3.9 和图 3.10 所示的简单的 DAC 的输出值都是离散值。为重建模拟信号, 常要使用插值技术。插值方法很易用电子电路实现, 这主要包括下列技术:(1)零阶保持或单点,(2)线性双点,(3)带限或低通。

3.4.3 ADC

ADC 的功能是将所给模拟输入转换成数字输出。正如前面所提到的, ADC 进行转换时, 常常要用到 DAC。通用的另一种方法包括一些积分或斜坡形式, 最后, 高速度的采样, 要用一个并行或快速转换器。很多转换器在输入处需要加一个采样/保持电路, 但因为它们不能转换变换着的输入信号, 而对于变化很慢的信号, 并不一定必须使用采样/保持电路。如果把输入和输出的定义互换, 则 ADC 的误差类似 3.4.1 小节所描述的那样。

ADC 转换器有如下几种类型。

1. 计数式

计数式 ADC 按一次增加一个 LSB 来不断增加计数, 直到它等于模拟输入后, 才能建立内部输出。当内部输出已增加到输入信号电平时, 一个比较器将发出信号停止计数器工作。在这点上计数值即等于数字输出。这种设计的缺点是转换时间随输入信号电平而变。因而对于低幅值信号, 转换时间很快, 但如果信号幅值翻倍, 则转换时间也将翻倍, 而且, 转换的精度依赖于斜坡产生过程中的误差。

2. 跟踪式

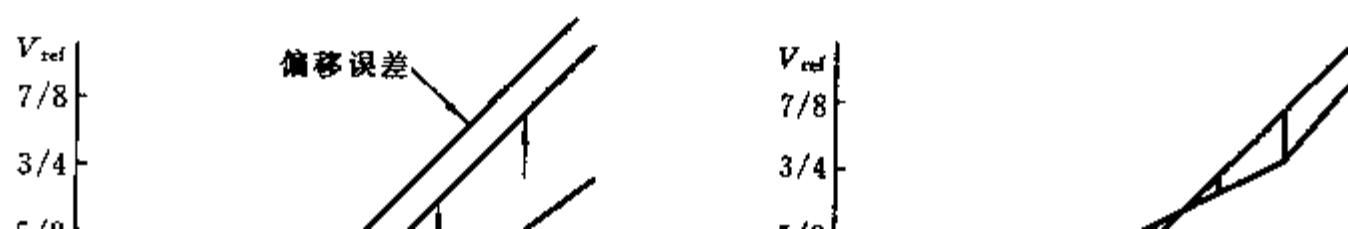
跟踪式 ADC 是计数式 ADC 的一种改进。计数式 ADC 转换器每次转换后要把内部输出复位置零, 而跟踪式 ADC 转换器的内部输出则继续跟踪模拟输入, 图 3.11 所示的是这两种 ADC 的区别。仅从外部看, 跟踪式 ADC 也能用做带数字输出的采样/保持电路, 而且通过设置上下限, 跟踪式 ADC 能用于寻找输入信号在给定时间内所达到的最大值或最小值。

3. 双斜坡式

双斜坡式 ADC, 可将模拟输入在一固定时间间隔内进行积分。这个时间的长度等于内部计数器的最大计数。积分器电容在这个积分时间内所积累的电荷正比于输入电压, 即

$$Q = CU \quad (3.3)$$

积分器输出的斜率正比于模拟输入的幅值。在时间 T_1 之后, 积分器的输入将切换到一个负的参考电压 V_{ref} 上, 这样积分器将以一个常量斜率作反向积分。若一个计数器的积分器达到零值所需的时间为 t_2 , 则积分器电容在时间 T_1 内获得的电荷必须等于 t_2 期间所失去的电荷, 即



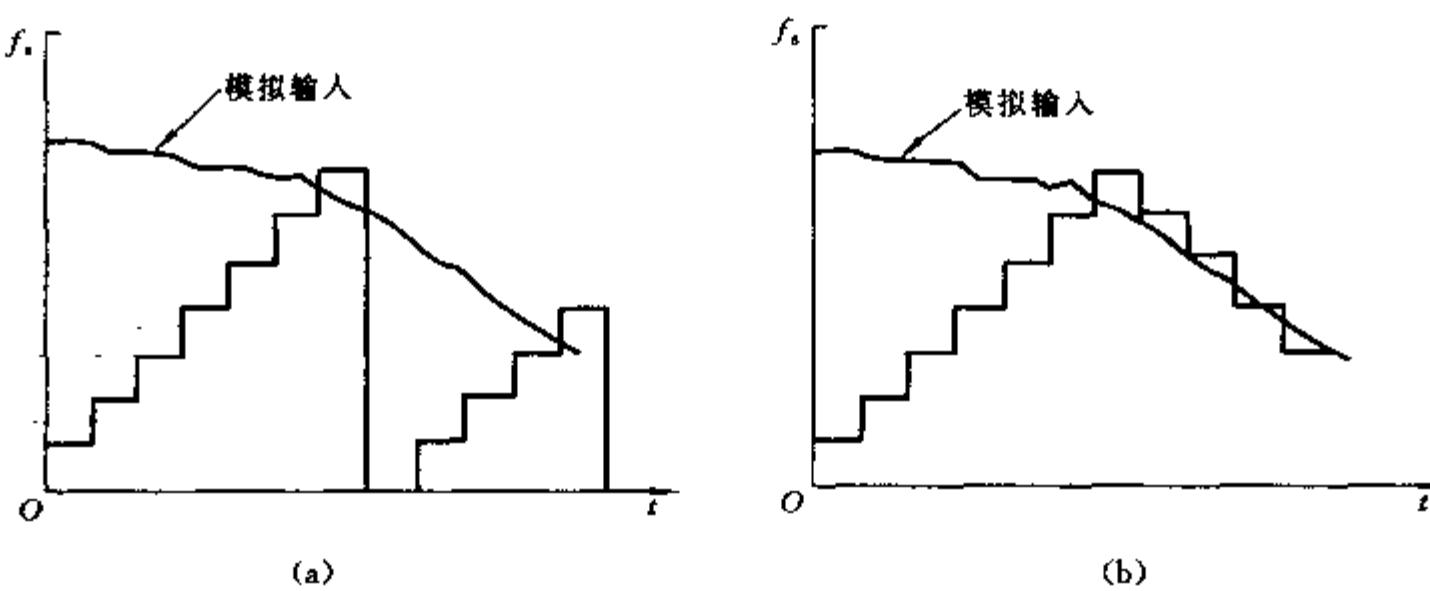


图 3.11 计数式和跟踪式 ADC 的内部输出

(a)计数式 ADC 在每次转换后复位的内部输出;(b)跟踪式 ADC 跟踪式模拟输入的内部输出

$$T_1 V_{in} = t_2 V_{ref} \quad (3.4)$$

注意 t_2 与 T_1 的比值也是计数器值的比率, 即

$$\frac{t_2}{T_1} = \frac{V_{in}}{V_{ref}} = \frac{\text{计数器值}}{\text{固定计数}}$$

因而 t_2 末端的计数等于 ADC 的数字输出。图 3.12 所示的是双斜坡 ADC 的框图和它的相关波形。注意, 双斜坡 ADC 的输出既非积分器斜率的函数, 亦非时钟频率的函数。就结果而言, 这种转换方法是很精确的。

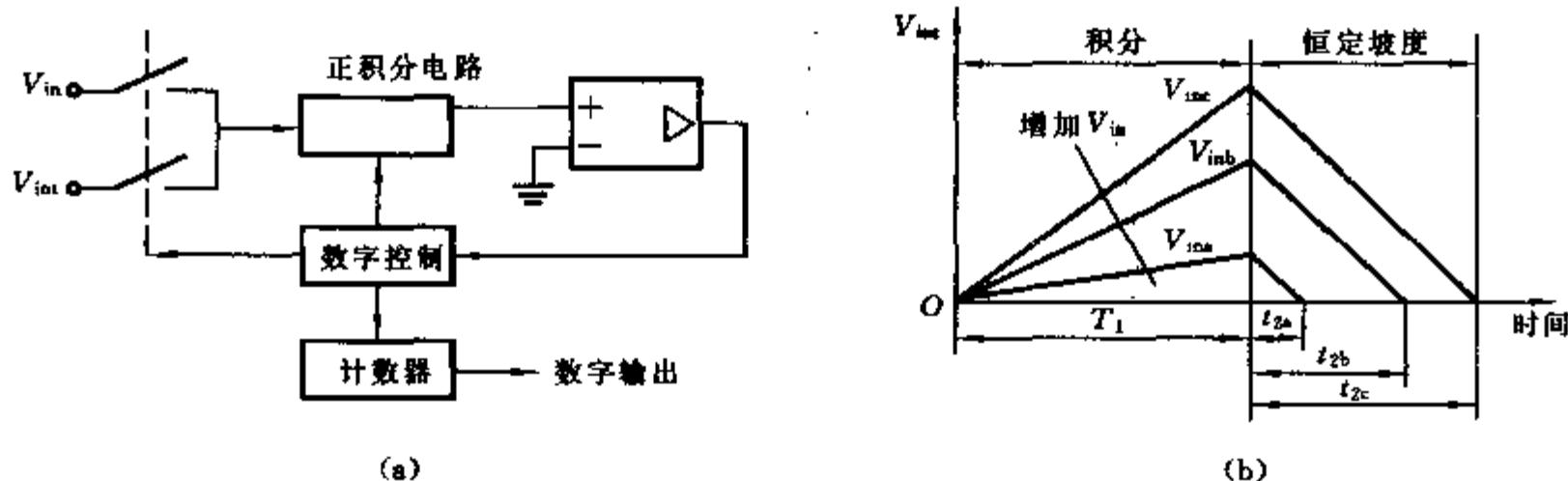


图 3.12 双斜坡 ADC

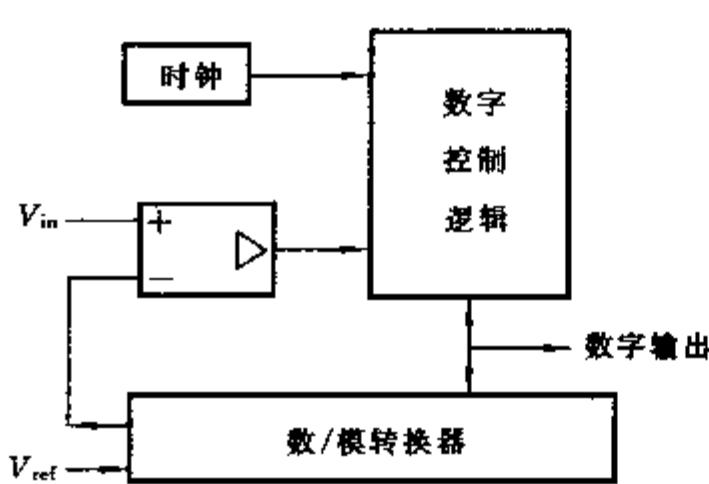
(a)方框图;(b)操作说明波形

4. 逐次逼近式

逐次逼近式 ADC 是电压标度和电荷标度 DAC 的结合。图 3.13 所示的是一个典型的逐次逼近式 ADC 的框图。它由一个比较器, 一个 DAC 和数字控制器逻辑组成。转换是从对待转换的模拟信号采样开始的。然后, 控制逻辑假定 MSB 为“1”, 而其他位为“0”, 这个控制作用于 DAC, 并产生 $0.5V_{ref}$ 的内部模拟信号。这里比较器用于比较这个内部模拟信号和模拟输入信号。如果比较器的输出是高电平, 那么 MSB 就是“1”; 如果比较器的输出是低电平, 那么 MSB 就被改成“0”。这样 MSB 就已确定了。其余各位就按顺序重复这个过程。

图 3.14 所示的是 3 位 ADC 可能的转换路径。注意, 转换一个 n 位字所需的时钟周期数为 n 。

模拟转换周期输入路径相当于图 3.14 所示的 $5/8 \times V_{ref}$ 的路径。



3.13 给出了一个 3 位转换器可能的转换路径

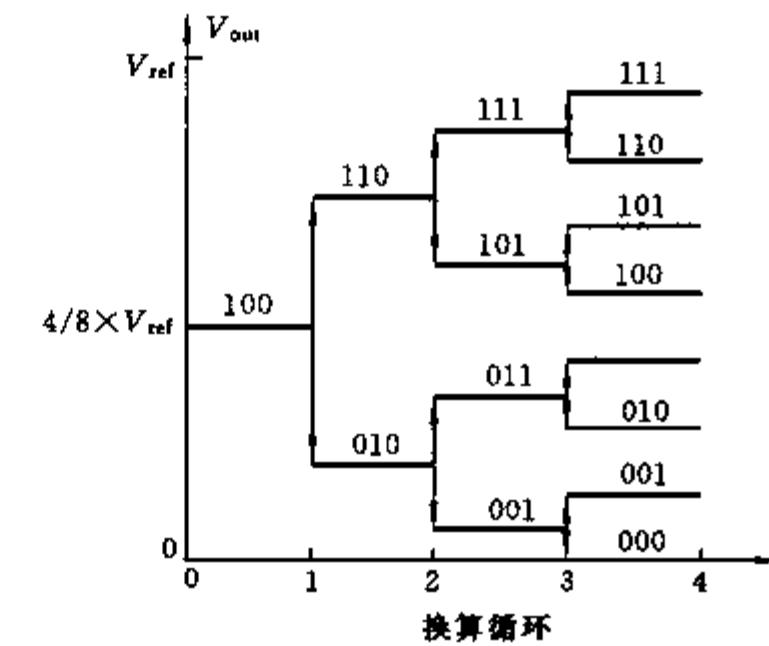


图 3.14 逐次逼近式步骤

5. 并行式或快速式

并行式或快速式 ADC 应用于很高速转换的场合。极限转换速度是一个时钟周期。在这种转换器中，采样时间常常是速度高低的限制因素。转换过程是直接的，如图 3.15 所示。转换一个 n 位字需要 $2^n - 1$ 个比较器。

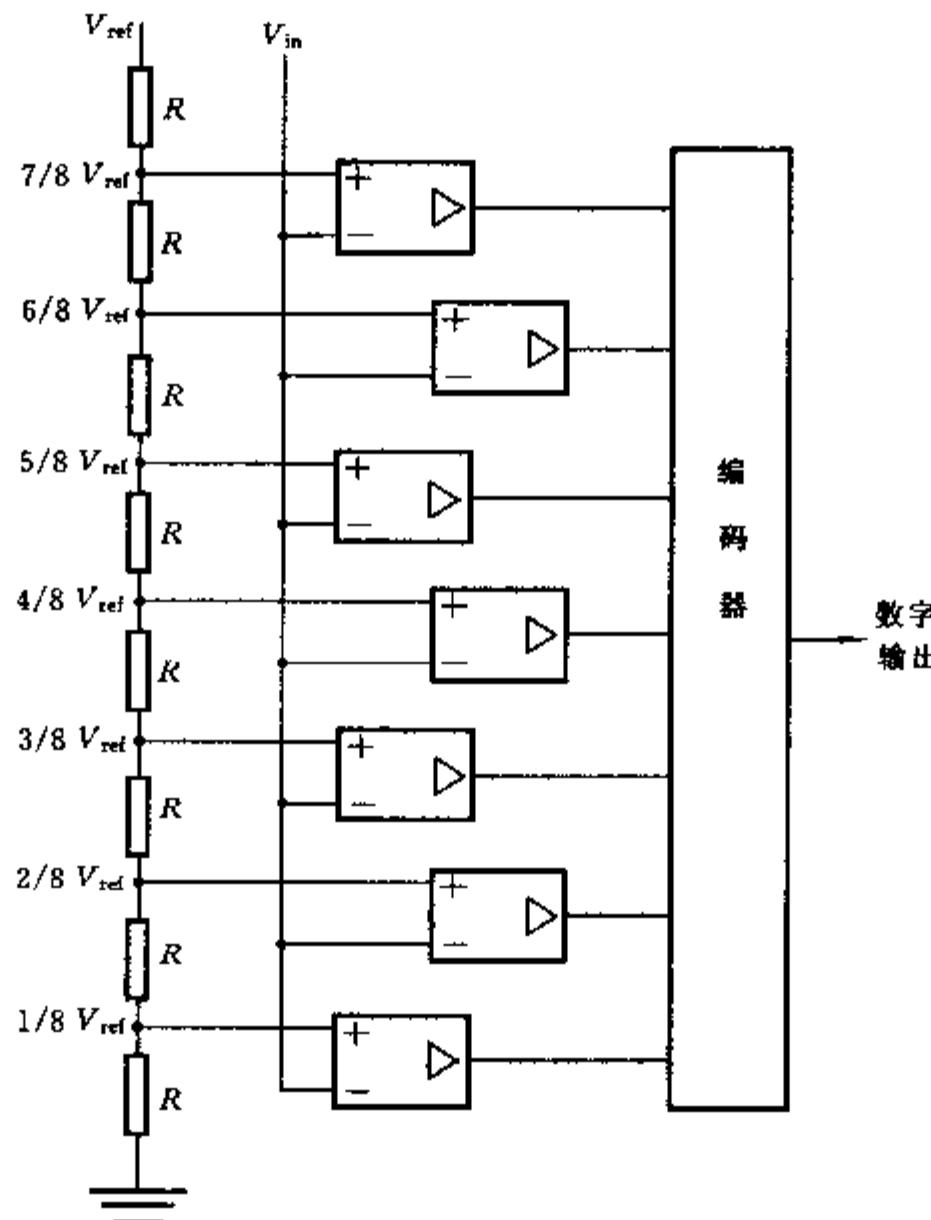


图 3.15 3 位快速式 ADC

3.4.4 采样/保持电路

既然把模拟信号转换成数字信号需要一定的时间,那么在转换期间把模拟信号保持为常值就是有利的。图 3.16 所示的是一个简单的采样/保持电路。它能用于采样模拟信号并在进行 A/D 转换期间冻结。

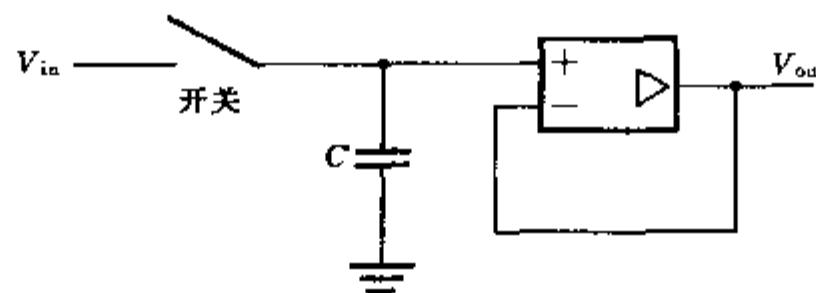


图 3.16 一个简单的采样/保持实现电路

由采样/保持电路引入的误差包括初始电压存储中的残留误差、放大器漂移和存储电压的慢放电。采样/保持电路的动态特性在 ADC 的整个性能中起着很重要的作用。完成一个采样所需的时间决定了 A/D 的最小转换时间,图 3.17 所示的是捕获时间(t_s)和调整时间(t_r)之间的关系。

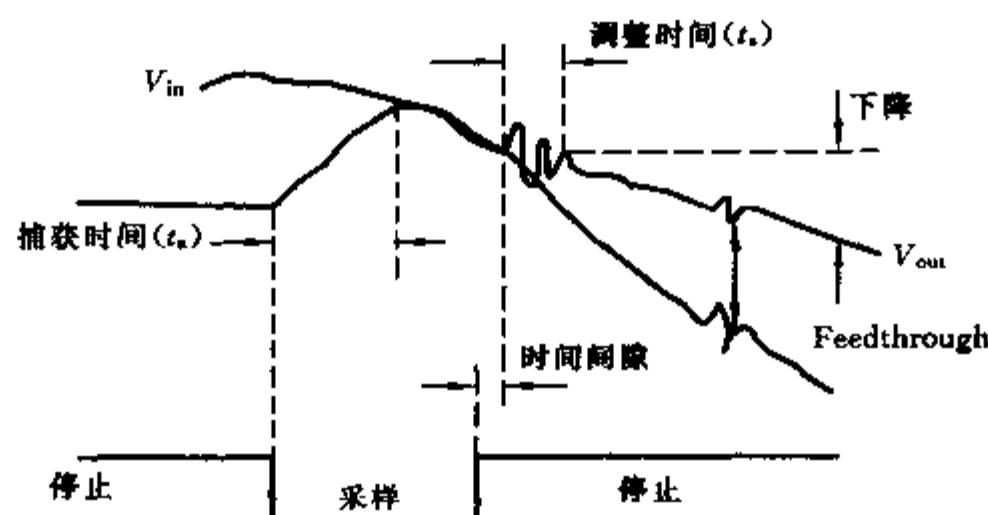


图 3.17 采样/保持电路的输入和输出电压示意图

3.4.5 模拟多路传输器

当有几路信号需要转换时,就必须要么为每一路信号提供一个 A/D,要么使用模拟多路传输器把各种信号分别导向单个的转换器。对于多数生物医学信号,所要求的转换速率很低,因此,适宜选择多路传输器。

通用的模拟多路传输器使用 JFET 或 CMOS 晶体管。图 3.18 所示的是一个简单的 CMOS 模拟开关电路。若干个这种开关接在一个 V_{out} 端就制成多路传输器。开关必须以工作前断开的方式工作,以保证两条输入线不会短接到一起。其他需要考虑的属性包括导通电阻、泄漏电流、串音和稳定时间等。

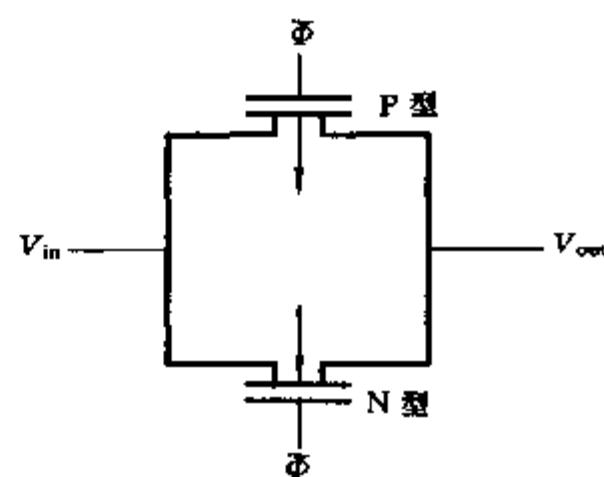


图 3.18 一个简单 CMOS 模拟开关
CMOS 模拟电路传输器基本功能块

3.4.6 放大器

从传感器或电极上得到的生物医学信号是典型的很小的信号,因而,A/D 转换处理前的第一步常常是对信号进行放大。模拟放大电路也能提供滤波,在采样之前,常常用模拟滤波器来带限信号,以减少满足采样定理所需的采样速率和消除噪声。

放大器有如下几种类型。

1. 一般式

对于一般生物医学信号,如 ECG 和 EEG,采用的是简单仪器用的放大器。它提供高输入阻抗和高 CMRR(共模抑制比)。2.4 节中已较详细地讨论了仪器用的放大器和模拟滤波器。

2. 微功率放大器

电池式便携、可移植的生物医学设备要求功率低,而 CMOS 放大器正是微功率设备,因而 CMOS 类放大器应运而生。微功率放大器工作在晶体管的弱转换区。这种工作极大地减小了电源的供给电流,也允许在超低供应电压(1.5V 或更低)下的工作。显然在这么低供应电压下,信号摆动必定很小。

3.5 实验:信号转换

本实验证实了采样速率对信号频谱的影响,并阐明了频域混叠的结果。

3.5.1 采样的应用

(1)依照附录 D 的要求安装 UW DigiScope 程序后,选择 ad(V)ops 项,然后选(S)ample 项。采样菜单允许读或显示一个波形数据文件,以各种速率采样波形,显示采样波形后,通过内插法重建原波形,以及寻找波形的功率谱。下面的步骤阐明了这些函数的功能。可以把它作为指导期使用。结束指导期后,即可使用采样函数完成实验过程。供研究的数据文件有三个:正弦波,不同 f 正弦波之和和方波。该程序不处理单个正弦波。

(2)同一波形同时在上、下通道显示。如果连续波形不能显示,就采用 5000Hz 的高采样速率。从菜单上选择(P)wr spect 来寻找这个波形的频谱。使用(M)easure 项来决定正弦波的 f 。

(3)选择(S)ample 项。在提示的采样次/s 中敲入所需的采样频率(试用 1000 次/s),并敲 ERTUEN,则在低通道上将显示时域采样波形。

(4)先用(R)ecreate 项,再选零阶保持(Z)oh 项,重建采样信号。所显示的波形就像你使用 DAC 直接显示采样速度后出现的波形一样。

(5)从菜单中选择(P)wrSpect 项以寻找波形的频谱。注意显示仅能在从 0 到所选采样 f 的一半之间运行。使用(W)easure 项确定波形的主要 f 。采样前和采样重建后信号的频谱区别在哪里呢?

(6)可选择(D)ata Select 项,其他三个波形的采样重复上述步骤。

3.5.2 实验步骤

(1)安装正弦波,计算其周期。以远高于 Nyquist f 的 f (例如 550 次/s)采样该波,使用零阶保持命令重建信号。你认为采样波的功率谱应是什么样子呢?对采样数据选用(P)wr Spect,解释同你的期望的差别。测量峰值发生的频谱。

- (2) 以 Nyquist $f \pm (5 \sim 10)\%$ 的采样速度采样。描述采样数据的出现和其功率谱，测量响应峰值发生的 f 。以几倍的 Nyquist f 采样，你发现了什么？信号能很好地重建吗？
- (3) 以零阶保持、线性和正弦内插三种内插法重建数据。这些方法的区别是什么？你认为重建数据的功率谱应是什么样子？
- (4) 以两个正弦波叠加和重复(1)~(3)步。
- (5) 以方波重复上述(1)~(3)步。

复习思考题

- 3.1 采样模拟信号时使用低通滤波器的目的是什么？
- 3.2 画出下列 DAC 的特性：①一个 LSB 的偏移误差；② ± 1.5 LSB 的积分线性误差；③ ± 1 LSB 的积分线性误差。
- 3.3 设计一个 4 位电荷标度 DAC。若 $V_{ref} = 5V$ ，当数字输入为 1010 时， V_{out} 是多少？
- 3.4 为什么认为双斜率 ADC 是一种精确的转换方法？若积分器随时间漂移，那么输出会发生什么现象？
- 3.5 若模拟输入为 $9116V_{ref}$ ，请画出 4 位逐次逼近式 ADC 的转换路径。
- 3.6 讨论 3.3 节中所列出的必须考虑的参数对设计生物医学转换系统的重要性。
- 3.7 列出 EEG 装置中所用 ADC 系统的规格。
- 3.8 画出计数式 ADC 的方块图。
- 3.9 解释香农采样定理。若所得的信号在一个周期中只有两个采样值，那么要重建信号应采用什么内插法？
- 3.10 设带宽为 100Hz 的 ECG 信号，其采样速率为 500 次/s，①画出来采样后所得新的数字信号的近似频谱，并在坐标轴上标上重要点。②采样该模拟信号所得新的数字信号的带宽是多少？解释其原因。
- 3.11 为减少畸变，对 400Hz 三角波应采用什么采样速率？解释其原因。
- 3.12 对于 100Hz 余波整流正弦波，若采样速率为 200 次/s，则将这些采样应用 DAC 直接重建波形能否很好地代表原来的信号？解释之。
- 3.13 一个 ADC 输入信号范围是 10V。在下列情况下它能分辨的最小信号(mV)是多少？①10 位转换器，②11 位转换器。
- 3.14 一个 10 位 ADC 能分辨的最小信号的数量级为 10mV，该转换器可能的满刻度电压范围(V)是多少？
- 3.15 一个 12 位 DAC 输出信号范围为 $\pm 5V$ ，其可能的输出最小步长是多少？
- 3.16 对于满刻度输入范围为 $\pm 5V$ 的 ADC，若分辨率为 0.5mV，应需要多少位的 ADC？
- 3.17 正常 QRS 波宽为 100ms。①对于临床 ECG，AHA (American Heart Association) (美国心脏学会) 所规定的采样速率是多少？②若使用 AHA 标准采样速率，则为定义一个 QRS 波需多少采样数据？
- 3.18 QRS 峰-峰值为 1mV 的 ECG 信号通过一个边缘陡峭、100Hz 带通滤波器后，以每秒采样 200 次。该 ECG 马上通过一个 DAC 和低通滤波器重建。比较原始信号和 DAC 输出，

评价下列原因造成二者之间差异的情况：①由采样过程本身；②幅度的峰-峰值；③临床对这种信号的接收。

3.19 QRS 峰-峰值为 1mV 的 ECG，其周期为 100ms，通过截止频率为 100Hz 理想低通滤波器。ECG 采样频率为 200Hz。由于缺乏存储器，采样每隔一点就要去掉一点，因而每秒只有 100 个据点保存下来，该 ECG 马上通过 DAC 和低通滤波器进行重建。比较原始信号和重建信号，并评价由于下列原因造成的不同情况：①畸变，②采样过程本身，③峰-峰值幅度；④该种信号的临床接受。

3.20 IBM PC 信号采集板采用 8 位 ADC 采样 ECG。ECG 放大器提供 1V 的峰-峰值，且转换器输入范围为 0~5V，需要多少位的 ADC 才能恢复信号？

3.21 商用 12 位信号获取板有±10V 输入范围，用于采样 ECG。ECG 放大器提供的峰-峰值为 1V，要代表 ECG 信号需要多少离散幅值？

3.22 解释信号频率和采样定理间的关系。

3.23 描述其非理想效应：①输入滤波器，②输出滤波器。

3.24 若要采样一个 ECG 信号，那么你将采用什么样采样速率和什么样滤波器特性（比如说截止频率）？

3.25 什么类型的 ADC 电路能提供最快的采样速度？

3.26 什么类型的 ADC 电路能平滑高频噪声？

3.27 在 8 位逐次逼近式 ADC 中，信号最初的数字逼近是多少？

3.28 4 位逐次逼近式 ADC 得到一个信号时，最终逼近是 0110，其最终结果前的一次逼近是多少？

3.29 设计 DAC 时，R-2R 双二相电阻网有什么优点？

3.30 对于 8 位逐次逼近式 ADC 而言，在下列情况下其下一次逼近（十六进制）值是多少？①对于输入信号而言，0X90 稍低，②0X80 稍高。

3.31 对于 8 位逐次逼近式 ADC 而言，若其某步逼近结果为下列值：① 0X10，②0X20，则其可能的下一步逼近值是多少？

3.32 8 位 ADC，其驱动内部逐次逼近电路的时钟为 80kHz，①该转换器可达到最快采样速率是多少？②若 0X80 代表的信号值为 1V，则该转换器能分辨的最小信号是多少(mV)？

3.33 在信号转换系统中为保存模拟电压值应采用什么电路？画出其原理图并解释其工作原理。

3.34 附录 A 中介绍的内部 IBM PC 信号获取板是用于采样 ECG 的。其中，放大器将 ECG 放大，使得 1mV 能用转换器的所有的 12 位，则其能分辨的最小的 ECG 幅值是多少 (mV)？

3.35 8 位逐次逼近式 ADC 用于采样 ECG。放大器将 ECG 放大，使得 1mV 数量级的信号能使用转换器的所有的 8 位。则其能分辨的最小的 ECG 幅值是多少？

3.36 Wisconsin 计算机(COW)的 ADC 计算机芯片采用 CMOS 技术，它包括一个 8 位逐次逼近式 ADC，其采样周期为 100ms。芯片上的模拟多路传输器可提供 8 路采样。①使用这种 COW 芯片，采样一路信号的最快速度是多少（单位：次/s）？②若要使用所有 8 个通道，则每个通道采样速度是多少？③为得到最大采样速度，驱动逐次逼近电路外部时钟所需的最小频率是多少？④若该板采用 TTL 技术，二者相比，列出采用 CMOS 的两个优点。

第四章 数字滤波器基础

本章主要介绍数字滤波器的基本概念、优缺点及其与模拟滤波器的区别。数字滤波器是相对于模拟滤波器而言的，它具有离散性质。不同类型的数字滤波器的实现将在以后各章中逐一加以介绍。

4.1 数字滤波器

数字滤波器的作用与模拟滤波器相同，但它们的实现过程是完全不同的。模拟滤波器是通过采用有源或无源电子电路来实现并工作在连续波形上的，而数字滤波器是通过采用数字逻辑电路或计算机程序来实现的，其信号是采样连续波形得到的一个数字序列，当今，由于计算机的有效使用，数字滤波器已得到广泛的应用。只要编写一个计算机程序就能实现任一类型数字滤波器的功能。

数字滤波器有着若干优于模拟滤波器的优点。在实现方法上的特点（软件/数字电路），使数字滤波器对噪声具有很强的抗干扰能力；其精确度仅取决于它的舍入误差，而舍入误差是由设计者选择滤波器参数时直接决定的，因此通常很容易且花费不多的费用就可改变滤波器的工作特性（如截止频率）；不像模拟滤波器那样，其功能不随某些因素诸如元件老化、温度和供电电压等的变化而改变。这种特性在医疗器械的应用中是很重要的。并且在模拟电路中存在漂移，这可能会引起低频信号的失真。

4.2 z 变换

正如第三章所讨论的，采样过程是把一个连续信号转换成数字序列的过程。图 4.1 所示的是描述该序列产生的过程。

$$\{a(0), a(T), a(2T), a(3T), \dots, a(kT)\} \quad (4.1)$$

这组数描述了波形 $a(t)$ 在 $0, T, \dots, kT$ 各时刻所对应的采样值，其中 T 为采样周期。

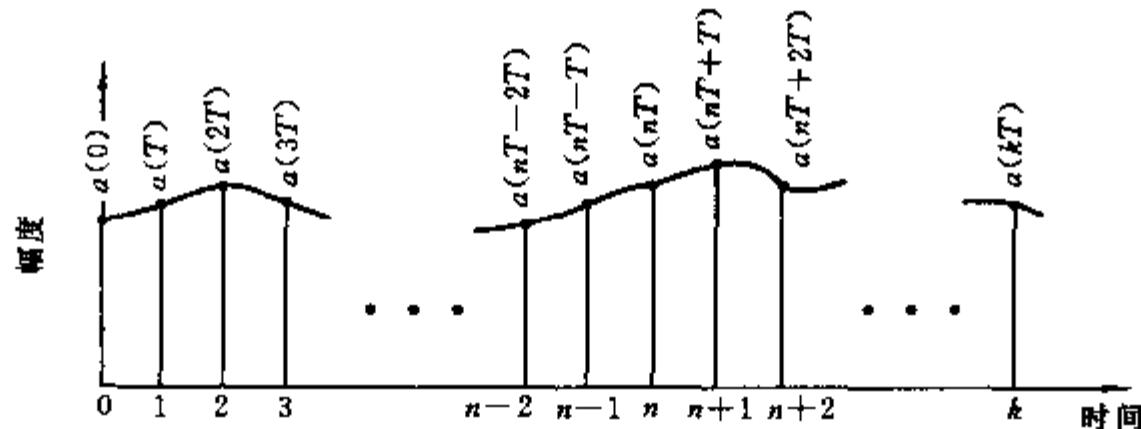


图 4.1 采样—连续信号所得到的脉冲序列

相邻脉冲间采样间隔为 T 秒

下面研究具有 z 变换性质的数字滤波器。根据定义,任何一个数列

$$\{f(0), f(T), f(2T), \dots, f(kT)\} \quad (4.2)$$

其 z 变换为:

$$F(z) = f(0) + f(T)z^{-1} + f(2T)z^{-2} + \dots + f(kT)z^{-k} \quad (4.3)$$

一般表达式为:

$$F(z) = \sum_{n=0}^k f(nT)z^{-n} \quad (4.4)$$

若上述变换成立,则式(4.1)的 z 变换为:

$$A(z) = a(0) + a(T)z^{-1} + a(2T)z^{-2} + a(kT)z^{-k} \quad (4.5)$$

或

$$A(z) = \sum_{n=0}^k a(nT)z^{-n} \quad (4.6)$$

假设想求出任何一个数列,如数列

$$\{1, 2, 5, 3, 0, 0, \dots\} \quad (4.7)$$

的 z 变换。则根据式(4.4),即能写出其 z 变换表达式:

$$X(z) = 1 + 2z^{-1} + 5z^{-2} + 3z^{-3} \quad (4.8)$$

在 z 变换中变量 z^{-1} 表示相邻两点间相差一个周期(T 秒)。 z 的负指数表示从采样过程开始到该项经过了多少个采样周期, z 项的系数为采样数据点的值。例如由式(4.8)可知,在 $t=0$ 时刻获得第一个采样数据点的值为 1,而 $t=2T$ 时刻获得第三个数据点的值为 5。

在数字滤波器中 z 变换是很重要的,它描述了一个采样过程,并在数字域中起到与模拟滤波器的拉普拉斯变换类似的作用。图 4.2 所示的是数字离散化时间信号与模拟连续时间信号

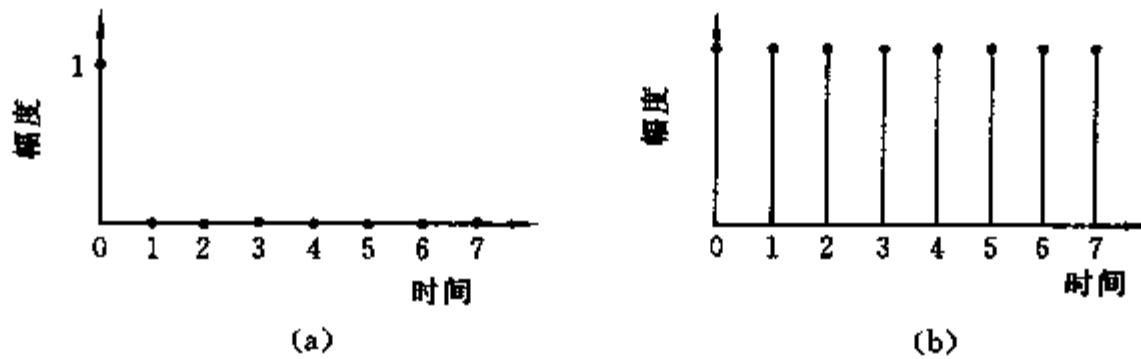


图 4.2 离散时间信号的例子

(a) 单位脉冲函数:当 $n=0$ 时, $\delta(nT)=1$;当 $n \neq 0$ 时, $\delta(nT)=0$;

(b) 单位阶跃函数, $f(nT)=1$, 对所有 n 值, 变量 n 取整数

的两个例子。图 4.2(a)所示的为单位脉冲函数,它类似于 δ 脉冲函数,可表达为:

$$\begin{cases} f(nT) = 1 & (n = 0) \\ f(nT) = 0 & (n > 0) \end{cases} \quad (4.9)$$

对应的数列为:

$$\{1, 0, 0, 0, 0, 0, \dots\} \quad (4.10)$$

因此,单位脉冲函数的 z 变换为:

$$F(z) = 1 \quad (4.11)$$

使用单位脉冲函数作为输入函数去研究滤波器的功能,这是一个很重要的发现。

对于单位阶跃函数(图 4.2(b)),有

$$f(nT) = 1 \quad (n \geq 0)$$

相应数列为：

$$\{1, 1, 1, 1, 1, 1, \dots\} \quad (4.12)$$

因此，单位阶跃函数的 z 变换为：

$$F(z) = 1 + z^{-1} + z^{-2} + z^{-3} + \dots \quad (4.13)$$

这种变换式是一个非零次的无限求和式。利用二项式定理，可将该式转换成一种比例多项式，即

$$1 + V + V^2 + V^3 + \dots = \frac{1}{1 - V} \quad (4.14)$$

令 $V=z^{-1}$ ，则单位阶跃信号的 z 变换可表示为：

$$F(z) = \frac{1}{1 - z^{-1}} \quad (4.15)$$

图 4.3 所示的是一些一般信号的 z 变换。

$f(t), t \geq 0$	$f(nT), nT \geq 0$	$F(z)$
1(单位阶跃)	1	$\frac{1}{1 - z^{-1}}$
t	nT	$\frac{Tz^{-1}}{(1 - z^{-1})^2}$
e^{-at}	e^{-anT}	$\frac{1}{1 - e^{-aT}z^{-1}}$
te^{-at}	nTe^{-anT}	$\frac{Te^{-aT}z^{-1}}{(1 - e^{-aT}z^{-1})^2}$
$\sin(\omega_c t)$	$\sin(n\omega_c T)$	$\frac{(\sin\omega_c T)z^{-1}}{1 - 2(\cos\omega_c T)z^{-1} + z^{-2}}$
$\cos(\omega_c t)$	$\cos(n\omega_c T)$	$\frac{1 - (\cos\omega_c T)z^{-1}}{1 - 2(\cos\omega_c T)z^{-1} + z^{-2}}$

图 4.3 连续时间函数和相应的模拟离散时间函数的 z 变换

4.3 数字滤波器的组成

只需要下列三种类型的电路就可以实现任何数字滤波器：①延时器，②乘法器，③加法器。

图 4.4 所示是这三种电路的符号。考虑下列序列

$$\{x(0), x(T), x(2T), \dots, x(nT)\} \quad (4.16)$$

它的 z 变换为：

$$x(z) = x(0) + x(T)z^{-1} + x(2T)z^{-2} + \dots + x(nT)z^{-n} \quad (4.17)$$

若把这个序列作用到图 4.4(a) 所示的输入端，则可得到输出序列

$$\{0, x(0), x(T), x(2T), \dots, x(nT - T)\} \quad (4.18)$$

这个输出序列的 z 变换为：

$$y(z) = 0 + x(0)z^{-1} + x(T)z^{-2} + \dots + x(nT - T)z^{-n} \quad (4.19)$$

在这种情况下， $x(0)$ 在 $t=0$ 时输入到存储区，与此同时，在此存储区内的原来初始化的内容（通常假设为 0）被输出。在 $t=T$ 时， $x(T)$ 为输入， $x(0)$ 为输出。在得到 A/D 转换的采样数据的每一个脉冲时刻，将有一个新的数据输入，而以前存储的一个数据被输出（存储了 T 秒）。

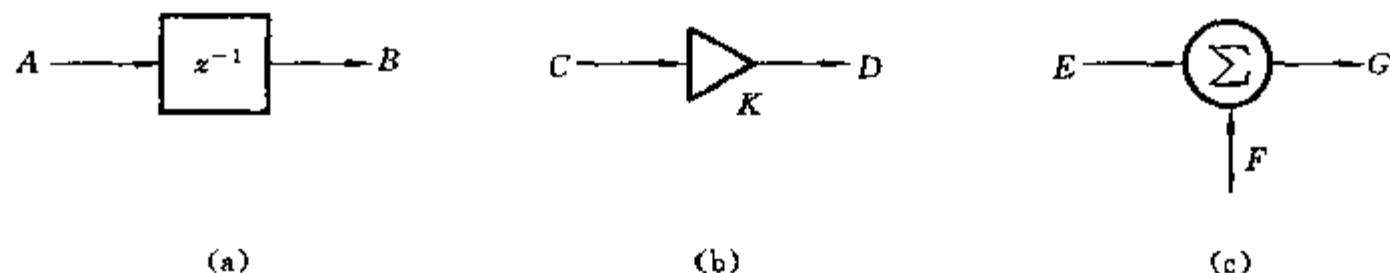


图 4.4 数字滤波器的操作符

- (a) 存储数字并延时一个时钟周期, $B = A$ 且 B 滞后 T 秒;
- (b) 乘以一个常数, $D = KC$, 无延时;
- (c) 两个数相加, $G = E + F$, 无延时

因而,一个输出序列对应于一个输入序列,只是整个序列被延迟了 T 秒。

根据式(4.19)和式(4.17),可得到输出的 z 变换与输入的 z 变换之间的关系,即

$$Y(z) = X(z)z^{-1} \quad (4.20)$$

延迟环节的传递函数为:

$$H(z) = \frac{Y(z)}{X(z)} = z^{-1} \quad (4.21)$$

微型计算机很容易在一个存储单元中存放一组连续数据,为以后某个恰当时刻调用。

图 4.4(b)所示的是实现数字滤波器的第二种功能的电路——乘法器。对于当前乘法器输入的序列中的每一个数据,都会立即输出相应的乘积

$$Y(z) = \beta X(z) \quad (4.22)$$

式中, β 是常数。理想情况下,在乘法器中没有存储或时间延时。一序列 {5, 9, 0, 6}, 乘以常数 5, 得一新序列 {25, 45, 0, 30}, 输出序列中的每一个数据都是由输入序列中相应数据在输入时产生的(无延时)。

数字滤波器最后实现的功能电路是加法器。如图 4.4(c)所示,在每个脉冲到来时刻,来自两列的数据相加后得到一个连续的输出序列。在理想情况下,相加后仍无延时。

输出与输入间的 z 变换为:

$$Y(z) = X_1(z) + X_2(z) \quad (4.23)$$

当然,乘法电路和加法电路需要一定时间去完成,但在数字滤波器中,控制时间的延时是由一些储能元件控制采样过程来实现的。因而,只要数字滤波器中的所有数字的操作在 T 秒内完成,单独的乘法和加法操作都被认为在 0 时间内完成,滤波器就能实现实时操作。加法器和乘法器所引起的非零延时不是大的障碍,因为在实际系统中,加法器和乘法器的输出不是立即需要的。由于它们的输入在 T 秒内恒定,实际上在 T 秒内产生输出,都可认为没有延时。

所有通用微处理机都能实现数字滤波器的三种必要操作: 延时器、乘法器和加法器的操作。它们能实现数字滤波器的基本功能。若能在下一个输入值出现前尽快地对所有操作产生一个输出值,就称为实时操作。实时滤波器非常像模拟滤波器,在施加输入信号的同时就有被滤波器进行了滤波的输出信号产生。

4.4 数字滤波器的类型

数字滤波器的传递函数是输出序列的 z 变换与输入序列的 z 变换之比,即

$$H(z) = \frac{Y(z)}{X(z)} \quad (4.24)$$

数字滤波器有两种基本类型——非递归系统和递归系统滤波器。对于非递归系统滤波器，传递函数具有一个有限的多项表达式：

$$H(z) = \sum_{i=0}^n h_i z^{-i} = h_0 + h_1 z^{-1} + \cdots + h_n z^{-n} \quad (4.25)$$

对于递归滤波器系统，其传递函数可表达成两个多项式之比：

$$H(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{1 - \sum_{i=1}^n b_i z^{-i}} = \frac{a_0 + a_1 z^{-1} + \cdots + a_n z^{-n}}{b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}} \quad (4.26)$$

$H(z)$ 为 0 时， z 的值称为传递函数的零点； $H(z)$ 为无限时， z 的值称为极点。由此可知，令分子等于 0 时，求得的 z 值为滤波器的零点；令分母等于 0 时，求得的 z 值则为滤波器的极点。于是，在零点处传递函数的值为 0；在极点处，系统变为不确定。

从非递归系统滤波器的传递函数可知，仅在 $z=0$ 时有极点。在以后的章节中将讨论在 z 平面上，极点的配置决定了它的稳定性。由于非递归系统滤波器仅在 $z=0$ 时有极点，因此，它总是稳定的。

4.5 差分方程的传递函数

一旦已可用于实现一个数字滤波器的数值算法的差分方程，就能很快地求出滤波器的特性变换方程。考虑差分方程

$$y(nT) = x(nT) + 2x(nT-T) + x(nT-2T) \quad (4.27)$$

式中， $x(nT)$ 和 $y(nT)$ 分别是与当前采样时刻相关的输入、输出序列中的采样点，经无延迟的 z 变量离散化，与 $X(z)$ 和 $Y(z)$ 相对应。同样，采样点 $x(nT-T)$ 经 z 变换后将延迟一个采样点或者对应于 $X(z)z^{-1}$ 。可写出 $Y(z)$ 关于 $X(z)$ 的方程，即

$$Y(z) = X(z) + 2X(z)z^{-1} + X(z)z^{-2} \quad (4.28)$$

其传递函数为：

$$H(z) = \frac{Y(z)}{X(z)} = 1 + 2z^{-1} + z^{-2} \quad (4.29)$$

从离散时间变量和 z 变量之间的关系可以看出，若知道差分方程，就能写出传递函数；反之亦然。

4.6 z 平面的零-极点分布图

从时域来看 z 变换时， z 变换的数学表达式定义为：

$$z = e^{j\omega T} \quad (4.30)$$

在复频域中，

$$s = \sigma + j\omega \quad (4.31)$$

因此

$$z = e^{\sigma T} e^{j\omega T} \quad (4.32)$$

根据定义， z 的模为：

$$|z| = e^{\sigma T} \quad (4.33)$$

相角为：

$$\angle z = \omega T \quad (4.34)$$

若令 $\sigma=0$, 即 z 的模为 1, 则有

$$z = e^{j\omega T} = \cos(\omega T) + j\sin(\omega T) \quad (4.35)$$

这就是在 z 平面上以 1 为半径的单位圆的方程。由于 z 平面是 s 平面的直接数学映射, 故可根据 s 域映射到 z 域的条件来讨论滤波器的稳定性, 即利用 s 域中的稳定条件去研究 z 域的稳定性。

由 s 平面到 z 平面映射可知, s 平面上的虚轴 ($j\omega$) 映射到 z 平面上为单位圆; σ 的负值表示 s 平面上的左半平面, 映射到 z 平面上的单位圆内部; σ 的正值表示 s 平面上的右半平面, 映射到 z 平面上的单位圆外部。

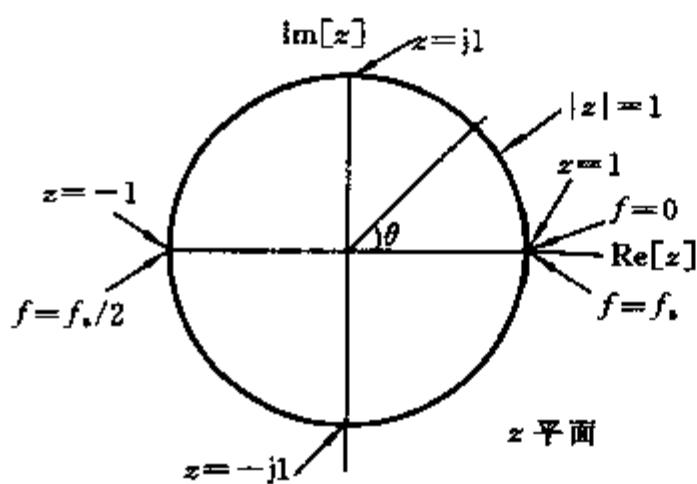


图 4.5 z 平面上的单位圆

$\theta = \omega T \approx f \approx f/f_s$, 表明可用几种方式来表达 z 平面上的角度。

在 s 平面上, 在虚轴右边的极点表示不稳定; 在虚轴上的任何一个极点一定是单极点。从 s 平面与 z 平面之间的映射关系, 可得出在 z 平面上稳定的一般规则: 所有极点必须位于单位圆内或圆上。若在单位圆上, 则一定是单极点。零点不影响稳定性, 可在 z 平面上任何位置。图 4.5 所示的是 z 平面的一些重要特征。任何一个相角 ω 都可确定单位圆上的一个点。由于 $\omega = 2\pi f$ 和 $T = 1/f_s$, 这个角度为:

$$\omega T = 2\pi \frac{f}{f_s} \quad (4.36)$$

单位圆上任一点所对应的角度定义为确定频率与采样频率的比值。若 $f = f_s$, 则 $\omega T = 2\pi$, 因此, 采样频率与 2π 弧度角一致。若 $f = 0$, 则 $\omega T = 0$; 因此, 重复信号

位于 0 度上。

另一个重要的频率是 $f = f_s/2 = f_0$, 即在 $\omega T = \pi$ 处, 这个频率称为转折频率, 其值等于采样频率的一半。由于采样定理要求采样频率是出现的信号中的最高频率的 2 倍, 因此, 这个转折频率代表了数字滤波器能产生的最大频率(见第二章)。不象在连续系统中, 频率可延伸到无穷。在离散系统中, 有意义的频率仅从 0 到 π 弧度, 与 DC 和 $f_s/2$ 频率相一致。在式(4.30)中 z 的原始定义的直接结果是从 s 平面到 z 平面的非线性映射。

请注意: 抗混消不能由任何数字滤波器完成, 而只能将采样频率升高到最高频率的 2 倍。但这个频率往往是不切实际的, 因而, 许多数字信号处理器都设有前端模拟调谐器——抗混消滤波器。

可用几种不同的方式来表达频率响应: 角度、采样频率的倍数或频率与采样频率的比率。360°角表示沿着单位圆旋转一周, 与采样频率一致。为避免产生旁瓣, 可将输入频率限制在采样频率的一半以内, 使得振幅和相位响应的有用范围在 0°~180°以内。

由图 4.6 可见, 可用几种方式来表达单位圆上的点所对应的角。若利用比率 f/f_s , 则它被称为规范化频率。若已知采样频率, 则可根据频率 f 间接地确定单位圆上的点所对应的角的位置。这说明了数字滤波器的一个重要特征——频率响应特性直接与采样频率有关。因而, 若采样频率为 200Hz, 则数字滤波器有零点位于单位圆上的 90°处(即 50Hz 处)。若设想有个零点在 25Hz 处, 则要达到这种结果的简单方法是将采样频率减少一半。

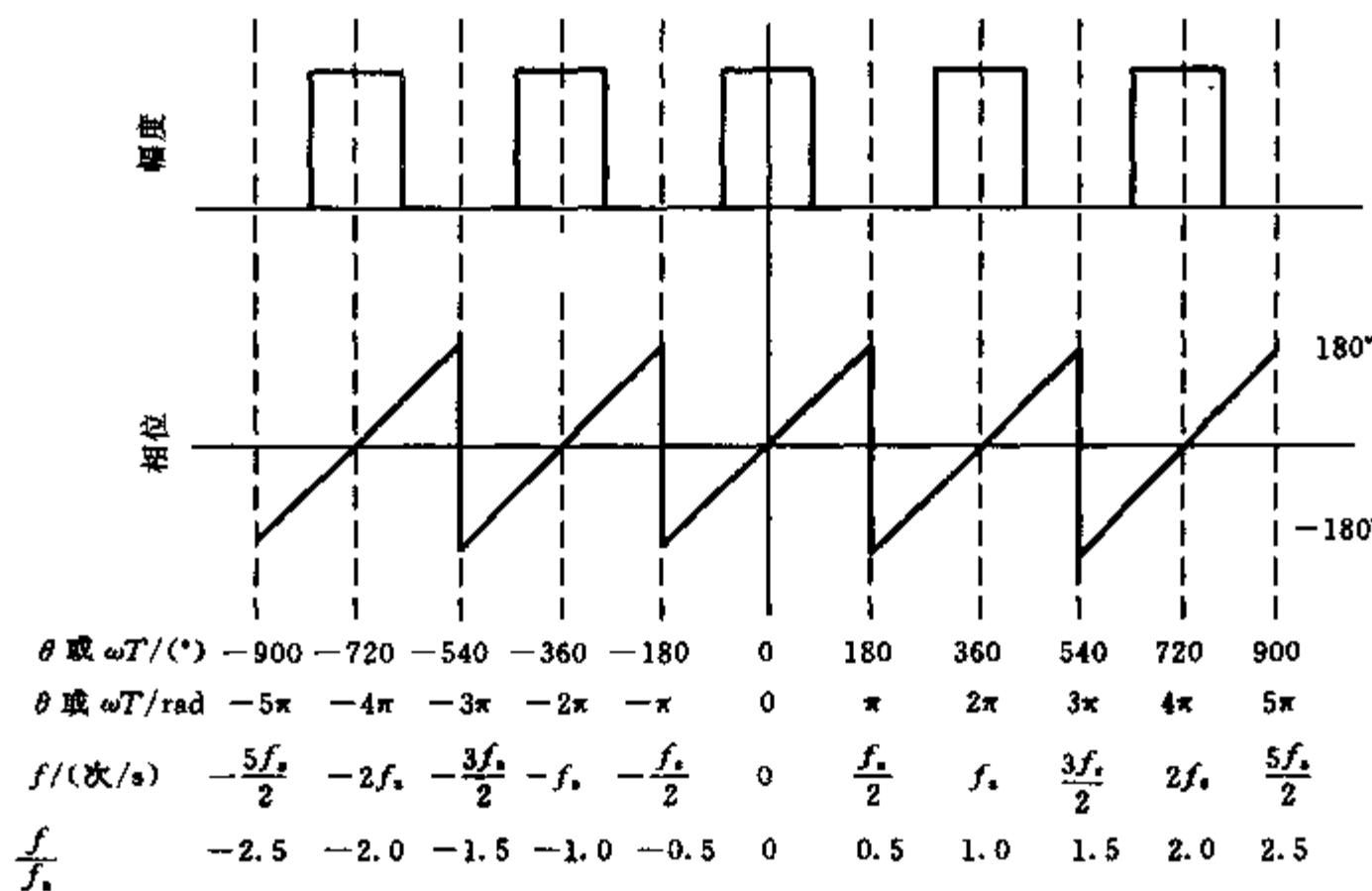


图 4.6 数字滤波器的振幅和相位响应的频率轴

例如,考虑下面的传递函数:

$$H(z) = \frac{1}{3}(1 + z^{-1} + z^{-2}) \quad (4.37)$$

为了找到 z 平面上的零点、极点位置,首先将式(4.37)的右边乘以 z^2/z^2 ,使传递函数中的变量为正指数幂,即

$$H(z) = \frac{1}{3}(1 + z^{-1} + z^{-2}) \frac{z^2}{z^2} = \frac{z^2 + z + 1}{3z^2} \quad (4.38)$$

令式(4.38)的分子等于零,求得零点,即

$$z^2 + z + 1 = 0 \quad (4.39a)$$

得两个共轭复数根为:

$$z = -0.5 \pm j0.866 \quad (4.39b)$$

这两个零点位于单位圆上 $\omega T = \pm 2\pi/3$ 处。若采样频率为 180Hz,则在 120°处的零点将完全消除在 60Hz 处的任何信号。令式(4.38)的分母等于 0,求得极点,即

$$z^2 = 0 \quad (4.39c)$$

可得到一个极点,位于 z 平面的原点上,即

$$z = 0 \quad (4.39d)$$

这个极点到单位圆上所有点都是等距离,在所有频率内对振幅影响相同。因此,在零-极点图上一般不标明。

4.7 橡胶膜的概念

为了对零-极点图有感性认识,设想 z 平面无穷大,一个扁平的橡胶膜片固定在这个无穷大平面上,且无约束。滤波器的零-极点模式决定了膜片上的点将增加至无穷(如极点)或固定在膜片表层(如零点)。图 4.7(a)所示的是 z 平面上的橡胶平面与单位圆重合的情况。

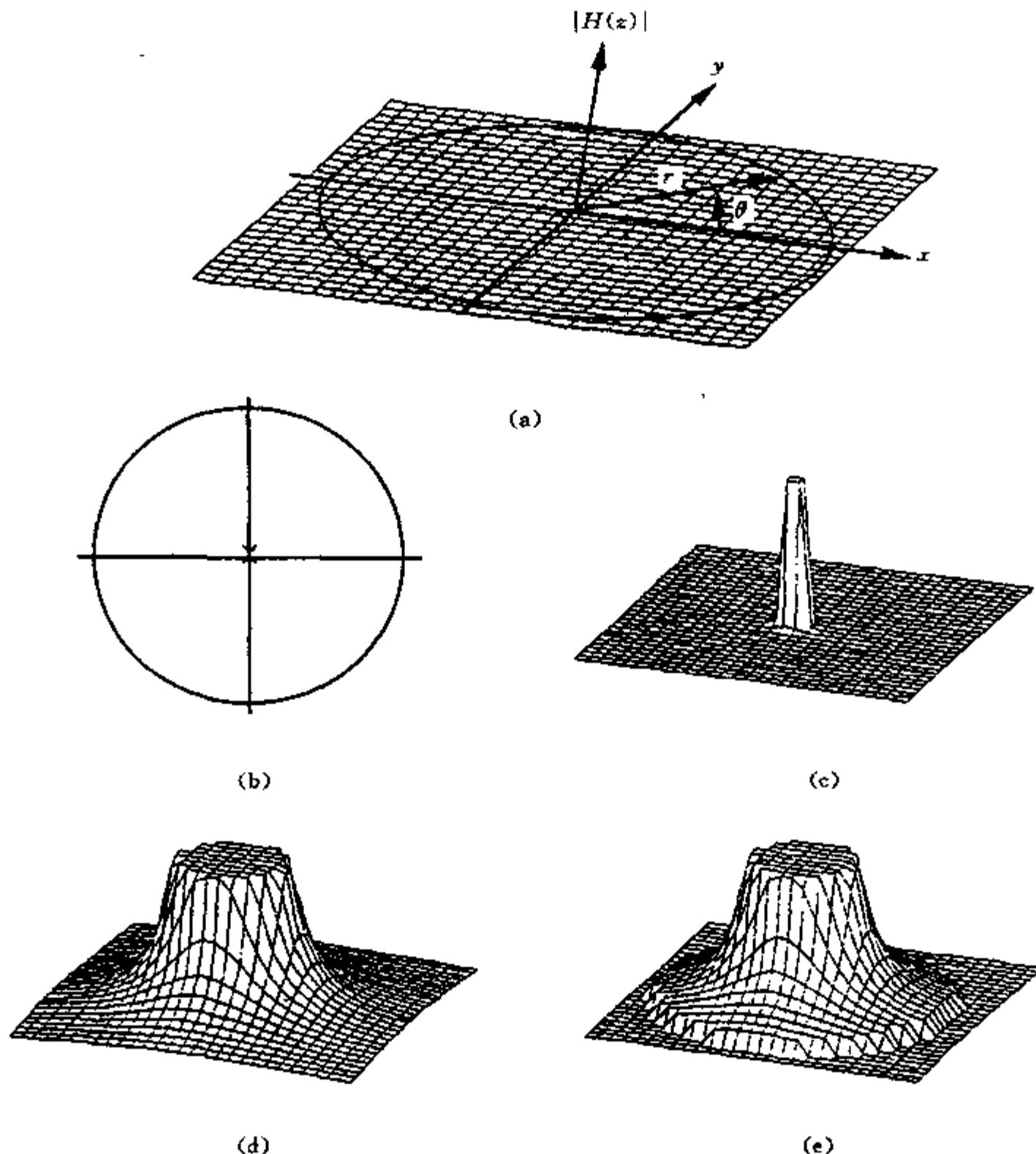


图 4.7 z 平面上的橡胶膜片模拟图

- (a)复数 z 平面区域上的单位圆, x 轴, y 轴和 $|H(z)|$ 轴;
- (b)单极点位于 $z=0$ 处的零极点图;
- (c)从高处俯视 z 平面上极点如何将膜片顶起;
- (d)从橡胶面平面看极点如何将膜片顶起;
- (e)在单位圆外令 $|H(z)|=0$ 来观察橡胶膜在单位圆内的隆起情况

考虑传递函数

$$H(z) = z^{-1} = \frac{1}{z} \quad (4.40)$$

它只有一个极点,位于 z 平面的原点上,即 $z=0$ 。假设在 $z=0$ 处设置一个无限极点,使它在 $z=0$ 处有突起。图 4.7(b)所示的是这个极点在零-极点图上的位置。图 4.7(c)所示的是从高处俯视 z 平面上的极点如何将膜片顶起来的。在图 4.7(d)中,主要考虑在单位圆上极点的直接影响,把观察点由这个橡胶平面高处移到平面上。在极点 $z=0$ 处,注意各个方向是怎样均匀突起的。

这样,这个单位圆从表面的各个方面等距离隆起。由于单位圆代表了频率轴,在单位圆上,

橡胶膜片突起的面积代表了 $|H(z)|$ 的值,即为振幅响应的大小。

为了描述滤波器的特性,主要关心的是在单位圆上膜片突起的面积,和单位圆内或圆外不重要的变化。因此,在图 4.7(e)中,我们限制传递函数的值在单位圆外任何处都为零,是为了更好地观察单位圆内发生的变化。可以看到,在单位圆内相同半径处传递函数的大小相同。用 $z=e^{j\omega T}$ 代入函数式(4.40),得复数频率响应

$$H(z) = e^{-j\omega T} = \cos(\omega T) - j\sin(\omega T) \quad (4.41)$$

这个函数的模值为:

$$|H(z)| = \sqrt{\cos^2(\omega T) + \sin^2(\omega T)} = 1 \quad (4.42)$$

相位响应为:

$$\angle H(z) = -\omega T \quad (4.43)$$

在单位圆上橡胶膜片突起的高度均为 1。单位圆上这个函数在 $0^\circ \sim 180^\circ$ 之间对应的频率范围为 $0 \sim f_s/2$,其值表示在单极点 $z=0$ 处的振幅响应。式(4.42)说明,任一允许频率范围内的周期信号通过滤波器后其幅值都不会改变。

相位响应公式(4.43)说明,输出信号与输入信号有相同频率,但在相位上滞后了一个相角 ωT ,这个滤波器是一个全通滤波器,所有频率的信号都能通过,只是有一定的相位延迟。

为了研究零-极点对橡胶膜片的影响,考察下面一个传递函数,它含有两个零点和两个极点:

$$H(z) = \frac{1 - z^{-2}}{1 - 1.0605z^{-1} + 0.5625z^{-2}} \quad (4.44)$$

为了找到零-极点在 z 平面上的位置,将传递函数乘以 z^2/z^2 ,使 z 的指数变成正的:

$$H(z) = \frac{1 - z^{-2}}{1 - 1.0605z^{-1} + 0.5625z^{-2}} \times \frac{z^2}{z^2} = \frac{z^2 - 1}{z^2 - 1.0605z + 0.5625} \quad (4.45)$$

令式(4.45)的分子等于 0,求得零点,即

$$z^2 - 1 = 0 \quad (4.46)$$

或

$$(z + 1)(z - 1) = 0 \quad (4.47)$$

因而,传递函数的两个零点为:

$$z = \pm 1 \quad (4.48)$$

令式(4.45)的分母等于 0,即

$$z^2 - 1.0605z + 0.5625 = 0 \quad (4.49)$$

可得到共轭复根极点,即

$$z = 0.53 \pm j0.53 \quad (4.50)$$

图 4.8(a)所示的是零-极点图,它表示了这个传递函数 $H(z)$ 的两个零点和两个极点位置。这零点固定在膜面上 $z = \pm 1$ (即模为 1,相角为 $1 \angle 0^\circ$ 和 $-1 \angle 180^\circ$)处。极点位于 $z = 0.53 \pm j0.53$ (即 $0.75 \angle \pm 45^\circ$)。图 4.8(b)所示的是单位圆的变形(即频率轴)。从这方向看,图 4.8(a)所示的平面以垂直轴为轴线按顺时针方向在 $0^\circ \sim 180^\circ$ 内以每 20° 方位角,增加 40° 的坡度倾斜度,在图 4.8(a)所示的最高点相差 90° 。这个橡胶平面在 $0^\circ \sim 180^\circ$ 之间的变形代表了滤波器的振幅响应。

在 0° 角处的零点完全减弱了 DC 输入。在 180° 角处的零点(即 $f_s/2$)限制了滤波器最高可能的输入频率。由采样定理可知,任何一个信号进入滤波器后,其频率将变为采样频率的一半。单位圆被固定在这两点上,其中极点将引起单位圆变形,在 0 频率和最高频率间提供一个通

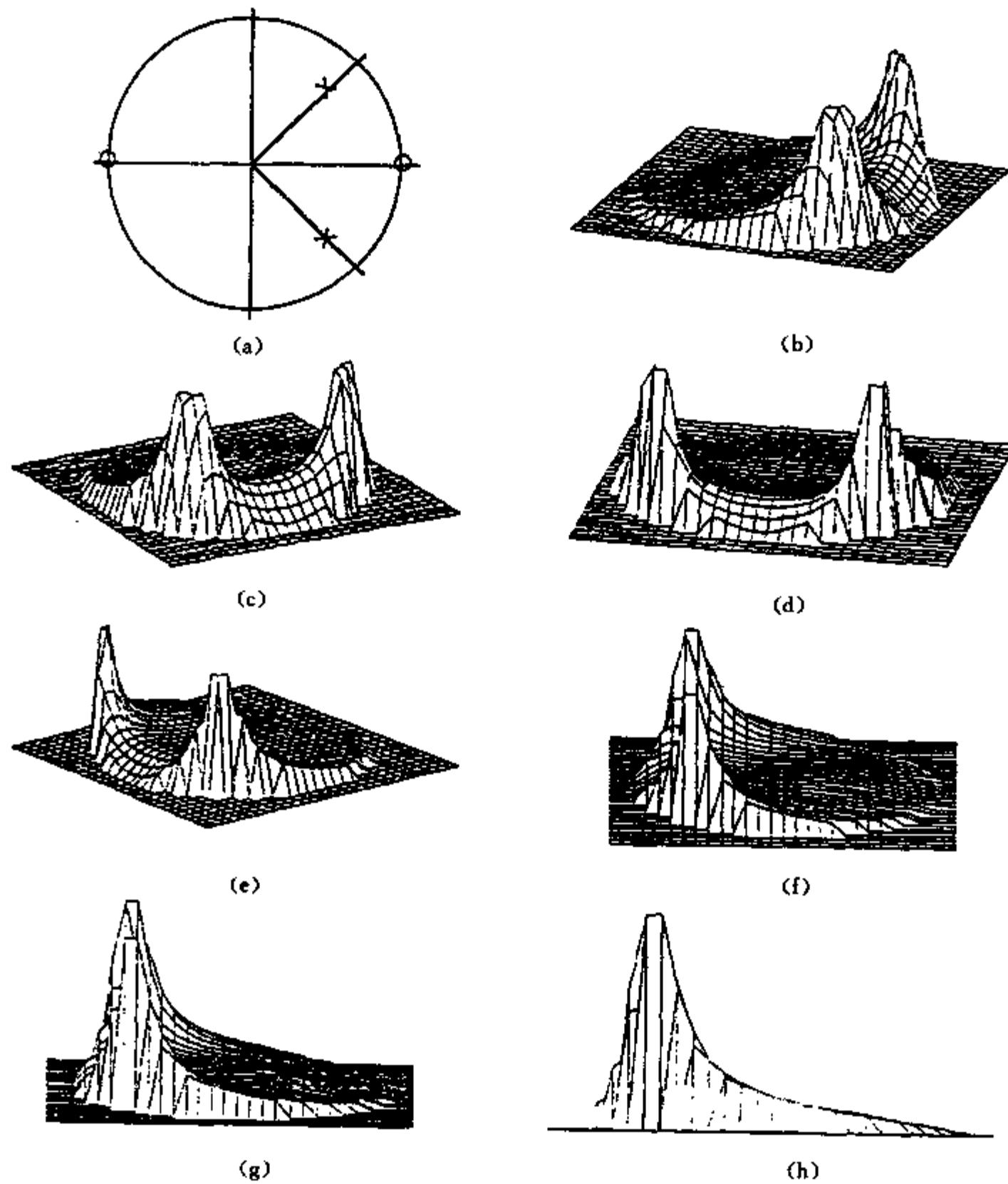


图 4.8 有两个零点和两个极点的滤波器

其零点位于 $z = \pm 1$ 处, 而极点位于 $z = 0.53 \pm j0.53$ (即 $\gamma = 0.75, \theta = 45^\circ$) 处。零点使得该处的橡胶膜下陷, 而极点使得该处延伸至无穷。(a) 零-极图, $Az = 0^\circ, EL = 90^\circ$; (b) $Az = 20^\circ, EL = 40^\circ$; (c) $Az = 60^\circ, EL = 40^\circ$; (d) $Az = 100^\circ, EL = 40^\circ$; (e) $Az = 140^\circ, EL = 40^\circ$; (f) $Az = 180^\circ, EL = 40^\circ$; (g) $Az = 180^\circ, EL = 20^\circ$; (h) $Az = 180^\circ, EL = 0^\circ$

带。这种滤波器称为带通滤波器。

正频率轴是在单位圆的上半部(见图 4.5), 在图 4.5 中, 正频率轴为不可见, 因为从当前的观察角来观察这个橡胶膜面时, 它是基底。当然, 负频率轴在 $0^\circ \sim 180^\circ$ 角之间, 我们可以看得见。

为了更好地观察振幅响应, 假设我们沿着这个膜面“走”, 看看被隐藏的正频率轴的一面。从图 4.8(c)到图 4.8(f)所示的是, 以方位角分别为 $60^\circ, 100^\circ, 140^\circ$ 和 180° 角顺时针方向旋转的图形, 在图 4.8(f)中, 正频率轴由虚轴左边的 DC 移到左边的 $f_s/2$ 处。

图 4.8(g)和图 4.8(h)所示的是坡度为 20° 和 0° 的倾斜情况。图 4.8(h)所示的图形提供了一个直接观察面, 水平轴代表了 z 平面的边缘。这种观察告诉我们, 在频率为 $f_s/8$, 幅角为 45°

极点处,该滤波器振幅响应有一峰值输出。

振幅响应在图的左边和右边变成0,相当于零点位于DC和 $f_s/2$ 处。这个图是单位圆利用幅值标度环形映射的。

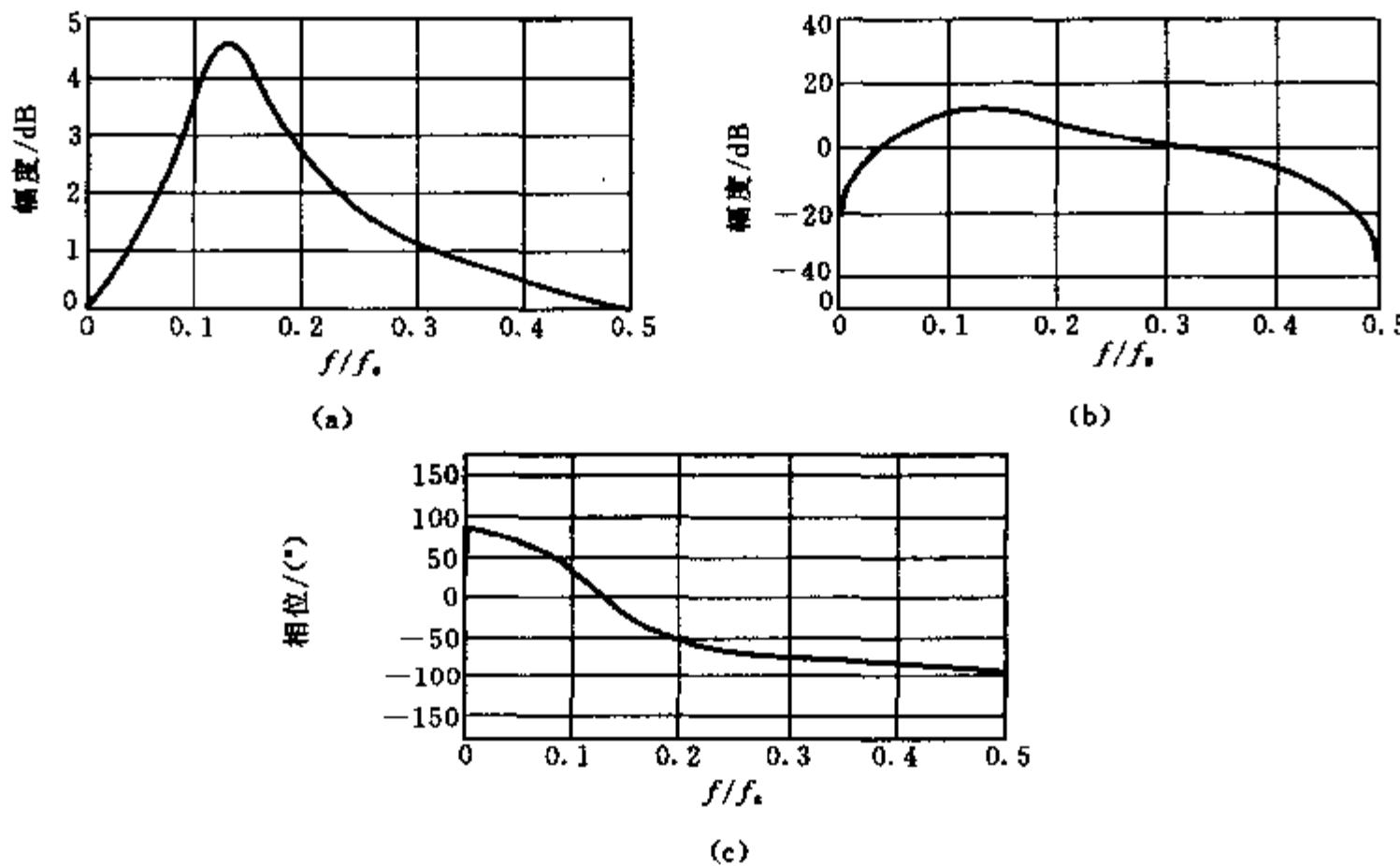


图 4.9 图 4.8 所示的滤波器的频率响应

(a)线性振幅模式下的振幅响应;(b)分贝振幅模式下的振幅响应;(c)相位响应

为了从频率响应得到响应曲线的条件,先计算一下振幅响应,把 $z=e^{j\omega T}$ 代入式(4.44)的传递函数中,得

$$H(\omega T) = \frac{1 - e^{-j2\omega T}}{1 - 1.0605e^{-j\omega T} + 0.5625e^{-j2\omega T}} \quad (4.51)$$

现变换一下函数 $e^{j\omega T}$ 的关系:

$$e^{j\omega T} = \cos(\omega T) + j\sin(\omega T) \quad (4.52)$$

得

$$H(\omega T) = \frac{1 - \cos(2\omega T) + j\sin(2\omega T)}{1 - 1.0605\cos(\omega T) + j1.0605\sin(\omega T) + 0.5625\cos(2\omega T) - j0.5625\sin(2\omega T)} \quad (4.53)$$

将实部和虚部分别写在一起,得

$$H(\omega T) = \frac{1 - \cos(2\omega T) + j\sin(2\omega T)}{1 - 1.0605\cos(\omega T) + 0.5625\cos(2\omega T) + j[1.0605\sin(\omega T) - 0.5625\sin(2\omega T)]} \quad (4.54)$$

上式可化简为:

$$H(\omega T) = \frac{A + jB}{C + jD} \quad (4.55)$$

式中,

$$A = 1 - \cos(2\omega T)$$

$$B = \sin(2\omega T)$$

$$C = 1 - 1.0605\cos(\omega T) + 0.5625\cos(2\omega T)$$

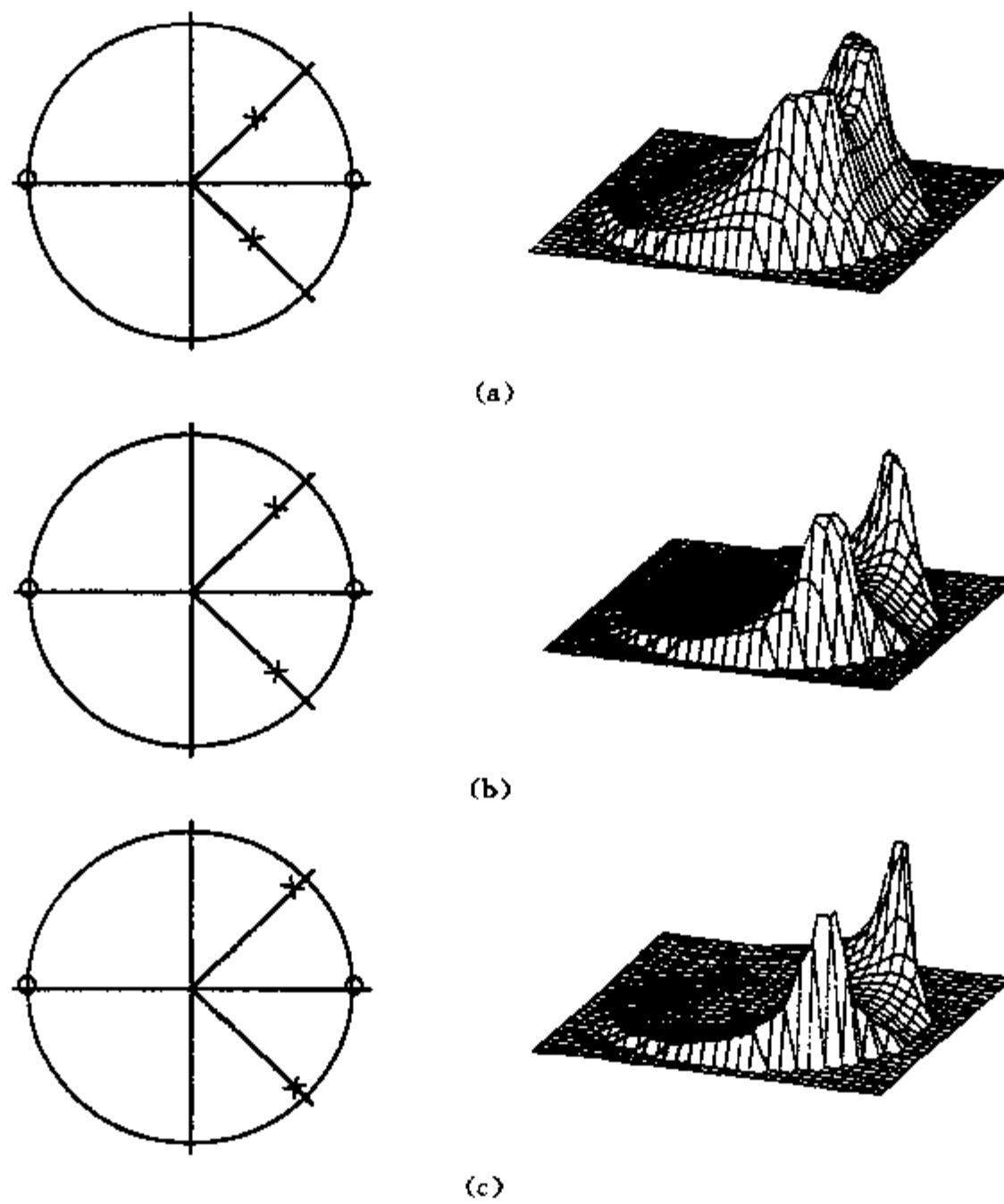


图 4.10 带通滤波器的零-极点图和橡胶膜面图

极点处角频率为 $f_s/8$; (a) $\gamma=0.5$; (b) $\gamma=0.75$; (c) $\gamma=0.9$

$$D = 1.0605 \sin(\omega T) - 0.5625 \sin(2\omega T)$$

为了得到滤波器的振幅响应和相位响应, 将分子分母同时乘以分母的共轭复数 $C-jD$, 即

$$H(\omega T) = \frac{A + jB}{C + jD} \cdot \frac{C - jD}{C - jD} = \frac{(AC + BD) + j(BC - AD)}{C^2 + D^2} \quad (4.56)$$

幅值响应为:

$$|H(\omega T)| = \frac{\sqrt{(AC + BD)^2 + (BC - AD)^2}}{C^2 + D^2} \quad (4.57)$$

这个响应曲线如图 4.9(a)所示, 横坐标为 f/f_s , 范围为 $(0 \sim 0.5)f_s/2$ 。

与图 4.8(h)所示的图形相比较可知, 来自单位圆的环形映射响应可得到相同的信息。图 4.9(b)所示的表明振幅响应曲线与普通的分贝曲线一样。

图 4.10 所示的是沿径向的极点怎样影响橡胶膜面的突起的情况。极点向单位圆周移近, 则峰值变得陡峭。极点离单位圆周越近, 滤波器向上倾斜的峰就越陡峭。

如图 4.9(c)所示, 滤波器的相位响应表达式为:

$$\angle H(\omega T) = \arctan \left[\frac{BC - AD}{AC + BD} \right] \quad (4.58)$$

图 4.11(a)所示的是振幅响应的叠加,对于这三个极点配置中的任一个,极点离单位圆最近的滤波器有最高点,因而在 3dB 以外有最好的抗频率干扰性能。

当极点向单位圆的中心移动时,陡峭程度变得越来越平缓。图 4.11(b)所示的是三个滤波器的相位响应。当极点逐渐向单位圆的中心移动时,相位响应渐渐直线化。由第一章可知,当所有的极点均在单位圆内时,相位响应就完全分段成线性化了。

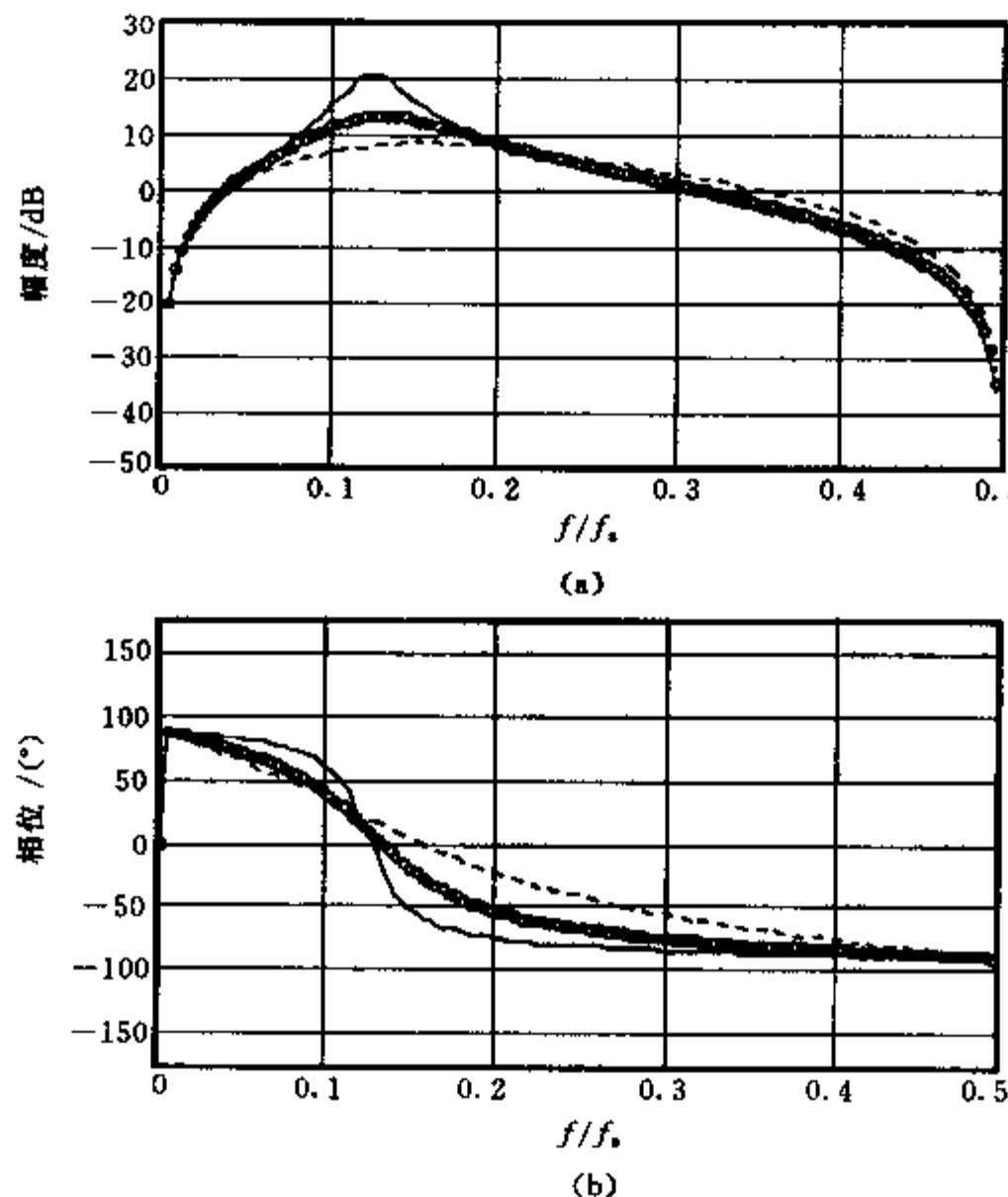


图 4.11 图 4.10 所示的带通滤波器的幅值响应和相位响应

极点位于 $\pm 45^\circ$ 角上(即 $f/f_s=0.125$)；

实线: $\gamma=0.9$; 圆圈: $\gamma=0.75$; 点画线: $\gamma=0.5$

因此,这种橡胶膜片的概念可帮助我们认识数字滤波器的特性曲线,它在数字滤波器的设计中成为一种重要的工具。

复习思考题

- 4.1 数字滤波器与模拟滤波器有什么区别?
- 4.2 在一个单位脉冲函数作用下,输出序列为{1,0,0,2,0,1},求数字滤波器的传递函数表达式。
- 4.3 根据传递函数 $H(z)=\frac{1}{4} + \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2}$,画出滤波器的零-极点图。

- 4.4 若有一个滤波器,它在单位圆的 30° 角处有零点,怎样用这个滤波器作为陷波滤波器去除掉 60Hz 的噪声? 用它又怎样抑制不同频率?
- 4.5 当阶跃函数幅值为 5(即 5,5,5,5,...)时,它的 z 变换是什么?
- 4.6 以指数函数 e^{-nT} 作为滤波器的输入,推导其离散化模式的 z 变换。
- 4.7 将单位脉冲函数作为滤波器的输入,产生序列 {1, -2, 0, 0, ...},若将单位阶跃函数作为滤波器的输入,试问输出是什么?
- 4.8 数字滤波器的传递函数为 $H(z)=z^{-1}+6z^{-4}-2z^{-7}$,它的输出 $y(nT)$ 的差分方程是什么?
- 4.9 当数字滤波器的输入为单位脉冲函数 (1, 0, 0, ...) 时,输出序列为 (1, 2, -3, 0, 0, 0, ...). 若它的输入为单位阶跃函数时,则输出序列是多少?
- 4.10 把单位脉冲函数施加于一数字滤波器,产生输出序列 {3, 2, 3, 0, 0, 0, ...},若把单位阶跃函数施加于同一滤波器,将产生什么结果?
- 4.11 滤波器的 z 变换为 $H(z)=2-2z^{-4}$. 它的幅频特性、相频特性和差分方程分别是什么?
- 4.12 有一个采样频率为 800 次/s 的传递函数: $H(z)=(1-0.5z^{-1})(1+0.5z^{-1})$,若把频率为 200Hz、振幅为 4 的正弦波作为输入,则输出信号的峰值多大?
- 4.13 将单位脉冲函数作用于数字滤波器,产生输出序列 {1, 2, 3, 4, 0, 0, ...},用单位阶跃函数作用于同一滤波器将产生什么结果?
- 4.14 有一采样频率为 600 次/s 的滤波器,其传递函数为 $H(z)=1-2z^{-1}$,以正弦信号 $10\sin(628t)$ 作为输入,问输出信号的幅值应为多大?

第五章 有限冲激响应滤波器

有限冲激响应滤波器(FIR)的单位冲激响应为有限项,而无限冲激响应滤波器(IIR),在单位冲激函数作用于其输入端时,将产生无限项的输出。FIR 滤波器的实现一般是非递推的,这意味着输出数据的计算中不含反馈项。滤波器的输出仅取决于当前的输入和过去的输入。这种性质对数字滤波器设计和应用具有重要的意义。本章将讨论几种典型的用于 ECG 实时处理的 FIR 滤波器,并对若干通用的 FIR 设计方法作出评述。

5.1 FIR 滤波器的特性

5.1.1 有限冲激响应

有限冲激响应是指对瞬态分量或初始状态的响应。不同瞬态分量或初始状态对滤波器输出的影响最终将消失。图 5.1 所示的是用非递归滤波器实现的 FIR 滤波器的信号流图。该滤波器仅是延时层的一组权系数。其单位冲激响应等价于这些权系数。

其差分方程为: $y(nT) = \sum_{k=0}^N b_k x(nT - kT)$ (5.1)

其传递函数为: $H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}$ (5.2)

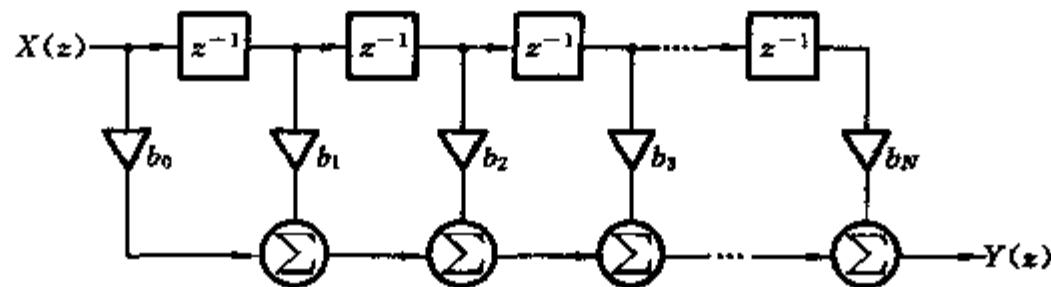


图 5.1 N 阶 FIR 滤波器的输出为延时线上
存储寄存器中数值的加权和

5.1.2 线性相位

在生物医学信号处理的许多应用中,通过滤波来保存信号的某些特性是重要的,例如 QRS 波的高度和间期。线性相位滤波器的相位响应是一纯延时,因此相位失真最小。如果某滤波器的频率响应 $H(e^{j\theta})$ 可表述如下:

$$H(e^{j\theta}) = H_1(\theta)e^{-j(\alpha\theta+\beta)} \quad (5.3)$$

这里 $H_1(\theta)$ 是一个实偶函数,则称其具有线性相位特性。由于 $H(e^{j\theta})$ 的相位为:

$$\angle H(e^{j\theta}) = \begin{cases} -\alpha\theta - \beta & (H_1(\theta) > 0) \\ -\alpha\theta - \beta - \pi & (H_1(\theta) < 0) \end{cases} \quad (5.4)$$

因此 FIR 滤波器很容易被设计为具有线性相位特性的滤波器。线性相位可用四种方法获得,即奇或偶对称与偶或奇长度的组合。

$$\left. \begin{array}{l} h(N-1-k) = h(k) \\ h(N-1-k) = -h(k) \end{array} \right\} \begin{array}{l} (\text{偶对称}) \\ (\text{奇对称}) \end{array} \quad (0 \leq k \leq N) \quad (5.5)$$

5.1.3 稳定性

因为非递推滤波器未采用反馈回路,故除了那些位于 $z=0$ 的极点外再没有别的极点。不可能有极点位于单位圆之外,这意味着它是稳定的。只要滤波器的输入是有限的,其输出也必将是有限的。这不但使滤波器设计变得简单,而且使 FIR 滤波器尤其适用于自适应滤波。自适应滤波器的系数将作为输入数据的函数随着输入数据的不同而改变。自适应滤波器将在第八章中进行讨论。

5.1.4 期望的有限字长寄存器效应

当数据从模拟形式向数字形式转换时,由于存储字节有限,因此,将会丢失一些信息。同样,当计算滤波器的系数时,实现的数字只能是理想数值的近似。这种数字存储所导致的近似性称为有限字长寄存器效应。有限字长寄存器效应对滤波器设计具有极大的消极影响,本书对此论题不作详细探讨。这些影响包括量化误差、舍入噪声、有限圆、条件稳定和系数敏感性等。在 FIR 滤波器中,由于这些误差未反馈回滤波器,因此,这些影响相对于 IIR 滤波器来说要小得多,且易于分析。

5.1.5 设计简单

以上所有特性都使得 FIR 滤波器的设计变得简单。有许多 FIR 滤波器的直接设计方法能够满足任意频率响应和相位响应指标,例如窗设计法或频率取样法。有许多软件包能够自动进行 FIR 设计过程,而且往往能求得在某方面最优的滤波器。

5.1.6 FIR 滤波器的实现

实现 FIR 滤波器有三种方法。最普通的方法是直接迭代法。在这种方法中,滤波器的单位冲激响应序列是用当前输入值和过去输入值迭代计算得来的。FIR 滤波器将通带外的信号逐渐减弱(因为它们具有慢收敛特性)。因为 FIR 滤波器比同样字长的 IIR 滤波器的收敛慢得多,而应用中要求迅速收敛,故 FIR 滤波器的阶数要很大才行。对于高阶滤波器,直接迭代法变得无能为力。

对于字长大于 30 的 FIR 滤波器,“快速迭代”法提供了计算存储技术。这种方法具有时域乘法运算,频域双重迭代的计算量小的特点。快速迭代的步骤是首先将一组数据进行快速傅里叶变换 FFT,再与单位冲激响应序列的 FFT 变换结果相乘,最后进行 FFT 反变换。下一组数据再重复这一处理过程。这种方法将在 11.3.2 小节中详细加以讨论。

实现 FIR 滤波器的第三种方法是包含一个梳状滤波器和一排平行数字谐振器的先进递推法。如果预期的频率响应中有许多系数为零,那么这种方法很适于频率取样设计,而且可用于整系数滤波器,正如第七章中所论述的那样。本章的余下部分仅探讨直接迭代法。

5.2 平滑滤波器

最常碰到的信号处理任务之一是平滑数据以抑制高频噪声。高频噪声源包括 60Hz 电网噪声*, 运动伪迹和量化误差。求几个数据点的平均值的方法是减弱高频噪声的一种简单方法。这种滤波器被称为移动平均滤波器。

5.2.1 海宁滤波器

海宁移动平均滤波器是最简单的平滑滤波器之一, 如图 5.2 所示。正如其差分方程所示, 海宁滤波器计算的是一组加权移动平均值, 其中间数据点的权系数是其余两个数据点的 2 倍:

$$y(nT) = \frac{1}{4}[x(nT) + 2x(nT - T) + x(nT - 2T)] \quad (5.6)$$

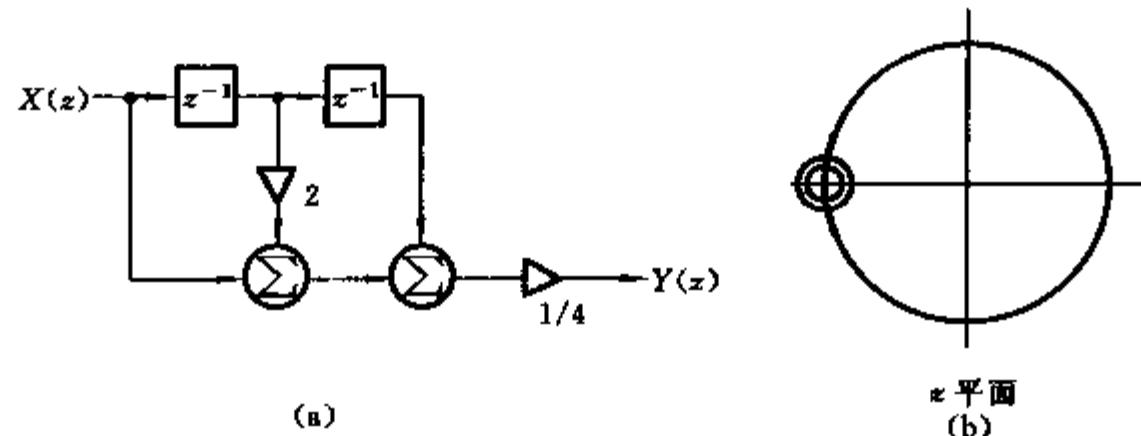


图 5.2 海宁滤波器

(a)信号流图; (b)零-极点分布图

正如 4.5 节中所述,一旦有了差分方程(实现数字滤波器的数值算法),即可确定传递函数。利用离散值和 z 域值间的关系类推,传递函数能够完全代表该滤波器的性能。

我们知道, $x(nT)$ 和 $y(nT)$ 是与当前采样时刻相联系的输入和输出序列中的点, 它们分别对应于无延时 z 域数值 $X(z)$ 和 $Y(z)$ 。同样, $x(nT - T)$ 是前一采样点的输入数值, 对应于滞后一个采样点的 z 域输入值, 或 $X(z)z^{-1}$ 。若将输出的 $Y(z)$ 视为输入 $X(z)$ 的函数, 则其 z 变换的方程式为:

$$Y(z) = \frac{1}{4}[x(z) + 2x(z)z^{-1} + x(z)z^{-2}] \quad (5.7)$$

图 5.2(a)所示的是用功能块直接实现方程中的各项的流程图。 z 的 -2 次幂决定了设计中需要 2 个延时器。为了与因子 2 和 $\frac{1}{4}$ 相乘, 必须有两个乘法器, 还需要两个加法器来将各项相加。此方程的传递函数为:

$$H(z) = \frac{1}{4}[1 + 2z^{-1} + z^{-2}] \quad (5.8)$$

该滤波器有两个零点, 均位于 $z = -1$; 有两个极点, 均位于 $z = 0$ (见 4.6 节, 如何求取极点和零点)。图 5.2(b)所示的是零-极点分布图。请注意极点是隐含的: 因为它们对幅值响应的所有频率影响相同, 故图中未画出。

* 美国的电网频率为 60Hz, 中国的电网频率为 50Hz。

在式(5.8)中用 $e^{j\omega T}$ 代替 z , 可得滤波器的幅频与相频特性:

$$H(\omega T) = \frac{1}{4}[1 + 2e^{-j\omega T} + e^{-j\omega T}] \quad (5.9)$$

直接代入三角关系中, 有

$$e^{j\omega T} = \cos(\omega T) + j\sin(\omega T) \quad (5.10)$$

而在代换之前, 将式(5.9)稍加变换可令表达式大为简化, 如提取 e 的幂次作为因子, 则提取结果有两个类似的幂次项, 它们的幂次相等, 符号相反, 即

$$H(\omega T) = \frac{1}{4}[e^{-j\omega T}(e^{j\omega T} + 2 + e^{-j\omega T})] \quad (5.11)$$

现将式(5.10)代入上式圆括弧中的各项, 得到

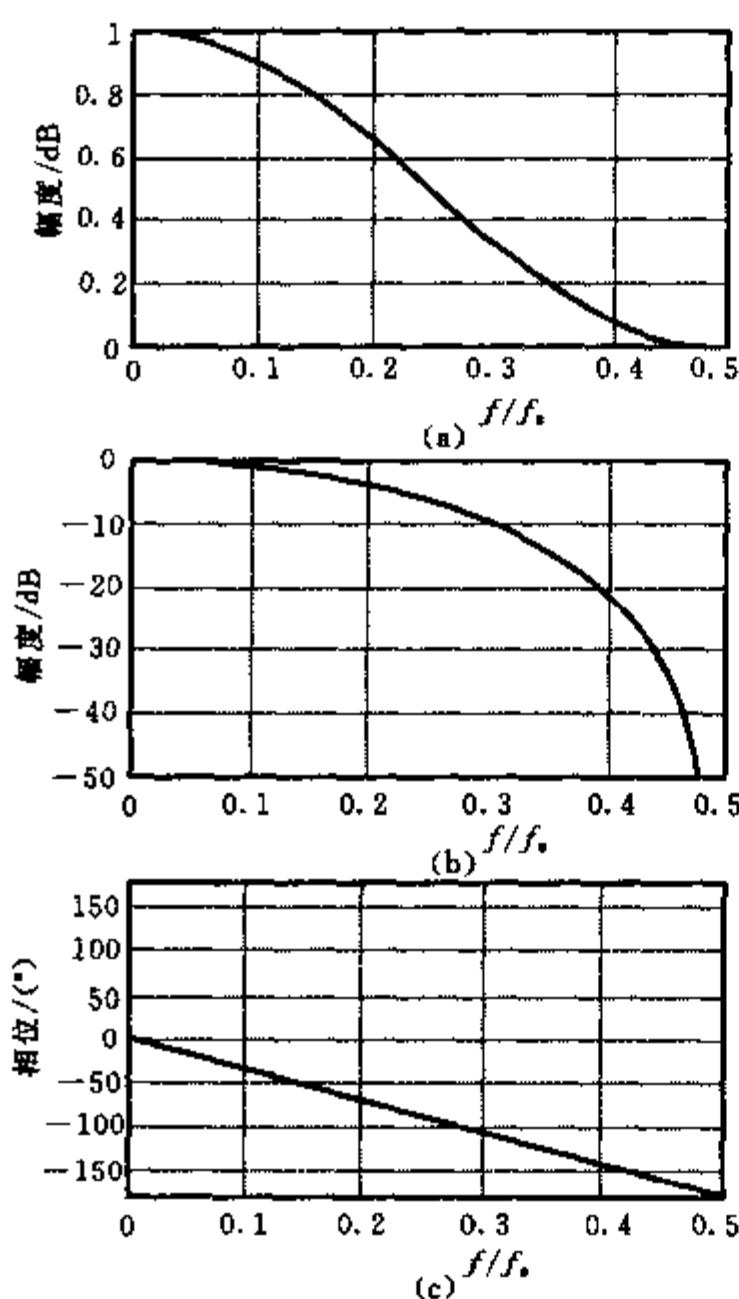


图 5.3 海宁滤波器
(a)频率响应(线性幅值轴);(b)频率响应
(分贝幅值轴);(c)相位响应

记作 x_{nt} 从数组 $idb[\cdot]$ 中得到。计算差分方程, 求得输出值 $y(nT)$ (即程序中的 y_{nt})。将该值存入数组, 代替 $x(nT)$ 的值。然后输入数据变量在延时器中移动。在下次输入之前, 前一采样点的数据 $x(nT-T)$ (程序中记作 $xm1$) 移至输入 $x(nT-2T)$ (即 $xm2$) 中。当前输入 $x(nT)$ 在时间上倒退一个点, 代替 $x(nT-T)$ 。在下一个 for 循环中, 重新获得 $x(nT)$ 的新值, 重复此过程直至 256 组数值全部被处理为止。滤波后的输出波形留在数组 $idb[\cdot]$ 中。

$$\begin{aligned} H(\omega T) = & \frac{1}{4}\{e^{-j\omega T}[\cos(\omega T) + j\sin(\omega T)] \\ & + 2 + \cos(\omega T) - j\sin(\omega T)\} \end{aligned} \quad (5.12)$$

消去 $\sin(\omega T)$ 项后, 得

$$H(\omega T) = \frac{1}{4}([2 + 2\cos(\omega T)]e^{-j\omega T}) \quad (5.13)$$

这是 $Re e^{j\theta}$ 形式, 这里 Re 是实部, θ 是相角。因此海宁滤波器的模是 $|R|$, 或

$$|H(\omega T)| = |\frac{1}{2}[1 + \cos(\omega T)]| \quad (5.14)$$

图 5.3(a)所示的是线性坐标下的余弦波幅值响应曲线, 而图 5.3(b)所示的是更为熟悉的分贝坐标图。读完本书, 就可以学会使用分贝坐标图。将海宁滤波器的输出作为另一相同滤波器的输入可使其相关慢收敛项迅速收敛。将多个滤波器串连在一起所形成的滤波器称为级联滤波器。图 5.3(c)所示的是线性相位响应等于相角 θ , 或

$$\angle H(\omega T) = -\omega T \quad (5.15)$$

海宁滤波器的实现是通过一段计算机程序来完成的。图 5.4 所示的是为离线应用的一段 C 语言程序, 这里首先用一个 ADC 来采样数据并将其放入一数组中。这段程序直接计算滤波器的差分方程式(5.6)。在 for 循环内, $x(nT)$ 的值(程序中

记作 x_{nt})从数组 $idb[\cdot]$ 中得到。计算差分方程, 求得输出值 $y(nT)$ (即程序中的 y_{nt})。将该值存入数组, 代替 $x(nT)$ 的值。然后输入数据变量在延时器中移动。在下次输入之前, 前一采样点的数据 $x(nT-T)$ (程序中记作 $xm1$) 移至输入 $x(nT-2T)$ (即 $xm2$) 中。当前输入 $x(nT)$ 在时间上倒退一个点, 代替 $x(nT-T)$ 。在下一个 for 循环中, 重新获得 $x(nT)$ 的新值, 重复此过程直至 256 组数值全部被处理为止。滤波后的输出波形留在数组 $idb[\cdot]$ 中。

```

/* Hanning filter

Difference equation
    Y(nT) = (x(nT) + 2*x(nT-T) + x(nT-2T))/4

C language implementation equation:
    ynt = (xnt+2*xm1+xm2)/4;

*/
main()
{
    int i,xnt,xm1,xm2,ynt,idb[256];
    xm2=0;
    xm1=0;
    for(i=0;i<=255;i++)
    {
        xnt=idb[i];
        ynt=(xnt+2*xm1+xm2)/4;
        idb[i]=ynt;
        xm2=xm1;
        xm1=xnt;
    }
}

```

图 5.4 实现海宁滤波器的 C 语言程序
数据由 ADC 预采样并存储在数组 idb[] 中。滤波后信号存放在数组 idb[] 中

由于所有的系数均为整数而且可用二进制的移位代替乘法，在实时应用中采用海宁滤波器尤为有效。图 5.5 所示的是一段实时海宁滤波器程序。在这段程序中，输出值 $y(nT)$ 的计算必须在一个采样间隔 T 内完成。这就是说，当 ADC 得到一个新输入数据后，在下一个 ADC 输入前必须计算出输出值，否则滤波器将无法跟上采样率，从而无法实时运行。

程序中，从 ADC 中采样，计算结果，并将滤波后数据送入 DAC，这些都在一个 for() 循环中完成。wait() 函数用于等待由 ADC 时钟引起的中断的到来。一旦发生中断，即用 adget() 函数采样数据得到 x_{nt} ，然后计算海宁滤波器的差分方程。计算中采用 C 语言的移位操作来有效地完成乘法运算。 $xm1 \ll 1$ 表示二进制的左移一位，对应于乘 2 运算； $(xnt+xm1 \ll 1 + xm2) \gg 2$ 表示二进制的右移两位，对应于被 4 除。

计算所得输出值由函数 daput() 送给 DAC。与前段程序一样，输入数据在延时器中移动。对于下一输入，前一采样点数据 $x(nT-T)$ （程序中记作 $xm1$ ）移至前两采样点，成为 $x(nT-2T)$ （即 $xm2$ ）。当前输入 $x(nT)$ （记作 xnt ）在时间上倒退一点，代替 $x(nT-T)$ （即 $xm1$ ）。然后 for 循环重复执行 wait() 函数直到一中断信号发出，再由 adget() 函数得到新的采样数据作为 $x(nT)$ 的新值为止。

5.2.2 最小二乘法多项式拟合的平滑

这类滤波器的输入数据的奇数 $(2L+1)$ 项在最小二乘法标准上与抛物线拟合，图 5.6(a) 所示的是滤波器的输出是抛物线的中点的情况。写出每一输入点的抛物线方程，得到

$$P(nT + kT) = a(nT) + b(nT)k + c(nT)k^2 \quad k \in [-L, L] \quad (5.16)$$

```

/* Real-time Hanning filter
Difference equation:
Y(nT)=(x(nT)+2*x(nT-T)+x(nT-2T))/4
C language implementation equation:
ynt=(xnt+xm1<<1+xm2)>>2;
*/
#define AD 31;
#define DA 32;
main()
{
    int i,xnt,xm1,xm2,ynt;
    xm2=0;
    xm1=0;
    tmic 2000;      /* Start ADC clock ticking at 2000 μs */
                    /* intervals (2 ms period for 500 sps) */
    for(;;)
    {
        wait();          /* Wait for ADC clock to tick */
        xnt=adget(AD);
        ynt=(xnt+xm1<<1+xm2)>>2;
        daput(ynt,DA);
        xm2=xm1;
        xm1=xnt;
    }
}

```

图 5.5 实现实时海宁滤波器的 C 语言代码

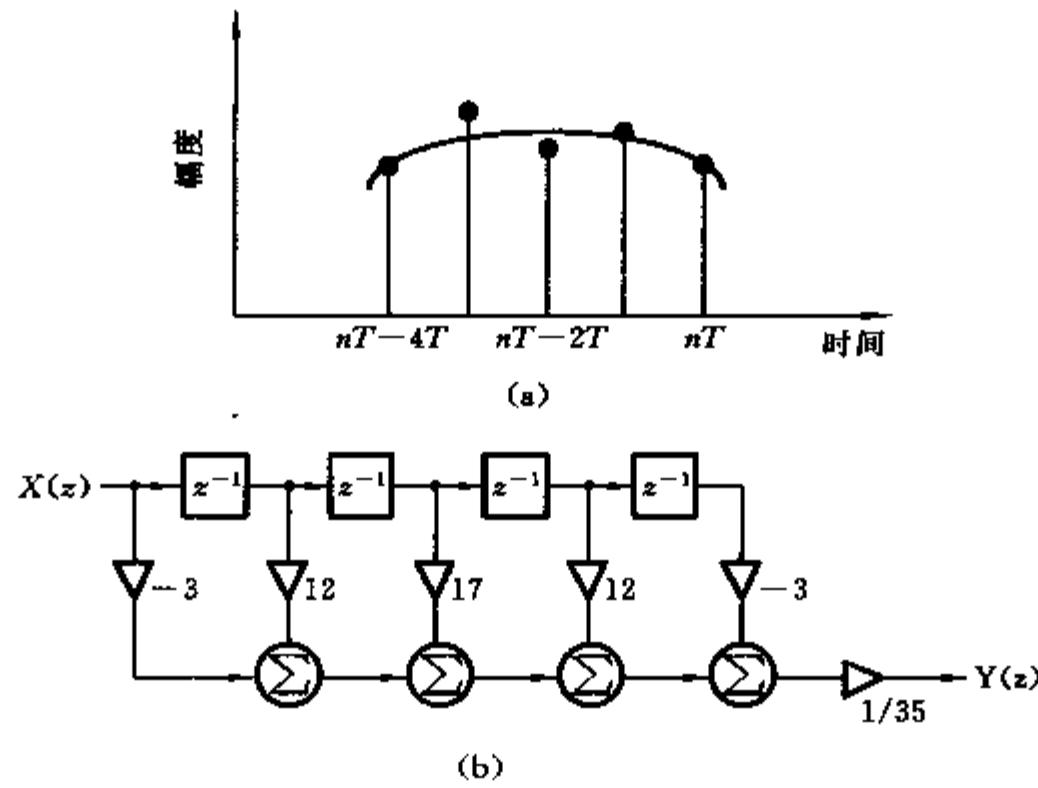


图 5.6 $L=2$ 的多项式平滑滤波器
(a)五采样点拟合抛物线;(b)信号流图

选择 $a(nT)$ 、 $b(nT)$ 和 $c(nT)$ ，使抛物线与输入数据间的方差最小，则输入数据将拟合抛物线。令方差分别对 $a(nT)$ 、 $b(nT)$ 和 $c(nT)$ 的偏导为零，得到一组关于 $a(nT)$ 、 $b(nT)$ 、 $c(nT)$ 、 k 和

$P(nT-kT)$ 的方程组,解之得到 $a(nT)$ (抛物线在 $k=0$ 时值)的表达式,此表达式为输入值的函数,表达式的系数是最小平方多项式滤波器的权系数。图 5.6(b)所示的为最小二乘法多项式五点滤波器的信号流图。五点抛物线滤波器的差分方程为:

$$y(nT) = \frac{1}{35} [-3x(nT) + 12x(nT-T) + 17x(nT-2T) \\ + 12x(nT-3T) - 3x(nT-4T)] \quad (5.17)$$

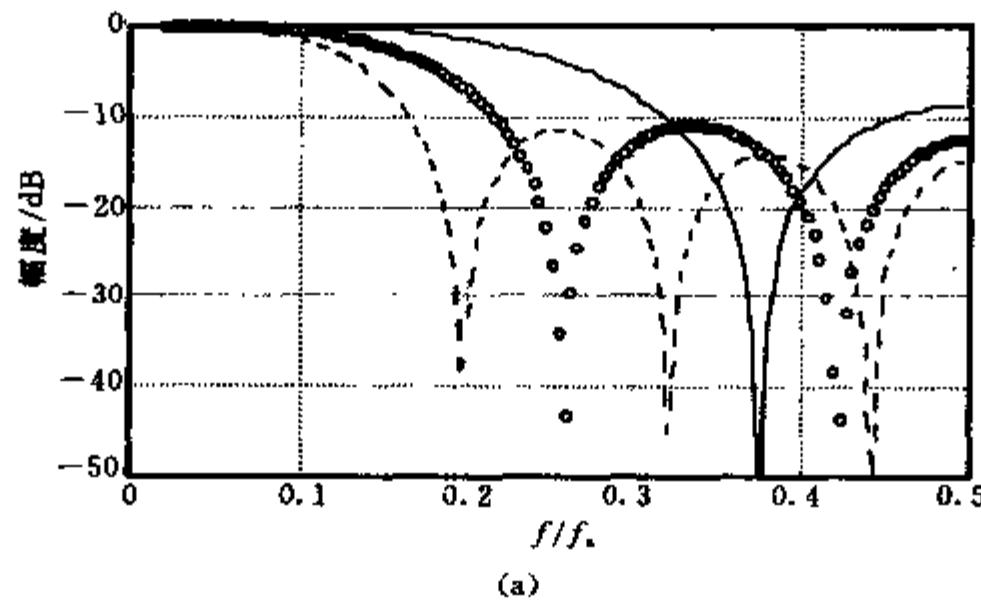
其传递函数为

$$H(z) = \frac{1}{35} [-3 + 12z^{-1} + 17z^{-2} + 12z^{-3} - 3z^{-4}] \quad (5.18)$$

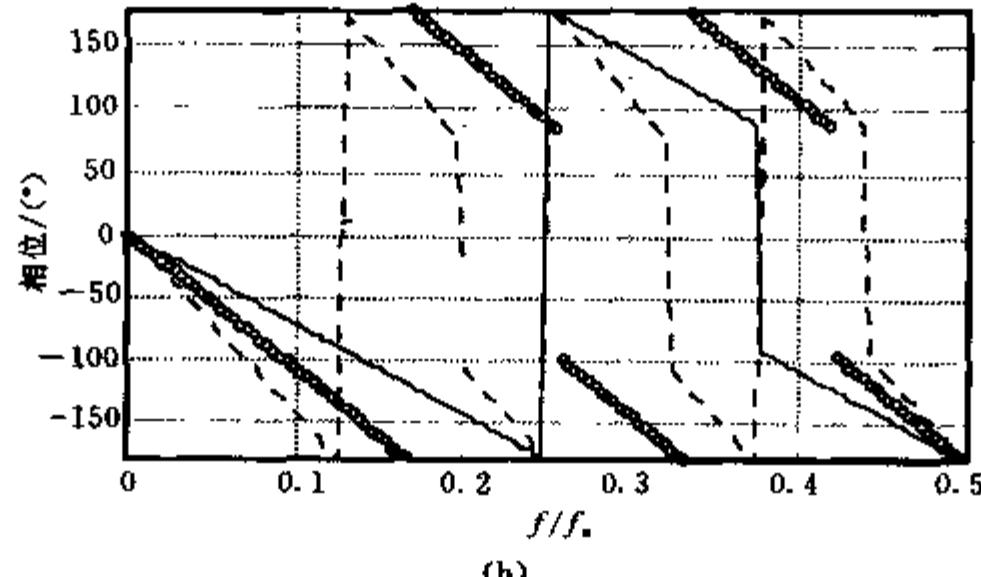
图 5.7 所示的为 $L=2,3,4,5$ 的滤波器权系数。图 5.8 所示的为它们的响应。滤波器的阶数可以根据期望的收敛速度来选择。

L	权重
2	$\frac{1}{35} (-3, 12, 17, 12, -3)$
3	$\frac{1}{21} (-2, 3, 6, 7, 6, 3, -2)$
4	$\frac{1}{231} (-21, 14, 39, 54, 59, 54, 39, 14, -21)$
5	$\frac{1}{429} (-36, 9, 44, 69, 84, 89, 84, 69, 44, 9, -36)$

图 5.7 多项式平滑滤波器的权系数



(a)



(b)

图 5.8 多项式平滑滤波器
(a)幅值响应; (b)相位响应
实线: $L=2$; 圆圈: $L=3$; 虚线: $L=4$

5.3 陷波滤波器

普通生物医学信号处理问题包括从信号中移去特殊频率或频率范围(如 60Hz 或 50Hz)的噪声,以便让更高或更低的频率可无衰减地通过。完成这一任务的滤波器称为陷波(或带阻、或带抗)滤波器。

为了从信号中完全去除指定频率噪声,一种简单的方法是在单位圆上对应于该频率的位置设置一个零点。例如,若采用 180 次/s 的采样率,则位于 $2\pi/3$ 处设置的零点,可以抑制信号中 60Hz 的电源频率噪声。差分方程为:

$$y(nT) = \frac{1}{4}[x(nT) + x(nT - T) + x(nT - 2T)] \quad (5.19)$$

滤波器零点为:

$$z = -0.5 \pm j0.866 \quad (5.20)$$

其幅值和相位响应分别为:

$$|H(\omega T)| = \left| \frac{1}{3} [1 + 2\cos(\omega T)] \right| \quad (5.21)$$

$$\angle H(\omega T) = -\omega T \quad (5.22)$$

图 5.9 所示的是这种滤波器的设计方案及其性能。这种滤波器的相关慢收敛性使得高于 60Hz 的信号也被极大地削弱了。

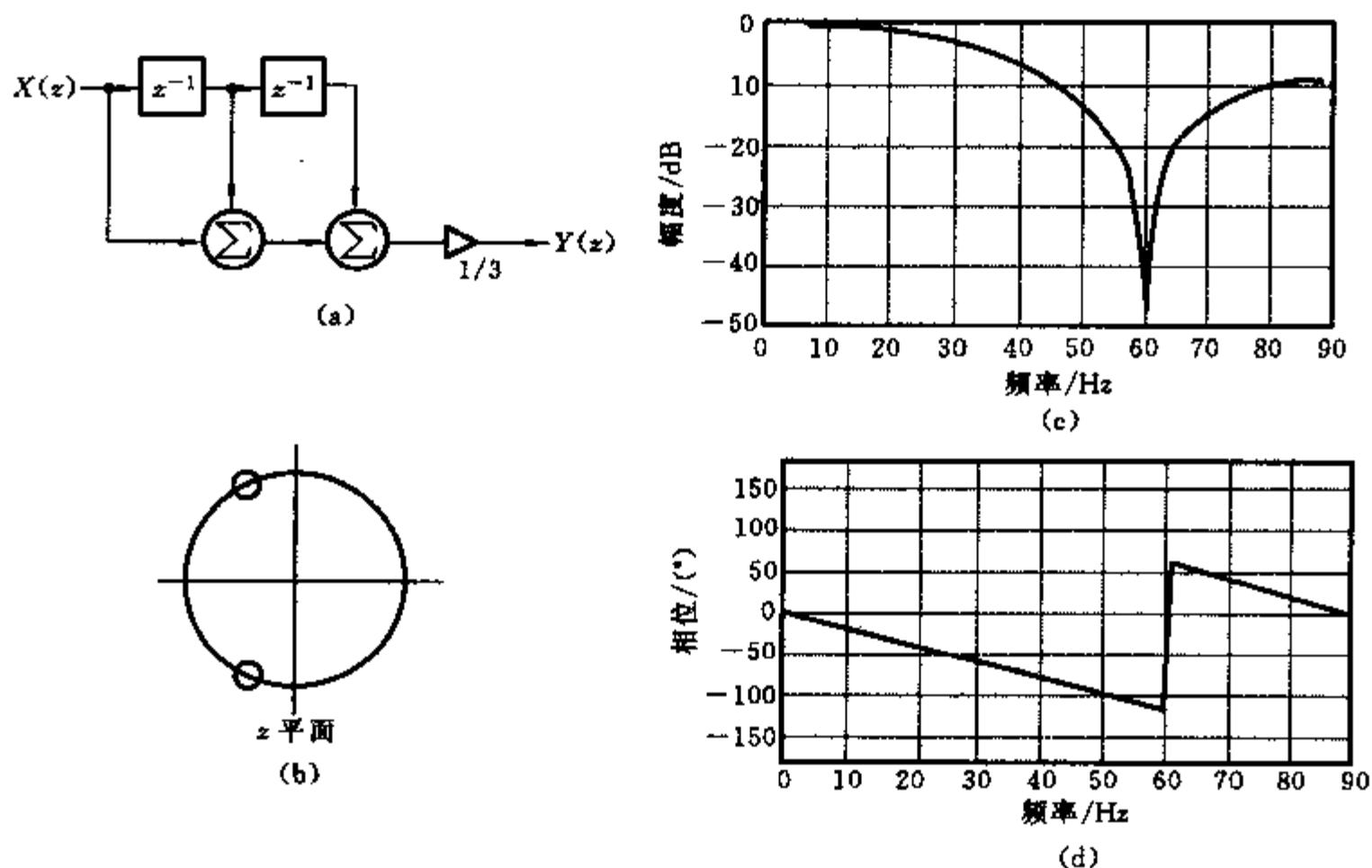


图 5.9 60Hz 陷波滤波器
(a)信号流图; (b)零-极点分布图;
(c)频率响应; (d)相位响应

5.4 求 导 数

实际求导函数的响应随着频率的增加而线性增长。而对于数字微分，由于频率响应的周期性，要得到这样的响应是不可能的。本节所讨论的方法是在计算的复杂性，与真实导数的接近程度和高频噪声的剔除之间折衷考虑的方法。图 5.10 所示的是三种不同的微分算法（两点差分、三点中心差分、最小平方多项式拟合）的信号流图和零-极点分布图。

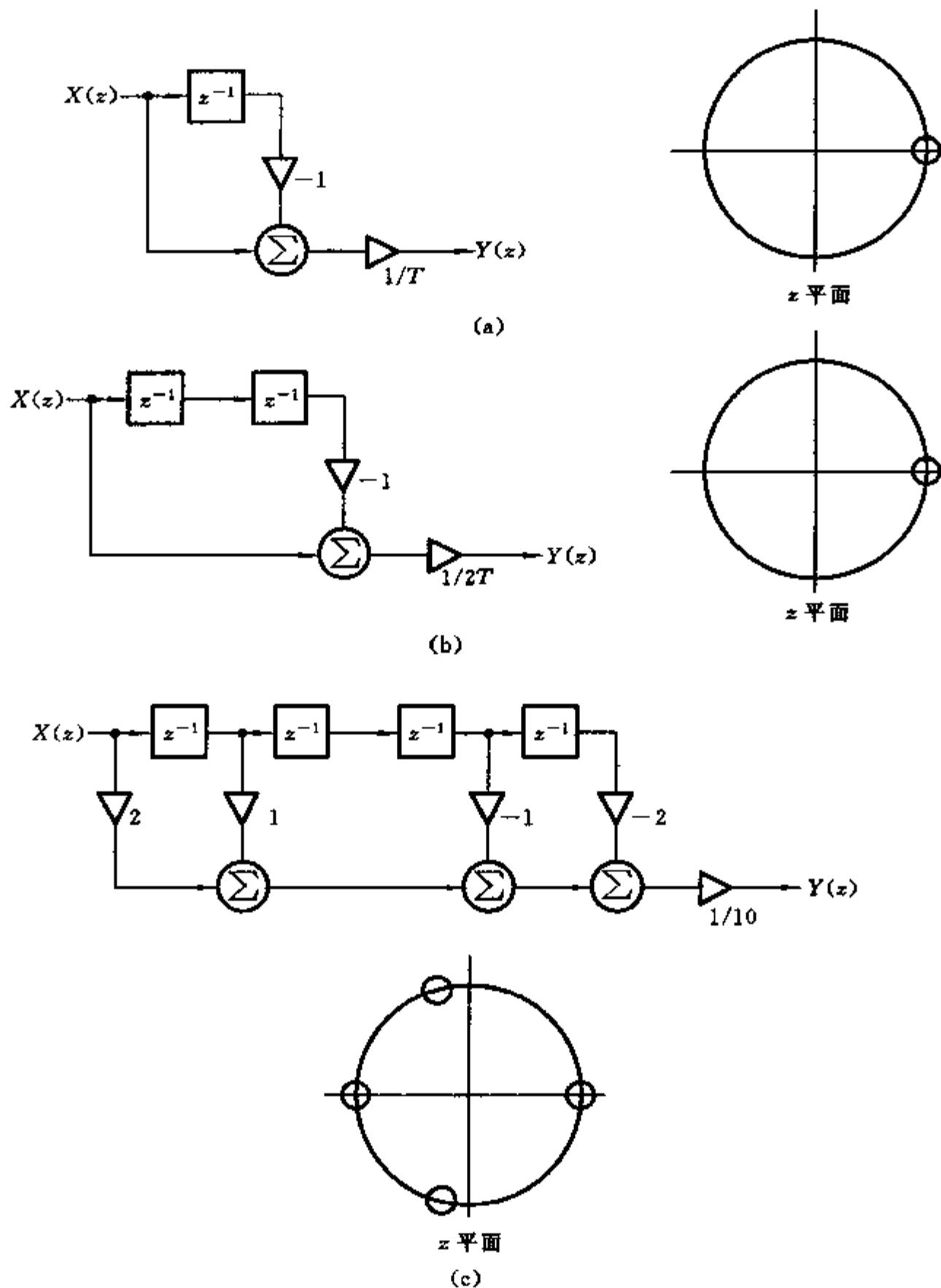


图 5.10 求导算法的信号流图和零-极点分布图
(a)两点差分；(b)三点中心差分；(c)五点最小平方多项式拟合

5.4.1 两点差分算法

求导算法中最简单的是两点差分算法,它在单位圆上的 $z=1$ 处有一个零点。其幅值响应(图 5.11(a))与理想响应非常接近,但由于它在 $f_s/2$ 处并未趋向于零,故将大大增强高频噪声。经常用一低通滤波器紧跟其后,以消除高频噪声的影响。其差分方程为:

$$y(nT) = \frac{1}{T}[x(nT) - x(nT - T)] \quad (5.23)$$

传递函数为:

$$H(z) = \frac{1}{T}(1 - z^{-1}) \quad (5.24)$$

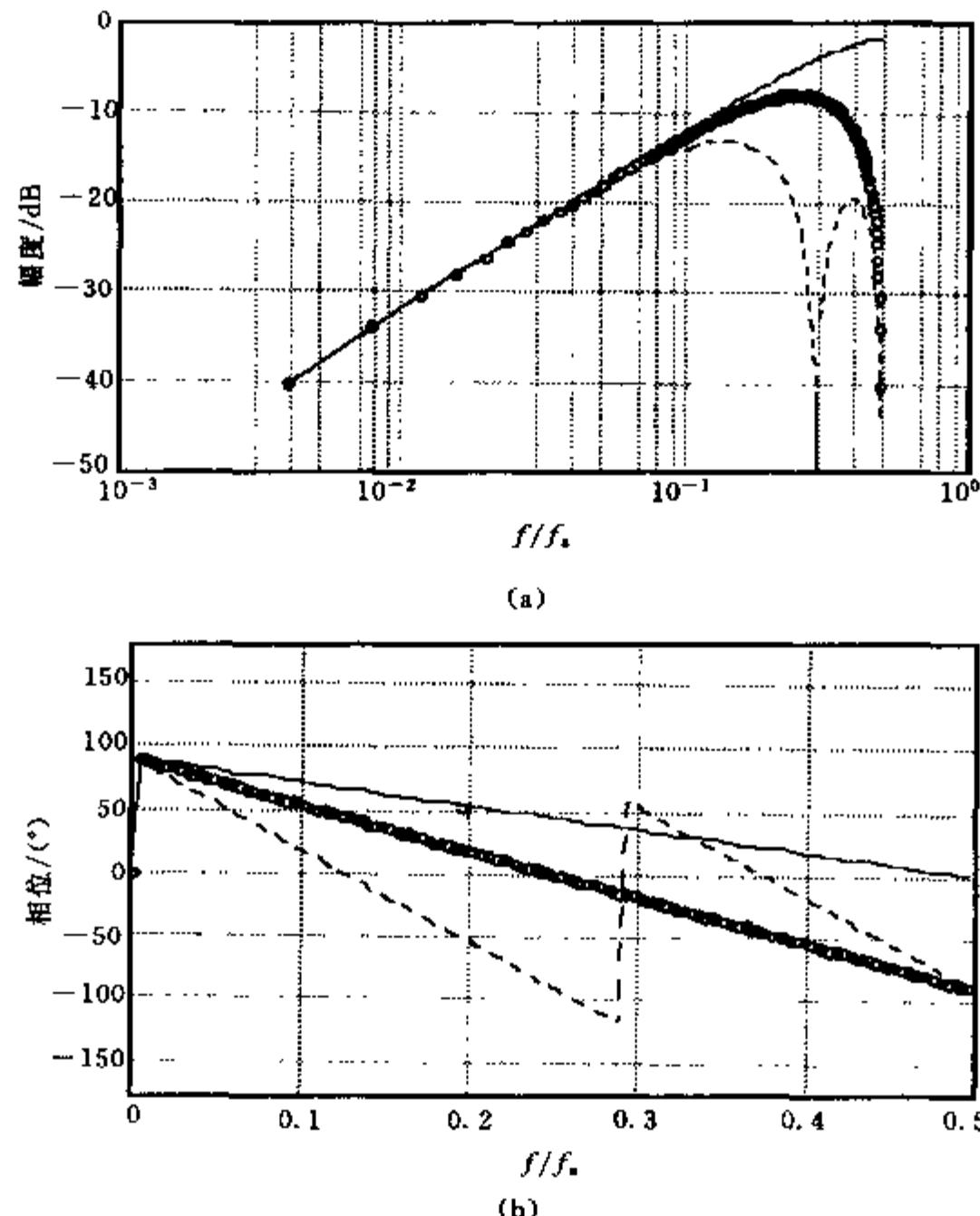


图 5.11 求导
(a)幅值响应; (b)相位响应

5.4.2 三点中心差分算法

三点中心差分算法的零点在 $z=1$ 和 $z=-1$ 处,因此,对于 $f_s/10$ 以上频率的信号,其求导差别较大(图 5.11(a))。但由于其响应在 $f_s/2$ 处趋向于零,故滤波器具有一些内在的平滑作用。其差分方程为:

$$y(nT) = \frac{1}{2T}[x(nT) - x(nT - 2T)] \quad (5.25)$$

传递函数为：

$$H(z) = \frac{1}{2T}[1 - z^{-2}] \quad (5.26)$$

5.4.3 最小二乘法多项式逼近求导

这种滤波器与前面的抛物线平滑滤波器很相似,但它是取抛物线中点的多项式斜率作为导数值的。图 5.12 所示的为 $L=2,3,4,5$ 的滤波器的传递函数系数。图 5.10(c)所示的为 $L=2$ 的滤波器的信号流图和零-极点分布图。请注意该滤波器的零点为 $z=\pm 1$,与三点中心差分法的相同,另外还有两个零点为 $z=-0.25 \pm j0.968$ 。五点抛物线滤波器的差分方程为:

L	权重
2	$\frac{1}{10T}(2,1,0,-1,-2)$
3	$\frac{1}{28T}(3,2,1,0,-1,-2,-3)$
4	$\frac{1}{60T}(4,3,2,1,0,-1,-2,-3,-4)$
5	$\frac{1}{110T}(5,4,3,2,1,0,-1,-2,-3,-4,-5)$

图 5.12 $L=2,3,4,5$ 的最小平方求导近似系数

$$y(nT) = \frac{1}{10T}[2x(nT) + x(nT) - x(nT - 3T) - 2x(nT - 4T)] \quad (5.27)$$

其传递函数为:

$$H(z) = \frac{1}{10T}[2 + z^{-1} - z^{-3} - 2z^{-4}] \quad (5.28)$$

如图 5.11(a)所示,由于抛物线拟合的平滑特性大大削弱了高频段的响应,故响应仅在低频段,这与实际求导是相近的。

实线:两点差分;圆圈:三点中心差分;

虚线: $L=2$ 的最小平方抛物线近似。

图 5.12 所示的是 $L=2,3,4,5$ 的最小平方求导近似系数。

5.4.4 二次求导

图 5.13 所示的为一简单的近似二次求导滤波器。其差分方程如下:

$$y(nT) = x(nT) - 2x(nT - 2T) + x(nT - 4T) \quad (5.29)$$

这种滤波器由两级三点中心差分求导(式(5.26))级联而导出。令幅值放大系数为 1,得传递函数为:

$$H(z) = (1 - z^{-2}) \times (1 - z^{-2}) = 1 - 2z^{-2} + z^{-4} \quad (5.30)$$

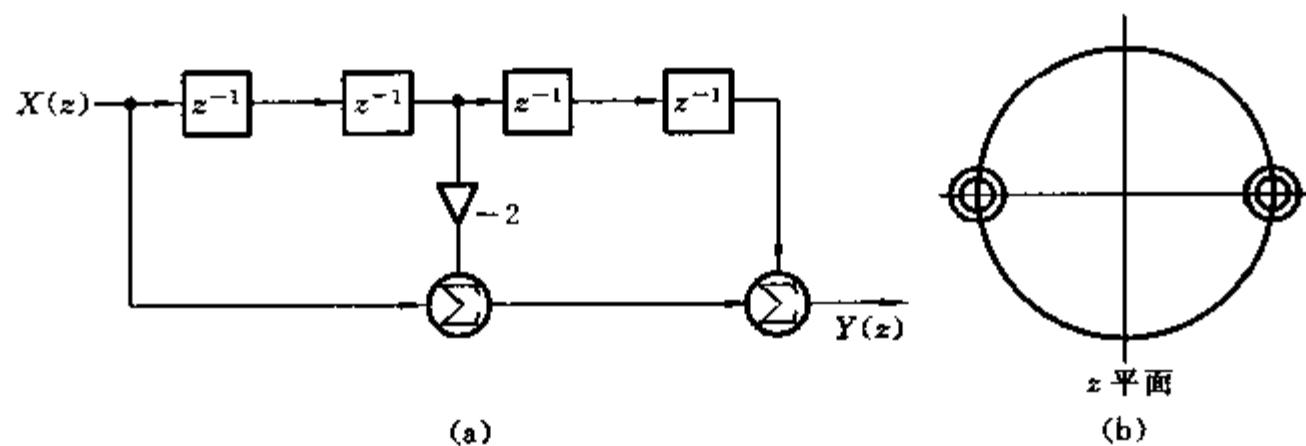


图 5.13 二次求导
(a)信号流图;(b)单位圆图

5.5 窗的设计

通常希望频率响应 $H_d(\theta)$ (连续函数) 的离散 FFT 反变换的结果就是预期单位脉冲序列 $h_d(k)$ (离散函数), 但此序列有无限项, 因此, 物理上不可能实现。窗设计的目的在于选择有限项的实际 $h(k)$, 使其频率响应 $H(e^j\theta)$ 从某种意义上说与 $H_d(\theta)$ 相接近。要使实际频率响应与预期频率响应之间的平均方差最小, 由 Parseval 理论可知, 直接截断 $h_d(k)$, 即可达到这一目标。换句话说, 就是脉冲响应 $h(k)$ 应选择 $h_d(k)$ 的前 N 项。然而, 这样的截断将导致频率响应在急剧转折时的大超调, 如图 5.14 所示, 这被称之为 Gibb 现象。

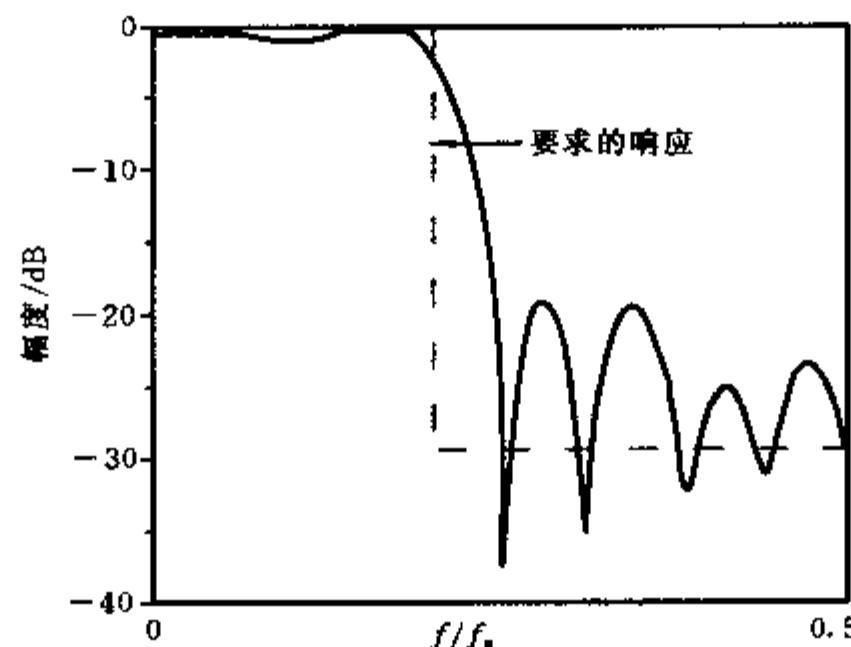


图 5.14 由于截断脉冲响应而在预期频率响应
急剧转折时产生超调的 Gibb 现象

为理解这一现象, 将 $H_d(k)$ 的直接截断结果看成是希望的单位脉冲序列与一矩形窗函数 $W_R(k)$ 相乘而得。由于时域中的乘法对应于频域中的迭加, 因此, 所得 $h(k)$ 的频率响应是预期频率响应与窗函数频率响应的迭加。

例如, 将有限长矩形窗的频率响应看成是一个简单的单位脉冲。预期频率响应与单位脉冲迭加仍等于预期响应。然而, 随着窗宽度变宽, 其频率响应变得不像冲激函数, 且旁瓣变得更加明显。这些旁瓣与希望的频率响应的迭加导致了转折时的超调。

对于矩形窗, 响应函数为:

$$W_R(e^j\theta) = \frac{\sin(\frac{N\theta}{2})}{\sin(\frac{\theta}{2})} \quad (5.31)$$

式中, N 是矩形窗的长度。窗与预期频率响应的迭代有两个重要结果。第一个是窗函数破坏了转折时的预期响应。这种过渡带的破坏拓宽了从通带到止带的过渡宽度。这将导致滤波器主瓣宽度增加。第二个是窗函数导致了滤波器阻带不期望的振荡波纹(被称为窗泄漏或旁带波纹)。

采用非矩形窗函数时,以主瓣的拓宽为代价可减小阻带波纹。已有多种类型的窗函数被详加研究,其中包括矩形窗、海明窗、布莱克曼窗、凯舍尔窗、契比雪夫窗、果西安窗。海宁窗和海明窗形式如下:

$$W_H(k) = W_R(k)[\alpha + (1 - \alpha)\cos(\frac{2\pi}{N}k)] \quad (0 < \alpha < 1) \quad (5.32)$$

式中, $\alpha = 0.54$ 对应于海明窗; $\alpha = 0.50$ 对应于海宁窗。

凯舍尔窗形式如下:

$$W_k(k) = W_R(k)I_0 \frac{\alpha \sqrt{1 - (\frac{k}{M})^2}}{I_0(\alpha)} \quad (5.33)$$

式中, α 允许设计者从矩形窗的极小值到布莱克曼窗的窗宽中选择主瓣宽度, 折衷考虑旁瓣幅值。有关契比雪夫窗和果西安窗的讨论请参阅 Roberts and Mullis(1987)所写的文献。所有这些函数在窗口边缘衰减为零。图 5.15 所示的是这几种窗的性能比较。

窗的类型	旁瓣纹波/dB	主瓣宽度
矩形窗	-13	$4\pi/N$
三角窗	-25	$8\pi/N$
海宁窗	-31	$8\pi/N$
海明窗	-41	$8\pi/N$
布莱克曼窗	-38	$12\pi/N$

图 5.15 各种窗的响应

由于难以预测频率响应被破坏的窗过渡带的长度, 设计窗时, 经常要反复。要设计一个指定频率响应的 FIR 滤波器, 首先应采用 IDTFT 变换, 计算希望得到的响应的单位脉冲序列, 用窗函数加以处理, 然后采用 DFT 变换计算实际响应。如有必要可调整带的边缘或滤波器的阶数并重复以上过程。目前, 已研制出了可利用计算机来完成这一处理的许多算法。这种窗的设计方法可以得到具有严格线性相位响应的滤波器。用窗设计所得的并非都是满足要求的最低阶数滤波器。第八章中还将继续讨论窗的问题。

5.6 频率采样

由于避开了从时域到频域的转换, 频率采样设计法比窗设计法更直接。它直接指定滤波器响应 $H(e^j\theta)$ 的各项与长度为 N 的 $H_d(\theta)$ 在围绕单位圆的统一空间频率上相对应。就像窗设计法一样, 在响应的急剧转折段将发生大超调。通过调节过滤带中的一些自由项可以减少超调。图 5.16 所示的与图 5.14 所示的具有相同预期响应和项数的频率采样设计, 它通过选择一自由变量值来减小阻带波纹。选择自由变量值可使 $H(e^j\theta)$ 和 $H_d(\theta)$ 间的误差按某种准则为最小。

用频率采样法所得到的滤波器一般比用窗设计法所得到的滤波器更加有效。

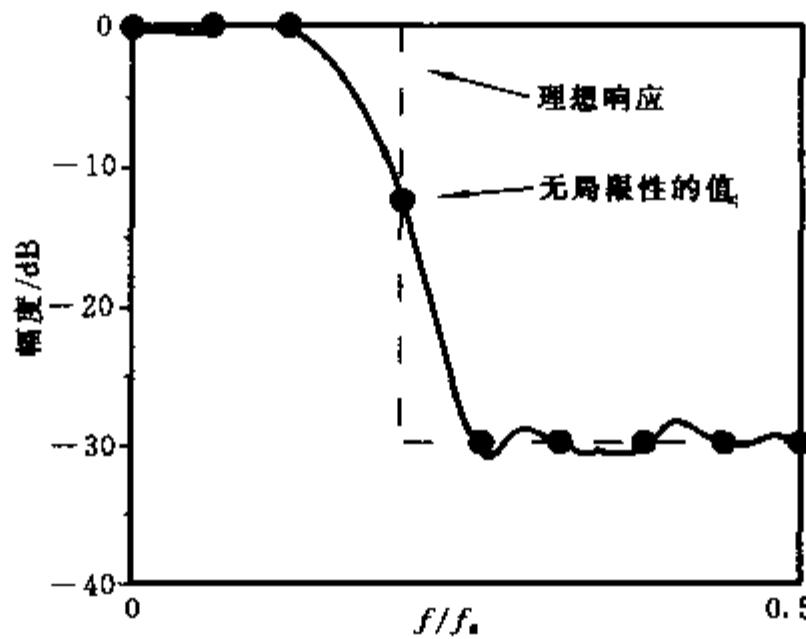


图 5.16 带有一自由项的频率采样设计($N=16$)

5.7 最小设计

窗设计的滤波器在频率响应的急剧转折段将产生大的误差。如果令误差能量为最小，则有

$$\epsilon^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |H_d(\theta) - H_1(e^{j\theta})|^2 d\theta \quad (5.34)$$

如同在窗设计法中的情形一样，在某些应用中使最大误差最小化将更为理想。其思想是有规律地、或以最一般的情况、或以穿越频率响应的某种权方式将误差展开。这比窗设计问题更困难，但目前已研制出可利用计算机来解决的算法。例如瑞梅兹算法，它允许设计者在整个通带和阻带内选择权因子。

5.8 实验:FIR 滤波器的设计

使用者可在 UW DigiScope 软件上做几个 FIR 设计方法的实验，这些设计方法包括直接指定脉冲响应法、频率采样法、窗设计法和设置转移方程零点法等。这个实验研究了一些基本的 FIR 滤波器，如海宁滤波器等，并将指定脉冲响应和设置转移方程零点作为设计工具。关于 UW DigiScope 软件的使用，请参阅附录 D。

5.8.1 平滑滤波器

先用 UW DigiScope 软件的主菜单“(G)enwave”功能生成带有一些随机噪声的 ECG 文件，然后先选择软件主菜单程序中的“(F)ilters”，再选择“(L)oad filter”，最后选择 hanning-fil 来运行海宁滤波器。此(G)enwave 功能装载后立即执行滤波函数。通过选择“(R)un filter”可以观察到两个滤波器级联的效果。此函数通常采用底部通道的数据进行运算。

滤波器 poly2.fil、poly3.fil 和 poly4.fil 分别为 $L=2, 3, 4$ 的最小平方平滑滤波器。为了用 ECG 数据测试这三个滤波器中的一个，首先用“(C)opy data”命令将初始信号从上通道移至下通道，然后应用 ECG 数据，用“(L)oad filter”来装载并运行滤波器。观察每个滤波器的幅值和相位响应，改变 ECG 数据的采样率，将怎样影响滤波器的性能？装载 poly4.fil 滤波器，并用“(M)easure”命令来测量来滤波数据和滤波后数据中 QRS 峰值间隔的差别。可能需要用

“(A)ctire channel”来改变通道进行测量。怎样考虑延时？

5.8.2 求导数

在 ECG 和方波数据上用不同的求导滤波器 deriv2.fil、deriv3.fil、deriv5.fil(分别为两点限制、三点限制和 $L=2$ 的最小平方求导滤波器)做实验。哪种滤波器最不适用于噪声信号？

5.8.3 脉冲响应

首先通过选择“(F)ilters”，然后选择“(D)esign”，接着选择“(F)IR”来产生一个计算二次导数的滤波器。选择“(P)ulse resp”，并指定滤波器字长为 5。输入合适的传递函数方程的系数并观察频率响应。在存储滤波结果后返回主菜单，用“(G)enwave”功能产生一个三角波，然后用“(R)un filter”来观察滤波效果。

用奇字长(偶阶数)和奇对称生成一个满足等式(5.4)的线性相位滤波器，观察其相位响应。

5.8.4 零点设置

使用者可在单位圆上任意设置零点，程序将自动生成离开实轴的复型数对。随后即可计算滤波器的频率、相位和单位冲激响应。

用“(Z)ero place”功能设置二阶滤波器的零点为 $z = -0.5 \pm j0.866$ ，并评价其频率响应。在带有或未带 60Hz 噪声的采样率为 180 次/s 的 ECG 数据上运行此滤波器并总结结果。

5.8.5 单位脉冲序列

在“STDLIB”库中有一个名为 ups.dat 的数据文件。它由单个脉冲组成，读 ups.dat 文件并运行 FIR 滤波器，将产生该滤波器的单位冲激输出序列。在所得输出信号上执行“(P)wr Spect”，则可获得滤波器的幅值响应。用这种方法，滤波器频率响应的测量要比用滤波器设计功能所产生的图形来测量要精确得多。用该方法可寻找海宁滤波器的 3dB 频率。

5.8.6 频率取样

输入以下数组作为围绕单位圆的响应：{0,0,-40,-40,-40,-40,-40} 和 {0,0,-6,-40,-40,-40,-40}，用频率取样法设计字长为 13 的 FIR 滤波器，通过比较它们的频率响应，可研究拓宽过滤带的影响的方法。优化转折值(序列中的第三项)以得到最小的阻带波纹。

用频率取样法设计字长为 21 的 60Hz 陷波滤波器。在带有和未带有 60Hz 噪声的 ECG 数据上运行滤波器。滤波性能优于 notch60.fil 吗？优化滤波器使其后在 60Hz 频率处，增益最小而在其他频率衰减最小。

5.8.7 窗设计

运行“(W)indow”功能用各种窗函数设计低通滤波器，比较它们的频率响应。设计采样率为 200Hz、截止频率为 25Hz 的滤波器。列表比较各种窗的主瓣宽度与旁带波纹的对应关系。在频率响应的第一个最小值处测量滤波器主瓣宽度。

5.8.8 线性对比片方式——线性相位响应

比较海宁滤波器和 $L=2$ 的最小平方求导滤波器(derir5.fil)的相位响应。为什么最小平方滤波器的相位响应并非真正线性？是什么引起了断续？更多的信息请参见第七章。

复习思考题

- 5.1 FIR 滤波器和 IIR 滤波器的主要区别是什么？
- 5.2 “直接迭代”和“快速迭代”的区别是什么？
- 5.3 为什么有限字长寄存器响应对 FIR 滤波器的影响要比对 IIR 滤波器的影响小得多？
- 5.4 计算并大略绘出两个海宁滤波器级联的频率响应。级联具有线性相位吗？
- 5.5 推导零点位于 $r\angle \pm \theta$ 和 $r^{-1}\angle \pm \theta$ 的 FIR 滤波器的相位响应并注释。
- 5.6 在选择最小平方多项式平滑滤波器的阶数时需折衷考虑的是什么？
- 5.7 完成 $L=2$ 的抛物线平滑滤波系数的推导。
- 5.8 列举 5.3.1 小节中所述的简单 60Hz 陷波滤波器的缺点。
- 5.9 用两点差分和三点中心差分算法来近似求导的主要差别是什么？
- 5.10 用窗设计方法设计滤波器的三个步骤是什么？
- 5.11 阐述各种窗的主瓣宽度与旁带波纹间的关系。
- 5.12 频率取样设计中过渡带的自由参数是如何影响滤波器性能的？
- 5.13 最小设计法和窗设计法最基本的区别是什么？
- 5.14 用窗设计法设计字长为 15 的 FIR 滤波器，要求从 0 到 50Hz 的通带增益为 0dB，从 100Hz 到 200Hz 的阻带衰减为 40dB。比较海宁窗和矩形窗(采用 400 次/s 的采样率)。
- 5.15 用频率取样法解答题 5.14 的问题。
- 5.16 海宁滤波器的传递函数为：

$$H_1(z) = \frac{1 + 2z^{-1} + z^{-2}}{4}$$

①求 DC 增益，②三级滤波器级联得到一个新的传递函数[即 $H(z) = H_1(z) \times H_1(z) \times H_1(z)$]，此滤波器的整体 DC 增益为多少？③从带有零相位延时的全通海宁滤波器中减去该滤波器的输出可设计出一个高通滤波器。所得滤波器有几个零点？都位于何处？

5.17 两个滤波器级联。第一个的传递函数为 $H_1(z) = 1 + 2z^{-1} - 3z^{-2}$ 。第二个的传递函数为 $H_2(z) = 1 - 2z^{-1}$ 。将单位冲激加在级联滤波器的输入端。①求输出序列。②求级联滤波器在：③DC，④1/2 折叠频率处，⑤折叠频率处的幅值响应。

5.18 两个滤波器级联。第一个的传递函数为 $H_1(z) = 1 + 2z^{-1} + z^{-2}$ ，第二个的传递函数为 $H_2(z) = 1 - z^{-1}$ 。①将单位脉冲加在级联滤波器的输入端，求输出序列。②求级联滤波器的幅值响应在 DC 的幅值。

第六章 无限冲激响应滤波器

本章介绍有陡峭转折特性的无限冲激响应(IIR)数字滤波器的分析与设计。

首先列举单极点的例子来简单说明极点位置与滤波器稳定性之间的关系,然后介绍如何设计具有两个极点的滤波器,包括低通、高通、带通、带阻等滤波器。本章还介绍全 IIR 滤波器的积分器的算法。最后提供了用于 ECG 分析的 IIR 滤波器的实验实例。

6.1 IIR 滤波器的通用表达式

IIR 滤波器传递函数的一般表达式为两个多项式之比值,即

$$H(z) = \frac{\sum_{i=0}^n a_i z^{-i}}{1 - \sum_{i=1}^n b_i z^{-i}} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 - b_1 z^{-1} - b_2 z^{-2} - \dots - b_n z^{-n}} = \frac{Y(z)}{X(z)} \quad (6.1)$$

或写成

$$Y(z) = b_1 Y(z)z^{-1} + \dots + b_n Y(z)z^{-n} + a_0 X(z) + a_1 X(z)z^{-1} + \dots + a_n X(z)z^{-n} \quad (6.2)$$

式(6.2)右边是延迟反馈项之和,把这些反馈项称为递归序列,因此,这种形式的滤波器又称为递归滤波器,如图 6.1 所示。

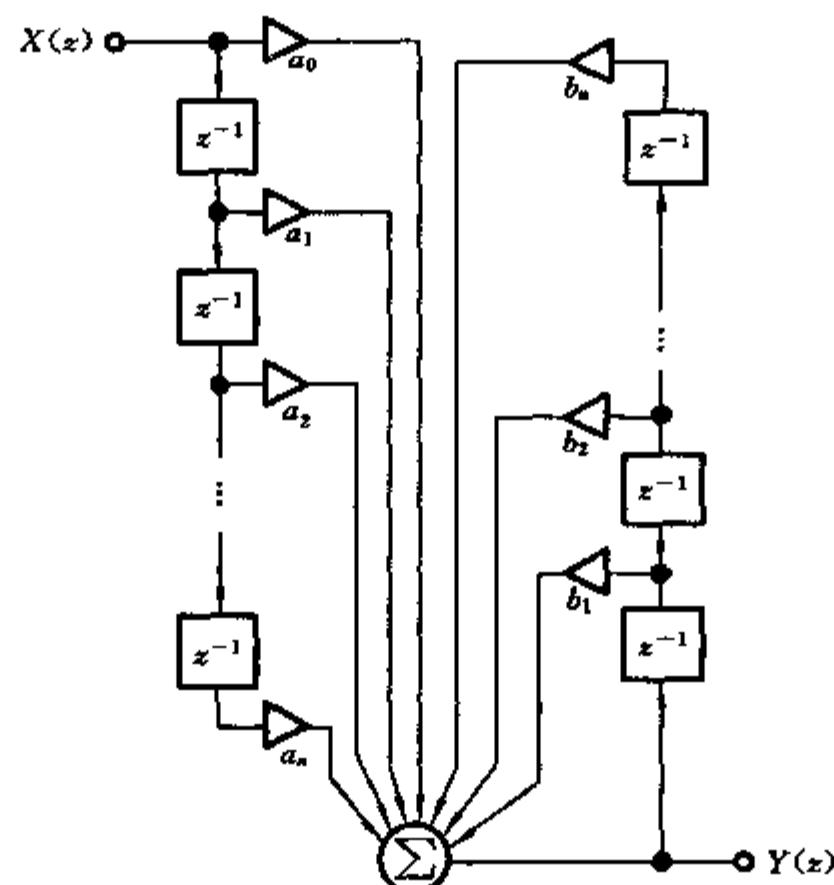


图 6.1 经延迟和反馈得到的 IIR 滤波器的输出

6.2 简单单极点实例

考虑图 6.2(a)所示的滤波器,通过在 $X(z)$ 端输入一个单位脉冲序列来求传递函数。图 6.2(a)所示的是在反馈系数为 $\beta=\frac{1}{2}$ 时,滤波器不同点所得到的序列。序列(1)定义为单位脉冲,从输出序列(2)可写出其传递函数,即

$$H(z) = 1 + \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2} + \frac{1}{8}z^{-3} + \dots \quad (6.3)$$

由二次项定理可以将上式写成两个多项式相除的有限和表达式,即

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2}z^{-1}} \quad (6.4)$$

上式经等式变换后,将 $Y(z)$ 用输入 $X(z)$ 和反馈函数表示为:

$$Y(z) = \frac{1}{2}Y(z)z^{-1} + X(z) \quad (6.5)$$

$x(nT)$ 和 $y(nT)$ 是目前采样时刻的输入和输出序列的一点,分别对应于 z 域中的 $X(z)$ 和 $Y(z)$,同样地, $y(nT-T)$ 表示过去某时刻的输出, z 变换后可得 $Y(z)z^{-1}$,因此上式可变换为:

$$y(nT) = \frac{1}{2}y(nT-T) + x(nT) \quad (6.6)$$

它与有限脉冲响应(FIR)不同之点在于,目前时刻的输出 $y(nT)$ 不仅与目前时刻的输入 $x(nT)$ 有关,而且与目前时刻之前的输出 $y(nT-T)$ 有关。因此,过去的输出影响下一次输出,依此类推,这一个输出又影响下一个输出。瞬态信号在从输出信号中消失之前要占用大量的采样点。FIR 滤波器不可能做到这一点,因为它不含反馈。为找到零点和极点,先把式(6.4)的传递函数乘以 z/z ,得到

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - \frac{1}{2}z^{-1}} \times \frac{z}{z} = \frac{z}{z - \frac{1}{2}} \quad (6.7)$$

式(6.7)中,令分子等于 0,得到零点 $z=0$;令分母为 0,得到单极点 $z=\frac{1}{2}$ 。

图 6.2(c)所示的为单极点滤波器的零-极点分布图,为确定响应的幅值和相位,用 $z=e^{j\omega T}$ 代入传递函数,得

$$H(\omega T) = \frac{1}{1 - \frac{1}{2}e^{-j\omega T}} = \frac{1}{\left[1 - \frac{1}{2}\cos(\omega T)\right] + j\left[\frac{1}{2}\sin(\omega T)\right]} \quad (6.8)$$

根据式(6.8),可画出如图 6.3 所示的相频和幅频特性图,由图可见,它是一个低通滤波器。如果把常数 β 由 $1/2$ 变为 2 ,则单位脉冲响应为 $(1, 2, 4, 8, 16, \dots)$,如图 6.2(d)所示。这个滤波器是不稳定的,输出随不断采样而不断倍增,单位脉冲响应并不随时间而衰减。通常希望单位脉冲响应能衰减到 0。在 $\beta=2$ 时计算极点位置,得到 $z=2$ 。图 6.2(e)所示的极点在单位圆外,所以滤波器是不稳定的。如果把 β 变为 1 ,则 $z=1$,极点正好在单位圆上,输出响应为 $(1, 1, 1, \dots)$,这是一种特殊的 IIR 滤波器,即矩形积分器。

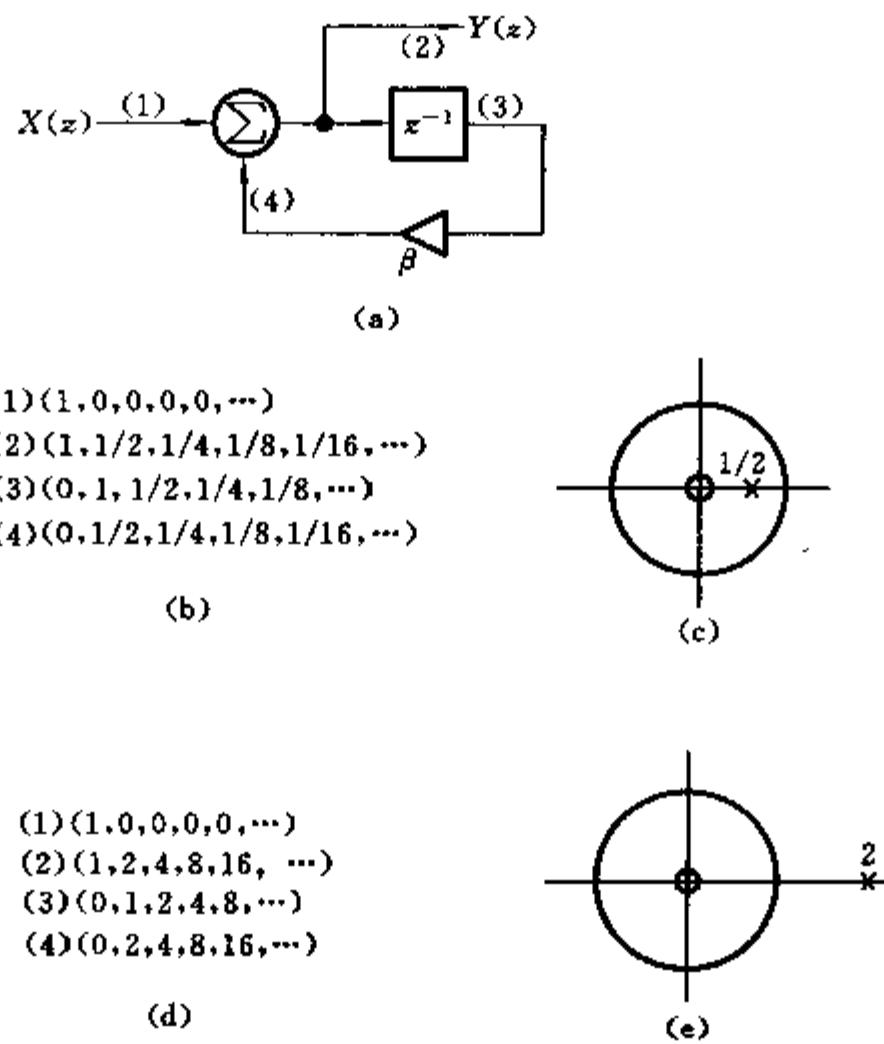


图 6.2 简单的单极点递归滤波器

(a) 框图; (b) $\beta = \frac{1}{2}$ 时单位冲激响应; (c) $\beta = \frac{1}{2}$ 零-极点图;
 (d) $\beta = 2$ 时单位脉冲响应; (e) $\beta = 2$ 零-极点图

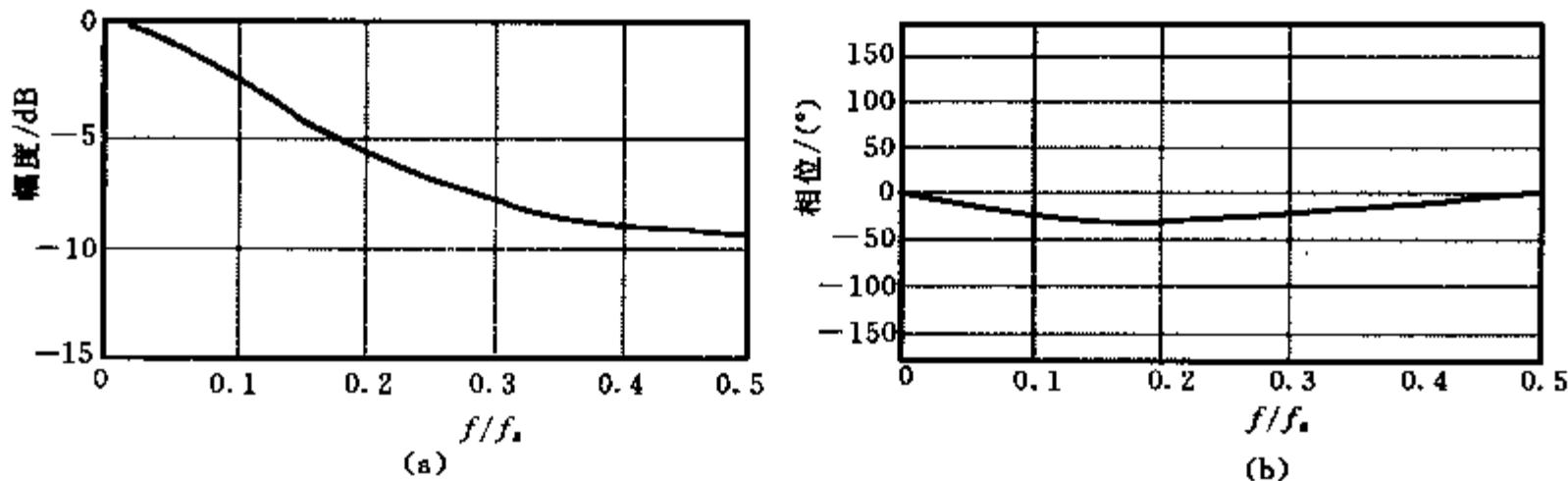


图 6.3 极点为 $z = \frac{1}{2}$ 的单极点递归滤波器特性

(a) 幅值响应; (b) 相位响应

6.3 积 分 器

积分器的通用形式为：

$$A = \int_{t_1}^{t_2} f(t) dt \quad (6.9)$$

式中, A 是函数 $f(t)$ 从 t_1 到 t_2 的积分。用有限个点和曲线拟合的方法对上式进行函数逼近, 可实现积分的数字算法, 即

$$A = \sum_{n=t_1}^{t_2} f(n) \Delta t \quad (6.10)$$

积分器的拉氏变换为:

$$H(s) = \frac{1}{s} \quad (6.11)$$

将 $s=j\omega$ 代入传递函数, 可得到相应的幅值和相位的关系, 即

$$H(j\omega) = \frac{1}{j\omega} \quad (6.12)$$

理想幅值响应与频率成反比例, 即

$$|H(j\omega)| = \left| \frac{1}{\omega} \right| \quad (6.13)$$

相位响应

$$\angle H(j\omega) = \arctan \left[-\frac{1/\omega}{0} \right] = -\frac{\pi}{2} \quad (6.14)$$

数字积分器与模拟积分器不同, 它是通过计算机程序实现的, 没有漂移的问题, 不受电容余电量的影响。下面讨论三种积分方法——矩形求和、梯形求和以及辛甫森法。

6.3.1 矩形求和法

矩形求和是最简单的积分算法, 它用矩形面积之和代替积分。如图 6.4(a) 所示, 每个矩形的宽是一个采样周期 T , 高是最近一次采样时刻输入 $x(nT-T)$ 的幅值。每个矩形面积是 $Tx(nT-T)$ 。差分方程为:

$$y(nT) = y(nT-T) + Tx(nT-T) \quad (6.15)$$

式中, $y(nT-T)$ 是 $(nT-T)$ 时刻前的矩形面积和。误差为实际面积与矩形面积和之差。式 (6.15) 的 z 变换为:

$$Y(z) = Y(z)z^{-1} + TX(z) \quad (6.16)$$

和

$$H(z) = \frac{Y(z)}{X(z)} = T \left[\frac{z^{-1}}{1-z^{-1}} \right] \quad (6.17)$$

如图 6.4(a) 所示, 传递函数有一个极点 $z=1$ 。幅值响应和相位响应分别为:

$$|H(\omega T)| = \left| \frac{T}{2\sin(\omega T/2)} \right| \quad (6.18)$$

和

$$\angle H(\omega T) = -\frac{\omega T}{2} \quad (6.19)$$

图 6.5 所示的是幅值响应曲线, 可通过提高采样率(远大于信号最高频率)来减少误差, 使每个矩形的宽度远小于信号的变化, 这样每个矩形面积更接近真实值。但提高采样频率只适用于低频范围, 因为在这种情况下, 矩形求和才与理想情况大致相同。不过, 提高采样频率会增加计算量和时间, 占去大量存储空间, 同时会产生大量不必要的数据, 所以希望选择其他方法。

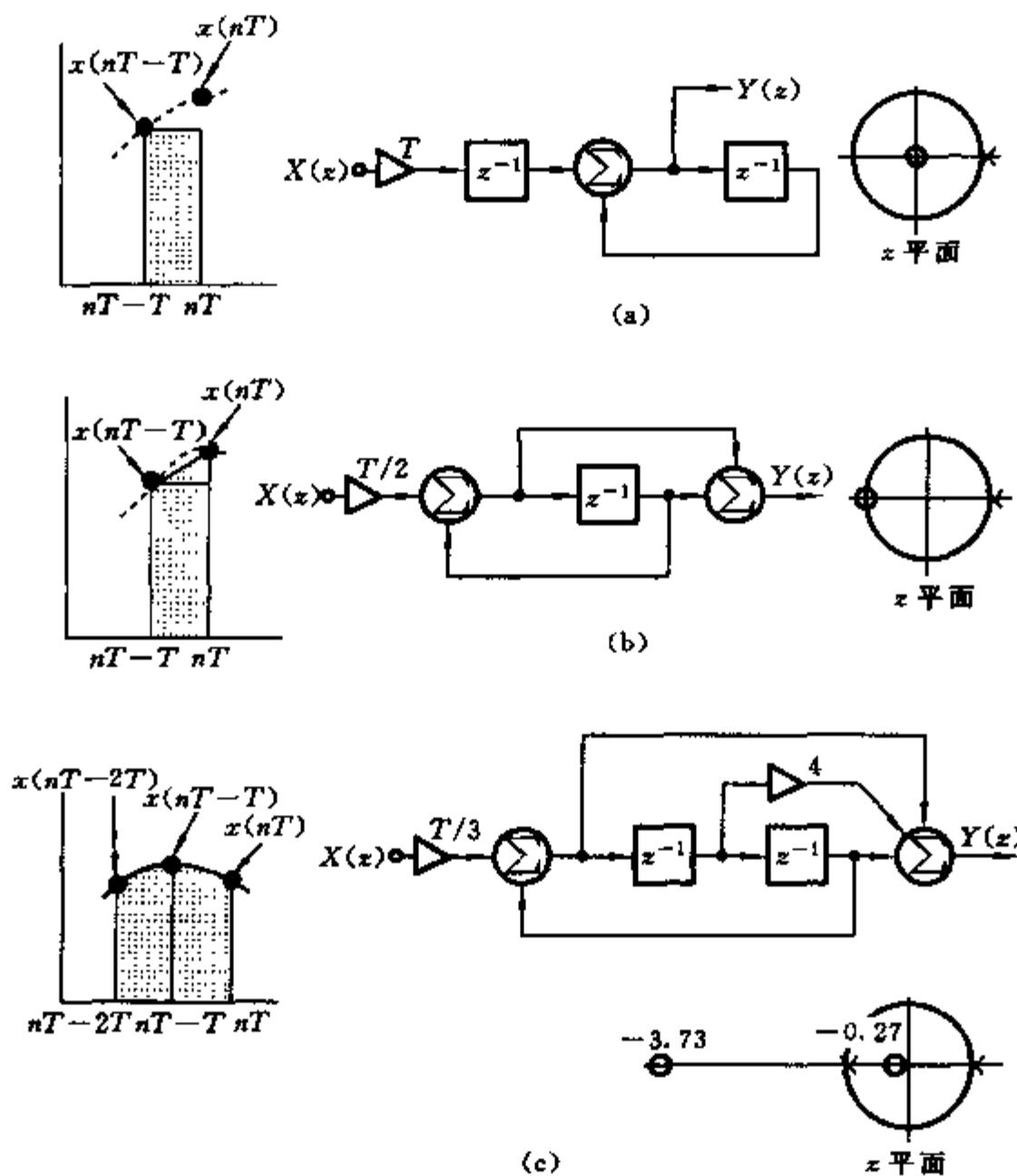


图 6.4 积分
(a)矩形法; (b)梯形法; (c)辛普森法

6.3.2 梯形积分法

这种方法是在矩形面积基础上再加一个三角形面积,如图 6.4(b)所示。除三角形部分外,差分方程与前面相同。梯形面积的差分公式为:

$$\begin{aligned} y(nT) &= y(nT - T) + Tx(nT - T) + \frac{T}{2}[x(nT) - x(nT - T)] \\ &= y(nT - T) + \frac{T}{2}[x(nT) + x(nT - T)] \end{aligned} \quad (6.20)$$

上式对应的 z 变换为:

$$H(z) = \frac{T}{2} \left[\frac{1 + z^{-1}}{1 - z^{-1}} \right] \quad (6.21)$$

为了得到准确的幅值和相位,用 $e^{j\omega T}$ 代替 z ,则式(6.21)变为:

$$H(\omega T) = \frac{1 + e^{-j\omega T}}{1 - e^{-j\omega T}} = \frac{e^{-j\frac{\omega T}{2}} [e^{j\frac{\omega T}{2}} + e^{-j\frac{\omega T}{2}}]}{e^{-j\frac{\omega T}{2}} [e^{j\frac{\omega T}{2}} - e^{-j\frac{\omega T}{2}}]} = \frac{2 \cos \left[\frac{\omega T}{2} \right]}{2 j \sin \left[\frac{\omega T}{2} \right]} = -j \cot \left[\frac{\omega T}{2} \right] \quad (6.22)$$

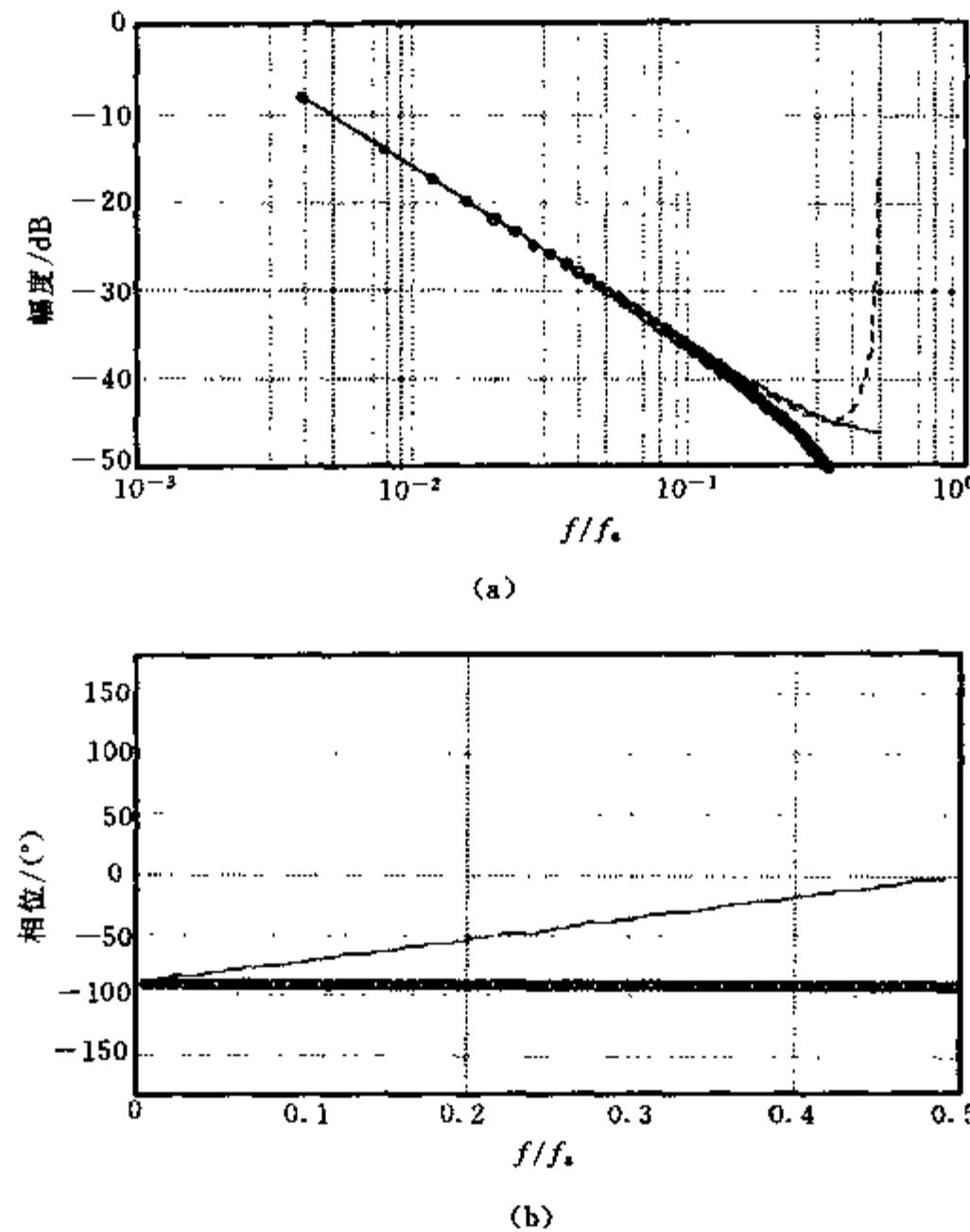


图 6.5 数字积分器

(a) 幅值响应; (b) 相位响应采样周期设定为 1;
实线: 矩形法; 圆圈: 梯形法; 虚线: 辛甫森法

幅值响应, 即 $|H(\omega T)|$ 的大小为:

$$|H(\omega T)| = \left| \cot\left(\frac{\omega T}{2}\right) \right| \quad (6.23)$$

相位响应为:

$$\angle H(\omega T) = -\frac{\pi}{2} \quad (6.24)$$

梯形积分器的结构框图和零-极点图如图 6.4(b)所示。其极点与矩形算法一样, 仍为 $z=1$, 但零点移到了 $z=-1$, 转折频率 f_0 处幅值为 0, 幅值和相位响应分别为:

$$|H(\omega T)| = \left| \frac{T}{2} \cot\left(\frac{\omega T}{2}\right) \right| \quad (6.25)$$

$$\angle H(\omega T) = -\frac{\pi}{2} \quad (6.26)$$

式(6.25)表示的幅值比矩形法的更加接近理想积分器, 而相位响应与理想积分器的一样。梯形算法提供了一种简单有效的积分算法, 图 6.6 所示的是用差分方程直接编写的梯形算法程序。

```

Turbo C source code implementing a trapezoidal integrator
Assume sampling period is 2 ms, the difference equation is
y(nT) = y(nT-T) + 0.001 * (nT) + 0.001 * (nT-T)

float Trapezoid(signal)
float signal;                                /* x[nT] */
{
    static float x[2],y[2];
    int count;
    x[1]=x[0];                                /* x[1]=x(nT-T) */
    x[0]=signal;                               /* x[0]=x(nT) */

    y[0]=y[1]+0.001*x[0]+0.001*x[1]; /* difference equation */
    y[1]=y[0];                                /* y[1]=y(nT-T) */

    return y[0];
}

```

图 6.6 C 语言的梯形积分器

6.3.3 辛甫森积分法

辛甫森积分法用得最广泛。如图 6.4(c)所示,它用三个顺序输入的点的多项式来实现面积逼近。增量面积加上每个输入面积为总的积分面积。其差分方程为:

$$y(nT) = y(nT - 2T) + \frac{T}{3}[x(nT) + 4x(nT - T) + x(nT - 2T)] \quad (6.27)$$

传递函数为:

$$H(z) = \frac{T}{3} \left[\frac{1 + 4z^{-1} + z^{-2}}{1 - z^{-2}} \right] \quad (6.28)$$

图 6.4(c)所示的为框图和零-极点分布图。幅值和相位分别为:

$$|H(\omega T)| = \left| \frac{T}{2} \left[\frac{2 + \cos(\omega T)}{\sin(\omega T)} \right] \right| \quad (6.29)$$

$$\angle H(\omega T) = -\frac{\pi}{2} \quad (6.30)$$

如图 6.5 所示,辛甫森积分器的幅值响应与梯形算法的一样,相位为理想的 $-\frac{\pi}{2}$ 。相对其他方法而言,辛甫森法以低于 $f_s/4$ 的频率逼近理想积分器,因而性能更好,但是它放大了转折频率处的高频噪声信号,所以它不适用于有噪声情况下的积分。噪声信号可以用梯形求和法求得积分。

6.4 双极点滤波器的设计方法

IIR 滤波器具有陡峭转折特性,与 FIR 的设计方法不同,FIR 一般是以数字量来逼近输入序列的,而 IIR 滤波器则是从希望逼近的模拟滤波器开始设计的,它的传递函数可以用多项式之比的有限和形式表示,而且相位响应是非线性的。

6.4.1 r 和 θ 的选择

低通、高通、带通、带阻四种滤波器的通用设计方法都有标准的递归形式,其传递函数为:

$$H(z) = \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} + b_2 z^{-2}} \quad (6.31)$$

其零点为:

$$z = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} \quad (6.32)$$

极点为:

$$z = \frac{b_1 \pm \sqrt{b_1^2 - 4b_2}}{2} \quad (6.33)$$

式中

$$b_1 = 2r\cos\theta, \quad b_2 = r^2 \quad (6.34)$$

$$\theta = 2\pi \left[\frac{f_c}{f_s} \right] \quad (6.35)$$

图 6.7 所示的为双极点滤波器的结构框图,给分子系数 a_1 和 a_2 赋值就确定了零点位置,并且定义了滤波器类型。变量 b_1 和 b_2 由极点在单位圆内分布的位置来决定。

现在来选择采样频率 f_s ,它必须大于或等于信号 $y(z)$ 的最高频率成分的 2 倍。下一步是选择临界频率 f_c 。对于低通或高通滤波器而言, f_c 是转折频率;对于带通滤波器来讲, f_c 是谐振频率;对于带阻滤波器来讲, f_c 是陷波频率。 f_c 和 f_s 决定了极点的角度 θ 。

图 6.7 双极点滤波器框图

再选择 r , r 为极点到坐标原点的距离。设 a 是衰减系数,则

$$r = e^{-aT} \quad (6.36)$$

从幅值响应上可看出,当极点接近单位圆(即 $r \rightarrow 1$ 或 $a \rightarrow 0$)时,幅值响应是欠阻尼的;当极点接近原点(即 $r \rightarrow 0$ 或 $a \rightarrow \infty$)时,幅值响应为过阻尼的。增大 r 或 θ 使极点移动,会导致幅值响应产生欠阻尼,而减少 r 或 θ 会导致幅值产生过阻尼。在最后的传递函数设计中,用 $e^{j\omega T}$ 代替 z 可得到频率特性。

应用传递函数

$$H(z) = \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} + b_2 z^{-2}} \quad (6.37)$$

求出 $Y(z)$ 的表达式:

$$Y(z) = b_1 Y(z)z^{-1} - b_2 Y(z)z^{-2} + X(z) + a_1 X(z)z^{-1} + a_2 X(z)z^{-2} \quad (6.38)$$

现在可以直接写出差分方程：

$$\begin{aligned} Y(nT) &= b_1 y(nT - T) - b_2 y(nT - 2T) \\ &\quad + x(nT) + a_1 x(nT - T) \\ &\quad + a_2 x(nT - 2T) \end{aligned} \quad (6.39)$$

图 6.8 所示的是式(6.31)的 a_1, a_2 字数表。

	a_1	a_2
低通	2	1
带通	0	-1
高通	-2	1
带阻	$2\cos\theta$	1

图 6.8 双极点滤波器的式(6.31)
实现的分子系数表

图 6.9 所示的为零-极点图和四种滤波器的相应的橡胶膜片图。

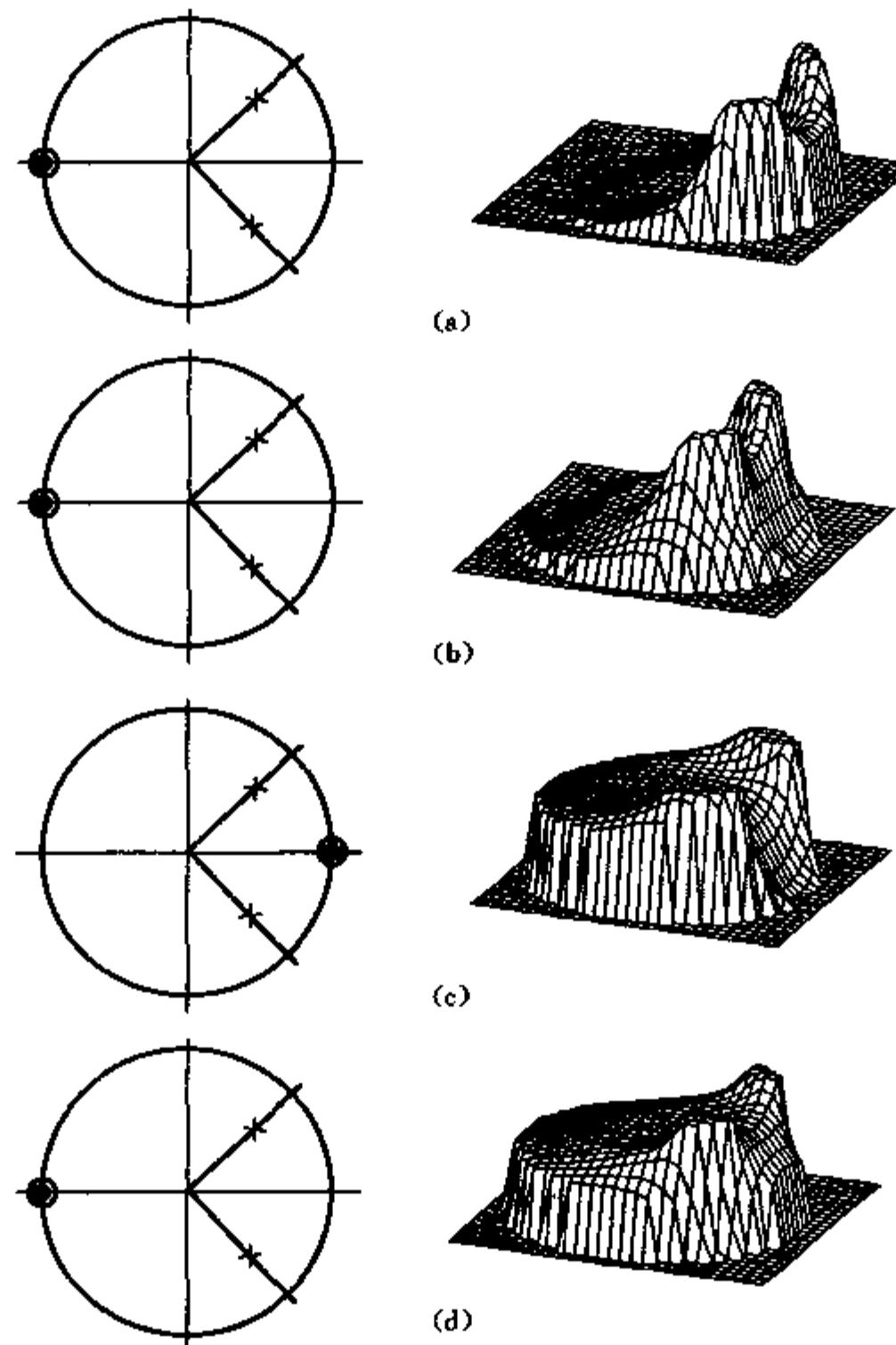


图 6.9 基本双极点递归数字滤波器的零-极点图和橡胶膜片图
(a) 低通; (b) 带通; (c) 高通; (d) 带阻

现在用 C 语言来设计程序(见图 6.10)。首先选择低通滤波器。低通滤波器具有两个零点，通过令 $a_1=2, a_2=1$ ，可确定这两个零点值，然后选择极点位置。这里选 $r=0.5, \theta=\pm 45^\circ$ ，可得 $b_1=0.707, b_2=0.25$ (见式(6.34))。

用软件在计算机中实现滤波器功能已经很直观了,可直接根据差分方程写出滤波器子程序。以式(6.10)为例,有以下五个步骤:

- (1)接收目前输入信号。
- (2)将阵列元移位以得到过去输入信号的延迟,并存储。
- (3)累加所有过去和,或目前输入以及过去输出量并乘上系数。
- (4)将阵列元移位得到过去输出信号的延迟,并存储。
- (5)输出信号返回。

这种算法可同样用于高通、带通和带阻滤波器。这个软件实现的滤波器唯一缺点就是速度有限。此程序已实际用于如ECG的生物医学信号的实时计算中。

Turbo C source code for implementing an IIR low-pass filter

Assume we use the r and q method where $r=0.5, q=45^\circ$

The difference equation is:

```
y(nT)=0.707y(nT-T)-0.25y(nT-2T)+x(nT)+2x(nT-T)+x(nT-2T)
y(nT)=0.707 * y[1]-0.25 * y[2]+x[0]+2 * x[1]+x[2]
float Lowpass(signal)
float signal;                                /* x[nT] */
{
    static float x[3],y[3];
    int count;
    for (count=2;count>0;count--)
        x[count]=x[count-1];      /* shift for x term delays */
    x[0]=signal;                      /* x[0]=x[nT] */
    /* difference equation */
    y[0]=0.707 * y[1]-0.25 * y[2]+x[0]+2.0 * x[1]+x[2];
    for(count=2;count>0;count--)
        y[count]=y[count-1];      /* shift for y term delays */
    return y[0];
}
```

图 6.10 用 C 语言编写的 IIR 低通滤波器

这种双极点滤波器与二阶模拟滤波器有相似的工作特性。其特性曲线转折慢,可以通过串联相同环节来增加下降斜率。当然实时系统中的有限响应时间也限制了串联的级数。在这种情况下就必须设计高阶滤波器。

6.4.2 双线性变换法

可以通过设计一个递归滤波器来实现模拟滤波器的功能。首先来研究 s 平面的传递函数。将传递函数中出现的 s 全部替换为:

$$s \approx \frac{2}{T} \left[\frac{1 - z^{-1}}{1 + z^{-1}} \right] \quad (6.40)$$

这个变换是非线性的。转折频率定义为：

$$\omega' = \frac{2}{T} \left[\tan\left(\frac{\omega T}{2}\right) \right] \quad (6.41)$$

模拟频域中的频率 ω' 对应于数字频域中的频率 ω 。要实现双线性变换，首先可将 ω' 的表达式代入式(6.40)，然后用 z 平面的方程式代替 s 平面的方程。比如以下传递函数：

$$H(s) = \frac{\omega_c}{s^2 + \omega_c^2} \quad (6.42)$$

根据式(6.41)代换 ω' ，再根据式(6.40)代换 s ，得

$$H(z) = \frac{\frac{2}{T} \left[\tan\left(\frac{\omega T}{2}\right) \right]}{\left(\frac{2}{T} \left[\frac{1 - z^{-1}}{1 + z^{-1}} \right] \right)^2 + \left[\tan\left(\frac{\omega T}{2}\right) \right]^2} \quad (6.43)$$

这就是用于完成与模拟滤波器大致相同性能的数字滤波器。此方法也适于设计高阶滤波器。

6.4.3 变换表方法

可以用变换表(例如 Stern, 1975)来设计任意阶的数字滤波器。这个表给出了连续或离散函数的拉氏变换和 z 变换，现在考虑一个如图 6.11(a)所示的二阶滤波器。模拟传递函数为：

$$H(s) = \frac{A}{LC} \left[\frac{1}{s^2 + (R/L)s + 1/LC} \right] \quad (6.44)$$

求极点，得

$$s = -a \pm j\omega_c \quad (6.45)$$

式中，

$$a = \frac{R}{2L} \quad (6.46)$$

$$\omega_c = \left[\frac{1}{LC} - \frac{R^2}{4L^2} \right]^{1/2} \quad (6.47)$$

传递函数就可以写成：

$$H(s) = \frac{A}{LC} \left[\frac{1}{(s + a)^2 + \omega_c^2} \right] \quad (6.48)$$

图 6.11(b)所示的为 s 域的零-极点图，相应的连续时间函数为 $e^{-at} \sin(\omega_c t)$ 。对应于离散时间函数， $e^{-\omega_c T} \sin(\omega_c T)$ 的 z 变换为：

$$H(z) = \frac{Gz^{-1}}{1 - b_1z^{-1} - b_2z^{-2}} \quad (6.49)$$

式中，

$$b_1 = 2e^{-aT} \cos(\omega_c T) \quad (6.50)$$

$$b_2 = -e^{-2aT} \quad (6.51)$$

而且

$$G = \frac{A}{\omega_c LC} e^{-aT} \sin(\omega_c T) \quad (6.52)$$

变量 a 、 ω_c 、 A 、 L 和 C 取自模拟滤波器设计值。传递函数有一个零点 $z=0$ 和两个极点

$$z = b_1 \pm j(b_1^2 + 4b_2)^{1/2} = b_1 \pm jc_1 \quad (6.53)$$

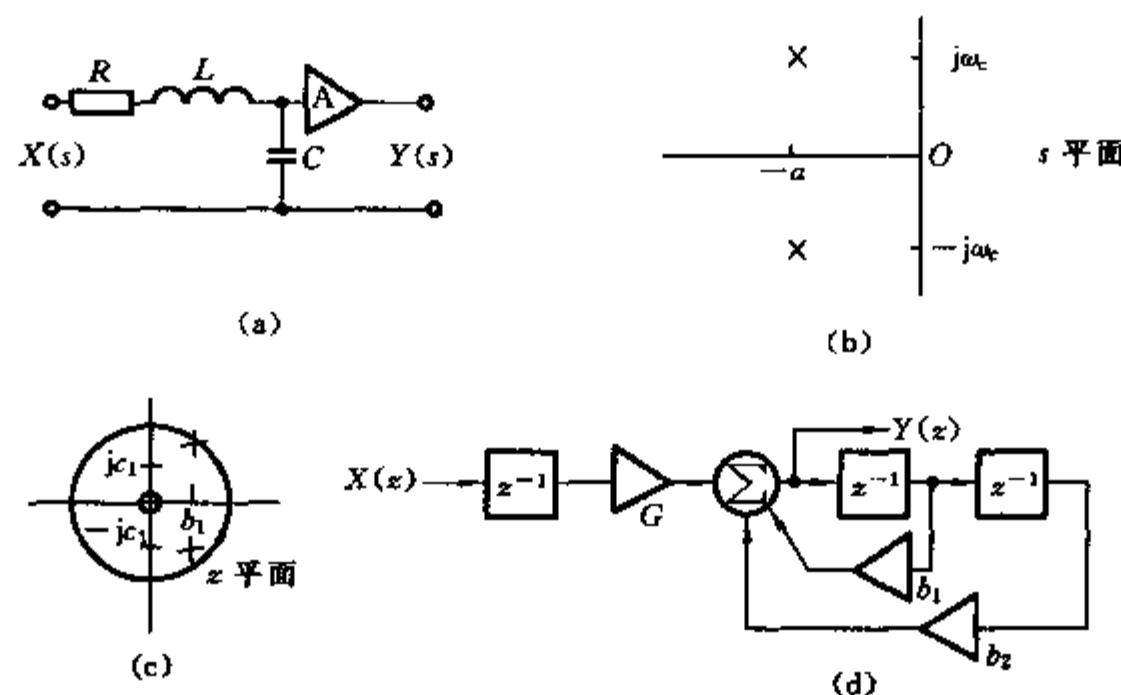


图 6.11 二阶滤波器

(a)模拟滤波器电路;(b)模拟滤波器零-极点图转换函数;
(c)二阶滤波器数字形式的零-极点图;(d)数字滤波器框图

图 6.11(c)所示的为 z 平面零-极点图。用 $Y(z)/X(z)$ 代替 $H(z)$, 得

$$Y(z) = GX(z)z^{-1} + b_1Y(z)z^{-1} + b_2Y(z)z^{-2} \quad (6.54)$$

其差分方程为:

$$y(nT) = Gx(nT - T) + b_1y(nT - T) + b_2y(nT - 2T) \quad (6.55)$$

根据这个差分方程可以直接写出实现滤波器的程序,还可以画出如图 6.11(d)所示的框图。

这种转换表设计方法提供了一种快速设计近似模拟滤波器特性的数字滤波器的方法。对于给定的模拟滤波器传递函数,通常还要用运算放大器和其他元件来实现滤波器的功能,然后,有了 z 变换,就有了完整的专门设计方法,仅需编写一个无循环的程序就可以实现滤波器的功能。

6.5 实验:用于 ECG 分析的 IIR 数字滤波器

本节介绍用于处理 ECG 的 IIR 滤波器的实验。在软件 UW Digi Scope 的主菜单中选(F)ilters, 然后选中(D)esign 以及(I)IIR 就可以设计积分器和双极点滤波器了。

6.5.1 积分器

本章讲述了三种不同的积分器(矩形、梯形、辛甫森法积分器),哪一种最费运行时间?在单位圆图上它们有什么共同点?

对 ECG 数据运行矩形积分器(需要用 ecg105.dat),解释结果。设计一个预处理滤波器解决所观察到的问题。对已适当处理过的 ECG 数据加或不加随机噪声,然后比较三种积分器的输出,哪种滤波器最适合在存在噪声信号场合中使用?

6.5.2 二阶递归滤波器

设计三个双极点带通滤波器, r 值分别为 $r=0.7, 0.9, 0.95$, 临界频率为 17Hz, 采样率为 200 次/s。测量三种滤波器的 Q 值。Q 定义为临界频率(谐振频率)与 3dB 带宽频率之比。对

ECG 数据文件运行三个滤波器并比较输出。当选择 r 时应权衡考虑什么?

6.5.3 传递函数

(G)eneric 工具允许输入下面形式的传递函数的系数来设计滤波器:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}$$

取 $R=10\Omega$, $L=10\text{mH}$, $C=1\mu\text{F}$, 用 6.4.3 小节的转换表方法的例子, 计算并输入 $H(z)$ 的系数。使用 200 次/s 采样率。滤波器 Q 值多大?

6.5.4 零-极点配置

按如图 6.12(a)所示的两个零点($r=1.0$, 角度 $\pm 120^\circ$), 以 180 次/s 采样率实现一个 60Hz 陷波 FIR 滤波器。如图 6.12(b)所示, 在零点附近加上极点($r=0.9$, 角度 $\pm 110^\circ$ 和 $\pm 130^\circ$), 实现 IIR 滤波器。比较两个滤波器的幅值响应。测量两个滤波器的 Q 值。能进一步提高 IIR 滤波器的 Q 值吗? 相位响应如何?

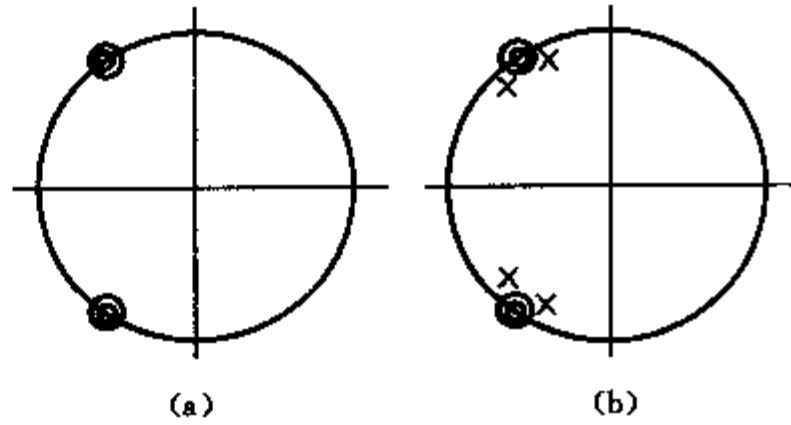


图 6.12 陷波滤波器
(a)FIR 滤波器; (b)IIR 滤波器

复习思考题

- 6.1 试描述递归滤波器的传递函数的特性。
- 6.2 IIR 滤波器与递归滤波器的区别是什么?
- 6.3 怎样推导递归滤波器的频率响应?
- 6.4 根据一个递归滤波器的传递函数式, 你如何写出其差分方程式?
- 6.5 采用 r 和 θ 的方法, 设计一个低通递归滤波器。
- 6.6 采用双线性变换的方法, 设计一个高通滤波器。
- 6.7 采用转换表的方法, 设计一个滤波器。
- 6.8 某数字滤波器有一个单位脉冲输出序列(即 1,0,0,...), 当一个单位阶跃(即 1,1,1,1,...)作用其输入端时, 该滤波器的传递函数是什么?
- 6.9 用 C 语言写出实现一矩形积分器的程序。
- 6.10 采用习题 6.5 得来的结果的差分方程式, 运行 6.4 节所提供的滤波程序。
- 6.11 请画出滤波器 $H(z) = \frac{1+2z^{-1}+z^{-2}}{1-z^{-2}}$ 的 z 平面的零-极点图。
- 6.12 有一滤波器, 当输入单位脉冲时, 输出序列为 $\{-2, 4, -8, 16, \dots\}$ 。试写出它的 z 变换形式(如: 多项式的比)。并指出这个滤波器是递归、非递归、稳定、非稳定、FIR、IIR 中的哪一种。

6.13 一个数字滤波器，它的传递函数为 $H(z) = \frac{1-bz^{-1}}{1+bz^{-1}}$ 。

①当 $b=(1)0.8; (2)-0.8; (3)1; (4)-2; (5)-\frac{1}{2}$ 时的滤波器为何种传统类型？

②当 $b=-\frac{1}{2}$ 时，滤波器增益为多少？

③当 $b=\frac{1}{2}$ 时，写出差分方程 $y(nT)$ ？

6.14 四个数字滤波器的模块图如图 6.13 所示。试写出①传递函数；②差分方程式

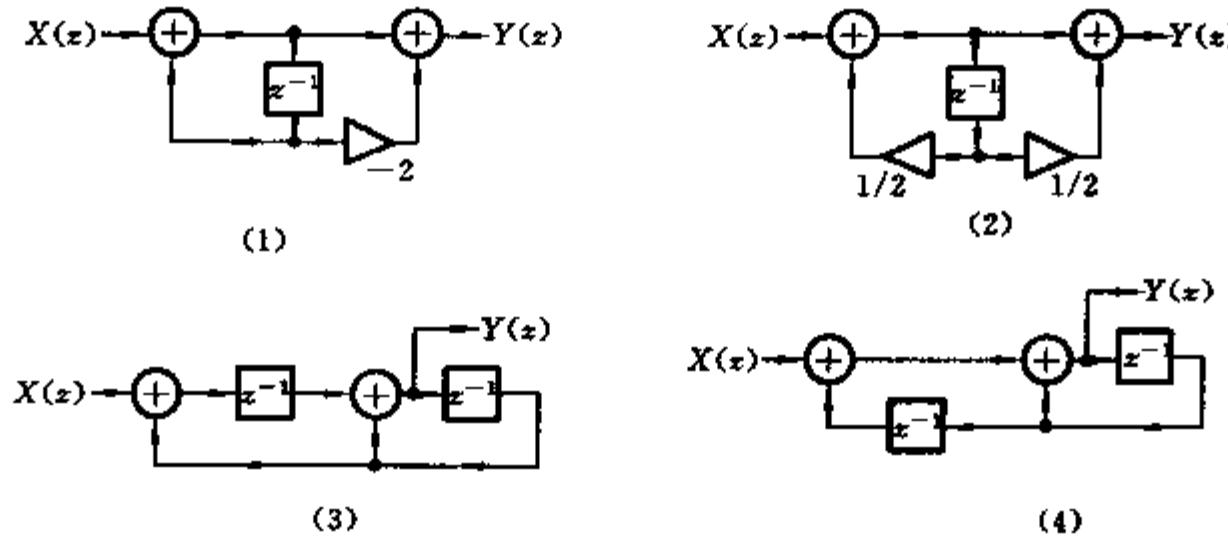


图 6.13

6.15 写出滤波器的传递函数，并评价其稳定性。当输入序列为单位脉冲时，输出序列为：

① $\{3, -6, 12, -24, \dots\}$ ，

② $\{2, -1, 1, -1/2, 1/4, -1/8, \dots\}$ 。

6.16 一个数字滤波器，其差分方程为 $y(nT) = y(nT-2T) + x(nT-T)$ 。问当输入单位脉冲时，输出序列为多少？

6.17 滤波器的差分方程为：

$$y(nT) = x(nT) + 3y(nT-T),$$

问当输入单位脉冲时，输出序列为多少？

6.18 滤波器的差分方程为 $y(nT) = x(nT) + x(nT-2T) + y(nT-2T)$ ，求其零点与极点的位置。

6.19 双极点数字滤波器的一般公式为：

$$H(z) = \frac{1 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} + b_2 z^{-2}} \quad \text{其中 } b_1 = 2r\cos\theta, b_2 = r^2,$$

①在以下条件下它为何种传统类型滤波器：

a. $a_1=2, a_2=1, -2 < b_1 < 2, 0 < b_2 < 1$ ，

b. $a_1=-2\cos\theta, a_2=1, r=1, \theta=60^\circ$ ？

②如果 $a_1=0, a_2=-1, r=\frac{1}{2}, \theta=60^\circ$ ，求差分方程方式。

6.20 一个数字滤波器的两个零点分别为 $z=0.5$ 与 $z=1$ ；两个极点分别为 $z=0.5, z=0$ ，求：

①单二角函数的幅度响应；

②相位响应。

6.21 数字滤波器的模块如图 6.14 所示，

- ①当开关打开时,求:
 - a) 滤波器的传递函数?
 - b) 差分方程式?
 - c) 当以一半采样频率采样时,幅度响应的大小?
- ②当开关合上时,求:
 - a) 滤波器的传递函数?
 - b) 差分方程式?
 - c) 如果 G 为负值,哪种传统的滤波器可以最好地描述它?
 - d) 当 $G=1$,它的零-极点位置如何?
 - e) 当 $G=1$ 时,输入一个单位阶跃函数,输出为多少?
 - f) 当 $G=-1$ 时,输入为单位阶跃函数,输出为多少?

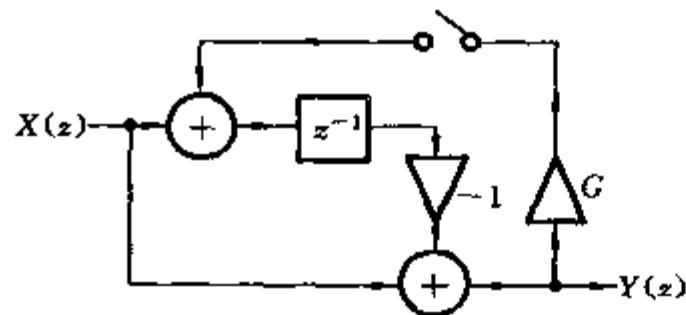


图 6.14

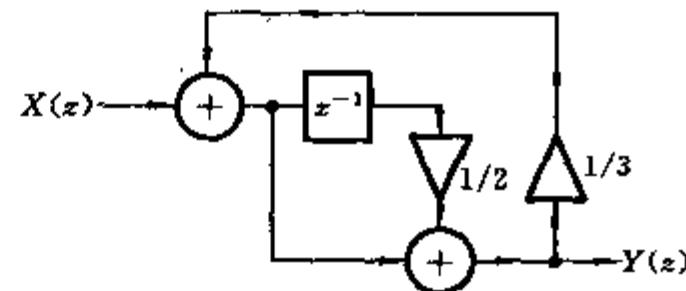


图 6.15

6.22 一个数字滤波器如图 6.15 所示,

- ①写出传递函数;
- ②零、极点的位置?

6.23 滤波器的差分方程为

$$y(nT) = 2y(nT - T) + 2x(nT) + x(nT - T)$$

试画出 z 平面零-极点图。

6.24 将一单位脉冲输入滤波器时,输出序列为 $(1, 0, 1, 0, 1, 0, \dots)$,求这个滤波器的差分方程式。

6.25 将一单位阶跃脉冲输入滤波器时,输出序列为 $\{1, 1, 2, 2, 3, 3, \dots\}$,求这个滤波器的差分方程。(提示:单位阶跃脉冲的 z 变换为 $\frac{1}{1-z^{-1}}$)

6.26 写出图 6.16 所示的数字滤波器的传递函数。

6.27 求传递函数为 $H(z) = \frac{1-z^{-6}}{1+z^{-6}}$ 的滤波器的相位响应。

6.28 写出传递函数为 $H(z) = \frac{z^{-2}}{1-z^{-2}}$ 的滤波器的幅值响应。

6.29 滤波器的零点为 $z = 0.707 \pm j0.707$, 极点为 $z = 0.5 \pm j0.5$, 问哪种传统的滤波器类型可以最好地描述它?

6.30 一个滤波器的采样频率为 1000 采样点/秒,在 $z=1/2$ 处有一极点, $z=3$ 处有一零点,求它在直流下幅值的响应的大小。

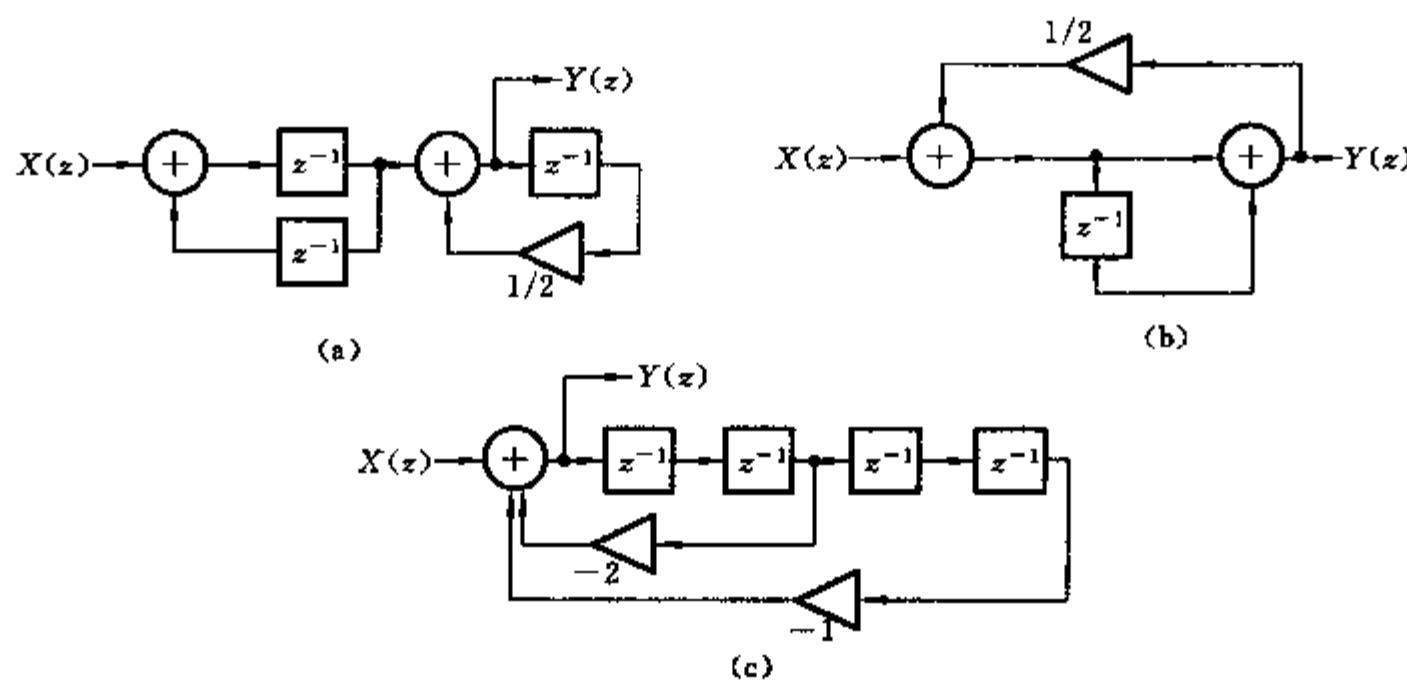


图 6.16

6.31 一个滤波器的差分方程为：

$$y(nT) = x(nT) + x(nT - T) + 0.9y(nT - T)$$

求其传递函数。

6.32 一个滤波器的差分方程为 $y(nT) = y(nT - 2T) + x(nT) + x(nT - T)$, 问哪种传递类型的滤波器可以最好地描述它?

6.33 一个滤波器输入为单位脉冲时, 输出序列为 $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots\}$, 求传递函数。

6.34 一个数字滤波器的差分方程为：

$$y(nT) = x(nT) - ax(nT - T) - by(nT - T)$$

变量 a 和 b 为正整数, 若 $a=1$ 以及 (a) $b=0.8$, (b) $b>1$, 问该滤波器分别为何种传统类型的滤波器?

6.35 滤波器的传递函数为 $H(z) = \frac{1-z^{-1}}{1+z}$, 试求其

- ①振幅响应;
- ②相位响应;
- ③差分方程式。

6.36 滤波器的传递函数为 $H(z) = \frac{z-1}{2z+1}$, 试求其

- ①振幅响应;
- ②相位响应;
- ③差分方程式。

6.37 采样频率为 1000Hz 的一个滤波器, 有一极点 $z=1$ 以及一个零点 $z_1=2$, 求 500Hz 时, 它的振幅响应的幅值是多少?

6.38 采样频率为 200Hz 的一个滤波器, 有极点 $z=\pm j/2$, 以及零点 $z=\pm 1$. 求在 50Hz 时, 振幅响应的幅值是多少?

6.39 一个滤波器的差分方程为 $2y(nT) + y(nT-T) = 2x(nT)$, 求其传递函数 $H(z)$?

6.40 一个滤波器有差分方程为:

$$y(nT) = y(nT-T) - y(nT-2T) + x(nT) + x(nT-T)$$

求它的传递函数?

6.41 当一单位脉冲作用于某滤波器的输入端时,滤波器有这样的输出序列{1,1,1/2,1/4,1/8,...},求该滤波器的传递函数?

6.42 一个滤波器与单位阶跃的 z 变换有相同的传递函数,一个单位脉冲作用在它的输入端,求其输出序列?

6.43 一个滤波器与斜波 z 变换有相同的传递函数,一单位脉冲作用该滤波器的输入端,求其输出序列?(提示:对于一个斜波方程是 $x(nT)=nT$,以及它的 z 变换为:

$$x(z) = \frac{Tz^{-1}}{(1-z^{-1})^2}$$

6.44 作用于数字滤波器输入端的斜波信号产生输出序列{0,T,T,T,T,...},求该滤波器的传递函数?

6.45 一个离散脉冲函数作用于四个不同滤波器的输入端。每个滤波器的输出序列如下:说明其是否是递归或非递归滤波器。

- ①{1,2,3,4,5,6,0,0,0,...};
- ②{1,-1,1,-1,1,-1,...};
- ③{1,2,4,8,16,...};
- ④{1,0.5,0.25,0.125,...}。

6.46 课文所讨论的三种积分算法(即矩形、梯形和辛甫森法则)相似的共同点是什么?

6.47 一个微分器与一个积分器级联。微分器采用两点差分方法: $H_1(z) = \frac{1-z^{-1}}{T}$;积分器采用梯形积分法: $H_2(z) = \frac{T}{2}[\frac{1+z^{-1}}{1-z^{-1}}]$,一个单位脉冲作用到它的输入端,它的输出序列是什么?

6.48 一个微分器与一个积分器级联。微分器采用三点中心差算法:

$$H_1(z) = \frac{1-z^{-2}}{2T}$$

积分器采用矩形积分器

$$H_2(z) = T[\frac{1}{1-z^{-1}}]$$

①一个单位脉冲作用到它的输入端,它的输出序列是什么?

②最好描述这种滤波器的传统滤波器类型是什么?

6.49 一个数字滤波器有两个位于 $z=0.5$ 和 $z=1$ 的零点,和一个位于 $z=0.5$ 的极点,对于①振幅响应为一个单三角_____,简单三角_____,以及②它的相位响应。

6.50 当一个单位脉冲作用到滤波器的输入端,滤波器有这样的输出序列,{2,-1,1,-1/2,1/4,-1/8,...},它的传递函数是什么?

6.51 滤波器的差分方程是:

$$y(nT-T) = x(nT-T) + 2x(nT-4T) + 4x(nT-10T)$$

它的传递函数 $H(z)$ 是什么?

6.52 具有差分方程数字滤波器为:

$$y(nT-2T) = y(nT-T) + x(nT-T) + x(nT-4T) + x(nT-10T)$$

该滤波器的传递函数是什么?

6.53 一个数字滤波器对一个单位脉冲响应有下列的输出序列{1, -2, 4, 8, ...}, 它的极点的位置在什么地方?

6.54 一个数字滤波器有一个位于 $z=0.5$ 的单零点和一个位于 $z=-0.5$ 的单极点, 它的振幅与相位响应是什么?

6.55 一个滤波器的差分方程是:

$$y(nT) = 2y(nT-T) + 2x(nT) + x(nT-T)$$

它的零、极点在什么位置?

6.56 具有 z 变换为 $H(z) = \frac{z^2-1}{z^2+1}$ 的滤波器最优描述传统滤波器的类型是什么?

6.57 一个离散脉冲函数作用到四个不同的滤波器的输入端, 这些滤波器的输出序列如下。这些滤波器中哪个在单位圆外有一个极点?

- ① {1, 2, 3, 4, 5, 6, 0, 0, 0, ...};
- ② {1, -1, 1, -1, 1, -1, ...};
- ③ {1, 2, 4, 8, 10, ...};
- ④ {1, 0.5, 0.25, 0.125, ...}.

6.58 试画出差分方程为:

$$y(nT) = y(nT-T) + y(nT-2T) + x(nT) + x(nT-T)$$

滤波器的结构图。

6.59 差分方程 $y(nT) + 0.5y(nT-T) = x(nT)$ 所描述的滤波器的传递函数是什么?

6.60 一个滤波器, 当输入序列{1, 3, 0, 0, ...}, 有下面输出序列{1, 5, 3, -9, 0, 0, ...}, 它的传递函数是什么?

第七章 整数型滤波器

数字滤波器用于实时系统时,存在计算时间过长的缺点。一种减少计算时间的方法是在滤波器的算式中采用整系数代替浮点系数的方法。用整数的移位及累加指令代替浮点乘法指令,前者比后者所需时钟周期少。整数型滤波器是一种特殊的数字滤波器,它仅含有整数系数,这个特点决定了设计上的限制,比如:很难实现锐截止。因为整数型滤波器比传统滤波器的速度要快,它常用于以低速微型处理器处理高采样率数据的情况。

本章讨论整数型滤波器的基本设计方法;滤波器的特征,如线性相位特性和如何推导传递函数;如何通过配置零极点位置得到各种不同的整数型滤波器;如何通过串联整数型滤波器改进特性,还介绍一种最新设计方法以增加设计的灵活性。最后介绍一种整数型滤波器的实际应用。

7.1 基本设计概念

Lynn(1977)提出著名的整数型滤波器的设计方法,可用来设计低通、高通、带通及带阻滤波器。这种方法可概括为:首先在单位圆上均匀分布零点,这些零点表明在与其位置相对应的频率处幅值完全衰减,其次选择位于单位圆上的极点,精确抵消一些零点,当极点与零点抵消后,与此位置相对应的频率处幅值就不会衰减了。

因为单位圆上每一点代表不同的频率,零-极点位置决定了频率响应,这种滤波器是递归滤波的特定形式。

7.1.1 传递函数的一般形式

Lynn 的传递函数的一般形式为:

$$H_1(z) = \frac{[1 - z^{-m}]^p}{[1 - 2\cos(\theta)z^{-1} + z^{-2}]^t} \quad (7.1a)$$

$$H_2(z) = \frac{[1 + z^{-m}]^p}{[1 - 2\cos(\theta)z^{-1} + z^{-2}]^t} \quad (7.1b)$$

式中, m 为单位圆均匀分布的零点数; θ 为极点的角度;指数 p, t 为滤波器的阶数,它们直接影响放大倍数和衰减系数。 p, t 提高相同的倍数相当于串联滤波器。滤波器功能实现时, p, t 为非负整数。

7.1.2 极点配置

传递函数的分母是单位圆上一对共轭极点相乘。二个复根都在单位圆上,由 $e^{j\theta} = \cos\theta + j\sin\theta$ 可知, $e^{j\theta}$ 的所有值都位于单位圆上,因此,可得:

$$\text{分母} = (z - e^{j\theta})(z - e^{-j\theta}) \quad (7.2a)$$

展开后得:

$$\text{分母} = z^2 - (e^{j\theta} + e^{-j\theta})z + (e^{j\theta} + e^{-j\theta}) \quad (7.2b)$$

已知

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2} \quad (7.2c)$$

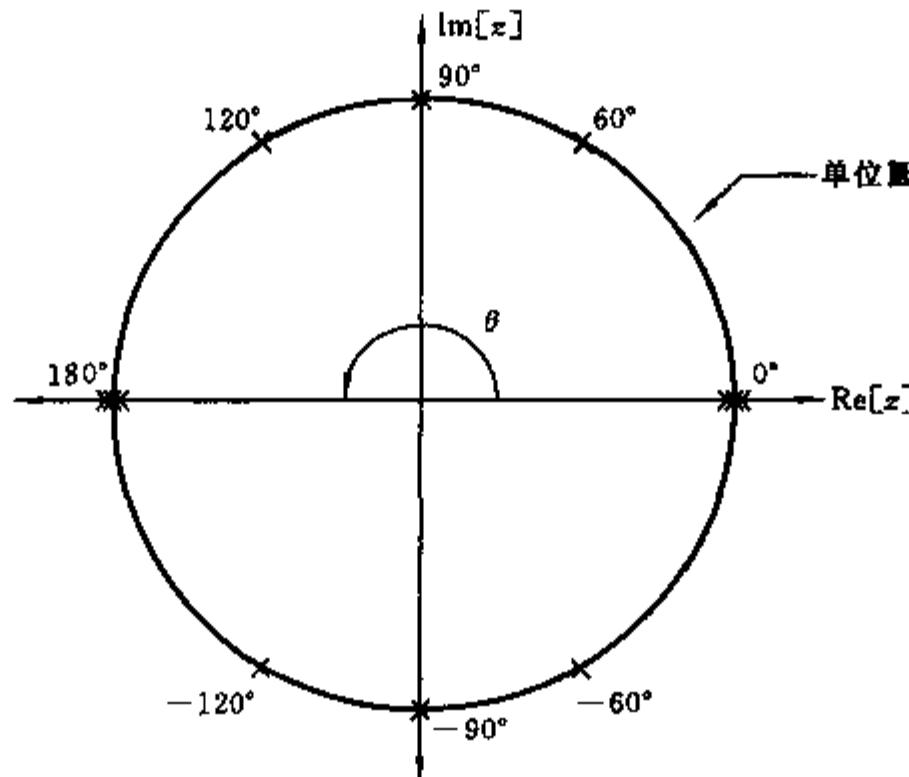
得到

$$\text{分母} = 1 - 2\cos(\theta)z^{-1} + z^{-2} \quad (7.2d)$$

一对共轭极点是整乘数或整除数的条件为 $2\cos(\theta)$ 是整数, 如图 7.1(a) 所示, 即 $\theta=0^\circ, \pm 60^\circ, \pm 90^\circ, \pm 120^\circ, \pm 180^\circ$, 如图 7.1(b) 所示。当系数为整数时, 乘法指令就可以被位移指令代替。

θ	$\cos(\theta)$	$2\cos(\theta)$
0°	1	2
$\pm 60^\circ$	$\pm 1/2$	1
$\pm 90^\circ$	0	0
$\pm 120^\circ$	$-1/2$	-1
180°	-1	-2

(a)



(b)

图 7.1 由式(7.1)传递函数的分母决定的可能的极点配置

(a) 能产生整数系数的仅有的极点; (b) 对应于(a)的零-极点图, 注意在 0° 和 180° 处有双极点

单位圆上的极点不可能完全抵消零点, 除非它的系数是整数型, 在其他位置上, 如 $\theta=15^\circ$, 由于浮点计算上的量化误差, 都存在小的不稳定区。

整数型滤波器的线性相位意味着信号频率各个部分经过滤波后都产生相同的时间延迟, 这对维持峰值的相互时间关系和输出波形的特征都是很重要的。

7.1.3 零点配置

极点在单位圆上的五个特定位置才有整数系数。设计者要求实现特定的截止频率时, 这个限制也影响了零点的配置和采样速率的选择。含零点时, 在传递函数中,

$$\text{分子} = (1 - z^{-m}) \quad (7.3a)$$

或者

$$\text{分子} = (1+z^{-m})$$

(7.3b)

式中, m 是正整数, 表示单位圆上均匀分布的零点个数。 $(1-z^{-m})$ 表示在 0° 处有一个零点, 其余零点按 $(360^\circ/m)$ 角均匀分布。为证明这一点, 令 $(1-z^{-m})=0$, 得 $z^{-m}=1$ 或 $z^m=1$ 。如果 z 在单位圆上, 则得到 $z=(1, 0)$ 且有 2π 的周期。即 $z^m=e^{j2\pi}=1$, 所以 $z=e^{j(2\pi/m)k}$, 给 m, n 以不同的整数值, 零点就在单位圆上按 $(360^\circ/m)$ 角度均匀分布, 并从零点 $z=1$ 开始。

当 n 超出 $n=\{0, 1, 2, \dots, m-1\}$ 的范围时也可以得到同样的结果。式(7.3b)说明了 m 个以 $(360^\circ/m)$ 间隔均匀分布于单位圆上的零点。设 $z^{-m}=-1$ 或 $z^m=-1$, 可以用类似的方法求出 z 的解为 $z=e^{j[(2n+1)/m]\pi}$, 其中 n 和 m 都是整数。 $(1-z^{-m})$ 与 $(1+z^{-m})$ 的差别在于 $1/2 \times (360/m)^\circ$ 的转角。图 7.2 所示的是它们的比较图。当 m 是奇数时, $(1+z^{-m})$ 总有一个在 180° 的零点, 而 $(1-z^{-m})$ 总是表现为“溢出”。

解法: $1-z^{-m}=0$

解法: $1+z^{-m}=0$

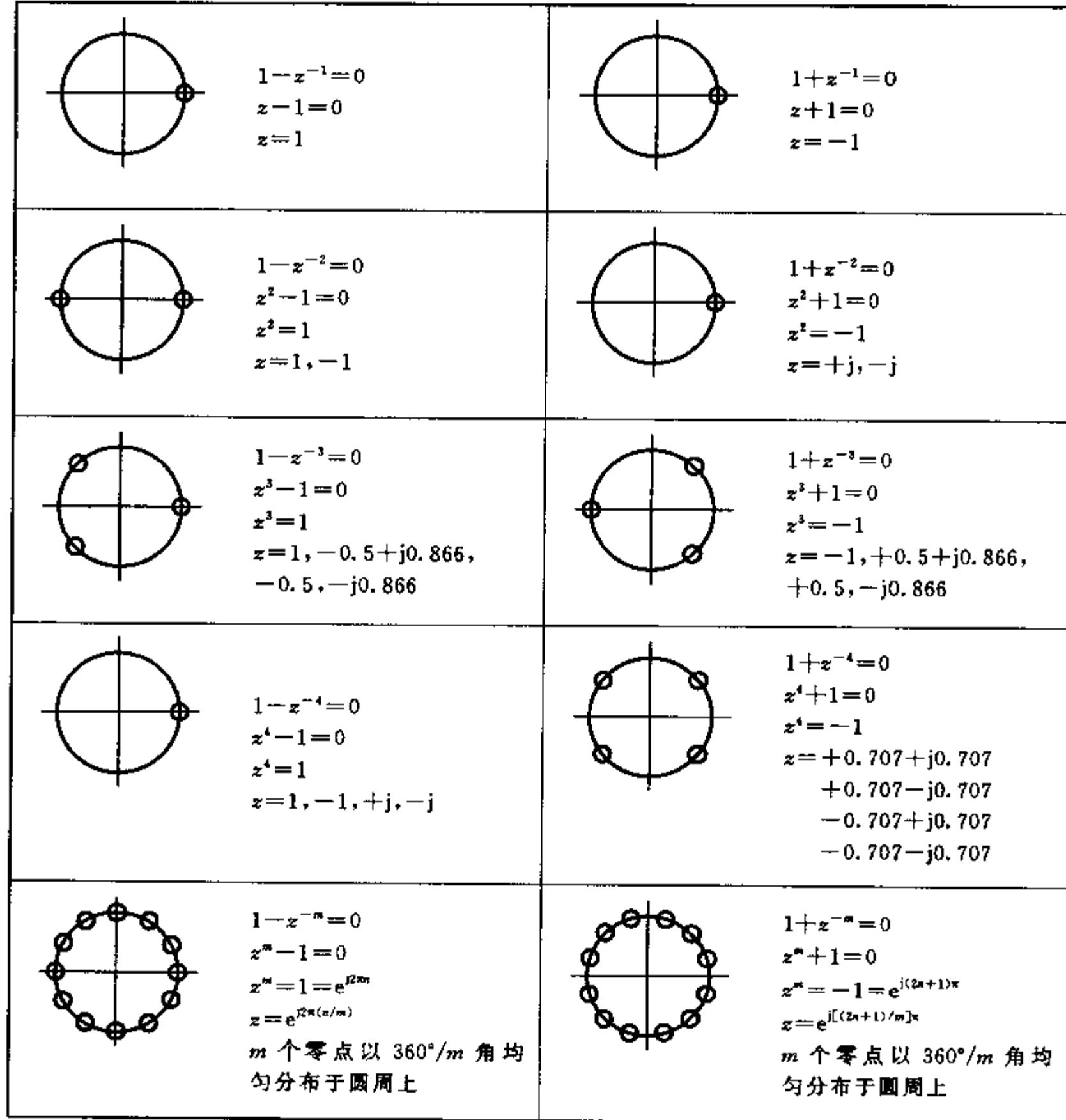


图 7.2 不同 m 值时, $(1-z^{-m})$ 与 $(1+z^{-m})$ 的零点对应关系

当 m 为正数时, 二者的零点互成“翻转”关系

7.1.4 稳定性及设计效益

对于一个稳定的、满足要求的滤波器，它的传递函数中的分母的最高次幂不能大于分子中零点的个数，即极点个数不能多于零点个数。在表示滤波器的式(7.1)中坐标原点处的点不显现，把传递函数都换成正幂数，如一个低通滤波器有五个零点，就可看到分母的五个极点（乘以 z^5/z^5 ）。

$$H(z) = \frac{1 - z^{-5}}{1 - z^{-1}} \times \frac{z^5}{z^5} = \frac{z^5 - 1}{z^4(z - 1)}$$

坐标原点处的极点有延迟作用，在频域中，坐标原点中 z 的阶数 m 决定了时间延迟 m 个采样周期。

有限冲激响应滤波器(FIR)的极点都在原点处，称无关极点。把式(7.1)中不是无关极点的极点用零点抵消就成为FIR滤波器。传递函数可以表示为不含抵消零、极点的式子。例如

$$H(z) = \frac{1 - z^{-m}}{1 - z^{-1}} = 1 + z^{-1} + z^{-2} + z^{-3} + \cdots + z^{-m+1} = \frac{Y(z)}{X(z)} \quad (7.4)$$

由于递推形式滤波器在计算上更加方便，如果 m 较大，则非递推的传递函数需要 m 次加法。而递推传递函数只需一次加法和一次减法，所以递推形式要比非递推形式大大减少计算量。

7.2 低通整数型滤波器

用式(7.1)设计低通滤波器时，设极点 $z=(1,0)$ ，但当 $\cos\theta=0$ 时，在 $z=(1,0)$ 处有两个极点，所以要加另一个零点 $z=(1,0)$ 或去掉一个极点。后者可使传递函数更简便。式(7.2)中应该用 $(1-z^{-m})$ ，而不是 $(1+z^{-m})$ ，因为它的零点在单位圆的 0° 处，这样传递函数为

$$H(z) = \frac{1 - z^{-m}}{1 - z^{-1}} \quad (7.5)$$

这个滤波器的低频旁瓣幅值比它的高频旁瓣的大，它放大低频，抑制高频。旁瓣是由位于原点处的极点引起的。图7.3所示的是当 $m=10$ 时的幅频和相频特性。对幅频特性的直观认识可用第四章介绍的橡胶膜片技术。在 z 域中，有一块平整且有弹性的薄膜，零点位置的薄膜被“钉住”，在极点位置，极点的作用是拉紧薄膜，极点相乘使这个位置的薄膜拉得更紧，在极点周围形成一个凸起。

从图7.3所示的高度可推出单位圆上的幅值响应。有效的幅值可从 $\theta=0^\circ$ 到 180° 中得到。大于 180° 范围是 0° ~ 180° 的映射，在此范围内，数字滤波器不响应任何频率，应该预先用模拟抗混叠滤波器滤除这个范围的所有信号。

为实现较低的截止频率，可以增加单位圆上的零点数或用较低的采样频率。前一种方法通常较好，它是锐截止的方法，可减少低速采样率时的信号丢失。最后，由于零点数增加，放大系数也会增加，这将产生大量不必要的输出和溢出错误。

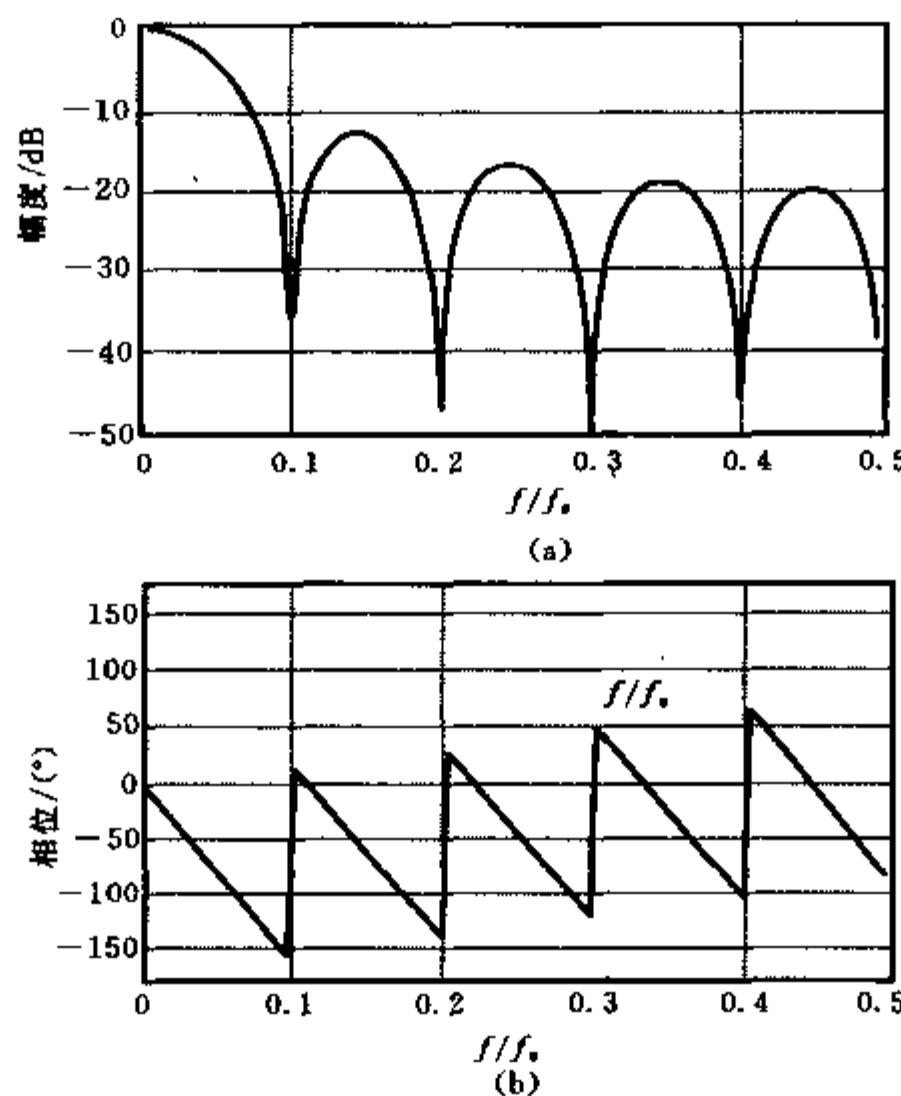


图 7.3 $m=10$ 的低通滤波器
(a)振幅响应; (b)相位响应

7.3 高通整数型滤波器

通常依据所要求的滤波器的截止频率来设计高通滤波器。

7.3.1 标准高通滤波器的设计

用 7.2 节的方法设计极点在 $z=(-1, 0)$ 、相角 $\theta=180^\circ$ 的滤波器时, 其分子的表达式必须参考 7.1.1 小节中的传递函数的分子表达式, 为了能实现这个滤波器, 在零点 $z=(-1, 0)$, 指数 m 为偶数时, 用 $(1-z^{-m})$ 表达式, 否则, 用 $(1+z^{-m})$ 表达式。对分母来说, 有两个极点都在 $(-1, 0)$ 处, 去掉一个含极点的因素, 简化后得

$$H(z) = \frac{1-z^{-m}}{1+z^{-1}} \quad (m \text{ 为偶数}) \quad (7.6a)$$

或 $H(z) = \frac{1+z^{-m}}{1+z^{-1}} \quad (m \text{ 为奇数}) \quad (7.6b)$

采样频率要大于 2 倍最高的输入信号频率。一个低通滤波器可消除大于 $1/2$ 采样频率的所有频率。当高通滤波器的截止频率大于 $1/2$ 采样频率时, 高通滤波器是不可能实现的。

为建立一个高通滤波器, 用极点抵消在单位圆上 $\theta=180^\circ$ 的零点。当实际滤波器截止频率大于 $1/4$ 采样频率 ($m \geq 4$) 时, 零点将增加, 带宽变窄, 要使频带带宽加大, 可采用下面介绍的减法设计法。

7.3.2 高通滤波器的减法设计法

一个复合滤波器的频率响应,可以用两个具有相同传输延迟的线性相位滤波器的输出相加得到,也就是各个滤波器响应的代数和。高通滤波器可以设计为一个全通滤波器与一个低通滤波器相减而成。如图 7.4 所示,全通滤波器是一个常系数的纯滞后滤波器,它可表示为 $H_u(z) = Az^{-m}$ (A 是放大系数, m 是零点个数)。理想情况下, $H_u(z)$ 与 $H_{low}(z)$ (低通) 有相同直流放大系数。如果 $A=2^i$ (i 为整数), 则滤波器的运算速度将更快, 因为这时可以用移位运算来实现高通滤波器。为实现最小相位不失真, 全通滤波器的延迟环节应与低通滤波器的延迟环节数目相同, 即 $\angle H_u(z) = \angle H_{low}(z)$, 从多零点低截止频率的低通滤波器得到低截止频率的高通滤波器。

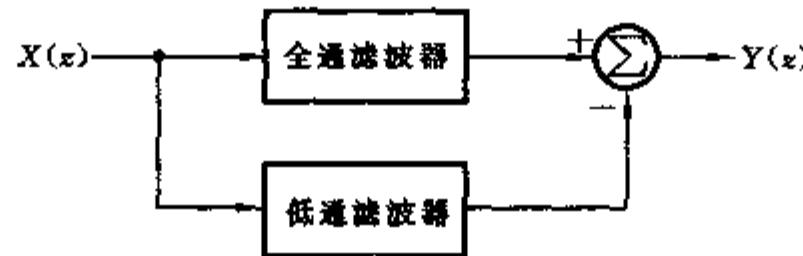


图 7.4 从全通滤波器中减去低通滤波器的框图

7.3.3 特殊的高通整数型滤波器

在特殊情况下, 单位圆上的零点将被极点潜在抵消, 这种情况在因子为 $(1-z^{-m})$, 而 $m=2, 4$ 或 6 时会出现。这种情况可以扩展到因子为 $(1-z^{-m})$, 同时包含有非整数因子, m 为任何偶数时。 $m=2$ 时, 180° 处的零点将被抵消, 设计的高通滤波器具有起于 0° 的锐截止频率。 $m=4$ 时, 可用 180° 和 90° 处的共轭极点抵消零点。当滤波器要通过 $1/4$ 采样频率时, 只抵消 180° 处的零点即可, 同样, $m=6$ 时抵消 $180^\circ, 120^\circ$ 和 60° 处的零点能达到同样的效果。

7.4 带通和带阻整数型滤波器

在设计带通整数型滤波器时要用到传递函数式(7.1)。但能产生整数系数的极点只能在 $\theta=60^\circ, 90^\circ$ 和 120° 处。选择采样频率时要保证通频带频率在极点处, 一对共轭极点的配置要求能在通频带中抵消零点。传递函数的分子表达式是选 $(1-z^{-m})$ 还是 $(1+z^{-m})$ 的原则是零点的位置能使 m 取得合适的值。零点数的选择要依据带宽的宽窄来进行。为使带宽变窄, 要增加 m 值, 即增加零点。然而放大系数与 m 一起增大, 将导致输出增大, 超过字长, 造成溢出错误。零点增加时,(1)带宽变窄,(2)旁瓣的幅值减小,(3)截止陡度增加,(4)差分等式要求有更多的过去输入值和采样数据。

带阻滤波器的设计有两种方法:一是, 在所要衰减的频率处设置零点;二是, 如 7.3.2 小节介绍的方法, 用一个全通滤波器和一个带通滤波器相减。

7.5 滤波器级联效应

滤波器的阶数表示相同滤波器的串联数, 前一个滤波器的输出是后一个的输入。增加传递

函数式(7.1)的阶数,即以相同整数倍乘以指数 p 和 t 。 n 阶滤波器幅值为 $|H(z)|^n$, n 增加,放大系数增加,所以这时要注意溢出错误。通过右移输出量可减小放大系数,然而这会导致量化误差。

前面介绍的滤波器都有相当大的旁瓣幅值和差的截止性。偶数倍地增加滤波器的阶数,旁瓣将会变小。如图 7.5 所示,增加阶数,曲线会迅速衰减。例如,由 $2\beta\pi$ 到 $4\beta\pi$,旁瓣越来越小。

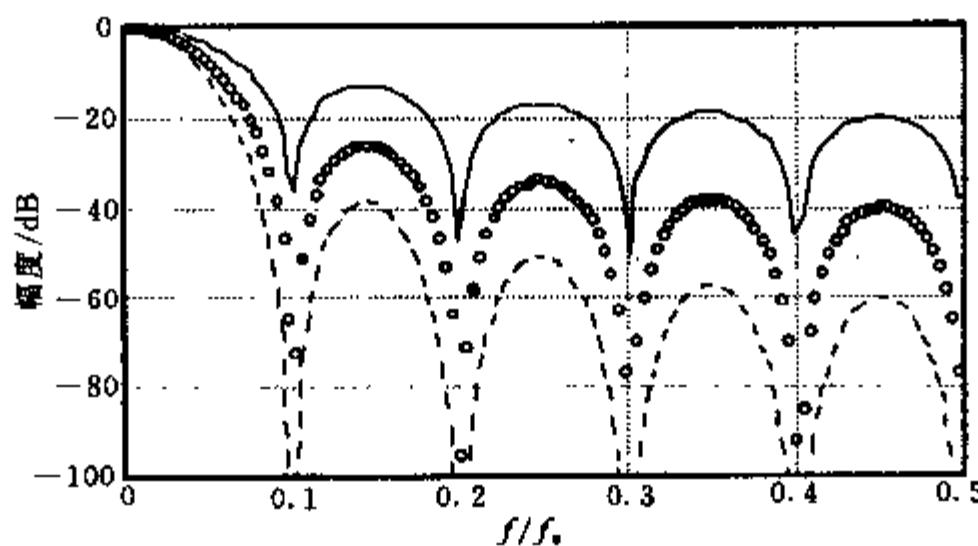


图 7.5 滤波器级数对增益的影响

这是 $m=10$ 的高通滤波器的例子;
实线:一阶,与图 7.3 滤波器一样;圆圈:二阶;虚线:三阶

滤波器的阶数既可为偶数,也可为奇数,但阶数为偶数时,相位响应有更好的特性,该类型的滤波器的相位特性可总结为:

- (1) 偶次阶滤波器具有纯线性相频特性,没有相位失真。
- (2) 奇次阶滤波器具有逐段线性相频特性,在相角为 180° 处相频特性不连续且幅值为零。

逐段线性特性表现为滤波器在相频特性变化时可以大大衰减信号。如果抑制频带时不能将频带很好抑制掉,在此范围内,那些具有很大能量已经过滤的信号会产生相位扭曲。也可把非理想滤波器组合起来,复杂的滤波器就是由一些简单滤波器组合起来的,如前面所述的滤波器相减方法得到的滤波器就是用这一方法得到的。

7.6 其他快速设计方法

在对 ECG 数据进行数字滤波时,Principe 和 Smimith 采用了与 Lynn's 略有不同的设计方法。他们的方法是根据 5.6 节中介绍的频率采样技术来设计的。这种设计方法分两步进行:第一步设置单位圆上的零点,产生几个带通频率,其中一个与所要求的频率对应;第二步,设置零点,抑制由放大系数产生的阻带。以图 7.6 所示的为例,所有极点都位于原点处,这些 FIR 滤波器性能稳定。用零点的位置抑制放大系数比零极点抵消的方法更加灵活。由分子 $1-2\cos(\theta)z^{-1}+z^{-2}$ 可求出共轭的零点。考虑到系数计算,把零点配置在只需二进制位移和加法指令计算系数而不需要乘法指令的位置。对于 $2\cos\theta$ 的三次逼近,可把零点配置在因式 $(1-z^{-m})$ 或 $(1+z^{-m})$ 中,而当 $m < 8$ 时,误差不超过 6.5%。这种方法不用乘法指令,而是在轻微降低带阻的抑制程度下,零点移到易于计算系数的位置上。

前面所述的滤波器或者是逐次分段线性的或为纯线性的,设计时,要求滤波器有锐截止频

率,没有相位失真。在这种情况下,内部极点配在单位圆零点附近,可以增大带通频率;对于抑制带阻频率,无论极点是否全部抵消零点,IIR 滤波器要优于 FIR 滤波器,另外,通常不希望用浮点法表示极点的系数。

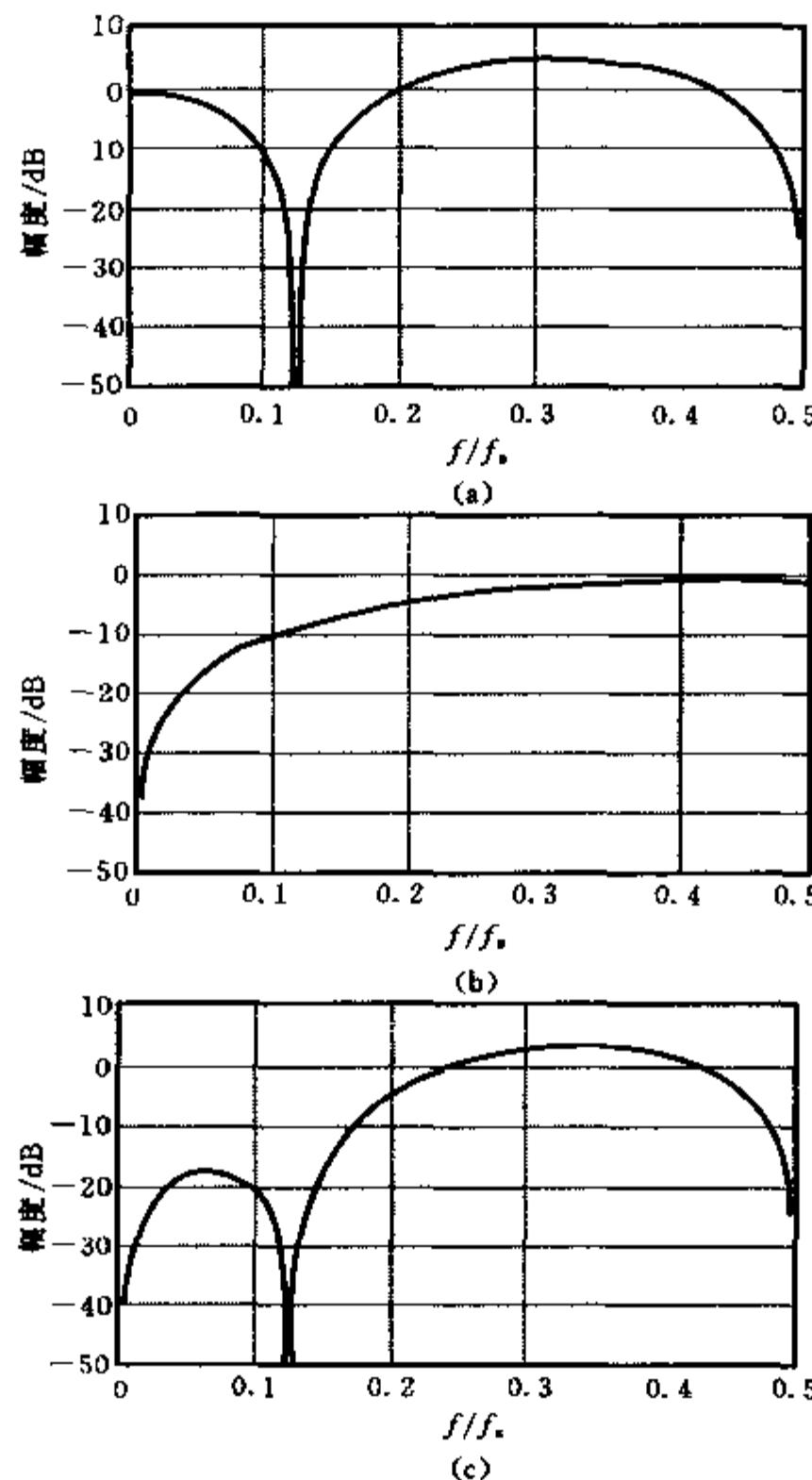


图 7.6 用零点产生的高通滤波器
(a)幅值输出;(b)正幅值;(c)相加产生幅值

Thakor 和 Moreau(1987)用“量化系数法”解决了这个问题。极点的系数由 x/zy 表示, x 是 1 与 $(2y-1)$ 之间的整数, y 是非负整数, 叫量化位。 y 由设计者决定, 受计算机的字长限制(对于 8 位机, $y=8$)。量化分解系数, 如 $1/8$, 可用位移指令实现, 但如果系数为 $17/32$, 则需要多次用位移和加指令, 因此, 其速度不会比乘法指令快。

7.7 设计举例和设计工具

本章介绍了许多设计方法, 本节介绍一个有具体要求的设计例子来说明常用的分析滤波器特性的方法和求解滤波器差分方程的 C 语言程序。

7.7.1 一阶和二阶带通滤波器

设计一个带通滤波器,它具有24个零点,从 0° 开始,均匀分布在单位圆上。带通峰值在 60° 位置,幅值和相位的理论值已被给出。

该滤波器的传递函数为:

$$H(z) = \frac{1 - z^{-24}}{1 - z^{-1} + z^{-2}} \quad (7.7)$$

把 z 用 $e^{j\omega T}$ 替代,并产生绝对值相等的正负指数,即

$$\begin{aligned} H(\omega T) &= \frac{1 - e^{-j24\omega T}}{1 - e^{-j\omega T} + e^{-2j\omega T}} \\ &= \frac{e^{-j12\omega T}(e^{j12\omega T} - e^{-j12\omega T})}{e^{-j\omega T}(e^{j\omega T} - 1 + e^{-j\omega T})} \end{aligned} \quad (7.8)$$

由于
得

$$H(\omega T) = \frac{e^{-j12\omega T}[\cos(12\omega T) + j\sin(12\omega T) - \cos(12\omega T) + j\sin(12\omega T)]}{e^{-j\omega T}[\cos(\omega T) + j\sin(\omega T) - 1 + \cos(\omega T) - j\sin(\omega T)]} \quad (7.10)$$

合并后得

$$H(\omega T) = \frac{j2\sin(12\omega T)}{2\cos(\omega T) - 1} \times e^{-j11\omega T} \quad (7.11)$$

将 $j=e^{j\pi/2}$ 代入上式得 $H(\omega T) = \frac{2\sin(2\omega T)}{2\cos(\omega T) - 1} \times e^{j(\pi/2 - 11\omega T)}$ (7.12)

图7.7(a)所示的实线表示幅频响应曲线,幅频响应为:

$$|H(\omega T)| = \left| \frac{2\sin(12\omega T)}{2\cos(\omega T) - 1} \right| \quad (7.13)$$

图7.7(b)所示的实线表示相频响应曲线,相频响应为:

$$\angle H(\omega T) = \frac{\pi}{2} - 11\omega T \quad (7.14)$$

接下来串上一个与它同样的滤波器,再计算串联后滤波器的幅值和相位,下面比较一阶和二阶滤波器。二阶滤波器传递函数为:

$$H(z) = \frac{(1 - z^{-24})^2}{(1 - z^{-1} + z^{-2})^2} \quad (7.15)$$

再次用 $e^{j\omega T}$ 代替 z ,得到二阶滤波器传递函数为:

$$\begin{aligned} H(\omega T) &= \left[\frac{2\sin(12\omega T)}{2\cos(\omega T) - 1} \times e^{j(\pi/2 - 11\omega T)} \right]^2 \\ &= \left[\frac{2\sin(12\omega T)}{2\cos(\omega T) - 1} \right]^2 \times e^{j(\pi - 22\omega T)} \end{aligned} \quad (7.16)$$

图7.7(a)所示的虚线表示幅频响应曲线,幅频响应为:

$$|H(\omega T)| = \left| \left(\frac{2\sin(12\omega T)}{2\cos(\omega T) - 1} \right)^2 \right| \quad (7.17)$$

图7.7(b)所示的虚线表示相频响应曲线,相频响应为:

$$\angle H(\omega T) = \pi - 22\omega T \quad (7.18)$$

比较两个滤波器的相位关系,可见一阶滤波器是逐段线性的,二阶滤波器是纯线性的。

为计算滤波器的增益,先求得临界频率时的幅值($\omega T = 60^\circ$),然后将此值代入式(7.17)后

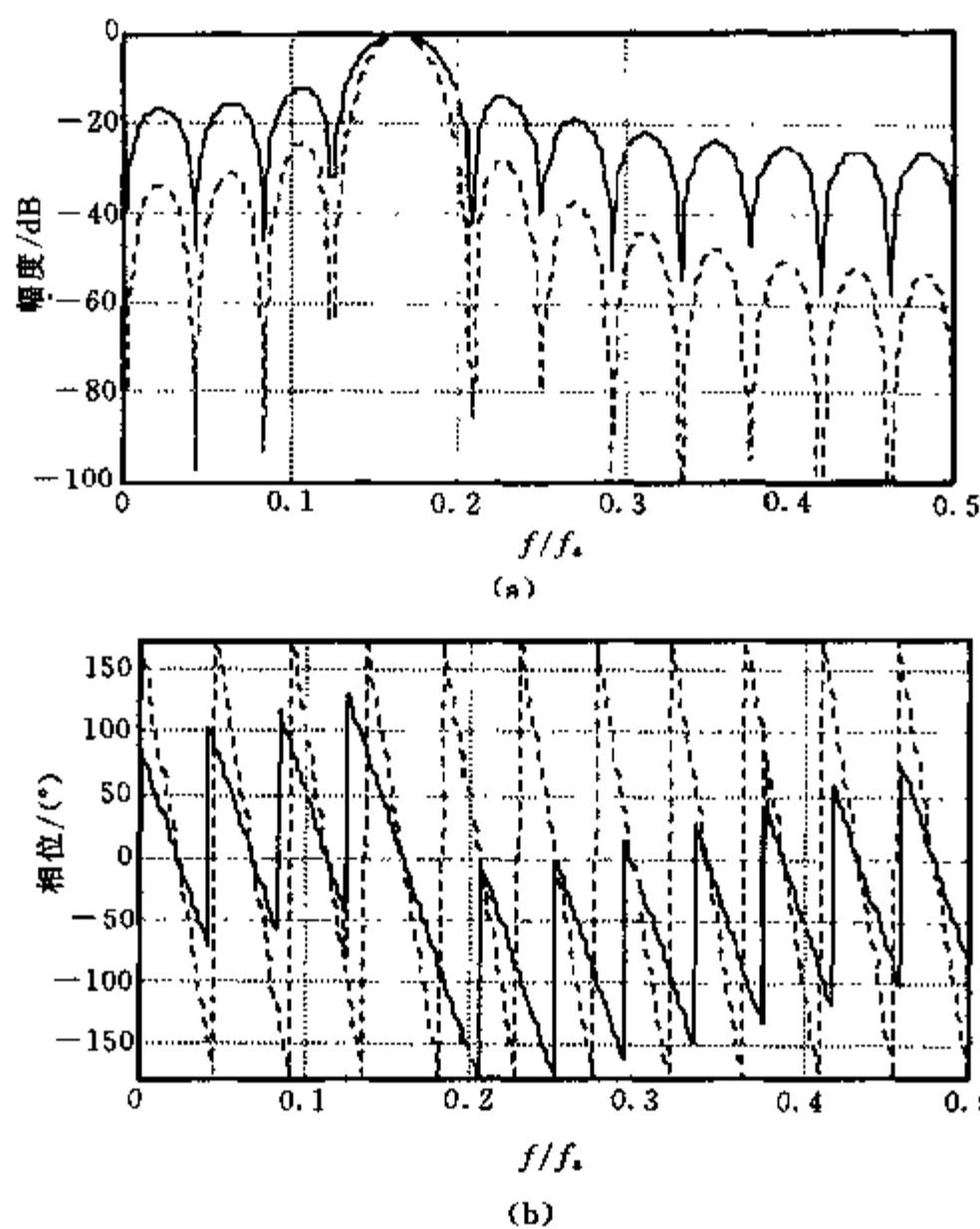


图 7.7 片段线性 FIR 带通滤波器

(a)幅值响应 实线:一阶滤波器,传递函数为 $H(z) = (1-z^{-24})/(1-z^{-1}+z^{-2})$;
虚线:二阶滤波,传递函数为 $H(z) = (1-z^{-24})^2/(1-z^{-1}+z^{-2})^2$;(b)相位响应
实线:一阶片段线性相位响应;虚线:二阶滤波器的真线性响应

得到一个不确定的值,即

$$\begin{aligned}|H(\omega T)| &= \left| \left[\frac{2\sin(12\omega T)}{2\cos(\omega T) - 1} \right]^2 \right| \\&= \left| \left[\frac{2\sin(60^\circ)}{2\cos(60^\circ) - 1} \right]^2 \right|_{\omega T = 60^\circ} = \frac{3}{0}\end{aligned}\quad (7.19)$$

为确定滤波器增益,用 L'Hopital's 定理先对上式的分子、分母分别微分,再求值,即

$$\begin{aligned}\frac{d(|H(\omega T)|)}{d(\omega T)} &= \left| \left[\frac{24\cos(12\omega T)}{-2\sin(\omega T)} \right]^2 \right|_{\omega T = 60^\circ} \\&= \left| \left[\frac{24}{-\sqrt{3}} \right]^2 \right| = 192\end{aligned}\quad (7.20)$$

二阶滤波器系数为 192,而一阶滤波器系数为 13.9,系数随阶次增加而增大,从图 7.7(a)可知,虽然系数增大,但旁瓣的衰减幅度比通频带旁瓣的衰减幅度大。

用滤波器的差分方程编写一个 C 语言程序来实现这种滤波器。滤波器差分方程为:

$$\begin{aligned}y(nT) &= 2y(nT - T) - 3y(nT - 2T) + 2y(nT - 3T) - y(nT - 4T) + x(nT) \\&\quad - 2x(nT - 24T) + x(nT - 48T)\end{aligned}\quad (7.21)$$

图 7.8 所示的是实时二阶带通滤波器的 C 语言程序,此程序并非是运行最快的,每个 for(cs)

语句对本循环内所有的值移位。

7.7.2 一阶低通滤波器

下面的简短 C 语言程序,可实现一个有 6 个零点的低通滤波器。先经过一个 A/D 转换采样,数据经滤波并返回。为得到前面的采样值,用一个 FIFO 缓冲器实现单位时间延迟。这个式子只有一个加法和一个减法指令。修改存储区的指针要优先用转移数据指令。这在增加滤波器零点时是要注意的。

含 6 个零点的低通滤波器的差分方程为:

$$Y(nT) = y(nT - T) + x(nT) - x(nT - 6T) \quad (7.22)$$

图 7.9 所示的是实现这个方程的 C 语言程序。

```
/* Bandpass filter implementation of
H(z)=[(1-z^-24)^2]/[(1-z^-1+z^-2)^2]
Notes: Static variables are automatically initialized to zero
Their scope is local to the function. They retain their value
when the function is exited and reentered.
Long integers are used for y, the output array. since this
filter has a gain of 192, Values for y can easily exceed the
values that a 16-bit integer can represent.

*/
long bandpass(int x_current)
{
    register int i;
    static int x[49];      /* history of input data points */
    static long y[5];       /* history of difference equation outputs */

    for (i=4; i>0; i--)          /* shift past outputs */
        y[i]=y[i-1];
    for (i=48; i>0; i--)         /* shift past inputs */
        x[i]=x[i-1];
    x[0]=x_current;           /* update input array with new data point */

    /* Implement difference equation */
    y[0]=2 * y[1]-3 * y[2]+2 * y[3]-y[4]+x[0]-2 * x[24]+x[48];

    return y[0];
}
```

图 7.8 二阶带通滤波器 C 语言程序

```

/* Low-pass filter implementation of

 $H(z) = (1 - z^{-6}) / (1 - z^{-1})$ 

Note 1: Static variables are initialized only once. Their scope
is local to the function. Unless set otherwise, they are
initialized to zero. They retain their values when the function
is exited and reentered.

Note 2: This line increments pointer x_6delay along the x array
and wraps it to the first element when its location is at the
last element. The ++ must be a prefix to x_6delay; it is
equivalent to x_delay + 1.

*/
int lowpass (int x_current)
{
    static x[6],           /* FIFO buffer of past samples */
    y,                     /* serves as both y(nT) and y(nT-T) */
    *x_6delay = &x[0];     /* pointer to x(nT-6T); see Note 1 */
    y += (x_current - *x_6delay); /* y(nT)=y(nT-T)+x(nT)-x(nT-6T) */
    *x_6delay = x_current;   /* x_current becomes x(nT-T) in FIFO */
    x_6delay = (x_6delay == &x[5])? &x[0] : ++x_6delay;
                           /* See Note 2 */

    return (y);
}

```

图 7.9 高效率实现一阶低通滤波器的 C 语言程序
用指针增量取代全部数据的移位

7.8 实验: 用于 ECG 分析的整数型滤波器

用于 ECG 分析的仪器往往要求具有实时性。这就要求每个采样点的数据处理必须在下一个采样点到来之前完成。在设计过程中, 大量限制条件说明要用 PC 机来完成。在本实验中, 要求设计几种滤波器并比较它们的性能。

在软件 UW DigiScope 中运行(F)ilters, (D)esign, i(N)teger 进入整数型滤波器设计界面。

(1) 用(G)enwave 函数产生正常 ECG 片断 1 和非正常 ECG 片断 5, 采样速率均为 100 次/s, 幅值分辨率为 8 位。

① 设计一个具有 6 个零点, 16.7Hz 中心频率的带通滤波器来处理以上信号。这种滤波器有时也用于根据 QRS 波测量心率的场合。

② 用这些滤波器对两个 ECG 信号进行滤波。因为这个滤波器具有增益功能, 所以要用(Y)sens 函数调整幅度。画出频响图。

③从 STDLIB 中读出单位脉冲响应 wps.dat，并对它进行滤波，画出频响图。用(P)Wr Spect 求单位脉冲响应的功率谱。其结果与滤波器的幅频响应是一样的，除非将频率轴放大一倍，因为单位脉冲信号是以 200 次/s 采样的，而不是用设计的滤波器的采样率 100 次/s 采样。用(M)easure 函数测量 3dB 点，并求带宽。注意，实际带宽只有测量值的一半，因为频率轴放大了一倍。计算滤波器的 Q 值。

④设计两个带通滤波器，零点数分别为 18 和 48，其余与上面完全一样。用这些滤波器重复步骤②和③。因为零点数增加，所以幅值也增加，因而要注意溢出问题，特别是对于单位脉冲。一旦溢出，输出就不连续。DigiScope 的 16 位数据表示出现溢出。在这种情况下，应该在滤波之前对单位脉冲信号的幅值进行衰减。以上设计的三种滤波器哪一种最适合探测 QRS 复波？为什么？

(2)设计一个有 10 个零点的低通滤波器。对正常 ECG 片断 1 进行滤波并画输出图。哪些波衰减了？哪些波放大了？3dB 通带在哪里？

(3)以 180 次/s 采样率产生正常 ECG 信号，其中含 10% 的 60Hz 噪声和 10% 的随机噪声。设计一个简单的低通滤波器，它具有①15Hz 的低通带宽用于衰减随机噪声，②完全抑制 60Hz 噪声的功能。测量实际 3dB 带宽，并对设计的滤波器性能进行评价。

复习思考题

- 7.1 为什么在数字滤波器的差分方程中使用整数系数更好？
- 7.2 说明为什么当所有系数都限定为整数时设计数字滤波器更难？
- 7.3 说明式(7.1)的分母对于所有的 Q 值都在单位圆上产生两个极点。什么样的 Q 值产生整数系数？为什么必须避免产生浮点数的 Q 值？
- 7.4 从数学上解释为什么传递函数分子 $(1+z^{-m})$ 和 $(1-z^{-m})$ 在单位圆上会产生零点？什么时候用 $(1-z^{-m})$ 比用 $(1+z^{-m})$ 更好？
- 7.5 什么时候式(7.1)是一个 FIR 滤波器？如何使它变得不稳定？将 FIR 滤波器的传递函数表达为递推形式有什么好处？
- 7.6 式(7.1)什么时候表示一个低通滤波器？讨论当希望滤波器具有低的截止频率时，设计者应考虑滤波器的什么特性？
- 7.7 严格线性相位响应和片段线性相位响应有什么差别？具有片段线性相位响应的滤波器能作为某个真线性响应的滤波器吗？
- 7.8 列举当整数型数字滤波器与自身级联时幅值和相位改变的四条途径，这些变化为什么有益或者为什么无益？
- 7.9 如果极点和零点处于非整数位置，如何使数字滤波器仍然具有高的计算效率？用此原则描述两种方法。
- 7.10 一个滤波器的 z 变换的传递函数式为 $H(z)=1-z^{-6}$ ，计算其幅值和相位响应。
- 7.11 传递函数的分子式为 $(1-z^{-10})$ 时，求其零点分布。
- 7.12 一个滤波器有 12 个零点，起始于 DC(直流)，并沿单位圆以 30° 均匀分布，有三个极点分别位于 $z=+0.9$ 以及 $z=\pm j$ 。采样频率为 360 次/s，①在什么频率处输出幅值最大？②

此频率处的增益多大?

7.13 一个数字滤波器具有如下的传递函数:

$$H(z) = \frac{1 - z^{-6}}{(1 - z^{-1})(1 - z^{-1} + z^{-2})}$$

①这种滤波器与什么样的传统滤波器相当? ②直流增益为多少?

7.14 对于具有如下传递函数的滤波器:

$$H(z) = \frac{1 - z^{-8}}{1 + z^{-2}}$$

求①幅值响应; ②相位响应; ③差分方程。

7.15 对于一个具有如下传递函数的滤波器:

$$H(z) = \frac{(1 - z^{-8})^2}{(1 + z^{-2})^2}$$

①它对应于什么样的传统的滤波器? ②画出其零-极点图; ③求幅值响应; ④求差分方程。

7.16 传递函数为:

$$H(z) = \frac{1 - z^{-6}}{1 - z^{-1}}$$

的滤波器的增益为多少?

7.17 具有传递函数

$$H(z) = \frac{1 - z^{-256}}{1 - z^{-128}}$$

的滤波器对应于什么样的传统滤波器?

7.18 具有以下传递函数

$$H(z) = \frac{1 - z^{-200}}{1 - z^{-2}}$$

的滤波器对应于什么样的传统滤波器?

7.19 一个数字滤波器有 4 个零点 $z = \pm 1$ 和 $z = \pm j$ 和 4 个极点 $z = 0, z = 0, z = \pm j$, 采样频率为 800 次/s。试问在什么频率下出现最大输出幅值?

7.20 当采样速率为 100 次/s 时, 具有如下传递函数的数字滤波器的最大增益大致出现在什么频率时?

$$H(z) = \frac{1 - z^{-36}}{1 - z^{-1} + z^{-2}}$$

7.21 一个滤波器的 z 变换为 $H(z) = 1 - z^{-360}$, 将 $x(t) = 100\sin(2\pi 10t)$ 施加在输入端, 采样频率为 720 次/s。①滤波器的峰-峰值输出多大? ②如果施加一个单位阶跃输入, 在 361 个采样周期后输出是多少? ③在何处分布极点可以将这个滤波器转换成整数系数的带通滤波器?

7.22 对于 $H(z) = \frac{1 - z^{-100}}{1 - z^{-2}}$ 的滤波器, 以 200 次/s 采样, 求相位延迟(ms)。

7.23 一个滤波器有 8 个位于单位圆上的零点, 起于 DC, 增量为 45° , 有两个极点位于 $z = \pm j$ 。采样频率为 360 次/s。求滤波器的增益。

第八章 自适应滤波器

本章讨论如何建立自适应滤波器以达到消除噪声和提取信号的目的。自适应滤波方法比不可调滤波方法优越,因为它不需要事先知道信号或噪声的特性。自适应滤波器通过采用期望值和负反馈值进行综合判断的方法来改变滤波器的参数。自适应滤波方法已被应用于从心电图信号中滤去频率为60Hz的干扰信号,分离出胎儿ECG信号,增强P波,去掉心电图中伪迹等方面。本章介绍了自适应数字滤波器的基本原理,列举了某些直接应用的实例。

在数字信号处理的应用中,通常一个有效信号中会有干扰信号掺杂进去。在不可调滤波方法中,得到理想滤波的基本前提是,必须知道信号和干扰的特性,通常假定,这两种信号稳定或在广义上稳定。自适应滤波器可获得输入信号的统计资料并在信号变化很慢时对它们进行跟踪。

8.1 基本的噪声消除器模型

在生物医学信号处理中,自适应滤波法对消除干扰影响是十分有用的。图8.1所示的是一种用自适应滤波器消除噪声的一般方法。在离散情况下,可将主输入设为 $s(nT) + n_0(nT)$ 。干扰是被加入的,并与信号源无关,另一个输入到滤波器的参考信号为干扰 $n_1(nT)$,它反馈到滤波器,产生输出 $\xi(nT)$, $\xi(nT)$ 与 $n_0(nT)$ 近似相等。这样干扰 $n_1(nT)$ 便与 $n_0(nT)$ 不自觉地联系在一起。

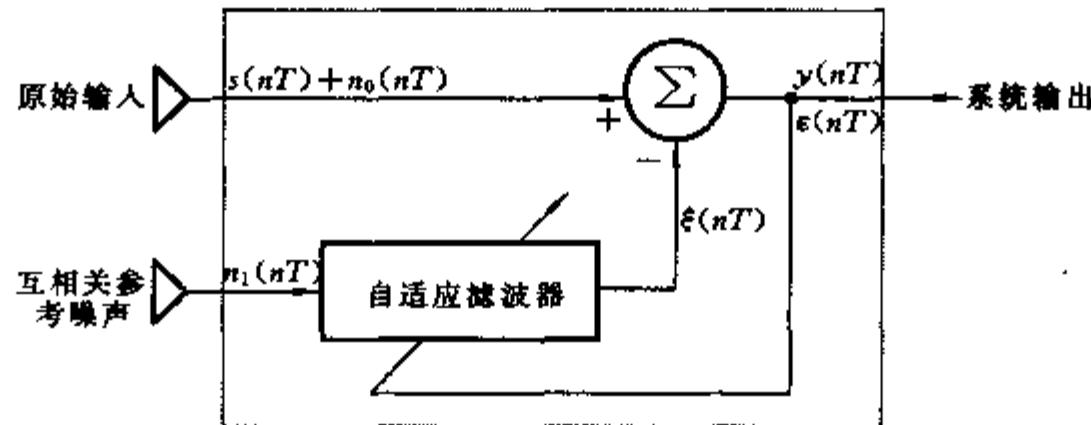


图8.1 自适应滤波器消除噪声的结构图

系统输出 $y(nT)$ 是由主输入减去 $\xi(nT)$ 得到的。此输出也被用于调整自适应滤波器滤波系数 $\{\omega(1, 2, \dots, p)\}$ 的误差信号 $\epsilon(nT)$ 。输出的差分方程为:

$$y(nT) = s(nT) + n_0(nT) - \xi(nT) \quad (8.1)$$

简化上式各项,将 (nT) 隐去,两端平方得

$$y^2 = s^2 + (n_0 - \xi)^2 + 2s(n_0 - \xi) \quad (8.2)$$

两边各取期望值,即

$$E[y^2] = E[s^2] + E[(n_0 - \xi)^2] + 2E[s(n_0 - \xi)]$$

$$= E[s^2] + E[(n_0 - \xi)^2] \quad (8.3)$$

由于信号源 $E[s^2]$ 不受滤波器调节的影响, 故期望最小值为:

$$\min E[y^2] = E[s^2] + \min E[(n_0 - \xi)^2] \quad (8.4)$$

当系统输出值根据式(8.4)取最小时, $(n_0 - \xi)$ 均方误差(MSE)最小, 且滤波器通过调节可得到综合噪声($\xi \approx n_0$)。这种通过均方误差反复改变滤波器系数的方法叫做最小均方算法(LMS)。

8.2 自适应滤波器消除 60Hz 干扰的正弦波模型

有许多资料表明, 心电图放大器容易掺杂进正弦 60Hz 线频率干扰信号。正如第五章所讨论的那样, 非循环带阻滤波器可用来削弱 60Hz 干扰信号。这种设计方法的缺点在于, 削弱输出干扰信号的同时, (1) 滤波器同时消除了信号中 60Hz 的组成部分, (2) 滤波器有缓慢的衰减, 不必要地衰减了有用信号, (3) 如果干扰信号的幅值频率特性有所改变, 则滤波器为非最优的。自适应横向滤波(选择延迟线)便于削弱干扰信号, 同时在不稳定过程中能维持最优的信噪比。

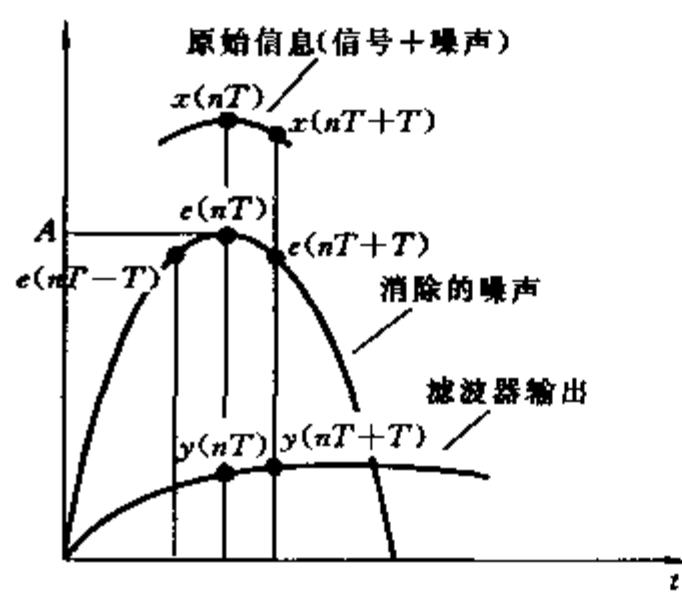


图 8.2 自适应消除 60Hz 干扰的正弦波模型

另一降低 60Hz 噪声的简便方法是模拟一个 60Hz 的正弦波的参考源。自适应滤波器的唯一参数是正弦波的幅值。图 8.2 所示的 3 个信号: $x(nT)$ 是掺杂了 60Hz 干扰信号的心电图输入信号; $e(nT)$ 是用 60Hz 正弦波建立的干扰信号; $y(nT)$ 是滤波器的输出。

上述算法是由建立一个假定干扰信号开始的, 幅值为 A , 频率为 ω 。

$$e(nT) = A \sin(\omega nT) \quad (8.5)$$

上式中, 用 $(nT - T)$ 代替 (nT) , 就得到前一周期的假定干扰信号的表达式:

$$e(nT - T) = A \sin(\omega nT - \omega T) \quad (8.6)$$

同样, 在等式(8.5)中用 $(nT + T)$ 代替 (nT) , 就可得

到后一个周期的表达式:

$$e(nT + T) = A \sin(\omega T + \omega nT) \quad (8.7)$$

由三角恒等式

$$\sin(\alpha + \beta) = 2 \sin \alpha \cos \beta - \sin(\alpha - \beta) \quad (8.8)$$

令

$$\alpha = \omega nT, \beta = \omega T \quad (8.9)$$

将式(8.8)和式(8.9)扩展, 建立超前一个周期的形如式(8.7)的等式:

$$e(nT + T) = 2A \sin(\omega nT) \cos(\omega T) - A \sin(\omega nT - \omega T) \quad (8.10)$$

注意到上式第一个下面划线的项同式(8.5)的相同, 第二个下面划线的项同式(8.6)的相同。 $\cos(\omega T)$ 这一项是由被分离出的干扰信号的频率 ω 和采样频率决定的常项 $f_s = 1/T$ 所决定的, 表达式为:

$$N = \cos(\omega T) = \cos(2\pi f_s T) \quad (8.11)$$

因此, 式(8.10)可以写成: 把采样正弦干扰波的超前估测信号与当前的采样值、滞后的采样值

联系在一起的式子,即

$$e(nT + T) = 2Ne(nT) - e(nT - T) \quad (8.12)$$

滤波器的输出是输入信号和估测信号的差值,即

$$y(nT + T) = x(nT + T) - e(nT + T) \quad (8.13)$$

因此,如果输入仅仅是干扰信号,建立的信号能完全跟踪它(即模拟),则输出将等于0,若ECG被噪声输入迭加,则输出中无噪声项。

现在,ECG信号被作为瞬态处理,滤波器反复地试图改变参考输入的“权重”或幅值,以适应期望的信号,60Hz的噪声。滤波器主要学习作为主输入的噪声,并把它消除。为了反复调整滤波器去适应噪声信号的变化,需要用反馈值去调整建立的正弦波信号的各采样时刻的幅值。

差分方程定义为:

$$f(nT + T) = [x(nT + T) - e(nT + T)] - [x(nT) - e(nT)] \quad (8.14)$$

由图8.3可了解此功能。在式(8.5)中最初的噪声 $e(nT)$ 的模型是假设为一个简单的正弦波,不显示直流组成部分。然而一般的情况下,在输入信号 $x(nT)$ 中用 V_{dc} 代替直流分量,即

$$V_{dc}(nT + T) = x(nT + T) - e(nT + T) \quad (8.15)$$

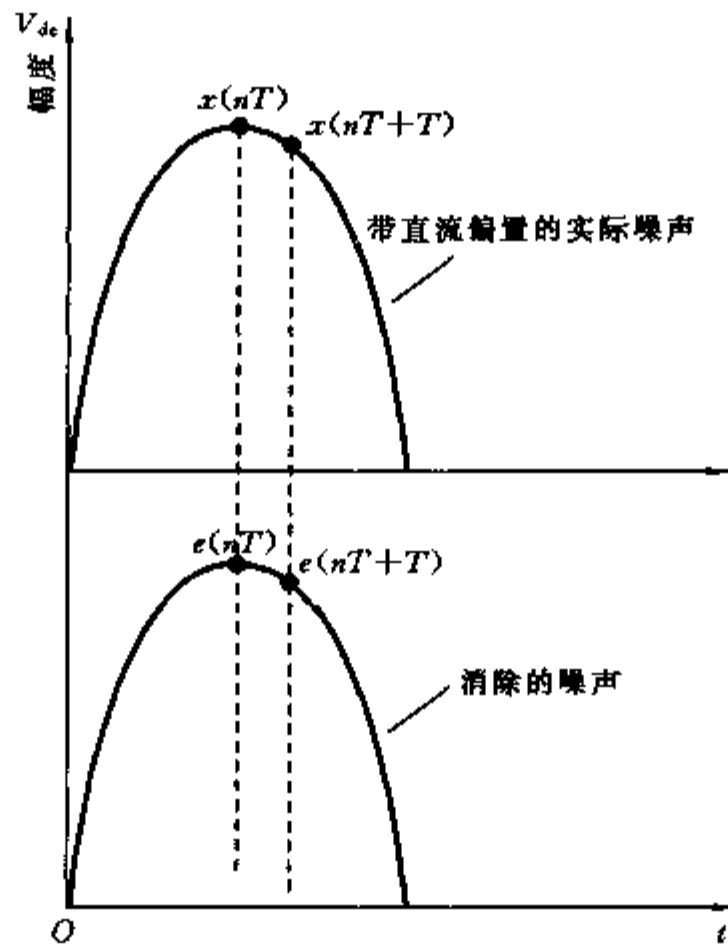


图8.3 实际干扰波会含有估计信号最初模型的直流偏移量

和

$$V_{dc}(nT) = x(nT) - e(nT) \quad (8.16)$$

假定直流分量没有较大地改变采样时刻的离散值,则

$$V_{dc}(nT + T) - V_{dc}(nT) = 0 \quad (8.17)$$

在式(8.15)或式(8.16)中代表直流分量的各项相减是式(8.14)的作用的基础。它在去掉直流部分的同时,比较输入和预测波形。

用 $f(nT+T)$ 来决定预测的 $e(nT)$ 是太大还是太小。如果 $f(nT+T)=0$,值是正确的,并且不必再去调整下一个估计值,或者

$$e(nT + T) = e(nT + T) \quad (8.18)$$

如果 $f(nT + T) > 0$, 则较估计值低, 应把估计值提高一个很小的步长 d , 即

$$e(nT + T) = e(nT + T) + d \quad (8.19)$$

若 $f(nT + T) < 0$, 则较估计值高, 应把估计值降低一个很小的步长 d , 即

$$e(nT + T) = e(nT + T) - d \quad (8.20)$$

d 的选择要建立在实验的基础上, 并依靠滤波器适应干扰噪声变化的速度。如果 d 大, 则滤波器会很快在受到 60Hz 噪声干扰后调整它的参数。若 d 稍大, 则滤波器不能准确跟踪噪声的变化。这个结果已在曾有准确幅值的预测信号的小振荡中被发现。若 d 小, 则滤波器需要一个长的学习过程, 但会对更平缓的噪声输出提供准确跟踪。如果 d 太大或太小, 则滤波器将永远不能准确跟踪噪声。

一般的 d 要比用来代替信号的最小有效位的整数值小。例如, 如果 8 位 ADC 的数值范围是从 0~225, 那么 d 的最佳值可能是 $1/4$ 。

要得到式(8.12)中的预测信号, 需要乘以式(8.11)中给出的 N 。设采样频率为 500 次/s, 噪声频率为 60Hz, 则

$$N = \cos\left(\frac{2\pi \times 60}{500}\right) = 0.7289686 \quad (8.21)$$

这样的乘法需要用到浮点小数算法, 这肯定会降低算法的速度。为了逼近算法, 可以选择一个 2 的幂的和, 由二进制算法来实现, 在一些硬件环境中, 这种算法比浮点小数乘法快。在本例中

$$N = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{128} + \frac{1}{512} + \frac{1}{2048} = 0.72900 \quad (8.22)$$

8.3 自适应滤波器的其他应用

自适应滤波器不仅应用在消除 60Hz 干扰上, 还应用在信号提取和消除人为干扰方面。自适应滤波方法对于从另外一个信号中产生出所需信号也是有益的, 此信号与原始信号无联系。

8.3.1 胎儿 ECG 中的母体 ECG

监测器已使测量未出生婴儿的心跳成为可能。然而, 人工的运动和母体的 ECG 会使观察胎儿的 ECG 非常困难, 因为它的信号幅度太小了。自适应滤波器已被用于消除母体的 ECG。Zhou 等人(1985)提供了一种算法, 用窗口激光质谱仪程序去检测微弱的心跳。腹部的跳动作为主输入, 母亲胸部的跳动作作为参考噪声输入。从包括胎儿和母体的 ECG 的腹部 ECG 中减去母体的 ECG, 得到的剩余信号就是胎儿的 ECG。

8.3.2 心脏产生的人为现象

在电阻抗呼吸记录领域中, 已经用自适应滤波器解决了心脏产生的人为现象问题(ECG)。血和心脏容器的变化将引起电阻抗变化, 于是就产生了人为现象。这会导致错误地分析呼吸运动。当监测婴儿是否窒息时, 会导致错误的报警。Sahakian 和 Kuo(1985)提出用可调谐激光质谱仪法去消除心脏产生的人为组成部分, 这样可建立最好的呼吸记录。为模拟心脏产生的人为现象, 他们制造了一个 ECG 模型, 这个模型与包括心律不齐的 ECG 中的固有干扰波同步。心脏产生的所有人为现象虽然同步, 却比心脏的收缩延迟, 所以 ECG 模型可被用于分离

并消除 ECG。

8.3.3 心室纤颤和心动过速的检测

频域技术已广泛应用于心室纤颤的检测,但它造价高而且不能总是实时实现。Hamilton 和 Touphins(1987)提供了一种自适应滤波的独特算法,设置与其频率谐振峰值一致的极点。通过二阶 IIR 滤波器,由系数得到的极点将给出对第一个频率峰值的很好的估计。

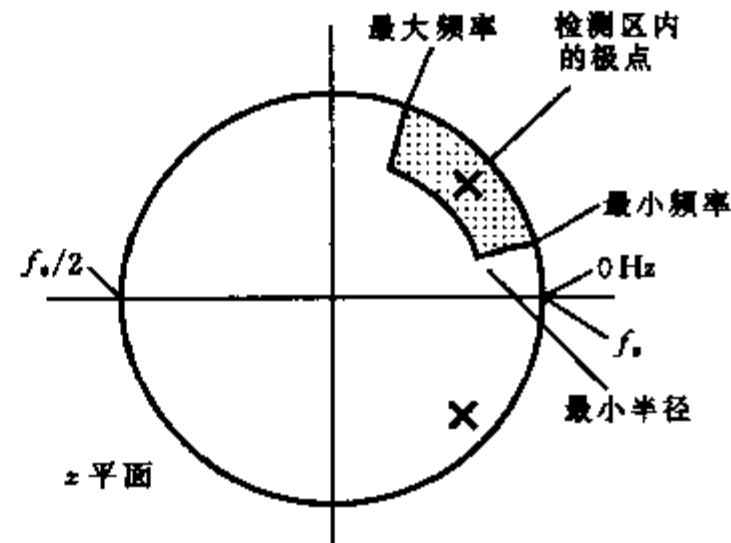
这种滤波器的传递函数为:

$$H(z) = \frac{1}{1 - b_1 z^{-1} - b_2 z^{-2}}$$

式中,不是所有的极点都在实轴上,但可通过下式求出极点的半径和角度:

$$b_1 = 2\gamma \cos\theta, \quad b_2 = -\gamma^2$$

从纤颤在 3~7Hz 频带产生的主要峰值,可决定它们是否处在 z 平面上的“探测区”。LMS 算法可不断校正滤波器的系数。图 8.4 所示的为自适应滤波器的 z 平面的零极点图。阴影区域表示 ECG 频谱的主要峰值在危险区域。这种方法唯一的缺点是,它会由于心室早搏(室早)过频心脏纤颤和严重的运动伪迹而产生心率大于 100 次/min 的误判。



8.4 实验: 60Hz 自适应滤波器

安装 UW DigiScope 程序,选择 ad(V)ops,然后选(A)daptive。这种模型是本书所描述的消除 60Hz 的自适应滤波器的方法。可通过滤波器的步长 d 来进行控制。在滤波器已经消除 60Hz 噪声而适应干扰后,可观察到滤波器已没有 60Hz 的组成部分。这种程序总使用包含有 60Hz 的数据文件 adpting.dat。

图 8.4 z 平面显示二阶自适应滤波器的复合共轭极点

复习思考题

- 8.1 自适应滤波器主要优点在什么地方?
- 8.2 叙述一下构造 Wiener 滤波器的准则。
- 8.3 为什么 Wiener 滤波器的输出一般有残留误差?
- 8.4 用最速下降法设计自适应滤波器。
- 8.5 用最小均方法(LMS)设计自适应滤波器。
- 8.6 为什么最速下降法和最小均方法在步长上必定有界?
- 8.7 在 60Hz 正弦波法中用不同的步长有何代价和受益?
- 8.8 解释一下 60Hz 正弦波法怎样适应噪声的相位。
- 8.9 60Hz 自适应滤波计算函数为 $f(nT+T) = [x(nT+T) - e(nT+T)] - [x(nT) -$

$e(nT)$],如果函数小于0,算法怎样调整预估值 $e(nT+T)$?

8.10 60Hz 自适应滤波器用等式 $e(nT+T)=2Ne(nT)-e(nT-T)$ 估计噪声时,若预估值太大,对 $e(nT-T), e(nT+T)$ 应做何调整? 写出 N 的等式并解释等式各项的含义。

8.11 60Hz 自适应滤波器计算函数为 $f(nT+T)=[x(nT+T)-e(nT+T)]-[x(nT)-e(nT)]$,根据函数大于、小于、等于0,调整预估值 $e(nT+T)$ 。用图表法表示并解释为什么函数不能简化为: $f(nT+T)=x(nT+T)-e(nT+T)$?

第九章 信号平均

前几章讨论的线性数字滤波器在信号与干扰的频谱重叠不十分严重时,会工作得很好。例如,截止频率为100Hz的低通滤波器一般可在衰减频率超过100Hz的ECG信号中很好工作。然而,如果高阶噪声的频域为50~100Hz,企图用50Hz低通滤波器去消除它们,则会同时削弱ECG信号中的一些组成部分。掺杂在信号频带中的大幅度噪声可以严重地污染信号。因此,常规的滤波方案在信号和噪声频率重叠十分严重的情况下会失效。信号平均是用于不使信号失真地分离出重叠噪声的数字技术。本章介绍信号平均技术如何用于提高信噪比,并讨论若干应用实例。

9.1 信号平均的基础

图9.1(a)所示的是被噪声掺杂的信号频谱,在此例中,噪声带宽与信号带宽被完全分离,于是噪声能轻易地用线性数字滤波器加以消除。而在图9.1(b)中,噪声带宽与信号带宽互相重叠,且噪声的幅值比信号幅值大。在此情况下,采用低通滤波器消除这些噪声时,会失掉一些信号,因此会使信号失真。

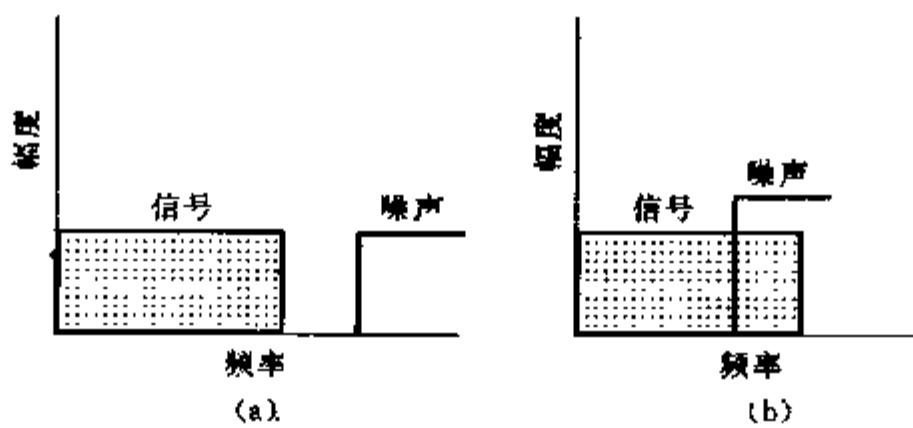


图9.1 信号和噪声的频谱图

(a)信号和噪声频带不重叠,故普通的低通滤波器可用来保留信号和消除噪声;(b)由于信号和噪声的频谱重叠,普通滤波不能被用于在不消除某些信号源的情况下消除噪声频率,信号平均在此例中是有用的

信号平均一个最重要的应用领域是绘制脑电图。从头皮电极处记录的脑电图是很困难的,因为它是数以百万计的脑细胞的活动的总和。从脑电图中推断出部分脑的听觉或视觉的活动是不可能的。然而,如果用一闪一闪的灯光或从听觉上用喀喳声刺激部分大脑,将会在脑区内产生诱发响应,传递被刺激的感觉系统的信息。对由许多刺激诱发的信号求和,再除以总的刺激个数,可得到一个平均响应。这个信号揭示了大脑感觉系统的活动。

信号平均将重叠的随机信号按时间位置进行求和。如果信号的时间位置十分一致,则信号波将直接组合在一起,另一方面,无关联的噪声则被及时平均。因此,信噪比(SNR)将得到提高。

信号平均方法是基于下列信号和噪声的特性而提出的:

(1)信号波形必须是重复性的(尽管它无需是周期性的)。这意味着信号肯定会发生一次以上,但不必以固定的时间间隔重复。

(2)噪声为随机且与信号不相关的。在此应用中,随机意味着噪声不是周期性的,它只能用统计方法描述(例如,通过它的均值和方差)。

(3)各信号波的时间位置必须是能被准确掌握的。

正是噪声的随机性使信号平均有效。每一个信号的时间位置(或扫描)有意对准前一个时间位置,这样从新的时间位置中的数字取样将被加入到前一时间位置的相应的取样上。因此各时间位置的时间校正重复信号 S 被直接相加起来,在 4 个脉冲过后,信号幅值是各时间位置的 4 倍($4S$)。如果噪声是随机的,平均值为 0,均方根值是 N ,4 个脉冲后的均方根值是平方值总和的平方根(即 $(4N^2)^{1/2}$ 或 $2N$)。一般地, m 个重复后的信号幅值是 $(m)^{1/2}N$ 。因此,信噪比将提高 $m^{1/2}$ 倍(即 $m^{1/2}$)。例如,一个信号 100 次重复的平均值可把信噪比提高为以前的 10 倍。这可用下面的数字推导来证明。

输入波形由信号 $S(t)$ 和噪声 $N(t)$ 组成,即

$$f(t) = S(t) + N(t) \quad (9.1)$$

令 $f(t)$ 每 7s 采样一次,那么,在采样周期内,任何采样点的值是噪声和信号的和,即

$$f(iT) = S(iT) + N(iT) \quad (9.2)$$

每一采样值都将被存储器存储起来,在第 i 个存储位置, m 个重复后存储起来的值为:

$$\sum_{k=1}^m f(iT) = \sum_{k=1}^m S(iT) + \sum_{k=1}^m N(iT) \quad (i = 1, 2, 3, \dots, n) \quad (9.3)$$

在采样点 i 时刻的信号组成部分,各周期相同。如果信号稳定,扫描十分均衡,则

$$\sum_{k=1}^m S(iT) = m S(iT) \quad (9.4)$$

以上假定与信号和噪声没有联系,并且噪声是随机的,平均值为 0,在许多周期后, $N(iT)$ 的均方差为 σ_n ,即

$$\sum_{k=1}^m N(iT) = \sqrt{m\sigma_n^2} = \sqrt{m}\sigma_n \quad (9.5)$$

由式(9.4)和式(9.5)中可给出 m 个周期后的信噪比为:

$$\text{SNR}_m = \frac{m S(iT)}{\sqrt{m}\sigma_n} = \sqrt{m} \text{SNR} \quad (9.6)$$

因此,信号平均将信噪比提高了 \sqrt{m} 倍。图 9.2 所示的是由式(9.6)得到的图形。

图 9.3 所示的是信号平均的问题,图中最上层曲线表示中间的心电图叠加了随机噪声后的结果。因为噪声是宽频带的,不同时滤掉 ECG 的一些噪声频率组成部分,是无法用传统的线性滤波器完全滤掉的,这将会使 ECG 失真。这种噪声信号的平均方法,需要将每个 QRS 复波在时间上对准。通过分析对波形的多次滤波形式,可以定位 QRS 复波的峰值,并将它们进行时间上的对准,图 9.3 中最下面的曲线表明这些时间基准标记(基准点),这些基准点被用于信号处理过程。

图 9.4 所示的是 QRS 复波如何集中于基准点,怎样集中起来并被叠加产生平均信号的。经过时间校正的 QRS 复波可直接叠加起来,同时噪声平均为 0,而且只要它们与信号在时间上有正确的对准,基准点定位可以在信号平均之前或之后进行。

心电图学的一个研究领域是晚电位,需要带宽 500Hz 的心电图放大器。在临床中,这些弱

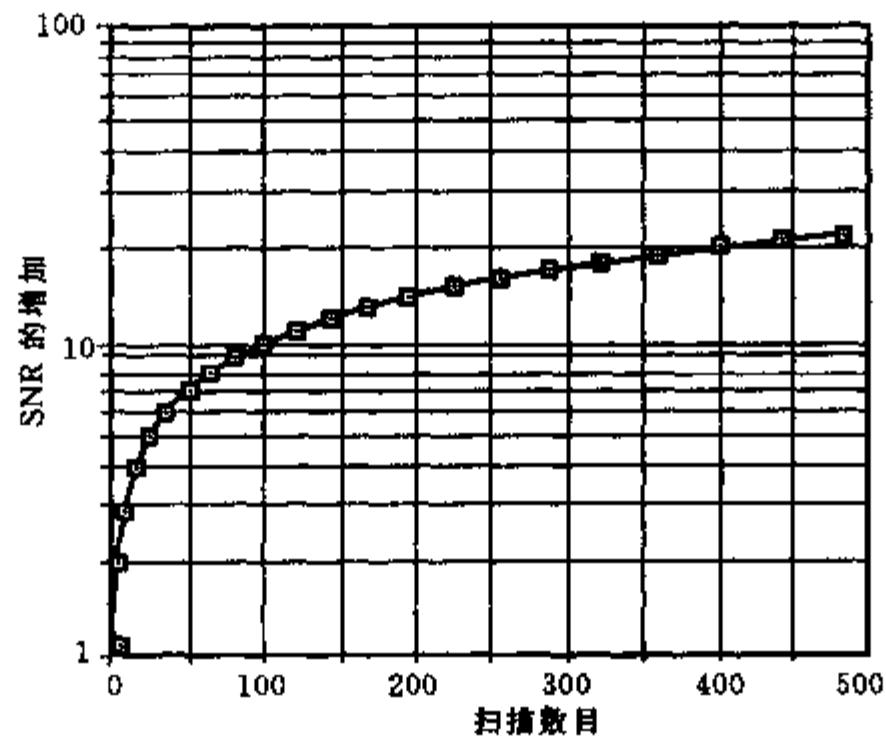


图 9.2 信噪比的增长为平均扫描个数的一个函数

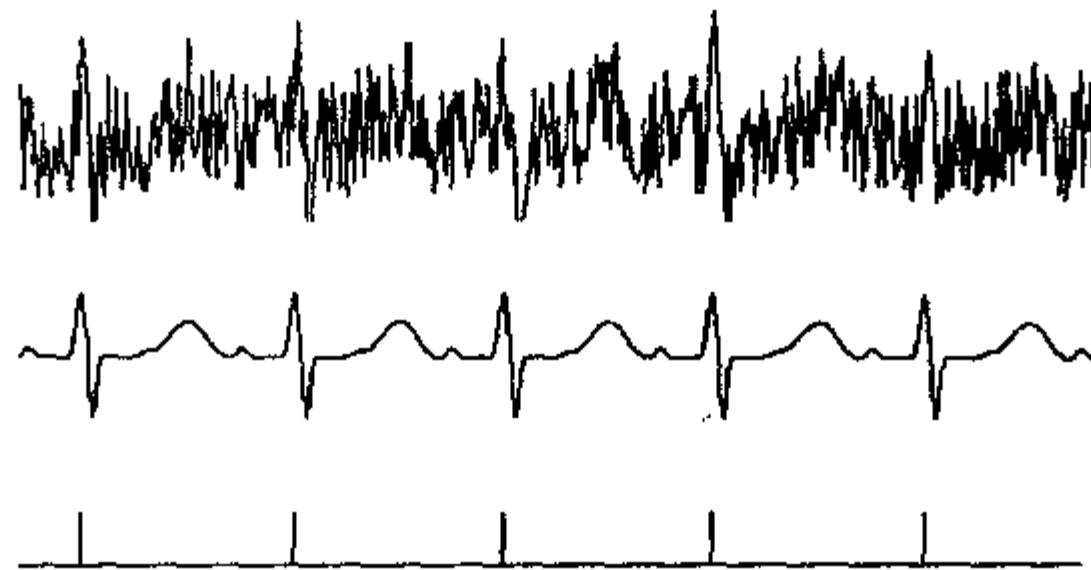


图 9.3 信号平均及波峰位置



图 9.4 信号求和

的高频信号可能会发生在反常的体表 ECG 的 QRS 复波中。这些信号与其他的相比是如此的弱，以至于叠加在噪声中不易被发现，也不会被平均。在此应用中，基准点可从 QRS 复波中得到，且平均频带是在时间上跟随各个 QRS 复波的频带上。

最高轨迹是中间轨迹掺杂了任意噪声的心电图。最低轨迹提供了基准标记，显示信号中复合波样波的位置。

图 9.4(a)、(b) 和 (c) 所示的为掺杂了随机噪声的经过时间定位的信号求和情况（它们从图 9.3 中被提取），以提高信噪比。100 个 ECG 的时间平均可将信噪比提高 10 倍，结果如图 9.

4(d)所示。

9.2 信号平均器

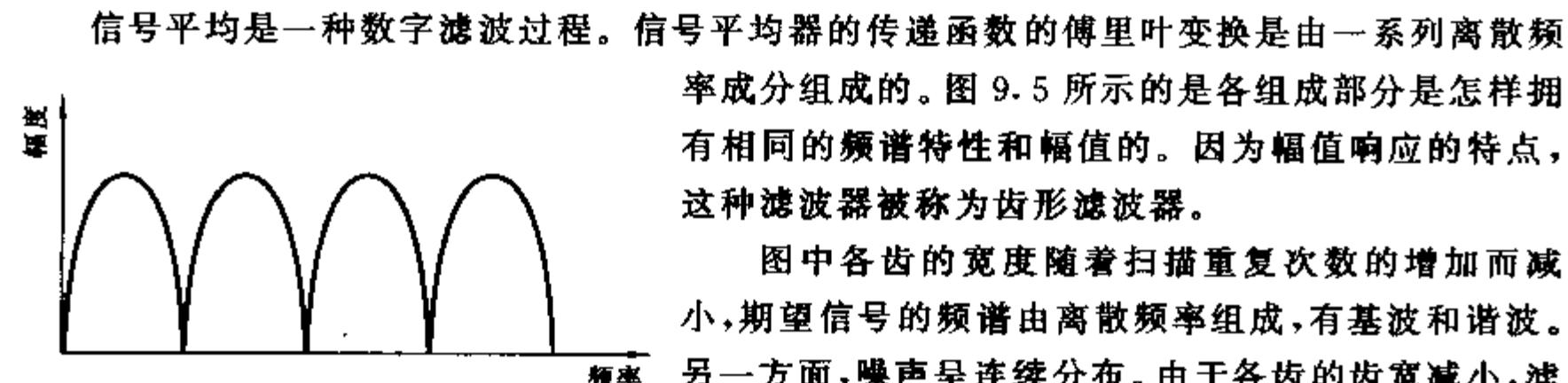


图 9.5 信号平均器的傅里叶变换
随着扫描数目的增长，各齿的宽度减小

图中各齿的宽度随着扫描重复次数的增加而减小，期望信号的频谱由离散频率组成，有基波和谐波。另一方面，噪声呈连续分布。由于各齿的齿宽减小，滤波器更多地选择一些信号的基波和谐波，同时阻止任何噪声频率落于齿间。因此，信号平均器可以通过信号而阻止噪声。

9.3 典型的平均器

图 9.6 所示的是典型平均器的方框图，为平均像这样一个对听觉刺激有皮层反应的信号，可刺激带有喀喳声的激励源系统，同时，提供一个从激励中得出的触发器，它触发对存储在缓冲寄存器中的前次响应采样数据求和（在此例中，采样数据为由刺激引起的脑电图波形），前次响应是指时间位置或曲线。当平均器接收到触发脉冲后，便以选定的速度采样脑电图波形，使信号数字化，并对应于那个采样间隔（在缓冲寄存器中）的存储单元的内容求和。这个过程是持续的，通过存储地址变动直到所有的地址都已被采样为止。扫描在此刻结束。一个新的扫描将在下一个触发脉冲时刻开始，这个过程重复着，直到期望的扫描曲线已被平均为止。平均过程的结果被存储在缓冲寄存器中，并可以显示在 CRT 上，作为平均响应。

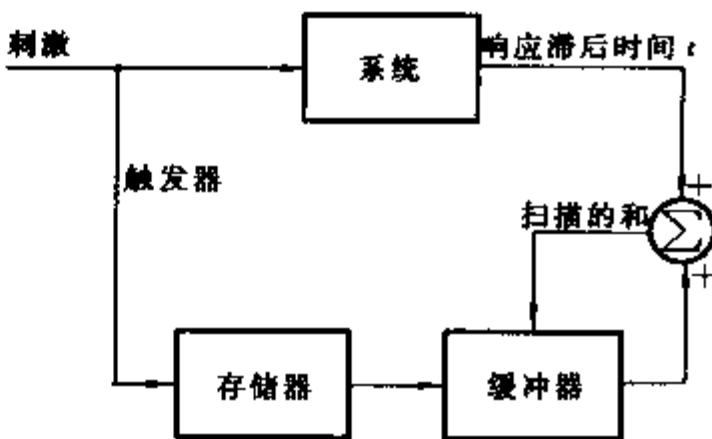


图 9.6 典型信号平均器的方框图

9.4 信号平均软件

图 9.7 所示的是用于信号平均的程序流程图，用于平均图 9.3 所示的 ECG 信号。程序用 QRS 复波探测算法在每个 QRS 复波的峰值得到基准点，一旦 QRS 探测到 128 个新采样点——包括在基准点前 64 个点和后 64 个点就被加入缓冲器中。

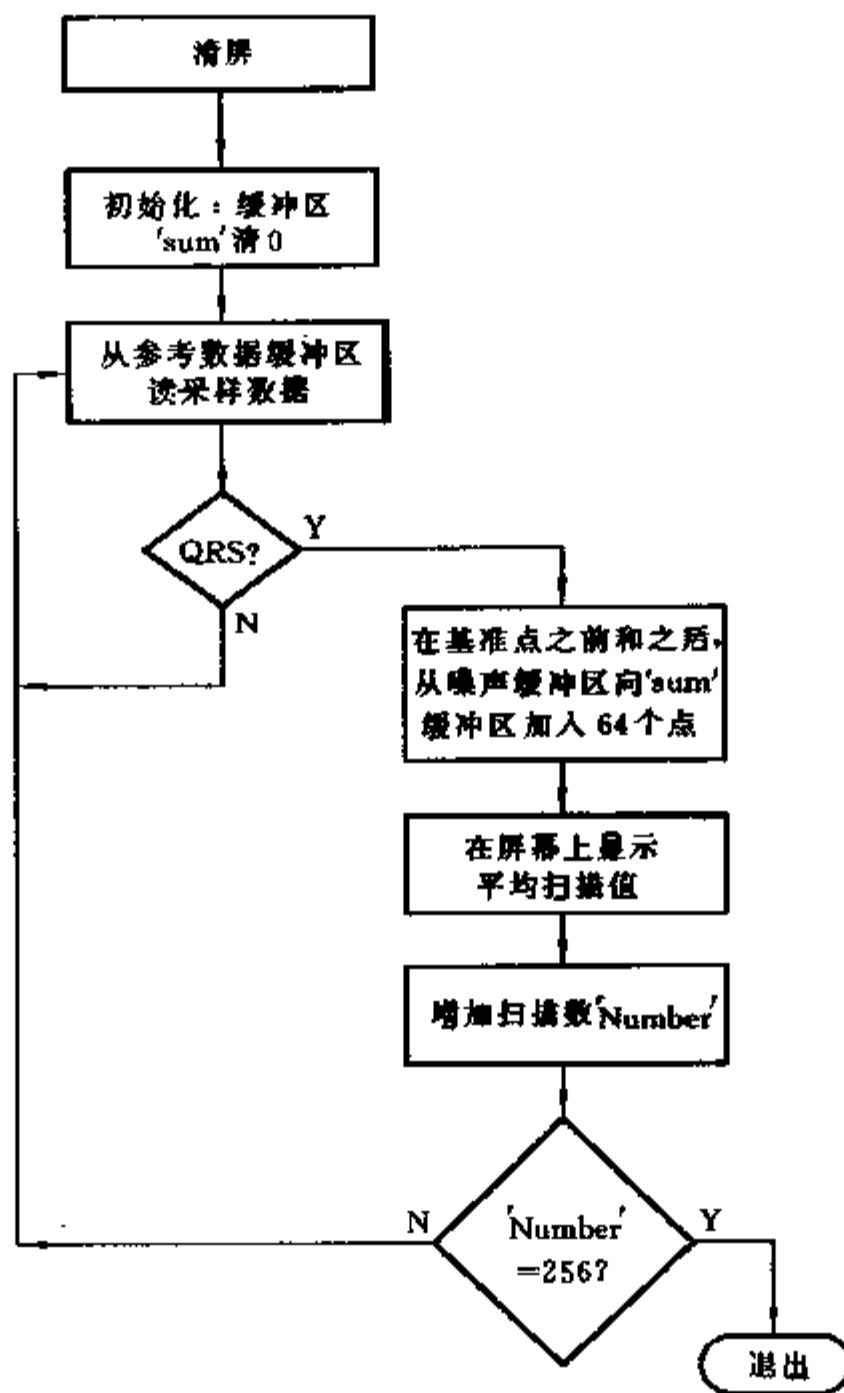


图 9.7 信号平均程序流程图

9.5 信号平均的局限性

在信号平均中所做的重要假设是噪声为高斯分布的。这个假设对生物医学上的信号并不十分有效。而且,如果噪声分布同信号相关便可能产生错误的结果。如果基准点从信号本身得来,则必须小心处理以保证噪声不会影响到基准点的瞬时位置,否则,每个信号波形的定位差便会在最后的结果中引起低通滤波效应。

9.6 实验:ECG 信号平均

装入 UW DigiScope 程序,选 ad(V)Ops,再选 a(V)verage。这是信号平均的演示模块。对于施加的任何随机的、高斯分布的噪声,源文件总是 average.dat。有一个简洁的数据版本用于作为触发信号。显示在输出通道上的数据平均版本在叠加连续波形时就逐渐建立起来了。屏上显示的是累加信号的轨迹之和,而不是平均信号,这样就需要将结果信号按尺度降低幅值才

能看到与原始信号有相同幅值尺度因子的真实平均信号。输出通道的尺度因子可以在获取信号轨迹时进行控制。每按一次向下的方向键会使幅值减半。例如，在 16 次心跳曲线叠加的情况下，就需要按 4 次向下的方向键使输出幅值除以 16，以得到合适的幅值尺度。

复习思考题

- 9.1 在什么噪声条件下信号平均不能改善 SNR？
- 9.2 在信号平均应用中，不相关的噪声幅值是原始信号幅值的 16 倍，需要多少次平均才能达到 4：1 的信噪比？
- 9.3 假设经过 4096 次 EEG 诱发电位的平均后达到 4：1 的信噪比，若 EEG 与噪声源不相关，那么信号平均之前的信噪比是多少？
- 9.4 在信号平均应用中，噪声幅度最初是信号幅值的 4 倍，为了达到 4：1 的信噪比，需要进行多少次平均的扫描？
- 9.5 在信号平均应用中，刺激产生的信号与噪声强弱相关，信号与噪声的频幅相互重叠，进行 100 次平均后能将信噪比改进到什么程度？

第十章 数据压缩技术

在典型的计算机医学信号处理系统中需要使用大量数据,它们是难以存储和传输的。这样,需要一种减小数据的存储空间,同时能够在信号重建时保存重要的临床信息的方法。在某些应用中,压缩和重建过程需要具备实时性。

数据压缩算法通过减少原始信号中的冗余度来减少所需存储代码位的数量。将原始信号的比特数除以存储压缩信号的比特数就得到压缩比。通常希望压缩比越高越好。但是,要注意不能将压缩比作为数据压缩算法的唯一标准。带宽、采样频率和原始数据精度等诸多因素会明显地影响压缩比。

一种好的数据压缩算法必须在容许的范围内具有好的保真度。在生物医学数据压缩中,一般可用目测的方法来决定重建信号在临床上的合理性,也可以用测量残差,即重建信号和原始信号的差的方法来确定其合理性。这样的一种数值量度,即均方根差值的百分比 PRD,由下式给出:

$$PRD = \left\{ \frac{\sum_{i=1}^n [x_{org}(i) - x_{rec}(i)]^2}{\sum_{i=1}^n [x_{org}(i)]^2} \right\}^{\frac{1}{2}} \times 100\% \quad (10.1)$$

式中,n 为采样次数; x_{org} 和 x_{rec} 分别为原始数据和重建数据。

无失真数据压缩算法没有残差,重建信号时能准确重现原始信号。然而,没有残差并不能保证是临床可接受的,较大的残差也并不能一概予以否定。例如,用于心电图记录的数据压缩算法可以删除小范围的基线漂移,残差只包含可忽略不计的临床信息。因此重建心电图信号只要能为临床所接受,不管其残差有多大都不影响质量。

本章将讨论用于 ECG 数据压缩的两种类型的算法。第一类算法是特征点提取法,它包括转折点(TP)算法,AZTEC(幅值零相位编码)算法和扇形算法。这些方法一般只保留具有重要信息的数据值,而除去其余的。由于产生了非零的残差,所以称为有失真算法。第二类算法是基于 Huffman 编码字数的算法,该类方法根据数据出现的频率进行可变码长的编码。预测算法一般与 Huffman 编码一起使用,它通过顺序检查邻近的采样信号数来进一步减小数据冗余度。

10.1 转折点算法

采用转折点算法的基本目的是使心电图信号的采样频率由 200 次/s 减少到 100 次/s。该算法是依据以下的观察提出来的:除了大振幅与陡峭的 QRS 复波外,对于 ECG 来说,100 次/s 的采样速率是足够的。

转折点算法基于以下认识:心电图信号一般被过分采样,其采样频率比其最高频率成分还要快 4 或 5 倍。例如,用于监护 ECG 的带宽是 50Hz,但往往采用高达 200 次/s 的采样频率,以

便容易地看到 QRS 复波的高频特性。由采样定理可知, 可用 100 次/s 的采样频率对这样的信号进行采样。转折点算法通过有选择地保留一些特征点(如波峰、波谷和转折点等), 提供了将有效采样率减至一半, 即减至 100 次/s 的一种途径。

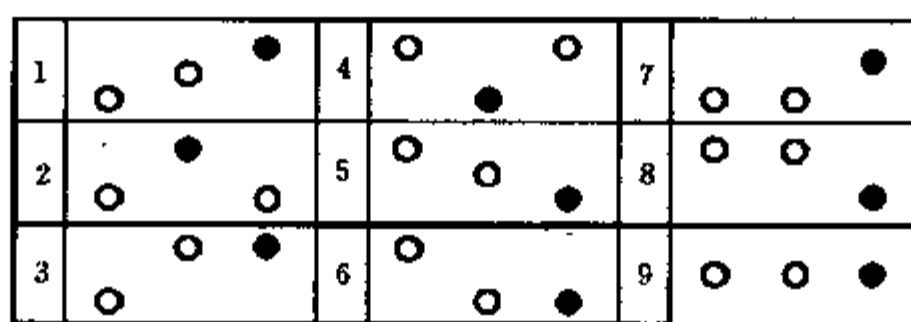
该算法同时处理三个数据点。它存储第一个采样点并将它作为参考点 x_0 。对于紧接着的两个点 x_1 和 x_2 , 该算法根据哪一点代表原始信号的转折点(通过判别斜率的改变)来选择保留 x_1 还是保留 x_2 。

图 10.1(a)所示的是 3 个连续采样点所有可能的排列形式。在每个框图中, 黑点代表 3 个原始点的斜率, 该算法保留这个点使之成为下次重复的参考点 x_0 。然后采样余下的两个点, 将它们作为 x_1 和 x_2 , 并重复此过程。

可用一简单的数学判据来定义存储点。首先考虑 $\text{sign}(x)$ 函数:

$$\text{sign}(x) = \begin{cases} 0 & (x = 0) \\ +1 & (x > 0) \\ -1 & (x < 0) \end{cases} \quad (10.2)$$

可得到 $s_1 = \text{sign}(x_1 - x_0)$ 和 $s_2 = \text{sign}(x_2 - x_1)$, 其中 $(x_1 - x_0)$ 和 $(x_2 - x_1)$ 分别是两相应的连续点连线的斜率。如果斜率为 0, 则函数值为 0, 正或负的斜率分别用 +1 或 -1 表示。转折点仅在斜率由正变负时产生, 反之亦然。



(a)

模型	$s_1 = \text{sign}(x_1 - x_0)$	$s_2 = \text{sign}(x_2 - x_1)$	$\text{NOT}(s_1) \text{OR}(s_1 + s_2)$	保留样本
1	+1	+1	1	x_2
2	+1	-1	0	x_1
3	+1	0	1	x_2
4	-1	+1	0	x_1
5	-1	-1	1	x_2
6	-1	0	1	x_2
7	0	+1	1	x_2
8	0	-1	1	x_2
9	0	0	1	x_2

(b)

图 10.1 转折点算法

(a) 所有可能的三点分布图, 每个框内包括三点 x_0, x_1 和 x_2 的序列, 实点被存储;

(b) 用于找到存储点的判据

C 语言中, 用逻辑运算符 NOT 或 OR 来判断转折点的, 如果 $C=0, \text{NOT}(C)=1$, 否则 $\text{NOT}(C)=0$ 。同样逻辑 OR 仅当 a 和 b 都为 0 时, $(a \text{OR} b)=0$, 因此保存 x_1 , 仅当 $\{\text{NOT}(s_1) \text{OR}(s_1 + s_2)\}$ 为零, 否则保留 x_2 。在此表达式中, $(s_1 + s_2)$ 是由 $\text{sign}()$ 函数得到结果的算术和。这个过程的最后结果是得出一个用以决定保留 x_1 还是保留 x_2 的逻辑值。点 x_1 仅当斜率由

正变负时才被保留,反之亦然。这个结果用算术运算也可以容易地得到,但是逻辑运算更快。

```
#define sign(x) ((x)? ((x>0)? 1:-1):0)

short *org,*tp;           /* original and tp data */
short x0,x1,x2;          /* data points */
short s1,s2;              /* signs */

x0=*tp++=*org++;        /* save the first sample */
while(there_is_sample){
    x1=*org++;
    x2=*org++;
    s1=sign(x1-x0);
    s2=sign(x2-x1);
    *tp++=x0=(!s1||(s1+s2))? x2:x1;
```

图 10.2 实现转折点算法的 C 语言程序

图 10.2 所示的是用 C 语言实现转折点算法的流程图。图 10.3 所示的是应用转折点算法合成 ECG 信号的一个应用实例。

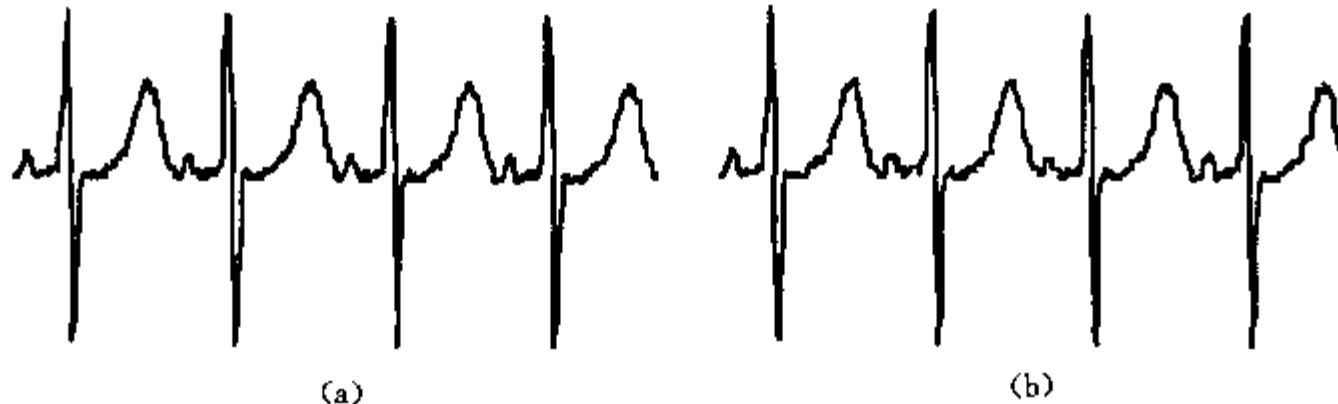


图 10.3 转折点算法应用实例

(a)由 UW DigiScope 的 Genwave 功能产生的原始波形;
(b)应用一次转折点算法之后的重建信号。压缩比为 512 : 256, PRD = 7.78%

转折点算法简单、快速、产生固定的 2 : 1 压缩比。在有选择地删除一半采样数据后,可以通过在被存数据对之间插值的方法来恢复原始数据。

将转折点算法再次应用于已压缩数据,可以将压缩比提高到 4 : 1。采用这种算法时,用 50 次/s 的有效采样率的压缩数据就可以得到 200 次/s 采样所得的信号。如果所需的心电图的带宽是 50Hz,则这种方法就违反了采样定理,因为有效采样频率比目前信号中最高频率的 2 倍要小。这样会导致重建信号时总有加宽的 QRS 复波和尖锐的边缘,因而降低了它的临床可靠性。此算法另一缺点就是,保留的点不代表平均时间间隔,这会导致短时间失真。不过,当再现信号用于标准临床控制和纸上记录观察时局部失真并不明显。

10.2 AZTEC 算法

AZTEC(幅值零相位编码)数据压缩算法将原始 ECG 数据分为水平直线和斜线段。它产生了一个线段序列,形成了心电图的逐段线性逼近。

10.2.1 数据压缩

图 10.4 和图 10.5 所示的是用 C 语言实现的 AZTEC 算法的完整流程图。算法包括两部分——线检测和线处理。

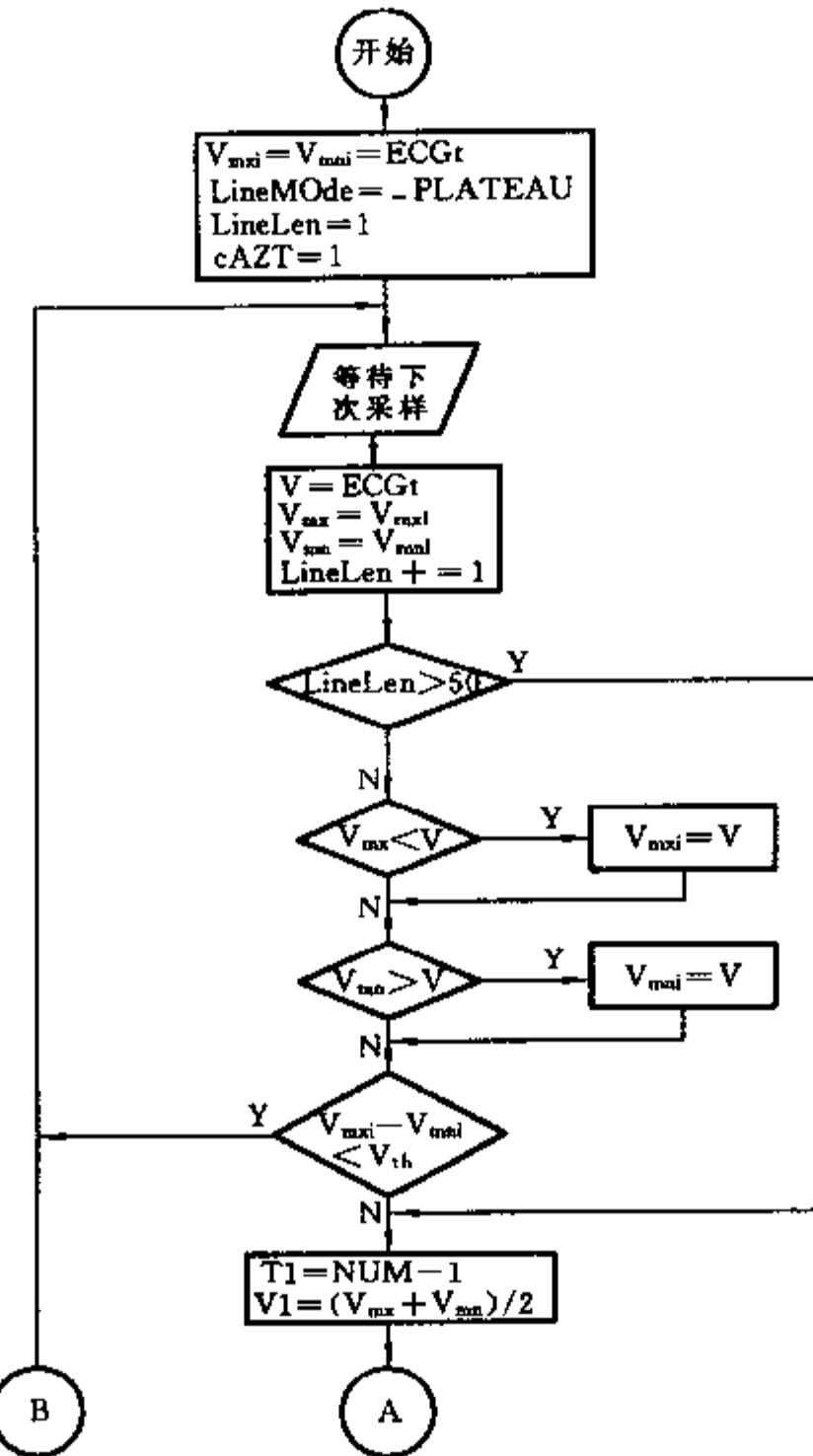


图 10.4 AZTEC 线检测工作流程图

图 10.4 所示的为线检测运算,它采用零阶插入法(ZOI)产生水平线。变量 V_{mx} 和 V_{mn} 总是反映过程线的最高值和最低值的变化。变量 LineLen 保留被测数据点的数目。如果 V_{mx} 与 V_{mn} 的差值大于预置临界值 V_{th} 或者 LineLen 大于 50, 就存储一段水平线段。被存储的值是长度 (LineLen - 1) 和水平线的平均幅值 $(V_{mx} + V_{mn})/2$ 。

图 10.5 所示的为线处理流程图,该算法根据变量 LineMode 的值决定产生水平段或斜线段。将 LineMode 值预置为 _PLATEAU 以产生开头水平段。当需要组成水平段的采样点数据不足 3 个时, AZTEC 斜线段便产生了。设置 LineMode 为 _SLOPE 意味着已进入了斜线段产生阶段。目前斜线段斜率符号或方向可以由当前幅值 V_{si} 与前一个线幅值 V_1 的差值决定。还可以重新设置斜线段长度 T_{si} 。变量 T_{si} 记录当前线幅值,这样就可以跟踪斜坡方向的任何变化。注意在线性检测开始前, V_{mxi} 和 V_{mni} 将更新为最近一次的采样值。

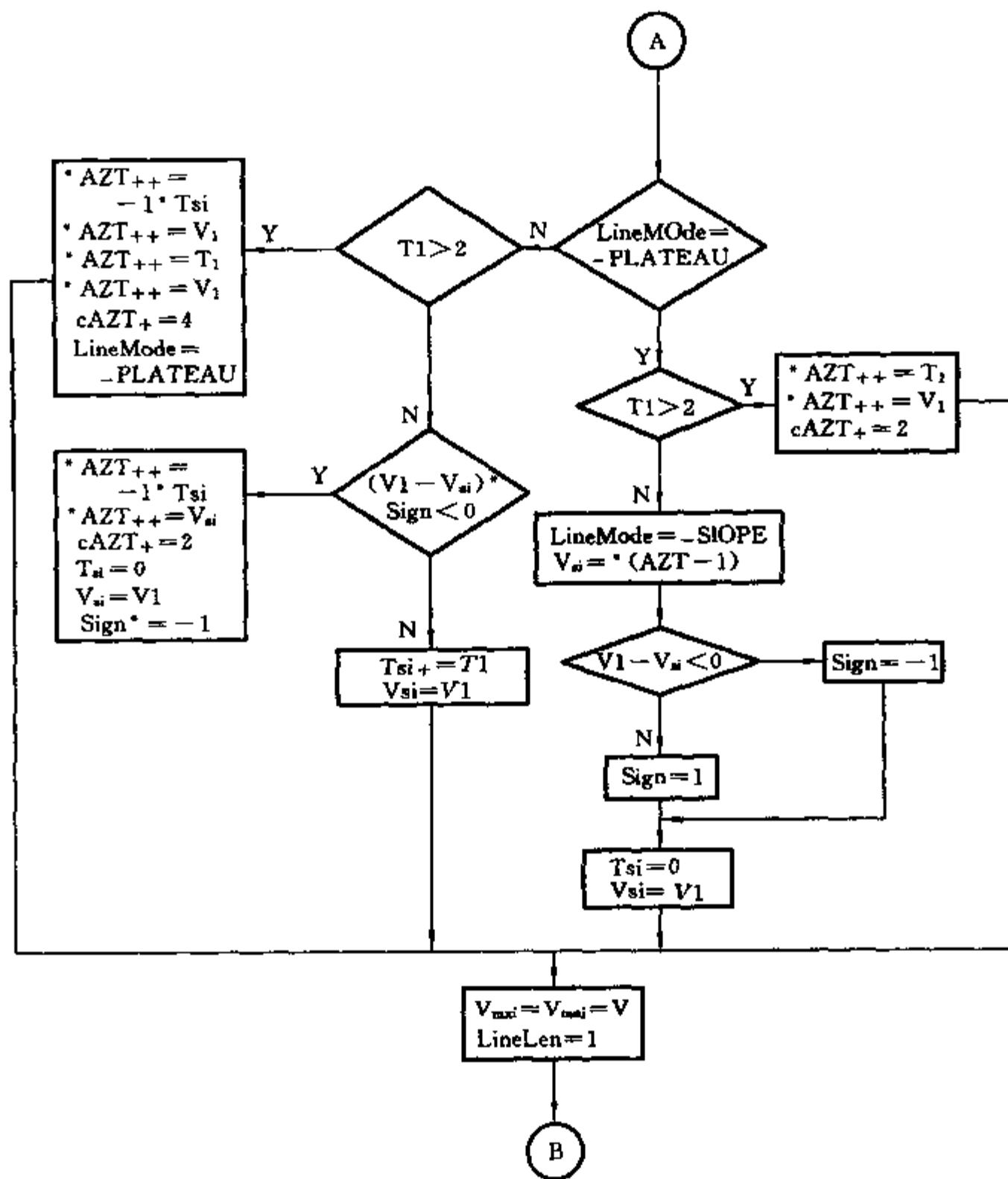


图 10.5 AZTEC 算法线处理工作流程图

LineMode 在 _SLOPE 等再次进入线处理过程时,要么存储斜线段,要么改变斜率。当能组成超过 3 个采样周期的平稳段,或者检测到方向变化时,就存储斜线段。如果检测到一个新的超过 3 个周期的平稳段,就存储当前斜线段和新的平稳段。对于斜线段而言,所存储的是它的长度 T_{si} 和它的末端幅度 V_1 。

注意 T_{si} 乘以 -1 是为了区别斜线段与水平线(即减号作为斜线段的标志)。存储新平稳段的长度和幅度后,可以重新设置所有的参数,回到产生平稳段的阶段。

当检测到斜线段方向的变化时,首先保留当前斜线段的参数,然后重新设置符号 V_{si} 、 T_{si} 、 V_{mx} 和 V_{mn} ,以便产生新的 AZTEC 斜线段。现在算法回到了线检测过程,但保留了斜线段产生模式。当没有新的平稳段或方向变化时,便要调整斜线段的参数 T_{si} 和 V_{si} ,回到线检测过程。此时 LineMode 设置为 _SLOPE。

AZTEC 不能产生恒定的数据压缩比。这个比值一般大于等于 10,这取决于信号特征点和由经验决定的检测阈值。

10.2.2 数据重建

由 AZTEC 算法得到的数据代表不断改变的线段的采样长度和幅值。例如：采样 AZTEC 数据排列是：

$$\{18, 77, 4, 101, -5, -232, -4, 141, 21, 141\}$$

将平稳段和斜线段扩展成离散点来重建 AZTEC 数据。上述数据中前两个值表示该线段有 18 个采样点、幅值为 77。其后两个值表示另一个线段的采样点为 4，幅值为 101。再后面两个值第一个为负，因为这个值代表线段长度，而线段长度必须为正，那么这个负号标志着非零斜率线段的特征点集。这个线段有 5 个采样周期长，始于上个线段的末尾（即幅值为 101 的一段）、止于幅值 -232 处。下一对点同样也表示非零斜率段，从幅值 -232 开始，有 4 个采样周期长，到幅值 141 为止。

这样的重建过程将产生阶梯状的心电图信号，这是临幊上所不能接受的。AZTEC 需要采取曲线平滑算法或低通滤波器进行后处理，以便除去跳动并产生更能被接收的输出。

第五章介绍的最小平方多项式平滑滤波是一种简单而快捷的平滑处理信号的方法。这组滤波器将奇数 $(2L+1)$ 个输入数据点拟合成抛物线，令 $L=3$ ，得到

$$P_k = \frac{1}{21}(-2x_{k-3} + 3x_{k-2} + 6x_{k-1} + 7x_k + 6x_{k+1} + 3x_{k+2} - 2x_{k+3}) \quad (10.3)$$

式中， P_k 是新数据点； x_k 展开为 AZTEC 数据。平滑作用同低通滤波器一样可减少不连续性，尽管它能产生便于接收的输出，但也同样使幅值失真。

图 10.6 所示的为将 AZTEC 算法用于 ECG 的例子。

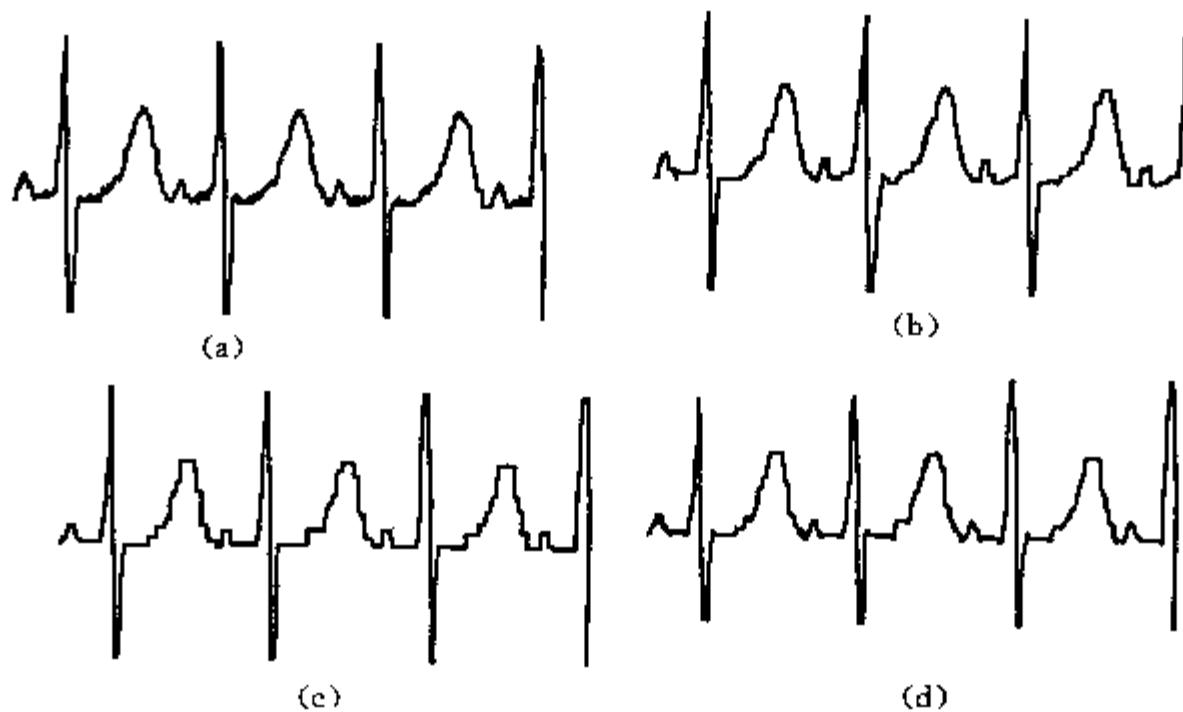


图 10.6 AZTEC 应用实例

(a) 由 UW DigiScope Genwave 产生的原始波形；(b) 小阈值，压缩比为 512 : 233, PRD = 24.4%；

(c) 大阈值，压缩比为 512 : 153, PRD = 28.1%；(d) 对(c)的平滑处理, L = 3, PRD = 26.3%

10.2.3 CORTES 算法

CORTES（坐标简化时间编码系统）算法是转折点算法和 AZTEC 算法的混合算法，它试图做到扬长避短。在高压缩比的等电位区内 CORTES 利用 AZTEC 舍弃的临幊上次要的数据，在临幊上非常重要的高频区内（QRS 复波）用转折点算法对接收的心电数据进行压缩。

CORTES 将对输入的心电图信号同时进行 AZTEC 算法和转折点算法处理。

每当一条 AZTEC 线产生时,CORTES 算法就根据线的长度决定是保留 AZTEC 数据还是保留转折点数据。如果这条线比经验决定的界限值长,就保留 AZTEC 数据点,否则保留转折点数据点。由于转折点算法用于 QRS 复波编码,故只能实现 AZTEC 平稳线。

CORTES 算法通过 AZTEC 平稳段线和在一对转折点数据点间的插值来重建信号,用抛物线来平滑 AZTEC 段以减少不连续处。

10.3 Fan 算法

Fan 算法最初用于心电图遥测,在成对的开始点和结束点间画线,这样所有的中间采样点都将在允许误差范围 ϵ 之内。图 10.7 所示的是 Fan 算法的原理。开始接受第一个采样点作为非冗余的恒定点,称为起始点。然后取第二个采样点 x_1 ,并画两个斜线段 $\{V_1, L_1\}$ 。 V_1 过点 $(x_0, x_1 + \epsilon)$, L_1 过点 $(x_0, x_1 - \epsilon)$ 。如果第三个采样点 x_2 在两个斜线段之间,那么将产生两个新的斜线段,分别通过点 $(x_0, x_2 + \epsilon)$ 和点 $(x_0, x_2 - \epsilon)$ 。比较两对斜线段,保留最收敛的(限制的)斜段(在本例中为 $\{V_1, L_2\}$)。下一步把 x_2 值赋给 x_1 ,下一个采样值赋给 x_0 。结果 x_2 总是最后一次采样值,而 x_1 是比 x_2 前一次的采样值。比较 x_2 和具有最大斜率的值,并重复以上的过程。如果 x_2 在此范围外,则保留线 T 的长度。过程重新开始,从起始采样到下一个采样的斜线组成一系列辐射线,外形像一把扇子,于是给这种算法取名为 Fan(即扇子)算法。

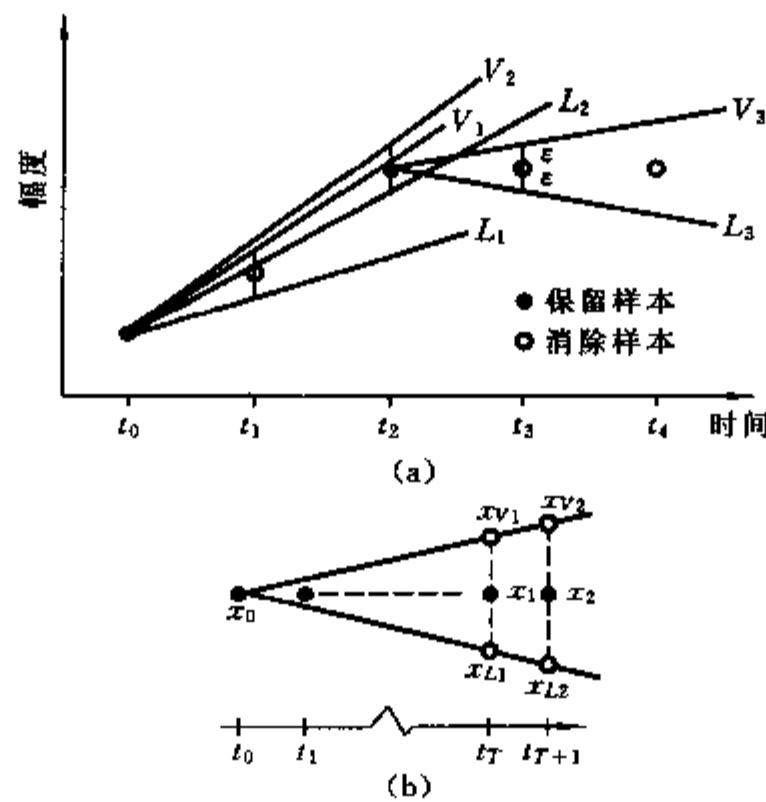


图 10.7 Fan 算法说明

下面讨论如何用 C 语言实现 Fan 算法。首先建立变量 x_{V1}, x_{L1}, x_{V2} 和 x_{L2} ,决定 x_2 的范围。由图 10.7(b),可得

$$x_{V1} = \frac{x_{V1} - x_0}{T} + x_{V1} \quad (10.4a)$$

$$x_{L2} = \frac{x_{L2} - x_0}{T} + x_{L2} \quad (10.4b)$$

式中,

$$T = t_T - t_0$$

图 10.8 所示的为用 C 语言实现 Fan 算法的程序。图 10.9 所示的是 Fan 算法在心电图中的应用。

```
short X0,X1,X2;           /* sample points */
short XV2,XL2,XV1,XL1;    /* variable to determine bounds */
short Epsilon;            /* threshold */
short *org,*fan;          /* original and Fan data */
short T;                  /* length of line */
short V2;                 /* sample point */

/* initialize all variables */
X0=*org++;
X1=*org++;
T=1;
XV1=X1+Epsilon;
XL1=X1-Epsilon;
*fan++=X0;

while (there_is_data) {
    V2=*org++;
    XV2=(XV1-X0)/T+XV1; /* upper bound of X2 */
    XL2=(XL1-X0)/T+XL1; /* lower bound of X2 */

    if(X2<=XV2&&X2>=XL2) { /* within bound */
        /* obtain the most restrictive bound */
        XV2=(XV2<X2+Epsilon)? XV2:X2+Epsilon;
        XL2=(XL2>X2-Epsilon)? XL2:X2-Epsilon;

        T++; /* increment line length */
        X1=X2; /* X1 hold sample preceding X2 */
    }
    else { /* X2 out of bound, save line */
        *fan++=T; /* save line length */
        *fan++=X1; /* save final amplitude */
    }
    /* reset all variables */
    X0=X1;
    X1=X2;
    T=1;
    XV1=X1+Epsilon;
    XL1=X1-Epsilon;
}
}
```

图 10.8 实现 Fan 算法的 C 语言程序

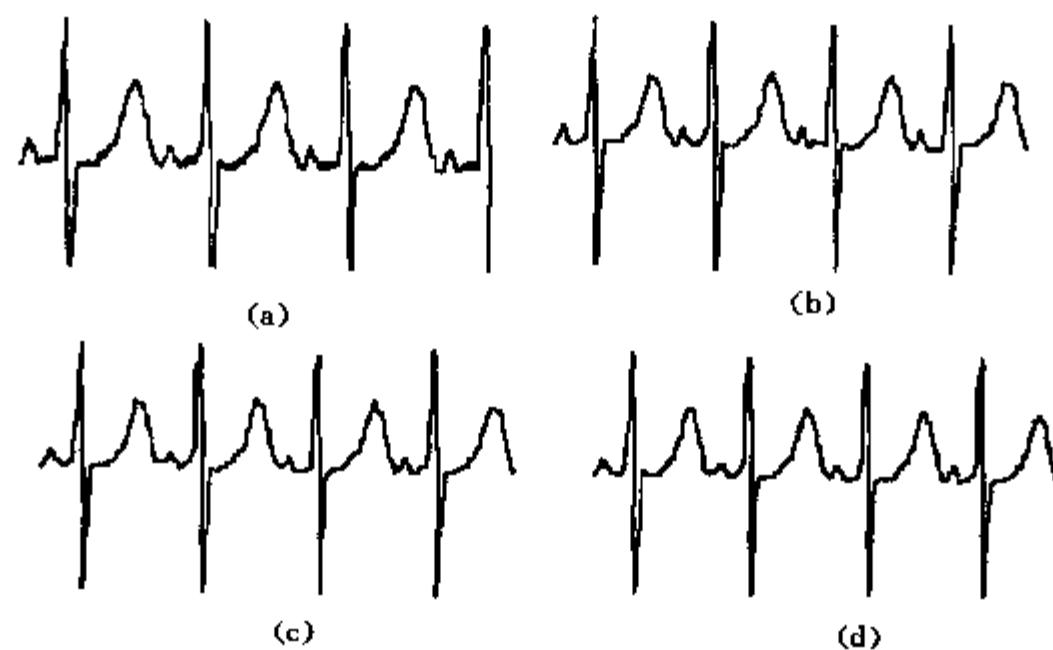


图 10.9 Fan 算法应用实例

- (a)由 UW DigiScope (Genwave) 产生的原始波形;
- (b)小误差下,压缩比 = 512 : 201, PDR = 5.6%;
- (c)大误差下,压缩比 = 512 : 155, PDR = 7.2%;
- (d)根据(c)的平滑处理, $L=3$, PDR = 8.5%

Fan 算法通过将线离散成离散点来重建压缩数据,保证两固定采样点间的线和任何在线上的实际(冗余)采样点小于等于预置容许误差。算法的压缩比由允许误差决定。与转折点算法和 AZTEC 算法相比,Fan 算法在相等的压缩比下能得到保真度较高的信号。

基于全线扫描逼近法技术(SAPA)的三种算法跟 Fan 算法很近似。SAPA—2 算法在 3 种算法中效果最好。跟 Fan 算法类似,SAPA—2 算法能保证直线重建信号与原始信号之间的误差不超过允许误差。

SAPA—2 算法不仅能算出 Fan 算法所算得的两个斜线段,而且还可算出在起始点与下一个采样点之间的称为中间斜线段的另三个斜线段。无论中间斜线段在不在两个收敛斜线段界限之间,最近的采样点均作为原始点。因此,在 SAPA—2 算法和 Fan 算法中唯一显示得不同的是 SAPA—2 算法用中间斜线段判据代替实际采样值判据。

10.4 Huffman 编码

Huffman 编码利用了这样的事实,量化信号的离散幅值不是等概率出现的。频繁出现的数据用短的码字代替。它根据数据的发生频率将可变字长的码字赋予不同的数据。

10.4.1 静态 Huffman 编码

图 10.10 所示的是 Huffman 编码的原理。举一个例子来讲,假设希望传输一个有 28 个数据点的集合: $\{1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 7\}$ 。集合包括 7 个不同的量化水平或标号。对于每个标号 S_i ,可通过将它的出现次数除以 28(整个数据点的数目),计算它的出现概率 P_i 。结果,这个集合的 Huffman 编码的结构以 7 个结点开始,每个都同 P_i 相关。每一步都将 P_i 按递减顺序分类,将结点任意地打散。最小概率的两个结点 P_i 和 P_j 合成一个新的结点,概率为 $P_i + P_j$ 。这一过程一直持续到得到一个简单值 1 或 0 为止,如图

10.10(a)所示。

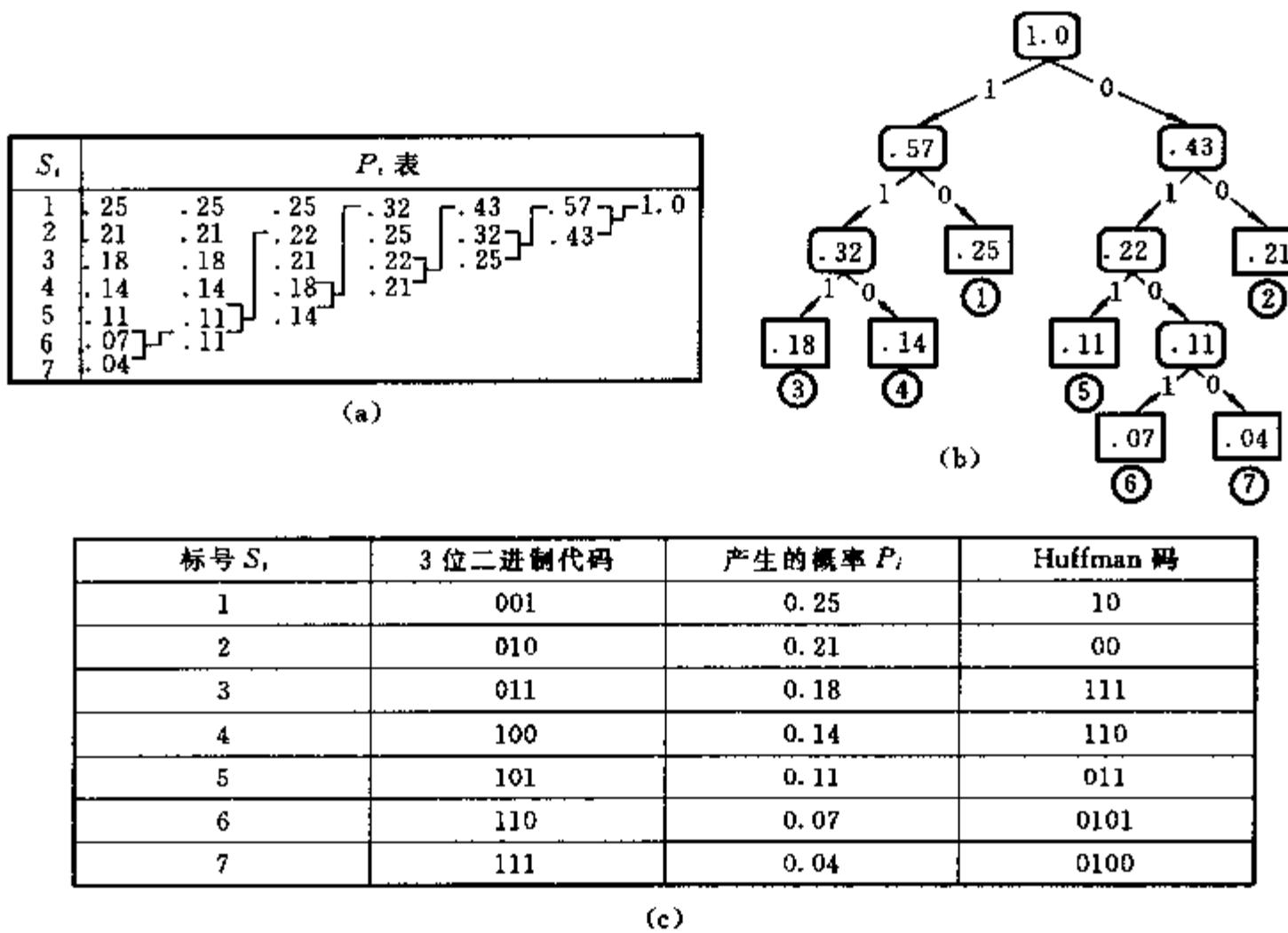


图 10.10 Huffman 编码示意图

合并结点的过程产生了如图 10.10(b)所示的二叉树。将两个结点合并, 概率为 $P_i + P_j$ 。一个父结点产生两个子结点, 代表 P_i 和 P_j 。树根的概率为 1.0。沿着树往下走, 将 1 给左边的子树, 0 给右边的子树, 得到的编码具有前缀性质(即没有编三字是其它码字的前缀)。图 10.9(c)所示的是其结果, 有 7 个标准 Huffman 编码。将这些编码送入变换表, 在压缩过程中, 根据此表将适当的码字填入输出比特流中。

Huffman 编码的压缩比取决于符号源的分布。在此例中, 原始数据需要 3B 代表 7 个离散值。在进行 Huffman 编码后, 可算出希望码字的长度:

$$E[l] = \sum_{i=1}^7 l_i P_i \quad (10.5)$$

式中, l_i 是标准的 Huffman 码的长度, 本例中为 2.65, 希望的压缩比为 3 : 2.65。

重建过程从树根开始。如果字节 1 收到, 则从左边分布算起。输出与此结点相对应的符号, 然后开始浏览右边的分支。

Huffman 编码重建过程完全覆盖了原始数据, 因此是无失真的算法。然而, 一个字节的传输错误将导致多个译码错误。传输错误的扩大是所有会产生可变长度码字算法的必然结果。

10.4.2 改进的 Huffman 编码

Huffman 编码的实现需要译码表。每个源符号对应唯一的码字。如果原始数据分成 16 个字节, 则表明需要保存 2^{16} 个记录, 这样大的表会占用不少内存, 从而会降低压缩效率。

为了减小码表的规模, 改进的 Huffman 编码将源符号分为常用的和非常用的。对于所有常用的符号, 将其 Huffman 码组成静态框。然后用特殊的码字作为非常用编码的前缀, 并将一般二进制表示编码作为后缀。

假定给出与前面相似的一组数据,假设期望离散水平要在一些将来的传输中出现,则可将离散水平{0,7}归为非常用类。再像以前一样进行 Huffman 编码得到如图 10.11 所示的结果。注意在非常用类中的离散水平码的前缀为 0100,其码长比常用类的长。因此,保证足够小的非常用类对得到合适的压缩比是很重要的。

符号, S_i	3 位二进制代码	产生的概率 P_i	Huffman 码
0	000	0.00	0100000
1	001	0.25	10
2	010	0.21	00
3	011	0.18	111
4	100	0.14	110
5	101	0.11	011
6	110	0.07	0101
7	111	0.04	0100111

图 10.11 改进的 Huffman 编码结果

量化水平{0,7}划归非常用类

在某些改进 Huffman 编码设计中,将集中于零附近的一组分散值作为常用类。在非常用类中有两个前缀码作为符号,一个前缀代表大的正值,另一个表示大的负值。

10.4.3 自适应编码

Huffman 码需要一个编码与译码的码表。要检测数据统计值,就必须检查整个或部分数据集。转换表还必须传输或存储正确的译码。

自适应编码试图建立一个表示数据的转换表。一个动态递归码表对当前统计信息很敏感。它可以根据当前统计信息更改码长以达到最大的压缩比。由于无需静态表格,所以可节省存储空间。

自适应编码的一个例子是 LZW(Lempel-Ziv-Welch)算法。LZW 算法使用固定大小的表格。它为某些选定的数据类在表格中预置一些位置。当遇到新数据时,便用未预置的位置放置新数据,这样每一个数据都有自己的位置。当表格装满后,根据新数据,LZW 算法将重新预置最早或最后使用的位置。在数据重建期间,根据译码数据,逐渐重建转换表。

10.4.4 残差差分

一般来说,相邻信号的幅值在概率意义上不是独立的。理论上讲,可将一采样值分离成两部分,一部分与以前的采样相联系,另一部分没有联系。既然采样之间的联系对应于过去采样的预测值,那么它便成为多余的和可以除去的,于是留下了没有联系的部分,它代表了预测误差或冗余信号。由于冗余信号的幅值范围比原始信号的小一些,所以可用较少字节来描述。对冗余信号采用 Huffman 码便可以压缩数据。下面简要地描述一下两种用于残差的心电图压缩算法。

Ruttimann 和 Pipberger(1979)将 Huffman 码用于预测和插入时得到的冗余信号中。在预测时,采样值是通过将合适数目的前采样点的数据求线性加权和而得到的,即

$$x'(nT) = \sum_{k=1}^p a_k x(nT - kT) \quad (10.6)$$

式中, $x(nT)$ 是原始的数据; $x'(nT)$ 是预测采样值; p 是预测时的采样次数。选取适当的 a_k 使均方误差的期望值 $E[(x - x')^2]$ 最小。当 $p = 1$ 时, 选 $a_1 = 1$, 也就是说, 取一阶信号差值。对心电图数据的初步研究表明, 高于一阶的信号差值分析比一阶的没有实质性的提高。在插入法中, 采样值的估值由过去和将来的采样值线性组合而成。预测的结果表明, 二阶估值已足够了。因此, 插入法仅用了前一个采样值和后一个采样值, 即

$$x'(n) = ax(nT - T) + bx(nT + T) \quad (10.7)$$

式中, a 和 b 取决于如何将期望均方误差减到最小。预测和插入的冗余值是用改进的 Huffman 编码法译码而得的, 其中常用类由一些集中在零附近的离散值组成。对于用插入法得到的冗余值, 在译码后可得到较高的压缩比, 大约为 7.8 : 1。

Huffman 和 Tompkins(1991)揭示了这样的事实, 即典型的心电图信号由跳动之间很少有变化的重复心跳组成。算法计算和修正了平均心跳值, 并作为结果数据。当观察心跳时, 从平均心跳中减去所测的心跳。冗余信号进行 Huffman 编码并存储在表格中表示心跳数据的位置。最后, 算法用所测的心跳去修正平均心跳值。在此方案中, 心跳位置和分离的值会严重影响压缩比。

10.4.5 扫描宽度编码

扫描宽度编码技术广泛地应用于传真。在传真比特流中的连续字节中存在有高的对比度。传真输出信号的字节是 1 还是 0 取决于对应的是黑还是白。在典型记录中, 白与黑两种颜色可提高对比度。扫描宽度编码仅仅将原始比特流转换成串 $\{V_1, L_1, V_2, L_2, \dots\}$ 。其中 V_i 为幅值, L_i 为长度。具有观察能力的读者会发现, AZTEC 和 Fan 码都是扫描宽度编码的特例。

举一个例子, 输出流 {1, 1, 1, 1, 1, 3, 3, 3, 0, 0, 0} 有 12 个元素, 扫描宽度编码 {1, 5, 3, 4, 0, 3} 只有 6 个元素。还可以用 Huffman 编码对此进行进一步数据压缩。

10.5 实验: ECG 数据压缩算法

本实验应用了本章所介绍的数据压缩技术。用附录 D 的有关方法, 安装 UW DigiScope 软件。

10.5.1 转折点算法

从 UW DigiScope 软件的 ad(V)Ops 菜单中选 c(O)mpress, 然后选(T)urn pt. 程序, 用转折点算法对显示在上方的通道的显示波形进行压缩, 然后再用解压、插入法重建信号, 并在下方的通道显示结果。从文件中读出两个不同的心电图信号, 一个正弦波和一个方波, 再运用转折点算法压缩, 注意观察:

- (1) 重建信号的质量。
- (2) 压缩比。
- (3) 均方根差值百分比(PRD)。
- (4) 原始和重建信号的功率谱。

将观测结果做成表并总结。

10.5.2 AZTEC 算法

AZTEC 算法从 UW DigiScope 软件的 COMPRESS 菜单中选择 (A)ztec 为研究对象, 重复 10.5.1 小节步骤。用至少 3 个阈值(试用全标度的峰-峰值的 1%, 5% 和 15%), 对 10.5.1 小节的表中各值进行观测和分析, 同时, 总结重建信号在用平滑滤波器前后的质量。

将观测结果做成表并总结。

10.5.3 Fan 算法

重复 10.5.2 小节介绍的 Fan 算法, 从 COMPRESS 菜单中选 (F)an。比较 Fan 算法与 AZTEC 算法的特性。你能推出什么? 将观察结果做成表并总结。

10.5.4 Huffman 编码

从 COMPRESS 菜单中选 (H)uffman, 选择 (R)un, 对显示在上方的通道信号进行 Huffman 编码, 实验中无需在此点用一阶差分法。记录压缩比。注意这个压缩比不包括表所需的空间。从 PRD 可以推导出什么? 选择 (W)rite table 将 Huffman 数据写入文件。退出 UW DigiScope 程序之后用 DOS 类命令可以看到码表。

安装一个新的心电图波, 重复以上步骤。当选择 (R)un 时用以前推导的转换表对数据进行编码。从压缩比中可以得到什么? 在用 (M)ake 从菜单中推导出新转换表后, 重新选择 (R)un, 并对新压缩比进行说明。

再选中 (M)ake, 用一次差值法推导新的 Huffman 码。使用新推导的码后压缩比有改变吗? 选中 (W)rite Table, 将 Huffman 数据写入文件, 再安装你用过的第一个心电图波形。不用推导新的 Huffman 码, 观察所得到的压缩比。评价你的观察。

从 UW DigiScope 程序退出, 观察你所得到的转换表。对转换表的上半部有何评价?

复习思考题

- 10.1 解释无失真和有失真数据压缩方法, 将本章描述的 4 个数据压缩算法分成两类。
- 10.2 已知数据 {15, 10, 6, 7, 5, 3, 4, 7, 15, 3}, 导出用于转折点算法的被存储的数据点。
- 10.3 解释为什么 AZTEC 重建信号对于心电图学不适用, 提供解决此问题的方法。
- 10.4 Fan 算法可用于其他类型的生物医学信号。对医学信号的期望特性列表时, 哪些信号用这种算法能得到满意的结果? 并举例说明之。
- 10.5 给定数据集 {a, a, a, a, b, b, b, b, c, c, c, d, d, e}, 用 Huffman 编码法求出码字。平均码字长度是多少?
- 10.6 描述修正 Huffman 编码的优点和缺点。
- 10.7 解释为什么希望将 Huffman 编码用于由原始采样点减去估计采样点得到的冗余处理?
- 10.8 数据压缩可通过使用参数提取技术实现, 即用分离和传输的信号特性代替原始信号。画出表示这样一个系统的可能形状的框图, 框图应该包括压缩和重建部分。决定这些技术

成功与否的因素是什么？

10.9 转折点算法可使：①在很长一段时间内产生极大的扫描失真；②保留每个信号中的每一个转折点（即波峰或波谷）；③如果不违反采样定理，对信号运用两次该算法可产生 4:1 的压缩比；④提供精确的重建原始信号；⑤将 AZTEC 用于脑电图（EEG）的压缩能达到同样的效果吗？解释你的答案。

10.10 以下哪些是 Huffman 编码算法的特征：①对于心电图，肯定比 AZTEC 有更大的压缩比。②不能完全重现采样数据点（在指定误差范围内）。③是可变长度码。④直接由 Morse 码结束。⑤对最常用的 A/D 值使用 ASCII 码。⑥作为自行纠正误差算法的一部分。

10.11 在转折点算法应用后，如果数据被 ADC 采样，则需要保留哪些数据结果？①{20, 40, 20, 40, 20, 40, 20, 40}，②{50, 40, 50, 20, 30, 40}，③{50, 50, 40, 30, 40, 50, 40, 30, 40, 50, 50, 40}，④{50, 25, 50, 25, 50, 25, 50, 25}。

10.12 在转折点算法用于对一个信号压缩后，被保留的数据点是{50, 70, 30, 40}。如要重建原始数据，则最佳逼近它的数据结果是什么？

10.13 ADC 采样到的 20 个数据点如图 10.12 所示。在图的顶端是采样数值，对这些采样值进行 AZTEC 编码：

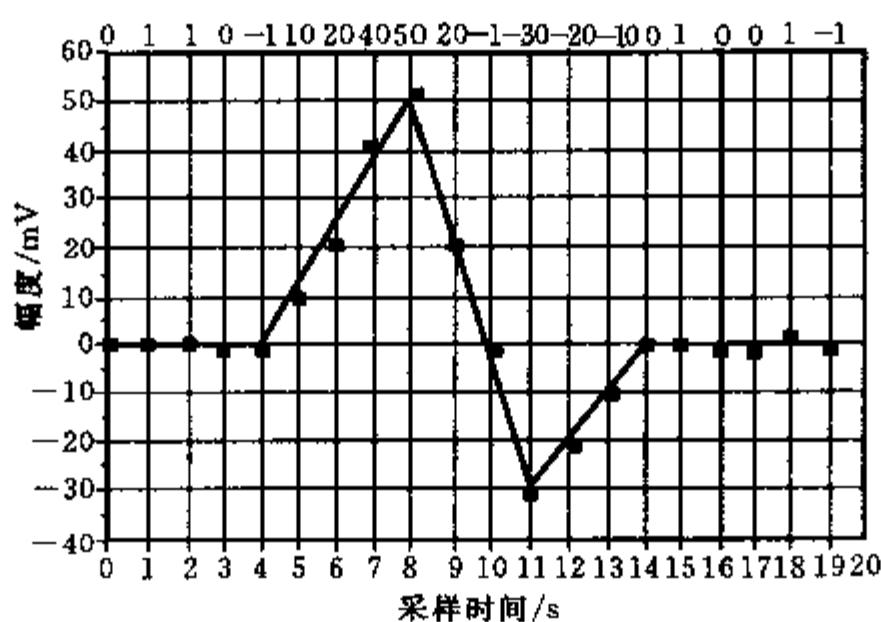


图 10.12

- ①列出代表这个信号的 AZTEC 编码的数据列。
- ②用 AZTEC 对此信号进行压缩，可实现的压缩比是多少？
- ③在下列原始数据表中哪个数据将被存储（如果用转折算法对此信号压缩）：
{0, 1, 1, 0, -1, 10, 20, 40, 50, 20, -1, -30, -20, -10, 0, 1, 0, 0, 1, -1}
- ④如果这个信号用 Huffman 类型的可变字长码表示，且以下 4 个字节模式是编码类型的一部分，指出分配给每个模式的幅值是多少。

编 码	幅 值
1	
01	
001	
0001	

- ⑤每种算法提供的压缩率是多少（假定无需编码表用于 Huffman 码的存储）？

10.14 AZTEC 将信号编码为 $\{2, 50, -4, -30, -4, 50, -4, 30, -4, 50, 2, 50\}$, 则最初被采样到的数据点有多少?

10.15 在对信号用 AZTEC 编码后得到的存储数据列为 $\{2, 0, -3, 80, -3, -30, -3, 0, 3, 0\}$ 。画出根据这些数据重建信号的波形。

10.16 AZTEC 对 8 位 ADC 得到的信号编码为 $\{2, 50, -4, 30, -6, 50, -6, 30, -4, 50, 2, 50\}$, ①数据压缩量是多少? ②根据这些数据重建信号的峰-峰值是多少?

10.17 AZTEC 对 8 位 ADC 得到的信号编码为 $\{3, 100, -5, 150, -5, 50, 5, 100, 2, 100\}$ 。将转折点算法用于对同样的原始信号压缩, 与转折点算法相比, AZTEC 能达到多少的数据压缩量。

10.18 图 10.13 所示的是由 8 位 ADC 得到的心电图采样信号, 共 20 个点。

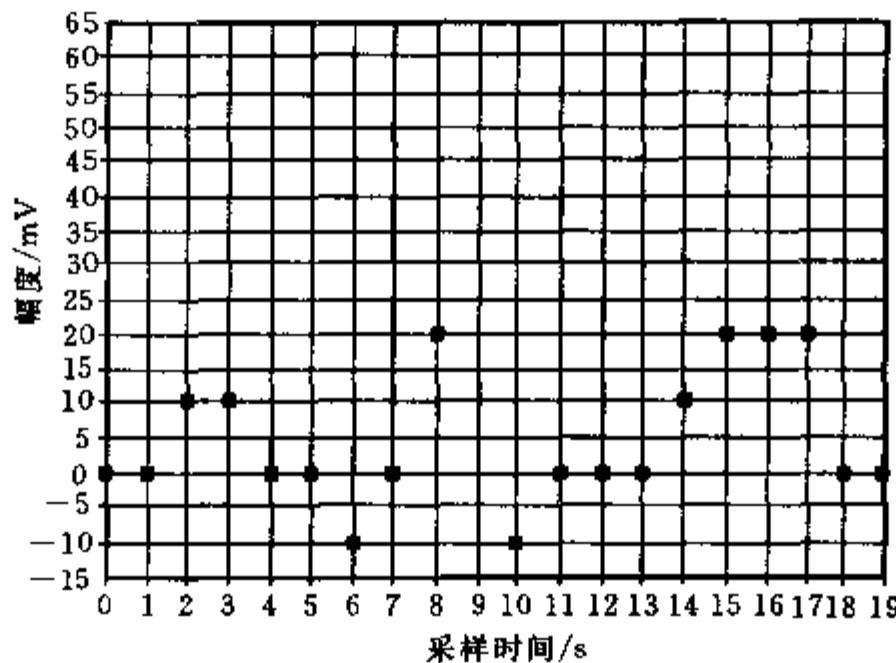


图 10.13

①画出与图 10.10 相似的包括这组数据出现概率的二元树。

②从二元树中取出 5 个点, 组成数据列, 如图 10.14 所示, 请将适当的 Huffman 码分配给这个数据列的每一个数:

数	Huffman 码
-10	
0	
10	
20	
60	

图 10.14

③假设不需要存储 Huffman 表格, 对这些数据进行 Huffman 编码能得到多大压缩比?

④请将 Huffman 编码的数据 01010110001 译码, 说明它代表的采样点。

第十一章 其他时域及频域分析方法

生物医学信息通常会受到噪声的干扰(如电源干扰,人体肌肉运动伪迹以及电外科手术仪或透热治疗仪的射频干扰)。为了滤除这些干扰信号,必须了解它们的频谱范围,以便设计出适当的滤波器。例如,如果想弄清病人的心血输出量与 QRS 复波面积的相关性,就需要采用合适的相关技术。本章介绍有关的时域及频域方法。

11.1 傅里叶变换

快速傅里叶变换(FFT)是进行信号频域分析的基本工具。

11.1.1 非周期信号的离散傅里叶变换

假设一个离散时间非周期信号是从一个模拟模板中以 T 为采样周期采集到的数据,采样角频率为 $\omega_s = 2\pi/T$,在时域中可以将其描述为 δ 函数的加权和,即

$$x(t) = \sum_{n=-\infty}^{+\infty} x(n)\delta(t - nT) \quad (11.1)$$

该表达式的傅里叶变换为:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt \quad (11.2)$$

或

$$X(\omega) = \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} x(n)\delta(t - nT)e^{-j\omega t} dt \quad (11.3a)$$

积分与求和的顺序可以交换为:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) \int_{-\infty}^{+\infty} \delta(t - nT)e^{-j\omega t} dt \quad (11.3b)$$

由此得到:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega nT} \quad (11.3c)$$

同样地,可得到傅里叶反变换:

$$x(n) = \frac{T}{2\pi} \int_0^{2\pi} X(\omega)e^{j\omega nT} d\omega \quad (11.4)$$

傅里叶变换的一个重要特性(图 11.1(b))是,它在正负方向上每隔一个采样频率区间就重复一次。另外,值得注意的是,在 $0 < \omega < \omega_s/2$ 区间内的分量是 $\omega_s/2 < \omega < \omega_s$ 内分量的共轭复数。现在习惯采用规一化频率,即将采样周期 T 设为 1。

考虑规一化频率时,离散信号的傅里叶变换可写为:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n} \quad (11.5a)$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n} d\omega \quad (11.5b)$$

这个傅里叶变换是连续的，且每隔一个采样频率重复一次。

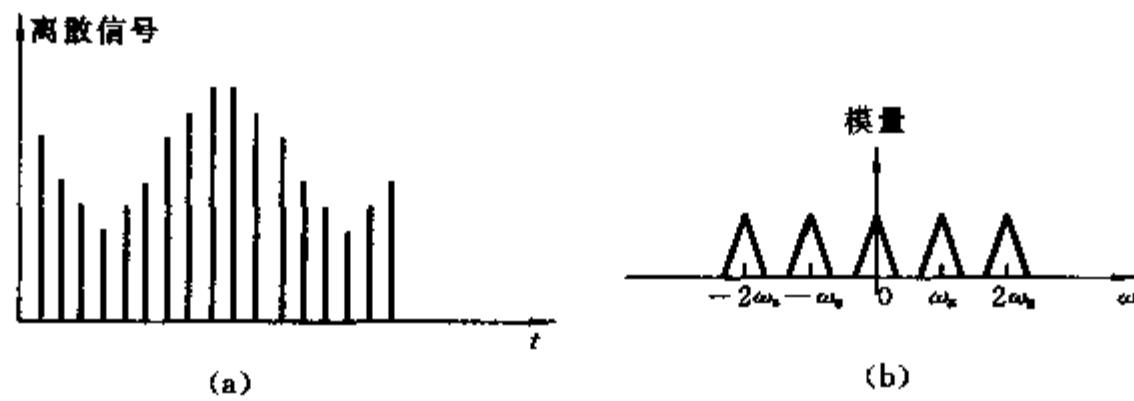


图 11.1 傅里叶变换
(a)一个离散时间信号;(b)傅里叶变换系数

11.1.2 周期信号的离散傅里叶变换

离散傅里叶变换(DFT)是指，对离散周期信号的傅里叶级数系数的计算，它类似于求周期信号的傅里叶系数，但也有某些显著的区别。第一是，在离散时间域中，积分变为求和；第二是，这种变换只能估算出有限数目的复系数，即原始信号在一个周期内的原始数据点数。因此，把每条谱线称为基本周期的第 k 个谐分量，而不使用 Hz 或 rad/s 单位来描述具体频率值。用代数方法可得到正、反变换的表达式，即

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn} \quad (11.6a)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi}{N}kn} \quad (11.6b)$$

图 11.2 所示的是一个离散周期信号及其 DFT 的实部和虚部。第一条谱线($k=0$)给出了信号

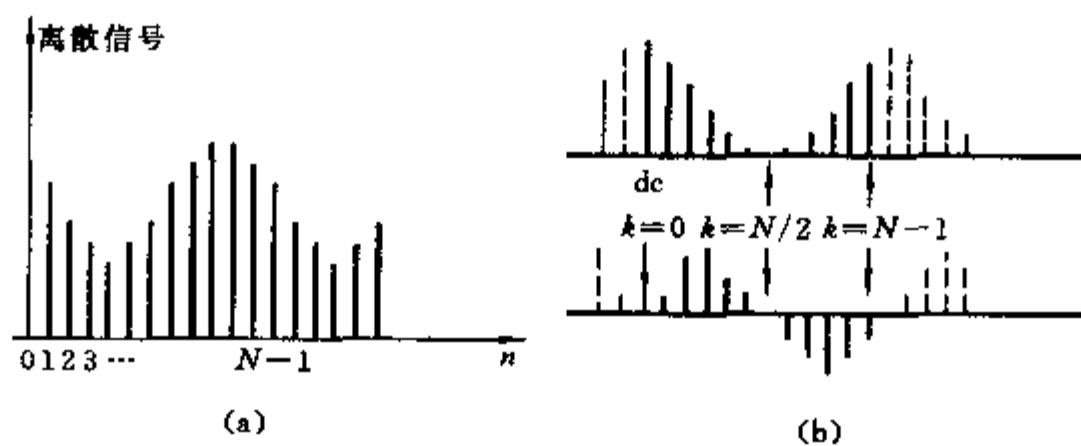


图 11.2 离散傅里叶变换
(a)离散周期信号;(b)DFT 的实部和虚部

中直流分量的幅值，第二条谱线对应于 N 点数据中一个周期的频率。该频率为 $2\pi/N$ 。第 N 条线对应于离散 N 样本数据序列的采样频率，而第 $k=N/2$ 条线代表奈奎斯特频率。利用 DFT 的对称性，发展了快速算法。此外，对称性还有两个重要的含义，第一是，变换将产生 N 条不同

的复谱线,第二是,其中一半谱线是多余的,因为一个真正的时域信号中的所有信息都包含在前 $N/2$ 条谱线中。这样就产生了 FFT。

11.1.3 快速傅里叶变换

Oppenheim 与 Schafer 合著的《数字信号处理》一书对 FFT 作了完整的讨论。FFT 一词泛指那些可用于估算含有 N 个等距离样本的信号的 DFT 的计算方法,其中 N 通常为 2 的幂。为了提高运算效率,必须把 DFT 分割成许多更小的 DFTs,为此要应用复数的对称性和周期性。

$$W_N^{kn} = e^{-j(2\pi/N)kn}$$

式中, W_N 以 $e^{-j(2\pi/N)}$ 递减。基于将序列 $x(n)$ 分割成更小序列的分解算法叫做时间抽取法。下面给出这种算法的原理,其中, N 为 2 的整数次幂。这时, $X(k)$ 可分解为奇和偶的两个 $N/2$ 点序列,即

$$X(k) = \sum_{n=2p+1} x(n)W_N^{nk} + \sum_{n=2p} x(n)W_N^{nk} \quad (11.7)$$

上式还可写成

$$X(k) = \sum_{p=0}^{\frac{N}{2}-1} x(2p)W_N^{2pk} + \sum_{p=0}^{\frac{N}{2}-1} x(2p+1)W_N^{(2p+1)k} \quad (11.8)$$

但是 $W_N^2 = W_{N/2}$,因此式(11.8)可写成

$$X(k) = \sum_{p=0}^{\frac{N}{2}-1} x(2p)W_{N/2}^{pk} + W_N^k \sum_{p=0}^{\frac{N}{2}-1} x(2p+1)W_{N/2}^{pk} = X_e(k) + W_N^k X_o(k) \quad (11.9)$$

式中,每个求和项分别是原始序列的 $N/2$ 个偶数点和 $N/2$ 个奇数点的 DFT。

将算出的两个 DFT 相加得到原始 N 点序列的 DFT。我们可以继续将每个 $N/2$ 点的 DFT 分解成两个 $N/4$ 点的 DFT,再将 $N/4$ 点 DFT 分解成两个 $N/8$ 点 DFT,等等。最终, N 点的 DFT 被分解为 2 点 DFT 的和与积。

图 11.3 所示的为计算 8 点原始序列的 $X(k)$ 的流程图。在 Oppenheim 与 Schafer 的《数字信号处理》中指出,需要进行 $N \log_2 N$ 复加和复乘的总次数为 $N \log_2 N$ 。求原始 N 点序列的 DFT 需要 N^2 个复加和复乘,由此可见,DFT 算法大大节省了计算时间。

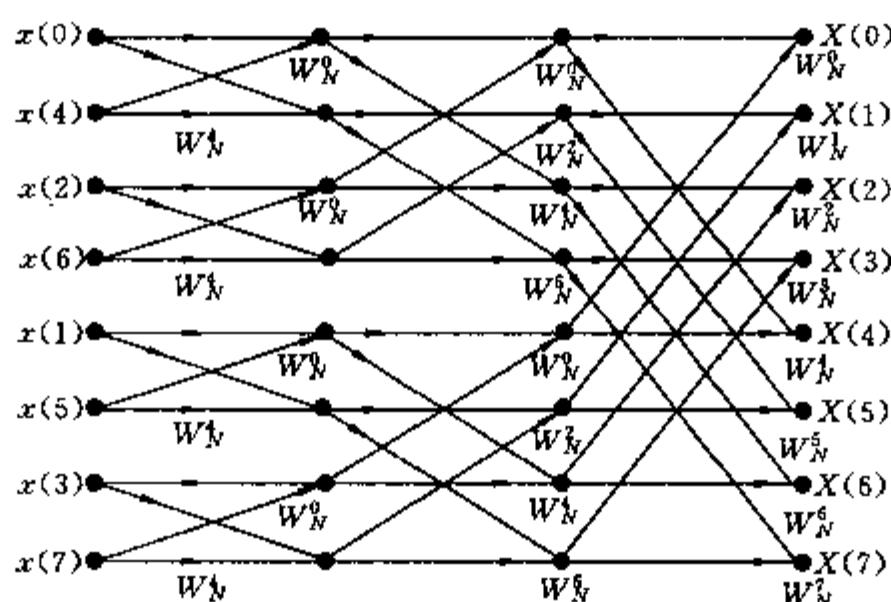


图 11.3 8 点 DFT 的时间抽取流程图

图 11.4 所示的是 FFT 和原始 DFT 的计算时间与 N 之间的关系曲线。FFT 的计算量比 DFT 至少小一个数量级。例如,配备了协处理器的微机可在不到 1s 的时间内完成 1024 点序列的 FFT 变换。当 N 不是 2 的整数次幂时,通常将有限长序列增加一些零值项,直到总点数达到最近的 2 次幂或可写出 FFT 算式的幂为止,这种技术称为补零法。为了尽可能减小误差,有时要将信号乘上一个有限长窗函数。当 N 是 2 的整数次幂而分析信号的 FFT 在 N 点内包含的周期数不为整数时也可使用窗函数法。这时由信号中不完整的周期所引入的误差可以通过适当选择窗口类型而减小。

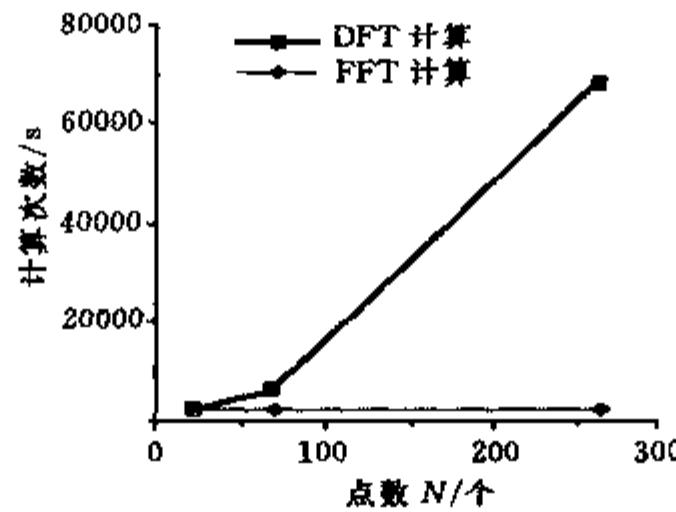


图 11.4 FFT 节省计算量

11.1.4 用 C 语言编写的 FFT 函数

图 11.3 所示的计算流程图,描述了当有限长序列点数为 2 的整数次幂时其 FFT 的计算方法。要完成这样的计算,可以使用两个存储寄存器阵列,一个用于存放被计算的阵列,另一个用于存放计算中所使用的数据。例如,在计算第一个阵列时,第一组存储寄存器存放输入数据,而第二组存储寄存器存放第一阶段的计算结果。

为了完成基于“蝶形”图的计算,输入数据必须倒序存储。实际上,输入数据是采用位颠倒方式存储的。

为了表明上述含义,下面用三位二进制数写出输出数据和对应的输入数据:

$$\begin{aligned}
 X(000) &\longrightarrow x(000) \\
 X(001) &\longrightarrow x(100) \\
 X(010) &\longrightarrow x(010) \\
 X(011) &\longrightarrow x(110) \\
 X(100) &\longrightarrow x(001) \\
 X(101) &\longrightarrow x(101) \\
 X(110) &\longrightarrow x(011) \\
 X(111) &\longrightarrow x(111)
 \end{aligned}$$

如果 $(n_2\ n_1\ n_0)$ 表示输入数据 $x(n)$ 的二进制形式,则 $\{x(n)\}$ 必须重新排序,使 $x(n_2\ n_1\ n_0)$ 的新位置为 $x(n_0\ n_1\ n_2)$ 。

图 11.5 所示的是 FFT 算法的 C 语言程序框架,它将输入阵列 $x[nn]$ 重新排序。UW DigiScope 中的 FFT 函数允许用户对信号进行补零或作不同窗函数处理。

```

#define RORD (a,b) tempr=(a);(a)=(b);(b)=tempr
...
...
float tempr,x[512];
int i,j,m,n,nn;
...
...
nn=512;
n=nn<<1;
j=1;
for (i=1;i<n;i++){
    if (j>1){
        RORD (x[j],x[i]); /* this is the bit-reversal section of */
        RORD (x[j+1],x[i+1]); /* a FFT computation routine */
    }
    m=n >>1;
    while (m>=2 && j>m){
        j-=m;
        m >>1;
    }
    j+=m;
}

```

图 11.5 用位颠倒计算的 C 语言程序

11.2 相关性

现在来研究一下数据群之间或信号之间相关性的概念。数据群之间的相关性是指它们彼此之间以一种结构化的方式移动或变化。在研究信号间的相关性时，只考虑已数字化形成数据群的信号。

11.2.1 时域相关性

对于 N 对数据 $\{x(n), y(n)\}$ ，其相关系数定义为：

$$r_{xy} = \frac{\sum_{n=1}^N (x(n) - \bar{x})(y(n) - \bar{y})}{\sqrt{\sum_{n=1}^N (x(n) - \bar{x})^2 \sum_{n=1}^N (y(n) - \bar{y})^2}} \quad (11.10)$$

如果要分析有限长的信号，则必须定义两个信号的互相关函数：

$$r_{xy}(k) = \frac{\sum_{n=1}^N (x(n) - \bar{x})(y(n+k) - \bar{y})}{\sqrt{\sum_{n=1}^N (x(n) - \bar{x})^2 \sum_{n=1}^N (y(n) - \bar{y})^2}} \quad (11.11)$$

在两个输入信号相同的情况下，互相关函数变成自相关函数。自相关函数的定义为：

$$r_{xx}(k) = \frac{\sum_{n=1}^N \{x(n) - \bar{x}\} \{x(n+k) - \bar{x}\}}{\sum_{n=1}^N \{x(n) - \bar{x}\}^2} \quad (11.12)$$

图 11.6 所示的是用阻抗描记法同时记录的 4 组人体呼吸信号的互相关函数。在图 11.6(a) 中, 4 组信号是沿被测者的腋线上不同点采集到的。记录开始时, 被测者做有节律的呼吸, 中期做不呼吸的移动, 后期又恢复有节律的呼吸。在图 11.6(b) 中, 给出每两个通道信号间的互相关函数, 以便将运动和节律性呼吸区分开。

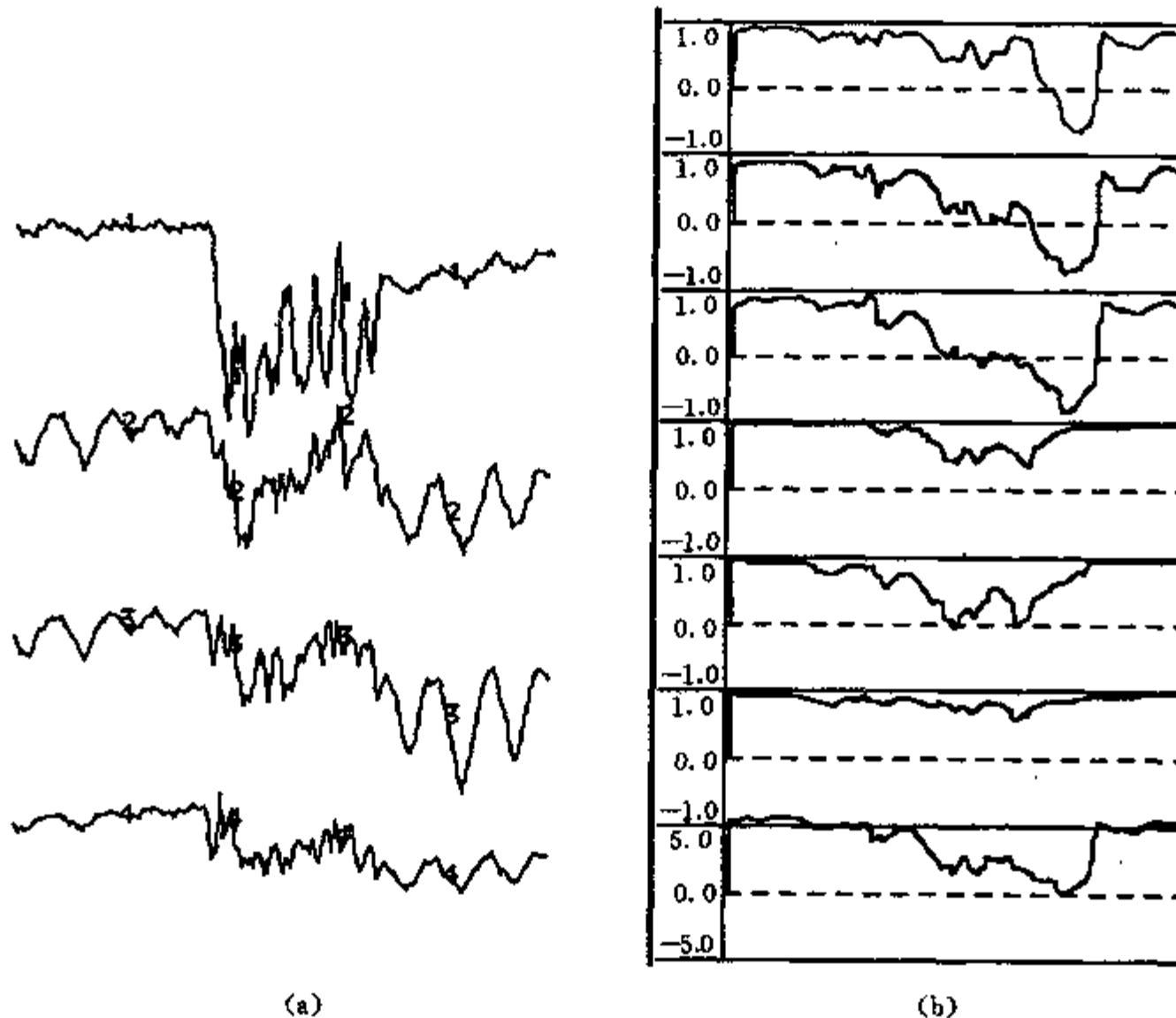


图 11.6 用阻抗呼吸运动描记法记录的呼吸信号

- (a) 从上到下, 信号 1, 2, 3, 4 是用 5 次/s 的采样率沿被测者的腋中线同时记录的;
- (b) 从上到下, 曲线分别表示通道 1—2, 1—3, 1—4, 2—3, 2—4, 3—4 间的互相关函数和平均相关的系数

11.2.2 频域相关性

互相关的定义最初用于连续信号。如果 $h(t)$ 和 $g(t)$ 是两个连续信号, 则它们的互相关函数定义为:

$$C_{gh}(t) = \int_{-\infty}^{+\infty} g(\tau) h(t + \tau) d\tau \quad (11.13)$$

互相关函数的傅里叶变换满足:

$$C_{gh}(\omega) = G(\omega)^* H(\omega) \quad (11.14)$$

式中, $G(\omega)^*$ 是 $G(\omega)$ 的共轭复数。

如果将 $h(t)$ 和 $g(t)$ 数字化，则可得到下面的近似互相关函数：

$$C_{gh}(m) = \frac{1}{N} \sum_{n=0}^{N-1} g(n)h(m+n) \quad (11.15)$$

式(11.5)也称为互相关函数的偏差估计量。在两个输入离散信号的 DFT 和偏差估计量的 DFT 之间有下列关系：

$$C_{\text{err}}(k) = G(k)^* H(k) \quad (11.16)$$

因此，两个离散信号的互相关函数也可以由求它们的 DFTs 积的反 DFT 得到。这可以用 FFT 和反 FFT(IFFT) 算法通过下式完成，以提高运算速度：

$$C_{\text{err}}(n) = \text{IFFT}(\text{FFT}^*(g) * \text{FFT}(h)) \quad (11.17)$$

结果表明，在不运动而做正常呼吸时，通道 2 和通道 3 相关性很强，而在运动并不呼吸时，二者相关性较差。

11.2.3 互相关函数

图 11.7 所示的 C 语言程序用于计算两个 512 点的输入序列 $x[512]$ 和 $y[512]$ 的互相关函数，并将输出数据存入 $r_{xy}[512]$ 。该程序的思想可用于开发求 ECG 信号和模板间互相关的 C 语言程序。此时，数组 $y[]$ 必须与模板维数相同。

```
/* The crosscorrelation function of x [] and y[] is */
/* output into rxy [] */

void corr(float *x, float *y)
{
    int i, m, n;
    float s, s1, s2, xm, ym, t;
    float rxy[512];
    n = 512;
    s = s1 = s2 = xm = ym = 0.0;
    for (i = 0; i < n; i++)
        xm = xm + x[i];
    ym = ym + y[i]; /* the arithmetic mean of x [] and y[] */
    /* computed */
    xm = xm / (float)n;
    ym = ym / (float)n;
    for (i = 0; i < n; i++)
        s1 = s1 + pow((x[i] - xm), 2.0);
        s2 = s2 + pow((y[i] - ym), 2.0);
    }
    s = sqrt(s1 * s2);
    for (m = 0; m < n; m++)
        t = 0.0;
        for (i = 0; i < n; i++)
            t = t + (x[i] - xm) * (y[(i+m)%n] - ym);
        }
        rxy[m] = t / s;
    }
}
```

图 11.7 计算互相关的 C 语言程序

11.3 卷 积

11.3.1 时域卷积

我们知道,信号通过一个线性系统的路径可以在频域中用系统的频率响应来描述。在时域中,系统对一个特定输入信号的响应可以用卷积表示。与频域中为求输出信号的频谱而将输入信号的频谱与分析系统脉冲响应的频谱相乘一样,卷积是时域的。

如果连续信号 $x(t)$ 加在一个脉冲响应为 $h(t)$ 的模拟系统输入端,则输出信号 $y(t)$ 的表达式为下面的卷积:

$$y(t) = \int_{-\infty}^{+\infty} x(t - \tau)h(\tau)d\tau \quad (11.18)$$

对离散信号,卷积式变为:

$$y(m) = \sum_{n=0}^{N-1} x(m-n)h(n) \quad (11.19)$$

式中, $x(n)$ 是输入信号, $h(n)$ 是经过采样的系统脉冲响应, $y(n)$ 是输出信号。式(11.19)可用于实现有限脉冲响应数字滤波器。此时, $h(n)$ 为代表 FIR 滤波器系数的有限长序列。

图 11.8 所示的是用阻抗呼吸描记法记录的呼吸信号,图(b)所示的是心动伪迹的 9 系数低通 FIR 滤波器的输入和输出信号。该滤波器的角频率为 0.7Hz, 阻带内的衰减约为 20dB。

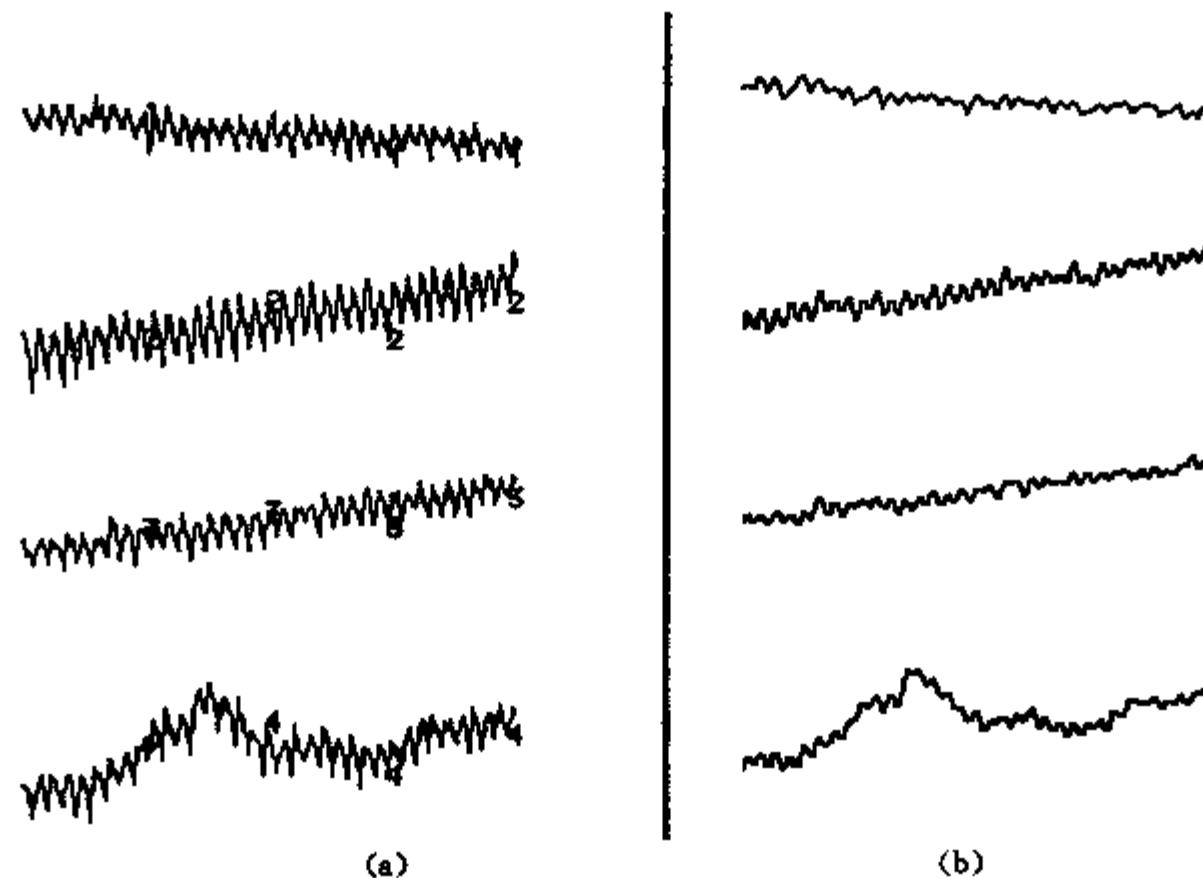


图 11.8 用阻抗呼吸描记法记录的呼吸信号

(a)4个信号是以5次/s的采样率从4个不同的通道采集的;(b)经过9系数低通滤波器后相应的信号,在图(a)中附加噪声所代表的心动伪迹被大大衰减了

11.3.2 频域卷积

时域卷积常用“*”符简单表示为:

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

或对于离散信号来说：

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

对于 DFTs 来说：

$$Y(k) = X(k)H(k)$$

如果用 FFT 计算时域卷积，则该方法叫“快速卷积”，即

$$x(n) * h(n) = \text{IFFT}(X(k)H(k)) \quad (11.20)$$

式中， $X(k)$ 和 $H(k)$ 用 FFT 算法计算。

频域卷积与时域卷积的定义相似，对于连续谱，可用积分表示为：

$$Y(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega - \Omega)S(\Omega)d\Omega \quad (11.21)$$

$Y(\omega)$ 对应的时域信号为：

$$y(t) = x(t)s(t)$$

因此，两个信号在时域内的乘积等于它们的傅里叶变换在频域内的卷积。

对于离散信号，其原理相同。我们知道，离散信号频谱的基本分量在 $-f_s/2 \leq f \leq +f_s/2$ 区间内，其中 f_s 是采样频率。在此区间外，采样信号的频谱在正负频率方向相同地重复。如果两个离散信号在时域内相乘，得到的频谱也在宽度为采样频率的各区间内相同地重复。重复函数显然是两个采样信号的傅里叶变换的卷积。值得注意的是，其形状与具有采样信号包络线的连续时间信号的卷积谱形状是不同的。

频域卷积是信号窗口法的基础。例如，如果 $h(n)$ 是频率特性为 $H(\omega)$ 的理想低通滤波器的脉冲响应，则 $h(n)$ 为无限长序列。为了实现近似于 $H(\omega)$ 的 FIR 滤波器，必须将 $h(n)$ 乘上窗口函数，得到

$$h'(n) = w(n)h(n) \quad (11.22)$$

式中， $w(n)$ 是有限长窗口序列。我们可以用频域卷积得到该 FIR 滤波器的傅里叶变换 $H'(\omega)$ ，即

$$H'(\omega) = H(\omega) * W(\omega) \quad (11.23)$$

式中， $W(\omega)$ 是窗口序列的傅里叶变换。 $h(n)$ 与 $h'(n)$ 近似程度的好坏有赖于窗口序列的特性。

分析窗口函数特性的一种方法是研究它的傅里叶变换。在这种方法中，需要分析阻带衰减和通带宽度。图 11.9 所示的是最常用于低通滤波器设计的适用于几种类型的窗口的重要参数。设计者应该在通带宽度、系数个数和阻带内最小衰减三者之间进行权衡。如图 11.9 所示，对于中等通道宽度，用海明窗可得到最佳结果。如果设计者对通带宽度不感兴趣，则用布莱克曼窗可得到最佳结果。若想详细了解窗函数法的原理，可参阅 Oppenheim 和 Schafer 合著的《数字信号处理》。

窗	主瓣的传输宽度	最小阻带衰减
矩形	$2f_s/N$	-21dB
海宁	$4f_s/N$	-44dB
海明	$4f_s/N$	-53dB
布莱克曼	$6f_s/N$	-74dB

图 11.9 不同窗函数的特性

N 代表函数中用于描述窗口的系数个数， f_s 是采样频率

11.3.3 卷积函数

图 11.10 所示的是一个 C 语言程序, 它计算时域内 512 点输入数据序列 $x[512]$ 和 FIR 滤波器系数阵列 $h[ncoef]$ 之间的卷积, 并将输出数据存入 $cxy[512]$ 。

```
/* The convolution between x[] and h[] is saved into cxy[] */

conv (float *x, float *h, int ncoef)
{
    int i, m, n;
    float cxy[512];
    n = 512;

    for (m = 0; m < n; m++) {
        cxy[m] = 0.0;
        for (i = 0; i < ncoef && i <= m; i++) {
            cxy[m] += h[i] * x[m - i];
        }
    }
}
```

图 11.10 计算卷积的 C 语言程序

11.4 功率谱估算

11.4.1 Parseval 定理

Parseval 定理描述了时域与频域间的能量守恒原理。对于周期为 T 的周期信号 $f(t)$, 由 Parseval 定理可知, 在已知傅里叶级数系数 $a_k, b_k, k=0, 1, \dots, \infty$ 的情况下, 可用下式计算信号中的平均功率:

$$\frac{1}{T} \int_{-T/2}^{T/2} f^2(t) dt = a_0^2 + \sum_{k=1}^{\infty} \left[\frac{a_k^2}{2} + \frac{b_k^2}{2} \right] \quad (11.24)$$

对于连续的非周期信号, 可以类似地得到 $f(t)$ 和它的傅里叶变换之间的关系, 即

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |F(\omega)|^2 d\omega \quad (11.25)$$

同样, 对于实域离散信号, 经傅里叶变换, 可得到

$$\sum_{n=-\infty}^{\infty} f^2(n) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |F(\Omega)|^2 d\Omega \quad (11.26)$$

在 DFT 情况下, 假定时域信号以 N 为周期相同地重复, 则其 DFT 将在以采样频率为宽度的各区间内重复。这时 Parseval 定理可表示为:

$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \quad (11.27)$$

为了估算信号的平均功率, 需计算均方幅值, 并作下述近似:

$$\frac{1}{T} \int_{-T/2}^{T/2} f^2(t) dt \approx \frac{1}{N} \sum_{n=0}^{N-1} f^2(n) \quad (11.28)$$

这里所用的功率谱估算(PSE)法是基于周期图的概念建立的。如果对一个函数 $c(t)$ 进行采样,

并用 FFT 法计算它的 DFT，则会得到

$$C_k = \sum_{n=0}^{N-1} c(n) e^{-jkn2\pi/N}$$

这时对 $\frac{N}{2}+1$ 个频率点的功率谱的周期图估算定义为：

$$\begin{aligned} P(0) &= \frac{1}{N^2} |C_0|^2 \\ P(k) &= \frac{1}{N^2} (|C_k|^2 + |C_{N-k}|^2) \quad k = 1, 2, \dots, \frac{N}{2} - 1 \\ P(N/2) &= \frac{1}{N^2} |C_{N/2}|^2 \end{aligned} \quad (11.29)$$

由 Parseval 定理可知，均方幅值等于 $N/2+1$ 个 P 值的和。我们必须知道该估算法的精度及如何对其进行改进。下面给出两种改善估算法性能的方法。

11.4.2 平均修正周期图的沃尔什法

周期图是不一致的谱估算法，其估算方差在记录长度接近无穷时不趋近于 0。沃尔什提出了一种改进方法，它是基于将 N 点数据记录 $x(n)$ 分割成一段段含有 M 点的部分 $x_k(n)$ ，各段之间重叠了 L 个样本点的事实提出的。如 $L=M$ ，则 $N=(K+1)M$ ， K 是段数。将一个窗函数作用于每段，然后计算每段的周期图。最后，将这些周期图平均，即得到沃尔什估算结果。

11.4.3 Blackman-Tukey 谱估算

Blackman-Tukey 估算法可由三步完成。第一步，从记录到的 N 点数据中估算出自相关序列 $\phi_{xx}(m)$ （其中， $-M \leq m \leq M$ ）的中间 $2M+1$ 个样本。第二步，将一个窗函数作用于估算后的自相关延迟。第三步，计算引入窗函数后的自相关估算的 FFT，得到 Blackman-Tukey 估算结果。参数 M 和窗函数类型必须根据应用场合适当地选取。

11.4.4 压缩谱阵列和灰度图

在压缩谱阵列(CSA)法中，将得到的谱结果按时间顺序画出（每个功率谱紧挨着前一个谱），以便产生三维效果，使之容易被理解。为了在打印输出的二维图像上显示这种效果，将每个代表下一时间周期的功率谱的原点依次在 x 轴和 y 轴方向上移动。原点在 y 轴上相对 x 轴移动得越多，视角就越窄，单个谱分离得就越好，但是这样就很难跟踪通过几个时间周期的频率成分。图 11.11(a)所示的是用 CSA 对一个糖尿病人的 EGG(胃电图)谱分析结果，该病人的时域记录表明，其存在快速节律不齐。

图 11.11(b)所示的是相应的灰度图。这是个二维图形， x 轴代表频率， y 轴代表时间，点的灰度代表谱功率大小。点越暗，则那点的谱功率越大。灰度图上每个数据点用一个 5×5 点阵表示。每个点阵的灰度水平为 26 个灰度级中的一级，从所有点全暗到全亮，其他值以最大值为标准比例列入最接近的代表灰度级的整数值。灰度图不能像 CSA 那样提供高幅值分辨率，但它推动了频移的研究。

11.4.5 功率谱函数

图 11.12 所示的是用 C 语言编写的计算 N 点数据序列 ($N=512$) 功率谱的 C 语言程序。

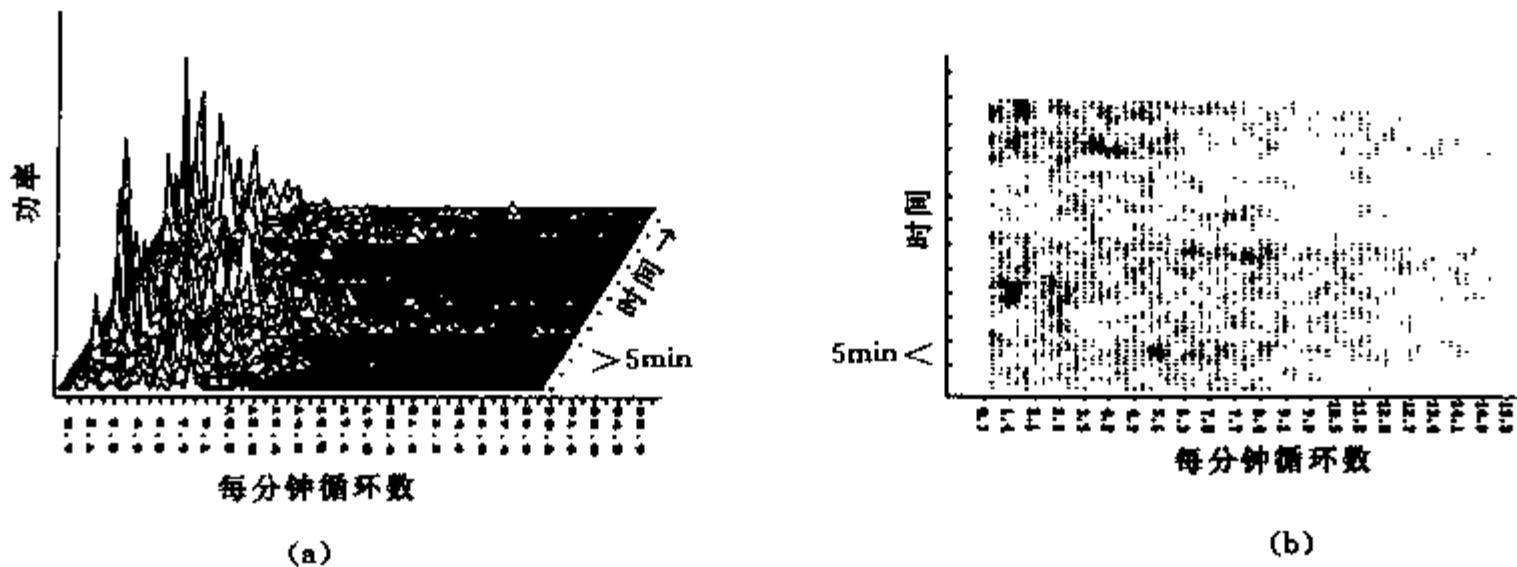


图 11.11 一个糖尿病人的胃电图(EGG)

(a)压缩谱阵列(CSA); (b)灰度图

输入数据是来自输入缓冲器输出的整数,被转换成浮点型数后用于计算功率谱。输出数据被按比例排列并由浮点型转换为整型数存入一个输出文件中。该程序是基于沃尔什的周期图平均思想设计的。因此,该函数将输入数据序列分解成两个 256 点的段,用窗函数作用于每段,完成每段的功率谱估算。最后,按照窗口的作用将每段结果平均并得到其灰度级。

```
#define WINDOW (j,a,b)(1.0-fabs(((j)-1)-(a)*(b))/* parzen */
#define SQR(a) (sqrarg=(a),sqrarg*sqrarg)/* modulus of a,squared */
...
for(j=1;j<=mm;j++) {
    w=WINDOW (j,facm,facp);

    w1[2*j-2] *=w; /* the real part of them data segment is windowed */
    w1[2*j-1] =0.0; /* the imaginary part is set to zero */
}

...
fft(w1-1,mm,1); /* the fft of the windowed signal is performed */
/* the power spectrum estimate for the windowed data segment is computed */
p[0] +=(SQR(w1[0])+SQR(w1[1]));
for (j=1;j<m;j++){
    j2=2*j;
    p[j] +=(SQR(w1[j2])+SQR(w1[j2+1])+SQR(w1[-j2+4*m])+SQR(w1[-j2+4*m+1]));
}
p[m] +=(SQR(w1[mm])+SQR(w1[mm+1]));
...
for (j=0;j<=m;j++) p[j] /=scale; /* the PSE is scaled */
/* considering the window effects */
```

图 11.12 计算功率谱的 C 语言程序

11.5 实验:ECG 的频域分析

下面为研究信号频率特性的实验,另外一个实践来源出自 Alkin(1991)写的 PC—DSP。

11.5.1 周期信号的功率谱分析

用 UW DigiScope 软件的(G)enwave 命令产生由不同的采样率得到的具有不同基频的正弦波和方波信号。用(P)wr Spect 命令计算每个信号的频谱,并讨论采样率对结果的影响。如何才能用功率谱获得实际的傅里叶系数呢?

11.5.2 ECG 的功率谱分析

- (1)用(P)wr Spect 命令得到一段 ECG 频谱中主峰的频率。所得的结果是所希望的吗?
- (2)从主菜单中选择 ad(V)Ops, 用(T)emplate 选择一个 QRS 复波, 用(P)wr Spet 命令得到经过补零的 QRS 复波的谱。QRS 复波频谱中主峰的频率是你所希望的值吗?
- (3)选择了 QRS 复波模板后, 用(W)indow 命令将窗函数作用于所选模板, 然后运行(P)wr Spect命令。确认不同窗函数的作用。
- (4)选择一个 P-QRS-T 段作为模板, 用(P)wr Spect 命令得到 ECG 的功率谱估算。找出对应于 ECG 频谱中主峰的频率, 它与第一部分的结果有何不同?

11.5.3 ECG 的互相关

用(T)emplate 命令选择一个 QRS 复波, 用(C)orrelation 命令求该 QRS 复波与 ECG 的相关性。解释互相关函数的输出与所选模板尺寸间的关系。互相关函数的峰值点与所造 QRS 复波间的时间延迟是什么? 从一个磁盘文件中读入另一段 ECG, 然后求模板与新 ECG 的互相关函数, 解释所得结果。

复习思考题

11.1 当 $x(n)$ 为每个周期内含有 N 个样本的数字周期信号时, 推导式(11.24)和式(11.27)。

11.2 如果 $x(n)=1.0+\cos(2\pi n/4.0)$, 求 $n=0,1,\dots,7$ 时, $x(n)$ 的 DFT 变换。用 C 语言编写程序, 使其生成一个存放信号的数据文件。用 UW DigiScope 程序读取该数据文件, 并补足相应数目的零值。用(P)wr Spect 命令求未施加窗函数的补零信号的功率谱, 将该结果与你的手算结果进行比较。解释二者间的差异。

11.3 产生一个存放 $x(n)=\sin(\pi n/512.0)$, $n=0,1,\dots,511$ 的文件。求其在未施加窗函数时的功率谱。用 0 代替该信号的最后 10 个样本值。求补零后的信号在有窗函数和无窗函数作用时的功率谱。解释其差别。

11.4 选择问题 11.2 中信号的一个周期作为模板。求该模板和信号间的互相关函数。解释互相关函数的形状。计算峰值并将其与你的实验结果相比较。

11.5 用 500 次/s 的采样率采集一个 100Hz 带宽的 ECG 信号。
①画出采样后得到的新数字信号的近似频谱, 并在坐标轴上标出重要的点;
②在同一个图中, 画出从一组正常 QRS 复波平均后得到的近似谱图。

第十二章 ECG 的 QRS 复波的检测

近几年来,采用微机对 ECG 进行分析处理已大有增长的势头。研究表明,基于微机的医学仪器系统能以极其有效的方式开展医疗服务。实际上,许多系统已被设计用来完成对诸如 12 导联离线 ECG 分析、Holter 磁带记录分析以及病人实时监护这样的信号处理任务。所有这些应用都要求对 ECG 的 QRS 复波进行准确的检测。例如,自由活动的病人身上的心律失常监护仪时刻实时分析着病人的 ECG,一旦发生心律失常,监护仪就存储这一时刻异常的 ECG。该类型的监护仪需要有精确识别 QRS 复波的能力。这样,QRS 复波的检测就成为许多 ECG 信号处理分析系统的重要组成部分。

本章将讨论业已开发并用于监测 QRS 复波的许多方法中的几种。首先讨论 QRS 复波的功率谱,然后评述 QRS 复波的多种检测算法。

12.1 ECG 的功率谱

ECG 的功率谱能提供 QRS 复波的有用信息。本节再次谈及以前提过的功率谱的概念,同时也给出 QRS 复波功率谱的概念。一段包含近两拍心跳的 512 个采样点的功率谱(基于 FFT)由一组系数构成,其中最大值位于心跳速率的频率附近。

将规范化频率与采样频率相乘可以确定心率,也可获得 QRS 复波频谱的有用信息。为了得到这些信息,ECG 中的 QRS 复波必须被选作模板,并在分析功率谱前将基线调至零电位。所得的频率谱峰值就对应于 QRS 复波的峰值。

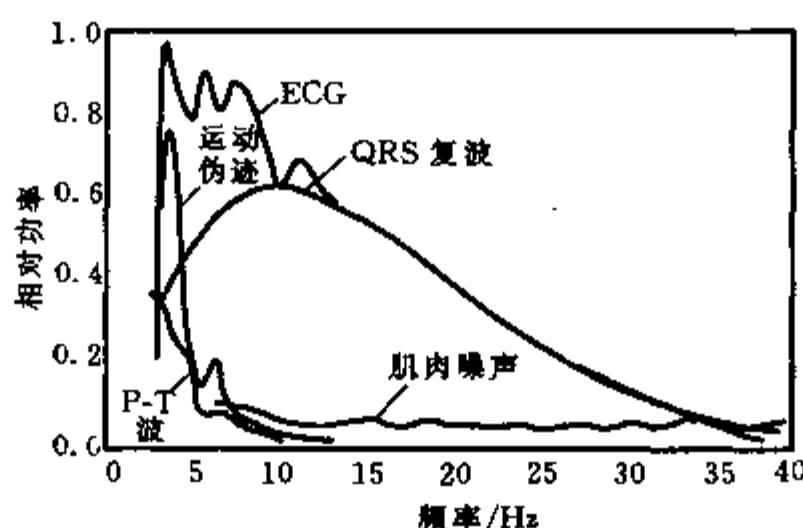


图 12.1 QRS 复波、P 波和 T 波,肌肉噪声和运动
伪迹的相对功率谱(150 次心跳的平均值)

除 QRS 复波外,ECG 波形还包括 P 波和 T 波、来自电源线干扰的 60Hz* 噪声、来自肌肉的 EMG、电极和皮肤接触处的运动伪迹以及来自手术室中电外科仪器的干扰。许多临床仪器

* 注:美国电网的频率为 60Hz。

如心率计和心律失常监护仪等都需要进行精确实时的 QRS 复波检测。因此,有必要从其他噪声源中抽取感兴趣的 QRS 复波信号。

将前期研究中获得的 ECG、QRS 复波、P 波和 T 波、运动伪迹及肌肉噪声的相对功率谱一并总结归纳成图 12.1 所示的图谱。

12.2 带通滤波方法

依据对 ECG 信号中各种信号分量功率谱的分析,能够设计出一种从 ECG 中有效地筛选出 QRS 复波的滤波器。我们完成的另一项研究检查了来自 3876 次心跳的 ECG 和 QRS 复波的功率谱图。图 12.2 所示的是信噪比(SNR)作为频率的函数的图形。由 ECG 信号、QRS 复波和其他噪声的功率谱分析表明,最大的 SNR 值可以通过中心频率为 17Hz、Q 值为 3 的滤波器获得。12.3 节和一项实验(指本章后面的实验——译者注)将研究这样的滤波器在不同 Q 值条件下所获得的效果。

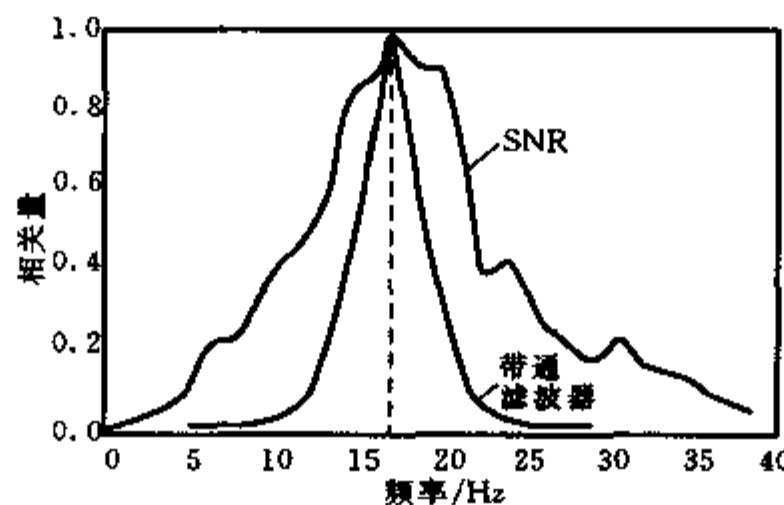


图 12.2 QRS 复波参比其他所有信号噪声的信噪比(SNR)图(基于 3875 次心跳)

由图可知,用于心率计的最佳带通滤波会使 SNR 达到最大。

12.2.1 双极点递归滤波器

可以用 C 语言来实现一个简单的双极点递归滤波器对 ECG 信号进行带通滤波。该滤波器的差分方程为:

$$y(nT) = 1.875y(nT-T) - 0.9219y(nT-2T) + x(nT) - x(nT-2T) \quad (12.1)$$

在该滤波器的设计中,假定 ECG 的采样频率为 500 次/s。1.875 和 0.9219 分别为实际设计值 1.87635 和 0.9216 的近似值。由于系数是用 2 的指数形式来表达的,所以借助 C 语言的移位操作能较快地完成乘法操作。图 12.3 所示的为实现式(12.1)的 C 语言程序段。

注意,在这段程序中,系数 1.87635 和 0.9216 由以下两式近似获得:

$$1.875 = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}$$

和

$$0.9219 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{32} + \frac{1}{64}$$

```

twopoleRecursive (int data)
{
    static int xnt, xm1, xm2, ynt, ym1, ym2 = 0;
    xnt = data;

    ynt = (ym1 + ym1 >> 1 + ym1 >> 2 + ym1 >> 3) +
        (ym2 >> 1 + ym2 >> 2 + ym2 >> 3 +
        ym2 >> 5 + ym2 >> 6) + xnt - xm2;

    xm2 = xm1;
    xm1 = xnt;
    xm2 = ym1;
    ym2 = ym1;
    ym1 = ynt;
    return (ynt);
}

```

图 12.3 实现一简单双极点递归滤波器的 C 语言程序

12.2.2 整数型滤波器

利用第七章给出的传递函数可形成整数型滤波器。心率计中的 QRS 检测器通常采用一个中心频率为 17Hz 的带通滤波器对 ECG 信号进行带通滤波。传递函数通式的分母允许在 60°、90°和 120°处有极点，这相当于带通滤波器的中心频率分别为 $T/6, T/4$ 和 $T/3\text{Hz}$ 。通过选择一个适当的采样频率就能得到所需中心频率。

Ahlistuom 和 Tompkins(1985)描述了一个可用于 QRS 检测的实用的滤波器，它是基于如下的传递函数建立的：

$$H(z) = \frac{(1 - z^{-12})^2}{(1 - z^{-1} + z^{-2})^2} \quad (12.2)$$

该滤波器在单位圆上的 12 个不同频率处有 24 个零点，极点位于 $\pm 60^\circ$ 处。ECG 信号以 200 次/s 的速率采样，然后使用转折点算法将采样速率压缩到 100 次/s。中心频率为 16.67Hz，标准带宽为 $\pm 8.3\text{Hz}$ 。振荡的周期近似为 240ms(下节将解释滤波器在不同 Q 值时的效果)。完成此传递函数的差分方程为：

$$\begin{aligned} y(nT) = & 2y(nT - T) - 3y(nT - 2T) + 2y(nT - 3T) \\ & - y(nT - 4T) + x(nT) - 2x(nT - 12T) + x(nT - 24T) \end{aligned} \quad (12.3)$$

12.2.3 不同 Q 值的滤波器响应

中心频率 $f_c=17\text{Hz}$ 的带通滤波器的 Q 值决定有用信号经过带通滤波时的衰减程度。增加有用信号(即 QRS 复波)的 SNR 也是必要的。滤波器的 Q 值由下式计算：

$$Q = \frac{f_c}{B_w} \quad (12.4)$$

Q 值太高将引起振荡响应。这种扰动必须在 20ms 内消除，以便一个 QRS 复波的扰动不至于干扰下一个 QRS 复波。对于中心频率为 17Hz 的滤波器，其最佳的 Q 的值为 5。图 12.4 所示的是不同 Q 值的效应。对于中心频率 $f_c = 17\text{Hz}$ 的带通滤波器，Q 值为 5 时，SNR 达到最大。

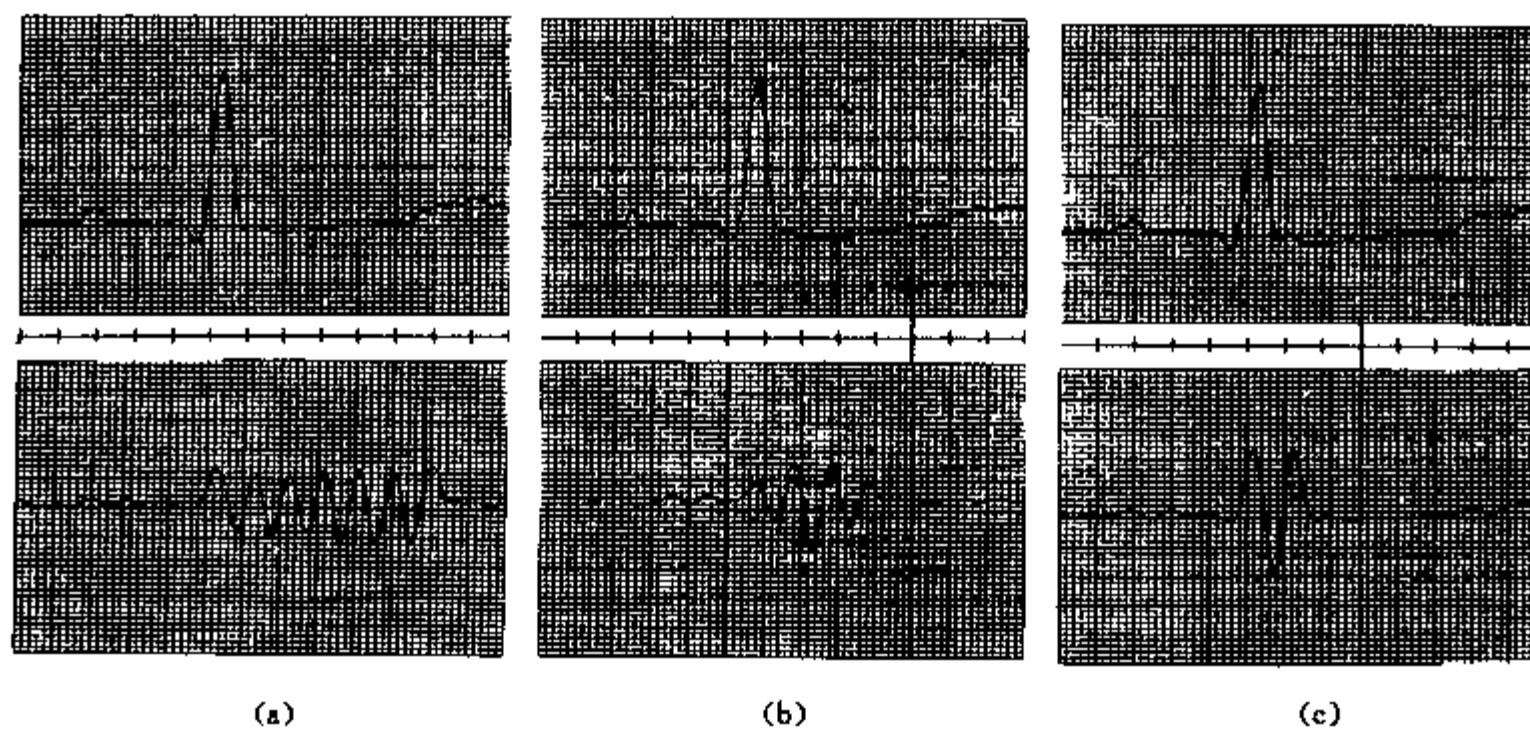


图 12.4 不同 Q 值效应。高 Q 值导致一个短暂振荡响应

(a) $Q = 8$; (b) $Q = 3$; (c) $Q = 1$

12.3 差 分 法

差分法构成了许多 QRS 复波检测算法的基础。由于它基本上是高通滤波器,因而微分增强了具有较高频率特征的 QRS 复波,同时削弱了具有较低频率成分的 P 波和 T 波。

以一阶、二阶微分为基础的一种算法首先由 Balda 等人(1977)提出,后经 Ahlstrom 和 Tompkins(1983)修改用于 ECG 的高速分析。随后,在比较某些类型的 QRS 检测算法中的噪声灵敏度的问题时,Friesen 等人(1990)也采用了这种算法。图 12.5 所示的是这种算法的信号处理过程。

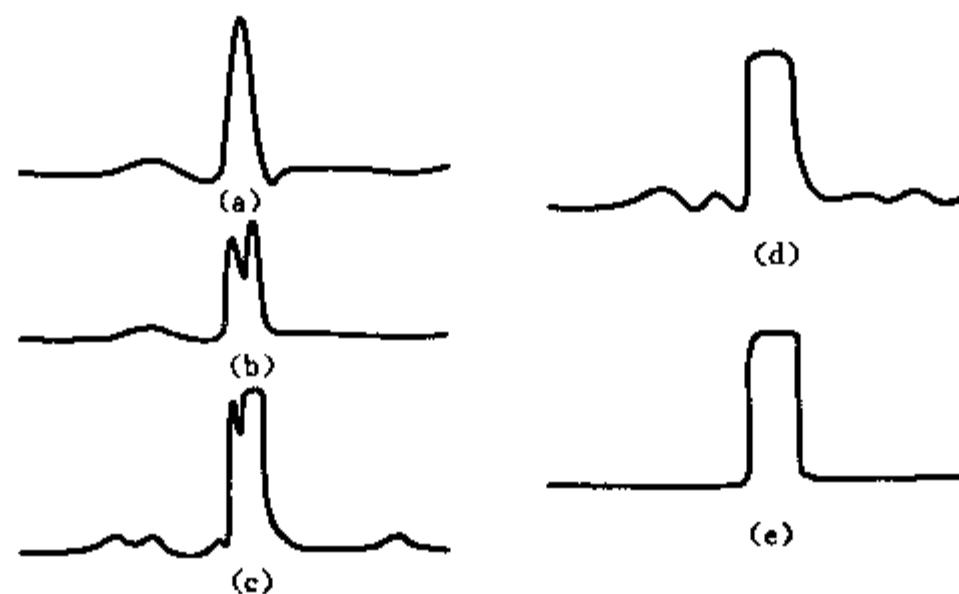


图 12.5 基于微分的 QRS 检测算法中阶段的信号

(a) 原始 ECG; (b) 平滑和整流后的一次微分;
 (c) 平滑和整流后的二次微分; (d) 平滑后的(b)和(c)之和;
 (e) 对应于每一个 QRS 复波的方波脉冲输出

ECG 信号一阶和二阶微分的绝对值用下式计算：

$$y_0(nT) = |x(nT) - x(nT - 2T)| \quad (12.5)$$

$$y_1(nT) = |x(nT) - 2x(nT - 2T) + x(nT - 4T)| \quad (12.6)$$

式(12.5)和式(12.6)分别乘以比例系数后相加，得

$$y_2(nT) = 1.3y_0(nT) + 1.1y_1(nT) \quad (12.7)$$

判别 $y_2(nT)$ 是否达到或超过给定的阈值，即

$$y_2(iT) \geq 1.0 \quad (12.8)$$

一旦 $y_2(iT)$ 中有一数据点满足此条件，后 8 个数据点就同这个阈值相比较。如果这 8 个数据点中有 6 个或更多点等于或超过此阈值，那么这一段就可能是 QRS 复波的部分。除了检测 QRS 复波外，这种算法还有一个优点，它能产生一个宽度与 QRS 复波成比例的脉冲。然而，该算法的缺点是对高频噪声特别敏感。

12.4 模板匹配法

本节将讨论 ECG 信号中进行模式分类的方法，该方法与人类的识别过程很相近。

12.4.1 模板互相关

如果两信号波形形状相互匹配，就称这些信号是相关的。相关系数是一个值，它确定两个或更多信号形状间匹配的程度。由 Dobbs 等人设计的一种 QRS 检测方法采用了互相关的方法。

求一个信号与另一个信号相关，要求这两个信号互相对准。在这种 QRS 检测方法中，希望检测到的信号的模板以数据形式存储起来。既然要求出模板与输入信号的相关，那么输入信号应该与模板相对准。Dobbs 等人提出了两种对准方式。

第一种方法是，利用每个信号上的基准点将模板和输入信号对准。这些基准点是通过其他处理方法得到的。如果模板上和信号上的基准点对准了，那么就能完成相关。

另一种方法是，考虑了一段输入信号和模板间的连续相关。每当一个新的信号数据点移进时，一个最老的数据点同时就从这段中移出（一种先进先出结构）。然后，求出这一段信号与模板（它有与信号点相同的数目）的相关系数。这种技术不需要指定信号基准点的处理时间。可以把模板看成一个窗口，它在输入信号上每次移动一个数据点。这样，当窗口移动穿过整个信号时，至少能发生一次感兴趣的信号段与此窗口匹配。

相关系数的值总是位于 +1 和 -1 之间。+1 值表明信号与模板准确匹配。正如前面提到的，这个系数值确定了研究中的两波形形状的匹配程度。实际信号的幅值对相关函数来说是无关紧要的。这种形状匹配，或 QRS 复波的识别过程，与识别信号的自然途径是一致的。

12.4.2 模板减去法

图 12.6 所示的是模板减去法。同本章中描述的其他方法相比，这是一种较简单的 QRS 检测方法。

在简单的模板匹配中，输入信号逐点从 QRS 模板中减去。如果两个波形完全匹配，则结果为零值。

这种算法首先存储一段对应于 QRS 波形的 ECG 输入信号。然后，这一段信号或模板与后

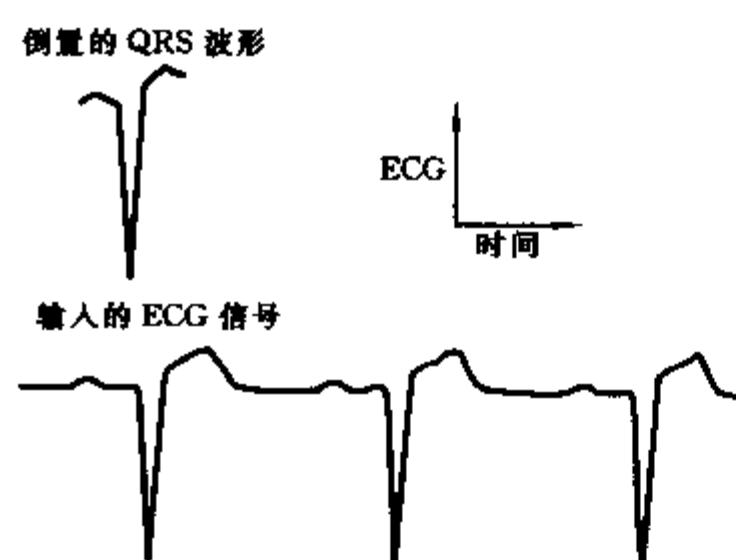


图 12.6 模板减去法

续输入的 ECG 信号相比较。将后读输入信号的每一点与模板上的对应点相减。当模板与这信号的一个 QRS 复波相匹配时，减去的结果是一个非常接近于零的值。这种算法仅需执行与模板点相同次数的减法操作。

12.4.3 基于句法的模板匹配

Furno 和 Tompkins (1982) 提出了一种建立在句法概念上的 QRS 检测器算法。这种算法采用了一些在许多图像识别系统中常用的基本方法。ECG 信号首先被简化成一组预先规定的标号，它们体现了 ECG 波形的特定形状。

图 12.7 所示的是一个正常 ECG 的一组标号，随后这组标号被送入如图 12.8 所示的有限状态自动机中。这个有限状态自动机基本上是一个状态转换图，它借助适用于大多数编程语言中的 IF...THEN 控制状态语句而构成。标号序列被送入这个自动机，如 zero, thrmup, normdown 和 normup 这样的序列标号将使自动机给出此 ECG 属正常类的报告结果。

标号序列必须从 ECG 信号推算出来，这可以通过分析输入信号差分序列得到。该算法将具有相同符号且超过某一给定阈值的差分值归为一组，并将组内差分值的和，以及差分个数赋给该组。

这种 QRS 检测算法开始具有一个学习的阶段，在学习阶段中，需要近似地确定正常 QRS 复波的峰值。出现一次幅值大于预定峰值一半的波形时，该算法就认为检测出一正常 QRS 复波。通过这一阶段，算法就告知有限状态自动机。此时的序号序列构成了一个正常的 QRS 复波。

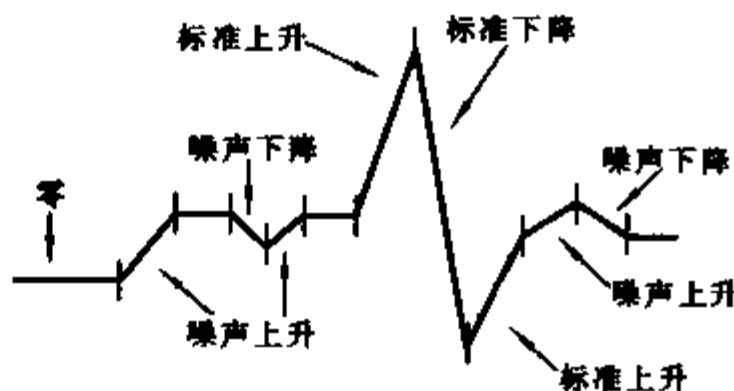


图 12.7 ECG 信号简化成标号

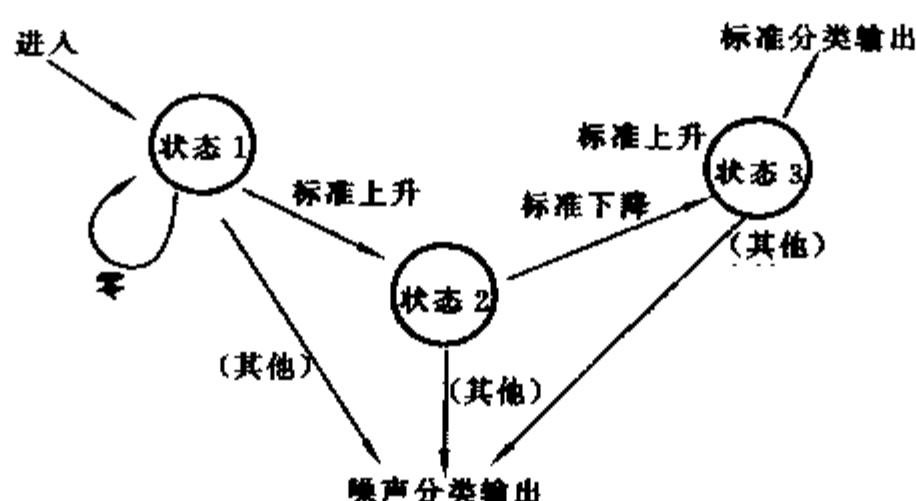


图 12.8 一个简单的自动机的状态转换图
它仅检测 QRS 复波与噪声状态转换适用于未来状态转换处
未注明的其他任何标号，从而离开一个特定状态

至此,可以设定对应正常 QRS 复波的微分组数目及和值的范围。根据其微分的数目及和值,算法可以给多个微分组赋予波形标志。例如,如果某微分组的和值与数目位于 QRS 复波上升或下降范围内,那么,将赋予该组一个 normup 或 normdown 的标号。如果数目值与和值并没有落入这一范围,则赋予 noiseup 或 noisedown 的标号。和值为零的组赋予 zero 标号。这样,这种算法就将 ECG 信号简化成为一组标号序列,然后输入给有限状态自动机用于 QRS 检测。

12.5 QRS 复波检测算法

一种实时 QRS 复波检测算法首先由 Pan 和 Tompkins(1985)创立,后来 Hamilton 和 Tompkins(1986)对此作出了进一步的描述。它通过分析斜率、幅度和宽度识别 QRS 复波。图 12.9 所示的是用于 ECG 信号分析中的各种滤波器级联构成的带通滤波器。随后,对信号进行微分、均方和时域平均。

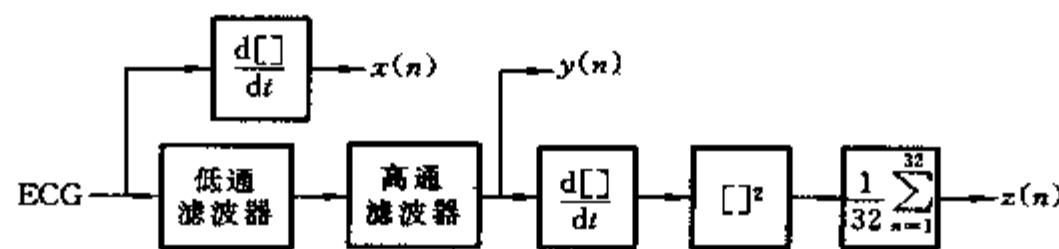


图 12.9 QRS 检测器的滤波步骤
 $z(n)$ 是时间或平均后的信号; $y(n)$ 是微分后的 ECG

在整数型数字滤波器的基础上可设计出一种带通滤波器,这种滤波器允许微处理器仅通过整数运算来进行信号处理,因而获得了通过浮点运算中难以达到的实时处理速度。由于不可能直接设计出这种具有特殊形式的带通滤波器,这种带通滤波器实际上是由级联的低通和高通滤波器构成的。这种滤波器能分离中心频率在 10Hz 处的 QRS 成分,削弱了具有低频特征的 P 波和 T 波以及基线漂移,同时也削弱了与肌电噪声和电源线干扰有关的高频分量。

下一步的处理就是进行微分,这是一个找出高斜率的标准方法,而高斜率通常能将 QRS 复波与其他的 ECG 波形区分开来。为了实现这种运算,所有的处理都通过线性数字滤波器来完成。

再下一步是进行非线性变换,即将信号采样点进行逐点平方运算。这种转换使数据在后续微分前为正值,同时增强了微分处理后信号中的高频分量,这些高频分量通常就是 QRS 复波的特征。均方后的波形通过一个移动窗口的积分器。该积分器求出均方的波形在 150ms 窗口下的面积后重新积分。选择的窗口宽度,窗口宽度要足以包括扩宽了的以及正常 QRS 复波在内,又要不至于使 QRS 复波与 T 波重叠。带通滤波器出来的波形和移动积分波形的自适应幅值的阈值与峰值信号水平和峰值噪声的不断修正的估计值有关。借助自适应阈值进行初步检测后,就可检测出该波形是否为 QRS 复波。

当检测到 QRS 复波后,测量算法还会计算出 QRS 的宽度。这样有两个波形特征适用于后续的心律失常分析——RR 间期和 QRS 宽度。

下面将逐步解释这种 QRS 检测算法。图 12.10 所示的是一采样的 ECG,它将作为下面处理步骤中的输入信号。

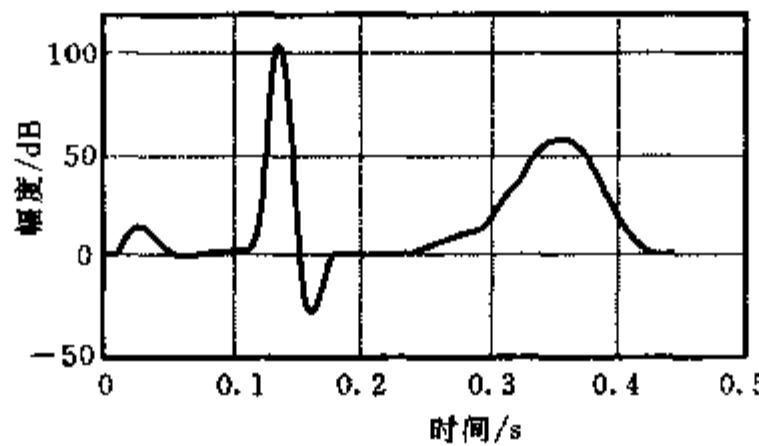


图 12.10 采样速率为 200 次/s 的 ECG

12.5.1 整数型带通滤波器

用于 QRS 检测的带通滤波器与 QRS 复波的平均功率谱吻合,因此可用来削弱 ECG 信号中的噪声。它能削弱肌电噪声、60Hz 干扰、基线漂移和 T 波干扰造成的噪声。如同 12.1 节中讨论过的,QR 主要能量的通频带近似在 5~15Hz 范围内。用这种算法构成的滤波器是一个整数型递归滤波器,其极点的配置能消除 z 平面上单位圆上的零点。低通滤波器和高通滤波器被级联在一起,以形成带通滤波器。

1. 低通滤波器

二阶低通滤波器的传递函数为:

$$H(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2} \quad (12.9)$$

此波滤器的差分方程为:

$$\begin{aligned} y(nT) = & 2y(nT - T) - y(nT - 2T) + x(nT) \\ & - 2x(nT - 6T) + x(nT - 12T) \end{aligned} \quad (12.10)$$

截止频率大约为 11Hz, 延迟为 5 个采样点(或 25ms, 当采样频率为 200 次/s 时), 增益为 36。图 12.11 所示的是实现这种滤波的 C 语言程序。为了避免饱和, 将输出除以 32, 这是离 36 最近能通过二进制位移运算实现的整数。

```
int LowPassFilter (int data)
{
    static int y1 = 0, y2 = 0, x[26], n = 12;
    int y0;

    x[n] = x[n+13] = data; y0 = (y1 << 1) - y2 + x[n] - (x[n+6] << 1) + x[n+12];
    y2 = y1;
    y1 = y0;
    y0 >>= 5;
    if(--n < 0)
        n = 12;

    return(y0);
}
```

图 12.11 实现低通滤波的 C 语言程序

这种低通滤波器的详细特性如图 12.12 所示。此滤波器具有纯线性相位响应。注意到在对应于 $0.3f/f_s$ 频率处有大于 35dB 的衰减。因为采样速率为 200 次/s，因此，此处对应频率为 60Hz。通过这种滤波后，电源线噪声将被明显地削弱，所有高频分量将衰减 25dB 以上。

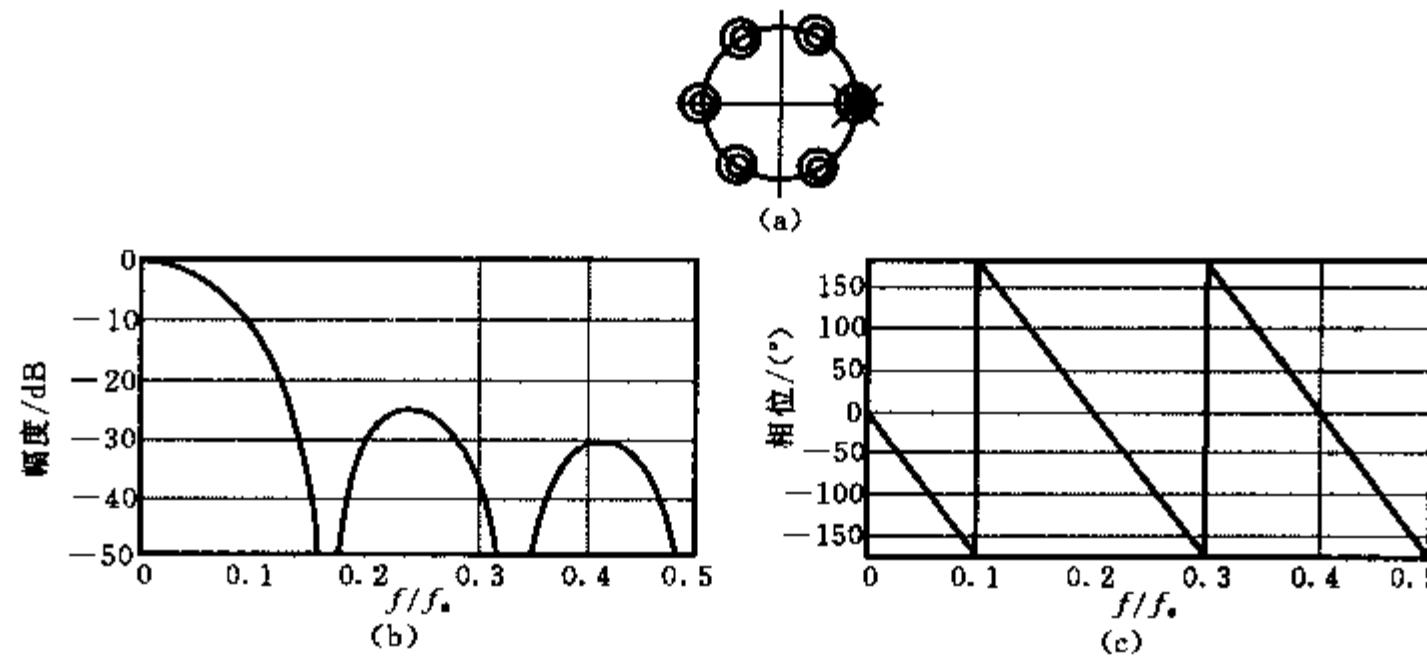


图 12.12 低通滤波器

(a)零极点分布图；(b)幅度响应；(c)相位响应

图 12.13 所示的是图 12.10 所示的 ECG 经低通滤波器处理后的结果。最显著的结果是 QRS 高频分量的减弱。任何存在的 60Hz 噪声或肌电噪声也将被明显地削弱。

2. 高通滤波器

图 12.14 所示的是将一个带延迟的全通滤波器减去一个一阶低通滤波器构成的高通滤波器。这个低通滤波器是一个整数型滤波器，其传递函数为：

$$H_{lp}(z) = \frac{Y(z)}{X(z)} = \frac{1 - z^{-32}}{1 - z^{-1}} \quad (12.11)$$

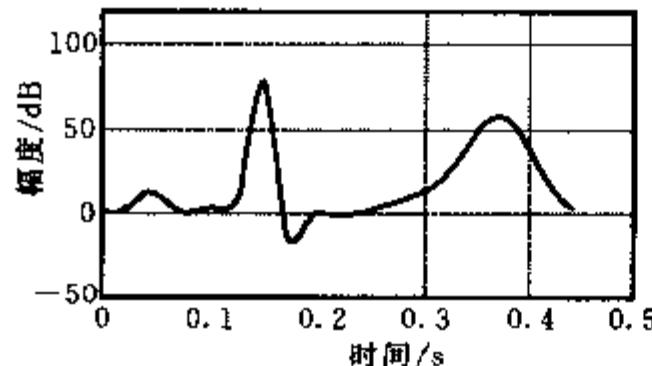


图 12.13 低通滤波后的 ECG

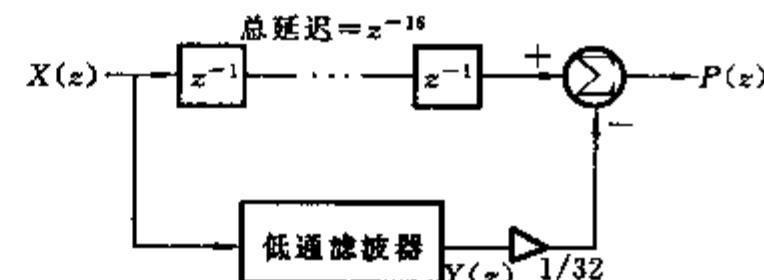


图 12.14 将带延迟的全通滤波器减去低通滤波器构成高通滤波器

其差分方程为：

$$y(nT) = y(nT - T) + x(nT) - x(nT - 32T) \quad (12.12)$$

该滤波器的直流增益为 32，延迟为 15.5 个采样点。

从原始信号中减去以直流增益的低通滤波器的输出，即可得到高通滤波。此高通滤波器的传递函数为：

$$H_{hp}(z) = \frac{P(z)}{X(z)} = z^{-16} - \frac{H_{lp}(z)}{32} \quad (12.13)$$

其差分方程为：

$$p(nT) = x(nT - 16T) - \frac{1}{32}[y(nT - T) + x(nT) - x(nT - 32T)] \quad (12.14)$$

高通滤波器的低通截止频率为 15Hz，延迟约为 $16T$ （或 80ms），增益为 1。图 12.15 所示的是实现这种高通滤波器的 C 语言程序。

```
int HighPassFilter (int data)
{
    static int y1 = 0, x[66], n = 32;
    int y0;

    x[n] = x[n+33] = data;
    y0 = y1 + x[n] - x[n+32];
    y1 = y0;
    if(--n < 0)
        n = 32;
    return(x[n + 16] - y0 >> 5);
}
```

图 12.15 实现高通滤波的 C 语言程序

这种高通滤波器的性能特征如图 12.16 所示。注意该滤波器具有纯线性的相位响应。

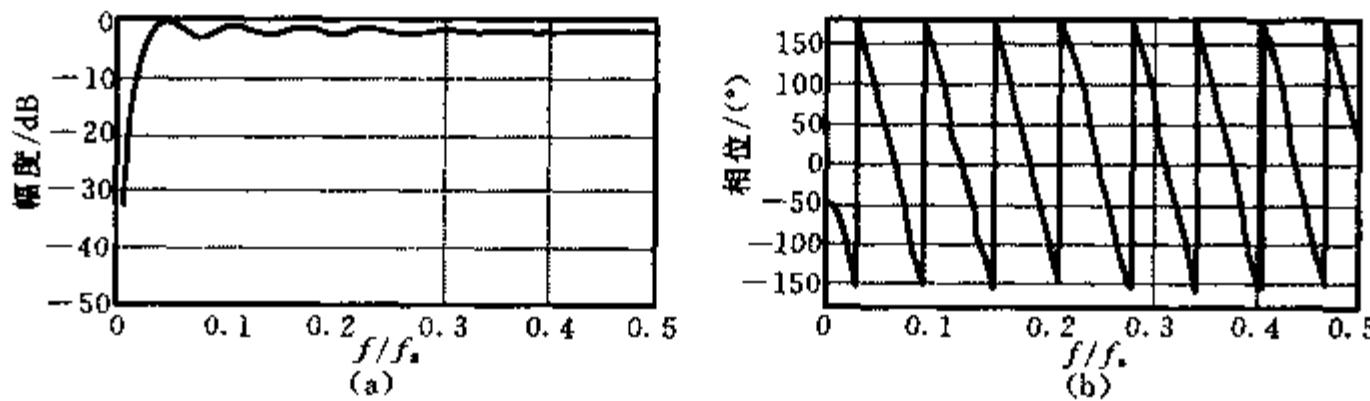


图 12.16 高通滤波器
(a)幅值响应 (b)相位响应

图 12.17 所示的是低通滤波器和高通滤波器级联构成的带通滤波器的幅频响应。其通带

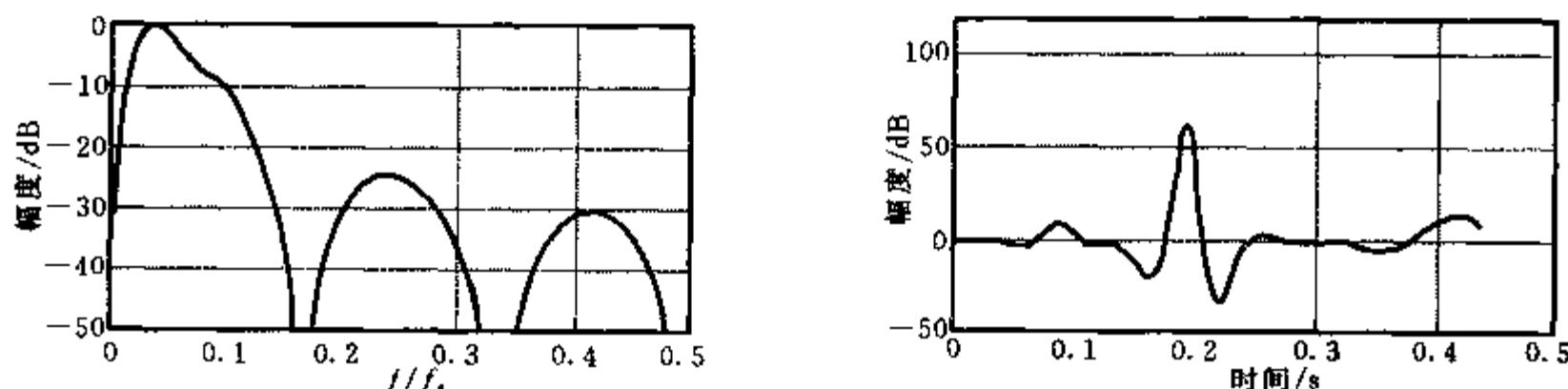


图 12.17 由低通滤波器和高通滤波器
组成的带通滤波器幅频响应

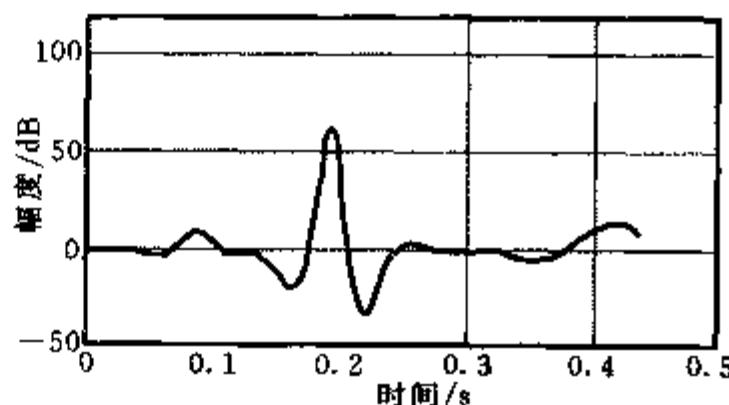


图 12.18 带通滤波后的 ECG

中心频率为 10Hz。此滤波器的幅频响应设计成接近于图 12.1 所示的平均 QRS 复波的频谱。这样，该波滤器很好地保持了平均 QRS 复波的频率特性，同时较低或较高频率的信号被削弱。

图 12.18 所示的是图 12.10 所示的 ECG 经过带通滤波后的信号。其中高通滤波器使得 T 波被削弱。

12.5.2 微分

信号滤波后, 经过微分可得到 QRS 复波的斜率信息。五点微分的传递函数为:

$$H(z) = 0.1(2 + z^{-1} - z^{-3} - 2z^{-4}) \quad (12.15)$$

差分方程为:

$$y(nT) = \frac{1}{8}[2x(nT) + x(nT-T) - x(nT-3T) - 2x(nT-4T)] \quad (12.16)$$

式中, 常数 $1/8$ 是实际增益 0.1 的近似值。在整个滤波器设计中, 把参数都近似为 2 的指数值, 这有利于实时操作。C 语言用左移或右移操作就能完成 2 的指数运算。

该微分器在直流为 30Hz 频域时近似为一理想微分器, 有一个 $2T$ (或 10ms)的滤波延迟。图 12.19 所示的是实现这个微分的 C 语言程序。

```
int Derivative (int data)
{
    int y, i;
    static int x_derv[4];
    /* y=1/8 (2x(nT) + x(nT-T) - x(nT-3T) - 2x(nT-4T)) */
    y=(data<<1)+x_derv[3]-x_derv[1]-(x_derv[0]<<1);
    y>>=3;
    for (i=0, i<3, i++)
        x_derv[i]=x_derv[i+1];
    x_derv[3]=data;
    return(y);
}
```

图 12.19 实现微分的 C 语言程序

这个微分运算器的性能特征, 如图 12.20 所示。注意幅度响应在直流约为 20Hz 时近似于一个真实微分器的响应。这是一个重要的频率范围, 因为所有的高频分量经带通滤波器后都显著地衰减了。

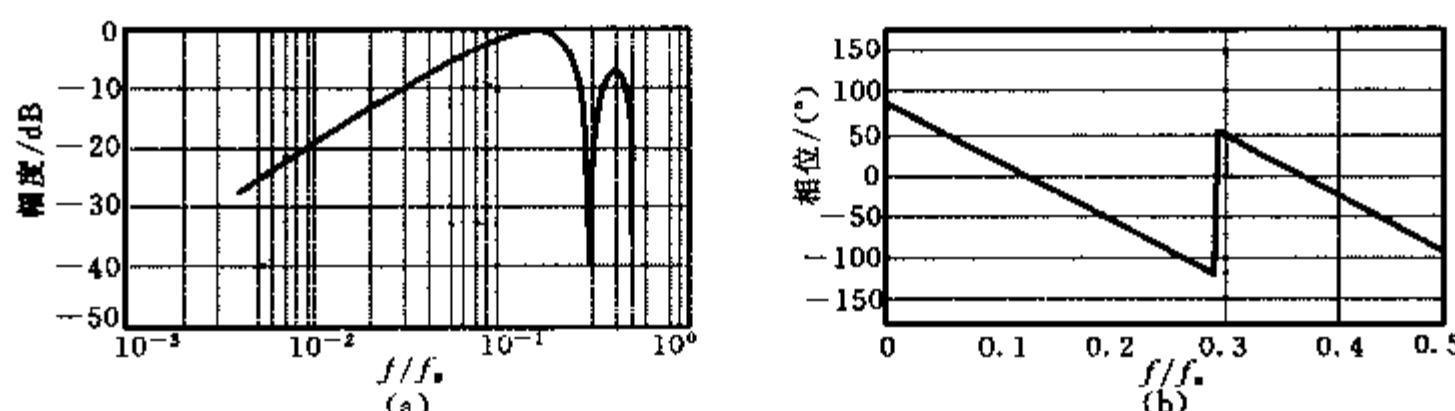


图 12.20 微分
(a)幅值响应; (b)相位响应

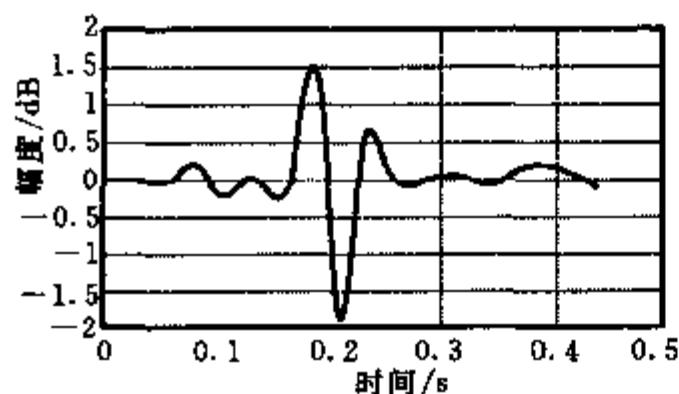


图 12.21 带通滤波和微分后的 ECG

图 12.21 所示的是信号通过带通和微分器的结果。注意,这时 P 波和 T 波有进一步的衰减,相应地 QRS 复波的峰-峰值反而进一步增强。

12.5.3 平方函数

上述的处理方法和将在下一节介绍的移动窗口积分,是 QRS 检测器的线性处理部分。信号逐点平方是一个非线性操作。完成这个操作的方程为:

$$y(nT) = [x(nT)]^2 \quad (12.17)$$

这个操作使所有处理过的信号数据点为正值,并且非线

性地放大了微分处理器的输出。它突出了信号中的高频分量,这分量主要是由 QRS 复波引起的。在该操作中要注意的是,该阶段的输出应严格限制在某一最大电平内,它对应着用来代表信号数据类型的位数。图 12.22 所示的是 ECG 采样经这种处理后的结果。

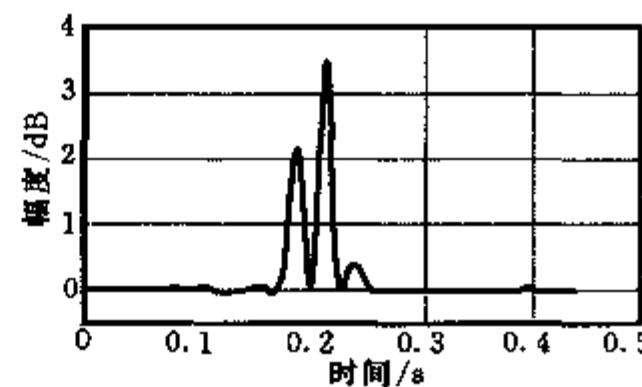


图 12.22 经平方函数后的 ECG 信号

单靠 R 波的斜率来检测 QRS 复波不是确定 QRS 事件的可靠方法。许多不正常的 QRS 复波有高幅度和长间期(斜率并不陡峭)的现象,仅利用 R 波的斜率信息可能检测不出它们,这样,为了检测出 QRS 事件,我们需要从信号中提取更多的信息。

移动窗口积分抽取了 R 波斜率之外的特征。它依靠如下差分方程来完成:

$$y(nT) = \frac{1}{N} [x(nT - (n-1)T) + x(nT - (n-2)T) + \dots + x(nT)] \quad (12.18)$$

式中, N 为移动窗口宽度中的采样点数,应当仔细选择这个参数值。

图 12.23 所示的是图 12.10 中 ECG 采样信号经移动窗口积分后的输出。图 12.24 所示的是 QRS 复波与窗口宽度间的关系。窗口宽度应当尽可能同 QRS 复波宽度相近。如果窗口尺寸太大,积分波形将把 QRS 复波和 T 波融合在一起。另一方面,如果窗口尺寸太小,一个 QRS 复波可能在输出中会产生几个峰值。窗口宽度应该通过实验来选择。当采样频率为 200 次/s 时,为这种运算选定的窗口应具有 30 多个采样点(相当于 150ms)。

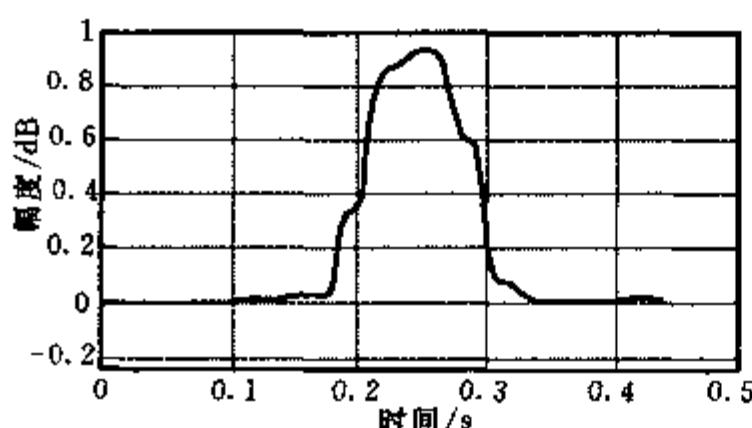


图 12.23 移动窗口积分后的信号

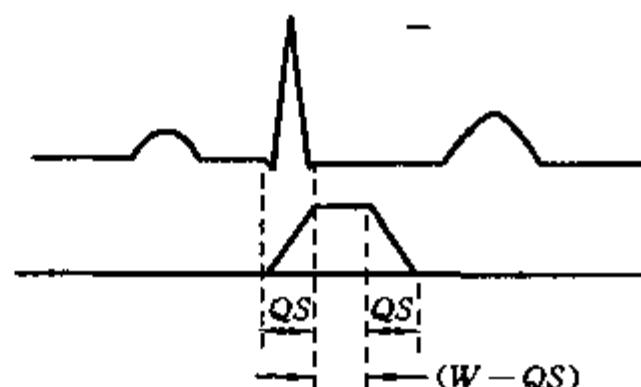


图 12.24 QRS 复波与移动积分波形之间的关系

(a)ECG 信号; (b)移动窗口积分器的输出;

QS : QRS 宽度; W : 积分窗口宽度

实现移动窗口积分的 C 语言程序如图 12.25 所示。

```
int MovingWindowIntegral (int data)
{
    static int x[32],ptr = 0;
    static long sum = 0;
    long ly;
    int y;

    if(++ptr == 32)
        ptr = 0;
    sum -= x[ptr];
    sum += data;
    x[ptr] = data;
    ly = sum >> 5;
    if(ly > 32400) /* check for register overflow */
        Y = 32400;
    else
        Y = (int) ly;
    return(y);
}
```

图 12.25 实现移动窗口积分的 C 语言程序

图 12.26 所示的是这种算法中某些滤波输出波形。这时应注意原始 ECG 信号和移动窗口积分的对应波形之间的处理延迟。

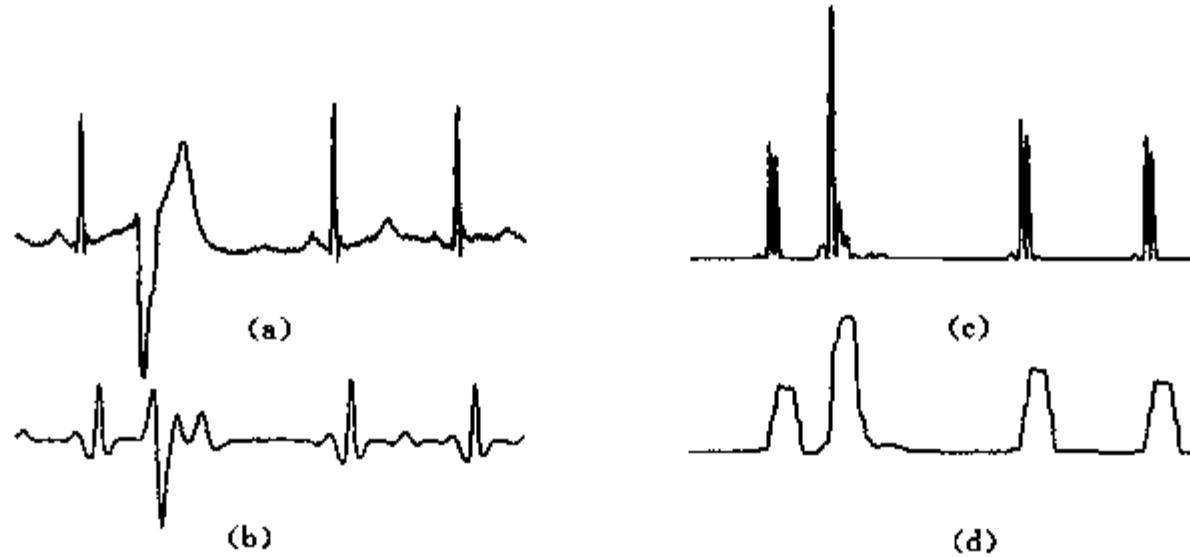


图 12.26 QRS 检测器信号

(a) 未滤波的 ECG; (b) 带通滤波器的输出;
(c) 带通、微分和平方处理后的输出; (d) 最终的移动窗口积分

12.5.5 设置阈值

Pan 和 Tompkins(1985)用于 QRS 检测算法的一组阈值是使信号峰值(即有效 QRS 复波)能被检测到的那一组阈值。信号峰值被定义为 QRS 复波的峰值,而噪声峰值为 T 波、肌肉噪声等的峰值。ECG 信号通过带通滤波阶段后,它的信噪比提高了,这就允许使用略高于噪

声峰值水平的阈值。这样，检测器的总灵敏度就提高了。

该检测算法有两组阈值被使用，每组都有两个不同的阈值。用于移动窗口积分波形的那组阈值是：

$$SPKI = 0.125PEAKI + 0.875SPKI \quad (12.19)$$

式中，PEAKI 是信号峰值。

$$NPKI = 0.125PEAKI + 0.875NPKI \quad (12.20)$$

式中，PEAKI 是噪声峰值。

$$THRESHOLDI_1 = NPKI + 0.25(SPKI - NPKI) \quad (12.21)$$

$$THRESHOLDI_2 = 0.5THRESHOLDI_1 \quad (12.22)$$

上述方程中所有变量都适用于积分波形信号，它们的含义如下：

PEAKI 是最大峰值。

SPKI 是信号峰值的实时估值。

NPKI 是噪声峰值的实时估值。

THRESHOLDI₁ 是应用的第一阈值。

THRESHOLDI₂ 是应用的第二阈值。

当一个信号在一定时间间隔内改变方向时，一个峰值就被确定了。SPKI 是之前已经确认的 QRS 复波峰值，而 NPKI 峰值是与有意义信号无关的任何峰值中的一个。从方程中可以看出，新的阈值能从以前的阈值中算出。这样，算法能适应某个特定人体的 ECG 信号的变化。

当一个新的峰值被检测出来时，它必须被归为噪声峰值或信号峰值。如果此峰值水平在信号首次分析期间就超过 THRESHOLDI₁，那么它是一个 QRS 峰。如果利用回溯查找（回检）方法（在下节解释），那么归为一个 QRS 峰的信号峰值应该超过 THRESHOLDI₂。如果用第二阈值水平来发现 QRS 复波，则峰值调整速度是通常的 2 倍。



$$SPKI = 0.25PEAKI + 0.75SPKI \quad (12.23)$$

在移动窗口积分之后，必须检测最终滤波阶段的输出峰值。一个峰值检测器算法用来寻找峰值，另一个检测器算法用来存储在这个阶段滤波后的信号自上一次峰值控制以来的最大值。一个新的峰值仅当其随后的水平低于刚才达到的峰值水平一半以下时，它才能被定义下来。图 12.27 所示的是峰值出现在下降沿的中点处（Hamilton 和 Tompkins, 1986）。

图 12.27 移动窗口积分后的输出，伴有峰值检测点

12.5.6 回检方法

回检方法保持着两个 RR 间隔的均值。一个均值 $RR\ AVERAGE_1$ ，是最近 8 次心跳的平均值。另一个均值 $RR\ AVERAGE_2$ ，是最近 8 次 RR 间隔均落入一定范围内的心跳的平均值。

$$RR\ AVERAGE_1 = 0.125(RR_{n-7} + RR_{n-6} + \dots + RR_n) \quad (12.24)$$

$$RR\ AVERAGE_2 = 0.125(RR'_{n-7} + RR'_{n-6} + \dots + RR'_n) \quad (12.25)$$

式中， RR' 是落入以下界限内的 RR 间隔。

$$RR\ LOW\ LIMIT = 92\% \times RR\ AVERAGE_2 \quad (12.26)$$

$$RR\ HIGH\ LIMIT = 116\% \times RR\ AVERAGE_2 \quad (12.27)$$

若在一间隔内 (RRMISSED LIMIT) 不能检测出 QRS 波形，则 QRS 波峰就是两个已经建立的

阈值间的峰,这两个阈值就是上节提到的应用于回溯查找期间的那两个阈值。

$$RR \text{ MISSED LIMIT} = 166\% \times RR \text{ AVERAGE}_2 \quad (12.28)$$

如果 8 个最近的 *RR* 间隔中的每一个数都处于由 *RR LOW LIMIT* 和 *RR HIGH LIMIT* 确立的极限内,则此时的心率认为是正常的。

12.5.7 性能测试

以 MIT/BIH 数据库为研究对象,测试这种算法的性能。此数据库由 48 位可活动的病人,每人 0.5h 的 ECG 记录所构成(MIT/BIH ECG Database)。这个数据库,装在 CD ROM 上,是麻省理工学院和 Beth Israel Hospital 创建的。分析约 16,000 次心跳的总失误率为 0.68%,对应于心跳的平均误差率为 33 次/h。事实上,大部分错误来自于 4 段特殊的 0.5h 带(即总数据库中的 2h 的数据。)

图 12.28 所示的是从结果中排除 4 小段问题最多的 0.5h 带的结果。注意到误检比漏检降低得更多。这种差别表明这种算法把噪声误认为 QRS 复波比丢失一个真实事件的可能性更大。除掉这 4 段 0.5h 带,心跳误差率将降低到 10 次/h 以下。另一种适用的 ECG 数据库是由

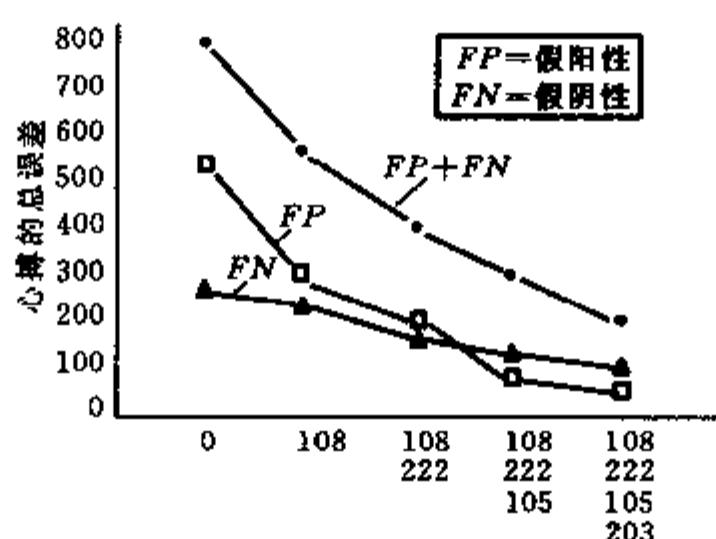


图 12.28 QRS 检测软件的性能

选择性地除去数据库中有问题的磁带能显著地降低 QRS 检测算法的总误差

美国心脏协会建立的(AHA ECG Database)。

12.6 实验:实时 ECG 处理算法

这个实验让你“窥视”QRS 检测算法的内部工作情况,此算法由 Pan 和 Tompkins(1985)创建,并在 12.5 节中已阐述。装入 UW DigiScope,选择 ad(V) ops 命令,然后检测出 QRS。

12.6.1 QRS 检测算法处理步骤

观察 QRS 检测器每一阶段的输出。描画或打印一个周期的原始 ECG 信号以及低通、带通、微分、平方和移动窗口积分各阶段的输出,注意每一阶段的滤波器延迟。

12.6.2 滤波器 Q 值对 QRS 检测的影响

为检测 QRS 复波,可构造几个中心频率为 17Hz 的两节点递归滤波器,以观察不同 Q 值对 ECG 的影响,如同 12.2.1 小节所述。*r* 为何值时将产生最佳响应?

12.6.3 ECG 整数滤波处理

使用(G)enwave 命令产生一个 ECG 信号,采样速率为 100Hz。用有如下差分方程的滤波器处理这个信号:

$$\begin{aligned}y(nT) = & 2y(nT - T) - 3y(nT - 2T) + 2y(nT - 3T) \\& - y(nT - 4T) + x(nT) - 2x(nT - 12T) + x(nT - 24T)\end{aligned}$$

观察输出并注意振荡周期。

复习思考题

- 12.1 怎样利用 QRS 检测算法来检测异位心跳?
- 12.2 怎样使用相关方法来检测异常波形中的 QRS 复波?
- 12.3 12.5 节介绍的移动窗口积分算法中,应当怎样选择窗口宽度? 选择太大或太小的窗口会造成什么影响?
- 12.4 12.5 节介绍的 QRS 检测算法中,应该怎样选择每组阈值的第一阈值,以便提高检测非规则心率的敏感度?
- 12.5 带通滤波器的 Q 值对 ECG 的 QRS 复波与噪声之比有什么影响?
- 12.6 设计一个算法,它能获得 ECG 上的基准点。
- 12.7 作为一个运行操作练习,写一段 C 语言程序,采用本章中描述的任一方法检测 ECG 中的 QRS 复合波。
- 12.8 以本章或其他相关文献中提出的某些方法为基础,给出一个能实时检测出 QRS 复波的检测算法。
- 12.9 检测平均 QRS 复波的频率特征的实验表明 QRS 复波的最大功率谱近似发生在何频率段?
- 12.10 带有差分方程, $y(nT) = [y(nT - T)]^2 + x(nT)$ 的滤波器是什么类型的最佳传统滤波器?
- 12.11 最佳 QRS 带通滤波器的中心频率并不在 QRS 复波的最大能谱处。^①此最佳滤波器使什么功能达到最大?^②用于心率计的最佳 QRS 滤波器的中心频率是多少?^③假如这个滤波器有一个恰当的中心频率,并且 $Q = 20$, 它能较好地工作吗? 如果不,为什么?
- 12.12 除了心率信息,基于一阶和二阶导数的 QRS 检测算法还能提供 QRS 的什么参数?
- 12.13 实时 QRS 检测器的微分算法有差分方程: $y(nT) = 2x(nT + x(nT - T) + x(nT - 2T) - x(nT - 3T) - 2x(nT - 4T))$ 。^①画出它的方框图。^②对于单位阶跃输入,其输出响应是什么? 画出其波形。
- 12.14 一个实时 QRS 检测器有传递函数

$$H(z) = \frac{1}{8}[-2z^{-2} - z^{-1} + z^1 + 2z^2]$$

写出此检测器中进行微分运算的幅度和相位响应的方程。

12.15 一个移动窗口积分器在一个窗口上积分,其宽度为 30 个采样点,总幅度换算因子为 $1/30$ 。当该积分器的输入为一单位脉冲,即 $\{1, 0, 0, 0, \dots\}$ 时,它的输出序列是多少?

12.16 一个移动窗口积分器的窗宽为 5 个积分点,幅度转换因子为 1。一个起搏脉冲描述为序列 $\{1, 1, 1, 1, 0, 0, 0, \dots\}$ 。这个脉冲输入给移动窗口积分器将产生怎样的输出序列。

12.17 用于实时 QRS 检测算法中的一个滤波器有传递函数

$$H(z) = \frac{(1 - z^{-6})^2}{(1 - z^{-1})^2}$$

当采样速率为 200 次/s 时,该滤波器将除去输入信号中的什么频率分量?

第十三章 ECG 分析系统

本章首先介绍用于分析和解释 12 导联 ECG 的技术,然后探讨一下用于心脏负荷测试系统的 ST 段偏移分析技术,最后总结便携式心电图心律失常监护器的软件和硬件的设计。

13.1 ECG 分析

导联心电图计算机分析技术使用一些算法来判定病人 ECG 是否正常,同时可以给出异常 ECG 的书面描述。

13.1.1 计算机 ECG 分析的历史回顾

在主机上开发和使用 ECG 分析技术始于 20 世纪 60 年代初。当时,计算机安放在计算中心进行 ECG 分析和解释。通过专门设计的可以推到病床边的 ECG 采集车,将 ECG 信号从远处的医护点传送给计算机。采集车中装有三个 ECG 放大器 FM。主体计算机运行的分析程序约有几十万条 FORTRAN 语句,可以同时采集到 3 个导联的信号,然后通过一个相互的通道 FM 调制解调器借助调整语音电话网传送这些数据。

随着技术的发展,医院的微型计算机取代了远处的主计算机。ECG 采集车开始安装嵌入式微处理器来捕捉 ECG。由于 ECG 一旦出现干扰,算法的失误率将增大,因此,微处理器通过使用数字信号处理算法,这提高了信噪比,剔除了基线漂移,抑制了工频干扰。

最后 ECG 分析程序与病床采集车结合在一起。这样,整个采集、数据处理和分析过程只需在病床边就可完成,无需再将数据传送到远处的计算机上。该技术如今已发展成独立的、采用微处理器分析的 ECG 设备。该设备可利用电池供电,并且体积小,可安装在小盒子中。

早期的 ECG 采集车装有三个 ECG 放大器,发送 2.5s 长同步采集的 3 导联信号。为了得到 12 导联的信号,将 12 导联分成 4 组,每组 3 个导联,因此,传送整个记录需要 10s 的时间。这 4 个 3 导联的信号描述了病人心脏在 4 个不同时间段的活动情况。一个 2.5s 的同期仅包括 2 个或 3 个心搏。早期的算法在判断异常,即心律失常时遇到了一些困难,因为在心律紊乱中可能会涉及好几个心搏。为了提高对 ECG 心律失常的分析能力,有学者额外记录了 3 个导联(通常向量心电图导联),记录时间为 6s,所得数据也加入到采集的数据中。

现代的基于微处理器的 ECG 分析仪器具有 8 个 ECG 放大器。因此,可同时采集和存储 8 个导联——肢导联 I、II 和胸导联电压 $V_1 \sim V_6$ 。

它们合在一起可以得到 4 个冗余的导联,即 II、 V_{aR} 、 V_{aL} 和 V_{af} (见第二章)。这些仪器具有足够存储容量,可以存储 8 个导联,采样频率为 500 次/s 的 10s 的数据。

13.1.2 12 导联 ECG 分析

进行 ECG 分析时,首先要进行特征提取。特征提取包括两部分,如图 13.1 所示。这一过程的目的是:(1)波形识别,对 ECG 所包括的 P 波、T 波和 QRS 波群进行识别。(2)测量出各波

的幅度和间期,以便提供给后面的分析程序。由于计算机不能像人类的眼睛-大脑一样直接分析 ECG 的波形,因此,必须为计算机提供一个可操作的、与波形相对应的数码集。

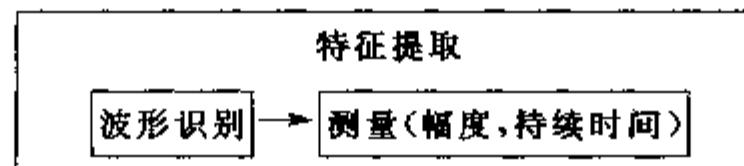


图 13.1 ECG 特征提取

波形识别的步骤是,第一步用 QRS 检测算法检出所有的心搏。第二步将各个导联中类似的波形作时间对准并求平均,由此得到 12 个导联中各导联的均波或中值波,通过分析这 12 个平均波,可以识别出其他的波形或 ECG 的其他特征,最后形成一个测试集,并汇集成一个矩阵,这些测试信息将在后续处理中得到分析。

图 13.2 所示的是一个普通男性病人的 ECG,该 ECG 是借助于一个 ECG 分析仪器获得

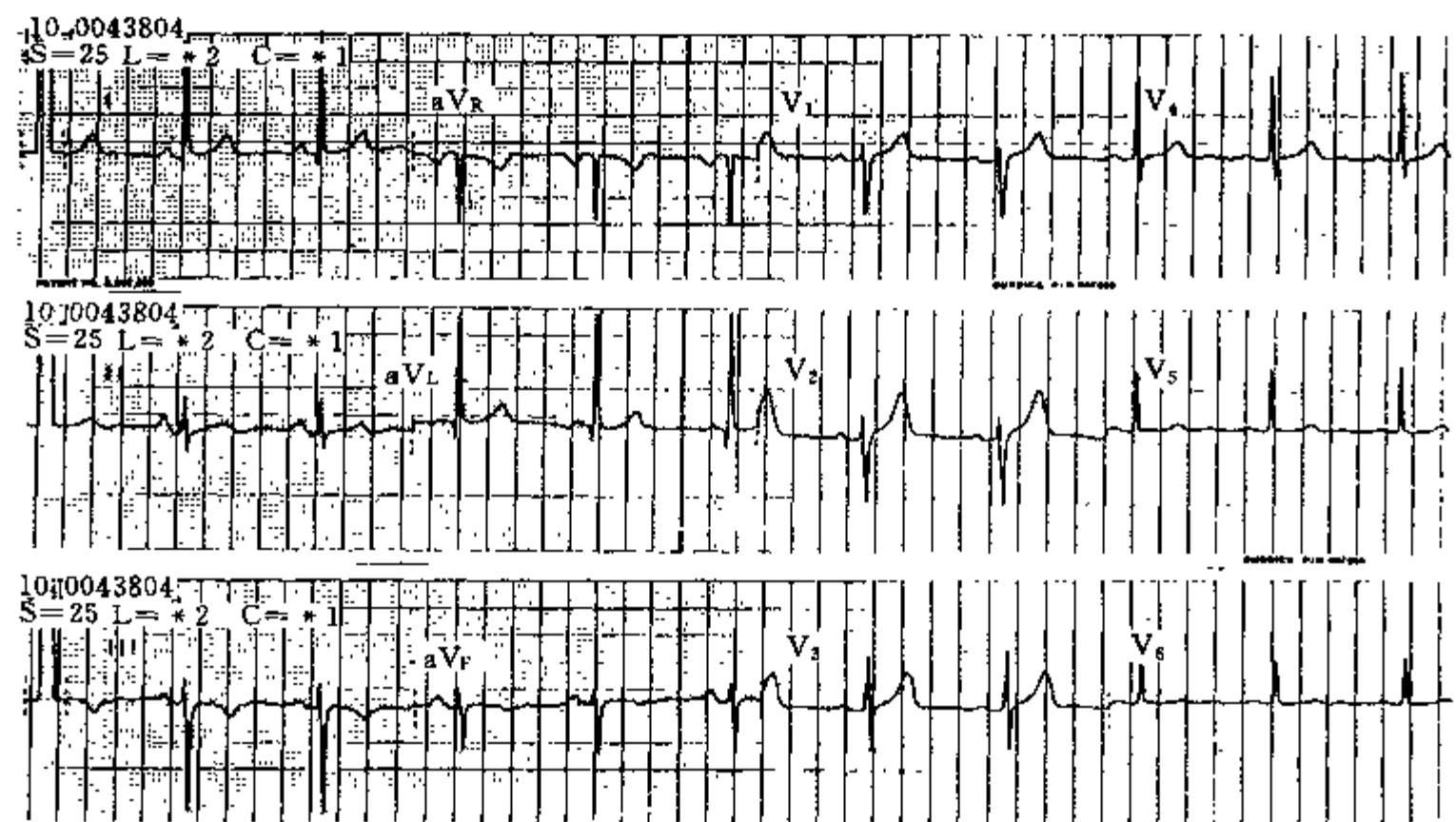


图 13.2 一个普通男性病人的 12 导联 ECG 图

左边的标准脉冲指定为 1mV。记录速度为 mm/s,每一个小的分格是 1mm,因而大的分格是 5mm。在导联 I 中, R 波幅度大约为 11mV,心肌间隔 1s(即心率为 60 次/min)。这些记录是从 Elite ECG 分析仪 (Siemens Burdick, Inc) 上测得的

的。尽管在仪器中存储了 8s 的数据,但只有 2.5s 的数据可以打印在报告单上。图 13.3 所示的为 ECG 仪器在这一病人身上得到的内部测量矩阵。R 波幅值使用 μ V 为单位,间期使用 ms 为单位。例如,在导联 1 中,R 波幅值(RA)为 1140μ V 或 11.4mV ,R 波宽度(持续时间)(RD)为 71ms 。

利用计算机分析 ECG 有两种基本方法。第一种方法是决策逻辑法,该方法通过计算机程序用基于规则的专家系统来模仿专家的决策过程。第二种方法将 ECG 分析看成一个模式识别的分类问题,用一套随机多变量模式识别方法来解决这些问题。

由 Elite ECG 分析仪 (Siemens Burdick, Inc) 得到的测量矩阵,其波幅用 μ V 为单位,间期用 ms 为单位。

图 13.4 所示的是整个 ECG 分析方法。特征提取过程产生一个称为测度矩阵的数据集。这

	I	II	III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6
PA	70	127	77	10	50	102	90	90	80	87	80	70
PPA	-10		-15	-90	-25		-25	-5		-33		
EOD	5			106	10							
OD	17				88	17						
OA	80			655	125							
RD	71	49	18		71	18	23	26	49	52	88	88
RA	1140	340	170		1055	130	250	430	1040	1560	1130	820
SD		39	70				9	62	57	31	32	
SA		110	970				60	1020	1190	760	390	
RPD							3					
RPA							30					
SPD							58					
SPA							430					
ORSA	681	105	576	-399	623	-241	783	-685	311	681	662	531
STJ	35	7	-28	-23	30	-10	42	70	55	12	10	20
STM	40	20	-20	-30	30		110	190	130	50	20	20
STE	50	30	-20	-40	35	-5	180	320	230	90	30	20
TA	230	120	-130	-170	175	-25	480	860	700	330	150	110
TPA												
MTA	180	90	-110	-140	130	-30	300	540	470	240	120	90
MXFG	1	1	1	1	1	1	1	1	1	1	1	1
	I	II	III	aVR	aVL	aVF	V1	V2	V3	V4	V5	V6

图 13.3 Elite ECG 分析仪

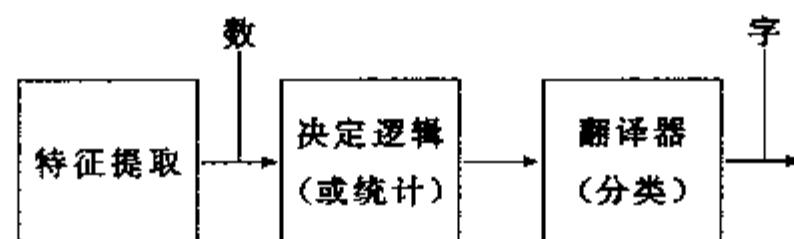


图 13.4 ECG 分析步骤

些数集成为输入量,又进入决策逻辑或统计过程。决策逻辑过程驱动确定和描述病人病情的分析过程。

决策逻辑方法用一组规则集进行推理,这组规则集由 ECG 测量矩阵推导而得。这些规则以一系列的逻辑 IF—THEN 语句形式出现在计算机程序中。

这些规则集是根据专家的知识来开发的。通过一组 IF—THEN 语句,最后成为一个或多个分析语句,并打印在最终的报告上。但是众所周知,一组心脏病专家对同一 ECG 报告的理解,相同点不超过 80%。事实上,如果一个心脏病专家在不同时间内观察相同的 ECG,那么医生对 ECG 的理解与他(她)前一次的相比,相同率也不超过 80%。因此,一个决策逻辑程序只能由参与开发 ECG 分析的医生或医生小组来制定。

决策逻辑法的一大优点是其结果和决策过程很容易被技术人员理解和操作。但其决策规则是直接从专家那儿得到而不是从数据中得到的,像这样一个系统,永远不能超过专家分类器。与专家系统不同,基于规则的系统不能直接利用波形。同时,使用该方法很难对一条线或几条指令进行局部更改。因此,为了适应某组特殊病情的患者,对其中某一条规则作轻微变动就可能导致对许多条逻辑语句都要做必要的修改。

用多元统计模式识别方法分析 ECG 时,每个决策都直接利用数据获得,因此,该方法不受人的影响。该方法是由一组病人数据出现的概率先确定一个统计范围,再在这个确定范围的

ECG 的测量矩阵中,根据数据的概率来得出决策的。该方法完全取决于数据而不是人们的知识经验。因此,利用该方法开发的分析系统比最好的医生所具有诊断水平还要高。

然而与决策逻辑方法不同,在多元统计模式识别方法中,由于不存在逻辑表达式,因此,无法向专家们说明算法是如何得到最终分析结果的。正是这种原因,使得该方法无法在商业设备中采用。

在临床应用中,医生可浏览并修改由计算机所给出的 ECG 分析报告。如果连续地分析相似的波形,计算机软件就会反复地出现相同的诊断错误。虽然 ECG 分析系统乐意从其错误中“学习”,但目前还没有开发出一种能提高计算机分析其错误性能的商用软件系统。

图 13.5 所示的是一个 ECG 分析仪提供给临床医生的最终摘要,该摘要是针对图 13.2 所示的 ECG 得出的,分析仪将该病人的 ECG 分为“正常(Normal)”和异常“窦性节律”。

ID	0043804	SINUS RHYTHM
50 Year Old	MALE	NORMAL ECG
Med		
16, 53 04/11/91	Loc	
Vent. Rate		61
PR interval		180
QRS duration		92
QT/QTc		404/407
P-R-T axes		52-20-11
Limb × 2	Chest, × 1	
25 mm/s		

图 13.5 由 Elite ECG 分析仪(Siemens Buraick, Inc)
得到的分析摘要

13.2 ST 段分析

ST 段为 ECG 的一个区段,它出现在去极化(QRS 复波)之后与复极(T 波)开始之间。ECG 的 ST 段出现变化,表示供应到心脏的血液出现不足。因此,对 ST 段进行测量是十分重要的。这一节将介绍带有微机的仪器是如何对 ST 段进行分析的。

图 13.6 所示的是一个标有几个特征点的 ECG 波。对 ST 段进行分析时,首先从检测 QRS 波形开始。许多有效的技术都可以完成对 QRS 波形的检测。通过寻找与 QRS 检测特征点之前和之后 60ms 相当的间期,识别 R 波峰,这个点具有最大值。Q 波是到达 R 波前的第一个回折点。该回折点可通过斜率、零斜率标记的心变或斜率出现显著的变化进行识别。斜率可根据三点差分法来计算。如,ECG 信号出现杂波时,在计算斜率之前,需使用一个低通数字滤波器滤波,以使得波形变得光滑。

必须首先确定 ECG 的基准线。在 P 波和 Q 波之间斜率接近零的 30ms 间期的线就是基准线。确定 QRS 宽度时,S 点定位与 R 波的定位方法相同。测量 QRS 宽度、R 波峰幅值时均以该基线为基线,这样就获得了 RR 间期。

J 点是 S 点后的第一个回折点,或者在某些 ECG 波形中,J 点本身就是 S 点。要得到 T 波(也可称为 T 点)的起始点,首先确定以基线为基准的 T 波波峰,该波峰为 $J + 80\text{ms}$ 和 $R + 400\text{ms}$ 之间的最大绝对值处。然后,通过寻找一个在 T 波的左侧的 35ms 间期,来寻找 T 波的起始点,即 T 点。在 ECG 中,T 点是最难识别的特征点之一。如果该点没有检测到,可设定该

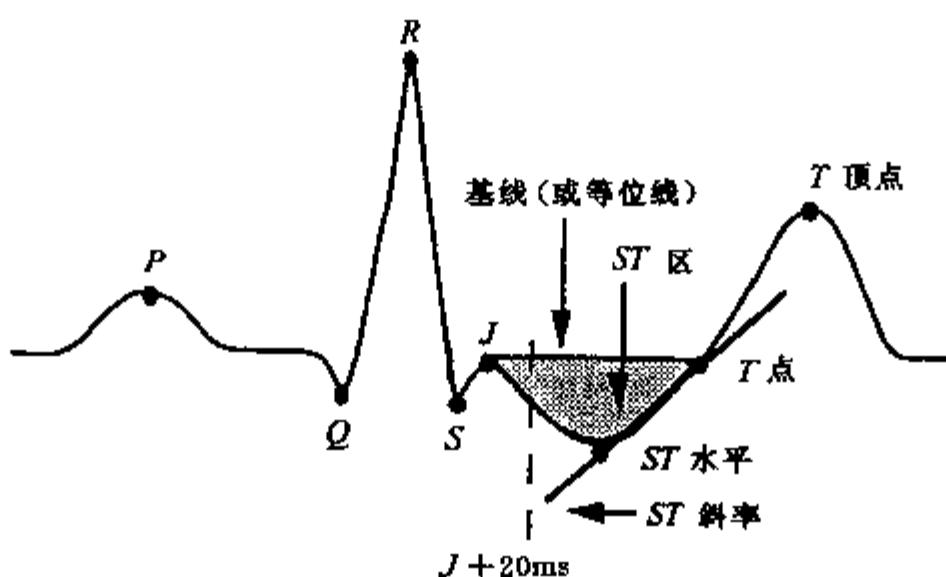


图 13.6 ST 段分析仪测量的 ECG

图中给出了 ECG 的相关点, J 、 T 、ST 偏移等; ST 偏移的搜索窗口由 $J+20\text{ms}$ 和 T 点所定义

点在 $J+120\text{ms}$ 处。

对 ECG 的各特征点进行识别后, 可使用窗口搜索法测量 ST 段。由 $J+20\text{ms}$ 和 T 这两个点作为窗口的边界, 在窗口中找出最大下降或上升点。ST 段偏移可用相对于基线的绝对值的变化值来表示。

除了 ST 段偏移外, 还应计算其他几个参数。由 ST 段起点与 T 点的幅值差除以相应的时间间隔, 便得到 ST 斜率。ST 面积是根据在 J 点和 T 点之间每个点减去基线值后, 所有点的信号瞬时值的总和来计算的。ST 指标即是 ST 段的水平值和与 ST 斜率的 $1/10$ 之和。

13.3 便携式心律失常监护仪

近年来, 出于经济的考虑, 人们对在家中监护病人表示出极大的兴趣。如果病人在家中同在医院中一样能获得相同的诊断信息, 那么在家中进行监测更经济。随着技术的发展, 出现了一些具有高分析计算能力的装备, 如袖珍式、可折叠的个人计算机。这种可使用电池供电的系统在家中或其他地方都可进行计算工作, 取代了以前使用的大型、非便携式、使用市电供电的计算机。

在外形体积缩小和耗电量减少的同时, 计算机的计算能力增强, 使得设计一种小型的、可携带在病人身上的智能生物医学仪器成为可能。

尽管该仪器可以实用, 但效果没有原先预测的那么好, 原因是用于该仪器的微型计算机技术还没有发展成熟(Sahakian 等人, 1978; Tompkins 等人, 1980; Weishes 等人, 1982)。不过, 便携式监护器最终将具有以前仅在医院内拥有的监护设备的全部功能。

13.3.1 Holter 磁带记录器

Holter 磁带记录器是目前用于诊断一个活动病人是否患有潜在的心脏病的仪器。设计便携式心律失常监护仪的初衷是取代它的作用。Holter 磁带记录器最佳替代设备是一个以微型处理机为核心的、便携式心律失常监护器, 该监护器使用的程序算法与当今医院用于心脏病护理设备的检测系统的算法相似。

图 13.7 所示的是 Holter 磁带记录器的工作过程。Holter 磁带记录器可全天候地将病人

的 ECG 记录在磁带上。将记录结果送回到实验室，经过回放和分析，筛选出可疑的心律失常并及时报告给医生。医生可以很方便地掌握患者的病情，而不必用上几天时间查看患者所有的心电波形。

称 Holter 监测是不确切的，因为它的“监测”不是普通意义的监测。一般的监测，可以连续地、及时地向医生提供病人的状况，如：在进行特别护理时进行的监测就是一般监测。

利用 Holter 磁带记录器来记录活动病人的 ECG，所采用的技术是比较落后的、有限的。虽然对小型磁带记录器进行了一些改进，但因其是电动机械设备，限制了其改进的潜力。现在很多先进的技术正在应用于 Holter 磁带记录器回放分析系统中。但现在仍需要大量地进行手工操作来对捕捉的信息进行筛选。由于这是一种高智能型的工作，在未来的一段时间内仍将使用该方法。

13.3.2 便携式心律失常监护仪的硬件设计

图 13.8 所示的智能便携式心律失常监护仪可以捕捉到可疑的心电波，并且立即通过电话网络将异常的心电片段传送至中心接收站。

这套仪器将为心脏病学家建立一个新的诊断方式，他们可以比现今更快、更准确地对病人作出诊断并制定治疗方案。除此之外，临床医生可以跟踪监察治疗效果并及时对其作出修改，这也是磁带记录器所不能比的。

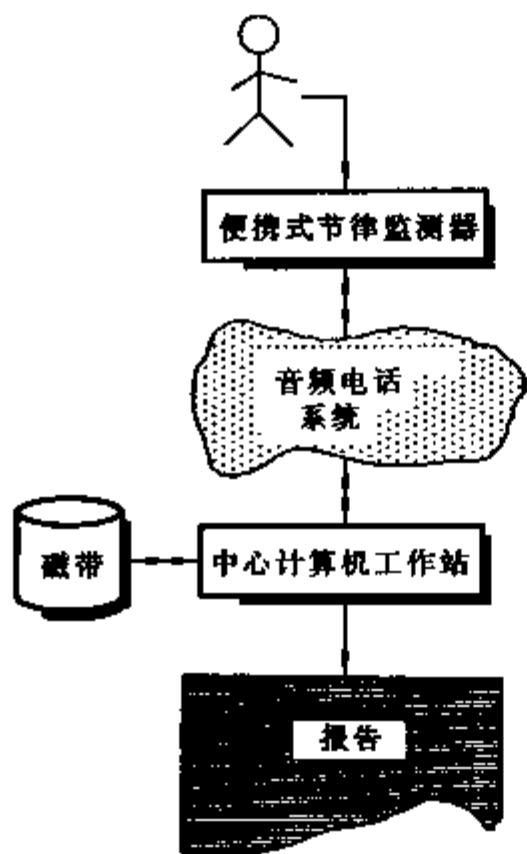


图 13.8 便携式心律失常监护仪
这套基于微处理器的仪器能实时分析 ECG，并通过声级电话线将捕捉到的数据传到主计算机

这种智能便携式心律失常监护仪的硬件设计非常简单明了，它所采用的是市场上就可买到的大型集成电路零件，且用电池供电。用电池作为电源的硬件设计一般首选的半导体技术为 CMOS。

硬件技术迅猛的更新换代是与半导体技术的飞速发展分不开的。例如，最初的便携式心律失常监护仪是在 1977 年设计的，它是基于 COSMAC 微处理器 (RCACDP1802) 而研制的，按如今的标准来看，只是一种信息处理单元。现代的商业装置一般采用 80C86 微处理器 (Intel)，这是一种 CMOS 集成电路块，IBM-PC 机采用的就是这种技术。

图 13.9 所示的是一种便携式监护仪的方框示意图。除了微处理器外，便携式心律失常监护仪还需配有 ADC。模拟放大器在前端将心电信号放大。模/数转换集成电路将模拟心电图转换为微处理器所能接受的数字信号。

ROM 存储器存有整套仪器的实现各功能的程序。RAM 则将捕捉到的 ECG 信号存储起来。输入/输出接口

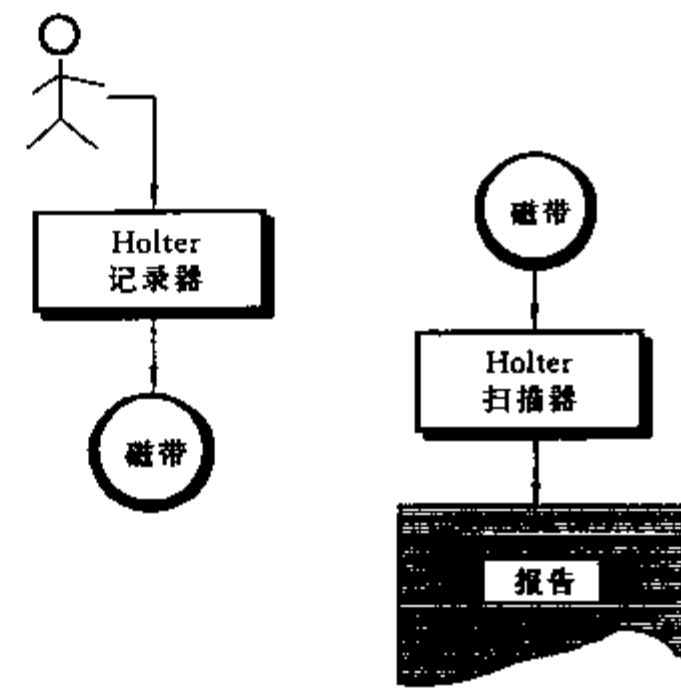


图 13.7 Holter 记录器的工作过程
当代 Holter 记录器可全天候地将两通道的 ECG 记录在磁带上

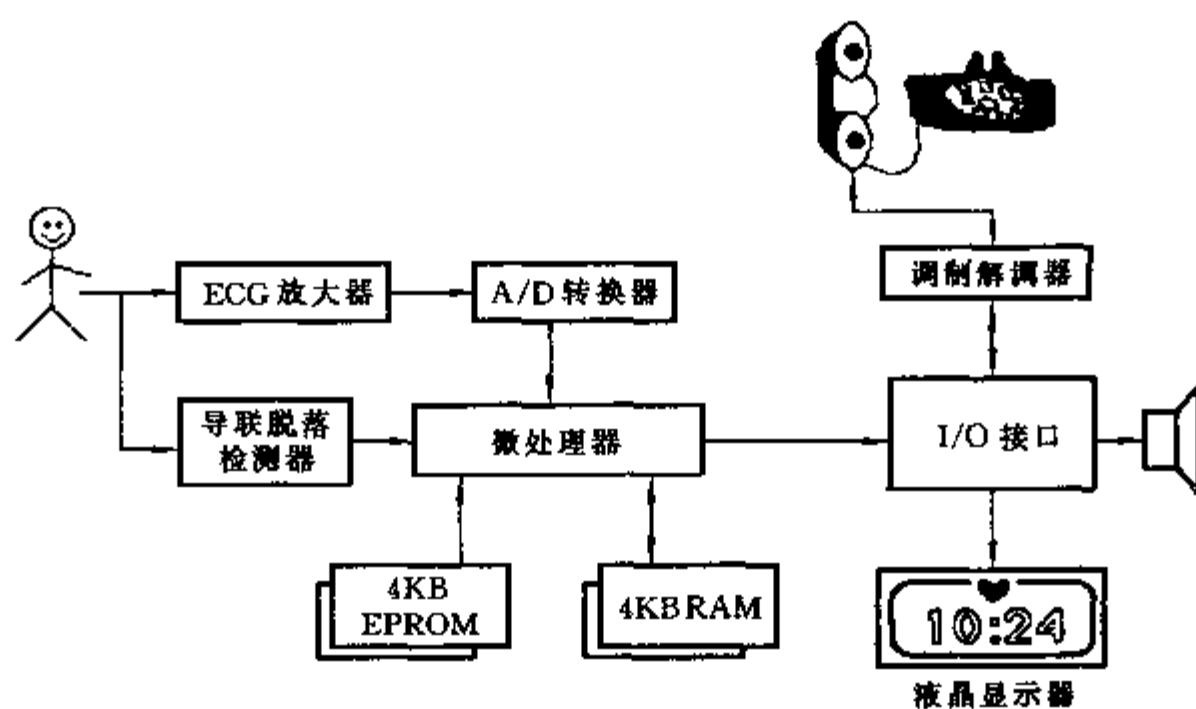


图 13.9 便携式心律失常监护仪的方框图

(I/O)将视频显示以及仪器的交互开关连接起来。调制解调器可使其与在另一处的计算机相接。这样就可将捕捉到的 ECG 信号送回到信息处理中心(Thakor 等人,1982)。

由于这些技术都在日新月异地改进中,因此,必须不断地考虑如何改进提高硬件设计水平。更新的设计可以使整套装置,(1)部件减少;(2)更加可靠;(3)尺寸减小及重量减轻;(4)耗电量小;(5)计算能力增强(从而可以实现更加复杂的信号处理及分析算法)。这样就必须不断地提高硬件设计水平,才能达到以上目的。

13.3.3 便携式心律失常监护仪的软件设计

硬件技术的不断更新使得软件也需要重新设计,以跟上其前进的步伐。一台仪器的软件是离不开硬件的。每一种新的微处理器都有其独特的机器语言。这样,需为不同的微处理器重新编制同样功能的程序,在这方面要浪费大量的程序设计时间。软件设计上的这一问题促使人们着手开发更高级的程序语言,使其可以从一种类型的微处理器上移植到另外一种机型上。

我们最终选择了 C 语言,通过在 Bell 实验室的试验,我们相信,它是适用于这种实时记录仪的最理想的程序语言。其带来的主要益处有:(1)用高级语言,就足以实现精确的机器控制,(2)可以从一种类型的微处理器移植到另一种机型上。倘若遵循一些软件设计规则,那么用 C 语言编制的这种类型的微处理器的实时程序只要稍加处理一下就可使用在另一种类型的处理器上。尽管 C 语言还不是很完善,但在很大程度上它还是减少了更换微处理器时重新编制软件所需的程序设计时间。这样就可以把程序设计的时间集中在如何改进算法上。

设计一套便携式心律失常监护仪软件的关键所在有两个方面:(1)QRS 复波检测,(2)心律失常分析(Abeustein,1978;Mueller,1978)。QRS 探测必须非常准确,否则会多报或漏报心脏搏动的次数,从而造成心律失常分析算法的错误。另外一个重要的(对于记忆容量受限的情况)设计问题则是数据压缩算法问题。

1. QRS 检测算法

QRS 检波器所采用的各种技术包括:线性数字滤波器、非线性变换、判定过程及模板匹配,一个 QRS 检测软件常采用其中的两种或更多的技术。

现今市场上所卖的 ECG 分析仪最常采用的方法是以模板匹配为基础的方法。常规 QRS

复波的模式称为模板,它是由软件通过对某个病人心电图进行学习而得到的。运用算术准则将此模板与随后出现的实时 ECG 相比较,尽可能地找出与之相匹配的波形。与模板极其相近的波形代表检测到一个 QRS 复波,如果出现的是一个不匹配的波形,但同时又可能是异常的 QRS 复波的波形,则它就可列为一个特定的波,以后出现的可疑 QRS 复波都将与之相比较。由于这种技术要求有相当数量的存储器来保存模板(这取决于使用者所用的数量,一个系统至少得有 40 个),以及有很强的计算能力来使模板与实时信号相比较,因而这里没有在检测处理过程中运用这一技术,而是运用一种完全基于数字滤波器的算法来代替这种技术。12.5 节所介绍的即为这套装置运用 QRS 检测算法的小结。

2. 心律失常分析

QRS 宽度及 RR 间期都是由 QRS 检测器所确定的。根据这两个参数就可将 ECG 信号进行分类。图 13.10 所示的为以这两个参数为基准的心律失常分析算法的说明(Ahlstrom 和 Tompkins, 1983)。在这个以两个参数为基准的映射图中,算法通过对一套由临床医生确定的 8 个 QRS 复波进行识别来建立一个称为“正常”的区域。这些 QRS 复波包括特定病人的“正常”心率与形态。这一识别过程在两组 2 维映射空间内,建立一个初始的“正常”区域的中心。除了“0”区域之外,对映射图中其他区域的边界均计算出距离常规区域中心位置(定位)的百分比。“0”区域的固定边界是根据生理指标(极限)而定的。所有被映射到“0”区域的指示点都被认为是噪声,因为它不在通常所希望的最小的 RR 间期或 QRS 宽度的生理限度之内。

心动异常诸如心动过速造成的一连串心脏搏动将落在 RR 间期非常短的“1”区内,而心动过缓则落在“6”区内。特别要注意的是,心动异常的划分归类必须考虑到心脏搏动的顺序。例如,伴有全代偿性暂停的心室性早搏的特点就是一个短的 RR 间期伴有一长的 QRS 宽度。随后再出现的一个长的 RR 间期伴有一正常 QRS 宽度。这一现象反映在图上就是两个有序的指示点,第一个点在“3”区,第二个点在“5”区。这样,心律失常分析还包括了心脏搏动落在映射空间的方式的分析。

正常区的中心要根据 8 个最近的正常心跳的 RR 间期的平均数来适时修正。这种方法可以使正常区在两参数空间内随着训练及其他生理变化引起的心率的正常改变而移动。其他区的界限则根据心跳的次数一次一次地修正,这是因为它们都是以正常区为基准而划分的,因此这种算法适合于心率正常改变的情况。

波形可以通过标记连续的心搏所在的区域和顺序进行归类。图 13.11 所示的是一些检测不同心律失常的算法。上述方法用于从 ECG 信号中抽取 RR 间期及 QRS 宽度是很有效的。根据所获得的信息,可以区分不同类型的心律失常。心律失常判断规则如下:

正常:心脏搏动位于正常区域。

停搏:无大于 1.72s 的 R 波,心率低于 35 次/s。

漏搏:长 RR 间期,心搏落在 6 区;R 波落在 T 波上,心搏落在 2 区。

代偿心室性早搏:心搏在 3 区,随后一次心搏落在 5 区。

无代偿心室性早搏:3 区出现一次心搏,随后一次出现在常规区。

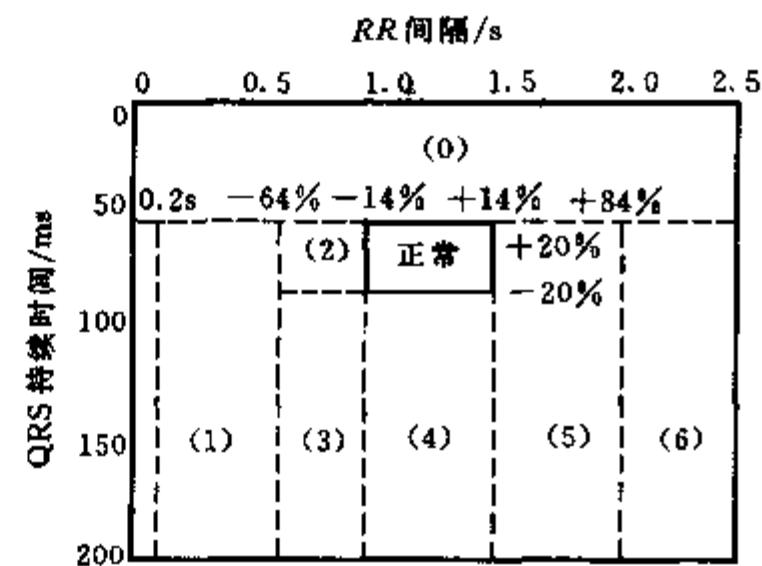


图 13.10 基于 RR 间隔和 QRS 间期映射的二维空间的心律失常分析算法

Normal:	If a beat falls in the normal box.
Asystole:	No R wave for more than 1.72 s; less than 35 beats/min.
Dropped:	A long RR interval; beat falls in Region 6.
R-on-T:	A beat falls in Region 2.
Compensated PVC:	A beat in Region 3, followed by another in Region 5.
Uncompensated PVC:	A beat in Region 3, followed by another in the normal region.
Couplet:	Two consecutive beats in Region 3 followed by a beat in the normal region, or in Region 5.
Paroxysmal Bradycardia:	If there are at least three consecutive points in Region 5.
Tachycardia:	Average RR interval is less than 120 beats/min.
Fusion:	A beat with a wide QRS duration; falls in Region 4.
Escape:	A beat with a delayed QRS complex; falls in Region 5.
Rejected:	A beat that has an RR interval of 200 ms or less, or QRS duration of 60 ms or less.

图 13.11 基于 ECG 信号中 QRS 间期和 RR 间对心律失常的分类

早搏成对：3 区出现两次连续的心搏，随后一次心搏在常规区或 5 区。

阵发性心动过缓：至少有 3 次连续心搏落在 5 区。

心动过速：平均 RR 间期小于 120 次/min。

融合：心搏出现在 5 区，伴有宽大的 QRS 复波。

逸搏：心搏出现在 5 区，伴有延迟 QRS 复波。

折返：心搏 RR 间期小于或等于 20ms，QRS 宽度小于或等于 60ms。

13.3.4 便携式心律失常监护仪的前景

现代高技术替代过时的 Holter 技术的困难之处在于，前者本身存在一些问题。也就是说，虽然医生从不需察看全面完整的 24h 的 ECG 信号记录，而是只看最后的报告，但显而易见的是技术人员在作 ECG 时已对其作了分析。同样，医生对最后的报告数据也是很放心的，因为他们并不需对此负很大的责任，这样一来，医生们大多不愿再从头到尾地将所得报告数据重审几遍。

第二个原因是这种以微处理机为基础、实时记录的现代记录的诊断算法还不是很完善。因而对 Holter 市场的冲击还不是很大。便携式监护仪必须能够完成现今使用的冠心病护理中心所做的大部分工作。它还必须可靠地检测 QRS 复波。也许还应比现在医院的医疗系统还准确些，这是由于错误的判断将造成无用的数据，而且也会占据本来就不足的存储器。如果出现异常情况的话，这种便携式监护仪还必须能进行自我诊断，并为病人提出如何应付这种情况的建议。

在医生们都纷纷急于接受这一仪器之前，必须在临床试验中证实该装置至少已能捕捉到重要的临床信息，并且所耗费用也应比现行的 Holter 记录法低廉。要做这种临床试验的证明确定是非常困难的，因为并没有什么绝对好的标准来衡量这种仪器的性能。MTT/BIH 和 AHA ECG 仅能作为起码的评定标准。

我们这套现行的仪器并不是要对病人的 ECG 作出解释分析。它的设计目的在于把某个病人的被认为是正常的 ECG 中的可疑波分离出来。然后通过电话，它将这些可疑波的数据传送给某个临床医生以求诊断。事实上它仅是一种屏幕检测显示装置。研究的目的首先是为了简化 ICU 病房的护理工作。

在现今的医疗机械市场中还没有一种便携式心律失常监护仪（有时也称为实时 Holter 监

护仪)取得过成功。其中部分原因是实时 QRS 检测及心律失常分析算法还不够完善,不能完全满足于该装置的有效运用。但是这些算法的进步性还是相当肯定的。这是因为现在临床监护系统正越来越多地采用微处理机。

Holter 记录器将来无疑会被这种以微处理机技术为基础的便携式监护仪所替代。届时它将降低诊断费用,提高诊断可靠性,提供更好的临床兼容性,并且不断改进其性能。这些装置的运用必将随着科学技术的进步而进步。

复习思考题

13.1 在新型便携式心律失常监护仪中,分析心律失常所使用的映射图是根据哪两个变量产生二维空间的?

13.2 目前在 UW 软件中开发的实时 QRS 检测算法可在标准的 24h 的数据库中检测百分之几的 QRS 复波?

13.3 在心律失常分析中,每次心搏的 RR 间期和 QRS 宽度都被映射到一个二维空间中。图 13.10 中的“正常”区的中心位置是如何确定的?

13.4 下面哪些是用 UW 软件研制的便携式心律失常监护仪的最佳描述?

①分布式处理方法;②选择出重要的信号,然后将其存储在磁带上,再通过电话将录音传回到中央计算机;③将 RR 间期和 QRS 宽度存储起来;④运用 ST 段分析作出心律失常分析算法的一部分;⑤在其存储器中保存 30 段 16s ECG 片段;⑥通过电话传送信息,使用的是一个安装在“T”形盒中的调制解调器;⑦现在使用的是 CMOS8088 微处理机,但将来还会有改进;⑧总是将警报前的 ECG 片段存储起来;⑨设计的目的在于取代 Holter 记录器;⑩利用遥测装置将其捕捉的数据通过新的医院卫星网络传送至中央计算机;⑪使用内装加速度计来监测病人的活动能力;⑫一共有 250KB RAM 存储 ECG 信号;⑬在心律失常分析中,利用 2 个从 ECG 中抽取的典型;⑭由于该装置的 QRS 检测近于最佳优化,因此,最快将在明年投入商业化生产;⑮保存 24h 2 通道 ECG;⑯将报警段的 ECG 片断存储起来;⑰分析 12 导联 ECG。

13.5 下面哪些最恰当地描述了惠普公司研制的便携式心律失常监护仪?

①CMOS280 是一种微处理机;②信号是通过电话传送到中央个人微型机中的;③ECG 采样波形存储于 RAM 中;④分析了 12 导联 ECG。

13.6 请描述一下市场上最常用的高性能心律失常监护仪所采用的 QRS 检测技术,诸如集中护理单元等。

第十四章 VLSI 在数字信号处理中的应用

本章重点讨论硬件在前面介绍的数字技术中的应用。首先讨论数字信号处理器(Digital Signal Processors,简称 DSP)及其必须实现的功能,然后介绍两种市上可购到的商品化 DSP 芯片。目前具有高性能的用于信号处理的超大规模集成电路(VLSI)的应用包括:并行处理、位串处理、收缩阵列和滤面阵列等。一种便携式 ECG 和数字式助听器是医学仪器上使用 VLSI 的例子。本章最后简要介绍超大规模集成电路在生物医学传感器中的应用及其相应的集成化技术。

14.1 数字信号处理器

在 25 年以前大多数信号处理都是采用专门的模拟信号处理方法来实现的。当数字系统开始实用化并且有了各种数字信号处理的算法以后,信号的数字处理便得到了越来越广泛的应用。早期的信号处理硬件是在一些通用的微处理器,例如 Intel 8088 上进行的。随着信号处理的要求越来越高,这种通用微处理器就显得太慢而不适于对信号做实时的分析,因此,出现了一些专门设计的信号处理芯片。

14.1.1 处理器的要求与组成

数字信号处理器实际上是一种专用的微处理器。典型的微处理器一般都需运行较长的软件,如操作系统,它们通常不能用于实时的计算。对于数字信号处理器,它不需具有太多的功能但必须以非常高的速度运行。数字信号处理器必须具有一些专门的计算能力,其具体的技术指标如下面所述。这些基本的要求包括:转换频域、平均技术以及各种滤波器技术。为了完成这些操作,一个典型的数字信号处理器应具有如下组成部分:

- (1) 控制处理器;
- (2) 算术处理器;
- (3) 数据存储器;
- (4) 定时控制;
- (5) 系统。

早期的数字处理系统的组成结构如图 14.1 所示,它包括许多芯片和集成电路,现在这样的一个系统只需一个单片的 VLSI 芯片就能实现。

14.1.2 单片数字信号处理器

1970 年以后的一些早期的设计是试图把图 14.1 所示的结构集成一块单片的数字信号处理器。到 1980 年一些公司如贝尔实验室、德克萨斯仪器公司、NEC 和 Intel 都先后开发出单片的数字信号处理器。最近几年推出的利用大规模集成电路技术所生产的数字信号处理芯片具有较高的性能价格比。

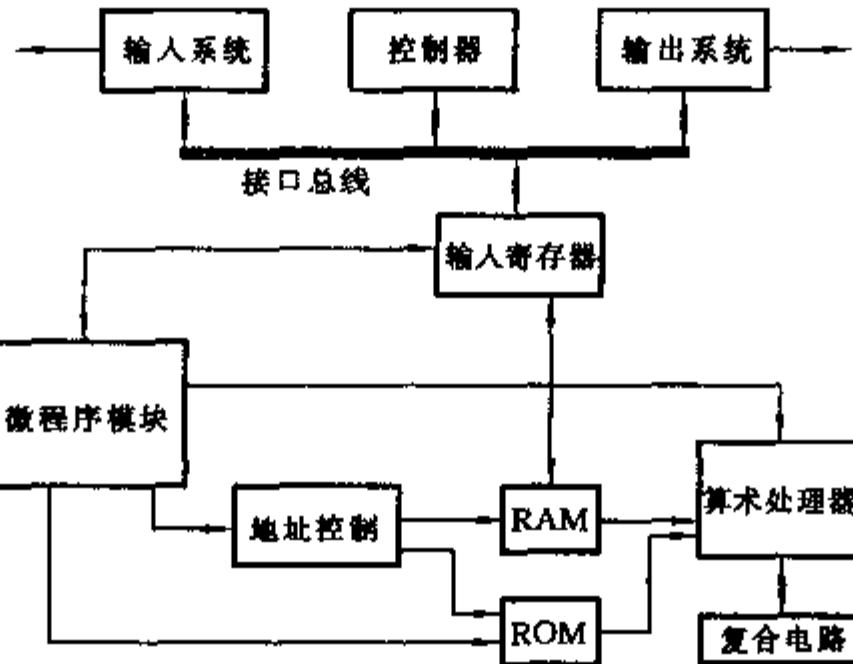


图 14.1 一个典型的数字信号处理器模块设计

比较各种 DSP 芯片的性能并不是一件容易的事。通常用于比较微处理器速度的指标是其每秒运行多少百万指令数(MIPS)或每秒运行多少百万浮点数(MFLOPS)。但这个指标并不适合于 DSP 芯片。一般用于比较 DSP 芯片性能的指标是其进行乘法和累加(MAC)所花的时间。 MAC 时间反映了含有乘法和累加运算指令的最大速率。这个指标对于计算 FFT 或者数字滤波器尤其有意义。然而这种比较会形成一个陷阱,因为这个指标并不能总是解释得很完整,因而它不能满足实际的需要。图 14.2 所示为一些常用的 DSP 芯片的 MAC 时间。

公司	部分	日期	指令运行时间	乘法位数
AT&T	DSP1	1979	800	16
德克萨斯仪器	TMS32010	1982	390	16
Fujitsu	MB8764	1983	100	16
NEC	μ PD77220	1986	100	24
摩托罗拉	DSP56001	1987	74	24
AT&T	DSP16A	1988	33	16
德克萨斯仪器	TMS320C30	1988	60	24
摩托罗拉	DSP96001	1989	75	32

图 14.2 比较一些常见的 DSP 芯片的 MAC 时间

下一节将介绍两种 DSP 芯片,首先介绍的是 TMS320 系列,这是由德克萨斯仪器公司生产的,它是目前使用得最普遍的 DSP 芯片。第二种要介绍的是由摩托罗拉公司生产的 DSP56001,可以粗略地估计一下它的速度,摩托罗拉的通用微处理器芯片每秒可运行 270,000 次乘法,而 DSP56001 每秒可运行 10,000,000 次乘法,可见专用的 DSP 芯片运算速度是通用微处理器速度的 37 倍。

1. TMS320

TMS320 是德克萨斯仪器公司 1982 年生产的微处理器系列,它是专为进行实时数字信号处理的应用而设计的。TMS320 的主要特点是,采用了大量的片内阵列乘法器,而在一般的通用微处理器中乘法运算是由指令系统的微代码实现的。微处理器所具有的多功能性是以花费更长的 MAC 时间为代价的,DSP 芯片采用的硬件乘法器能大大地减少 MAC 时间。在典型的微处理器中大约有 10% 的芯片空间用于进行运算,但在 DSP 芯片,如 TMS320 中,有大约 35% 的空间

来进行运算,这种大的运算区域是进行数字信号处理算法所必须的。这种采用 16/32 位算术和逻辑运算单元的芯片可在 200ns 之内完成两个 16×16 位的乘法运算。这种高速的乘法运算能力是进行快速 FFT 变换的基本条件,程序采用汇编语言,最大时钟速率是 20MHz。

2. DSP56001

DSP56001 是 1987 年推出的,被广泛地用于音响设备、科学仪器以及其他应用。该芯片具有每秒运行 1 千万次乘法,1 千万次加法,2 千万数据传输以及 1 千万次循环运算能力。要取得这些高速的运算能力,其芯片要有高度的并行结构,具有阵列乘法的硬件结构,两个算术逻辑单元,两个独立的片内存储空间,以及片内的程序存储空间。更重要的是 DSP56001 采用一种能进行多独立单元同时运算的并行流水线技术,使得不同的独立单元可同时运作。摩托罗拉的 DSP 芯片具有独特的能在一个周期内完成一个 MAC 的能力,该芯片的主要性能有:512B 的 RAM、24 位数据通道能提供 144dB 的动态范围、一个数据 ALU、地址算术单元以及以并行方式运行的程序控制器、74ns 的 MAC 时间。

14.2 高性能的大规模集成电路信号处理器

真正实现实时数字信号处理只是近年的事,这是因为数字信号处理需要高速的计算能力,尤其是对于实时方面的应用。这种对 DSP 芯片实时性的需求持续增长,导致了开发出诸如 TMS320 和 DSP56001 那样的大规模集成电路数字信号处理器。这种 DSP 芯片的发展趋势是将其算法硬件化,换句话说,这种采用 VLSI 设计的方法将导致设计出一些越来越专门化的硬件结构。如前所述,数字信号处理的应用包括有 FFT 计算、FIR 和 IIR 数字滤波器。这些技术仅需要执行三种类型操作,即存储、乘法和加法。在这些例子中,要将那些有限的不同运算与 VLSI 技术有机地结合成一整体,这就导致为数字信号处理应用而设计出专用的 VLSI 结构。这些结构的主要特征是,使用多处理机制和并行处理、阵列处理器、精减指令集(RISC)、计算以及采用流水线的方式来获得非常高的处理速率。

目前还没有一个专门的术语能用来描述和对这种专用的 VLSI 结构进行分类。目前已提出了一些方法来建立一种有用的分类法,如 Flynn 所提出的基于指令和数据流的 Flynn'S 术语。下面将要讨论的是试图用一些最常用的术语来定义这种结构的方法。然而对于同一结构而用不同术语来解释的方法,从字面上看似乎有些难以理解。

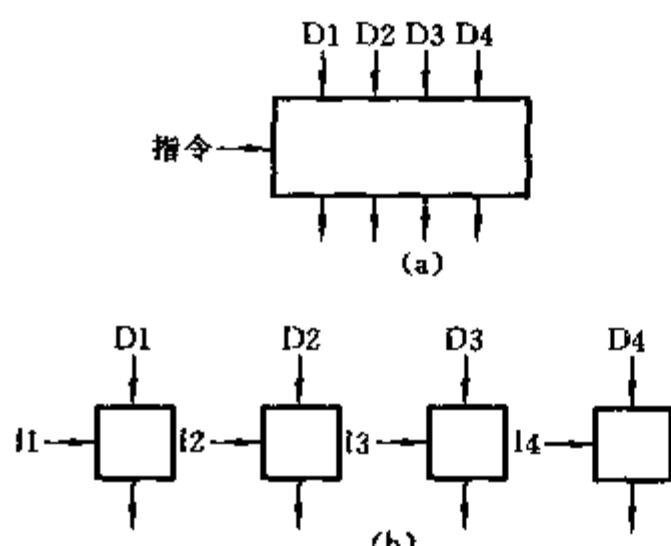


图 14.3 并行处理器结构的分类

- (a) 单指令多数据(SIMD);
- (b) 多指令多数据(MIMD)

• 172 •

并行处理或多任务处理是通过使用多个处理器来实现这种同时进行处理的方法。流水线处理方式是实现多重处理器的最佳方法,尤其在进行大量运算时更显示出其独特优点。典型阵列处理器可解释为一个二维的阵列处理器,在处理器中数据流是通过流水线方式来处理的,因此,这是一种非常有效的工作方式。

14.2.1 并行处理

并行处理技术应用于信号处理一般可分为两种类型,如图 14.3 所示。

(1) 单指令多数据(SIMD)操作,由单运算指令来

操作多数据元素构成的计算域。

(2) 多指令多数据(MIMD)操作,由一定数量的指令流来操作多个数据元素。这种方法一般用于图像处理方面。

并行处理结构已被用于数学方面的余数数据处理系统(RNS)。在该系统中计算域被分解为若干个独立的子域。在这些独立子域中的计算是由那些类似于 SIMD 的并行结构完成的。RNS 提供了一种高速的数学运算,因为在这里进行的加和减不需进位和借位,而且乘法也不需要分步求积。

14.2.2 位串处理

在位串处理中,每次只操作一个字中的 1 位。这种位串组织通常应用于对多个字的处理。因此,“位片”或“位列”的概念便产生了。图 14.4 所示的是简单的位片草图。

在位串结构中,数字信号按位在单条数据线上顺序传递,而不是在并行总线上同时传递。这种方法与并行方法相比有几个优点:(1)在 VLSI 芯片间通信更有效,这对于信号处理中涉及的通信操作是很重要的。(2)位串结构可以在位级上转变成有效的管道结构,从而加快运算速度。

另外,这种位串方案还有利于设计 VLSI 芯片。由于不需使用并行接口,这种位串芯片更易于连成网络,同时这种结构还可大大减少所需的芯片输入输出引脚。最后,在芯片中这路以位流水方式在存储器和处理元素的分布是以模块化和正规方式进行的。这种方案极大地方便了设计、制造和应用这种硅编译器。

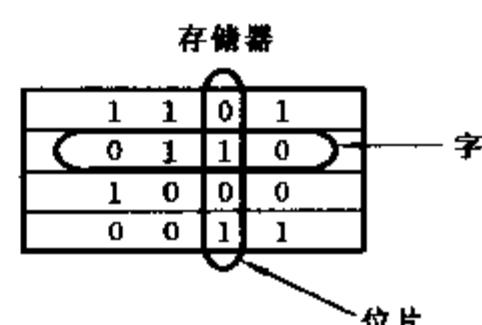


图 14.4 简单的位片组织
当进行数据的位片操作时,多个字的同一位便同时操作了

14.2.3 收缩阵列

收缩阵列(systolic arrays)是以位序结构并应用流水线原理而组成的一种阵列结构。这种阵列能以位的、字节或二者兼有的运行方式进行操作。收缩阵列这个名字是借用血液循环中的泵而得来的。在收缩操作中数据系数的其他信息整个地进入系统,通过一些附加处理后再作为结果“泵出”。这种多处理器结构通过采用流水线数据处理方式可获得高度并行化处理的效果,这种处理方式的典型应用是在一种二维结构的应用。当数据进入阵列后,它可不用存储就直接传到任何一个处理器。图 14.5 所示的是这种数据流,这个数据流与整体时钟同步并显示时间延迟。

收缩阵列结构特别适用于 VLSI 的制造。为了能充分发挥 VLSI 芯片高度集成化的优点,芯片的设计必须简单、规则和模块化。芯片中收缩阵列使用简单的处理单元,相互连接的相邻元素间沿着一或二维折转。事实上,大多数的连接仅与其最相邻的部分进行通信。这种收缩阵列的结构如图 14.6 所示。芯片就像一个格子,格子中的每一个点代替处理器,每一条线连接相邻点。

14.2.4 波阵面阵列

波阵面阵列是一种最新的结构。这种波阵面结构由 Kung 所开发,它类似于收缩阵列,其特征包括模块化处理器的规则的局部内部连接网络。二者的不同之处在于,波阵面阵列的整体时钟和时间延迟被异步的握手信号所替代。这种结构消除了时滞、容限误差和峰值功耗带来的

问题。

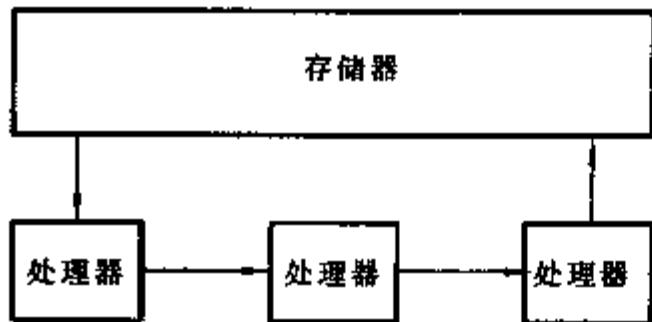


图 14.5 数据的收缩流进入和流出存储器

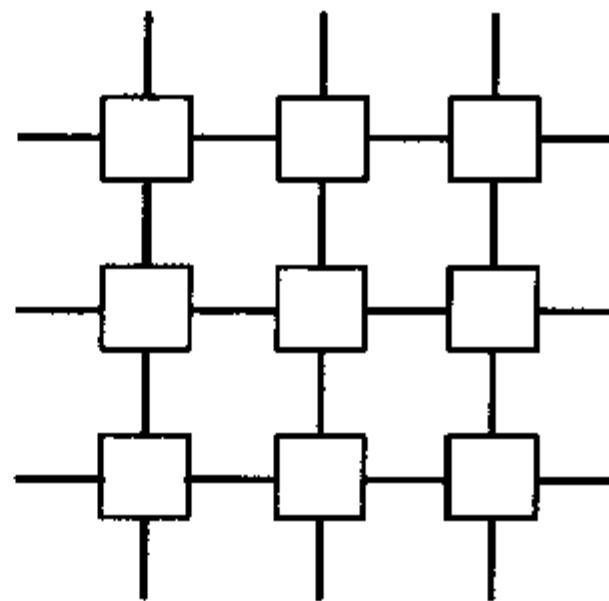


图 14.6 一个简单的收缩矢量结构

每一个盒子代表一个处理器或晶体管，
每根线代表两个处理器间的连接

有关这种结构的详细讨论已超出了本书的范围，在这里只能讲述有关应用VLSI技术于数字信号处理的一些概况和应用。

14.3 VLSI 在医学中的应用

VLSI 目前已广泛地用于医疗产品中，其产品包括，核磁共振成像系统普及型心电图机、Holter 监护仪和血流分析仪等。

14.3.1 便携式 ECG 机

便携式的 ECG 机能够实用化是因为采用了 VLSI 设计。当今的便携式的 ECG 机，如 Elite (Siemens Burdeck 公司)，在性能上与全功能的 12 导联 ECG 设备相当，而全功能心电图机体积大，纸的耗用和存储空间都较大。将它与采用 VLSI 技术生产的便携式 ECG 相比，后者比前者体积小 $\frac{1}{15}$ ，重量轻 $\frac{1}{10}$ ，其成本只有前者的一半，而且便携式 ECG 要节约 90% 电源和少用 60% 的元件。

14.3.2 数字式助听器

一个传统的模拟式助听器是由并行连接的带通滤波器组成的。为了获得较高的补偿精度需要较多的滤波器，但是助听器的尺寸、结构和成本都限制了使用大量的滤波器，另外要使用大量的可调整元件，使得其滤波特性要适合于任一个用户也是十分困难的。

随着通用的 DSP 芯片和专用集成电路(ASIC)的发展，在助听器中使用数字技术已成为现实。这种数字式助听器的设计包括在前面讨论过的 A/D 转换、D/A 转换和数字滤波等。采用数字设计技术的最大优点是，其滤波特性可变，能最大限度地补偿每一个听力丧失的人的听力。这种补偿是采用软件来实现的，因此，具有极好的可变性，能满足每一种特殊需求，而且这种数字助听器可按自动化规格生产和安装。

14.4 用于生物医学信号的传感器

最近从标准的微电子学技术中已派生出固态传感器和灵敏器这一新的分支。集成电路处理技术已被用于制造各类敏感元件。目前新的进展是,开发出新一代智能的或 VLSI 传感器,这种传感器将敏感元件和信号处理电路集成在单个的基底上。图 14.7 所示的为一般的采用 VLSI 的传感器的原理框图。

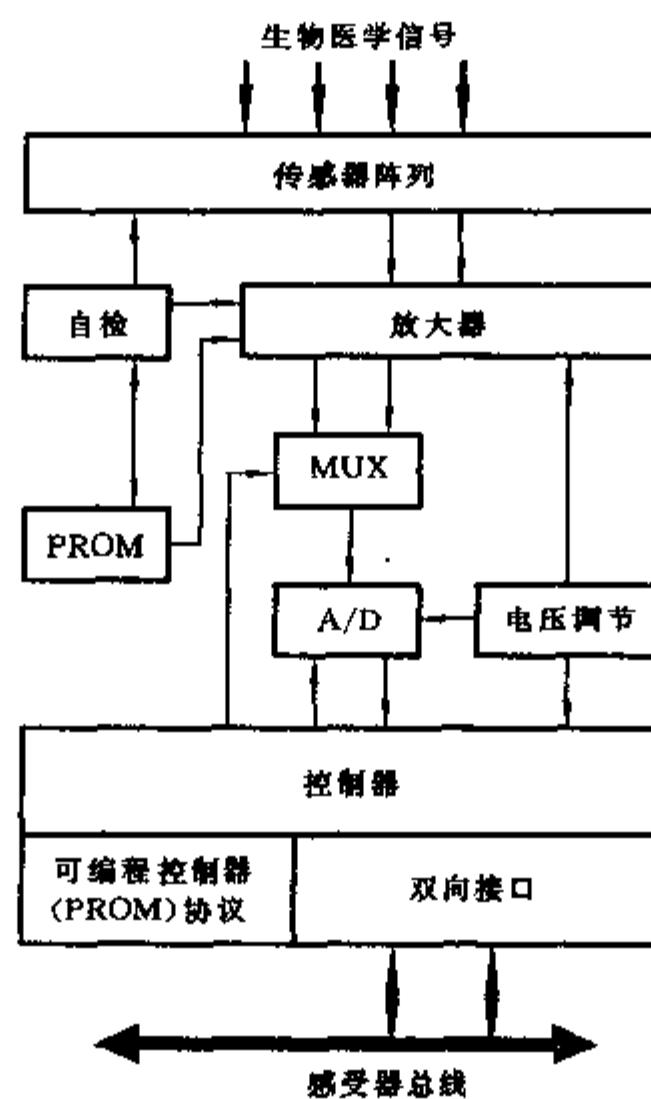


图 14.7 VLSE 传感器的原理框图

这种传感器是可寻址的,具有自检测和提供标准数字输出功能

图 14.8 所示的为传感器的整体结构,其电路包括一些功能电路,如放大器、模拟多路转换器、移位寄存器和时钟等,把这些功能集成在传感器中,这些功能已在第三章讨论过。该传感器长 3~4mm,宽 200 μm ,厚 30 μm 。

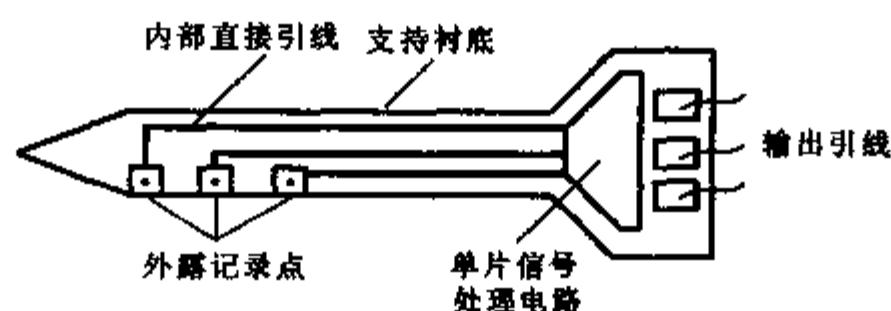


图 14.8 由 Najafi 和 Wise 组装的灵敏传感器的整体结构

14.5 VLSI 开发工具

在科研环境中使用的大多数 VLSI 开发工具都是 Berkeley VLSE 工具。该系统允许设计者在器件水平做出设计并仿真。一个典型的设计是以功能结构框图开始的，然后逐步地向实际器件设计。这种器件水平使用的是另一种称为 MAGIC 的 VLSI 设计软件。MAGIC 是一个模拟的 CAD 设计程序。设计者要根据设计图画出每一个将要制造出的晶体管元件。显然这将是一个艰巨的工作，因为通常一个典型芯片都将有成百上千个晶体管。MAGIC 有许多性能可简化设计和去掉部分多余的设计。当一部分设计完成之后，程序，例如 CRYSTAL 和 ESIM，可用于测试这部分电路的逻辑和时序状态，这点是非常重要的，因为它不必等到整个设计完成之后进行测试，而可以在部分电路设计完后就进行测试。然后在各部分设计完成并通过性能测试之后，再将这些部分连接起来，最终构成完整的设计。

一旦这种设计完成并通过性能仿真测试，就可以开始制造了。MOSIS(MOS 结构系统)的制造工具在南加州大学是作为一种无需太多经验的工作，最后器件的测试和修正是必须的一道工序。

14.6 设计方案的选择

当开始设计一种仪器时，首先必须确定方案：是采用现有的芯片，或 ASIC——应用专门集成电路，还是采用上述几种方案的某种组合设计。在 10 年之前采用已有的芯片设计方案是唯一的选择。但现在，即使一些很小的公司也能够使用前述的开发工具来设计自己的 ASIC 或者专用集成电路。总之设计方案的选择应根据特定项目的要求以及可得到的已有芯片两方面来综合考虑。其设计时间或成本是与设计时采用已有芯片设计或完全重新设计相关的。

复习思考题

- 14.1 叙述通用微处理器与 DSP 之间的不同之处。
- 14.2 为什么要用 MAC 时间而不用通常的 MIPS 或 MFLOPS 作为评价 DSP 的指标？
- 14.3 为什么 VLSI 非常适合设计 DSP？
- 14.4 解释下述术语：①并行处理；②流水线；③阵列处理；④SIMD；⑤MIMD。
- 14.5 解释收缩阵列与波阵面阵列之间的不同，其中哪一个更有效。
- 14.6 列出至少 4 个优点来说明 VLSI 在医学仪器设计中的应用。

附录 A UW DigiScope 软件的硬件配置

本附录提供为建立各种 UW 支持的硬件系统所需的步骤。UW DigiScope 软件可以同三种不同设备连接：(1)IBM PC 机或其兼容机内部的实时设备 ADA2100 模数 I/O 接口卡，(2)通过串行口 RS232 与一个外设单板机通信(摩托罗拉 S68HC11EVBU 学生设计箱与 MCM68HC11EVB 计算箱)，(3)虚拟 I/O 设备(数据文件)。当你用 INSTAL.EXE 安装 UW 时，你必须指示在你的系统中可以用哪些物理设备。这种设置在你将来添加一台转换设备时将被更改。

图 A.1 所示的是三种不同设备与它们彼此之间的兼容性。利用基于 80286 的 IBM PC/AT 兼容机，可以达到大约 500 次/s 的采样速率。利用基于工作在 4.77MHz 较慢的 8088 的 IBM PC 机或其兼容机，最大采样速率可达 250 次/s。于是，根据不同的设备，采样能力将有所不同。

项 目	内设(ADA 2100)	外设(摩托罗拉 EVBU)	虚拟 I/O 设备
采取频率/次/s	1 至 500	31 至 500	1 至 500
模拟输入范围/V	-5 至 +5	0 至 +5	N/A
模拟输入数	8	4	20
模拟输出数	2	无	无
心输出范围/V	-10 至 +10	N/A	N/A
数字输入数	4	8	无
数字输出数	4	8	无

图 A.1 三种设备比较

在你的系统中安装三种设备的重要步骤是，运行 INSTALL 程序。INSTALL 程序为数字示波器建立所有的软件。如果要在系统中建立实时设备 ADA2100，那么你必须在 AUTOEXEC.BAT 中加入 DACINIT、COM、DACINIT、用 COM 重设的 ADA2100 定时器，以防止定时器产生中断。ADA2100 的记录解释了为什么要这样做。

在下面的三部分中，给出如何准备 PC 机系统，以及利用 UW DigiScope 软件进行数据采集的具体步骤。首先进行覆盖配置，安装 ADA2100，然后给出将摩托罗拉 EVBU 和 EVB 与一个使用 RS232 通信口的 PC 机相连的步骤。最后，为建立一个虚拟数字采集的好的系统，给出一些建议。在讨论硬件设置前，必须知道怎样组织工作区才能使它正确、安全地工作。请阅读下面框中的信息：

注意,准备工作区:

安全第一,拔下所有你所用工作设备的电源。在未拔下插头前,千万不要打开机壳,这将有助于保护你和你的设备。

为用 ADA2100 或摩托罗拉 EVBU 安装和配置你的系统,需要你移动包含精密集成电路(ICs)的电路板。你必须十分小心地操作,防止损伤设备,首先需要有一个干净的工作区。防静电罩或袖口等工具将有助于消除静电。如果你没有这些工具,那么你必须在接触任何 ICs 或电路板时使自己接地。这将有利于防止静电在你体内积累。

ADA2100 和 EVBU 上的许多 ICs 是 CMOS 器件,即使是很少的静电荷也会永久地损坏 CMOS 部分。损坏部分看不出什么特征,但不能进行正常工作。

A. 1 在 IBM PC 机中安装实时设备 ADA2100

安装一个实时设备 ADA2100 接口板到 IBM PC 机中,有三个步骤,每一步都必须十分小心。第一步是配置 ADA2100;第二步是在 IBM PC 机总线上进行物理安装;第三步是建立 CONFIG.WDS 文件,这将由 INSTALL.EXE 程序自动完成,并可运行 ADINSTAL.EXE 加以更改。

A. 1. 1 配置 ADA2100

UW DigiScope 软件给出了一些关于 ADA2100 配置的假设。图 A. 2 所示的是设备跳线和开关设置的情况。还必须检查你系统中别的已装入的内部卡,以判断 ADA2100 是否同你系统中别的设备相冲突。ADA2100 的设备是十分灵活的,因此不会产生什么问题。

跳线/开关	功 能	所需设备
P2	基本 I/O 地址	同 CONFIG.WDS 中定义
S1	模拟输入信号类型	置 1、2、3 端口为上,置 4 端口为下
P3	PIT I/O 头	将定时器连接,见下文
P5	PIT 中断头	同 CONFIG.WDS 中定义
P6	EOC 监视头	PA7
P7	EOC 中断头	EOC 不同任何 IRQ 连接
P9	A/D 转换电压	10V
P10	D/A 转换电压	二者均设为+/-极

图 A. 2 RTD ADA2100 跳接配置

基本 I/O 地址(P2)必须设置成软件安装时定义的值。模拟输入信号类型(S1)必须设置,以便形成 8 个具有+/-极的单向通道。PIT I/O 头(P3)必须设置,以保证 OVTO 与 CLK1 相连,KVT1 与 CLK2 相连,OVT2 与 CO2 或-CO2 相连。EG0、EG1 和 EG2 线必须与 +5V 相连。PIT 中断头(P5)必须配置,以便 OVT2 与安装时定义的 IRQ 线相连。EOC 监视器头(P6)必须设为 PA7,EOC 中断头(P7)必须配置,以便 EOC 与任意中断都不相连。A/D 转换电压范围(P9)必须设为 10V。最后,D/A 转换输出电压范围(P10)必须使 AOVT1 和 AOVT2 都与+/-极相连。

A. 1. 2 安装 ADA2100

本节不介绍安装一个卡的具体步骤。假设读者熟悉 IBM PC 机并能完成这一工作。如果

你不能安装一个 I/O 卡,你的系统手册将会告诉你如何操作。本节只介绍一些注意事项。必须确保没有别的卡与 ADA2100 发生冲突。这里首先需注意基本 I/O 地址 +0×17 的一段地址。例如,如果 ADA2100 的基本 I/O 地址是 0×240,那么别的卡一定不能使用 0×240 和 0×257 之间的地址。这些地址必须分配给 ADA2100,而且必须保证没有别的卡使用设置在 PIT 中断头(P5)的 IRQ 线。

A. 1.3 安装 ADA2100 启动初始化

为执行 DACINIT.COM,必须修改 AUTOEXEC.BAT 文件。这可由任何一个文本编辑器来完成。DACINIT.COM 将定时器 82C54 设置在一个可知状态。82C54 定时器不能自动恢复到一个可知的启动状态。这使 82C54 能够在系统不关闭中断情况下在某种模式下启动而中断系统。系统启动时将关闭这个中断,并将中断矢量设为“虚拟”中断。如果因为某种原因系统不能照此运行,则一定产生了错误。运行 DACINIT.COM 不能使 82C54 产生中断。而且,UW DigiScope 程序将使 82C54 的退出失效。如果 UW DigiScope 程序不能正常退出(用 Control Break 命令),则定时器将持续运行,导致中断产生。

A. 1.4 连接 ADA2100

图 A. 3 所示的是输入输出的连接点。“—”信号引脚是地线。ADA2100 上所有的地线都一样。

信 号	+信号引脚	-信号引脚
模拟输入通道 1~8	1~8	21~28
模拟输出通道 1,2	10,11	30,31
数字输入 0~3	36,16,35,15	37
数字输出 0~3	34,14,33,13	37

图 A. 3 ADA2100 的信号连接

于是,当有几个通道分享同一地时,只需一个地导联连入 ADA2100。模拟输入被配置为 8 个单向±5V 的输入电压范围。数字信号直接出/入 82C55,但其电压值不要超过 82C55 的 0 到±5V 范围。并且像 82C55 这样的 CMOS 器件不能驱动大的电流。我们推荐使用诸如 74LS244 的单向缓冲器以保护 82C55。读者可参阅 ADA2100 用户手册的附录 C,以得到关于 82C55 输入输出的更多信息。

A. 2 摩托罗拉 68HC11 EVBU 的配置

选择便宜的摩托罗拉 EVBU 学生设计箱用作数据采集和控制单元。为什么选择该器件?一是因为它易购买,二是它便宜。在本书出版时,摩托罗拉以 68.11 美元出售该箱,这个价格大致不会变化。EVBU 设计用来仿真摩托罗拉 68HC11。EVBU 使用的处理器具有 8 个通道的 8 位 A/D,一个 RS232 串行通信接口,24 位的数字 I/O。作为在数字示波器上的具体实现,EVBU 有 4 个通道的 A/D(0~±5V),它们具有 8 位数字输入和 8 位数字输出功能(0~±5V)。

直接将 EVBU 同主计算机相连,这里仅需一个电缆和一个+5V 直流电源。+5V 电源可用一个电池代替。关于使用电池为 EVBU 供电的介绍随 EVBU 一同出售。

EVBU 出售时不附带电缆,但附带有怎样将 EVBU 同 IBM PC 相连的说明书。我们提供

另一种不同的电缆设计。因为我们的软件不需要硬件握手，可以仅使用带有三股导线的电缆和一些跳线。这些跳线能保证 PC 串行通信卡中目标硬件握手的正常运行。图 A.4 所示的是 PC25 针和 9 针接插件的电缆设计。

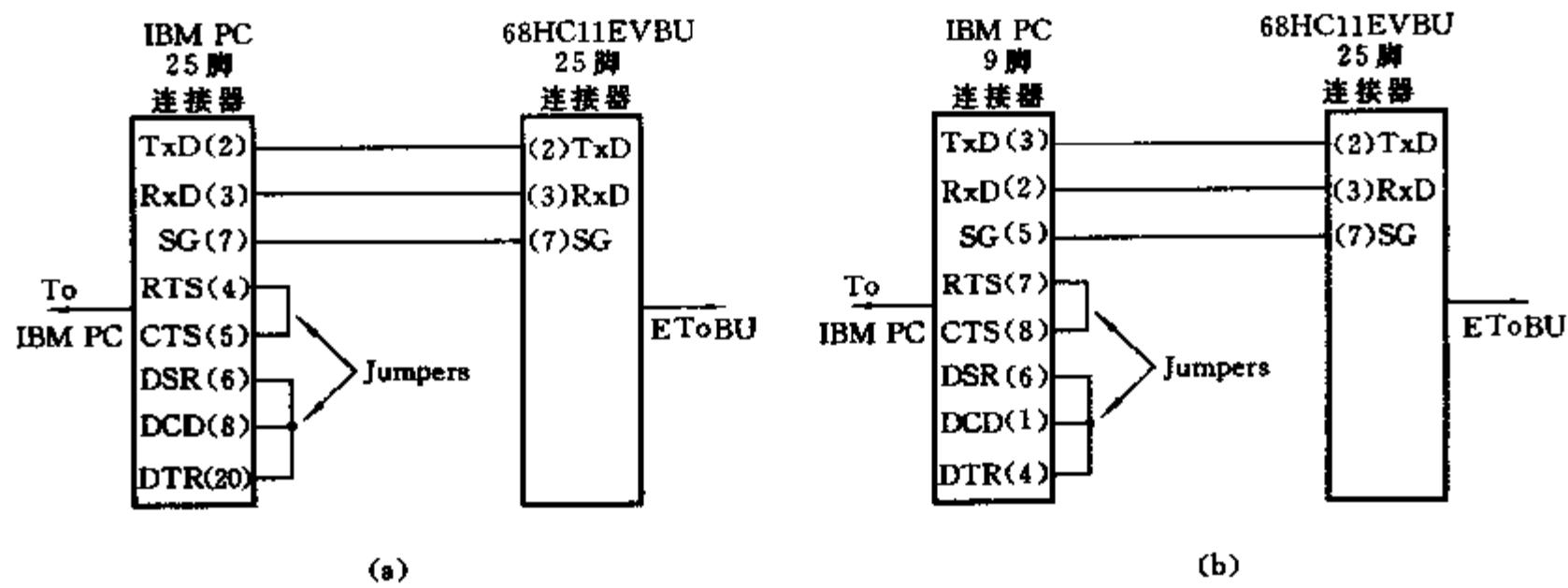


图 A.4 连接 PC 机和 68HC11 EVBU 板的电缆设计

(a)25 脚 PC 机连接器；(b)9 脚 PC 机连接器

A.2.1 EVBU 的配置

图 A.5 所示的是最初的配置，在 EVBU 内部程序存储器部分设定后将有一些小的改动。除了 J2 和 J4，所有的设置都是缺省的。不需要撤除 J2，但需要一个通过 J4 的跳线，将 68HC11 置于自举模式。BUFFALO 监视器使用 J2 来确定监视器是否工作或处理器是否跳到位于 0×B600 的 EEPROM。这只要撤除 J2，使之通过 J4 就可以了。

在进行“安装”和“撤除”工作时，必须操作一个物理的跳线。当进行“缩短”和“打开”时，不需要操作跳线，但需操作位于印刷电路板底部的一个消迹开关。不用消除电路板上任何痕迹，事实上，仅需将 J2 变到 J4 即可。

跳 线	功 能	所 需 设置
J1	输入电源选择头	按应用类型需要设置
J2	程序执行选择	必须拨到 J3
J3,J4	MCU 模式选择头	两个都必须安装
J5,J6	MCU 时钟配置	两个都必须打开
J7	跟踪允许头	可安装或撤除
J8,J9	SCI 配置	二者均为短路
J10、J11、J12、J13	SPI 配置头	可安装或撤除
J14	实时时钟 INT 头	必须打开
J15	TxD 配置头	必须短路
J5	终端波特率选择	穿过 11 针和 12 针
J6	主机口 Rx 信号屏闭	可安装或撤除

图 A.5 摩托罗拉 EVBU 配置

A. 2. 2 连接 IBM PC 和 EVBU

你必须准备一个如摩托罗拉 EVBU 用户手册中描述的电缆。UW DigiScope 程序的 CONFIG. WDS 文件用于决定与 EVBU 通信的串行口，缺省的串行口是 COM1。你可以运行 ADINSTAL. EXE 程序来产生 CONFIG. WDS。

A. 2. 3 在 EVBU 中安装 EDAC 68HC11 程序

你可以在得到本书时得到装有 EDAC. S19 和 EDAC. ASM 的磁盘。EDAC. ASM 是程序的源代码，需驻留在 EVBU 上并和 UW DigiScope 程序通信。EDAC. S19 包含 EDAC. ASM 的十六进制代码。EDAC. S19 是摩托罗拉 S 记录的形式。在 EVBU 手册中有关于 S 记录文件形式的详细描述。

和设计箱一起的是一个称为 pcBug11 的软件设计工具。它可用于 68HC11 的软件设计。利用 68HC11 仿真模式，pcBug 可以用于 PC 机向 68HC11 RAM、EPROM 或 EEPROM 安装程序。pcBug11 中一个称为 LOAD. MCR 的文件允许用 EDAC. S19 命令对 68HC11 的 EEPROM 进行简单的程序设计。在将 EDAC. S19 装载入 68HC11 的 EEPROM 后，必须用 pcBug11 启动 EDAC，使之运行在 68HC11 上。

如果上述软件工具不能工作，则需要为使用 12V 电源和 1 个 100Ω 电阻的监视器在 EEPROM 中设计程序，在这一过程中，J2 必须配置上，以便处理机在复位后能执行 EEPROM 中的代码。这意味着你只需拨动复位键，而不需运行 pcBug11。其中启动 EDAC 的操作是很重要的。图 A. 6 所示的是启动 EDAC 的方法。

启动 EDAC。

在每次复位 EVBU 时都必须利用 pcBug11 来重新启动 EDAC 程序。完成这一任务的命令是：

PCBUG11-E port=N macro=go

这里 N 是 EVBU 连接的 COM 口，N 必须是 1、2、3 或 4。

而且下面的文件必须在当前目录中：

PUBUG11. EXE, TALK. XOO, TALK. BOO, GO. MCR

其中除了 GO. MCR，所有的文件都在 pcBug11 盘上。

GO. MCR 包含在你的 UW DigiScope 程序中。

图 A. 6 启动 EDAC 的方法

A. 2. 4 将信号连入 EVBU

图 A. 7 所示的是将模拟和数字信号与摩托罗拉 EVBU 相连的管脚号，注意模拟输入通道 1~4 与 E 口的 4~7 位相连。

信 号	十信号管脚	一信号管脚
模拟输入通道 1 至 4	44,46,48,50	1
数字输入 0~7	9~16	1
数字输出 0~7	42,41,40,39,38 37,36,35	1

图 A. 7 摩托罗拉 EVBU 的信号连接

这一工作虽然不是必须的,但仍建议那些使用 EVBU 做数字采集和控制的用户加一些输入输出保护措施。像 68HC11 这样的 CMOS 器件对在 0~5V 范围外的电压十分敏感,而且不能承受或产生大电流。

保护模拟输入。

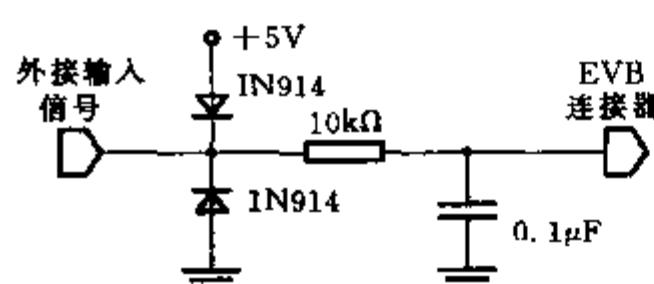


图 A.8 摩托罗拉 EVBU 模拟输入的简单保护电路

图 A.8 所示的是一个对模拟输入进行简单保护的电路。为了不使输入电压超出 +5V 或低于 0V,建议使用二极管进行电压箝位。A 点的电压不会低于 -0.7V 或高过 +5.7V。这足够保护 68HC11。在箝位二极管和 IC 的输入端加了一个限流电阻。10kΩ 的电阻将限制电流大小但不会明显改变输入阻抗。最后,在输入与地之间加一个 0.1μF 的瓷片电容。这个电容将有助于削波噪声。保护数字输入/输出。

数字输入和输出较易保护。将一个单向缓冲器设在外界和微控制器之间,如果一个输入/输出信号达到了破坏电压,则缓冲器将被损坏,而保护了微控制器。另外,用低功耗肖基特(LSTTL)技术制造的缓冲器对过载电压有较大的承受能力。A74LS244 用来做缓冲器是十分合适的。在一个封装中有 8 个缓冲器,在每一路输入/输出中只需使用一片 74LS244。

A.3 摩托罗拉 68HC11 EVB 的配置

对摩托罗拉 EVB 进行设置与对 EVBU 系统的设置相似。EVB 也是用于仿真摩托罗拉 68HC11 的软件。

因为电路板不同,在 EVB 上有不同的跳线配置。它们之间的主要区别是 EVB 的 EEPROM 中已装有监视器程序。对于 68HC11 的内部 EEPROM,不需要编程。监视器用于对 EEPROM 编程。

A.3.1 设置 EVB

配置 EVB 的步骤包括两步。首先是设置跳线并连接 RS232 电缆。第二步是将 EVB 同电源相连接。EVB 需要 +5V、+12V 和 -12V 的电源。EVB 上的跳线必须按照图 A.9 所示的设置,所有这些设置都是缺省的。在用 EDAC.S19 对 EEPROM 编程后,J4 必须拨到 EEPROM。UBFFALO 监视器使用这一跳线来判断监视器是否运行,或者处理器是否跳到位于 OxB600 的 EEPROM 中。开始时,这个跳线必须处于 TBD 的位置,表明 EEPROM 可以运行。

	功 能	所需设置
J1	复位选择头	能安装或撤除
J2	时钟选择头	并联 2 和 3
J3	RAM 选择头	能安装并撤除
J4	程序执行选择	必须最早设为 EEPROM
J5	终端波特率选择	并联 11 和 12
J6	机口 Rx 信号屏蔽	能安装或撤除

图 A.9 摩托罗拉 EVB 跳线配置

A. 3. 2 在 EVB 中装入 EDAC 68HC11 程序

你得到本书时,同时可以得到带有 EDAC.S19 和 EDAC.ASM 文件的磁盘。EDAC.ASM 是程序的源代码,将驻留在 EVBU 中,并与 UW DigiScope 程序通信。EDAC.S19 含有 EDAC.ASM 的十六进制代码。EDAC.S19 是摩托罗拉 S 记录格式的文件。在 EVBU 手册中有关于 S 记录文件形式的详细描述。

和设计箱一起的是一个称为 pcBug11 的软件设计工具,它可用于 68HC11 的软件设计。利用 68HC11 仿真模式,pcBug11 可以用于以 PC 机,向 68HC11 PAM、EPROM 或 EEPROM 安装程序。pcBug11 中有一个称为 LOAD.MCR 的宏文件,允许用 EDAC.S19 命令对 68HC11 的 EEPROM 进行简单的程序设计。在将 EDAC.S19 装载入 68HC11 的 EEPROM 后,必须用 pcBug11 启动 EDAC,使之运行在 68HC11 上。

A. 3. 3 将 EVB 与 IBM PC 机相连

必须准备一个如摩托罗拉 EVB 用户手册所介绍的电缆。用 UW DigiScope 程序来检查 CONFIG.WDS 文件,以便确定用于与 EVB 通信的串行口情况。缺省的串行口是 COM1。你可以运行 ADINSTAL.EXE 程序来产生 CONFIG.WDS。

A. 3. 4 将信号接入 EVB

关于将信号接入 EVB 的方法,请参阅 A. 2. 4 小节中的介绍。相同的法则适用于保护模拟输入以及输出信号的输出、输入。图 A. 10 所示的是将模拟和数字信号接入 EVB 的管脚号。

信 号	十信号引脚	一信号引脚
数字输入通道 1~8	43,45,47,49,44,46,48,50	1
数字输入 0~7	9~16	1
数字输出 0~7	42,41,40,39,38,37,36,35	1

图 A. 10 摩托罗拉 EVB 的信号连接

A. 4 虚拟输入/输出设备(数据文件)

虽然为虚拟设备讨论硬件配置有些令人奇怪,本节还是介绍了一些怎样建立 PC,以运行一个虚拟 I/O 设备的办法。所需 PC 的最小配置是 640KB 内存、VGA 或单显卡及一个软驱。为了更好地进行工作,建议使用一个硬驱。如果数字示波器在读每一个样本光点时都读一下软盘,那么“实时数据采集”将看起来十分逼真。

A. 5 在二进制文件上加一个文件头:ADDHEAD

由 UW DigiScope 软件产生的文件有特殊的格式。可以用 UW DigiScope 的 Stat(U)S 命令在一个文件头中读有关信息,而且还提供了一个特殊程序以产生一个文件头。

如果数据文件中包含有严格二进制下的十六位整型数(例如:语言的十六位整型数),则可以用 ADDHEAD 在文件上加一个文件头。ADDHEAD 将提示用户要加在文件头上信息内容。如果用户不输入任何信息,则 ADDHEAD 将使用缺省值。缺省值位于提示的括号中。当所有信息被写入后,ADDHEAD 读二进制文件,并将文件头和数据写入一个称为 filename.dat 的新文件中。

附录 B 数据采集与控制程序

本附录介绍 DACPAC、LIB 中用于数据采集和控制的程序。DACPAC 的程序设计有两个目的。第一个是用于 UW DigiScope；第二个目的是给出信号转换设备的简单的软件接口，以便其它程序使用。DACPAC.LIB 与其他所需文件头都存在磁盘上，它们可以同本文一起得到。DACPAC 程序执行模拟信号采集、数字信号采集、模拟信号输出、数字信号输出以及定时功能。

DACPAC 为三种设备提供接口，它们是装在 IBM PC 机或兼容机内部的实时设备：模拟和数字 I/O 接口卡、用 RS232 通信的外部信号板（摩托罗拉 S68HC11EVBU）以及虚拟输入设备（数据文件）。

这三种设备有相同之处。所有设备均由四种基本操作控制。它们是 OPEN、GET、PUT 和 CLOSE。OPEN 初始化设备；GET 从设备中检索到一段数据；PUT 将数据转给设备进行输出；CLOSE 终止设备上运行的所有操作（包括关闭一个文件、屏蔽中断和消除通信连接）。

本附录包括五部分。第一部分介绍 DACPAC.LIB 中的数据结构；第二部分介绍主程序，用于调用每个设备的程序。后面三部分讨论 DACPAC 中的程序。对于每种设备，用于控制设备的程序被分为一些子部分。除了上面四种程序，还加入两种特殊的用于 GET 和 PUT 的程序，用于数字信号的输入/输出。各子部分详细描述程序所需数据和程序所设置数据。

在进行详细讨论前，我们给出一个简单警告：在退出程序时，一定要保证关闭所有的设备。这意味着，必须十分小心使用 C 的库函数 exit()。如果你退出程序时未关闭设备，则你的 PC 机就会死机。

B. 1 数 据 结 构

这是本部分的程序所用的重要数据结构。最重要的是 Header 数据结构。这些数据结构中包含设备已经采集或将要采集的数据的所有重要信息。

title, creator, source 和 type 是程序中所用的字符串。外壳程序不使用这些字符串。当打开任何一个由 DACPAC 支持的设备时，用户必须十分小心地初始化一些将传给 OPEN 程序的 Header 文件结构，而不是初始化由 OPEN 程序建立的数据结构。图 B.1 所示的是 DACPAC 中数据采集程序使用的数据结构。

Header 结构中的一个重要部分是 CHANTYPE 数组。这个数组包含有每一通道的特殊信息。其中有一个称为 TYPE 的枚举类型变量，它不由任何 DACPAC 程序使用，而只提供给用户。

DATATYPE 不仅仅是 Short 类型，而且所有 DATATYPE 类型的变量都有一个约定，即它们是二进制补码的 16 位整型数，它的 0 值对应于 0V。在计算机 DATATYPE 数据给出电压值时，只需乘上被读设备给出的数据的分辨率（如：位/V）即可。将电压除 2 的分辨率次幂，就可以得到这个数。由 Header->voltlow 减去 Header->volthigh，可以得到电压范围。分辨率可由 Header->resolution 得到。

```

typedef short DATATYPE; /* data type will be 16 bit integer */
typedef enum (ECG,EMG,EEG,CV,RESP,EKG,ABC,ERROR) chantype_t;
typedef struct ChannelRecord {
    chantype_t type;
    float offset;
    float gain;
} CHANTYPE;

typedef struct HeaderRecord {
    char title[80];           /* title to be used for display */
    char filename[40];        /* filename containing data */
    FILE *fd;                /* file pointer returned by fADOpen */
    char creator[80];         /* name of person who gathered data */
    char source[80];          /* name of A/D card or other device */
    char type[80];            /* i.e. 12-lead ECG */
    float volthigh;           /* High limit of input voltage */
    float voltlow;             /* Low limit of input voltage */
    int stepsize;              /* step size used by data compress. */
    char compression;          /* Data compression type */
    int rate;                  /* positive integer */
    int resolution;            /* number between 8 and 16 */
    int num_channels;          /* number of channels (in array) */
    int num_samples;            /* number of samples */

CHANTYPE channel[20]; /* pointer to array of Channel info */
DATATYPE *data; /* pointer to data buffer */
} DataHeader_t;

```

图 B. 1 文件头和通道类型数据结构(defns.h)

B. 2 顶层设备程序

DACPAC 中包含了调用文件句柄的程序。这些程序包含在 DAC.C 中。含有原型的文件头在 DAC.H 中。这个原型如图 B. 2 所示。图 B. 3 所示的是每种设备的调用约定。为确定传送什么值给这些程序,请参阅你使用的设备的相关说明。顶层程序 PUT()用数值 1 来调用 faDput-buffer()。

```

char OPEN(DataHeader_t * Header,char * dir,int device);
char CLOSE(DataHeader_t * Header,int device);
char GET(DataHeader_t * Header,DATATYPE * data,int device);
char PUT(DataHeader_t * Header,DATATYPE * data,int channel,int device);

```

图 B. 2 顶层程序原型(DAC.H)

```

#define VIRTUAL    1
#define EXTERNAL   2
#define INTERNAL   3

/* calling conventions for Internal device */

OPEN(&Header,"",INTERNAL);
CLOSE(&Header,INTERNAL);
GET(&Header,&data,INTERNAL);

/* calling conventions for External device */

OPEN(&Header,"r",EXTERNAL);
CLOSE(&Header,EXTERNAL);
GET(&Header,&data,EXTERNAL);

/* calling conventions for Virtual device */

OPEN(&Header,"rt",VIRTUAL);
CLOSE(&Header,VIRTUAL);
GET(&Header,&data,VIRTUAL).

```

图 B. 3 三种设备的调用设置

B. 3 内部 I/O 设备(RTD ADA2100)

RTD(实时设备)ADA2100 可以比本附录中描述的其它两种设备执行更多的 I/O 功能。使用下面的程序,ADA2100 可以阅读 8 个单向模拟输入,驱动 2 个-10~+10V 的模拟输出,读 4 个数字输入,驱动 4 个数字输出。数字 I/O 使用范围在 0~5V 的标准 TTL 电平内。

关于 ADA2100 的配置,请参阅附录 A。一些项目,例如输入/输出的配置是由软件完成的。图 B. 4 所示的是执行上述功能的程序和调用的设置。图 B. 4 所示的这些程序的原型在 IDAC.H 中定义。如果调用成功,则所有程序的返回值都是 1,若失败,则返回 0。有一些程序总是成功的,返回值总是 1。

```

char Iopen (DataHeader_t * Header);
char Iclose (DataHeader_t * Header);
char Iaet (DATATYPE * data);

```

图 B. 4 内部卡调用(IDAC.H)

B. 3. 1 打开设备 Iopen()

Iopen()程序可初始化 ADA2100,并在指定的采样速率下进行数字输入、输出和模拟输入。定时器以指定的速率发中断。中断矢量被设置为指向一个设置的小程序,由 Iget()检查。

图 B. 5 所示的是怎样使用 Header 数据结构,以及结构中什么元素需要初始化,什么元素

由 Iopen() 初始化的情况。

未列出的元素不为 Iopen() 所初始化或使用,因而可以在调用程序外壳中使用。

The following items must be initialized before the device is opened.

```
Header->rate  
Header->num_channels
```

The following items are initialized by the Iopen() routine.

```
Header->volthigh = 5.0;  
Header->voltlow = -5.0;  
  
Header->resolution = 12;  
if (channels>7) channels = 7; /* should fix this sometime */  
Header->num_channels = channels;  
Header->num_samples = 0;  
  
for (i=0;i<Header->num_channels;i++) {  
    ChanOffset(Header,i) = 0.0;  
    ChanGain(Header,i) = 1.0;
```

图 B.5 Iopen() 程序

B. 3.2 关闭设备 Iclose()

Iclose() 程序关闭 ADA2100 上的时钟,重置中断矢量到调用 Iopen() 前的值。Iclose() 既不修改也不需要 Header 数据文件中的元素,仅仅为了通用性,而将 Header 传入 Iclose()。

B. 3.3 进行模拟输入的读操作 Iget()

Iget() 程序检查标志是否已由定时中断程序设置,如果标志已被设置,Iget() 读 Iopen() 需要的通道。将这些通道中的数据装入一个数据 DATATYPE 并返回一个 1。如果标志未被设置,Iget() 返回一个 0。如果速度 0 被选择,则 Iget() 总是将一个值读入文件,并返回一个 1。

B. 4 外部 I/O 设备(摩托罗拉 68HC11 EVBU)

通过 RS232 通信线路,可以与摩托罗拉设备相连。EVBU 和 EVB 使用相同的接口。DACPAC 中提供的程序如图 B.6 所示。它们与 ADA2100 提供的程序相似,但存在两个主要的不同点。

虽然接口不太重要,但仍建议在使用 EVBU 进行数据采集和控制时,用户在输入和输出端上要加一些保护电路,A. 2.4 小节给出了一种保护电路:

```

char Eopen (DataHeader_t * Header,char * dir);
char Eclose (DataHeader_t * Header);
char Eget (DATATYPE * data);

```

图 B. 6 Edac 调用约定(EDAC. H)

B. 4. 1 设备启动 Eopen()

Eopen()程序如图 B. 7 所示。

Eopen()程序初始化串行口后,发送一个软复位命令,随后发送对 EVBU 的设置命令。这些设置命令告诉运行在 EVBU 上的软件以什么样的速率对模拟输入进行采样,以及对哪些输入采样。因为 EVBU 只能以 9600 位/s 的速率进行通信,在某些情况下需对数据进行缓冲,并在 PC 机有时间读数据时将缓冲区中的数据送往 PC 机。所以在这些命令的结尾将指明是否在使用实时数据传输,这可使数据快速传送。将 dir 设为 r,可调用 Eopen(),从而将 EVBU 设为实时传输状态;将 dir 设为 b 可调用 Eopen(),从而将 EVBU 设为快速传输状态。

The following items must be initialized before the device is opened.

```

Header->rate
Header->num_channels

```

The following items are initialized by the Eopen();routine.

```

Header->volthigh = 5.0;
Header->voltlow = -5.0;

Header->resolution = 8;
if (channels>8) channels = 8;
Header->num_channels = channels;
Header->num_samples = 0;

for (i=0;i<Header->num_channels;i++) {
    ChanOffset (Header,i) = 0.0;
    ChanGain (Header,i) = 1.0;
}

```

图 B. 7 Eopen 程序

在这里没有列入的元素将不为 Eopen()所初始化或使用,所以可以在调用程序中无条件使用。

如果你不想执行模拟输入,则只需要简单地给 Eopen()一个值为 0 的采样速率即可。这样做时,将无任何模拟输入被读入。在这种情况下,不能使用 Eget(),Eopen()假定调用者不希望进行模拟输入,而数据输入和输出仍将被执行。

B. 4. 2 关闭设备 Eclose()

Eclose()程序将停止 EVBU 上所有的模拟信号采集工作。它将一个软复位命令发送给 EVBU,这意味着 EVBU 将停止读入模拟输入,而执行数字 I/O,直到 Eopen()被调用, EVBU 重新接收到设置信息为止。

B. 4.3 进行一个模拟输入的读操作 Eget()

Eget()程序检查是否已有数据由 EVBU 传输过来。如果数据无误传输过来了，则 Eget()程序将通道中的数据输入到数据指向的 DATATYPE 变量中，并返回一个 1，如果没有数据到来，则 Eget()程序返回一个 0。

B. 5 虚拟 I/O 设备(数据文件)

图 B.8 所示的是用户可调用的各种程序的约定。设计这些程序有两个目的。第一个目的是，在用户没有数据采集硬件的条件下，帮助用户进行实时数据采集。第二个目的是，允许对使用上面介绍的数据采集程序得到的数据进行存储、标定和回放。这里介绍的程序不需要使用特别的硬件。如果你有一台具有一个软区、640KB RAM 和单显或 VGA 监视器的 IBM PC/XT/AT 机或兼容机，则你就能运行这些程序了。

```
char fADopen (DataHeader_t * Header, char * dir);
char fADclose (DataHeader_t * Header);
int fADget_buffer (DataHeader_t * Header, int size, DATATYPE * buffer);
char fADget (DataHeader_t * Header, DATATYPE * signal);
int fADput_buffer (DataHeader_t * Header, int size, DATATYPE * buffer);
void fADputfile (DataHeader_t * Header);
void fADgetfile (DataHeader_t * Header);
```

图 B.8 fAD 调用约定(fAD.H)

B. 5.1 文件结构

文件结构使用一个 ASCII 文件头来描述文件的数据。该文件头由一个单独的空行终止。数据则紧接在这一空行后，并以二进制形式存储。图 B.9 所示的是在文件中的一个 Header 的例子。注意，为了使 fADopen()正确运行，字段必须与图 B.9 所示的一致。虽然文件以文本方式存储，但是不能使用常规文本编辑器进行文件的编辑，因为文件是以二进制存储的。A.5 节介绍了怎样利用 ADDHEAD 程序在现存文件中加入一个文件头的方法。

```
Title: ECG data from file ecg105
Creator: unknown
Source: unknown
Type:ECG single channel
Volthigh: 12
Voltlow: -12
Step: 0
Compress: N
Resolution: 12
Rate: 200
Channels: 1
Samples:12000
Chan: 0 Gain 1.000 Ofst 0.0000 Type ECG
```

图 B.9 文件的例子

B. 5.2 打开设备 fAD open()

fADopen()程序可打开一个文件、初始化定时器、为伪真实时数据采集提供软件延时。如果你不希望进行模拟输入，则只需将 dir 设为 r，调用 fADopen()进行读操作即可；如果希望进行定时输入，则将 dir 设为 rt，调用 fADopen()，进行定时输入的读操作；如果希望进行输出，则将 dir 设为 w，调用 fADopen()，进行写操作。当打开文件进行输出时，Header 结构中所有文件头都需要设置。当打开文件读时，只需设置 Header—>filename。如图 B. 10 所示，所有其他内容将由文件中读入。函数 fADopen()只打开文件读(选择项为 r 和 rt)或写(选择项为 w)。图 B. 10 中给出怎样使用 Header 文件结构，结构中什么内容必须初始化，以及哪些元素由 fADopen()初始化的方法。在图中未列人的元素或者不由 fADopen()初始化，或者不为其所用，因此，可以在调用程序中任意使用。

The following items must be initialized before the device is opened.

Header—>filename;

The following items are initialized by the Iopen(),routine.

```
Header—>title;
Header—>creator;
Header—>source;
Header—>type;
Header—>fd;
Header—>stepsize;
Header—>compression;
Header—>rate;
Header—>resolution;
Header—>num_samples;
Header—>volthigh;
Header—>voltlow;
Header—>compression;      /* no compression */
Header—>resolution;
Header—>channels = channels;
Header—>num_samples = 0;

for (i=0;i<Header—>num_channels;i++) {
    ChanOffset (Header,i);
    ChanGain (Header,i);
    ChanType (Header,i);
```

图 B. 10 fADopen 程序

B. 5.3 关闭设备 fADclose()

fADclose()程序执行两个非常重要的操作。首先，非常明显，它关闭由 Header—>fd 的文件；其次，如果 fADopen()由 dir 指向 t 进行调用，fADclose()将从 time-of-day 中断序列中除去时钟程序。

B. 5.4 进行一个模拟输入的读操作:fADget(), fADget-buffer(), fADget-file()

fADget()程序检查由中断服务程序设置的标志,如果标志已被设置,则 fADget()从由 Header->fd 的文件的每一通道中采一个样,并返回一个 1。将采集到的数据放入由数据指向数组的数组中。调用程序必须开辟足够的空间以存储数据。如果标志未由中断服务程序设置,则 fADget()程序返回一个 0。

如果不需定时,并且所有的数据都需要读入进行处理,则可利用 fADget file()程序进行处理。fADget file()程序可以打开由 Header->filename 的文件,将文件句柄读入 Header 的数据结构,而且将数据读入到一个由 Header->data 的数组。注意 fADget -file()程序同样可将由 Headerd->data 的数据分配给内存。

如果文件太大,以致不能一次读完,或者只需要文件中的一部分数据,则 fADget-buffer()将十分有用。fADget-buffer 在空间允许时将读入尽可能多的采样数据。如果读入成功,则 fADget-buffer 将返回由文件中读出的采样数目。一次采样即是一个通道中的一次读操作。

B. 5.5 在文件中写入数据:fADput-buffer()和 fADput-file()。

用于将数据写到文件中的两个程序是,fADput-buffer()和 fADput-file()。用于写入文件的程序不需要定时。在这里,不需要有同 fADput()对应的 fADget(),因为调用缓冲区为 1 的 fADput-buffer()将会把每个通道中的一次采样写入到由 Header->fd 的文件中。这正如我们期望的名叫 fADget()的程序所作的事。记住每一次采样都是对每一通道的一次读操作,顶层程序 PUT()以数值 1 来调用 fADput-buffer()。

fADput-file()使数据存储进一个文件变得简单。fADput-file()将打开名叫 Header->filename 的文件,把所有的文件头信息写入文件,将所有数据写入文件然后关闭文件。

附录 C 数据采集与控制——若干提示

附录 A、B 介绍了如何建立硬件以便使用 UW DigiScope 软件,以及如何使用子程序。其中对关于如何开发这些程序和建立能实现 DAC 的硬件提供了某些提示。这里并非详细地把每一步应做的步骤都加以说明,而只给读者提供一些提示和建议,例如做什么和哪些不要去做。这里还对如何在 IBM PC 机上执行一些必要的操作,来进行采样和控制数据做了介绍,并提供了一些有用的参考资料。

本附录将指导读者如何建立一简单的数据采集与控制系统。该系统包括前面附录中提到的作为基本要素之一的、将模拟和数字信号读入计算机的三种不同型号的装置。其中第一种型号装置是一个内部装置,该装置通过内部扩展总线连接在电脑上,并通过扩展总线与电脑进行通信。另一种是 I/O 装置,可以是外部的。外部通信装置通过一个串行或并行通信接口与计算机交换信息。最为普通常用的通信接口为 RS232(串行)和 IEEE488(并行)。下面仅对 RS232 通信装置加以说明,因为,几乎所有的 IBM PC 机在结构上都设置了一个现成的 RS232 串行接口。最后一种装置是一个虚拟 I/O 装置。DigiScope 程序使用数据块和一个定时器来中断模拟信号。这些相同的公共数据块被用来存储及恢复通过内部和外部模拟输入装置获得的数据。

I/O 装置包括四种基本装置:开启装置、输入装置、输出装置、关闭装置。启动装置用于对 I/O 的请求作出响应,并发出必要的中断信号以提供精确的时间。输入装置将决定数据请求可以得到并通过该装置恢复数据。输出装置用于输出一组数据。关闭装置用于终止所有正在执行的操作,同时屏蔽开启装置的任何中断。下面对前面提到的三种装置各自的四种操作加以准确描述,最后将给出一些提示,以帮助你编写自己的 DigiScope 程序。

所有为 PC 机提供的程序都用 Turbo C 语言编写。我们提供给读者 Turbo C 语言的手册,以便读者了解诸如串行通信总线以及数据 I/O 的一些知识。

C. I 内部 I/O 装置(RTD ADA2100)

这种装置安装在 IBM PC 机的主板上。图 C. 1 所示的为 ECG 放大器与 I/O 插卡的连接图,请注意不需要其它的硬件装置。

C. 1.1 采用 Turbo C 语言的 I/O 卡接口

装在 PC 机中插卡的接口采用 Turbo C 语言的库函数从处理器的 I/O 接口读/写数据。`inportb()` 和 `outportb()` 分别用于 8 位数据的输入/输出操作,或一起用于 16 位数据的输入/输出操作。ADA2100 是一个 8 位 I/O 卡,因此,使用了 `inportb()` 和 `outportb()`。本文给读者提供了 Turbo C 语言的使用手册,以便读者了解更多信息。图 C. 2 所示的为如何用 `inportb()` 和 `outportb()` 建立 8259 的中断控制器的一个例子。8259 虽然是 PC 机系统的一个组成部分,但是采用 `inportb()` 和 `outportb()` 对 8259 进行读/写操作与对插卡的读/写操作是一样的。

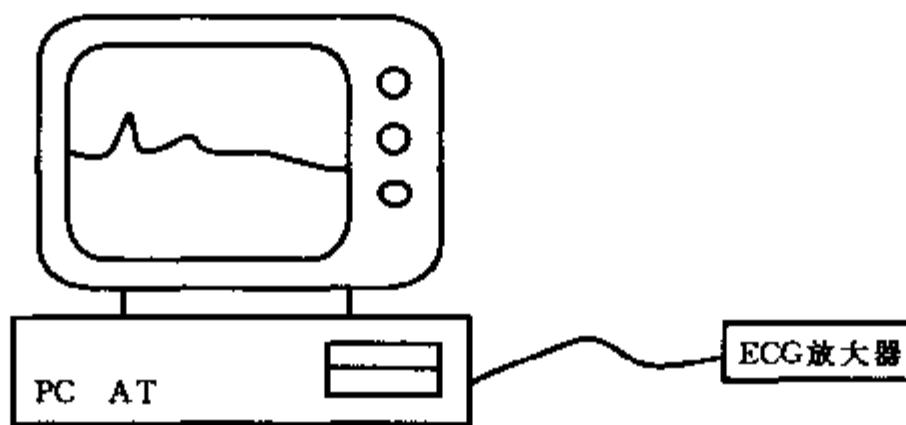


图 C.1 外部卡数据采集和控制系统

```

main (char * argv[], int argc)
{
    disable ();                                /* disables all interrupts */

    OldVector = getvect (10);                  /* set interrupt vector */
    setvect (10,Itimer);

    OldMask= inportb(0x21);                   /* unmask IRQ 2 on 8259 */
    mask =OldMask & 0xFB;
    outportb(0x21,mask);
    enable ();                                /* enables all interrupts */

    /* code that can be interrupted by INTERRUPT 10 */

    disable ();
    outportb(0x21,OldMask);                  /* return mask to original value */
    setvect (10,OldVector);                  /* return interrupt vector to */
                                              /* original value */
    enable ();
}

```

图 C.2 设置中断矢量及中断屏蔽寄存器

C. 1. 2 IBM PC 机上用 Turbo C 语言处理中断

下面将讨论正确产生中断的一些基本操作,而更重要的是,如何确保在不希望产生中断时屏蔽掉这些中断。编写有中断功能的程序较为困难,因为中断可能出现在程序运行的任何时刻,而当中断出现时,我们的控制手段是有限的。当需要停止一个内部处理转去解决另外一个问题时,我们就会采用中断。中断有很多起因。中断可能来自键盘、磁盘驱动器、串行接口、日期中断或者其他许多因素。在这种情况下希望事件总是在固定的时间间隔内发生。

与中断控制器的接口。

按 C 语言的习惯使用十六进制数。当中断控制器告诉处理器外部运行过程已申请中断时,如果中断被允许,那么处理器将对该中断作出信号响应。这时,中断控制器就会告诉处理器

到哪里去寻找中断矢量,中断矢量指的是中断服务器在总线上的地址。

8259 中断控制器有许多功能,但建议只改变中断屏蔽寄存器 IMR 即可。若想要更详细地了解 8259,请参阅《Intel 微处理器和外围设备(第 1 卷)》,或者 1983 年的 Eggbrecht。IMR 位于 I/O 区的 0×21。IBM PC 机采用了两块 8259 中断控制器。主中断控制器位于 0×21,从中断器位于 0×70。这两个中断控制器串联起来可提供 15 种不同优先级别的中断。主中断控制器的 IRQ2 与从中断控制器相连。

当屏蔽或打开一个中断时,仅仅屏蔽所需的中断或打开所需的屏蔽,在使用中断之后,关闭中断也很重要,图 C. 2 所示的为在程序的开始打开中断,而在程序的结尾关闭中断。

与操作系统的接口——中断处理。

中断控制器打开中断的同时必须将中断矢量初始化,使中断矢量指向正确的中断处理器。Turbo C 语言提供了两种简单方法来达到这一点。在建立中断矢量,以指向新的中断处理器之前,当前的中断矢量应该保留,以便中断处理之后系统可以返回原位置。

在 PC 机上使用中断时,应该记住的重要一点是,将中断控制器和中断矢量返回到它们在程序开始时所在的位置。这样就得先存储 IMR 和中断矢量(用两个全变量代替),并在程序的结尾时恢复 IMR 和中断矢量的值。如果不恢复,则程序运行时将会出现严重的问题。因为不论什么原因一旦发生这种情况,用户必须重新启动系统。

C. 2 外部 I/O 装置(摩托罗拉 68HC11 EVBU)

对于外部 I/O 装置,我们选择了摩托罗拉 68HC11 EVBU 学生级评估板,因为它物美价廉。使用该装置的缺点是建立它较为困难,将 EVBU 转换为 I/O 装置,需要装配一套 68HC11 系统程序,以便能与 PC 机通信,并且提供必要的输入/输出。我们已经帮读者完成了这个任务,如图 C. 3 所示的为一个例子,另外 EVBU 编程也不难。这套学生级评估板包括了充足的工具和文件,因而任何有编写汇编语言经验的人都会对 EVBU 编程。

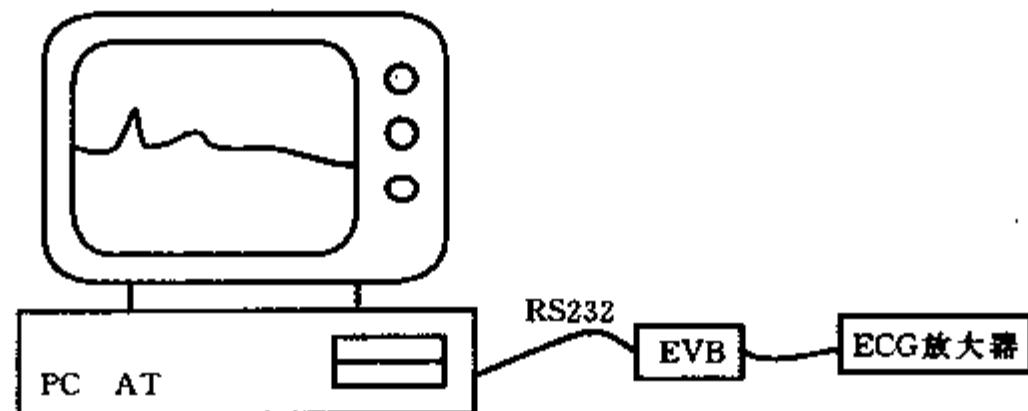


图 C. 3 摩托罗拉 EVBU 数据采集控制系统

使用这套装置的优点之一是,所有严格的定时在 68HC11 上都可以完成,这意味着能避免在 PC 机上使用中断。然而,68HC11 有一个基准定时器中断,这个中断在一个微处理器上比在一个 PC 机上容易处理,PC 机因为有操作系统和驱动器等等,因此结构复杂得多。

这一节将简要介绍如何用 Turbo C 语言进行串行通信。我们给出两种方法,一种用 Turbo C 语言对 PC 机编程;另一种用汇编语言对 EVBU 编程。这两种方法都能产生一个简单的通信序列。

C. 2.1 一般操作

这里主要介绍一下如何操作启动、关闭、获取、设置等功能。

(1) 设备启动 启动装置必须首先初始化 RS232 串行接口,用 EVBU 建立一个通信环节,同时根据所需的执行时间和日期对 EVBU 进行初始化。对串行接口的初始化可以很容易地用 Turbo C 语言的 biescom()实现(见图 C. 4)。在 C. 3.3 小节,我们举出了一个例子。图 C. 5 所示的为外部装置启动时连续通信的情况。

```
#define COM1      0
#define COM2      1
#define COM3      2
#define COM4      3
define SETTINGS  (0xE0 | 0x03 | 0x00 | 0x00)
                  /* 9600 Bd 8 bits 1 stop no parity */

                  /* CPORt contains COM port is being used */
int CPORt = COM1;

                  /* initialize serial port */
bioscom(0, SETTINGS, CPORt);

                  /* read the status of serial port */
status = bioscom(3, 0, CPORt);

                  /* read the input buffer of serial port */
port = bioscom(2, 0, CPORt);

                  /* Send an ASCII Q out on serial port */
bioscom(1, 'Q', CPORt);
```

图 C. 4 bisscom()作为 Turbo C 库函数操作

- PC sends reset command “R” to EVBU for soft reset.
- EVBU gets “R” and jumps to reset vector. As part of reset, EVBU sends “R” to PC to echo Reset command.
- PC sends channel mask “Cx” to EVBU. x is an 8-bit mask with 1’s in positions corresponding to analog input channels to be read.
- EVBU echoes channel mask “Cx” and saves it.
- PC sends delay setting, “Dxxxx” to EVBU. xxxx is in Hex and is timer ticks of 68HC11 timer.
- EVBU echoes “Dxxxx” command and saves delay setting.
- PC sends go command “G” to EVBU.
- EVBU echoes “G” command and starts timer interrupts.

图 C. 5 用于实时数据传输的初始化 EVBU 通信库列

(2) 设备关闭 要关闭设备时,仅仅需要给 EVBU 送一条指令,以终止所有的数据采集,并同时停止给 PC 主机发送信号。发送信号的步骤如下。

开启 EDAC。

每次 EVBU 的复位都需要使用 PC Bug11 去重新启动 EDAC 程序。其指令格式为：

PC Bug11-E port=N macro=go

式中，N 是 COM 接口的号码，即 EVBU 连接的端口。N 只能是 1,2,3 或 4，而且当前目录下还有以下文件：

PCBUG11.EXE, TALKER.XOO, TALKER.BOO, 以及 GO.MOR。

除 GO.MORE 外的所有这些文件都包括在与 EVBU 一起给出的 PCBUG11 盘中，GO.MOR 包含在 DigiScope 盘中。

(3) 数据输入 输入数据时，必须给 EVBU 读入数据的请求，并等待响应。记住应该限制主机等待响应的时间，以免计算机因为等待一个不会发生的事情而死机。

(4) 数据输出 输出数据时，必须给 EVBU 一个指令，告诉 EVBU 取出一组特定数据。

C. 2.2 用 Turbo C 语言进行串行通信

完成串行通信必须采用 bioscom()方法。bioscom()是一种可以用于初始化串行口的多功能手段。检查初始状态、读入串行接口、通过串行接口输出一个字节，Turbo C 语言的参考手册给出了许多执行以上操作的实用例子，图 C. 6 所示的是为本文开发的一些代码的例子。

```
send_command ("R"); /* reset the EVBU */

/* Tell EVBU which channels the host wants to read. */
/* The EVBU will send this many bytes to the TERMINAL */
/* at the sample rate */

for (i=0, mask=0, i<Header->num_channels; i++) {
    mask |= 1 << i;
}
sprintf (buf, "C%X\r", mask);
send_command (buf);

/* calc delay betw. samples */
if (Header->rate != 0) {
    delay = 2000000 / Header->rate;
} else {
    delay = 0; /* if rate is 0 send 0 delay */
}
sprintf (buf, "D%X\r", delay); /* Tell EVBU delay */
send_command (buf);

send command ("G"); /* Tell EVBU to start sampling */
```

图 C. 6 执行图 C. 5 过程的 PC 机 Turbo C 程序

C. 2.3 采样通信程序及其执行码

下面介绍 Eopen()中用以开启 EVBU 的通信程序，其中去掉了些错误检查子程

序,以便易于理解其指令。这些程序像在 PC 机上执行 Eopen()和在 EVBU 上执行 EDAC.ASM 一样被精确执行。这些程序如图 C. 5 所示,PC 机应该给 EVBU 提供采样速率和

LDAA	# 'R'	send signon character to host	
JSR	DATOUT		
SETUP	JSR	DATIN	
BRCLR	FLAGS		
RCVDAT	SETUP	wait for a character	
JSR	DATOUT	echo to host for host's error checking	
CMPA	#'D'	Check for delay	
BNE	SETUP1		
JSR	GETDELAY	Read in the hex value and save	
BRA	SETUP		
SETUP1	CMPA	#'C'	Check for number of channels
BNE	SETUP2		
JSR	GETCHAN	Read in the hex value and save	
BRA	SETUP		
SETUP3	CMPA	#'G'	Check for Start
BNE	SETUP		

* set up A/D converter and start interrupts

图 C. 7 执行图 C. 5 过程的 68HC11 汇编语言

通道号,并指出是用块传递还是实时传输。在 PC 机和 EVBU 上执行的程序分别如图 C. 6 和图 C. 7 所示。如图 C. 8 所示,Send-Command()命令送一个空字符串给 EVBU。

请注意,一个单字节码被送到 EVBU 后将指明使用哪些通道读入。每一位的位置都对应于一个输入通道。所有标号均在 0~7 选取。如果只有第 2 位在码里置位,那么通道 2 将成为唯一的读入通道。

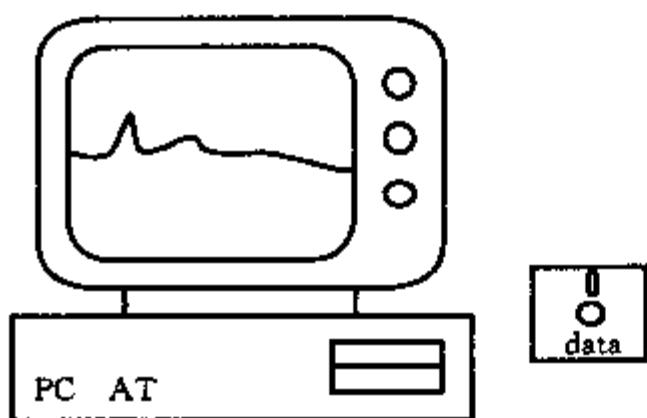


图 C. 8 虚拟模拟输入的微机和硬件

C. 3 虚拟 I/O 装置(数据文件)

I/O 操作对于大多数程序员而言是一个普通的操作。这里将介绍一种使用数据文件模拟物理 I/O 装置的方法。首要任务是产生一个数据文件的 I/O 操作,以便能模拟一个物理 I/O 装置的操作。接下来需要提供一些定时操作的程序,以便输入、输出以特定的速率进行。用数据文件模拟物理装置的困难之一是要提供定时器。为了完成虚拟模拟输入的定时操作,系统必须有与 IBM PC 机或 PC/AT 机在时间上相匹配的中断。

C. 3. 1 从日期中断中节省时间

图 C. 9 所示的为怎样从日期中断中节省时间的程序。旧时间向量要保存下来,以便恢复。
• 198 •

同时也是为了能够进行大约 18.2 次/s 的新的定时中断服务。这使得日期时钟能以接近正确时间快慢的速率运行,否则日期时钟将会在程序处理中断之后变成一个未知的数据。如果要了解如何从日期中断中节省时间,请参阅 Bovens 和 Brysbuer 的文献(1990 年版)。

```
#define TIMER0      0x40      /* I/O mem locations for 8253 */
#define TIMER_CTRL   0x43

void interrupt (*OldTimer)(void); /* global to save old ISR */

    disable();           /* disable interrupts */
    OldTimer = getvect(0x08); /* save the old ISR */
    setvect(0x08, SuperTimer); /* set new ISR to our routine */

    SetUpTimer(rate);      /* initialize timer to rate */

    enable();             /* enable interrupts */

/* * * * * code that can be interrupted goes here * * * * */

    disable();           /* disable interrupts */
    setvect(0x08, OldTimer); /* restore ISR */

    outportb(TIMER_CTRL, 0x36); /* timer 0, mode 3, LSB and MSB */
    outportb(TIMER0, 0x00);
    outportb(TIMER0, 0x00);

    enable();             /* enable interrupts */
```

图 C.9 节省日期中断(INT8)的程序

C. 3.2 指定采样速率的 8253 初始化

图 C.10 所示的为按指定速率建立一个定时器的程序。这种方法要求,先累计采样之间的时钟脉冲数,然后设立定时器。定时器的建立需要将 TIMER0 设置为 MODE2,并先写最低位,后写最高位。定时器的控制器位于 I/O 区的 0×43,而 TIMER0 在 I/O 区的 0×40。8253 有三个定时器,TIMER1 用于扬声器的输出,而 TIMER2 用于动态存储刷新,所有这些定时器的顺序不能打乱。我们给读者提供了《Intel 微处理器及终端手册(第三卷)》,以便对 8253 有更多了解。

C. 3.3 用 Turbo C 语言编写中断服务程序

非常重要的一点是 ISR 必须在中断服务程序执行之后使系统返回到中断前的状态。也就是说,ISR 应该在开始启动时保留处理器的当前状态。所幸的是,Turbo C 语言可以在程序出现“空中断”的情况下做到这一点。

图 C.11 所示的为一个在 DACPAC.LIB 中用于实际 A/D 转换的 ISR 程序。请注意程序 Super Timer() 仅仅是设置标志,同时使实时时钟准时。通过限制 ISR 工作任务的数量,我们可以消除一些由中断引起的问题。

```

#define TIMER0          0x40      /* I/O mem locations for 8253 */
#define TIMER_CTRL      0x43

#define TIMER_CLOCK     (long)1192755    /* Hz crystal for 8253 */
#define BIOS_TIC        (double)18.2   /* in ticks per second */

int old_timer_call;

void SetupTimer(frequency)
int frequency;
{
    long             divisor;
    int   data;

    old_timer_call = (int) (((double) frequency) / BIOS_TIC);

    divisor = TIMER_CLOCK / ((long) frequency);

    outportb(TIMER_CTRL, 0x34); /* timer 0, mode 3, LSB and MSB */
    data = (int) (divisor & 0xFF);
    outportb(TIMER0, data);
    data = (int) ((divisor >> 8) & 0xFF);
    outportb(TIMER0, data);
}

```

图 C. 10 设置 8253TIMER0 为一定采集率的程序

```

void interrupt SuperTimer ()
{
    static int super_counter = 0;

    TIME_OUT = TRUE;

    if (++super_counter == old_timer_call)
    {
        OldTimer ();           /* execute old timer approx. 18.2 */
        super_counter = 0;     /* times per second */
    } else {
        outportb(0x20, 0x20); /* interrupt acknowledge signal */
    }
}

```

图 C. 11 中断服务过程和设置标志及调用实时时钟(18.2 次/s)

C. 4 自编接口软件

用户从 UW DigiScope 程序中所发现的最为普遍的应用就是, 增加一个新的接口装置。这里将对怎样在 UW DigiScope 程序中增加一个不同的内部插卡以设计接口提供一些指导。

C. 4. 1 编写接口程序

C. 2 节介绍了如何在 Turbo C 语言与内部插卡之间建立接口的方法, 所有的 I/O 插卡都将装上一些接口程序, 和/或一本手册以及编写这些程序的一些例子。这些使 DigiScope 与内

部插卡相匹配的技术也使接口与插卡相匹配。附录 B 介绍了 DigiScope 的一些功能,这些功能能与各种各样的 I/O 装置进行接口联系。例如,若想设计一个与内部插卡连接的接口,则需要用到 IDAC.OBJ 的程序,这些程序的功能如图 B.4 所示。

图 C.12、图 C.13 和图 C.14 所示的是 IDAC.OBJ 文件中的程序范例。图 C.12 所示代码

```
/* Global Variables */

unsigned int BaseAddress = 200; /* base address for RTD card */
unsigned int IRQline = 3;
static char AD_TIME_OUT;
static char DA_TIME_OUT;
static int OldMask;
static int NUM_CHANNELS; /* used to remember how many */
                         /* channels to read */

static char OPEN=NO;
static char TIMED=NO;

/* RTD board addresses */

#define PORTA      BaseAddress + 0
#define PORTB      BaseAddress + 1
#define PORTC      BaseAddress + 2
#define PORT_CTRL  BaseAddress + 3

#define SOC12     BaseAddress + 4
#define SOC8      BaseAddress + 5
#define AD_MSB    BaseAddress + 4
#define AD_LSB    BaseAddress + 5

#define DA1_LSB   BaseAddress + 8
#define DA1_MSB   BaseAddress + 9
#define DA2_LSB   BaseAddress + 0xA
#define DA2_MSB   BaseAddress + 0xB
#define UPDATE    BaseAddress + 0xC
#define CLEAR_DA BaseAddress + 0x10

#define TIMER0    BaseAddress + 0x14
#define TIMER1    BaseAddress + 0x15
#define TIMER2    BaseAddress + 0x16
#define TIMER_CTRL BaseAddress + 0x17

char Iopen(DataHeader_t * Header)

{
    int register i,j,in, out;
    int delay, num_channels;
    char buf[20];
    int mask;
```

图 C.12 打开内部模拟 I/O 卡的开始程序

```

AD_TIME_OUT =FALSE;
DA_TIME_OUT =FALSE;

Header->volthigh = 5.0; /* initialize header data structure */
Header->voltlow = -5.0;

Header->resolution = 12;
if (Header->num_Channels > 8)
    Header->num_Channels = 8;
NUM_CHANNELS = Header->num_channels;
Header->num_samples = 0;
Header->data = NULL;

/* initialize internal card */
/* PORT C low is input */
/* PORT C high is output */

outportb (PORT_CNTRL,0x91);

/* set gain and offset for each channel during Iget routine */
/* type should be set by user */
/* for now the gain is always set to one */

for (i=0;i<Header->num_channels;i++) {
    ChanOffset (Header,i) = 0.0;
    ChanGain (Header,i) =1.0;
}

if (! CheckTimer()) return (NO); /* This routine is used to */
/* check if the board is installed */

/* setup timing function */
if (Header->rate !=0) {
    disable (); /* only perform timing if rate is */
                /* nonzero */
    ISetupTimer (Header->rate); /* setup 8253 timer */

    OldVector = getvect (IRQnumber); /* set interrupt vector */
    setvect (IRQnumber,Itimer);

    /* unmask interrupt mask */
    OldMask = inportb (0x21);
    mask =OldMask & ~ (1<<IRQline) & 0xFF;
    outportb(0x21,mask);
    enable ();
    TIMED=YES;
} else {
    TIMED=NO; /* rate is zero => don't use timing */
}
OPEN=YES;
return(YES);
/* Iopen () */

```

图 C.13 打开内部模拟 I/O 卡的结束程序

```

char Iget (DATATYPE * data)

{
    int msb, lsb, i;

    if (! OPEN) {
        DEVICE_NOT_OPEN ();
        return (NO);
    }
    if (AD_TIME_OUT || ! TIMED) {

        for (i=0;i<NUM_CHANNELS;i++) {
            /* read all channels requested */
            /* for now the gain is tied to 1 */
            outportb (PORTB,0x00 | i); /* select channel */
            outportb (SOC12,0); /* start a conversion */
            while (! (inportb (PORTA) & 0x80));
                /* wait for conversion to end */
                /* after EOC then read MSB and LSB */
            msb = inportb (AD_MSB) * 16; /* read in data from RTD card */
            lsb = inportb (AD_LSB)/16; /* the card should be in +/- mode */
            data [i] = msb + lsb - 2048;
        }

        AD_TIME_OUT = FALSE; /* Clear AD_TIME_OUT flag */
        return (YES);
    } else {
        return(NO);
    }
} /* Iget */

```

图 C.14 从内部卡上得到模拟数据的程序

用来启动装置并设置必要的定时及 I/O 功能, 应该记录下初始化后的内部数据结构“Header”的一些条款。

C. 4. 2 UW DigiScope 程序的新接口

DACPAC.LIB 文件中包含了所有 UW Digi Scope 的数据采集程序。为了用目前接口代替 I/O 卡, 只需在 IDAC.OBJ 有关程序中用 DACPAC.LIB 的程序更换。为了替换外部装置和虚拟装置, 须用 DACPAC.LIB 的相关程序分别替换 EDAC.OBJ 和 FAD.OBJ。另外应该包括附录 B 介绍的那些模块的功能, 否则至少将会导致一个编辑错误, 最坏的情况会导致时序混乱。如果你不想使用其中某个功能, 你可以简单地在其位置上插上一个虚设的功能, 如果想了解更多情况, 请查阅 UW DigiScope 程序上的 README 文件。

附录 D UW DigiScope 程序使用手册

UW DigiScope 程序给用户提供了一些有关数字示波器的基本典型功能,包括数据采集与存储、灵敏度调节控制、波形检测,等等。更为重要的是,该程序也是一个数字信号处理库,有一整套包括 FFT 和滤波器设计在内的内部函数,另外还有一系列特别的功能,包括 QRS 复波的检测、信号压缩和波形产生等。在此集中讨论 UW DigiScope 程序产生波形的功能和它的基本指令。学会将 UW DigiScope 程序装到硬盘上,这可参考 UW DigiScope 程序软盘上的说明。同时应该阅读盘上的 README.DOC 文件,以了解本书上未曾包括的有关 UW DigiScope 程序的附加信息。

D. 1 UW DigiScope 程序的简要介绍

为了运行这个程序,在 UW DigiScope 程序目录中敲入 Scope。在整个附录中,单词 Scope, digiscope 以及 UW DigiScope 是等价的。

虚线框为激活区。上面为从磁盘读出的 ECG 图,然后用推导算法处理 ECG 图并将波形放在下面的激活区。

D. 1. 1 主显示屏

图 D. 1 所示的为 UW DigiScope 的主显示屏。在屏幕左边是一个菜单,而在底部有一个指令行窗口。有两个显示窗口,实线所示的为激活窗口。该窗口可以用(A)ctive ch 菜单指令加以选择,通常在被激活的窗口上处理数据信息。屏幕顶端显示磁盘文件上读到的 ECG 图,而在最后一行则会显示用某种指定算法的 ECG 处理结果。

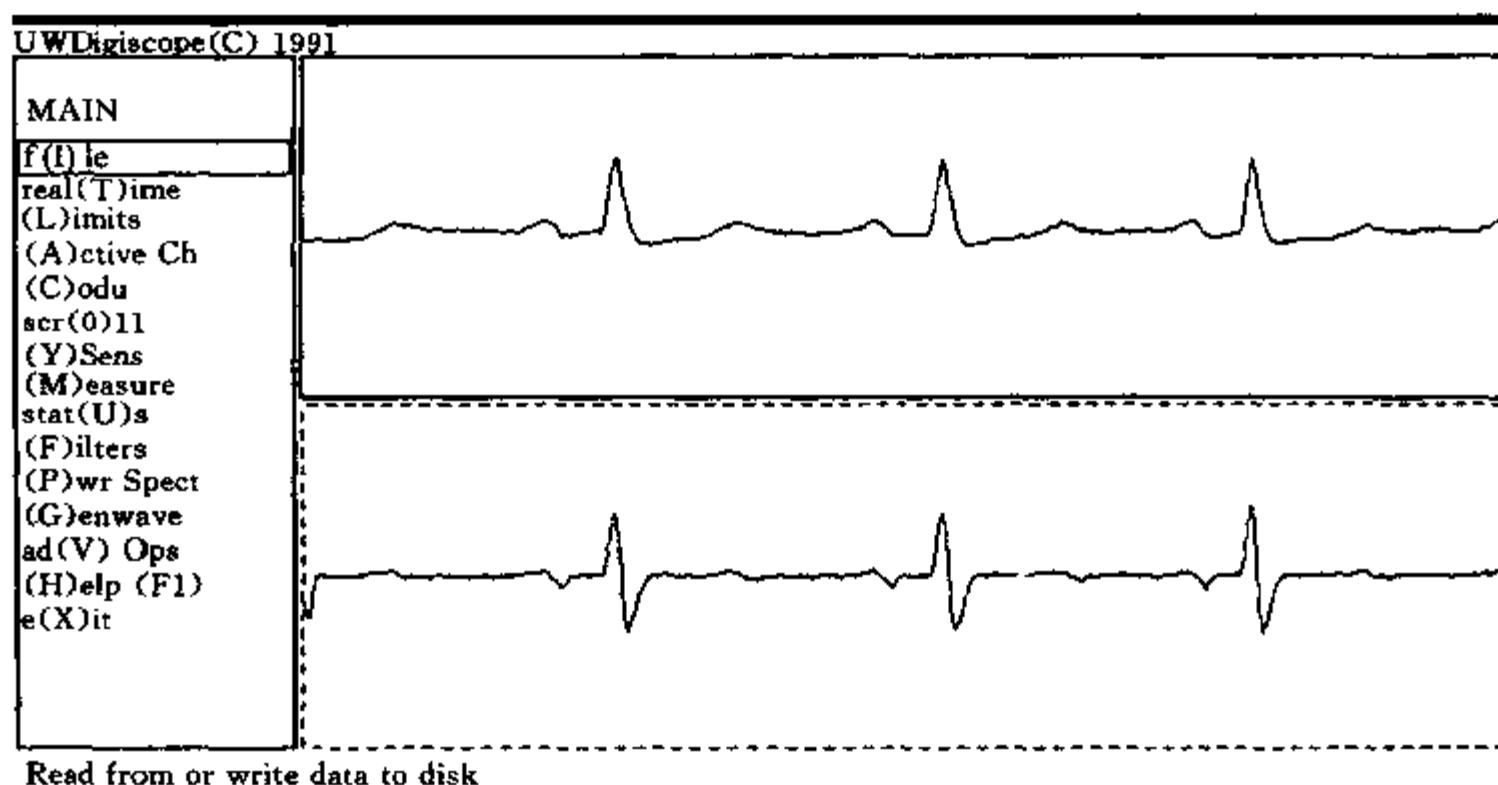


图 D. 1 UW DigiScope 的主屏

对 UW DigiScope 的操作通过使用菜单完成。“上”和“下”键控制选择框在菜单的上下移动。按回车键则选中。另外，键入菜单上每个指令括号中的字母，也可以执行该指令（例如，按 F 键，则立刻执行 (F)ilters 指令）。

在当前菜单下执行退出 e(X)it 指令，回到母菜单（即高一级菜单）的操作，既可以选择框套住 e(X)it，按回车完成，也可以按 RETURN 键，或 X 键，或按 ESC 键。实际上，在 UW DigiScope 的所有操作中都可用按 ESC 键来退出当前操作。而按 RETURN 键则可执行选择功能。

D. 1. 2 与 UW DigiScope 程序的通信

有时用户需要从键盘输入资料，这种资料的输入可以在屏幕底端窗口的指令行上完成，输入资料时同样可按 RETURN 键来结束，这样数据将被接受，或者按 ESC 键使用户中断数据输入的同时回到前一个屏幕。按 BACKSPACE 键，然后按 RETURN 键，可对 SCOPE 程序进行校正，也可以通过这个窗口的提示提供信息。

D. 2 各种功能的总结

图 D. 2 所示的为主菜单与其它菜单之间的关系。f(I)le 指令允许对磁盘数据文件进行读/写操作。real(T)ime 指令使你能够从磁盘文件，或者一个外部扩展摩托罗拉 68HC11 微控制器插卡，或者一个内部实时信号转换插卡中选择采样数据源。

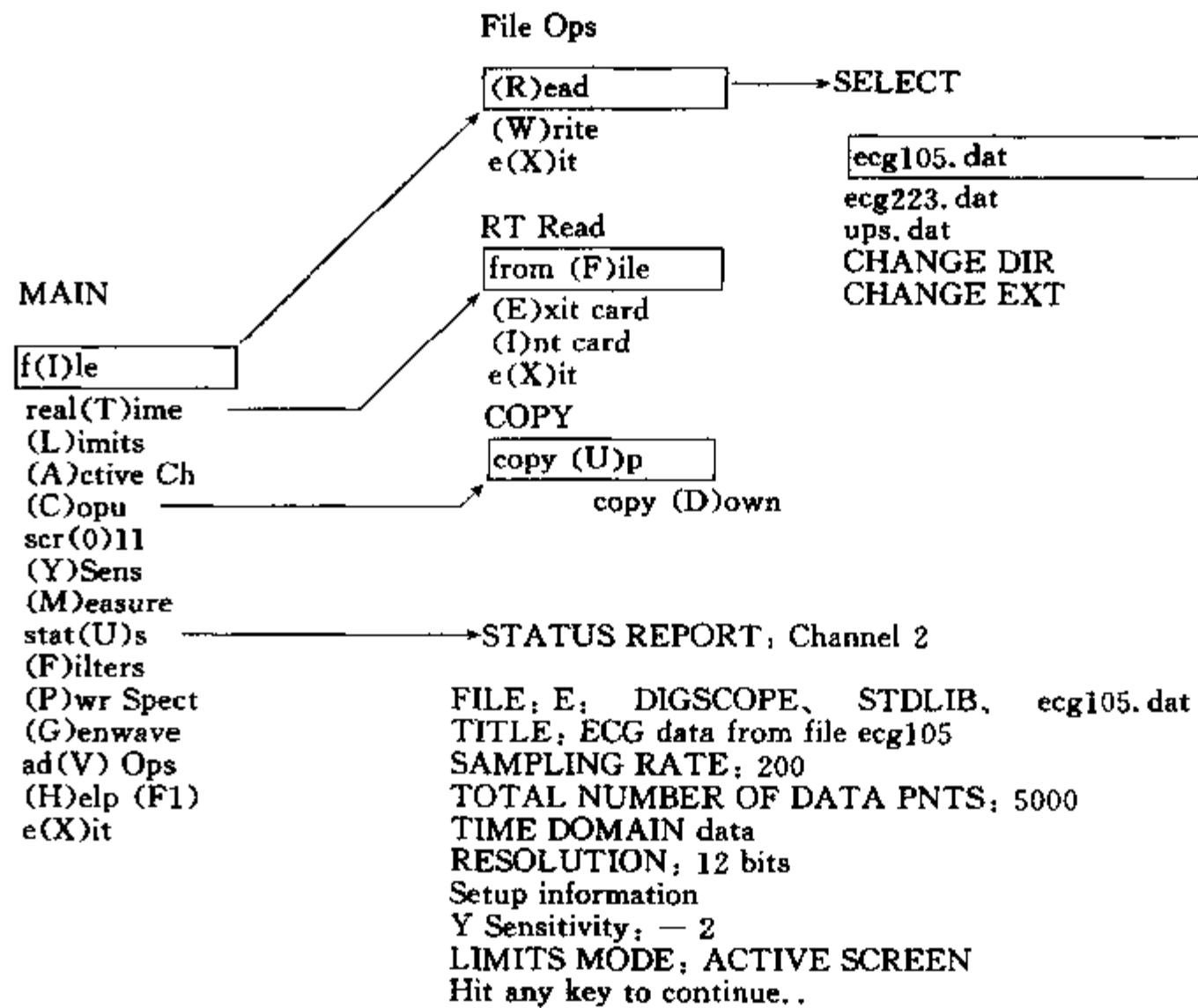


图 D. 2 主菜单的子菜单分支

(L)imits 指令使你可以选择对所有文件或对当前文件的一部分进行 SCOPE 操作。一旦取消限制，就能从当前文件下获得 512 个数据。最大的文件容量为 5120 个数据采样点。当用 f(I)le(W)rite 指令向磁盘写入一个文件时，将显示仅用当前数据（512 点）还是整个文件的指

示。通过 Scroll 指令,你可以用方向键查阅整个文件,同时选择文件中的哪一个 512 个点的部分显示当前状态,(C)opy 指令可将当前窗口信息复制一份。

为了调整激活通道的幅值,选择(Y)sens 指令,并通过按键盘上“上”或“下”箭头键来增加或减少通道的灵敏度。这种功能的操作如同在一个释放程序上的灵敏度操作一样。(M)easure 指令将两个光标重叠在一起,这两个光标位于你可以用箭头键移动的激活通道的波形上。在显示屏的底部,将会有一个窗口显示这两个光标的幅度和时间周期。

stat(U)s 指令提供了一个关于当前数据显示特征的全屏显示。这些信息将在数据文件写入磁盘时记录在文件的开头。

(P)wr Spet 指令可计算和显示一个信号的功率谱,而(H)elp 指令主要用于解释每个指令的功能。无条件选择 e(X)it 指令,将使操作从主菜单退回到 DOS 状态下,而 e(X)it 指令则可以使你回到一个当前菜单之前的任何子菜单中。

图 D. 3 所示的为一个从主菜单(F)ilter 指令中分支出来的滤波器设计的菜单。任何使用这些工具的设计数据都将保存在一个磁盘文件上,并生成标号。为设计而提供的工具分为三个部分:FIR、IIR 及整数滤波器。(F)IR 和(I)IR 中的每一个功能都提供了四种设计技术,而(I)nteger 指令则全面支持扩展新文件。最后设计的文件被存储在寄存器中,这样若选择(R)unfilter 指令,则在激活窗口下进行波形的滤波过程。(L)oad filter 指令将装入并执行前一个被保存在磁盘上的滤波功能。

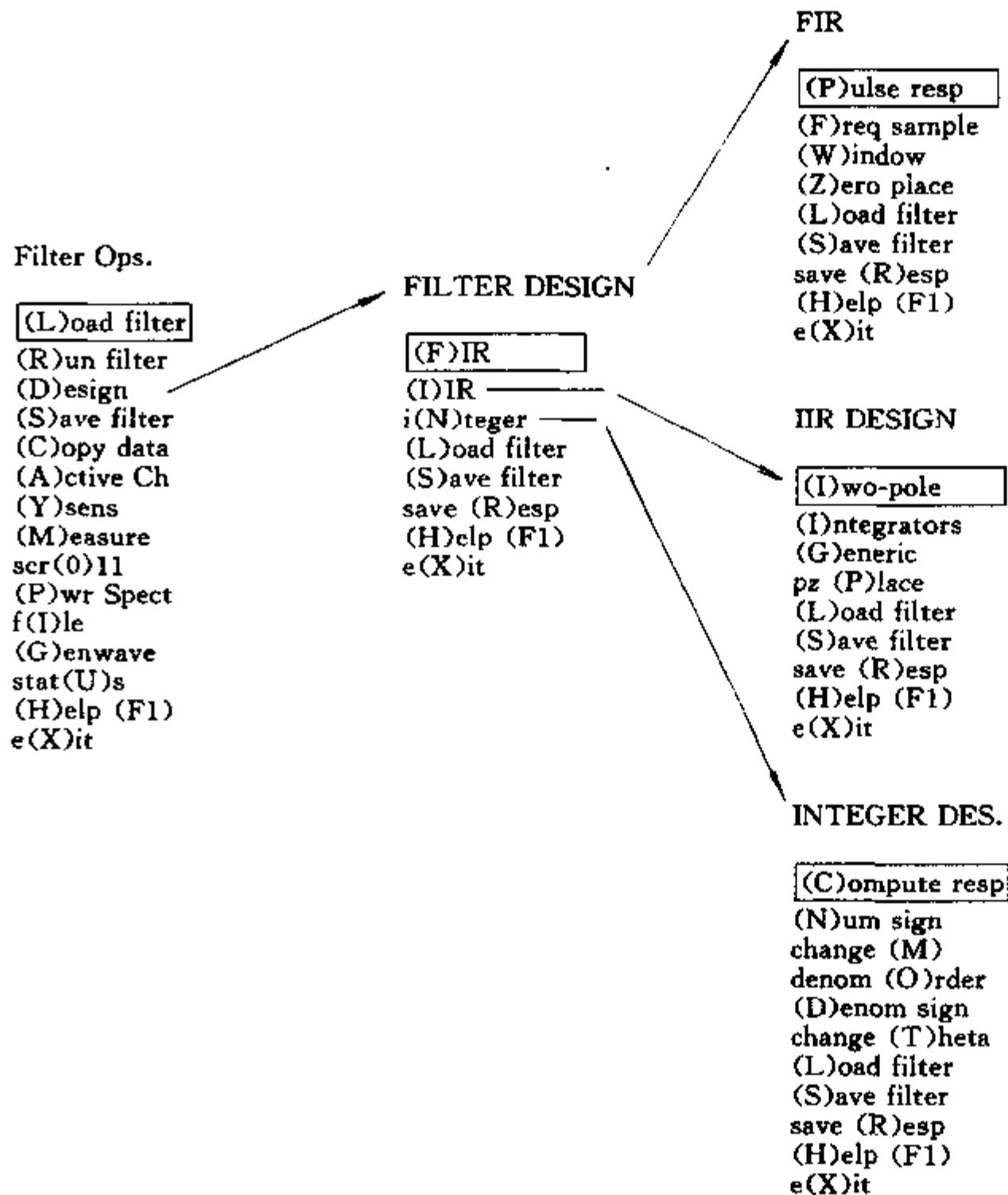


图 D. 3 滤波器设计方法

图 D. 4 所示的为用于设计两极 IIR 滤波器的设计窗口。在这个例子中,首先选择一个子菜单中的通频带滤波器,并且为了极点的配置而设定半径及幅角,程序 SCOPE 将显示零极点坐标,反馈回来一个包含脉冲、幅值和相位的响应信号,以及用于实现滤波器的差分方程。通过选择 e(X)it 指令或者按 ESC 键可回到上一个屏幕(见图 D. 3),执行(R)unfilter 指令可观察这个滤波器对信号的处理效果。

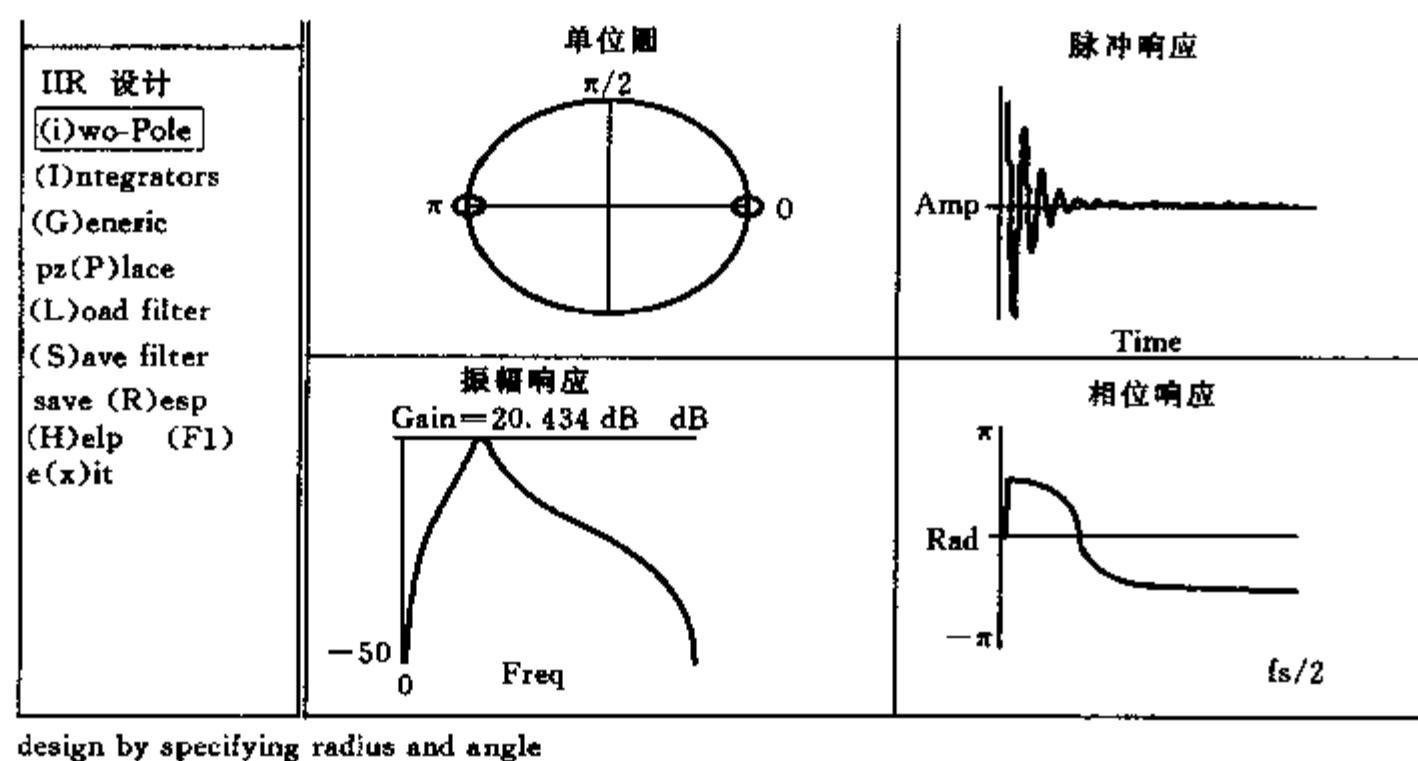


图 D. 4 UW DigiScope 设计窗口

注意将幅度响应调整为 0dB,并且记录滤波器的增益。高增益滤波器或级连滤波器可以引起信号数据的溢出。这常表现为不连续地输出波形,而且信号数据是用 16 位数来表示的。如果你有一个 12 位的数据文件(即值为大约 ± 2000),并且让这些数据通过 40dB(即放大 100 倍)的滤波器或者级连滤波器,就会产生一个 ± 200000 范围内的数字,这将会产生超出 16 位整数(大约 32000)的算术溢出,并且将输出错误的波形。为了防止这个问题的出现,在高增益滤波器前要加一个可衰减 40dB 的称为 atten 40.fil 的全通滤波器。这样做的缺点是,信号转换位将会丢失,因为原始信号幅度将会按 100 的系数下降,这样将导致信号衰减到原来的 $\frac{1}{100}$,从而丢失转换信的信号位。

图 D. 5 所示的为从主菜单选择 ad(V)ops 指令产生的专用函数。这些高级功能为频率分析、互相关分析、QRS 复波测试、数据压缩以及采样分析等。

(F)requanal 指令用于通过两个可移动的光标来对波形信号的一部分(称为样本)进行功率谱分析。该功能描述了零填充和/或数据窗口的效果。一个样本可通过光标加以选择,而零填充位于光标之外,任何偏移都被零填充去掉。可以给已获得的样本提供一个窗口,在这个被选定的窗口中样本的中间部分集中显示了全部信息,而在样本的边缘则衰减为零。re(S)tore 指令则将原始缓冲器重新装入样本。

(C)orrelation 指令对上一通道样本与信号进行互相关分析,并将结果存在下一通道。如果在选出样本之后不读入新的文件,则样本会与取到该样本的数据进行互相关处理(一种自相关方式)。在一个样本被选定后,输出将被集中在选定的样本附近。

(Q)RS detect 指令允许对采用本书中所描述的 QRS 测量算法的各种内部时变滤波器进

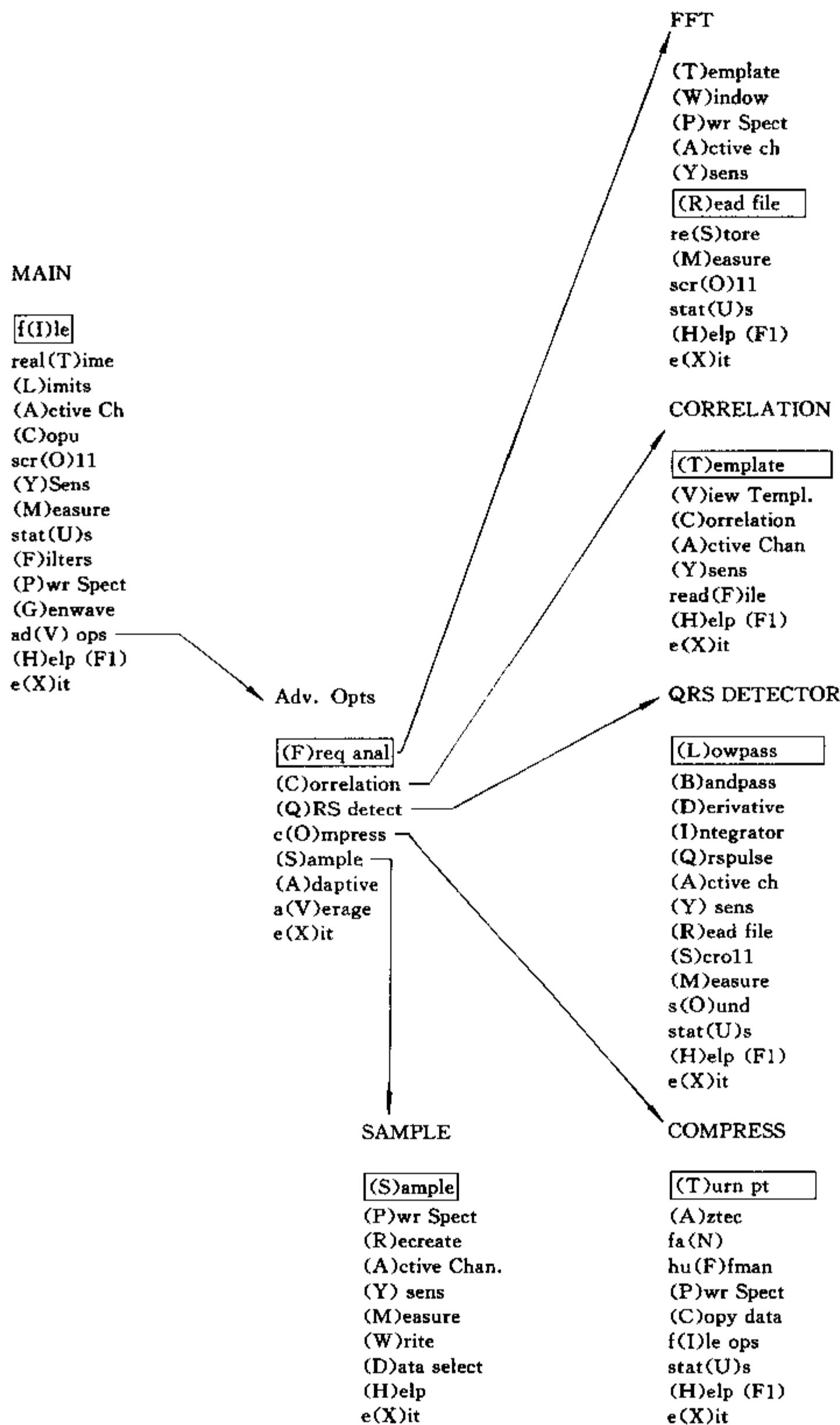


图 D. 5 高级选择菜单

行检查。这种算法适用于 200 次/s 采样率的信号。QRS 测量适合于整个文件数据，它将重新设置滤波器阈值和内部数据存储器，并可多次重复。这样，就可以在数据信号很短的情况下也能看到一个静止波形。

C(O)mpress 指令可通过选择转折点算法、AZTEC、FAN 和 Huffman 编码算法对波形数据进行压缩。这种数据压缩技术能计算上一通道信号的近似值，这种算法仅对当前数据进行操作。转折点算法通过对两个严格精确的点的推算减少数据点个数，而 FAN 和 AZTEC 算法则要求有一个阈值，该阈值在信号衰减与失真的临界点时将终止计算。用户在提示下输入算法所要的基本值，最好是数据列的一部分。在当前状态下，Huffman 编码算法是一种不失真的算法，该算法可以随着数据值出现的频率建立一个数据表，这个可查阅的数据表在用(R)un 指令压缩之前应该用(M)ake 指令进行计算。(R)un 指令在精确压缩之后再解压，则会显示压缩比，当制作这个表时，可以利用的是，这个表通常将减少数据的个数，并且增加数据压缩比。若在运行(M)ake 指令时数据值过大，则这个表会被改成 8 位数表，超出这个范围的值将被放在备用设备中并加上前缀，这样数据的减少将是微不足道的，最为有利的是这样做将会使数据减少到一个很低的范围（通过一个，例如 atten40.fil 的滤波器），然后重复这个过程。

(S)ample 指令用于研究如何通过不同的速率进行采样，以及用三种不同技术重组波形的采样过程。这个模块使用内部产生的三种波形之一，所以可以对已知数据进行再度采样。若数据以 5000 次/s 速率产生，那么可以选 1~2500 次/s 采样速率再度采样。如果要进行已采样数据的功率谱分析，则该程序可以产生一个临时的缓冲区存储以某一采样速率产生的 512 个点。也就是说，虽然显示的数据是根据内部采样区而来的，但是功率谱并不是根据显示的数据来计算的，而是根据对模拟信号以特定采样速率得到的数据来计算的。

(A)daptive 指令描述自适应滤波的基本原理，而 a(V)erage 指令则用于时间上的信号平均技术。

图 D.6 所示的为用于产生波形的 GENWAVE 函数。将一定时的 ran(D)om 信号和 60Hz 噪声进行综合可用于测试滤波器。用 t(E)mplate 指令可以产生(S)ine、(T)riangle、s(Q)uare、ECG 以及其他周期信号。图 D.7 所示的为 9 个综合产生的正常或异常 ECG 信号。图 D.8 所示的为综合应用该函数产生的两个信号。

<p>GENWAVE</p> <p>(S)ine (T)riangle s(Q)uare t(E)mplate (F)req/BPM sample (R)ate (P)ts/(B)eats ran(D)om 60-H(z) (A)mplitude resol(U)tion (G)enerate (W)rite file (H)elp (F1) e(X)it</p>	<p>GENWAVE PARAMETERS</p> <p>ECG1 Heart rate is 120 BPM Sampling rate is 200Hz There are 5.1 beats There is 0% random and 0% 60-Hz noise resolution is 8 bits amplitude of waveform is 100%</p>
---	--

ECG or other TEMPLATES from default.tpl

图 D.6 波形产生

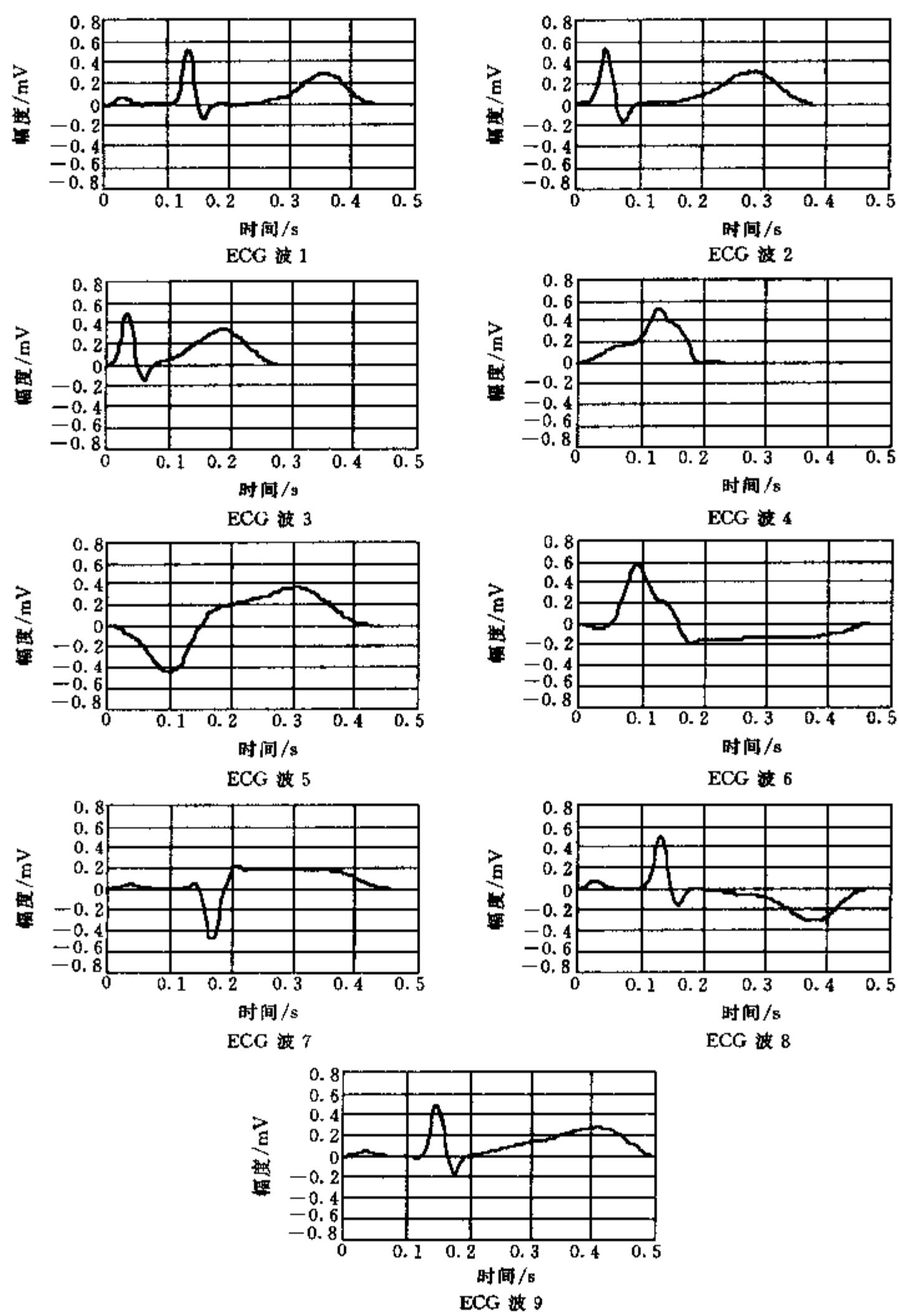


图 D.7 用 GENWAVE 产生 ECG 波形的样本

样本 ECG WAVE1 是正常的,其余异常

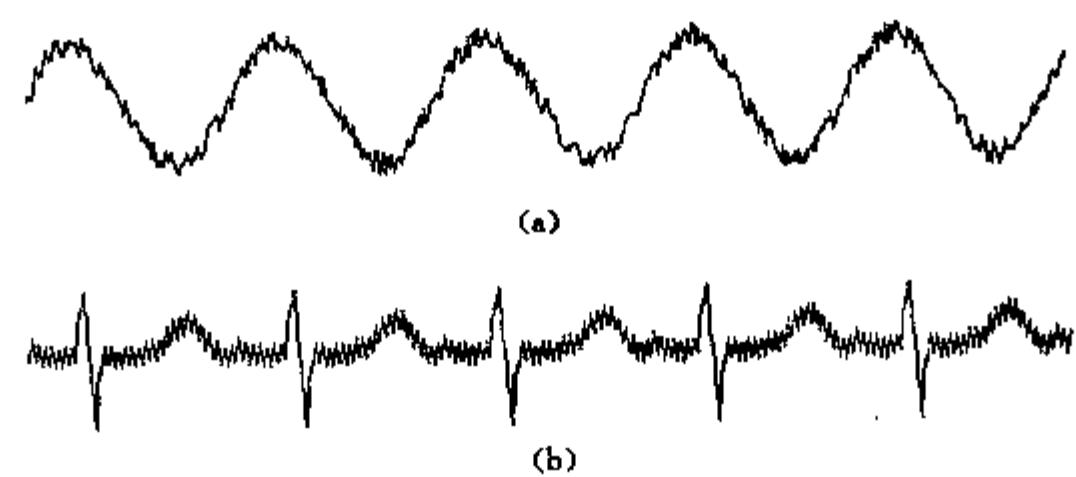


图 D.8 用(G)enwave 产生的波形
(a)含 20% 随机噪声的 2Hz 正弦波; (b)含 5% 随机噪声的 ECG 波形 1

附录 E 信号发生器

在设计数字滤波器和信号检测器算法的过程中,测试它们的性能是很重要的。测试过程经常包含将信号输入到滤波器或检测器算法中,然后观察输出端的结果。对生物医学应用来说,能采用 ECG 和 EEG 这样的像正弦、三角和方波的基本物理信号作输入信号更好。但是物理信号经常不是很容易得到的,这使滤波器和检测器算法测试很难进行。然而,可以使用计算机和软件生成理想的生理信号,下面介绍的是怎样使用 UW DigiScope 软件的 GENWAVE 功能来生成标准信号和生理信号的方法。

E. 1 信号发生方法

模型方程法和波形模板法是两种合成信号的基本方法。

在模型方程法中,信号的数学模型用于计算构成信号的数值序列。信号生成方法非常灵活,能够产生任何频率和幅值的信号。然而,要对 ECG 这样的特殊信号进行准确的建模可能很困难。如果信号很复杂,模型方程就很复杂,计算时间也就很长。在 UW DigiScope 软件中,采用方程方法来产生正弦、方波和三角波信号。

波形模板法简单、灵活,而且是生成复杂、重复信号的快速方法。波形模板很容易通过模/数转换来保存一个周期的采样数据,从而获得所需的生理信号。这种方法适用于任何形式的重复信号,无须考虑所要生成信号的复杂度。在 UW DigiScope 软件中,使用模板法来进行 ECG 信号的仿真。

E. 2 信号发生器程序(GENWAVE)

GENWAVE 是 UW DigiScope 软件中的程序模块,只需调用(G)enwave 命令即可生成各种信号波形。用户可以设置周期数、采样频率、波形重复率、噪声幅度和位分辨率。得到的信号形式取决于采用的波形模板。新波形模板可以加到缺省模板文件中,也可保存在单独的模板文件中。下面介绍如何生成模板文件以及怎样使用 GENWAVE 程序。

E. 2. 1 生成模板文件

GENWAVE 提供了两个生成模板的程序:ASC2TPL. EXE 和 TPL2ASC. EXE。ASC2TPL. EXE 将正确格式的 ASCII 文件转化成缺省的模板文件。TPL2ASC. EXE 将二进制模板文件(如缺省的模板文件 DEFAULT. TPL)转化成 ASCII 文件,以便这些文件可以用编辑器或 WORD 软件进行编辑。其中,有 9 个 ECG 信号模板的缺省模板文件是 UW DigiScope 软件中附带的。将新的模板加到缺省模板文件中的方法是:首先将缺省模板文件转化成 ASCII 文件,然后将新模板数据加到 ASCII 文件末尾,最后将 ASCII 文件转化回二进制的缺省模板格式。在转化成二进制时,ASC2TPL. EXE 自动将波形定为 12 位的,以获得最大

分辨率。图 E.1 所示的是 ASCII 模板文件格式。

Waveform Name	(80 characters max.)
Waveform ID number	(1-12 are reserved; max. number 50)
Number of Datapoints	(32767 max.)
Sample Rate of Signal	(in Hz. - 32767 max.)
Data(1)	
Data(2)	(the first and the last datapoints must be the same baseline values)
.	
.	
.	
Data(Number of Datapoints)	
(Repeat above for each waveform template)	

图 E.1 ASCII 模板文件格式

要将缺省的模板文件转化成 ASCII 文件时, 可键入下列命令:

TPL2ASC DEFAULT.TPL <ASCII File Name>

其中, ASCII File Name 是存储缺省模板的数据文件名, 文件名 DEFAULT.TPL 是所读波形模板的二进制文件名。这种格式命令的一个例子为

TPL2ASC DEFAULT.TPL

这个命令把二进制模板文件 DEFAULT.TPL 转化成名为 ASCTEMP.ASC 的 ASCII 模板文件。两个文件名都必须具体化, 且除了 DEFAULT.TPL 外的名字都可以作为不同的二进制模板文件名。

将 ASCII 模板文件转化成缺省的二进制模板文件时, 首先要重命名当前的缺省模板文件, 接着键入下面的格式命令:

ASC2TPL <ASCII File Name> DEFAULT.TPL

其中, ASCII File Name 是将被转换成缺省模板文件的文件名。新的 DEFAULT.TPL 文件必须放在 DIGSCOPE / STDLIB 目录下, 以便 DIGSCOPE 软件可以使用。这种格式命令的一个例子为

ASC2TPL ASCTEMP.ASC DEFAULT.TPL

这个命令把 ASCII 模板文件转化成名为 DEFAULT.TPL 的二进制模板文件。两个文件名都必须具体化, 且除了 DEFAULT.TPL 外的名字都可以作为不同的二进制模板文件名。

一个模板文件最多可以有 50 个波形模板。每个波形必须有一个波形序列号 ID。在一个给定的文件中, ID 号应从一开始, 并随每个模板逐个递增。波形名长度可达 80 个字符。为确保开始模板和末尾模板的幅度匹配, 第一个和最后一个的波形值必须相同。图 E.2 所示的是 ASCII 模板文件的一个范例。

E. 2. 2 GENWAVE 功能

信号发生器 GENWAVE 生成波形时, 要用到如下几个参数。如果信号被存储为一个文件, 这些参数则放在文件头(默认值在方括号中)中。

- (1) 输出文件名(带 3 位扩展名的 8 位字符) [WAVEFORM.OUT]。
- (2) 输入文件名(带 3 位扩展名的 8 位字符) [DEFAULT.TPL]。
- (3) 以次/s 为单位的输出采样频率(最大 10 000) [500]。

(4) 以周期/min 为单位的输出波形频率 (最大 10 000) [60]。

Normal ECG	(Waveform Name)
4	(Waveform ID)
157	(Number of Datapoints)
360	(Samplerate)
0	(Data1)
0	(Data2)
0	(Data3)
624	(Data4)
1560	(Data5)
.	.
.	.
.	.
920	(Data155)
25	(Data156)
0	(Data157 Last)
PVC	(Waveform Name)
5	(Waveform ID)
83	(Number of Datapoints)
360	(Samplerate)
0	(Data1)
0	(Data2)
43	(Data3)
324	(Data4)
260	(Data5)
.	.
.	.
.	.
320	(Data1)
35	(Data82)
0	(Data83 Last)

图 E. 2 ASC II 模板文件范例

(5) 数据点数 (最大 5 000) [512]。

(6) 噪声所占满尺度的百分比 (0~±100) [0]。

① 随机噪声。

② 60Hz 噪声。

(7) 波形序列号 ID (1~50) [50]。

(8) 以位为单位的满尺度分辨率 (3~12) [8]。

由于正弦、三角和方波信号已经内置在程序中了，即使 DEFAULT.TPL 文件找不到了，GENWAVE 也总能够产生这些波形。既然 GENWAVE 读的总是存在的文件，就可望生成新的缺省文件，即将旧的 DEFAULT.TPL 文件以另外的名字存储，而把你自己的新文件重命名为缺省的模板文件名 DEFAULT.TPL。

附录 F 有限长寄存器效应

设计数字滤波器时可以使用硬件来达到某些特殊目的，也可使用计算机，如 PC 机来实现一般目的。两种方法都涉及使用有限长数据寄存器(FLR)，它只能表示有限数目的数值。在第三章，讨论了模/数转化和由于能被表示的幅值数目有限的量化效应。在信号处理器或微处理器上设计自己的数字滤波器时，必须考虑由 FLR 效应带来的问题。本附录讨论溢出特性、舍入噪声、有限循环、缩放比例和由于定点寄存器带来的 I/O 偏差。本附录还将比较浮点和定点寄存器各自的功能优势。

F. 1 量化噪声

在数字滤波器中使用定点寄存器时，它只能存储有限数目的整数数据。这种表现形式导致两种结果：(1) 构成滤波器的状态变量只能代表最小量化的整数乘法；(2) 存在寄存器可以以 1 对 1 表示的最大量化值。第一个效应称为量化，第二个则称为溢出。

量化误差可能在任何有乘法和加法的定点寄存器中产生。例如，如果想实现 FIR 滤波器，则输出应该是带权的各个输入之和。最后的加法值只能代表最小值的准确度。一个位长为 $B+1$ 的寄存器所能表示的最小量化值为

$$q = \Delta 2 - B \quad (\text{F. 1})$$

式中， q 通常称为量化步长。可以通过选择 Δ 的值来任意定义 q 。如果想设计整数滤波器，则 Δ 的值为 $2B$ 时， $q=1$ 。选择大的 Δ 的优点是可以扩大所能表示的值的范围。然而，增大了量化步长，其结果也就失去了精度。图 F. 1 所示的是 3 位寄存器舍入的量化特性。

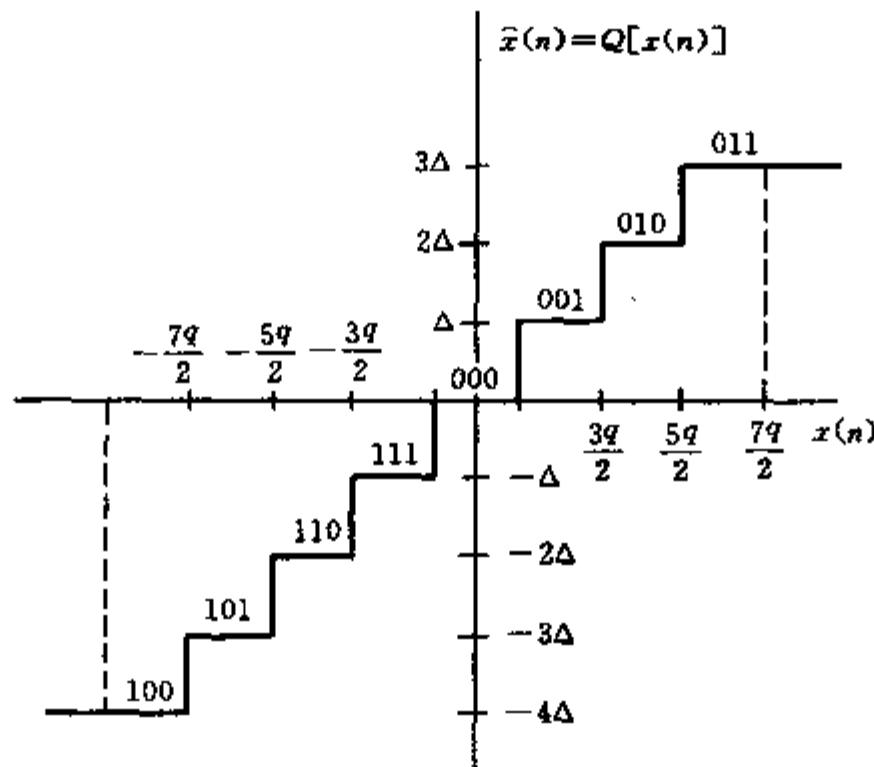


图 F. 1 3 位寄存器的二进制实现舍入量化效应

实际数据 $x(n)$ 和有限二进制表示之间的误差为：

$$e = x(n) - Q[x(n)] \quad (\text{F. 2})$$

这就是由 FLR 所带来的量化误差。在任何 A/D 转换器的数据提取中，也要产生量化误差，但 A/D 转换器的量化是硬件问题，而 FLR 量化是由于软件的局限性所致。

F. 1.1 舍入

实现量化的一个方法是将真实值向最近的量化水平进行舍入。量化误差限在正负 $q/2$ 之间。因此，量化在功能上就等同于在最初的真实值上加上范围在正负 $q/2$ 之间的随机噪声。可以把这些噪声看成是随机产生的，噪声产生的概率密度函数(PDF)如图 F. 2 所示：

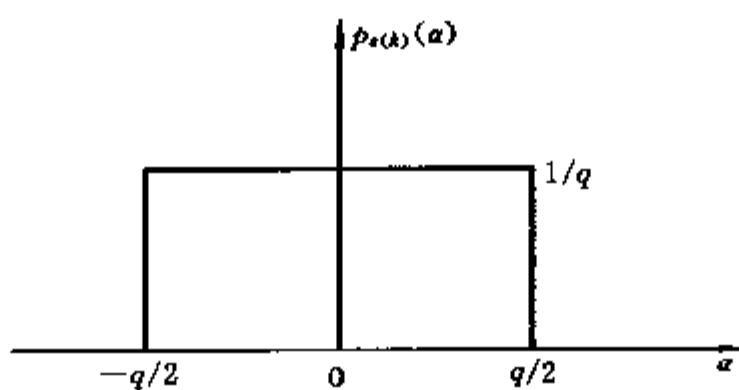


图 F. 2 舍入产生的量化误差的概率密度函数

理论研究和数字仿真表明，这个模型对于相对宽的频率谱输入和宽的概率密度函数来说是准确的。因此，输入不能长时间地保持在同一量化水平上。可以计算量化的乘法和加法带来的噪声的功率理论值，即

$$\delta_e^2 = E[e(k)^2] = \int_{-q/2}^{q/2} \frac{\alpha^2}{q} d\alpha \quad (\text{F. 3})$$

F. 1.2 截断

量化的另一方法是对值进行截断，或者将值减小到可表示的水平。这与舍入有点近似，除了这种量化误差的概率密度函数外还有一个漂移平均。截断噪声范围在 $-q \sim 0$ 之间。然而，不管是舍入方法还是截断的方法，它们之间均没有内在的优势。舍入的方法更可取是因为它具有零均值特性。既然噪声功率比噪声方差理论计算更简单，那么这种方法也就容易得多。

F. 2 有限循环

如果数字滤波器对输入进行放大，则状态变量中便存在内部增溢。如果寄存器的累加值超过可表示的最大二进制值，就会发生溢出。图 F. 3 所示的是几种函数表现形式。饱和溢出是一个严格的限制器的特性。两种补余溢出是环绕式效应的特性。溢出后为零是能简单地将输出强制为零的特性。

上述的三种方法在提供整个舍入误差和连续的溢出可能性上都有各自折衷的优势。饱和溢出方法的输出与真实值(虚线表示)最接近。然而，将输出保持在最大的量化水平对下一次滤波器的溢出有很大的潜在影响。两种补余溢出方法是两种补充表示的自然方法。最大正表示值比最小负表示值小一位，但是两种补余将产生最大的舍入误差。这种方法的输出落在量化范围的任何位置的可能性是均等的。而溢出后为零的方法则介于饱和方法和两种补余方法之间。

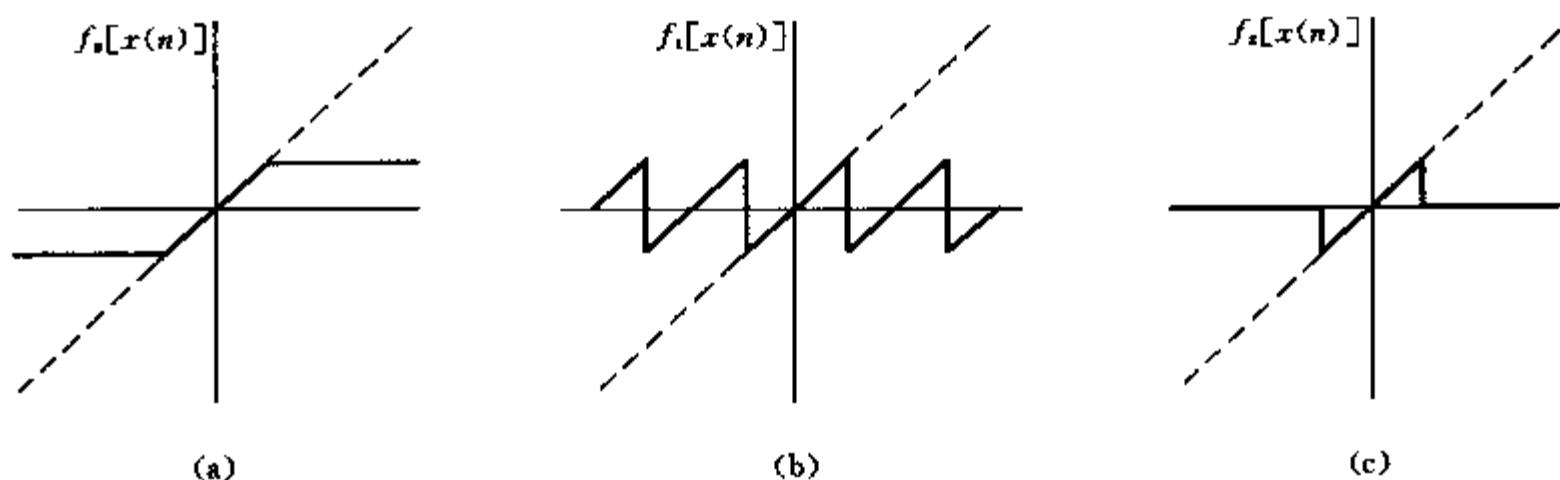


图 F.3 溢出特性
(a)饱和;(b)两种补余;(c)溢出后为零

是一种折衷的方法。

F. 2.1 溢出摆动

从溢出特性的不同形式可以看出,重复溢出可能发生在三种情况中。当溢出连续发生时,由于存在溢出摆动,紧跟初始化的非零输入的输出不会平均为零。溢出摆动的含义是输出不再依赖于输入。

图 F.4 所示的是一个二阶滤波器利用两种补余溢出,再利用舍入量化,从可能的状态空间跳到溢出状态的例子。在这个例子中,通过设置输入为零,简化了典型的状态空间方程。滤波器的方程可用下式表示:

$$x(k+1) = Ax(k) + Bu(k) \quad (\text{F.4})$$

$$y(k) = Cx(k) + Du(k) \quad (\text{F.5})$$

式中, A 是描述当前状态变量 $x(k), x(k+1)$ 的矩阵。如果 A 的特征值大于单位圆,滤波器便是不稳定的,会发生溢出。对零输入来说, $x(k+1)$ 在状态空间有几个特殊的区域会超出范围,在这些区域会发生溢出摆动。非零输入有时会改变路径,以至于跳出这个循环。然而,与状态空间的整个范围比较,输入不会很大。

F. 2.2 死区效应

溢出不是有限寄存器效应发生的唯一有限循环形式。在具有连续输入的递归无限冲击响应滤波器(IIR)的输出实际上应当连续衰减情况下,可通过量化产生稳定状态的输出。这个问题由于舍入误差使得状态变量停留在同一状态空间产生的结果。因此,存在一个死区,在死区,连续输入可防止滤波器产生线性输入-输出曲线特性。这在发生的舍入误差的大小上与溢出摆动的危险不同,但在误差输入响应方面仍然是个问题。

下面列举这种一阶滤波器的例子,令

$$y(nT) = -0.96y(nT - T) + x(nT)$$

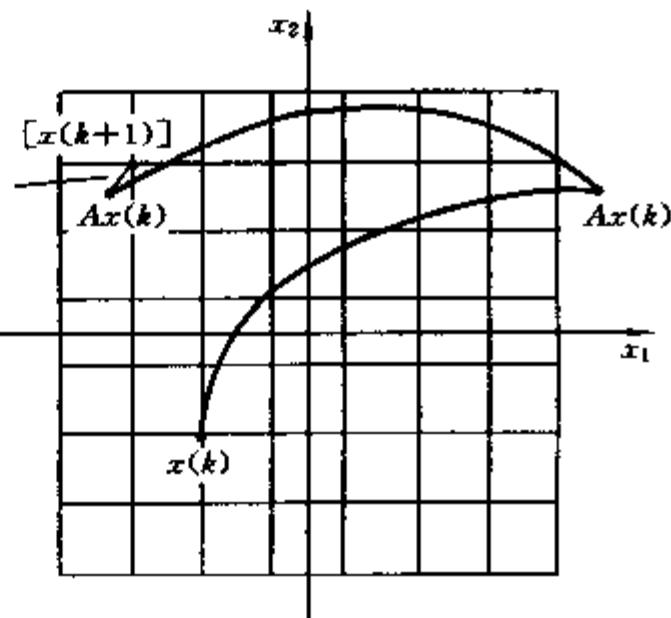


图 F.4 显示增溢和舍入的二阶数字滤波器的状态空间描述

其中,

$$x(0)=13, \quad x(nT)=0 \quad (n>0)$$

显然,不经过量化的实际输出会衰减到零,但如果确保输出向最近的整数舍入,就会得到下面的输入-输出特性。

n	$y(nT) = -0.96[y(nT-T)]$	$[y(nT)]$
0	13	13
1	-12.48	-12
2	11.52	12
3	-11.52	-12
4	11.52	12

这种滤波器具有两个有限循环。有一个周期的有限循环的滤波器具有连续保持在同一栅格内的状态变量。

解决死区效应的一个办法是,向状态变量 $x(n)$ 加入少量的白噪声。然而这就意味着将永远不能获得滤波器真正的稳定状态。另一个方法是,使用幅度截取而不是舍入。但同样的问题是,截取在消除旧的死区的同时会引入新的死区。

F. 3 缩放比例

有几种方法可以避免有限循环带来的灾难性的影响。一种是,增加 Δ 使得状态空间栅格能覆盖更大的区域,然而,这将增加外来噪声的能量。有限循环最常见的影响是,使滤波器具有太大的增益。缩放比例是一种可减小溢出可能性,且仍能保持滤波器传递函数,而不是在限制量化误差方面折衷的办法。

图 F. 5 所示的是怎样对节点 v 进行缩放比例以使滤波器不产生溢出的例子。通过传递函数 $F(z)$ 将 u 输入到节点,然后除以一个常数 β ,这样对节点进行缩放以减小溢出就成为可能。为保证整个滤波器的输入-输出特性,必须在后续的传递函数 $G(z)$ 中加上增益 β 。

有两种不同的缩放准则,即 L1 和 L2 缩放准则。第一种准则适用于将输入的绝对值进行舍入的情况。第二种准则适用于保持输入的能量。两种准则分别为:

$$\beta = \|f\|_1 = \sum_{l=0}^{\infty} |f(l)| \quad (\text{F. 6})$$

$$\beta = \delta \|f\|_2 = \delta \left[\sum_{l=0}^{\infty} f^2(l) \right]^{1/2} \quad (\text{F. 7})$$

参数 δ 可以人为选择,以满足滤波器的理想要求。如果输入是零均值,且在单位内变化,则 δ 可看成在节点 v 中表示的标准偏差数。 $\delta/5$ 则被认为是很保守的值。

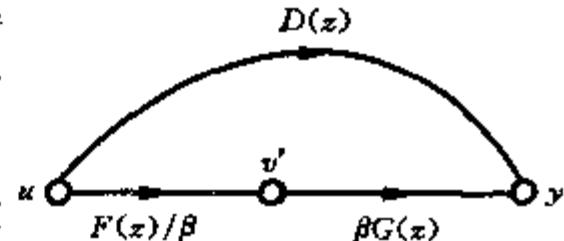


图 F. 5 对节点 v 进行缩放比例的状态变量描述

F. 4 IIR 滤波器中的舍入误差

在 IIR 滤波器中,舍入误差和量化误差均与状态变量有关。在输出端所积累的影响称为舍入噪声。理论上,可以对每种舍入误差进行建模,将它们看成是在 $q/2/12$ 内变化的附加的白噪声,以此来估计舍入对整体的影响。如果节点 i 到输出的单位脉冲响应序列是 g_i ,并且输出经过累加器后再进行量化,则整体舍入噪声可估计为:

$$\delta_{\text{tot}}^2 = \frac{q^2}{12} \left\{ \sum_{i=1}^n \|g_i\|^2 + 1 \right\} \quad (\text{F. 8})$$

式(F. 8)的累加有时称为滤波器的噪声增溢。选择不同的滤波器结构型式可改进噪声增溢,使之与二阶幅度相等。直接型式滤波器所产生的噪声增溢比最小噪声滤波器的高,因为后者使用了适当的缩放,对 g_i 进行了改变,从而减小了舍入噪声量。

F. 5 浮点寄存器滤波器

尽管浮点寄存器的指数性能使之能提供更宽的区域,但它表示的状态数还是有限的。状态空间栅格不再是均衡的,而是取决于状态向量和初始值之间的距离。图 F. 6 所示的是利用指数范围,获得可能的状态数间的更宽的边界的方法。

浮点与定点有两个方面的差别:(1)加法存在量化误差;(2)输出舍入噪声的变化与状态变量数成正比。尽管浮点寄存器可以大大提高滤波器输入的范围,但量化所带来的舍入误差却远远大于定点寄存器的误差。

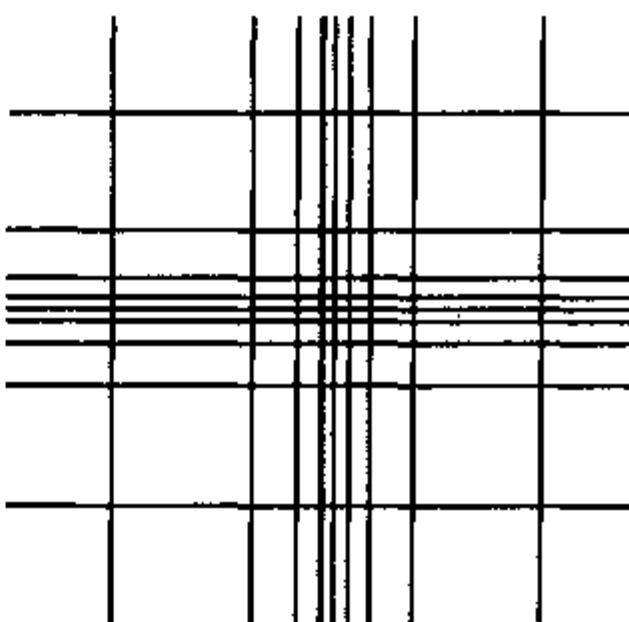


图 F. 6 浮点寄存器的状态空间栅格

F. 6 总 结

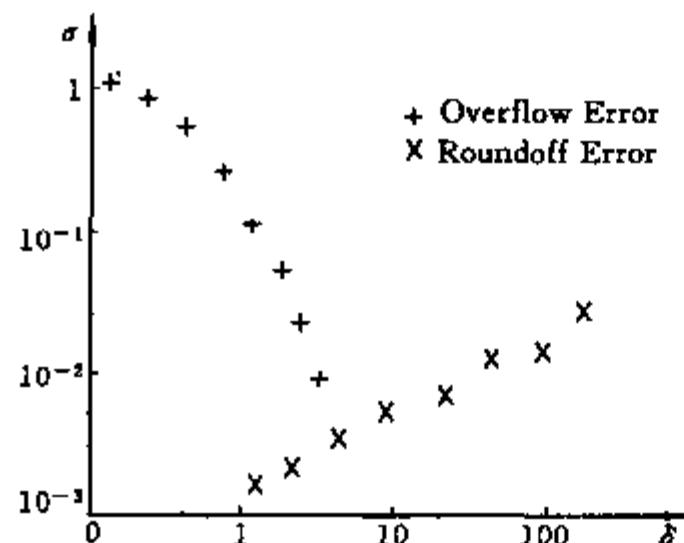


图 F. 7 整体误差输出中溢出和舍入误差的比较

在许多实时应用中数字信号处理都需要使用有限寄存器(FLR)。这里总结了使用 FLR 所带来的影响和误差形式。不管寄存器是定点还是浮点,也不管是否对内部节点进行缩放,这些影响取决于舍入的形式和寄存器的溢出特性。

图 F. 7 所示的是对滤波器的整体误差与变量缩放度进行的比较。在没有进行缩放时,除非对输入进行舍入,否则极可能会产生溢出。当增大缩放因子后,溢出发生的可能性则小得多,但由于节点寄存器的动态范围减少,由量化所带来的舍入误差则开始增加。为减小输出的整体误差就必须找到一个可以减小两种误差的折衷方法。

F. 7 实验:数字滤波器中的有限长寄存器效应

编写使用舍入特征的子程序,并对 8 位整数寄存器的 2 种补余溢出特性进行仿真。试用下列传递函数实现一个高通无限冲击响应滤波器(IIR):

$$H(z) = \frac{1}{1 + 0.5z^{-1} + z^{-2}}$$

用正弦作为输入,找出开始发生溢出的幅值点。检查滤波器这种输入所得到的输出。看看溢出是否反映了 2 种补余响应特性。

附录 G 商用 DSP 系统

当前,各种商业数据采集的硬件和软件都可以在市场上买到。它们大多数以功能完善的数据采集系统的形式出现,支持各种硬件卡,而且具有数据分析和显示功能。一个完整的数据采集系统基本上由三个模块构成:采集,分析和显示。

G. 1 数据采集系统

G. 1. 1 采集

采集数据的 4 种常见方法是:(1)使用 RS-232 串行接口,(2)使用 IEEE488(GPIB)并行设备接口,(3)使用 VXI 总线,(4)使用 PC 总线的可插数据采集卡。

1. RS-232 接口

这种方法由与简单设备有关的串行通信协议组成,这些设备有数字体温计,面板表和数据记录器等。这种方法适用于以采样频率低于 1KB/s 来控制采集远端的数据。既然 RS-232 接口是许多计算机的标准,这种方法便不再需要额外的硬件。

2. IEEE 488(GPIB)并行接口

许多复杂的实验和工业设备,如数据记录器和数据示波器,都配备有 GPIB 接口。它们通过最长为 20m 的电缆进行设备的通讯,使用 8 位并行协议,最大数据传送率为 2MB。这种接口既支持控制又支持数据采集。IEEE 488 使用 ASCII 命令集(Baran, 1991)。

3. VXI 总线

这种总线是高性能设备卡上的体系结构,适用于复杂设备。这种体系结构于 1987 年引入,已用于在架栈仪器系统中减小物理体积,实现严格定时,在多个设备间实现同步,得到更快的传输率。这个标准可以实现超过 10MB 的传输率。

4. 即插式数据采集板

数据采集板直接插到特殊的计算机(如 PC 机和 Macintosh 机)中。这种方法成本低,而且可得到适中的性能。这些卡通常包括很多功能,如 A/D 转换,D/A 转换,数字 I/O 和定时器操作。它们具有 8 到 16 位的分辨率,采样频率可高达 1MB/s。对于实现一般目的的数据采集是很灵活的。

G. 1. 2 分析和显示

数据分析是将原始数据转化成有用的信息。这本书原理上就是关于数据分析的。许多软件包都提供了如数字信号处理,统计分析和曲线匹配操作之类的子程序。

数据显示则是用一种直观的,有意义的格式将数据呈现给用户。除了用图形表示数据外,也可以将数据记录在袋状图上,在宽范围的打印机或绘图仪上产生有意义的报表。

G. 2 DSP 软件

现在的趋势是使用能提供数据采集、分析、显示于一体的使用商业 DSP 软件。这里我们要讨论商业上用于一般目的的即插式数据采集板的软件。由于这种数据采集方法具有很大的灵活性,这方面的软件具有很好的市场,需求量也高。

软件性能是随设备供应商所提供的重点和价格而变化的。如有些公司以模块的形式销售他们的软件,用户可以有选择地购买。

一些通用的商业 DSP 软件包括以下内容:

- (1) 支持很宽领域的信号转换卡。
- (2) 包含 FFT、卷积、低通、高通和带通滤波器在内的 DSP 算法库。
- (3) 能进行数据的存档。更复杂的软件允许导出数据到 Lotus 123, dBase 和其它的通用分析程序。
- (4) 宽范围的采样率。
- (5) 图形显示和菜单或者驱动用户界面的图标。
- (6) 用户可编程。
- (7) 支持用 C 语言,BASIC,ASCII 命令进行高级编程。
- (8) 可制定生成报表和图形(如彩色控制,自动或手动缩放)。

这里有几个有趣的软件包,它可以让读者了解商业 DSP 软件提供的内容。

Tektronix 开发的 DSP 软件包是为 Tektronix 数字转换器和数字示波器以及 PEP 系列系统控制器或 PC 控制器设计的。它的工具箱提供了 200 多种功能,包括积分和差分,脉冲测量,统计,窗口化,卷积和相关,任意长短阵的前项和后项傅里叶变换,任意波形的正弦波成分,插值和抽取,标准波形生成(正弦,方波,随机噪声),FIR 滤波器生成。

DSP 公司开发的 DADISP 提供了在 80286 和 80386 机上依保护模式运行的版本。可对 16 位的地址进行寻址。DADISP 被看作是交互式的电子表格。电子表格取代单一的数字来表示波形,信号,或者图形。每个单元格都由包含全部波形的窗口所表示。例如,如果窗口 1(W1)包含一个信号,而窗口 2(W2)包含公式 DIFF(W1) (有关时间的差分),经过差分的信号便显示在 W2 中。如果 W1 中的信号发生变化,自动重新计算差分并显示在 W2 中。它也包括设备分配和管理单元。下面举例说明:如果 W1 是电压测量,W1 将反映伏特数,而 W2 则反映每秒钟的伏特数。允许最多 100 个窗口进行缩放,滚动和移动光标。由于 DADISP 能够自动地在磁盘和内存之间对数据进行编页,因此数据点数只受磁盘空间的限制。

英汉对照索引词

A

accelerometer	加速度测量仪
adaptive filter	自适应滤波器
AHA database	AHA 数据库
alarm	警报器
algorithm	算法
alias	别称
alphanumeric data	字母数字编码数据
Altair 8800	阿泰尔 8800
ambulatory patient	能行走的病人
amplifier	放大器
ECG	心电图
schematic diagram	原理图
instrumentation	仪器
micropower	微功率
operational	可操作的
amplitude accuracy	振幅精度、运算的
analog filter	模拟滤波器
analog multiplexed	模拟多路开关
analog-to-digital	模数转换器
antialias filter	(see filter, analog)
apnea	呼吸暂停
Apple II	苹果 II
application software	应用软件
application specific integrated circuit	应用特殊集成电路
architecture	结构
array processor	阵列信号处理器
arrhythmia	心律不齐
monitor	监护仪
artificial neural network(ANN)	人工神经网络
ASIC	美国标准码
assembly language	汇编语言
autocorrelation	自相关
automation	自动调节器
averaging signal	平均信号

AZTEC algorithm (Amplitude Zono Time Epoch coding)幅值区
时间分段编码

B

Babbidge	磁泡
BASIC	BASIC 语言
Basic Input/Output system	基本输入/输出系统
battery power	电池电源
benchmark	基准
binomial theorem	二项式定理
biomedical computing	生物医学计算
BIOS (Basic Input/Output system)	基本输入/输出系统
bit-reversal	位取反
bit-serial processor	位串行处理器
Blackman-Tudey estimation method	Black Turkey 估计方法
blind person	盲人
block diagram language	块图表语言
blood pressure	血压
blood pressure tracings	血压跟踪
body core temperature	体核温度
bradycardia	心动过速
brain	大脑

C

C language	C 语言
C++ language	C++ 语言
C-language program	C 语言程序
bandpass filter	带通滤波器
bit-reversal computation	位取反计算
convolution	卷积
cross correlation	互相关
derivative	求导数
Fan data reduction algorithm	扇形数据压缩算法
Hanning fitter	海宁滤波器
High-pass fitter	高通滤波器
low-pass fitter	低通滤波器
moving window integrator	移动窗口积分器
power spectrum estimation	功率谱估计
real-time Hanning fitter	实时海宁滤波器

TP data reduction algorithm	TP(转折点)数据压缩算法
trapezoidal integrator	梯形积分器
two-pole band pass filter for QRS detection	QRS 检测的双极点带通滤波器
CAD(Camputer Aided Design)	计算机辅助设计
Calories	卡
CALTRAC	一种用小型加速度测量仪来监测人体的运动的计算的仪器
cardiac	心脏的
atria	心房
attioventricular(AV)node	房室结
depolarization	去极化
equivalent generator	等效发生器
output	输出
pacemaker	起搏器
purkinje system	普尔顿系统
repolarization	复极化
sinoatrial(SA)node	窦房结
spread of excitation	兴奋传导
ventricles	心室
ventricular muscle	心室肌
cardiogenic artifact	心脏伪差
cardiology	心脏病学
cardiotachometer	心率计心脏血流计
cascade	级联
catheter	导管
circular buffer	循环缓冲器
clinical engineering	临床工程
clinician	临床医生
closed loop control	闭环控制
CMOS	互补型金属氧化物半导体(缩写词)
CMRR(common mode rejection ratio)	共模抑制比(缩写词)
coefficient sensitivity	系数灵敏度
common mode rejection ratio	共模抑制比
compressed spectral array(CSA)	压缩谱阵列
compression	压缩
concurrent execution	并发操纵
converter	转换器
analog-to-digital	模/数转换
analog multiplexer	模拟多路开关
counter	计数器

dual-slope	双斜率
flash	闪烁
parallel	平行
successive approximation	逐次近似法
tracking	跟踪
charge scaling	容量标度
differential	微分
digital-to-analog	数/模转换
design	设计
interpolation method	内插法
dynamic properties	动态性能
gain error	增益误差
integral linearity	线性积分
monotonicity	单调性
offset error	偏移误差
resolution	分辨率
setting time	置定时
static properties	静态性能
voltage reference	参考电压
voltage scaling	电压标度
convolution	卷积
diret	直接
fast	快速
correlation	相关性
autocorrelation	自相关
coefficient	系数
cross correlation	互相关
C-language program	C 语言程序
VW Digiscope command	威斯康辛大学数字化显示器命令
CORTES algorithm	(Coordinate Reduction Time Encoding System 坐标压缩时间编码系统)算法
CP/M	用于微处理器的控制程序
CRYSTAL	晶振电路
cutoff frequency	截止频率

D

DADisp	
damping factor	衰减系数
data compression	数据压缩
data reduction	数据压缩

adaptive coding	自适应编码
AZTEC algorithm	算法幅值零相位编码 AZTEC
compressed signal	压缩信号
facsimile	复制
Fan algorithm	扇形压缩算法
Huffman coding	哈夫曼编码
frequent set	频率装置
Infrequent set	非频率装置
translation table	转换表
loss less algorithm	无丢失算法
reduction ration	压缩比
residual difference	剩余差分方程
run-length encoding	扫描宽度编码
SAPA algorithm	SAPA 算法
signal reconstruction	信号重建
significant-point-extraction	特征点提取
turning point algorithm	转折点算法
C-language program	C 语言程序
UW Digscope commands	
dBase	数据库
deadband effect	死区效应
decimation in time	时间抽取
derivative	微分
desktop computer	终端计算机
DFT	离散傅里叶变换
diabetic	糖尿病患者
diagnosis	诊断
diathermy	透热疗法
differentiator	微分器
Digiscope	软件名称
digital signal processing	数字信号处理
Motorol DSP56001	摩托罗拉 DSP56001
Texas Instruments TMS	得克萨斯仪器 TMS
digital-to-analog converter	数模转换器
dipole current source	双极电流源
Dirac delta function	迪拉克 delta 函数
DOS(disk operation system)	磁盘操作系统
DRAM	动态 RAM
DSP(digital signal processing)	数字信号处理
DSP chip	数字信号处理器芯片

E

ECG(electro cardiogram, electracardiography)	心电图
EEG(electroencephalogram)	脑电图
EEPROM	电可擦除的可编程 ROM
EGG (electrogastrogram)	胃电图
Einthoven	等边三角形
EIT(electrical impedance tomography)	电阻抗断层 X 线照相术
electrical stimulator	电刺激器
analysis of	分析
body surface	体表
clinical	临床的
electrical	电极
fetal	胎儿的
Interpretation	解释
average beat	平均心率
decision logic	判断逻辑
feature extraction	特征提取
knowledge base	知识库
measurement matrix	测量矩阵
median beat	中速心率
statistical pattern recognition	统计模式识别
waveform recognition	波形识别
isoelectric region	等电位区
J point	J 点
monitoring	监护
power spectrum	功率谱
ST lever	ST 段偏移
standard 12-lead clinical	临床标准的 12 导联
electrocardiography	心电图
acquisition cart	采集车
amplifier	放大器
Schematic diagram	时间表
arrhythmia analysis	心律不齐分析
clinical	临床
database	数据库
electrogram	电图
forward problem	前向问题
Holter recording	Holter 记录
late potential	晚电位

lead coefficients	导联系数
lead system	导联系统
objective of	客观性
QRS detester	QRS 检测器
adaptive threshold	自适应阈值
amplitude threshold	幅度阈值
automata theory	自动理论
bandpass filter	带通滤波器
crosscorrelation	互相关
first and second derivation	一阶和二阶导数
Searchback technique	回溯技术
template matching	模板匹配
recording bandwidth	带宽记录
rhythm analysis	节律分析
ST-Segment analysis	ST 段分析
transfer coefficients	传输系数
Vector	矢量
electrode	电极
ECG	心电图
nonpolarizable	非极性的
offset potential	偏移电位
polarizable	极性的
silver-silver chloride	银-氯化银电极
electroencephalogram	脑电图
electrogastrogram	胃电图
electrolyte	导电膏
electromyogram	肌电图
electrosurgery	电外科
ENIAC	电子数字积分计算机
Euler's relation	欧拉相关
evoked response	诱发电位
auditory	听觉

F

facsimile technology	传真技术
Fan algorithm	Fan 算法
fast Fourier transform(FFT)	快速傅里叶变换
butterfly graph	蝶形图
fibrillation	纤维性颤动
fiducial point	可靠点

FIFO	先进先出
fitter	滤波器
adaptive	自适应的
analog	模拟
antialias	去假频
bandpass	带通
bandstop	带阻
differentiator	微分器
high-pass	高通
integrator	积分器
low-pass	低通
notch	陷波
QRS detector	QRS 检测器
cascade	级联
comb	梳状
digital	数字的
adaptive	自适应的
advantage	优点
all-pass	全通
amplitude response	幅度响应
band-reject	带阻
band pass	带通
integer	整数
bilinear transform	双线性变换
derivative	求导的
least-squares polynomial	最小平方多项式
three-point central difference	三点中心差分
two-point difference	两点差分
difference equation	差分方程
feed back	反馈
finite impulse response	有限冲激响应
frequency sampling design	频率采样设计
Hanning	海宁
high-pass	高通
infinite impulse response(IIR filter)	无限冲激响应
integer coefficient	整系数
integrator	积分器
rectangular	矩阵的
Simpson's rule	辛普逊法则
trapezoidal	不规则四边形的

learning	学习
least squares polynomial smoothing	最小二次多项式平滑
low-pass	低通
minimax design	最小最大设计
moving average	移动平均
moving window integrator	移动窗口积分器
nonrecursive	非递归的
notch	陷波
parabolic	抛物面的
phase response	相位响应
recursive	递归的
rubber membrane concept	橡胶膜概念
second derivative	二阶导数
smoothing	平滑
stability	稳定性
time delay	时延
transfer function	传递函数
transform table design	转换表设计
transversal	反相
two-pole	双极
window design	窗口设计
gain	增益
finite element resistivity model	有限元阻抗模型
finite-length register effect	有限长寄存器影响
FIR filter	FIR 滤波器
firmware	微程序语言
floppy disk	软盘
fluid loss	流失
FM modem	频率调制解调器
foldover frequency	折叠频率
Forth	第四
FORTRAN	FORTRAN 语言
Fourier transform(also FFT)	傅里叶变换
inverse discrete-time	逆离散时间

G

galvanometer	电流计
Gibb's phenomenon	Gibb 现象
GPIB	IEEE 488 总线接口
graphical user interface(GUI)	图形用户接口

gray-scale plot

灰度图

H

hardware interrupt structure	硬件中断结构
health care	健康护理
hearing aid, digital	听力辅助,数字
heart	心脏
disease	疾病
monitor	监护仪
rate	速率
high-level language	高级语言
Holter	为一种可携带的监护仪
real-time	实时
recording	记录
scanning	扫描
tape recorder	磁带记录仪
home monitoring	家庭监护
Huffman coding	哈夫曼编码
human-to-machine interface	人机接口

I

IBM PC	国际商用机器公司的个人计算机
IBM PC/AT	IBM 个人计算机与兼容机
icon	插图
Icu(intensive care unit)	集中护理单元
IDTFT(Fourier transform Inverse Discrete-time)	反傅里叶变换
IEEE 488 bus	电气与电子工程师协会,488 总线
IIR filter	无限冲激响应
implanted pacemaker	植入式起搏器
infusion pump	灌输泵
insensate feet	无知觉的脚
integrator(see filter, digital)	积分器
Intel	Intel 公司微处理器型号
4004	1971 首先推出 4 位微处理器
80286	
80386	
8080	1973 推出 8 位微处理器
8086 18088	
8088	

80C86

i486

microprocessor

微处理器

intensive care unit

集中护理单元

interactive graphics

交互性图示

K

Kirch hoff ' s voltage law

基尔霍夫的电压定律

L

L'H ? pital's Rule

L'Hôpital's 法则

LabVIEW

Labwindows

LAN(local area network)

局域网

Laplace transform

拉氏变换

laptop PC

LCD display

液晶显示

lead

导联

augmented limb

加压导联

coefficient

系数

frontal limb

额侧的肢体

limb

肢体

orthogonal

正交的

redundant

压缩

lead system

导联系统

12-lead

12 导联

Frank

人名

monitoring

监测

standard clinical

临床标准的

VCG

心电矢量图

learning filter

学习滤波器

Least Mean Squared algorithm(LMS algorithm) 最小均方差算法

Lempel-iv-Welch algorithm(LZW algorithm) LZW 算法

limit cycle

限制循环

linear phase response

线性相位响应

Lisa

线路阻抗稳定网络

local area network—LAN

局域网

M

MAC time	乘和累加时间
Macintosh	苹果公司生产的一种型号的计算机
MAGIC	一种与医学 CAD 相似的程序
magnetic resonance imaging	磁共振成像
math coprocessor	数学协处理器
MathCAD	数学计算机辅助设计
MatLAB	一种计算机语言
medical care system	医学护理系统
medical history	医学历史
medical instrument	医学仪器
Mflops	每秒运算一亿万个浮点数
microcomputer	微型计算机
microcomputer-based instrument	基于微机的仪器
microcomputer-based medical instrument	基于微机的医学仪器
microprocessor	微处理器
MIPS	每秒钟执行一百万条指令
MIT/BIH data base	数据库
model of cardiac excitation	心脏激励模型
modeling of physiological system	生理系统模型
Moore's Law	Moore 法则
MOSIS	金属氧化物半导体集成电路
Motorola	摩托罗拉
68000	
68040	
MSDOS	微软磁盘操作系统
multitasking operating system	多任务操作系统
multiuser	多用户

N

native code	原码
NEXT computer	NEXT 计算机
noise cancellation	噪声消除
noninvasive procedure	非侵入式工艺
nonrecursive	非递归
Nyquist frequency	奈奎斯采样频率

O

object-oriented software	面向对象的软件
--------------------------	---------

offset potential	补偿电位
op amp (operational amplifier)	运算放大器
open heart surgery	剖心外科手术
open loop control	开环控制
operating room	操作室
operating system	操作系统
operational amplifier	运算放大器
optical disks	光盘
OS/2	操作系统
over damping	过阻尼
overflow	溢出

P

pacemaker	起搏器
PACS	图片存档及通信系统
palmtop PC	掌上电脑
parallel processing	并行处理
parallel processor	并行处理器
parseval's theorem	Parseval 定理
Pascal	一种计算机语言
pattern recognition	模式识别
PDP-12	一种小型计算机型号
peak detector algorithm	峰值检测算法
percent root-mean-square difference (PRD)	均方差的百分比
personal supercomputing	个人超级算机
phase distortion	相位失真
phase response	相位响应
linear	线性
nonlinear	非线性
physiological signal	生理信号
pipeline	流水线
pneumo graphy	呼吸图表
portable arrhythmia monitor	便携式心律不齐监护器
portable device	便携装置
portable instrument	便携仪器
power spectrum estimation	功率谱估计
premature ventricular contraction	心室过早收缩
pressure distribution under the foot	脚下的压力分布
pressure sensor	压力传感器
probably density function	概率密度函数

punched card

穿孔卡

Q

Q of a band pass filter

带通滤波器的 Q 值

QRS defector

QRS 检测

quantization error

量化误差

R

RAM

real Time Devices

实时装置

read-time

实时

application

应用

computing

计算

digital filtering

数字滤波

environment

环境

patient monitoring

病人监测

processing

处理

signal processing algorithm

信号处理算法

recursive

递归

reduction

衰减

rehabilitation engineering

修复工程

Remez algorithm

Remez 算法

reprogramming

程序重调

residual difference

残差

resolution

分辨力

resonator

谐振器

 digital

 数字的

respiratory signal

呼吸信号

RF interference face

RF 接口

RISC processor

RISC 处理器

roll off

滑离

ROM

只读存储器

RS232 interface

RS232 接口

round off noise

舍入噪声

RTD ADA 2100 interface card

d RTD ADA 2100 接口卡

 language calls

 C 语言程序调用

 calling conventions

 调用规则

 configuration

 结构配置

 Installation

 安装

Interrupts
jumper and switch settings
rubber membrane concept
run-length encoding

中断
跳线和开关设置
橡胶膜概念

S

S-plane	S平面
sample-and-hold circuit	采样保持电路
sampling theorem	采样定理
Scope	范围、余地
searchback technique	回溯技术
sense of touch	触觉
integrated circuit	积分电路
smart	灵敏度
serial interface	串行接口
signal	信号
biomedical frequency, and amplitude ranges	生物医学的频率和幅度范围
discrete-time	离散的时间
generator	发生器
equation method	平衡法
template method	放样法
signal averaging	信号平均
signal conversion card	信号转换卡
external	外部
internal	内部
signal processing	信号处理
signal-to-noise ratio(SNR)	信噪比
simulation	仿真
68HC11EVB interface board	68HC11EVB 接口板
configuration	连接方式
68HC11EVBU interface board	68HC11EVB 接口板
cable design	电缆设计
calling conventions	调用规则
communication with PC	与 PC 交流
jumper settings	跳线装置
programming	编程
protecting analog input	保护模拟输入
software	软件
stability	稳定性
conditional	条件的

stimulation	大脑刺激
structured programming	结构化编程
supercomputer	超级计算机
systolic array	收缩期阵列

T

tachcardia	心动过速
tactile(touch) image	可触摸像点
tact or	向量
tap weight	开关重量
telemetry	遥测技术
temperature ,body core	温度,体核
therapeutic device	治疗装置
therapy	治疗
Thevenin equivalent	Thevenin 等效
resistance	阻抗
voltage	电压
tomography	断层成像
transducer	传感器
transportability	可运输性
trans portable	可携性
Turbo C	Turbo C 语言
turning point algorithm(TP)	转折点算法

U

ultrasound	超声
under damping	欠阻尼
unit circle	单位圆
unit impulse	单位脉冲
Unix	一种操作系统
Unstable digital filter	不稳定的数字滤波器
UW Digi Scope	威斯康辛大学编写的数字信号处理软件
advanced options menus	高级可选择菜单
command summary	命令一览
file header	文件头
file structure	文件结构
filter design menus	滤波器设计菜单
filter design screen	滤波器设计屏幕
installation	安装

main menu	主菜单
operation	操作
prototype device handlers	原型装置处理机

V

Vacuum tube	真空管
VCG(vector cardiogram)	心电向量图
vector ,cardiac	向量,心脏的
vector cardiogram	心电向量图
virtual I/O device	虚拟 I/O 装置
calling conventions	调用规则
virtual instruments	虚拟设备
visual computing language	可视计算语言
VLSI	超长规模集成电路
Voice-grade telephone network	声级电话网络
VXI bus	UXI 总线

W

wave front array	波前排列
Welch's method	Welch 方法
window	窗口
Gaussoian	高斯
Hamming	海明
Hanning	海宁窗
Kaiser	开塞窗
leakage	漏
rectangular	矩形的
triangular	三角形
windows	窗口

X

X-ray	X-射线
-------	------

Z

Z plane	Z 平面,定义
Z transform	Z 变换,定义
zero padding	零垫片
zero-order interpolation	零阶插入
zilog z-80	zilog 公司,z-80CPU

参考文献

第一章

- 1 Doumas T A, Tompkins W J and Webster J G. An automatic calorie measuring device. *IEEE Frontiers of Eng. in Health Care*, 1982, 4: 149—51
- 2 Friskin-Gibson S, Bach-y-Rita P, Tompkins W J and Webster J G. A 64-solenoid, 4-level fingertip search display for the blind. *IEEE Trans. Biomed. Eng.*, 1987, BME-34(12): 963—65
- 3 Hamilton P S and Tompkins W J. Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. *IEEE Trans. Biomed. Eng.*, 1986, BME-33(12): 1157—65
- 4 Hua P, Woo E J, Webster J G and Tompkins W J. Iterative reconstruction methods using regularization and optimal current patterns in electrical impedance tomography. *IEEE Trans. Medical Imaging*, 1991, 10 (4): 621—28
- 5 Kaczmarek K, Bach-y-Rita P, Tompkins W J and Webster J G. A tactile vision substitution system for the blind: computer-controlled partial image sequencing. *IEEE Trans. Biomed. Eng.*, 1985, BME-32(8): 602—08
- 6 Kernighan B W and Ritchie D M. The C programming language. Englewood Cliffs, NJ: Prentice Hall, 1978.
- 7 Mehta D, Tompkins W J, Webster J G and Wertsch J J. Analysis of foot pressure waveforms. *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1989, pp. 1487—88
- 8 Pan J and Tompkins W J. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.*, 1985, BME-32 (3): 230—36
- 9 Pfister C, Harrison M A, Hamilton J W, Tompkins W J and Webster J G. Development of a 3-channel, 24-h ambulatory esophageal pressure monitor. *IEEE Trans. Biomed. Eng.*, 1989, BME-36(4): 487—90
- 10 Tompkins W J. A portable microcomputer-based system for biomedical applications. *Biomed. Sci. Instrum.*, 1978, 14: 61—66
- 11 Tompkins W J. Portable microcomputer-based instrumentation. In Eden H S and Eden M. (eds.) *Microcomputers in Patient Care*. Park Ridge, NJ: Noyes Medical Publications, 1981, pp. 174—81
- 12 Tompkins W J. Digital filter design using interactive graphics on the Macintosh. *Proc. of IEEE EMBS Annual Conf.* 1985, pp. 722—26
- 13 Tompkins W J. Biomedical computing using personal computers. *IEEE Engineering in Medicine and Biology Magazine*, 1986, 5(3): 61—64
- 14 Tompkins W J. Ambulatory monitoring. In Webster J G. *Encyclopedia of Medical Devices and Instrumentation*. New York: John Wiley, 1988, 0001: 20—28
- 15 Tompkins W J and Webster J G. *Design of Microcomputer-based Medical Instrumentation*. Englewood Cliffs, NJ: Prentice Hall, 1981.
- 16 Woo E J, Hua P, Tompkins W J and Webster J G. 32-electrode electrical impedance tomograph-software design and static images. *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1989, pp. 455—56
- 17 Xue Q Z, Hu Y H and Tompkins W J. Neural-network-based adaptive matched filtering for QRS detection. *IEEE Trans. Biomed. Eng.*, 1992, BME-39(4): 317—29

- 18 Yorkey T, Webster J G and Tompkins W J. Comparing reconstruction algorithms for electrical impedance tomography. IEEE Trans. Biomed. Eng., 1987, BME-34(11):843—52

第二章

- 1 Tompkins W J and Webster J G. Design of Microcomputer-based Medical Instrumentation. Englewood Cliffs, NJ:Prentice-Hall. 1981.

第三章

- 1 Allen P E and Holberg D R. CMOS Analog Circuit Design. New York:Holt, Rinehart and Winston. 1987.
- 2 Cromwell L, Arditti M, Weibel F J, Pfeiffer E A, Steele B and Labok J. Medical Instrumentation for Health Care. Englewood Cliffs, NJ:Prentice Hall. 1976.
- 3 Oppenheim A V and Willsky A S. Signals and Systems. Englewood Cliffs, NJ:Prentice Hall. 1983.
- 4 Tompkins W J and Webster J G. Design of microcomputer-based medical instrumentation. Englewood Cliffs, NJ:Prentice Hall. 1981.
- 5 Tompkins W J and Webster J G. Interfacing sensors to the IBM PC. Englewood Cliffs, NJ:Prentice Hall. 1988.

第四章

- 1 Antoniou A. Digital Filters Analysis and Design, New York:McGraw-Hill. 1979.
- 2 Bogner R E and Constantinides A G. Introduction to Digital Filtering, New York:John Wiley and Sons. 1985.
- 3 Gold B and Rader C. Digital Processing of Signals, New York:Lincoln Laboratory Publications, McGraw-Hill. 1969.
- 4 Rabiner L R and Rader C M. Digital Signal Processing. New York:IEEE Press. 1972.
- 5 Stearns S D. Digital Signal Analysis. Rochelle Park, NJ:Hayden. 1975.

第五章

- 1 Antoniou A. Digital Filters; Analysis and Design. New York:McGraw-Hill. 1979.
- 2 Bogner R E and Constantinides A G. Introduction to Digital Filtering. New York:John Wiley and Sons. 1985.
- 3 Friesen G M et al. A comparison of the noise sensitivity of nine QRS detection algorithms. IEEE Trans. Biomed. Eng., 1990, BME-37(1):85—98
- 4 Hamming R W. Digital Filters. Englewood Cliffs NJ:Prentice Hall. 1977.
- 5 Rabiner L R and Rader C M. Digital Signal Processing. New York:IEEE Press. 1972.
- 6 Roberts R A and Mullis R T. Digital Signal Processing. Reading, MA:Addison-Wesley. 1987.

第六章

- 1 Soderstrand M A. On-line digital filtering using the PDP-8 or PDP-12. Computers in the Neurophysiology Laboratory, 1:31-49. Maynard, MA:Digital Equipment Corporation. 1972.
- 2 Stearns S D. Digital Signal Analysis. Rochelle Park. NJ:Hayden. 1975.
- 3 Tompkins W J and Webster J G. Design of Microcomputer-based Medical Instrumentation. Englewood Cliffs, NJ:Prentice Hall. 1981.

第七章

- 1 Ahlstrom M L and Tompkins W J. Digital filters for real-time ECG signal processing using microprocessors. *IEEE Trans. Biomed. Eng.*, 1985, BME-32(9):708—13
- 2 Kohn A F. Phase distortion in biological signal analysis caused by linear phase FIR filters. *Med. & Biol. Eng. & Comput.*, 1987, 25:231—38
- 3 Lynn P A. Online digital filters for biological signals; some fast designs for a small computer. *Med. & Biol. Eng. & Comput.*, 1977, 15:534—40
- 4 Lynn P A. Recursive digital filters for biological signals. *Med. & Biol. Eng. & Comput.*, 1971, 9:37—44
- 5 Lynn P A. Recursive digital filters with linear-phase characteristics. *Comput. J.*, 1972, 15:337—42
- 6 Lynn P A. Transversal resonator digital filters; fast and flexible online processors for biological signals. *Med. & Biol. Eng. & Comput.*, 1983, 21:718—30
- 7 Principe J C and Smith J R. Design and implementation of linear phase FIR filters for biological signal processing. *IEEE Trans. Biomed. Eng.*, 1986, BME-33(6):550—59
- 8 Thakor N V and Moreau D. Design and analysis of quantised coefficient digital filters; application to biomedical signal processing with microprocessors. *Med. & Biol. & Comput.*, 1987, 25:18—25

第八章

- 1 Ahlstrom M L and Tompkins W J. Digital filters for real-time ECG signal processing using microprocessors. *IEEE Trans. Biomed. Eng.*, 1985, BME-32(9):708—13
- 2 Hamilton P S and Tompkins W J. Detection of ventricular fibrillation and tachycardia by adaptive modeling. *Proc. Annu. Conf. Eng. Med. Bio. Soc.*, 1987, 1881—82
- 3 Haykin S. *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall. 1986.
- 4 Huhta J C. and Webster J G. 60-Hz interference in electrocardiography. *IEEE Trans. Biomed. Eng.*, 1973, BME-20(2):91—101
- 5 Mortara D W. Digital filters for ECG signals. *Computers in Cardiology*, 1977, 511—14
- 6 Sahakian A V and Kuo K H. Canceling the cardiogenic artifact in impedance pneumography. *Proc. Annu. Conf. Eng. Med. Bio. Soc.*, 1985, 855—59
- 7 Sheng Z and Thakor N V. P-wave detection by an adaptive QRS-T cancellation technique. *Computers in Cardiology*, 1987, 249—52
- 8 Widrow B, Glover J R, John M, Kaunitz J, Charles S J, Hearn R H, Zeidler J R, Dong E and Goodlin R C. Adaptive noise canceling: principles and applications. *Proc. IEEE*, 1975, 63(12):1692—1716
- 9 Zhou L, Wei D and Sun L. Fetal ECG processing by adaptive noise cancellation. *Proc. Annu. Conf. Eng. Med. Bio. Soc.*, 1985, 834—37

第九章

- 1 Tompkins W J and Webster J G. *Design of Microcomputer-based Medical Instrumentation*. Englewood Cliffs, NJ: Prentice Hall. 1981.

第十章

- 1 Abenstein J P and Tompkins W J. New data-reduction algorithm for real-time ECG analysis, *IEEE Trans. Biomed. Eng.*, 1982, BME-29:43—48
- 2 Bohs L N and Barr R C. Prototype for real-time adaptive sampling using the Fan algorithm, *Med. & Biol. Eng. & Comput.*, 1988, 26:574—83

- 3 Cox J R, Nolle F M, Fozzard H A and Oliver G C Jr. AZTEC:a preprocessing program for real-time ECG rhythm analysis. *IEEE Trans. Biomed. Eng.*, 1968, BME-15:128—29
- 4 Hamilton P S and Tompkins W J. Compression of the ambulatory ECG by average beat subtraction and residual differencing. *IEEE Trans. Biomed. Eng.*, 1991, BME-38(3):253—59
- 5 Hamilton P S and Tompkins W J. Theoretical and experimental rate distortion performance in compression of ambulatory ECGs. *IEEE Trans. Biomed. Eng.*, 1991, BME-38(3):260—66
- 6 Huffman D A. A method for construction of minimum-redundancy codes. *Proc. IRE*, 1952, 40:1098—1101
- 7 Ishijima M, Shin S B, Hostetter G H and Skalansky J. Scan-along polygonal approximation for data compression of electrocardiograms, *IEEE Trans. Biomed. Eng.*, 1983, BME-30:723—29
- 8 Jalaleddine S M S, Hutchens C G, Coberly W A and Strattan R D. Compression of Holter ECG data. *Biomedical Sciences Instrumentation*, 1988, 24:35—45
- 9 Jalaleddine S M S, Hutchens C G and Strattan R D. ECG data compression techniques—A unified approach. *IEEE Trans. Biomed. Eng.*, 1990, BME-37:329—43
- 10 Moody G B, Soroushian and Mark R G. ECG data compression for tapeless ambulatory monitors. *Computers in Cardiology*, 1988, 467—70
- 11 Mueller W C. Arrhythmia detection program for an ambulatory ECG monitor. *Biomed. Sci. Instrument.*, 1978, 14:81—85
- 12 Ruttmann U E and Pipberger H V. Compression of ECG by prediction or interpolation and entropy encoding. *IEEE Trans. Biomed. Eng.*, 1979, BME-26:613—23
- 13 Ruttmann U E, Benson A S and Pipberger H V. ECG data compression by linear prediction. *Proc. Comput. Cardiol.*, 1976, 313—15
- 14 Tai S C. SLOPE—a real-time ECG data compressor. *Med. & Biol. Eng. & Comput.*, 1991, 175—79
- 15 Tompkins W J and Webster J G. Design of Microcomputer-based Medical Instrumentation. Englewood Cliffs, NJ, Prentice Hall 1981.

第十一章

- 1 Alkin O. PC-DSP, Englewood Cliffs, NJ, Prentice Hall. 1991.
- 2 Oppenheim A V and Schafer R W. Digital Signal Processing. Englewood Cliffs, NJ, Prentice Hall. 1975.
- 3 Pfister C J, Hamilton J W, Bass P, Webster J G and Tompkins W J. Use of spectral analysis in detection of frequency differences in electrogastrograms of normal and diabetic subjects. *IEEE Trans. Biomed. Eng.* 1988, 935—41

第十二章

- 1 AHA ECG database. Available from Emergency Care Research Institute, 5200 Butler Pike, Plymouth Meeting, PA 1962.
- 2 Ahlstrom M L and Tompkins W J. Automated high-speed analysis of Holter tapes with microcomputers. *IEEE Trans. Biomed. Eng.*, 1983, BME-30:651—57
- 3 Ahlstrom M L and Tompkins W J. Digital filters for real-time ECG signal processing using microprocessors. *IEEE Trans. Biomed. Eng.*, 1985, BME-32:708—13
- 4 Balda R A, Diller G, Deardorff E, Doue J and Hsieh P. The HP ECG analysis program. *Trends In Computer-Processed Electrocardiograms*. vanBemmel JH and Willems JL. Amsterdam, The Netherlands: North Holland, 1977, 197—205
- 5 Dobbs S E, Schmitt N M, Ozemek H S. QRS detection by template matching using real-time correlation on a microcomputer. *Journal of Clinical Engineering*, 1984, 9:197—212

- 6 Friesen G M, Jannett T C, Jadallah M A, Yates S L, Quint S R, Nagle H T. A comparison of the noise sensitivity of nine QRS detection algorithms. *IEEE Trans. Biomed. Eng.*, 1990, BME-37, 85—97
- 7 Furno G S and Tompkins W J. QRS detection using automata theory in a battery-powered microprocessor system. *IEEE Frontiers of Engineering in Health Care*, 1982, 4, 155—58
- 8 Hamilton P S and Tompkins W J. Quantitative investigation of QRS detection rules using the MIT / BIH arrhythmia database. *IEEE Trans. Biomed. Eng.* 1986, BME-33, 1157—65
- 9 Pan J and Tompkins W J. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* 1985, BME-32, 230—36
- 10 Thakor N V, Webster J G and Tompkins W J. Optimal QRS detector. *Medical and Biological Engineering*, 1983, 343—50
- 11 Thakor N V, Webster J G and Tompkins W J. Estimation of QRS complex power spectra for design of a QRS filter. *IEEE Trans. Biomed. Eng.*, 1984, BME-31, 702—05

第十三章

- 1 Abenstein J P. Algorithms for real-time ambulatory ECG monitoring. *Biomed. Sci. Instrum.*, 1978, 14, 73—79
- 2 Abenstein J P and Thakor N V. A detailed design example—ambulatory ECG monitoring. In Tompkins, W J and Webster, J G. *Design of Microcomputer-Based Medical Instrumentation*. Englewood Cliffs, NJ: Prentice Hall. 1981.
- 3 Abenstein J P and Tompkins W J. A new data reduction algorithm for real-time ECG analysis. *IEEE Trans Biomed Eng.*, 1982, BME-29, 43—48
- 4 Ahlstrom M L and Tompkins W J. An inexpensive microprocessor system for high speed QRS width detection. *Proc. 1st Annual IEEE Compmed. Conf.*, 1981, 81—83
- 5 Ahlstrom M L and Tompkins W J. Automated high-speed analysis of Holter tapes with microcomputers. *IEEE Trans. Biomed. Eng.*, 1983, BME-30, 651—57
- 6 Bonner R E and Schwetman H D. Computer diagnosis of the electrocardiogram II. *Comp. Biomed. Res.*, 1968, 1, 366
- 7 Furno G S and Tompkins W J. QRS detection using automata theory in a battery-powered microprocessor system. *IEEE Frontiers of Eng. in Health Care*, 1982, 4, 155—58
- 8 Klingeman J and Pipberger H V. Computer classification of electrocardiograms. *Comp. Biomed. Res.*, 1967, 1, 1
- 9 Macfarlane P W, Lorimer A R and Lowrie T D V. 3 and 12 lead electrocardiogram interpretation by computer. A comparison in 1093 patients. *Br. Heart J.*, 1971, 33, 226
- 10 Mueller W C. Arrhythmia detection program for an ambulatory ECG monitor. *Biomed. Sci. Instrum.*, 1978, 14, 81—85
- 11 Pan J and Tompkins W J. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* 1985, BME-32 (3), 230—36
- 12 Fordy L, Jaffe H, Chesky K, et al. Computer diagnosis of electrocardiograms, IV, a computer program for contour analysis with clinical results of rhythm and contour interpretation. *Comp. Biomed. Res.*, 1968, 1, 408—33
- 13 Thakor N V. Reliable R-wave detection from ambulatory subjects. *Biomed. Sci. Instrum.*, 1978, 14, 67—72
- 14 Thakor N V, Webster J G and Tompkins W J. Optimal QRS filter. *Proc. IEEE Conf. on Frontiers of Eng. in Health Care*, 1980, 2, 190—95
- 15 Thakor N V, Webster J G and Tompkins W J. A battery-powered digital modem for telephone

- transmission of ECG data. *IEEE Trans. Biomed. Eng.*, 1982, BME-29, 355—59
- 16 Thakor N V, Webster J G and Tompkins W J. Optimal QRS detector. *Med. & Biol. Eng. & Comput.*, 1983, 21: 343—50
 - 17 Thakor N V, Webster J G and Tompkins W J. Design, implementation, and evaluation of a microcomputer-based portable arrhythmia monitor. *Med. & Biol. Eng. & Comput.*, 1984, 22: 151—59
 - 18 Thakor N V, Webster J G and Tompkins W J. Estimation of QRS complex power spectra for design of a QRS filter. *IEEE Trans. Biomed. Eng.* 1984, BME-31: 702—706
 - 19 Tompkins W J. A portable microcomputer-based system for biomedical applications. *Biomed. Sci. Instrum.*, 1978, 14: 61—66
 - 20 Tompkins W J. Modular design of microcomputer-based medical instruments. *Med. Instrument*, 1980, 14: 315—18
 - 21 Tompkins W J. Portable microcomputer-based instrumentation. In Eden, M. S. and Eden, M. (eds) *Microcomputers in Patient Care*, Park Ridge, NJ, Noyes Medical Publications, 1981, 174—81
 - 22 Tompkins W J. Role of microprocessors in ambulatory monitoring. *Proc. AAMI*, 99. 1981.
 - 23 Tompkins W J. Trends in ambulatory electrocardiography. *IEEE Frontiers of Eng. in Health Care*, 1982, 4: 201—04
 - 24 Tompkins W J. Arrhythmia detection and capture from ambulatory outpatients using microprocessors. *Proc. AAMI*, 122. 1983.
 - 25 Tompkins W J and Abenstein J P. CORTES—A data reduction algorithm for electrocardiography. *Proc. AAMI*, 277. 1979.
 - 26 Tompkins W J, Tompkins B M and Weisner S J. Microprocessor-based device for real-time ECG processing in the operating room. *Proc. AAMI*, 122. 1983.
 - 27 Tompkins W J, Webster J G, Sahakian A V, Thakor N V and Mueller W C. Long-term, portable ECG arrhythmia monitoring. *Proc. AAMI*, 278. 1979.
 - 28 Webster J G. An intelligent monitor for ambulatory ECGs. *Biomed. Sci. Instrum.*, 1978, 14: 55—60
 - 29 Webster J G, Tompkins W J, Thakor N V, Abenstein J P and Mueller W C. A portable, microcomputer-based ECG arrhythmia monitor. *Proc. 31st ACEMB*, 60. 1978.
 - 30 Weisner S J, Tompkins W J, and Tompkins B M A compact, microprocessor-based ECG ST-segment monitor for the operating room. *IEEE Trans. Biomed. Eng.*, 1982, BME-29: 642—49
 - 31 Weisner S J, Tompkins W J and Tompkins B M. Microprocessor-based, portable anesthesiology ST-segment analyzer. *Proc. Northeast Bioeng. Conf.*, 1982, 222—226

第十四章

- 1 Denyer P and Renshaw D. *VLSI Signal Processing: A Bit-serial Approach*. Reading, MA: Addison-Wesley. 1985.
- 2 Duncan R. A survey of parallel computer architectures. *Computer*, 1990, 23(2): 5—16
- 3 Einspruch N G and Gold R D. *VLSI in Medicine*. San Diego: Academic Press. 1989.
- 4 Flynn M J. Very high speed computing systems. *Proc. IEEE*, 1966, 54: 1901—1909
- 5 Fortes J A B and Wah B W. Systolic arrays—from concept to implementation. *Computer*, 1987, 20(7): 12—17
- 6 Kung S Y. *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice Hall. 1988.
- 7 Kung S Y, Whitehouse H J and Kailath T. *VLSI and Modern Signal Processing*. Englewood Cliffs, NJ: Prentice Hall. 1985.
- 8 Lee E A. Programmable DSP architectures: Part 1. *IEEE ASSP Magazine*. 1988, 4—19

- 9 Mo F. Basic digital signal processing concepts for medical applications. *Biomedical Science and Technology*. 1991, 1(1):12—18
- 10 Motorola. Technical Data # DSP56001/D, Motorola Literature Distribution, P. O. Box 20912, Phoenix, Arizona, 85036. 1988.
- 11 Quarmby D. (ed.) *Signal Processor Chips*. Englewood Cliffs, NJ: Prentice Hall. 1985.
- 12 Szabo N S and Tanaka R I. *Residue Arithmetic and Its Applications to Computer Technology*. New York, McGraw-Hill. 1967.

附录 A

- 1 M68HC11EVB Universal Evaluation Board User's Manual. Motorola Literature Distribution, P. O. Box 20912, Phoenix, AZ 85036. 1986.
- 2 ADA2100 User's Manual. Real Time Devices, Inc., 820 North University Dr., P. O. Box 906, State College, PA 16804, 1990.
- 3 M68HC11 User's Manual. Motorola Literature Distribution, P. O. Box 20912, Phoenix, AZ 85036. 1990.

附录 B

- 1 ADA2100 User's Manual. Real Time Devices, Inc., 820 North University Dr., P. O. Box 906, State College, PA 16804. Ph: (814)234-8087; 1990, Fax: (814)234—5218
- 2 Eggbrecht, L. C. *Interfacing to the IBM Personal Computer*. Indianapolis, IN: Howard W. Sams. 1983.
- 3 M68HC11 User's Manual. Motorola Literature Distribution, P. O. Box 20912, Phoenix, Arizona 85036. 1990.
- 4 M68HC11EVB Evaluation Board User's Manual. Motorola Literature Distribution, P. O. Box 20912, Phoenix, AZ 85036. 1986.
- 5 Turbo C Reference Guide. Borland International, Inc., 1800 Green Hills Road, P. O. Box 660001, Scotts Valley, CA 95055-0001. 1988.
- 6 Turbo C User's Guide. Borland International, Inc., 1800 Green Hills Road, P. O. Box 660001, Scotts Valley, CA 95055-0001. 1988.

附录 C

- 1 ADA2100 User's Manual. Real Time Devices, Inc., 820 North University Dr., P. O. Box 906, State College, PA 16804. 1990.
- 2 Bovens N and Brysbaert, M. IBM PC / XT / AT and PS / 2 Turbo Pascal timing with extended resolution. *Behavior Research Methods, Instruments, & Computers*, 1990, 22(3):332—34
- 3 Eggbrecht L C. *Interfacing to the IBM Personal Computer*. Indianapolis, IN: Howard W. Sams. 1983.
- 4 M68HC11 User's Manual. Motorola Literature Distribution, P. O. Box 20912, Phoenix, Arizona 85036. 1990.
- 5 M68HC11EVB Evaluation Board User's Manual. Motorola Literature Distribution, P. O. Box 20912, Phoenix, AZ 85036. 1986.
- 6 Microprocessor and Peripheral Handbook, Volumes I and II. Intel Literature Sales, P. O. Box 8130, Santa Clara, CA 95052-8130. 1988.
- 7 Turbo C Reference Guide. Borland International, Inc., 1800 Green Hills Road, P. O. Box 660001, Scotts Valley, CA 95055-0001. 1988.
- 8 Turbo C User's Guide. Borland International, Inc., 1800 Green Hills Road, P. O. Box 660001, Scotts Valley, CA 95055-0001. 1988.

附录 F

- 1 Oppenheim A V and Schafer R W. Digital Signal Processing. Englewood Cliffs, NJ: Prentice Hall. 1975.
- 2 Roberts R A and Mullis C T. Digital Signal Processing, Reading, MA: Addison-Wesley. 1987.

附录 G

- 1 Baran N. Data Acquisition: PCs on the bench. Byte, 145—49, (May). 1991.
- 2 Coffee P C. Tools provide complex numerical analysis. PC Week, (Oct. 8). 1990.
- 3 National Instruments. IEEE-488 and VXI bus control. Data Acquisition and Analysis. 1991.
- 4 Omega Technologies. The Data Acquisition Systems Handbook. 1991.
- 5 The Math Works, Inc. Signal Processing Toolbox User's Guide. 1988.

[G e n e r a l I n f o r m a t i o n]

书名 = 生物医学数字信号处理

作者 =

页数 = 2 4 6

S S 号 = 0

出版日期 =

V s s 号 = 9 3 1 9 8 3 8 5

[封面](#)
[书名](#)
[版权](#)
[前言](#)
[目录](#)
[正文](#)