目　录(Contents)

第一部分　基础概念
(Fundamental Concepts)

PDG

第一部分概述(Overview to Part Ⅰ)

从前——也并非很久以前——数字音频的录音、合成、处理和重放还是实验室专家的特权,可今天,它们几乎像电视机一样普通,基本上每台计算机都有数字音频装置。

第1章的主题"数字音频"是计算机音乐的核心。采样(sample)——不过是个数字——则是声音的原子。从理论上说,借助一系列跟踪记录时间轴声音波形的标本,并通过扬声器播放,我们就可以构造出任何声音。然而,只是在采样率和采样宽度(sample width)满足严格的相关技术条件时,理论才成为现实。如果采样率过低,结果会因失真而造成消音或脏音。采样宽度则是指用于表示一个样本的数码字的长度,如果它过小,那么声音简直会被噪音劈碎。

第2章介绍程序设计的艺术。要想在计算机音乐领域真正有所出新,关键要知道如何编写程序。所以,通晓程序设计的各种概念对于学生是不可或缺的课题。

第一部分的构成(Organization of Part Ⅰ)

第一部分介绍了贯穿全书的数字音频与程序设计的基本概念。第1章与第2章以概要方式涵盖了大量材料,目的在于向读者传达对这部分所涉及的领域的一种基本认识,从而为此后的章节做好准备。

第1章由约翰·斯特朗(John Strawn)与柯蒂斯·罗兹(Curtis Roads)共同撰写,涵盖的基本内容包括:数字录音的历史、采样定理、混叠、相位校正、量化、抖动、音频转换器、过采样以及数码音频格式。本章部分内容最初发表于《键盘》(*Keyboard*)杂志,但在收入本教程前又做了大量修订。

第2章《音乐系统编程》出自柯蒂斯·阿博特(Curtis Abbott)这样的大师

之手，介绍了程序设计的艺术。作者梳理了各种程序语言以及程序风格要素的演变。他总结了程序设计语言的基本概念、控制与数据结构，讲述了面向对象的程序设计基础。

第 1 章　数字音频概念
(Digital Audio Concepts)

柯蒂斯·罗兹(Curtis Roads)、约翰·斯特朗(John Strawn)

数字录音与计算机音乐技术的融合创造了一种灵活而强大的艺术形式。本章将介绍数字录音与重放的历史与技术。通过这部分的介绍，读者可以熟悉数字音频的基本术语与概念。简要起见，我们对那些其本身已经是大型专业的主题加以浓缩；更多的文献资源可参见 D. 戴维斯(Davis1988，1992)。

背景：数字录音史
(Background: History of Digital Audio Recording)

声音的录制有其丰富的历史，它始于托马斯·爱迪生(Thomas Edison)和埃米尔·贝利纳(Emile Berliner)在 19 世纪 70 年代的实验，并以 1898 年普尔森(V. Poulsen)研制的留声电话机磁线(译注：录音钢丝)记录器为标志(Read and Welch 1976)。早期的录音是一个机械过程(图 1.1)。

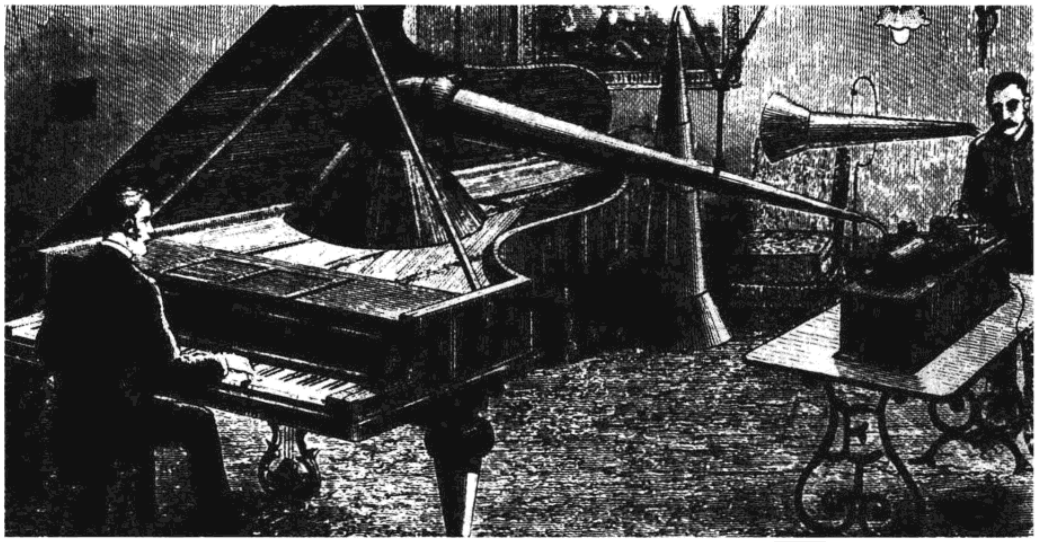


图 1.1　1900 年之前的机械录音。声音的振动由伸向钢琴正上方的大锥斗拾取，转换为一枚刻纹针的振动，使后者在一个旋转的蜡圆筒上刺刻。

虽然 1906 年三极真空管的发明开启了电子新时代，然而电子录音直到 1924 年才真正可行(Keller 1981)。图 1.2 所示为 20 世纪 20 年代一种典型的号筒加载式扬声器。

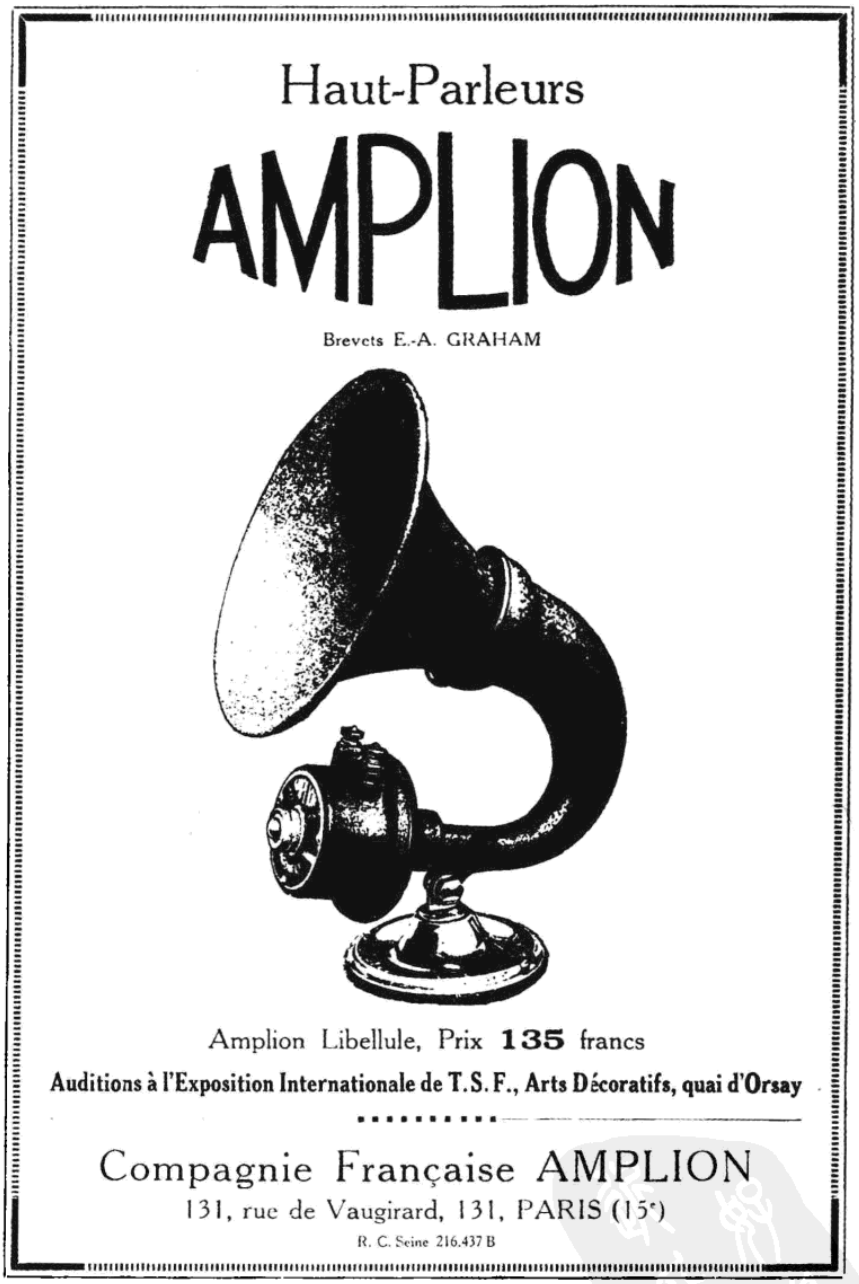


图 1.2 Amplion 艾普力昂扬声器 1925 年广告。

在胶片上的光学录音于 1922 年首次试验成功(Ristow 1993)。20 世纪 30 年代,德国发明了在附有磁性材料粉末涂敷层的条带上录音的技术(图 1.3),但这一技术直到第二次世界大战之后才在世界范围内推广。德国的磁带录音机相比此前的钢丝录音机与钢带录音机有巨大的进步,后者每次接合都需要焊

接完成。磁带录音机及其后继者都是模拟式(analog)录音设备。所谓“模拟”指的是这样一个事实:在磁带上以编码记录的波形是对由麦克风拾取的原声波形最接近的相似物。模拟录音虽然得到不断改进,却存在着根本的物理局限。当从一个模拟媒介复制到另一个的时候,这种局限就极为凸显,无可避免地出现附加噪音。

有关模拟录音的历史,尤其是针对多轨设备的,参见第 9 章。

图 1.3　1935 年便携式 Magnetophon 磁带录音机原型,AEG 制造。(图片由 BASF Aktiengesellschaft 巴斯夫公司提供。)

实验性数字录音(Experimental Digital Recording)

采样(Sampling)是数字录音的核心概念,即将连续的模拟信号(例如来自麦克风的信号)转化为非连续时间取样(time-sampled)信号。采样的理论基础是采样定理(sampling theorem),它特别规范了采样率与音频带宽之间的关系(参见本章稍后关于采样定理的专门部分)。虽然这一定律在贝尔电话实验室的 H. 奈奎斯特(Harold Nyquist)的工作之后也被称为奈奎斯特定律(Nyquist 1928),不过,该定律的另一种形式则早在 1841 年即由法国数学家 A. 柯希

(Augustin Louis Cauchy,1789—1857)率先提出。英国研究者 A. 里维斯(A. Reeves)开发出并注册了专利的第一个脉冲编码调制系统(pulse-code-modulation, PCM)以"振幅对分,时间量化"的(数字)形式传递信息(Reeves 1938, Licklider 1950,Black 1953)。直到今天,数字录音有时还被称为"PCM 录音"。信息论(information theory)的发展对理解数字音频传输很有帮助(Shannon1948)。为解决在模拟信号与数字信号之间进行转换的难题人们花费了几十年的工夫,至今仍在不断改进之中。(稍后我们将进一步详述转化过程。)

20 世纪 50 年代后期,贝尔电话实验室的马克斯·马修斯(Max Mathews)和他的工作小组从一台数字计算机中生成了第一批人工合成声音。这批样本继而通过数字计算机写到昂贵而笨重的卷对卷(reel-to-reel)计算机磁带存储驱动器上。由数字生成声音的这一过程与通过由 Epsco 公司为客户定制的 12 比特(bit)真空管"数字—声音转换器"重放磁带的过程完全分离开来(Roads 1980;见第 3 章)。

20 世纪 50—60 年代,哈明(Hamming)、胡夫曼(Huffman)与吉尔伯特(Gilbert)发明了数字纠错理论(digital error correction)。稍后,萨托(Sato)、布莱瑟(Blesser)、斯托克海姆(Stockham)与多伊梅德(Doimade)对于数字纠错的贡献则使第一个实用的数字录音系统得以问世。第一个专用单声道(基于录像带机制的)数字录音机首先由日本广播公司 NHK 发布(Nakajima et al. 1983)。很快,Denon 公司开发出升级版(图 1.4),数字录音机推向市场的竞赛由此开始(Iwamura et al. 1973)。

时至 1977 年,第一个商用录音系统 Sony PCM-1 处理器推向市场,用以将 13 比特数字音频信号的编码记录在 Sony Beta 制式的盒带录像机上。不到一年即被诸如 Sony PCM-1600 这样的 16 比特 PCM 器所取代(Nakajima et al. 1978)。产品由此开始分作两条线索发展:专业型与"消费者"型,尽管这种数字录音从未发展出真正意义上的大众市场。专业型 Sony PCM-1610 与 1630 成为压缩光盘(CD)原版片制作的标准,而 Sony PCM-F1—compatible 兼容系统(亦称为 EIAJ 系统,EIAJ,日本电子工业协会 Electronics Industry Association of Japan 的缩略)则成为在录像带上进行低成本数字录音的实际标准。这些标准在整个 20 世纪 80 年代一直通用。

音频工程协会(Audio Engineering Society)于 1985 年制定了两个标准采样频率:44.1kHz 与 48kHz。1992 年加以更新(Audio Engineering Society 1992a, 1992b)。(用于广播的 32kHz 采样频率也同时存在。)其间,一些公司还发展了高分辨率数字录音机,能以高于 16 比特的采样率编码。例如,Mitsubishi 的一款 X-86 卷到卷数字带录音机可以 96kHz 的采样频率进行 20 比特编

码(Mitsubishi 1986)。各种各样的高分辨率录音机至今仍可使用。

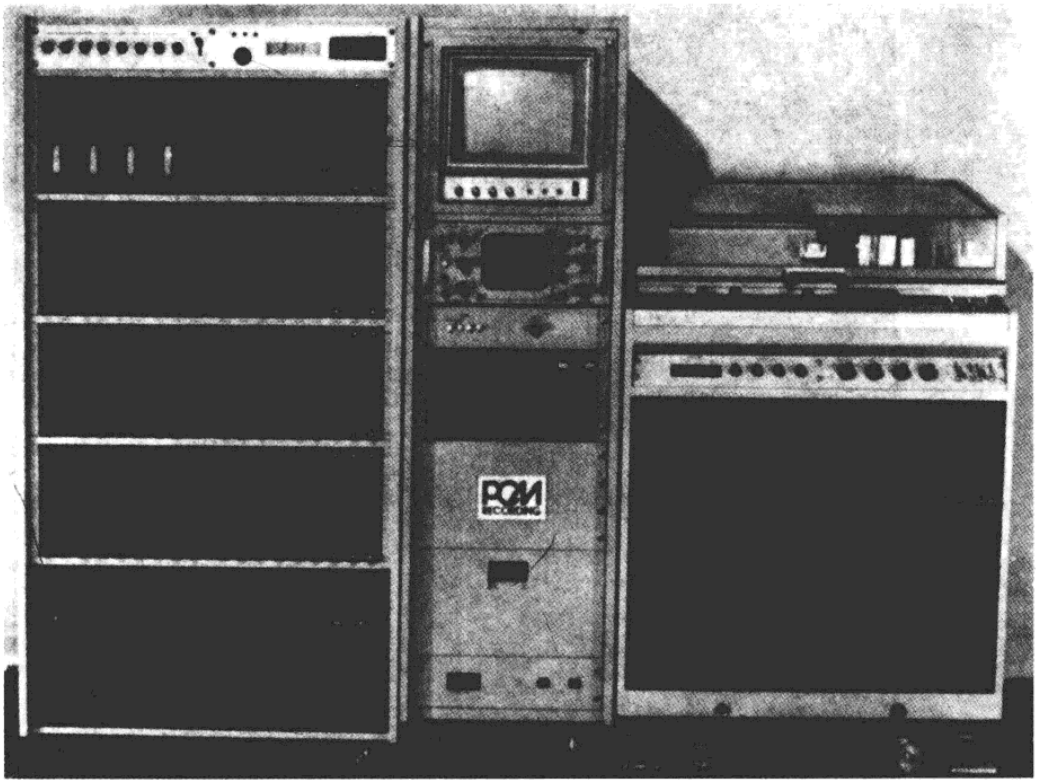


图 1.4　1973 年基于 1 英寸录像机(右侧)制成的 Nippon Columbia 日本哥伦比亚(Denon 德农)数字录音机。

面向公众的数字声(Digital Sound for the Public)

数字化声音在 1982 年通过压缩光盘(CD)的形式——一个由激光束读取的 12 厘米光学碟片——最初抵达普通大众。CD 格式由 Philips 和 Sony 公司联合研制多年而成。它在两年中销售 135 万台播放机和数千万张光碟(Pohlman 1989),在商业上获得了巨大的成功。自此,一系列由 CD 技术延伸的产品应运而生,包括 CD-ROM(Read Only Memory 只读存储器)、CD-I(互动),以及混合了音频数据、文字和图像的其他格式。

直至 20 世纪 90 年代早期,制造商主要针对的是可录数字媒介方面的需求。不同类型的立体声媒介相继出现,包括数字录音带(Digital Audio Tape, DAT)、小型数字盒带(Digital Compact Cassettes, DCC)、小型光盘(Mini-Disc, MD)以及可写光盘(recordable CDs, CD-R)。(参见下文数字音频媒体部分。)

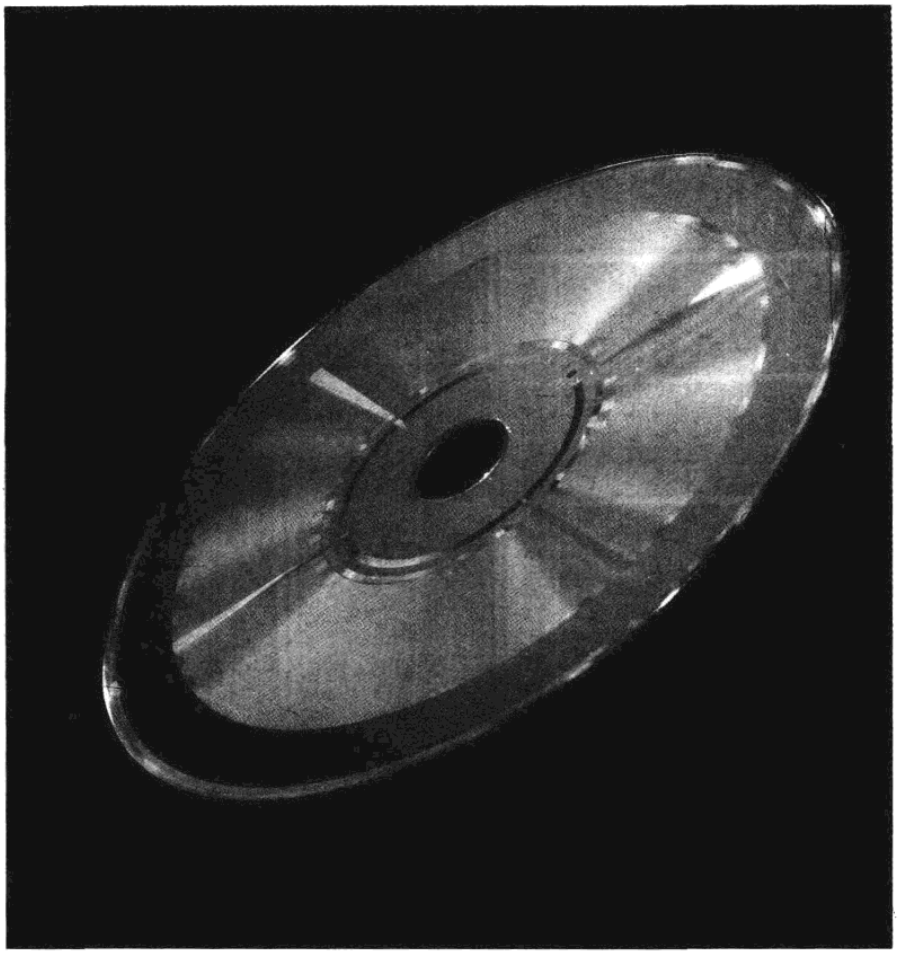


图 1.5 Sony-Philips 索尼—飞利浦压缩盘片。

适合音乐家的数字声(Digital Sound for Musicians)

虽然CD机本身拥有价钱不高的16比特数模转换器,但1988年前,配置有高品质转换器的计算机却并不普遍。在此之前,尽管少数计算机音乐中心等机构也特制了模数转换器和数模转换器,个人电脑系统的用户却还得等待些时日。他们可以购买数字合成器,并用MIDI协议(见第21章)通过计算机控制合成器,但却无法直接用计算机合成或录制声音。

直到20世纪80年代后期,品质高、价格低的转换器开始进入个人计算机。这一进步宣告了计算机音乐的新时代。不久,通过计算机进行声音合成、录制并处理开始广泛流行。许多种不同的音频工作站(audio work-stations)进入音乐市场。这些系统允许音乐人将音乐直接录制在与个人计算机连接的硬盘上。

这些音乐可以从硬盘读取播放，在计算机屏幕上被精确地编辑。

数字多轨录音(Digital Multitrack Recording)

与立体声录音这种左右声道同时录制的方式所不同，多轨录音机(multitrack recorder)拥有分立的各个声道(channel)或音轨(track)，可以在不同的时间分别录制。例如，每个音轨可以录制一个单独的乐器，这样就使稍后各轨的混音留有余地。多轨录音设备的另外一个优点在于，它可使音乐人的录音呈现得很有层次；每一层新的声音都是对之前已录声音层的补充。

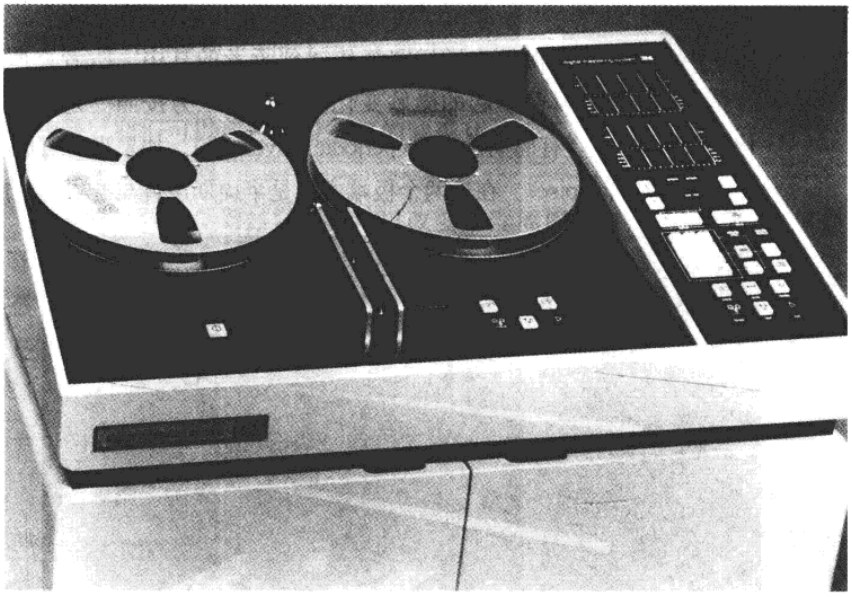


图 1.6　1978 年 3M 公司研制的 32 轨数字磁带机。

1976 年由英国广播公司(BBC)开发研制了一种试验性的 10 声道数字磁带机。两年后，3M 公司与 BBC 合作，推出了第一台 32 轨商用数字录音机(图 1.6)，以及初具雏形的数字磁带编辑机(Duffy 1982)。第一台基于计算机光盘随机读取的声音编辑机与混频器则是由美国犹他州盐湖城的 Soundstream(声流)公司研发(见图 16.38)。该系统允许在计算机上同时对多达 8 个音轨或计算机磁盘的 8 个声音文件(sound files)做混频(Ingebretsen and Stockham 1984)。

至 20 世纪 80 年代中叶，3M 与 Soundstream 公司相继撤出数字多轨磁带式记录机市场，当时这一市场已经完全是 Sony 和 Mitsubishi 集团的天下，稍后 Studer 公司又分占一席。很多年间，数字多轨录音是一种极为昂贵的事业(图 1.7)。直至 20 世纪 90 年代初，随着 Alesis 与 Tascam 推出了低价的多轨带式录音机，以及很多其他企业研制的多轨圆盘录音机，这种状况才得以改观，开始

进入一个新阶段。(第 9 章将详述模拟多轨录音的历史。)

声音信号基础 (Basics of Sound Signals)

这一节将介绍如何描述声音信号的基本概念和术语,包括频率、振幅与相位。

频率与振幅(Frequency and Amplitude)

声音由一个音源发出通过空气传递到听者的耳中。听者之所以能听到声音是由于气压在耳朵里起的微妙变化。如果这压力按照一定的重复模式在变化,我们说这声音有周期性波形(periodic waveform)。如果没有可以辨识的模式,那么就称噪音(noise)。在这两个极端之间是半周期声音与准噪音的广大区间。

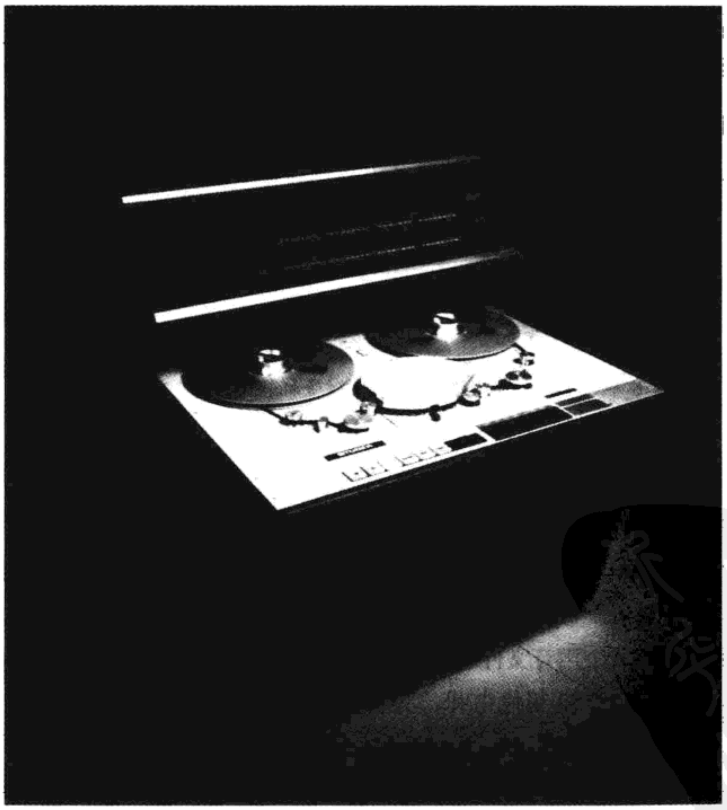


图 1.7 1991 年 Studer 公司推出的 D820-48 DASH 型数字式多轨录音机,零售价 270 000 美元。

周期性波形的一个反复称一个周波(cycle)；波形的基频(fundamental frequency)指每秒钟发生的周波的数量。而当周波的长度——被称为波长(wavelength)或周期(period)——上升时，每秒周波的频率就下降，反之同理。在本书中，为了与标准的声学术语规范相配合，我们用 Hz 指代“每秒周波”(“cycles per second”)。(Hz 是 Hertz 的缩略，以德国声学家 Heinrich Hertz 的名字命名。)

时域表示(Time-domain Representation)

一种描述声音波形的简单方法是绘制一张以空气压力对应时间的坐标图(图 1.8)，称为时域(time-domain)表示。当曲线接近坐标图底部时，表示气压下降；当曲线接近坐标图上端时，气压则上升。波形的振幅(amplitude)指气压变化量。我们可以通过一段给定的波形从气压零点到最高点(或最低点)的垂直距离来测量振幅。

一件声学乐器通过发出振动以改变乐器周围空气压力从而产生声音。扩音器根据电子信号的电压变化而经由前后运动发出声音。当扩音器从初始状态“向内”移动时，气压下降；当扬声器“向外”移动时，扬声器附近的气压上升。要使这通过向内或向外的振动生成可以被听见的声音，振动的频率就必须在 20—20 000Hz 之间。

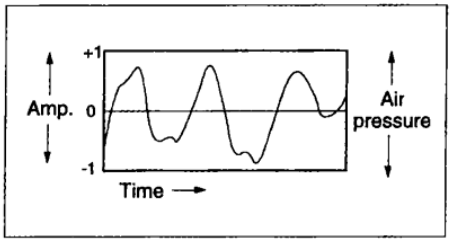


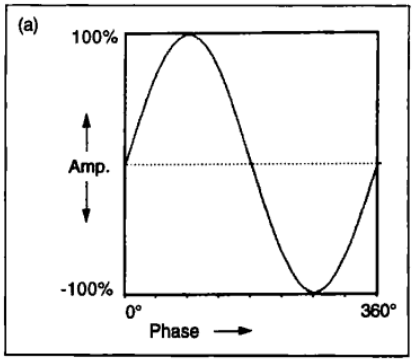
图 1.8　信号的时域表示。垂直方向上表示气压。当曲线接近坐标顶部时，气压加大；低于水平线的话，气压下降。以声音听到的大气压力变化可以发生得很快。对乐音而言，这整个图表不过千分之一秒(1 毫秒)。
Amp.=振幅　Time=时间　Air pressure=气压

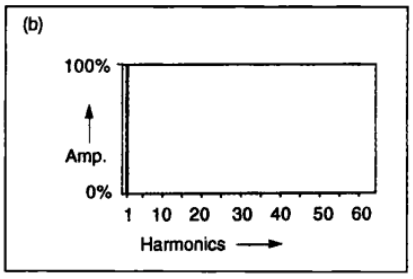
频域表示(Frequency-domain Representation)

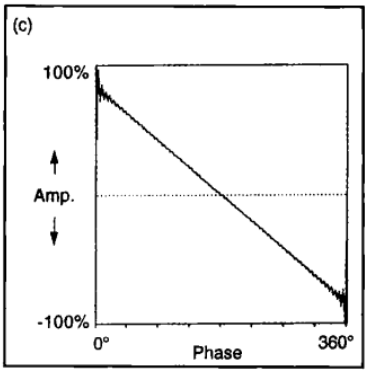
除了基频，在一个波形中还可以呈现很多频率。一个频域(frequency-domain)或频谱(spectrum)表示可以显出声音的频率内容。频谱的单一频率分量可称为谐波(harmonics)或分音(partials)。谐波频率是基频的简单整数倍。假设一个基频或第一谐波为 440Hz，那么，第二谐波就是 880Hz，第三谐波即为

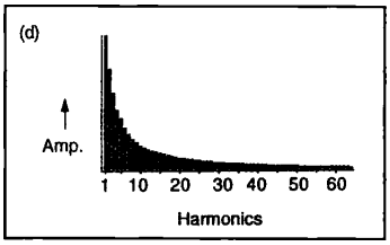
1 320Hz，以此类推。更常规地说，任何一个频率分量都可称为一个分音，无论它是否是基频的整数倍。事实上，很多声音并没有独特的基频。

波形的频率内容可以多种方式表示。标准方法是将每个分音沿 x 轴线性排开。每条线的高度则表示每个频率分量的强度（或振幅）。最纯粹的信号呈正弦（sine）波形。用正弦命名是因为它可以运用一个角度的正弦的三角公式来计算。（参见附录解释算法。）一个纯粹正弦波表示一个频率分量，或在频谱中表示为一条线。图 1.9 就描述了几个波形在时域与频域内的表示。注意频谱图中的水平轴被标为“谐波”，因为分析算法设定输入部分是一个周期性波形的完整的基本周期。而在图 1.9g 中的噪音信号情况下，这个假设就不适用，所以，我们把分音重新标示为“频率分量”（“frequency components”）。









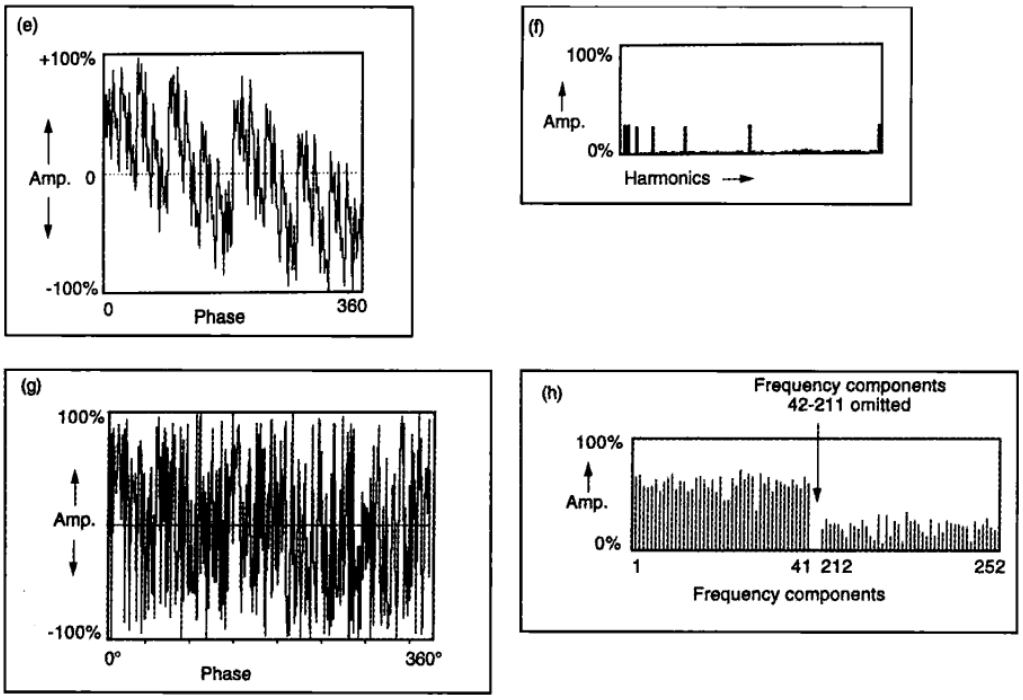


图 1.9　四个信号的时域与频率表示。(a)时间域的一个正弦波周期视图;(b)一个正弦波频率分量的频谱;(c)一个周期的锯齿状波形在时域的视图;(d)一个锯齿波形频率分量呈指数下降的频谱;(e)一个复杂波形周期在时域的视图。虽然波形看来复杂,但当它不断重复时,声音实际上很简单,类似簧片风琴的声音;(f)波形(e)频谱,表明其由几个频率分量主导;(g)一个随机噪音的波形;(h)如果一个波形不停改变(每一个周期都与此前的不同),我们听到的就是噪音。噪音的频率非常复杂。这里就分析出 252 种频率。这个快照并不能揭示它们的波幅是如何随时间而不停地改变的。

Amp.=振幅　Phase=相位　Harmonics=谐波　Frequency components=频率分量　omitted=缺省

相位(Phase)

在 y 轴或振幅轴上的周期性波形的起点就是它的初始相位(initial phase)。例如,一个典型的正弦波始于 0 振幅点,一个循环后止于 0。如果我们将水平轴上的起始点置换为 $\pi/2$(或 90 度),那么,正弦曲线波就将在振幅轴上起始于 1 并止于 1。按常规,这被称作余弦波。实际上,一个余弦相当于一个 90 度相位移(phase shifted)的正弦波(图 1.10)。

当两个信号始于同一点时,就称为同相(in phase)或对准相位(phase aligned)。与此形成对照,那些相对于另一信号略有延时的信号,我们称这两个信号为异相(out of phase)。当信号 A 和另一信号 B 的相位正好相反(错位 180 度,故信号 A 的每一个正值都对应信号 B 的一个负值),那么我们说,B 相对 A 是反极性(reversed polarity)。也可以说,B 是 A 的反转相位(phase-inverted)

副本。图 1.11 即演示了反转相位关系的两个信号相加的效果。

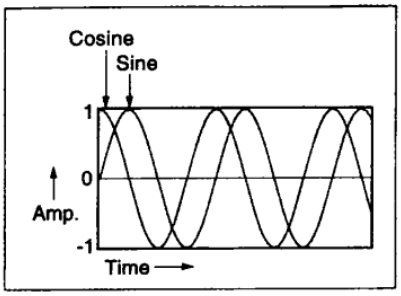


图 1.10 一个正弦波相当于一个被略微延时或移相的余弦波。
Amp.=振幅 Cosine=余弦 Sine=正弦 Time=时间

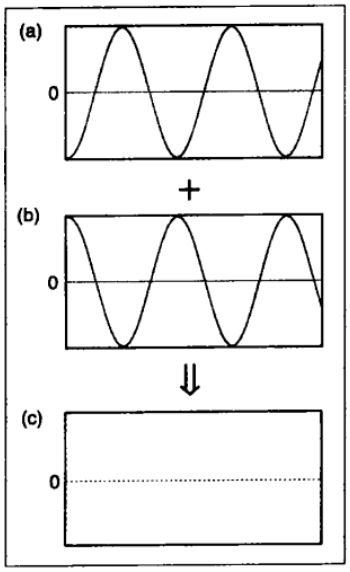


图 1.11 反转相位的结果。(b)是(a)的反转相位副本。如果两个波形叠加在一起,其和为零(c)。

相位的重要性(Importance of Phase)

时常有人说,相位对于人的耳朵并不重要,因为两个初始相位不同而信号完全相同的情况是极难分辨的。事实上,研究表明绝对相位或绝对极性(polarity)上的 180 度差异对于一些在实验室环境下的人而言是可以分辨的(Greiner 与 Melton 1991)。然而,即便除了这一特殊情况外,也还有很多理由表明相位的重要性。每一个滤波器运用移相来改变信号。一个滤波器相位(通过短时间延迟输入)将一个信号移位,然后将移位后的版本与原信号合在一起,产生"频率依赖式相位抵消"效应(frequency-dependent phase cancellation effects)从而

改变原初的频谱。用频率依赖(frequency-dependent)的概念,我们试图说明,并不是所有的频率分量都受到同等的影响。当相位移是时变的,受影响的频带也就会产生变化,产生一种我们称为整相(phasing)或镶边(flanging)的拂音效果(见第 10 章)。

在以分析一个已存在的声音为基础来重新合成声音的系统中,相位同样重要。尤其是,这些系统需要知道每一个频率分量的初始相位,从而将不同的分量按照正确的顺序排列起来(参见第 13 章以及附录)。相位数据在再生短促而迅速变化的经过音(transient sounds)时尤为关键,比如突然插入的一个乐器声。

最后,人们近年来更多关注那些能够对输入信号尽可能小地进行移相的音频组件,因为频率依赖性移相会让音乐信号在听觉上失真,干扰扩音器的成像(imaging)。〔成像是指由一组扩音器生成稳定的声象(audio picture)的能力,其中每一音响源在声象中被定位到明确的位置。〕多余的移相被称为相位失真(phase distortion)。打一个视觉比喻,相位失真的信号就像"焦距没对准"。

如上,我们已经介绍了音频信号的基本属性,现在,让我们就其两种表示法做一比较:模拟与数字。

声音的模拟表示法(Analog Representations of Sound)

正如空气压力可因声波发生变化,同样,一条将放大器与扩音器相连的电线中被称为电压(voltage)的电量也可以随声波的变化而变化。在此,我们无需对电压加以定义。考虑到本章主旨,我们只需简单地假设:改变一个与电线相关联的电特性是可能的,它多少与气压的变化吻合。

我们已经介绍的时变量(气压和电压)的一个重要特征是任何一方对于另一方或多或少恰好相似。一个由麦克风拾取的气压变化的图形看上去与这个声音重放时扬声器部位的(电压)变化图形极为相似。"模拟"一词提示我们,这些变量是如何密切关联的。

图 1.12 显示一个模拟音频链。一个音频信号的曲线可以沿传统留声机唱片的凹槽写入,正如图 1.12 所示。留声机唱片的凹槽的两壁上包含着存储在唱片中声音的连续时间表示(continuous-time representation)。当唱针在凹槽中滑行时,唱针横向前后运动,进而转化为电压,然后被放大,最终抵达扬声器。

虽然近年来模拟声音的复制进入了一个很高的水平,但模拟录音有其本质性局限。当你把一个模拟录音复制到另一个模拟录音时,拷贝永远不会和原始

录音同样好。这是由于模拟录音的过程总在增加噪音。对于第一代(first generation)或原始录音而言,噪音可能还不那么令人反感,但当我们进而录到第三代或第四代,一而再、再而三地复制,原始录音就越来越多地失之于噪音。正如我们稍后会进一步介绍的那样,相比之下,数字技术则允许生成无数代却依旧完美的(无噪音)原始录音的克隆。

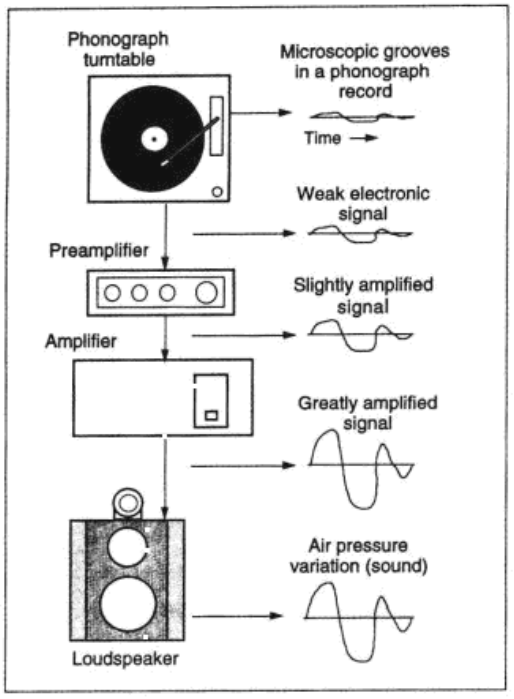


图 1.12 模拟音频链,从唱片凹槽传感为模拟波形开始,转为电压发送到前放大器、放大器、扬声器并被发射到空气中。
Phonograph turntable=留声机转盘 Microscopic grooves in a phonograph record=唱片凹槽放大图
Time=时间 Weak electronic signal=弱电信号 Preamplifier=前置放大器
Slightly amplified signal=略加放大的信号 Amplifier=放大器
Greatly amplified signal=充分放大的信号 Loudspeaker=扬声器
Air pressure variation(sound)=气压变化(声音)

从本质上说,生成或复制数字声涉及将一连串数字转换为我们刚刚讨论过的某种时变性的变化。如果这些数字能够转为电压,那么这些电压就能被放大并馈送到扬声器中而产生声音。

声音的数字表示法(Digital Representations of Sound)

这个部分将介绍最为基础的数字信号概念,包括信号的二进制数转化,音

频数据与 MIDI 数据的比较、采样、混叠、量化和抖动。

模拟—数字转换(Analog-to-digital Conversion)

我们先看一下从数字录音到重放的过程。与模拟环境中的连续时间信号不同,数字录音处理不连续时间信号(discrete-time signal)。图 1.13 以图表方式展示了数字录音和重放的过程。图中,麦克风感应气压变化并转化为电压,电压经线路通过模拟—数字转换器(analog-to-digital converter),通常缩写为 ADC(读为 A-D-C)。这个设备将电压在每一个采样时钟(sample clock)周期上转换为一连串二进制数(binary numbers)。这些二进制数则被储存在数字录音介质——一种存储器之上。

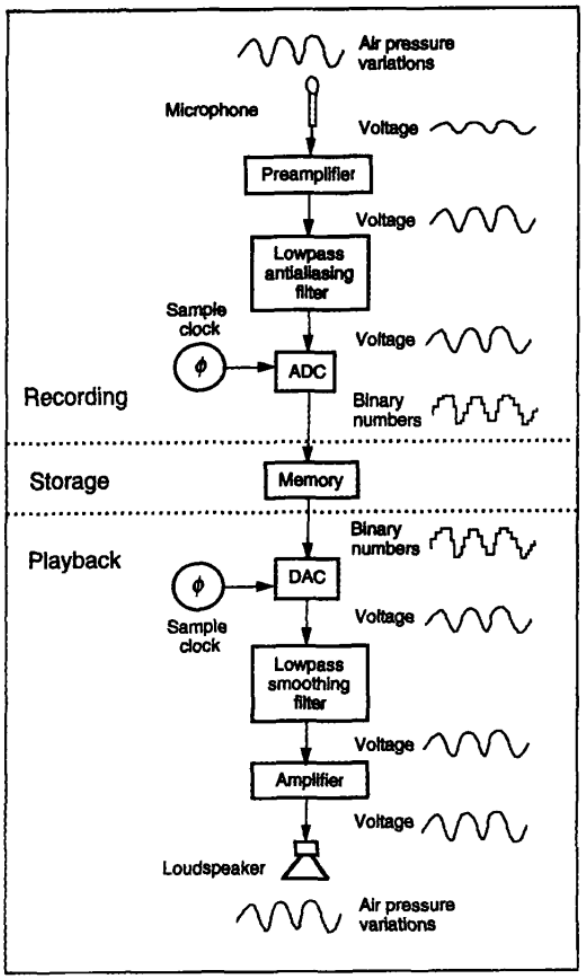


图 1.13　数字录音与重放概览
Recording=录音
Air pressure variations=气压变化
Microphone=麦克风
Voltage=电压
Preamplifier=前置放大器
Lowpass antialiasing filter=低通抗混叠滤波器
Sample clock=采样时钟
ADC- Binary numbers=二进制数
Storage=存储
Memory=存储器
Playback=重放

二进制数(Binary Numbers)

与采用由 0 到 9 这 10 个数字的十进制(或以 10 为基)不同,二进制(或以 2 为基)只采用 2 个数字,0 和 1。比特(bit)一词是二进制数字(binary digit)的缩略语。表 1.1 罗列了一些二进制数及其对应的十进制数。在二进制中可以有多种方法表示负值。在很多计算机中最左端的数被转译为符号指示器,如为 1 即为正数,如为 0 即为负数。(真正的十进制或浮点数同样可以被表示为二进制。见第 20 章更多关于在数字音频信号处理中的浮点数的内容。)

在一种录音媒介中将比特进行编码的物理方法有赖于那种媒介的属性。例如,在数字磁带录音机上,1 可能表示为一个正磁荷,而 0 则表示无磁荷。这与模拟录音带录音不同,后者以连续变化的脉冲来表示。在光学介质上,二进制数据可能被编码为在特定位置上的反射比的变化。

表 1.1 二进制数及其对应的十进制数

二进制	十进制
0	0
1	1
10	2
11	3
100	4
1000	8
10000	16
100000	32
1111111111111111	65535

数字—模拟转换(Digital-to-analog Conversion)

图 1.14 表示了转化一个音频信号(a)到数字信号(b)的结果。当听者希望再听一遍声音,这些数字就被从数字存储器上一一读取,然后传输到数字—模拟转换器上,简称 DAC(发音为"dack")。这个设备由采样时钟驱动,将一连串数字转换为一系列电压电平。由此,该过程即与图 1.13 所示相同,即一系列电压电平被低通滤波到时间连续的波形中(图 1.14c)放大,然后导入扬声器,扬声器的振动引起气压变化。这样,信号再度发出声音。

简而言之,我们可以将空气里的声音变为可以被数字化存储的一连串二进制数字。这一转换过程的中心构件是 ADC。当我们希望再次听到声音的时候,DAC 就可以把那些数字变回声音。

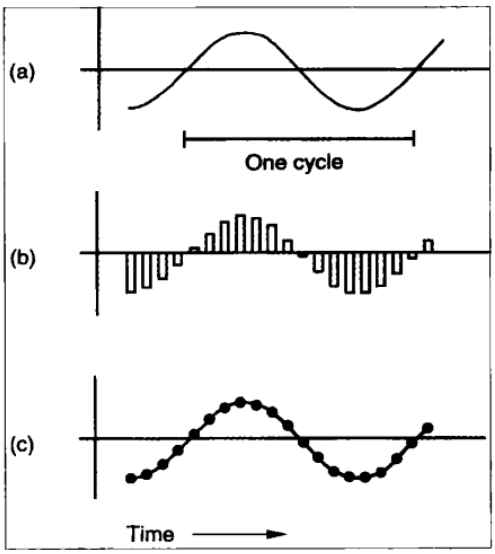


图 1.14　模拟和数字信号表示。(a)模拟正弦波形。波形下方的水平线指一个周期或循环。(b)是(a)正弦波形的采样版本，通常在 ADC 的输出端如此显示。每一个垂直柱形代表一个采样。每一个采样以一个数字被记录在存储器上，而这些数字则以这些垂直柱形的高度来表示。一个周期表示为 15 个采样。(c)是(b)波形的采样版本的重建。简要地说，将采样的上端以低通平滑滤波器相连，形成最终可以抵达听者耳中的波形。

数字录音与 MIDI 录音(Digital Audio Recording versus MIDI Recording)

这最后一点可以消除混淆：通过 ADC 生成的数字流与 MIDI 数据无关。(MIDI 指乐器数字接口 Musical Instrument Digital Interface specification，一种广泛应用于数字音乐系统控制的协议，见第 21 章。)数字录音机与 MIDI 音序器同样都是数字的，都可以记录多“轨”，但各自处理的信息量与种类却有差别。

当 MIDI 音序器通过键盘记录下人的行为，实际上只有相对很少量的控制信息从键盘传输给音序器。MIDI 并不传输声音的采样波形。对于每一个音符而言，音序器记录的只是开始和结束的时间、音高，以及起始音符的振幅。如果这个信息传输回原来弹奏的合成器，这就让合成器依照原来的声音弹奏，就像钢琴卷帘录音。当音乐家在 MIDI 合成器上以每分钟 60 拍的节奏演奏 4 个四分音符时，那么只有 16 个信息项捕捉了这 4 分钟的声音(4 个开始、结束、音高和振幅)。

相比之下，如果我们用连接着数字磁带录音机的麦克风录制同样一段声音，将采样频率设定在 44.1kHz，那么就有 352 800 个信息项(以音频采样的形式)来记录同样的声音(44 100×2 声道×4 秒)。数字录音需要大容量来存储。以 16 比特采样为例，录制 4 秒钟的声音就超过 700 000 字节。这是存储在 MIDI 上的数据的 44 100 倍。

由于所处理的数据量小，MIDI 音序器录音的一个优势是成本低。例如一个在小型计算机上运行的 48 轨 MIDI 音序器录音程序的价格大约是 100 美元，允许处理每秒 4 000 字节。相比之下，一个 48 轨数字磁带机则上万美元，每分钟可处理 4.6M 字节的音频信息，是 MIDI 数据率的上千倍。

数字录音的优势在于，它可以捕捉包括人声在内的任何麦克风能够捕捉的声音。MIDI 音序器录音则仅限于录制对一系列音符事件指示其开始、结束、音高和振幅的控制信号。如果你将 MIDI 电缆从音序器接入一个与最初演奏该音序的合成器不一样的另外的合成器，出来的声音就可能完全变了。

采样(Sampling)

图 1.14b 中所示数字信号与图 1.14a 所示的初始模拟信号有显著的不同。首先，由于信号是在特定时间里“被取样”，所以，数字信号只由特定的时间点来定义。图 1.14b 中的每一个垂直柱形代表一个原始信号的采样。这些采样以二进制数储存；图 1.14b 中的柱形越高，数值就越大。

用于表示每一个采样值的比特数，决定了系统能够控制的噪音电平和振幅范围。压缩光盘用 16 比特数来表示一个样本，而比这更多或更少的比特同样可用。我们稍后将在量化(quantization)的部分再回到这个话题。

获取样本的速率即采样频率(sampling frequency)，以每秒的采样数表示。这是数字音频系统中很重要的规约。采样频率常被称为采样率(sampling rate)，以赫兹表示。1 000Hz 被缩写为 1kHz，因此我们说：“压缩光盘录音的采样率是 44.1kHz，”这里的“k”取自公制单位“kilo”，意为“千”。

模拟信号的重建(Reconstruction of the Analog Signal)

在数字音频系统中常用的是 50kHz 左右的采样频率，也可见到更高或更低的频率。无论如何，每秒 50 000 个数是一个快速的数字流；这意味着每分钟的立体声就有 6 000 000 个采样。

图 1.13b 中的数字信号并没有显示柱形之间的值。一个柱形的持续性极窄，可能只持续 0.00002 秒(十万分之二秒)。这就意味着如果一个原始信号在柱与柱“之间”变化的话，这个变化并不会体现在柱形的高度上，至少等到获得下一个采样。专业术语称图 1.13b 中的信号是以离散(discrete)时间定义的，每一个这样的时间由一个采样(垂直柱形)来表示。

数字声的一个神奇之处就在于，如果这信号是带宽受限的，那么 DAC 以及附属的硬件就可以根据采样精确地重建原信号！这就意味着，在一定条件下，在“采样之间”丢失的信号可以被复原。这在数字经过 DAC 和平滑滤波器时会发生。平滑滤波器在离散样本之间“连结各点”(见图 1.13c 的点线)。由此，一个发往扬声器的信号看上去和听起来就像原信号一样。

混叠(迭影)〔Aliasing(Foldover)〕

采样的过程并不像看上去的那样直接。正如音频放大器或扬声器会产生失真,采样同样也会对声音变戏法 。图 1.15 就是个例子。采用图 1.15a 中所示的输入波形,假设该波形的采样取自图 1.15b 中的垂直柱形所表示的时间点(每一个垂直柱形生成一个采样)。如前所示,图 1.15c 的采样结果以数字形式存在数字存储器上。但当我们试图重建原始波形的时候,就会出现如图 1.15d 所示的完全不同的结果。

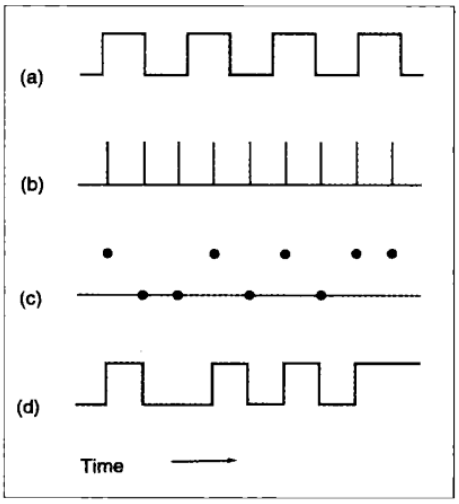


图 1.15　采样的问题。(a)要录制的波形;(b)采样脉冲;每当采样脉冲发生时,就生成一个采样;(c)经采样的波形并被存在存储器上;(d)当(c)的波形被发送到 DAC,输出信号则可能如此呈现(出自 Mathews 1969)。

为了能够更好地理解采样过程中会产生的问题,我们看一看当我们在不改变采样与采样之间的时长而只改变原始信号的波长(wavelength,一个周期的长度)会发生什么。图 1.16a 表示一个信号,其周期为 8 个采样的长度;图 1.16d则表示 2 个采样长的周期;图1.16g表示一个每 10 个采样有 11 个周期的波形。这个关系也可以表示为每个采样为 11/10 个周期。

同样,当每个采样组经过 DAC 和串联的硬件时就重建了一个信号并被传到扬声器。图 1.16c 中以虚线表示的信号就这样被比较准确地重建出来。在图1.16f中的采样结果可能就不那么令人满意。但在图 1.16i 中,重新合成的波形则在极为关键的方面与原始信号大相径庭。即重新合成的波形的波长(循环的长度)与原始波形的波长不同。在现实中,这意味着重新构成的信号听起来在音高方面与原始信号不同。这一类错位被称作混叠(aliasing)或迭影(foldover)。

发生混叠时的频率是可以预计的。假设,尽量取简单的数字,我们每秒取 1 000 个采样,那么,图 1.16a 中的信号就是每秒 125 个周波的频率(由于这里

每周波有 8 个采样，即 1 000/8＝125)。在图 1.16d 中，信号的频率是每秒 500 周波(因为 1 000/2＝500)。

在图 1.16g 中的输入信号的频率是每秒 1 100 周波。然而输出信号的频率却并不相同。在图 1.16i 中，你可以在输出波形中数出每周期为 10 个采样。实际上，输出波形发生在每秒 1 000/10＝100 周波频率上。因此，图 1.16g 中的原始信号的频率就已经被采样率转换(sample rate conversion)的过程改变了，而这对于音乐信号而言是不可接受的改变，必须尽量避免。

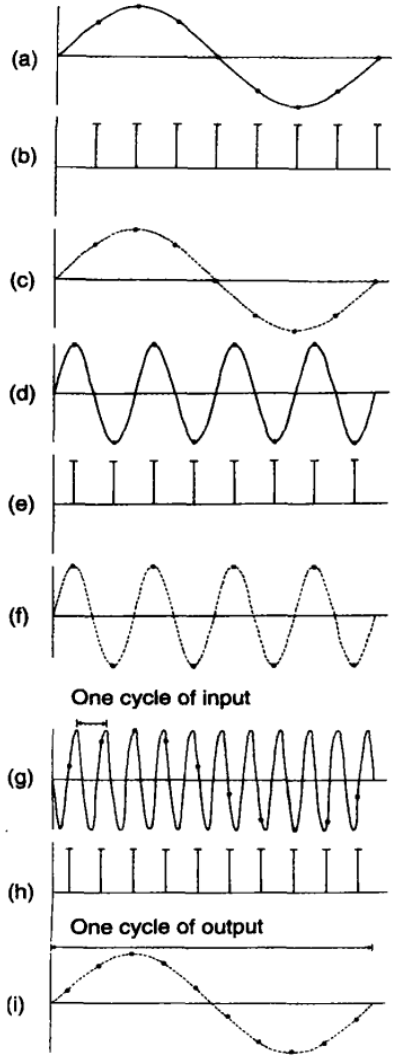


图 1.16 选影效应。在每一组的三个图表的底部，黑色点代表采样，连线则表示 DAC 重建的信号。正弦波形(a)的每个周波在(b)中采样 8 次。用同样的采样频率，(d)的每一个周波在(e)中仅采样 2 次。如果(e)中的采样脉冲向右移动，即便输出频率可能保持不变，但(f)的输出波形就可能发生相位移。在(h)中，这里有(g)中的 11 个周波的 10 个采样。当 DAC 试图重建信号时，如(i)中的虚线所示，是一个正弦波结果，但频率则因选影效应被完全改变的。注意在(g)的上方水平双向箭头，表示输入波形的一个周期，而(i)上方的箭头则表示输出波形的一个周期。
One cycle of input=一个输入周期
One cycle of output=一个输出周期

采样定理(The Sampling Theorem)

我们可以从图1.16概括出,只要原始波形的周期有不少于2个采样,我们就可以假定重新合成的波形会保持同样的频率。当每个周期少于2个采样,原始信号的频率(或音质)就会丢失。在这种情况下,新的频率可以根据以下公式获得。如果原始频率高于采样频率的一半,那么:

新频率=采样频率-原始频率

这个公式从数学角度看并不完整,但对于我们这里的讨论是充分了。它意味着:假定我们选择一个固定的采样频率,我们从一个低频信号开始,对其采样,采样后重新合成。当我们调高输入信号的音高(但仍保持采样率不变),那么,重新合成的信号的音高就与输入信号的音高相同,直到我们达到呼应采样频率的一半的音高。当我们进一步调高输入信号的音高,输出信号的音高就会降到最低频率!当输入信号达到采样频率时,全过程开始自行反复。

举一个实际的例子。假定我们将一个26kHz的模拟信号引入一个工作频率为50kHz的模拟-数字转换器中,转换器读取的是24kHz的音,因为50-26=24kHz。

采样定理描述采样率与被传递信号带宽之间的关系。H. 奈奎斯特(Harold Nyquist 1928)曾如下表述:

"对于任何一个给定的接收信号的畸变,传输频域必须与信号速率成正比上升……结论是:该频带与速度成正比。"

采样定理的要旨可以简述如下:

"为了能够重建一个信号,采样频率必须至少2倍于被采样信号的频率。"

为了纪念他对采样定理做出的贡献,数字音频系统所能产生的最高频率(采样率的一半)被称为"奈奎斯特频率"(Nyquist frequency)。这运用在音乐中,奈奎斯特频率通常是人耳所能听见的最高频率,超过20kHz。这样,采样频率即可设定在至少2倍,即40kHz以上。

在一些系统中采样频率被设定在略高于最高频率的2倍,因为转换器和相连的硬件通常无法完美地重建接近采样频率一半的信号(图1.16f显示了此种情况下的理想化重建)。

理想采样频率(Ideal Sampling Frequency)

关于什么样的采样频率才是最理想的录制和复制音乐的频率的问题一直众说纷纭。部分原因出自数学理论和工程实践之间的冲突:转换器时钟不稳定、转换器电压呈现非线性或滤波器导致失真,等等。(见关于修正相位和过采样的部分。)

另一个原因是,很多人在听 20kHz 范围内的信息(称为"空气")时受听力限制(Neve 1992)。的确如此,鲁道夫·科尼格(Rudolf Koenig)在年届 41 岁时,他发现自己的听力已经延伸到 23kHz(Koenig 1899)。他所发明的精确单位设定了音效的国际标准。看来奇怪,新数字压缩光盘应该比 20 世纪 60 年代发明的留声机录音设备的带宽更窄,一个新型数字录音机也应该比一个 20 年老的模拟磁带录音机更窄。很多模拟系统可以生成超过 25kHz 的频率。科学试验也从物理学和主观观点两方面验证了 22kHz 以上的声音效果(Oohashi et al. 1991, Oohashi et al. 1993)。

在声音合成的运用中,在 44.1kHz 与 48kHz 标准采样率下缺乏"频率净空"会产生严重问题。这要求合成算法只生成 11kHz(44.1kHz 采样率)或 12kHz(48kHz 采样率)以上的正弦波,否则就会产生迭影。这是由于任何带有基音以外分音的高频分量含有超过奈奎斯特速率的频率。例如,12.5kHz 的音高,其第三谐音是 37.5kHz,这在一个以 44.1kHz 采样率运行的系统中会被反落到可听见的 6 600Hz 音。在采样和音高移位的应用中,频率净空的缺乏要求样本在向上调变之前通过低通滤波器。这些限制强加的麻烦带来了不便。

虽然高频采样率会有诸多问题,例如会增加存储量,以及对高品质重放系统的需求从而不使这些努力徒劳,但从艺术的角度来看,高采样率录音显然更受欢迎。

抗混叠与抗镜像滤波器(Antialiasing and Anti-imaging Filters)

两个重要的滤波器确保数字声系统正常工作。一个放在 ADC 之前以确保输入信号中不包含任何(或越少越好)高于一半采样率的频率。只要这个滤波器正常工作,那么,录制过程就不会产生混叠现象。所以,这个滤波器就很逻辑地被称为"抗混叠滤波器"(antialiasing filter)。

另一个滤波器被放置在 DAC 之后,主要的功能是将数字化存储的采样转化为平滑而连续的信号加以呈现。实际上,低通"抗镜像"(anti-imaging)或"平滑滤波器"(smoothing filter)生成了如图 1.14c 中表示的通过连接黑色实心点

而成的点线。

相位校正(Phase Correction)

在介绍了第一代数字录音和重放设备之后，很快就面临“相位校正”的问题。很多人抱怨数字声刺耳，问题可以回溯到 ADC 之前的抗混叠的“砖墙”(Brickwall)(Woszczyky and Toole 1983，Preis and Bloom 1983)。之所以称之为砖墙，是由于它们陡型频率抑制曲线(典型是奈奎斯特频率超过 90dB/8 度)。这些陡型滤波器可能在中端和高音频率(图 1.17)造成严重的延时(相位失真)。较小的频率相关延时由置于 DAC 输出端的平滑滤波器造成。

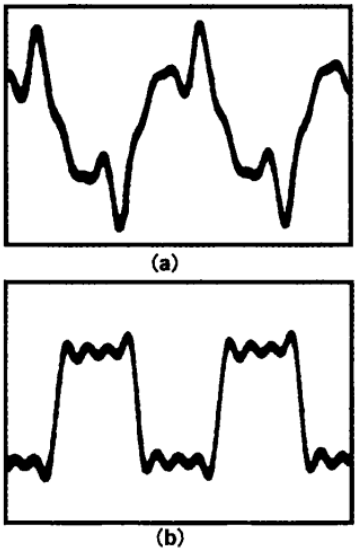


图 1.17　由抗混叠滤波器造成的失真。(a)被“砖墙”抗混叠滤波器干扰的 2.5kHz 方波；(b)相位校正方波。

没有一个模拟滤波器可以既呈极端陡型又在切割点上呈“线性相位”(phase linear，线性相位指的是完全没有或很少由滤波器引起的频率相关延时)。由此，陡型滤波器的效果溢出(spill over)到声音范围中。对于以 44.1kHz采样率录制的压缩光盘，奈奎斯特频率是 22.05kHz，陡型抗混叠则带来远远低于 10kHz 的失真(Meyer 1984)。这种类型的失真导致一种极其不自然的刺耳的高频声音。

有不少方法可以处理这个问题。最简单的方法就是，为了减少失真而置换滤波器的抗混叠的特性。一个较少陡型的抗混叠滤波器(例如 40-60dB/8 度)就较少带入失真，但这样就会有产生极高频声音的叠像的危险。另一个解决方法是在 ADC 之前再置入一个“时间校正滤波器”(time correction filter)，从而让输入信号的相位关系略有偏移，以在录音过程中保留初始相位关系(Blesser 1984，Greenspun 1984，Meyer 1984)。不过，如今高科技的解决方法是采用“过

采样"(oversampling)的技术在系统输入与输出阶段进行相位校正转换。我们将稍后讨论"过采样"。

量化(Quantization)

此前已经讨论过的在不连续时间的间隔的采样,构成了数字与模拟信号之间的一个主要差异。另一个差异是"量化"(quantization)或离散幅度分辨率。由于数字只能在特定范围和精度内被表示,而这又因运用的硬件而不同,所以,被采样的信号是不会呈现为任意假想值的。量化的应用在数字音频品质方面是一个重要的因素。

量化噪音(Quantization Noise)

采样通常以整数表示。例如,如果一个输入信号的在对应 53—54 数值之间的电压值,那么,转换器有可能以四舍五入的方式将其分配在 53 的值上。通常,对于每一个选定的采样,采样值与原始信号值有少许差异。这一数字信号的问题叫做"量化误差"(quantization error)或"量化噪音"(quantization noise)(Blesser 1978, Maher 1992, Lipshitz et al. 1992, Pohlmann 1989a)。

图 1.18 表示了一系列可能发生的量化误差的种类。当输入的信号像交响乐那样极为复杂,我们只听误差,如图 1.18 下方所示,它就听起来像噪音。如果误差较大,那么就能听到像模拟录音带在输出系统中发出的那种嘶嘶的声音。

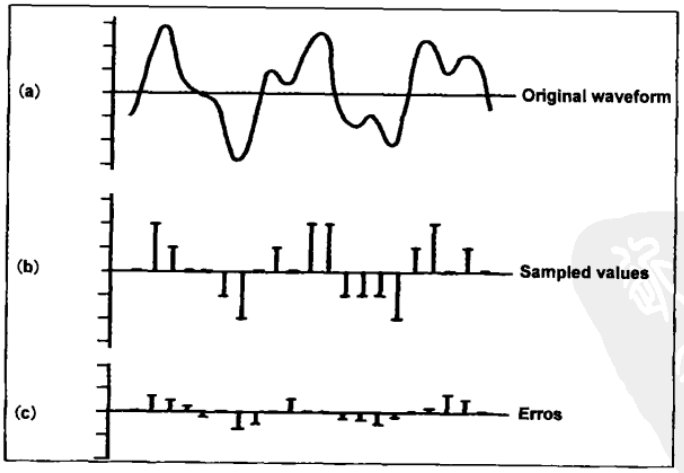


图 1.18 量化效应。(a)模拟波形;(b)波形(a)采样版本。每一个采样都被赋予一个特定的值,该值由左侧坐标上的水平短横线标示。采样和原始声音之间的误差在(c)中标示,每一个柱形的高度就代表量化误差。
Original waveform=原始波形 Sampled values=采样值 Erros=误差

量化噪音出于两个方面：信号自身，以及信号在数字形式中表示的精确度。要解释信号的灵敏度，只需留意一下在模拟式的磁带录音机中，磁带施加的那种带有柔和光晕且持续不断的，甚至在安静的空白部分也能听见的噪音。而在数字系统中，当没有录制（或默声）时就不会产生量化噪音。换句话说，如果输入的信号是无声，而当信号用一系列采样表示时，每个呈现的就都是零，在图 1.18c 中出现的细微差别就消失了，也就意味着量化噪音消失了。另一方面，如果输入信号是一个纯粹的正弦曲线，那么量化误差就不是一个随机功能而是一个确定性舍位效应（Maher 1992）。这种粗糙的声音被称为“粒化噪音”（granulation noise），可以在极低正弦曲线降为无声的时候听到。当输入信号完整时，粒化噪音就随机成为白噪音。

第二个量化噪音的方面是数字信号的精确度。在 PCM 系统中，由一个整数代表每一个采样值〔线性 PCM 系统（linear PCM system）〕，量化噪音直接与用于标示采样的位数关联。这一规格即系统的采样带宽（sample width）或量化水平（quantization level）。图 1.19 将 1 比特与 4 比特量化这两个分辨率加以比较，图解了不同量化水平的效应。通常在线性 PCM 系统中，用于表示采样的比特数越多，量化噪音就越少。图 1.20 就显示了通过增加分辨率的比特数，正弦波的准确率获得了显著的提高。

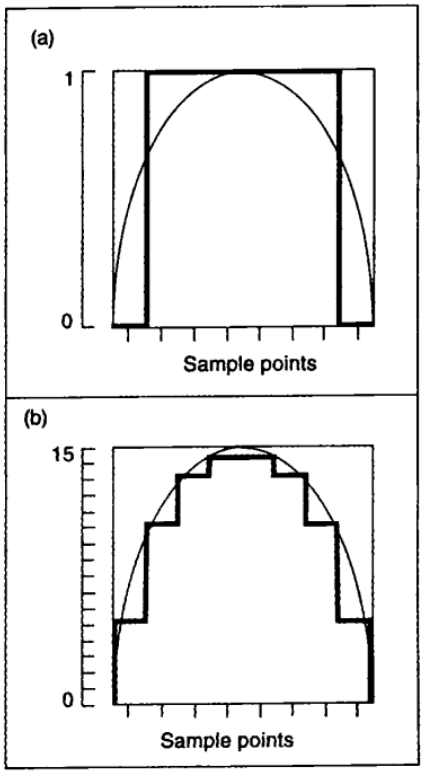


图 1.19　比较 4 比特量化与 1 比特量化的精确度。细曲线代表输入波形。(a)1 比特量化提供 2 个振幅分辨率的水平，而(b)4 比特量化则提供了 16 个不同的分辨率水平。Sample points=采样点

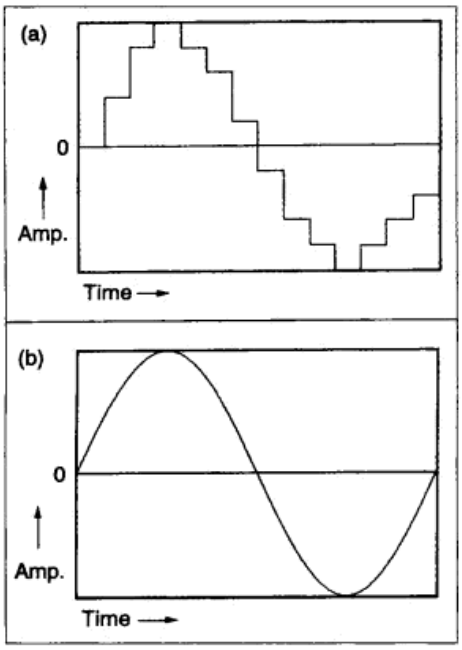


图 1.20 在平滑正弦上的量化效果。(a)10个量化水平的"正弦"波,针对一个由4比特系统发出的相对较大的声音;(b)由8比特系统发出了平滑的正弦曲线。Amp.=振幅 Time=时间

量化标准会被使用高速"1 比特"转换器的"过采样"(oversampling)系统干扰。使用"1 比特"转换器的量化系统比远远优于运用实际的 1 比特。参见稍后关于过采样部分。

低电平量化噪音与抖动(Low-level Quantization Noise and Dither)

尽管在没有信号输入的情况下,数字系统表现为没有噪音,但是在非常低(但非零)的信号电平下,量化噪音还是一个有害的东西。非常微弱的信号也会引发微弱的变异。这些 1 比特的变异看起来就像一个奇次谐波丰富的方波。不妨设想一个钢琴音的衰退吧,其高列分音由波动到逐渐衰减——直到最低的程度即改变了性质并成为一个刺耳的方波。方波的谐波甚至可能扩展到超出奈奎斯特频率,导致混叠现象并引入初始信号中没有的新频率成分。如果信号保持在一个较低的监听级,这些人为因素可能被忽略,但是如果信号在较高的电平上被听见或是对它进行数字化的重新混合以达到较高的电平,那么这些人为因素将变得很明显。因此,信号在输入阶段尽可能的精确量化是很重要的。

面对这些低电平量化问题,一些数字化录音系统的处理方式乍看起来显得很奇怪。在模数转换进行之前,它们在信号中引入了少量(叫做抖动)的模拟噪音(Vanderkooy and Lipshitz 1984,Lipshitz et al. 1992)。这样做使得模数转换围绕低电平信号产生一个随机的变化,它抹平了方波谐波的有害的影响部分(图 1.21)。由于抖动的存在,量化误差(它通常是随信号而定)被转化成一个

与具体信号没有联系的宽频带噪音。就像前面提到的钢琴音的渐弱现象，随着声音平滑地衰退到一组低电平随机噪音，其变化效果就像一种“软着陆”。增加的噪音数量通常近似于 3dB，但是人耳能重新构建这些振幅降低到抖动信号之下的乐音。参见 Blesser(1978, 1983), Rabiner and Gold(1975), Pohlmann (1989a), and Maher(1992)，可以了解量化噪音更多的细节以及减少量化噪音的各种方法。Lipshitz, wannamaker, and Vanderkooy(1992)提出了一种关于量化和抖动的数学分析方法。参见 Hauser(1991)对过采样转换器中有关抖动的讨论。

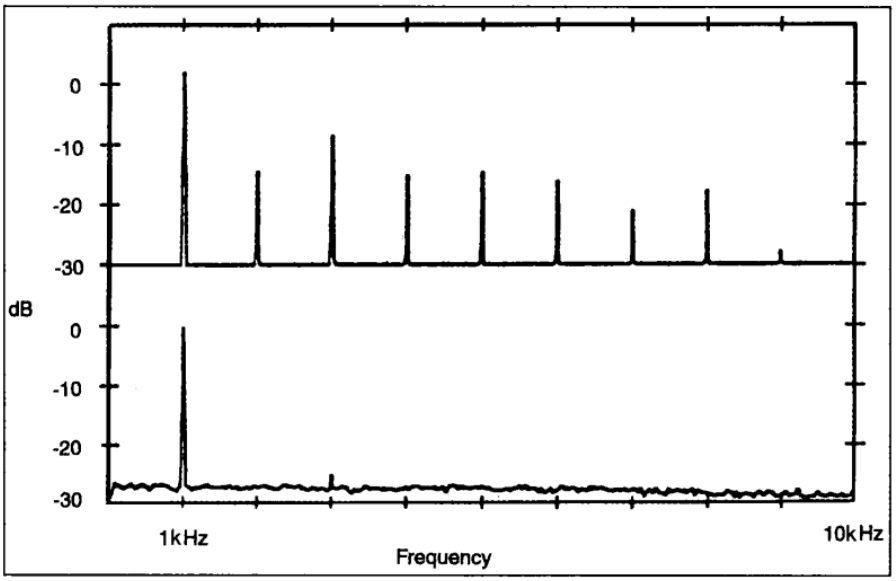


图 1.21　在数字系统中，抖动减少了谐波失真。图中上半部分显示了一个振幅为 1/2 比特的 1kHz 的正弦波的频谱。注意由于 ADC 活动产生的谐波。图中下半部分显示的是在转换前和振幅大约为 1 比特的抖动产生后的同一信号的频谱。在较宽的频谱噪音中，只有第三谐波中的很少一部分噪音仍然存留着。人耳可以辨别噪音层下的这些正弦波。

抖动技术对于精确到 20 比特的转换器来说也许不必要，因为低比特代表了一个极弱的信号，它们低于最大信号达 108dB 以上。但是，举例来说，当从 20 比特到 16 比特转换信号格式的时候，抖动技术对于保证信号的高保真则是必须的。

转换器线性特征(Converter Linearity)

转换器能导致多种失真(Blesser 1978, McGill 1985, Talambiras 1985)。这里要指出的一个实例就是 N 比特的转换器通过 N 比特的输入与输出未必能准确地响应出它的整个动态范围。当 N 比特的转换器的精度只能达到 2 的 n 次

方(2^n)的一部分时,转换器的线性度即模拟和数字输入与输出信号级别必需和它们可量化的值相一致。这就是说,一些转换器使用 2^n 作为基本梯级(步长),但是,这些梯级(步长)不是线性的,这就将导致失真。因此,这好比看到一个"18 比特的转换器",而它只有"16 比特线性",这是很有可能的。这样的一个转换器可能比一个普通的可能达不到 16 比特的线性度的 16 比特转换器更好。(关于这些问题的讨论参见 Polhmann 1989 a。)

数字音频系统的动态范围 (Dynamic Range of Digital Audio Systems)

标准数字声音设备的规范说明详细阐述了系统的精确性或者说是精度,它反映了系统用来存储每个采样的比特数。每一个采样的比特数量在计算数字声音系统的最大动态范围时是很重要的。总的来说,动态范围是系统产生的最大的和最小的声音之间的差,它用分贝(dB)来计量。

分贝(Decibels)

分贝是用来指示电伏,强度特别是在高频系统中的能量关系的一个计量单位。在音频测量方法中,分贝的取值范围显示了一个数据水平与一个参考水平的比率,根据这种关系:

分贝数 $=10\times\log_{10}$(电平/基准电平)

分贝	具体声音
195	火箭点火声
170	喷气式飞机发动机声
150	螺旋式飞机声
130	摇滚音乐会(持续地)
120	75 人的交响乐队(瞬态高度)
110	大手提钻的声音
100	钢琴声(瞬态高度);公路上汽车声
90	叫喊声(普通级别)
80	

续表

分贝	具体声音
70	说话声(普通级别)
60	
50	
40	
30	悄声说话的声音
20	经过音响处置的录音室
10	
0	听觉阈值

图 1.22　典型的来自不同声源的声音力度水平。所有数字的得出基于这样的公式：0dB=$10^{-12}/m^2$。

其中的基准电平通常就是听觉阈值($10^{-12}/m^2$)。分贝的算法基础意味着如果两个音一起发响，每一个是 60dB，增加的级数就只有 3dB。在强度上的数百万倍的增加将导致 60dB 的提升。(见第 23 章，Backus 1977，Pohlmann 1989 有关分贝的更多细节。)

图 1.22 显示了分贝刻度表和一些从 0dB 开始的大致的声音能量级别。两个重要事实描述了一个数字音频系统所需的动态范围：

1. 人耳听觉范围从大约 0dB(大致地在这个级别上最微弱的声音可以被听到)扩展到大约 125dB(这大致就是人耳听觉阈值)。

2. 两个声音之间稍稍小于 1dB 的差别相当于可听声音的最小区别值。

这些数字随着年龄、受训练情况、音高以及不同的个体而变化。

在录制音乐时，如果我们要再现音乐的丰富表现力，获得尽可能宽的动态范围是很重要的。例如，在一个现场管弦乐队音乐会上，动态范围可以从“寂静”变化到 60dB 的一个乐器的独奏，甚至是整个乐队全奏时的超过 110dB。模拟磁带设备的动态范围由模拟录音过程中的物理性所指示。使用专业没有降噪设备的磁带录音机，它可承受一个 1kHz 音高的音产生的大约 80dB 动态变化。(降噪设备在产生大量失真的代价下可以增加动态范围，见第 10 章可以获得更多关于降噪的内容。)

利用一个没有较宽动态范围的媒介来录音时，(例如批量生产的模拟盒带)，声音较弱部分将被操作人员处理放大，而大音量的部分将被处理变小。如果这些问题没有处理好的话，那么，最大音量的部分在录音时将产生失真，而最小声音的部分将被嘶嘶声和其他噪音所掩盖。

数字系统的动态范围(Dynamic Range of a Digital System)

要计算一个数字系统最大的动态范围值,我们可以使用下面简单的公式:

最大的动态范围分贝数=比特数×6.11

在这个公式中的 6.11 是理论上最大值的近似值(van de Plaasche 1983, Hauser 1991),实践中,6 是一个更实际的数字。这个公式是由 Mathews (1969)和 Blesser(1978)提出的。

因此,如果我们使用 8 比特的硬件系统来录音,那么它的动态范围的上限大约是 48dB,这比模拟磁带录音机的动态范围要小一些。但是,如果我们使用 16 比特的采样率,动态范围将增加至最大到 96dB,这是一个重要的提高。一个 20 比特的转换器可以提供 120dB 的动态范围,这个范围基本相当于人耳的听觉范围。量化噪音和比特数有直接的关系,因此,较微弱的声音部分由于不会使用到系统的整个动态范围,故听起来将更清楚一些。

这个讨论假设我们正采用一个线性的 PCM 方案,它使用整数来代表每一个采样的数值并储存每一个采样。Blesser(1978), Moorer(1979b) and Pohlmann(1989a)回顾了其他一些编码方案,这些方案可以将声音转换成了十进制的数字、分数以及获得在连续样本之间的差分等。其他的编码方案通常具有这样一个目标,即减少系统必须存储的比特数。在一些实际应用中,就像压缩磁盘介质,它是将图像和音频数据混合在一起(CD-ROM, CD-I 等),为了合理安排磁盘上所有需要的信息,采用储存更少量的比特数的办法而牺牲一点动态范围也是必要的。当然,节约空间的另一个方法就是降低采样率。

过采样(Oversampling)

我们已经重点讨论关于线性 PCM 转换器的内容。线性的 PCM 数模转换器基本上就是使用一个直接的步骤将一个采样转换成一个模拟电压。相比较利用线性 PCM 转换器,过采样的转换器比起在使用录音媒介的存储来说,在转换的过程中要使用更多的采样。然而,过采样的理论是一个更高层面的话题,在这里我们的目标是阐述一个基本概念,本书后面列有相关文献,有兴趣进一步关注这个话题的人可以阅读相关参考资料。

过采样不是一种技术而是一系列增加转换器精确度的方法。我们要区别

以下两种不同类型的过采样：

1. 多比特过采样数模转换器(DAC)是由飞利浦公司的工程师于 20 世纪 80 年代早期开发的。它主要应用于 CD 播放器(van de Plassche 1983，van de Plassche and Dijkmans 1984)。

2. 1 比特过采样，它使用西格玛-得塔(sigma-delta)调制技术或是在目前更多应用于模数转换和数模转换中的相关方法(Adam 1990，Hauser 1991)。

第一种方法在采样时钟的每一个单位时间内要转换大量的比特数值(例如 16)，而第二种方法在同一时间内只要转换一个比特，但要在一个非常高的采样频率下进行。这两种方法的差别并不总是那么明显的，因为一些转换器两种方法都使用了。这就是说，它们首先执行多比特过采样然后再使用 1 比特的过采样方法。

多比特过采样转换器(Multiple-bit Oversampling Converters)

在 20 世纪 80 年代中期，CD 制造商在他们产品中开始采用飞利浦公司的数模转换(DAC)芯片，该芯片主要是针对家庭用户进行音乐欣赏而开发的。这些转换器的优点主要在于它们的数字滤波器相较于在常规的数模转器中使用的旧式模拟滤波器可以提供更多线性声象响应。(基于这种概念而生产的模数转换器也已经出现，但我们还是把讨论限制在数模转换器方面。)在 CD 播放器中，每个通道每秒可以存储 44 100 个 16 比特的采样，但是在回放时它们依托系统的帮助可以过采样到 4 倍(176.4kHz)甚至是 8 倍(352.8kHz)。这种效果是通过在每两个原始样本之间插入三个(或者七个)新的 16 比特的样本而实现。在同一时间所有样本被线性的相位数字化滤波器进行滤波，而不是由产生相位失真的模拟滤波器来滤波。〔这种数字化滤波器就是一个有限脉冲响应(finite-impulse-response)滤波器，相关内容见第 10 章。〕

除了声象的线性度以外，过采样的一个主要优点就是它能在整个音频带宽内简化量化噪音，也就是增加信噪比。这些设备的设计理念来自于转换器的基本应用原理，即整个量化噪音的能量和转换器的分辨率是一致的，而与它的采样率无关。理论上，这些噪音均匀地分布在系统的这个带宽上。更高的采样在更宽的频率范围上会分布着等量的量化噪音。后面低通滤波在音频带上减少了量化噪音的能量。因此，一个 4 倍的过采样录音有少于 6dB 的量化噪音(相当于增加一个比特的分辨率)，一个 8 倍的过采样录音有小于 12dB 的噪音。这个系统的最末段是一个逐渐倾斜的模拟低通滤波器。例如说，通过音频带宽的

移动，它可以除去所有高于 30kHz 以上的因素。

1 比特过采样转换器(1-bit Oversampling Converters)

尽管 1 比特过采样转换器的理论可以追溯到 20 世纪 50 年代(Cutler 1960)，但是这种技术最终使用到数字音频系统也经历了多年的时间。1 比特过采样转换器包含了一系列的不同技术，主要有：sigma-delta、delta-sigma、noise-shaping、bitstream 和 MASH 等转换器品牌。这些转换器具有相同的思路，即它们在同一时间内只取样一个比特，而且是在较高的采样频率上。这些转换器并没有尝试使用一个单一的采样来描述整个波形，而是在连续的样本之间测量它们的差别。

1 比特转换器的应用利用了信息理论的一个基本法则(Shannon and Weaver 1949)，这个法则指出，我们可以交替使用一个采样率的采样宽度并在相同的分辨率下进行转换。也就是说，一个具有 16 倍储存采样比率过采样的 1 比特转换器相当于一个没有过采样的 16 比特的转换器。它们能处理相同数量的比特数。当它处理的比特数比输入的比特数还要大时，过采样的优势就产生了。

对于使用者来说，准确理解 1 比特转换器过采样的比率并不容易，因为，显示究竟有多少个比特数值被处理或是被存储，并不是必须的。要解读这个过采样规范的方法就是测定被处理的所有比特数的总量，根据以下公式：

过采样因数×转换器宽度

例如，一个使用 1 比特转换器的 128 倍过采样系统每个采样时段可以处理 128×1 的比特数。这样的处理能力，可以和一个传统的处理 1×16 比特数或者少于 8 倍数据的 16 比特的线性转换器相比较。理论上，这个 1 比特转换器可以获得更清楚的声音。然而，在实际应用中，做这样的测定有时会被转换器给弄混淆了，因为，转换器要使用好几个采样步骤并改变内部比特宽度。

无论如何，过采样的所有优势是由于有了 1 比特转换器，在数字滤波器的帮助下它提升了分辨率和声象的线性度。在多比特转换器技术条件下难以达到的高采样率，在 1 比特转换器技术下变得容易实现了。在 MHz 的过采样率范围中，它可以实现每个样本的 20 比特的量化。

1 比特过采样转换器应用的另外一个技术就是噪音修正(noise shaping)，它具有好几种格式(Hauser 1991)。该技术的基本理论阐明：在过采样过程中产生的再量化错误由一个高通滤波器在输入信号的反馈循环中被转移到一个

超出了音频带宽、更高的频率范围。这些噪音修正循环通过高通滤波器只发送再量化错误，而不是音频信号。

过采样转换器的末端部分是一个信号抽取器(decimator)/滤波器，它们减少了需要存储(用模数转换器，ADC)或是回放(用数模转换器，DAC)的信号的采样比率，并用低通滤波器过滤了信号。在噪音修正的转换器中，信号抽取器/滤波器还去除了再量化的噪音，并导致了信噪比戏剧性的提升。由于二级噪音修正技术的应用(这样的称谓是由于在反馈循环中应用了二级高通滤波器)，1 比特转换器的最大信噪比水平每个过采样八度大约可以达到 15dB(2.5 比特)，减去一个固定的 12.9dB 的损失(Hauser 1991)。因此，一个 29 比特的过采样元素给一个 16 比特的转换器提高了 10 比特的或是 60dB 的信号比。

关于过采样的信号修正转换器的内部结构的更多内容，参见 Adams(1986，1990)，Adams et al. (1991)，and Fourré，Schwarzenbach，and Powers(1990)。Hauser(1991)写过相关的概述文章来阐述它的发展历史、理论和过采样技术的实际应用，文章中也包括许多参考书目。

数字音频媒介(Digital Audio Media)

音频采样可以存储在任何使用数字录音技术(例如电子磁、光磁或光学技术)的数字媒介中，如磁带、光盘或是集成电路。使用一个给定的媒介，数据可以被以多种格式写入。一种格式就是一种数据结构(见第 2 章内容)。例如，一些数字音频工作站的制造商推出了专有的样本磁盘存储格式。由于技术和市场的原因，新媒介和格式不断出现。表 1.2 列出了一些媒介和它们与众不同的特点。

一些介质具有每秒可以处理更多比特的能力，所以具备了进行高质量的录音的潜力。例如，某个数字磁带录音使用恰当的转换器可以对每个样本进行 20 比特的编码处理(Angus and Faulkner1990)。一个硬盘能够以超过 100kHz(同时处理几个声道)的速率来处理 20 比特的样本，而半导体的媒介(存储芯片)它们潜在的采样宽度和采样率将更大。

媒介另外的一个特点就是它的使用期限。使用镀金防腐的玻璃材料制成的档案级磁盘可以使用几十年，历经无数次的使用(Digipress 1991)。像 DAT 那样的磁质媒介和软盘都是非常便宜而且便于携带的，就是使用时间有限。

数字存储媒介的另外一个优势就是我们将数据从一个媒介向另外一个媒介转移时，它的内容基本没有损失。我们可以对一个原版录音或者它的复制品任意次地复制。这也意味着我们可以将一个录音从一个便宜的媒介上(例如

DAT)转移到一个可以读写的媒介上(例如磁盘),这种方式更适合编辑和处理。当我们完成了编辑后,还可以将样本转回到 DAT 上。这些转移行为的实现主要是通过数字输入和输出连接器(回放和录音系统上的硬件)和标准数字音频传输格式(在设备之间发送音频数据的软件协议,见第 22 章)来完成。

表 1.2 数字音频媒介

媒介	连续的或随意的访问	备注
固定头(磁质带)	连续的	典型运用是在专业的多轨(24,32,48 轨)录音;几种格式保存;有限的编辑。
旋转头影带(磁质带)	连续的	专业的和民用格式;民用消费录影盒带比较便宜;左综合编辑时需要两台机子(见第 16 章);多种磁带格式(U-matic、Beta、VHS、8mm,等等)还有三种不可兼容的国际视频编码格式(NTSC、PAL、SECAM)。
旋转头音频带(磁质带)		四轨录音的专业的 Nagra-D 格式。
数字音频磁带(DAT)(磁质带)	连续的	小型便携式盒带录音机;广泛的兼容;一些近期还可以处理 SMPTE 编码(见第 2 章)。
数字压缩盒带(DCC)(磁质带)		可以在传统模拟盒带录音中使用的数字格式;使用数字压缩;和 CD 格式相比它的音质较差。
硬盘(磁质和光介质)	任意的	不可拆卸的硬盘,有较快的速度(访问时间只需几毫秒);可移动的硬盘对于文件备份和声音样本的移动也是很方便的。注意:计算机上可移动的光学硬盘通常和 CD 格式不一样,虽然它们看起来很相似。
软盘(磁质)	任意的	软盘相比较小、便宜而且使用方便;但是它们运转慢能储存的声音文件很短。长时间的储存还不可靠。
索尼迷你磁盘(MD)(磁质)	任意的	是软盘的声音格式,使用声音压缩技术;和 CD 格式相比音质较差。

续表

媒介	连续的或随意的访问	备注
光盘(光学)	任意的	小而薄的光盘，存储量最大可达 782M，74 分钟。档案级的磁盘可以使用几十年，可以重放影像和音频。根据使用的情况，有从语音(CD-ROM)到高保真(20 比特格式)等各种音频质量级别。同其他任意存储媒介相比，压缩光盘的访问速度较慢(Pohlmann 1989b,d)。
半导体存取器(电子)	任意的	非常快的访问时间(少于 80 毫微秒)；良好的临时存储(为了编辑处理)但是较大的数据库价格较贵。

合成与信号处理(Synthesis and Signal Processing)

正如我们所见到的，采样将声音信号转成二进制数字，实现了数字音频录音。音乐应用中，采样不仅仅可以用于录音，还可以进行合成和信号处理。合成就是利用算法手段进行样本数据流处理的过程。第二部分的第 6 章中列举了许多种合成方式。

信号处理传输着样本数据流。在音乐中，我们使用信号处理工具来进行声波处理。典型的音频信号处理包括以下几种：

- 动态范围(振幅)处理——重新修正声音的振幅
- 混音——合并多轨音频，包括交错渐变
- 滤波器和均衡器——改变声音的频谱
- 延时效果——回声、合唱效果、镶边、整相
- 卷积——时间同步和频率变化
- 空间投射——包括混响
- 降噪——清除坏的录音
- 采样率转换——改变持续时间而不影响音高，或改变音高而不变动持续时间
- 声音分析、转化和再合成
- 时间压缩与扩展——不影响音高和声音质量的情况下改变声音的持续时间

尽管这是一个相对新的领域，数字信号处理(DSP)已经发展成为了一门大的理

论科学和应用艺术。第三、四部分阐述了在音乐中应用的 DSP 的基本概念。

结论(Conclusion)

这一章介绍了数字音频录音和重放的基本概念。这种技术还在不断得到发展。在模数和数模转换、信号处理和存储技术领域,还有许多可以发展的空间,我们可以期待在未来几年它还有更新发展。

当录音技术还在大步发展的时候,录音的艺术开始朝着两个相反的方向发展。一方面是录音领域的"自然主义者"或是"纯化论者"流派,他们试图利用尽可能人为的手段重现理想的音乐厅效果。听他们的录音,就像我们也悬在空中一个理想的听音位置(话筒放置的地方)倾听艺术家的演奏。与之相对的发展方向主要应用在流行音乐、电子音乐和计算机音乐领域。人工声音舞台的构建,它可以实现声音移动以及幻觉声场,例如,多个声音从不同的空间同时发出。这种由不同信号处理操作产生的幻觉声场效果将在第三部分描述。

第 2 章　音乐系统编程
(Music Systems Programming)

柯蒂斯·阿博特(Curtis Abbott)

在计算机音乐领域若要有所创新，程序设计是必不可少的。程序设计中出现的一些问题不仅对计算机音乐具有重要的实际意义，其本身的理性也令人深感兴趣。因此，本章不仅讲述什么是程序设计，也论述其趣味之所在。虽然仅用一个浓缩的篇章来简述如此宏大的题目，我们还是希望能努力使读者窥一斑而知全豹，对程序设计有所了解。

本章将首先介绍主流程序设计语言的一些基本要素，其中包括控制结构和数据结构。此外，还将从音乐系统编程的角度纵览一些经过挑选的高级课题，如函数型编程、逻辑编程以及约束编程。

程序设计就是问题求解(Programming is Problem-solving)

从根本上说，程序设计就是问题求解。一个程序可从多个方面进行质量评估，包括实用方面的运行速度和美学上的简洁精致，但是对一个程序最重要的检验在于其正确性(correctness)——它是否解决了设计预期要解决的问题。这个标准并不像一眼看上去那样简单。应用计算机求解的问题变得越来越大，而且同我们的日常生活联系日趋紧密，要准确无误地将它描述出来变得越来越难。如果没有精确而不含糊的问题陈述，那么就很难判断一个程序是否正确。而且，当程序非常庞大或者含有不可预测的因素存在，就不可能进行详细彻底的测试。

因此，程序设计就是一个问题求解的过程，其中的问题通常难以给出完整准确的定义。的确，和非程序设计人员以及程序理论家所想象的往往不同，在有创造性的程序设计任务中，大多数的时候是在考虑如何把含糊不清的问题更准确地加以陈述，研究不同定义方式所产生的结果，同其他人(如潜在用户，尽管这种情况极少发生)讨论最佳方案等。

音乐系统编程也要面对一般程序设计所具有的技术和智力上的挑战。作曲中的问题很难进行准确完备的定义，这已是众人皆知，因此，满足某个作曲家的需要并不一定就能产生整体解决方案。有时候提供一个使用灵活的能让用户自由操作的工具包，要比试图一次性解决所有音乐问题效果好。很多音乐任务要求与计算机系统之外的不确定的组件(例如转换器、合成器或音乐家本身)进行通常是实时性的交互操作。实时演奏的需要对计算机软件提出了特别的要求，它必须对各种情况及时做出从容的反应。

程序设计语言的基本要素 (Basic Elements of Programming Languages)

最早的计算机编程类似以前的模拟合成工作室。用户设置开关转换器，通过接插线将计算机各个计算部件连接起来(见图 2.1)。今天，程序都是用程序语言写成的。在许多语言的使用中，新的程序设计语言被不断开发出来。通常，一种程序语言的开发是基于对程序设计的某种特别的观察方式。本章将着重介绍一些被广为接受的高级语言，而略去低级的面向机器的代码，如汇编语言。此外，本章中的大部分篇幅将介绍顺序编程(sequential programming)，即为一次只执行一个任务的计算机进行编程。并行程序(parallel programs)能运行在可以同时执行多个操作的计算机上，它正变得越来越重要，但是要熟悉编程还是从顺序程序设计开始最容易上手。

程序设计规划可以分为两个大的方面。控制(control)与计算机所执行的操作及其发生的顺序有关，而数据(data)与计算机执行操作所涉及的存储的对象有关。在这样的讨论框架下，我们便默认以下的隐含假设，即计算机由两个部分组成：主动作用的部分称为中央处理单元(central processing unit)(通常称作 CPU 或处理器)，控制权居于其中；被动作用的部分称为存储器(memory)，数据居于其中。这条假设适用于目前的计算机，不过，随着计算机的发展，也许它会变得不大合适。

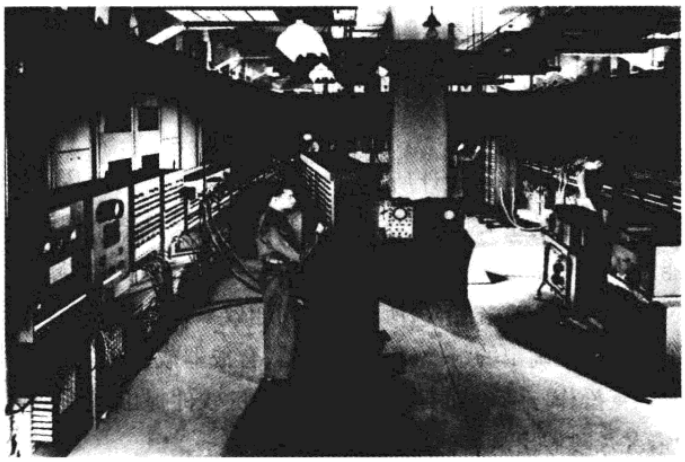


图 2.1 欧文·戈尔茨坦(Irwin Goldstein)下士通过设定功能开关为埃尼阿克(Eniac，电子数字积分计算机)编制程序。1945 年 12 月 Eniac 计算机开始运行。这台具有历史意义的计算机有 18 000 个电子管，70 000 个电阻器，10 000 个电容器以及 6 000 个开关转换器。它有 30 米长，3 米高，1 米深，现在被华盛顿史密森博物馆(Smithsonian Museum)收藏。(照片使用由宾夕法尼亚大学摩尔电子工程学院授权。)

执行程序(Executing Programs)

计算机能够执行准确而具体的机器指令(machine instructions)。一条简单的机器指令所完成的任务有限,但把成百上千条指令连在一起就能有效地完成工作。与机器指令不同,程序是写好的文件,通常用高级语言写成,使人比较容易读懂。图形化编程语言,如 Max(Puckette and Zicarelli 1990)使得程序设计看起来更加直观。虽然本章中所举的例子均为文本的形式,但实际上我们提到的每一点也适用于图形语言。在详细介绍程序语言之前,我们有必要了解高级语言程序是如何被转化成无数条机器指令的。

实现一个可执行程序的常用方法是使用一个翻译程序(translator),它可将你所编辑的代码转化为可执行形式(executable form)或对象代码(object code)(机器指令的有序组合)。这个翻译程序被称为编译器(compiler)。从历史看,另外一种方法一直以来也很重要,即用一个叫做解释器(interpreter)的程序。解释器读取程序中的每一条语句,迅速将其翻译成机器指令,并在读取下一条语句之前将指令提交给计算机来直接执行。因此解释器在某种意义上可被视为另外一种不同的计算机,它能执行比任何真实硬件计算机复杂得多的程序。支持这种方法的一个有力论据就是编译器的翻译过程相对较慢,因为它更全局性地看待程序,并基于所发现的结果进行最优化操作。在程序开发中,减少花在对改动所作测试上的等待时间是很重要的。然而,现在功能强大的计算机比过去便宜了,尤其是由解释器执行的程序运行得更慢,这条论据就不那么具有说服力了。(对于某些程序设计语言而言,使用解释器仍有很多好处,这里限于篇幅就不深入探讨了。)

流程图与结构化程序设计(Flow Graphs and Structured Programming)

程序的可执行形式是必需的,它包含告诉处理器如何操作机器指令。然而,在程序被执行时,并非每一条指令每次都会被执行,因为计算机具有条件指令(conditional instructions),条件指令可以运行一个测试来决定哪一条指令将被执行。指令的条件执行(conditional execution)是计算机灵活性的关键。

当程序运行时,由于指令会或不会被执行均有可能,因此思考条件执行模式的方法就变得很重要。数学家们有一种他们称之为图(graphs)的便利的结构,这个结构被计算机科学家作为控制流图(control flow graphs)而采用。控制流图由一组方框表示,每个方框中包含一组不可分割执行的指令,在这些方框之间,用箭头来表示可能的路径。以这种方式画出来的图形也被称为流程图

(flow charts),我们会给出一些简单的示例。由这些图形所表示的指令执行的模式被称为程序的控制流。在本书中,我们通过引入结构化控制流(structured flow of control)来介绍程序设计构造。

结构化控制流的概念是更通用的领域即结构化程序设计(structured programming)(Dahl, Dijkstra ,and Hoare 1972)的一部分,它涉及一系列优良程序设计风格的规则和思想。接下来,我们将列举结构化程序设计的不同方面,并展示如何将结构化程序设计应用到控制和数据上。

过程(Procedures)

构造一个程序的最基本的方法是将它划分为一组过程(procedures)〔也称为函数(functions),子程序(subroutines),或者有时也称为方法(methods)〕。过程普遍存在于程序设计中。一个过程就是一个可以被调用(called or invoked)的程序单元,就像数学函数,比如加法或乘法。过程是可以被嵌套(nested)的,也就是说,一个过程可以在将控制归还给原始调用者之前,调用另一个过程(图 2.2)。

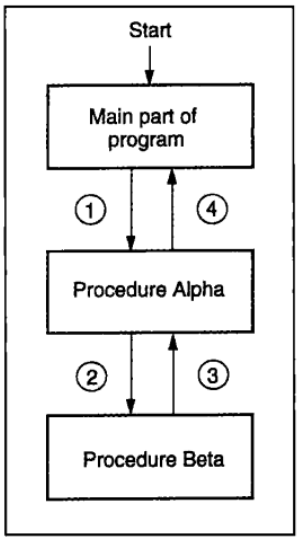


图 2.2 嵌套过程调用中的程序控制流。程序的主要部分调用过程 Alpha,该过程依次调用 Beta。当 Beta 完成后将控制返回给 Alpha,当 Alpha 完成后控制被返回给主程序。
Start=开始 Main part of program=程序的主要部分
Procedure Alpha=过程 1(Alpha) Procedure Beta=过程 2(Beta)

过程可以具有参数(arguments)(这一点也很像数学函数),这使其成为活跃的单元,适应于不同的环境,因为过程的参数是由调用它的程序部分指定的。通常,过程也返回一个值(value)——一个数字或者是能被调用部分使用的更复杂的数据结构。这一点也和数学函数类似——加法和乘法根据它们的参数产生结果。

程序设计语言一般会预定义一些过程以作为较大程序的构件。这样的

一些过程反映了计算机硬件的潜在能力，比如数字的加法和乘法。其他的则提供了对经常用到的输入输出设备的访问，比如在磁盘上保存或恢复文件的能力，或者在显示器上显示字符等。由程序员定义的过程则扩展了这些基本的能力。

赋值(Assignment)

大多数程序语言提供的另一个基本能力就是赋值(assignment)。赋值是改变一个变量(variable)的值的操作。在程序中，变量是内存中值所在位置的名字。这些值可以用许多不同的方式解释，如数字、字母、机器指令、其他内存地址等。因此，赋值可以改变与变量相应的内存中的内容。

在 C 语言中，赋值用等号表示如

```
a=b;
```

这行代码的意思是，将 b 所对应的内存中的值复制到 a 所对应的内存中。这和数学中的等号的意思大不相同。有些程序语言为了避免混淆，使用等号的变体来表示赋值。一种常见的变体形式如

```
a:=b;
```

这是 Pascal 语言的写法举例。一个语言的控制模式主要是基于赋值和过程调用的。接下来我们将就此进行讨论。

控制结构(Control Structures)

除了过程和赋值，程序语言还进行控制流的操作。贯通整个程序的最简单的控制方式是按顺序进行：每条程序语句依次执行。

选择(Alternation)

一些控制结构允许程序定义两个或更多可选(alternative)的执行通路。在这样的控制结构中，根据测试结果选择一个通路，控制流就从所选路径通过。

基于选择的控制结构可用下面这种先分支后又合并的结构流程图来表示。

图 2.3 显示了几个这样的结构，其中圆圈表示测试(在程序中这部分用来决定哪一分支将要被执行)。方框表示可能的选择。

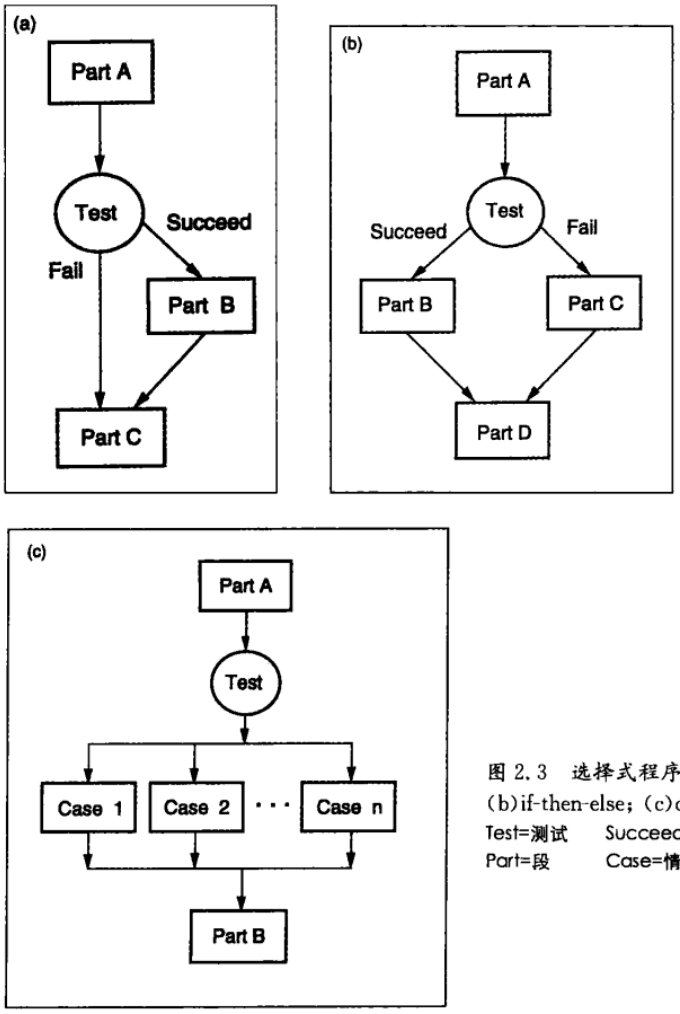


图 2.3 选择式程序控制结构。(a)if-then；(b)if-then-else；(c)case
Test=测试 Succeed=成功 Fail=失败
Part=段 Case=情形

选择结构中最简单的一种情况就是当我们不想运行程序的某个部分的时候，这种情况可以用 if-then 结构来表示。以下是 if-then 结构的形式：

if<*test*>then<*program part*>

其中<*test*>对应图 2.3a 中的圆圈，它决定<*program part*>是否将被执行。

另外一种稍微复杂点的选择情况是当要从两个不同程序部分中选择其一的时候，见图 2.3b。在程序语言中，这种情况被表示为在 if-then 后扩展出 else，其表达形式如：

if<*test*>then<*part1*>else<*part2*>

在现有的程序语言中，从数个可选对象中选择其一的方法有很多种。其中之一就是多级串联 if-then-else 结构，如下例所示：

if<*first test*>then<*part 1*>
else if <*second test*>then<*part 2*>
…
else if <*last test*>then<*part n*>

狄克斯特拉（Dijkstra1976）曾提出一个卫式命令表（guarded command list），对以上的结构作了有趣的总结。这是一个测试和结果的列表，用来概括级联的 if-then-else 结构，规定如果有一个以上的测试通过，那么与之相关的任何结果都可以执行。这条总结可能很有用处，因为它能让我们更清晰地表达自己对底层状况的想法，而不用去指定测试进行的顺序。尽管精致优雅，卫式命令表却并未得到广泛应用。

另一个被广泛应用的结构是 case。当一系列可选对象依赖于单一表达式时，这种结构非常有用。其构成的形式如下：

case<*expression*>
(<*first value*>=><*part 1*> <*second value*>=><*part 2*>)
…
<*last value*>=><*part n*>

其中，符号=>表示控制被传递给相应的程序部分。以上结构本质上等价于下面的 if-then-else 级联：

if<*expression*>=<*first value*>then<*part 1*>
else if <*expression*>=<*second value*>then<*part 2*>
…
else if <*expression*>=<*last value*>then <*part n*>

case 的构造较为优越，不仅因为它使得控制结构更清晰而易于阅读，还因为它通常能使计算机更快地对适当的程序部分做出选择。

重复(Repetition)

控制流的另一个重要模式是,一段程序一再地重复执行,这种模式通常被称为循环(looping)。在重复模式中,确定重复执行多少次以及如何结束重复是必需的。重复的一个典型构成是 while 结构,其形式如下:

while <*test*> do <*program part*>

它的效果是,在每次执行<*program part*>之前评估<*test*>,当<*test*>失败时重复结束。图 2.4 为 while 结构的流程图。

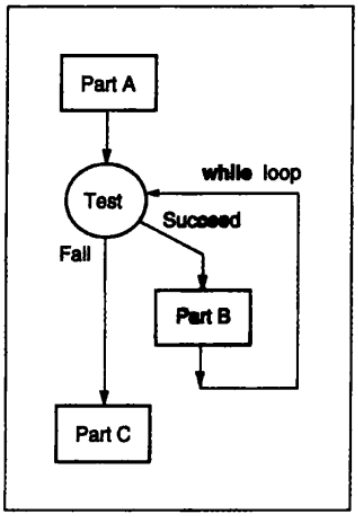


图 2.4 while 结构的流程图。当测试条件为真时,Part B 继续重复执行。否则,程序运行至 Part C。

程序语言中有许多重复结构。有些结构将测试放在程序部分之后(这保证了程序部分最少会被执行一次),有些则反用测试的含义(这使得重复一直执行到测试成功为止)。

关于重复结构的一个更重要的变体是那些支持使用与重复相关的辅助变量(auxiliary variables)的结构。辅助变量对于枚举数据集合中的元素非常有用,无论这些集合是如何组织的:矩阵、表、组或者其他形式。这种变体带来的可能性是巨大的,各种程序语言都提供了这样的结构,展示了该种结构非常可观的多样性。一个规范的例子是 Pascal 中的 for 语句(Jensen and Wirth, 1974)。其构成形式如下:

for <*auxiliary variable*> :=<*initial value*> to

<*final value*> do <*program part*>

这里，<*auxiliary variable*>是一个新建的整数，它随 for 一起变化。它的初值<*initial value*>在每次重复之后都会增加。重复操作直到变量到达终值<*final value*>为止。

Pascal 的 for 结构具有特殊的限制条件，这为其他语言设计者提供了一个很好的关于约束的范例。这个限制条件如下：

1. 除了被每次循环增加之外，<*auxiliary variable*>不能被改变。

2. <*final value*>只在循环开始前被估值一次。

这两条规则保证了当 Pascal 的 for 循环开始时，重复的次数是已知的。这样做的好处是 for 循环总是会结束，而且永远也不会失去控制。缺点是，在某些情况下，提前决定循环应被执行多少次是不合适或者不现实的。（在这种情况下，在 Pascal 中必须使用 while 结构。）

对比 Pascal 中的 for 结构与 C 中的类似结构（Kernighan and Ritchie 1978）是颇有意思的。C 中 for 语句的形式如下：

for (<*initialize*>; <*test*>; <*increment*>)
<*program part*>

这个结构的使用与 Pascal 中的 for 语句相近。但是，C 的 for 语句更容易被理解为一组用来重写该结构部件的指令，因为它没有 Pascal 中那样的约束。具体来说，<*initialize part*>最先被执行，接着如果<*test*>成功，<*program part*>被执行，接着是<*increment part*>，然后返回到<*test*>。这个 for 循环看起来很方便，使得程序员可以用一种清晰简练的方式来写循环。相比之下，虽然引入约束和辅助变量的 Pascal 的 for 循环缺乏全面性，但同时它更适合它所应用的场合。

这些例子难以穷尽所有重复结构的可能性。Knuth（1974）的文献包含一个迷人而又易于理解的有关重复结构的讨论。

数据结构(Data Structures)

在任何计算机中，数据都是以存储器中的比特（bits）（二进制数）来表示，比特可以是值 1 或 0。这些比特被组织为字节（bytes）或者字（words）。字节（几乎通行地）是 8 比特，它可以表示西方字母表中的许多字符。因此，字节经常被用来表示文字字符。

字是计算机中数据操作的“自然单元”,因而一个字所包含的位数对于不同的计算机是不同的。今天,大部分通用计算机使用 32 或 64 比特字,但是过去并不是这样,将来很可能也不是这样。

了解计算机字的变化性质很重要。举例来说,图 2.5 显示了一个 16 比特字及其相关的解释。对我们的目的而言,特定的解释并不重要。我们更想说明的是,如果一种位组合形式有任何意义,那么有关的解释我们就必须去了解。这就引出了下一个概念:数据类型(data types)。

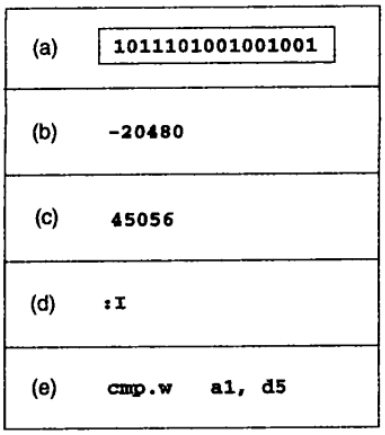


图 2.5 一个 16 位计算机字和一些相关解释。(a)二进制表示;(b)2 的补码;(c)未赋值整数;(d)ASCII字符;(e)机器指令。

数据类型(Data Types)

数据对象(一个或多个计算机字)的类型涉及它如何被解释。因此,我们说数据对象 x 的类型是整数、浮点数、字符,等等。

从“原始的”计算机内存,以及发生在它上面的机器操作的硬件层面来看,只存在少数几种类型。因此,如果我们暂停一个程序并随机查看内存中的一个字,要确定这个字表示什么将是非常困难的。它可能是一个样本值,一小组字符,一个对其他数据对象的引用,一个机器指令,或者其他任何东西。程序语言的一个重要任务就是,为程序员提供合适的类型概念,这个类型概念适用于该语言被设计用来解决的那类问题。通常,这个概念与机器层面的类型有些不同,像我们说的那样,它“更高级”。因此,程序语言执行(implementation)(也就是,编译器或者其他程序使得用高级语言编写的程序能够运行在机器上)的一个任务就是,对由两个层面提供的不同类型概念进行翻译。

什么类型的概念适当,这个问题一直有争议,至今没有结束。我们已经说过,这取决于程序语言被设计用来解决哪一类问题。这个观点通常作为一个主要原则被接受,但是这个观点并不清晰,也使得精确的技术讨论难以进行。就

像我们将要看到的，一些非常成功的程序语言使用与大部分计算机提供的类型非常接近的类型概念。另外，很多推动程序语言设计发展的学术工作已经将精力集中在更加抽象的类型概念上了。

类型声明(Type Declaration)

争议不仅在于程序语言应当表达什么样的数据类型，而且在于这些类型是否应当在编写程序时可见，如此，使得这一原本就使人困惑的课题更加令人困惑了。在那些类型可见的语言中，我们说变量和过程随它们的类型一起被声明(declared)。在这样的上下文中，声明说明了特定的对象(变量，过程等)具有特定的类型。例如，我们可以假定：

integer v_1 ;

用来声明变量 v_1 具有 integer(整数)类型。显示(explicit type declaration)类型声明有两个主要的作用。一是翻译程序可以根据声明中的信息来捕捉编程错误，并使程序更有效率地执行。二是声明使程序更容易被人理解。

在程序语言中，如果变量和过程的类型没有被声明，那么这种语言通常被说成是无类型的(untyped)。这绝对是误导！任何程序语言都包含一些类型概念，它们的翻译程序必须应付类型概念不同的计算机硬件。不需要类型声明的语言的不同之处在于，它们的翻译程序全权负责确保所有的数据都会根据它们的隐式类型定义得到正确的解释。对于“无类型”语言，一个常用的论据是类型声明过于费事。这个论据变得越来越不具有说服力了，因为程序变得越来越大，生命期越来越长，同时也越来越难以理解了。

在业界被普遍使用的程序语言都有类型声明。本章最后讨论的几种语言(Lisp, Smalltalk, Prolog)没有类型声明，或者没有可选的类型声明。

总结一下我们到目前为止所讲到的，计算机内存字表示数据对象。这些对象由它们的数据类型所表现，数据类型决定了计算机将如何解释数据对象。程序语言也提供了类型概念，这些概念有时与机器层面上的类型概念截然不同。关于程序语言中什么类型概念是合适的，存在一些争论，某种程度上，答案取决于该语言想要实现什么样的应用。关于是否需要类型声明，或者类型声明能否是可选的，同样存在争论。

现在，让我们更详细地考虑类型的一些通用属性。最简单的类型是原子的(atomic)，也就是说，无法分割为更简单的类型了。原子类型的例子包括整数，浮点数和字符。通常，原子类型由计算机系统的机器层面直接支持，某种意义

上，计算机具有能够操作整数、浮点数和字符的硬件。

类型构造实施(Type-building Operations)

程序语言通常提供构造比基本类型更复杂的类型的方法。在验证这些类型构造操作时，我们将看到一些很好的例子，这些例子中既有抽象的也有具体的类型构造方法。

数组(Arrays)

作为第一个例子，让我们考虑数组(array)，数组是项目的集合，数组中的项目可以通过下标(index)被选中。数组中的所有项目具有相同的类型。如果 A 是类型为 X 的数组，我们可以通过 A 的下标来选择一个 X。加有下标的操作通常被写为 $A[i]$，这里 i 就是下标索引。被索引的值也能够被赋值，因此语句：

$A[i] = 1\ 023.99$;

设置浮点数组 A 中的第 i 个元素为 1 023.99。数组在计算机底层通常被直接实现为内存块。下标提供了合适的方式来访问内存块中的不同部分。作为直接实现的一个结果，下标类型在大部分语言中是受限制的。在一些语言中，下标是从 0 到某个最大值的整数，这个最大值在数组初次被声明时就确定了。在另一些语言中，这样或那样的限制连同其实现上的额外的复杂性都被去掉了。

数组声明的形式在某种程度上取决于所给语言中什么样的限制是有效的。例如，在使用从 0 到某个固定大小的整数下标的语言中，只有数组大小和类型是必须的，因此像下面这样的表述：

array[*12*] of *pitch*;

声明了一个类型为 pitch，具有 12 个项目的数组，这个数组下标为整数，从 0 到 11。如果我们可以指定下限和上限，声明可以是这样：

array[*19*, *108*] of *MIDI-pitch*;

上面的代码根据 MIDI 协议规范声明了一个可以容纳 88 个平均律音高(equal-tempered pitches)的数组，数组索引从整数 19 到 108。(关于 MIDI 的更多信

息，请参考第 21 章。）

数组也可以是多维的(multidimensional)。这意味着确定数组中的一个元素需要使用两个或多个索引。二维数组可以表示矩阵(matrix)，矩阵在数学上是对矢量空间(vector space)的线性操作。由于矩阵和矢量空间在数学的科学应用中极为重要，因此数组在科学计算中也总是非常重要。

数组在计算机音乐程序中很普遍，因为一个一维数组可以表示一个音频波形或包络线的振幅，而一个二维数组可以表示用于作曲的马尔可夫链的概率(见第 19 章)或魔术方格(magic square)中的音阶(在其所有交叉线上显示一组十二音)。

记录(Records)

数组提供了一种组织相同类型数据对象的集合的方法。另一种类型构造操作是记录(records)，它允许程序员组织类型异质的数据对象。在记录中，每个不同种类的信息被称为记录的一个组件(component)。有时组件也被称为记录的槽(slots)或成员(members)。

记录非常有用，因为它允许程序员将本应在一起的东西放在一起，从而表现逻辑上的相关性。对记录的合理使用，可以使程序更容易被理解。例如，可以使用记录来收集一个音符的几个方面的信息：它的音高、音域、音长等。记录被声明为一个组件的名字和类型的列表：

```
record
pitch: character_string;
register: integer;
duration: integer;
...
end
```

这样的记录的集合就形成了数据库(database)。实际上，记录的概念是由 Cobol 程序语言最先引入的，这种语言是为数据库设计的，并与商业应用相关。但是就如我们将要看到的那样，记录广泛而有用得多。

指针及其不足(Pointers and Their Discontents)

数组和记录类型是组织数据的基本方式，因为它们将数据对象集合转变为

一个单一的，易于管理的对象。另一种基本的概念就不同了。指针(pointer)类型表示了指向数据对象的引用，这里引用被视为对象自身。因此，如果 *X* 是一个类型，指向 *X* 的指针类型就是引用 *X* 的数据对象的类型。指针都具有相应的解参(dereferencing)操作，对指向 *X* 的指针解参即得到 *X*，也就是指针所指向的对象。解参操作通常使用星号，因此 * *ptr* 解参指针 *ptr* 来获得它所指向的值。通过指向某个对象的指针，我们可以使用解参操作通过赋值改变该对象的值。此时指针很像数组元素，也有点像变量。

当我们说指针指向(refer)某个对象时，我们的意思很简单——比自然语言中的所指要简单得多。要确定像“可点击的东西”这样的英文表达的所指是很困难的。相比而言，指针的所指的概念很简单。指针被实现为一个内存字，它的内容被解释为另一个内存字的地址(也就是被指向的对象所在的地址)。因此，指针指向内存地址上的对象，此对象即指针的值。

虽然基本的概念很简单，接下来的衍生物却并不简单。系统地将一个计算机内存字解释为另一个内存字的地址的思想是一种基本的程序设计技巧，它对于使用计算机做任何有用或有趣的事情都是绝对必要的。理由有些微妙。计算机内存是按照严格的线性方法被寻址的。(也就是说，有一个内存字地址为0，接下来一个内存字地址便为1，以此类推。)大部分问题并不适用于这个严格的线性结构。指针技术(将内存字解释为其他内存字的地址的技术)能够使得线性寻址内存模拟任何想象得到的结构。在细节上学习如何做到这一点，是学习如何编程的重要一步。

为了了解将值解释为地址这个概念有多大说服力，让我们暂时回到数组来思考这个问题。假设有一个整数(内存字)数组 *A*，索引从 0 到 3 999。这就像一个有 4 000 个字的内存数组。由于它既包含整数，又以整数作为索引，就像指针对计算机底层内存那样，我们也可以在这个数组上使用同样的方法。像 *A*[*A*[*12*]]的表达式意味着将 *A*[*12*]中的整数解释为一个索引(地址)，并返回那个索引所对应的数组值。实际上，对于没有显示指针类型的语言而言，这个方法在数组上的应用很广泛。

Lisp 语言采用的方法就更有趣了，它提供了强有力的证据证明了前面的说法，即指针能将线性寻址数组映射到任何结构当中。(注意：下面的描述对于现代 Lisp 语言并不完全正确，但却准确地反映了它们的整体特点。)Lisp 中的内存结构由 cons 组成，cons 就是指向另外两个对象的小对象，这两个对象可以是其他的 cons 或者更基本的对象，比如数字(图 2.6)。有了 cons，我们可以建立列表和其他种种奇特但出色的结构。Lisp 中的 cons 是建立在底层计算机内存硬件的线性阵列之上的。

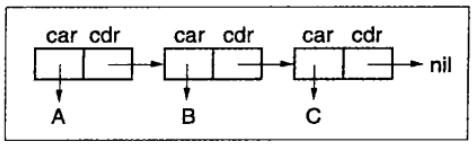


图 2.6　一个由 3 个 cons 来表示的 Lisp 列表(A B C)。每个 cons 有两个指针。Cons 的 car 指向值，而 cdr 指向列表中的下一个 cons 单元。最后一个 cdr 指向 nil，也就是说，cdr 包含零，这表明列表的结束。

指针非常强大有用，但是在使用时也会有一定的危险。让我们再次返回那个数组，去看看为什么指针会比较危险。先前我们写了 *A*[*A*[*12*]]来解释将 *A*[*12*]的值作为索引，但是如果 *A*[*12*]的值并不是一个正确的索引(比如，并不是一个在 0 到 3 999 之间的整数)呢？而且，我们现在正用至少两种方式来解释 *A* 中的元素，作为数字和作为索引。至少 *A*[*12*]是一个索引，并且被它索引的是一个数字。使用指针时，有许多途径会导致错误，造成误解。

正由于存在这样的危险，在程序语言中显而易见地提供指针一直存在争议。和其他问题一样，正确的答案取决于程序语言的目的。很多语言大胆地提供了指针，C 语言就是其中之一，但是这与它的设计目的——系统程序语言(systems programming language)是相符的，即始终坚持靠近计算机底层。Lisp 语言根本不设指针——尽管指针在其语言的执行中普遍存在，部分原因在于它的设计目标与 C 语言大相径庭。

为了使讨论更具体，我们来看一个使用指针的例子。一个经典的例子是建立链表(linked lists)。链表是具有相同类型的记录的集合，在链表中，每个记录的每个组件都是指向下一个链表元素的指针。图 2.6 以 Lisp 中的隐式表示展示了一个简单的链表。下面的代码是一个显式程序设计示例：

```
linked_list_node record
pitch: string;
next_node: pointer to linked_list_node;
end
```

这个记录有两部分：一部分叫做音高(pitch)，它包含一个被引用的字符串；另一部分是一个叫做 next_node 的指针，指向另一个 linked_list_node 记录。链表经常用一个包含简单方框和箭头的图示来表示，见图 2.7。

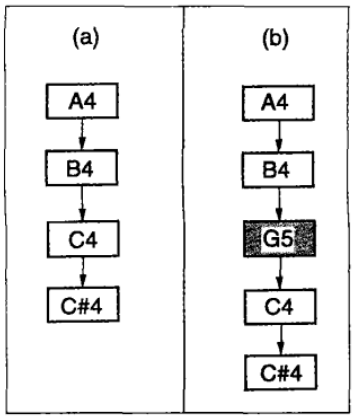


图 2.7 一段旋律的链表数据结构，包括被引用的字符串所表示的音阶名。最后一个指针值为 nil，在本图中略去。(a)原旋律；(b)修改"B4"的指针，并添加新元素"G5"之后的新的旋律。

一个更加复杂的例子是有两个指针的链表记录。这就形成了一个被称为二叉树(binary tree)的通用数据结构。在一个二叉树中(图 2.8)，每个节点可以有两个分支：

linked_list_node record
starting_time: integer;
left_node: pointer to linked_list_node;
right_node: pointer to linked_list_node;
end

在编辑多声部乐曲的演奏进程时，二叉树是一个很好的表示方法。通过二叉树组织的信息也能被有效地搜索，因此这个数据结构在计算机科学中是很常见的(Knuth 1973b)。

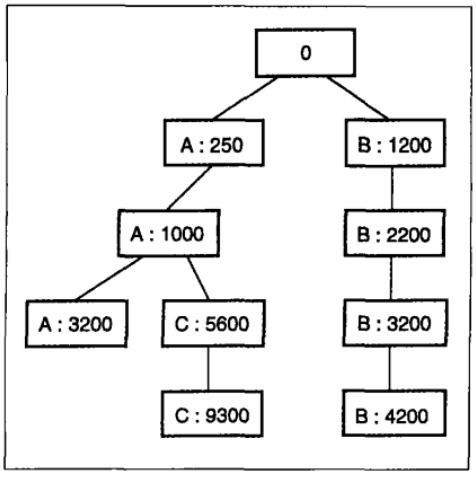


图 2.8 一个用二叉树表示节点的链表，每个节点都可以有两个子节点。每个节点的特性代表三种乐声的演奏进度。每个节点中的数字表示以毫秒计的每个事件的起始时间。

这些链表的例子带来了几个问题，其中之一是链表被使用的方式。总的来说，程序遍历(traversed)链表，并跟随子节点中组件的指针，直至到达某个特定值，该值表明已到达链表的结尾。链表之所以有用，原因之一是它能够在表中的任何位置，通过操作指针组件的值来方便地添加或删除元素。这一点与数组形成了鲜明的对比，在数组中部插入一个值，意味着新元素右边的每一个值都需要向右移动，以便为新元素腾出位置。增大或减小数组很费时，然而在链表中添加或删除元素就非常有效率。

许多程序设计环境都提供标准的过程来操作链表数据类型，例如 insert_node 和 delete_node。这一将过程与数据类型联系起来的思想在程序设计语言的研究中已变得非常重要，而且它被两个不同的研究机构以不同的方式分别采纳和运用于抽象数据类型(abstract data types)及面向对象程序设计(object-oriented programming)。关于这两者，我们稍后会介绍更多。

下面是另一个由我们的链表示例引出的问题。设想已经定义了一个二叉树链表，我们现在需要一个新的树，它的 starting_time 不是整数，而是浮点数。在大部分程序语言中，只能通过定义一个新的数据类型并给它一个不同名称来完成。这个方法不是很好，至少由于两个原因。对于阅读程序的人来说，这样做易于混淆这些记录类型(也就是链表)的常规应用模式。从技术的角度来看，与这些数据类型相关的许多过程(比如添加和删除元素，通过遍历列表计算元素的个数)对于链表数据类型而言都是完全相同的。

解决这个麻烦的一个办法是使用多态(polymorphic)数据类型。大致上，多态类型有一个关于自己的类型的参数。列表是一个很好的例子：我们可以有整数列表，浮点数列表以及无数其他类型列表。

稍抽象的数据类型(Somewhat Abstract Data Types)

早期的程序语言设计工作主要集中在控制方面，比如，提供什么样的选择和循环结构。从那时起，人们就清楚地意识到，涉及数据的问题非常困难，但也非常重要。这是由于程序中的数据结构是对(现实的或想象的)部分世界的模仿。从数据结构的重要性看起来，人们就不会惊讶于有许多不同的研究陷入困境是因数据结构方面的原因。

一个基本的见解是，数据类型是由数据结构(记录、数组、指针等)和它所允许的操作(operation)共同定义的。通常只有被授权的操作可以访问数据，其他的必须通过额外的操作获得授权。我们说操作封装(encapsulate)了数据，这是一个很有益处的重要主题。

至少有两组独特的研究者已经开发出了自己的语言，它们带有针对封装数

据的各种过程所设计的数据类型。其中一组以别的传统语言为基础，并加入了所谓抽象数据类型(abstract data type)的结构。这项工作的实例有 CLU (Liskov et al. 1979)和 Alphard (Wulf, London, and Shaw 1976)。其中最显著的成就(也许是巅峰)是由美国政府(国防部 1980)资助的 Ada 语言。

面向对象的程序设计(Object-oriented Programming)

另外一组非常不同，他们提出要更加强调数据，并开发出一种称为面向对象程序设计的方法。他们还研究出专门讨论这种程序设计法的一些新法子以强调其激进性。具体来说，数据元素被称为对象(objects)。数据类型被称为类(classes)。(至少大部分数据类型被称为类，尽管在面向对象语言中也存在更传统的数据类型。)与类相关的操作通常被称为方法(methods)。调用一个过程被称为发送一条消息(sending a message)。这种不同不仅仅是表面的。在传统程序设计中，一个过程的所有变量都是平等的，而在面向对象程序设计中存在一个有特权的变量，它就是消息的接收者(receiver)(消息被发送到的对象)。

Simula 语言(Dahl and Nygaard 1966; Dahl, Dijkstra, and Hoare 1972)是面向对象语言中的先行者，它被设计用来进行离散事件仿真程序设计，在这种仿真中，时间作用于一系列模拟真实世界的对象上。尽管 Simula 是第一个面向对象语言，Smalltalk (Goldberg and Robson 1983a,b)却可能是最有影响的。在施乐帕洛阿尔托研究中心(PARC)和 ParcPlace Systems 公司经过长期开发之后，它具备了一些与面向对象程序设计内涵相同的本质属性。Smalltalk 是动态类型的(dynamically typed)，某种意义上，它的变量可以是任何类的对象，因此合适的消息在被发送之前必须被检查。换句话说，Smalltalk 是那种有时会被误认为是"无类型"的语言之一。同时，目前版本的 Smalltalk 和 Smalltalk-80 包含一个超级开发环境，对开发图形用户界面的音乐程序提供了出色的支持(Krasner 1980; Pope 1991a,b; Scaletti 1989a,b; Scaletti and Hebel 1991)。

继承性(Inheritance)

面向对象程序设计对于数据构造的一个重要贡献是类继承(class inheritance)思想。该思想以已有的类型(类)为基础，而不是完全从头开始定义新类型。当一个类需要特化(specializes)为另一个类时，继承尤其合适。

举例来说，假设在一个演奏程序中有一个用来管理音乐键盘的类，我们称它为 Keyboard 类。它将键的数字转换为音高，以及当键被按下时开始一个音符的操作(方法)。现在，假设我们需要挂接一个速敏式键盘(Velocity-Sensitive

Keyboard)(如果我们越用力也越快速地击键,该键盘奏出的声音也就越大)。由于需要传送速率数据,我们需要重写"音符开始"操作,而键到音节的转换操作则不用改变。我们可以使用继承创造一个 Velocity-Sensitive-Keyboard 类,该类继承了 Keyboard 类的所有数据成员,也许还要添加一个新的处理速率信息的数据成员。这个类在继承了转换操作的同时将重新定义"音符开始"操作。

在面向对象程序设计中一个不太常用的技术是多重继承(multiple inheritance),即一个类可以从数个其他类继承行为(和数据成员)。这对于"混入"功能非常有用,所谓"混入"功能就是为一个已存在的类添加新的特性。例如,我们可以写一个音符记录 mixin,它可以被添加到任何有音符实例化操作的类中,它可以自动重定义操作以便它每次被调用时,其参数能够被保存到一个辅助记录数组中。

当程序变得越来越庞大,越来越复杂的时候,能够便利地对其进行组织、日渐增多的修改以及其他操作,就变得越来越重要,这种便利正是面向对象程序设计的要义所在。因此,随着问题的规模和复杂性的增加,面向对象程序设计在计算的世界里已经变得越来越受重视,也越来越有影响了。早期的语言(尤其是 Smalltalk)有些受执行速度缓慢和难于学习的阻碍,鉴于此,一些语言已开发出来,它们在如 C 和 Pascal 这些主流语言的基础上加入了面向对象的功能。这种趋势的一个例子是 C++语言(Soustrop 1997)。

高度抽象的数据类型(Highly Abstract Data Types)

到目前为止所介绍的抽象数据类型的方法都不是高度抽象的。例如,如果定义了一个数据集类,就必须精确地定义它的一个实现(例如,一系列的操作),所有的数据集都使用这个实现。但是,实现数据集有很多好方法,每种方法都适用于具有不同的规模和作用域的数据集。目前,没有一种语言完全地解决了这个问题。然而,另一项研究却陷入了困境,在该研究中,数据类型以一种非常抽象的形式实现,这被认为是为底层的自动系统决定如何最好地实现数据类型留下了可能。在这些方法中,只需要支持数据类型的操作的属性必须满足什么样的条件,而不是精确地说明如何执行它们。在大多数情况下,可用公式来说明。在这一研究领域内经常出现的一个典型例子是对栈(stack)的描述。栈是一个含有 push, pop 和 top 单个操作的数据结构,其中 push 操作将对象存储在栈上;pop 移除最近存储的值;top 返回最近存储的值。

这些简单的原则可用以下公式来解释,假设 S 是一个栈,x 是一个值。如果有一些背景假设,那么两个公式就可以定义栈的行为。

$$\text{pop}(\text{push}(S, x)) = S$$
$$\text{top}(\text{push}(S, x)) = x$$

第一个公式是说 pop 可以撤销 push，第二公式是说 push 必须记录被推入的值以便 top 能够读取这个值。

对抽象数据类型的抽象，其背后的核心思想是，将公式化定义解释为能够有效执行所定义操作的指令。这样，虽然不使用计算机语言，也不使用任何明显的指令，公式操作起来却非常有效。事实上，我们把这称为对公式的运算解释(operational interpretation)。从程序设计的角度来看，这一点之所以如此重要是因为我们以一种非常抽象的方式，即公式，来描述栈的行为，而不是通过给出实现它的行为的具体指令来描述它。这是一个被称为说明性程序设计(declarative programming)的例子，我们将在本章稍后讨论。

公式的运算解释背后是大量的数学理论支持。通常第一步是给出公式的择优取向(preferred orientation)，以便它们能够成为涉及所给术语(在我们的例子中是 push，pop 和 top)的表达式的重写规则(rewrite rules)，或者换句话说，简化规则(simplification rules)。注意，在我们的例子中，两个公式右边的部分要比左边的部分简单得多，因此，择优取向是从左到右的。通常方向可以被指定，有时甚至能够被自动完成。然而，有时根本就不能指定方向，更别说自动完成了。例如，一个通用代数规则是交换性的，对于加法操作符而言，它可以被写成下面这样的公式：

$$x + y = y + x$$

直观的解释是，参数的顺序对于加法操作符并不重要。这个公式就不能被给定择优取向，直观的原因是公式的两边都并不比另一边简单。

对于非常抽象的规范的自动解释是一个很受期望，却也充满困难的目标。包括戈德尔(Godel)、图林(Turing)和其他数学先驱的很多理论研究结果都表明，严格来讲，很多问题都不能解决。其中之一就是确定能否将择优取向自动指定给公式的问题。

除此之外，公式级的抽象并不能解决本节开头时所提出的实践问题，即对于数据集这样的抽象类型，不同的实现适用于不同的环境。对这个问题，需要一个更加实际的方法。到目前为止，还没有人能够令人信服地完成这项工作。

程序设计语言的课题(Programming Language Themes)

到现在为止,我们已经探究了程序设计中的控制和数据两个问题。现在简单总结一下程序设计中最重要的三个课题:函数程序设计、逻辑程序设计和约束程序设计。

函数型编程(Functional Programming)

函数型编程(functional programming)认为数学函数(mathematical function)是我们用来求解问题的最重要的工具。一个数学函数定义一系列输入(也称参数 arguments)和一个输出(也称作结果 result)之间的关系。也就是说,对于每一组特定的参数,总会有一个不变的结果。这是算术数字运算中应该具有的一个观点,它也能很自然地扩展运用到很多其他地方,的确是一种很有用的思维方式。在本章开始处,我们曾说起过程是程序中最重要的组成部分,而且与数学函数的功能最为相似。

函数型编程的与众不同之处在于其认为过程丝毫不依赖于上下文,而且同样的输入总是应得到同样的结果。这对于要从外界接受输入(比如所有的交互式计算机音乐程序)或者以某种方式对外界产生影响(不对外界产生影响的程序是没用的)的操作造成困难。当然,函数型编程的鼓吹者对于这一缺陷和其他的问题有他们的答案。

相比深入探讨关于函数式方法的普遍性的争论而言,我们更希望指出一些在此方法支持下已完成的无疑是有价值的工作。函数型编程兴起于 Church 的数学工作,为了研究函数,Church 发明了 λ 演算(lambda calculus, Church 1941)。λ 演算产生了 Lisp 程序语言,该语言最初就是函数式语言。

现在很重要的基本计算知识来自于对并行计算机和 λ 演算的研究成果。比如,这项研究阐明了递归过程(recursive procedure)的概念,并通过展示该过程在非常多的情况下都极端有用来影响程序设计实践。〔一个递归过程在操作上可以被视为是对自身的调用,或者更抽象一些,作为自身的一部分被调用。它也可以是相互递归(mutual recursion),即一组过程以相互引用的方式来定义。〕该研究的另一个有价值的方面是确定了评估函数及其参数的排序策略,演示了很多情况下最明显和最易实现的策略可以带来无尽的计算,而更复杂的策略却并不能。

历史上,函数型编程很早即以 Lisp 的形式被实现。当 Lisp 被开发出来后,

它变得更加注重实效而丧失了它函数型的纯洁程度。最被广泛使用的现代Lisp形式是Common Lisp，一个同时具有数个Lisp变体的大型编程环境(Steele 1984)。Scheme，一个自身也产生了其他变体的Lisp变体，试图恢复紧凑和纯洁的特点(Abelson and Sussman 1985)。

有一段时间函数型编程被认为是效率低下因而无法实际应用的，但是它已经重新崭露头角，一个重要的原因是它在并行计算机上的潜在应用。具体原因有以下几点：首先，如果函数是真正"纯的"(对于相同的参数总是返回相同的结果)，那么一个函数的参数能够被并行计算，这些参数的参数也如此，等等。其次，函数型编程似乎非常适用于函数型操作符作用于大型统一的数据集(例如矢量、矩阵等)的问题，这给所谓"数据并行操作"(Hillis 1987)带来了许多机会。

逻辑编程(Logic Programming)

λ演算是本世纪一项重要的数学成就，相较而言，谓词演算(predicate calculus)尽管始于上世纪，却更加重要。一个谓词可以被理解为一个返回真或假的函数，因而谓词演算是数学逻辑的基础。

就像函数型编程产生于λ演算，逻辑编程(logic programming)产生于谓词演算。谓词演算可以被视为一种用来讨论数学定理的语言，它的一个可用的阐释可以是一段定理的证明。事实上，这是对逻辑编程的正确的抽象。

前面我们描述了高度抽象数据类型的公式化方法，并指出在这一领域的很多问题严格来讲都无法解决。对谓词演算而言也是这样。因此，逻辑编程就是要找到谓词演算的严格形式或严格的定理证明策略，使得计算易于进行，同时保持描述事物时的优点，即类似于数学定理的抽象形式。

到目前为止最流行的逻辑程序语言是Prolog语言(Clocksin and Mellish 1987)，它流行的原因是一系列使Prolog程序能够非常有效率地执行的聪明的思想。一个Prolog程序可以被视为是一系列被称为霍恩子句(Horn clauses)的严格的谓词演算形式的声明。通过描述谓词间的关系，这种声明有效地定义了谓词。这一思想很像我们前面看到的公式化定义，除了左右两边并非对等关系，而是推论(implication)关系，也就是说，如果右边为真，那么左边也为真。这种方法与函数型编程的一个重要不同在于，谓词可以像表现函数那样容易地表现任何关系(relations)。关系比函数的限制要少，这使得关系在某些情况下成为更加灵活的代表性手段。

Prolog程序试图解决涉及由程序定义的谓词的问题，该程序的部分参数是变量(variables)。程序的目的就是为这些变量找到一个或多个解决方案。它按照预定的方式，一个接一个地搜索声明(子句clauses)。当子句的左边与问

题匹配时，Prolog 解释器试图以与求解原始问题相同的方式求解右边的部分。如果到达绝境(没有匹配的子句)，它就回溯(backtracks)，返回到最后一个匹配的字句，放弃在这条子句上的匹配，并尝试寻找另外的匹配规则。

Prolog 内建的回溯性能非常有力，并且很适合于涉及搜索的问题，但是也有些不可预知和难以控制。在实践中，很多谓词并不要求回溯，而且引入了一些不同的机制来对它进行限制。

Lisp 与 Prolog 举例(An Example in Lisp and Prolog)

为了对 Lisp 和 Prolog 有一个更加具体的印象，我们用这两种语言介绍一个简单的函数的定义。这个函数连接两个列表，即 *append*。在 Lisp 中，append 以两个列表为参数，并产生连接后的列表作为结果。在 Prolog 中，所有的一切都是谓词，并且所有的注意力都放在参数上，因此 append 有三个参数，第三个就是产生的结果。图 2.9a 是 Lisp 的定义，图 2.9b 是 Prolog 的定义。

(a)

```
(defun append (list1 list2)
      (cond ((null list1) list2)
            ((null list2) list1)
            (t (cons (car list1) (append (cdr list1) list2)))))
```

(b)

```
append([], X, X).
append(X, [], X).
append([X1 | X2], Y, [X1 | Z]) :- append(X2, Y, Z).
```

图 2.9　append 操作的定义。(a)Lisp 定义；(b)Prolog 定义。

让我们先来解释这些定义。在 Lisp 和 Prolog 中，列表都是基本数据类型。在 Lisp 中，表达式被放在括号中。第一行说明函数的定义(*defun*)，并给出它的名字和参数。定义仅包含一个表达式，其第一个字是 cond。这是一个条件表达式，它按顺序测试每个字表达式的第一部分，如果测试为真，就执行并返回第一个表达式的剩余部分。这样，这个表达式的本质就是：如果 *list1* 是 null(空)，返回 *list2*；如果 *list2* 是 null，返回 *list1*；否则("*t*"表示"true")，返回：

(cons (car *list1*) (append (cdr *list1*) *list2*))

这个复杂的表达式将 list1 的第一个元素与将 list1 的剩余部分附加到 list2 后的结果连在一起。换句话说，这是一个递归定义。

在 Prolog 中，谓词参数位于函数名之后的括号中，这是一种更加传统的数学定义。以大写字母开头的名字是变量。通常子句有很多变量，append 也不例外。图 2.9b 的第一条规则说明，如果第一个参数是空列表（写作“[]”），那么第二和第三个参数相等。（记住第三个参数是“结果”。）第二条规则对第一和第三个参数作了同样的说明。第三条规则只在所有的列表都不为空时起作用（因为前两条规则将先被 Prolog 解释器检查），它以一种稍微不同的方式说明了本质上与 Lisp 程序中那个复杂的表达式相同的内容。这条规则的右边部分等价于 Lisp 对 append 的递归调用。

三行的程序并不能告诉你太多关于程序语言的内容，这个简单的例子只是为了让读者略知一二。现代 Lisp 和 Prolog 编译器可快速执行此类程序。

约束编程（Constraint Programming）

约束编程（constraint programming）与逻辑编程相关，尽管时间使这种关系变得并不明显。然而，逻辑编程研究一直由精通数学逻辑的人们所掌控，而约束编程更依赖于直觉，不那么严格精确。这一领域内的大多数工作集中在以下几个方面：几何约束问题的解决（例如，保持线的平行或垂直，等等），图形系统（Sutherland 1963；Borning 1979），还有电路中的欧姆定律（Sussman and Steele 1981）。近些年，经过数学培训的研究者们已经开始认真研究逻辑编程和约束之间的关系（Jaffar and Lassez 1987；Saraswat 1992）。

约束编程的最吸引人之处就在于它允许程序设计者陈述比 Prolog 的霍恩子句，或者其他类似的逻辑编程方式更复杂的规则，并且用功能更强大、更专业的解释器来解释这些规则。最终，约束编程能让程序设计者灵活访问许多各种各样的专业数学算法，选择和使用这些算法目前要求精通专业知识的人员。

在另一个稍有不同的研究方向上，约束对于阐明需要渐进解决方案的问题，有潜在的帮助作用：一个给定的系统状态被表示为约束系统的解决方案，任何干扰都会导致一个新的解决方案的产生，且通常会对前一方案的组件加以相对少量的改变。最近几年，随着交互式和图形化应用的发展，渐进算法已经变得越来越重要。

结论(Conclusion)

我们已经讲了什么是程序设计，以及与程序设计相关的问题和各种解决方式。在最开始，我们曾说程序设计从根本上说是一个问题求解的过程，并讨论了为何正确的基本原则有时难以实行。许多人喜欢问题求解的过程本身，但程序设计如此有趣的原因之一是计算机应用，而不是程序设计本身。计算机根据用数学语言表达的简单规则来工作。很多程序设计方法和语言都来自或受到数学的启发。这一点在本章中有充分的讨论。在另一方面，计算机是具体的实物，可用各种有趣的方式与现实世界相连。当计算机控制一个合成器时，输出的就是声音，而声音比任何数学表达的性质都要丰富(至少对于耳朵而言)。计算机的可编程性将抽象的数学形式与实际的应用和效果联系起来。让一个原型音乐系统运作所带来的乐趣就是音乐工程的乐趣。程序设计不同于更早出现的工程类型，它的工作对象是无形的。但是，计算机程序设计仍是工程的一个分支，因为它能产生有形的结果。

另外，让人感到兴奋的一点就是计算的跨学科特点，这对于那些对计算机音乐抱有兴趣的人们就无须赘言了。计算机科学吸取了数学和工程领域的许多思想，也回馈了大量有益之物。随着计算机在我们这个社会的深入使用，计算机知识和其他领域的知识的结合也变得更加普遍。

尽管计算机的普及应用不过区区数年，但已出现数以百计的程序设计语言及其分支，将来还会出现更多。有了这么多种语言，对这个领域不了解的人们也许会认为该研究的已经被挖掘得很彻底了。其实远非如此。虽然有些语言如 Cobol 和 Fortran 已确立了牢固地位，但计算机程序设计正不断地快速向前发展。因此，本章中提到的许多观点和技术在 10 年或 20 年后还有用，而有些也许就将过时。几乎可以肯定，吸引我们大部分注意力的传统程序设计方法将会变得不那么重要。随着事物发展，本书提到的一些仍会有效和有用的技术会以不同的角度继续呈现，以不同的方式继续应用。

到目前为止，基于现有的计算机程序设计经验，最重要的一个经验可能就是：程序设计是一个组织和控制复杂性的问题，这一复杂性比我们起先所了解的程度更深。结构化程序设计运动在过去 20 年中扮演了重要的角色，它可被看作是对这一复杂性的反应。结构化程序设计所产生的语言支配着程序设计实践，也构成本章的基础。面向对象程序设计有潜力进一步发挥这些特点，因为它的很多重要贡献都与控制复杂性有关。约束编程，或许还有逻辑编程的某些形式，也将扮演更加重要的角色。

PDG

第二部分　声音合成
(Sound Synthesis)

PDG

第二部分概述(Overview to Part Ⅱ)

1906 年 9 月 26 日,位于纽约百老汇第 39 街的电传乐大厅(Telharmonic Hall)敞开了大门。约有九百位听众前来聆听了一场新乐器的演奏会,这个新乐器是由萨迪尤斯·卡希尔(Thaddeus Cahill)制作的庞大的电传簧风琴(Telharmonium),这是历史上第一个、同时也是最大的声音合成器(Cahill 1897, Rhea 1984, Weidenaar 1991)。它虽以电力驱动,但信号并未经过电子放大,它平稳旋转着的音调发生器,播放出比自然更纯净的合成音——以精确整数比的正弦波构成的音调。年迈的美国作家马克·吐温(Mark Twain, 1835—1910)被这次展演深深地打动,他写道:"每当我听到或看到一个像这样令人惊奇的事物,我就只好推迟我的死期,在一次又一次聆听它之前,我恐怕不能离开这个世界!"(Rhea 1972)

大约在同一时间,意大利神秘主义者路易吉·鲁索洛(Luigi Russolo)的想象力则被另一个声音世界所吸引——浑浊的工业噪音以及战争的破坏性声响,如同在其著作《噪音的艺术》(*The Art of Noises*, Russolo,1916)中所激动宣称的一样。鲁索洛(Russolo)制作了一组声学噪音乐器,并在 20 世纪 20 年代的一系列公开演奏会上用它们来表演。这些戏剧性的序幕,为更加系统化地探索电子声音合成——一个深刻地影响了 20 世纪音乐理论与实践的发展进程——奠定了基础。

在数字技术之前,电子的声音合成方式分为两大类:(1)真空电子管、离子管振荡器或晶体管振荡电路。(2)使用机械、静电、光电方式驱动的旋转或振动系统。

1940 年间发明了可储存程序的电子数字计算机,开启了声音合成的新纪元。自马克斯·V. 马修斯(Max V. Mathews)于 1957 年所做的第一个实验开始,数十种声音合成技术陆续被发明出来。现代声音合成则是这些技术的总和。与计算机图形学领域一样的是,任何时候我们都难以断言哪些技术会蓬勃

发展，而哪些技术行将消逝。这种状况由于音乐产业的竞争压力而更加激烈，合成方法不可避免地来去更替，时冷时热，有时还出现反复。声音合成技术的种类必定会不断地增加，因为，没有哪一种方法可以满足所有音乐家的需求。人的品位和取向各异，对新的音乐感受与体验也会继续追寻不止。

第二部分的构成(Organization of Part Ⅱ)

第二部分讲解当代合成方法的基本原理。这些课程依教学需求编排。我们以直观的方式介绍这些合成技术是如何起作用的，并略过特定厂商的实现细节，以免陈述混乱。在不断变化的技术环境中，这似乎是最谨慎的课程安排。在教学环境上，本书还应辅之以实习作业并利用一些现有的工具，包括合成器、交互式声音合成程序或声音合成语言等。

这里的材料建立在对基本术语理解的基础上，如频率、振幅、频谱等，并假设读者已具有如第 1 章所介绍的数字音频基础的知识背景。

我们将不同的技术组合为 11 个类型，放在 5 个等长的篇章里来介绍。有些章内的组合是任意安排的，如第 5 章所介绍的三种技术并不紧密相关。然而，篇章的顺序则不乏刻意的安排。第 3 章是阅读其他部分之前的必要课程，讨论大致由最基本的合成方式开始，一直到较为奇特少见的合成方式。注意在第 20 章中讲述了两个附加的合成方法：射频解调(radio wave demodulation)及脉冲音调合成(pulse tone synthesis)，之所以介绍它们主要在于其历史趣味和新鲜感。

对于电子音乐或计算机音乐的作曲家而言，一个陷阱就是太过依赖单一的某种合成方法。采用任何一种技术的同时而排斥其他的技术，会导致过度使用而陈腐乏味，除非是以不同寻常的艺术家态度为之。合成技术本身不能解决计算机音乐作曲与编配中的所有问题。最令人期待的合成技术延伸之一，就在于不同技术间的对比——以混频及信号处理来造成复合声音对象(compound sound objects)(Roads 1985f)。本书第三部分将探索这些主题。

第3章 数字声音合成引论
(Introduction to Digital Sound Synthesis)

柯蒂斯·罗兹(Curtis Roads)、约翰·斯特朗(John Strawn)

音乐声的表述(Specifying Musical Sounds)

声音对象(Sound Objects)

加法合成表述问题举例

(Example of the Specification Problem for Additive Synthesis)

音乐家界面(The Musician's Interface)

音乐化输入装置(Musical Input Devices)

演奏类软件(Performance Software)

编辑器(Editors)

语言(Languages)

算法作曲程序(Algorithmic Composition Programs)

声音分析(Sound Analysis)

结论(Conclusion)

本章概略讲述数字声音制作的基本方法。在简要的历史回顾后，我们将介绍查表合成的理论——这也是大多数合成算法的核心。接下来介绍各时期有所不同的声音合成策略。之后是“软件合成”与“硬件合成”间的实际比较，也就是在计算机程序与专用合成器间的比较。最后，我们对计算机或合成器表述乐声的各种方法作一概览。阅读本章的必要前提是第一章所介绍的数字音频基本概念的相关知识。

背景：数字化声音合成的历史 (Background: History of Digital Sound Synthesis)

最早的计算机声音合成实验始于1957年，出自新泽西州墨雷山贝尔电话实验室的研究者们(David, Mathews, and McDonald 1958; Roads 1980; Wood 1991)。马克斯·马修斯(Max V. Mathews)(图3.1)与其同事们在其最早的实验中证明了计算机可以依任何音高或波形——包括时变频率与振幅包络来合成声音。

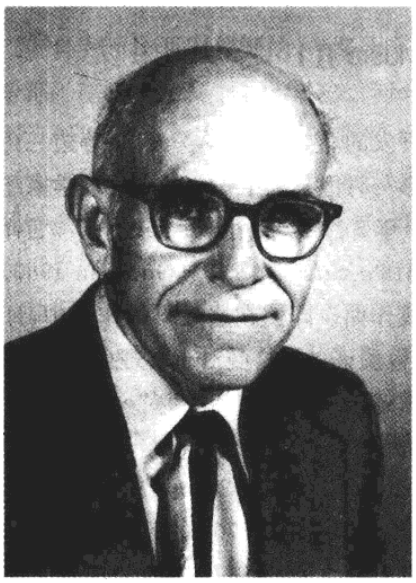


图3.1　马克斯·马修斯(Max V. Mathews)，1981年(照片由AT&T贝尔实验室提供)。

他们的第一个程序是在电子管电路大型计算机IBM 704上直接用机器指令撰写的(图3.2)。在当时，704是相当强大的机器，具有36比特字长，而且内置浮点部件以供高速数字运算。它可以加载多达32K字到其磁芯存储器内。当时计算机是如此稀有，贝尔实验室竟没有一台适合的机器，以致合成运算必

须在纽约市的 IBM 总部完成。每次前往曼哈顿计算一个声音后，马修斯与同事总是将数字磁盘带回贝尔实验室，在那里，用一台较慢的计算机，连同一个 12 比特电子管的“数字-声音转换器”(digital-to-sound converter)把磁带上的样品转换成可听的形式。这台由伯纳德·戈登(Bernard Gordon)设计的转换器，是当时世界上唯一一台能够制造声音的机器。

图 3.2 IBM 704 型计算机，1957 年(照片由 IBM 提供)。

音乐Ⅰ与音乐Ⅱ(Music Ⅰ and Music Ⅱ)

Music Ⅰ程序是由马修斯开发的，它可产生单一波形：等边三角波。使用者必须很有耐心地按照音高、波形与时值来确定一个音。心理学家纽曼·格特曼(Newman Guttman)曾在 1957 年 5 月 17 日用 Music Ⅰ作了一首单声部的练习曲，起名为《在塞尔佛音阶中》(*In a Silver Scale*)(Guttman 1980)。这是最早经由数字-模拟转换过程的合成作曲。即使在这最早的作品中，计算机能精确产生任何频率的这一潜力也得以验证。格特曼对于心理声学很感兴趣，他以此曲作为塞尔佛(Silver 1957)描述的“等拍半音阶”(“equal-beating chromatic scale”)与音准之间的一个对比测试。

马修斯在 1958 年完成了 Music Ⅱ(“音乐 2”)软件，它是由 IBM 7094 型计算机的汇编语言编写而成。此计算机是 IBM 704 的后继改进机型，由晶体管构成。7094 比此前的电子管机器速度快了数倍，所以能实现更为艰巨的合成算法。它可提供四个独立声部，内存中有 16 种波形可供选择。Music Ⅱ由贝尔实验室内的一些研究者使用，其中包括马修斯、约翰·皮尔斯(John Pierce)以及格特曼。

1958 年，在纽约举行了一场名为“计算机音乐”的新音乐会，会后是由约翰·凯奇(John Cage)主持的讨论会。同年晚些时候，格特曼在赫尔曼·谢尔

欣(Hermann Scherchen)位于瑞士格拉夫萨诺(Gravesano)的别墅演出他自己的计算机合成的作品《音高变奏曲》(*Pitch Variations*),作曲家伊恩内斯·克赛纳基斯(Iannis Xenakis)也在场观赏(Guttman 1980)。

单元发生器的概念(The Unit Generator Concept)

在数字声音合成语言的设计中,最具意义的成果之一就是单元发生器(unit generator,UGs)这个概念。单元发生器就是像振荡器、滤波器、放大器这样一些信号处理模块,它们可以彼此连接起来,构造出能产生不同声音信号的各种合成乐器(instruments)或音色排秩(patches)(本章后半部分将详细讲述单元发生器)。第一个使用单元发生器概念的合成语言是 Music Ⅲ,是由马修斯与其同事琼·米勒(Joan Miller)于 1960 年开发的。Muisc Ⅲ让使用者由单元发生器出发来设计自己的合成网络。在声音信号通过一连串这种单元发生器之后,大量不同类型的合成算法就可以相对容易地实现。

Music N 语言 (Music N Languages)

自 Music Ⅲ时代开始,各路研究者开发了一系列的软件声音合成系统——全都是基于单元发生器的概念。Music Ⅳ是用贝尔实验室开发的宏汇编语言 BEFAP(Tenney 1963, 1969)编写,由 Music Ⅲ重新编码而成。1968 年开发的 Music Ⅴ,则是马修斯在软件合成上的集大成之作(Mathews 1969)。Music Ⅴ几乎全部用 Fortran Ⅳ(一种标准计算机语言)写成,在 20 世纪 70 年代卖给了全世界数十所大学以及实验室。对于许多音乐家,包含本书作者,这个软件成为他们进入数字声音合成艺术之门的引路人。

以 Music Ⅳ与 Music Ⅴ为模型,其他开发者也开发了许多合成程序,如 Music 4BF、Music 360、Music 7、Music 11、Csound、MUS10、Cmusic、Common Lisp Music 等。作为一个大类,这些程序可以概称为"Music N"语言(见第 17 章)。

固定波形查表合成(Fixed-waveform Table-lookup Synthesis)

如第 1 章所解释,数字声音合成产生一连串的数字流,来表示声音波形的每个取样点。只要将取样值送入数字—模拟转换器(DAC),从而把这些数字转换成持续变化的电压值,再经放大后馈送到扬声器,我们就能听到这些合成的声音。

对此过程我们可以这样观察：设想一个计算机程序，它根据数学公式计算出波形取样点的值，并将取样值一个接着一个送入数模转换器，这一过程虽然可以正常运行，但非最有效率的数字合成基础。

一般来说，乐音的声波具有极高的重复性，这一事实表现为各个频率与音高的概念。所以，另一种更有效率的技术，是令硬件计算出一个周期的波形数据后，以数据清单的形式记录在存储器中，如图 3.3 所示。此清单称为波表(wavetable)。要生成周期性的声音，计算机只要来回反复读取波表，同时将这些读入的瞬时值送至数模转换器，即可转换成声音。

这一反复扫描存储器波表的过程叫做查表合成(table-lookup synthesis)。由于计算机从内存读取一个值通常只需花费几个毫微秒，所以查表合成比从最开始计算每个样本值要快得多。对于数字振荡器——合成器中基本的声音发生器来说，查表合成就是其核心操作。

现在我们把整个表的谷值检视一遍。假设第一个取样是由波表内的第一个数值给定(图 3.3 的位置 1)。要想由此简单合成器生成每个新样本，只需读取波表下一个取样值。到了波表末端时，只要返回到起点重新开始读取样本即可。这个过程又称为固定波形合成(fixed-waveform synthesis)，因为在整个声音事件过程中它的波形始终不变。

比如说，假设我们的波表内有 1 000 个取样点(entries)，每个录入都是 16 比特数值，这些录入编成 0—999 指数。我们称波表内的当前位置叫做相位指数值(phase_index)，相当于波形的相位。要读取波表时，振荡器由表中第一个录入(phase_index = 0) 开始，以递增(increment)的方式渐增，直到表末(phase_index = 999)。在这个点上，相位指数会末端“折回”到波表的前端，重新开始。

改变频率(Changing the Frequency)

使用波表产生的声音频率是多少呢？这取决于波表的长度和取样频率。如果取样率是每秒 1 000 个取样点，而在表内有 1 000 个数值，那么得到的结果是 1 000/1 000 = 1Hz。如果取样率是 100 000Hz，而表内有 1 000 个录入点，那么输出的频率就是 100Hz，因为 100 000/1 000=100。

那么，要如何改变输出信号的频率呢？如我们所见，最简单的方法是改变取样频率，但这个方法很受限制，尤其是在我们处理或混合那些不同取样率信号的时候。比较好的做法，是以不同的扫描率浏览波表，跳过其中的一些取样点。这实际上相当于缩减波表的大小，以产生不同频率。

比如说，我们只取偶数取样点，而我们的扫描频率便成了两倍。这将使输

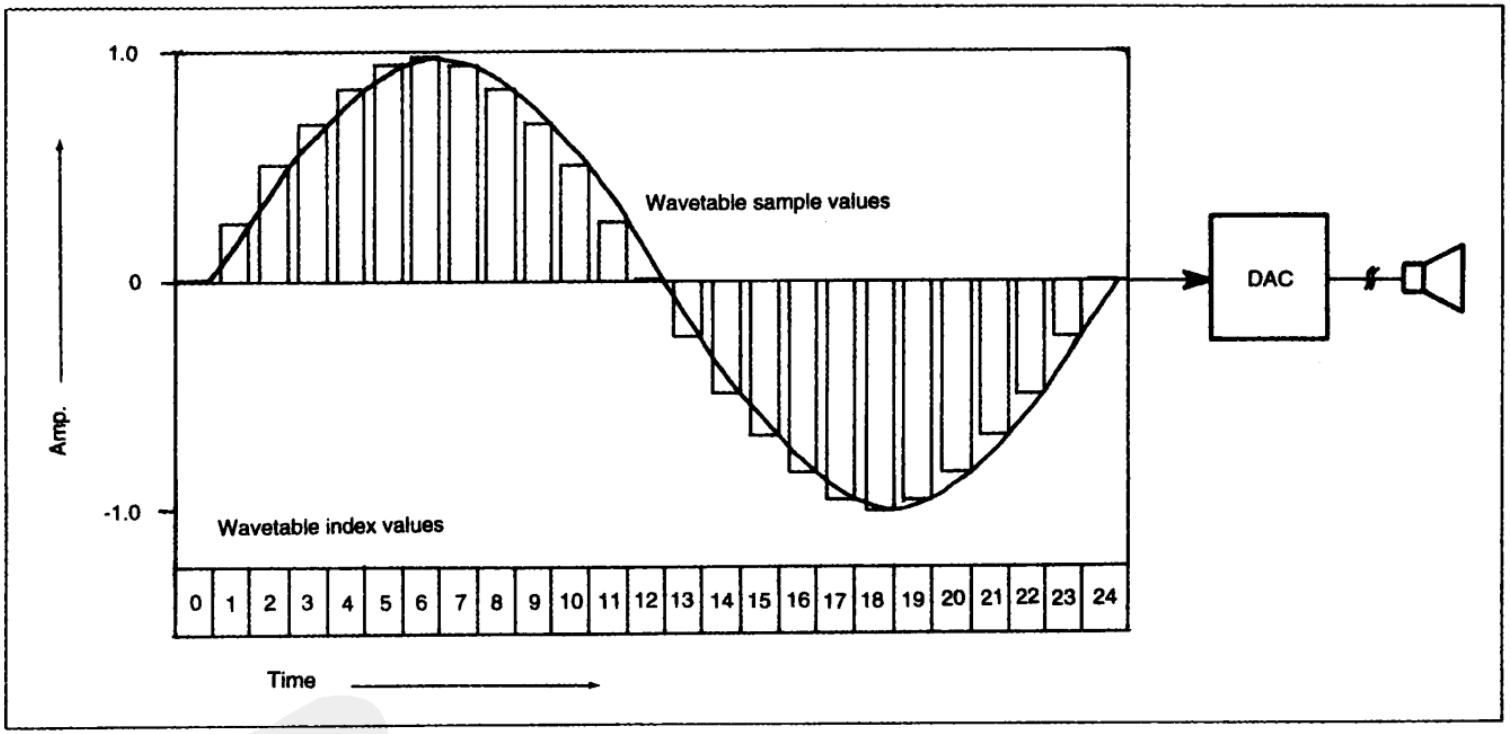


图 3.3　波表查表合成的图形描述。图下方的 0—24 是一些编号位置，或叫做“表索引值”(table index values)。每个索引点的声音样本保存在存储器中。这些样本用很多长方条形描述，它们勾画出图上方的正弦波。举例来说，Wavetable[0] = 0，Wavetable [6] = 1。要合成正弦波时，计算机查表得到连续索引位置内的取样值，并将其送入数模转换器，这样反复由表内读取。

出信号的频率比原来高一个八度。如果我们一次跳过两个取样点，那么频率又更高了(正确的说是一个八度加上五度)。在查表运算法中，递增值决定了要跳过几个取样点。此递增值会加上原先的相位位置，得到下一个取样点的读取位置。在最简单的情况下，如果我们要读取表内每一取样点，那么递增值就是 1，如果我们只想读取奇数点或偶数点，递增值就是 2。

数字振荡器算法(Algorithm for a Digital Oscillator)

我们可说振荡器对波表重新取样，以得到不同的频率。也就是说，它以现在的相位位置不断加上递增值，来跳过表内的某些取样点。所以最基本的振荡器运算法可以用下面两步的程序解释：

1. $phase\_index = mod_L(previous\_phase + increment)$
2. $output = amplitude \times wavetable[phase\_index]$

运算法中的第一步，是加法与模操作(以 mod_L 表示)。模操作是将加法所得值除以波表长度 L，而保留其余数，余数永远小于或等于 L(波表长度)。第二步则包含查表与相乘。这个动作所需要的计算量很少，但它假定波表内已经填满了波形值。

在一般情况下，如果波表的长度与取样率已固定，那么由振荡器传送的声音频率仅取决于递增值大小。给定的频率与递增值的关系可用下面的公式表示，这也是查表合成中最重要的公式：

$$increment = \frac{L \times frequency}{sampling\ Frequency} \tag{1}$$

比如说，若表长 L 为 1 000，取样率为 40 000，而指定的频率值为 2 000Hz，那么递增值就是 50。

这也导出了下面的频率公式：

$$frequency = \frac{increment \times sampling\ Frequency}{L} \tag{2}$$

到目前为止，已经对数字振荡器理论介绍得很多了。现在我们来看实际运算层面。

查表噪音与插值振荡器
(Table-lookup Noise and Interpolating Oscillators)

在前例中，所有变量都是 1 000 的倍数，所以得到的递增值都是简洁的整数解。然而，对于大部分算式(1)中的波表长度、频率以及取样率来说，得到的递增值将不是整数，而是小数点后带有诸多分数的真实数值。但是我们查表的方法是通过波表的指数定位的，波表的指数是整数。所以我们需要由实数的递增值来得到一个整数值。

实数值可以直接舍去小数部分(truncated)，从波表数值中得到整数值。这意味着要略去小数点后面的分数部分，如 6.99 的数字省略后会变成 6。

表 3.1　振荡器波表内的相位指数值以舍去分数法计算

相位指数	
计算	整数部分
1.000	1
2.125	2
3.250	3
4.375	4
5.500	5
6.625	6
7.750	7
8.875	8
10.000	10
11.125	11
12.250	12
13.375	13
14.500	14
15.625	15
16.750	16
17.875	17
19.000	19

假设我们使用的递增值为 1.125，如上面表 3.1 所示将计算出来的数值与舍去分数的值相比，这些因舍去而造成的不精确，表示我们得到的波形值只是近似值，而不是我们实际需要的严格意义上完全相同的值。其结果，会产生一些波形失真，称作查表噪音(*table-lookup noise*)(Moore 1977，Snell 1977b)。有许多种补救方法可以降低噪音。可以把波表加大，因为精细识别率的波表能够降低噪音。另外一个方法是四舍五入法，将递增值向上或向下移到最靠近的整数值，而不是整个舍去。在这方法中，6.99 会变成 7，远比 6 来的正确得多。但

是效果最好的是由内插振荡器所得。这在运算上所费的工夫更多，但它能产生非常干净的信号。

内插振荡器计算出该索引值上波表的数值该是多少，就像波表可以用该递增值所给定的精确相位来查询。换句话说，它对波表间的相邻数值作内插运算，所得的数值精确地对应于指定的相位指数的递增值(图 3.4)。

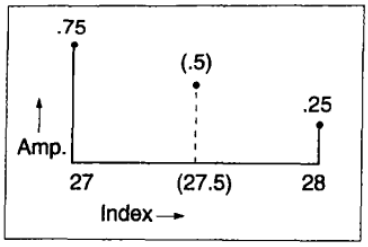


图 3.4 内插振荡器的原理，图示显示了波表的两个 x 点，在 27 和 28 的位置上，振荡器的相位递增标明是在 27.5 数值上，而这个地方没有录入点，于是，内插式振荡器在 27 和 28 数值间设定了 y 值来计算。

有了内插振荡器，就算使用较小的波表，也能够得到使用较大非内插波表的声音品质。假设 1 024 个数值的波表以内插振荡器运算，那么得到的正弦波的信—噪比约是 109dB(最差的情况下)，而使用非内插振荡器的相同大小波表，所得到的信噪比则是可怕的 48dB(Moore 1977)。这只是线性内插的结果，用更精细的内插方式可得到更好的结果 (Chamberlin 1985, Crochiere and Rabiner 1983, Moore 1977, Snell 1977b)。

这样，我们就完成了固定波形查表合成的介绍，后面，我们将解释在时间段中合成的哪些方面能够改变。

时变波形合成(Time-varying Waveform Synthesis)

目前为止，我们所见到的正弦波合成都是固定频率，这很容易理解。正弦波的最大值并不会随时间改变，所以信号的响度是固定的。因为只能控制频率跟时长，而不能控制声音的其他参数，所以在音乐上用途不大。即使用振荡器读取其他的波表，它还是不断地重复。制作更有趣声音的关键在于随时间改变的波形，在声音事件的时程内改变一个或多个合成参数。

包络、单元发生器与音色排秩(Enveolpe, Unit Generators, and Patches)

要制作时变波形，我们需要可以由包络控制的乐器，包络就是随着时间改变的函数。比方说，如果声音振幅依照时间改变，那么声音振幅的包络就叫做振幅包络。一般设计乐器的方法，是将它视为一个模块化系统，包含许多功能

不同的信号处理单元，当结合在一起时，可以建立依时间改变的声音。

单元发生器是数字合成的基本观念，一个单元发生器可以是一个信号产生器，或是信号调制器。信号产生器（如振荡器）合成如音乐波形或包络的信号。信号调制器如滤波器，可以接收信号，并以某种方式改变信号。

作曲家可将单元发生器连起来，组成排秩（patch），来构建一个用于声音合成的乐器。Patch 这个词是自早期模块化模拟合成器沿用过来的。在当时，声音模块是通过接插线串联。当然，音乐程序的内部联机是以软件撰写，不需串联任何实体电线。如果单元发生器输出的是数字，这个数字也可以当作其他单元发生器的输入。

合成乐器的图形表示法（Graphic Notation for Synthesis Instruments）

现在我们要介绍在数字声音合成文献中常用在出版物中来表示排秩路径的图形表示法。这个表示法的出现是为了解释最早的数字声音合成模块语言的操作，如 Music 4BF（Howe 1975）与 Music Ⅴ（Mathews 1969），和目前仍然有实用价值的一些数字声音合成模块语言操作。

每种单元发生器的符号有其独特的形状。图 3.5 是一个用图形来表示的查表振荡器（table-lookup oscilator），称为 osc，这是一个基本的信号产生器。它可以接收三个输入（振幅、频率、波形），并且产生一个输出（信号），振荡器读取单一波表，此波表在振荡器播放时不会改变。（更复杂的振荡器可以在一个声音事件的过程中，从数个振荡器中读取，详见第 5 章多重波表合成）。

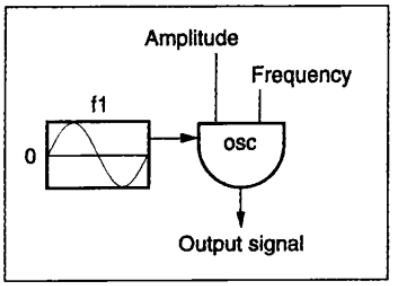


图 3.5　这是一个振荡器的图像表示，请参看书中的文字解释。
Amplitude=振幅　Frequency=频率
Output signal=输出信号

图 3.5 振荡器右上方的输入是它的频率值。左上方的输入则决定振荡器输出信号的波峰振幅。左边的方块则是波表 f1，内有一个正弦波（注：在某些实际应用中，不是输入频率值，而是直接输入原始的相位递增值到振荡器内。由于相位递增值不是音乐可感性的参数，我们假定此系统会根据算式 1，自动将频率转换至相位递增值。）

在排秩内使用包络(Using Envelope in Patches)

如我们将一个恒定值(比如说 1.0)设到振荡器的振幅输入端,那么波形振幅会在整个声音事件的过程中保持不变。但恰恰相反,大多数有趣声音的振幅包络都依时间改变。最典型的例子是,一个音符开始时振幅为 0,升高到其最大值后(通常经过规格化后就不会超过 1.0),再缓慢地减低到 0,(经规格化后的波形是将波形按比例调整,令其振幅包络在 0 到 1 之间,或是令其波形落在−1 到 1 之间。)包络的开始部分称为起冲(attack),而包络的结束部分称为消去(release)。

商用模拟合成器将振幅包络定义为四个部分:起冲(attack)、衰减(initial decay)、延留(sustain)(延持的时间,比方说,琴键依然被按着的时候),以及消去(release)。通常可缩写为起衰延消(ADSR)(图 3.6)。ADSR 的概念对于描述包络形状很有用,比方说"让起冲再锐利一点"。但要描述音乐包络时,四段式的限制是不足且不合时宜的。描述振幅形状是个非常精细的操作,所以需要更灵活的包络编辑器,让音乐家任意编辑其形状(见第 16 章)。

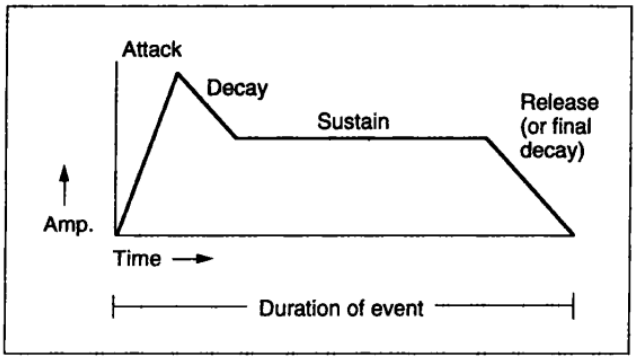


图 3.6 简单的 ADSR 振幅包络,显示音的振幅随着时值而变化。
Attack=起冲 Decay=衰减 Sustain=延留 Release(or final decay)=消去(或最终衰退)
Amp=振幅 Time=时间 Duration of event=事件的时值

图 3.5 的乐器,将包络信号输入到振荡器的振幅输入端,就很容易产生时变振幅。我们现在更接近以音乐为目的来控制振荡器。如果设定好包络的时长与曲线,就能控制每个音的振幅变化。

在作曲时,以手动方式给定每个声音事件的包络实在非常烦琐,所以我们需要的是一种更简单的程序,能依不同事件的长度来调整包络。有种解决方式是用另一个查表振荡器(图 3.7 中以 env_osc 表示),但此时的波表 f1 内填入的不是正弦波,而是 0 到 1 之间的振幅包络值。包络振荡器(envelope oscillator)的渐增值非由频率值,而是由音的时长来计算得来。比方说,若音长度为 2 秒,

那么包络振荡器的一个周期是两秒，频率值便是 0.5Hz。所以包络振荡器(env_osc)在整段时间内只读取一次波表。因为，每次取样，包络振荡器是根据储存中的包络 f1，产生它的输出值的。这个数值输入正弦波振荡器(osc)的左输入端(振幅)。当正弦波振荡器(osc)由波表 f2 查询取样点后，此取样值依振幅输入值作内部变化，这样，就通过包络振荡器(env_osc)得出相应的结果。

图 3.7a 是如 17 章所述的，由合成语言定义的典型乐器(instrument)，而图 3.7b 则是此结构的另外一个实现方式，在合成器上可能更常见。此图将包络振荡器换成单纯的包络产生器(envelope generator)env_gen。此包络产生器 env_gen 接收时长、波峰振幅以及波表；它以一定的时长读取整个波表，并依给定的波峰振幅调整大小。

如你所想的，我们也可以将包络产生器接到振荡器(osc)的频率输入端，以得到如揉音(vibrato)或滑音(glissando)的音高改变。的确，是可以用许多不同方式连接振荡器及其他单元发生器，来产生许多不同的声音。这种交互连接的振荡器，是第 4 章到第 8 章间所述的合成技巧的基础。

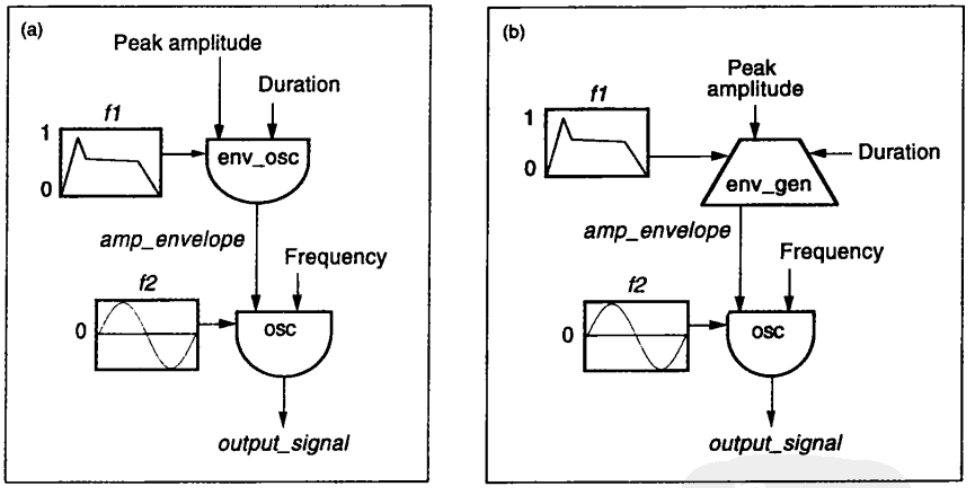


图 3.7　是振荡器的时变振幅控制。(a)作为包络产生器的振荡器。上图显示的包络振荡器(env_osc)，是用来产生控制下方振荡器(osc)的正弦波振幅包络的。包络振荡器仅读取一个周期，此结构在合成语言中也能找到。(b)是与(a)相同的结构，用了一个简单的包络产生器单元(env_gen)。此单元的输入有时长、振幅峰值以及波形。这个结构是更具合成器的典型性。

Peak amplitude=振幅峰值　Duration=时值　amp_envelope 振幅包络　Frequency=频率
output_signal=输出信号

软件合成(Software Synthesis)

到目前为止,我们以抽象方式讨论了数字合成,后续章节将以更实际的方式讨论合成系统。数字声音合成中最精确且最有弹性的方式,是在普通型计算机上执行的软件合成程序。软件合成意味着由程序执行所有计算,包含计算取样点串流,所以使用者可任意改变之。最典型的软件合成语言是 Music V 语言(Mathews 1969)或其他许多 Music N 的不同版本。

相对于软件合成的,是使用特别电路计算的硬件合成。硬件合成虽有速度快的好处,但其弹性与合成运算的复杂度都受限于硬件设计。最典型例子就是功能固定化了的商用键盘合成器。其内部电路无法被重新修改来执行其他厂商所开发的技术。

在某些情形下,软件与硬件合成的差别并不那么明显。假设有个系统是用可程序化的数字信号处理器(digital signal processor, DSP)构成,并具备大的存储器,它也可能执行普通型计算机上的某些合成软件。(详见第 20 章,DSP 的架构。)

总之,这些在计算机音乐上的先驱工作都是以软件合成开始的。今天有很多合成软件可在便宜的个人计算机上执行,甚至已经内建高品质的模/数转换器(ADC)与数/模转换器,或可自行添购。软件合成的一大好处是,只要音乐家有足够的耐性等待结果,使用小计算机也能够实现任何合成算法,即便它的计算需要耗费许多时间。今天,你只要有少许器材与创作音乐的意图,计算机已准备好为你做出高品质的音乐合成。

乐器编辑器与合成语言(Instrument Editors and Synthesis Languages)

当代软件合成程序可分为两大类:(1)图形乐器编辑器(graphical instrument editors),以及(2)合成语言(synthesis languages)。有了图形乐器编辑器,音乐家可以直接在计算机屏幕上连结图标,做成排秩(patches)。每个图标都代表一个单元发生器。(在第 16 章会介绍这方面内容并举例。)

使用合成语言编辑时,音乐家可撰写文字,由合成程序转译来给定声音。图 3.8a 显示了一个文本表述,它与图 3.7a 显示的是一个乐器。这个例子使用的是一个简单的假想合成语言,我们称作 Music 0 语言。"←"符号表示"所给定的值"。比方说,包络振荡器的输出设为(连接到)振幅包络的信号变量。接着,振幅包络数值,在每个取样时间距,会被送入振荡器(osc)模块的振幅输入。

(a)

```
Instrument 1
     /* env_osc arguments are wavetable, duration, amplitude */
     amp_envelope ← env_osc  f1  p3  1.0;
     /* osc arguments are wavetable, frequency, amplitude */
     output_signal ← osc  f2  p4  amp_envelope;
     out output_signal;
EndInstrument 1;
```

(b)

```
/*  Score line for Instrument 1      */
/*  p1        p2      p3        p4    */
    i1        0       1.0       440
```

图 3.8　乐器及其乐谱的文字表示。(a)是与图 3.7 相同的乐器。在"/ * "与" * /"两个标记之间的文字为注释(comments)。由 p 开始的参数域(parameter fields)给定了一些数值,它们是来自字母数字谱。如(b)所示。p_3 给定时值,p_4 则是频率。注意第二个振荡器(振幅)的第三个自变量(argument)是由第一个振荡器所产生的振幅包络(amp_envelope)信号提供的。(b)谱是为(a)的乐器服务的。第一个区域是乐器编号,第二个参数域给定起始时间,第三个参数域是时长,第四个是频率。

图 3.8b 显示了一个简单乐谱,它提供了此乐器所需要的参数值。(第 17 章解释合成语言的基本语法与特点。)

Instrument=音础　env-osc arguments=包络振荡器自变量　wave table=波表
duration=时值　amplitude=振幅　frequency=频率　output-signal=输出信号

声音合成的计算要求(Computational Demands of Synthesis)

执行声音合成算法的每个步骤都需要一段时间。对于较复杂的合成算法,计算机可能无法在一个取样间距时间内就完成计算。

更具体的分析,可参见下列使用查表法计算声音取样所需要的步骤。

1. 将原本波表相位位置加上递增值,得到新的位置。

2. 如果新的位置超过了波表末端,便减去波表的长度。(也就是说,执行取余动作)

3. 将新的位置储存,供下次计算使用。(见步骤 1)

4. 在新的位置上查询波表内的值。

5. 将此数值乘上振幅输入。

6. 将结果送出。

很重要的一点是，执行每个步骤都需要一些时间，比方说，可能需要一微秒来计算以上步骤。但如果我们的取样率是每秒 50 000 个点，那么每个取样点的时间只有 1/50 000 秒，也就是 20 毫秒(20 000 纤秒)。这意味着，就此计算机而言，若有数个简单振荡器时，是很难完成实时运算的。如果程序更为复杂，加入了滤波器，延迟，或者更多的查表运算，随机功能等，以及与音乐家互动的时间，那就会造成甚至一个乐器都不可能在实时中实现。那么，什么是实时呢？在这里，实时指的是可以在一个取样间距时间内完成这个取样点的计算。

非实时合成(Non-real-time Synthesis)

某些合成与信号处理技术的运算量很大，所以很难实现实时运算。这意味着在运算开始到能听到声音之间起码有数秒钟的延迟。这种有延迟的系统称作非实时系统。

早期的计算机音乐中，非实时系统是唯一的选择。比方说，在 1965 年到 1968 年间，J. K. Randall 在普林斯顿大学所作的 *Lyric Variation for Violin and Computer* 的两分钟(Cardinal Records VCS 10057)，需要花 9 个小时来计算。当然，如果发生了什么错误，整个程序必须重来。即使这是如此费力的工作，许多能使用这些设备的作曲家还是做出了相当长的计算机合成音乐作品(参见 Tenney 1969，Von Foerster and Beauchamp 1969，Dodge 1985，Risset 1985a)。

声音文件(Sound Files)

因为计算每个取样点的时间超过取样间距，所以软件合成程序输出的是声音文件。声音文件就是存放在磁盘或磁带上的数据文件。当曲子的所有取样点计算完成后，可以用数/模转换器将之播放。

声音文件包含标头文字(header text)以及表示声音取样点的数值。标头包含文件名称，取样的相关信息(取样率，每个取样点的比特数目，声道数等)。通常，在数据结构安排中，取样被称作帧(frames)；如果有数个声道，那么每个帧里就有数个取样点。所以，取样率其实给定了每秒钟帧的数量。

如同其他计算机程序一样，不同的档案格式可以共存，在不同的格式间转换是在计算机音乐工作室中很常见。

实时数字化合成(Real-time Digital Synthesis)

由于计算机已经变得更快、更小、更便宜,数字合成技术也变得更有效率。早在 20 世纪 70 年代中期,建立足够快的数字合成器已经有实现的可能,该数字合成器能够为一个取样在取样间距内作出所有必要的计算。电路技术的先进,使得过去体积过大的合成器已经被微小的集成电路所淘汰,集成电路能在实时状态下实现多声的合成运算。

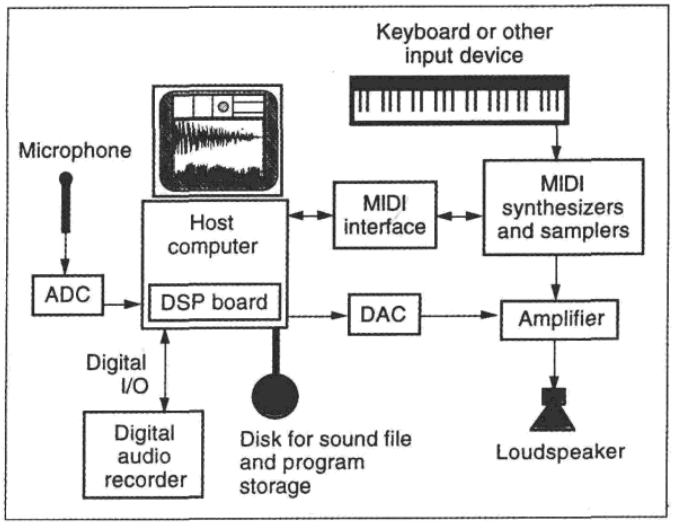


图 3.9　简化后的典型数字录音合成工具的略图。音乐家利用琴键或其他输入工具,或由在主计算机上运行的程序来控制合成器。声音可以由模/数转换器(ADC)录下后存在磁盘内,之后由数/模转换器播放。在能够处理多媒体的计算机上,除了 MIDI 键盘,所有组件都可能是内建在计算机中的。
Keyboard or other input device=键盘或其他输入装置　Microphone=话筒　ADC=模数转换器　Host computer=主机　(MIDI interface) =MIDI 接口　(MIDI synthesizers and samplers)=MIDI 合成器和采样器 DSP board=数字信号处理板　DAC=数模转换器　Amplifier=扩大器　I/O=数字　Digital audio recorder=数字录音　Disk for sound file and program storage=声音文件和程序储存硬盘　Loudspeaker=音箱

图 3.9 显示了实时计算机音乐合成系统的略图。事实上此系统有三种产生数字声音的方式:(1)非实时软件合成,直接在计算机上运算,后由数/模转换器播放。(2)由数字信号处理板(DSP)实时合成,经数/模转换器播放 。(3)采用合成器,通过乐器数字接口(Musical Instrument Digital Interface)(MIDI,见第 21 章)的控制,实现实时合成。

实时合成器的一大优点是音乐输入装置(musical input devices)(或称演奏控制器 performance controllers)如键盘、脚踏板、摇杆、按钮或旋扭等。所以音

乐家在听见声音时可以直接改变。音序器(Sequencers)以及乐谱编辑器(score editor)可以录下并编辑这些操作,而计算机上的音色排秩编辑器(patch editor)可以随时改变合成,改变信号处理排秩。

本书后半部分将更详细地讨论实时系统。尤其是在第五部分,将讨论数字合成器的内在,以及 MIDI 协议。第 14 章与第 15 章将讨论演出控制器及演出软件(详见 Alles 1977a, Buxton et al, 1978, Strawn 1985c, Roads and Strawn 1985, Roads 1989)。

非实时与实时合成的比较 (Comparing Non-real-time Synthesis with Real-time Synthesis)

非实时软件合成是数字声音输出的基本方法,而今仍占有一席之地。如我们已强调过的,采用排秩化音乐语言软件合成的优势是它的可程序化,因此它提供了音乐上的处理弹性。商用实时合成器总是设置一些出厂公司的限制,而软件合成则是全开放式的,让使用者自由建立个人化的乐器,或任意复杂的合成算法。许多新的声音合成实验及信号处理方法只有在非实时软件的形式上才能做到。

软件合成的另一个巨大优点是其可编辑乐谱的灵活性。即使只有一个简单的合成设备,经由乐谱语言(score language)(会在之后讨论)可以变得极端详细而复杂,远超过人类演奏者或是 MIDI 器材能够传输的极限。

然而,非实时软件合成的缺点也很明显。时间都浪费在等待运算取样的花费上了。切断声音与实时人体动作的关系——我们不能在听到声音时直接改变它。这种困境使计算机音乐令人不容易亲近。而可程序化的好处同时也是缺点,使得我们要得到一个简单的乐句都得花上很大的力气来撰写程序,如同撰写较难的乐句一样。即便比较任意自然的包络都需要先计算并键入许多数字。因此,使用非实时软件合成来做音乐的确较为困难。

幸运的是,硬件的大幅进步使得越来越多的合成方法成为实时运算。由 DSP 微处理器电路所建构的商业合成器,纳入了编程合成运算法的灵活性。只有那些最深奥而复杂的方法,如参数预测(parameter estimation)与分析—重合成(analysis-resynthesis)(第 7 章与第 13 章)在低价实时的硬件上仍无法实现。所以我们现在可以根据音乐应用的需要,在实时与非实时合成之间进行选择。除了节省时间外,实时合成器有能够演奏的优点——令音乐家的动作转为可听见的声音。

音乐声的表述(Specifying Musical Sounds)

现在我们把方向转到合成系统如何实现一段音乐。传统的作曲方式是先选择乐器,再在纸上写谱,由演奏者演出给定的音乐事件,其中留有演奏者演绎及把握乐器的空间。但在数字声音合成里,其可能性远远超过传统的谱纸与墨水。

声音对象(Sound Objects)

在传统乐理中,音符是静态的、同质的、一致的事件。现代合成技巧将音乐事件视为一个更广义的概念,称为声音体(sound object)。(Schaeffer 1977, Chion and Reibel 1976, Roads 1985f)。此声音体的观念非常有用,因为它可涵盖比一般定义下的音符更长、更复杂的声音。一个声音体可能有数百个短的子事件(如向量及粒子合成)。或者它可由数十个或更多随时间变化的参数控制,使得它能够由一个音高/音色变化到另外一个。

控制声音对象复杂参数变化之重担便落到了作曲家身上。这里必须回答此问题:我们如何给定所有随着时间改变的数值?在下一节当中,我们将说明在一般合成中会需要多少资料,接着在音乐家界面一节中将介绍五种策略。

加法合成表述问题举例 (Example of the Specification Problem for Additive Synthesis)

加法合成是声音合成中历史最悠久的一个。如其名,它将数个正弦波振荡器的输出相加,成为一个合成的音波。

图3.10显示了加法合成的数字合成乐器。此乐器的每个振荡器都有频率及振幅包络。频率包络是在[-1.0, +1.0]之间的时变函数,此包络会影响到送入包络振荡器(env_osc)输入端的峰值偏差(peak deviation)。如果峰值偏差为100,而频率包络的最低数值为-0.1,那么由频率包络输出的值就会是-10。此加法器(+)将此频率值与更低的振荡器的中心频率相加,使频率比原先的中心点低。如果中心频率给定为440Hz,那么频率包络会使它在某一时刻降到430Hz。

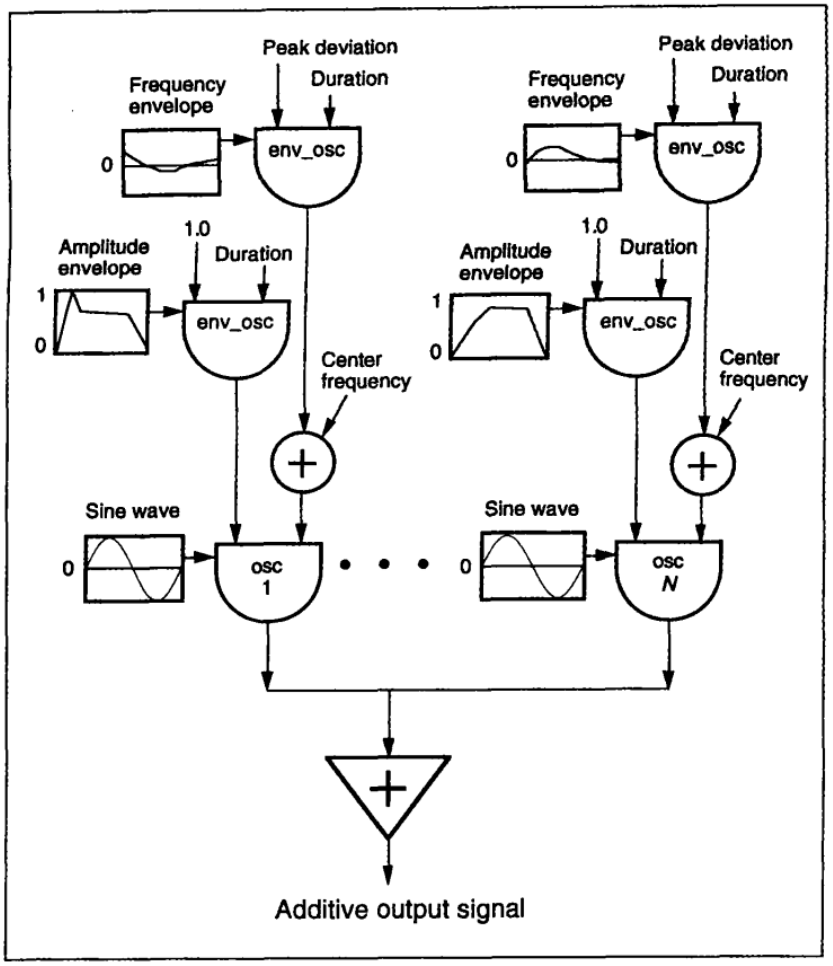


图 3.10 这个排秩扩展了图 3.7 的乐器，成为简化的加法合成乐器。每个正弦波振荡器会经过振幅与频率包络调整。许多正弦波的输出相加后形成一个样本。此排秩可加入更多的三振荡器单元以便做出更复杂的声音。

Frequency envelope=频率包络　Peak deviation=峰值偏差　Duration=时值
Amplitude envelope=振幅包络　Sine wave=正弦波　Center frequency=中心频率
Additive output signal=加法输出信号

注意此乐器中的每条垂直行内都有两个包络产生器和一个声音振荡器。我们把这样的一个单位叫做一个声源(voice)。在此只绘出了两个声源，中间的圆点表示省略了其他声源。只要确定了资料，这样的乐器可以产生非常多样的声音。

现在，问题回到如何为图 3.10 的乐器选定参数上了。对每个声源和每个事件，乐器需要以下的一些参数：

1. 音频振荡器 osc 的中心频率
2. 峰值振幅（图中设为 1）

3. 振幅包络
4. 振幅包络的起始时间
5. 振幅包络的时值
6. 频率包络
7. 频率包络的起始时间
8. 频率包络的时值

如果这个乐器有15个声源，每个声源需要这8个参数值，这代表一个声音事件需要120个参数值。

所以无论声音合成硬件变得多么强大，指定这些控制参数的问题依然存在。在第4章中我们将更仔细介绍加法合成的资料需求。下一个部分将介绍适用于所有合成技巧的6种一般性策略。

音乐家界面(The Musician's Interface)

为计算机与合成器提供所需合成资料的方法大概可分为以下6种：

1. 音乐性输入装置
2. 演出软件
3. 编辑器
4. 乐谱语言
5. 算法作曲程序
6. 声音分析程序

图3.11将这6种方式绘出，前五种对应于本书第五部分中提到的音乐家界面。最后一种在第六部分会提到，下面的6个部分将简单介绍以下每一个种类。

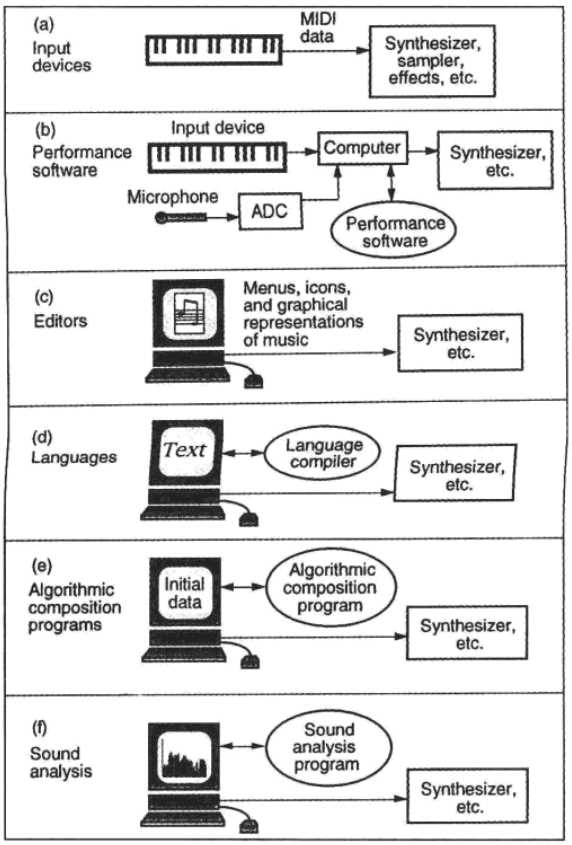


图 3.11 显示了音乐家界面：为计算机或合成器指定合成数据的 6 种不同方式。(a)输入装置可以将必要的信息直接传输到合成器，无论是否通过计算机作为中介。(b)演出软件转译演出者的动作，甚至可以即兴演出。(c)编辑器让使用者通过交互式图形技术给定资料。(d)使用语言将规约编码为精确文本。(e)算法作曲程序在开始产生音乐以前，通常需要来自作曲家少量的初始参数资料。(f)声音分析自动从输入的声音撷取数据，以供修改或重新合成。

(a)
Input device=输入装置
MIDI data=MIDI 数据
Synthesizers, samplers, effects, etc.=合成器，采样器，效果器等

(b)
Performance software=演奏类软件
Input device=输入装置
Microphone=话筒
Computer=计算机
ADC=模拟到数字转换器
Synthesizers, etc.=合成器等

(c)
Editors=编辑器
Menus, icons, and graphical representations of music=菜单，图标以及音乐的图形化表示

(d)
Language=语言
Language compiler=语言编译器
Synthesizers, etc.=合成器等

(e)
Algorithmic composition programs=算法作曲程序
Synthesizers, etc.= 合成器等

(f)
Sound analysis=声音分析
Sound analysis program=声音分析程序
Synthesizers, etc.=合成器等

音乐化输入装置(Musical Input Devices)

音乐性输入装置是音乐家所控制的实体乐器(见第14章)。这个乐器直接与音乐家的动作相连,以产生声音。电子输入装置切断所需原物理电源发声系统,而直接当成资料输入界面,这样,这些装置就比传统乐器更有潜在的适应性。比方说,使用电子乐器时,一个单吹管控制器就能像创造女高音声音一样便利地创造出男低音音响。的确,电子输入装置使用上如此简单,所以有一股研究方向,是朝重新添加物理性的难度,创造出费力的感觉,以得到富有表情的演出。

使用实时音乐性输入装置的好处很明显,虽然将它们与计算机串联的技术问题是无法避免的。传统声学乐器经过数百年发展,然而它们数字化的发展却才刚刚开始。音乐性输入装置最适合对几种音乐参数作精细的控制。比方说,琴键可以指定音高,同时琴键压力的速度可以决定高频振荡器的强度。大多数MIDI键盘有一个或多个持续音控制器(continuous controllers)(如踏板,调制轮,或是摇杆等)。这些控制器可以用来支配任何可操控的参数,所以我们可以设定脚踏板来控制整个声音的振幅,而调制轮则可使基本音高的变成弯音。

演出类软件(Performance Software)

实时演出软件的应用,因MIDI系统的进一步普及而越发受到欢迎(详见第15章)。演出软件包含音序器等,它可记忆琴键演奏并重新播放。音序器记录纯控制数据(如琴键落下时的激活时间,表示一个音符的开始。),而不是音频波形的取样。计算机音乐亦提供超越传统独奏演出的机会。比方说操控乐团内指挥层级的控制信号。

安装了眼(摄像机或其他种类的感应器)和耳(麦克风及声音分析软件),通过基于计算机的乐器,透过演出软件内定义的程序,可以用更任意而复杂的方式来响应人的演出动作。在音乐会中越来越常见由计算机控制的合成器与人类演奏者一同即兴。另外的用法,是弹性地播放预先准备好的乐谱,取代僵化的磁带录音机的演出方式。

一个简单的演出软件范例是:先设定好某些情况,当由键盘演奏某段乐句后,驱动一段已预先录好的乐谱,而用一个高音C来停止播放,并用调制轮决定预录好段落的速度。

编辑器(Editors)

编辑器程序可以让音乐家创造和改变一个文本、一个声音或一段影像(详见第 16 章)。许多交互式的编辑器使用图示技术,为音乐家提供更有效率的环境。被编辑的素材可由简单的动作来快速地剪贴或改变。

图示编辑器可将音乐想法快速模型化,所以它们在个人工作室里用得最多,在那里,要用一定时间去研究推敲。音乐想法可由编辑器逐渐建立起来,而音乐家常常可以听到正在修改中的结果。

因为音乐存在不同的层次和视角,所以有许多不同形态的音乐乐器也是很合理的。要设置加法合成器的演奏,可以使用乐谱、音础器和函数编辑器。我们在文本编辑器中对每个声音对象输入参数,或操纵图形符号(如一般的乐谱,或钢琴卷帘谱)。在乐器编辑器中,由振荡器及包络发生器等单元发生器来配置加法合成器。在最后的编辑阶段,令程序将排秩写入合成器内。函数编辑器提供数种定义时间函数的方法(波形与包络),包含图示方式或数学公式等。我们使用函数编辑器以建立不同振荡器的振幅与频率包络。

语言(Languages)

或许指定音乐的最精准方式,包括准备音符列表(note list)或演奏列表(play list),都是乐谱语言(score language)的一部分(详见第 17 章)。乐谱语言定义乐器参数的语法,而将之列在个别参数项中(parameter fields,简写为 pfields)。

第一个乐谱语言的范例是如图 3.8b 中显示的简单的乐谱。习惯上,在乐器名称后的第一个参数给定起始时间,第二个参数给定时长。后续的参数则依照乐器特性而有不同定义。比如图 3.12 中数字谱的第一行,说明在这一音乐事件中使用乐器 1,从时间 0 开始,演奏 1.0 秒,频率为 440Hz,振幅为 70dB,并且使用 3 号波形。(谱下方粗体的两行就是乐谱,其余行是注释。)

```
;  p1     p2        p3       p4          p5          p6
; Ins    Start      Dur.   Freq.(Hz)   Amp.(dB)   Waveform
  i1       0        1.0      440          70          3
  i2      1.0       .5       660          80          4
```

图 3.12 数字谱示例。两行批注后接两行谱。第一行为乐器 1(i1)指定一个音符,第二行是为乐器 i2 指定的音符。
Ins=乐器 Start=开始 Dur.=时值 Freq.=频率 Amp.=振幅 Waveform=波形

乐谱语言同时也包含函数表之定义，也就是由乐器所使用的包络与波形之定义（见第17章）。

传统乐谱语言都是以数字表示的：乐器编号、音高，以及振幅都以数字表示。其他种的乐谱语言支持更为“自然”的音乐记述方式，比如说，允许使用等音距之音名。（更多对乐谱语言的讨论，详见 Smith 1973；Schottstaedt 1983，1989a；Jaffe 1989 以及 Loy 1989a 与第17章。）

乐谱语言的主要优点恰恰也是它的缺点：它的精准性与详细程度。使用乐谱语言时，音乐家必须用文字及数字键入乐谱。并不是所有作曲家都愿意每次如此详细输入乐谱的，比如，在前面提到过的加法合成，音乐家得对每个声音体键入约120个数值。另外，乐谱语言使音乐家精准地给定乐谱，其精细程度使演奏家永远无法如实演出。

算法作曲程序（Algorithmic Composition Programs）

有些早期的计算机音乐作品是算法作曲：乐谱的创造是由作曲家/程序设计师设计的程序完成的。（Hiller and Isaacson－1959，Xenakis 1971，Barbaud 1966，Zaripov 1969）。比方说，计算机可以依概率分布或其他程序来计算声音的参数。

例如，假设我们把初始数据输入到算法作曲程序，然后，让它自动生成包含加法合成所需所有参数的完整乐谱。第19章显示算法作曲程序中可能采取的诸多策略。这样我们就明白了每个程序的初始数据的特点都不相同。对基于概率来计算乐谱的程序而言，作曲家只需要给定下述乐谱的一般特性：

1. 段落的数目。
2. 段落的平均时长。
3. 段落内音符密度的最大值及最小值。
4. 将频率与振幅包络编组，纳入音色类别。
5. 在音色类别内每种乐器演奏的概率。
6. 每个乐器最长与最短的演奏时长。

在这种情况下，控制自然是全面且带有统计性的。作曲家可以决定乐谱整体的特性，但所有细节都是由程序计算。在其他程序中，这些数据可更为详尽且予以更具风格的限制。

声音分析（Sound Analysis）

如同音乐一样，声音也可由许多种方法来分析。目前已建立的声音分析领

域着重在三个方向：音高、节奏以及频谱。我们可以用这些分析的输出结果来驾驭合成，如同以一个卷积器(convolver)将一个声音的节奏对应到另外一个声音的音色上(Roads 1993a，第 10 章)。音高分析器可以分析人声，而驱动另外一个数字振荡器的合音(accompaniment pitch)(第 12 章)，或者用频谱分析仪分析出时变频率与振幅曲线，以供加法合成使用(第 13 章)。

结论(Conclusion)

物理与电子声学的发展，开启了许多音乐化声音生成实验的新途径。在此领域的创造，代表当今最前卫的音乐发展。新的声音，加在新的节奏、和声、调性观念之上，这使得以一般的音乐美学标准来评论今日音乐变得极端困难。

——H. Miller(1960)

数字声音合成的音乐潜力之探索才刚刚开始，而我们还有许多未知之处。对现在而言，数字科技允许精确且可重复的声音合成。有了适当的硬件、软件以及播放系统，我们可以产生音频质量极高的音乐信号。也许，比精准性更为重要的是可编程性，它将转为音乐上的可塑性。有了足够的内存与演算时间，无论有多么复杂，计算机都可以实现任何合成运算。

虽然硬件不断在速度上有所突破，但寻找适当的控制数据以驱动合成引擎的问题始终存在。声音合成所面临的挑战之一，是如何将我们想要制造的声音的参数构想出来并传达给机器。这些给定方式包括如本书第五部分中六个章节所讨论的音乐家界面，以及如第四部分所讨论的声音分析。

音乐理论已落后计算机音乐的实践约半个世纪。引领作曲家向前的合成技术不断探索着所有可能性，留下的声音地图供后人按图索骥。这样的实验时期的音乐历史表明，此阶段将会走向一个日渐稳固的时代——当许多今日的实验变得平常，目前看来似乎很激进的资源日后将会变得随手可得。音乐作曲会进入一个更细致的、新的时代，而在系统架构内的编曲配器法(orchestration)的问题也会被重新提出，一如交响乐队的年代。

第 4 章　采样合成与加法合成
(Sampling and Additive Synthesis)

thesis)

加法合成中的声音分析方法(Methods of Sound Analysis for Additive Synthesis)

附加分析/再合成中的数据缩减(Data Reduction in Analysis/Resynthesis)

线段近似法(Line-segment Approximation)

主要成分分析(Principal Components Analysis)

频谱插值合成(Spectral Interpolation Synthesis)

频谱建模合成(Spectral Modeling Synthesis)

沃尔什函数合成(Walsh Function Synthesis)

结论(Conclusion)

本章介绍声音采样和加法合成的多种形态。这些技术是计算机音乐的基础，每一位对声音合成有兴趣的音乐家都应当理解。

采样合成(Sampling Synthesis)

在一般说法中，采样(sampling)是将一段相对短暂的声音以数字方式录下。“采样”一词是由已有的概念即数字样本(samples)以及采样率(sampling rate)的概念衍生而来的。在市面上，不管是否带有琴键，采样乐器(sampling instruments)已非常普遍。所有的采样乐器设计都围绕着一个基本理念，是使其能将先期录制的声音移到期望的音高上回放出来。

采样合成与第1章所介绍的固定波形合成的经典技术不同。采样系统不是由一小段固定波表中读取一个波形周期，而是读取一个大的波表，里面有数千个单独周期，即几秒钟的先期录制声音。由于采样波形在声音事件的起冲、延留、衰减等不同部分的变化，所以能得到丰富且随时间变化的声音。我们可随意决定采样波表的长度，唯一的限制是采样器的存储容量。大部分采样器提供光盘或磁盘驱动的接口，可以迅速地将很多采样加载到采样器内。

具体音乐与采样：背景(Musique Concrète and Sampling: Background)

对录制的声音加以操纵的作曲法最早在20世纪20年代就已出现，如作曲家米约(Darius Milhaud)、欣德米特(Paul Hindemith)和托赫(Ernst Toch)在演奏会中就已利用变速留声机实验(Ernst 1977)。磁带录音原由德国在20世纪30年代开发，磁带可以剪贴，所以可有弹性地编辑，重新安排录下的声音序列。直到第二次世界大战之后，磁带录音机才得以为音乐家所使用。

经过了20世纪40年代后期在变速唱机上的实验后，皮埃尔·舍费尔(Pierre Schaeffer)在1950年间，于巴黎成立具体音乐工作室(Studio de Musique Concrète)(见图4.1)。他与皮埃尔·亨利(Pierre Henry)开始使用磁带录音机录制并制作具体声音。具体音乐(musique concrète)所指的，就是使用由麦克风录下的声音，而不是在纯电子音乐中用合成手段生成的声音。但该词同时也指用声音来操作的一种作曲方式，具体音乐作曲家是直接用声音对象来作曲的(Schaeffer 1977, Chion 1982)。他们的作品需要新的图示记谱法，而不属于传统交响乐记谱的范畴(Bayle 1993)。

图 4.1 1960 年舍费尔(Pierre Schaeffer)的具体音乐工作室,位于 rue de l'Université,巴黎。工作室左方有三台磁带录音机,以及一个电唱盘。右边是另一台磁带录音机,以及多拾音头的"浮诺振"(Phonogène)器材(见图 4.2)。(照片由巴黎 Groupe de Recherches Musicales 所提供)

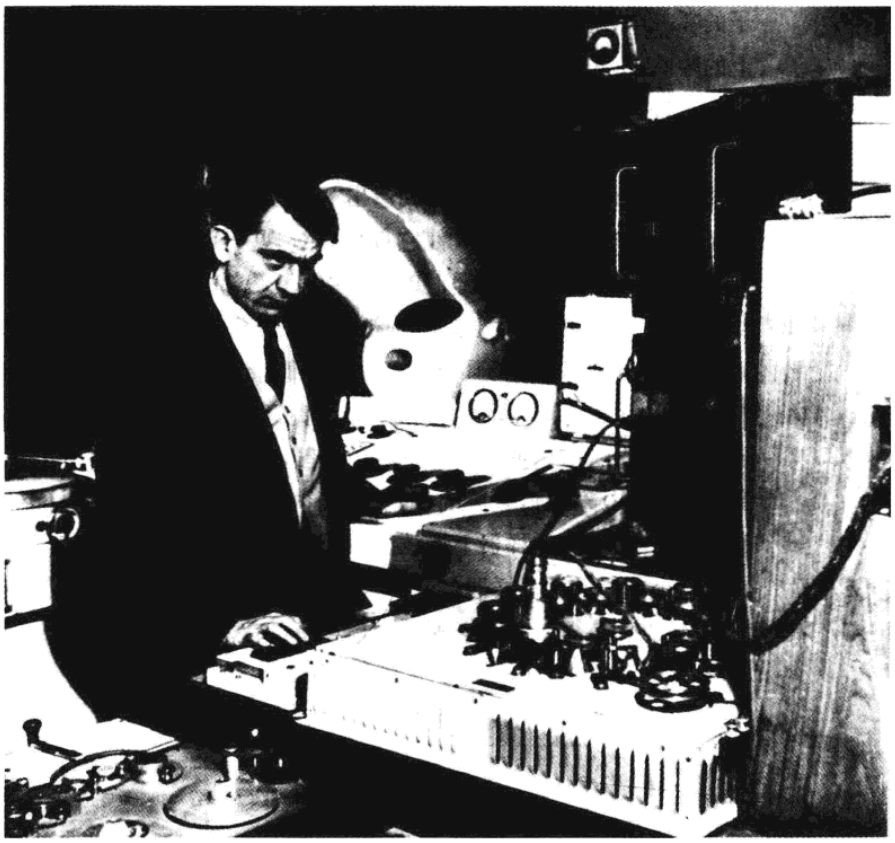


图 4.2　舍费尔(Pierre Schaeffer)和 Phonogène 设备，即一个磁带移调器和磁带时间延展器。1953 年，巴黎。

现代取样乐器是建立在应用光电或磁带循环设备基础的原则上，如韦尔特(Edwin Welte)的 Light-tone Organ(柏林，20 世纪 30 年代)，Sammis 的 Singing Keyboard(Hollywood，1936)、舍菲尔(Pierre Schaeffer)的 Phonogène(图 4.2，巴黎，20 世纪 50 年代早期)，休·勒凯恩(Hugh LeCaine)的 Special Purpose Tape Recorder Ottawa，1955)、钱姆博仑(Chamberlain，洛杉矶，20 世纪 60 年代晚期)以及美乐特朗(Mellotron，伦敦，20 世纪 70 年代早期)。这些设备播放储存声音的光盘(将波形以光学影象方式编码)或是磁带循环。依据所选择的磁盘或磁带和按下的琴键，这些设备内的回放头会依照所按琴键的音高，来播放磁盘或磁带上的声音。

Singing Keyboard 的设计者弗雷德里克·塞米斯(Frederick Sammis)，在 1936 年曾描述这样一种乐器的潜力：

“假想我们要使用这个机器，来当作卡通配音的特殊设备。很明显的是在这个机器上，作曲家可以实验许多台词与音乐的组合，并能立刻听到它完成后的样子。这个设备可能有超过十个声轨，可将声音与影片同步录下，可以是鸭

子的呱呱声，猫的喵声，牛的哞声……也可以是狗吠，或是人声在某一恰当音高上的哼鸣。"(Frederick Sammis, quoted in Rhea 1977)

也许，最著名的数字化技术以前的采样器是美乐特朗，它是带有数个旋转磁带循环的昂贵乐器。美乐特朗于 20 世纪 70 年代在摇滚乐团中大受欢迎。他们使用这乐器，在流行乐中创造出"交响乐的"或"合唱的"背景音。但是美乐特朗复杂的电子机械设计，使它仅昙花一现。因为磁头的磨擦，磁带毁损得很快，且在多个磁带间选择播放时，机械的可动部分也常出故障。即使有这些问题，美乐特朗能在演出时，播放预录好的自然声音能力，还是激起了广泛的兴趣。

几年之后，数字电子技术的兴起，使得录制并以数字内存存放声音日渐可行。在 20 世纪 70 年代，内存仍十分昂贵，所以最早的采样器仅是简单的延迟装置，多在录音室内使用，将采样声音延迟数毫秒后与原本声音相混，以丰厚声音(详见第 10 章延迟效果的讨论)。当内存价格滑落后，便可能储存数秒的声音，之后在琴键式采样器上播放。Fairlight Computer Music Instrument(CMI)是第一个商用键盘采样器(1979 年，澳大利亚)。CMI 有 8 比特的精准度，售价超过 25 000 美金。借数字硬件价格滑落的优势，E-mu Emulator 在 1981 年推出(图 4.3)，更进一步降低了 8-比特单声道采样器的价格，售价约 9 000 美金，提供 128K 的采样内存。

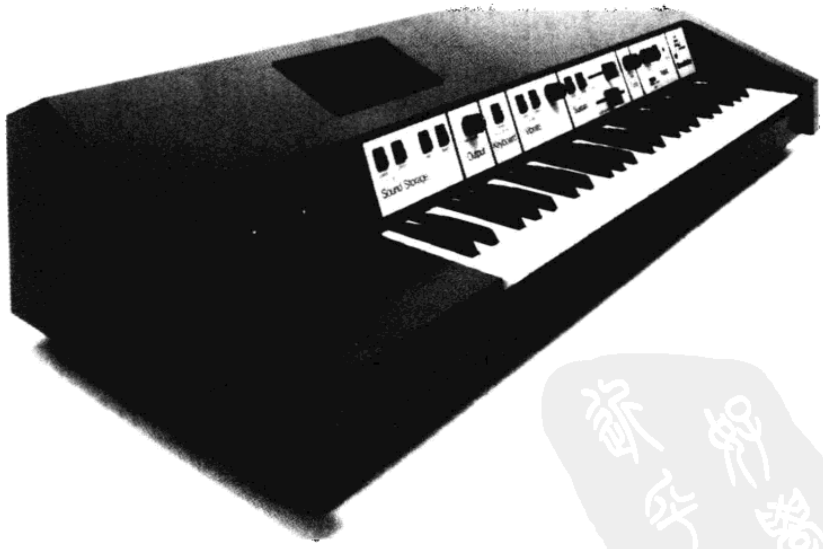


图 4.3 E-mu Emulator 采样键盘乐器(1981)。

为了创造商用采样器，必须探讨三个基本问题：循环、音高移位，以及数据缩减，我们在下面三个段落中将讨论之。

循环(Looping)

循环可将键盘上播放的采样声源延长，如果音乐家按着琴键，采样器会“无接缝的”读取此声音，直到放开琴键为止。这可由指定采样循环的开始点与结束点来实现。在音符的起音结束后，采样器会重复读取波表中的循环部分，直到放开琴键，接着播放此音符波表的完结部分。

出厂时的内建声源通常已先定义好循环点(prelooped)，但对于新的采样声音，就得由音乐家自行定义循环开始与结束点。制造无接缝而自然的传统乐器取样需要格外小心。循环应从音符起音后开始，在衰减之前结束(图 4.4)。

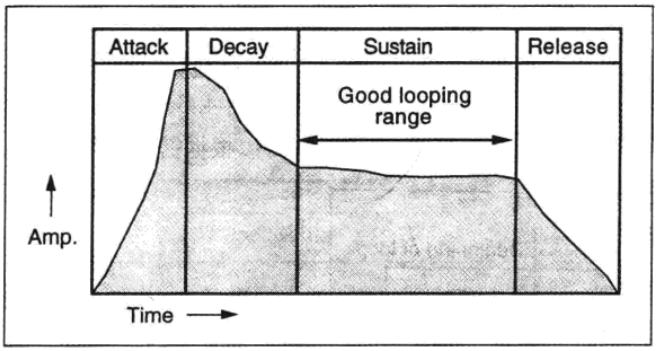


图 4.4　有着典型起衰延消(ADSR)振幅包络的声音，建立平顺循环的最佳区域是包络的持续部分。
Attack=起冲　Decay=衰减　Sustain=延留　Release=消去　Amp.=振幅　Time=时间
Good looping range=最佳循环范围

有些采样器提供自动寻找可能的循环点的功能。其中一个方法是对取样声音作音高侦测(pitch detection)(Massie 1986)。(见第 12 章，对于音高侦测的讨论。)音高侦测算法寻找波表中重复的样式，也就是基本音高周期。音高周期是波形一个周期的长度(图 4.5)。一旦估计好音高后，采样器给定一对循环点，符合波表内音高周期的某个倍数。此种循环运算法能够建立平顺的循环，并保持一定音高。然而若循环部分太短，得到的结果可能会像固定波形合成一样乏味，例如，一个循环若只包含一个或两个小提琴音符的音高周期，就会使弓弦演奏中的时变品质消失，产生很人工化的声音，而失去原本声音的独特性。

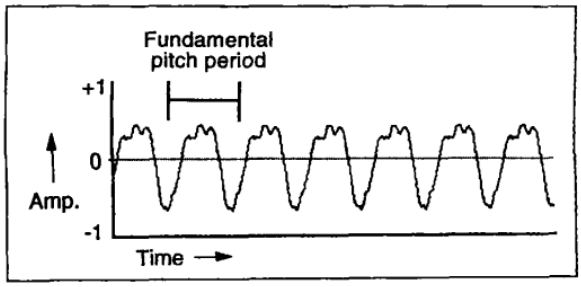


图 4.5 基本音高周期，相当于周期波的一个循环。此例是中音萨克斯的声音。
Fundamental pitch period=基本音高周期 Amp.=振幅 Time=时间

一个循环的开始点与结束点可以在两者共同的取样点上直接叠接(spliced)，或用交互淡出(crossfaded)方式连接。叠接是从一个声音直接切到另一个声音。波形叠接会在切点上造成噼啪声，除非循环的开始点与结束点恰好相配。交互淡出表示循环的末端会逐渐渐出，而循环前端再度缓慢渐入；交互淡出的程序会在按住琴键时一遍一遍重复(图 4.6)。典型的交互淡出时间约从 1 毫秒到 100 毫秒，但交互淡出的时间可以根据需要延长。

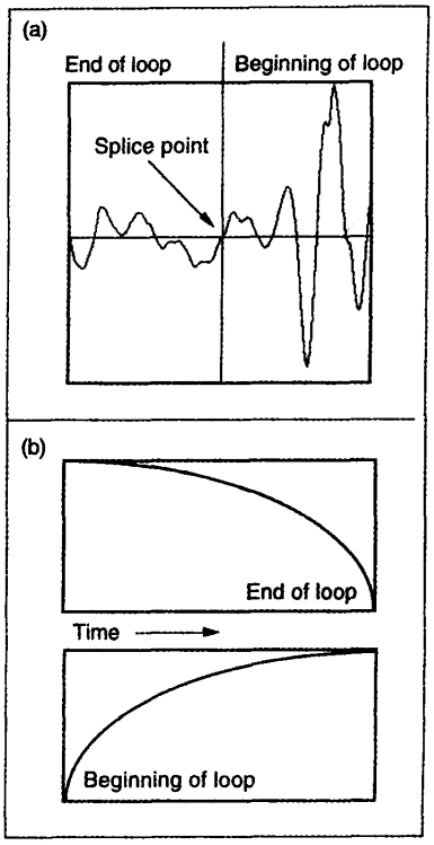


图 4.6 叠接与交互淡出的循环(a)垂直叠接一个波形的两个部分，交会在两者共同的取样点 0 点上。循环的末端与同个波表循环的起始端相接。(b)交互淡出的循环，可以视作将循环末端渐出，与渐入的循环始端重叠。
End of loop=循环末端
Beginning of loop=循环起始
Splice point=叠接点
Time=时间

由于揉音或其他信号变化，使得以上的技术都无法制造平顺循环时，必须使用更复杂的方式，如双向循环(*bidirectional looping*)。双向循环来回更换读取的方向(图 4.7a)。正向与反向的循环可以用分层的方式交互重叠，以掩盖任一方向的不连续处(图 4.7b)。更精细循环技术是根据频谱分析来实现的。比方说，我们可以分析一个声音，将循环内每个频率成分的相位随机化，再重新合成(Colllins 1993)。

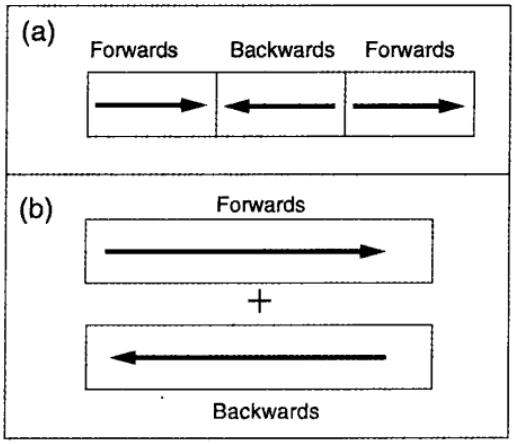


图 4.7　使变化平顺的循环方式。(a)双向循环的三个周期；(b)将分层的正向与反向循环相叠。
Forwards=前进　Backwards=后退

音高移位(Pitch Shifting)

在廉价采样器上，可能无法储存原始乐器演奏的每个音高。这些采样器在每三到四个半音间储存一个声音，从邻近的音高移位后得到中间的音高。如果你自己录下一个声音，存到采样器内存内，再按下不同琴键播放之，采样器也运用了相同的音高移位技巧。简单的音高移位的副作用是，由于所按琴键不同，声音延长的时间长短不一。第 10 章将叙述保持原始声音时长的音高移位方法。现在我们仍先回到简单的音高移位。

有两种音高移位的方法。

方法 1：改变数/模转换器(DAC)输出的时钟频率(clock frequency)，从而改变播放的取样率。这将使音高变高或变低，并改变时长。

方法 2：取样率变换(在数字领域下对信号重新取样)使采样器内部的音高改变，并允许以相同的取样率播放所有音高。

有些采样器使用第一种方法，有些则用第二种方法。两种方法都称为时域(time-domain)技术，因为他们都直接操纵时域上的波形。这与第 10 章频域(frequency-domain)音高变移技术不同。接着我们来比较这两种时域方法。

由于方法 1 改变播放取样率，所以在琴键上每个音高都需要独立的数/模转换器，可以同时播放声音（通常要多达 10 个数/模转换器）。每个数/模转换器必须允许可变时钟频率，并有与之相接的频率可变之平滑滤波器（smoothing filter）。为了要在整个范围间完整移调，数/模转换器与其滤波器必须有相当大的运作空间。比方说，如果一个声音的音高是 250Hz，以 44.1kHz 取样，要向上移六个八度到 16kHz 时，此数/模转换器的时钟频率必须也向上移六个八度，高达2.82MHz。

因为这些空间需要的缘故，要么只能使用较高价的组件，要么在系统声音品质上有所妥协（较典型的做法）。比方说，一个使用这种音高移位方式的采样器在最高取样率 41.67kHz 所录得的声音，仅允许音高移位一个半音（时钟频率改变小于 6%）。在此例中数/模转换器及它的滤波器不需要在高于44.1kHz 的取样率下工作。其他的采样器则不允许在任何频率上进行任何向上方的音高移位。

音高移位的第二种方法，是进行取样率转换。取样率转换事实上就是在数字领域内对信号重新取样。这与第 3 章所用到的查表合成变换音高的技术相同。输出数/模转换器（DAC）的取样频率保持固定。若加快一个声音，升高这个声音的音高，其方法是将其以较低的取样率重新取样来获得。这与定时摄影相类似，在定时摄影中，将每张摄影构图呈象频率放慢，从而获得播放时速度加快的效果。在数字音频系统中，有些取样点在重新取样时会被跳过。被跳过的取样点数目与希望音高移位的幅度成正比（如查表合成）。跳过取样点的重新取样程序称为抽取（decimation）（图 4.8a）。通过抽取技术的重新取样也称为减采样（downsampling）。比方说，要将声音向上调高三个八度，那么信号就应被减缩取样，在播放时仅读取每三个取样点中的一个。

若要降低声音的音高，并将它变慢，声音必须用更高的取样率重新取样，将之拉长。这与慢动作相机的操作类似，在慢动作相机操作中，通过加快呈象频率，来获得在播放时减缓动作的效果。在数字音频系统中，会在原本的取样点中间，以插值方式（interpolation）安插新的中间取样点（图 4.8b）。插值技术的重新取样也称作增采样（upsampling）。

许多种重新取样率与音高移位间的关系一开始可能颇令人困惑，因为音高移位方法 1 与方法 2 是以不同方向达到同样的效果。方法 1 提高播放时的取样率，来提高音调；而方法 2 则是由利用抽取方法（或称减采样法）降低取样率，来提高音调，而播放的取样率则是保持不变的。

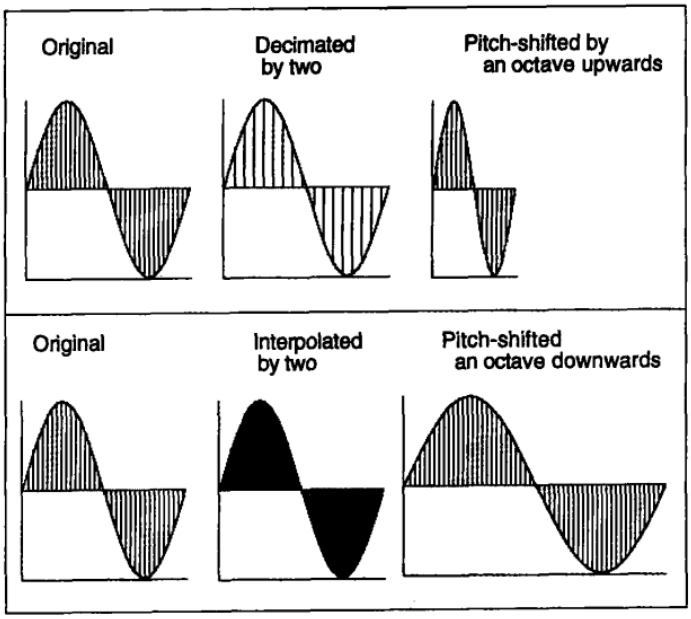


图 4.8　用恒定的采样播放频率方式获得取样率转换的音高移位。(上图)如果在播放时,每隔一个采样就略过一次的话,信号会被抽取,音高就会向上移动一个八度。(下图)如果在播放时用插值的方式使采样数量增加一倍,信号就会向下移动八度。
Original=原始信号　　Decimated by two=以 2 为单位的抽取
Pitch-shifted by an octave upwards=向上八度的音高移位
Interpolated by two=以 2 为单位的插值
Pitch-shifted an octave downwards=向下八度的音高位移

到目前为止,我们已经介绍过如何将音调升高或降低八度了,要将音高以任何一个整数比调整,就要采用插值法与抽取法的结合。(Schafer and Rabiner 1973a, Moorer 1977, Rabiner 1983, Lagadec 1983, Crochiere and Rabiner 1983, Hutchins 1986a, Duncan and Rossum 1988)。具体的公式是,若按照 N/M 的比率作音高移位,可先将之以 M 倍数插值后,再将之以 N 倍数抽取。举例说明,如要将音调降低 3/4(一个纯四度),我们先增取样,以四倍的比率插值,再以三倍比例作减采样和抽取。如若要将音调升高为 4/3,则要先以三倍插值,再以四倍抽取。

不改变音高的取样率转换(Sample-rate Conversion without Pitch Shifting)

许多数字音频录音机在标准取样率 48kHz 或 44.1kHz 的标准下运作。那么,我们怎样能用 48kHz 或 44.1kHz 的标准频率中的一个频率将录音重新取样,以达到在不改变音高的前提下以另一种频率回放呢? 在这种情况下,重新取样率与新的数/模转换器(DAC)输出取样率相同。

将信号在标准的取样率 44.1kHz 与 48kHz 中转换，而不改变音高，需要一个相当复杂的转换程序。首先先分解取样率比例：

$$\frac{48\ 000}{44\ 100}=\frac{2^5\times 5}{3\times 7^2}=(4/3\times 4/7\times 10/7)$$

这些比例可以作为六个阶段用因数 2,3,5,7 实现插值与抽取。

1. 以 4 倍插值，由 44 100—176 400Hz
2. 以 3 倍抽取，由 176 400—58 800Hz
3. 以 4 倍插值，由 58 800—235 200Hz
4. 以 7 倍抽取，由 235 200—33 600Hz
5. 以 10 倍插值，由 33 600—336 000Hz
6. 以 7 倍抽取，由 336 000—48 000Hz

这样，就能以在音高不会改变的情况下，实现 48kHz 的取样率播放声音信号了。

只要输出与输入的取样比例可用简单分数表示，转换的过程就相对简单。如果取样率间的比例值非整数，或者它们不断变化着，那么就需要更复杂的数学运算，在这里我们不会深入探究。（见 Crochiere and Rabiner 1983，Rabiner 1984，Lagadec 1984）。这是镶边效果（flanging effects）（见第 10 章）与音频擦蹭（模仿以手动来回控制磁带，重复播放以决定切点）。

重新取样的问题（Problems in Resampling）

重新取样的声音品质会受限于转换时使用硬件的精准度。当经过很多个中继转换程序时，会不可避免的加入一些噪音，而降低声音品质。混叠（Aliasing）（详见第 1 章）也可能是问题之一。这是由于重新取样时，就跟原始取样方法一样，可能会因混叠造成某些频谱上意外的非自然信号。这些中间的取样点可能会将波形两个连续点的中间部分过度平顺化，而经过抽取的信号往往会凹凸不平而不连续（图 4.9）。同时，所有频率向上移动，表示在播放时可能会造成混叠。这可以由在抽取后加上一个低通滤波器，将这问题减轻。滤波可将凹凸不平的波形变得平顺。

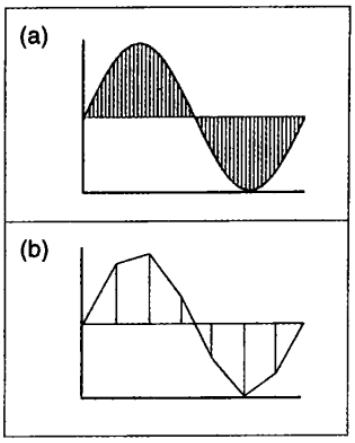


图 4.9 当有足够的抽取时，甚至连正弦波也会变成参差不齐的波形；(a)原始的弦波波形；(b)当因数是 8 时的抽取。

在插值计算时也必须使用滤波，因为简单的线性插值将造成混叠成分。在转换取样率时，与其使用更复杂的插值运算，通常较简单的做法是结合线性插值与滤波，将频率成分移位，同时减轻混叠。

采样器上的数据缩减与数据压缩 (Data Reduction and Data Compression in Samplers)

自 20 世纪 70 年代早期，半导体内存被发明以来，它的价格已大幅度降低。然而在今日，将大笔声音数据库完全存放在内存内仍然不切实际。为了要将数据库内的一部分子集合填入内存中，许多采样器仍需要数据缩减(data reduction)或数据压缩技术(data compression)，以减轻储存负担。这两种方法有很大的不同。数据缩减是将它认为不重要的数据丢弃，而数据压缩是将数据中的冗余部分，以编码方式将之以更有效率的方式放置在内存内。数据压缩可以重新组合成原始数据，而数据缩减则会造成失去部分原始数据。两种方法在音频文献中都归类于编码(coding 或 encoding)类配置中。

数据缩减(Data Reduction)

大部分采样器没有声音分析与“智能的”数据缩减功能。为了减少所需的储存声音采样内存量，制造商有时候采取较粗糙的手段，而影响声音品质。比方说，最浅而易见的数据缩减方法，是限制取样的分辨率或量化程度(quantization，详见第 1 章)。有些较便宜的采样器使用 12 比特或更少的量化值来表示一个采样点。减缩方法的一种变通手段是将采样存成低分辨率形式的浮点数编码方式(*floating-point*)连同几个表示原始声音振幅的比特(Pohlmann

1989a)。即便这明显改变了动态反应范围,但在低分辨率取样环境下的信/噪比依然很低。另一个方法是降低取样率。这将减少每个时间单位下所需要存放的声音数据,但其代价是音频带宽的降低。第三种方法则仅储存每三或四个音符之间的一个,再将这些取样以移调方式填入中间部分。这将造成频谱偏移的副作用,也不甚理想。如果声音中包含任何如震音或揉音之类的变化,这些变化的速度也会因音高移位的影响而被察觉。当内存价格降低后,就越来越没有理由使用这些会降低声音品质的方法了。

有另外一种更复杂的数据缩减方法,是从分析阶段开始的,将声音以数据缩减的方式与控制函数(*control functions*)一并储存,而后以近似方式重新将声音组合。这种分析后重新合成的方式有很多种。比方说,分析可以利用人耳的遮蔽效应(*masking phenomena*),把会被较大音量掩盖的声音成分舍弃。(对遮蔽效应的简介,见第 23 章;更详细的内容见 Buser and Imbert 1991)。本章后半部分,我们将介绍四种基于加法合成模式的实验性数据缩减方式。几种商用的数据缩减方式已内建在消费者的音频产品中。我们这里不是全面讨论以听觉感知模型为基础的数据缩减方式的地方。更不用说,任何数据缩减方式都会有数据丢失引起的声音品质的降低。若是在良好精密音频系统上全范围开发的音乐材料中,这种丢失会更明显。

数据压缩(Data Compression)

为了节省内存空间,有些系统使用数据压缩技术来限制取样串流所占用的空间。数据压缩技术通过对数据的冗余部分的减少来实现,而不造成声音品质上的降低。一种常用的压缩方式是运转周期编码(*run-length encoding*)。这种压缩方式的基本概念在于,不是存入所有的取样值,而是只储存与前面的取样值不相同的取样,还一并记录下后面相同的取样所重复的次数。(更多有关音频数据压缩的知识可参见 Moorer 1979b。)

采样库(Sample Libraries)

由于采样器是一种录音系统,采样的品质取决于录音技术的品质。制作高品质采样需要优秀的演奏者与好的乐器、高品质麦克风以及合适的录音环境。安排上述所有元素以录制声音数据库需要费很大工夫。因此,大部分采样器用户偏好以磁盘或光盘流通,由专业人士录制的数据库。

采样器的评估(An Assessment of Samplers)

尽管在采样技术上有大幅进展,采样器听起来仍然很"机器"味儿,与优秀的人类表演者演奏活生生的声音仍有很大差异。比方说,大部分打击乐手不会将取样鼓录下的声音误认为真实鼓手的声音。在打击乐现场的演出中,每个敲击都是不同的,最主要的差异,在于敲击过程中形成的音乐前后关系(musical context)情境。这不是说机械性的演出是毫无价值的。鼓机在商业上的成功证明了固定节奏以及不变的打击乐音响仍吸引相当多的听众。

任何情况下,把采样器的"自然"或"真实"作为不同品牌间的评断标准是可以理解的。众所皆知,某种乐器的声音在某个采样器上会比其他设备上来得更真实。

某些乐器,如管风琴,在大部分的采样器上都能有较真实的模拟。也就是说,它们都能产生管风琴或电子风琴的高品质声音。其他乐器的声音,如人声音色、小提琴、萨克斯、电吉他、锡塔尔琴等,以目前的采样技术就其本质上讲是较难以捕捉的。对于单一个声音可以录得很好,但当我们将这些声音组合成乐句、旋律以及和弦时,明显地会流失大量真实乐器演出的信息。

厂商内建的取样是在模仿一般歌手,或一般乐手吹奏的一般萨克斯,或是在普通音乐厅内的一般性管弦乐团。但大多数有一定知识素养的欣赏者能够分辨出两个声乐家、两个萨克斯演奏者,或两个交响乐团指挥之间的区别。几乎没有人会将MIDI音序器/采样器所重建的John Coltrane的萨克斯独奏与原本充满个性的演出版本相混淆。这也点出了现存采样器的基本局限。在某个程度上,除非在技术上能有大的跃进,并了解了声音结构与音乐表现之间的关系,否则,不可能再增加目前采样器的真实度。一种明显具革命性的采样器发展方向是分析/再合成技术(见13章),这种技术带有一定伸缩性,且具有与音乐声音前后关系场景相关的变化功能。

在人声音色、萨克斯、锡塔尔琴、吉他等这样富有表现力的乐器中,每个音符都是在其音乐前后关系场景(musical context)下产生的。在一个乐句内,一个音符是从另外一个音符(或从无声)开始,并延续到下个音符上(或无声)。除了这些音乐上的前后线索,还有一些过渡性的声音,如呼吸、舌触、按键音、手指滑过弦的擦弦音等。受音乐风格与品位的左右,决定了我们要在什么时候适当地加入如速度自由、滑音、颤音、渐强、渐弱以及其他一些细微变化的音乐前后相关效果(context-sensitive effects)。

这些问题可以分为两个部分:(1)如何模拟音符与音符之间过渡声音的细微结构?(2)我们如何解读(分析)总谱,依照特定音乐风格的规则来实现音乐前后场景敏感的演出?这些问题将在后面的两节中解说。

模仿音符间过渡(Modeling Note-to-note Transitions)

音符与音符之间的过渡是斯坦福(Standford)大学斯特朗(John Strawn)的博士学位研究课题(1985b)。他研究了九种非打击乐性管弦乐乐器的音符过渡。从研究图示反映出的时域及频域图结构看,显示出了在音的延续过程中音乐前后相关细微变化的特点。

在管乐器中,运舌是表现音符过渡的方法之一,它是用舌头将气流暂时阻断,就如演奏者发出 t 或 k 的声音。图 4.10 显示小号手以运舌(a)及无运舌(b)的方式吹奏的时域图。此两种方式对比非常明显。

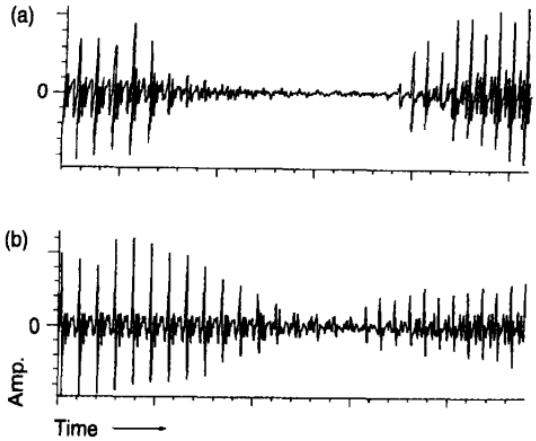


图 4.10 小号乐手吹奏上行大三度的音符过渡之时域图(a)运舌;(b)无运舌。总时长约为 120 毫秒。Amp.=振幅 Time=时间

图 4.11 绘出音符过渡时的频谱图。斯特朗的研究显示,有些过渡非常平顺,在音与音的音量上仅有小于 10dB 的差别。其他的过渡则在振幅和频谱上显示出很强的过渡预示,开始了第二个音的起冲。

模仿音与音之间过渡的细微结构,因其解决方案取决于可期望的科技进展,故似乎是个有迹可循的问题。这个问题可由增加采样器内存容量(储存两个音符间所有的过渡)、快速的信号处理,或是某些将这两者加以结合的方式来解决。比方说,双音素(diphone)方法将过渡数据储存为可延展或压缩的形式(Rodet,Depalle and Poirot 1988)。哈罗威(Halloway)与哈肯(Haken1992)则是在追踪相位声码器(tracking phase vocoder)上,以重叠音轨模仿音符过渡(见第 13 章)。

如果我们能够像音乐家在琴键上演奏般自动计算出过渡部分,乐器就必须要能迅速地决定其音乐前后内容。(第 15 章将讨论机器演绎乐谱的相关问题。)

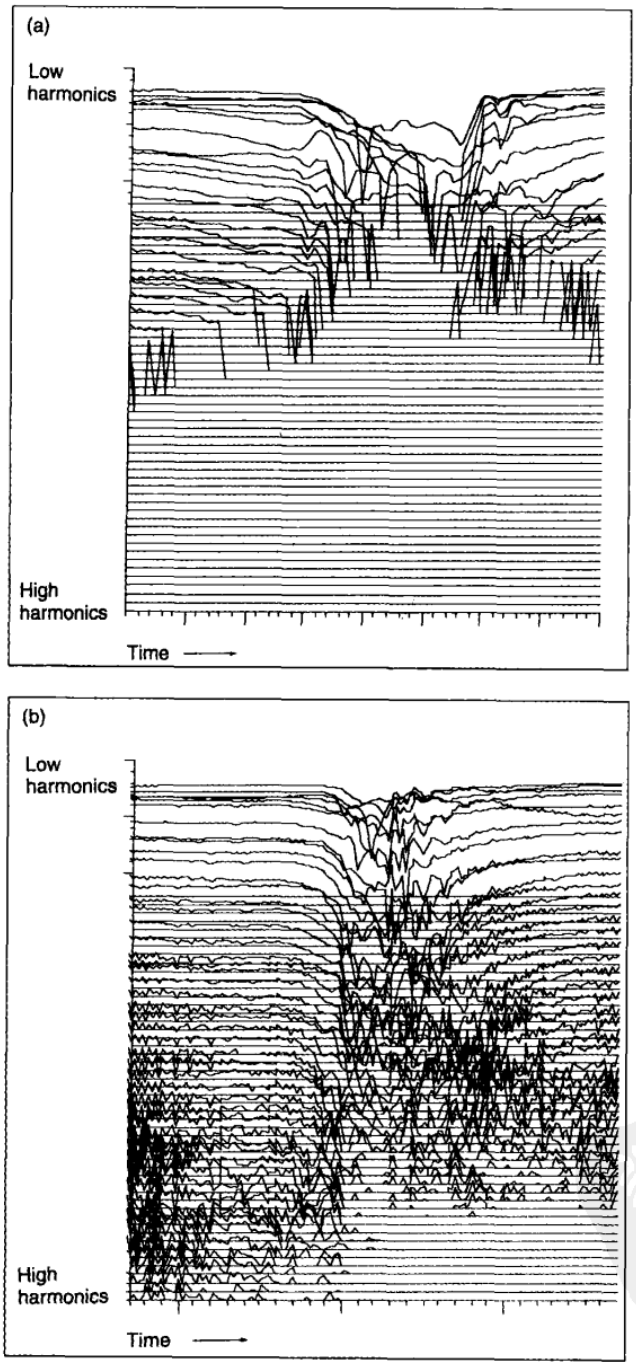


图 4.11　图 4.10 过渡部分的频谱图。此图显示 300 毫秒内的 50 个谐波成分，在后部带有较低频的谐波。(a)运舌；(b)无运舌。注意，在(a)图中央的“破口”，在无运舌时是怎样被填满(更连续)的。(由 John Strawn 提供。)

Low harmonics=低频谐波　　High harmonics=高频谐波　　Time=时间

加法合成(Additive Synthesis)

加法合成是种将基本波形相加后,创造出更复杂波形的声音合成技术。加法合成是最古老,也是最被深入研究的一种合成技术。此节将由说明加法合成的历史开始,接着再解释它的固定波形与时变表现。随后一节将着手讨论分析/再合成的方法,也就是将声音分析后,再以加法合成方式重新合成声音。

加法合成:背景(Additive Synthesis:Background)

加法合成的观念已有几个世纪的历史了,最先是以多重通风音栓(*register-stops*)的方式应用在管风琴上。拉动管风琴的通风音栓,空气可以通过不同的风琴管组。按下风琴琴键,令空气通过风琴管后产生声音。以不同比例调整通风音栓,我们可以将每个由琴键控制的风琴管之声音相加。据某位学者说:"中世纪时代特别偏好'混合',在这种混合中,每个音符都伴有数个在其上的五度和八度音。"(Geiringer 1945)。这种频率"混合"的观念是加法合成的基础。

加法合成自电子音乐初期即开始应用(Cahill 1897, Douglas 1968, *die Reihe* 1955, Stockhausen 1964)。1906 年问世的庞大电簧琴(Telharmonium)合成器将几十个电子音频发生器的声音相加,以得到叠加之复音(图 4.12)。

著名的 Hammond 风琴是纯加法合成乐器(图 4.13),它是以缩小版的 Telharmonium 的旋转音频发生器为基础。加法合成的能力在于,理论上,通过基本波形元素相加,可以极其近似地实现任何复杂的波形。另外,也有一种方法,先是对声音如小提琴进行分析,然后用时变的不同频率、相位及振幅的正弦波的组合将其重新合成。然而,由于分析分辨率上的基本限制,此重建版本永远不会是原始信号每个取样点的复制(见第 13 章)。

任何将数个基本波形相加,以创造新波形的方法,都可以算是加法合成。比方说,第 5 章讨论的粒式合成的某些形式也可以称作加法合成技术。不过,我们还是将粒式合成技术与加法合成技术区分开来,以阐明正弦波加法合成的传统手法与那些粒式合成手法两者之间的区别。

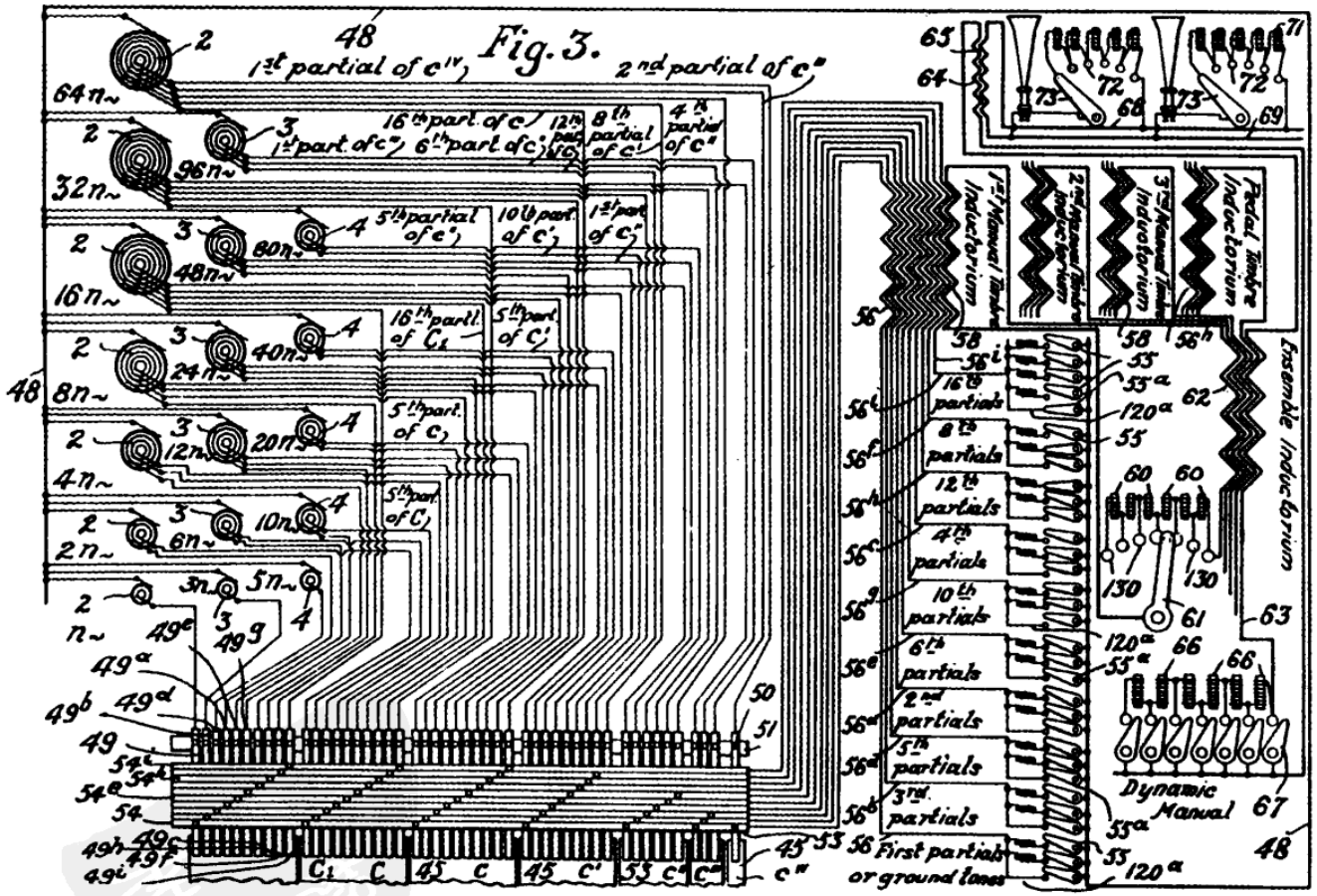


图 4.12　电传簧风琴中一个复音的加法合成。由生音振荡器产生的正弦波谐波被送入母线(54)。按下一个琴键(此例中为 C),该琴键将每个谐波连接到多线圈变换器(56"inductorium"),在这里,它们被混合在一起。每个谐波会被串联在线圈上的感应器减弱到适当的强度(56a, b, etc)。分线开关感应器(60)调整混音转换器输出的强度,传输线中听者一端的靠近扬声器的感应器(72,73)也做类似的工作。(Cahill 专利图,由 Johnson et at. 复制,1970。)

图 4.13 哈蒙德(Hammond)B3 型电风琴，是以电子机械式的转速脉冲轮(tone-wheels)为基础的加法合成乐器。多个谐波的不同混合可以由琴键上的拉杆调整。(由 Institute of Organology, Kunitachi College of Music, Tokyo 提供。)

固定波形加法合成(Fixed-waveform Additive Synthesis)

有些软件包与合成器让音乐家以谐波叠加方式(harmonic addition)来建立波形。为了要建立给定频谱的波形，使用者要调整已给定基频谐波的相对强度。("谐波"是指基频的整数倍率，最早由绍弗(Sauver，1653—1716)在 1701 年使用。比方说，400Hz 是 200Hz 的第二谐波，因为 200 的两倍是 400。谐波可以用条线图(bar graph)或直方图(histogram)表示，每条的高度代表该谐波之相对强度。图 4.14 显示谐波之频谱与相对应的波形。

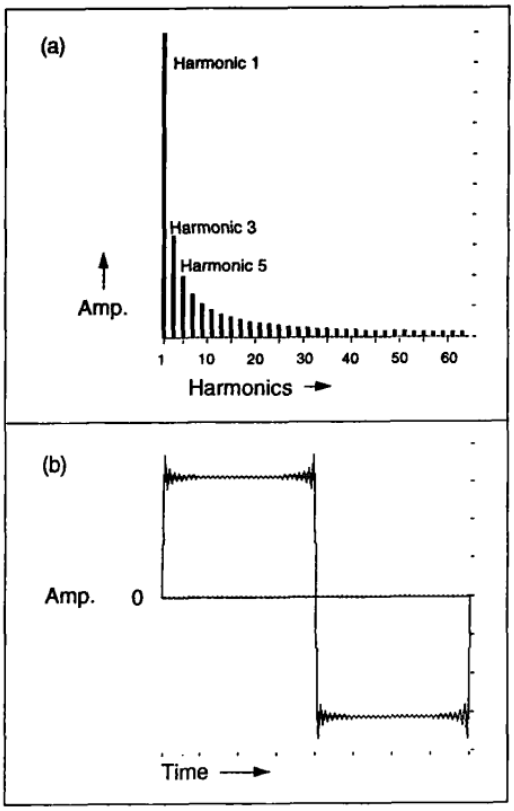


图 4.14　谐波叠加的波形合成。(a)直方图显示在线性比例下，谐波间的相对强度。在此例中直方图只有在奇数成分上有能量。第三谐波上的强度为基频的三分之一，第五谐波上的振幅为基频的五分之一，依此类推；(b)使用(a)直方图通过谐波叠加手段合成的近似方波。
Amp.=振幅　Time=时间

一旦调整好需要的频谱，软件便能计算出该波形的频谱，以数字振荡器播放出来。此频谱的样本会依振荡器音高而做出相对的调整。图 4.5 显示以加法合成做出近似方波之连续等级。

相位因素(The Phase Factor)

相位是个魔术师。依其所在的情况，它在加法合成中可能是也可能不是一个重要的因素。比方说，如果改变了固定波形的频率成分中的起始相位，并重新合成声音，听者可能感觉不到变化。然而这改变可能对波形在视觉上有着截然不同的影响，如图 4.16 所示。

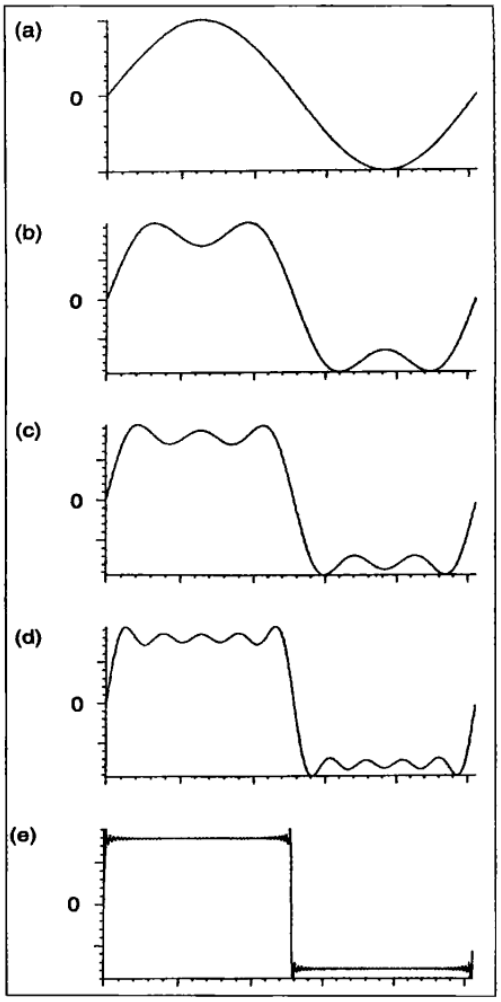


图 4.15 一系列时域波形不同阶段的谐波叠加。(a)仅有基频;(b)第 2(基频)与第 3 谐波之和;(c)奇数逐级相加到第 5 谐波之总和;(d)奇数逐级相加到第 9 谐波之总和;(e)由奇数逐级相加到第 101 谐波之总和形成的近似方波。

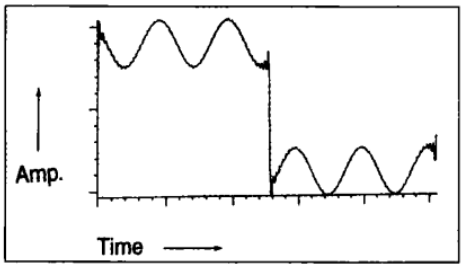


图 4.16 加法合成中相位的效应。此波形是由图 4.15e 中相同的正弦波成分叠加所形成的结果,只有一点不同,就是在第 5 谐波上的起始相位为 90 度,而不是0 度。Amp.=振幅 Time=时间

相位关系在短暂而明亮的声音如起音、颗粒或是过渡部分较容易察觉。耳朵对于复杂的声音内某些频率成分的相位随时间改变也极为敏感。

我们在后面一节中的声音分析与再合成中将会看到,适当的相位数据调整将有助于按正确顺序来重组那些短暂无常的成分,所以对重建一个分析后的声

音极为重要。

分音叠加(Addition of Partials)

我们可进一步将谐波叠加一般化为分音叠加。在声学中,分音指的是频谱上的任何频率成分(Benade 1990)。分音可能是基频 f 的谐波(整数倍)成分,也可能不是。图 4.17a 显示含有四个分音的频谱:两个谐波,两个非谐波。非和谐分音不是基频的整数倍率。图 4.17b 则是上四个分音之和的波形。

分音叠加的限制在于,它只能制造出有趣的固定波形声音。因为固定波形合成的频谱在一个音符的时间内保持不变的,所以分音叠加永远无法精确地重现原始乐器的声音。它只能接近达到这个乐器的稳态部分(steady-state)。研究表明,音的起冲部分的频率成分变化是以毫秒为单位,在辨认原始乐器时,音的起冲部分比这个音的稳态部分更为有用。在任何情况下,时变音色总比固定音色对人耳更有刺激作用(Grey 1975)。

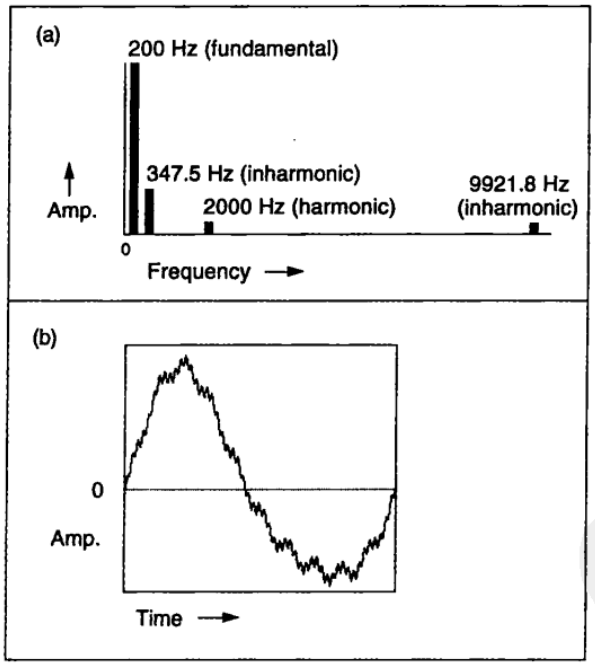


图 4.17　四个成分的分音叠加。每个成分的振幅百分比分别是 73,18,5,4。(a)频域图;(b)时域图。
Amp.=振幅　Time=时间　Fundamental=基频　Inharmonic=非谐波　Harmonic=谐波
Frequency=频率

时变加法合成(Time-varying Additive Synthesis)

将正弦波混合的声音随着时间改变,我们可以得到非常有趣的音色及更逼真的乐器音。在图 4.18 中的小号音符中,需要 12 个正弦波才能重现此事件的起冲部分。过了 300 毫秒之后,这个音只需要三或四个正弦波即可。

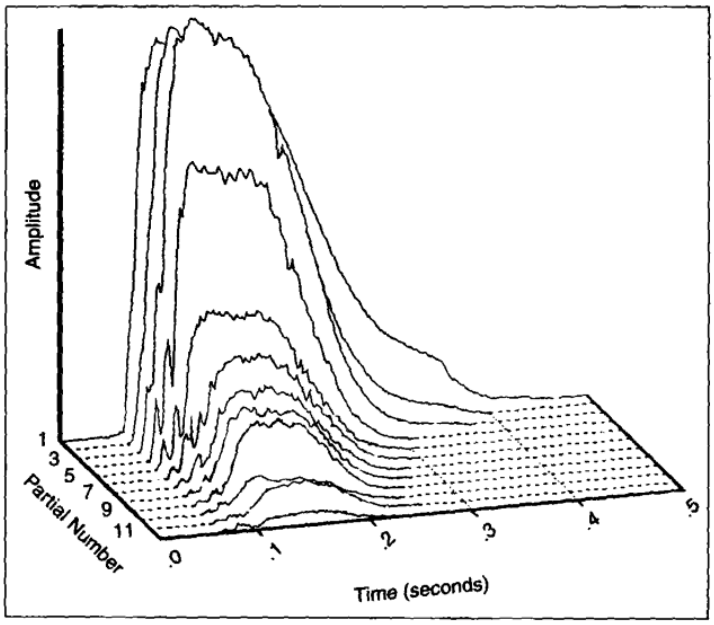


图 4.18 有 12 个分音的小号音色的时变频谱图。最高的分音在图的前景。时间从左至右。注意,(在后面的)基频的振幅不是最高的,但是最长的。
Amplitude=振幅 Time=时间 Partial Number=分音数

我们可以用很多图像方式来观察分音叠加的过程。图 4.19a 显示 20 世纪 50 年代实现的斯托克豪森(Stockhausen 1964)在模拟环境下的叠加合成。此图中有数个振荡器硬件模块,每个都带有一个手动控制的频率旋钮。振荡器的输出被送入混音器。作曲家在实时中调整振荡器间的平衡,改变时变频谱。在此设定下,手动控制是唯一选择。要精确实现时变混音,需要数人同时控制。(Morawska-Büngler 1988)。

图 4.19b 是数字加法合成。音频振荡器以半圆符号表示,有一对输入,一个是振幅输入,另一个则是频率输入。要产生时变频谱,每个振荡器的频率及振幅输入都不是常数,而是在整个声音事件时程内读取的随时改变的包络函数。正弦波音频振荡器被输入到将信号相加的模块,然后,此模块再将相加的结果送入数/模转换器(DAC)转为声音。

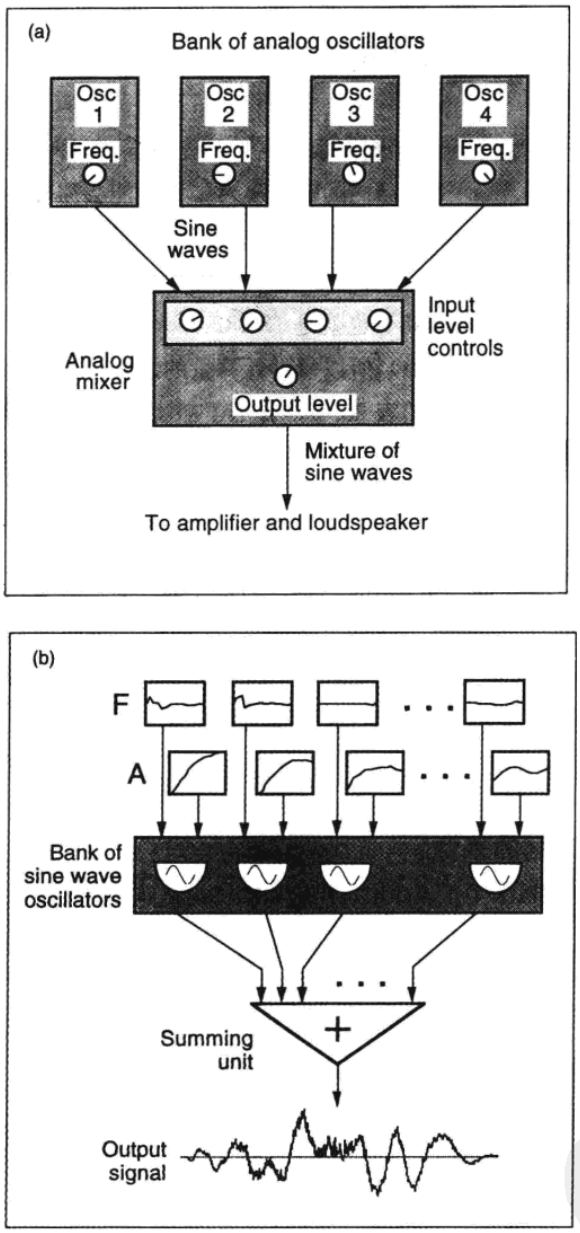


图 4.19　两个加法合成图示。(a)是在模拟状态下多个振荡器输入一个混合器中;(b)是数字加法合成。是分别带有不同频率(F)和不同振幅(A)的时变加法合成。图 3.10 显示了加法合成设备图示的更多细节。

Bank of analog oscillators=模拟振荡器组　Osc1=振荡器　Freq. =频率　Sine waves=正弦波
Analog mixer=模拟混频器　Input level controls=输入电平控制　Output level=输出电平
Mixture of sine waves=正弦波的混合　To amplifier and loudspeaker=到放大器和扬声器
Bank of sine wave oscillators=正弦波振荡器组　Summing unit=求和单元　Output signal=输出信号

加法合成的要求(Demands of Additive Synthesis)

时变加法合成对数字音乐系统要求很高。首先,它需要许多振荡器。如果我们在音乐上合理推测,一个作品内的每个声音事件可能会需要 24 个分音(每个都由一个正弦波振荡器产生),而至少同时能播放 16 个声音事件,所以我们在任何时间上至少要有 384 个振荡器。如果系统在 48kHz 的取样率下运行,每秒就必须产生 48 000 乘以 384,也就是 18 432 000 个取样点。因为每个取样点需要 768 个运算(相乘—相加),所以每秒计算量的总和超过 14 亿个运算,这还不包含查表动作。第 20 章的表 20.1 推算出了加法合成每个取样点的需求。如此的计算需求虽然令人畏惧,但还未超出今日硬件的极限。比方说,一个特别用来处理加法合成的合成器,能具有在实时中提供使用数千个正弦波的能力(Jansen 1991)。

虽然计算能力是加法合成的唯一要求,但此方法对于控制信号的需要也极难以满足。如果一个作品有 10 000 个事件(如典型的交响乐谱),每个都需要 24 个分音,所以,总共需要 240 000 个频率包络与 240 000 个振幅包络。就算是在多个事件中重复使用相同包络,这些控制数据要从哪里来呢? 这是下节所要讨论的主题。

加法合成中的控制数据的来源
(Sources of Control Data for Additive Synthesis)

任何数字合成技术有效的使用(包括加法合成)都取决于是否给予合成设备良好的控制数据。要创造生动而又具有丰富内在变化的声音,我们必须通过控制数据来驾驭合成器;所以,控制数据通常也可以称作合成器的驱动函数(*driving function*)。控制数据可以由下列数种来源获得:

1. 来自另外一个领域并对应到合成参数上。比方说,有些作曲家以山峦起伏或城市天际线的曲线来作为控制函数,这是早期道奇(Charles Dodge)的计算机音乐作品 *Earth's Magnetic Field*(1970)的创作方法,还有完全从地理的、随机方式的或其他数学与物理模式中获取数据来驾驭作品。

2. 由作曲家给定音乐细微结构的限制后,让作曲程序自动产生。一个例子是约翰·乔宁(John Chowning)的作品 *Stria*(1977),作品采用非和谐频谱的加法合成实现。

3. 由体现了高水准音乐观念的交互式作曲系统产生,如"乐句"(phrases,Rodet and Cointe 1984 的 Formes 语言),"趋向掩饰"(tendency masks,在 Tru-

ax 1977 的 POD 系统),“声音对象”(sound objects,如 Buxton et al. 1978 的 SSSP 系统),或者“粒云”(clouds,Roads 1978c, 1991 的异步粒式合成)等观念,对应到合成参数上。

4. 由作曲家手动输入,把前面叙述过的种种方法,或是作曲家的直觉、理论或心理听觉上的知识结合在一起。一个使用此方法的范例是 Jean-Claude Risset 的作品 *Inharmonique*(1970)。

5. 由分析子系统提供数据控制资源,该子系统可分析自然声音,输出重新合成所需要的控制数据,也可编辑这些控制数据,以产生原始声音的变形。Trevor Wishart(1988)使用声音分析,作为改变人声的中间程序,以制作他的作品 *Vox-5*(详见 Murail 1991)。

由于方法 1 到方法 4 是基于作曲审美原则上的,故将不在此章深入探讨。第 5 个方法需要声音分析的子系统,也就是下一节所要介绍的。

附加分析/再合成(Additive Analysis/ Resynthesis)

附加分析/再合成包含不同的技术,但都要有三个步骤(图 4.20):

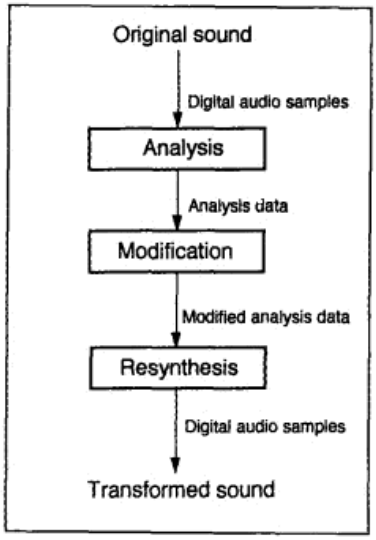


图 4.20　附加分析/再合成的略图。修改阶段可能牵涉到手动编辑分析数据,或者由交互合成(cross-synthesis)来改变,即一个声音的分析数据与另一个声音的分析数据相乘。
Original sound=原声
Digital audio samples=数字音频样本
Analysis=分析
Analysis data=分析数据
Modification=修正
Modified analysis data=修正分析数据
Resynthesis=再合成
Digital audio samples=数字音频样本
Transformed sound =转化声

1. 将录制的声音加以分析。
2. 音乐家修改分析数据。
3. 修改过的数据使用在对改变声音的再合成上。

此附加分析/再合成的观念并不单只用于加法合成上,也可用在减法再合成(subtractive resynthesis)(见第 5 章)、加法与减法再合成的结合(Serra

1989, Serra and Smith 1990)或其他种种方法上(见第 13 章)。

弗莱彻(H. Fletcher,即著名的 Fletcher-Munson 振幅曲线)以及他的同事(Fletcher, Blackham, and Stratton 1962; Fletcher, Blackham, and Christensen 1963)做了最早期的附加分析/再合成实验。他们使用纯模拟器材。当数字叠加方法开始应用在再合成上时,整个系统就如图 4.21 所示。不断地对输入信号的细小片段作分析,这种将输入信号分割的程序称作窗口化(*windowing*)(在第 13 章及附录中讨论)。我们可以将每个经过窗口化的片段,视作通过了一组窄频带通滤波器,在这里,每个滤波器调整在特定的中央频率上。实际操作上,常用快速傅里叶变换(FFT)替代滤波器组,并在此程序中执行相同工作,也就是测量每个频带上的能量(这也将在第 13 章及附录中讨论)。

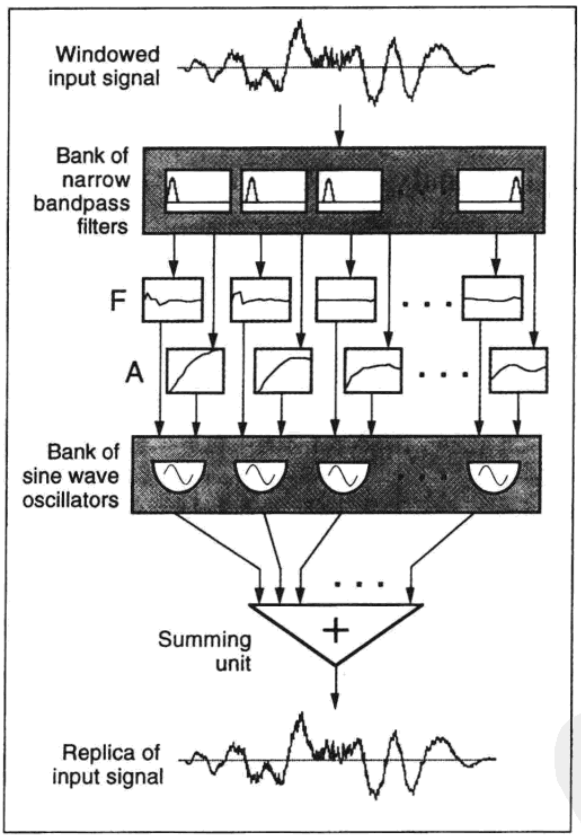


图 4.21 附加分析/合成。经过窗口化的输入信号经由一组滤波器分析,成为一套频率(F)与振幅(A)包络或驱动一组振荡器的控制函数。如果分析数据没有改变,输出信号也应该很接近于输入信号。

Windowed input signal=窗取输入信号 Bank of narrow band pass filters=窄带通滤波器组
Bank of sine wave oscillators=正弦波振荡器组 Summing unit=求和单元
Replica of input signal=输入信号的仿形

来自每个滤波器的信号振幅都经过了测量,这种时变值成为该频带下的振

幅控制函数(amplitude control function)。同时,系统依照其少量频率偏移(frequency deviations),通过检视相邻滤波器(在 FFT 中的 analysis bins)的输出,来计算控制函数。

频率与振幅控制函数驱动再合成阶段中的一组振荡器。换句话说,我们使用分析现存声音所得的数据,来创造一套控制函数,以便满足对那些用正弦波加法合成出来的声音做再合成的需要。如果输入声音可以被模拟,转化成一系列正弦波之和,那么,这个由振荡器产生的相加在一起的正弦波信号应与原始信号非常相近。

当然,从音乐的角度讲,直接的声音分析/再合成并没有太多意思,为了要创造有趣的音乐效果,我们必须改变分析后产生的数据,也就是下节探讨的主题。

附加分析/再合成的音乐应用
(Musical Applications of Additive Analysis/Resynthesis)

在分析步骤完成后,音乐家可以编辑控制函数,来创造出原始输入信号的变化。可以用此技巧得到许多不同效果,如表 4.1 所示。20 世纪 80 年代的三部作品是分析数据作曲操作的优秀范例:如 J. 哈维(Jonathan Harvey)的《死去的哭声,活着的叫喊》(*Mortuos Plango, Vivos Voco*,1981),T. 穆雷尔(Tristan Murail)的《裂变》(*Désintegrations*)(1983, Salabert Trajectoires),以及 W. 卡洛斯(Wendy Carlos)的《数字月面》(*Digital Moonscapes*)(1985, CBS/Sony)。

表 4.1　使用附加分析/再合成之音乐转变

音乐效果	技术
已录制声音的变化	通过编辑或乘上任意的函数,以改变选定的频率或振幅包络。
频谱定标 (无时间标度)	将所有分音频率(可能除了基频以外)乘上一个系数 n 或任意函数。因为乘法将不会保留共振峰结构,就会使人声及器乐声失去其原有的独特音质。
频谱位移 (无时间上标度)	在所有分音上(可能除了基频以外)加上一个系数 n 或任意函数。当此值很小时,共振结构会保留下来。

续表

音乐效果	技术
频谱倒置	在再合成操作前，颠倒频率成分的顺序，所以第一分音的振幅被指派到最后一个分音上，反之亦然；接下来的第二分音与倒数第二分音等其他成分的振幅交换依此类推。
混合音色	用选定的另一个声音的包络替换当前声音的某些包络。
无音高移位的时间扩展和压缩	延长频率及振幅包络的时值，或者回放时改变跃幅(hop size)的间距(见第 13 章)
将一个打击乐音色，延长为一个拉长了的合成音片段	延迟每个分音的爆发时间并将这些分音的包络平顺化。
由一个器乐声到另一个器乐声的音色插值	在两个乐音包络间的时值中插值。
使合成音变异	在任意合成音的包络内插值。
增强已录制声的谐振区	加大选定频率分音的振幅。
交叉合成	方法 1：用一个声音的分音振幅包络以相同比例关系改变另一个声音的分音振幅包络〔见第 10 章快速卷积(fast convolution)。〕 方法 2：将一个声音的振幅包络套用在另一个声音的频率(或相位)函数上。 方法 3：将一个声音的噪音残余，套用于另一个声音的准谐波部分(详见频谱建模合成与第 13 章中的梳状小波变换)。

在哈维的作品中，作曲家分析了一个大钟的声音。在再合成中，作曲家将对于每个正弦波成分替换为适当频率上的男孩取样声音。此人声取样后接续经分析过的谐和的钟声频率与钟声振幅控制函数，得到奇异的男孩/钟声合唱效果。在穆雷尔的作品里，作曲家分析传统乐器的声音，创造出对传统声音的合成的补充成分，不仅与原始乐器声音无缝地接合，而且在声音停止时还起到了戏剧性的衬托的作用。作品 *Désintegration* 是频谱作曲法的典型范例，作品中的谐波结构是建立在器乐声音分析的基础上的(Murail 1991)。在《数字月面》(*Digital Moonscapes*)中，卡洛斯使用分析数据作为灵感，以创造一个包含类似打击乐、弦乐及木管铜管的音色的特别的合成乐团，并套用传统管弦乐团的风格。

下节将简单讨论有加法再合成的现有的声音分析技术，并着重在数据缩减的问题上。这可视为是第 13 章与附录中更深入来探讨的前奏。

加法合成中的声音分析方法
(Methods of Sound Analysis for Additive Synthesis)

许多频谱分析方式都是傅里叶分析频率成分基本技术之变化，其中包括音高同步（pitch-synchronous）分析（Risset and Mathews 1969）、相位声码器（phase vocoder）（Dolson 1983，1986，1989b）、以及常数 Q（*constant-Q*）分析（Petersen 1980，Schwede 1983，Stautner 1983），实用的傅里叶分析是短时傅里叶变换（short-time Fourier transform，STFT）。这个方法可以被认为是通过精选（由窗口函数 window function 显示出来的）一连串短时、重叠的声音片段，并将这些声音片段通过一组滤波器的滤波来分析取样声音。每个滤波器的输出都经过测量，并在那个特定频率上标出频谱的振幅与相位。一连串这样的短时分析接续成一个时变频谱（就像电影中许多帧的画面）。STFT 的核心是 FFT，即傅里叶分析的高效率运算法（Cooley and Tukey1965，Singleton 1967；Moore 1978a，1978b，Rabiner and Gold 1975）。

在这里值得一提的是相位声码器（Phase vocoder，PV，Flanagan Golden 1966，Portnoff 1978，Holtzman 1980，Moorer 1978，Dolson 1983，Gordon and Strawn 1985，Strawn 1985b），它是在许多音乐软件包中常见的声音分析/再合成方法。相位声码器（PV）将输入信号取样后，转为时变频谱共振峰。特别是，它能产生一组时变频率与振幅曲线。许多有趣的声音变形可以通过编辑、再合成相位声码器的资料来获得。比方说，相位声码器可以应用在不造成音高改变前提下的时间压缩（time compression）或时间延展（time expansion）。这种技术效果，可以使声音变长或缩短，而不影响音高或音色。（详见第 10 章有关数种时间压缩/延展的方法的讨论。）

但与这些技术发明者（他们一直在寻找有效的编码技巧）的期望相反，声音的分析技术可能造成“信息爆炸”（Risset and Wessel 1982）。也就是说，分析数据（控制函数）可能比原始声音数据多占用内存很多倍。数据量部分取决于输入声音的复杂度，即需要多少个正弦波才能重新合成，也部分取决于分析程序中的内部数据表示方式。比如，使用追踪相位声码器时，一段占用 2MB 的小声音档可能产生数十个 MB 的分析数据。如此大的储存需求使建立分析声音的数据库十分困难，而且难以处理。这种情况使得我们必须减缩某些控制数据，如下节所述。

附加分析/再合成的数据缩减(Data Reduction in Analysis/Resynthesis)

数据缩减对于有效率的分析/再合成十分重要。数据缩减有两个步骤。首先分析出数据,也就是一组振幅与频率控制曲线,然后,再用算法,将原始数据转为更精简的表达方式。数据缩减的目的是,在不丧失重要的输入声音感知特征的前提下压缩数据。在计算机音乐中另外一个重要目的是将分析数据保留为一种可以由作曲家编辑的格式。此目的并不是简单的节省比特,而是人们希望使这些数据缩减材料变得易于操作。

有大量关于数字音频采样的数据缩减研究工作被记录在文献中,其中包括:Risset(1966), Freedman(1967), Beauchamp(1969, 1975), Grey(1975), Grey and Gordon (1978), Charbonneau (1981), Strawn (1980, 1985a, 1985b), Stautner(1983), Kleczkowski(1989), Serra (1989), Serra and Smith (1990), Holloway and Haken(1992), and Horner , Beauchamp , and Haken (1993)。由于实时演出对于音乐家来说变得非常重要,所以,分析/再合成研究其中的一个目的就是加快数据缩减的处理速度,并由这些数据做出实时合成。Sasaki and Smith(1980)和 Schindler(1984)的论文解释了高速数字合成处理减缩数据的硬件设计。

有许多探讨数据缩减方式的工程文献著述,这里我们仅浏览四种应用在计算机音乐上的方法:线段近似法(line-segments approximation)、主成分分析(principal components analysis)、频谱插值合成(spectral interpolation synthesis)以及频谱建模合成(spectral modeling synthesis)。〔详见克德伯格(Goldberg)1989 年有关基因算法的描述,近来才应用在减缩数据上。(Horner, Beauchamp, and Haken 1993)〕

线段近似法(Line-segment Approximation)

在振幅与频率控制函数上的线段近似法可以消除每个取样分析特定值的存储必要性。因为,该分析系统只储存两个一组的断点(break-point pairs),也就是时间(x 轴)与振幅(y 轴)的点,在这个点上,波形有明显的变化。线段近似法通过只储存最大的变化点来表现波形整体的轮廓。在再合成阶段,系统通常使用插值在两个一组的断点之间直线将两个点连接起来。

早期的线段近似法是以手动方式,用互动性图像编辑器来建立每个四到八个线段的函数(Grey 1975),数据缩减量约有百倍之多。这种手动编辑工作可以部分地转为自动执行,如斯特朗(Strawn1985a, 1985b)所展示的。图 4. 22a

显示小提琴音色的16个谐波的透视图，以25kHz取样，图4.22b绘出(a)用三个线段完成的一个(a)的近似值。

除了线段近似法的储存以外，比彻姆(Beauchamp)从一个声音的第一个谐波曲线开始，为该声音所有的谐波近似振幅曲线开发了一种启发式技术(heuristic)。在简单的周期性声音中，查邦尼乌(Charbonneau 1981)发现了更加极端的数据缩减技术，他将某声音的一个单包络的简单变化用在这个声音的所有的振幅函数中。(详见Kleczkowski 1989，Eaglestone and Oates 1990有关这些技术提议的简介。)

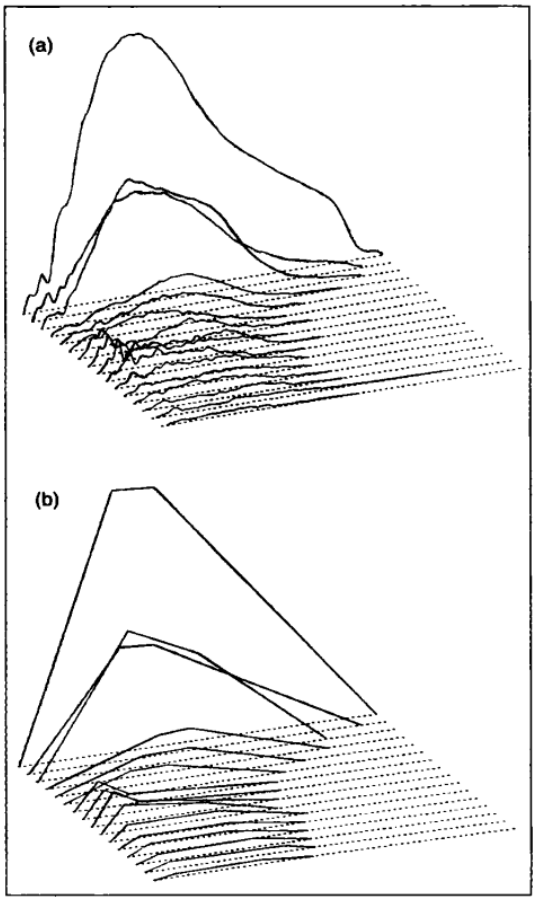


图4.22　加法合成中大幅度的分析数据缩减。垂直高度为振幅，频率由后往前走，时间则由左到右。(a)是一个小提琴音的原始的频率—时间—振幅曲线构图；(b)是(a)中显示的小提琴音的每个分音的三线段法近似的结果。

主要成分分析(Principal Components Analysis)

主成分分析(PCA)技术已被使用在数个分析/再合成系统上(Stautner 1983, Sandell and Martens 1992, Horner, Beauchamp, and Hakken1993)。主成分分析用数学上的协方差矩阵(covariance matrix)计算方式将波形拆解。得到的结果是一组基本波形(主要成分)和一组这些基本波形的额外系数。当成分依照额外系数相加在一起时,结果就会很近似于原始波形。

主成分分析(PCA)的好处在于它的数据缩减潜力。主成分分析将采样点间潜在的关系进行概括,所以只要最少的成分,就能得到最大可能的信号变量(variance)。决定主成分及其额外函数的方法。是以递归近似法来应用的,递归近似法尝试着使平方误差(squared numerical error,即原始值与预测值间的差)最小化。第一个主成分是以单一波形对整组数据。第二个主成分则是对应其余冗(residual,有时称为 residue),或原始值与第一次近似之间的差。第三个成分则是对应第二个主要成分的次余冗,以此类推。有关主成分分析(PCA)更详尽的描述可参见 Glaser and Ruchkin(1976)。

频谱插值合成(Spectral Interpolation Synthesis)

频谱插值合成(SIS)(Serra, Rubine and Dannenberg 1990)则是通过在分析频谱之间作插值以产生时变声音的实验性技术。频谱插值合成并非在时域的取样声音中间交互淡出(如第 5 章讨论的多重波表合成),而是由分析录下的声源开始,以加法合成在连续的频域频谱分析之间交互淡出。必须使用自动的数据缩减运算法,以将分析数据压缩为一个很小的两个连续声音和一组渐增函数的共同频谱路径。该共同路径描述了由一个频谱向另一个频谱的转变。此方法最主要的难处在于处理声音的起冲部分。

频谱建模合成(Spectral Modeling Synthesis)

频谱建模合成(SMS)(Serra 1989, Serra and Smith 1990)将分析数据缩减为可确定性成分(*deterministic* component,即原始声音的窄频成分)和随机成分(*stochastic* component)。可确定性成分是分析的数据缩减版本,它模拟频谱中最突出的频率。这些频率通过波峰侦测(*peak detection*)程序和波峰延续(*peak continuation*)程序,从每个分析的帧(frame)中独立出来,波峰延续是追踪连续帧间的波峰。频谱建模合成用正弦波再合成这些被跟随的频率。这与

13 章中追踪相位声码器的方法相同。

当频谱建模合成更进一步分析余冗部分，并分析可确定性成分与原始信号间的差，这就被称为信号的随机(stochastic)成分。随机成分以一连串的包络的形式，控制一组可以使白噪声通过的频率成型滤波器(frequency-shaping filter)。所以作曲家可分别处理可确定的(正弦波)包络与随机(经过滤波的噪音)成分，如果需要(图 4.23)，即便在经过转换后(如滤波后)，噪音成分仍可保持嘈杂。这与纯正弦波模型形成对比，在纯正弦波模型中，经过转换(如时间压缩/伸展)，噪音成分会成为有序的正弦波集合，而使得原本的噪音音质变得不自然。

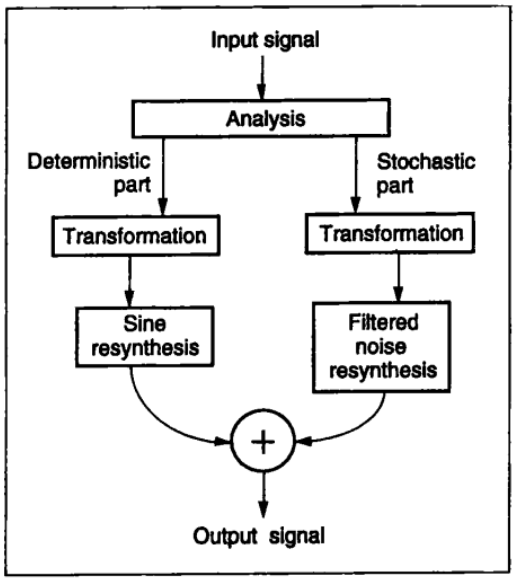


图 4.23　频谱建模合成的略图。输入信号被分为可确定部分与随机部分。每个部分可以在再合成之前分别被改变。(详见图 13.16，将有更详细的分析步骤。)
Input signal=输入信号　Output signal=输出信号　Analysis=分析　Deterministic part=可确定部分
Transformation=转换　Stochastic part=随机部分　Sine resynthesis=正弦再合成
Filtered noise resynthesis=经过滤波的噪音再合成

产生伪随机噪音(pseudorandom noise)的高效运算法(Knuth 1973a, Keele 1973, Rabiner and Gold 1975)很有影响。因为使用经过滤波的噪音可以得到相当大的数据缩减。在纯正弦波再合成中，没有这种类型的数据缩减，噪音成分必须由数百个正弦波来近似模拟。控制这些正弦波的函数会占据很大的内存，因此，从计算的角度考虑，正弦波再合成非常昂贵。

频谱建模合成的精准度问题仍未定论，比如，用来重建随机成分、经过滤波的伪随机噪音并不一定与原始声源的噪音品质相当。在许多声音里，“噪音”是那些具有听觉特性和特征的复杂音流产生的结果。在某些声音上，相同噪音的近似模拟工作还有待改进。

沃尔什函数合成(Walsh Function Synthesis)

目前为止我们所讨论的分析/再合成,都是建立在傅里叶分析的基础上,且以正弦波相加的方式再合成。傅里叶正弦波方法在研究上有久远的传统及应用方式,基于其原始理论,即对于周期性信号,可用不同频率的正弦波组合,以任意程度近似于原始信号。数学研究显示,除正弦波外,其他波形组也可以用于近似化模拟信号。一组称为沃尔什函数的方波,可以通过 Walsh-Hadamard transform 的分析后,用来近似模拟原始信号。沃尔什函数由长方波构成,因为它只有+1 与-1 两种值,所以是一种数字域的系列"digital domain series"(Walsh 1923)。

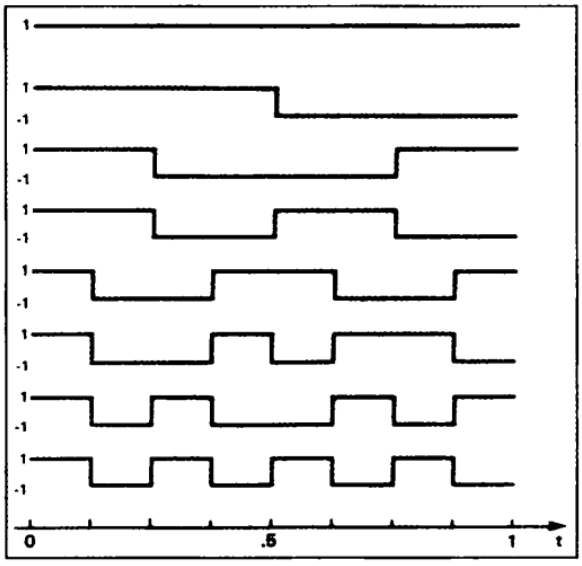


图 4.24 前八个沃尔什函数。

如傅里叶系列和该系列中的正弦波一样,一个任意周期的波形可以由无限个沃尔什函数的叠加获得近似值。傅里叶系列由不同成分的频率建构波形,而沃尔什合成则用不同的序列(sequences)的函数建构波形。序列被定义为每秒钟零交叉(zero-crossing)平均数值的一半(zps)(Hutchins 1973)。图 4.25 显示一个由几个沃尔什函数相加形成的混合波形。图示中显示,正弦波加法合成与沃尔什函数合成在观念上的对比,也就是说,沃尔什函数合成中,最难合成的波形就是纯正弦波。除非使用大量的序列项,否则,正弦波的沃尔什近似版本就会保留锯齿状。任何锯齿状都会破坏正弦波的实现,同时也令人反感。相对的,在傅里叶的正弦波合成技术中,最难合成的波形就是如方波那样带有直角

边的波形。比如，图 4.15，说明一个类方波要由 101 个正弦波相加获得。

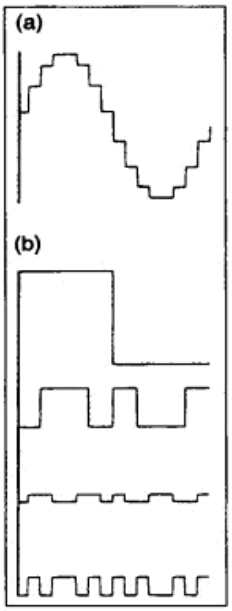


图 4.25　沃尔什函数相加之和的展示。通过将沃尔什函数相加(b)而构建的一个简单的正弦波近似值(a)。(出自 Tempelaars 1977。)

数字声音合成上，沃尔什函数合成的主要优点在它的长方形形状，可用便宜的数字线路并以高速运算而得。但相对于正弦波合成，沃尔什函数合成缺点是每个沃尔什函数并不能如在正弦波合成中那样与特定频率的谐波相关。然而，它可在数学上将傅里叶频率域转到沃尔什频率域(Tadokoro and Higishi 1978)。因此，我们可以用不同频率成分(partials)相加来解读声音，再将此解读转变成一组沃尔什函数合成器的参数值。更进一步，自然声音也可以被取样并使用 Walsh-Hadamard 转换技术，转换到沃尔什域，然后，用快速沃尔什转换(fast Walsh transform，FWT)作再合成(Hutchin 1973，1975)。

很多音乐合成操作系统已经按照沃尔什信号处理线路来设计，例如，哈钦斯(Hutchins)就用沃尔什函数线路设计了一个包络发生器(1973)，罗参伯(Rozenberg)和哈钦斯(1975)展示了实现振幅变换、减法合成、频率变换、频率位移以及混响——所有这些都是沃尔什域的技术。

即便沃尔什函数合成极有潜力，但基于此理论的实验器材仅有数个(Hutchins 1973，1975；Insam 1974)，而且没有一个曾被商品化。这可能是由于正弦波加法合成线路的价格不断下跌(包含内存芯片与乘法器)，所以沃尔什合成线路的经济优势就不大了。在傅里叶/正弦波方法上所累积的研究，以及在频率与感知之间的直觉关系上的研究，也造成在当代合成器设计中，正弦波加法合成的流行。

结论(Conclusion)

此章讨论了两种广为人用的合成技术:取样与加法合成。取样器能模仿其他乐器的声音。它的创造性合成能力较差,但它可以通过它的记忆功能来复制任何声源并回放。因为它能模仿真实乐器的丰厚声音,取样器是最受欢迎的电子乐器。

在数十年间,加法合成技术曾被深入研究。当加上了分析阶段,加法合成能够有效地模拟自然声音,且做出不同变化。如我们所见的,加法合成的主要缺点是为了得到模拟各种声音的一般性而牺牲了计算效率。为了模拟一个声音,必须先做过详细的分析。此分析可能会大量增加数据,所以需要经过数据缩减,才能够加以编辑。声音分析工具需要大量的计算,所以过去仅有昂贵的研究中心级机器上才能运算。但今日此情况已改变,复杂的声音分析与编辑工具可在笔记本计算机上作业。

下一章将讨论两种与采样及加法合成相关的合成技术,即多重波表合成与粒式合成。第5章其他的讨论技术是减法合成,在观念上与加法合成相反。

第 5 章　多重波表合成、波貌合成、粒式合成与减法合成
(Multiple Wavetable, Wave Terrain, Granular and Subtractive Synthesis)

本章内容包含许多种不同的合成技术。首先，我们从商用采样器与合成器中，经常使用的多重波表合成法开始；接着解释波貌合成，及同类型的粒式合成。余下的部分则探讨减法合成，是一种使用滤波器来塑造声音的有效技术。

多重波表合成(Multiple Wavetable Synthesis)

多重波表合成一词，指两种简单而在声音上很有效果的方法：波表交错渐变(wavetable crossfading)以及波堆积(wavestacking)。使用多重波表的合成技术不止这两种，事实上，很多技术都套用相同原则。但我们将此处所讨论的技术与其他类似技术区分开来，其原因是，此二者只有使用多重波表时才能工作。这两个技术在商用合成器与采样器中皆属常见。

霍纳(Horner)、比彻姆(Beauchamp)与哈肯(Hakken)在1993年发展了另一项技术，也称作多重波表合成(multiple wavetable synthesis)。它被归类为一种加法分析/再合成的变化型(如第4章所述)。但它也可视为此处所介绍的波堆积(wavestacking)方法的一种。在波堆积方法中，波表是分析和数据缩减阶段形成的正弦波的总和。

波表交错渐变(Wavetable Crossfading)

如第1章所述，固定波表合成中，数字振荡器重复读取已填入单一波形的波表。因为波形不断重复而未随时间改变，所以产生的音色是固定的。相对地，波表交错渐变是实现音色随时间变化的直接方法。振荡器交错渐变不是在不断重复扫描一个单一波形，而是在一个声音事件的全过程中，由两个或更多的波表做交错渐变。也就是说，声音事件由波形1开始，当波形1渐出时，波形2渐入，依此类推。图5.1绘出整个交错渐变的程序。波表交错渐变是复合合成(compound synthesis)(Roads 1985f)，向量合成(vector synthesis)(by Sequential Circuits, Korg and Yamaha Companies)，以及线性运算合成(L/A or Linear Arithmetic synthesis, Roland)的核心技术。

波表交错渐变可以在一个时间段内制造出由一声源变化为另一声源的声音。比方说，可用交错渐变技术将真实乐器如吉他、钢琴、或者打击乐器的丰富起音，嫁接到一个合成波形的延音部分。图5.2是使用波表交错渐变技术的乐器。

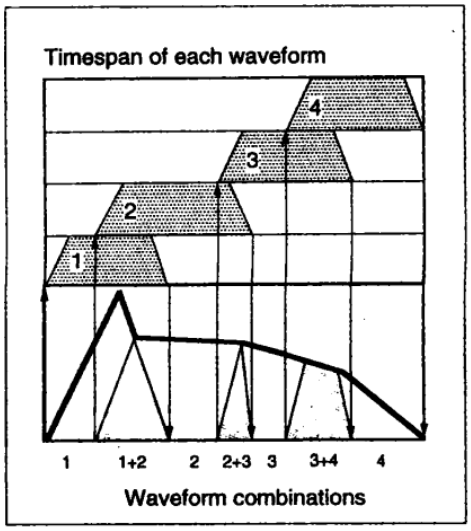


图 5.1 波表交错渐变。粗的线条是音符事件的振幅。四个波形在整个事件中交错渐变。下方的数字代表使用哪个波形，以及哪种波形结合。底端每个部分代表一个独立的音色，所以整个事件会在七种音色中交错渐变。
Timespan of each waveform=每个波形的时间间隔
Waveform combinations=波形组合

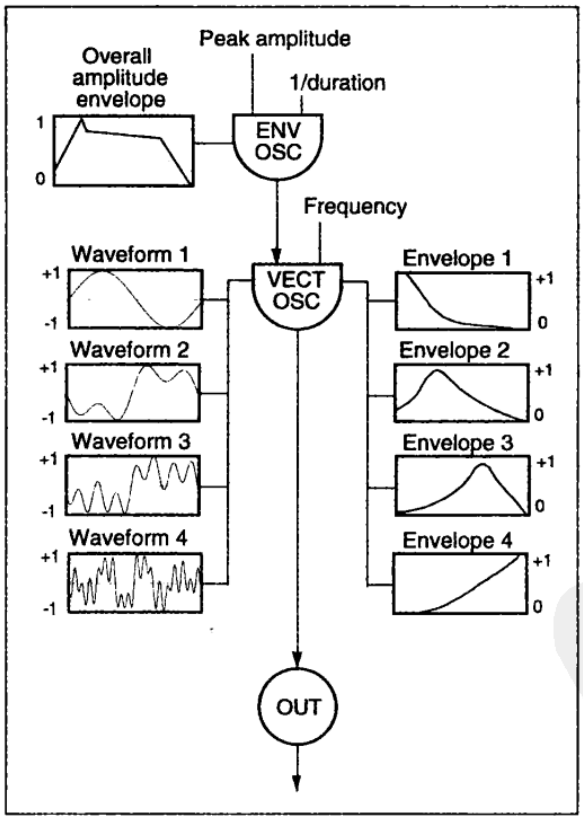


图 5.2 使用了四个波表的波表交错渐变(向量合成)乐器。右边的每个包络会用在左边相对的波表上。
Peak amplitude=振幅峰值
Overall amplitude envelope=总的振幅包络
1/duration=1/时值
Frequency=频率
Waveform=波形
Envelope=包络
ENV OSC=包络振荡器
VECT OSC=向量振荡器
OUT=输出

第一台使用多重波表交错渐变的商用合成器是 Sequential Circuits Incorporated Prophet VS，在 1985 年问世(图 5.3)，可在四个波形间交错渐变。更新的合成器可以在一个声音事件中让使用者指定任意数目的波表作交错渐

变(图 5.4)。交错渐变可由自动控制(由一个声音事件驱动)或由摇杆手动控制,如同 David Smith 所设计,Korg 及 Yamaha 所制造的向量合成(vector synthesis)。

图 5.3　Prophet VS 数字合成器,由 Sequential Circuits Incorporated 制造(1985)。

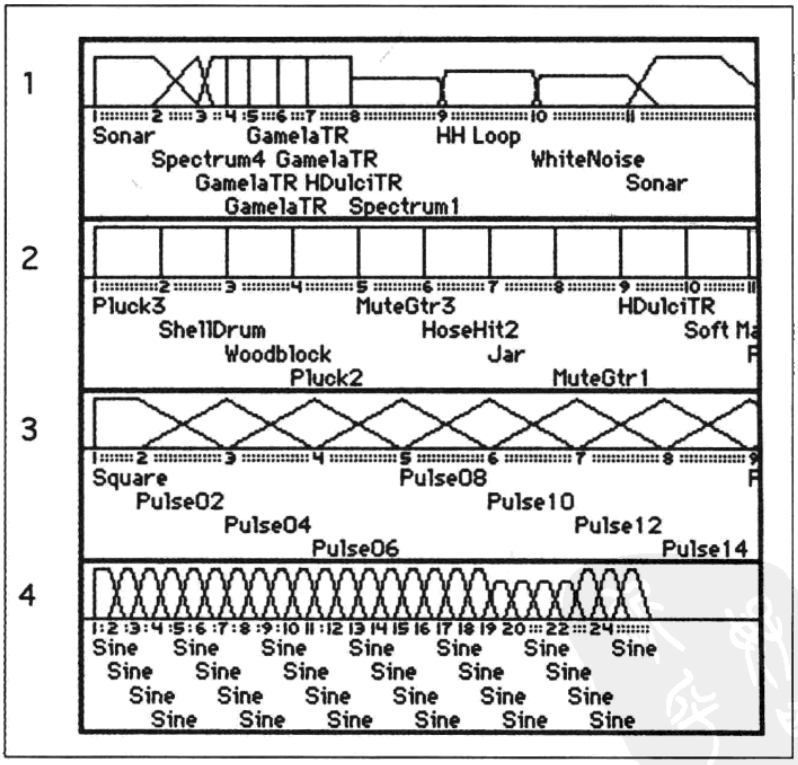


图 5.4　向量合成器的线路排秩编辑器画面,显示四种声音的波表序列。注意第四个声音会在 24 个波表内交错渐变。虽然每个波形都标为“sine”,这些“sine”可能会有不同的强度,并有不同数量的周期,从而造成瞬时变化。

波堆积(Wavestacking)

波堆积(wavetable stacking 或 wavestacking)是简单而有效的加法合成的变种。此方法中,每个声音事件是由数个波形的叠加合成而得(在商用合成器上通常有四个到八个波形),这与传统的加法合成不同。传统的加法合成是叠加正弦波,而波堆积的每个波形都可以是一个复杂的信号,如取样声音(图 5.5)。

将数个取样声音堆积后,可以做出混合音色,如萨克斯管/笛子或小提琴/竖笛。每个堆积中的波形有个别的振幅包络,以便在一个声音事件中的波堆积内交错渐变。当堆积四到八个复杂声波后,每个声音事件都会相当深邃而丰厚。

波堆积的做法,是储存波形数据库后,再用查表振荡器扫描读取。每个波形的包络必须经 $1/n$ 的比例参数测量,n 值与堆积波形的数量相当,以避免数字超过可表示范围(overflow)。(也就是波形之和应该在合成器所能表示的量化范围之内。)波堆积已在许多商用合成器上使用,有时候它会与多重波表交错渐变法结合,以创造绚丽的内在动感与频谱变化的声音。

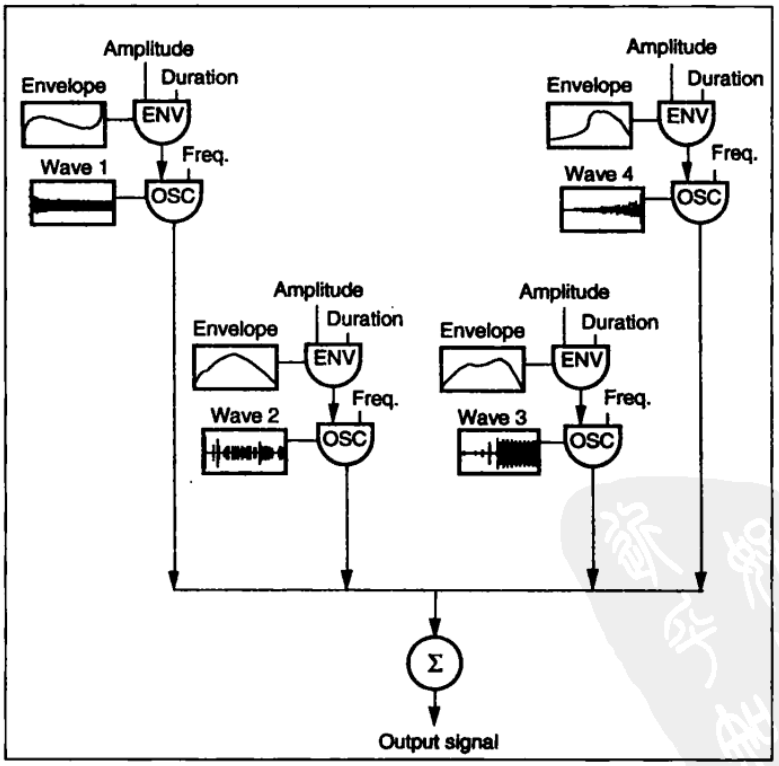


图 5.5 波堆积。把由四个振荡器送出的信号加在一起。注意波表内不是单一周期函数,而是长的取样声音。

Amplitude=振幅 Envelope=包络 Duration=时值 Freq.=频率 Wave=波 Output signal=输出信号

波貌合成(Wave Terrain Synthesis)

如第1章所解释，许多合成技术都是由最基本的波形查表法开始：也就是以一个索引值，每隔一取样间距渐增其值，读取波表中的取样。也可将此查表原则应用在扫描读取三维的"波形面"，我们沿用了戈尔德(R. Gold)的用法，即称此波形面为波貌(wave terrain，WT)(Bischoff，Gold，and Horton 1978)。Gold，Leonard Cottrell (Bischoff，Gold，and Horton 1978)，Mitsuhashi (1982c)，以及 Borgonovo 和 Haus(1984，1986)等计算机音乐研究者相互沟通了意见，他们共同开发了使用两个索引值读取波貌技术的可能性。(Borgonovo and Haus 1986 的论文中包含实现此技术的原始码。)

波貌与轨道(Terrains and Orbits)

传统的波表可以画成二维图，相当于由索引值 x 表示的函数 wave(x)。一个有两个索引值的波貌可表现为 wave(x，y)，在三维的面上作图(图 5.6)。此情形下，z 点或表面的每一个点的高度，代表两个索引值(x，y)上的波形值。波形储存在名为"双变量函数"(function of two variables)的表中，所以此技巧也被称为双变量函数合成(two-variable function synthesis)(Borgonono and Haus 1986)。

波貌读取的轨迹称为轨道(orbit)。虽然在天文学上此词指椭圆函数，但此处轨道可包含波貌上任何一个点的序列。我们将很快会进一步讨论轨道的问题，但，首先，我们面临的问题是如何用波貌合成(WT)产生可以预测的波形。

由波貌产生可预测的波形 (Generating Predictable Waveforms from Wave Terrains)

在音乐用途上，任何三维面都可以当作一种波貌——从严格定义下的数学函数，到任意的地形学设计，如地理学上的地形图。这并不令人惊讶，然而，对此技术的系统性讨论，着重在如何由简单数学函数来产生波貌。与频率调制及波成形(waveshaping)技术相同(第6章)，使用简单数学函数的好处在于可以精确预测由波貌所产生的波形与频谱。Mitsuhashi(1982c)、Borgonovo 和 Haus(1986)发明了其范围在[$-1\leqslant x\leqslant 1$，$-1\leqslant y\leqslant 1$]的平滑数学波貌公式。为预测输出波形，以下的条件必须满足：

1. x 与 y 两个函数，以及它们第一次序的泛音派生物(derivatives)要在整个波貌中持续(在数学意义上)。

2. 在波貌的边界上，x 与 y 的函数两者皆为 0。

第二个特点确保当轨道从波貌一个边缘到另外一边跳跃时，其函数以及其派生物的可保持其连续性。如此跳跃相当于在单一指数波表扫描中由右到左的来回环绕(wraparound)。

在图 5.6 中所绘的波貌符合以上条件，并可由下式定义：

$$\text{wave}(x,y) = (x-y)\times(x-1)\times(x+1)\times(y-1)\times(y+1)$$

我们将会看到这个函数如何由轨道扫描，来产生不同的波形。参见 Mitshuhashi(1982c)、Borgonovo 和 Haus(1986)有关对类似函数的定义。

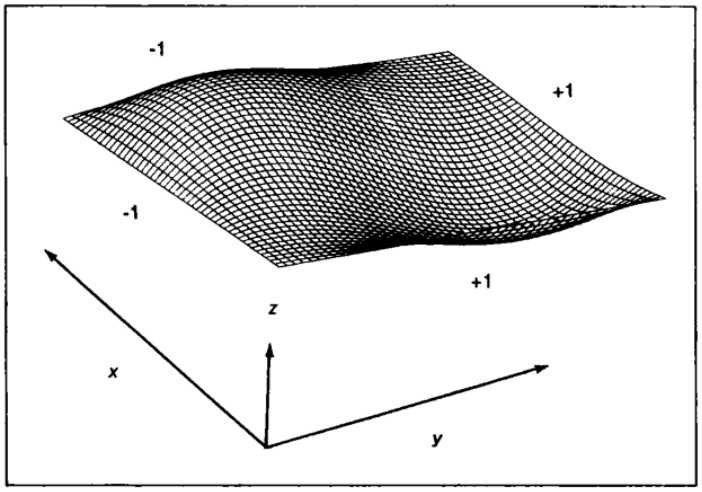


图 5.6 是一个三维面的波貌图，波貌的高度(z-轴)表示该波形的值。

周期性的轨道(Periodic Orbits)

由波貌产生的输出信号既取决于波貌也取决于轨道的轨迹。轨道可以是波貌表面上的直线、曲线、随机曲线、正弦曲线函数或由 x 与 y 两个维度的正弦曲线产生的椭圆函数。如果轨道有重复性(周期性)，输出信号也会有周期性。图 5.7 的顶端显示周期性的椭圆轨道，由下列函数定义。

$$x = 0.5\times\sin(8\pi t+\pi/5)$$
$$y = \sin(8\pi t)$$

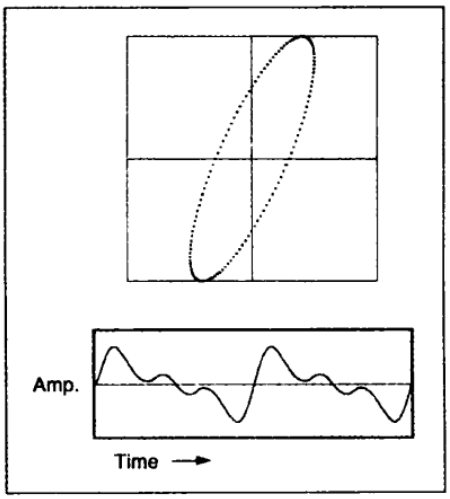


图 5.7　椭圆轨道以及所产生的信号。(上)椭圆轨道之轨迹。x 与 y 两个维度都从 −1 变化到 +1。(据 Borgonovo and Haus 1986)。(下)由椭圆轨道扫过公式 1 定义的波貌产生的波形。(注:此波形是由 Borgonovo and Haus 1986 所重绘。)
Time=时间　　Amp.=振幅

图 5.7 的底部,显示由公式 1 波貌上的椭圆轨道所产生的周期波形。

图 5.8 为由以下函数定义的,在波貌上循环的周期轨道的另一示例

$$x=0.23\times\sin(24\pi t)$$

$$y=(16\times t)+0.46\times\sin(24\pi t+\pi/2)$$

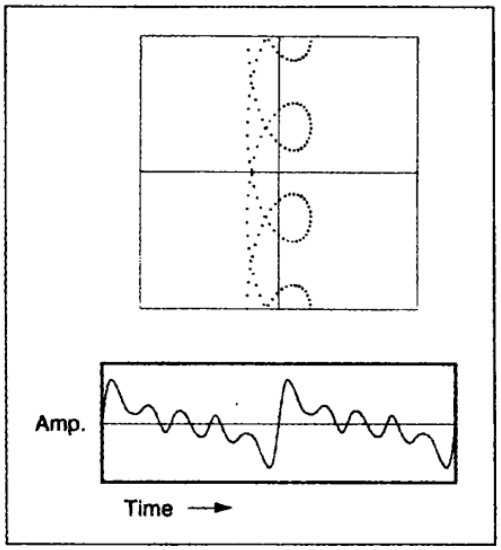


图 5.8　这是一个循环轨道和它所实现的信号。(上)轨道的轨迹图,x 与 y 两个维度都从 −1 变化到 +1(据 Borgonovo and Haus 1986);(下)由循环轨道扫过公式 1 定义的波貌产生的波形。(注:此波形是由 Borgonovo and Haus 1986 所重绘。)
Time=时间　　Amp.=振幅

时变轨道(Time-varying Orbits)

当轨道固定后,得到的声音也会是固定波形,有固定频谱。要产生时变波形的方法,是令轨道随时间改变(图 5.9)。比方已知螺旋状轨道可产生有趣的结果。

我们也可想象它的延伸应用方式。将轨道固定,但让波貌随时间改变。在此情形下,读取波形的程序相当于在一个起伏的表面扫描,就像是海面的波动。

波貌(WT)合成已被证明是能产生合成波形的高效率实验性技术,然而,为了要近似于语音信号,或者真实乐器音色,仍需研究如何调整此方法内的参数。

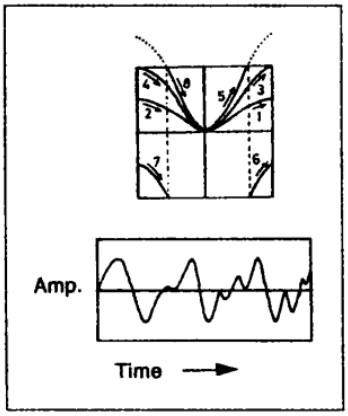


图 5.9 非周期性轨道以及其输出信号。(上)通过波貌的八个轨迹;(下)注意此波形随时间改变。(根据 Mitsuhashi 1982c。)
Time=时间 Amp.=振幅

粒式合成(Granular Synthesis)

如同光可以同时拥有波的性质和粒子的性质(光子),我们也能如此地想象声音。粒式合成是由数千个声音颗粒建构起一个声音事件。声音颗粒的时间非常短暂(通常是 1 到 100 毫秒),这接近了人感知分辨声音的时值、频率、强度的时长极限。

以颗粒方式表示声音,是检视复杂声音现象的有用方法,如同基本能量单位的聚合,而每个单位有其时间与频率。这样的表示方法在合成与信号处理算法上很常见。当然,还有很多不同的名词来表示同一个现象,如"量子论"(quantum)(Gabor1946,1947)、"高斯元素信号"(Gaussian elementary signal)(Helstrom 1966, Bastiaans 1980),"短时片段"(short-time segment)(Schroeder and Atal 1962),"短时权重函数"(short-time weighting function)(Flanagan 1972),"窗口"(window)(Arfib 1991, Harris 1978, Nuttal 1981),"滑动窗"(sliding window)(Bastiaans 1985),"窗口函数脉冲"(window function

pulse)(Bass and Goeddel 1981),"小波"(wavelet)(Kronland-Martinet and Grossmann 1991),"共振峰—波—函数"(formant-wave-function 即 FOF)(Rodet 1980),"VOSIM 脉冲"(Kaegi and Temperlaars 1978),"波信息包"(wave packet)(Crawford 1968),"音爆裂"(toneburst)(Blauert 1983, Prince 1990),"音脉冲"(tone pulse)(Whitfield 1978)以及"音果仁"(tone pip)(Buser and Imbert 1992)等,都是将音乐信号描述为颗粒的表示方式。

以颗粒方式表示声音极为恰当,因为它把时间信息(开始时间、时值、包络形状、波形)与频域信息(在颗粒间的波形所占时间,波形的频谱)结合了在一起。这与仅表示声音强度,但无法捕捉频域信号的表示法不同,也与认为声音为无限长的正弦波总和的、抽象的傅里叶表示法不同。

粒式合成:背景(Granular Synthesis: Background)

将声音视为原子状的"颗粒"观念,可以追溯至科学革命的起源。荷兰学者依萨克·比克曼(Issac Beekman, 1588—1637)在 1616 年提出微粒子"corpuscular"声音理论(Beekman 1604—1634; Cohen 1984)。比克曼(Beekman)相信任何振动物体,如弦,会将周围的空气切割为螺旋状的空气微粒,经由振动向所有方向扩散。比克曼的理论是,当这些微粒撞及耳膜,我们就能听见声音。虽然这个理论在严格科学定义上不成立,但是它提出了粒式合成在听觉上的生动比喻。

几个世纪后,英国物理学家丹尼斯·伽伯(Dennis Gabor)再次提出声音的颗粒或量子描述方式,在他的两篇精彩论文中,把量子物理理论性的洞见与实际实验结合了在一起(1946,1947)。根据伽柏(Gabor)的理论,颗粒表示方式可以描述任何声音,这个假设在数学上由巴斯蒂安(Bastiaans)证明(1980,1985)。20 世纪 40 年代,伽柏根据由投影机改装的齿链光学录音系统,实际建立了声音粒式合成器。他使用这机器做了时间压缩/延伸且带有移调的实验——改变声音音高而不改变时长及其相反的实验。(见第 10 章,对于时间压缩与延伸及移调的讨论。)

20 世纪 60 年代所发展的短时傅里叶转换(short-time Fourier transform)所用的窗口化(windowing)技术也隐含了声音的颗粒表示法(Schroeder and Atal 1962,见第 13 章与附录)。MIT 的神经机械学家诺伯特·维纳(Norbert Wiener)(1964)与信息理论家阿布拉哈姆·莫洛斯(Abraham Moles)(1968)也曾提出声音的颗粒表示法。

作曲家克赛纳基斯(Iannis Xenakis)(1960)是阐明声音颗粒作曲理论的第一人。他开始采取以下的原则:"所有声音,即便是连续的音乐变化,都可想象为适当分布在时间中的大量基本声音的组合。在一个复音的起音、音体以及衰减部分,数千个纯音在或多或少短暂的时间内出现。"克赛纳基斯使用模拟声音

产生器及磁带拼接，创造了颗粒声音。这在他为弦乐团及磁带所作的作品 *Analogique A-B* 中可以找到。此作曲由克赛纳基斯所陈述(1992)。〔作品乐谱与磁带来自萨拉伯特版(Editions Salabert)。〕

本书作者在 1974 年间于加州大学圣地亚哥分校(Roads 1978c)及 1981 年于麻省理工学院(Roads 1985g)，发展了第一个计算机上的粒式合成程序。此技术在许多作品中出现，包括《nscor》(1980，Wergo compact disc 2010-50)，《Field》(1981，MIT Media Laboratory compact disc)，以及《Clang-tint》(Roads 1993b)。粒式合成曾以不同方式实现，最著名的是加拿大作曲家巴里·杜亚士(Barry Truax)(1987，1988，1990a，b)，我们会在之后详述。

声音颗粒(Sonic Grains)

振幅包络塑造了每个颗粒的形，此包络可改为许多不同的形状，从高斯(Gaussian)钟形曲线到简单的三段折线：起音/延音/衰减(图 5.10)。以下公式定义了高斯(Gaussian)曲线 $P(x)$：

$$P(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-1(x-\mu)^2/2\sigma^2}$$

σ 表示标准差(钟型的宽度)，而 μ 表示平均值或中间峰值。

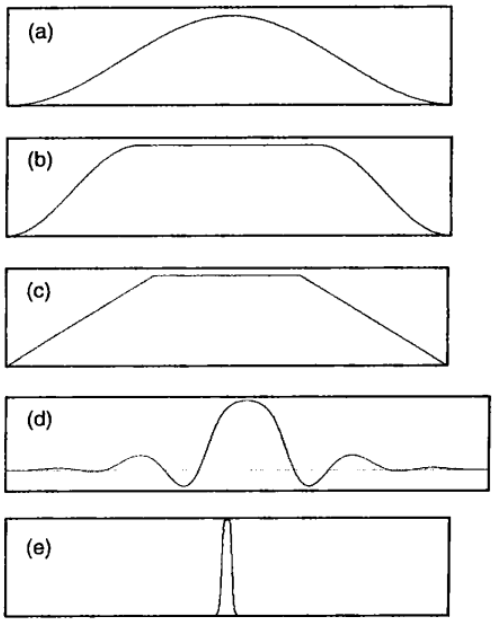


图 5.10 颗粒包络。(a)高斯(Gaussian)；(b)准高斯(Quasi-Guassian)；(c)三段折线；(d)脉冲(Pulse)；(e)窄脉冲(Narrow impulse)，也可当成是短时间范围内的(a)模式。

图 5.10b 显示准高斯(Quasi-Gaussian)曲线，或称作土耳其窗口(Turkey window)(Harris 1978)，其峰值占至整个颗粒的 30%－50%。此形状被证明在声音应用上相当有用(Roads 1985g)。

如带限脉冲(band-limited pulse)般的复杂包络(图 5.10d)会产生带有共振的颗粒，当颗粒小于 100 毫秒时，听起来像是稀疏的敲木块声音。当窄包络小于 20 毫秒，如图 5.10e，会造成爆裂(crackling and poping)的音质。如人们所想的，包络中如有尖锐的角，将对其频谱带来大幅的副作用。这些副作用是因为包络频谱与波形颗粒的卷积(convolution)所造成。(第 10 章对于卷积有更详细的解释。)

颗粒长度可以是固定常数，随机的，或是随频率而改变。比方说，我们可以给较高频率的颗粒设定较短的长度。颗粒频率与颗粒长度间的相关性，是小波分析/再合成(wavelet analysis/resynthesis)的重要特征，我们将在本章和第 13 章中讨论。

在颗粒中的波形可为两类：合成波形和采样波形。合成波形通常是以特定频率扫描的正弦波的总和。而采样颗粒，通常读取来自储存声音文件中波形的规定位置，同时带有或没有音高位移。

在颗粒层级上，有许多参数可调整，包含时值、包络、频率、声音文件中的位置(对采样颗粒而言)、空间定位以及波形(合成颗粒的波表，或是文件名称，或是采样颗粒的输入声道)。恰恰由于有许多颗粒层级上的控制，才能使此方法产生许多独特的效果。

粒式产生器设备(Grain Generator Instrument)

粒式合成可用简单的合成设备所实现：一个由包络产生器控制的正弦波振荡器(图 5.11)。也可轻易地延伸此设备，以允许选取不同的波形函数。

即便此设备如此简单，要产生单纯、不复杂的声音也需要大量的控制资料——每秒的声音需要上千个参数。这些参数描述每个颗粒的起始时间、振幅等。因为不可能用手动方式设定每个颗粒的参数，所以必须发展出高阶的组织单元。此组织单元应可自动产生上千个个别的颗粒指定参数。

高阶粒式组织(High-level Granular Organizations)

由粒式合成产生声音的复杂度，来自于输入的控制资料量。如果 n 是每个颗粒所需参数的数量，d 是每秒钟声音颗粒的平均密度，那么每秒钟就需要 $d \times n$ 个参数值。由于 d(每秒密度)值通常是数十个到数千个，因此很明显，为了作

曲上的控制，我们需要一个高阶组织单元来控制颗粒。此单元的用途，是让作曲家使用少数几个全域参数来控制大量的颗粒。

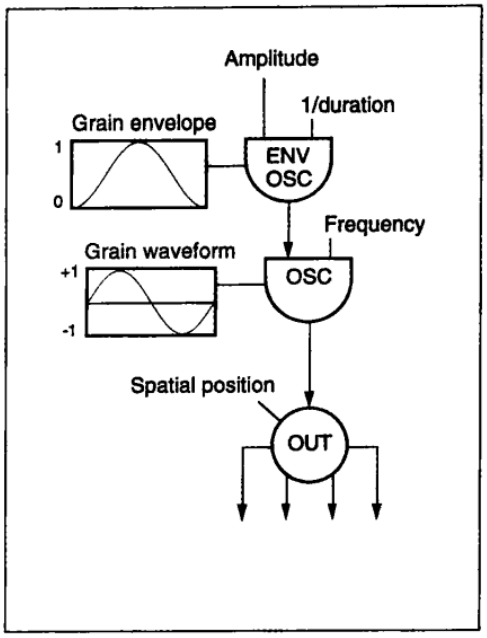


图 5.11 这是一个简单的粒式合成设备，由包络产生器及多信道输出的振荡器组成。
Amplitude=振幅 1/duration=1/时值 Frequency=频率 Grain envelope =颗粒包络
Grain waveform=颗粒波形 Spatial position=空间位置

根据颗粒的组织方式来区分，现有的粒式合成法，可分为五类：

1. 傅里叶与小波栅格(Fourier and wavelet grids)
2. 音高同步重叠串流(Pitch-synchronous overlapping streams)
3. 准同步串流(Quasi-synchronous streams)
4. 异步粒式云(Asynchronous clouds)
5. 时间颗粒或采样声源串流，可带有重叠、准同步、或不同步回放(Time-granulated or sampled-sound streams, with overlapped, quasi-synchronous, or asynchronous playback)

在下几节中，我们将简要叙述每一种方法。

傅里叶与小波栅格及声幕(Fourier/Wavelet Grids and Screens)

短时傅里叶变换(short-time Fourier transform, STFT)与小波变换(wavelet transform)，可处理时域声讯，并量取时间上的频率成分，是两个相关的频谱分析技术(第 13 章将介绍此两者)。事实上，这些方法将分析栅格(anal-

ysis grid)内的每个取样点与时间—频率能量单位联系在了一起，也就是颗粒或小波的联系。(图 5.12)

著名的短时傅里叶变换(STFT)可用快速傅里叶变换演算 FFT(Rabiner and Gold 1975)。“颗粒”在这里，是指在傅里叶分析器 N 个信道(图 5.12 中的横向列)的每一个信道内的一组叠盖分析窗。我们可以将这些颗粒视为排列在二维时间/频率栅格上的颗粒，每个栅格的间距相同。Arfib(1991)以颗粒的动作描述短时傅里叶变换(STFT)的应用方式。

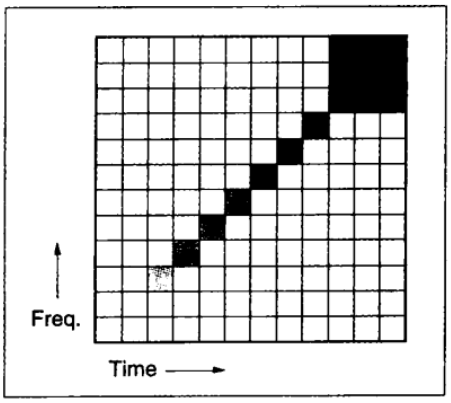


图 5.12　傅里叶栅格将时域与频域分成格的单位，每一列代表一个频率信道，每一行代表一个时间段。每个方格内黑色的程度代表该时间—频率区域内的强度。这个范例显示一个声音频率在上升，能量在增加的状态。在短时傅里叶变换(STFT)中，频率栅格是线性的，而小波转换中则是对数的。
Time=时间　Freq.=频率

小波转换(Kronland-Martinet and Grossmann 1991)执行同样的动作，但是分析的信道间隔以及取样窗的时值(称为分析小波，analyzing wavelet)与短时傅里叶变换(STFT)不同。在 STFT 中，频率信道上的间隔是线性关系，而小波转换的间隔则是对数关系。也就是说，在小波转换中，每个频率信道的间距(带宽)$\Delta f/f$ 是固定的。同样在 STFT 中，窗格时值是固定的，而小波转换中的时值则是随着频率函数而改变(见第 13 章)。

两种技术都能够分析、转换、重新合成，是操控取样信号极有潜力的音乐工具。使用傅里叶/小波栅格最明显的转换包括采用栅格延伸或压缩，以得到带有音高移调的时间延伸或压缩，也就是改变音高而不改变时值，或是其相反的效果。

另外一个由栅格所衍生，但与傅里叶或小波分析无关的观念，是 Xenakis (1960,1992)的声幕(screen)观念。声幕是散布着声音颗粒的振幅—频率栅格。同步的声幕序列(称为 book)构成复杂声音的演化。不同于由声音分析开始的傅里叶/小波转换，由声幕为基础的声音合成以运算法自动产生颗粒，填入声幕

中。克赛纳基斯(1971,1992)提出以随机方式将颗粒散布在声幕上,然后用集合理论(set-theory)的运算法来形成新的声幕,这些集合运算包含交集、联集、补集、差集等:

"采用带有这些颗粒集合的所有种类的操作,我们希望创造的不单是传统乐器的声音和有弹性的声音形体,及在具体音乐中所偏好的声音,而是至今为止空前未有且无可想象的,持续变化发展的音波扰动。"

另外一个基于声幕观念的方法,则由细胞自动体(cellular automata)互动得到颗粒参数(Bowcott 1989)。

音高同步粒式合成(Pitch Synchronous Granular Synthesis)

音高同步粒式合成(PSGS)是为产生频谱上带有一个或多个共振峰区的音而设计的一种技术(De Poli and Piccialli 1991)。音高同步粒式合成的处理有多个阶段,包含音高侦测、频谱分析与再合成,及以脉冲响应为基础的滤波程序,诸多技术步骤在后续章节中将一一叙述;所以此处的介绍将很简短(见 De Poli and Piccialli 1991)。

分析的第一个阶段是音高侦测(见第 12 章)。将每个音高周期视为一个独立单位,或是一个颗粒。针对每个颗粒做频谱分析。系统得到频谱的脉冲响应,而用这个系统来设定再合成滤波器的参数。(第 10 章将讨论脉冲响应的测量。)

在再合成阶段,使用侦测到的音高周期上之连续脉冲(pulse train),驱动一组有限脉冲响应(finite impulse response,FIR)滤波器。(FIR 滤波器会在第 10 章中讨论。)输出信号的结果来自在所有滤波器脉冲响应的分量总和上建立的连续脉冲刺激。在每个时间帧(frame)上,系统送出与前一个颗粒交迭相加的颗粒,以做出平顺的变化信号(图 5.13)。由 De Poli 和 Picialli 提出的音高同步粒式合成(PSGS)的应用方法有许多声音变形功能,可以创造出原始声音的变化版本。后续的延伸,使声音的准和谐(quasi-harmonic)部分与余冗的非和谐部分得以分离(Piccialli et al. 1992)。

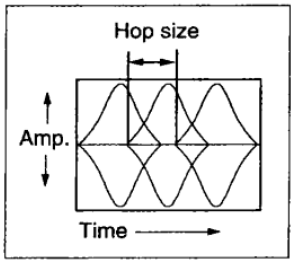


图 5.13 为重叠的颗粒串流。跃幅(hop size)则是相邻颗粒间的延迟时间间距。
Time=时间 Amp.=振幅 Hop size=跃幅

准同步粒式合成(Quasi-synchronous Granular Synthesis)

准同步粒式合成(quasi-synchronous granular synthesis,QSGS)产生一个或数个颗粒串流,一个颗粒接着另一个,每个颗粒间有可改变的延迟时间。串流观念的优点是明确且感性。Orton,Hun and Kirk (1991)发展了图像界面,可直接在屏幕上绘出曲线般的串流轨迹。

图 5.14 显示了一个有五个颗粒的串流,每个都套用准高斯(quasi-Guassian)包络,可改变颗粒间的延迟时间。我们说"准同步",是因为颗粒后接着或多或少间距相当的时间间隔。当相邻颗粒间的间距相同时,整体的颗粒串流包络形成一周期函数。因为此包络是周期性的,所以准同步粒式合成(QSGS)所产生的信号可以被分析,如振幅调制(AM)分析。振幅调制(AM)是以一个信号的形状(调制器 modulator)控制另一个信号(载波 carrier)的振幅大小。(见第 6 章的调制说明。)在此例中,载波者是颗粒内的波形,而颗粒包络则是调制器。

从信号处理的角度看,我们观察到对载波中的每个正弦波成分来说,包络函数会将一连串边带(sidebands)提供给终端频谱。(边带是指在载波频率上下的附加频率成分。)边带通过与包络函数周期倒置相对应的距离与载波区分开来。对一个 20 毫秒接续不断的颗粒串流,其输出信号频谱的边带是以 50Hz 为间距,该颗粒包络的形状决定这些边带的振幅大小。

周期性包络调制效果产生的结果,是载波周围的共振峰的面貌。也就是说,在频谱上不是单一的直线(表示一个单一频率),而会像是一条山脊(表示一组在载波周围的频率)。从这个意义上讲,准同步粒式合成(QSGS)相当于共振峰合成(VOSIM, Kaegi and Temperlaars 1978)、共振峰—波—函数合成(formant-wave-function)即 FOF 合成(Rodet 1980; Rodet, Potard and Barrière 1984)。(见第 7 章对 VOSIM 与 FOF 合成法的介绍。)

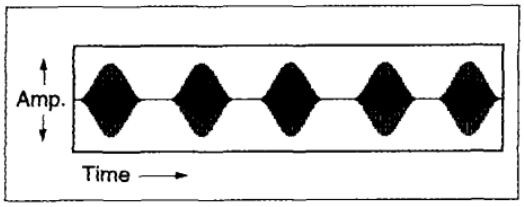


图 5.14　是一个有五个 40 毫秒时值颗粒的串流,在 1.06kHz 频率上带有一个汉宁(Hanning)包络,在这个串流中,颗粒间的延迟间距略有变化。
Time=时间　　Amp.=振幅

通过平行结合数个准同步颗粒串流(每个串流围绕相分割的频率创造它自己的共振峰),此信号可以模拟人声歌唱与真实乐器声音的共振。

当颗粒间的间距变得不规则,如图 5.15,将会产生通过"模糊"共振结构来控制声音织体的粗细(Truax 1987,1988)。在其最简单的形式下,可变延迟法相当于使用低频有色噪音(colored noise)作为调制器的振幅调制(AM)。(见第 6 章对调制的说明。)单单使用此技巧本身并非特别有趣。但在粒式合成法上能做到的远超过简单的噪音调制的 AM(振幅调制)。尤其是,我们可以同时在颗粒等级上改变数个其他参数,如颗粒波形、振幅、时值以及空间定位。在更全域的等级上,我们还可以动态改变每秒钟颗粒的密度,以创造出多变有冲击力的效果。

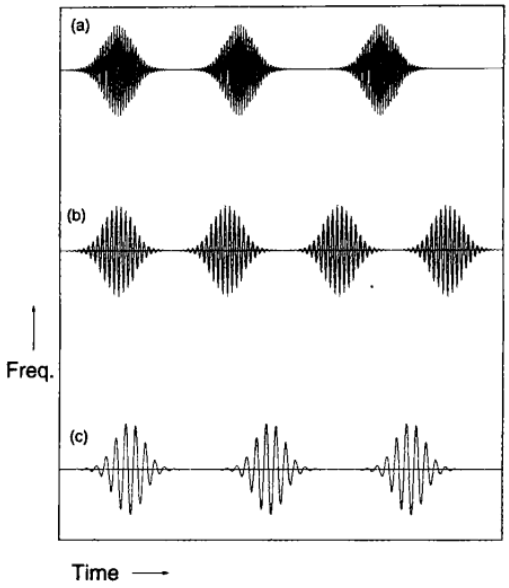


图 5.15 是三个准同步粒式合成中的颗粒串流。串流的纵向分布标明颗粒的频率(也就是波形的频率)。颗粒间的开始时间是随机的。
Time=时间 Freq.=频率

异步粒合成(Asynchronous Granular Synthesis)

异步粒合成(AGS)给了作曲家一把精密的声音喷枪,每个喷点是一个声音颗粒(Roads 1991)。异步粒合成以统计方式将颗粒散布在频率—时间的平面上。这些区域称为颗粒云(clouds)——也就是作曲家赖以工作的单元。

作曲家通过下述参数来设定颗粒云,如图 5.16。

1. 颗粒云的起始时间与时值。
2. 颗粒的时值(通常是 1 到 100 毫秒,但也可以超过或低于此界限)。颗粒

时值可以设为常数，或是在限定范围内由几率曲线得到的随机值，或随着颗粒频率的函数改变，在这种改变中，高频颗粒的包络较短。

3. 每秒钟的颗粒密度，比方说，如果颗粒密度很低，便只有少一些的颗粒随机散布在云中；如果颗粒密度高，颗粒会重叠而得到复杂频谱。在云的时程内，密度可随时间改变。

4. 颗粒云的带宽(bandwidth)。通常由两个曲线给定高低频的边界，在两个曲线内是颗粒散布之处(积云，cumulus clouds)；另外，在云中的颗粒频率可以设定在具体的音高集合上(如层云，stratus clouds)。

5. 颗粒云的振幅包络。

6. 颗粒中的波形，这是颗粒云最有效参数之一。比方说，云中的每个颗粒可以有不同的波形，波形可以是合成的或是取样的声音。

7. 颗粒云中颗粒的空间散布定位，在指定应用中其输出通道数量是特定的。

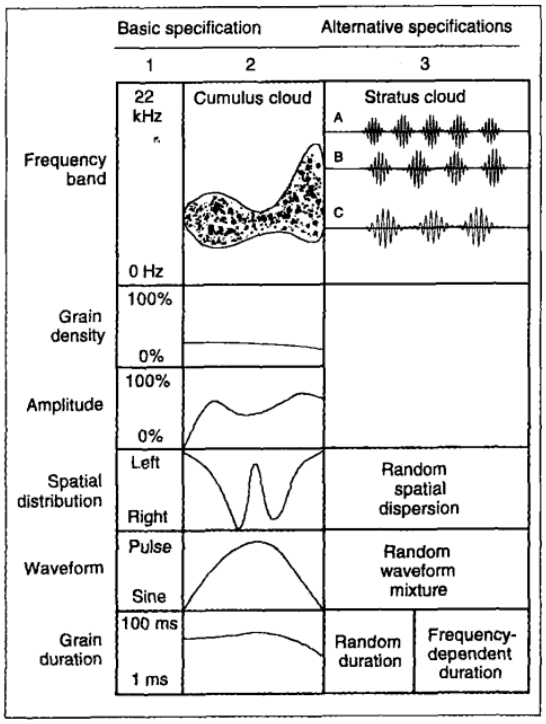


图 5.16　是异步粒合成中颗粒云参数的图解。第一列是典型的参数分布。第二列是标准颗粒云的基本说明。第三列是频带、空间分布、波形与颗粒时值参数的变化说明。
Basic specification=基本设定　Alternative specifications=替换设定　Frequency band=频带　Grain density=颗粒密度　Amplitude=振幅　Spatial distribution=空间分布　Waveform=波形　Grain duration=颗粒时值　Left=左　Right=右　Pulse=脉冲　Sine=正弦　cumulus cloud=积云　stratus clouds=层云　Random spatial dispersion=随机空间散布　Random waveform mixture=随机颗粒混合　Random duration=随机时值　Frequency-dependent duration=频率依赖式时值

通过改变异步粒合成(AGS)的这七个参数,我们可以得到相当广泛的效果。这节的其他部分简要地概括了时值、波形、频带、密度以及空间效果。波形与带宽参数只应用在合成中,而不应用在取样颗粒上。要更详尽地了解异步粒合成的参数效果,参见 Roads(1991)。

如图 5.16 所示,颗粒时值可以是常数(一条直线)、变量或在两极限值间的随机数或是依频率改变。

颗粒时值改变颗粒云的声音织体。较短的时值会造成裂声(crackling)、爆炸声响。而时值较长会较平顺。在颗粒时值的设定采用了信号处理成熟的定律:一个声音事件的时值越短,带宽越大。图 5.17 三个基本信号验证了此定理。

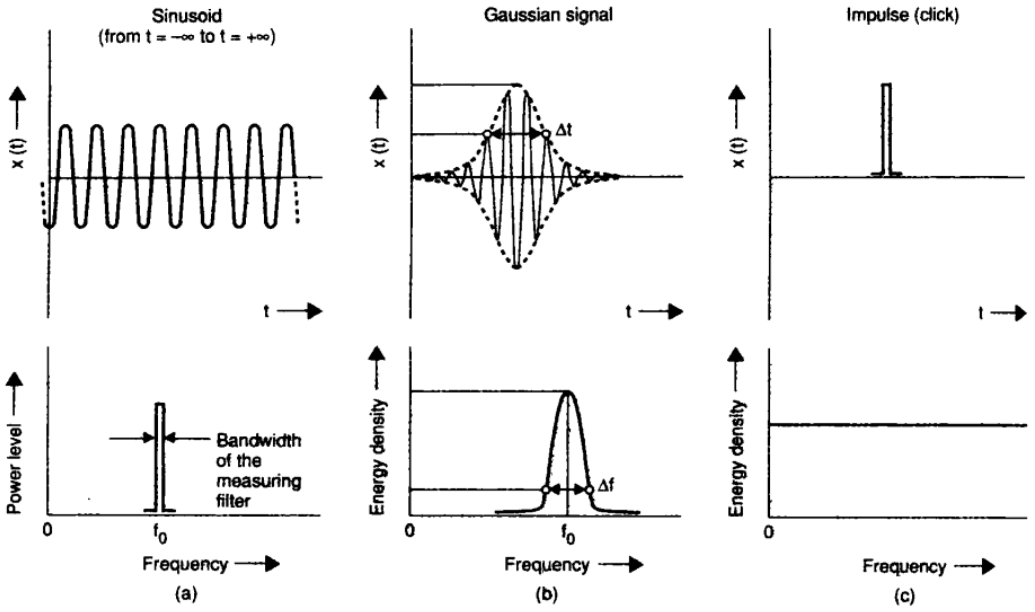
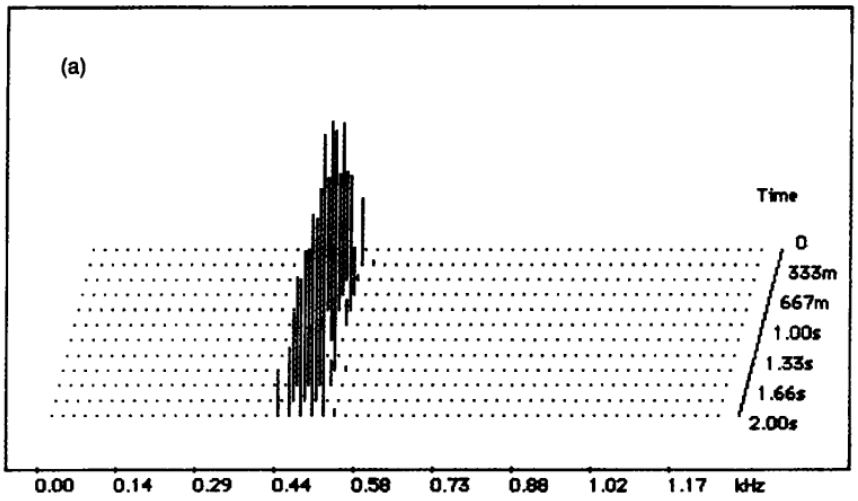


图 5.17 是三种基本信号的时域函数(上)和频谱(下)Blauert(1983)。(a)无限长的正弦波,相当于频谱上的一条直线;(b)高斯颗粒及相对应的共振峰频谱;(c)短脉冲以及其相对应的无限带宽频谱。

Power level=功率水平 Sinusoid=正弦 Impulse=脉冲 Energy density=能量密度
From t=- ∞ to + ∞=从- ∞到+∞ Click=杂音 Frequency=频率 Gaussian signal=高斯信号

图 5.18 显示缩短颗粒时值时的频谱效果。注意带宽会由于颗粒时值缩减而大幅增加。



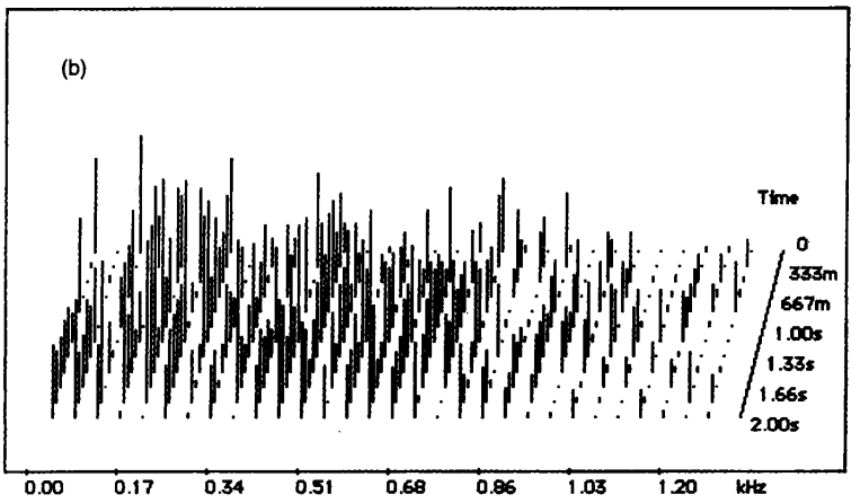


图 5.18　颗粒时值的频谱效果。(a)一个颗粒云的频谱，颗粒长为 100 毫秒固定频率 500Hz。注意中心设在 500Hz 的共振峰区域。时间由图后方向前增加；(b)一个颗粒长为 1 毫秒的颗粒云的频谱，固定频率在 500Hz。注意频谱的宽度。
Time=时间　　kHz=千赫兹

由于在颗粒层面上每个颗粒可以变化，所以我们能把单个波形或多个波形的颗粒填入颗粒云中。比方说，单色调云(monochrome cloud)使用单一波形，而多色调云(polychrome cloud)有数个波形的随机混合。而转换色调云(transchrome cloud)则在一个云的时程内，在统计上从一个波形变化到另外一个波形。

对于积云(cumulus cloud)(图 5.19a，也可见图 5.11，第二列)而言，发生器将颗粒随机撒在高频带与低频带间。若将频带变窄成为一个很小的音程，就可

以产生具有音高的声音。也很容易做出许多种形态的滑音(glissandi)(图 5.19b)。另外一个设定法是层云(stratus cloud)(图 5.19c,也可见图 5.11,第三列),颗粒被限制在一个音高或几个音高上,做出和弦和音块(pitch clusters)。

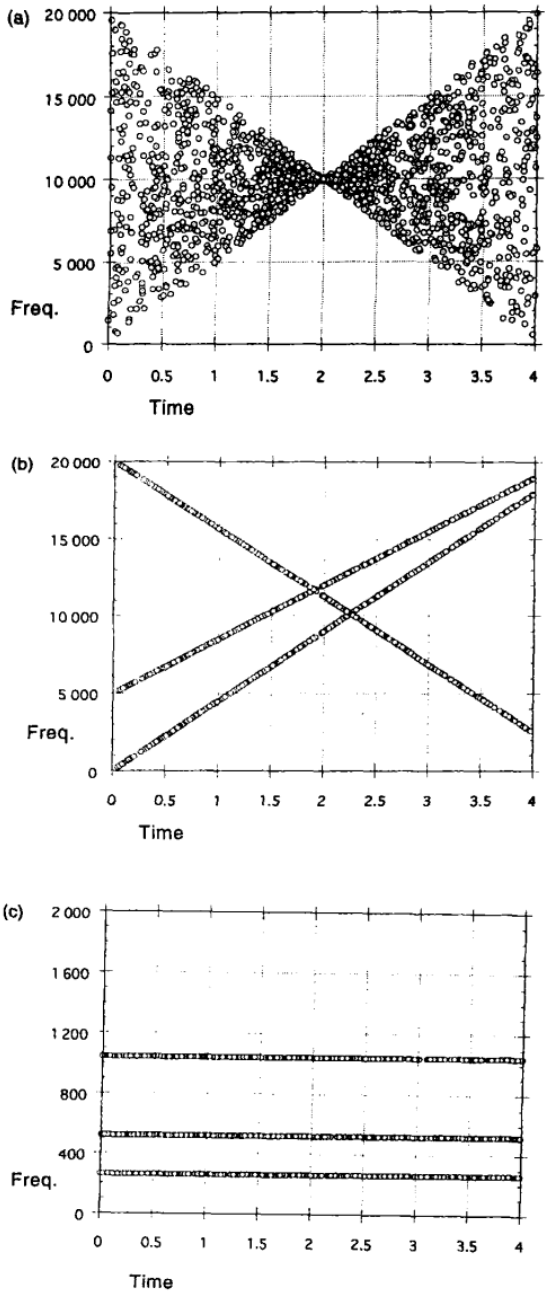


图 5.19 颗粒云的形态。(a)积云;(b)滑音;(c)层云。
Time=时间 Freq.=频率

颗粒密度与带宽参数结合，可以制造许多效果。当密度较小，不管带宽大小为何，都会造成类似点描的织体。当密度高时，窄带宽，可以做出带有音高及共振频谱的串流，而宽带宽（一个八度音或以上）可以得到巨大的声块。

最后，就如同所有粒式合成法一样，异步粒合成（AGS）的多个声道的空间定位可以增加颗粒质感。颗粒云的空间算法可以采用随机散布或在整个颗粒云事件时程内的左右偏位效果。

取样声音的时间颗粒（Time Granulation of Sampled Sounds）

录制取样声音的时间颗粒是将客观声音材料送入逻辑绞碎机——以新的排列方式释放带有新的微细节奏的颗粒。也就是说，粒式合成器读取小段的采样声源（由声音档案或直接由模数转换器），并将包络应用于该段。颗粒输出的顺序（它的延迟）取决于作曲家在设定上的选择。

时间粒式合成有三条途径：

1. 将储存的声音文件颗粒化，如音乐中的乐音、动物声或人的语音。

2. 对输入声音的持续实时粒式合成，或带有时间不规则性（time scrambling）（Truax 1987，1988，1990a，b）。

3. 对以不同速度回放的输入声音的持续实时粒式合成（Truax 1987，1988，1990a，b）。

第一种方式变化弹性最大，因为能以任何顺序从声音文件中选取颗粒。比方说，我们可抽取小鼓的、单一的、大的颗粒，复制成数百个颗粒的周期性序列，创造出小鼓的交替击鼓声（drum roll）（图5.20a）。另外，颗粒发生器也可以从较长的文件中任意抽取颗粒，如人声或几个音符，然后重新排列（图5.20b）。一种该技术的进一步应用，是随机取样数个声音文件，并交织这些取样颗粒，以创造多种音色的织体（图5.20c）。这些交织在一起的声音织体，根据其内部单一颗粒音色与音高的不同而形成很大的变化。

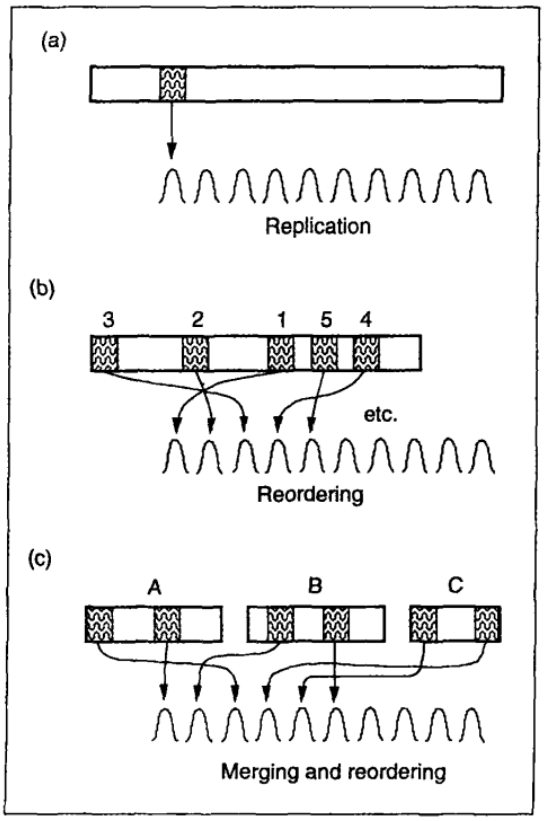


图 5.20 来自储存的声音文件的时间粒式合成之三种方法。(a)提取一个颗粒,使颗粒滚动出现;(b)由一个声音文件随机抽取并重新排序的颗粒;(c)由数个声音文件任意选取并重新排序的颗粒,颗粒不一定要按照顺序安排,且可以重叠。
Replication=复制 etc.=等等 Reordering=新排序 Merging and reordering=合并与重新排序

上述的第二个情况牵涉到对连续声音的实时粒式合成,此处计算机与延迟线(delay line)或取样窗(window)的作用类似,声音被延迟后成为不同的颗粒。(见第 10 章对延迟线与 tap 的描述。)此情况下,声音的颗粒化会造成频谱上的反作用,以可操控的方式将声音扭曲,并使声音丰富。

第三个情况除播放速度可由读取采样的速度控制参数来改变外,与第二个情况很相近。播放速度是在正常速度到慢速之间改变,即重复读取单一取样点。所以此方法可以看作是上述两种情况的综合。

粒式合成的评估(Assessment of Granular Synthesis)

粒式合成构成了只分享声音颗粒概念的微小声音技术体。傅里叶与小波分析中的颗粒描述方式是纯内部性的,它隐藏起来不被使用者所察觉。的确,这些方法的技术目的,是创造持续的幻觉、模拟似的信号处理。声音只有在不

正确的失真后——如叠加再合成中跃幅(hop size)过大时(见第 13 章)——才会出现颗粒感。A. Picciallli 和他的同仁将音高同步分析/再合成的表示方式说得更清晰。像准同步粒式合成〔如杜亚士(B. Truax)所发展〕已在许多不同平台上实现。

异步粒合成(AGS)已证明了它在创造早期技术所难以达到的声音上的价值。异步粒合成将声音颗粒散布于整个频谱上成为云状的形式。得到的结果往往是粒子声音复合体,它的作用可作为对数字振荡器所产生的较平顺无味的声音的衬托。颗粒云的时间变化结合可带来更戏剧化的效果,如蒸发(evaporation)、连接(coalescence)以及由云的相互交迭产生的变化。这些处理手法与在视觉领域中粒子合成(particle synthesis, Reeves 1983)创造的东西之间有明显的类似。视觉领域中的粒子合成是用来做出火、水、风、雾以及类似草的质感,这与异步粒合成技术上的声音效果(火在燃烧时的噼啪声、水流的汩汩声、风声、爆炸)很相似。最后,结合时间颗粒与卷积(convolution, Roads 1993a),粒式技法可以由纯合成技术转为声音变形应用技术。

减法合成(Subtractive Synthesis)

减法合成是利用滤波器来对一个资源声音的频谱塑形。当声源信号通过滤波器,会加强或减弱所选定的某部分频谱。在原始声源中的频率若很丰富,且滤波器很有伸缩性,减法合成不仅可以塑造许多新而无法分类的声音,而且也可以塑造许多自然声音的近似(如人声和真实乐器)。

此节的剩余部分介绍减法合成的主要工具——滤波器——并带到下节主题,减法分析/再合成技术。在第 10 章中,我们将更进一步介绍滤波器的内部操作,此处我们仅描述它们的作用。

滤波器简介(Introduction to Filters)

在字面上,滤波器一词所指的可以是任何信号处理动作(Rabiner et al. 1972)!但是,此词最常指的是增强或减弱部分频谱区域的装置,即是此处所要说明的用法。这些滤波器的工作采用下述方式之一,或两个方式都采用:

- 稍稍延迟输入信号(约一个或数个取样周期)的拷贝,并将延迟信号与新的原始信号相加(图 5.21a)。
- 延迟输出信号,并将之与输入信号相加(图 5.21b)。

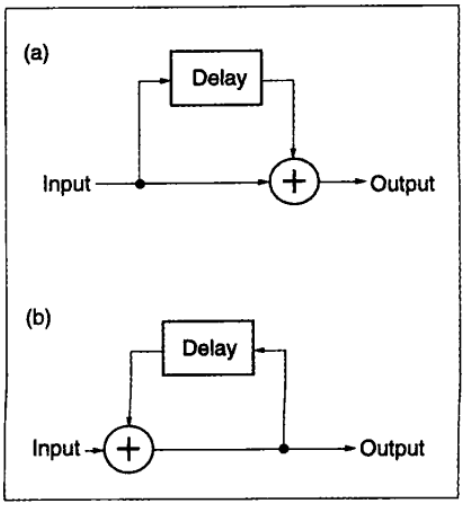


图 5.21 是两种基本的数字滤波器。(a)将输入延迟，并与其他信号相加(FIR)(前馈，feedforward)；(b)将输出延迟，再加回(IIR)(反馈，feedback)。
Delay=延迟 Input=输入 Output=输出

虽然图 5.21 中信号结合以加法(+)表示，但同样能够以减法(−)表示之。不管是哪一种，原始与延迟信号的结合会造成带有不同的频谱的新波形。将延迟增多，或以不同的结合方式混合加减法中的和与差，就能建立许多种类的滤波器。

下一步我们讨论各种滤波器的性质。因为我们的主要目的是在音乐用途上使用减法合成，所以将不会详加介绍数字滤波器的建构方式，或滤波器理论的数学理论。第 10 章有对这个浩瀚领域的基本介绍。(见 Moorer 1977 and Moore 1978a, b。)具有工程背景的人，也可以参阅 Moore (1990)、Smith (1985a, 1985b)、Oppenheim 和 Willsky (1983)、Rabiner 和 Gold(1975)以及 Oppenheim 和 Schafer(1975)等学者的相关资料。

滤波器种类与响应曲线(Filter Types and Response Curves)

要表示不同形态滤波器的性质，最重要方法之一是绘出它的振幅—频率响应曲线(amplitude-versus-frequency response curve)。一般音频器材的规格文献中，都会有一张“频率响应图”(frequency response)。此词是振幅—频率响应的缩写。最准确的频率响应是一条直线，它标明一个横穿整个频谱的线性(linear)，即平直的(flat)振幅。这意味着在此器材范围内的所有频率，不会带有任何的增强或衰减。图 5.22a 显示典型的高级音频系统接近平直的频率响应。现在我们看到的上限值为 25kHz。对于高级模拟音频组件，如前置放大器和放大器，频率响应可以延伸高达 100kHz。如第 1 章所解释的，数字音频系统的频率限制取决于其取样率。

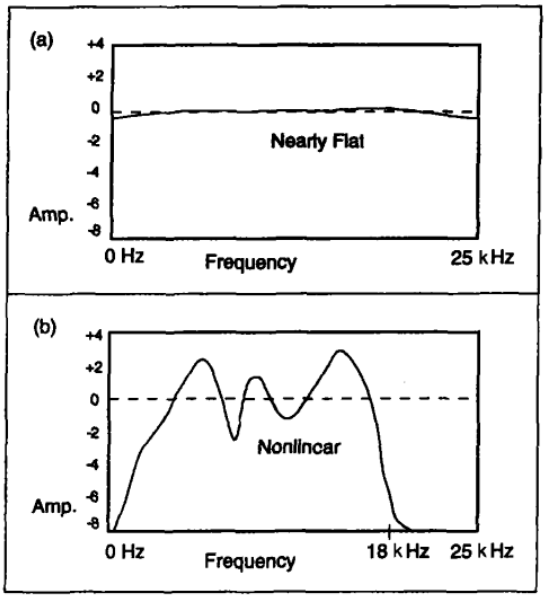


图 5.22　振幅对应频率的响应，俗称“频率响应”。纵向轴是用分贝来表现的振幅，横向轴为频率。(a)近似水平振幅的响应；(b)非线性振幅响应。
Frequency=频率　Amp.=振幅　Nearly Flat=近似平直　Nonlinear=非线性

实际器材通常不会有完美的平直响应。图 5.22b 是小音箱的非直线性系统频率响应。我们可以如下描述此响应：从 100Hz 到 16kHz 频率间的＋3dB 到－2.5dB 的变化。这说明了此音箱最多增强某些频率达＋3dB，而减弱某些频率达－2.5dB。小于 100Hz 或大于 16kHz 的频率响应都急速衰减。由于它改变输入信号的频谱，所以此音箱的作用相当于滤波器。

每种滤波器有它本身独特的频率响应。图 5.23 绘出四种典型的基本滤波器的频率响应曲线：低通(lowpass)、高通(highpass)、带通(bandpass)、带阻或带除(notch)滤波器。

图 5.24 显示的平层滤波器(shelving filters)，会将在某个阈限值以上或以下的频率增强或全部删除。平层滤波器的诸多类型的名字有时令人混淆，因为高平层(high shelving)滤波器设定为截止高频时，作用与低通滤波器相同。而低平层(low shelving)滤波器设定为截止低频时，其作用与高通滤波器相同。

滤波器的重要特性就是截止频率(cutoff frequency)。图 5.23 与 5.24 显示低通与高通滤波器的截止频率。习惯上，这是频率范围中的一个点，在这个点上，滤波器将信号削减至它的最大值的 0.707。为什么是 0.707 呢？在截止频率上的信号功率与信号振幅的平方成正比，由于 $0.707^2=0.5$。所以，截止频率又称为半功率点(half-power point)。另一个术语是 3 分贝点(Tempelaars 1977)。这是因为 0.707 与 1.0 的比例接近于－3dB。

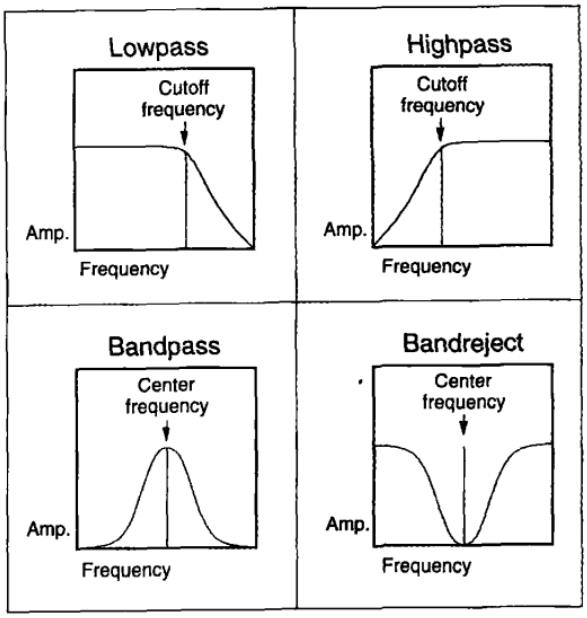


图 5.23 一般滤波器的四种形态。
Lowpass=低通 Cutoff frequency=截止频率 Amp.=振幅 Frequency=频率
Bandpass=带通 Center frequency=中心频率 Highpass=高通 Bandreject=带阻

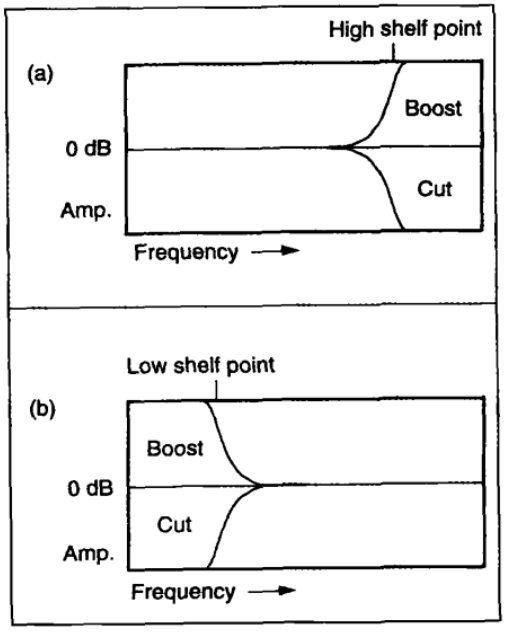


图 5.24 平层滤波器(a)高平层滤波器。在平层点以上,信号可以被增强或删除。如果信号被删除,那么高平层滤波器就相当于低通滤波器;(b)低平层滤波器,在平层点以下,信号被增强或删除。
High shelf point=高平层点 Low shelf point=低平层点 Amp.=振幅 Frequency=频率
dB=分贝 Boost=增强 Cut=截断

在半功率点以下，被减弱的频谱部分称为滤波器的阻带（stopband）。在半功率点以上，则称为滤波器的通带（passband）。在带通滤波器的高、低截止频率间的差值称为滤波器的带宽（bandwidth）。带通滤波器的中心频率是强度的最大值；而在带阻滤波器的中心频率则是强度的最小值。

在理想的锐利滤波器（sharp filter）中，截止频率就像是一堵墙：任何在此之外的频率都会被最大限度地减弱，将频率响应单纯地分为带通与带阻（图 5.25a）。实际滤波器中，滤波器的斜率不是直接通向截止频率的直线〔在频率响应中还存有波纹（ripple）〕，而带通与带阻之间的这段区域被称为过渡频带（transition band）（图 5.25b）。

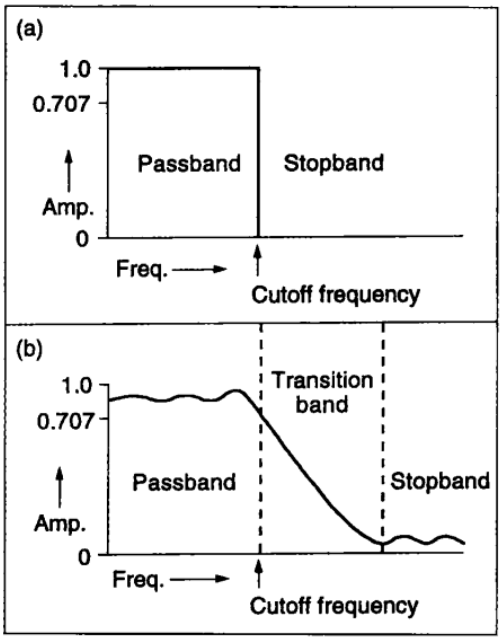


图 5.25　理想与非理想滤波器。(a)理想滤波器，其频率可以被干净地区分成带通与带阻，而它的截止是直角方形的；(b)非理想（实际）滤波器，响应曲线上会有波纹，且在带通与带阻间有或多或少倾斜的过渡频带。
Passband=通带　Stopband=阻带　Cutoff frequency=截止频率　Amp.=振幅　Freq.=频率
Transition band=过渡频带

滤波器斜率的倾斜程度，通常以每八度音之间增强或减弱分贝（dB）数来定义，可写为 dB/octave。比方说，6 dB/octave 的低通滤波器的衰减（rolloff）会较平顺，而 90dB/octave 的滤波器则将频谱锐利的截断（图 5.26）。

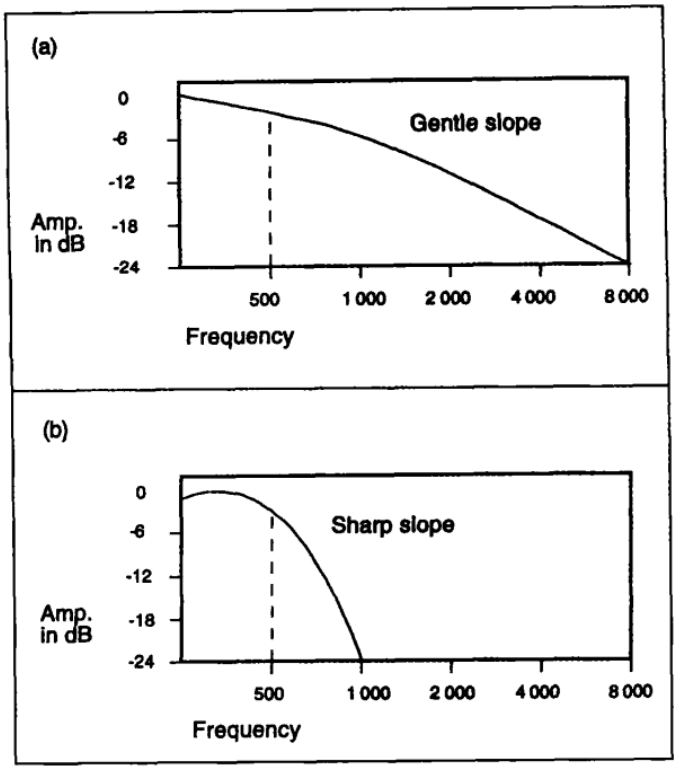


图 5.26 滤波器的斜率。(a)较缓的斜率;(b)较陡的斜率。
Frequency=频率 Amp.=振幅 In dB=以分贝表示 Gentle slope=平缓斜率
Sharp slope=陡峭斜率

要依照音乐不同情况来选择较锐利或较平顺的斜率。比方说,较锐利的带除滤波器(notch filter)可以用来消除处在特定频率中央的音,而使用较平顺的低通滤波器是降低高频段中背景噪音的最谨慎方式。

滤波器的 Q 值与增益(Filter Q and Gain)

许多带通滤波器都有 Q 值的控制旋钮(无论是软件或硬件)。Q 值的直觉定义就是它反映带通滤波器的共振程度。图 5.27 显示不同程度 Q 值的滤波器。当 Q 值较高,会造成较窄的内曲线,频率响应在峰值(共振点)频率周围特别陡峭。如果通过信号能量在它的中心频率附近,出现高 Q 值滤波器,此滤波器会在共振频率上回响(ring),也就是说,当信号通过时会引发信号振荡。

带通滤波器的 Q 值可以用中心频率与它的 −3 分贝点(截止点)带宽展开之间的比例关系来作精确的定义:

$$Q=\frac{f_{center}}{f_{highcutoff}-f_{lowcutoff}}$$

f_{center} 是滤波器的中心频率，$f_{highcutoff}$ 是上方的 3dB 点。而 $f_{lowcutoff}$ 则是下方 3dB 点。注意，当中心频率为常数时，调整 Q 值相当于调整带宽。这里是一个滤波器的 Q 值计算：滤波器中心频率为 2 000Hz，3dB 点分别为 1 800Hz 及 2 200Hz。此滤波器的 Q 值则是 2 000/(2 200－1 800)＝5。像这样的高 Q 值共鸣滤波器适合产生打击乐的声音。我们可用连续脉冲波送入高 Q 值共鸣滤波器，来产生如塔布拉手鼓(tablas)、盒梆(wood blocks)、响棒(claves)这样有音高的鼓或玛林巴的效果。

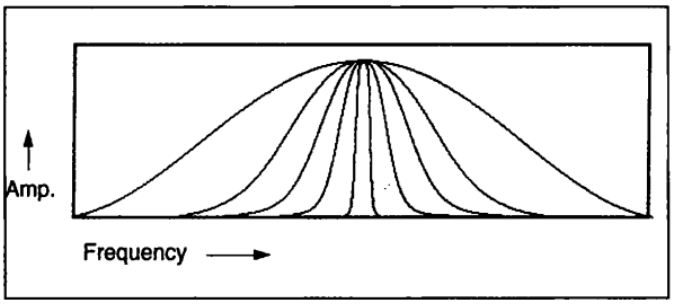


图 5.27　不同 Q 值下的滤波器设定。较高的 Q 值对应到较窄的频率响应。增益值维持相同。
Frequency=频率　　Amp.=振幅

增益是带通或带阻滤波器的另一个性质。它指的是频带增强或减弱的量。它显露出在响应曲线中频带的高度(或深度)(图 5.28)。当信号输入高 Q 值滤波器时，必须要小心避免在共振频率上的增益(峰值的增益值)造成系统过载，而引起信号失真。许多系统在其滤波器内部中有增益补偿(gain-compensation)电路，以避免发生这种过载。

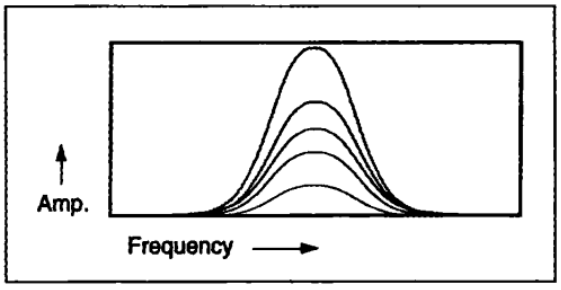


图 5.28　采用同一个滤波器的不同增益系数。带宽与 Q 值保持相同。
Frequency=频率　　Amp.=振幅

有一种特别的带通滤波器称作固定 Q 值滤波器。要保持固定的 Q 值，此滤波器必须依照中心频率调整带宽。比方说当中心频率是 30Hz，Q 为 1.5(或

3/2)时,因为 30/20=1.5,所以带宽是 20Hz。但是如果我们将滤波器调整为 9kHz,并将 Q 保持在 1.5,那带宽就必须是中心频率的 2/3,也就是 6 000Hz。图 5.29 显示以线性和对数坐标绘出的两个固定 Q 值滤波器的曲线。在线性坐标上(图 5.29a),以 30Hz 为中心的滤波器看起来非常窄,而以 9kHz 为中心的滤波器的曲线就看起来非常宽了。而在对数坐标上,两个滤波器具有相同的形状(图 5.29b)。

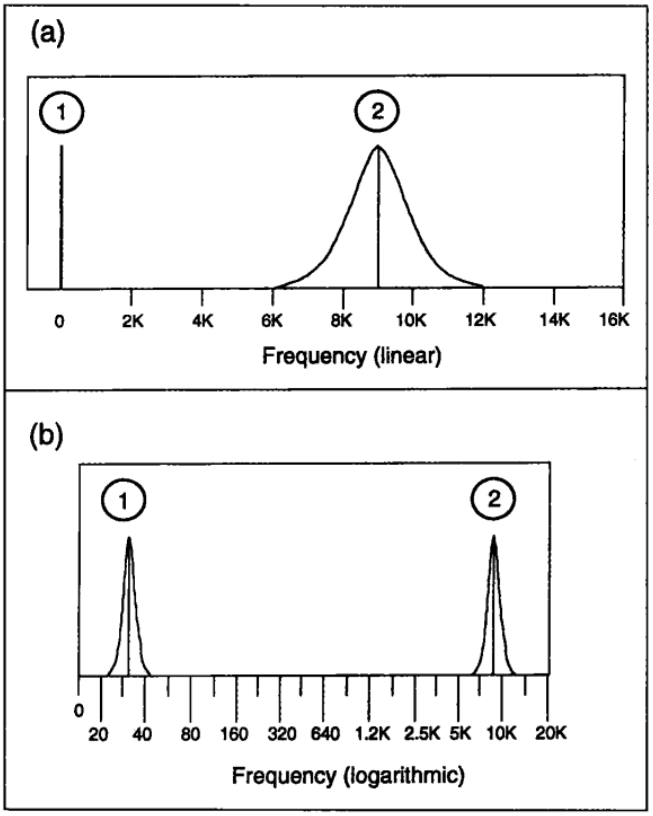


图 5.29 画在线性及对数坐标范围之内的相同常数 Q 值的滤波器。滤波器 1 的中心频率在 30Hz,带宽则是由 20Hz 延伸到 40Hz。滤波器 2 的中心频率在 9kHz,带宽则是由 6kHz 延伸到 12kHz。(a)线性坐标;(b)对数坐标。
Frequency=频率 Linear=线性的 Logarithmic=对数的

固定 Q 值滤波器具有音乐性质,因为其所跨的频率音程并不会因为中心频率改变而改变。比方说,以 A440Hz 为中心,Q 值为 1.22 的固定 Q 值滤波器与以 A880Hz 为中心,Q 值为 1.22 的滤波器所跨的音程是相同的(分别为 C260 到 D620,与 C520 到 D1240)。

滤波器组与均衡器(Filter Banks and Equalizers)

滤波器组(filter bank)，是平行送入同样的输入信号的一组滤波器(图 5.30)。每个滤波器皆是设定在特定的频率上的窄频带通滤波器。经过滤波的信号通常会组合在一起来形成输出声音。当每个滤波器有自己的输入信号水平来控制滤波器组时，该滤波器组就称为频谱成形器(spectrum shaper)，因为每个控制都可以直接改变输入信号的频谱。频谱成形器可以用来强化某些频带，也就相当于减弱另一些频带。

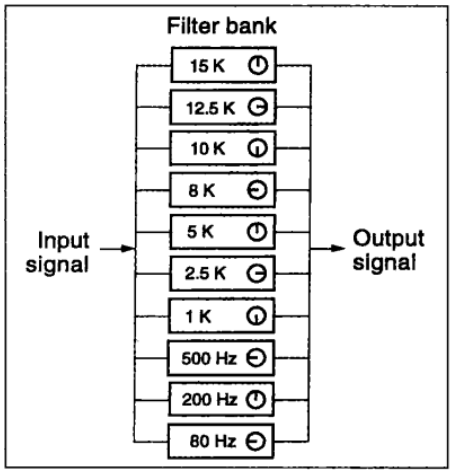


图 5.30　是一个有控制旋钮(增强或减弱)并与每个频率库相连的十等级频谱成形器。
Filter bank=滤波器组
Input signal=输入信号
Output signal=输出信号

频谱成形器的另一个名字是均衡器(equalizer)。这个滤波动作称为均等化(equalization)。字面上，均等化一词来自其原始应用方式，也就是补偿电话信道或播音系统(public address systems) (Fagen 1975)的频谱响应上的不规则处。比方说，一个演奏厅存在频率在 150Hz 处的隆隆共振声，就可以用电子均衡器将此频段减弱，从而抵消演奏厅对此频段的强化。

图示均衡器的控制器反映出频率响应曲线的形状(图 5.31a)。每个滤波器有一固定中心频率、固定带宽(通常是八度音的三分之一)以及固定 Q 值。(有些设备可以在不同 Q 值间切换。)每个滤波器的响应可以用线性推杆来增加或减弱该频带。图 5.31b 显示了这样的滤波器可能的频率反应。

参数式均衡器(parametric equalizer)用到滤波器上较少，但每个滤波器的控制更有弹性。典型的安排是有平行的三到四个滤波器，使用者可以独立地调整每个滤波器中心频率、Q 值及增益或衰减值。半参数式均衡器则有一个固定的 Q 值。

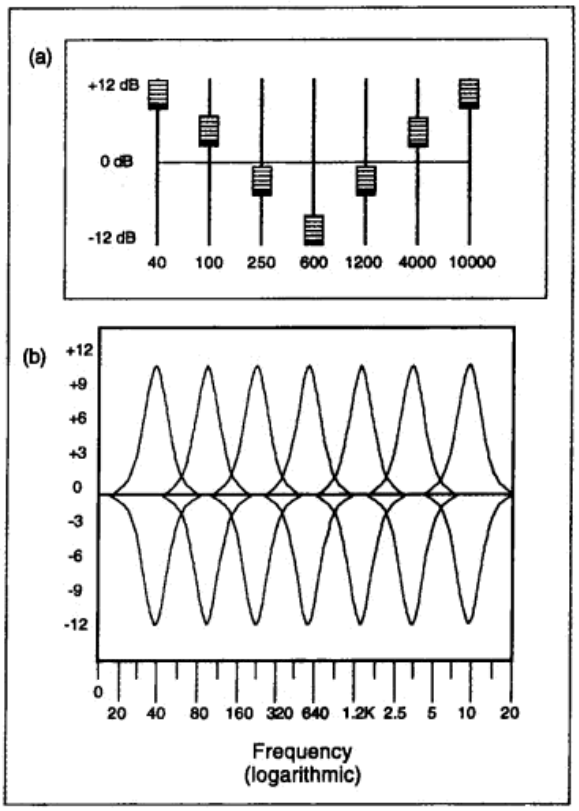


图 5.31 图像式均衡器。(a)七个频带的图像均衡器，带有设在任意强度上的线性电位器。(b)七个频带图像式均衡器的可能频率响应曲线。
Frequency=频率 Logarithmic=对数的

梳状滤波器与全通滤波器(Comb and Allpass Filters)

在第 10 章与第 11 章讨论这两个滤波器之前，我们先讨论这两个滤波器的优点。在频率响应上带有数个很陡的曲线的滤波器被称作梳状滤波器。图 5.32显示了两种类型的梳状滤波器的频响曲线，一个是频率反应带有很深的凹口，另一个则是带有锐利的波峰。此“梳形”一词明显地来自其曲线的形状。第 10 章将对此滤波器与其音乐应用作更完整的解说。

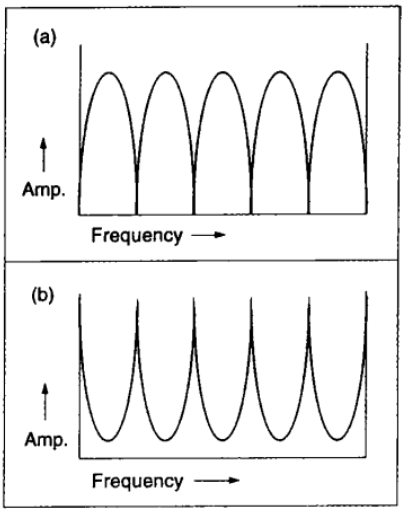


图 5.32　梳状滤波器频率响应曲线。(a)FIR梳状滤波器;(b)IIR 梳状滤波器。(详见第 10 章对 FIR 与 IIR 的解释。)
Frequency=频率　　Amp.=振幅

最后一个要讨论的滤波器是全通滤波器(allpass filter)。对于稳态(不改变的)输入声音,全通滤波器将以相同增益值输出所有频率——这也就是其名称的由来。全通滤波器的用途在于制造与频率相关的相位偏移。所有滤波器在增强或减弱某些频率时,都会造成某种程度的相位偏移,而全通滤波器的主要效果就是在偏移相位。如果输入信号为非稳态信号,那么全通滤波器由于这种与其频率相关的相位偏移的效果,可以对声音进行粉饰(color)。这种粉饰在短暂声音上尤其明显,其相位关系对于音质有绝对影响力。

全通滤波器可用来校正不想要的另一滤波器产生的相位偏移。全通滤波器也可以用在音乐信号处理上。在此,全通滤波器可对输入信号带来时变、频率上的相位偏移,从而产生声音的丰富感。全通滤波器是数字混响的建构部分之一。第 10 章与第 11 章将讨论全通滤波器的应用。

时变减法合成(Time-varying Subtractive Synthesis)

滤波器可以是固定的,或随时间改变的。在固定滤波器中,所有滤波器性质都是预先定义好的,不会随时间改变。这种形式是传统音乐录音时的典型情况,录音工程师在音乐作品的一开始就已设定好每个信道的均衡。

时变滤波器有许多音乐应用空间,尤其对于旨在超越传统乐器限制的电子与计算机音乐上。随时间改变的 Q 值、中心频率及衰减程度会使带通滤波器产生大量丰富的声音色彩,尤其是当送入经过滤波的声音同时又随时间改变时,更能带来难以想象的声音变化。时变滤波器的一个例子是在混音器上的参数均衡器。混音工程师在混音程序的任何时间都可以改变 Q 值、中心频率以及增强、衰减量,或用设定好的参数令其自动改变。

时变减法合成的最重要的例子是 SYTER——在 20 世纪 70 年代晚期由 Jean-François Allouis 及其同仁(Allouis 1979, Allouis and Bernier 1982)在巴黎 Groupe de Recherches Musicale(GRM)开发的数字信号处理器。目前,许多 SYTER 软件已被移植在个人计算机的信号处理卡上执行(INA/GRM 1993)。

SYTER 被作曲家当作时变减法合成器的引擎来使用,如 Jean-Claude Risset 在他 1985 年的作品《Sud》(Wergo recording 2013-50)。如运行 B. Maillard 所撰写的软件,可使 SYTER 系统实时控制数十个带有动态参数变化的高 Q 值带通滤波器。这些滤波器也可以通过傅里叶声音分析的生成数据来驱动(见下节对减法分析/再合成的介绍)。当像水声或风声这样的全频带声音通过此系统时,共振滤波器会以和弦和音串的方式回响(rang),也可以产生丰富的梳状滤波器以及相位的变化效果(见第 10 章)。

减法分析/再合成(Subtractive Analysis/Resynthesis)

与加法合成一样,减法合成的能力也可因加上分析阶段后再提升。基于减法滤波器,而非加法振荡器的分析/再合成系统能逼真地模拟任何声音。在实践上,在减法分析/再合成中,大多数资料分析与数据缩减技术是用于语音合成上的。因此,大多数研究也主要集中在语音领域(Flanagan et al. 1970, Flanagan 1972)。

在音乐上的减法分析/再合成的研究焦点,已由语音用途的工具(如线性预测编码 linear predictive coding,在此章后半部分讨论)延伸到宽频带的乐音声音的领域。

声码器(The Vocoder)

原始的减法分析/再合成系统是声码器,是在纽约 1936 年的世界博览会中由说话机器人展示出来的(Dudley 1936, 1939a; 1939b, 1955; Dudley and Watkins 1939; Schroeder 1966; Flanagan 1972)。传统的模拟声码器有两个阶段。第一阶段是一组分布在音频带宽上的固定频率带通滤波器。每个滤波器的输出连接到包络侦测器(envelope detector)上,产生与滤波器所跟踪频率能量成比例的电压。

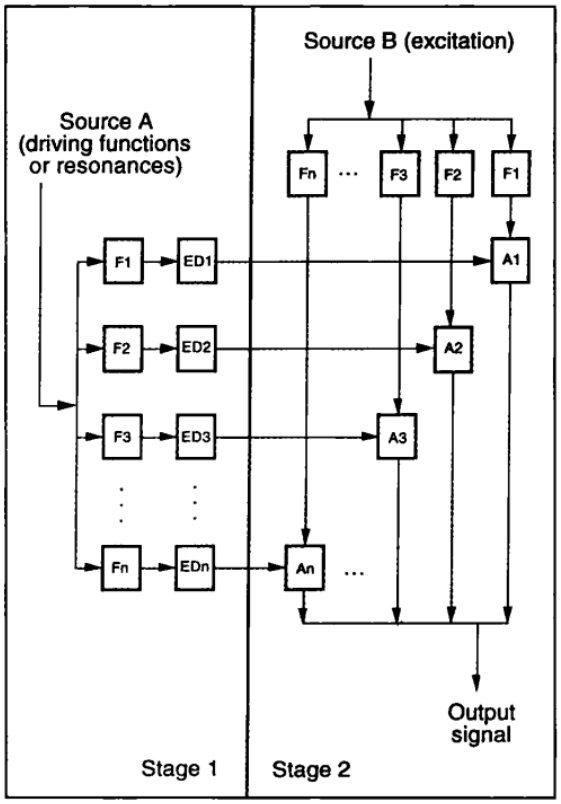


图 5.33　声码器。第一阶段是分析部分，第二阶段是合成部分。"F"代表滤波器，"ED"代表包络侦测器，"A"则代表由电压控制的放大器。放大器的增益由包络侦测器输入的控制电压来控制。相同的架构亦在数字形式下实现。
Stage=阶段　Source=声源　Driving functions or resonances=驱动函数或共振
Excitation=激励　Output signal=输出信号

声码器的第二个阶段是与第一阶段相同的一组带通滤波器。将相同的输入信号送进所有滤波器，每个滤波器的输出信号送到它自己的电压控制放大器上(voltage-controlled amplifier，VCA)。所有 VCA 的输出加在一起成为一个输出信号。第一阶段的滤波器与侦测器产生控制信号(也叫做驱动函数 driving functions)，该控制信号决定来自声码器第二阶段滤波器的音频信号强度。

参看图 5.33，声源 A 是共振峰频谱赖以产生的信号，如歌唱的人声。如果我们要追踪此频谱的轮廓，便会形成频谱包络(spectral envelope)或共振曲线(resonance curve)。声源 B 是激励函数(excitation function)。激励函数通常是一个宽频带信号，如白噪音或连续脉冲。声码器的输出由声源 B 的激励函数组成，该激励函数带有声源 A 人声时变频谱的包络。图 5.34 以图像方式解释作用在激励函数上的共振滤波器的处理方法。

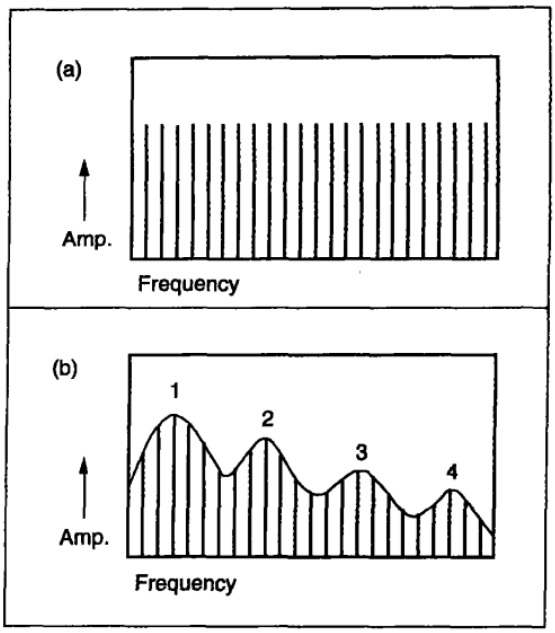


图 5.34 共振滤波器作用在激励函数上的效果。(a)激励函数的简化图标，如由开放声带所产生的频谱，具有多个相同强度谐波的嗡嗡声；(b)有四个共振峰 1、2、3、4 的元音频谱的简化图标。
Frequency=频率 Amp.=振幅

最初的声码器研究方向是为合成语音作资料缩减。对于缓慢移动的驱动函数之数据率和信道需求，它的确比原始信号的数据率和信道需求小了很多。

在音乐应用上，将驱动函数(或共振)与激励函数分离，意味着节奏、音高及音色都可独立控制。比方说，作曲家可以改变歌声的音高(通过改变激励函数的频率)，但保留人声的原始频谱清晰度。通过在整个时段上延伸或缩短驱动函数，可以在不改变音高或影响共振结构的情况下减慢或加速一段朗读文字。

线性预测编码(Linear Predictive Coding)

线性预测编码(LPC)或线性预测(Linear Prediction)，是一个在语音与音乐应用上大量采用的减法分析/再合成方法(Atal and Hanauer 1971; Flanagan 1972; Makhoul 1975; Markel and Gray 1976; Cann 1978, 1979, 1980; Moorer 1979a; Dodge 1985; Lansky 1987; Lansky and Steiglitz 1981; Hutchins 1986a; Lansky 1989; Dodge 1989; Depalle 1991)。线性预测编码将一个声音，如人声，分析成资料缩减的形式，再重新合成一个近似版本。线性预测编码语音合成相当有效率，因其需要的资料远比采样语音少很多，LPC 的廉价集成电

路在 1980 年早期开发，并内建在不是很贵的可说话玩具内（Brightman and Crook 1982）。

从作曲家的角度而言，线性预测编码技术的能力来自可编辑的分析资料以及重新合成出不同版本的原始输入信号。线性预测编码实现了一种声码器，也就是将激励信号与共振分离，使得独立操纵节奏、音高以及音色成为可能，并允许一种交互合成（Cross Synthesis）（将在之后讨论）。

在语音内，声带会产生嗡嗡振动的激励函数，而声道的其他部分会去过滤声音，从而产生共振。激励脉冲的频率决定输出声音的音高。既然线性预测编码让使用者独立操纵激励函数，我们就可以改变激励频率，比方说，可将说话的声音转为歌唱。

什么是线性预测？（What is Linear Prediction?）

线性预测这个模糊的名词，其出处是由于，在系统中的频谱分析部分，输出取样是由滤波器参数（系数）和先前取样的线性结合来预测的。预测算法（prediction algorithm）试着在现有取样点之外的位置找出取样，也就是说，一组取样点的任何推算（extrapolation）都是预测。预测的内部是有出错的可能性的，所以预测算法中总是包含错误的估算部分。

简单的预测器仅是延续最后的取样与该最后取样前面的取样之间的斜率（图 5.35）。这种类型的预测可将更多采样纳入计算，而变得更复杂。如果已知真实信号值（在线性预测编码 LPC 中已知），可将实际的信号值与预测采样值间的错误或差值纳入计算。由于预测器会计算时间延迟取样的相加之和与差值，所以也可将预测器看作是一个滤波器——一个描述它现在处理的波形之滤波器。（见第 10 章对数字滤波器的讨论。）

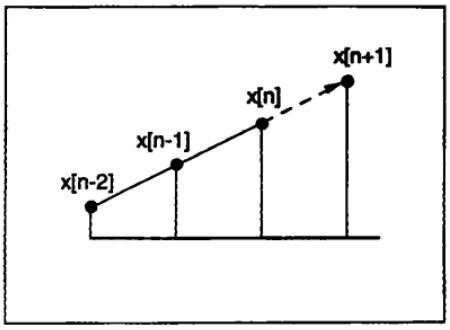


图 5.35　线性预测将一组取样点推算出来。

如果将这些在整个时间段内的滤波器系数有规则地记载下来，颠倒过来，接着用丰富、宽频带的声音驱动结果滤波器，我们应该能得到与原始输入信号

的时变频谱很接近的信号。所以，这种预测的一个“副作用”是预测输入信号的频谱。这是重点之所在。但频谱预测仅是线性预测编码分析的一个阶段，LPC的其他阶段则用来表示音高、强度以及判断声带音/非声带音(voice/unvoiced)的部分。这些将在后续节中简述。

线性预测编码分析(LPC Analysis)

图 5.36 显示了线性预测编码分析的略图。线性预测编码分析分成四个部分：(1)频谱的共振峰分析。(2)音高分析。(3)振幅分析。(4)决定一个声音是否有发声(voiced)（有音高的 pitched)或是未发声(unvoiced)(噪音性质的)。每个分析阶段都以帧(frame)为单位来分析。一个帧相当于一个信号的快照，LPC 分析的典型帧频(frame rates)约在每秒 50 到 200 个帧。

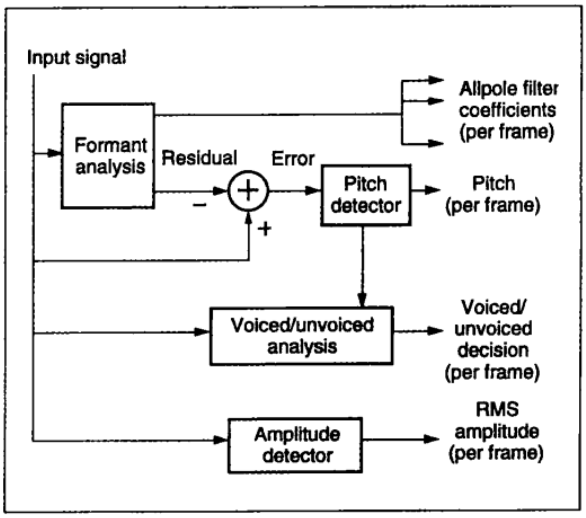


图 5.36 LPC 分析的四个阶段。频谱(共振)分析、音高侦测、声带音/声分析以及强度侦测。
Input signal=输入信号 Formant analysis=共振峰分析
Allpole filter coefficients (per frame)=全极点滤波器系数(每帧) Residual=余冗 Error=错误
Pitch detector=音高侦测 Pitch (per frame)=音高(每帧)
Voiced/unvoiced analysis=声带音/非声带音分析
Voiced/unvoiced decision (per frame)=声带音/非声带音判定(每帧) Amplitude detector=振幅侦测
RMS amplitude (per frame) =RMS 振幅(每帧)

滤波器预测(Filter Estimation)

下几个段落将概述 LPC 分析，但我们先从 LPC 分析中所使用的滤波器词汇开始。工程师根据滤波器的极点(poles)与零点(zeros)位置来说明带通或带阻滤波器(Rabiner and Gold 1975)。我们不会在这里深入说明极点—零点图(pole-zero

diagrams)(可参阅任何信号处理文献),我们此处仅将滤波器的极点当作共振点——频谱图上的共振峰或共振区域,相对的,零点则是频谱上的空点(null point)与波谷处。

当滤波器有数个平顺的峰点时,称作全极点滤波器(allpole filter)。此种滤波器是线性预测编码(LPC)的特征,它根据数个共振峰模仿频谱。这样的模型是许多种人声及某些乐器的合理近似模拟方式。

如前所述,线性预测或自动回归分析(autoregressive analysis)(见第 13 章)一次接受多个采样点,使用最近的取样点作为参考。它试着由滤波器的加权系数之和以及之前的取样来预测现在的取样点。作为预测的"副作用",运算法将反向滤波器调整为输入信号的频谱。全极点滤波器的倒转是全零点滤波器,会在通过的信号频谱上造成数个波谷。

线性预测编码(LPC)分析器会近似模拟最终合成所需要的滤波器之反向。如果这近似模拟做得很好,线性预测的结果应该就是激励信号(图 5.37)。换句话说,反向滤波器抵消了声音频谱包络的效果。此近似模拟永远不会是完美的,所以会有称作余冗(residual)的信号,也就是激励信号(一连串脉冲)加上噪音。LPC 频谱分析的目的是要将余冗最小化。

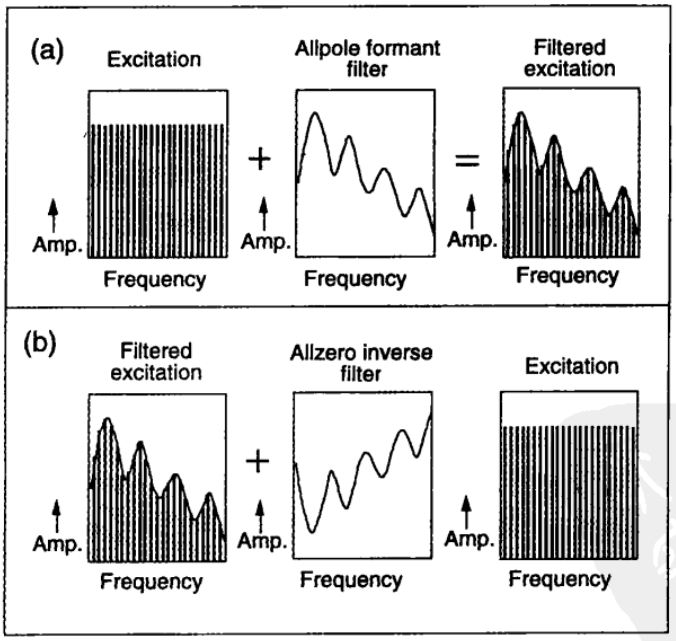


图 5.37　在理想状态下共振峰与反向共振峰滤波器的关系。(a)共振峰滤波器的结果;(b)反向共振峰滤波器的结果。

Excitation=激励　Allpole formant filter=全极点滤波器　Filtered excitation=滤波后的激励
Allzero inverse filter=全零点滤波器　Frequency=频率　Amp.=振幅

一旦找到了适宜的反向滤波器，该反向滤波器是造就一个再合成滤波器的自反转。在数学上，将滤波器反转非常直接(Rabiner and Gold 1975)；只要将所有滤波器参数的正负号相反，并将之用在输出信号上，而非输入信号。滤波器会由 FIR 转为 IIR 滤波器(见第 10 章)。第 13 章有更多 LPC 滤波器内部分析的讨论。工程上的讨论可见(Markel 1972, Makhoul 1975, Moore 1990)。

读者可能会有这样的疑问：线性预测编码(LPC)怎么知道输入任意一个信号时，它的激发函数是什么？答案是：它的确不知道。它假设激发信号不是带有音高的连续脉冲，就是白噪音。这些假设在语音与某些乐器的近似模拟上效果很好，但它不是对所有声音都有效的模式。所以，LPC 法在声音的再合成中通常会留下人工的痕迹。有些改进的 LPC 分析方法加入了多脉冲集群(multi-pulse cluster)，而不是只使用单一音高周期的脉冲，此脉冲集群(脉冲的强度与间距)是由分析数据中得来的(Atal and Remde 1982)。这可以帮助减少 LPC 再合成时的人工质感。

音高与振幅分析(Pitch and Amplitude Analysis)

应用在线性预测编码(LPC)中的音高侦测技术有很多，这将在第 12 章中描述。要根据执行方式的不同来采用具体的技术方法。图 5.36 显示了尝试从余冗信号中预测音高的一个方案。

有几种定义每个帧(frame)的振幅的技术。一个典型的方法是，针对帧所描述的输入波形，在帧域上，将它作为一个平均值来计算。

声带音/非声带音的判定(Voiced/Unvoiced Decision)

音高侦测后，LPC 分析试着为每个帧(frame)决定声带音/非声带音。此判定十分重要，因为它将决定在再合成时声音是否带有音高。声带音(voiced)的声音带有音高，如元音 a、e、i、o、u，由声带的振动所发出；而非声带音(unvoiced)的声音像是齿擦音 s 和 z，爆发音 t 与 p 或是摩擦音 f 等子音。除了声带音与非声带音外，第三种激励的种类是“混合声”(mixed voice)，将带有音高的音及噪音结合在一起，如“azure”中的|z|。

分析管乐器声音时，声带音/非声带音的数据通常代表送气量，对类似小提琴的声音时，声带音/非声带音则指擦弦噪音。在再合成时，会通过带音高的连续脉冲模仿声带音，而非声带音则是以白噪音来模拟。当然两者都会经过滤波处理。

声带音/非声带音的判定很难完全自动化(Hermes 1992)。在音乐上使用的 LPC 系统中,分析会先做出第一次判定,但需要作曲家对帧(frame)作调整(Moorer 1979)。第一次判定是采用各式各样的探索。图 5.36 显示音高侦测的结果将送入声带音/非声带音判定。比如,如果分析无法判定输入信号的音高,那它将会产生很大的音高预测误差。当这个误差(经规格化于 0 到 1.0 之间后)大于某个值(约 0.2)时,这个时点上的声音很可能是如辅音这样的嘈杂的非声带音。余冗的平均振幅是另外一个线索。如果余冗的振幅与原始输入信号相比时较低,此信号可能是声带音。

分析帧(Analysis Frames)

分析阶段的结果是一连串的帧(frame),它们代表经大幅数据缩减后的输入信号。每个帧(frame)由以下的参数列表描述:

余冗声音的平均振幅。

原始声音的平均振幅。

两振幅间的比例(用来帮助判定该帧是声带音还是非声带音)

音高预测

帧的时值

全极滤波器的系数(每个极点会在频谱上形成一个共振峰)

图 5.38 显示“sit”这个词的帧(frame)数据(Dodge 1985)。为清晰起见,省略了滤波器系数。

ERR 列是判定该帧是否为声带音的重要线索。当 ERR 较大(大于 0.2),通常是非声带音帧。但是因为很难完全自动化判定声带音/非声带音,所以应再次检查此标记。注意 ERR 值在“S”与“I”的边界上的明显改变。RMS1 与 RMS2 值是“I”与“T”边界间的较好判定标记。

Phoneme	Frame	RMS2	RMS1	ERR	PITCH	DUR
S	197	813.27	1618.21	0.252	937.50	0.010
	198	1189.36	2090.14	0.323	937.50	0.010
	199	553.71	838.38	0.436	937.50	0.010
	200	742.59	1183.17	0.393	937.50	0.010
	201	1041.95	1918.33	0.295	123.95	0.010
	202	1449.16	2677.06	0.293	123.95	0.010
	203	1454.84	2920.50	0.248	937.50	0.010
	204	1430.03	2496.88	0.348	937.50	0.010
	205	1570.88	2981.21	0.277	142.84	0.010
	206	1443.27	2665.22	0.293	142.84	0.010
	207	1172.67	2150.50	0.297	150.00	0.010
	208	1200.73	2080.20	0.333	150.00	0.010
	209	1095.51	2055.25	0.284	116.26	0.010
	210	1260.36	2408.14	0.273	116.26	0.010
	211	1105.17	2293.05	0.232	937.50	0.010
	212	809.10	1659.80	0.237	937.50	0.010
	213	428.20	784.93	0.297	250.00	0.010
I	214	419.45	3886.15	0.011	250.00	0.010
	215	925.86	6366.20	0.021	208.32	0.010
	216	746.28	8046.81	0.008	208.32	0.010
	217	829.82	8277.42	0.010	192.29	0.010
	218	754.64	8049.50	0.008	192.29	0.010
	219	771.84	8001.70	0.009	197.35	0.010
	220	726.81	7955.17	0.008	202.69	0.010
	221	807.63	7835.20	0.010	202.69	0.010
	222	874.27	7732.59	0.012	205.42	0.010
	223	776.87	7491.86	0.010	205.42	0.010
	224	684.64	7317.04	0.008	205.42	0.010
	225	560.87	6297.36	0.007	102.03	0.010
	226	175.63	1842.81	0.009	102.03	0.010
	227	46.53	1329.09	0.001	197.85	0.010
T	228	38.25	793.00	0.002	197.85	0.010
	229	39.26	316.92	0.032	202.69	0.010

图 5.38 是以易于编辑的方式显示的线性预测编码(LPC)的帧(Frame)序列,(Dodge, 1985)。为了更清楚解释数值,加上了音素的数列。RMS2 的数列表示余冗的振幅,RMS1 则是原始信号振幅。ERR 则是两个振幅的比例值,数值较高时代表无发音信号。PITCH 则是音高,单位为 Hz,DUR 则是帧的时值,单位为秒。
Phoneme=音素 Frame=帧

线性预测编码合成(LPC Synthesis)

图 5.39 说明 LPC 的合成阶段。第一个参数是帧(frame)的时值,它决定给定的一组参数所要产生的输出取样点的数值。下一个参数决定此帧是声带音或非声带音。对于标准的声带音帧,合成器会使用音高参数来仿真人声的激发函数(the glottal wave)。此哼声(通常是限频的连续脉冲)用于元音与复元音(如"toy"中的"oy")。对于非声带音的帧,合成器使用噪音产生器来仿真声道中的扰动。

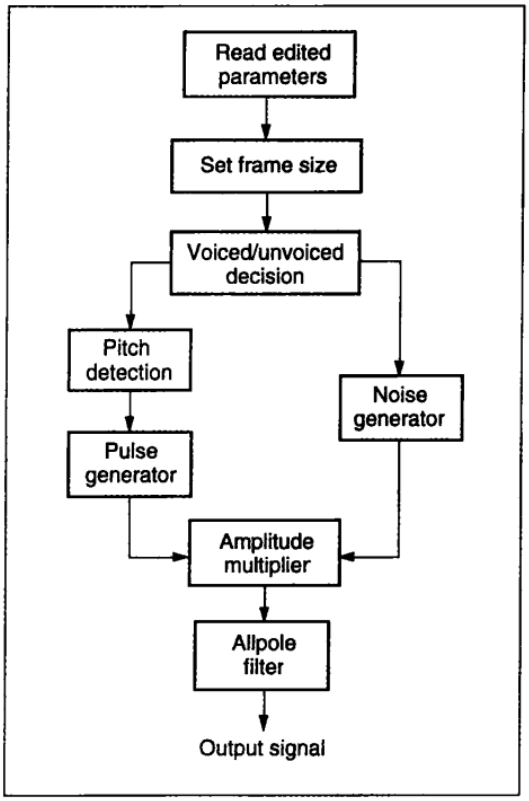


图 5.39　是线性预测编码(LPC)合成的略图
Read edited parameters=读取编辑过的参数　set frame size=设定帧的大小
Voiced/unvoiced decision=声带音/非声带音判定　Pitch detection=音高侦测
Pulse generator=脉冲发生器　Noise generator=噪音发生器　Amplitude multiplier=振幅乘法器
Allpole filter=全极点滤波器　Output signal=输出信号

经过振幅参数整形的适宜的发生器输出，用于全极点滤波器的输入。在模仿说话和唱歌中，全极点滤波器仿真语音及歌唱时声道的共振。在语音合成中最多使用 12 个极点的滤波器，而在音乐合成上可能需要高达 55 或更多个极点(Moorer 1979a)。

编辑线性预测编码的帧数据(Editing LPC Frame Data)

加入编辑及混音子系统后，线性预测编码(LPC)技术可以由纯语音应用调整为音乐上的应用。Dodge(1985)在一篇 LPC 作曲法的文章中，描述了一种编辑程序，所执行的 LPC 帧操作，如表 5.1 所示。在 LPC 帧上的一个主要操作方式是，将平板的说话声音转为歌唱。使用 LPC，可以延长字的时长，原本说话的音高曲线可以被流畅的旋律线替换。单字跟句子可以被任意重复，或者重新安排。句子还可以在不影响原始音高的前提下在时间上压缩。

表 5.1 LPC 帧的操作

帧时值的延伸或缩减
A 帧与 B 帧之间的时值的膨胀
在一组帧中改变特定参数值
在一组帧中的插入值(或比方创造出音高滑音)
将帧由 A 点移往 B 点
增强帧的强度
在一组帧间渐强
设定一个帧的音高
在每隔一个帧上做出颤音

作曲家 Charles Dodge 和 Paul Lansky 曾使用过 LPC 来做出所有这些效果,如在 Dodge 的作品《*Speech Songs*》(1975)以及 Lansky 的作品《*Six Fantasies on a Poem by Thomas Campion*》(1979),和作品《*Idle Chatter*》(1985, Wergo compact disc 2010-50)中。

标准 LPC 的音乐延伸应用(Musical Extensions of Standard LPC)

LPC 可以作为交叉合成(cross-synthesis)的实现方式(Mathews, Miller and David 1961; Petersen 1975;Moorer 1979a)。交叉合成在不同的使用系统上代表不同的意义(线性预测编码、卷积、相位声码器、小波等)。一般而言,它所指的技术是由分析两个信号开始,用其中的一个声音的某些特征来改变另一个声音的某些特征,通常是频谱上的变形。LPC 交叉合成利用一个声源的激励函数(音高及事件出现时间)来驱动另外一个声源的时变频谱包络。比方说,我们可以用一个复杂的波形,如管弦乐的声音,来代替创造发声语音的简单连续脉冲信号。得到的效果会像是"说话的管弦乐团"。图 5.40 本质上与图 5.33 的声码器相同,除了在声码器使用的简单激发函数被换成宽频的音乐声源(声源 B),且内部的分析/再合成方式是 LPC 法。

当希望的效果是使声源 B"说话",相对于窄频声源如小提琴独奏,使用宽频声源可提高语音信号的理解程度,如使用整个管弦乐团或是合唱团音源等。如果需要,激发函数也可以变白(whitened),以便将所有频谱成分提高到一致的水平(Moorer 1979)。

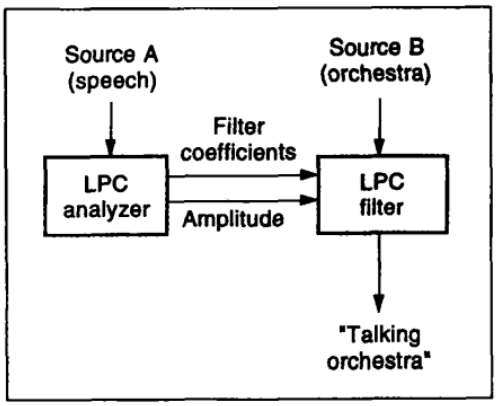


图 5.40　交叉合成推演频谱包络从一种声音转变成另一种声音。
Source A (speech)=声源 A (语音)　Source B (orchestra)=声源 B (乐队)　Filter coefficients=滤波器系数
LPC analyzer=LPC 分析器　LPC filter=LPC 滤波器　"Talking orchestra"="说话的乐队"

另一种 LPC 合成的延伸，是将单一乐器的滤波器响应向外推演，成为一整组的类似乐器。比方说，从小提琴的分析开始，我们可以复制中提琴、大提琴或者低音大提琴，成为完整编制的弦乐四重奏。(Lansky and Steiglitz 1981; Moorer 1981b, 1983a)。在理论上，这些滤波转换技术可以延伸，去模拟任何乐器的共振。在 Paul Lansky 的音乐里，此方式称为变形线性预测(warped linear prediction)，并用在合成弦乐器、萨克斯和口琴的电子版本(New Albion Records NA 030CD 1990)。

线性预测编码的评估(Assessment of LPC)

LPC 的语音合成结果容易辨识，以此方式模拟的传统乐器原始声音也很容易辨认。然而，LPC 所产生的语音与音乐的声音品质都不高。也就是说，仍可分辨出合成的复本与原始版本的差异。虽然这不能阻止该技术在音乐上使用，但在作曲应用上若能将音质提升将会更令人满意。穆尔(Moorer 1977, 1979a)曾实验使用高阶全极点滤波器以及更复杂的激励函数来提高 LPC 的音质。他的结论是："令人满意的音质还尚未出现。"他将之归咎于缺乏有效的工具来仿真激励函数。详见 Depalle(1991)对于不同的 LPC 频谱仿真的研究。

如果 LPC 模型的声音品质可以进一步提升，减法合成将比正弦波加法合成多一些优势。比方说，对于音高、频谱以及时域上的操纵，在减法合成中更为独立。在加法合成中，频谱通常与基频的音高有关。这表示当音高改变时，谐波的频率也会改变。另外，LPC 模型对于激励函数的频率不敏感，因此，它所产生的滤波器，不仅可以应用在基频上面的和谐频谱，而且也可以应用在基频上面的非和谐频谱。

双音素分析/再合成(Diphone Analysis/Resynthesis)

双音素分析/再合成的概念，是在数十年前于语音研究的背景中建立(Peterson and Barney 1952；Peterson，Wang，and Silvertsen 1958；Olive 1977；Schwartz et al. 1979)。其基本概念为，大多数语音声音是由过渡声音所分开的一连串稳定声音所组成。虽然此方法是为了创造易于分辨的语音信号，但是，其串联的一些点上仍有失真。双音素的概念第一次被检验是在减法分析/再合成技术的说明上。这也是为什么我们在本章引用这个技术概念，在这里，这个概念已经被延伸为另一种形式的再合成。

将此观念由语音信号泛化推展到音乐信号的范围，我们可以建立稳态及过渡声音的"字典"，来描述某些种类的声音，比方说，传统乐器声音。每个双音素可以解释为在特定强度上的一个音高。为了减轻双音素边界的失真问题，最近的研究工作集中在为每种乐器建立过渡规则的字典上，将相邻的双音素的串联平顺化(Rodet，Depalle，and Poirot 1988；见 Depalle 1991)。所以此研究也与在两个音之间建立仿真的过渡有关(Strawn 1985a，1987a)。但是它也提供了建立混合声音，从而将双音素与不同乐器联结起来的可能性。我们也可以做出合成的双音素。

为制作字典，每个声音都要被分析，此处我们假设分析方式是 LPC，以输入声音每秒 200 帧来运行。如果为了音乐效果将数据延展或压缩，可能会在迅速改变的信号如起音部分或两个声音间的过渡上造成不连续情况。因此，为了快速转换，双音素法以确保连续过渡的形式重塑这些分析数据，即使在受发音及句子变化的影响下也能确保这种连续过渡。比方说，延伸或压缩双音素的规则，会依照是来自哪个双音素，以及连接到哪个双音素的线索来改变。(Depalle 1991)。在每个双音素内部称为非内插区域(zone of noninterpolation)，该区域将保持完好，不会因转接而改变(图 5.41)。

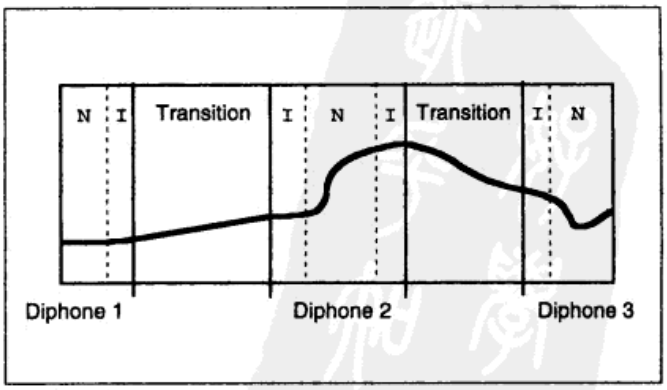


图 5.41 这是贯穿三个双音素的过渡。粗线标示声音在时间上延展的合成参数轨迹。此延展由每个双音素的内插区域(I)开始，延伸到过渡区域。非内插区域(N)不会被延伸，以保留双音素的中心部分。
Diphone=双音素 Transition=过渡

结论(Conclusion)

此章介绍了多重波表、波貌、粒式合成以及减法合成技术。多重波表合成是许多畅销合成器的核心技术。它以交错渐变及堆栈混合，让声音更丰富。如果需要的话，这些混合可以与合成频谱结合，以创造出奇特的、类似实际的声音。

波貌合成在计算上极有效率，且很有潜力，但是需要额外的音乐上的发展。比方说，它能够以音乐上有效的方式使用在取样的波表上吗?

粒式合成是基于短时值声音粒子的集合来产生声音的一组技巧。同步及准同步粒式合成产生带有音高的共振峰频谱，而异步则创造出更粒子化的声音效果，如云般在声音频谱上的变化。我们也可以将采样声源颗粒化，并创造出混合不同声源的颗粒，以做出特别丰富的音质。

减法合成是在一个历史悠久的学科基础上建立的。然而数字滤波设计仍属萌芽期间。因此，可以预期未来它将更为精练。计算能力的增加，将使实时滤波器组与减法分析/再合成的应用发展更为顺利。

部分原因是由于加法合成与减法分析/再合成过于复杂，故出现了种种有效的和专门的技术。这些技术除计算效率的长处外，也有其他的优点。通常需要音乐家给定较少的控制信息。它们在硬件上的实现较为廉价，并可制成小巧的可批量生产的集成电路。在许多情形下，由这些技术所产生的音乐化声音很难使用一般的分析/再合成方式来模拟和控制。因此说，这些技术存在的意义，是为现有的声音添加了一种调色板。第 6 章与第 7 章将探讨这些方法。

第6章 调制合成 (Modulation Synthesis)

在电子学与计算机音乐上，“调制”一词指信号（载波）的某些性质，依第二个信号（调制器）的某些性质而改变。在传统乐器与人声中，常见的震音（缓慢的振幅变化）与揉音（缓慢的频率变化）就是声音调制的例子。此处的载波是带有音高的声音，而调制器则是相对缓慢的函数（小于 20Hz）。在恰当的时刻下，以合适的速度使用震音与揉音，可让电子乐器及原声乐器更富表现力。

当调制的频率到达可听带宽内（约 20Hz 以上），会出现可听见的调制结果（modulation products）或边带（sidebands）。这些是在载波频谱上外加的频率成分（通常会在载波的两侧）。

就获得同等复杂程度的频谱来说，使用调制合成在参数数据、内存需求、计算时间上都要比加法合成或减法合成有效率得多。调制合成仅需数个振荡器（通常是 2—6 个），而加法或减法合成所需的计算消耗则多上数倍。根据所需的调制类型，调制技术只需要几个对应表、乘法器，以及加法器就可以实现。因为所需要的参数比加法或减法合成少，音乐家常认为调制技术更易于操控。

随着时间改变参数，使用调制技术可以轻易地做出时变频谱。经仔细调整后的调制将产生丰富的、极为接近自然乐器音响的动态声音。也可以将调制用在非模仿的方向上，以探索未分类的合成声音领域。

在此章对调制的讨论中，我们将使用很少量的数学公式，并穿插乐器线路图或排秩（patches）。这些图表以基本信号处理单元发生器（unit generators）的结构形式来说明合成乐器音色（见第 1 章对于单元发生器的介绍）。

调制信号的选择，可以简单到一个固定频率的纯正弦波，也可以复杂到含有所有频率的纯白噪音，详见第 8 章的噪音调制。

双极与单极信号（Bipolar and Unipolar Signals）

环形调制（ring modulation，RM）与振幅调制（amplitude modulation，AM）是两个关系很近的合成方法。为了要理解其间差异，必须先厘清它们所处理的两种信号形态：分别是双极（bipolar）与单极（unipolar）信号。通常音频信号都是双极信号，在时间轴上，波形在零点上下游移（图 6.1a）。相对的，单极信号则在整个系统的半边内振荡（图 6.1b）。可以把单极信号想成双极信号加上一个定值，此定值将所有取样点提高到零点以上。这个定值的另外一个名称是直流偏置（direct current offset，或 Dc offset）——也就是 0Hz 频率变化的信号（没有变化）。

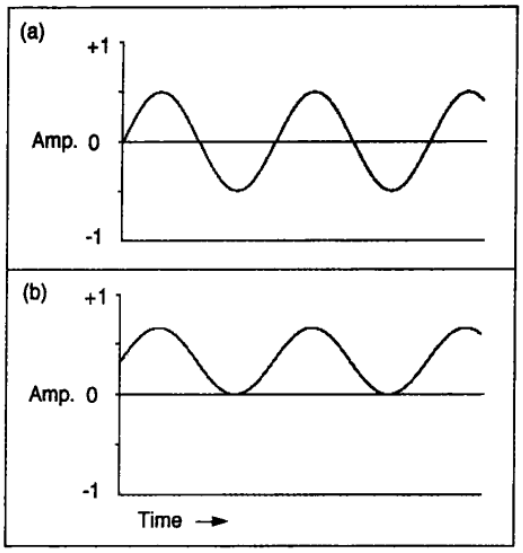


图 6.1　双极与单极正弦波。(a)双极正弦波在－1 和＋1 之间变化；(b)单极正弦波在 0 和＋1 之间变化。
Time=时间　Amp.=振幅

这种区分非常重要，因为 RM 与 AM 的基本差别，在于 RM 是两个双极信号的调制，而 AM 则用单极信号调制双极信号。下两节将进一步说明这两个方法。

环形调制(Ring Modulation)

我们的讨论从环形调制开始。理论上，环形调制是振幅调制的一种(Black 1953)。在数字系统中，环形调制就是单纯地将两个双极音频信号相乘。也就是说，载波信号 C 乘上调制器信号 M。C 与 M 的基本信号都来自两个已储存的波形，其中常用的是正弦波。环形调制信号 *RingMod* 的公式是直接相乘：

$$RingMod_t = C_t \times M_t$$

图 6.2 中绘出在 RM 乐器中，两个相类似的实现方式。在图 6.2a 中，假定载波振荡器通过振幅输入的数值乘上从波形表得到的数值。在图 6.2b 中，乘法的执行方式更为明显。两个情况中，调制器与载波都在－1 到＋1 之间变化，所以都是双极信号。

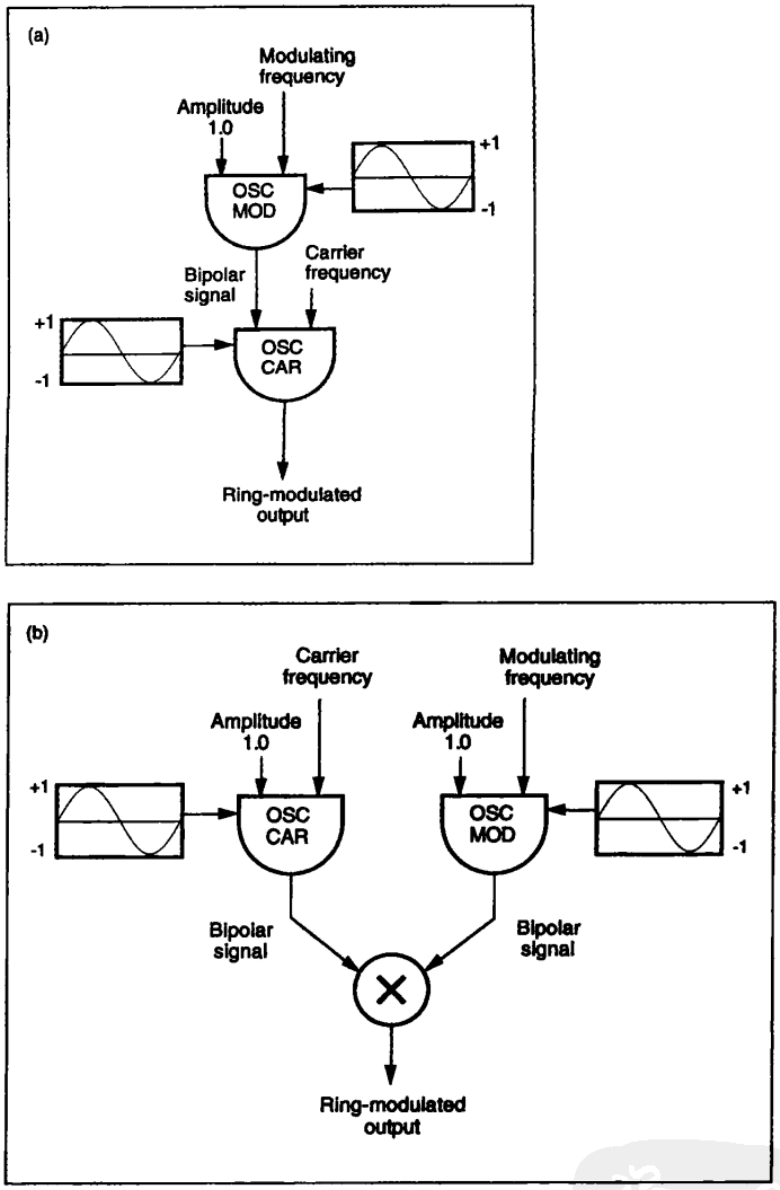


图 6.2 两个相关的环形调制或双极信号相乘的实现方式。每个振荡器的左边方块是它的波形。振荡器上方左边的输入是它的振幅，上方右边的输入则是频率。(a)环形调制 RM 是隐含在载波振荡器内的乘法中；(b)RM 是以载波及调制器的外部乘法表示。
Modulating frequency=调制频率 Amplitude=振幅 OSC MOD=调制振荡器
Bipolar signal=双极正弦波 Carrier frequency=载波频率 OSC CAR=载波振荡器
Ring-modulated output=环形调制过的输出

当调制器 M 的频率小于 20Hz 左右，环形调制的效果会使载波 C 的振幅依调制波 M 的频率而改变——也就是震音效果。但是当调制波 M 的频率到了可听范围内后，C 的音色将会改变。对于载波中的每个正弦波成分，调制器都

会在最后的频谱上使其形成一对边带(sideband)。若使用两个正弦波作为输入,RM 产生的频谱将有两个边带。这两个边带是 C 与 M 的频率差及频率和。奇特的是,原始的载波频率消失了。另外,若 C 与 M 之间的倍数为整数比,那么由 RM 产生的边带将是和谐的,否则将是非和谐的。

信号相乘得到的边带可由下面的三角函数公式得到:

$$\cos(C)\times\cos(M)=0.5\times[\cos(C-M)+\cos(C+M)]。$$

另外一种理解环形调制的方式,是将它想成一种卷积(convolution),如第 10 章所介绍。

此处举一个环形调制(RM)的例子,假定 C 是 1 000Hz 正弦波,而 M 是 400Hz 正弦波。如图 6.3 所示,其 RM 频谱将包含 1 400Hz(C 与 M 相加)与 600Hz(C 与 M 相减)的成分。

输出信号成分的相位,也将是两输入波形的相位之和及相位差。如果 C 与 M 是比正弦波更复杂的信号,或者它们的频率随着时间改变,那么得到的输出频谱将会有许多和及差的频率。频谱图将会出现许多条线,显示出非常复杂的频谱。

负频率(Negative Frequencies)

如图 6.3b 所示,当调制器频率比载波频率高时,将产生负频率,如当 $C=100$Hz 而 $M=400$Hz,所以 $C+M=500$,而 $C-M=-300$。在频谱图上,负频率可以用 x 轴上延伸向下的直线来表示。正负号的改变仅会改变信号的相位正负号。(当符号改变,波形将正负颠倒。)当将相同的频率值相加时,相位将会变得非常重要,因为反相(out-of-phase)成分将会抵消掉正相(in-phase)成分。

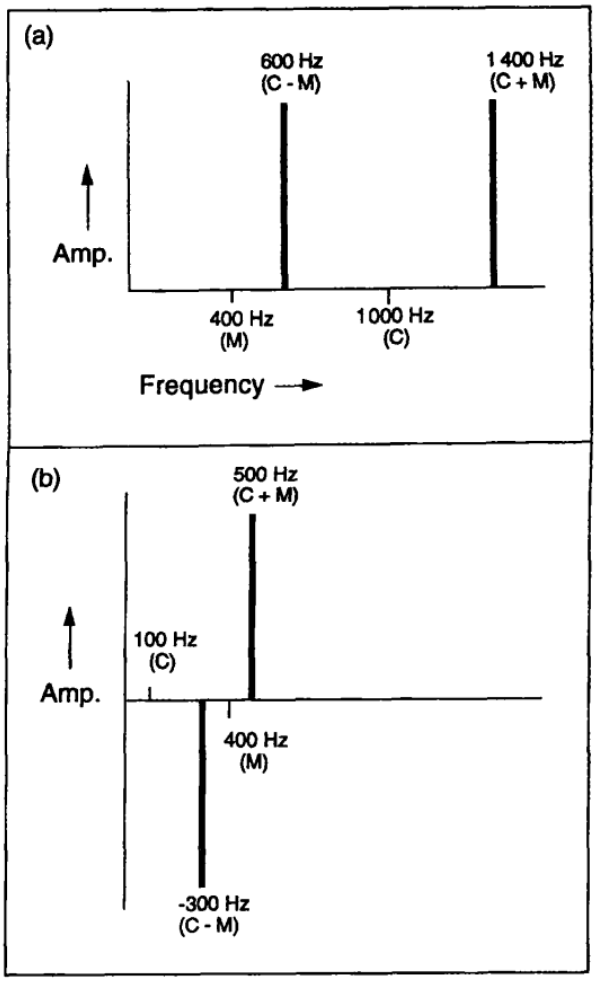


图 6.3 环形调制的频谱。(a)当载波为 1 000Hz,而调制器为 400Hz,频率和及差分别是 1 400Hz与 600Hz;(b)当载波为 100Hz,调制器为 400Hz 时,频率和及差分别是 500Hz 与 −300Hz。
Frequency=频率 Amp.=振幅

环形调制的应用(Applications of RM)

RM 在音乐上的典型用法,是用正弦波调制器来改变载波的取样信号(人声、钢琴等)。另外一个策略是用正弦波创造出纯合成声音,无论它是和谐的或非和谐的比例。这是作曲家 James Dashow 在作品 *Sequence Symbols* 所采取的方法(Dashow 1987)。

模拟环形调制与频率位移
(Analog Ring Modulation and Frequency Shifting)

数字环形调制以信号相乘来实现。一般来说，数字环形调制应该听起来都一样。相反地，许多模拟 RM 信号依其所使用的电路与组件而不同，会有不同的“特质”。这是因为模拟 RM 的实现方式，是在一个环形结构中用四个二极管(diodes)的电路来模拟环形调制中的乘法。这些电路会依二极管的形态(硅或锗)而产生额外的频率(Bode 1967，1984；Stockhausen 1968；Duesenberry 1990；Strange 1983；Wells 1981)。比方说，在以硅晶二极管为基础的模拟环形调制器中，电路在到达调制器某些短暂强度时，将会使载波产生截断失真(造成类似方波)。这将以如下形式，造成数个载波的奇数谐波上的和及差值：

$$C+M,\ C-M,\ 3C+M,\ 3C-M,\ 5C+M,\ 5C-M\cdots\cdots$$

图 6.4 比较了相乘 RM 与经过二极管截断的 RM 所输出的信号。模拟环形调制在 1950 年、1960 年和 1970 年间的电子音乐实验室中被大量使用。德国作曲家施托克豪森(Karlheinz Stockhausen)尤其钟爱环形调制，并使用在许多 1960 年间的作品上，包括 *Kontakte*、*Mikrophonie* Ⅰ *and* Ⅱ、*Telemusik*、*Hymnen*、*Prozession* 以及 *Kurzwellen*(Stockhausen 1968，1971b)。

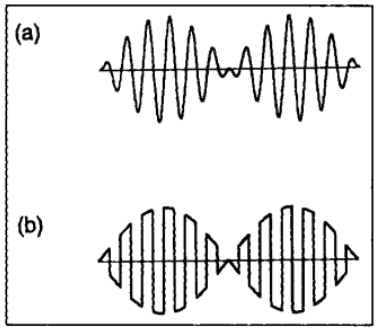


图 6.4　两种环形调制。(a)乘法环形调制；(b)二极管截波或断路器(chopper)RM。

环形调制的一位先驱，发明家 Harald Bode 还发展了环形调制的一种变化，称为频率位移(frequency shifting)(Bode 1967，1984；Bode and Moog 1972)。频率位移器(Klangumwandler)，将频率之和与频率之差分开，这种方法的另外一个名称是单边带调制(single-sideband modulation)(Oppenheim

and Willsky 1983)。

振幅调制(Amplitude Modulation)

振幅调制是最古老的调制技巧之一(Black 1953),并被广泛地应用在模拟电子音乐内。如 RM,一般载波的振幅会随着调制器波形的振幅而改变。这两种调制的不同是 AM 的调制器信号是单极性的(整个波形都在零点之上)。

也许,最常见到的次声(infra audio)AM 例子,是在正弦波上叠上包络。因为包络在 0 到 1 之间改变,所以是单极性的,如调制器般。而因为正弦波在 −1 到 +1 之间改变,所以是双极性的,如载波般。要将信号加上包络,即是将波形 C 与波形 M 相乘:

$$AmpMod_t = C_t \times M_t$$

$AmpMod_t$ 是振幅调制信号在时间 t 上的值。图 6.5 说明其结果。

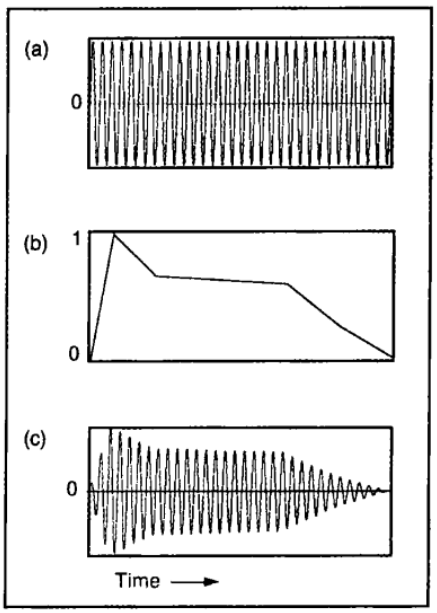


图 6.5 将信号加上包络,即是最简单的次声(infraaudio)AM。(a)中的正弦波信号与(b)中的包络信号相乘,得到经包络化的信号(c)。
Time=时间

如环形调制 RM 一样,振幅调制 AM 将在载波及调制器的每个正弦波成分上产生一对边带。边带与载波的分离,是通过与倒置的调制器周期相符的

距离完成的。RM 与 AM 在声音上的差别是，AM 频谱还包含载波频率（图 6.6）。两个边带的振幅会与调制量成比例增加，但是永远不会超过载波振幅的一半。

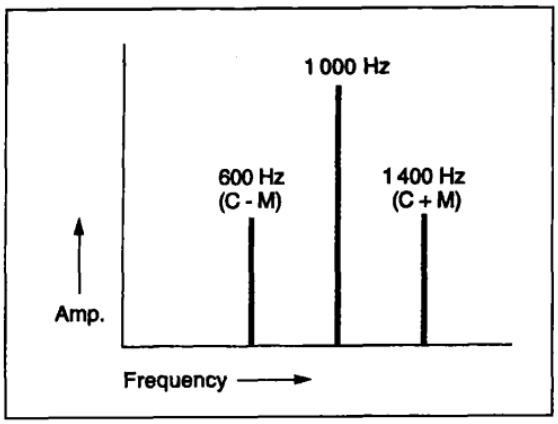


图 6.6　由 1kHz 正弦波与 400Hz 正弦波经 AM 产生的频谱。两个边带是载波频率的和及差值。每个边带的振幅是 index/2。
Frequency=频率　　Amp.=振幅

图 6.7 显示出在声频带中，两个正弦波信号调制完成的 AM 时域图。

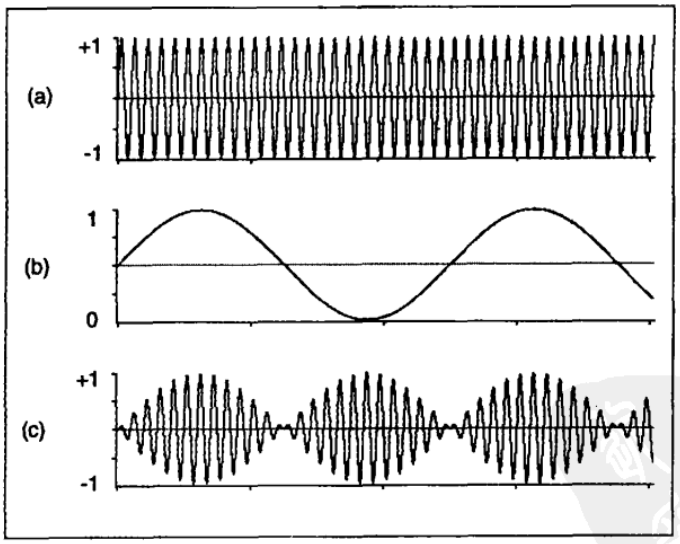


图 6.7　音频 AM 的时域图。(a)中的 1kHz 正弦波被(b)40Hz 正弦波调制，得到(c)的振幅调制结果。

振幅调制乐器(AM Instruments)

要实现典型的 AM 调制,我们将调制器的信号限制在单极性信号上——也就是在 0 到+1 之间。图 6.8a 显示简单的振幅调制乐器,调制器是单极性信号。

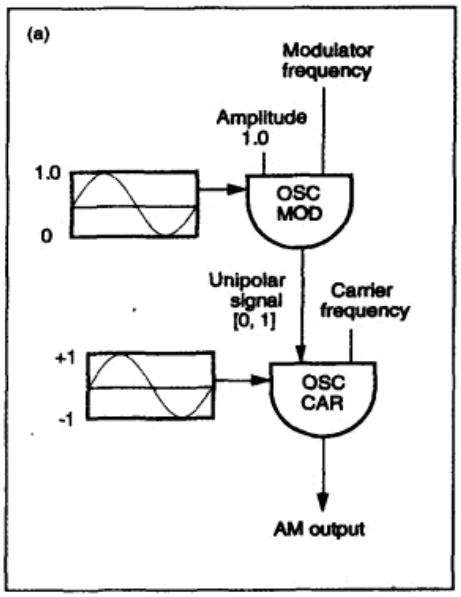


图 6.8 AM 的两种实现方式。(a)简单的振幅调制乐器,调制信号假设为单极性信号;(b)更复杂的振幅调制乐器,可控制整体的音符事件时程内的调制量和所有振幅。每个振荡器左边的方块是它的波形。在包络振荡器的例子中(以 ENV OSC 表示),频率周期是 1/note_duration。这代表仅会在一个音符事件的时程内读取波形表一次。正级计数器模块(scaler module)确保送入加法器的调制输入在 0 到 0.5 之间改变。

(a)

Modulating frequency=调制频率　Amplitude=振幅　OSC MOD=调制振荡器

Unipol ar signal=单极信号　Carrier frequency=载波频率　OSC CAR=载波振荡器

AM output=AM 输出

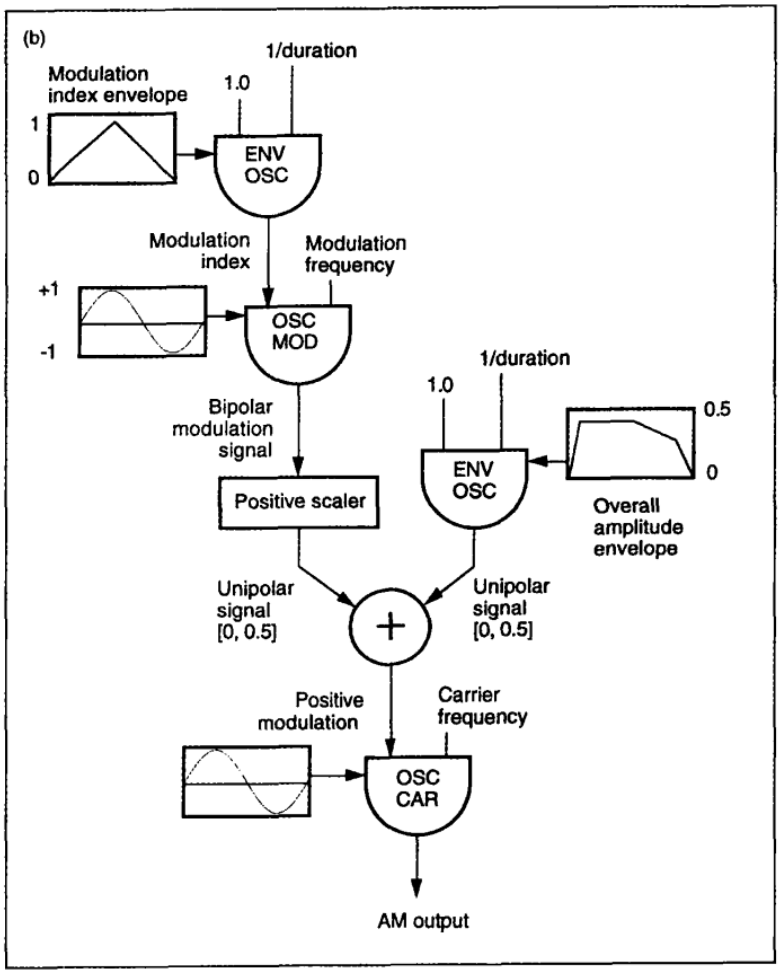


图 6.8 （续）

(b)

Modulation index envelope=调制指数包络　1/duration=1／时值　ENV OSC=包络振荡器
Modulation index=调制指数　Modulating frequency=调制频率　OSC MOD=调制振荡器
Bipolar modulation signal=双极调制信号　Positive scaler=正定标器
Overall amplitude envelope=总的振幅包络　Unipolar signal=单极信号　Positive modulation=正调制
Carrier frequency=载波频率　OSC CAR=载波振荡器　AM output=AM 输出

调制指数(Modulation Index)

要控制调制的量，以及控制整体的振幅包络，需要稍微复杂的乐器。图 6.8b说明用一个包络控制调制量的 AM 乐器(图的左上方)。在调制理论(之后将会讲解)中，包络的作用称为调制指数(modulation index)。此乐器将双极性调制信号调整为 0 到 1 之间的单极性信号，接着将其加在一个声音事件整个时

程的全部振幅包络上。下面的等式说明所得的 AM 波形：

$$AmpMod = A_c \times \cos(C) + (I \times A_c)/2 \times \cos(C+M) + (I \times A_c)/2 \times \cos(C-M)$$

$AmpMod$ 是经振幅调制了的信号，A_c 是载波振幅，I 是调制指数，C 是载波频率，M 则是调制器频率。

频率调制(Frequency Modulation)

频率调制(frequency modulation，FM)因 Yamaha 公司的采用，是个非常著名的数字合成法。然而，FM 并不单指一种技术，而是指基于非线性振荡函数的波形查表方法原理的一组技术。

背景：频率调制(Background: Frequency Modulation)

在通讯系统上，频率调制的应用可溯源至 19 世纪。射频频率(在 MHz 范围)的振幅调制理论在 20 世纪早期建立(Carson 1922; van der Pol 1930; Black 1953)。这些研究至今仍值得一读，尤其是 Black 书中为读者完整介绍波形调制之全貌。

斯坦福大学(Standford University)的约翰·乔宁(John Chowning)是系统性地探索数字 FM 调制合成音乐潜力的第一人(Chowning 1973)。在此之前，大多数数字声音都由固定波形、固定频谱技术产生。时变加法合成或减法合成极为罕见，计算起来也十分昂贵。由于大多数字合成工作都是在多人使用的大型计算机上完成，因此，有一种强大的吸引力，去开发重在时变频谱上的、更有效率的技术。乔宁这样解释这种动机：

“在自然声音中，频谱中的频率成分是动态的或称时变的。这种频率成分的能量常常以一种复杂的方式变化，这尤其体现在声音的起音和衰减阶段。”(Chowning 1973)

因此，乔宁开始探索一条能生成一种带有自然声音鲜活频谱特性的频率音响的途径。突破是在他从事极限振动技术(extreme vibrato techniques)实验时获得的。在这个实验中，振动是如此的快，以至变成了信号的音色效果。

“我发现只要用两个简单的正弦波，我就能创造出许多种不同的复杂声音，而这对其他的方法来说，需要许多更强大及更昂贵的工具。比方说，如果你想要有 50 个谐波的声音，你就得要有 50 个振荡器。而我只要用两个振荡器就能得到十分相近的声音。”(Chowning 1987)

在乔宁为探索此技术的潜力而经历了认真的实验后，他提出了 FM 合成实现的专利权。1975 年，日本公司 Nippon Gakki(Yamaha)得到在其产品上使用此专利的许可。经过数年的发展及基本技术的延伸(之后将介绍)之后，Yamaha 的昂贵 GS1 数字合成器在 1980 年间问世(16 000 美元以类似钢琴的木质盒装)。但它仅是 1983 年秋季推出的 DX7 合成器(2 000 美元)的序幕，DX7 的大幅成功，使得成千上万的音乐家将 FM 调制视为数字合成的同义词。

频率调制和相位调制(Frequency Modulation and Phase Modulation)

频率调制 FM 和技术上与其密切相关的相位调制(Phase Modulation，PM)反映了同类型的角调制(angle modulation)中两个实质上相同的例子(Black 1953，pp. 28-30)。由两种方式得到的泛音强度略有不同，但是在实际音乐应用中，尤其是在时变频谱上，PM 与 FM 并没有极大的分别。所以我们将不会在本书中深入探讨 PM。〔但本章将讨论另一种称为相位失真(phase distortion)的变化〕。要仔细厘清 PM 与 FM 之间的差异，详见 Bate(1990)、Holm(1992)与 Beauchamp(1992)。

简单 FM(Simple FM)

在基本频率调制技术中(称为简单 FM 或 Chowning FM)，载波振荡器的频率是由调制振荡器所控制(Chowning 1973，1975)。图 6.9 说明简单 FM 乐

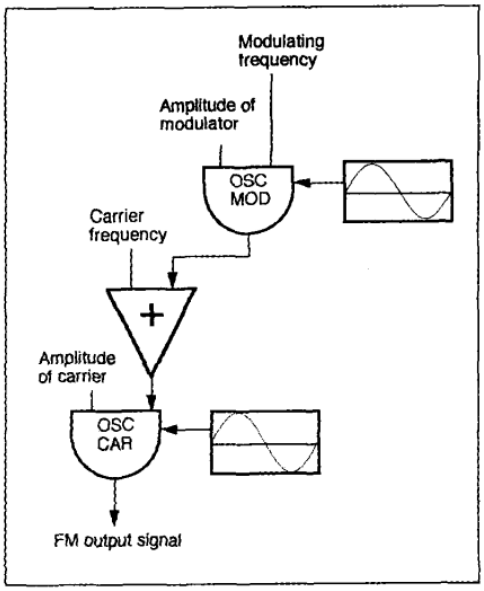


图 6.9　一个简单 FM 乐器，在基本的载波频率上加入了调制振荡器的双极输出，使得载波上下改变。调制器的振幅决定调制的量值，或说决定了基本载波频率的频率偏移。
Modulating frequency=调制频率
Amplitude of modulator=调制振幅
OSC MOD=调制振荡器
Carrier frequency=载波频率
Amplitude of carrier=载波振幅
OSC CAR=载波振荡器
FM output signal=FM 输出信号

器。(在图 6.9 中的乐器所描述的频谱成分的振幅与典型的 FM 公式之间偶尔有些许差异,但整体说来这些差异不大,有关摘要可参见 Holm 1992 与 Beauchamp 1992。)

看一看图 6.10 中的频谱,我们马上可以看到 FM 与前述的 RM 及 AM 之间的差异。两个正弦波的 FM 得到的并非只是一个边带的和与差,而是围绕载波 C 频率产生一连串的边带。每个边带以调制频率 M 的倍数的距离扩展。我们再查看一下边带的数目,其所产生的边带数目相当于用在载波上的调制量大小。

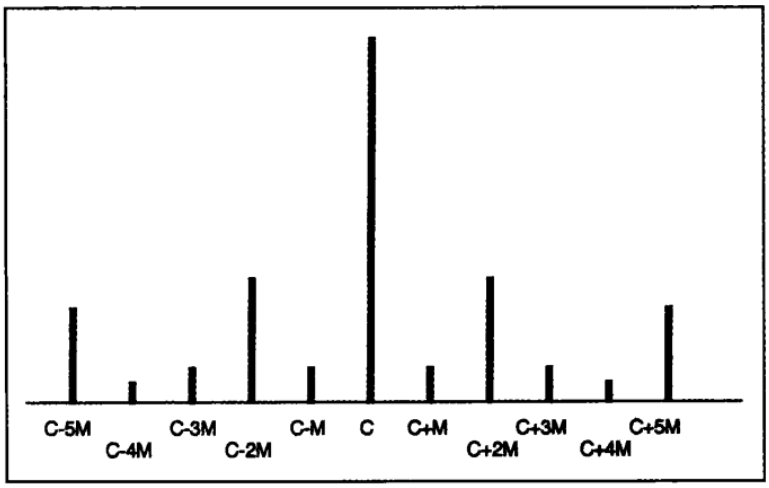


图 6.10 FM 频谱,边带围绕载波 C,以调制器 M 的倍数等距散布。

$C:M$ 比值($C:M$ Ratio)

由 FM 产生的频率成分位置取决于载波频率与调制频率间的比率。这称为 $C:M$ 比值($C:M$ ratio)。当 $C:M$ 是简单整数比时,如 4:1(例如在两个信号为 800Hz 与 200Hz 的情况当中),FM 将产生和谐频谱,也就是,边带是载波与调制频率的整倍数:

C=800Hz	(载波)
$C+M$=1 000Hz	(和)
$C+(2\times M)$=1 200Hz	(和)
$C+(3\times M)$=1 400Hz,等	(和)
$C-M$=600Hz	(差)
$C-(2\times M)$=400Hz	(差)
$C-(3\times M)$=200Hz,等	(差)

当 $C:M$ 并非简单整数比时，如 8：2.1（例如在两个信号为 800Hz 与 210Hz 的情况当中），FM 将产生非和谐频谱（载波与调制器的非整数乘积）：

C=800Hz	（载波）
$C+M$=1 010Hz	（和）
$C+(2\times M)$=1 120Hz	（和）
$C+(3\times M)$=1 230Hz，等	（和）
$C-M$=590Hz	（差）
$C-(2\times M)$=380Hz	（差）
$C-(3\times M)$=170Hz，等	（差）

调制指数与带宽（Modulation Index and Bandwidth）

FM 频谱的带宽（边带的个数）是由调制指数 I（modulation index 或 index of modulation）所控制，调制指数 I 在数学上的定义如下：

$$I=D/M$$

D 是载波频率的偏移量（单位为 Hz）。由于 D 表示调制的深度（depth），所以当 D 为 100Hz 且 M 为 100Hz 时，其调制指数为 1.0。

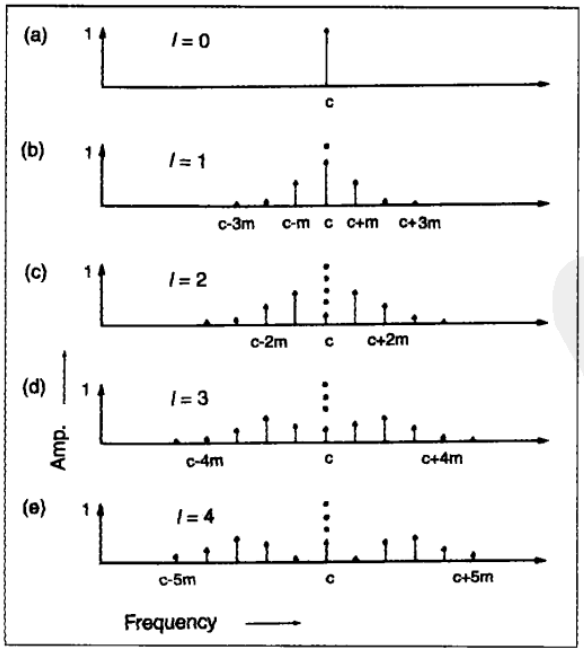


图 6.11　带有调制指数增加的 FM 频谱。(a)载波；(b)-(e)载波加上边带，由 $I=0$（见 a），到 $I=4$（见 e）。边带以调制频率 M 为间距，并与载波 C 形成对称。（出自 Chowning 1973。）
Frequency=频率　　Amp.=振幅

图 6.11 说明增加调制指数的效果。当 $I=0$ 时(图 6.11a),频率偏移为 0,所以无调制;当 I 大于 0 时,边带频率开始以调制器 M 为间距在载波 C 上下出现;当调制指数 I 增加时,边带的数目也跟着增加。注意,当调制指数 I 增加时,在载波上的能量被"偷走",分布在逐渐增加的边带上。

第一个原则是,成对的有效边带(大于载波强度 1/100 者)之数目,会近似于 $I+1$(De Poli 1983)。整体带宽接近于频率偏移 D 与调制频率 M 相加之和的两倍(Chowning 1973)。其公式的形式如下:

$$FM\ bandwidth \sim 2\times(D+M)$$

因为带宽会随调制指数增加而增加,所以 FM 可以模仿器乐音的一个重要性质,即当振幅增加时,带宽也会增加。这是许多乐器如弦乐器、圆号以及鼓等的典型状况。因此,带宽随调制指数增加而增加的现象可以在 FM 中,通过对载波振幅及调制指数使用类似的包络形状来实现。

反射边带(Reflected Sidebands)

对于某些载波与调制频率的值及调制指数(I)而言,最边缘的边带会在频谱的最上端与最下端反射回来,造成听觉上的副作用。超过奈奎斯特(Nyquist)频点(采样率的一半)之上的泛音,会"折回"(aliases),并反射到频谱的下端。(第 1 章有对折回的详细说明。)

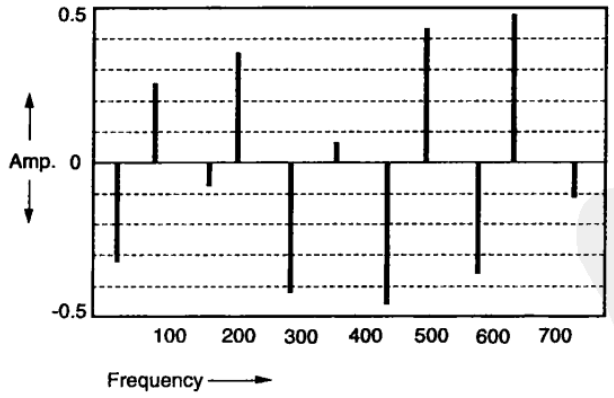


图 6.12 频谱的绘制显示了反射低频边带的效果。$C:M$ 的比率是 $1\sqrt{2}$,调制指数是 5,向下的线标示了相反转后的反射成分。(出自 Chowning 1973。)
Frequency=频率 Amp.=振幅

当下方边带低于 0Hz 时,它们将以 180 度的相反转(180-degree phase-in-

verted form)折回到频谱上。我们所说的“相反转”,指的是波形以 x 轴为准,上下颠倒,所以正弦波的正的部分成为负的,而负的部分成为正的。相反转的泛音可以绘成向下的线段,如图 6.12。一般来说,负频率成分会使得频谱的低频更丰富,但是如果负成分恰好与正成分相叠,将会彼此抵消掉。

FM 公式(The FM Formula)

当载波与调制器皆为正弦波时,在时间 t 上的频率调制信号 FM 可写为:

$$FM_t = A \times \sin\{C_t + [I \times \sin(M_t)]\}$$

这里,A 是载波的振幅峰值,$C_t = 2\pi \times C$,$M_t = 2\pi \times M$,而 I 为调制指数。如公式所述,单纯 FM 相当有效率,仅需要两个乘法器、一个加法器和两个波表查寻即可。波表查寻指的是存放在内存中的正弦波形。

贝塞尔函数(Bessel Functions)

单一边带成分的振幅,会依照一类称作贝塞尔函数〔Bessel functions of the first kind and the nth order $J_n(I)$〕的数学公式而改变,其公式的自变量是调制指数 I。上述的 FM 公式可用加入了贝塞尔函数的同等方式来重新表示(经 De Poli 1983 调整):

$$FM_t = \sum_{n=-\infty}^{\infty} J_n(I) \times \sin\{2\pi \times [f_c \pm (n \times f_m)]\}t$$

每个 n 都是一个独立泛音。所以,比方说要计算第三泛音的强度,我们在调制指数(I)点上乘上第三个贝塞尔函数,也就是 J_3(I),乘上载波频率两侧的两个正弦波。奇数低边带频率分量是反相的。

图 6.13 以三维方式说明贝塞尔函数,由 n=1—15,调制指数为 0—20。纵平面(起伏的平面)显示边带的振幅随着调制指数改变而改变。此图说明当边带的数目较少时(在整张图后端),振幅的变化将很剧烈。当边带数字增加(在整张图的前端),振幅的变化(波浪)较小。

从音乐的观点来看,重要的性质在于每个贝塞尔函数的起伏就像是一种带阻尼(译注:物体在运动过程中受各种阻力的影响,能量逐渐衰减而运动减弱的现象)的正弦波(damped sinusoid)——在调制指数 I 较小时变化很大,而调制指数 I 较大时变化很小。当我们改变调制指数,简单 FM 的这种起伏可以明显

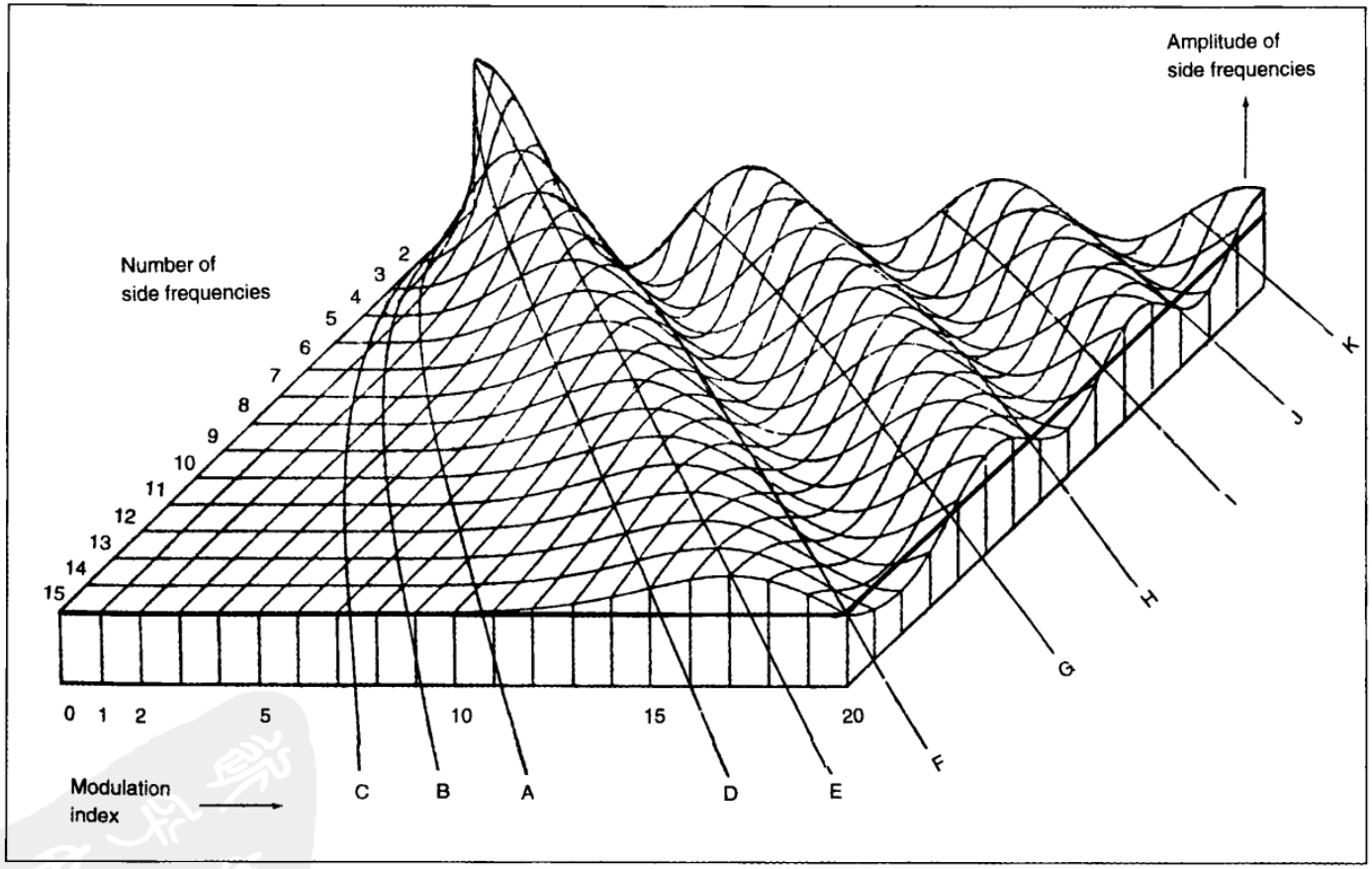


图 6.13 是 Bessel functions 1 15 的三维图(由后向前),作为调制指数 I 的函数(由左到右),显示所产生的边带数量(Chowning 1973)。A、B、C 线显示振幅分别衰减了−40dB、−60dB、−80dB。D 线显示"感知显著"的边带区分点。E 线则是每阶的最大振幅。F 线到 K 线则显示函数的过零点(zero crossing),也就是在不同的边带频率上,所产生的振幅为 0 或无效的调制指数值。

Number of side frequencies=边带频率数量　　Amplitude of side frequencies=边带频率振幅　　Modulation index=调制指数

地听得见。注意对于不同的 n 值，$J_n(I)$会在不同的调制指数 I 上到达 0 点。所以当调制指数 I 改变时，边带会以近似随机的方式而时隐时现。

FM 的一个方便的特性是，最高振幅以及信号能量并不是一定要依照调制指数 I 而改变。这表示当调制指数 I 增加或减少时，整体的声音强度并不会大幅改变。在音乐上，这表示我们可以独立地使用不同的包络，来操纵振幅与调制指数，而不需担心调制指数 I 的数值会不会影响整体强度。如本章后段所述，其他几种合成技术没有这种特性。这在波成形和离散求和公式上体现得更明显，因为调制会剧烈地改变输出振幅，所以这些技术需要振幅规格化(amplitude normalization)。

FM 的数字实现方式(Digital Implementations of FM)

图 6.9 说明一简单 FM 乐器，其调制深度可由常数频率偏移值来控制。但因为带宽直接相关于调制指数，而不直接相关于频率偏移，所以指定 FM 声音的较方便方法，是使用调制指数。在此例中，该乐器需经过改动，以执行下列公

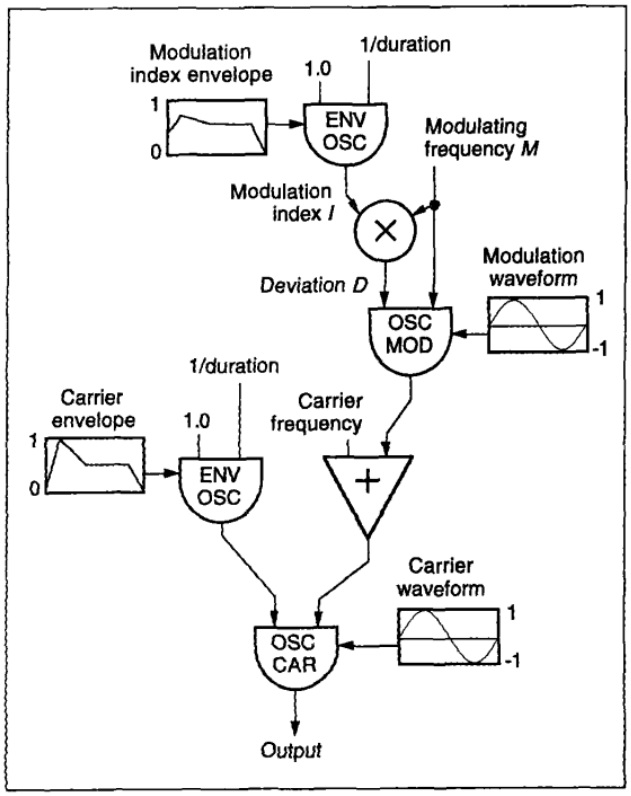


图 6.14　带有振幅和频率包络的一个简单 FM 乐器。这种乐器可以将使用者所规定的调制指数包络转换成频率偏差参数。
Modulation index envelope=调制指数包络
1/duration=1/时值
ENV OSC=包络振荡器
Modulating frequency *M*=调制频率 *M*
Modulation index=调制指数
Deviation *D*=偏差 *D*
OSC MOD=调制振荡器
Modulation waveform=调制波形
Carrier envelope=载波包络
Carrier frequency=载波频率
Carrier waveform=载波波形
OSC CAR=载波振荡器
Output=输出

式的额外计算：

$$D = I \times M$$

音乐家总希望不仅能动态地调制指数，也能动态地控制整体振幅。图6.14给定了这些包络，在 Chowning 的原始论文(1973)中，他描述了一个带有一个调制指数的乐器的变体，调制指数会在两个 *I1* 与 *I2* 值间，依照包络改变。(见 Maillard 1976，有另一种实现方式。)

简单 FM 的应用(Applications of Simple FM)

最直接的简单 FM 应用，是产生类似铜管乐器的声音。这类声音在振幅及调制指数包络上都有锐利的起音，并保持 $C:M$ 比值为 1。调制指数应在 0 到 7 之间改变。

当 $C:M$ 比值为 1∶2 时，将出现奇数谐波，可模仿真实的单簧管。像这样的无理数的 $C:M$ 比值：

$C:\ \sqrt{2C}$

会造成非和谐复杂波，可模仿打击乐器或类钟声的声音(Moorer 1977)。

除了模仿器乐声以外，另外一种通过 FM 作曲的方式，是利用它“非自然”的性质，以及其特殊的合成频谱。这是作曲家 James Dashow 及 Barry Truax 的方法。Dashow 使用 FM 来“和谐化”(以和谐“harmony”一字的延伸义)成对音高(pitch dyads)(Dashow 1980，1987；Roads 1985c)。Truax 系统性地以不同 $C:M$ 比值，制造出频谱的“家族”(Truax 1977)。比方说，某些 $C:M$ 比值会产生和谐频谱，而其他的会产生和谐与非和谐频谱的结合。每个 $C:M$ 值都可归类成产生类似频谱的一组比例“家族”的成员。此组频谱的差别只在于频谱能量集中处的载波位置。仔细地选定载波与调制频率后，作曲家可以通过同一组边带产生相关音色的级变。

用 FM 创作音乐的另一种方法是设定一个不变的载波 C 或调制波 M，然后用不同的 $C:M$ 比值来生成一些相关的音色。

指数 FM(Exponential FM)

在一般 FM 的数字实现上，边带会以相同的间距散在载波频率周围；我们称此为线性 FM(linear FM)。然而，某些模拟合成器的 FM，载波周边边带的间距是不均匀的，所以会造成不同形态的声音，我们称此为指数 FM(exponential

FM)。此节将解释这两种 FM 实现方式的差异。

多数模拟合成器，电压控制振荡器(voltage-controlled oscillator，VCO)的频率调制是由另一个振荡器实现的。然而，为了使平均律键盘(equal-tempered keyboard)可以控制 VCO，VCO 必须以频率相互关联的手法对给定的电压形成反应。特别的是，典型 VCO 会以每八度音一伏特的关系改变，与模拟合成器中的伏特/八度音的协议相符。比方说在这系统中，要得到音高 A880Hz，就输入比 A440Hz 的电压多 1 伏特即可。

在 FM 的情况中，一个调制信号由－1 到＋1 伏特之间的改变，会引起设在 A440Hz 的载波频率由 A220Hz 到 A880Hz 间的改变。这表示它向下调制了 220Hz，但向上调制了 440Hz——也就是非对称调制。载波平均中央频率的改变，通常表示感知的中央音高在音程上有很大偏移。此偏移是由调制指数造成的，这说明中央频率与带宽间是相互关联的。从音乐的观点来看，这种关联并不理想。我们希望增加调制指数，而不改变中央频率。详见 Hutchins(1975)对于指数 FM 的分析。

在数字调制中，边带以相同的间距散布在载波周围，所以称为线性 FM (linear FM)。当调制指数增加，中央频率仍保持不变。所有的数字 FM 都是线性的，且至少有一个制造商 Serge Modular 制作了线性 FM 模拟振荡模块。

分析与 FM(Analysis and FM)

由于 FM 技术可以创造许多不同类的频谱，它对于分析/再合成程序可能非常有用，一如在加法合成与减法合成加入分析程序。此程序可用一个已有的声音，将它转换为 FM 乐器的参数值。将这些数值送入乐器中，我们能听见 FM 合成出类似的声音。此种程序的一般名称叫参数预测(parameter estimation，见第 13 章)。已有许多使用 FM 来自动模拟稳态频谱的尝试(Justice 1979；Risberg 1982)。但预测 FM 参数中复杂且不断改变的声音的问题是很困难的(Kronland-Martinet and Grossmann 1991；Horner，Beauchamp，and Haken 1992)。

当数字硬件能力增加，一些预测 FM 参数的动机变得不重要了。FM 合成原先是为了节省运算，但现在许多强大的合成方式(如加法合成)已不再如此困难。调制合成方式只能将某类声音模仿得好。加法合成及物理模型(见第 7 章)可能是传统乐器更适当的模型。

多重载波 FM(Multiple-Carrier FM)

多重载波 FM(multiple-carrier frequency modulation，MC FM)一词指的

是FM乐器中以一个振荡器同时调制两个或更多的载波振荡器。载波的输出相加，得到调制频谱叠加后的合成波形。多重载波可以在频谱上创造出共振峰区域(formant regions，peaks)，如图6.15所示。共振峰区域是人声与大多数传统乐器的频谱特征。另一个使用分离的载波系统的理由，是为每个共振区域设定不同的衰减时间。这对于模仿铜管类乐器声音十分有用，因为高频分音衰减的速度远比低频分音快。

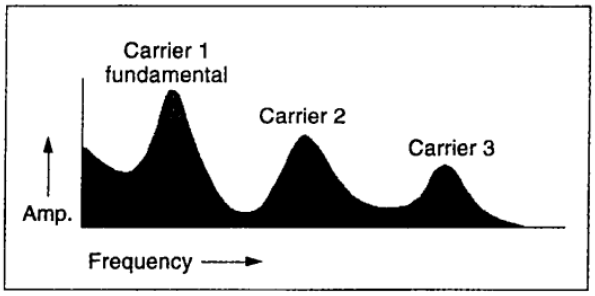


图6.15 由三个载波FM乐器所创造的三个共振峰区域的频谱。
fundamental=基频
Carrier=载波
Frequency=频率
Amp.=振幅

图6.16显示了一个三载波FM乐器。为了要指明多重载波的结构，此图省略了包络控制与波表。载波的振幅是独立的。当载波2与载波3振幅为载波1的分数时，此乐器会在第二与第三载波处产生共振峰区域。

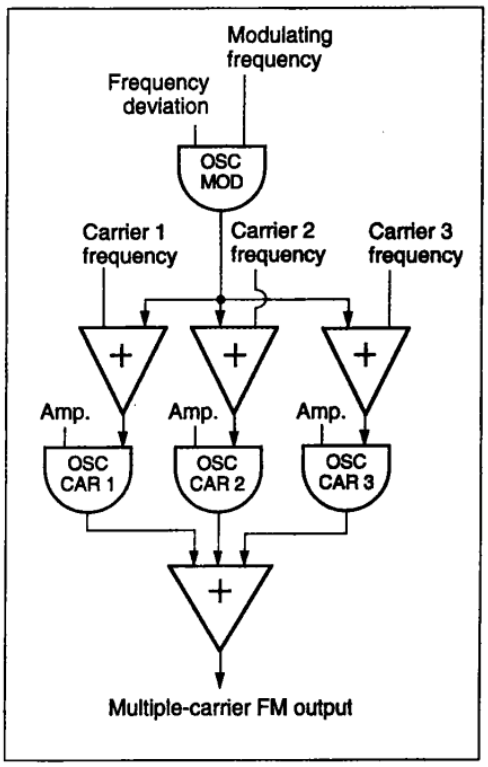


图6.16 由单一调制振荡器(OSC MOD)驱动的三个载波FM乐器。
Modulating frequency=调制频率
Frequency deviation=频率偏移
OSC MOD=调制振荡器
Carrier=载波
frequency=频率
Amp.=振幅
OSC CAR=载波振荡器
Multiple-carrier FM output=多重载波FM输出

时间 t 上的多重载波 FM 波形公式，是单纯地将 n 个简单的 FM 公式相加：

$$MCFM_t = A^{w1} \times \sin(C1_t + [I1 \times \sin(M)]) \cdots + A^{wn} \times \sin(Cn_t + [In \times \sin(M)])$$

A 为振幅常数，$0 < A \leqslant 1.0$，
$w1$ 为载波 1 的权重，
wn 为载波 n 的权重，
$C1$ 为基频 $= 2\pi \times$ 载波 1 的频率(单位为 Hz)，
Cn 为共振频点 $= 2\pi \times$ 载波 n 的频率(单位为 Hz)，而 Cn 为 $C1$ 的整数倍，
M 是调制频率，通常设为与 $C1$ 相同(Chowning 1989)，
$I1$ 是 $C1$ 的调制指数，
In 是 Cn 的调制指数，
指数 $w1$ 与 wn 决定载波如何随着整体强度 A 的变化而变化。

MC FM 的音乐应用(Musical Applications of MC FM)

文献中的 MC FM 应用方式致力于模仿传统乐器的声音。使用 MC FM——或是任何为此目的合成技术——要得到逼真模仿的诀窍，在于注意这声音的所有细节，包括振幅、频率、包络、揉音以及音乐的语境。

MC FM 的直接应用，是合成小号类的声音。里塞与马修斯(1969)对小号类的声音分析，说明了它具有近似于和谐的频谱，20—25 毫秒的振幅包络上升时间(其中高频泛音上升的速度较慢)，少量近随机性的频率波动，以及在 1 500Hz 区域的共振峰。Morrill(1977)以这些数据研发了单载波及双载波 FM 乐器，来模仿铜管乐器。双重载波乐器听起来较为真实，因为每个载波会产生频谱上的不同频段。特别是，$C1$ 产生基频以及前五到七个分音，而 $C2$ 设定为 1 500Hz，即小号的主要共振峰区域。每个载波有它自己的振幅包络，以调整组合频谱中两个载波系统间的平衡。比方说，在音量较大的小号声音上，高频泛音会较突出。

Chowning(1980，1989)将 MC FM 技术应用在对女高音和男低音歌唱的元音的声音合成上。他决定对所有频率参数都组合使用周期性与随机揉音信号，以得到人声的逼真模仿。“没有揉音，合成的声音会非常不自然”(Chowning 1989，p. 62)。类周期性的揉音会使频率“融合”成类似人声的声音。在 Chowning 的模仿中，揉音偏移率 V(vibrato percent deviation V)的定义如下：

$$V = 0.2 \times \log(\text{pitch})$$

所以当音高为 440Hz,V 相当于 1.2%,或是 5.3Hz 的深度。根据音高 F3 到 F6 之间的基频范围,揉音的频率分布在 5.0Hz 到 6.5Hz 之间。

多调制器 FM(Multiple-Modulator FM)

多调制器 FM(multiple-modulator frequency modulation, MM FM)一词,指的是使用超过一个振荡器来调制单一载波振荡器。有两种可能的基本配置:并联式或串联式(图 6.17)。要理解 MM FM 的最简单方式,是将调制器数目限制为两个,且其波形为正弦波。

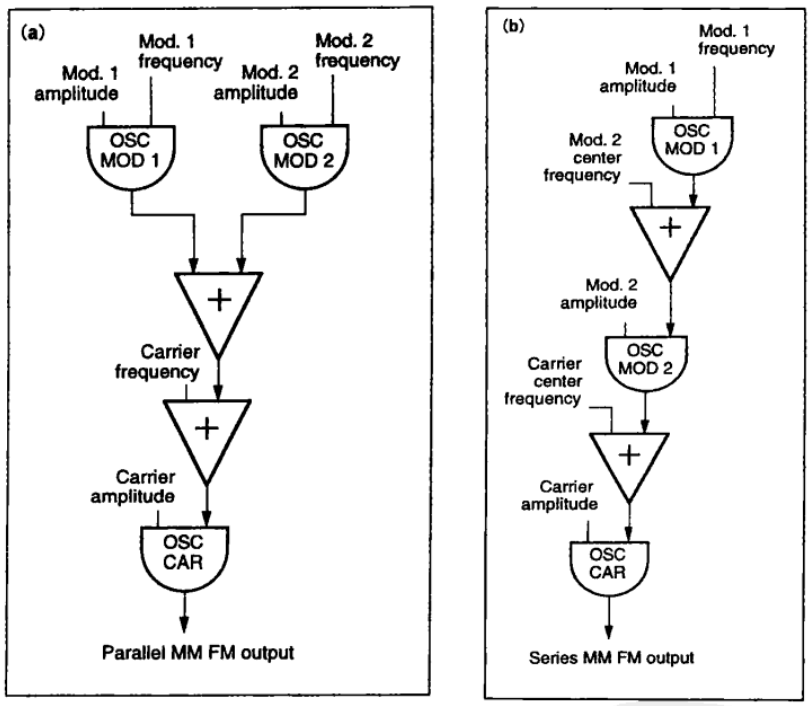


图 6.17 MM FM 乐器。
(a)并联式 MM FM;(b)串联式 MM FM。

(a)
Mod.=调制
amplitude=振幅
OSC MOD=调制振荡器
frequency=频率
Carrier frequency=载波频率
Carrier amplitude=载波振幅
OSC CAR=载波振荡器
Parallel MM FM output=并联多调制器 FM 输出

(b)
frequency=频率
Mod.=调制器
OSC MOD=调制振荡器
Mod.2 center frequency=调制器 2 中心频率
amplitude=振幅
Carrier center frequency=载波中心频率
Carrier amplitude=载波振幅
OSC CAR=载波振荡器
Series MM FM output =串联 MM FM 输出

并联式 MM FM(Parallel MM FM)

在并联式 MM FM 中，两个正弦波同时调制一个正弦载波。调制产生的边带频率会以此形式出现：

$$C \pm (i \times M1) \pm (k \times M2)$$

i 与 k 皆为整数，而 $M1$ 及 $M2$ 为调制频率。在并联式 MM FM 中，就好比每个调制器所产生的边带，会被另一个调制器当作载波来调制。在图 6.18 中列出了第一与第二调制结果，可以明显看到泛音数目的大幅增加。

并联双重调制器 FM 信号在时间 t 的波形公式为：

$$PMMFM_t = A \times \sin\{C_t + [I1 \times \sin(M1_t)] + [I2 \times \sin(M2_t)]\}$$

要深入了解利用数学来描述此技术所产生的频谱，可见 Schottstaedt (1977)与 LeBrun(1977)。

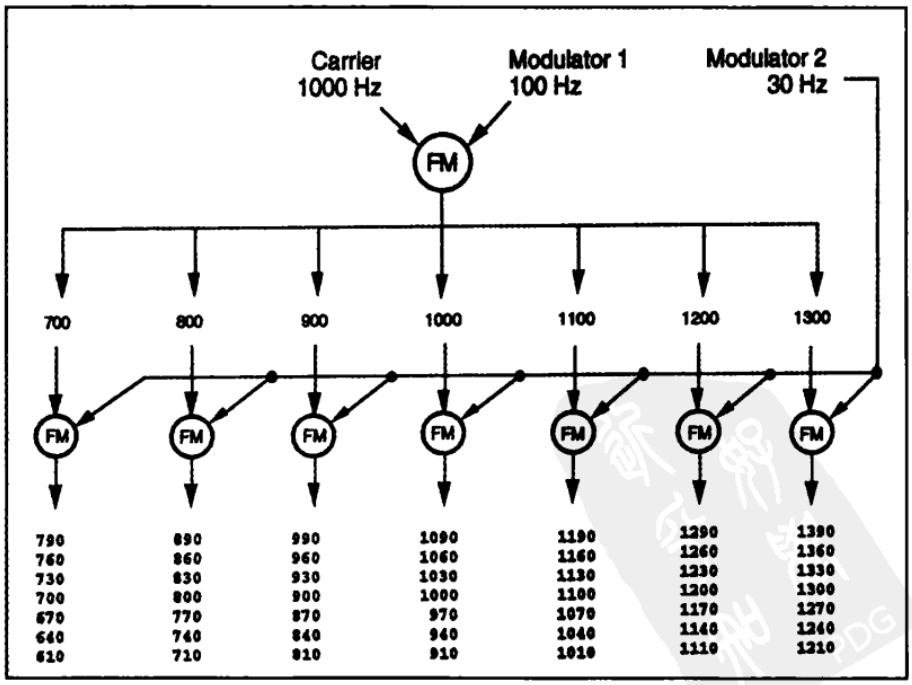


图 6.18　本图表描述由并行多调制器 FM 产生的分音的数字展开。由调制器 1 作用的载波调制所发出的每一个分量再经调制器 2 的信号调制，然后生成频谱分量，如图底部所列。
Carrier=载波　　Modulator=调制器

串联式 MM FM(Series MM FM)

在串联式 MM FM 中,调制正弦波 *M1* 本身被 *M2* 所调制。如此,根据调制指数的情况,会创造出带有大量正弦波边带成分的复杂调制波形。串联式双调制器 FM 的瞬时振幅可由从 Schottstaedt(1977)修改过来的下面的公式给定:

$$SMMFM_t = A \times \sin\{C_t + [I1 \times \sin(M1_t + [I2 \times \sin(M2_t)]\}$$

并联式与串联式的公式差别,反映在振荡器的配置不同。在实践中,*I2* 决定调制信号中显著边带的数目,而 *I1* 决定了输出信号的边带数目。即便 *I1* 与 *I2* 的数值很小,也会产生复杂的波形。*M1* ∶ *C* 的值决定载波的边带位置,每个边带位置都有诸多其间距由 *M2* ∶ *M1* 决定的自己的边带。所以每个边带既被调制,又都是调制器。

MM FM 的音乐应用(Musical Applications of MM FM)

Schottstaedt(1977)使用双调制器 FM,来模仿钢琴声音的某些特质。他将第一个调制器设定在接近于载波频率上,而第二个调制器接近于载波频率的四倍。根据 Schottstaedt 所说,如果载波及第一调制器完全相同,那么所产生的纯净和谐声音听起来很人工化,就像是电子(放大调音棒)钢琴。钢琴音中这种对不和谐的需要与声学家们的研究相一致(Blackham 1965; Backus 1977)。

Schottstaedt 将调制指数的振幅与频率相关,也就是说,当载波频率增加,调制指数将减少。得到的频谱将在低音上较丰富,而当音高升高,将逐渐变得单纯。由于钢琴音色的衰减长度会依音高而改变(低音的衰减较长),所以他使用了根据音高变化的衰减时间。

乔宁与 Schottstaedt 也曾利用 FM 三重调制器模仿弦乐类音色,其中 *C* ∶ *M1* ∶ *M2* 的比例是 1 ∶ 3 ∶ 4,而调制指数也是根据频率变化而变化的(Schottstaedt 1977)。乔宁同时也利用 MC FM 与 MM FM 两个乐器的结合,开发低沉的贝司音色。详见 Chowning(1980,1989)。

反馈 FM(Feedback FM)

由于 Yamaha 在数字合成器上的专利应用(Tomisawa 1981),反馈 FM 成

为广泛应用的技术。此节中我们描述三种反馈 FM：单振荡器反馈(one-oscillator feedback)、双振荡器反馈(two-oscillator feedback)以及三重振荡器间接反馈(three-oscillator indirect feedback)。

反馈 FM 解决了与简单(没有反馈的)FM 手法相关的某些问题。当简单 FM 中的调制参数增加时，其泛音的振幅将不会平衡地增加，而是依照贝塞尔函数上下移动(图 6.19)。这种泛音振幅上的波动会对简单 FM 频谱造成非自然的“电子声音”特质；这使得模仿传统乐器更为困难。

反馈 FM 使得频谱在变化时更趋于线性。一般来说，在反馈 FM 中，当调制指数增加时，泛音数以及其振幅皆相对线性地增加。

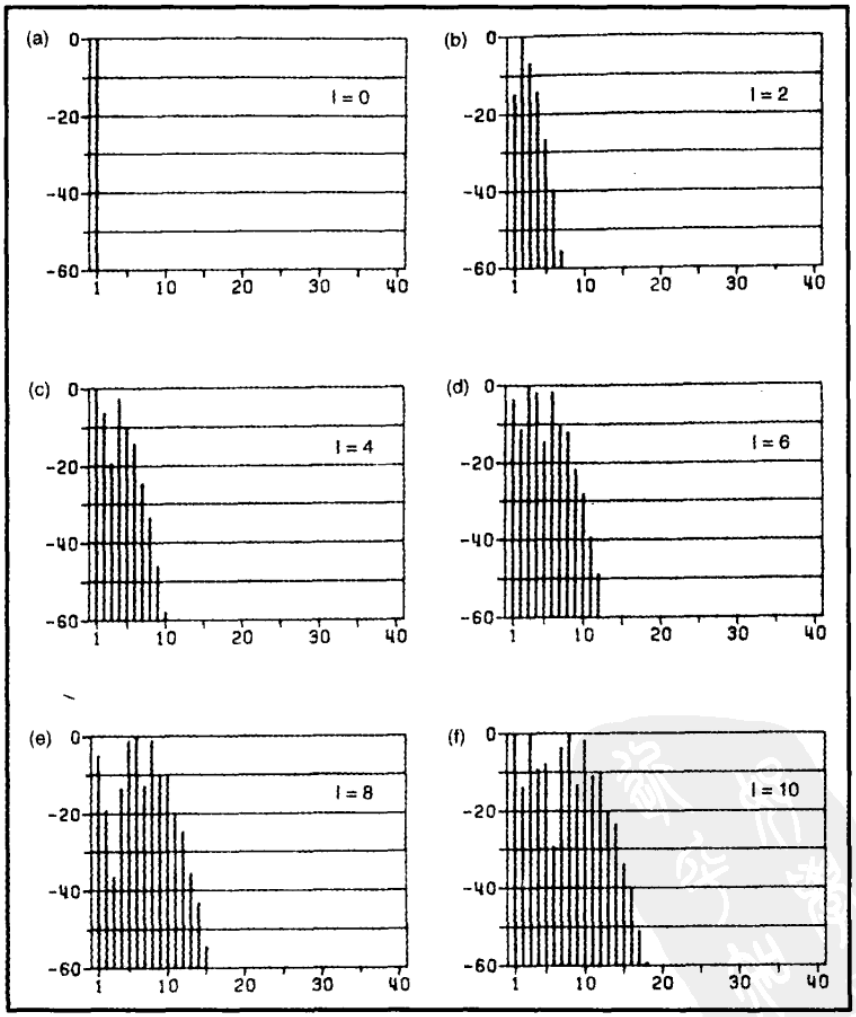


图 6.19　描述当载波 C 频率与 M 调制波频率相等，调制指数 I 由 0 增加到 22 时，FM 的谐波频谱(Mitsuhashi 1982b)。由左上角看此图，接着是右上，之后是第二列左图、右图，以此类推。注意频谱不平均的过程：当调制指数增加时，泛音先增加后减少。

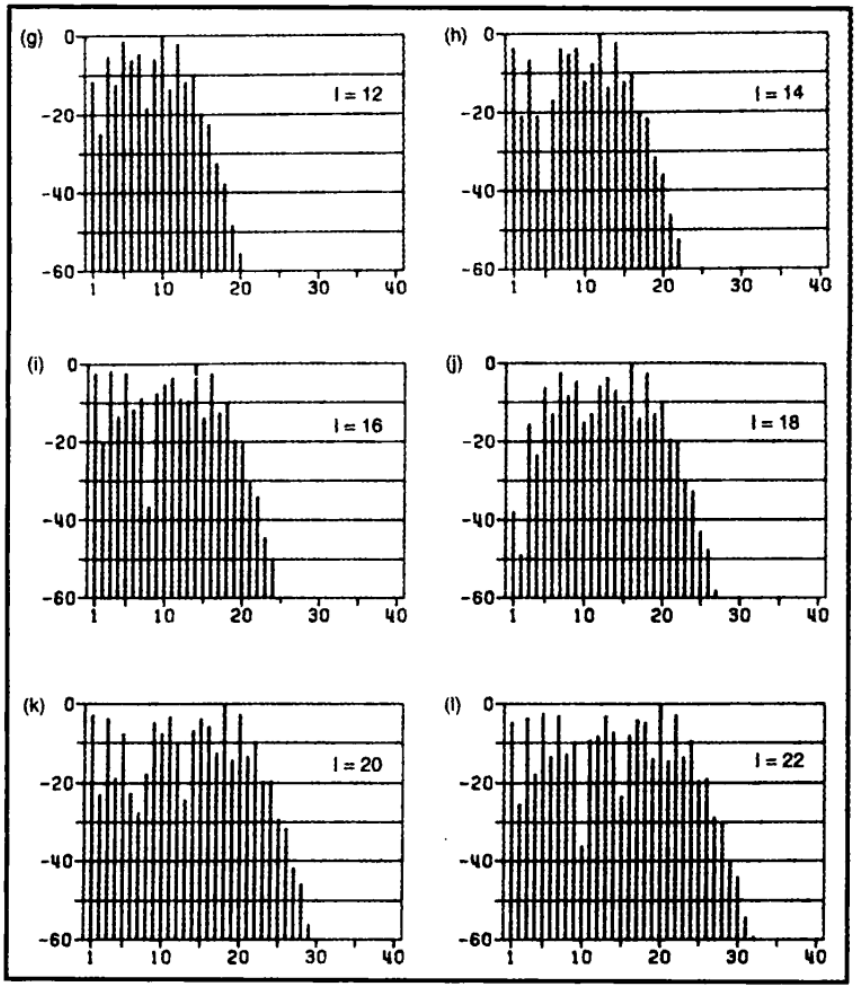


图 6.19 （续）

背景：反馈振荡器(Background: Feedback Oscillators)

反馈振荡器乐器最早在 1969 年，让-克劳德·里塞(Jean-Claude Risset)的 *Introductory Catalog of Computer Generated Sounds* 一文中出现。由于此目录并非公开发行，故此技术第一次的公开出现，是在一篇模糊的论文中，以一个隐密标题《计算机合成化声音的某些特异面貌》(*Some idiosyncratic aspects of computer synthesized sound*)发表(Layzer 1971)。在论文中，Arthur Layzer 描述了在贝尔电话实验室，开发一种其输出会反馈作为输入的自我调制振荡器的工作。此工作是与里塞、马克斯·马修斯(Max Mathews)和穆尔(F. R. Moore)共同合作实施的。穆尔(F. R. Moore)在 Music V 语言上以单元发生器(unit gen-

erator)的方式,实现了反馈振荡器。(Music V 在 Mathews et al. 1969 中有描述。)

贝尔实验室的反馈振荡器,与 Yamaha 的反馈 FM 技术的最大不同在于前者将信号反馈到振幅输入,而后者将信号反馈到频率或相位增量输入。所以早期的反馈振荡器是以"反馈 AM"形式,而不是反馈 FM 形式实现。

单一振荡器反馈(One-oscillator Feedback)

单一振荡器反馈 FM 的基本概念很容易描述。图 6.20 说明振荡器将输出经过乘法器与加法器送回到频率输入端。加法器为这个振荡器内的正弦检表操作计算相位索引值。在每个取样周期,一个数值 x(频率增量)会与现有的相位相加。正弦波波表上以新相位所得到数值即是输出信号 sin(y)。在合成器中,x 通常由按下的琴键所给定。若琴键音高较高,将转换为一个大的相位增量值;若音高较低,则得到一个较小的相位增量值。

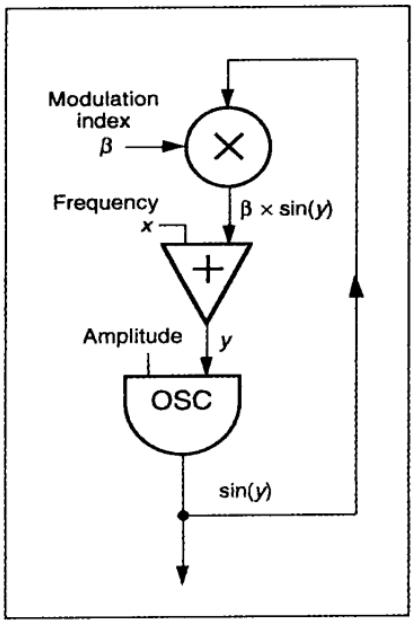


图 6.20　反馈 FM 乐器。x 为正弦波检表的相位增量,x 乘上一个反馈因数(feedback factor) β,加上了从输出反馈的信号。

在反馈 FM 中,输出信号 sin(y)会先经过乘法器乘上反馈因数(feedback factor β)后送到加法器。因数 β 的作用相当于缩放函数或反馈的"调制指数"。在反馈循环中,下个取样点的地址是 $x+[\beta\times\sin(y)]$。

图 6.21 绘出当 β 渐增时,一个单振荡器反馈 FM 乐器的频谱。注意泛音数目的增加,以及有规律的、在泛音之间的振幅上增量的不同,所有的性质都造成一种类似线性频谱的构成。当调制渐增,信号会以持续的方式由正弦波变化为锯齿波。

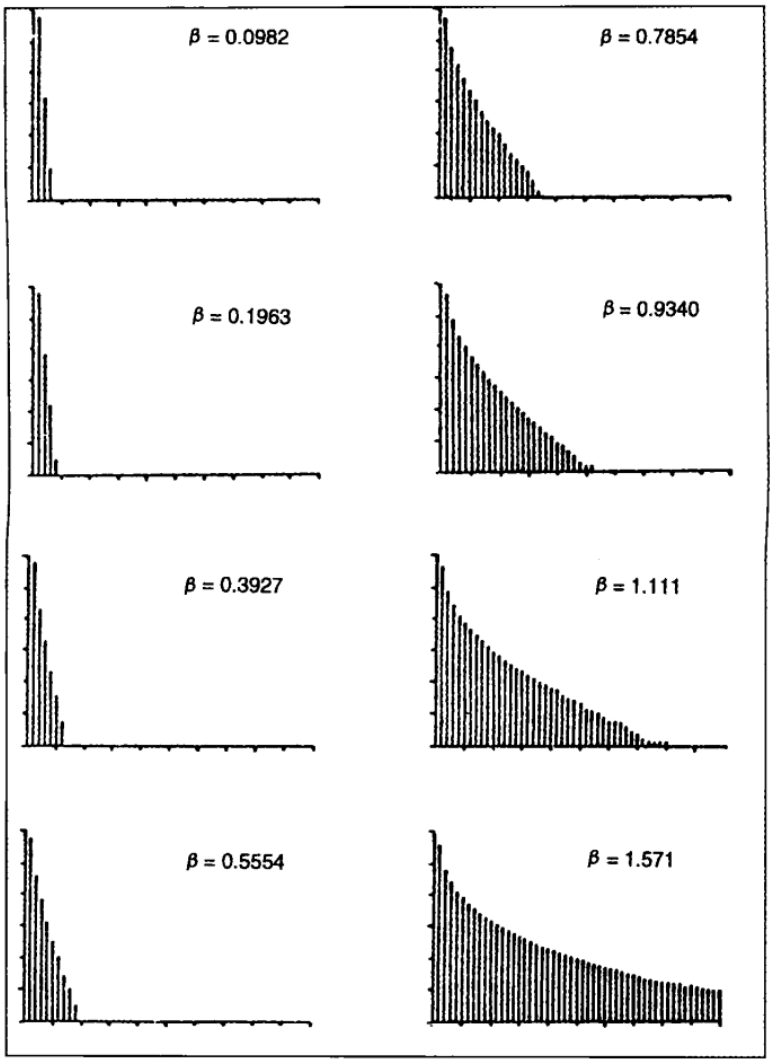


图 6.21 当反馈因数 β 增加时，单一振荡器反馈 FM 乐器的频谱，其相位增量 x 设为 200Hz。水平轴显示 0 到 10kHz 的频率，垂直轴显示 0 到 60dB 的振幅。

单振荡器反馈 FM 的公式可以由贝塞尔函数给定(Tomisawa 1981)：

$$FFM_t = \sum_{n=1}^{\infty} \frac{2}{n \times \beta} \times J_n(n \times \beta) \times \sin(n \times x)t$$

这里，$J_n(n)$为位数 n 中的贝塞尔函数，$n \times \beta$ 则是调制指数。在反馈 FM 中，贝塞尔函数会以不同于简单 FM 的方式作用。在简单 FM 中，调制指数 I 对于每个贝塞尔成分 $J_n(I)$都是一样的。这表示每个贝塞尔函数的 $J_n(n)$值是通过一个由同样调制指数完成的位置高度表示的。于是，当 FM 的调制指数有

规律地增加时，频谱包络会呈波动特征。而在反馈 FM 中，贝塞尔函数 $J_n(n\times\beta)$ 的位数 n 是包含在调制指数内，且因数 $2/(n\times\beta)$ 会当作一个系数，乘在贝塞尔函数上(Mitsuhashi 1982a)。

在反馈 FM 中，调制指数 $n\times\beta$ 会对每个位数 n 而有所不同，且以接近于单调函数(也就是以常数增加)的形态增加。缩放系数 $2/(n\times\beta)$ 确保当泛音的位数 n 增加时，其振幅减少。

双振荡器反馈(Two-oscillator Feedback)

另一个反馈 FM 的排秩(patch)，是使用反馈振荡器的输出来调制另外一个振荡器(图 6.22)。该图中的乘法器 M 的作用相当于两个振荡器间的调制控制指数。

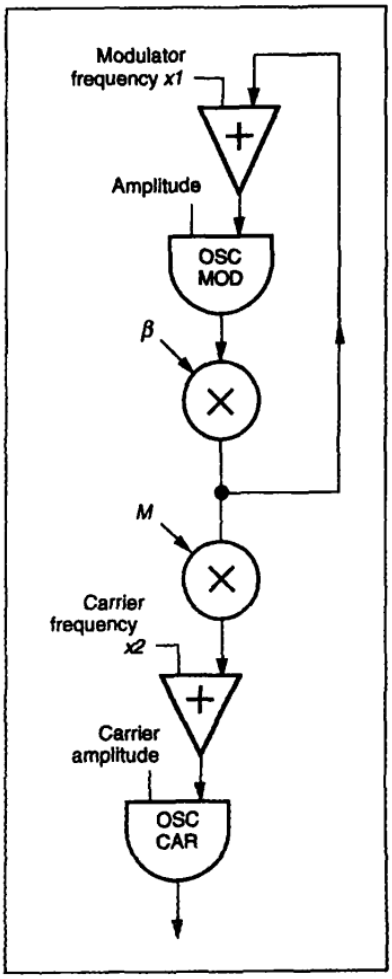


图 6.22　双振荡器反馈 FM 乐器。反馈 FM 振荡器输出信号调制第二个非反馈振荡器。
Modulator frequency=调制频率
Amplitude=振幅
OSC MOD=调制振荡器
Carrier frequency=载波频率
Carrier amplitude=载波振幅
OSC CAR=载波振荡器

当 M 的范围在 0.5 到 2 时，频谱将有单调递减的趋势，在这种情况下，随着泛音的数目增加，其振幅会递减(图 6.23)。当反馈参数 β 大于 1 时，高阶泛音的整体振幅会增加。这将创造出可变滤波器的效果，所以它将有更刺耳、尖锐的声音。然而，当 M 设为 1，且 $x1$ 与 $x2$ 相等时，此乐器将创造出与单一振荡器反馈 FM 乐器相同的频谱，如图 6.20 所示。

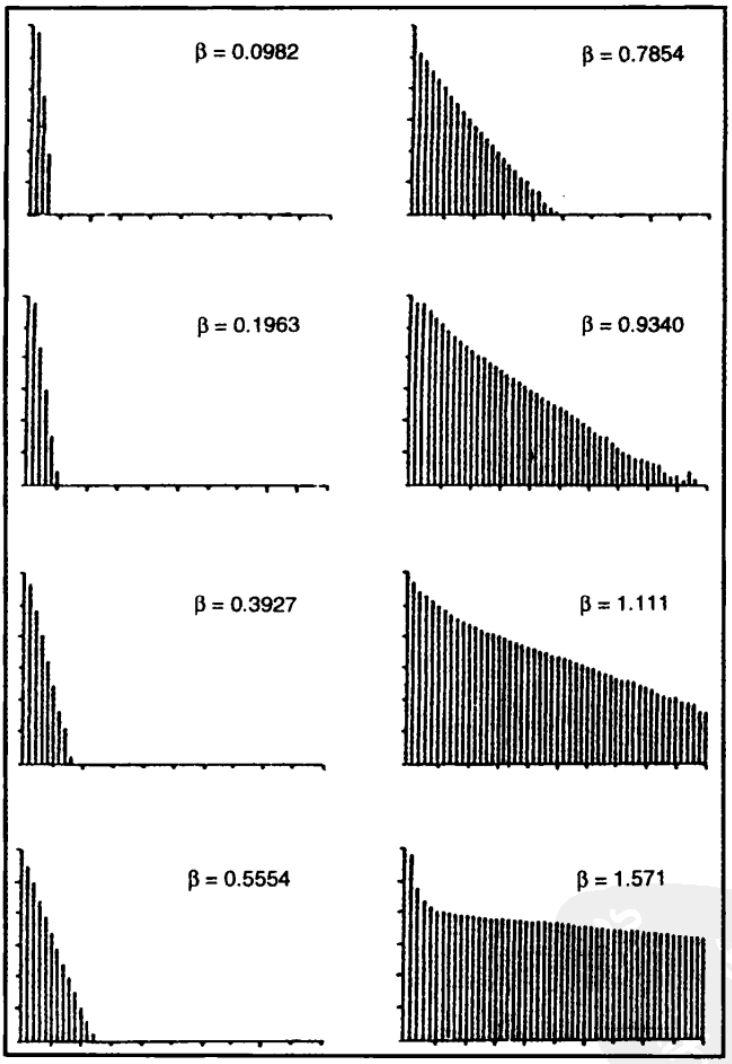


图 6.23 当反馈因数 β 由 0.0982 增加到 1.571 时，双重振荡器反馈 FM 乐器 t 的频谱。其频率 $x1$ 与 $x2$ 均设为 200Hz，调制指数 M 设为常数 2。水平轴显示 0 到 10kHz 的频率，垂直轴显示 0 到 60dB 的振幅。

当 $x2$(载波)与 $x1$(调制器)的比例为 2∶1，调制指数 M 为 1，反馈指数 β 在 0.09 与 1.571 之间变化时，得到结果是由类正弦波到类方波间持续的变化。

三个振荡器的间接反馈(Three-oscillator Indirect Feedback)

另外一个反馈 FM 的变化，是带有间接反馈的三个振荡器技术，如图 6.24 所示。此反馈参数为 $\beta 1$。间接反馈将造成复杂的调制形式。当频率 *x1*、*x2* 与 *x3* 以非整数相乘时，将造成没有确切音高(nonpitched)的声音。当这些频率是十分接近的整数关系时，会造成拍音合唱效果(beating chorus effect)。根据声音设计师大卫·布里斯托(David Bristow)(私人书信 1986)所述，此乐器将产生丰富的频谱，当反馈增加，能量倾向集中在频谱的高频端。

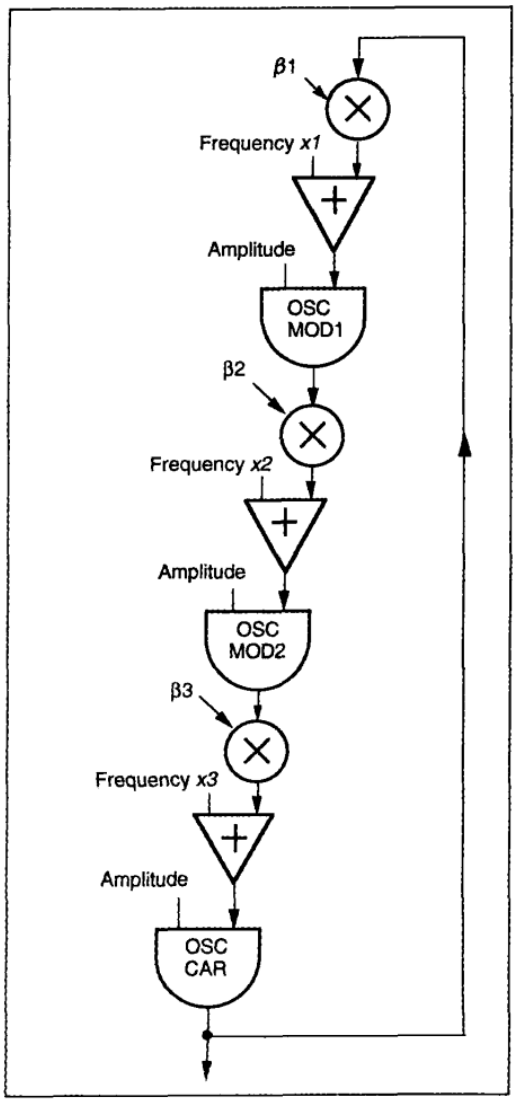


图 6.24　三个振荡器的间接反馈调频音础器。串联的三个振荡器彼此调制，三个调制指数要素 *β1*、*β2* 及 *β3* 决定调制的量。总体输出再反馈到第一个调制振荡器。
Frequency=频率　Amplitude=振幅
OSC MOD=调制振荡器
OSC CAR=载波振荡器

相位失真合成(Phase Distortion)

相位失真合成(Phase distortion, PD)一词由卡西欧公司发明,以描述一个为该公司的几种数字合成器开发的简单调制技术。PD 合成使用正弦波波表振荡器,而其扫描率在整个波表周期中变化,扫描的时间间隔在 0 到 π 间加速,随后在 π 到 2π 间减速。整体的频率依照音符的音高维持一定,但输出波形不再是正弦波。图 6.25 绘出扭曲(加速后减速)扫描函数所得到的输出波形的效果。

当加速与减速的量增加(进一步扭曲扫描函数),原本的正弦波波形变成类似三角波,最后成为带有丰富谐波的、类似锯齿波的波形。

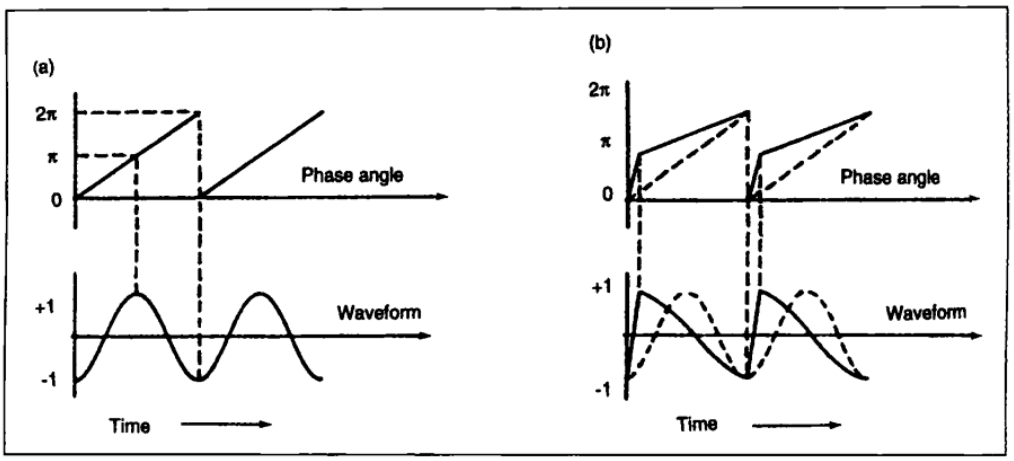


图 6.25 卡西欧的相位失真合成波形。声波会因读取正弦波波表的速度改变而改变。(a)以恒量的读取率生成的一个正弦波;(b)读取速度每周期改变两次,将正弦波失真为类似锯齿波波形。

Phase angle=相位角度 Waveform=波形 Time=时间

波成形合成(Waveshaping Synthesis)

里塞在纽泽西的贝尔电话实验室工作时,执行了第一个当时还不为人知的波成形合成(waveshaping synthesis)实验(Risset 1969)。Daniel Arfib(1979)与 Marc LeBrun(1979)独立发展了此基本方法的理论与经验的细节。波成形合成在音乐上很有趣,因为,如 FM 合成一样,它可以让我们简单地以相当节省的计算方式把握一个音的时变带宽与频谱。

波成形(也被称为非线性失真 nonlinear distortion)基本观念是将声音信号

x 通过“失真方块”(distortion box)。在数字形式中,失真方块是储存在计算机内存内的表(或数组)中的一组函数。该函数 w 将[−1,+1]范围内的任何输入值 x,对应到相同范围内的输出 $w(x)$。

最简单的情况,x 是由一个振荡器所产生的正弦波。然而,x 可以是任意一种信号,并不只是正弦波。对每一个要计算的输出取样,我们都将数值 x 用在指数表 w 上。指数表 w 含有成形函数(shaping function,亦称转换函数 transfer function)。这样,我们就能简单地把由 x 指数化的 w 中的数值变成输出值 $w(x)$。

简单波成形乐器(Simple Waveshaping Instrument)

图 6.26 所显示的是一个简单波成形合成乐器。此处包络振荡器控制送入成形函数表中的正弦波振荡器的振幅。振幅包络 x 很重要,因为它有缩放输入信号的效果,使其对应到成形函数 w 的不同区域。我们下节介绍这些关系。

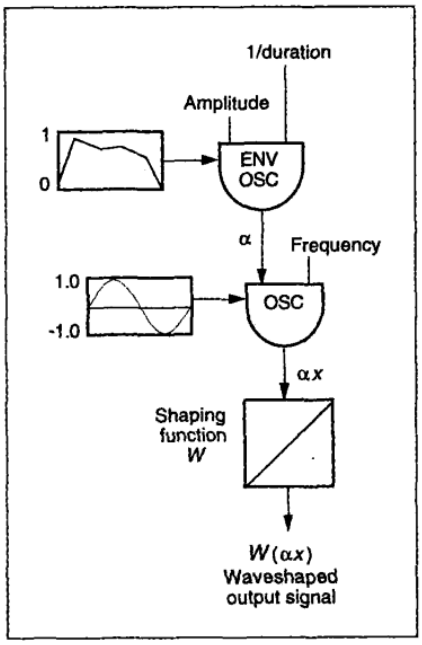


图 6.26　简单的波成形乐器。一个振幅被另一个振幅包络信号 x 控制了的正弦波振荡器,被当作成形函数表 w 的一个索引值。如同其他的范例乐器一样,输送到包络振荡器频率输入端中的输入 1/时值,表示其走过了一个音时值的整个一个周期。
1/duration=1/时值　Amplitude=振幅　ENV OSC=包络振荡器　Frequency=频率
OSC=振荡器　Shaping function=成形函数
Waveshaped output signal=经波成形后的输出信号

成形函数的范例(Example Shaping Functions)

如图 6.27 所示,如果波形表 w 内成形函数为从−1 到+1 直的对角线,那么输出 w 就会是输入 x 的精确复制。这是因为对应表将输入值的−1(函数的最下方)对应到输出值的−1(函数的最右方),0 对应到 0,1 对应到 1,以此类推。因为这种简单的输入与输出间的关系仅发生在成形函数为直线的时候,所以我们称此输出为输入的线性函数(linear function)。

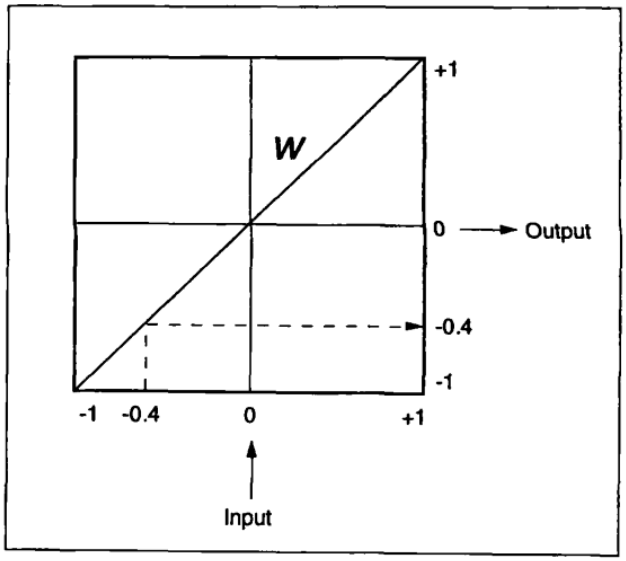


图 6.27 线性关系的成形函数。此函数将以下方坐标为基准的输入信号对应到以右方坐标为基准的输出信号。要检视函数如何将输入对应到输出值上,从下方纵向读取,然后向右找到相对应的输出值。所以当输入值为下方的−0.4,将对应到右方的−0.4。这种输出与输入的一致性仅在线性成形函数中才会发生。
Output=输出 Input=输入

如果成形函数表内的关系并非−1 到+1 的直线,那么成形函数 w 就会使 x 失真。图 6.28 显示数种成形函数作用于相同正弦波输入的结果。图 6.28a 显示一个颠倒的成形函数。对于每个输入振幅的正值,成形函数都会将其转化为相对应的负值,反之亦然。图 6.28b 则是直线,但是角度要比图 6.27 小一些。所以对应到右边(输出)的成形函数区域也较小,表示将减弱输入信号。图 6.28c 则将较弱信号增强,将较强信号送入截断失真(clipping distortion)中。波成形的振幅波动特性在图 6.28d 中可以看得更清楚。成形函数在 0 附近,即低强度的部分为直线。低强度的输入信号经过此函数将不会被失真。当输入振幅增加,成形函数的两端会令输入信号变成非常复杂的失真形状。

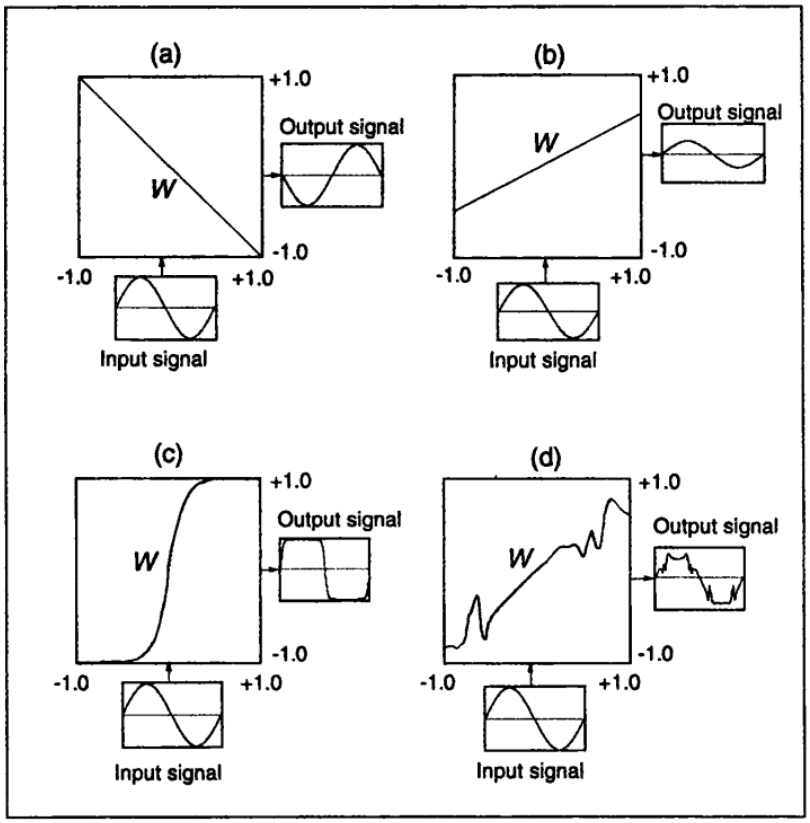


图 6.28　四个成形函数。(a)输入信号的颠倒;(b)减弱;(c)增强(延伸)低强度信号,并造成高强度信号截断失真;(d)复杂的振幅波动失真。
Output signal=输出信号　　Input signal=输入信号

成形频谱的振幅灵敏度(Amplitude Sensitivity of Waveshaping Spectrum)

不难发现波成形的振幅灵敏度可用来模仿真实乐器的特性。也就是说,当更用力演奏乐器时,比方说,更用力地弹奏吉他,更刺耳地吹奏萨克斯管,或是用力打鼓,都会使得频谱更丰富。在波成形合成中,我们可以将整体振幅随着时间改变的信号送入成形函数,来模仿这些效果。当输入信号的振幅改变,就能得到相对应的时变输出频谱。换一个说法是,输入信号时域上的改变,将以输出信号频域上的改变显现出来。这是一个很重要的特点。只要简单地改变信号的振幅或改变输入信号的偏移量(offset),以便应用成形函数的不同区域,那么我们只需要一个成形函数(预先计算好,储存在内存内)便能得到许多输出波形的变化。所以波成形合成是种非常有效率的合成技术。阿菲博(Arfib 1979)在波成形的具体音乐应用上给出了该技术的实用范例。

切比雪夫成形函数(Chebychev Shaping Functions)

勒布伦(LeBrun 1979)与阿菲博(1979)的研究,证明了能在数学控制的状态下准确预测波成形技术所产生的输出频谱。通过将信号 x 限于恒定的余弦波上并使用名为切比雪夫函数的一组平滑多项式,后者值域为[-1,+1]以建立成形函数 w,我们便能轻易在稳态频谱下创造出任何想要的谐波组合。此原理来自下面的公式:

$$T_k\times(\cos[\theta])=\cos(k\times\theta)$$

这里,T_k 是第 k 个切比雪夫函数。换句话说,通过将第 k 个切比雪夫多项式应用到一个输入正弦波中,我们就能得到在第 k 个谐波上的余弦波。这表示将每个独立的切比雪夫多项式当作成形函数使用时,会制造出输入 x 的特定谐波。通过加入一个调整了的切比雪夫多项式组合,再将所得的结果放入成形函数表中,我们就能将其作为波成形技术的输出,来获得相对应的谐波混合。比方说,为了获得带有第一谐波(基频)和振幅为基频的 0.3 倍的第二谐波,以及振幅为基频的 0.17 倍的第三谐波这样的稳态波形,我们需要加入一些算式。

$$T_0+(0.3\times T_2)+(0.17\times T_3)$$

然后,将相加的结果放入转移函数表中。如果将余弦波送入此表,那么输出的频谱就应该有预期的谐波比例。

使用切比雪夫函数的好处是,我们可以确保波成形器的输出是在限定带宽内。也就是说,不会有超过奈奎斯特率的频率,所以不会发生折回失真(foldover distortion)。表 6.1 列出了当 x 等于 $\cos(q)$时的 T_0 到 T_8 的算式。

表 6.1 切比雪夫函数的 T_0 到 T_8

$T_0=\mathbf{x}$
$T_1=\mathbf{x}$
$T_2=2x^2-1$
$T_3=4x^3-3x$
$T_4=8x^4-8x^2+1$

$T_5 = 16x^5 - 20x^3 + 5x$

$T_6 = 32x^6 - 48x^4 + 18x^2 - 1$

$T_7 = 64x^7 - 112x^5 + 56x^3 - 7x$

$T_8 = 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1$

振幅规格化(Amplitude Normalization)

如图 6.28 中的简单波成形乐器所看到的，波成形合成的主要缺点在于，即便只使用单一成形函数，其输出振幅的变化也是过于剧烈。此变化是因为使用了不同部分的成形函数，也就是说，这种变化是根据作用于成形函数的输入信号的振幅来决定的。

在波成形中，x 的振幅实际上是用来控制音色，而不是控制整体声音的强度。如果我们要完全独立地控制音色与输出音量，必须做某些形式的振幅规格化。至少有三种可能的规格化：响度规格化(loudness normalization)、功率规格化(power normalization)以及峰值规格化(peak normalization)。

从音乐角度上看，我们的理想是响度的规格化，在响度的规格化中，我们所感受的乐器响度对所有的 α 值是维持不变的。不过，这牵涉到复杂的心理声学的听觉互动，且与音乐具体情境相关，所以大部分的实现方式都过于复杂，计算上也过于昂贵。功率规格化则建立在以特定成形函数所产生的谐波振幅均方根值的除法上(root mean square)。勒布伦(1979)曾详述此技术。峰值规格化可能是三者中最简单也最实用的。它是以最大输出值的关系来调整输出。峰值规格化确保不同声音的输出振幅起码会有相同的波峰值，所以不会形成超过范围的输出，造成数模转换器的过载。

图 6.29 是带有峰值规格化的波成形乐器。最简单的实现方式，是先准备好一个包含应对所有 α 值的规格化要素的表，因为包络 a 决定 x 的振幅。比方说，如果输入规格化表中的 α 值为 0.7，我们就把这个对应于 α 的规格化表中的输入值乘上输出。

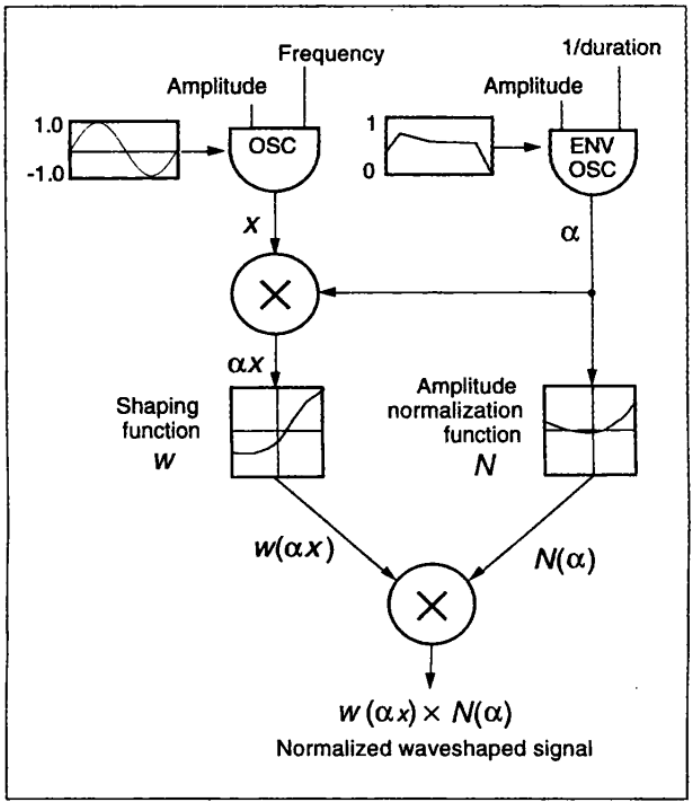


图 6.29 带有峰值规格化的波成形乐器。α 值索引规格化表中的一个数值，按比例改变波成形器的输出。

Frequency=频率 Amplitude=振幅 OSC=振荡器 1/duration=1/时值 ENV OSC=包络振荡器
Shaping function=成形函数 Amplitude normalization function=振幅规格化函数
Normalized waveshaped signal=规格化波成形信号

波成形的变化(Variations on Waveshaping)

传统波成形技术——将余弦波送入切比雪夫多项式成形函数——能产生一个频域的谐波频谱。我们可以通过改变输入信号，或改变成形函数，来扩充这种波成形频谱。另一个可能性是不通过波成形乐器，而是通过另一个信号处理器，如滤波器，来修改声音信号。

如前所述，波成形器的输入 x 可以是任意信号，并不只是余弦波而已。比方说，Reinhard(1981)详细描述当 x 是两个不同频率的余弦波相结合时的情况。另外一种变化则是使用经 FM 调制后的信号作输入信号 x，这样做的优点在于可以得到泛音以及共振峰结构的非和谐式的结合(Arfib 1979)。

信号 x 也可以是取样的或具体的声音。当成形函数 w 是一个简单的平顺多项式，效果有点像是相位调整(phasing)，因为输入的谐波部分会以时变的方

式波动。所以波成形乐器可以产生非常有趣的自然音与电子音的混合。如果成形函数 w 含有任何笔直的横向或纵向线段，那么会造成强烈的失真，就像是晶体管化的电吉他振幅调到最强后的失真。

w 也不一定只能是切比雪夫多项式。使用切比雪夫多项式作为成形函式的最大好处是限频输出，所以不会造成折回失真。但是若这个优点不是最为重要的，那么将可以由任何其他公式产生 w。也可以直接用手工绘制(Buxton et al. 1982)。详见第 8 章，使用噪音调制的成形函数的波成形。

可动波成形(Movable Waveshaping)

另外一种变化称作可动波成形，由北京中央音乐学院(Beijing Central Music Conservatory)的 Xin Chong(译注：中文名暂不可考)发明(Xin 1987)。此技术中成形函数会随着时间改变。这种技术的实现，是通过储存一个较长的成形函数，并在不同时间将索引值移动，以对应成形函数的不同部分。用简单的输入信号以及简单的时变成形函数，我们可以获得非常多样的结果。

分数波成形(Fractional Waveshaping)

德波立(De Poli 1984)把成形函数看作是一个分数，更确切地说，是看作两个多项式间的比例来借以分析成形函数的结构。他称此为分数波成形(fractional waveshaping)。分数波成形可以产生指数频谱的效果，或产生其频谱形状如同经过阻尼的余弦波的效果。多块经阻尼的余弦波频谱，听起来就如同共振峰一样。通过改变输入余弦波信号的振幅及偏压值，我们就可如常规波成形一般，获得动态变化的频谱。

后处理与参数预测(Postprocessing and Parameter Estimation)

经波成形的信号可以再次经过另一个信号处理器的处理，这种技术我们称作波成形信号的后处理(postprocessing)。所谓另一个处理器可以是 AM 振荡器、FM 振荡器或滤波器。比如，通过在谐波频谱内加上非和谐分音，AM 和 FM 就会使波成形的频谱更丰富(Arfib 1979; Le Brun 1979; De Poli 1984)。

德波立(De Poli 1984)与 Volonnino(1984)发展了一种实验性的滤波法，称作由频率依赖式波成形(frequency-dependent waveshaping)。其目的是为了对每个由波成形程序所生成的谐波，提供独立的相位与振幅控制。详见这些技术列举的文献。

比彻姆(Beauchamp 1979)在他的铜管音色波成形模式中加入了高通滤波器,以模仿铜管乐器阻音效果。近期比彻姆与 Horner(1992)又通过多重波成形加滤波器来模仿乐器声音。他们先对乐器音色做参数预测,再用一组波成形器加滤波器的模式求取近似值。由原始声音减去此近似值后,得到差或余冗信号。再用另一组波成形器加滤波器模式模拟此余冗部分。如此使用两个或三个波成形模型,可以得到比单一模式更为接近的模仿。

泛调制(General Modulations)

只要把时变函数取代原始技术算式中的常数,许多合成技术都可以改为调制技术。如果时变函数是周期性的,那么此技术可算是波形参数调制(waveshape parameter modulation)类合成技术的一种。比方说,AM 调制及 FM 调制可以算做波形参数调制技术之一。要深入了解该分类的架构,可参见 Mitsuhashi(1980)。

穆勒(James A. Moorer 1976)认为单一 FM 的算式,是称为离散求和公式(discrete summation formulas, DSFs)种类的一个例子。离散求和公式(DSFs)指的是一组公式,它们是有限或无限的三角数列相加之和的闭合式解(closed form solution)。所谓闭合式解意味着比一个较长的求和公式更简约、更有效率的表现方式。如果我们假定这些公式描述由正弦波所叠加的波形,那么这些公式便与声音合成有关。比方说,下面公式的右半边是左半边之和的封闭型解:

$$\sum_{k=1}^{n}\sin(k\theta)=\sin[1/2(n+1)\theta]\sin[(n\theta)/2]\csc(\theta/2)$$

此公式说明我们可以仅用五个乘法、三个除法以及三个查表运算来表示 n 个正弦波相加的结果。作为一闭合式公式,DSFs 仅须操控数个参数,便可简单地以数字方式实现。穆勒的论文中叙述四个 DSFs,展示了声音合成的前景。另外也有一种更宽泛的 DSFs 分类方式(Hansen 1975),但大多数在音乐合成上都没有用处。

有些 DSFs 可以产生时变声音,听起来很接近 FM 频谱。穆勒也叙述 DSFs 可以产生简单 FM 所无法产生的频谱,如单边带频谱,其泛音仅在载波频率的一边。另外一类采用 DSFs 的频谱,能产生单是振幅增大(也就是通过常数要素)的泛音。

相对于如 FM 的技术而言,DSFs 的缺点在于缺少振幅规格化。所以必须对 DSF 合成算法的输出使用某种缩放或规格化处理。(见波成形一节对于振

幅规格化的讨论)。对技术有兴趣并且有兴趣探索 DSF 方法的读者,可以参考 Moorer(1976,1977)或 Moore(1990)。

结论(Conclusion)

信号调制是音乐效果与声音的丰富技术资源。由于 AM 与 RM 在无线电通讯上的应用,它们拥有悠久历史。在听觉频带间,它们能够产生典型的"无线电声(radiosonic)"。但是它们较 FM 更受限,部分原因是因为它们无法产生与 FM 同等数目的边带,部分原因是因为 FM 参数的弹性较大。在 FM 技术中,经过数十位美、日工程师多年的耐心研究,得到了许多成果。音乐家也投下了许多时间去调整 FM 乐器的参数,以创造出许多种有趣的声音与音色。

基本调制技术的缺点是调制公式本身所固有的。由调制技术产生的声音频谱受限于数学公式,使其限定在某类行为上。在实践中,这意味着每种简单调制都有一种特别的声音"记号",接触过此技术的人便能够分辨得出来。

此声音记号可能会是很恼人的陈腔滥调,或是很吸引人的音乐力量,这要视作曲家的技巧而定。在后续的应用中,Louis 和 Bebe Barron 为科幻片 *Forbidden Planet*(1956)所作的电子音乐原声带,即是调制技术在音乐应用上的杰出范例。在未来,会发展出更精细的合成技术,但如何艺术性地应用调制技术,仍旧依赖人内心的深层感性。

第7章 物理模型与共振峰合成
(Physical Modeling and Formant Synthesis)

"自太初之时，最早的人类远祖第一次张开口，发出的第一个音节，在乐园的芳香气息逸散开来为始，直到今日为止，没有任何事物能够完美地逼近这上天所赋予的'语言之音乐'。虽然为了达到这看似不可能的目的，不断地有许多尝试，但终究徒劳无功。直到幸运而且令人称奇的"机器头脑"的发明者，君士坦丁堡的吉亚克普·桑古斯先生（几与奇迹相比）发明了最好、最精妙建构的机械，能够与自然之母所赋予的人声相提并论。"〔《人类学家或者机械歌唱家》（*Anthropoglossos or Mechanical Vocalist*）的描述，London ca. 1835，Ord-Hume 1973 重印。〕

本章介绍三种相互重叠的合成方法，每种技术都是为了仿效原音的发声方式。物理模型合成（Physical modeling synthesis）建立传统原音乐器的模型，如气流经过吹嘴，进入共振管内。卡普拉斯-斯特朗合成（Karplus-Strong synthesis）作为一种物理模型合成的简单变化，可仿真拨弦乐器的声音，如吉他、曼陀铃、古钢琴也可产生类似鼓声或其他声音。共振峰合成（Formant synthesis）是一组在频谱上产生波峰的技术。这些技术可仿真人的发音器官的共振，也能仿真传统与合成乐器。我们将由物理模型合成开始，依次介绍每一种技术。

物理模型合成（Physical Modeling Synthesis）

物理模型（PhM）合成（Physical Modeling Synthesis）是从乐器发声的物理声学的数学模型开始的。也就是说，物理模型的公式描述乐器发声的机械与声学行为。此方法也被称作规则合成（synthesis by rule）（Ferretti 1965，1966，1975），第一原理合成（synthesis from first principles）（Weinreich 1983）以及近期出现的虚拟声学（virtual acoustics）（Yamaha 1993）。

物理模型合成的目标有两个方面：一是科学上的，另一个是艺术上的。首先，物理模型研究以数学式及演算逻辑，来仿真现有的乐器发声机制所能达到的程度。这种方式基于这样一个前提，即仿真越逼真，我们对此系统的了解就越透彻。从这个意义上来说，物理模型是一复杂的机械——声学机制的精确数学模型，而具体实现了牛顿古典力学的理想。（对于机械与声学系统中的波动物理学之简明导论，可参见 Pierce 1974，Crawford 1968，Olson 1991。）

物理模型的第二个目标则更艺术性：仿真物理模型，创造出现实世界无法打造的幻想乐器的声音。在此范畴下，我们所说的幻想乐器，其特质或形状可以随着时间改变——比方说是有弹性的大提琴，可以在乐句内被拉长或缩短，或者是无论用多大力量打击都不会损坏的鼓。物理模型技术通常都是可以缩放大小的，所以比方说，只要有了一面锣的描述方法，就能制造出从直径 30 厘

米到30米，十几面不同大小的锣。从对一根弦的特定描述来推断，音乐家能建立出弦长及粗细与吊桥上的悬吊钢缆相等的虚拟吉他。为了取悦这些音乐炼金术士，我们更可改变乐器的材质——从银到铜，到奇异的木质，到塑料——只要键入几个常数，就是这么简单。

物理模型的特长在于仿真音符与音色间的过渡。动态改变虚拟乐器某些部分的尺寸(如延长共鸣管)，通常能得到令人信服的声音过渡。物理模型的另一个特征是，它们捕捉演出时发生的意外，如嘎吱声(squeak)、锁模(mode locking)及多音和弦(multiphonics)。这些声响是初学者尝试演奏时所不可避免的，但是适量使用，将会加入一种更逼真的味道。PhM合成中，这些声音是因某些参数的设定而自然产生的副产品。相比之下，在加法合成中要得到这些效果，则必须要提供这个声音在各个方面的全部细节。

物理模型合成法不试图去创造某一乐器"完整的"物理模型。不处理乐器所有可能发生的情况，而仅需处理在高度限制的演奏情况下的乐器物理情况。在演奏中，演奏者通常只使用为数不多的，具有该乐器特点的演奏方式，这种相对低带宽的控制讯息可以用软件简洁地表示出来。

物理模型合成的效率(Efficiency of Physical Modeling Synthesis)

物理模型合成是由许多研究者经超过30年所发展的一组技术。因为这些技术的许多数学特性，以及大量的计算需求，物理模型从实验室研究到进入音乐工作室的脚步较为缓慢。

只有在近几年内，才发展出了对某些物理模型合成较有效率的实现方式(McIntyre, Schumacher, and Woodhouse 1983; Smith 1986, 1987a, b, 1992; Keefe 1992; Adrien 1991; Woodhouse 1992; Cook 1991a, b, 1992, 1993; Borin, De Poli and Sarti 1992)。这些有效率的运算法(如波导)是以一般数字信号处理架构为基础，如延迟线、滤波器以及查表运算等。然而，一般来说，这种效率的提升是以过度简化为代价的，这代表通常它们产生的是"类似乐器的声音"，不一定会十分逼真。这并不是说这些仿真并不有趣。从作曲的观点来看，这种灵活的类似乐器声响可能很有用。Woodhouse(1992)则说明了现代模型所面临的不足。

本章将描述一些有效率的方式，如模型合成、McIntyre、Schumacher和Woodhouse合成以及波导合成法，也会谈及"古典的"或运算密集型的方法。我们也将介绍一种称为卡普拉斯－斯特朗合成(Karplus-Strong synthesis)的高效率方法。

背景:物理模型(Background: Physical Modeling)

物理模型所用的观念、术语以及某些公式,可以溯源至19世纪关于声音性质的论文,如雷利(Lord Rayleigh)的杰出著作《声音理论》(*The Theory of Sound*,1894/1945)。雷利详尽阐述了振动系统,如膜、盘、棒、壳等的原理,并以数学物理描述空气中、管中与箱子中的振动。其他19世纪先驱则建立了机械模型,来仿真乐器的物理性质(Helmholtz 1863, Poynting and Thomson 1900, Tyndall 1875, Mayer 1878)。随着真空管的发明,也建立了模拟电子模型(Steward 1922, Miller 1935, Stevens and Fant 1953)。在奥尔逊(Olson 1967)的文章里可以看到使用模拟电路所建立的物理模型,包含打击乐器、笛类乐器、号类乐器、拨弦乐器以及人声等。但在计算机发明之前进度缓慢。

贝尔实验室的约翰·凯利(John Kelly)与卡洛尔·洛克鲍姆(Carol Lochbaum)是将人类声道的物理模型移到计算机上的先驱(Kelly and Lochbaum 1962)。他们计算机版本的《双人自行车》,收录在1960年由马修斯(Max V. Mathews)制作的贝尔实验室 *Music from Mathematics* 一碟中,成为当时数字计算机能力与日俱增的象征。〔斯坦雷·库布立克(Stanley Kubrick)的电影《2001太空漫游》(*2001: A Space Odyssey*),曾提及这一成果。当超级计算机HAL被拔去数万枚芯片,如人临死前忆起它学会的第一首歌。不过电影中的版本是由人演唱的。〕

伊利诺伊大学的希勒(Lejaren Hiller)、比彻姆(James Beauchamp)与鲁伊斯(Pierre Ruiz)是最早利用物理模型来仿真乐器合成的(Hiller and Beauchamp 1967, Ruiz 1970, Hiller and Ruiz 1971)。他们的研究集中在合成振动物体的声音,如弦、柱、盘以及鼓膜等,以拨弦或敲击的方式使之进行运动。另一个物理模型合成的先驱是费雷蒂(Ercolino Ferretti),在1960年与1970年间曾指导过麻省理工学院(MIT)、哈佛大学与犹他大学的学生在这方面的研究(Ferretti 1965, 1966, 1975)。

利用波导(waveguides)合成声音的兴趣是由卡普拉斯-斯特朗发明了拨弦算法而引起的(本章后半部分将详述)。此种节省运算方法的出现具有偶然性,而不像是在物理模型方面有意尝试的结果(Karplus and Strong 1983, Jaffe and Smith 1983)。Keefe(1992)对自1963年以来其他的发展做了总结(同时参见Fletcher and Rossing1991)。1993年Yamaha公司推出了以波导为基础的商用合成器,VL1与VP1。

激励与共振(Excitation and Resonance)

"问题:管乐器的共振模式(resonant mode)并非完美和谐,但是音高却可以完美和谐。另外,打击乐器有非和谐共振,而产生非和谐声音,差别在哪里?"

"答案:关键在于要考虑的并不只是共振模态,或是它们被如何放置的,而是此乐器如何被激发的(excited)。如果你捡起小号,并用锤子敲它,那么声音就会是敲击声。如果你拿起小军鼓,用振动器激发它,那么声音就会是和谐的。"(B. Hutchins 1984)

物理模型合成的基本原理在于激励器(exciter)与共振器(resonator)间的互动。激励(excitation)指的是造成振动的动作,如琴弓摩擦,鼓槌敲打或者气流吹动。共振(resonance)则是乐器本体对于激发的反应。从信号处理的观点来看,乐器本体相当于是对激励信号的时变滤波器。

一般来说,激励器有非线性行为,而共振器有线性行为。我们曾在第5章中接触过此主题。"线性"声学系统的直觉解释,就是输出会成比例地对输入能量有所反应。如果我们将两个信号放在系统内,便可预期输出结果是其相加之和。而"非线性"则指系统内有个内定的门槛值,如果超过的话,会使得系统以新的方式反应,如同开启开关。

激励器/共振器的互动可以分成两种基本类型:退耦〔decoupled,又称前馈(feedforward)〕,以及耦合〔coupled,又称反馈(feedback)〕。在减法合成技术,如线性预测编码中(见第5章),激励信号来源注入共振滤波器。在来源与激励间除能量从激励器传递到振荡器上,没有其他的互动。

相对的,萨克斯管的发声机制则是耦合激励的例子。此"耦合"代表共振部分的振动会回授到激励部分。例如,在最初被嘴吹出的气流激励之后,簧片的振动频率,会被乐器共振管身(bore, tube)的声音回授所强烈影响。

这种激励与共振间的互动,创造了我们所听到的许多大师演奏中声音的变化与精妙处。由于物理模型技术可以模仿这种互动,所以它们倾向于传达出声音背后的姿态感觉(Florens and Cadoz 1991; Adrien 1991)。与此相对立的,是与声音背后的姿态感觉无关的由数学公式所控制的抽象合成方式。

在某些物理模型合成的实现上,激励来自于演奏者演奏的输入装置(或表演控制器)(Cadoz, Florens, and Luciani 1984; Cook 1992)。详见后述的物理模型合成输入装置。(关于对一般音乐输入装置的讨论,见第14章。)

古典物理模型的方法论(Classical Physical Modeling Methodology)

物理模型的"古典"方法可以由早年 Hiller 与 Ruiz(1971)的研究,以及后续

的许多研究者为代表。古典方法论如下所述。

首先，先指定振动物体的物理尺寸及其常数，如其质量或弹性。因为原声乐器中，声音是由振动物体如弦、簧片、鼓膜、管中气流或乐器本体所发出。

其次，规定限制振动物体的边界条件(Boundary conditions)。这些是不能超越的变量的限制值。边界条件也允许这样一种可能——系统在前一个输入后尚未完全"平静"或稳定。

需要指定初始状态(initial state)，比方说，弦在平静时的起始位置。

再次，以算法描述激励，表示以某种方式作用在振动物体上的作用力。典型原声乐器的激励包含敲击性来源，如鼓键、槌以及钢琴琴键的运动；空气来源，如簧片间的气流，以及弦乐器的琴弓。可用此算法描述激励器与共振器的耦合。

阻抗(impedence)效应也必须要加以考虑。阻抗是对于驱动力的阻力；在高阻抗的介质中，需要用较大的力量来产生较小的振幅。当波由乐器的一部分传递到另一部分时，不同部分的阻抗会改变波的传导。比方说，假设两根接合在一起的弦，一部分远比另一部分重。如果我们拨了较轻的弦，那么波撞到较重的弦后，几乎所有能量都会被反弹回来到较轻的弦上。但如果两弦的阻抗相同，就不会有反射。研究者已量取了各种乐器成分的阻抗，并能在物理模型内安插适当的公式(Campbell and Greated 1987)。

最后，也须明确因摩擦以及声音放射的形态，而产生的滤波效果，作为对振动条件的进一步限制。

在此，我们得到一套相当复杂的公式系统，代表乐器的物理模型。相对应的波动方程(wave equation)结合了所有因素，受初始状态与激励所影响(Morse 1936)。之后可由反复连续近似程序解出波动方程，试图同时为很多互相依赖的变量，找出合理数值解。此方程式产生一离散取样值，表示某个时间点上的声音压力波。

在古典方法论底层的，是一组基于质量—弹簧模型(mass-spring paradigm)的差分方程(difference equations)，也就是下节所要描述的振动结构模型。

差分方程(Difference Equations)

在使用古典手法进行物理模型合成时，声音取样是通过对描述一个实体的振动行为的差分方程(difference equations)的评估来产生的。差分方程包含方程式的差与微分。这些方程式在描述时变信号时经常使用。巧合的是，贝尔努利(Joseph Bernoulli)在1732年的第一个差分方程的应用，即是在仿真有限长度的弦振动—物理模型合成的核心技术。差分方程也描述数字滤波器的动作。所有第10章的FIR与IIR滤波器的等式都是差分方程的例子。(对差分方程

更仔细的讨论，可见 Rabiner and Gold 1975 或其他数字信号处理的教科书。）

物理学家使用差分方程描述物理量的改变的定理。以此方式描述现象时，第一步是找出可以精确描述仿真现象的状态变量的最小数值。下一步是设定最简单的差分方程，其精确描述支配这些变量变化的物理定理。某类差分方程有一般代数解，而其他的只能由耗费时间的逐次近似计算法（successive approximation）解出（Press et al. 1988）。在这些方法中，是先猜想一个解，之后在连续迭代地求更逼近的解。

弦振动的质量—弹簧模型 (The Mass-spring Paradigm for Vibrating Strings)

乐器中的弦振动令科学家与音乐家着迷了数个世纪。所以不令人意外地，希勒与鲁伊斯（Hiller and Ruiz 1971）的先驱性研究是由弦振动着手。他们解出在弦中央、接近底端与在底端拨弦或擦弦时的差分方程。运弓的速度，使用的压力以及摩擦系数都被作为初始条件的一部分。他们同时也考虑其他因素，包括空气的阻力、弦的粗细、琴桥的移动，从琴桥到共振体的能量传输，以及由共鸣箱发散的能量等。

在近期的试验中，这项研究是以传统方式，将弦仿真成一连串由弹簧连接在一起的离散质量中心。长久以来，此质量—弹簧模型被物理学家与声学家用来描述振动物体以及其放出的波（Crawford 1968；Benade 1990；Cadoz，Luciani，and Florens 1984；Weinreich 1983；Smith 1982，1983；Hutchins 1978；Adrien and Rodet 1985；Boutillon 1984；Chafe 1985）。此质量—弹簧模型捕捉了振动介质的两个重要性质。首先，振动介质带有密度，也就是介质的每单位质量。对于弦而言，其密度可以由量测其重量得知。其次，振动介质是有弹性的。如果介质的某一个部分离开了平衡位置，马上就会有一股恢复力将它拉回。如果我们拨动弦上的某部分，产生扰动，那么介质的位移部分便会在邻近部分施力，依序造成下个部分的移动，并且继续下去，此过程称为波动传导（wave propagation）。因为介质带有质量，该部分不会马上自其平衡位置移开，而需要一点点时间，所以拨弦的冲力会以一定的速度在介质中传导。

图 7.1a 以数个由轻弹簧连接的相同质量块来描述弦振动。如果第一个质量块向右移动，那么第一个弹簧将被压缩，施力在第二个质量块上（图 7.1b）。然后第二个质量块向右移动，压缩第二个弹簧，以此类推，如图 7.1c 所示。由于质量块依次的位移与扰动传播的方向一致（也就是水平的），此被称为纵波（longitudinal wave）。

图 7.1d 与图 7.1e 显示横波(transverse wave)的传播,其最初位移是垂直于波传导的方向。当琴弦被拨、敲击或被琴弓拉过时,这是波振动的主要形态。另外一种振动是旋转式(rotational),但在声音合成中并不常用。

将弦拆解成离散的质量块,有其计算的优点。在弦上某点的激励效果,可以被仿真成作用在单一质量块上的作用力,经由弹簧传递到其他质量块。当弦被敲击后,某个时间点弦的形状可以由一系列差分方程解出。

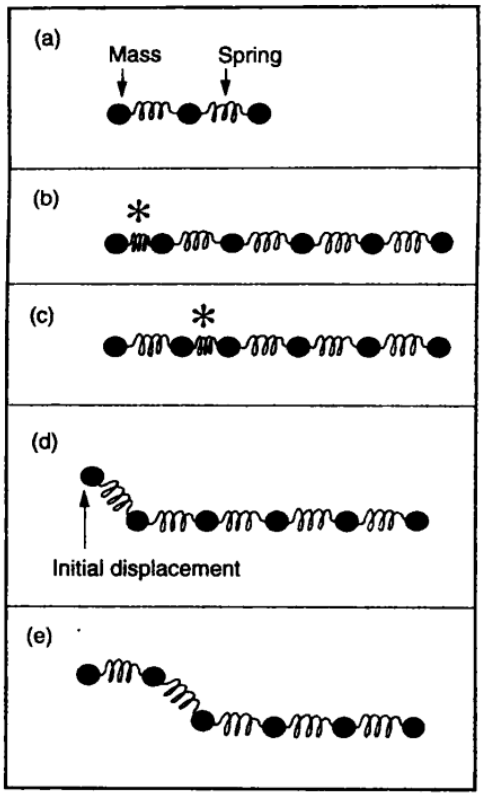


图 7.1 弦振动的质量-弹簧模型。(a)弹簧模仿弦的弹性;(b)在纵波中,扰动与波动传导的方向是一致的,起始移动(弹簧的压缩)由星号标出;(c)跟随状态;(d)在横波中,起始的扰动垂直于波动传导的方向;(e)跟随状态。
Mass=质量 Spring=弹簧 Initial displacement=初始位移

平面与体积的质量－弹簧模型
(The Mass-spring Paradigm for Vibrating Surfaces and Volumes)

质量－弹簧的表示方式可以延伸到振动平面及体积上。可以把质量的分布与多个弹簧加以连接来对表面进行仿真(图 7.2a),或以圆形的排列来仿真鼓皮(图 7.2b)。当质量的相互连接多于六个方向时,体积就形成了方格

形状(图 7.2c)。

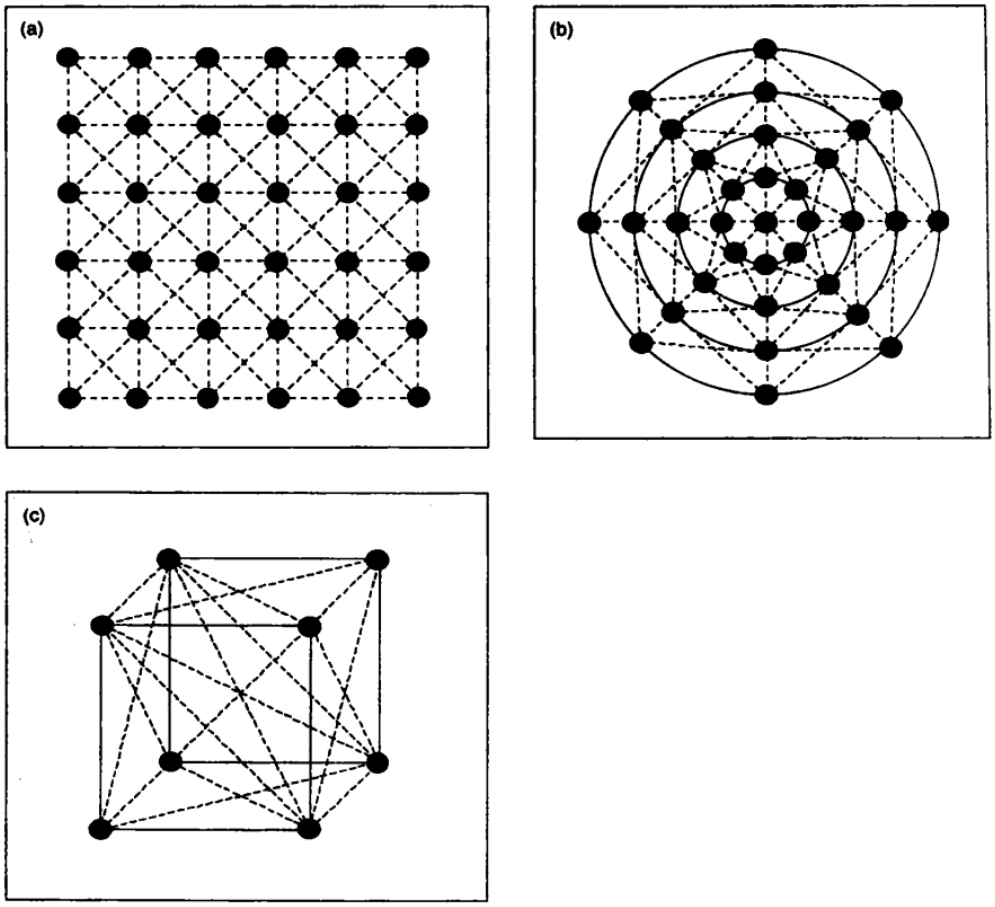


图 7.2 以弹簧连接质量的模型,表示振动平面或物体。黑点代表质量,线段代表弹簧。(a)振动平面的模型;(b)鼓面的模型,质量与弹簧以同心圆方式排列;(c)振动立体可以被仿真以格状结构排列的质量,六边分别为弹簧所连接。

激励的质量－弹簧模型(The Mass-spring Paradigm for Excitation)

到目前为止,我们已经描述了作为共振器模型的质量系统以及线性弹簧。如果可定义弹簧带有非线性行为,它们就能当作良好的激励模型。在物理模型法中常被用做激励器的非线性振荡器,可以视为以质量－非线性弹簧表示的模型(Rodet 1992)。质量表示其惯性行为,而非线性弹簧表示激励器本体的弹性。非线性摩擦成分表示激励器与共振器间的接触情况。举例来说,如此的表示方式可以应用在如琴槌敲打在钢琴弦上的情况(Suzuki 1987)。

模态合成(Modal Synthesis)

“拥有许多个移动部分的复杂系统的运动,可被视为较简单运动的混合,这些被称为模态的运动都是同时发生的。无论系统有多么复杂,我们会发现每个模态都有非常接近于简谐振荡器的性质。”(克劳弗德 F. Crawford 1968)

模态合成(Calvet, Laurens, and Adrien 1990; Adrien 1991)是质量－弹簧模型的一种替代方案。它所根据的是,发声物体可以被表示成一组振动次结构的集合。次结构的数目通常远比质量－弹簧法来得少。典型的次结构包含小提琴琴桥、琴身、发声管、钟、鼓面等。在质量－弹簧模型中,次结构会反应外界所作用的激励(作用力、气流、压力或者移动)。当它们被激发时,每个次结构有其自然振动模态(modes of vibration),这些模态是专属于一个特定的结构,并与数个物理因素有关,在此我们将不深入介绍(见 Benade 1990)。使用模态合成的好处是:由于其在工业上的频繁应用,现在已经有了定义完整的使用模态分析的方法(Hurty and Rubinstein 1964, Hou 1969),这些方法可以被改写为声音合成所用。参见博克(Bork 1992)对于乐器模态分析的简述及其他参考资料。

模态合成将每个次结构描述为一组模态资料,包括:(1)次结构共振模态的频率与衰减函数;(2)一组表示振动模态形状的坐标。所以乐器的一般性瞬间振动,可以被表示为每个模态所贡献的总和。

在阿德仑(Adrien)的实现方式中,瞬间振动可由结构中所选定的 N 个点所组成的 N 维向量所描述。这些坐标,以及接近于乐器特性的几何与机械特征结合在一起。N 点的组合相当于 N 组的模态资料。可以用 N 点的相对位移来描述一个振动模态。

对于单纯振动结构,如不带阻尼的弦,模态资料可在机械工程文献中找到其方程式。对于较复杂的振动结构,我们可以直接对实际乐器做实验,量取其模态资料。对于这种机械工程的分析工具,如传感器(Transducer)与分析软件,皆已在工业应用中所使用,如航空器设计等,所以研究者可轻易取得这些工具。

模态方法优于质量－弹簧模型之处在其弹性。这是由模态次结构调制性的特性所决定的。模态合成将发声机制分成独立部分,可以增加或减少它的次结构,以创造时变合成效果,如“拉长”或“缩短”乐器的尺寸。此方法同时允许以非自然方式结合次结构,创造出由一个乐器到另一个乐器音色的间插。

摩赛克:模态合成的实际应用
(MOSAIC: A Practical Implementation of Modal Synthesis)

由阿德仑(Jean-Marie Adrien)与约瑟夫·莫里松(Joseph Morrison)发展的 MOSAIC 系统,是以模块软件工具形式所实现的模态合成(Morrison and Waxman 1991,Morrison and Adrien 1991)。为了教学的目的,我们将在此完整介绍一个范例。

在摩赛克(MOSAIC)的世界中,你坐在有一组对象的虚拟工作台前,将这些对象组合成乐器。这些对象包含弦、空气柱、金属片、鼓膜以及小提琴与大提琴的琴桥。其他对象可激发乐器,如弓、槌子或是琴拨。对象之间的互动称为连接(connections)。连接可以想象成是接通两个物体间的一个黑盒子,并且明确两个物体之间的关系。比方说,两物体可以用黏着、拉弓、拨、打、推等方式连接在一起。每个连接上有控制器(controllers)——一些规定控制参数的旋钮。比方说,弓的连接有规定诸如运弓速度、松香的量等控制器。最后,对象的物理位置称为接入(access)。比方说要连接两个物体时,我们必须规定它们的接入。

图 7.3a 是用对象、连接、控制器与接入所建立的范例。此范例程序代码见图 7.3b。

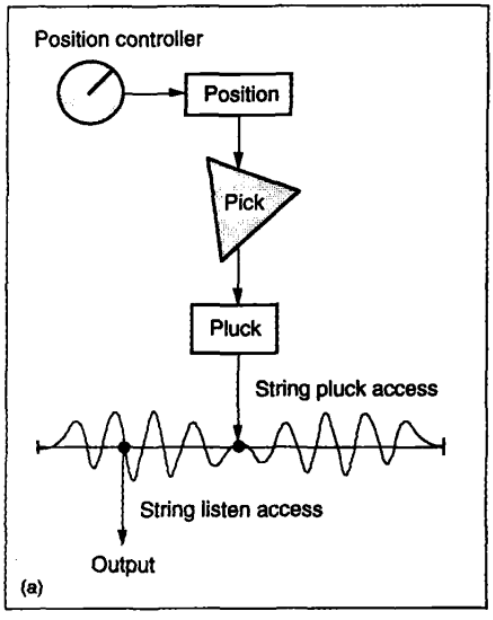


图 7.3　由 MOSAIC 程序所仿真的拨弦乐器。(a)图像表示方式。(b)表示(a)的 MOSAIC 码。以分号开始的行是批注,详见本文对程序代码的说明。
Position controller=位置控制器
Position=位置
Pick=挑
Pluck=拨
String pluck access=拨弦接入
String listen access=弦监听接入
Output=输出

(b)

```
;;; MOSAIC plucked string example, written in Scheme

;;; Make string and plectrum objects

(define my-string (make-object 'monostring))
(define my-plectrum (make-object 'bi-two-mass))

;;; Make pluck connection between plectrum and string

(define my-string-pluck
      (make-access my-string (const .6) 'trans0))
(define my-plectrum-pluck
      (make-access my-plectrum (const 1) 'trans0))

(make-connection 'pluck my-string-pluck
      my-plectrum-pluck 0 .1 (const 50))

;;; Make position connection to push plectrum

(define my-plectrum-move
      (make-access my-plectrum (const 0) 'trans0))

;;; Move plectrum from .1 meter to -.5 meter in .5 secs

(make-connection 'position my-plectrum-move
      (make-controller 'envelope 1
                       (list (list 0.00  .1)
                             (list 0.50 -.5))))

;;; Make listening point on string

(define my-string-out
      (make-access my-string (const .3) 'trans0))

(make-point-output my-string-out)

;;; Run the synthesis and play the sound

(run 2) ; Make 2 seconds of sound
(play)
```

图 7.3 （续）

这个范例是用 Lisp 程序语言的一个分支，即 Scheme 语言（Abelson and Sussman 1985）来写的。Scheme 语言遵循以下句法形式：

（*function arguments*）

此行表示先指定“动词”或动作，后面是该动作的自变量。当出现嵌套插句时，会先执行内部的，再执行外部的。比方说此命令：

〔define my-string (make-object'mono-string)〕

会建立名为 my-string 的对象，并将它放在虚拟控制台中。当摩赛克执行

此命令时，它将执行完整的模态分析。此名称 my-string 将指向由此分析产生的资料。除了弦，我们还需要琴拨：

```
〔define my-plectrum (make-object'bi-two-mass)〕
```

我们要告诉摩赛克使用琴拨来拨弦，但是摩赛克需要我们确定接入点。这可由下式定义：

```
〔define my-string-pluck
(make-access my-string (const . 6)'trans0)〕
(define my-plectrum-pluck
(make-access my-plectrum (const 1)'trans0))
```

my-string-pluck 与 my-plectrum-pluck 是两对象接触点的名称。下一行则连接拨弦动作。

```
(make-connection'pluck my-string-pluck
my-plectrum-pluck 0 . 1 (const 50))
```

在'pluck 后的第一个自变量，是被拨的对象(object plucked)与拨片(plucker)的接入点。下两个自变量说明被拨的对象位置为 0，而拨片的位置距离该点 0.1 米。第三个自变量指定控制器数值，决定何时释放琴弦。数字 50 是作用力，单位为牛顿(牛顿为力的单位，一牛顿的力将对一公斤的质量每秒加速一米)。当拨片力道超过 50 牛顿时，期间的连接脱离。下一行则在琴拨上产生第二个接入，以便它可以被包络控制器移动。

```
(define myplectrum-move
(make-access my-plectrum (const 0)'trans0)
(make-connection'position my-plectrum-move
(make-controller'envelope 1
(list (list 0.00 . 1))
(list 0.50 -. 5))))
```

包络的数值是由成对的形式(时间数值)指定的。list 函数将从两个列表的成对数值建立新的列表。最终的表达式(define my-string-out...)建立聆听弦

发声及命令乐器演奏的接入。

MSW 合成(McIntyre, Schumacher, and Woodhouse Synthesis)

另外一种物理模型的方法是 McIntyre, Schumacher, and Woodhouse 模型(1983)。他们描述了一种优雅而高度简化的乐器发声机制模型。由此基础开始:振荡(自我延续的往返振动)会在木管中、琴弓拉过的弦上以及管风琴的风管中产生音高。MSW 将焦点集中在音高的时域行为。也就是说,他们研究波形的生成与变化,以及在这些现象之后的物理机制。在 MSW 以前,前人的研究(如 Benade)着重在决定乐器声响的共振频率上。但是这并没有考虑乐器波形的重要细节,如起音时的过渡。MSW 的时域方法得以窥探在乐器层面上波形变化的物理原因,并考虑如弓拉过的弦所产生的音高降低、次谐音以及起音暂态的时长等现象。

研究数种乐器后,MSW 描述了一种高效率的合成方式,称为 MSW 合成。MSW 合成的优势在于其控制参数与那些通过开发音乐演奏发掘出的结果具有相关性。

下一节将讨论 MSW 方法背后的理论,并说明 MSW 合成技术的草图。

非线性激励与线性共振(Nonlinear Excitation and Linear Resonance)

在 MSW 合成中,发声机制可以被分为两个主要部分:非线性激励与线性共振(图 7.4)。在单簧管的 MSW 模型中,非线性激励是由气流吹进单簧管的吹嘴,而簧片相当于某种开关,重复开启关闭进入共振管(单簧管管身)中的气流(Benade 1960,1990)。此开关动作是由吹嘴的压力变化所造成。一开始簧片是在半开放状态,但是当气流进入吹嘴,造成吹嘴内的压力时,将关闭簧片。这又给了气流进入由吹嘴进入管身的机会,而由单簧管的开放端溢散出去,所以簧片会使连续的气流变成一连串的喷气。喷气的频率是由管身的有效长度所决定,有效长度会依照指孔的开关而改变。也就是说,管身中的声波以单簧管所演奏的音高发生共振。管身的质量与硬度几乎完全控制了簧片,而决定音高。此互动将持续由共振器到激励器产生反馈,如图 7.4 所示。所以 MSW 模型说明了激励器/共振器的耦合。

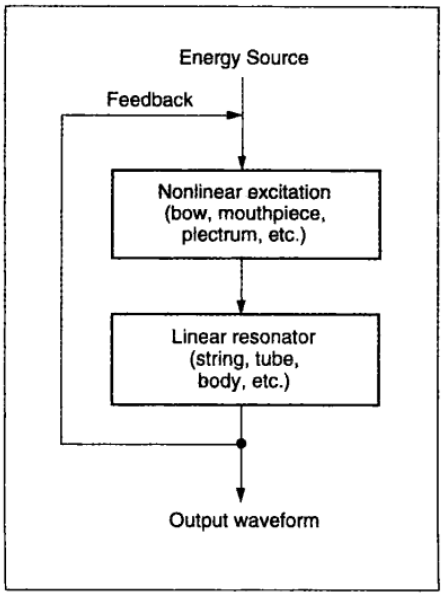


图 7.4　McIntyre, Schumacher, and Woodhouse (1983) 的乐器振荡模型。从线性共振体而来的波反射会影响非线性激励体，构成反馈路径。
Energy Source=能量源
Feedback=反馈
Nonlinear excitation (bow, mouthpiece, plectrum, etc.)=非线性激励(弓、吹嘴、弦拨等)
Linear resonator (strings, tube, body, etc.)=线性共振(弦、吹管、琴身等)
Output waveform=输出波形

在 MSW 合成经弓拉过的弦时，非线性的转换会发生在琴弓与弦摩擦时，弓在一个短暂时间内“捕捉”到弦，之后弦又滑开，被弓“释放”开来。然后再次产生摩擦，弦再次被“捕捉”到，如此反复进行。在长笛或风琴管中，非线性激励是由管的短端建立的气压所产生。当建立起的气压较高时，其释放的压力超过输入的气流，造成气流流入管中短暂的中断。

在以上三种情况中(木管、弓擦弦及风琴管)，激励都是非线性的转换机制，送出尖锐的脉冲波到乐器的线性部分。线性部分的动作有如滤波器，将波形化为该乐器的独特音色。

MSW 合成的草图(Sketch of MSW Synthesis)

对某一乐器，MSW 合成以简洁的方程式组仿真其对象及动作。最复杂，专属于某种乐器的方程式描述其激励。主要变量是能量源(energy source)(双簧管、长笛、风琴管中的气流，或弦乐器中弓的摩擦力)，振动的非线性元素的能量(energy of nonlinear element)，以及描述由系统非线性部分所造成的波形滤波效应的反应函数(reflection function)。要深入了解这些方程式，请参见 McIntyre, Schumacher and Woodhouse(1983)。Smith(1986)和 Keefe(1992)描述了 MSW 的高效率实现方式。他们的实现方式是以对应表及乘法器，替换掉计算上较昂贵的在每个取样点上同时解方程的方法。

由于纯 MSW 模型的许多简化，它做出的声音并不是那么逼真。需要做相当多的调整才能做出令人信服的真实乐器声音模型。比方说，Keefe(1992)描

述 MSW 合成铜管乐器的延伸应用。它使用了一个精细的子程序，来详细说明空气柱(如铜管乐器、长笛或风琴管)的特性，以测试不同设计下的声音精准程度。

波导合成(Waveguide Synthesis)

波导是一种有效率物理模型合成的实现方式，是 Yamaha 与 Korg 于 1993 及 1994 年所采用的合成器引擎(Smith 1982，1983，1986，1987a，b；1991b，1992；Garnett 1987；Garnett and Mont-Reynaud 1988；Cook 1991a，b，1992，1993；Hirschman 1991；Hirschman，Cook，and Smith1991；Paladin and Rocchesso 1992；Van Duyne and Smith 1993)。波导(或波导滤波器)是声波传播介质的计算模型。在音乐应用上，这种介质通常是管或弦。长久以来，物理学家利用波导描述声波在共振空间内的行为(Crawford 1968)。

波导的基本建构要素是一对数字延迟线(digital delay lines)(见第 10 章)。每个延迟线被注入激励波，以相对方向传播，并于到达底端时，反射回中心。延迟线是这种程序的优秀模型，因为波阵面(wavefront)会在一段有限时间内通过共振介质。波在介质内上下来回运动，以其尺寸相关的频率形成共振与干扰。当波导网络在每个方向都对称，受到激发时所产生的声音较偏和谐。如果波导扭曲，改变大小，或是与另一个波导交叉，将改变共振模式。如我们将看到的，人声与乐器如铜管、木管、弦乐器都可由振荡器驱动波导网络来仿真。Garnett(1987)以波导建立了钢琴的简化模型。第 11 章叙述波导产生混响的应用。

波导吸引人的特质是它可以与 Music N 合成语言在相当程度上兼容。这表示波导网络的建构要素可以与标准的单元发生器结合(Link 1992)。

下面四节将叙述拨弦时的波导模型，一般性波导乐器可仿真弦乐器或管乐器，以及更具特性的单簧管与圆号的模型。

击弦的波导模型(Waveguide Model of Struck Strings)

最简单的波导模型也许是单弦琴或单弦乐器。我们可图解此模型，解释当敲打弦上某点时的状况：两股波由冲击点向相反方向传播(图 7.5)。当振波到达琴桥，某部分能量会被吸收，某些能量则会以相对方向反射回去——朝向冲击点，并在两波交会后继续前进，而造成共振与干扰。在波导的文献中，琴桥的作用相当于散射结(scattering junctions)，因为它们将能量散开至所有连接的波导上。

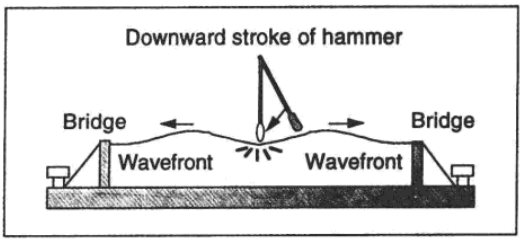


图 7.5 敲在弦中央时，会产生两个波，朝向相反方向移动。这是弦振动的延迟线模型基础。
Downward stroke of hammer=槌向下敲击 Bridge=琴桥(马) wavefront=波阵面

一般性的波导乐器模型(Generic Waveguide Instrument Model)

图 7.6 说明一般性的波导乐器模型，能够仿真弦乐器或管乐器(Cook 1992)。尖锐的非线性激励波注入延迟线，直到它撞上散射结后，将一部分能量发散出去，并反弹部分能量。散射结可能是线性或非线性滤波器，它可以仿真手指或琴弓压在弦上，或是木管乐器的指孔上的情形。在底端的滤波器仿真琴桥、琴身或是管口(bell)的效应。

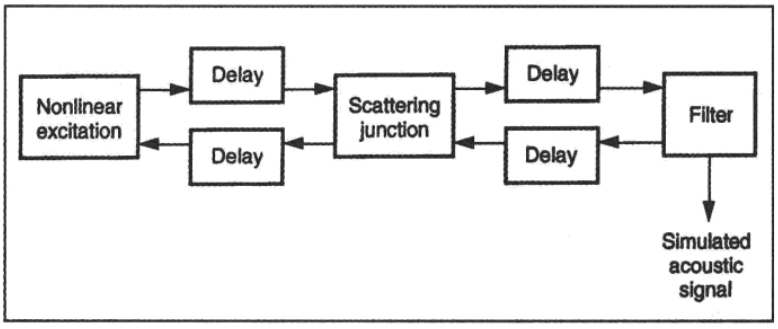


图 7.6 波导乐器的一般性模型，能够仿真弦乐器或管乐器(出自 Cook 1992)。非线性激励注入上方的延迟线，直到它碰到散射结，其将仿真声学系统中，接合点的能量损失及逸散。某些能量会回到振荡器结点，某些能量送到以滤波器仿真的输出结点。
Nonlinear excitation=非线性激励 Delay=延迟 scattering junction=散射结 Filter=滤波器
Simulate acoustic signal=仿真音频信号

为了要接近于非圆柱状管道，如号角或声道，管道会被切成等长部分，每个部分代表一波导滤波器。此称为空间中的取样(sampling in space)，与时间中的取样(sampling in time)相当，因为波阵面通过空间的某段距离需要一段时间。在相邻波导间的散射结得自该点管道的物理尺寸。

图 7.7 说明一个平滑的声音管道被切成一连串小部分，每个部分由波导仿真。类似的近似方法可以用在二维平面或三维空间内(对于混响的仿真)(Smith 1991b, Cook 1992)。

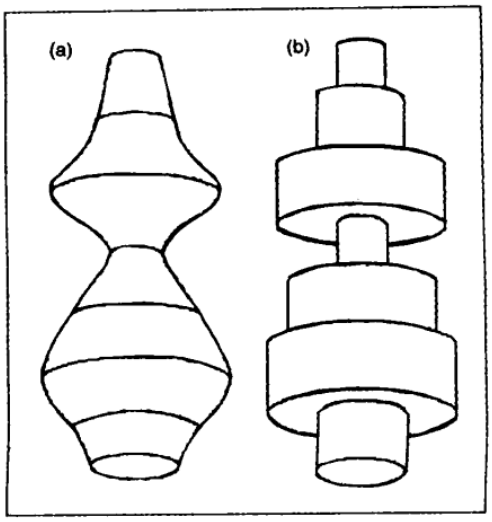


图 7.7 对非圆柱状管道的波导近似。(a)平滑的声音管道,如异域色彩的号角,或是声道的某部分。(b)将管道分成小部分,相当于在空间上采样。

当模拟铜管与木管乐器时,波导仿真乐器管道的每个部分。作为激励的簧片或吹嘴,是由简单的查表振荡器或更复杂的非线性振荡器仿真来驱动波导网络。非线性振荡器是由质量一弹簧一阻尼机制仿真,如之前所述。同样的架构(非线性振荡器驱动波导网络),也可以应用在弦合成上,其非线性振荡器仿真琴弓与琴弦间的互动(Chafe 1985)。

通过利用散射结将不同的波导结合,再在特定点上加上滤波器,并插入非线性接合点,激活波导网络,研究者已将全部类别的乐器建立了模型。下面两节将介绍一些特定的波导乐器模型的例子。

波导单簧管(Waveguide Clarinet)

图 7.8 描述了基于 Hirschman、Cook、Smith(1991)以及 Hirschman(1991)所研究出的单簧管的波导模型。单簧管模型有五个部分:

1. 簧片
2. 上半管身
3. 音高指孔
4. 下半管身
5. 钟形开口

此处只需要仿真单一指孔,因为上半管身与下半管身的尺寸,会依照演奏的音高而改变。这种模型可产生数种有逼真的、类似单簧管的音色,包括依照输入强度而产生的泛音,以及在给予适当的输入时乐器所产生的尖叫声。

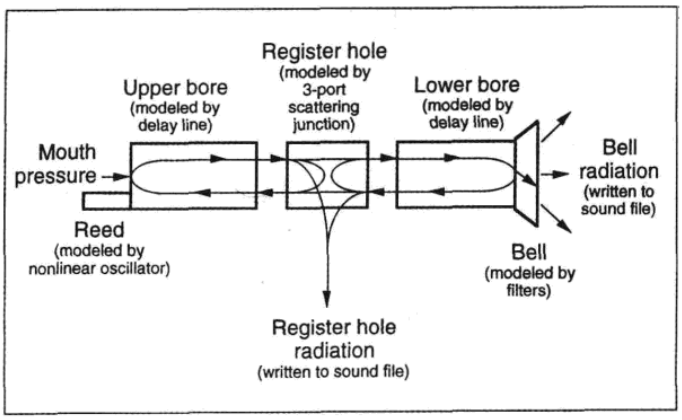


图 7.8　使用波导技术将单簧管作为五部分结构加以仿真。只需要仿真一个指孔，因为上半管身与下半管身的大小，会因为演奏的音高而改变。
Mouth pressure=嘴的压力
Reed (modeled by nonlinear oscillator)=簧片(由非线性振荡器模仿)
Upper bore (modeled by delay line)=管身上部(由延迟线模仿)
Register hole (modeled by 3-port scattering junction)=音域孔(由三个端口的散射结模仿)
Register hole radiation (written to sound file)=音域孔辐射(写进声音文件)
Lower bore (modeled by delay line)=管身下部(由延迟线模仿)
Bell (modeled by filters)=钟形开口(由滤波器模仿)
Bell radiation (written to sound file)=钟形开口辐射(写进声音文件)

波导圆号(Waveguide Horn)

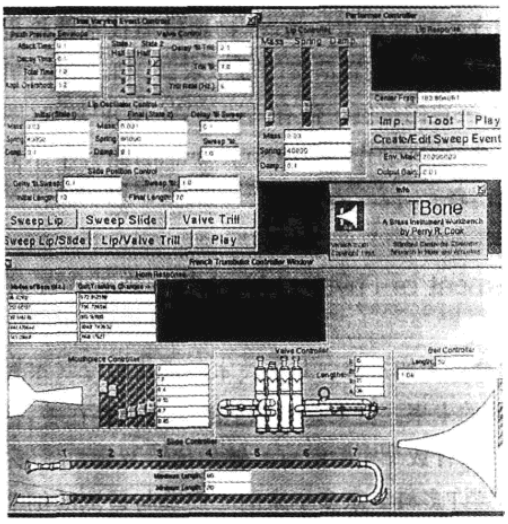


图 7.9　Tbone 铜管乐器工作台，详见本文。

图 7.9 显示由图形界面控制的波导铜管乐器仿真程序 TBone 的画面(Cook 1991b)。画面分成三个窗口,法国 Trumbuba 控制器、演奏者控制器以及时变事件控制器。

位于下方的窗口 French Trumbuba Controller(法国 Trumbuba 控制器)提供修改乐器的图形方式控制器。用滑杆可控制长号滑管的位置、喇叭口的外环以及吹嘴的个别部分。文字输入部分可让使用者指定喇叭口、滑管以及与四个活门相连管道的长度。按下活门按钮会使它们在开与关之间切换,使得在声学线路中加入或移除适当的管路部分。频谱显示在目前圆号的结构中,其脉冲响应经傅里叶转换的强度。这通常会称作传递函数(transfer function),描述每个频率部分经过整个圆号系统后的增益。

演奏者控制器(Performer Controller)窗口位于上右方,可改变演奏者舌头的模型。对质量、弹簧系数以及阻尼的简单控制即已足够指定舌头振荡器的自然频率。舌头的传递函数将显示于频谱中。当"Toot"键按下时,会合成并播放一个短的音。按下 Play 键会再次播放同一个声音文件。

物理仿真合成的输入装置(Input Devices for Physical Modeling Synthesis)

视觉界面提供了物理模型乐器的良好视觉显示,但是因为要实时操控许多参数,用鼠标与数字键盘来实际演奏乐器十分困难。有些研究是将参数群组化,但是实际演出时的理想控制器是拥有数种自由度的音乐性输入装置。当模型可以实时合成时,如许多波导模型的情况,物理模型技术就接近了一个完整的循环:由实际的乐器到用输入装置演奏的虚拟乐器。第 14 章的图 14.5 显示两个物理模型合成输入装置的范例。带有马达的 ACROE 键盘(图 14.5),可仿真机械键盘的许多种实际物理动作。HIRN(图 14.5i)结合了不同形态的控制器(吹嘴控制器、按钮、滑杆、不同吹奏角度)。

模态合成的评估(Assessment of Physical Modeling Synthesis)

近年来,物理模型合成大幅进步。某些技术的进步程度甚至预言物理模型工具套件将是数字合成的未来。但某些观念上的问题仍存在着,仍有许多领域的声音模型尚未建立。只有数个实验室有足够的仪器及能力能够执行此任务。声学研究的历程中,充满许多科学家多年来耐心实验与量测的细节。世界上有上千种不同形态的原音乐器,但是只建立了其中寥寥数种的物理模型。一旦模型建立后,则要决定如何恰当地设定每个声音的几十个参数。

物理模型合成的基本问题之一,是一个乐器本身并无法构成一个声音构造

的完整系统。演奏物理模型乐器的初期尝试,有时候听起来像是初学者的痛苦练习过程。对于我们创造的每个乐器,仍然需要努力练习,才能够掌握得当。若纯粹由软件控制乐器(而不是输入装置),我们必须也如定义乐器般同时定义演奏者的物理模型。这个演奏者模型应该能够实现演奏者个人特色的动作和优秀的演奏技巧——无论对该乐器是以何种方式定义的。已有人对演奏者模型做过初步的尝试,但仍有许多工作有待完成(Garton 1992)。

若有了传统乐器的模型后,使用对实际演出求取参数的分析系统,将有助于建立演奏者模型。下一节将对物理模型合成分析阶段的一些初步尝试做一个概览。

物理模型的来源与参数分析(Source and Parameter Analysis for Physical Modeling)

所有的声音分析都可视为一种参数估计(parameter estimaiton)的形式。也就是说,分析试着以参数设定的方式,来说明输入声音的特性,这些参数设定用来接近一个指定的再合成方法中的声音(Tenney 1965, Justice 1979, Mian and Tisato 1984)。

有了现有乐器的物理模型之后,通常决定适当的演出参数的方法,是与有经验的演奏者合作,对每个单独的音、过渡以及姿态等进行一系列非常费工的试验。这个详细的工作可因加入分析阶段而加快,分析阶段能够聆听大师演奏,并且自动估计其特征参数。

另外一个在物理仿真中加入分析阶段的动机是自动乐器建构(automatic instrument construction)。现有的物理模型仅是声音宇宙的一个小角落。那么对于无法用现有模型实现的声音该怎么办呢?我们可想象有个自动编译器,对任何输入的声音,即便是合成声音,建立一个虚拟乐器。这个自动建构物理模型,将使乐手能以动作控制该声音及类似的声音家族。这种观念可能看起来不太自然,但请记得傅里叶分析其实就相当于某种编译器,对任何输入的声音都能以加法合成乐器实现。

参数估计实验(Parameter Estimation Experiments)

物理模型合成参数估计的初步实验,说明了此方向的难处及其潜力(Szilas and Cadoz 1993)。此处我们将报告三种设计。

来源分离(Source Separation)

沃德(Wold 1987)的文章是一篇以物理模型方式为基础,对再合成非常重要的参数估计研究论文。他的终极目标并不是合成本身,而是多声声源的分离。也就是说,把两个不同乐器发出的混合信号输入系统之中,该系统就会参照物理模型合成的方式,而不是如加法合成等方式,来预测每个乐器各自的再合成参数。

他从设计原声乐器,如人声音色、木琴与单簧管的近似物理模型开始。这些乐器的形式是一组参数化的状态方程。对于一个输入声音,目的是将输入声音与状态方程模型作比较,并试着确认出一组参数集合,以产生相同声音。

图 7.10 说明了沃德参数估计系统的图表。系统的第一部分阐释了评估者都会遇到的问题,即在哪里开始进行预测。他的系统使用频谱分析及音高侦测来做出"第一轮"估计。根据此初步估计,系统会用递归技术改善其分析,并检查其结果,与再合成的状态方程式模型相比较。使用卡尔曼(Kalman)滤波器法改善初次估计。时变的卡尔曼滤波器提供了一种基于噪音观察,对错误取样信号的估计,而类似均方根逼近法的技术。它拥有基于统计性标准的重要特质。(Kalman 滤波器理论是非常先进的研究,详见 Rabiner et al. 1972。)

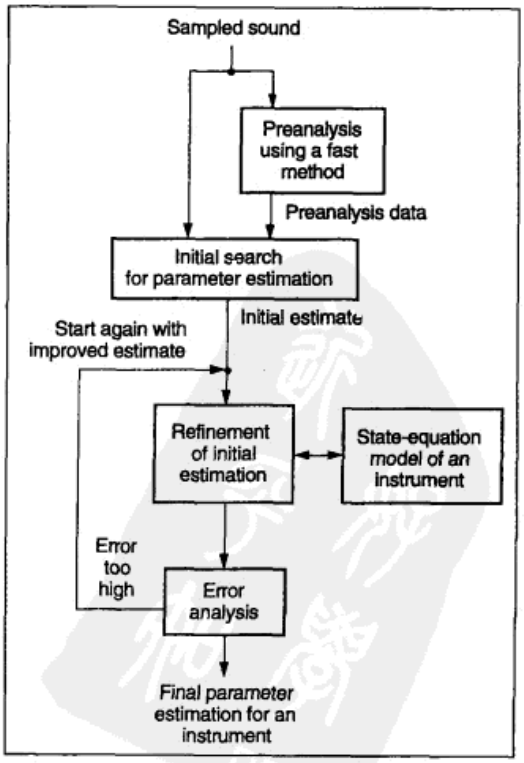


图 7.10 由 Wold(1987)实现的参数估计声音分析。目的是以将两个混合信号分离的方式,来对以物理模型合成为基础的合成器进行参数估计。如果某一个估计距状态方程模型偏差太大,此系统就会尝试用另一种估计。

Sampled sound=采样声音
Preanalysis using a fast method=用快速方法所做的预分析
Preanalysis data=预分析数据
Initial search for parameter estimation=为参数分析而做的初始搜索
Initial estimate=初始估计
Start again with improved estimate=用改进的估计重新开始
Refinement of initial estimate=初始估计的提炼
State-equation model of an instrument=乐器的状态方程模型
Error too high=错误过大
Error analysis=错误分析
Final parameter estimation for an instrument=对乐器的最终参数估计

物理模型合成参数的卡尔曼滤波器估计程序的计算量非常大。对打击乐、人声与单簧管模型的高保真参数预测，每秒钟的分析声音需要几十亿的浮点运算(Wold 1987)。意义重大的是，为了要实时实现此运算法，沃德以对计算机结构新形态的讨论作为其论文的结束部分。

库克(P. Cook)的“歌手”程序(Singer)是人声道的波导物理模型(Cook 1991a, 1993)。此物理模型法与其他人声合成法，如线性预测编码(第 5 章)或本章后半所讨论的共振峰法的差别在于，“歌手”程序包含舌头、声道、鼻道的模型，得以更真实地捕捉发声细节。

从图 7.11 中的线路，就能知道此合成模型的复杂度。对于每个声音都需仔细调整数十个参数。此模型的问题在于：要从何处得到适当的资料，以做出逼真的说话与歌唱声音？库克根据“歌手”程序模型对语音资料进行参数估计，以使模型参数与语音信号相匹配。

此研究的另外一个重要方向，是仿真声门波形(glottal waveform)的成果——由声带发出的语音激励信号。库克使用去卷积法(deconvolution)得出声门波形，并用梳状滤波器法估计音高。声道噪音则用流体动力方法仿真。(见 Blake 1986 对声音与振动的流体动力模型。)D. Matignon 也延续了卡尔曼(Kalman)滤波分析法，由状态方程模型开始，并采用波导再合成模型(Matignon 1991; Matignon, DePalle, and Rodet 1992)。

高阶频谱分析(Higher-order Spectrum Analysis)

这里顺便要提到另外一个策略，是称为高阶频谱分析(Higher-order spectrum, HOS analysis)的一组新技术。HOS 法是非常先进的研究题目。HOS 分析的目的是描述非线性系统。HOS 分析的好处是，它显示出成分间的关系。这对于非线性系统尤其重要，因为他们往往有交互调变效果。HOS 可以显示出一个成分是由其他成分经非线性程序而得来的。如我们所见，许多声音是由非线性激励开始的，故 HOS 法显示出其作为对这种来源进行分析的有效工具(Wold 1992, Nikias and Raghuveer 1987)。

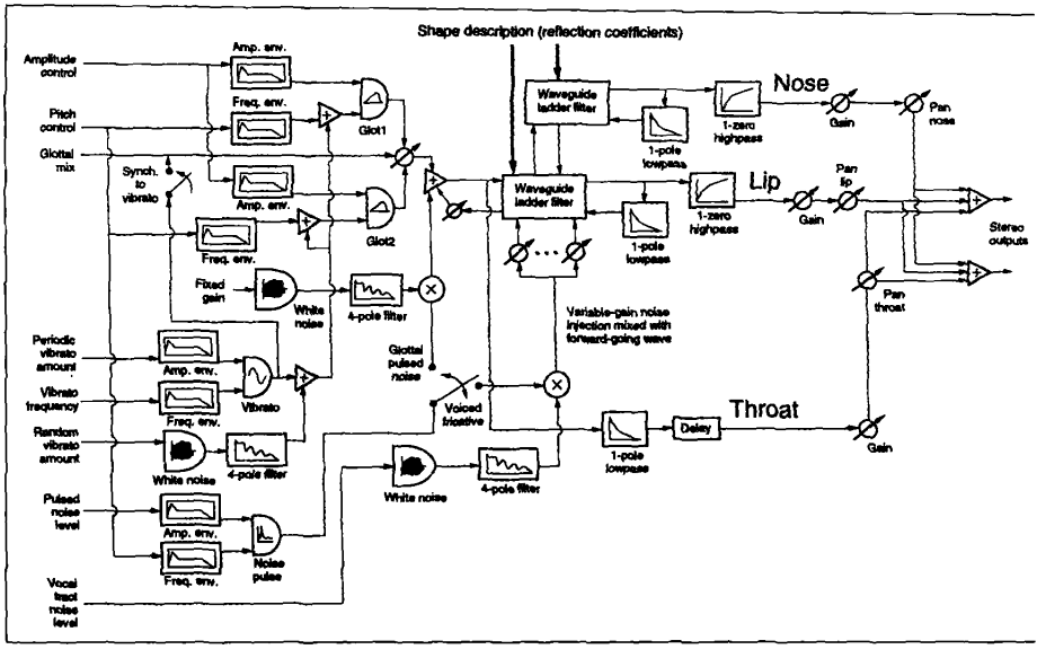


图 7.11 “歌手”程序的方块图，为人声的物理模型合成器。图的左端描述激励来源，中间部分描述波导共振器，右半部分描述输出阶段。两个声门波表振荡器(Glot 1，Glot2)允许积分信号中缓慢的揉音同步变化。声门噪声源包括经过滤波的白噪音，乘以任意指定的、与声门振荡器同步的时域波形。此模型允许脉冲噪音与周期性声源混合。弦波振荡器仿真揉音，揉音的频率以噪音随机化。经过滤波的白噪音注入向前移动的声门波。声门反射以简单的反射系数仿真，并以低通滤波器仿真唇及鼻孔的影响。低通滤波器与延迟线仿真由喉咙输出信道经皮肤所发散的声音。

Shape description (reflection coefficients)=形状描述(反射系数)　Amplitude control=振幅控制
Pitch control=音高控制　Glottal mix=声门混合　Periodic vibrato amount=周期揉音量
Vibrato frequency=揉音频率　*Random vibrato amount*=随机揉音量　*Pulsed noise level*=脉冲噪水平
Vocal tract noise level=声带揉音水平　Synch. to vibrato=同步到揉音　Amp. env.=振幅包络
Rreq. env.=频率包络　Fixed gain=固定增益　White noise=白噪音　Vibrato=揉音
4-pole filter=4 极滤波器　Glottal pulsed noise=声门脉冲噪音　Glot =声门波形表振荡器
Voiced fricative=发声的摩擦　Waveguide ladder filter=波导梯状滤波器
Variable-gain noise injection mixed with forward-going wave=变量增益噪音注入与前进波形相混合
1-pole lowpas=1 极低通 (s)　Delay=延迟　1-zero highpass =1-0 高通　Gain=增益
Nose=鼻子　*Lip*=嘴唇　*Throat*=喉　*Pan nose*=鼻音声象　*Pan lip*=唇音声象
Pan throat=喉音声象　Stereo outputs=立体声输出

卡普拉斯一斯特朗(拨弦与鼓)合成〔Karplus-Strong(Plucked String and Drum Synthesis)〕

拨弦与鼓合成的卡普拉斯一斯特朗(Karplus-Strong)(KS)算法,是一种根据延迟线或再循环波表(recirculating wavetable)原则的高效率技术(Karplus and Strong 1983; Jaffe and Smith 1983)。在它的实现上,KS与MSW及前述的波导技术相关。基本KS合成法的计算需求极少。所以不难想象这种技术已从较慢的8-bit微处理器,到大型数字合成器,以及称作Digitar芯片的集成电路(Karplus and Strong 1983)等许多地方都在应用。

拨弦声音(Plucked Strings)

基本KS算法由填入了随机数值,长度为p的波形表开始。当由波形表右方开始读取数值时(图7.12),数值会以某种方式被改变,而其结果会重新安插至波形表的左边。最简单的改变方式,是将现在的取样点与前一个取样点平均——也就是简单低通滤波器的核心活动(见第10章,对于平均低通滤波器的解释)。波形表的读取指针及写入指针以每个采样时间为间距渐增。当指针抵达波形表底端,将折回起点并重新开始。这个简单运算的结果是产生一个有明确音高的声音,最开始很"明亮",但当它衰减时,音色迅速转暗,成为单一正弦波——很像拨弦时的声音。

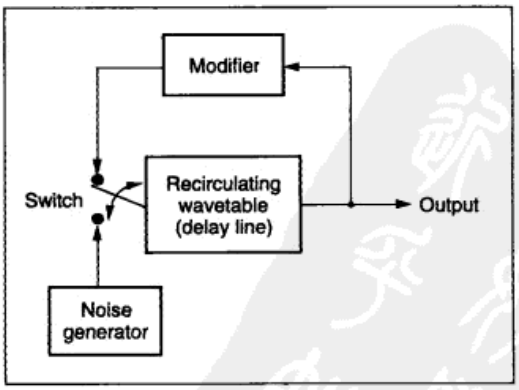


图7.12 卡普拉斯一斯特朗(Karplus-Strong)再循环波形表的核心部分。再循环波形表的输入开关,在每个事件一开始时连接到噪声源,接着在事件的其他时候切换到修改器(modifier)循环。修改器对连续采样进行平均,模仿阻尼效果。
Modifier=修改器 Switch=开关 Recirculating wavetable (delay line)=再循环波形表(延迟线)
Noise generator=噪音发生器 Output=输出

如果波形表一开始填满了随机数值，那么读者可能会怀疑，为什么听起来不会是噪音——至少在声音的一开始部分。声音带有音高的原因，是因为在每次经过波形表时，波形表一再重复(有少量改变)。因为这些每秒发生数百次重复，一开始听起来随机的波形，便成了类周期性的波形了。如果没有算法的衰减部分(低通滤波器)，那么波形相当于二分之一取样率的和谐成分(理论上)，带有类似簧片一风琴的音色(Karplus and Strong 1983)。

在实践中，对每个音重新填入带有一组新的随机数值的波形表是个很好的主意。这会给每个音提供一个稍稍不同的泛音结构。伪随机数值发生器(pseudorandom number generator)程序(如 feed back shift－register random bit generator；Knuth 1981，p. 29)能够提供随机数值。

类似鼓的音色(Drumlike Timbres)

KS 利用稍微复杂的改变器回授采样，产生类似鼓的音色。此音色是由设定几率参数 b 控制，称为混合系数(blend factor)，即 $0 \leqslant b \leqslant 1$。改变器算法如下：

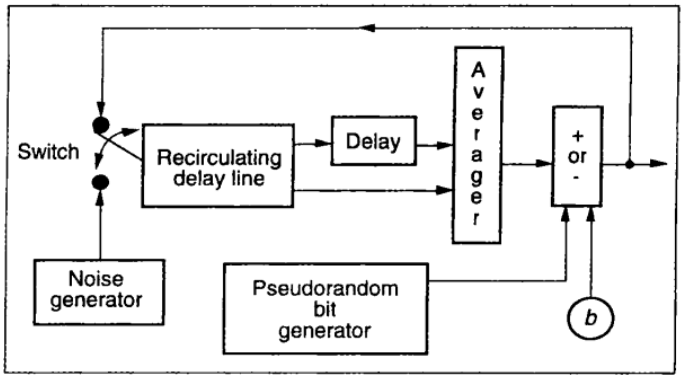


图 7.13 Karplus-Strong 鼓声合成算法，数值 b 是混合系数。
Switch=开关 Noise generator=噪音发生器 Recirculating delay line=再循环延迟线
Delay=延迟 Averager=平均器 or=或 Pseudorandom bit generator=类随机比特发生器

$$Signal_t = \begin{cases} +1/2(Signal_{t-p} + Signal_{t-\lceil p-1 \rceil}), \text{几率为 } b \\ -1/2(Signal_{t-p} + Signal_{t-\lceil p-1 \rceil}), \text{几率为 } 1-b \end{cases}$$

t 为现在的采样索引值，而 p 为波形表的长度。

当 b 为 1 时，修改器就跟以前一样，是个低通滤波器，而声音会像是拨弦。当 b 为 0.5 时，声音不再像是弦了，它失去了它的音高，而听起来比较像鼓。当 b 为 0 时，每 $p+0.5$ 个取样，信号会正负号相反。这将使得感知频率减为一半，而只留下频谱中的奇数谐波，造成类似竖琴的低音区的声音。

图 7.13 说明鼓声合成的 KS 乐器。注意由再循环波形表而来的采样会与之前的采样平均，并依照混合参数 b 的几率给予正号及负号。当 b 接近于 0.5 时，波形表长度不再控制音高，因为波形不再是周期性的了。在鼓声的开始处，长度 p 决定噪音爆破声的衰减时间。当 p 相当大时（超过 200），乐器的声音会像是嘈杂的小鼓。当 p 很小（小于 25），相当于刷击嗵嗵鼓（brushed tom-tom）。要做出共振鼓，波形表须预先加载定值，而非随机数值。

延伸衰减时间（Stretching Out the Decay Time）

由 KS 产生的声音衰减时间与波形表长度 p 成比例，也就是说，使用较小的波形表的音符将很快衰减。理想情况下，我们希望将延迟时间与波形表长度独立开来，这可由称为衰减延伸（decay stretching）的技术达到，此算法如下：

$$Signal=\begin{cases}+Signal_{t-p}，几率为\ 1-(1/s)\\ -1/2(Signal_{t-p}+Signal_{t-[p-1]}，几率为\ 1/s\end{cases}$$

s 是延伸系数（stretch factor），当 s 设为 1 时，会使用一般的算法，不会延伸衰减时间；当 s 接近于 0 时，声音不会被平均，所以会在整段延迟时间中延伸。

KS 的扩展应用（Extensions to KS）

卡普拉斯与斯特朗的同事，贾菲与史密斯（Jaffe and Smith）将许多 KS 技术加以扩展（Jaffe and Smith 1983）。在基本 KS 线路上加上滤波器，他们可以得到以下的效果：

- 消除最初的“拨动”声音
- 与带宽相关地改变声音音量大小
- 滑奏与连音
- 模仿共鸣弦振动效果
- 仿真拨片与琴桥间相对移动的声音
- 仿真上下拨奏

这些技术很专门，其描述在 Roads（1989）中重印。读者可参考这些说明。另外的扩展应用是仿真电吉他的声音，特别重点是带有失真与回授，经过高增益前置放大线路后的电吉他音色。详见 Sullivan（1990）。Karjalainen et al.（1991）将 KS 模型应用在类似笛子的声音上。

共振峰合成(Formant Synthesis)

共振峰(formant)是频谱上的能量峰(图 7.14),可以含有泛音、非泛音的分音以及噪音成分。共振峰是人说话的元音与许多乐器声音的特征。

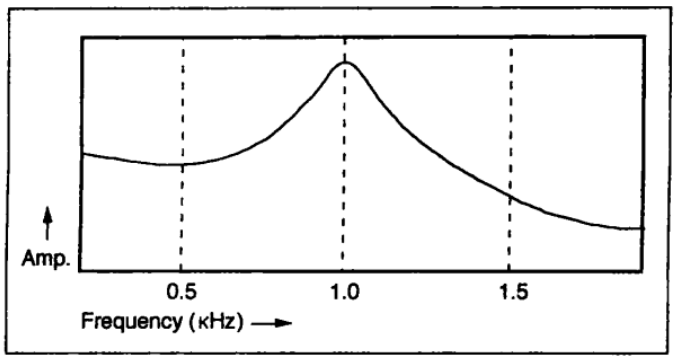


图 7.14 共振峰在频谱上会呈现峰状。此共振峰集中在 1kHz。

如图 7.15 所示,在 0 到 5 000Hz 的范围内,声道通常有五个共振峰区域(包含基频)。见 Bennett and Rodet (1989),有女高音、女低音、假音男高音、男高音以及男中音的不同音素的共振峰图。

共振峰区域可当做某种"频谱记号"或是许多声源的音色特征。(见 Grey 1975 and Slawson 1985 的简介,以及对音色研究的参考资料。)但这并不是说声音或乐器的共振峰是固定的,它们是根据基础音的频率而产生相对变化的(Luce 1963; Bennett and Rodet 1989)。许多情况下,共振峰是耳朵判定声源种类的唯一线索。

了解人声的共振峰特质,长久以来都是科学研究的目标。历年来,不断有精巧的合成方法出现,意图合成类似元音的共振峰,包含"歌唱的火焰(singing flames)"、"歌唱的水注(singing water jet)"以及设计用来仿真狗吠及人声的共振峰机械装置(Tyndall 1875)。巴黎的 Marage 博士用实体物理模型的方法建立了人声仿真器。每个元音都由连接着人工口腔的一对橡胶唇所发声。语音气流是由一对电子机械肺所供给:由电子马达驱动的肺(Miller 1916)。其他实验装置使用了风琴管的特殊结合,产生类似元音的声音。

理所当然地,语音研究成果被当成在音乐上共振峰合成的观念来源。本节的其余部分讨论三个产生共振峰的合成技巧:共振峰波形函数(formant wave-function)或 FOF 合成、VOISM 以及窗口函数(window-function WF)合成。FOF 与 VOSIM 是直接由仿真人声的尝试变化而来,而 WF 是为了仿真传统乐

器的共振峰所发展。

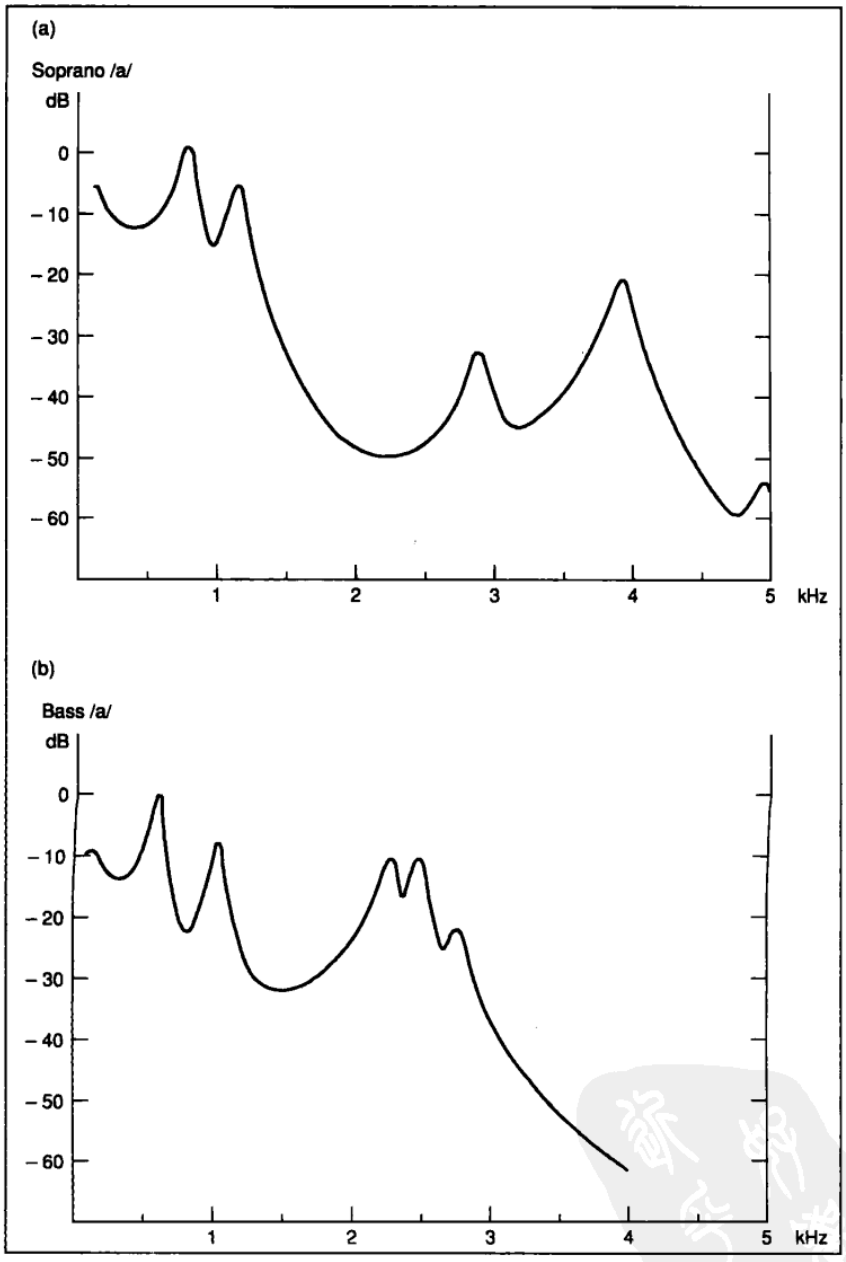


图 7.15　0 到 5kHz 间的人声共振峰区域。(a)女高音唱元音[a];(b)男中音唱元音[a]。(出自 Bennett and Rodet 1989。)

可以确定的是,除本章中所介绍的技术之外,许多技术都可以产生共振峰。这些技术有加法合成(第 4 章)、减法合成(第 5 章)、颗粒合成(第 5 章)、频率调制(第 6 章)以及物理模型(第 7 章),我们仅只提过其中一些。我们将 FOF、

VOSIM 与 WF 区分开来的理由有二：第一，因为他们并不属于任何之前所论及的合成领域；第二，是因为他们主要就是设计用来合成共振峰的。

共振峰波形函数合成与 CHANT 程序 (Formant Wave-Function Synthesis and CHANT)

共振峰波形函数合成(formant wave-function synthesis，由法文原文 fonction d'onde formantique 的缩写是 FOF)，是 CHANT 合成系统的基础。(法文中 chant 代表歌、歌唱。)自从它发表后的数十年来(Rodet and Santamarina 1975；Rodet and Delatre1979；Rodet and Bennett 1980；Bennett 1981；Rodet，Potard and Barrière 1984)，CHANT 已被移植到许多平台，从大型合成器如 4X (Asta et al. 1980)到个人计算机上(Lemouton 1993)。FOF 产生器也被实现于 Csound 合成语言中(Clarke 1990)。

CHANT 是设计用来仿真广泛的自然机制，它们被激励时会共振，但最终会因为物理力量，如摩擦力而衰减。比方说，钟的共振时间很长，而木块则是种阻尼共振，几乎会使共振立刻停止。我们可以用指头敲打脸颊来激发一个共振，这单一的动力就会引起短促的爆破声。声带产生一连串快速的脉冲，连续激发声道的共振，形成带有音高的声音。这些系统都属于 FOF 所操纵的种类。

CHANT 内含的基本发声模型是人声。然而，使用者可以调整 CHANT 的许多参数，使之合成人声以外的声音——如乐器或合成效果的仿真。夏维尔·罗代(Xavier Rodet)和他的同事们使用 CHANT 研发出男性与女性歌手的模型，传统弦乐器、木管乐器、号类乐器以及打击乐器等。如我们将见到的，CHANT 也可以用来当作采样声音的滤波器组处理器，这种模式为某些作曲家所偏好。

FOF 合成的基本原理(Fundamentals of FOF Synthesis)

FOF 是 CHANT 系统的核心，它与基于传统减式方法的合成，如线性预测(见第 5 章)不同。传统减式方式中，来源信号带有宽频谱，如连续脉冲波或噪音信号——经过复杂的滤波器。滤波器切开(carve)几乎所有的频率，留下几个共振频率，即频谱上的共振峰。

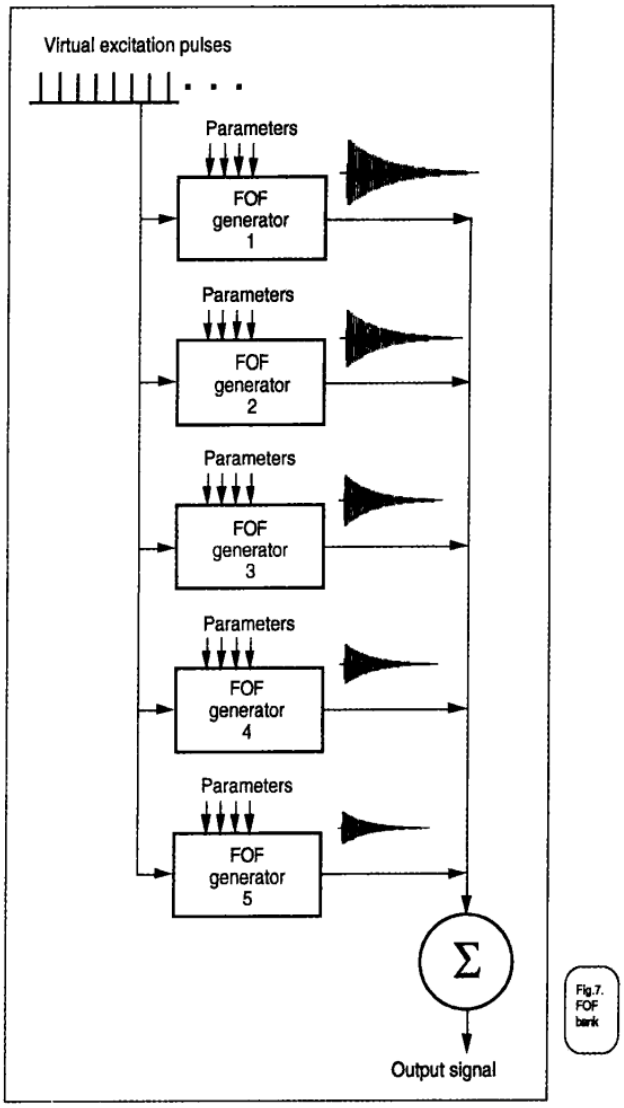


图 7.16　一组 FOF 发生器，在每个音高时长内，由输入的脉波驱动 FOF“颗粒”。所有 FOF 发生器的输出被相加以产生一个复合的输出信号。
Virtual excitation pulses=虚拟激励脉冲　　Parameters=参数
FOF generator=FOF 发生器　　Output signal=输出信号

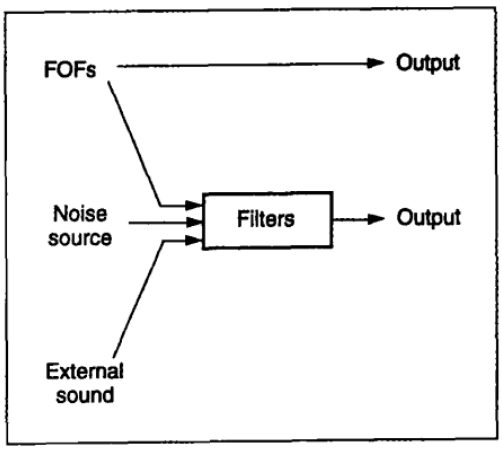


图 7.17 FOF 合成与处理状态。输出可以是弦波，经滤波的噪音，经滤波的取样声音，或者上述的结合。
Noise source=噪声声源 External sound=外部声音 Filters=滤波器 Output=输出

罗代将减式合成中所使用的复杂滤波器，拆解为一组由脉冲波激发的数个相同平行带通滤波器。(滤波器是二阶部分，如第 10 章所述。)一个 FOF 实现了这些平行带通滤波器之一，平行的数个 FOF 就能仿真带有数个共振峰的复杂频谱包络。频谱包络是一个平滑的轮廓线，可以描出频谱中的峰类(Depalle 1991)，类似于线性预测编码分析产生的曲线。

然而 FOF 有双重特性，FOF 实现方式的一种变化，是用一组经阻尼的正弦波发生器替代滤波器。这些发生器的信号与频谱等同于由脉冲波驱动的滤波器(图 7.16)。由罗代所述，将滤波器改为正弦波发生器有数个优点。正弦波发生器较为有效率，而且比起滤波器来，对计算精准度的需求也不高。而且，一个以上的共振峰可以连续地改变为正弦波，其强度与频率都可控制，做出共振峰合成与加法合成间的连续转换(Rodet 1986)。

滤波器与经阻尼的正弦波发生器方法，可以结合，而产生单一声音，如图 7.17 所示。

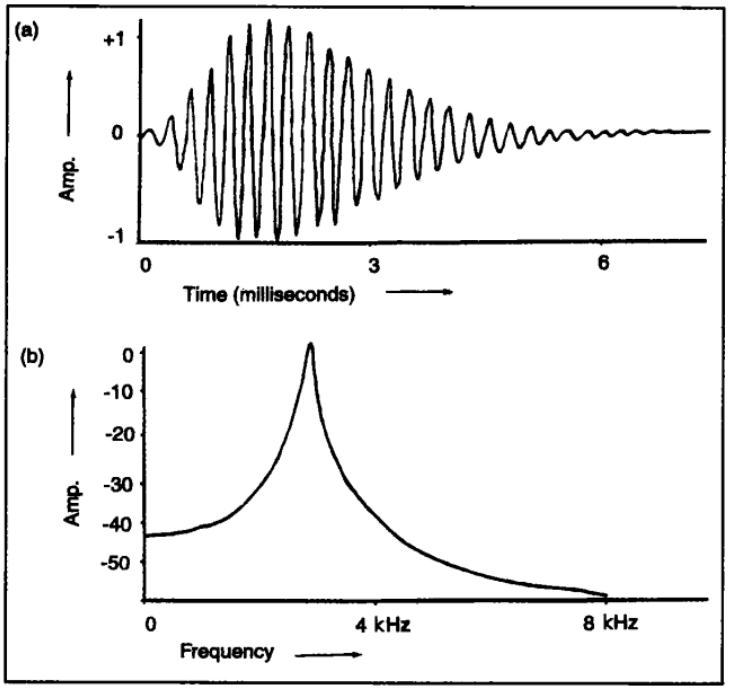


图 7.18　FOF 颗粒与频谱。(a)单一颗粒或是由 FOF 产生器发出的爆音;(b)在(a)中的频谱,以对数强度比例绘出。(出自 d'Allessandro and Rodet 1989)
Frequency=频率　Amp.=振幅　Time(milliseconds)=时间(毫秒)

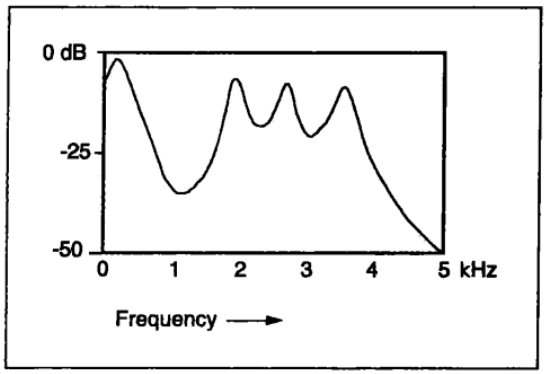


图 7.19　人声共振峰频谱,由数个平行的 FOF 产生器所产生。

剖析 FOF(Anatomy of a FOF)

在合成中,FOF 产生器在每个音高周期间产生一个声音颗粒。所以一个音符内含有数个颗粒。为了要将这些颗粒与第 5 章所提的区别开来,我们称这些叫做 FOF 颗粒。FOF 颗粒是经过阻尼的正弦波,可能有较陡或较缓的起音及类指数衰减(图 7.18a)。FOF 颗粒的包络称作局部包络(local envelope),而

不同于整个音符的整体包络。

在形式上，局部包络用下列定义：对于 0≤t≤tex：

$$env_t = 1/2[1-\cos(\pi_t/tex)] \times \exp(-atten_t)$$

对 $t \geqslant tex$

$$env_t = \exp(-atten_t)$$

这里 π 为 FOF 信号的初始相位，*tex* 是局部包络的起音时间，*atten* 则是衰减时间(D'Allessandro and Rodet 1989)。

因为每个 FOF 颗粒的时长仅有几毫秒，FOF 颗粒的包络会对正弦波周围加入可听见的边带成分，而造成共振峰。〔这是由于正弦波与包络的卷积(convolution)所造成，见第 10 章对于卷积的解释〕。经阻尼的正弦波发生器相当于一个带通滤波器的频率响应曲线(图 7.18b)。

将数个 FOF 产生器的结果相加，就能得到带有数个共振峰的频谱(图 7.19)。

FOF 参数(FOF Parameters)

每个 FOF 产生器是由数个参数所控制，包含基频与振幅。图 7.20 说明四个共振峰参数，我们称为 *p1* 到 *p4*：

*p*1 为共振峰的中心频率。

*p*2 为共振峰带宽，由共振峰中心频率向外到−6dB 间的宽度。

*p*3 为共振峰的最高振幅。

*p*4 为共振峰边缘(formant skirt)的宽度。共振峰边缘是共振峰的较低部分，大约共振峰下−40dB，相当于山周围的小丘。边缘参数与指定共振峰带宽的参数相互独立。

这种内在的时域与频域间相互关系，以指定 FOF 参数的形式得到证明。两个主要的共振峰(频域)参数是在时域上指定，即 FOF 颗粒包络的性质，不过这种方法对不熟悉信号处理理论的音乐家来说是违背直觉的。首先，FOF 的起音长度控制参数 *p*4，即共振峰边缘的宽度(约−40dB)。也就是说，当起音的时长增加，边缘宽度就变窄。图 7.21 说明此关系。

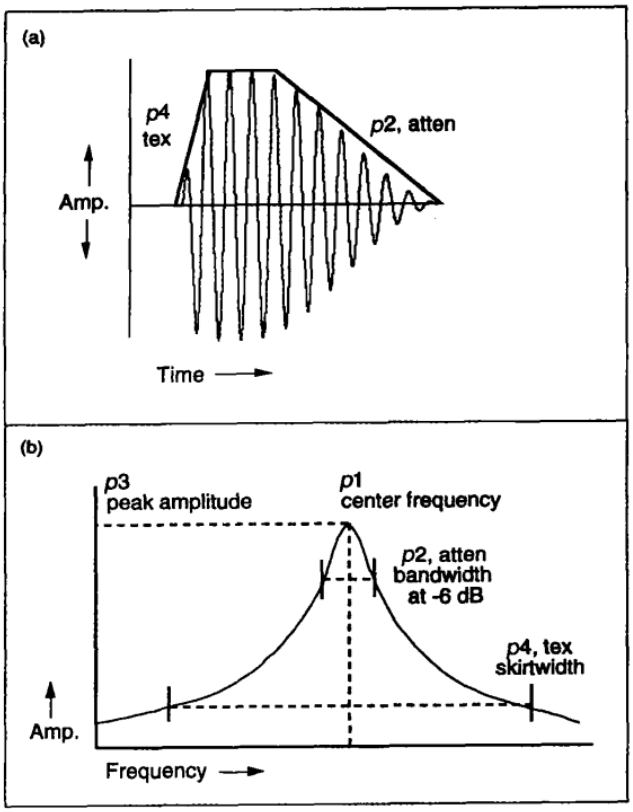


图 7.20　FOF 参数。(a)FOF 的时域图。参数 *p4* 表示起音时间(在大部分的实现中称作 *tex*),而 *p2* 表示衰减(称作 *atten*)。(b)四个共振峰参数的频域图。参数 *p1* 是共振峰的中心频率,而 *p2* 是共振峰带宽,*p3* 是共振峰峰值强度,*p4* 则是共振峰边缘宽度。

(a)
Amp.=振幅
Time=时间
p4 tex=参数 p4 起音时间
p2 atten=参数 p2 衰减

(b)
p3 Peak amplitude=参数 p3 振幅峰值
p1 Center frequency=参数 p1 中心频率
p2, atten bandwidth at - 6dB=参数 p2 衰减带宽至 -6 分贝
p4 tex skirtwidth=参数 p4 起音时间边缘宽度
Amp.=振幅
Frequency=频率

其次,FOF 衰减的时长决定 *p2*,也就是共振峰的−6dB 带宽大小。所以较长的衰减时间,会转成较尖锐的共振峰,而较短的则会使信号带宽扩大。(这声音的时长与其带宽间的关系在颗粒合成中已出现过,如第 5 章所详细讨论过的。)

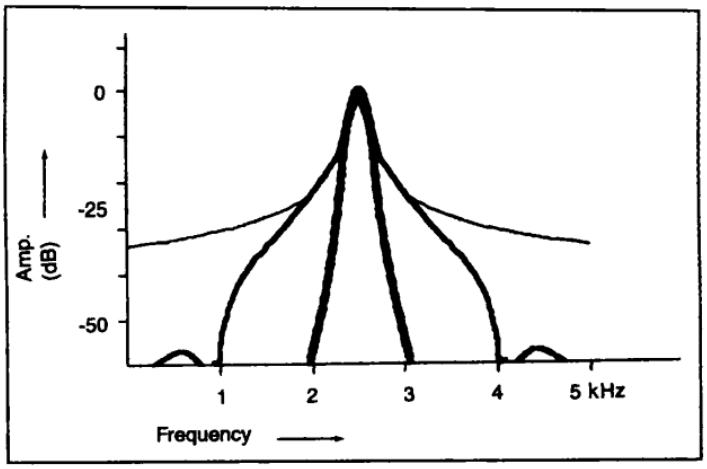


图 7.21 改变起音时间造成共振峰边缘带宽改变。细线，最宽的共振峰：$p4=100\mu s$；中等粗细的线，中等共振峰：$p4=1ms$；粗线，窄共振峰，$p4=10ms$。
Frequency=频率 Amp.=振幅

FOF 合成的典型应用是将 FOF 产生器平行排列配置。在每个 FOF 产生器的六个主要参数外，CHANT 的实现提供了额外的参数，以控制声音整体。表 7.1 列出主要的参数。有些实现方式超过 60 个参数。这众多的参数要求构建一个符合控制合成引擎规则的数据库。这在人声及器乐仿真时特别需要，因需要大量的参数设定。CHANT 及相关的高级语言，如 FORMES(Rodet and Cointe 1984，见第 16 章)以及 PatchWork(Barrière, Iovino, and Laurson 1991)的部分工作，是提供规则数据库。

CHANT 程序(The CHANT Program)

CHANT 合成程序(Baisnée 1985)向使用者提供三种互动模式。第一种是最简单的，使用者提供一组预设歌唱合成的变量值。这些变量会转换成每个 FOF 产生器从 $p1$ 到 $p4$ 的参数，如前所述。变量可以被分成下列几个领域：

- 声音强度
- 基频
- 基频的揉音与随机变化
- 频谱形状与共振峰强度
- 共振波形的局部(local)包络
- 整体振幅曲线

表 7.1　主要的 FOF 参数

对每个 FOF 产生器
振幅
基频
波幅线——替代颗粒的减弱
共振峰中央频率(p1)
自共振峰顶的－6dB 共振峰带宽(p2)
共振峰最高振幅(p3)
共振峰边缘带宽(p4)
颗粒重叠
波形表(通常为正弦波)
初始相位
人声合成的频谱校正
滤波器参数
共振锋中心频率
共振峰振幅
共振峰带宽

- 频谱形状与共振峰振幅
- 共振峰波形的局部包络
- 整体振幅曲线

在第二种模式下，FOF 作为时变滤波器作用于采样声。这一模式已被作曲家作为一种声音转换技术使用。

在交互作用的第三种模式下，由用户写出规则——描述各个音色之间转变与插值的算法。像排秩作业(PatchWork)一类的辅助作曲环境也支持这一策略(Iovino 1993；Barrière，Iovino，and Laurson 1991；Malt 1993)。

FOF 分析/再合成(FOF Analysis/Resynthesis)

借助于共振峰与正弦波，FOF 合成代表一种很有潜力的通用方法。我们在此简单介绍产生 FOF 再合成参数的分析系统的成果。

共振模型(Models of Resonance)

共振模型(Models of Resonance, MOR)指捕捉传统原声乐器音色的方法，再合成使用 FOF(Barrière, Potard, and Baisnée 1985; Potard, Baisnée, and Barrière 1986, 1991)。MOR 所依据的是典型的激励——共振模型。也就是说，发声机制被分成激励阶段与共振阶段。MOR 假定激励是脉冲造成的，如以琴拨弹奏，或以鼓棒敲打。共振则是乐器本体对于激励的声学响应。

在 MOR 中，每个共振是以在特定频率的正弦波仿真的，并带有时间上的对数衰减。(这与窄带通滤波器的脉冲响应相符，在第 10 章讨论)。当脉冲(如钢琴琴槌敲击)激发共振时，每个共振会以其特有的强度与频率发声。由于 MOS 仿真乐器本体，乐器的声音不单是相关于现在所演奏的音符，同时也相关于之前演奏音符所造成的状态。

MOR 分析仅捕捉共振部分。这并不是乐器完整的物理或频谱模型。它也不是被设计用来准确复制输入信号的。其目的是萃取出可用来“调校并控制音色结构”的特质(Barrière, Potard, and Baisnée 1985)。

MOR 的研发者认为，MOR 的分析方法是个既有些费力也不太完美的过程(Potard, Baisnée and Barrière1986; Baisnée 1988, Potard, Baisnée, and Barrière 1991)。基本上，它涉及提取一个声音片段的单一的快速傅里叶转换(FFT)。(FFT 在第 13 章中与附录中解释)。波峰萃取算法将最重要的频谱共振分开来，剔除其他成分。接着进行另一个使用较大时间窗的分析，频谱峰会在一公共文件(common file)中合并。从这些峰值进行的再合成可以用来比较它与原始信号接近的程度 。使用者连续地使用较大的窗口重复分析，直到得到满意的再合成为止。对于复杂声音，分析可以被切成从不同时间开始的数个部分，对每个部分都可以用不断反复的程序来执行。能得到最好效果的是带音高的打击乐声音，如马林巴、钟琴与管钟(Baisnée 1988)。

MOR 再合成最多使用数百个标准 FOF 产生器，可以是指数衰减的正弦波振荡器，或是由脉冲噪音激发的带通滤波器。另外一种实现方式使用了某种特殊硬件，允许经由 MIDI 协议进行实时控制(Wessel et al. 1989)。(MIDI 将在第 21 章中介绍。)

MOR 变形(MOR Transformations)

MOR 的目的之一，是作为自然声音与合成声音间的桥梁。激励与共振部分的分离，提供了分析声音变形实验的肥沃土壤。比方说，可将一般的激励(脉

冲波或白噪声)，更换为采样的乐器声音，做出交叉合成(cross-synthesis)效果。

研究者使用了一组分析模型的数据库和另一组规则数据库，用来将一个 MOR 转换成另一个 MOR。这些规则可以在频率或时间上延展 MOR，或将共振模型相加创造出音色混合。其他规则在一乐器与另一乐器的共振之间，于时间上做内插计算。

当激励是单一脉冲或爆破噪音时，MOR 法极有效率，但是当激励与共振/激励结构间的耦合现象相关时，如弓与小提琴弦的情况，可能就不那么适用了。能够处理耦合现象是前述的物理模型合成的强项。

FOF 与频谱包络匹配(Matching the Spectrum Envelope with FOFs)

D'Allessandro and Rodet(1989)报告了由线性预测编码(linear predictive coding，LPC)频谱分析(见第 5 章)开始的 FOF 分析/再合成实验。在一帧一帧地找出频谱包络的轮廓后，此程序萃取出符合于一组 FOF 产生器的共振峰值。此结果并不是同等的重建(作者举出了最前面第二个与第三个谐波的问题)，而是与原始信号相似。DePalle(1991)大致上采用 FOF 分析/再合成，利用技术逼近于原始分析声音的时变频谱包络。他的许多研究集中在自回归(autoregressive，AR)频谱分析法，如第 5 章与第 13 章所述。

VOSIM

VOSIM 合成技术是由 Utrecht 的 Institute of Sonology 的 Werner Kaegi 和 Stan Tempelaars，在 20 世纪 70 年代早期所发展(Kaegi 1973，1974；Tempelaars 1976；Kaegi and Tempelaars 1978)。其核心观念是产生重复的爆音信号，造成强力共振峰成分。在此意义下，此技术与前述 FOF 相关。如同 FOF，VOSIM 原先是用来仿真元音的。之后，被扩展到仿真摩擦音，如辅音[sh]——以及准乐器声音(Kaegi and Tempelaars 1978)。

VOSIM 波形(VOSIM Waveform)

VOSIM 波形是对人声产生的信号粗略近似所得。此近似表现为连续的脉冲串形式，脉冲串内的每个单一脉冲是正弦波函数的平方。脉冲的最大强度以参数 A 设定。每个脉冲串带有 N $\sin^2$ 个脉冲，以衰减参数 b 来减弱强度(图 7.22)。单一脉冲的宽度(时长)T，决定了共振峰频谱的位置。每个脉冲串之后是可变长度延迟 M，用来决定脉冲串的整体周期，所以有助于决定基频周期。

周期可由$(N\times T)+M$计算出，所以200微秒的七个脉冲加上900微秒的延迟，总时长是3毫秒，基频为333.33Hz。共振峰中心在5 000Hz。

通常在VOSIM信号中，有两个明显的感知对象：一是基频，相对于整个信号的重复频率；另一个则是在频谱上的共振峰，相当于$\sin^2$脉冲波的脉冲宽度。每个VOSIM振荡器会造成一个共振峰。要建立带有数个共振峰的声音，必须将数个VOSIM振荡器的输出加以混合(如FOF产生器)。

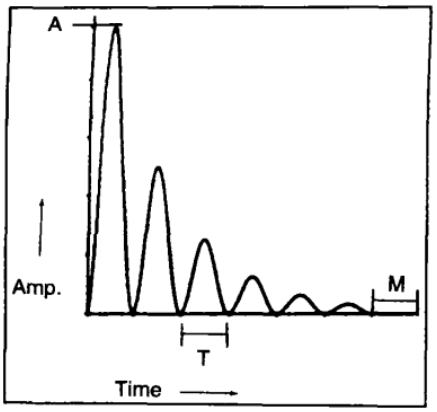


图7.22 VOSIM脉冲串，参数于文本中介绍。
Time=时间 Amp.=振幅

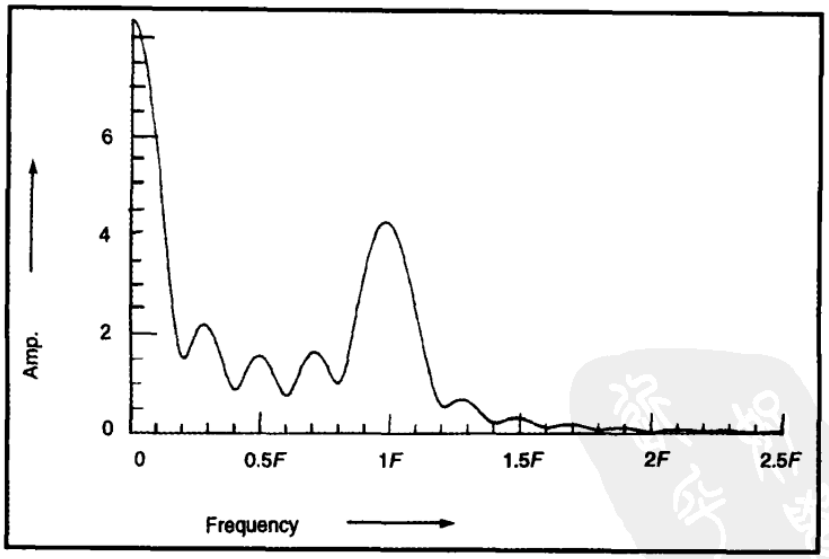


图7.23 有五个脉冲，衰减函数为0.8的VOSIM振荡器所产生的频谱。(出自De Poli 1983。)
Frequency=频率 Amp.=振幅

VOSIM振荡器是通过改变一组影响产生声音的参数来控制的。T、M、N、A与b是最主要的参数。要得到揉音、频率调制以及噪音，我们必须调制延

迟时间 M。由于这种限制，Kaegi 和 Tempelaars 认为必须引入三个新的参数：S、D 以及 NM，分别对应调制的形态（正弦波或随机），最大频率偏移，以及调制率。他们同时也想提供“过渡声音”，所以便引入了变量 NP、δT、δM 及 δA。这些分别是 T、M 与 A 的在 NP 个周期之内的正负渐增值。

改变脉冲宽度 T，可以随时间改变共振峰。此效果称为共振峰偏移（formant shifting），与在频率调制合成中渐进的频谱强化所得到的声音不同（见第 6 章）。

未混合的 VOSIM 信号并非限频信号。这可能会在低取样率时造成系统的混淆（aliasing）问题（见第 1 章）。在两倍共振峰频率以上，频谱成分的强度比起基频至少要降低了 30dB。共振峰频率六倍以上，频率成分的强度至少要降低 60dB（Tempelaars 1976）。

声响技术研究所（Institute of Sonology），在荷兰的乌得勒支的舍尔彭尼斯（J. Scherpenisse）设计并建造数个由小型计算机控制的 VOSIM 振荡器（Tempelaars 1976；Roads 1978a）。在多伦多大学，VOSIM 振荡器则建构于 SSSP 数字合成器的硬件之中（Buxton et al. 1978b）。

表 7.2　VOSIM 参数

名称	描述
T	脉冲宽度
δT	T 的渐增值或渐减值
M	一串脉冲后的延迟时间
δM	M 的渐增值或渐减值
D	M 的最大偏移
A	第一个脉冲的强度
δA	A 的渐增值或渐减值
b	脉冲串的衰减参数
N	每个周期内的脉冲个数
S	调制类型（正弦波或随机）
NM	调制率
NP	周期数

窗函数合成（Window Function Synthesis）

窗函数（window function，WF）合成是使用完全和谐分音的共振峰合成的多阶段技术（Bass and Goeddel 1981；Goeddel and Bass 1984）。此技术由建造

宽带(broadband)谐波信号开始。之后的是调整权重阶段,强化或减弱信号中的不同谐波,创造出时变共振峰区域,仿真传统乐器的频谱。

在 WF 合成的初始阶段所使用的宽带信号建构模块,是窗函数脉冲(window function pulse(图 7.24a)。窗函数是特别的波形,使用在许多信号处理工作中,如滤波器设计与声音分析。详见第 13 章与附录。

目前已发明了多种窗函数(见附录与 Harris 1978; Nuttall 1981 的讨论)。窗频谱图会显示其独特的中央波瓣(center lobe)与旁波瓣(side lobe)。中央波瓣的强度远比旁波瓣高,表示信号事实上是限频的。Bass 和 Goeddel 选择了布莱克曼—哈里斯(Blackman-Harris)窗函数,旁波瓣的强度衰减至少 60dB(图 7.24b)。由于可听见的谐波位于中央波瓣内,这可确保不会发生混淆的问题(见第 1 章对混淆的讨论)。

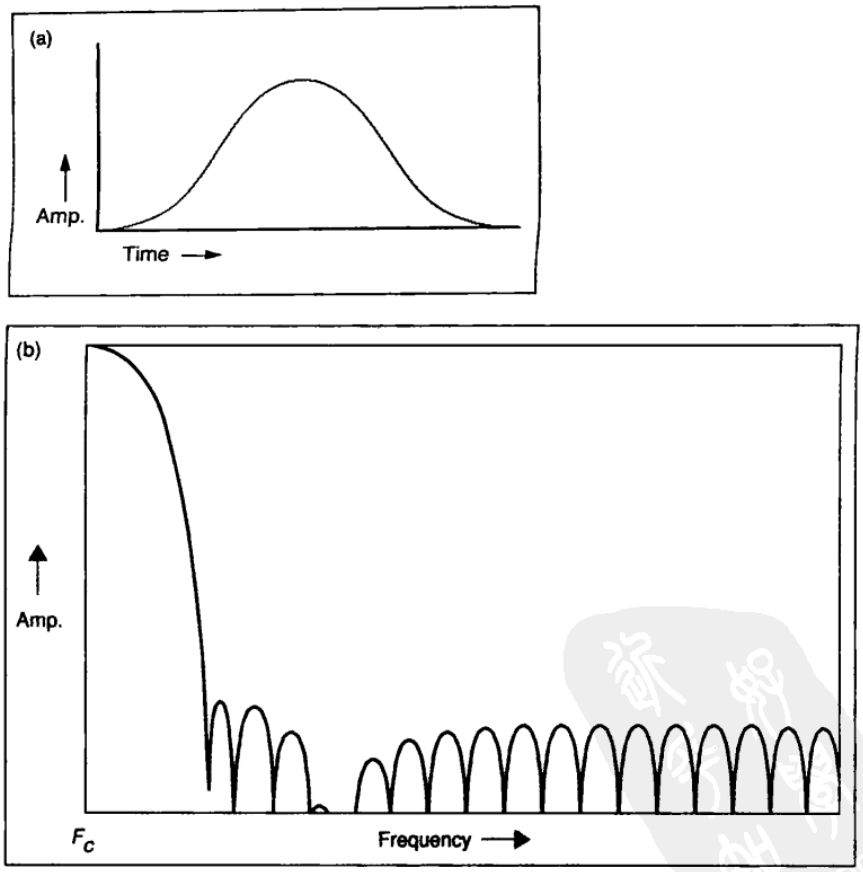


图 7.24 窗函数脉冲。(a)在时域的脉冲;(b)频谱的一边。图的左端相当于脉冲的中央频率,而波瓣代表边带,所有都比中央频率峰值小了起码 70dB(Nuttall 1981)。

宽带信号是将 WF 脉冲连接成串所建立，在脉冲间，强度为 0 的时段称为静止时间(deadtime)。对不同的基频，WF 脉冲的时长保持相同；仅有脉冲间的静止时间改变。图 7.25 显示相距八度音的两信号，其中不同处仅有静止时间的长度。WF 技术所采取的脉冲后接静止时间的方式，与前述 VOSIM 及 FOF 法不同。我们之后将详细解释，WF 合成，如同 VOSIM 与 FOF，是将数个发生器相加，产生复杂的时变频谱。在其他层面上，这些技术并不类似。

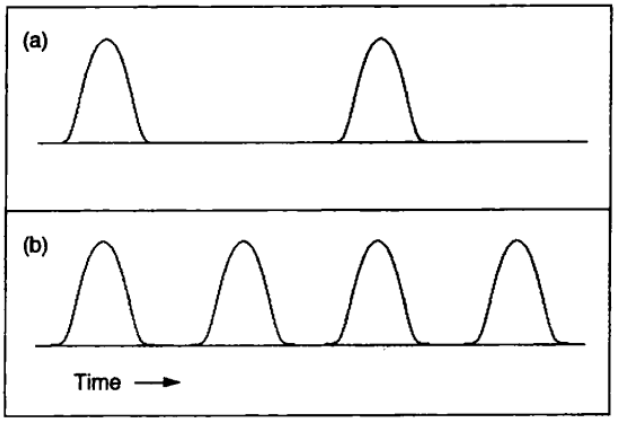


图 7.25　相距一个八度的两 WF 信号之时域图。(a)低频信号；(b)高频信号。

在 WF 合成中，泛音的数目会随着基频降低而增加。这是因为高频谐波会超过 WF 脉冲频谱的中央波瓣。所以较低音的音色较丰富，而较高音的音色则较薄。这是传统乐器如风琴与钢琴的特征，也就是 Bass 与 Goeddel 所要仿真的。但要注意在其他乐器上，如古钢琴，并没有这种性质。另外，有些乐器并没有完美谐波频谱，所以并无法用 WF 技术建构良好的模型。

目前为止，我们所讨论的架构是产生固定音质的。这些声质可以是宽带(较低基频)到窄带(较高基频)间的音色。为了要创造频谱上的共振峰区域，需要进一步的，称为时隙加权(slot weighting)的处理。

时间槽(time slot)为单一 WF 脉冲加上静止时间的整段时长。以一个周期性的 N 槽权重序列(N slot weights)，为不同时槽加权(即一个数值与一个槽相乘)，就能够操控输出信号的音色。乘法器计算每个脉冲与其相对的权重值，得到的结果是带有不同强度的 WF 脉冲输出串流(图 7.26)。此串流的频谱显示不同频率上的波峰与波谷。对于时变音色，每个时槽权重可以由时变函数来指定。

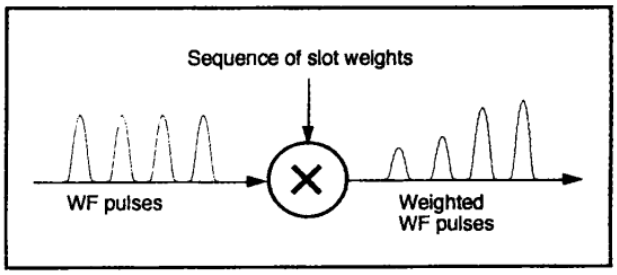


图 7.26 一串窗口函数脉冲乘以一个周期性的槽权重序列，以获得一系列的加权窗口函数脉冲。
Sequence of slot weights=槽权重序列 WF pulses=窗口函数脉冲
Weighted WF pulses=加权窗口函数脉冲

WF 合成需要强度补偿方案，因为低频信号的脉冲较少，而且有较多强度为零的静止时间，而高频信号有较多脉冲，并且几乎没有静止时间。可使用接近线性，与频率成反比的比例函数调整强度。也就是说，低频声音会被强化，而高频声音会被减弱，以求得频率范围间的平衡。

如同卡普拉斯—斯特朗的拨弦与鼓算法，基本的 WF 算法也可以由许多技术继续延伸，增加弹性，同时保持计算效率。更多的细节，可见 Bass and Goeddel (1981)，Goeddel and Bass (1984)。根据 Bass 与 Goeddel，在实际应用中，使用 8 个 WF 振荡器，一周期内有 256 个时槽(最大值)，取样率 40kHz，WF 脉冲宽为 150 微秒，以及对仿真每个时槽权重的 28 线段时间函数的情况下，可以得到合理的传统乐器仿真音色。

图 7.27 显示两个中音萨克斯管声音。(一般来说，中音萨克斯管对于声音合成是个很困难的测试。)图 7.27a 显示原始声音，图 7.27b 显示由 WF 技术产生的合成声音。

结 论(Conclusion)

本章探索广泛的合成技术，包括物理模型合成，卡普拉斯－斯特朗(Karplus-Strong)算法，以及许多不同的共振峰方法。并非所有技术都在商业系统中采用，还留有许多实验机会。

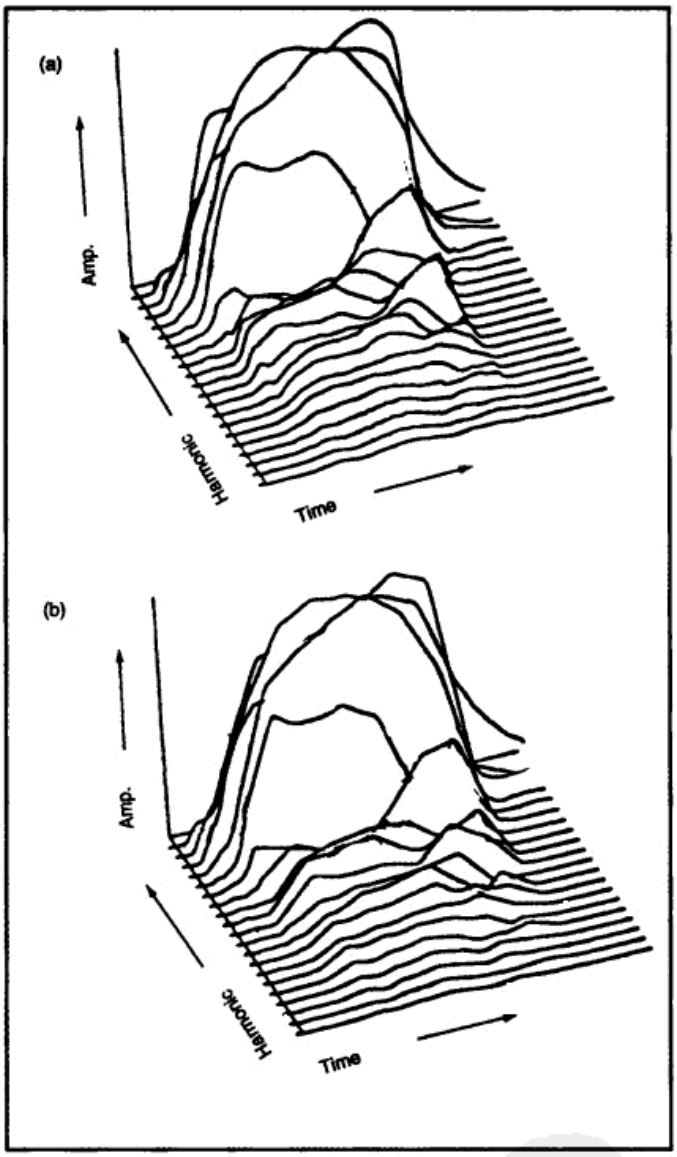


图 7.27　中音萨克斯管的前 20 个泛音的时变频谱。低频在后方。(a)原始的萨克斯管声音;(b)由 WF 合成法产生的合成音。(出自 Goeddel and Bass 1984。)

物理模型呈现了一个庞大而且具有潜力的声音合成资源,而这在音乐创作上的应用才刚开始。物理模型可以捕捉许多表现性的方面,是计算机音乐家目录中受欢迎的新项目。

共振峰合成让音乐家直接处理最重要的声音记号之一,也就是频谱峰的强度与间隔。

第8章 波形片段、图形以及随机合成 (Waveform Segment, Graphic, and Stochastic Synthesis)

波形片段技术(Waveform Segment Techniques)

波形内插(Waveform Interpolation)

线性内插方程式(Linear Interolation Equation)

振荡器与包络发生器间的内插(Interpolation in Oscillators and Envelope Generators)

GEN 函数的内插(Interpolation in GEN Functions)

内插合成(Interpolation Synthesis)

SAWDUST

SSP

指令合成(Instruction Synthesis)

图形声音合成(Graphic Sound Synthesis)

声音合成中的图形:背景(Graphics in Sound Synthesis: Background)

与 UPIC 的互动(Interaction with UPIC)

最初的 UPIC (The First UPIC)

实时的 UPIC (Real-Time UPIC)

使用 MIDI 的图形合成(Graphic Synthesis with MIDI)

图形声音合成的评估(Assessment of Graphic Sound Synthesis)

噪声调制(Noise Modulation)

对于噪声的论述(Discourse on Noise)

噪声调制的 AM 与 FM(Noise-modulated AM and FM)

使用随机成形函数的波成形合成(Waveshaping with a Random Shaping Function)

随机波形合成(Stochastic Waveform Synthesis)

动态随机合成(Dynamic Stochastic Synthesis)

GENDY

结论(Conclusion)

本章探讨四种“非常规”合成法。所谓非常规，是指它们并不是以仿真传统乐器为出发点，而是创造出新的电子声音。由作曲美学，以及对新的音乐材料的创造性想象的追求，刺激了这些技术的发展。

这些技术研究包括以下几个方面：

波形片段(waveform segment)技术是计算机特有的。它们由指定个别强度点，以及建构连结各点的简单波形片段的命令开始。这些命令将线段连接成更复杂的波形。这种技术的一种版本可以自动产生波形，作为一个由“虚拟机器”执行作曲家编程指令的副产品。

图形合成(graphic synthesis)依循视觉与雕塑方面的原则对声音进行规定。作曲家创作时以绘图板或屏幕绘出声音图形。这些图形被解译成声音。另外，也可以用图形工具转换出取样声音的图形。

噪声调制(noise modulations)是能产生从异步揉音或颤音，到一整个频带区间的有色噪声的理想方式。

随机合成技术(stochastic synthesis techniques)由参考受时变限制所控制的几率函数，计算出声音波形。

这些技术的共通点是它们都可以轻易地产生丰富、宽频和嘈杂的声音，要以其他技术达成十分困难。所以，它们构成了整个声音连续整体的重要支点。

波形片段技术(Waveform Segment Techniques)

所有声音感知中的不同，可以追溯回其波形瞬时结构(temporal structure)的差异……如果声音的所有经验特性，都可以上溯为一种次序的单一原理——如被瞬间组成的连续脉冲——作曲思考将从最根本处重新定义方向……我们不会再从那些已被人体验过的声音特质下手，并允许这些性质决定那些瞬时变化；取而代之的是，我们将就那些脉冲本身的瞬时安排进行作曲，并以实验发现它们所产生的性质(Karlheinz Stockhausen 1963)。

波形片段技术构建了一套方法，将单一取样点以至波形片段加以连接，以创造出较大的波形、段落及整首乐曲。事实上，数字声音是由不可再细分的粒子，也就是取样点，所创造出来的。波形片段技术代表合成的时域方法，因为它们由独立振幅点创造出声音。诸如“频率”或“频谱”的观念，可能无法明确地由合成参数显示出来，但可以在作曲的处理过程中以副产品的形式出现。

此节描述四种波形片段技术：

- 波形内插法(waveform interpolation)
- SAWDUST

- SSP
- 指令合成法(Instruction synthesis)

波形内插可直接与频域相关，因为可以预期内插法对于信号频谱的效果，这一点我们以后会谈到。此处介绍的两个技术 SAWDUST 与 SSP 中，作曲家直接在取样点上操作。时变频谱可以在作曲家对波形的操作过程中得到。指令合成是一种抽象的合成方式，因为作曲家以逻辑指令指定声音，与声学参数间没有直接关系。

波形内插(Waveform Interpolation)

内插是种产生两个端点(endpoints)或折线点(breakpoints)之间联机的数学技术。而每个折线点都是由一对(x 坐标，y 坐标)数值所描述。现有许多内插算法，包含常数(constant)、线性(linear)、指数(exponential)、对数(logarithmic)、半余弦(half-coscine)以及多项式(polynomial)等。每一种方法都会在两个端点间建立不同种类的曲线。如图 8.1 所示，常数内插在两端点间绘出与横坐标平行的直线。线性内插则绘出连接端点间的直线。

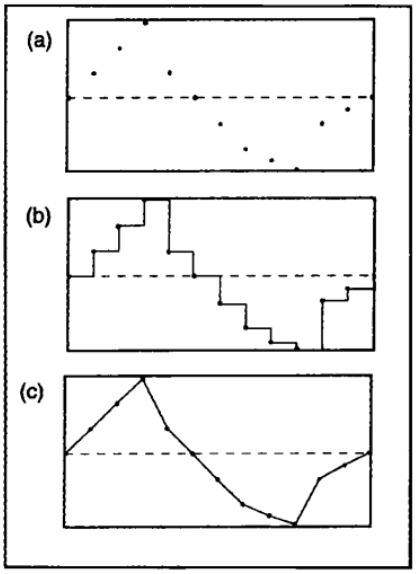


图 8.1　简单内插技术。(a)原始折线点；(b)常数；(c)线性。

在半余弦内插中，两曲线端点(曲率)确保了两端点间为平滑曲线。图 8.2(a)显示两点间的半余弦内插。图 8.2(b)显示数点间的半余弦内插。多项式内插技术[包含三次样条(cubic splines)与切比雪夫(Chebychev)多项式]，则依其所使用的多项式，指定两端点间平滑或任意改变的曲线。

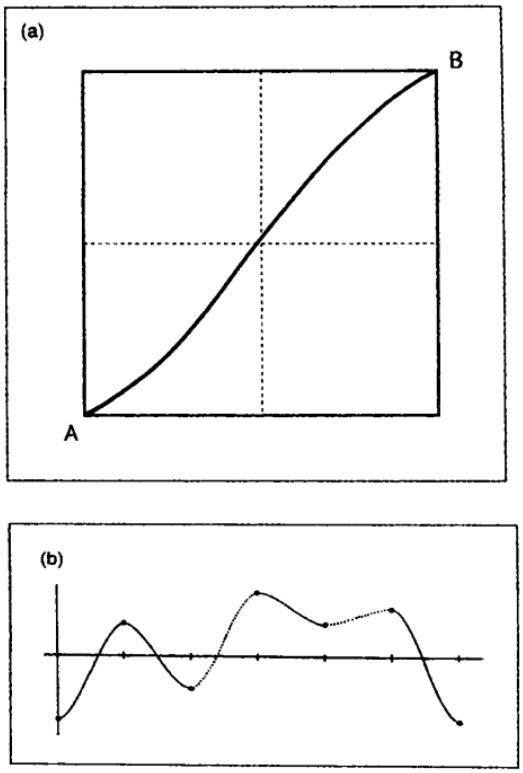


图 8.2 半余弦内插。(a)A、B 两点间半余弦曲线，注意两曲线点(弯曲点)；(b)数点间的半余弦内插。(出自 Mitsuhashi 1982b。)

线性内插方程式(Linear Interolation Equation)

线性内插是最简单且最广泛使用的内插方式。它试着找出两已知点间的中间点 i。要达到此目的之方程式如下：

$$f(i)=f(start)+\{([i\text{-}start]/[end\text{-}start])\times[f(end)-f(start)]\}$$

其中 $f(start)$与 $f(end)$分别是起始与结束端点，i 是在起始与结束点间横坐标上的中间点。线性内插找出 i 与开始点及结束点间的距离，将此比例乘上 $f(end)$与 $f(start)$之间的差，并把它加在 $f(start)$之上。

振荡器与包络发生器间的内插 (Interpolation in Oscillators and Envelope Generators)

计算机音乐系统经常用到内插法。例如，可在振荡器(Moore 1977)与包络

发生器内找到它们。第 3 章解释了内插振荡器比起非内插振荡器，有大幅改进的信噪比。在包络产生器内，也使用内插法来连接描述包络轮廓的成对的折线点(x-y 坐标)。这种方法较将包络每点储存的方法要节省内存，但需要较多计算。

内插也可用来在现有的波形之上产生新波形。比方说在 Music N 语言的某些应用当中就包含了波形内插的单元发生器(Leibig 1974)。这些单元在输入端接收两个信号，并产生两个信号间的权重内插信号(图 8.3)。随着时间改变权重，我们就能时变地交迭两个输入波形。

GEN 函数的内插(Interpolation in GEN Functions)

第 3 章与第 17 章中，几种 Music N 语言内的表生成函数(table generation function，GEN)会在作曲家指定的端点间做内插运算。这些 GEN 函数会建立包络与波形，供 Music N 乐器使用。Music N 内典型内插 GEN 函数，包含线段(线性内插)、指数、三次样条(多项式)以及切比雪夫(多项式)。

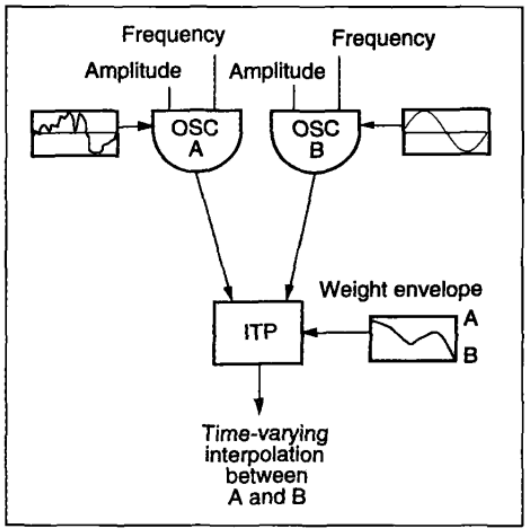


图 8.3　波形内插乐器，使用 Music N 合成软件语言中的 ITP 单元生成器。权重包络指定哪个波形所占成分较大。当权重包络为 1 时，会听见左边的振荡器；当它是 0 时，会听到右边的振荡器；当它是 0.5 时，则会是两波形点对点的平均。
Amplitude=振幅　Freiquency=频率　Weight envelope=权重包络
Time-varying Interpolation between A and B=在 A 和 B 之间进行时变性内插

内插合成(Interpolation Synthesis)

Bernstein 和 Cooper(1976)提出了一种完全由线性内插为基础的波形合成方式。在此方法中,假定一段波形内有 n 个以相等时间间距分布的端点。波形的线性内插最主要的缺点在于会造成波形上的尖角,而产生尖锐刺耳,无法控制的高频泛音。Mitsuhashi(1982b)提供了几种线性内插法的替代方式,包含常数内插、半余弦、多项式内插。他证明了常数内插法在产生的波形上(全为直角),以及建立波形所需要的参数数目上,接近于沃尔什(Walsh)函数合成(见第4章)。相对于沃尔什函数合成,常数内插不需要沃尔什合成所需的权重系数相加。所以,它具有更高效率的潜力。但不幸的是,如同线性内插,常数内插也有产生无法控制的高频泛音问题。

半余弦内插就没有这种问题了。Mitsuhashi 使用半余弦内插函数,可决定波形中的谐波混合成分,造成等同于加法合成的结果。半余弦内插的优点,在于它比加法合成使用了较少的计算资源。

Mitsuhashi 也分析了任意多项式产生的内插。当使用了等距的端点间距,可用前向差分(forward difference)方法,有效率地计算出多项式。使用前向差分方法,做出多项式内插的数学细节在本书讨论的范畴外。详情可见 Mitsuhashi(1982a, 1982b),以及 Cerruti 和 Rodeghiero(1983)。

经内插得到的信号频谱,是以下两项条件的结果:端点 $f(i)$ 的纵坐标以及所选择的内插函数。当合成一周期内带有 n 个端点的周期波形时,可以由改变端点高度(纵坐标)控制 $n/2$ 个谐波的强度(Mitsuhashi 1982b)。所以,如果端点数目为 20,可以控制 0 到第 10 个谐波。

接着,也可以每周期改变端点的纵坐标,来产生时变频谱。端点在纵坐标的线性变化,会方便地造成谐波强度上的线性改变。

到目前,我们所考虑的都是端点等距分布的情况,但也可使用非等距端点间距。使用仔细选定后的非线性端点间距,会比线性间距更好地接近指定波形。这会有较少的失真。图 8.4 指出等距分布端点所提供的波形近似,远逊于将端点放置在变化最大处的非线性端点的近似。Bernstein 和 Cooper(1976)指定了傅里叶系数,决定由非线性端点间距所近似波形的频谱。此方法需要更深入的研究,以决定其优点与缺点。

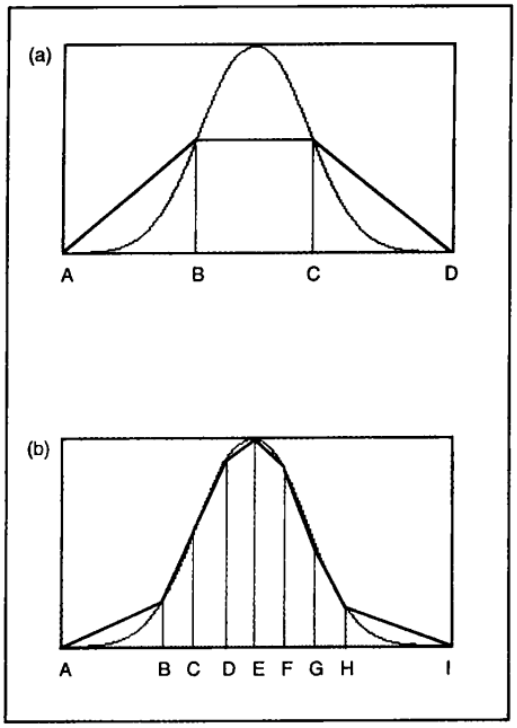


图 8.4　非线性端点的效果。(a)由平均分布端点所绘出的波形;(b)由非线性分布端点所绘曲线,得到一个较佳的曲线拟合。

SAWDUST

SAWDUST 系统由赫伯特·布隆(Herbert Brün)设计,由伊利诺伊大学(University of Illinois)的一组程序设计师所实现(Blum 1979),该系统代表一种声音合成的原创方式。(见 Grossman 1987 从实施的观点对 SAWDUST 的评论。)

"sawdust"一词由两个字结合而来:"saw"——工具与"dust"——碎屑,处理的副产品。在布隆的观念中,"saw"是计算机,而"dust"是数据。由微粒强度点(取样)所组成。SAWDUST 系统是个操纵强度点(Brün 称为元素 elements)的交互式环境,并以层级的形式结合成波形,段落,最终成为完整的作品。对其他波形片段技术来说,由 SAWDUST 所产生的信号通常具有生硬,带锯齿状的音质。

SAWDUST 的基本操作,包含结合元素循环(looping)、混合以及改变。此操作是由次程序 LINK、MINGLE、MERGE 与 VARY 所执行。LINK 是种排序函数,将未排列的元素 *A* 集合转换为一组称为 link 的有序列表。

MINGLE 是循环活动,将一组已排列的 link,组成重复 *n* 次的一组新集合。此机制是在 SAWDUST 中建立周期波形。比方说 MINGLE(*2*, *L3*, *L4*)=

{*L3*, *L4*, *L3*, *L4*}。

MERGE是排列动作,交替地选择两 link 中连续元素,以建立新的 link。比方说,如有两 link L_j 与 L_k,L_j = {*e1*, *e2*,... *e10*},L_k = {*e21*, *e22*, ... *e30*},那么 $MERGE(L_j, L_k)$ = {*e1*, *e21*, *e2*, *e22*,... *e10*, *e30*}。

VARY 将一个 link 转换为另一个。作曲家指定初始 link、时长以及最终 link。另外,作曲家规定多项式的阶数(degree),直到它折回到目的 link 中相应的端点。

SSP

SSP 是一种波形片段合成系统,由德国—荷兰作曲家 G. M. 凯尼格(G. M. Koenig)设计,并由声响技术研究所(Institute of Sonology)(乌得勒支,Ultrecht)的保罗·贝尔格(Paul Berg)在 20 世纪 70 年代晚期所实现(Berg 1978b)。如同布隆的 SAWDUST,SSP 也是操纵单一元素,形成波形与大型作曲结构(large-scale compositional structures)的互动结构。

SSP 是由序列主义或后序列主义作曲家设计。所以,此系统主要受益于第二次世界大战后的作曲理论,而非信号处理理论。特别是 SSP 的程序数据库的使用可以直接追溯到凯尼格的作曲程序 Project 1(Koenig 1970a)和 Project 2(Koenig 1970b)中序列及后序列中的选择原则(selection principle)。这些程序操作作用于元素与片段。SSP 中的元素是时间与振幅点,也就是取样点。SSP 系统以内插方式连接作曲家选定元素间的取样点。片段则是操作元素后建立的波形。

使用 SSP 工作时,作曲家准备时间点数据库及振幅点数据库。作曲家再将时间点与振幅点联合,可以指定常见的波形样式,如正弦波、锯齿波、三角波或者可能是由几率程序得来的更特别的样式。SSP 的选择原理,创造或萃取出元素数据库的部分,结合成波形片段。作曲家以其他的选择原理的程序,决定出片段的时间顺序。表 8.1 列出 SSP 中的六种选择原理。

表 8.1 SSP 中的选择原理

选择原则	自变量	解释
Alea	A,Z,N	*A* 到 *Z* 间 *N* 个随机选取的数值。
Series	A, Z, N	*A* 到 *Z* 间 *N* 个随机选取的数值。数值选取后,会从现存可用值集合中移除。当此集合无物时,会被再次填满。
Ratio	Factors, A, Z, N	*A* 到 *Z* 间 *N* 个随机选取的数值。由 *A* 到 *Z* 的发生几率,是以被称为 *Factors* 的几率权重来确定的。

续表

选择原则	自变量	解释
Tendency	N，M，A1，A2，Z1，Z2...	对每个 *M* 趋势屏蔽(*tendency mask*)选取 *N* 个随机数值。此 *N* 个数值会在初始边界 *A*1 与 *A*2 与末端边界 *Z*1 与 *Z*2 间选取。
Sequence	Count，Chunks	直接指定一串元素。Count 是指定元素的个数。Chunk 是它们数值的列表。
Group	A，Z，LA，LZ	在 *A* 到 *Z* 之间选取一个随机值，这将发生一次或多次，形成一组集合。此集合的大小是在 *LA* 到 *LZ* 间随机选取的。

布隆的 SAWDUST 与凯尼格的 SSP 都适合在带有数字模拟转换器的小型计算机上运作。这两种合成方法所产生的声音材料，都比较倾向具有未经处理的，并有丰富频谱的波形，这是在标准声学与信号处理模型中所无法得到的。

指令合成(Instruction Synthesis)

指令合成(也被 G. M. 凯尼格称作“非标准合成”，见 Roads 1978a)使用计算机指令序列(比方说，二进制相加、相减、AND、OR、循环、延迟、分支)来产生并操纵二进制数据。这资料被视为一组声音取样，送至数字模拟转换器。当然所有合成法都在软件的最底层使用计算机指令处理。而指令合成的重点在于，声音是完全由逻辑指令指定的，而不是从传统的声学或信号处理概念产生的。

“由指令合成”在观念上与规则合成或第 7 章所讨论的物理模型合成相对。物理模型从声学机制的数学描述开始。此模型可以非常复杂，需要大量计算。相对地，指令合成从计算机指令的特有操作方式开始，没有声学模型。此技术相当有效率，并且可以在最便宜的微机上实时运行。

由指令合成产生的声音，在特性上不同于由规则产生的合成声音。许多情况下，使用“标准”数字或模拟合成技术很难产生这种声音，更不用说是使用机械—声学方法了。

指令合成研究的主要部分是声响技术研究所的研究人员完成的，最开始是在乌得勒支，后来移至海牙。指令合成系统的一个领域，是发展虚拟机器的汇编器(assembler)(Berg 1975；Berg 1978a，1979)。汇编器是一种底层的程序语言，每一行程序有一个相应的机械指令。虚拟机器是仿真抽象计算机运作的程序，它有自己的指令集、资料形态等。这些系统要求作曲家撰写相当完整的程序，以产生个别的信号。此程序是作品的详细说明，所以也是乐谱。

保罗·贝尔格的 PILE 语言(Berg 1978a，1979)是指令合成的典型范例。PILE 语言的动机是源于"计算机迅速产生并操纵数字及其他符号信息，这可以视为计算机的惯用语"的美学信念(Berg 1979)。为了实现这一理念，贝尔格设计了一个处理数字及符号的虚拟机器，由一个为小型计算机撰写的程序对此加以模仿。PILE 语言是虚拟机器的指令集。由虚拟机器执行这些程序，制造出取样，送入数模转换器(DAC)。

PILE 的指令集包含如 RANDOM(选取随机数)、INCR(对一个数值加一)、SELECT(赋予变量一个随机值)以及 CONVERT(将一个样本送入数模转换器)的动作。另一些操作则改变比特屏蔽，并且通过各种随机操作和安插延迟等方式来控制程序的流程。虽然可以在 PILE 中仔细控制音高、时长、音色贝尔格做了一首流行歌曲，证明此说法)，此语言的使用，较倾向互动的声音实验以及尝试—错误式的即兴创作。由于随机变量的存在，无法预期由 PILE 一组指令所产生之声音结果。这与此语言发明者的探索性美学思维相一致。

*Holtzman 的系统(1979)是从较高等级来控制指令合成的尝试。*他发展了产生短暂声音合成程序的程序生成器。作曲家通过使用高级语言可以指定这些程序执行的顺序。

由于指令合成的性质，它所产生的声音总是无法预测。接受此特性后，作曲家可用尝试—错误式的方式来使用指令合成。由于这可以很简单而快速地产生非常广泛的声音类型，可以在录音间内尝试许多可能性，之后再由作曲家选取最有用的声音。

图形声音合成(Graphic Sound Synthesis)

图形声音合成是从视觉方法开始到产生声音的研究。这些系统将图形转为声音。此节我们将检视此方法的历史，接着把焦点放在根据这一原则所发展的近期成果上。

声音合成中的图形：背景(Graphics in Sound Synthesis: Background)

"自由音乐(Free Music)不需要由人来演奏。如同大多数的音乐，它是带有感情的，而不是大脑的产品，而且应该以精细控制的音乐机器，直接将作曲家的想象传递到听者的耳中。"(Percy Grainger 1938，出自 Bird 1982)

产生声音的图形技术拥有丰富的历史。在 1925 年，R. Michel 申请了摄影音乐记谱法的专利，与制作光学影片音轨的技术相似(Rhea 1972)。四年后，

A. Schmalz 以光电声音发生器(photoelectric tone generator)发展了电子音乐乐器。将新的表音图(phonogram)(一张将波形刻在玻璃上的图形)放入乐器内,由声音发生器发出的音色也随之改变。

这些早期实验由基于旋转光电声音发生器(rotating photoelectric tone generator)的商业性乐器延续下来,如"细胞音素"机,超级钢琴,维尔特风琴,"合创"风琴,以及光电声机(Celluophone, Superpiano, Welte Organ, Syntronic Organ, and Photona)。后两者是由 Ivan Eremeef 发展,在费城的 WCAU 广播电台完成。Eremeef 的顾问及赞助者是著名的指挥家斯托科夫斯基(Leopold Stokowski)(在 1920 年间首演许多瓦雷兹的作品)。这是 20 世纪 50 年代以前,工程师与重要音乐家的少数合作代表之一。可参见 Clark(1959)对于光电乐器的描述。

也许最具想象力且最精细的光学技术应用,是由加拿大的电影制作人 Norman McLaren。他费尽苦心一次一帧地,将声音波形直接画在影片扣链齿轮的光学音轨上(McLaren and Lewis 1948)。

光学技术同时也应用在控制模拟合成上。在英国的 Daphne Oram 所发展的 Oramics Graphic System(Douglas 1973),作曲家在透明胶片上画出模拟合成器的控制函数。这些函数决定音高、揉音、颤音、滤波器设定以及声音的振幅。扣链齿轮胶片通过光学扫描头,此扫描头将图形转为电子控制电压,送入合成器中的许多模块。

另外一组乐器可以扫描图形记谱。L. Lavallée 的 sonothèque 或"声音图书馆"读取以图形编码的声音。可导电的墨水通过一连串充电的电刷加以感应(Rhea 1972)。Cross-Grainger 的自由音乐机器(Free Music Machine)(1944 年的最初版本)读取书写在纸上的图形记谱(Bird 1982),并能用八个真空管振荡器来合成声音。

Hugh LeCaine 的"编码音乐机"(Coded Music Apparatus)(1952)让作曲家以五个连续的曲线控制声音:音高、振幅以及三个音色控制(Young 1989)。他的振荡器组(Oscillator Bank)(1959)是由光学器材驱动来扫描类似频谱图(sonogram)的乐谱。由 O. Kendall 在 20 世纪 50 年代晚期开发的"作曲器"(Composer-Tron),扫描在阴极射线管(显示画面)表面上由手绘出的包络曲线,然后用这些包络控制模拟合成器材。

对数字声音的图形控制是由 Mathews 与 Rosler 的实验开始的(1969)。(见第 16 章关于图形数据项)。近年来,在个人计算机上已实现了数种图形合成系统。(举例而言,可见 Oppenheim 1987)下节所介绍的 UPIC 系统,是发展程度最高的。

与 UPIC 的互动(Interaction with UPIC)

UPIC(Unité Polyagogique Informatique de CEMAMu)是由克赛纳基斯构思,由巴黎 Centre d'Etudes de Mathématique et Automatique Musicales(CEMAMu)的研究者所建构的合成系统(Xenakis 1992)。UPIC 系统结合了许多合成方法以及有弹性的图形使用者界面,以建立独特的声音作曲法。

最初的 UPIC(The First UPIC)

UPIC 的最初版本早在 1977 年出现。在此实现中,互动是由大型高分辨率的绘图板为中介,垂直地架设在画板上(Lohner 1986)。我们介绍此系统的某些功能,因为许多其他功能会在其后的 UPiC 系统中见到。

在创作细微结构的声音时,波形与事件包络可以直接绘于板上,并显示在图形终端。或者,作曲家可以敲出一组点,由计算机以内插方式连接。指定了波形与包络,就能试听它们的结果。

在较高层级的组织当中,作曲家可以画出频率/时间结构的乐谱页面(page)。当作曲家移动位置指定装置时,在 UPIC 的术语中称为弧线(arcs)的线段,就会出现在显示屏幕上。图 8.5 就是 Iannis Xenakis 的 Mycenae-Alpha,由 UPIC 系统所制作。

图 8.5 克赛纳基斯的 Mycenae-Alpha 的一页(1980),在 UPIC 系统上制作。纵轴为频率,横轴为时间。

音乐家也可以拥有录音、编辑以及为采样声音记谱(scoring)等选项。经采样的信号可以当作波形或是包络使用。当采样被当作包络使用时，会造成密集的振幅调制效果。如果需要，图形式乐谱可与相结合的合成及采样声音一起进行配器。

如图 8.6 所示，动作与图形间的互动使作曲家轻易地创造乐谱结构，任何其他方法都会非常麻烦。在页面的层级上，UPIC 可以同时捕捉细微结构的细节以及宏观结构的变化。

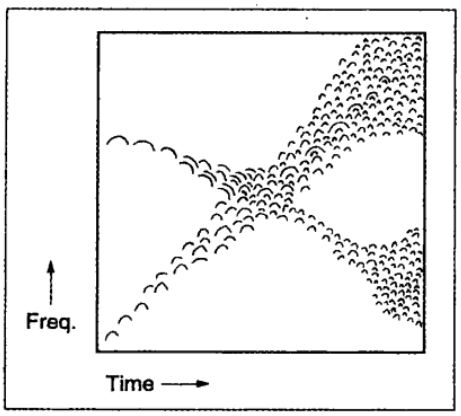


图 8.6　Curtis Roads 的 Message 的一页(1987)，每个弧线代表一个中音萨克斯管的音高曲线。

实时的 UPIC(Real-Time UPIC)

最初的 UPIC 系统在缓慢笨重的微机上运作。虽然图形设计是相当具互动性的程序，但需要一段时间从作曲家的图形乐谱计算出声音取样。以 64 个振荡器合成引擎为基础而研发的实时处理版本，是 UPIC 系统的重大突破(Raczinski and Marino 1988)。在 1991 年，此引擎可在个人计算机的视窗(Windows)操作系统中，执行复杂的图形界面(Marino, Raczinski, and Serra; 1990 Raczinski, Marino, and Serra 1991; Marino, Serra, and Raczinski 1992; Pape 1992)。

图 8.7 是由实时的 UPIC 所作的页面，此页同时可有 64 个弧线，一页可以有超过 4 000 个弧线。每一页的时间可以是 6 毫秒到超过 2 小时。编辑动作如剪、贴、拷贝、重新安排弧线位置等，也可以在时间或频率上延展或压缩。这些操作可以在演奏该页时执行。在同一页上可分配 4 种不同的音阶。当以不连续的音阶演奏时，跟踪频率梯级的弧线是由调音表(tuning table)来控制的。

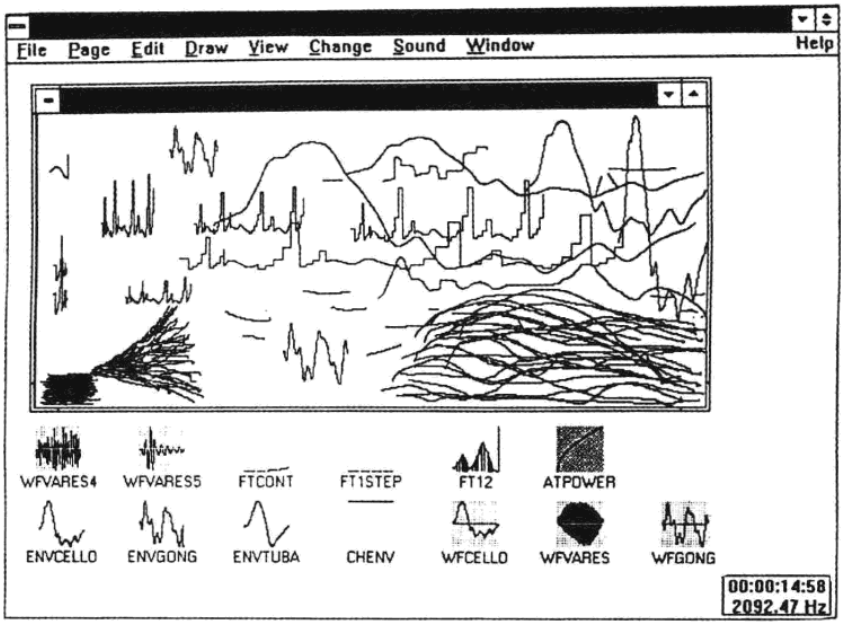


图 8.7 Gerard Pape 1992 年所作的乐谱页面，由实时 UPIC 系统实现，Paris，Les Ateliers UPIC。画面下半部分的图标表示可用的波形与包络。

实时合成使 UPIC 成为演出乐器。通常合成单元是由左至右演奏乐谱，由使用者定义该页时长后以匀速进行。然而，读取乐谱的速度和方向，也可以用鼠标实时控制。比方说，它允许由乐谱的一个区域跳到另一个区域。控制动作的顺序可以在演奏乐谱时由系统录下，接着可以再次演出或编辑同样的演奏。

使用 MIDI 的图形合成(Graphic Synthesis with MIDI)

跟随着 UPIC 的发展，出现了许多输出 MIDI 信号的图形式作曲环境(Yavelow 1992)。有些有很复杂的功能，如“多重谐波”模式，由鼠标选择的线条与其他数条在和谐音程上的线条会同时出现(Lesbros 1993)。

这种方法带来的问题，在于如何将大量的图形控制资料对应到有限的 MIDI 协议(见第 21 章对 MIDI 的讨论)之中。如图 8.8 的图形可能有超过 100 个同时发生的事件。很少 MIDI 合成器能够接收这么大量的数据，所以需要详尽地规划 MIDI 设置，才能够加以处理。

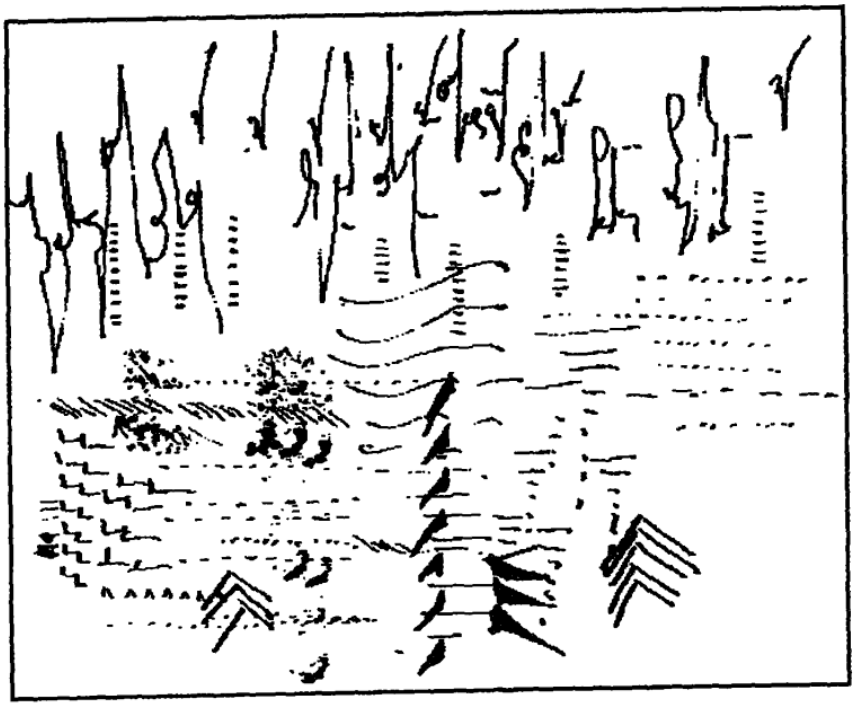


图 8.8　Vincent Lesbros 所绘的 Poly5 乐谱中的一页，它使用 1993 年发展的表音符号程序(Phonogramme)。注意使用多重谐波的绘图模式。在这种模式中，使用者绘出一条线段，就会在其上方自动产生多重谐波。

图形声音合成的评估(Assessment of Graphic Sound Synthesis)

图形声音合成是种直接的和直觉的声音雕塑方法。在时间一频率平面中绘出事件层级的声音，与图形合成间的互动可以是精准或不精准的，视使用者如何使用这些程序而定。若作曲家设计每一行程序，并将其对应到声音上，即可获得确切的结果。但若作曲家在屏幕上即兴，就可以将此界面视为他的素描簿，渐渐修改最初的草图，而完成最终的设计。

对于作曲家来说，音高的图形控制很自然，这使得建立旋律线与乐句变得相当简单，而这若以其他方式来做可能会很复杂。明显的例子包含带有多重滑音的微音程乐句或是有滑音或揉音内的精细变化。

包络形状的图形设计已在许多系统中证明是极为有效的。但手绘声学波形的问题仍然存在：我们无法一眼看出波形图听起来会是什以样子(第 1 章与第 16 章讨论此问题)。除了形状以外，任何没有改变的重复波形都只会是静态的音质。所以，与其他方式一样，在图形合成系统中，波形生成已由单一固定波形，演化为不断变化的声源，如取样声音或者一组时变波形等。

UPIC 系统是一种易于控制的音乐工具，因为它将不同的作曲层级结合在单一使用者界面内。在屏幕上直接做出图形函数，可以作为包络、波形、音高－时间谱、节奏曲线或者演出轨迹。这种对每个层级作曲资料的相同处理方式，建立了可以延伸到更多计算机音乐系统上的一般性准则。

噪声调制(Noise Modulation)

“我相信，使用噪声制作音乐，会持续下去并日渐增加，直到我们到达由电子器材制作音乐的时代……然而在过去，争论是在不谐和音与谐和音间。在即将到来的未来，争论将是在噪音与所谓乐音之间的。”(John Cage 1937)

本节探讨产生噪音的方法。原始想法是用经过滤波的噪声，来调制其他波形，如正弦波等。这范畴内的技术包含由噪声驱动的振幅调制、频率调制以及波成形合成。

对于噪声的论述(Discourse on Noise)

要实现噪声调制，我们需要噪声的数字来源。这将以一连串随机数值取样点的形式出现。但在数学上定义产生随机数的算法非常困难。(Chaiten 1975)。任何以计算机为基础的方法，所产生的随机数值终究会是有限的，已决定了的程序。所以我们称产生“随机”数值的算法为“伪随机数值发生器”(pseudorandom number generator)，因为这些算法产生的序列，会在数千或数百万次反复之后重复出现。程序语言环境提供的伪随机数值发生器有着不同的特性，如频率范围以及序列长度等。所以我们不会深入介绍算法的细节。Knuth(1973a)与 Rabiner 和 Gold(1975)的论文中有所介绍。第 19 章有此主题的额外批注。

由统计标准所定义的伪随机噪声，只是噪声音质的一种而已。许多合成技术也可以产生有趣的混乱噪声，包含正弦波调制(第 6 章与第 8 章)以及颗粒合成(第 5 章)。

的确，“噪声”一词是语言学中对于更加复杂而无法很好理解的信号，如管乐及弦乐中非谐波及混沌部分，或者打击乐器的起音瞬间的一种替代。在科学上正在开始理解产生这些空气压力曲线的程序，并非一定是“随机”行为(不管它指的意义是什么)。

当前音乐声学的主要挑战，是建立更多仿真噪音模型的高深算法。比方说，以整体的统计标准定义伪随机数值序列，并不是描述许多种噪声的最佳方

式。如在早期的鼓机的经验中，以白噪声替代铜钹的效果很差。非线性混沌(nonlinear chaos)模型——建立复杂行为的可确定算法，已经在科学家所观察到的某些现象中取代了随机模型。(Gleick 1988)。第 19 章将介绍此主题。

噪声调制的 AM 与 FM(Noise-modulated AM and FM)

"将自己陷于偶然调制的作曲家将……发现这种调制会直接将他带往一个以前被描述为'噪音'现象的世界。"(Meyer-Eppler 1955)

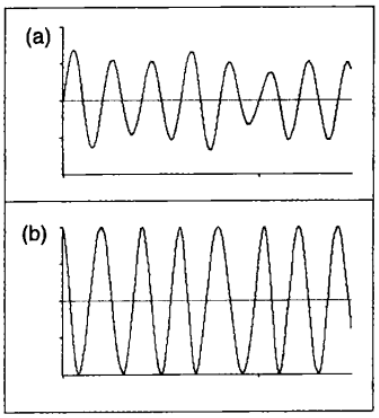


图 8.9　由噪音调制产生的波形。(a)经过低通滤波噪声，以 50%的强度的振幅调制正弦波；(b)经过低通滤波噪声，以 50%的强度的频率调制正弦波。注意每个音高周期的宽度会有些许改变。

噪声调制技术使用伪随机信号发生器或噪声发生器，控制振荡器的 AM 与 FM 调制。(见第 5 章对于 AM 及 FM 的讨论)。如图 8.9 所示，当噪声经过低于可听频率范围(小于 20Hz)的滤波器后，得到的效果是种偶然的颤音(在 AM 的情况中)或揉音(在 FM 的情况中)。

当噪声有较大频宽，得到的调制结果是有色噪声，也就是说，噪声带会集中在某个振荡器的载波频率周围。图 8.10 说明此噪声调制的 AM 与 FM 线路图。在两种情况中，使用经过低通滤波噪声源是个较好的主意。如此，噪声的随机程度接近于载波频率。如果噪声未经滤波，效果可能听来像在原始载波上加入高频噪声成分。

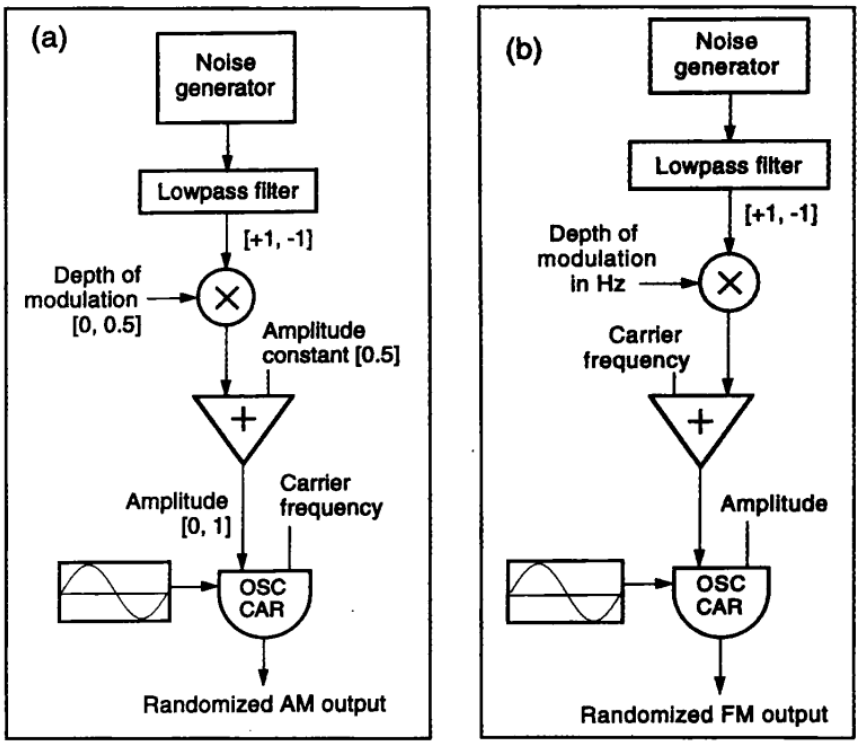


图 8.10 噪声调制乐器的线路图。(a)噪声调制 AM。噪声发生器的输出经过滤波，再经调制参数的深度缩放大小。接着，再加入振幅常数，以得到混合振幅值，送入振荡器；(b)噪声调制 FM，噪声发生器的输出经过滤波后，再经调制参数的深度缩放大小，指定了载波两边的带宽。

(a)
Noise generator=噪声发生器
Lowpass filter=低通滤波器
Depth of modulation=调制深度
Amplitude constant=振幅常数
Amplitude=振幅
Carrier frequency=载波频率
Randomized AM output=随机性 AM 输出

(b)
Noise generator=噪声发生器
Lowpass filter=低通滤波器
Depth of modulation in Hz=用赫兹来表示的调制深度
Carrier frequency=载波频率
Amplitude=振幅
Randomized FM output=随机性 FM 输出

使用随机成形函数的波成形合成 (Waveshaping with a Random Shaping Function)

利用第 6 章所介绍的波成形合成，可做出另一种噪声调制。在波成形合成中，实时信号振幅会依成形函数对应输出。随机的成形函数将扭曲周期波，成为更宽频的声音。图 8.11 说明四种越来越嘈杂的成形函数，而图 8.12 说明正弦波通过这四种成形函数的效果。

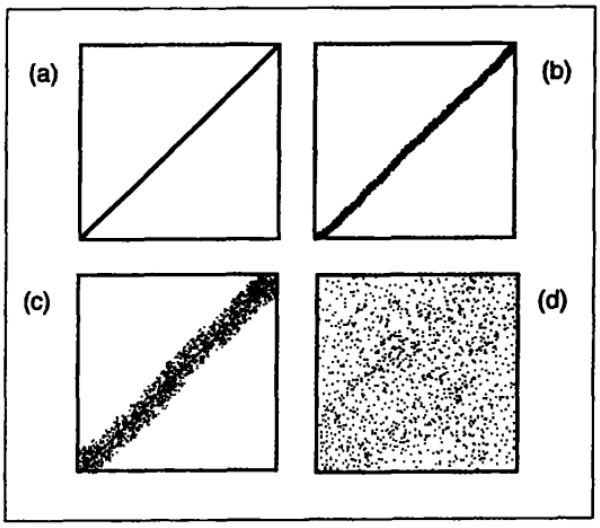


图 8.11　四种逐渐随机的成形函数。成形函数将输入数值(依下方坐标)重新对应为输出数值(依右方坐标)。详见第 6 章对波成形合成的讨论。

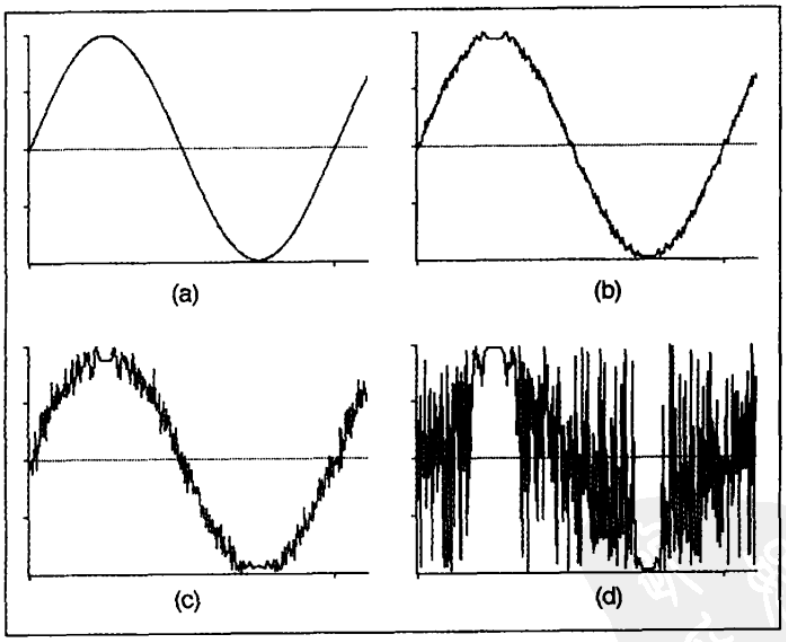


图 8.12　经过图 8.11 的成形函数的正弦波。

更微妙地应用波成形合成中随机性质，是在低振幅部分使用平滑的成形函数，而在高振幅部分引进渐强的随机性质。另外一种可能性，是将成形函数的随机程度与声音时长或其他的事件参数相连接。

随机波形合成(Stochastic Waveform Synthesis)

“乐音在其音色变化上过于受限。最复杂的管弦乐团,在音质上也可被简化为四五种乐器组:弦乐器、铜管乐器、木管乐器、打击乐器。现代音乐在这小圈圈里挣扎,徒劳地试图创造出新的音色变化。我们必须破除这声音的限制框架,征服无限多种的噪音!”(Luigi Russolo 1916)

随机波形合成由比较伪随机数值与几率分布(probability distribution),来产生声音取样。几率分布是一个曲线(储存在计算机内存数组中),用来指定可能发生事件的数值几率。在波形合成的情况中,“发生事件”是取样点的振幅值。第19章说明由几率查表法产生旋律。将此运算法调整为产生波形,所需的只是将音高数值替换为波形数值(也就是-1到1之间)。

由于波形产生算法每秒必须产生数万个声音取样点,少量地计算改进,也会为整体节省不少计算时间。比方说,其中的一个改进是先加载填满伪随机数值的大对应表,而不是为每个取样点调用产生随机数值的程序。如此,要得到随机数值仅需查表动作即可。

为某一作品的应用而发明一种适当的几率分布是种艺术。第19章介绍此主题并说明几率分布的种类。已有许多几率理论的文献资料;例如Drake(1967)的文章。具有音乐范例及程序代码的一篇卓越参考文章是《Lorrain》(1980);这篇文章修改和更正过的版本可在广泛流通的文选中找到(Roads 1989)。其他对于随机技术的文献,在作曲上的有Xenakis(1992)、Jones(1981)以及Ames(1987a, 1989a)。在碎型波形产生的相关实验可见Waschka和Kurepa(1989)(见第19章对碎型的描述)。

由简单的几率查表法,而没有后续限制的波形产生法,会产生固定频谱的噪声。所以加入限制是十分重要的——这些限制是外加的条件,改变几率以得到更有趣的时变声音。这是动态几率合成的目的,将在下面介绍。

动态随机合成(Dynamic Stochastic Synthesis)

在《形式化的音乐》中,作曲家I. 克赛那基斯(1992)提出了另外一种声音合成方式。与其由简单的周期信号,且试着让它们更生动,而加入“无序”(也就是许多失真与调制),何不由伪随机函数开始,而加入秩序(也就是权重、限制、界限)以驯服它们?这个想法以8种策略实行,开拓了波形合成的动态随机方法,列在表8.2中。

表 8.2　克赛纳基斯所提出的随机波形产生方法

1. 直接使用几率分布[如泊松(Poisson)、指数、高斯(Gaussian)、平均、柯西(Cauchy)、反正弦以及逻辑]以创造波形。
2. 将几率函数彼此相乘。
3. 以加法结合几率函数，有可能是随着时间改变的。
4. 使用振幅及时间的随机变量，作为弹性强制力或其他随机变量的函数。
5. 使用随机变量，在可变界限间来回弹跳。
6. 使用几率函数，产生其他几率函数参数的数值(这些后续的函数用来产生波形)。
7. 将几率曲线分成数个种类，并将这些种类视为高阶集合与程序的一个元素(也就是说，引入不同级别的波形产生控制)。
8. 将随机声音合成技术的选择注入随机作曲程序内(第 7 项的延伸)。

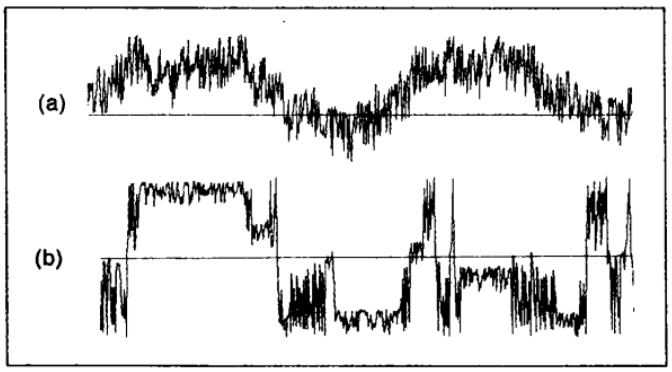


图 8.13　显示由随机法产生的两个波形。双曲余弦(hyperbolic cosine)函数与使用界限和非随机时间的指数密度的积。图 8.13(b)使用与(a)相同的算法，但其时间间隔为随机的。(出自克赛纳基斯 1992。)

GENDY

GENDY 程序(GENeration Dynamique)是随机动态合成的实际应用，在观念上与本章所提到的内插合成技术有关。此节特别对 GENDY3 程序加以说明(Xenakis 1992, Serra 1992)。

GENDY 由重复初始波形产生声音，接着将波形在时间及振幅上失真。因此，此合成方式由前一个波形的随机变化来计算新的波形。

在此程序中，波形是由多边形所表示的，限制在时间轴与振幅轴的边界中。多边形的线段是由时间与振幅轴上的顶点所表示(图 8.14)。此程序对顶点进行内插，算出期间的线段。

GENDY 依照许多随机分布，合成出这些顶点。如果此随机变化不在有限区间内，信号将迅速地变为白噪声。因此，此程序将时间与振幅的变化限制在镜线(mirror)边界内。镜线是由强度与时间边界组成。落在镜线之外的点会被折回到镜线之内(图 8.15)。事实上，镜线过滤随机变化以增加或减少强度界限，作曲家可控制反射的量值多寡。反射表示波形上的不连续，所以这是种音色控制。因为时间界限设定时间点之间的间隔，它将影响声音的感知频率。

因此，GENDY 系统的控制参数是时间线段的数值，镜线的界限，以及选定的时间与强度顶点的随机分布。这些是以每种音色的基础而设定的。图 8.16 显示 GENDY 的波形变化。它是伪周期性的由小镜子所决定。加上第二层镜线之后，可以在波形变化上加上类似揉音或颤音的效果。

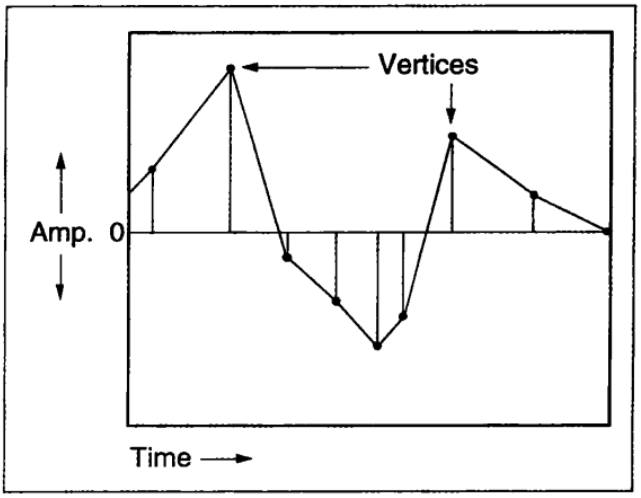


图 8.14 GENDY 中的波形结构。此波形是通过在时间—频率平面的顶点间画直线段而形成的一组多边形集合。注意在顶点间有不同的时间间隔。
Vertices=顶点 Amp.=振幅 Time=时间

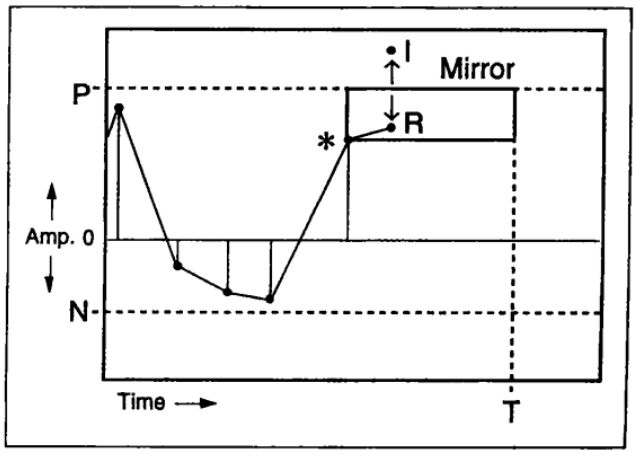


图 8.15 定义镜线的时间与振幅界限(P, N, T)，限制由星号点产生的下个顶点的位置。如果下个随机产生顶点(原始投射点 I)落在界限之外，界限 P 将取代该选择，将它反射回方块内(反射 R)。
Mirror=镜线 Time=时间

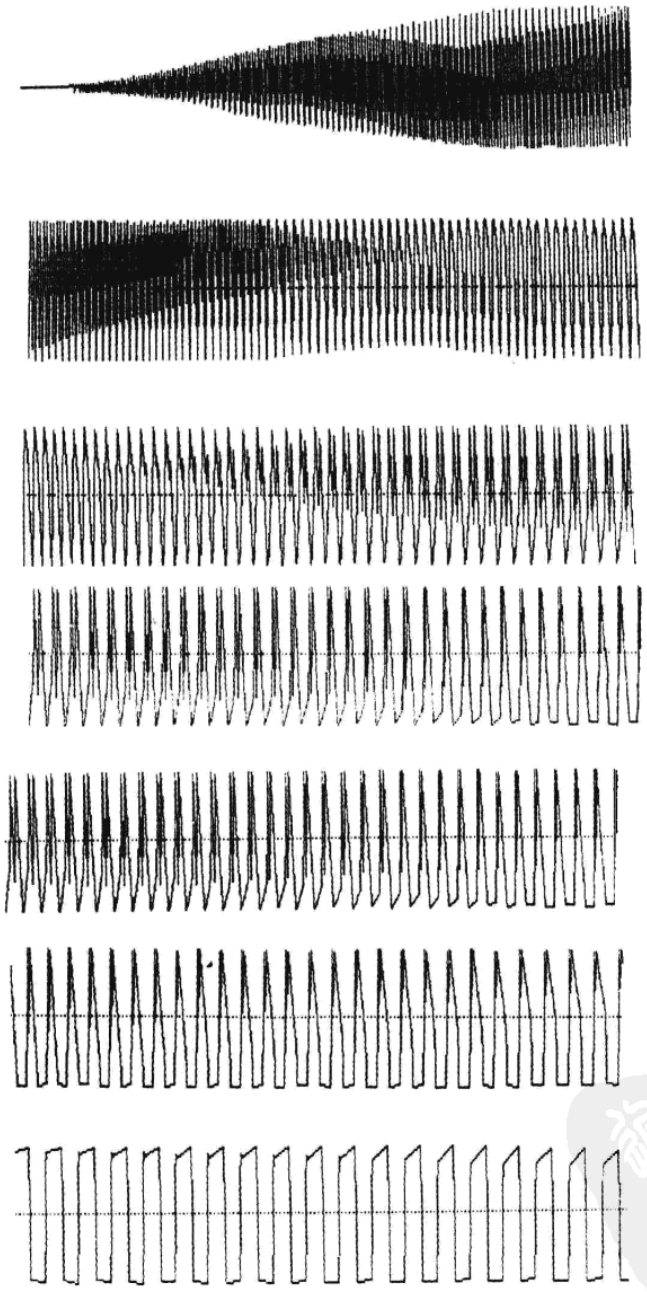


图 8.16　由 GENDY 产生的波形变化。此变化是由上方开始，持续向下，每行由左到右绘出波形。

结 论(Conclusion)

在1916年,路易吉·鲁索洛(Luigi Russolo)预言了建筑在"噪声艺术"基础上的音乐新世界。本章所介绍的技术,包含波形片段、图形、噪声调制和随机波形合成,将我们引领到未涉足的合成声音银河中。许多我们所遭遇的信号带有频谱上的暗流,淹没了基频的顶峰,并将其能量散在整个频域上。我们称此散布的能量为"噪声",一个会有不愉悦联想的名词。然而噪声元素长久以来是乐音的基础成分之一,包括从打击乐的尖锐敲击声音,到弓擦过弦时的柔和刮音,以及管乐器中感性的呼吸气息效果。本章解释的技术所产生的嘈杂声音,可以作为较标准方法所创造的平顺且熟悉音色的结构性装饰。所以,不应该忽视它们在音乐编配安排上的重要性。

第三部分　缩混与信号处理
(Mixing and Signal Processing)

PDG

第三部分概述(Overview to Part Ⅲ)

声音是一种易于调整的媒介形式。声波既可以被精确地融合到舒缓的音乐旋律中,也可以被灵活地并行放置,来产生清晰、强烈或震撼的效果。我们把对几个音频通道中的音频信号电平进行平衡处理的过程称为声音的缩混。在进行缩混时,经常会在声音上使用诸如滤波、延时、混响、定位等处理效果。这些操作归根到底就是信号处理的具体应用。在本书中的这一部分,包括第 9 章到第 11 章,是介绍在缩混和声音变换中的数字信号处理技术。

信号处理的另一部分内容是声音分析,由于它本身就是一个大的学科,因此在本书中单独成章来对声音分析的部分进行介绍。

为什么要学习信号处理?(Why Study Signal Processing?)

对于许多音乐家来说,对信号处理最初的接触是在使用效果处理器时发生的,如使用均衡器、延时器或是混响器等设备。大量商业性的设备试图减少其算法并通过少量的参数来进行控制。因此,还有必要做进一步的了解吗?事实上,对于音乐家来说,第一次面对信号处理的概念时,往往会觉得进入了一种纯数学的氛围,同时远离了音乐体验。那么为什么要学习信号处理呢?

答案有许多。一是为了掌握音乐素材的本质特性,我们必须要了解音乐素材的基础声学特性。那种将作曲和配器两者人为分开的做法,在计算机音乐中是不需要的。通过计算机来产生声音信号并对这些信号进行处理比使用笔写在纸上要来得更直接。

从理论的观点来说信号处理的一些概念并不是非常的深奥,但是它们却被普遍地应用,因此相应的基础知识就显得比较重要了,比如在缩混和动态范围处理中的一些应用。

而另一些论题概念突显出来，则是由于他们不仅具有深奥的理论意义，同时还直接用于实践应用中。在这一类型中有通过延时处理实现的一系列效果：从相移到梳状滤波、空间定位、复制和混响等；另外我们还要讨论在滤波处理中带有大量分支的卷积、时间和空间处理以及调制。

在计算机音乐形成的初期阶段所涉及的一些合成技术的概念现在也被不断地纳入到信号处理的范畴中了。例如一些声音可以通过各种效果和空间处理得到增强。不断涌现出的“合成(synthesis)”技术实际上就是“分析、转换、重新合成”的综合技术，这需要对第四部分阐述的声音分析进行了解。

总之，通过对信号处理的解密，我们希望帮助那些自身不断提高的音乐家减少他们对技术助理的依赖。同时对基础知识的了解也会引导那些仍在学习的音乐人跳出迷惘的阶段，而步入到能够更直接的对他们的音乐素材进行控制处理的阶段。

第三部分的结构(Organization of Part III)

第 9 章描述了许多关于缩混和多轨录音的方法。现在的作曲家和缩混工程师已经不断地接管了原来由指挥完成的音乐工作。而作为一个指挥则必须要清楚地了解在一个音乐片段中如何平衡各个声部的强弱，调整在不同阶段需要突出和减弱的音乐角色。在第 9 章的最后部分介绍了一些媒体之间的同步问题，包括音频连接、视频、灯光以及其他效果之间的联系。同步问题对于那些积极的音频工程师来说是非常必要的，因为在现在的专业演播室中这些工作是必不可少的。

第 10 章和第 11 章介绍了大量的通过信号处理实现声音转换的技术。这两章主要是针对那些对信号处理领域没有任何了解的音乐家，当然第 1 章中所介绍的数字音频基础概念除外。我们的主要目的就是对那些厂商闭口不谈的和看似神秘的、缺乏解释的概念进行解密。我们要告诉读者如何使用信号处理工具对音乐进行“处理(handles)”。在第 10 章和第 11 章的最后，对许多的概念和术语进行了解释，并对他们之间的联系进行了说明，并强调了声音转换中最大的可能性。

第 10 章被分为几个部分：第一部分介绍了动态范围处理器所涉及的基本概念，其中以限制器和压缩器为例；接下来的部分介绍了简单数字滤波器的本质，解释了卷积对音乐处理的意义；最后一部分分析了时间延时效果(固定和可变的)和时间/音高的变化算法。

声音的空间定位从电子音乐出现以来就是一个非常重要的合成因素。第

11 章研究了诸如混响、空间定位、移动声源和声场空间模型的基本概念。

作为一个专业方向，数字信号处理需要对应用数学（微积分、线性系统理论、概率论、线性代数）进行掌握。在第三部分我们尽可能地使用更少的公式而采用更多的音乐观点。虽然我们给出了一些简单的等式，但是在一些地方我们也尽量避免使用数学符号。另外我们也将相关的工程著作的引用标记在整个部分中，以方便对技术层面感兴趣的读者参考。

第 9 章 声音缩混(Sound Mixing)

在空气中传播的声音可以产生自然的混合效果，譬如在交响音乐会上“融合”的交响声，或是在城市街头巷尾中各种声音的混合。模拟电子线路也可对声音信号进行混合，这些信号以时变电压的形式表现。通过电路加法运算可以将多个信号合成为一个信号。

在数字领域，音频信号的混合通过简单的加法就可以实现。为了进一步了解这个过程，通过图 9.1、9.2 和 9.3 三个不同时间轴表示的混合示意图来进一步说明。在图 9.1 中，信号源(a)在 t_1 时刻的采样值为 32 767，它与来自信号源(b)、其 t_1 采样值为－32 767 的采样相加，采样值相加的结果为 0[见(c)所示]。当两个正极信号 10 000 都在 t_2 时相加，其相加的结果为 20 000。

在图 9.2 中，显示了一个高频信号和一个低频信号波形电平的混合结果。

在最后的图 9.3 中，显示了两个持续时间大约为 2.5 秒的不同声音文件的混合。

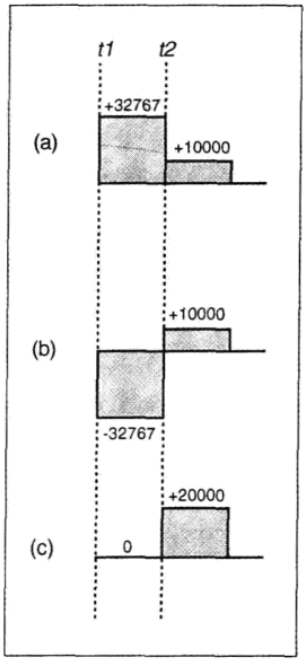


图 9.1 信号(a)和信号(b)两个采样值在 t_1 和 t_2 时间点的合成，结果如(c)所示。

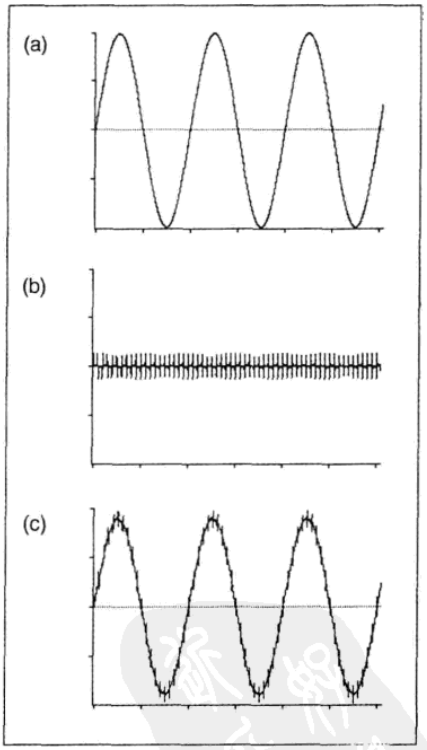


图 9.2 波形混合示意图。(a)50Hz 正弦波信号；(b)500Hz 正弦波信号；(c)为(a)＋(b)混合后的结果。

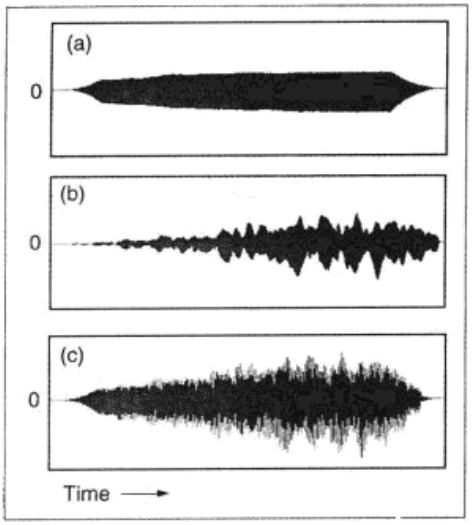


图 9.3　声音文件的混合。(a)中音萨克斯声音文件;(b)粒式合成声音文件;(c)为(a)+(b)合成后的文件。
Time= 时间

缩混和动态范围(Mixing and Dynamic Range)

动态范围(DR)描述的是一个系统所能处理的最小与最大的声音的范围,用分贝(dB)来表示。譬如,人耳的动态范围大约是 120 分贝。正如第 1 章所介绍,数字音频系统的动态范围是由量化比特数来决定的,每个比特大约对应 6 分贝。因此,一个 16 比特的系统动态范围大约是 96 分贝,而一个 20 比特的数字系统所能够处理的动态范围与人耳听觉系统的动态范围大致相当。

由于缩混将许多取样值相加产生混合值,因此在数字缩混系统中,动态范围的限制就带来了一些问题。如果相加的混合值超出了量化范围,结果将导致非线性或是错误。当几百个取样值都超出了量化范围的话,将会导致数模转换器中的"数字削波"或过载,从而导致音频信号产生强烈的"喀喀"声。

许多数字调音台进行加法处理的信号通道都提供了从 24 比特到 64 比特的量化精度,这么多的比特数可以满足调音台对 16 路或更多通道同时进行结合的要求将 16 个 16 比特的有效值相加会得到一个 20 比特的值。采用高量化精度的另一个原因是许多数字滤波器的处理至少需要 24 比特(144dB 的动态范围)的量化精度,才能保证得到高质量的音频信号。在调音台输出上,各种舍入模式能够降低取样的量化比特数。

在调音台(或是在缩混程序)内部,音频采样使用整数表示的称为定点

(fixed-point)表示法,与此对应,称为浮点(floating-point)表示法,其中采样由两部分数值组成:尾数(mantissa)和指数(exponent)。其中指数部分代表一个比例因数,可以表示一个很大或是很小的数值,提高了动态范围。因此,系统设计者可以采用浮点表示法设计方案,来避免在数字系统中动态范围过大或过小而出现问题。关于定点表示法与浮点表示法系统的更多细节请参看第 20 章。

非实时软件缩混(Non-real-time Software Mixing)

数字音频缩混可以通过硬件或软件进行。软件系统调音台的缩混工作主要有两个步骤:首先是音乐家安排计划,然后通过软件来执行。一般来说,我们称这种方式为非实时缩混,或更准确地称做无实时控制的缩混。

尽管没有实时控制,软件程序缩混仍比某些硬件调音台有一定优势。第一,软件缩混可以执行更准确、更复杂的运算,而对于某些硬件调音台来说是无法实现的。例如,输入通道个数超过自动化调音台的输入能力等情况。而一个设计优秀的软件调音台,即使同时处理一百甚至更多个声音文件时,还可在更短时间内应用更精细的信号处理;第二,软件调音台可以在软件环境(如一个声音文件在另一个程序窗口中,也可在本软件进行处理)中集成其他各种工具;第三,软件调音台是一种开放的系统,可通过程序语言来任意扩展其功能。

软件调音台主要可分为三大类:通过脚本语言缩混、通过面向对象软件缩混、图形缩混,下面一一介绍。

脚本语言缩混(Mixing by Script)

在第 3 章和第 17 章中描述的 MUSIC N 语言是一种合成软件,它包括把声音信号结合在一起的一些工具。声音素材首先转化为声音文件,然后通过软件打开进行精确的编辑。同时对独立的音频文件还可通过包络来进行电平调整,MUSIC N 语言的所有信号处理工具都具有附加的转换功能。其缩混的结果是一个典型的声音文件。

这种方法的一些例证包括由普林斯顿大学保罗·兰斯基(Paul Lansky)开发的 MIX 和 Cmix 系统。其中 MIX 系统是一种灵活的、非实时的缩混系统,可将磁带上最多 20 轨 32 比特浮点音频缩混为磁盘上的双声道声音文件。这个文件可以重录到另一个磁带上,或在需要的时候重新缩混。

Cmix 系统比 MIX 系统开发得要晚一些,同时 Cmix 系统用于 UNIX 兼容的计算机平台上。它是一个绑定在 C 语言上的信号处理程序的脚本库,通过预

置的缩混脚本与 Cmix 系统进行交互操作。最基本的就是通过这些脚本设定缩混中开始、停止的时间，以及使用什么样的振幅曲线。信号处理程序能够为用户提供对声音文件的滤波、混响和空间处理。用户还可编写自己的声音文件处理程序，并通过预先定义的程序来对声音文件进行混合处理。

在 Cmix 系统的程序处理方法，使音乐家可以对独立的音符、乐句、乐段或声部进行计算，并分步骤地把这些预混素材结合在一起。这种操作方法通过对组合素材的预演与独立素材的预演对比，实现整体素材更加有效的预演效果。

面向对象软件缩混(Object-oriented Mixing)

另一种用于缩混处理的软件是由李·博因顿(Lee Boynton)编写，用在 NeXT 计算机系统中的 Sound KIT。Sound KIT 是通过 C 语言编写的面向对象例程库(译者注：例程，即具有通用性的，被别的程序调用的程序或序列)，用来对音频文件进行处理。Sound KIT 中设计的一系列指令可以对声音文件进行分割、剪切、粘贴、删除、存取独立声音采样、组合声音文件，并且能对声音文件的任意部分进行重放(Jaff and Boynton 1989；Lansky 1990c)。

图形缩混(Graphical Mixing)

许多图形缩混软件程序都可运行在价格低廉的个人计算机上。其中一个经典的例子就是 MacMix 程序，MacMix 程序最初由阿德里安·弗里德(Freed and Goldstein 1988)开发，后来进一步发展并用在了基于苹果计算机平台的 Studer Dyaxis 音频工作站上。

MacMix 软件如图 9.4 所示，进行缩混的独立声轨以水平条的形式排列，用户可选择水平条并在时间线上水平移动，来调整水平条相对声轨的时间位置，而这种操作对于传统的线性媒体如多轨录音机是无法实现的。每一声轨都可在屏幕上通过调整包络线控制音频块的淡入淡出，在右边的柱状条则可以对声轨的增益进行调整。

按照 MacMix 程序思路，许多公司也开发了他们自己的与 MacMix 控制相似的多声道图形缩混软件。如图 9.5 所示的由数字设计(Digidesign)公司和 Opcode 公司开发的带有 MIDI 序列数据处理的多音轨音频缩混系统。

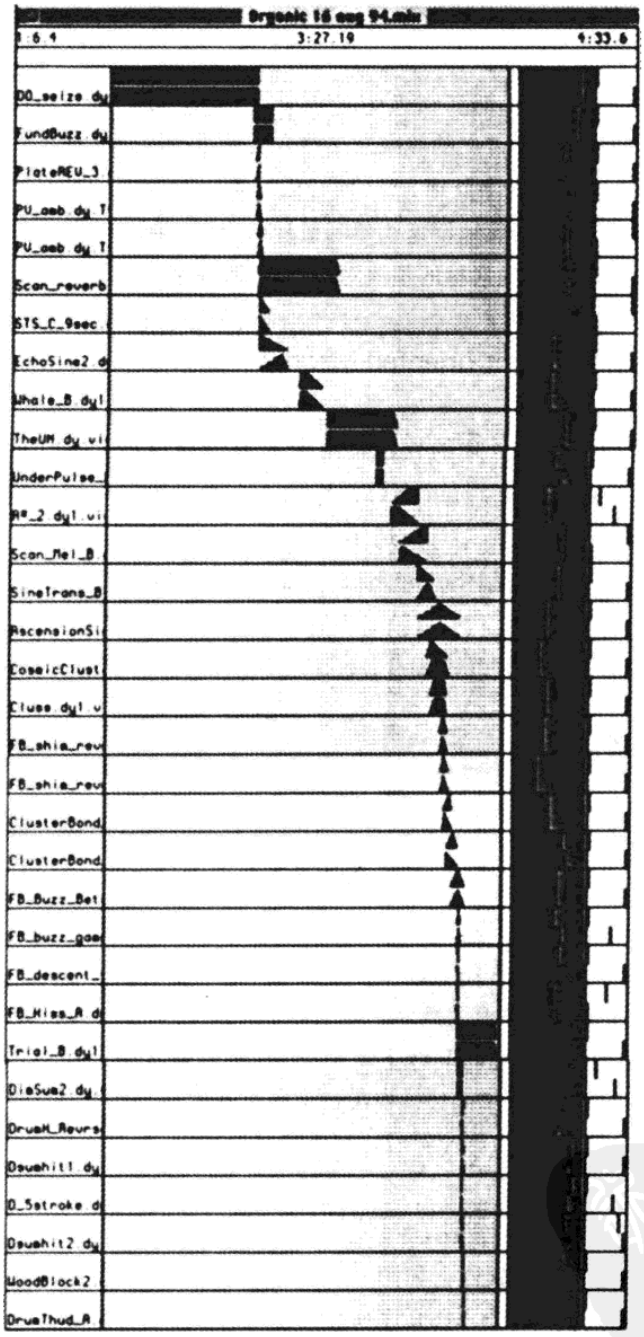


图 9.4 在 MacMix 程序中进行多轨合成的截图。其中阴影栅条代表有 33 轨的声音进行合成，每一个立体声轨对应一个在磁盘上存储的声音文件。在声轨前后的斜度变化表示声轨的淡入淡出。右边黑色的水平条表示每一轨的相对电平幅度大小，在最右边的垂直线代表着每一个音轨出现的空间位置。截图来自作者的合成作品《Clang-tint》(1994)。

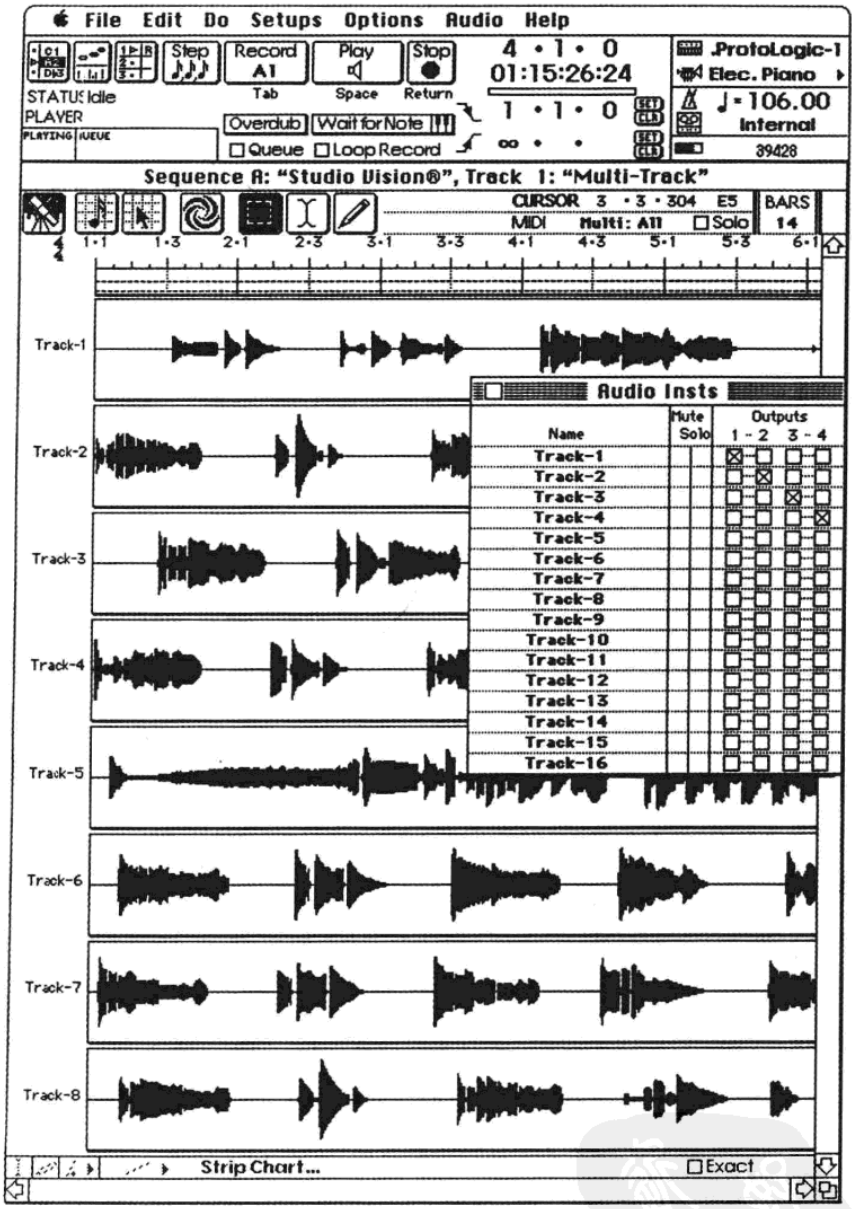


图 9.5　Opcode 的 Studio Vision 程序的屏幕截图，显示出了在 MIDI 音序编辑器中的音频声轨。其中 MIDI 序列数据并没有显示出，而是通过在音频第一轨上方的 4/4 标记和节拍数来表示，同时还包括一些音序器控制工具。

软件缩混的评价(Assessment of Software Mixing)

软件缩混可以运行在通用计算机平台上,具有良好的适应性和精确性。软件处理应用在前期准备过程中可以给工程师更多的时间来进行精确的缩混处理。在可编程的缩混环境中,可对任意数目的声道进行组合或是缩混操作,远远超过了人们在缩混调音台上(甚至是一台具有自动化控制的调音台)所能达到的工作能力。

软件缩混在具有适应性和精确性之外,也有其自身的不足。即便在简单缩混中,精心的制作安排也是非常必要的。在非实时的缩混过程中,对演示片段进行精确的时间上和包络线的调整,是软件缩混的优势。非实时软件缩混中一个主要的不足就是缺少"感觉"。由于没有实时推子进行操作,因此不能与听感之间产生直观对应。这些基于计算机的调音台通过鼠标和键盘进行操作来控制通道的幅度,在这种要求精确的处理过程中感觉上无法与实际的推子控制相同,因此具备一套精确的推子是更可取的方案。所以,一个具有精确推子的硬件实时缩混调音台(模拟或数字)与软件缩混相结合使用将会是最佳的解决方案。

调音台(Mixing Consoles)

一般来说,调音台(也被称为 mixing desk 或 mixer)是将一定数目的输入通道进行组合并实时的送入到一定数目的输出通道中的设备,同时还可以实现如滤波器或信号分配等辅助功能。多年来,调音台都是通过专用模拟电路实现的,并且已经具有非常高的声频标准。随后,数字技术逐渐引入到模拟调音台中,形成了数控模拟调音台,它是将数字化控制和自动化电路加入到模拟音频电路中实现的,关于数控模拟调音台将在后面进行介绍。现在全数字调音台的使用率也在快速地增长。

一般来说,调音台的功能不仅仅是一个功能强大的音频信号加法器,同时也作为演播室或现场制作中的控制中心。因此在调音台上集成了专门的控制部分和一些相应的功能。当然一些特殊的控制要求及性能要求取决于设备生产商和客户的需要。因此,我们在这一章中仅对基本原理进行介绍,而不对某些调音台上特殊功能进行说明。

调音台的属性(Properties of Mixers)

调音台可以根据所能处理的输入通道数目及其所能产生的输出通道的个数来进行标识。比如说,一个调音台可以同时处理 8 路输入信号,并能将它们缩混成 2 路信号输出,我们就把它称为 8/2 调音台("8 到 2")。另外,许多调音台还具有一定数量的用于信号分配的输出母线。再比如说,一个调音台有 8 路输入,4 路输出母线,2 路主输出母线,我们称为 8/4/2 调音台。在这样的一个调音台上通过 4 路输出母线和 2 路主输出母线,可以同时实现独立的 4 路录音和 2 路录音。

一个典型的录音演播室调音台包括 6 个主要的模块部分:输入模块、输出模块、辅助返送模块、对讲模块、监听模块和仪表模块。如图 9.6 所示为一个 8/4/2 调音台示意图,其中 8 个输入通道中的信号可以通过输出母线和主输出设置按钮(包括 LR、1/2、3/4)和声象旋钮(pan pot)分配到 6 个输出通道中的一个或多个通道中去。按下其中的一个输出母线和主输出设置按钮可以将输入信号送入到相应的两个输出母线或主输出上去,然后再通过声象旋钮的左右旋转选择其中的某一个输出通道。同时输入也可以分配到 2 路辅助输出(AUX)母线上,将信号输出到外部的声音信号处理器中进行处理;外部效果器的信号可以被返送到输出母线或主输出上,通过调整输出电平推子上方的辅助返送旋钮(RET)来进行控制。另外效果器的输出也可以再次送入输入通道中进行进一步的处理和控制。在调音台右上方的 CM 和 SM 旋钮用来对监听扬声器的电平幅度进行控制,其中 CM 为控制室,SM 为演播室,它们都从 L/R 主输出上获得信号。在右边的对讲传声器可以实现在控制室的录音师与演播室的演员之间的沟通和标记。仪表模块显示的是 8 个输入通道和 6 个输出通道的电平幅度的大小指示。下面将对调音台的模块部分进行详细的阐述。

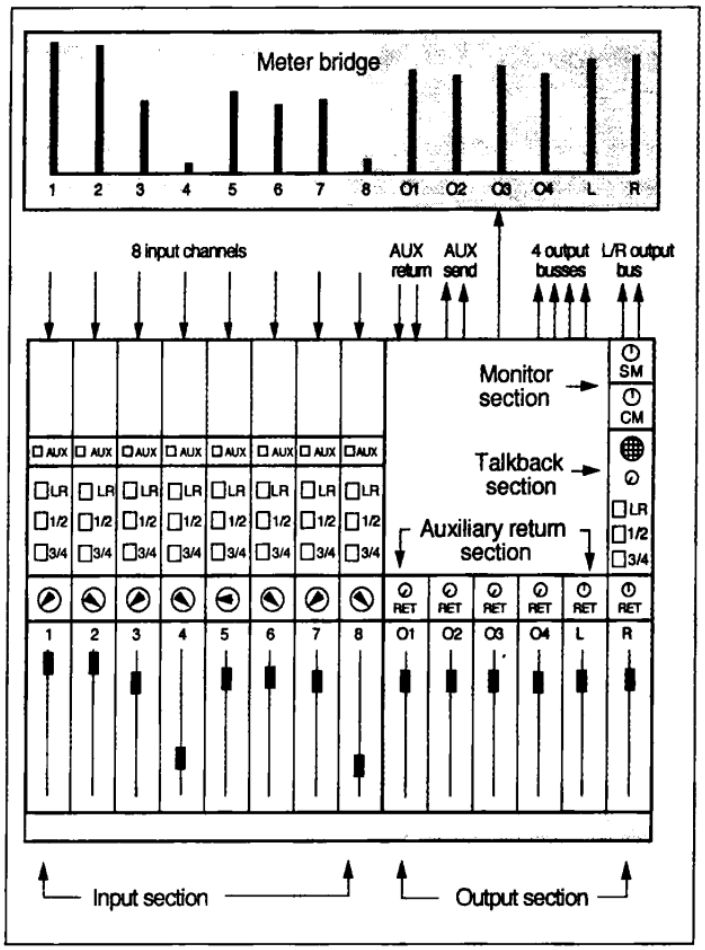


图 9.6 一台简单的 8/4/2 调音台的信号流向示意图，同时也显示出了调音台的不同模块。其中方块代表着按压式开关，圆圈代表着旋钮调节器，O1 到 O4 代表输出母线，L 和 R 对应左右输出，监听模块中的 CM 和 SM 用来控制监听控制室和演播室中的监听电平大小。
Meter bridge=表桥 Input channel=输入通道 AUX return=辅助返回 AUX send=辅助送出 Output busses=输出母线 L/R output bus=左/右输出母线 Monitor section=监听模块
Talkback section=对讲模块 Auxiliary return section=辅助返回模块 Input section=输入模块
Output section=输出模块

输入模块(Input Section)

输入模块一般是由一定数量、相同的输入单元组成，如图 9.7 所示。表 9.1 对输入模块的每个部分进行了说明。

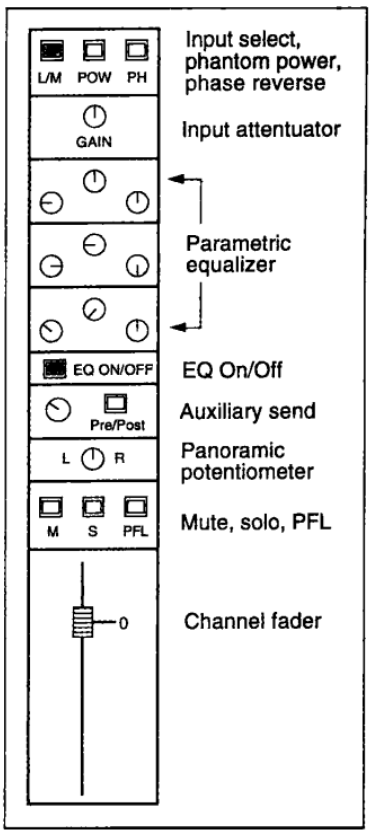


图 9.7　调音台输入模块示意。表 9.1 详细解释其中的每一个部分。

表 9.1　调音台输入模块功能

输入选择、幻象供电、反相开关	选择开关。输入选择包括传声器输入、线路输入或子编组输入。幻象供电开关可以为电容传声器提供工作所需的直流电信号(DC)。反相开关可以将输入信号进行相位反转处理(主要用于多传声器设置中)。
输入衰减/pad	输入衰减旋钮可以避免输入通道中的输入信号过载的情况。对于线路输入来说，输入衰减旋钮可以调节调音台与外部输入设备之间电平的匹配(如磁带录音机或乐器等)。
参数均衡器	通过对某一频带内频率成分的提升或衰减，来改变输入信号的频谱结构。图示为一个三段参数均衡器。每个频带内的三个控制分别为带宽调整，中心频率调整和提升及衰减。一个半参数均衡器省略了频带控制调整。
EQ 开/关	均衡电路是否对输入信号进行处理。

续表

辅助输出	将信号输出到外部的效果器(如延时器或混响器)或是提示输出(cue output)。提示输出常用于演播室中演员的耳机监听或是舞台上的返送监听。因此提示输出可以将平衡后的音乐信号预缩混输出,使得每个演奏家都可以感受到他的乐器声音信号在总的音乐信号中的比例及强度。Send 旋钮控制用于输出到外部处理设备或参考输出的信号的输出电平大小。Return 旋钮控制从外部处理设备返送回来的信号电平的大小(参见效果/辅助返送部分)。如果 pre/post 选择为 post(post fader),表示输出到外部设备信号的控制为推子后控制,意味着推子电平的改变影响输出信号的大小;如果选择 pre(pre fader),表示输出到外部设备的信号不受推子的控制,意味着推子电平的改变不影响输出信号的大小,甚至是推子在关闭情况下,信号仍然输出到辅助母线上去。
声象旋钮(pan pot)	控制在两个或两个以上的声道中声音的空间定位。
哑音(Mute)、独奏(Solo)、预听(PFL)	哑音键可以关闭此通道声音,不影响其他通道声音。独奏键只打开此通道声音,关闭其他通道声音,当独奏键打开时,除此通道外的其他通道全部处于哑音状态。当需要对某一通道的声音进行单独监听的同时而又不影响到其他通道的输出时,使用预听(pre-fader listen)来实现。如广播电台的工程师可以通过按下 PFL 键后,利用耳机来监听录音开始的部分,由于此通道的推子是关闭的,则信号不会被送出,听众是无法听到的。因此,通过 PFL 键,我们可以实现某一通道的声音在不输出的情况下进行电平设置或均衡处理的目的。
通道分配设置(没有示出)	通道分配设置一般由一组按钮组成,每一个输出母线或主输出对应一个按钮。它的功能是将输入信号分配到需要的输出母线或主输出中去的。
通道推子(或旋钮)	一个线性的推子或一个可旋转的旋钮用来对通道中声音信号的幅度(或增益)进行调整。

输出模块(Output Section)

调音台输出模块的控制相对来说比较简单。它主要就是通过推子来控制送入到输出母线上信号电平的大小,同时通过仪表显示出来。

辅助返送模块(Auxiliary Return Section)

辅助返送模块也被称为效果、提示或返送模块。所有这四种叫法使用在不同场合的录音工艺中。混音师可以通过辅助返送模块将经过效果器处理的信号混入到输出信号中去，另外他们还可以通过辅助返送模块产生一些特定的监听信号，通过耳机送到演播室中的演员或是送到舞台上的返送监听扬声器中。

对讲模块(Talkback Section)

对讲模块的主要作用就是实现控制室中的录音师与演播室中的演员之间的沟通交流。对讲模块的另一个作用是在计算机音乐录音演播室中可以产生一些“提示”声或“打板”声，以备于后期工作中的参考。从技术上来说，对讲模块包括一个传声器、一个电平控制，以及一些控制开关用于决定录音工程师语音的分配。

监听模块(Monitor Section)

监听模块的功能就是将左右输出母线上的信号进行分配，送入到控制室(调音台所在的房间)和演播室(演员所在的房间)的扬声器或耳机中。

仪表模块(Metering Section)

仪表模块上的仪表用来对输入通道信号和输出母线或主输出信号的电平进行指示。其中峰值表(Peak meter)反映的是信号的瞬时峰值幅度，它的上升时间(响应时间，指的是达到信号幅度的 99%所需的时间)一般只有几毫秒，延时时间(复位时间)大约一秒以上(请注意，不同的峰值表可以具有不同的参数指标)。VU(Volume Unit)表则有相对较长的上升时间，一般来说 300 毫秒左右，因此更适合表示短时间内信号的平均幅度值(类似于响度)。也有一些仪表组合了峰值表和 VU 表的特点。

编组功能模块(Grouping Section)

一些调音台具有次编组功能，录音工程师可以将一些输入通道的控制指定到一个单独的推子上，这个推子被称为预混或次编组推子。这样的话，当调整次编组推子时，所有被指定的输入通道的推子也同时做相应的动作。

数控模拟调音台(Hybrid Consoles)

在调音台上最早应用数字硬件技术是在 20 世纪 70 年代。第一个数字技术的受益就是推子自动化——重新调用开关旋钮的设置和推子的位置用以重建特殊的缩混过程(参见后面的自动化缩混部分)。现在一些调音台混合了数字和模拟技术,将数字化的自动控制功能与宽频带的模拟信号处理技术相结合(如图 9.8 所示)。在模拟电路中频率响应可以达到 100kHz,远远超过了数字调音台工作设计的 44.1kHz 和 48kHz 的标准采样频率。

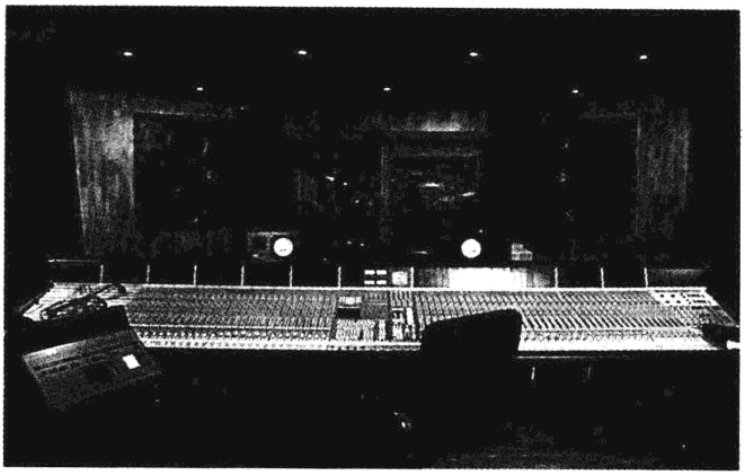


图 9.8 一个大型的数控模拟调音台(由 Solid State Logic 生产),可通过调音台内置的计算机对模拟信号处理进行控制。自动化对于操作这样大型的调音台是必不可少的。[图片来源于卡普里(Capri)数字工作室,卡普里]

数控模拟调音台的自动化系统允许录音工程师保存缩混控制数据,并能在以后从存储器中重新取出保存的数据,用于重建缩混过程。缩混数据的存储是通过将推子的位置(相应的模拟电平)通过模数转换器进行采样编码后存储在调音台的计算机中实现的。

缩混重放(Playing Back the Mix)

有两种方法用于调音台上缩混数据的恢复和重建。一种是每个通道的数字缩混数据被送入数模转换器中,将数据转换为模拟信号形式,用以控制压控放大器(VCA)的电平,如图 9.9 所示,达到重建的目的。

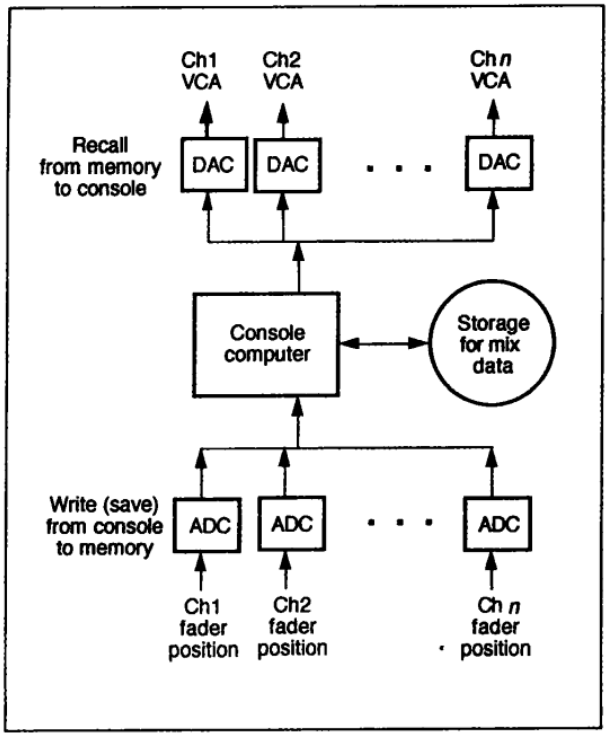


图 9.9　基于压控放大器的数控模拟调音台缩混数据的记录和重建示意图。
Ch1,2…*n* VCA=通道压控放大器　　DAC=数模转换器　　ADC=模数转换器
Recall from memory to console=从存储器将记忆设置调出送到调音台
Console computer=调音台计算机　　Storage for mix data=用于混音数据存储
Write(save) from console to memory=将调音台状态写入存储器中
Ch1,2 ...*n* fader postion=通道推子位置

由于采用压控放大器并不会提高音频质量，所以一些厂家开始使用小型电机马达来实现自动化控制。也就是说，通过 DAC(数模转换器)直接控制电机马达，根据录音工程师前面对推子所作的动态调整记录数据，对推子进行动态控制(意味着控制了通道的电平)。这种电机马达可以达到很高的精度，可以在 100 毫秒之内实现从最高到最低 4 096 个 0.1dB 步进的控制，同时避免了压控放大器在音频处理过程中的干扰。采用电机马达的另一个好处是录音工程师可以根据推子的运动看到上一次录音师对推子电平的动态控制，另外在运动控制的同时，录音师还可以根据自己的需要手动的对运动的推子进行控制，如按住正在运动的推子，来取代此时刻计算机的控制。

数字调音台的特性(Features of Digital Mixing Consoles)

数字调音台有着模拟调音台和数控模拟调音台所不具备的功能和特点。以下为数字调音台的一些特性(不一定适用于每一台数字调音台):

1. 数字信号处理全部在数字域中完成,因此避免了多次采用 DAC(数模转换器)和 ADC(模数转换器)而产生的声音损失。

2. 控制面板可以被重新设置,以采用更少的旋钮进行控制,替代每个电路部分都有一个旋钮来控制(大型的调音台有超过 4 000 个旋钮、按键及滑动开关等),可设置的旋钮可以在不同的时刻实现不同的处理功能。在任何一个通道中都可以调用集中控制工具(如图 9.10 所示),如可以指定一个推子来对任意数量的输入通道进行控制。

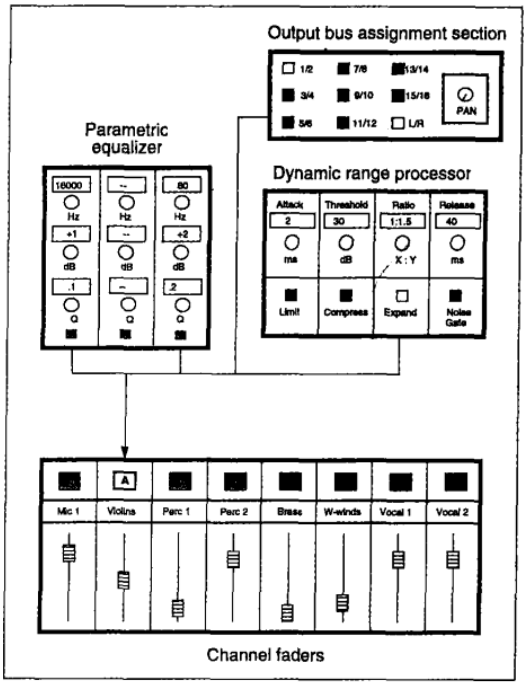


图 9.10 在可设定调音台中,每一个输入通道都有一个独立的推子控制,但是用于均衡、动态、输出母线控制等功能部分只有一套控制,对每个通道的这些控制都可以通过打开推子上方的设定按钮(标记为 A)来进行独立控制,这时相应的控制旋钮按键仅对选中的通道进行控制。在此图中,通道 2 采用了 2 路参数均衡控制,一个动态范围扩张器控制,并被分配到几路不同的输出母线上。通过对旋钮的旋转来对指定的通道进行控制。

Output bus assignment section=输出母线设置模块　　Dynamic range processor=动态范围处理器
Parametric equalizer=参数均衡器　　Channel fader=通道推子

3. 控制面板可以与缩混硬件独立开来，因此数字控制面板所占用的空间与模拟调音台相比可以小得多。

4. 数字效果如延时、混响及动态处理等功能都可以集成在数字调音台内部实现。

5. 随着更多设备的“数字化”，其他的一些数字技术如推子自动化、自动化信号分配、图形显示、硬拷贝打印(hardcopy printing)、网络连接及计算机接口等都可以被更加容易的集成在调音台系统中。

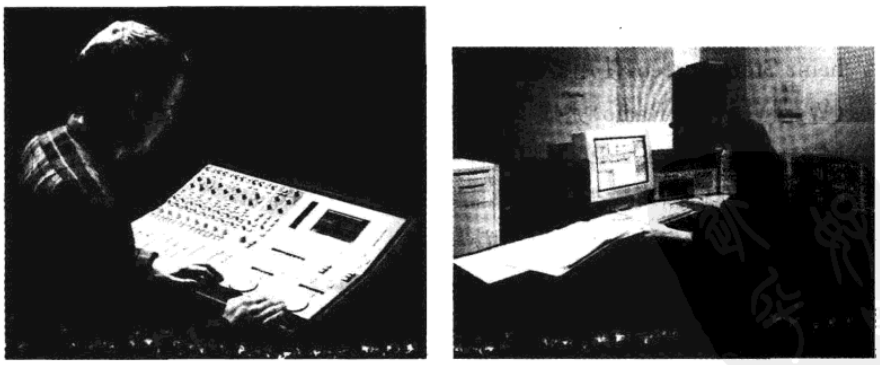
6. 由于调音台系统是基于软件平台的(如使用程序或微代码控制硬件)，所以随着软件系统的升级就可以提供更丰富的功能和新特性，从而提高整个系统的水平。

7. 如果缩混硬件具有一定的灵活性，那么通过软件的重新设置就可以实现不同数量的输入输出通道、均衡器，等等，来适应各种不同需要的缩混工作。一个演播室应该具有一系列不同配置的“排秩”用于各种类型的演播室缩混工作。

8. 软件系统中的诊断程序可以分析和显示错误代码，相应的错误日志可以帮助工程师的后期维护工作。

独立调音台与音频工作站(Stand-alone Mixers versus Audio Workstations)

在第20章我们将对固定功能(fixed-function)和可变功能(variable-function)硬件结构进行区分。在数字调音台系统中，两者的内部区别主要体现在外部程序的使用和操作上。固定功能系统被设计在独立调音台系统中，这种调音台可以完成已经设定好的特定的音频功能和操作效果[如图9.11(a)所示]。



(a) (b)

图9.11 独立调音台与通用工作站。(a)独立调音台，型号SSL01，用于CD母带制作。控制面板左边看上去像一个模拟调音台；(b)一台基于标准的个人计算机(左边为Apple Quadra计算机)的多轨音频工作站(Studer Dyaxis Ⅱ)，可以运行大量的软件包。其他的数字录音缩混设备包括一个8轨数字磁带录音机(在图片右边)，在它上面是一台CD刻录机，在右边是两台专业DAT。(图片来源于Cornelia Coyler, Center for Computer Music and Music Technology, Kunitachi College of Music, Tokyo。)

可变功能系统常被用于基于计算机系统的音频工作站中[如图 9.11(b)所示],因此可以运行大量的软件包,同时可以与相同构架计算机进行连接。通过板卡或外置接口箱,带有推子的录音缩混硬件设备可以与计算机相连接,从而取代独立调音台实现实时控制音频参数。拥有大量的软件包的好处是不言而喻的,当然如果他们不能协调工作除外。

多轨录音和缩混(Multitrack Recording and Remixing)

早期的录音都是单声道的——录制在一个声道中。相应的音频后期制作也是单声道的方式,并通过一个扬声器进行重放。立体声录音方式始于 20 世纪 30 年代(Blumlein 1933,Keller 1981),那时候的录音主要都基于一个或两个声道的录音。立体声录音再现了现场的效果,同时各个声源的相对平衡感也在录音的那一刻就进行了确定。

相反,多轨录音机包括了多个独立的通道或音轨,每一个音轨都可以在不同的时间进行录音。在下面的部分中我们将对多轨录音的历史进行介绍,描述多轨录音的优点,以及它在缩混中表现出的一些问题。

多轨录音:背景(Multitrack Recording: Background)

20 世纪 50 年代,电吉他演奏家 Les Paul 在与位于加利福尼亚的安培(Ampex)公司合作的过程中,首先提出了使用多轨技术进行配音复制的概念。在 1960 年,多轨磁带录音机投放市场,同年,卡尔海茵茨·施托克豪森(Karlheinz Stockhausen)使用德律风根(Telefunken)T9 四轨录音机在西德广播电台(West German Radio)完成了他的电子音乐作品 *Kontakte*(Stockhausen 1968, Morawska-Büngeler 1988)的录音工作。到 1964 年,在瑞士的 Studer 公司生产了他们第一台四轨磁带录音机,制作人乔治·马丁(George Martin)使用此设备完成了甲壳虫(Beatles)著名的专辑 *Sgt. pepper* 的制作。

在第 1 章我们介绍了数字多轨录音的历史。现在专业的数字多轨磁带录音机能够实现 48 甚至更多轨的录制,当需要更多轨数的时候,还可以将几台多轨录音机连在一起同步使用。当然现在的专业多轨录音机还是比较昂贵的,在小型的录音棚采用相对廉价的基于录像带的录音机或硬盘工作站是更为可取和现实的。

多轨录音的优势(Advantages of Multitrack Recording)

多轨录音媒介为一些录音实践提供了一定的灵活性。首先,录音工程师可以将不同的声源安排在不同的声轨上,这样就避免了在录音时对所有声道的平衡做大量的工作,而是将这些工作安排在后期的缩混制作过程中,进行更准确的处理和调整。

对于合成音乐,多轨方式进行录音和分层声轨方式也具有很强的吸引力。数字录音可以实现并轨(在一台设备上同时将几个不同的音轨合成为一个音轨)和多代无损的复制(相比而言,在模拟介质上一个严重的问题就是在复制时噪声的累加)。

一些系统还提供了数字混录(sound-on-sound)的功能。在这种录音过程中,一个新的声音信号(比如一个两声道的信号)可以简单地叠加在一个原有的信号上,从而形成另一个新的两声道信号。通过对原始信号和新信号平衡度的仔细调整,建立合成结构,或是对声音接点一步一步地进行精确的加工。

多轨缩混中出现的问题(Problems Posed by Multitrack Remixing)

虽然多轨录音具有良好的灵活性,但是它也不是一剂万能药。为了利用多声道的独立性,每个声道的声音,都必须与在其他声道上同时进行录制的声音相隔离。为了达到这个要求,就需要录音工程师采用独立的录音小室、挡板及有指向性的传声器,同时还要减小传声器的拾音范围以达到录制声音的最大独立性。当然一些电子乐器不需要使用传声器,它们的信号可以直接送入到独立的声轨中。

当这些各自独立的声音同时重放时,结果往往是产生非常不自然的声场效果。尤其是通过耳机监听时,每个声轨的声音听上去就好像是一个人的耳朵里塞满了各种不同的乐器。对于某些音乐来说,我们的目的是建立一个合成的声场(如大部分的流行音乐和电子音乐),这个问题并不突出。为了对多轨的声音进行统一融合,录音工程师通过增加混响等类似的效果,仔细地调整平衡关系和空间方位感来达到立体声的整体感觉。如果我们不是很在意这种人工"统一"这些不同的音乐素材的话,我们倒是可以通过对每个独立声轨应用空间效果后,创作出奇幻的、非自然的人工空间。

但是如果我们的目标是重建类似听众在音乐厅中所感受到的空间感声象的音乐时,多轨处理就不可取了,尤其是对众多的声学音乐节目(如交响乐、合唱、独奏、声乐等形式)。作为多轨录音的反例,一些录音工程师开始重新回到

了使用较少的传声器和较少声道的“纯”录音方式中(Streicher and Dooley 1978)。这种“纯”录音方式需要工程师正确地安排音乐家和传声器之间的位置关系,从而在具有良好声学效果的演奏厅中得到最好的录音效果。这种采用传统录音技术的方式带给录音工程师更多的压力,因为在录音的同时也就必须完成缩混的工作了。

音频监听(Audio Monitoring)

在录音或缩混时,音频监听的环境是非常重要的。各种监听理论层出不穷,在此我们并不从理论上去争论每一种理论的优缺点,但是在此我们的选择主要强调的是个人的感觉和经费预算的问题。

耳机(Headphones)

对于现场录音来说,由于现场环境的限制,没有独立的听音房间,在监听时耳机则是唯一的选择。但并不是说,耳机最适合现场录音监听。即使通过质量好的耳机监听,也好像是在放大镜下观看声音。但是耳机确实是一种监听录音作品的最好方式,它可以用来监听一些录音过程中细微的缺陷,如断点、咔咔声、噪声、畸变和相位等问题。这在中等音量下的扬声器中是无法明显感知的。

扬声器监听(Loudspeaker Monitoring)

扬声器与房间是紧密相关的。在这一部分我们重点来讨论三种扬声器监听环境:近场、控制室和听音室。虽然只对比这三种环境,但并不意味着只有这三种监听环境,其实还存在着许多不同的环境。事实上,选择现在的监听环境很大程度上取决于音乐作品的风格。

近场监听(Near-field Monitoring)

在小型录音棚或演播室中,近场监听扬声器是比较常用的(如图 9.12a 所示)。当然在一些大型的演播室中也会使用近场监听扬声器,录音工程师通过这些近场扬声器来模拟类似于在家庭音响系统扬声器音乐重放时的听音感受。一般来说,近场监听扬声器安装在调音台上方与耳同高的位置,同时这些小型的动圈扬声器与录音工程师之间的距离应小于两米。这样的安装设计,可以保

证来自扬声器的声音响度远远超过其他方向的反射声的强度。另外，近场扬声器的尺寸也应该小一些，因为听声者距离扬声器比较近，扬声器可以辐射出一个相对“融合”的空间声象；而这种效果通过一个尺寸较大的多驱动单元的扬声器是无法实现的，因为大型扬声器的高音单元(tweeter)与低音单元(woofer)之间的距离可以达到一米或更远。

另外近场监听扬声器的一个比较大的问题是，由于扬声器的尺寸较小从而缺乏低频响应的声音辐射。在后期制作中使用近场监听扬声器会造成2到3个八度的声音损失和较强的衰减。

控制室监听(Control Room Monitoring)

另一种音频监听的应用来源于传统录音棚的设计，录音棚往往被分隔成为两个部分：演播室——演员进行表演的场所；控制室——录音工程师和调音台所在的空间。在控制室中，扬声器安装在经过声学处理的前面墙面上(如图9.11b所示)。每一个扬声器驱动单元由独立的功放进行驱动，根据扬声器系统的扬声器驱动单元个数来决定使用双路功放系统(biamplification)或是三路功放系统(triamplification)。整个监听系统(包括房间)经过均衡处理后在录音工程师头部“最佳区域(sweet spot)”位置应具有平直的频率响应。另外控制室监听应支持高声压级重放，比如一些大型的流行音乐演播室中的监听环境。

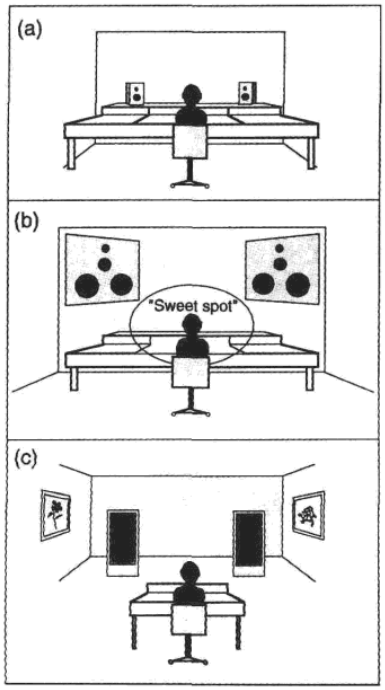


图9.12 图示为三种监听环境。(a)在近场监听环境中，小型的扬声器应摆放在距离听声者1到2米的范围内；(b)在控制室监听环境中，大型墙面固定安装的扬声器应距离听声者3到5米的范围内，并保证听声者位于调音台面前的最佳听音位置处；(c)在起居室监听环境中，大型的落地式扬声器应摆放在距离听声者2到5米的范围内。
Sweet spot=最佳听音区

听音室监听(Listening Room Monitoring)

听音室中的扬声器一般安装在地板或地板支架上,当然在一些特殊的声学环境中也可以根据需要安装在相应的位置,如图 9.12c 所示的典型的起居室。房间一般应经过一些声学处理,但是又不必像流行音乐演播室那样做严格的声学处理。扬声器相对较大,具有全频带平坦的频率响应和精确的空间声象定位的能力。可以采用三分频动圈扬声器(包括高音单元、中音单元、低音单元)或超指向性静电扬声器进行声音辐射。许多古典音乐录音师和制作人都比较倾向于采用中等音量条件下起居室监听方式。如图 9.13 所示为一个 CD 母盘制作监听环境的应用。在前面墙中心的矩形区域安装了一个不规则面的反射板,对声波进行反射,来衰减房间内的共振。

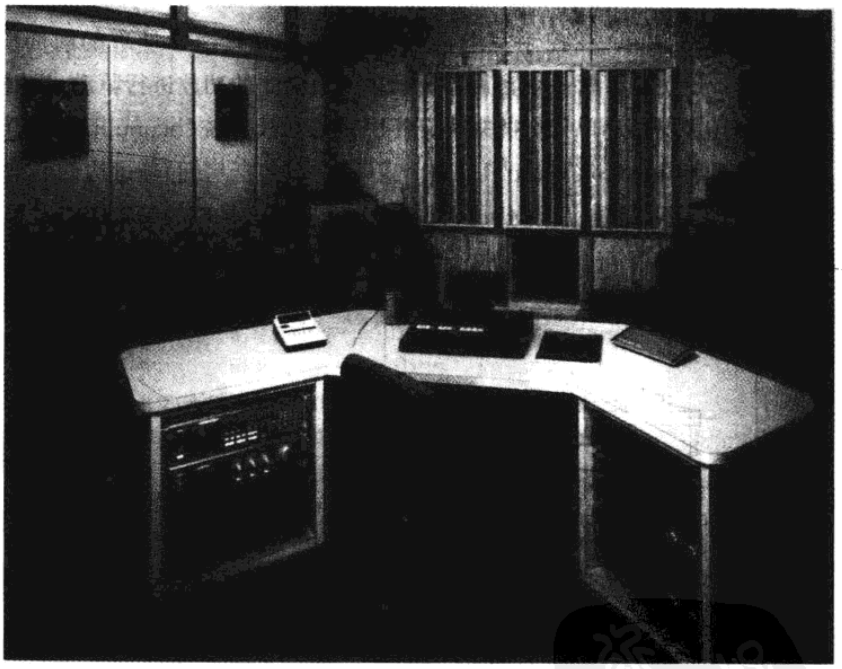


图 9.13 CD 母盘制作监听环境示意图。(照片来自 John Newton,位于 Soundmirror,Boston。)

演出时的缩混和监听(Mixing and Monitoring in Performance)

从评判的立场来说,听众听到了什么,以及通过房间中的扬声器重放后的音乐再现对于听众在哪一点聆听是最佳位置,这些问题都与扬声器的设置有关。虽然扬声器的设置问题仍然是一个开放的议题,但是,归根结底还应该由

使用它的艺术家来决定。另外一个问题是在电子音乐中再现声学乐器时，是将它们融合在一起还是将它们独立开来？关于这个问题可以参见作曲家 Morrill (1981b)关于这些问题的讨论，Morrill 创作了大量的电子音乐作品和计算机声音作品。也可以参看第 11 章，关于声音在空间中的辐射。

自动化缩混(Mix Automation)

多轨录音缩混是一个复杂的过程，一个人的能力往往不能胜任。直到出现自动化缩混之前，一个复杂的多轨缩混(如一个电影声轨的缩混)都必须要有至少 4 个人在一个调音台上来完成。而自动化缩混的最大好处就在于仅仅需要一个独立的录音工程师通过简单的步骤就可以完成一个复杂的缩混工作。例如，录音工程师从缩混两个立体声节目开始，节目分别位于音轨 1-2 和 3-4 上。调音台内置的自动化缩混系统可以实时地记录第一次缩混时的控制信息，这一步完成后，另一个位于音轨 5-6 的立体声节目就可以直接缩混到第一次的缩混节目中，这样依次下去，不断地将节目缩混到上一次的缩混节目中去，直到最终完成全部的缩混工作。

调音台上的自动化的概念比较宽泛。“自动化”可以指在调音台上按压一个按键时调用不同的调音台设置功能，也可以指推子自动化(通道的推子按记忆及时进行运动)，甚至是将在大型调音台上的一个处理过程中所有功能设置全部进行记录、存储并重新调用。

调音台上全功能的自动化系统在一秒钟内会对所有设置进行多次的扫描。在扫描的过程中，推子或按钮的当前位置都会与上一次扫描后存储的位置进行比较，如果位置发生了改变，将会产生一个触发数据，进行控制的识别和新位置的确定。在重放时，调音台计算机按照相同的速率将已存储的控制数据读出来，从而控制调音台，同时在这种状态下，录音工程师仍然可以随时手动调节推子或旋钮，进行修改。

音频调音台的 MIDI 控制：跳线和通道哑音 (MIDI Control of Audio Mixers: Patching and Channel Muting)

MIDI 全称为音乐乐器数字接口(Musical Instrument Digital Interface)。虽然它并不是为调音台的自动化而设计的，但是在 MIDI 1.0 的规范中一些内容却可以用来对调音台进行控制，尤其是那些小型演播室中的调音台。(见第 21 章关于

MIDI 更多的说明。)调音台上的一些控制功能可以通过 MIDI 指令来实现,比如说,MIDI 指令中的程序转换信息(program change messages)能够对调音台的输入/输出分配进行重新排秩安排,或者是在指定的时间里对某一通道进行哑音(关闭)控制。当然这些功能的实现是通过在调音台的内部安装一个廉价的微处理器实现的,微处理器对 MIDI 信息进行翻译后,得到相应的调音台所能够接受的指令,从而对调音台内部的开关旋钮进行控制实现所需要的处理要求。

在流行音乐的制作过程中,使用多轨磁带录音时,通道哑音技术是必不可少的。举个例子来说,假设录制一个鼓的声音,将鼓的声音同时录制在三个不同的声轨上,而且每个声轨上的鼓都施加了不同的效果处理,这时我们通过 MIDI 音序器中的动态模式,根据乐曲节奏点的位置,切换三个声轨的哑音控制,从而得到特殊效果的鼓声。另外,哑音控制还可以应用到演唱表演中,首先对演唱表演进行多次录制,将声音记录到不同的声轨上,然后通过哑音控制选取每一段声轨中最好的部分进行组合,得到较完美的录音作品。

在缩混中通过 MIDI 控制并不仅仅能够完成排秩和哑音控制,它还可以实现如连续的推子变化、均衡改变、空间声象定位及一些效果控制。当然在一些大型专业调音台上,由于 MIDI 数据率的限制,并不能通过 MIDI 信息来完成所有功能的动态控制(Cooper 1989,Rogers 1987,McGee 1990)。也就是说,MIDI 可以控制一些小型调音台的部分功能,但不能同时控制或连续进行控制。MIDI 可控的调音台需要进行相应的调整以适应 MIDI 信息的数据率,从而达到接受 MIDI 控制信息的范围。

调音台的自动化是通过专用功能计算机上的音序器来进行控制得以实现的。尽管还没有建立相关的调音台 MIDI 自动化控制信息的标准,但是已有三种基本方案得以应用。每一种方案都使用了不同种类的 MIDI 信息:程序转换、音符/力度和 MIDI 时间码。(见第 21 章对 MIDI 信息的解释。)

MIDI 控制衰减系统 MCA(MIDI controlled attenuator)是 MIDI 程序交换信息的典型示例应用(如图 9.14 所示)。对于小型录音棚来说采用 MCA 可以比较廉价地实现推子自动化的解决方案。系统(通常为 8 到 16 个音频通道)连接到传统模拟调音台的输入端,然后 MIDI 音序器将程序转换信息发送到 MCA 的每一个独立的通道中,改变每个通道的幅度变化。在缩混处理时,就是创建和保存在音乐中每一时刻、每一点的推子状态发生变化的静态捕捉或"场景"。在一些系统中,还可以设置从上一状态到下一状态自动化变化时的淡入淡出时间(crossfade time),来模拟类似连续的控制方式。

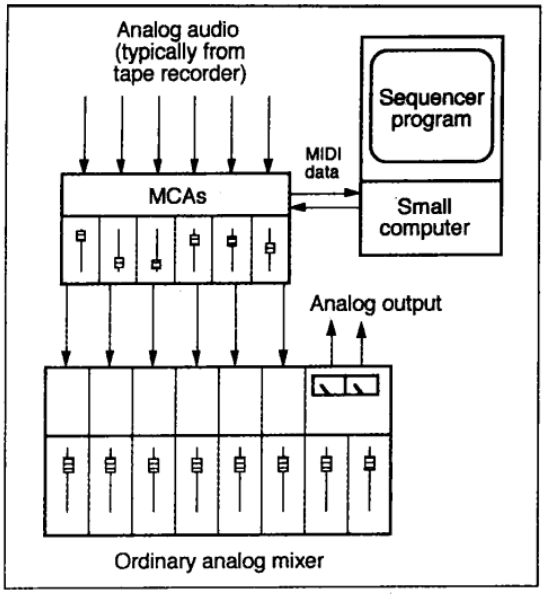


图 9.14　在传统的模拟 6/2 调音台上采用 MIDI 控制衰减系统进行缩混的连接示意。调音台上的电平推子位置相同，录音工程师只需操作 MIDI 控制衰减系统即可控制信号的电平大小。每一个通道的控制信息都以 MIDI 轨的方式记录到 MIDI 音序器中，因此可以实现精确和更加复杂的缩混操作。

Analog audio(typically from tape recorder)=模拟音频信号(来自磁带录音机)
Sequencer program=音序器程序　　MIDI data=MIDI 数据　　Small computer=小型计算机
MCAs=MIDI 控制衰减器　　Analog output=模拟输出　　Ordinary analog mixer=普通模拟调音台

另外一种应用方式是采用 MIDI 音符信息来对调音台上的每一个旋钮和按键进行设置和控制。当录音工程师改变一个旋钮的设置时，调音台发出一个音符事件用来指示那个旋钮的改变。因为每个 MIDI 音符事件都是通过 7 比特来表示其“变化幅度”，这种变化的值就可以被用来表示旋钮当前新的设置值。7 比特意味着可以有 128 种幅度状态。尽管推子本身的分辨率较低，但是通过 7 比特就可以表示为 128 个步进变化了。

第三种应用是通过 MIDI 时间码系统发送提示信息(cue messages)来控制调音台的自动化(见第 21 章和本章后面的 MIDI 时间码部分内容)。这些提示信息可以在操作时间之前发送，这样做的目的在于实现控制调音台在特定时间码位置完成指定速率的切换变化。

音频缩混和视频的同步(Synchronization of Audio Mixing and Video)

这一部分我们来讨论在缩混演播室中的一个逐渐突出的重要问题:多台设备的同步(synchronization)。所谓同步就是指一台以上的设备可以同时进行并行操作。典型的同步应用主要就是视频节目的后期制作(缩混声音到视频节目中去)。目的就是将音频多轨录音机(包括对白、音效以及独立音轨中的音乐)与编辑后的视频节目进行同步。因此就需要音频设备能够与重放的视频节目同步工作,以实现所看到的图像与所听到的声音同步。

在后面 MIDI 时间码的部分中,我们将阐述几种同步的应用,如通过 MIDI 实现音序器控制、效果和声音文件重放。另外,在这部分我们也不讨论在数字音频中的采样时钟同步(sample clock synchronization)的问题。因为在第 22 章中,我们会详细地分析这种同步应用,所以在这一部分我们仅讨论 SMPTE 和基于 MIDI 的同步连接。

多台设备的同步(Multiple-machine Synchronization)

多台设备的同步是通过线缆将设备连接到同步器(synchronizer)上实现的,同步器已经逐渐成为音视频后期制作和音乐录音棚中必不可少的标准设备了。同步器的工作就是将两台设备上已经记录的时间码读出来,以确保一台设备与另一台设备之间的跟随。在每一台机器中,时间码都记录在特定的轨道上,来唯一标示称为帧的位置。正是因为每一帧位置的独立性,才使得如编辑、同步和帧对位的操作成为可能(见后面的 SMPTE 时间码部分)。

与同步器连接中的设备,其中一台起主导作用,这台设备被称为主机,其他与同步器相连的设备则被称为从机,从机根据主机的时间码位置来进行跟踪定位。比如在磁带录音机中,同步器通过控制从机的磁带传动系统来完成与主机的同步,当主机运转到一个指定的时间码位置时,同步器控制从机也到达相应时间码的位置来完成同步。

在基于磁盘的系统中,同步器是通过控制从机的磁盘控制器存取指定位置的方式完成同步的。比如一些作为从机的磁盘录音机是根据从主机上读出的特定时间码来编排重放的声音文件来完成同步。

如图 9.15 所示为一个典型的音视频后期制作系统,用来完成将声效、对白及音乐缩混到录像带的处理。一个音频多轨录音机和一个专业磁带录像机(VTR)都连接到同步器上。其中每一个音频轨和每一个视频轨分别由各自独

立的时间码发生器将时间码记录在各自的轨道上，同时保证在多轨音频上的声音与相应位置上的视频画面相对应。

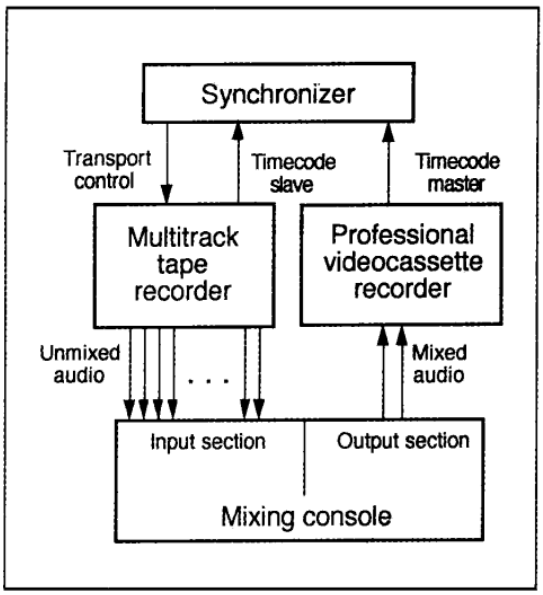


图 9.15　对多轨磁带录音机中的多轨音频进行缩混用于专业视频录像机的音/视频后期制作系统示意图。其中音频多轨录音机和视频录像机都连接在同一台同步器上，视频录像机作为主设备，音频录音机作为从设备。意味着视频录像机在重放时将时间码送入到同步器，经转换后同步器将时间码信息送到音频录音机中用以控制录音机的运转。
Synchronizer=同步器　Transport control=传输控制　Timecode slave=从机时间码
Timecode master=主机时间码　Multitrack tape recorder=多轨磁带录音机
Professional videocassette recorder=专业视频磁带录像机　Unmixed audio=未混音的音频信号
Mixed audio=混音后的音频信号　Input section=输入模块　Output section=输出模块
Mixing console=混音调音台

当录制工程师播放录像带时，同步器将录像带上的新时间码读出，然后将时间码信息发送到多轨录音机上，并指示多轨录音机在第一时间快速地进行跟踪。在多轨录音机启动的同时，录制工程师按下录像机上的按钮将声音记录到录像带的音频轨道上，然后录音工程师就将多轨录音机上各轨的音频缩混成可以为磁带录像机所使用的立体声音频(这种缩混也可以在节目制作的不同阶段来完成。)

SMPTE 时间码(SMPTE Timecode)

SMPTE 时间码规范包括了各种标准的时间码格式。SMPTE(Society of Motion Picture and Television Engineers)是电影与电视工程师协会的简称。SMPTE 时间码主要有两种：LTC(Longitudinal timecode)纵向时间码沿着磁

带的水平方向进行记录；VITC(vertical interval timecode)垂直间隔时间码记录在旋转扫描的视频帧上，也叫帧时间码(旋转扫描指的是当磁带水平通过时，重放和记录磁头垂直旋转，将信息按照垂直条带形式记录在磁带上)。其中LTC纵向时间码又可分为24帧/秒(电影)、25帧/秒(PAL制)、30帧/秒(black-and-white)和29.97帧/秒(掉帧NTSC)。SMPTE的数据率是2 400比特/秒。

VITC时间码的优点在于可以读出磁带上的静止图像。对于基于磁盘的系统来说，可以使用SMPTE时间码格式，但是在同步应用中一定要指定正确的帧率，否则将会由于时基上的动态变化造成可闻噪声。

所有的SMPTE时间码都由一个80比特的数字表示小时、分钟、秒和帧。举个例子，SMPTE时间码为"01：58：35：21"表示1小时58分钟35秒21帧位置。时间码本身并不能全部占用80比特的数字，因此在时间码中还带有一些如消隐时间、指示数或标记的信息。当一个位置被SMPTE时间码标记后，时间码将成为这个位置的永久地址信息。关于SMPTE时间码更多的信息请参看Hickman(1984)。

*接上所述，在具体应用中，每个用于同步的设备都必须在其记录轨道上记录SMPTE时间码。*许多录音机都有特定的轨道来记录SMPTE时间码。作为从机时，能够通过由主机读出的时间码来进行跟踪同步。

MIDI时间码(MIDI Timecode)

MIDI也可以被用于准同步方式缩混过程中(固有的MIDI传输延时意味着要实现毫秒级的同步是不可能的)。其应用主要包括以下内容：

1. 一个MIDI键盘可以控制多个合成器和采样器，因此作曲家可以通过控制键盘实现对多台基于MIDI连接的合成器发出的声音进行缩混。

2. 一个MIDI音序器可以存储一系列的音符信息，通过这些音符信息控制在缩混过程中指定位置的触发。

3. MIDI音序器还可以记录一个预先编码的、基于MIDI控制效果器的一系列程序转换信息。通过这种方法，一些复杂的效果序列就可以自动地应用到用于缩混的声音上。(其中一种应用如MIDI控制下的通道哑音，可参见前面的部分。)

4. 一些基于计算机的音频文件系统可以对通过MIDI发送过来的触发信息进行响应来进行音频文件的重放。

在2、3和4的应用中，主要的技术问题在于：我们如何触发MIDI序列的启动来实现其他序列音频信号的准同步重放？一个直接的解决办法就是手动控

制，通过按下计算机键盘上的按键来触发序列。还有一些类似的方法是通过 SMPTE 时间码和 MIDI 时间码(MTC)之间的连接来实现。

在这种方案中，多轨音频录音机需要在某一轨道上记录 SMPTE 时间码，与其他音频轨道的数据保持同步，然后将 SMPTE 时间码所在的轨道接入到 SMPTE-to-MIDI 转换盒中，通过这个盒子将 SMPTE 时间码转换为 MIDI 时间码，从而触发 MIDI 序列的启动或声音文件的重放(如图 9.16 所示)。在同步领域中触发点也被称为提示(cues)。现在的许多音序器软件都提供了对这种方案的支持。

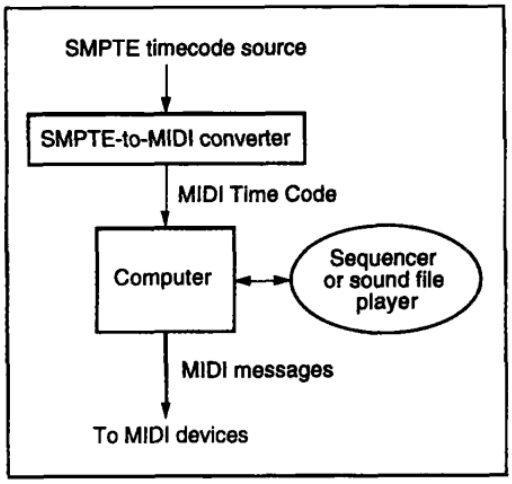


图 9.16　SMPTE 到 MIDI 时间码数据转换示意图。如视频录像机中的 SMPTE 时间码读出后送到转换器中，输出的 MIDI 时间码通过 MIDI 音序器或声音文件重放程序进行翻译，然后输出到 MIDI 设备进行控制。
SMPTE timecode source=SMPTE 时间码信号源　SMPTE-to-MIDI converter=SMPTE-到-MIDI 转换器
MIDI Time Code =MIDI 时间码　Computer=计算机
Sequencer or sound file player=音序器或声音文件播放器　MIDI messages=MIDI 信息
To MIDI devices=到 MIDI 设备

一些国外的音乐同步方式随着交互式 MIDI 操作软件的出现也在不断地发展。可参见第 15 章和第 21 章了解更多的内容。

结论(Conclusion)

声音的缩混仅是制作过程环节中的一个步骤，制作过程从录音开始，然后进行编辑和信号处理。但是缩混不仅仅是一个技术工艺，它还需要具有对音乐的洞察力和判断力。在演播室或舞台演出时，录音工程师扮演着与音乐厅中乐队指挥同等重要的作用，负责对作品中的各个声部进行总体把握。

对监听环境的严格选择更多地在于使用者的品位和喜好，而调音台控制自动化和同步就与价格、质量和性能有更加直接的关联了。同时，缩混技术也在不断地发展，这表现在其在各种领域中的广范应用，从模拟和数控模拟调音台，到大量软件系统，独立式数字调音台和音频工作站中的应用。我们也试图说明不存在某种单一的应用方式适用于所有的音乐制作环境。

不断增大存储容量的数字媒体也使得在单一系统中记录存储数以千计的音频文件成为可能。在一个复杂的缩混中可以使用数以百计的音频文件，相应的问题是如何有效地对这些大量文件进行组织和存取，对于未来的音频数据管理系统也存在着相似的问题(见第 16 章)。

第 10 章　信号处理基础
(Basic Concepts of Signal Processing)

数字化时间颗粒化(*Digital Time-granulation*)

通过调谐器实现变速/变调(Time/Pitch Changing with a Harmonizer)

通过相位声码器实现变速/变调(Time/Pitch Changing with the Phase Vocoder)

重叠相加转换(*Overlap-add Transformations*)

追踪相位声码器转换(*Tracking Phase Vocoder Transformations*)

通过小波变换实现变速/变调(Time/Pitch Changing with the Wavelet Transform)

通过线性预测编码实现变速/变调(Time/Pitch Changing with Linear Predictive Coding)

结论(Conclusion)

这一章介绍的内容主要是音乐家常用的一些重要的与信号处理操作相关的基本概念，其中包括：动态范围变化、滤波器、卷积、延时效果和变速/变调。这些操作的结果将直接导致声音的变化。在后面的第 12 章和第 13 章中将对另一层次的信号处理操作进行介绍，其中包括：声音分析技术，如音调检测、节奏识别和频谱分析。

动态范围处理(Dynamic Range Processing)

动态范围技术可以改变信号的幅度大小，运用这项技术的设备主要包括：包络整形、噪声门、压缩器、限制器、扩张器、降噪设备和压缩扩展器(McNally 1984)。从实际的一些工作如清除噪声信号到具有创造性的工作，如通过重新整形乐器或人声的包络，都是动态范围处理的具体应用。

包络整形(Envelope Shapers)

在许多声音编辑系统中，音乐家都可以对采样声音的全部振幅包络进行重新调节，这会直接引起增益的改变(如在振幅上升高或降低一些分贝)，或是对声音的全部包络进行重新的设计。这种整形的方法可以适用于一个独立的声音对象，也可以针对一个音乐片段来进行。

如图 10.1 所示为一个大键琴的音头形状如何被图 10.1b 所示的包络进行了整形处理。结果造成中间部分的声音在衰减之前保持了持续的声响。

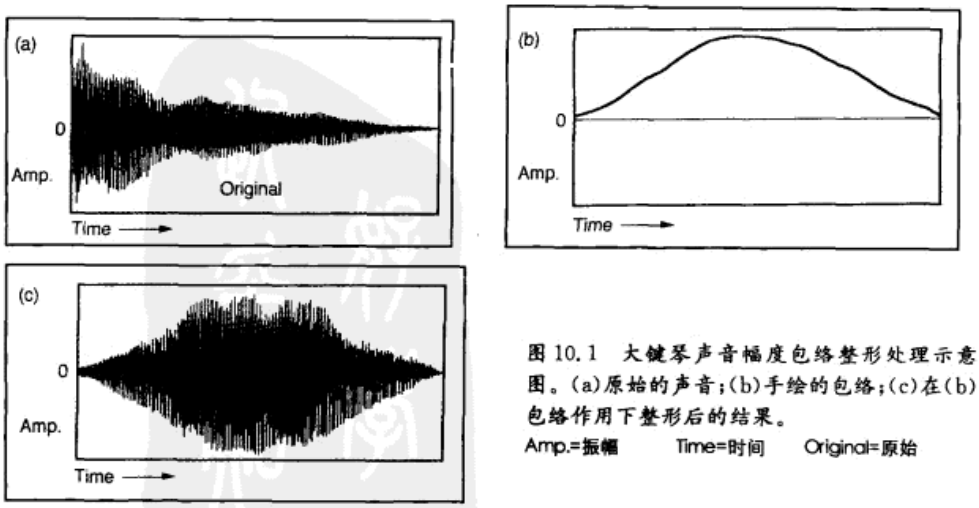


图 10.1 大键琴声音幅度包络整形处理示意图。(a)原始的声音；(b)手绘的包络；(c)在(b)包络作用下整形后的结果。
Amp.=振幅 Time=时间 Original=原始

噪声门(Noise Gates)

噪声门是一种去除音乐信号中不明显的连续噪声如嘶嘶声或嗡嗡声的处理手段。一般来说,噪声电平要比音乐信号电平低。噪声门就像一个切换开关,当高电平的音乐信号通过时,切换开关为开状态;而当音乐信号停止时,切换开关为关闭状态,将残余的系统噪声完全去除。但是当通过噪声门的音乐信号电平低于噪声门设定的门限值时,噪声门也会最大程度地衰减(切换开关为关闭状态)输入的音乐信号,如图 10.2 所示。在图 10.2a 中,一个有噪声干扰的信号逐渐淡出,直到噪声完全可闻;在图 10.2b 中,同样的信号淡出到低于门限值时,噪声门将信号和噪声完全去除。

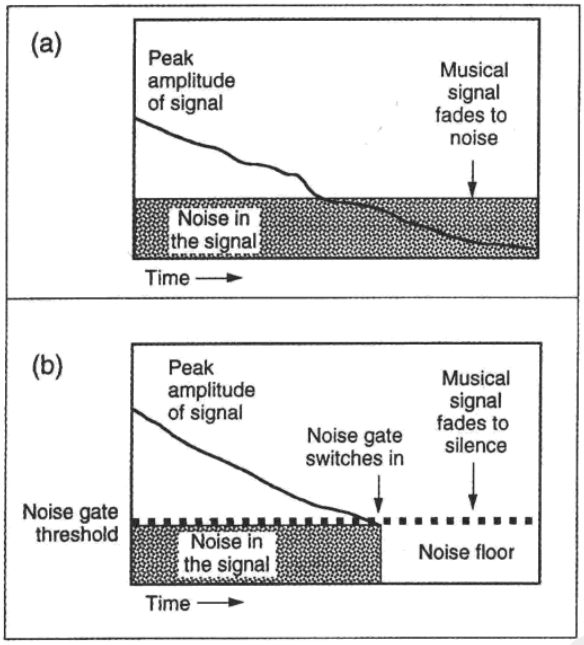


图 10.2　噪声门操作原理示意图。(a)不使用噪声门,低于噪声电平的音乐信号包含了噪声信号;(b)使用噪声门,当信号电平低于噪声门限时,噪声门打开,信号低于噪声门的部分用静音代替。

Peak amplitude of signal=信号的峰值振幅　　Musical signal fades to noise=音乐信号被噪声覆盖
Noise in the signal=信号中的噪声　　Noise gate threshold=噪声门限
Noise gate switches in=噪声门开关打开　　Noise floor=噪声级　　Time=时间

从图 10.2 中我们可以明显地看出,一个简单的噪声门在音乐重放的过程中不能够消除已存在的噪声,因此只有当音乐信号完全掩蔽掉噪声信号时,噪声门设备才能很好地工作。

压缩器(Compressors)

一个压缩器也是一个放大器,它通过控制输入信号的增益(放大的数量)来实现。压缩器的一种用法是保持输出信号相对稳定不变,当输入信号的增长超过了设定的上限时,压缩器对其进行衰减来保持输出的相对不变。

通过转移函数(transfer function)对深入了解压缩器功能的实现是一个比较好的方法。转移函数解释了送入到压缩器中给定振幅的输入信号如何被映射成为指定振幅的输出信号的过程。对于转移函数的描述与在第 6 章中使用转移函数解释波形整形分析是一样的。

图 10.3 显示了几种动态范围处理器的转移函数。我们可以想象一下,输入信号来自正方形的下方,从正方形的右边输出。图 10.3a 显示一个线性转移函数,正方形底部的−1 映射到右边的−1,底部的+1 映射到右边的+1,以此类推。

图 10.3b 所示的转移函数对输入波形的处理可以看成是一个相对"软"的压缩效果,可以看到输入信号的峰值通过转移函数后被映射成为较小振幅值的输出信号了。

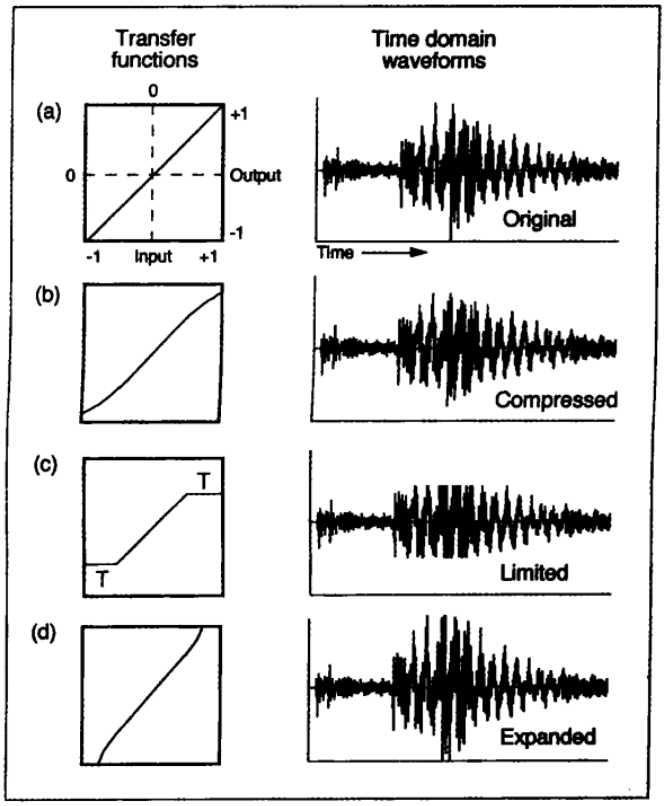


图 10.3 动态范围处理示意图。左边的图示代表不同处理模式下的转移函数。(a)原始信号——经过线性转移函数处理后的镲声;(b)采用软压缩后,使得峰值大小下降几分贝;(c)采用硬压缩后,信号峰值按照门限 T 进行峰值平坦处理;(d)采用扩张模式,会产生更大一些的峰值信号。
Transfer function=转移函数
Input=输入
Time domain waveforms=时间域波形
Time=时间
Original=原始波形
Compressed=压缩后波形
Limited=限制后波形
Expanded=扩张后波形

峰值和平均检测器(Peak versus Average Detectors)

在压缩器的内部,存在着一个检测器电路对输入信号的振幅进行监测。压缩器中的检测器电路可以对输入信号振幅的峰值或平均值进行响应。峰值检测器可以对振幅峰值起反应,即使这些峰值仅仅是瞬时的信号。图 10.3 中所示的动态范围处理都属于峰值检测方式。相反,平均值检测器的响应就比较慢了,它是通过对信号的全部振幅值来进行的,一般为 1 到 2 秒。峰值检测器的快速响应,可以防止振幅过载,但从另一方面来说,平均值检测器则可以对输入信号的变化提供一个平滑的响应。

压缩比(Compression Ratio)

压缩比或输入/输出比是表示输入信号的变化与其相应的输出信号的变化的比值。一个普通的放大器的压缩比为 1∶1。如果比值为 4∶1 则代表着输入信号改变 4 分贝,只带来输出信号改变 1 分贝;如果压缩比大于 8∶1 的话,由于对输入信号的"压扁(squash)",改变了信号的瞬态特征,从而造成了音色上的变化。

高压缩比在流行音乐的后期制作中是一种常用的方法。比如压缩比为 10∶1 左右,可以使得流行音乐中的人声听上去更"亲和(intimate)",原因是所有的那些由于舌头变化、嘴唇闭合、唾液飞溅以及呼吸噪声等发出的夸张了的声音都被缩放到一个相同幅度大小的范围内了。对于一些拨弦乐器如电吉他,高的压缩比也可以产生持续的声音效果。这是因为压缩器降低了弹拨的瞬态特性,同时通过一个大的比例因数进行了整体的提升。比如在吉他中,当压缩后的信号被大幅度地放大时,将会进一步增强弦的持续振荡。

扩张器(Expanders)

扩张器恰好与压缩器相反,它是将输入信号小范围幅度的变化转变为相应的输出信号幅度上的大范围变化。扩张比决定了扩张的程度。举个例子,扩张比为 1∶5 意味着输入信号上 1 分贝的变化将被转换为相应输出信号上 5 分贝的变化。扩张器主要应用在对老的录音作品进行恢复。在降噪设备中也常常包括压缩-扩张这两个环节,这部分内容后面进行讨论。图 10.3d 显示了应用在图 10.3a 所示的输入信号峰值的扩张效果。

限制器(Limiters)

限制器是一种极端的压缩——它的压缩比往往在 10∶1 以上。如图 10.3c 所示,输入和输出在达到指定电平值之前保持线性关系,这个指定电平用正负门限 T 来表示(注意:在一些实际应用系统中,一般只指定一个绝对值作为门限,而不分为上下限度)。当超过这个门限时,输出信号保持持续不变,而与输入信号电平无关。

限制器常被用于音乐会现场录音,用以保证在录音中各个环节的绝对动态范围不出现过载。例如,数字录音机有一个绝对的输入电平门限,一旦超过这个门限值将导致刺耳的数字削波失真,因此录音工程师必须在录音机输入之前插入一个限制器用来保证输入信号不会超过录音机的过载门限。

降噪设备和压缩扩展器(Noise Reduction Units and Companders)

降噪设备通常在录音机的输入部分使用压缩器,在输出部分使用扩张器(如图 10.4 所示)。正是由于这个原因,降噪设备也常常被称为压缩扩展器(压缩器和扩张器的缩写)。其中压缩器部分用来降低瞬态的同时提升其他的输入信号到一个指定的高电平,在重放时则通过扩张器部分将原始动态范围恢复出来。因为压缩后的录音节目仅仅包含很少的噪声信息(通过设定一个适度的超过录音机噪声门限的电平进行录音来达到这个目的),结果就可以得到一个低噪声且大动态范围的录音节目了。

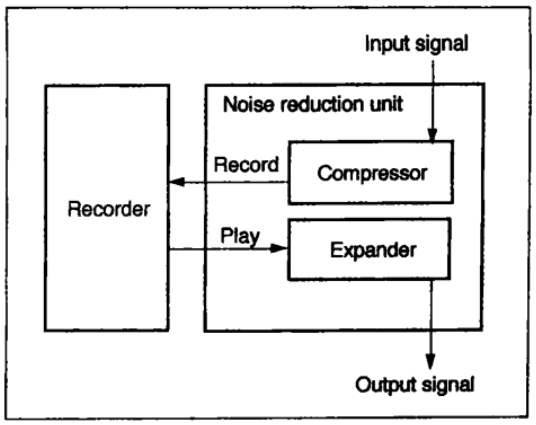


图 10.4 降噪设备在录音时压缩,回放时扩张示意图。
Recorder=录音机
Input signal=输入信号
Noise reduction unit=降噪设备
Record=录音
Play=播放
Compressor=压缩器
Expander=扩张器
Output signal=输出信号

图 10.5 描述了这种扩展的过程。在一个存在噪声干扰的通道中(如模拟磁带录音机或低比特率数字录音机),将录音节目的动态范围进行压缩,录制的信号电平保证在一个足够高的电平值之上,以避免记录在通道中的某些噪声。

同时，还要保证具有低于门限足够的余量，以避免出现过载和削波失真。

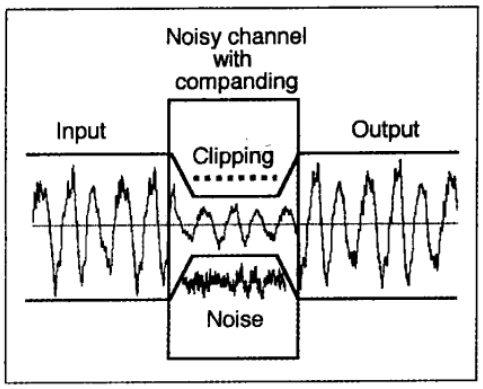


图 10.5　扩展型降噪设备将信号首先进行动态范围压缩后再送入到具有噪声的信道中，这样来保证信号的电平高于噪声电平同时低于削波电平。压缩扩展器的最后部分实现对信号的动态范围进行扩展。
Input=输入　Output=输出　Noisy channel with companding=对带有噪声的通道进行扩展处理
Noise=噪声　Clipping=削波

另外还有一些降噪的方法，如杜比实验室提出的频率相关法来实现压缩和扩张。即首先将输入信号通过滤波器分成一些不同的频带，每个频带都独立地进行压缩和扩张——这个过程被叫做子带分割。对不同的频带进行独立的分割扩展，可以实现每一个频带采用各自不同的压缩和扩张曲线，同时扩展的边界效应也可以降到不易被察觉。这样做的好处在于某些频带可能需要扩展，而另一些频带可能就不需要了。

对于那些覆盖几个频带的声音来说，如连续的滑音，都可能造成上述系统出现问题，这是因为每个频带内降噪电路触发进行操作时会产生可察觉的失真。此外，在频带内振幅对频率的响应即使存在很小的不规则（小于 1 分贝），也会带来整个声音的可闻声染色（Lagadec and Pelloni 1983）。关于振幅对频率响应的描述请参见第 5 章。

动态范围处理的缺陷（Dangers of Dynamic Range Processing）

总体来说，想要改变一个随时间变化的声音幅度而不引起波形瞬态的变形是非常困难的，因为波形瞬态中尖锐的起冲和某些衰减是作为音色表现的基本要素。在动态范围处理中，很容易地就可以对瞬态特性造成损伤，因此在使用这些技术之前应该对它们可能造成的副作用有全面的了解。

动态范围处理器改变了声音整体的起冲和衰减包络，影响了通过它们的全部声音，而无论声音的前后关系，同时对通过它们的信号振幅也进行了改变。另外在“因(信号的振幅变化)”和“果(处理的切换)”之间的反应延时也是一个众所周知的问题。一些设备为了降低这种影响，采用了对输入信号进行很小的延时，并且提前预测可能触发动态范围处理的波形的方法。如果预测到可以触发，它们将实现与这些波形基本同步的效果切换。其他一些产品没有采用这种预测方式。这种方法能够改变触发门限以实现更快的响应，但是这种频繁变化的效果切换将会导致可察觉的“抽动(pumping)”声。

没有任何一种诸如阈值、包络和延时等设置可以适用于多种声音。因此对于这些参数的调整通常是在不进行处理和进行处理出现变形之间折中考虑的。就压缩而言，许多流行音乐制作人寻求的就是这种变形所带来的音质。总而言之，压缩也像其他的效果处理一样，很容易就会使用过度。

数字滤波器(Digital Filters)

某个信号处理工程师委员会对滤波器的定义如下：

“数字滤波器是一种计算处理或算法，它将数字信号或一组序列的数字(作为输入)转化为另一组序列的数字，形成输出信号。”(Rabiner et al. 1972)

因此，任何带有输入输出的数字设备都可以看作是一个滤波器。这些设备最重要的作用就是对声音频谱的区域进行提升和衰减。混响器和音频延时线也是滤波器。这也就意味着事实上滤波器不仅可以改变输入信号的频谱特性，同时还能改变输入信号的时域结构——如精细的调整(对一些指定频率范围作几微秒的延时)，或是较大的调整(对整个信号延时几百毫秒)。

针对音乐家的滤波器理论(Presenting Filter Theory to Musicians)

数字滤波理论是一门纯专业，通过数学语言来表达和描述，远离人类的感受。例如，数字滤波方程没有必要去揭示声音的品质。不幸的是，对于滤波后的效果往往是通过感觉和情感的体验来实现的。对于这种滤波处理后所体现出的美学主观感受很少在信号处理的著作中出现过(Gerzon 1990, Rossum 1992, Massie and Stonick 1992 are exceptions)，即使滤波器对音乐的影响程度可以是从一个极端到另一个极端。音乐家往往使用“刺耳的”、“温暖的”、“悦耳

的”这些词汇来试图描述滤波器所产生的这些不同效果。随着这门艺术的不断成熟，也许会出现更多更准确的术语对其进行描述。

在对于滤波器的主观感受和滤波器的应用集成之间存在着大量的理论。许多描述解释了滤波器的工作情况。在电子工程文献中对滤波器的描述不可避免地都会涉及到 z 变换。z 变换将样本延时的效果映射到一个被称为复数 z 平面(complex z plane)的二维频率域图像中，其中平面上的极点(poles)代表共振峰值，零点(zeros)代表幅度为零的点。例如，一个两极性滤波器(two-pole filter)具有两个共振峰值。对专业滤波器设计师来说，z 变换是一个基本的概念，这是由于 z 变换在滤波器所要求的特性和它的集成参数之间构建了一个数学的桥梁。但是对于 z 变换的描述以及它的应用需要一系列的推导且非常抽象，只是间接的与某些具有物理意义的参数相关。

因此我们对滤波器理论的描述会尽可能地简单一些，而更多地针对在音乐中的处理。我们会按照对样本的一系列延时和简单的算法操作来定义滤波器的本质，用以说明在软件中滤波器如何工作。我们也会将信号流程图、冲激响应和频率响应用图示来进行进一步的补充说明。与第 5 章中涉及的基本滤波器概念相结合，在这一部分涵盖音乐家在创作和演出时使用滤波器所必须了解的基本知识。

乐于挑战的读者在对滤波器理论的研究过程中将会发现有数以百计的文章需要阅读。其中较好的，在音乐方面加以考虑的文章包括 Moore(1978b，1990)、Cann(1979—1980)、Smith(1985a，b)、Moorer(1981b，1983a)。同时还可参见 Hutchins(1982—1988)所著的优秀滤波器设计教程，其中有完整的编码列表。还有大量的工程手册也全部或部分地对滤波器进行了介绍。

在对滤波器的历史进行简要的回顾之后，后面的部分将对滤波器冲激响应的基本概念进行阐述，并说明简单的低通和高通滤波器的处理过程。其中对比两个基本滤波器的结构，讨论滤波器的设计以及现在的滤波器部件、梳状滤波器和全通滤波器。

滤波器：背景(Filters：Background)

最初的电子音乐设备使用模拟滤波器来对由自身音调发生器产生的未加工的波形进行整形，1968 年 Douglas 通过减法音调合成实现了混录处理。在许多著名的乐器中都包括了滤波处理器，如 Mixtur-Trautonium、索洛沃克斯乐器(Solovox)、Clavioline(电子小提琴)、“瓦伯共振峰”风琴(Warbo Formant Organ)、Hammond Novachord、RCA Synthesizer 以及 Ondioline(Jenny 1958，Rhea 1972，Bode 1984)。

西德电台(WDR)演播室中的独立模拟滤波器——Albis Tonfrequenz 滤波器,是众多电子音乐设备中的一个标准单元,如图 10.6 所示。在 20 世纪 50 到 60 年代里,施托克豪森、凯尼格(G. M. Koenig)、埃卢瓦(J. -C. Eloy)和其他一些作曲家曾经在这个西德电台演播室中工作过。后来压控滤波器成为了模块化模拟合成器的黄金时代的代表。

数字滤波电路的试验尝试开始于 20 世纪 50 年代,随着 z 变换微积分的综合采用,数字滤波器理论在 20 世纪 60 年代向前推进了一大步。在声音合成算法语言如 Music Ⅳ和 Music 4B 中已经出现了简单数字滤波器模块。在大型和昂贵的合成器如 Systems Concepts Digital Synthesizer(Samson 1980,1985)和 Giuseppe Di Giugno's 4X(Asta et al. 1980)中实现了大量实时数字滤波器的使用。但是直到 20 世纪 80 年代后期,随着硬件设备的加速发展,使得在低成本的合成器、插卡式信号处理卡、效果器、数字调音台中集成实时的数字滤波器才成为可能。

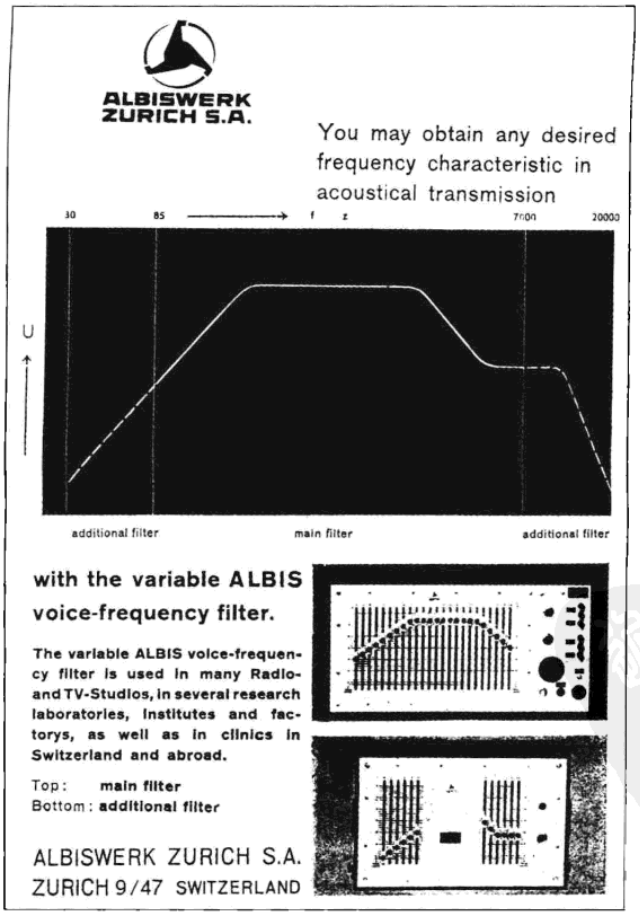


图 10.6 Albis"语音频率"(Tonfrequenz)滤波器是一种在西德广播电台电子音乐演播室广泛使用的图示均衡器。

滤波器的脉冲、频率和相位响应
(Impulse, Frequency, and Phase Response of a Filter)

如图 10.7 所示，我们可以通过在时域或频域中滤波器处理前、后的图示，来观察滤波器的处理效果。当然有些输入信号对滤波器效果的展现比其他的信号更清晰。但是否存在一种理想的输入信号可以对所有滤波器的响应都具有清晰的特征表现呢？一般来说，我们需要将能够包含所有频率信息的信号送入到滤波器中，用于对滤波器的效果进行全面的测试。白噪声信号就具有这样的特点，它包含了所有的频率信息，能够反映出滤波器在频率域的响应。另外一个同样重要的测量则是了解滤波器对瞬态信号的响应，这就意味着我们必须测量滤波器在时域的响应。

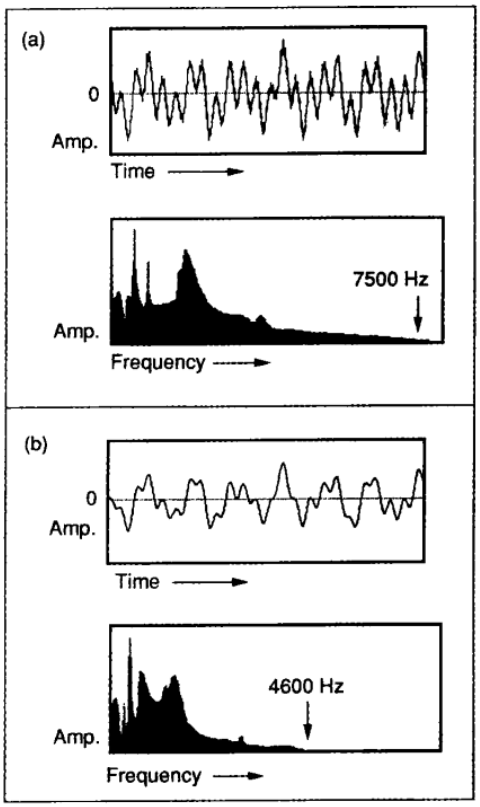


图 10.7　信号通过低通滤波器衰减后的效果示意图，其中包括时域和频域图示。(a)通过音叉琴(19 世纪英国的一种键盘乐器)发出的原始信号片段；(b)上述信号通过一个在 3 000Hz为拐点、衰减 12 分贝的低通滤波器处理后的结果，请注意其中带宽的减小。
Amp.=振幅
Time=时间
Frequency=频率

18 世纪傅里叶揭示出了信号的持续时间和信号的频率内容之间的反比关系。一个无限时长的正弦波代表着一个单一的频率，当我们缩短正弦波的持续时间时，它对应的傅里叶频谱也变得越来越复杂。换句话说，我们需要将更多的正弦波叠加在一起，使得它们彼此相互抵消，从而产生一个持续时间短的信号。因此，信号持续时间越短，相应的频谱就越宽。

在数字系统中，最短的信号持续时间为一个样本的时长，在这个信号中包括了在所有频率上的能量，所有的频率通过给定的采样频率进行表示。因此一个描述滤波器特征的通用方法就是观测其对一个样本脉冲的响应，这是一种概念上的近似和无限短时间的脉冲响应，即克罗内克尔增量。由送入滤波器中的这种单位脉冲得到的输出信号被称为滤波器的冲激响应(IR)。这种冲激响应准确地反映了系统的振幅对频率的响应(在第 5 章中对此进行了说明，即频率响应)。冲激响应和频率响应都包含了相同的信息——滤波器的单位冲激响应——但是对应为不同的域中。也就是说，冲激响应是时域中的表示，频率响应是在频率域中的表示，而两个域之间的转换则通过卷积来实现，我们在后面会进行介绍。

如图 10.8a 所示，一个尖锐的提升带通滤波器扩展了冲激响应后的能量。一般来说，由于表现为尖锐狭窄带宽特性的滤波器通常将导致在原始脉冲之后存在明显的时间延迟。一个长的脉冲响应对应一个窄的频率响应，正如我们在第 13 章中看到的，在窄带滤波器中存在的长延迟时间就成了在频谱分析中的难题。另外，一个短的脉冲响应对应一个宽的、平坦的频率响应。图10.8b 示出了平缓的低通滤波器频率响应的效果。

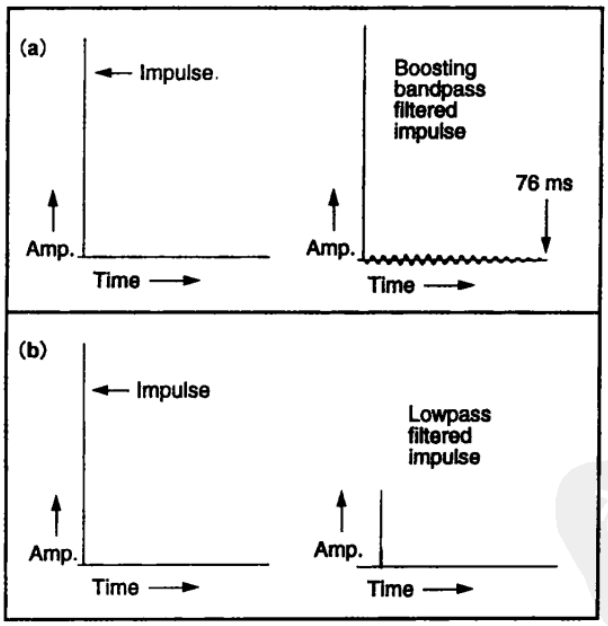


图 10.8 对脉冲信号进行滤波效果示意图。(a)经过一个提升的带通滤波器。左边的竖线代表脉冲信号，右边的信号表示经过一个在 200Hz 处提升 24 分贝、带宽为 20Hz 的带通滤波器后的脉冲信号，注意低电平的“迸溅”扩展响应到了 76ms 处；(b)低通滤波器。经过一个截止频率为 1kHz、衰减 15 分贝的低通滤波器后的结果。

Amp.=振幅 Time=时间 Impulse=脉冲信号

Boosting bandpass filtered impulse=经过一个提升的带通滤波器后的脉冲信号

Lowpass filtered impulse=经过低通滤波器处理后的脉冲信号

滤波器的另一个特性是对输入滤波器中正弦信号相位的影响。相位响应指的是滤波器对每一个输入的正弦信号分量的相位的偏移(以弧度表示)。也许更直观的测量是相位延时,用每一个输入到滤波器中的正弦信号分量的时间延时(以秒来表示)来表示相位的偏移。

滤波器方程式(Filters as Equations)

除了观测冲激响应的图像以外,我们还可以使用与输入信号和输出信号相关的方程式对数字滤波器进行描述。方程式的输出往往通过对当前的和过去的输入样本进行加、减、乘运算得到。这种方程在术语上被称为线性差分方程(linear difference equation)。所谓线性指的是如果将两个具有一定比例的方程之和送到滤波器中,结果与这两个方程独立送入到滤波器中得到的结果之和相同。参见 Rabiner and Gold(1975)或一些信号处理的书籍来了解更多的线性差分方程的内容。

在信号处理书籍中,送入到滤波器中的输入信号被习惯称为 x,输出信号称为 y。输入和输出样本被标号或序列化(如样本在时间 n 处,那么像一个样本就在时间 $n+1$ 处),并且样本的标号往往放到括号中。因此 $x[0]$就代表第 0 时刻的输入样本,$x[1]$就代表下一时刻的输入样本,并以此类推。

简单的低通滤波器(Simple Lowpass Filter)

一个简单的低通滤波器就是对当前的输入样本和之前的输入样本进行平均,换句话说就是,它将当前的输入样本和之前的输入样本相加后再除以 2 得到相应的结果。这种平均化滤波器可以实现对具有尖刺的输入信号的平滑输出。所谓的尖刺就是指那些突然的变化所表现出来的高频分量。平均化滤波器方程式如下:

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

在方程式中的变换常数(0.5)被称为滤波器系数(filter coefficients)。图 10.9 显示了实现这个方程式的框图。

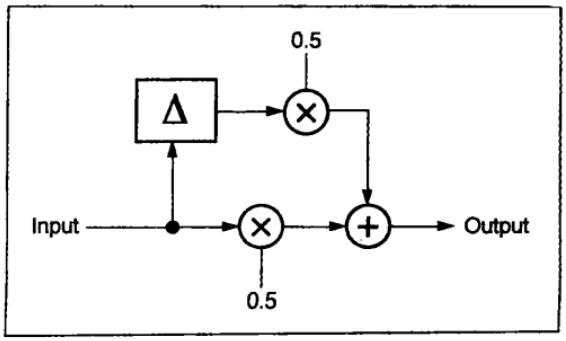


图 10.9 一个简单的平均化滤波器衰减了高频分量，同时导致了采样频率的减半。其中的符号等请参见下面的解释。
Input=输入 Output=输出

注意在图 10.9 中以及后面出现的图示中，符号的含义描述如下：箭头代表信号的流向，没有箭头的直线指示的是输入的系数（用于相乘或相加），小黑点代表支路点，在这里信号被分配到不同的流向，×符号代表着乘法，＋符号代表着加法，Δ 符号代表着一个样本周期的延时。

图 10.10 显示出上述滤波器的频率响应，从图上看上去很像在第一象限的余弦信号。可以看出，不仅可以对两个样本进行平均，同样还可以对三个、四个甚至更多的样本进行平均，从而增强滤波器对高频的衰减效果。这种对多个样本的平均，等于将两个或多个相同的滤波器串联起来使用。

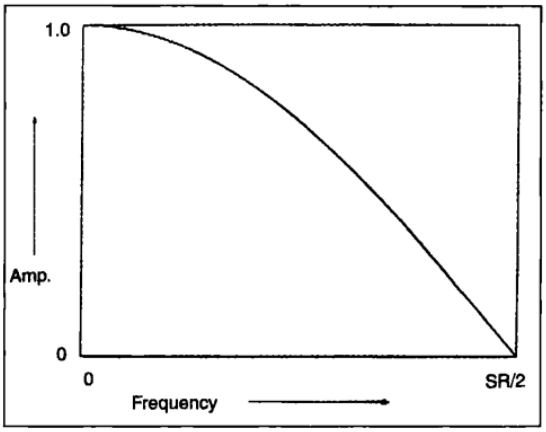


图 10.10 简单平均型低通滤波器的频率响应。
Amp.=振幅
Frequency=频率

简单的高通滤波器(Simple Highpass Filter)

下面我们来讨论用于衰减低频分量的高通滤波器。这种滤波器通过对样本的相减来实现，而不是对它们的相加。换句话来说就是，它计算两个相邻样本之间的差值：

$$y[n]=(0.5\times x[n])-(0.5\times x[n-1])$$

其中 $y[n]$表示当前的输出，$(0.5\times x[n])$表示当前输入信号的一半，$(0.5\times x[n-1])$表示之前输入信号的一半。

现在输出样本 $y[n]$表示的是当前的输入样本减去之前的输入样本后除以 2 得到的结果。这样的高通滤波器对低频分量进行抑制是由于在相邻样本之间低频分量的差异非常小，通过相减即可去除；对高频分量进行扩大是因为相邻样本之间的高频分量相差很大。图 10.11 显示出实现这种方程式的流程框图。图 10.12 显示出这种滤波器的频率响应。

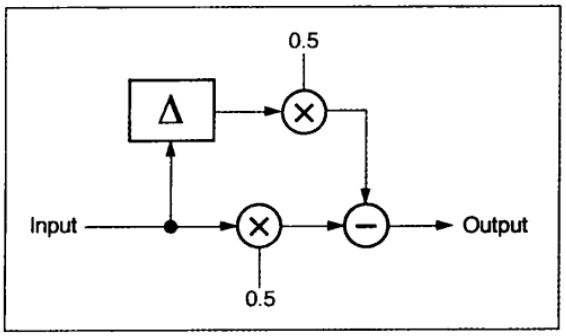


图 10.11　一个简单的高通滤波器框图显示了相邻输入样本之间的相减。
Input=输入　　Output=输出

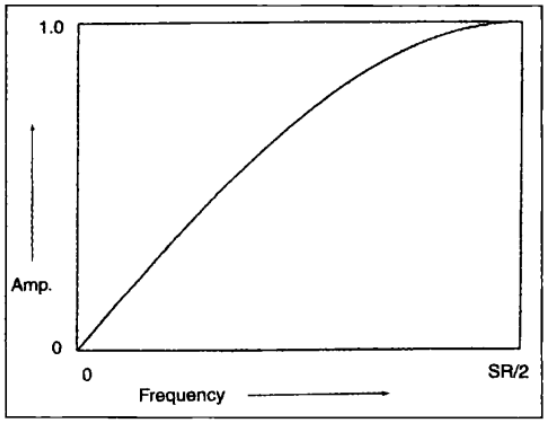


图 10.12　图 10.11 描述的简单高通滤波器所对应的频率响应示意图。
Amp.=幅度　　Frequency=频率

为了更加灵活地使用这种滤波器(同样可适用于低通滤波器)，我们可以将方程式中的变换常数 0.5 替换为变量 a_0 和 a_1，如下所示：

$$y[n]=(a_0\times x[n])-(a_1\times x[n-1])$$

其中系数下标为 0 代表着没有延时的信号，系数下标为 1 代表着延时一

个样本周期的信号。通过改变这些系数的值，就可以相应地改变这些滤波器的频率响应。

通用型有限冲激响应滤波器(General Finite-impulse-response Filters)

这种滤波器的通用方程式如下：

$$y[n]=(a_0\times x[n])\pm(a_1\times x[n-1])\pm\cdots(a_i\times x[n-i])$$

其中 a_i 代表最后的系数，$x[i]$代表着最后存储的样本。系数可以是正数或负数，分别对应于低通和高通滤波器。

这种类型的通用滤波器与延时线(decay line)相类似，延时线通过一个具有 i 样本的循环存储单元对输入信号进行延时。延时线的存储空间在有限的时间长度内循环——i(样本——意味着它对应的是最长的延时线。因此这种滤波器对短时间输入信号(如脉冲)的频率响应随着有限的时间周期逐渐衰减。正是由于这个原因，这种滤波器被称为有限冲激响应(finite impulse response，FIR)滤波器。

图 10.13 显示出这种滤波器的结构，它也被称为横向滤波器(transversal filter)。该图显示出输入信号送入到具有 n 样本时长的延时线中的效果，滤波器将输入信号和所有的延时信号与它们相应的系数相乘后再进行相加，获得输出信号。通过调整这些系数，滤波器响应可以被控制到更低的限制频率，近似为采样频率除以延时的级数。举例来说，在采样频率为 44.1kHz 时，经过 10 级的 FIR 低通滤波器后频率降到大约为 4 400Hz。

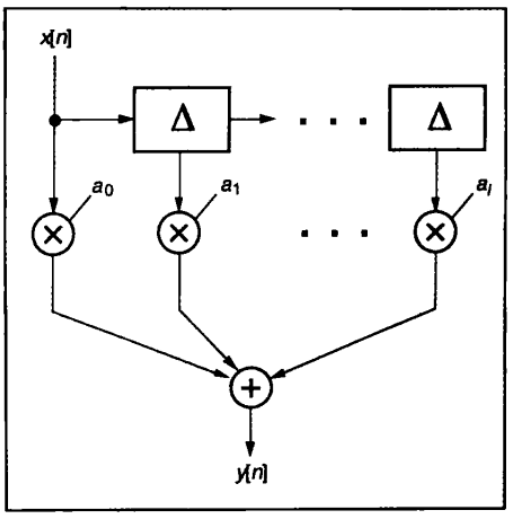


图 10.13 通用性 FIR 滤波器结构图，其中包括一系列的单样本延时，也就是说通过最后的延时单元，输入信号会被延时 i 个样本时长。同时每个延时的样本信号与相应的系数 a 相乘，最终输出的信号就是这些延时后并乘以相应系数的信号总和。

滤波器的延时长度越大，转换带宽就越窄，拐点斜率就越尖锐。毫无疑问，

滤波器延时长度越长，需要的计算量就越大。在实践中，增加长度会带来频率响应上的尖锐化，可是在滤波器的频率响应中主瓣一侧的峰值（如涟漪峰值，ripple peaks）会变得多而密（图 10.14）。

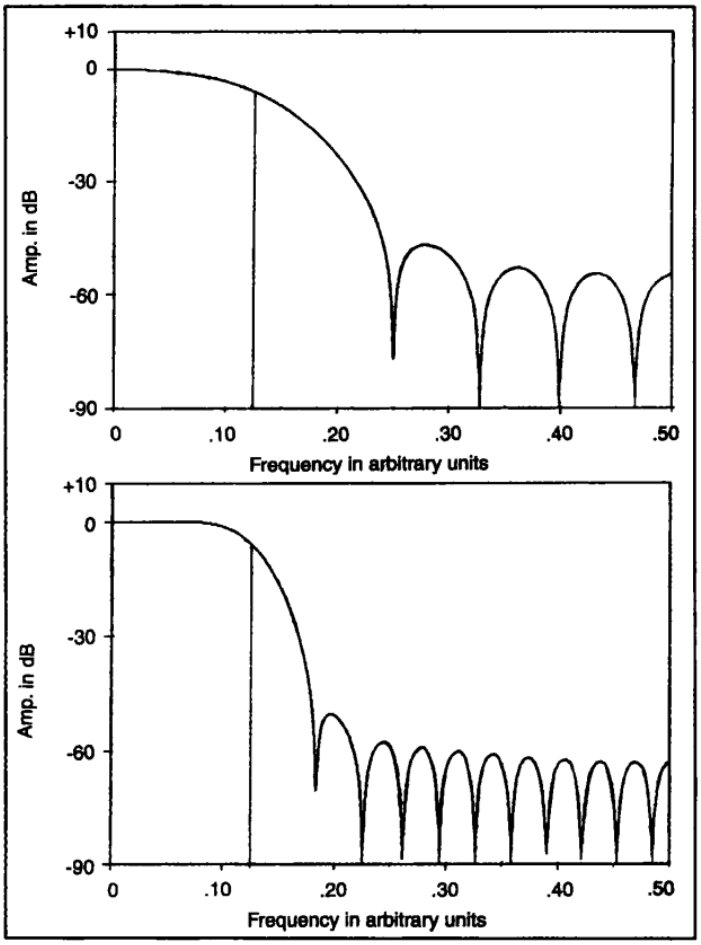


图 10.14　不同长度的 FIR 滤波器对应截止频率斜率的频率响应对比图。频率轴采用任意单位标记。(a)15 阶滤波器；(b)31 阶滤波器。其中图中的竖线代表截止频率(0.125)，可以看出增加更多的延时会增强截止频率斜率。
Amp. in dB=以分贝为单位的振幅
Frequency in arbitrary units=任意单位的频率

简单的无限冲激响应滤波器(Simple Infinite -impulse -response Filters)

如果我们将滤波器的输出再反馈到输入端中，那么滤波器将会对更多的早期信号进行混合，超过使用一个简单 FIR 滤波器所达到的效果，同时可以采用更少的系数。更少的系数意味着更少的乘法运算以及更少的计算量。一个滤

波器使用过去的输出样本被称为反馈或递归(feedback or recursion)。因为这种处理的长度具有潜在的无限性,因此这种滤波器被称之为无限冲激响应(infinite impulse response)或递归(recursion)滤波器。

一个简单无限冲激响应滤波器的例子就是指数时间平均(exponential time average,ETA)滤波器。ETA 滤波器将当前的输入 $x[n]$ 与它的上一个输出 $y[n-1]$ 相加后除以 2,得到一个新的输出样本:

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

图 10.15 为滤波器的信号流程图,显示出反馈回路。图 10.16 为 ETA 滤波器的频率响应示意图。对这个滤波器进行分析后可以看到它等于一个"无限长"的 FIR 滤波器:

$$y[n]=(1/2\times x[n])+(1/4\times x[n-1])+(1/8\times x[n-2])\cdots$$

与 FIR 滤波器相类似,我们也使用变量来代替常数系数:

$$y[n]=(a\times x[n])+(b\times y[n-1])$$

这里根据符号使用惯例,我们采用 b 作为反馈回路的系数。随着 b 的增大,滤波器的截止频率将向更低移动(截止频率的定义请参见第 5 章)。系数 b 的绝对值必须要小于 1,否则滤波器将变得不稳定。在不稳定的滤波器中,输出值 $y[n]$ 将越来越大,导致运算超载(overflow,数字大到音频转换器无法处理)和声音变形。

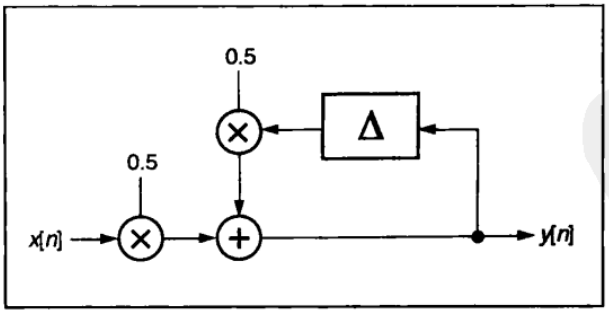


图 10.15 ETA IIR 滤波器信号流程图,注意其反馈回路。

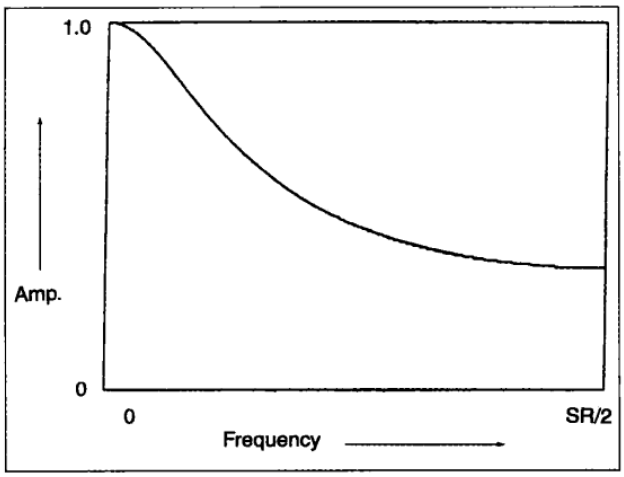


图 10.16　图 10.15 所对应的 ETA IIR 滤波器频率响应示意图。
Amp.=振幅
Frequency=频率

一个简单的递归高通滤波器将当前的输入样本与之前的输出样本相减，然后除以 2 得到当前的输出。图 10.17 显示出这种滤波器的频率响应。滤波器方程式如下：

$$y[n]=(a\times x[n])-(b\times y[n-1])$$

其中 $a=b=0.5$。如果增大 b 值，会提高滤波器的高通截止频率，衰减更多的低频分量。

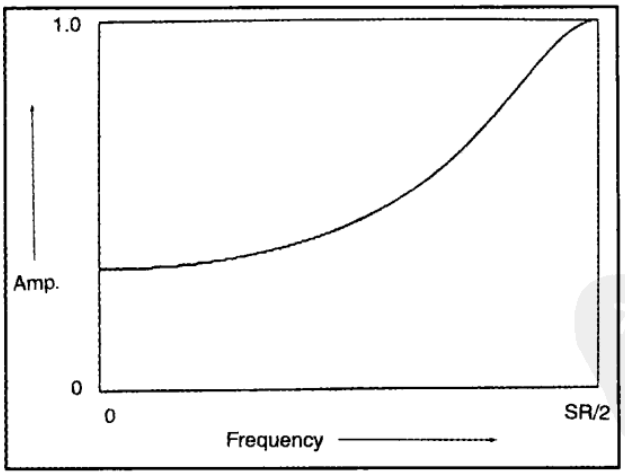


图 10.17　IIR 型高通滤波器频率响应示意图。
Amp.=振幅　　Frequency=频率

通用型无限冲激响应滤波器(General Infinite-impulse-response Filters)

更为复杂的 IIR 滤波器可以被设计为包括前一个输入信号样本(与非零系数相乘)和来自于前一个输出样本的反馈。一个 IIR 滤波器的通用形式如下:

$$y[n]=(a_0\times x[n])+\cdots(a_M\times x[n-M])-(b_1\times y[n])-\cdots(b_N\times y[n-N])$$

也可写为:

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

FIR 与 IIR 滤波器(FIR versus IIR Filters)

两种基本类型滤波器(FIR 和 IIR)的存在带来了一些问题:如何确定在某些应用中采用 FIR 滤波器,在另一些应用中采用 IIR 滤波器?每一种类型的滤波器都有各自的优点和不足。一般来说,可以容易地设计一个具有线性相位响应(linear phase response)的 FIR 滤波器。通常认为这种滤波器对音频信号处理更好一些,因为这种滤波器可以有效地避免相位失真(phase distortion)——一种由于频率相关性的延时而造成瞬态不足和声象模糊的声音质量衰减现象。另外,FIR 滤波器不存在反馈回路,因而比较稳定不会产生自激振荡。而 FIR 滤波器不足之处在于与 IIR 滤波器实现相同频率特性时,FIR 需要更多的算术处理和更多的存储单元(一些 FIR 滤波器具有上千个延时步骤,这部分内容在后面的卷积部分进一步说明)。因此在实现相同效果条件下,FIR 滤波器的硬件实现造价要远远高于 IIR 滤波器。

相比于 FIR 滤波器来说,IIR 滤波器可以使用更少的算术处理来实现尖锐的、指数型的衰减和提升效果。这是因为 IIR 滤波器采用之前的输出信号进行反馈,从而减少了许多算术处理的步骤和对存储器的存取操作,而可以实现与 FIR 滤波器相同的处理效果。但是 IIR 滤波器存在着相位畸变和共鸣(ringing)失真(Preis 1982)。共鸣失真意味着瞬态会激励滤波器,造成在某些时刻当瞬态信号通过系统后产生振荡。换句话说,IIR 滤波器会对信号的整体瞬态进行削弱,模糊高频分量造成音质的粗糙。而且由于 IIR 滤波器中递归的计算性质,使得 IIR 滤波器相对于 FIR 滤波器在滤波器算术处理中具有更敏感的舍入误差(roundoff errors)累积。(关于这个问题可见第 20 章。)

任意规格滤波器设计(Filter Design from an Arbitrary Specification)

以上我们已经介绍了几种基本的滤波器类型实例，每一种都具有自己特殊的性质。但是滤波器设计工程师的任务则是向另一个方向发展，他们必须设计一种可实现的滤波器——包括可对其系数进行设置——从一系列需要的特性开始进行。这些特性设置应包括音频规格如幅度对频率响应、相位对频率响应、冲激响应、群延时、截止频率等，同时还要对诸如字长、计算速度以及与现有软件和硬件的兼容性等实际限制进行设置，更不用说在经济上的限制。

总体来说，实现任意系列规格滤波器是一件不平凡的工作。即使在所需要的规格之间没有什么冲突，也仍然要进行大量的运算和函数变化。因此结果往往是对所需规格的一种近似处理，需要在一种特性和其他特性之间找到平衡。

如前所述，滤波器设计理论涉及大量的学科知识能力，同时还有各种竞争设计策略。许多包含这种理论的工程学教科书都以异常严格和详细的方式来阐述，而在这样一本音乐教程当中进行这种讨论是不现实的。因此我们建议具有一定技术基础的读者进行相关内容的了解。其中我们提及多次的文献(Rabiner and Gold 1975)，是一些关于此内容的经典教材。

幸运的是，滤波器设计理论中那些让人头疼的细节已经被编入到滤波器自动设计系统中了(McClellan, Parks, and Rabiner 1973)。在普通的计算机中就可以调用这些代码库(Smith 1981)并在交互式程序中运行(Hebel 1987, 1989; Zola Technologies 1991; Hyper-ception 1992)。这些交互性程序使得用户可以自行定义设计策略以及所设计滤波器的特性需要，同时还避免了对大量运算处理所需的算术和算子进行设置。另外许多的交互式程序还允许用户测试带有音频信号的模拟滤波器。

建立复杂滤波器模块(Building Blocks of Complicated Filters)

在任意滤波器中用于产生每个输出样本所消耗的最大时间间隔被称为滤波器的级数(order)。如一级滤波器只有一个样本的延时，而二级滤波器可以包括两个样本时长的延时。一般来说设计复杂滤波器就是通过这些一级、二级滤波器形成网络来构成，其中这些一级、二级滤波器相对比较稳定和耐用，要好于直接进行大型的设计和复杂精密的结构。参见 Rabiner and Gold(1975)关于此问题的论述。

一个具有二级部件的 IIR 滤波器主要适用于数字音频系统中(Shpak 1992)。作为二级 IIR 滤波器，对于输出 y 来说，它有两个之前的样本。术语

“部件(section)”意味着这种滤波器能够与相类似的滤波器共同组成更复杂的滤波器组。它可以实现带通频率响应,因此常常被用作组建参数或图示均衡器的模块。通过设置其中的一些系数为0,它还能够实现高通和低通滤波器的效果,因此具有广泛的应用。

在一些书籍中描述了这种二级滤波器的多种形式,我们在此仅给出最通用的形式,方程式如下:

$$y[n]=(a_0\times x[n])+(a_1\times x[n-1])+(a_2\times x[n-2])-(b_1\times y[n-1])-(b_2\times y[n-2])$$

其中系数 a 代表正向路径比例,系数 b 代表反馈路径比例。通常反馈路径贡献响应的峰值,而正向路径则造成凹陷。

对于二级部件的另一个术语是双二次(biquadratic)或双二次滤波器,指的是在方程式中有两个二次结构(一个用于 a,一个用于 b)。图 10.18 显示出了以上阐述的方程式所描述的线路实现框图。这种设计常常被用于音频处理的DSP系统中,采用大量的二级滤波器在某些时候可以实现DSP的实时处理(Moorer 1983b)。

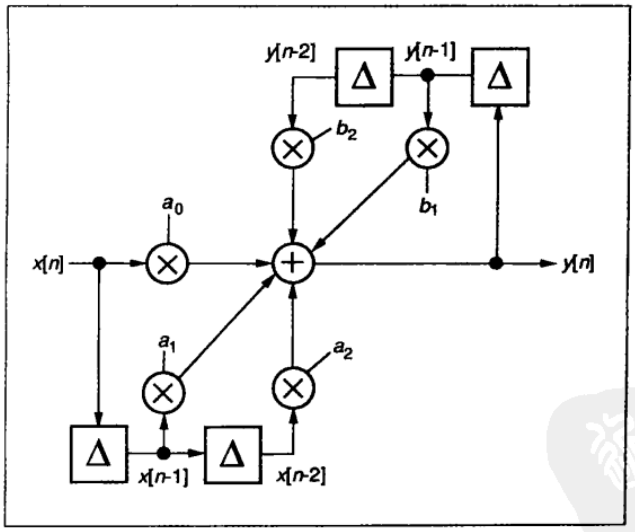


图 10.18 二级滤波器部分通用示意图。其中正向路径位于中线下方,由系数 a 控制;反馈路径位于中线上方,由系数 b 控制。

梳状滤波器(Comb Filters)

梳状滤波器可以产生规则的波峰和波谷序列——频率排列相等——在输

入信号的频谱中。之所以叫这个名字就是由于产生的波峰和波谷序列像梳子的梳齿一样。FIR 梳状滤波器对之前的输入信号进行处理，IIR 梳状滤波器对之前的输出信号进行处理。在这一部分，我们解释这两种类型的梳状滤波器。

FIR 梳状滤波器(FIR Comb Filters)

一个简单的 FIR 梳状滤波器将输入音频信号分为两路，其中一路插入一个多样本的时间延时 D 后，再将两路信号进行相加，如图 10.19 所示。一个简单的 FIR 梳状滤波器方程式如下：

$$y[n]=x[n]+x[n-D]$$

FIR 梳状滤波器与 FIR 低通滤波器的结构非常相似。但是在 FIR 梳状滤波器中既没有原始信号的缩放也没有延时信号的缩放(尽管这是可以实现的)，更重要的是 FIR 梳状滤波器的延时时间 D 是比较长的。在采样频率为 48kHz 情况下，回路中一个样本的延时会产生一个轻微的低通滤波器效果。这是因为延时时间仅仅是 1 秒钟的 0.00002083 分之一或大约为 0.02 毫秒。只有当延时时间大于 0.1 毫秒时，由于相位抵消，滤波器才会在频谱上产生多个空点(null points，这些点的位置上振幅为 0)，从而导致梳状滤波器效果。

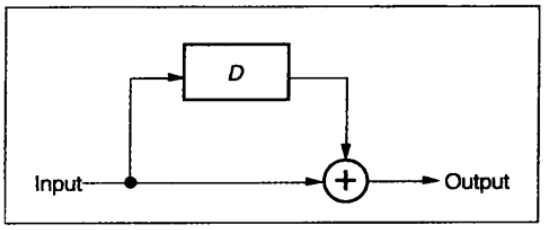


图 10.19　简单的正向梳状滤波器流程示意图。
Input=输入　Output=输出

梳状滤波效果是由延时信号和非延时信号之间的相位抵消和增强的关系产生的，其中同相增强，反相抵消。如果原始信号和延时信号进行相加——当送入到正相相加梳状滤波器中——滤波合成结果中第一个峰值出现在频率 $f=1/D\times f_s$ 的位置，其中 D 为样本的延时时间，f_s 为采样频率。连续的峰值会出现在 $2f$、$3f$、$4f$ 等位置上。因此这种滤波器可以用来对基波频率 f 及其所有的谐波成分进行增强。

举个例子，如果采样频率是 48kHz，延时为 12 个样本时长(0.25 毫秒)，原始信号和延时信号正向相加，第一个听得见的峰值出现在 1/12×48kHz=

4kHz 的位置，依次的峰值出现在 8kHz、12kHz 等位置，直到到达奈奎斯特频率(Nyquist，24kHz)。同样滤波器会在 2kHz、6kHz 及其他相差 4kHz 间隔的位置出现空点，直到到达奈奎斯特频率，如图 10.20 所示。

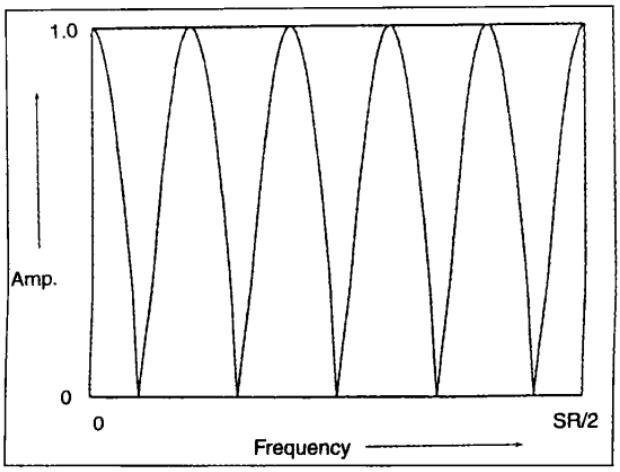


图 10.20 f=4kHz 延时为 0.25 毫秒所对应的 FIR 梳状滤波器频谱示意图。
Amp.=振幅 Frequency=频率

相位抵消和增强也可以用下述说明进行解释。在低频时，延时基本上不会对信号的相位造成影响，当两个信号(原始和延时信号)叠加时，输出信号会得到增强。而对于高频信号来说，延时对它们的影响很大，因为它们的相移越来越接近 180 度。在 2kHz 时，一个 0.25 毫秒的延时刚好会造成 180 度的相移，当这个信号与原始的信号进行相加，两个信号在这个频率上相互抵消(如图10.21 所示)。在 4kHz 时，这个延时会造成 0 度或 360 度的相移，此时两个信号叠加后，输出信号得到增强。在 6kHz 时，这个延时再次造成 180 度的相移，当延时信号与原始信号进行叠加时，两个信号也互相抵消，输出为零，以此类推。

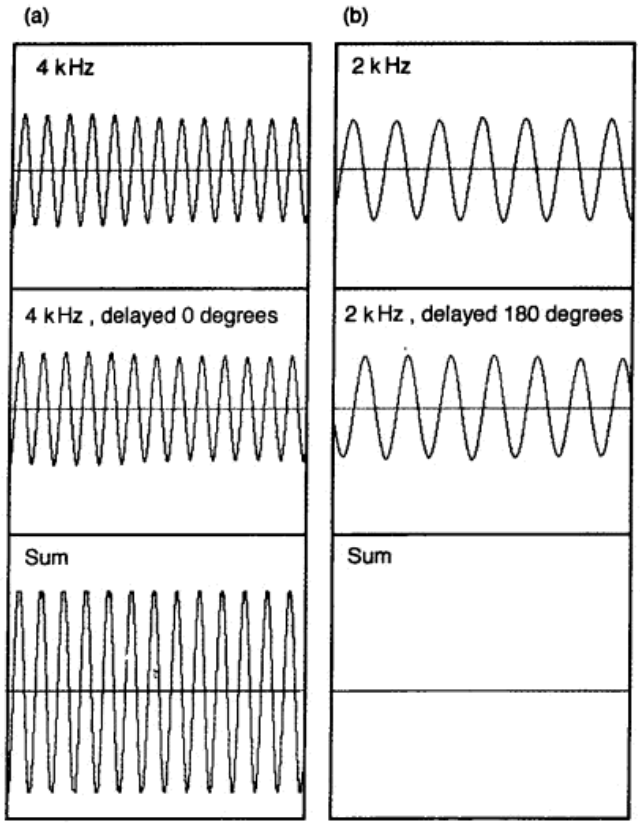


图 10.21　相位增强和抵消效果示意图。左边为(a)图示，右边为(b)图示。(a)最上面为4kHz音调波形图，中间为不加延时的4kHz音调波形图，下面为上面和中间两个波形信号相加后的结果，可以看出信号得到了增强；(b)最上面为2kHz音调波形图，中间为相位延时180度后的2kHz音调波形图，下面为上面和中间两个波形信号相加后的结果，可以看出信号互相抵消了。

delayed 0 degrees=延时了0度　　delayed 180 degrees=延时了180度　　Sum=和

如表 10.1 所示，更长的延时时间会造成更紧密的梳齿(梳状峰值)之间的排列。比如，延时为 50 毫秒，那么第一个空点出现在 10Hz 处，后面依次在 30Hz、50Hz、70Hz 等处出现。小于 5 毫秒的延时时间会产生丰富的梳状滤波效应，这是因为峰值与空点之间的距离也相应增加了，在频率上梳齿变得更宽，对耳朵听觉的冲击力也越强。

表 10.1 FIR 梳状滤波器峰值

延时时间(毫秒)	第一个峰值和峰值间隔
20	50Hz
10	100Hz
2	500Hz
1	1kHz
0.5	2kHz
0.25	4kHz
0.125	8kHz
0.1	10kHz

当两个信号(原始信号和延时信号)不是相加而是相减时会出现什么情况呢？这就是反向叠加(negative summing)的情况。它其实与将两个互为 180 度相差的信号进行相加是一样的。简单的减法 FIR 梳状滤波器方程式如下：

$$y[n]=x[n]-x[n-D]$$

其中 D 为样本的延时时间。如果两个信号进行减法处理而不是相加，那么第一个空点出现在 0Hz，后面将连续在 f、$2f$、$3f$ 等处出现空点，以此类推。在这种情况下，梳状滤波器可以去除掉基波频率以及它的谐波分量，而在 $f/2$、$3f/2$、$5f/2$ 的位置上得到增强，以此类推。

IIR 梳状滤波器(IIR Comb Filters)

一个递归 IIR 梳状滤波器会将它的一些输出信号馈送到输入中去。简单地递归 IIR 梳状滤波器方程式如下：

$$y[n]=(a\times x[n])+(b\times y[n-D])$$

系数 a、b 作为比例因子范围从 0 到 1。图 10.22 显示出了这种滤波器的频率响应。特别是通过 b 值的调整，这种 IIR 梳状滤波器比相应的 FIR 梳状滤波器对信号具有更明显的“共振”效果。事实上，如果 b 值设定得太大，滤波器的反馈将急剧增加，造成算术溢出以及其后的连续畸变。

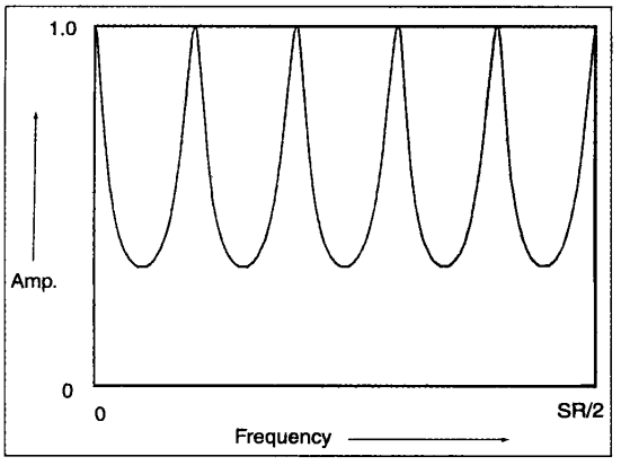


图 10.22　IIR 梳状滤波器频谱示意图。
Amp.=振幅　Frequency=频率

全通滤波器(Allpass Filters)

全通滤波器(allpass filter)是一种特殊的信号处理器。当将稳态的音调馈送到其输入端时，全通滤波器会将所有的频率分量不做任何幅度的变化进行输出——正如它的名字一样。因此全通滤波器在整个的音频带宽上有着平坦的频率响应。但是全通滤波器会对输入信号引入频率相关的相位移位，也就是说它对不同的频率区域进行了不同数量的延时。这种频率相关的延时也被称为离散(dispersion)。

图 10.23 显示出一个全通滤波器的延时与频率的曲线。请注意低频分量如何被延时了。当信号通过与频率相关的相移“染色(colors)”后，全通滤波器产生的可闻效果揭示了信号具有尖锐的上升沿和衰减过程(Preis 1985；Deer，Bloom and Preis 1985；Chamberlin 1985)。Moorer 用这样的语言描述了全通滤波器：

“我们必须清楚全通滤波的特性只是从理论的角度而言，而并非是从我们所感知的效果出发。我们也不能由于频响表现具有绝对的统一性，就单纯地认为滤波器具有感知透明的特点。事实上，一个全通滤波的相位响应表现得相当复杂。全通滤波一般只应用在持续时间较长的稳态声音信号上，这样可以使得频谱平衡不会受到改变。全通滤波通常不使用在短持续时间的瞬态信号上。实际上，一个经验丰富的听众很容易辨别出梳状滤波与全通滤波在听感上的明显区别。”(J. A. Moore 1979)

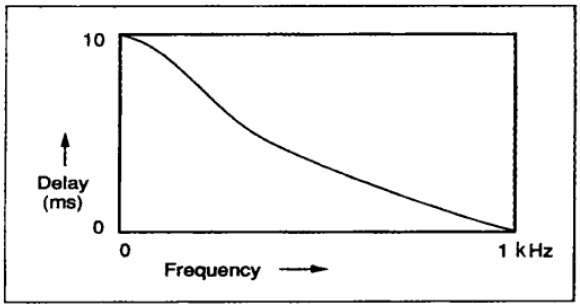


图 10.23 离散全通滤波器的延时与频率的响应曲线
Delay=延时 Frequency=频率

下面的方程式描述了一个具有平直长期频率响应(从 0 到半采样频率),以及对不同频率分量进行不同数量延时的全通滤波器。当样本延时 D 比较大时,全通将会产生一系列的衰减回声,这种效果用在全通混响器中(allpass reverberators,见第 11 章)。

$$y[n]=(-g\times x[n])+x[n-D]+(g\times y[n-D])$$

图 10.24 显示出这种全通滤波器的结构图,与 Schroeder 提出的基本相同(1961,1962;见 Moorer 1977)。这个全通滤波器由一个 IIR 梳状滤波器组成,其中滤波器带有一个反馈回路(通过 g 控制)部分,以及一个嵌入的通过 $-g$ 控制输入信号的正向回路部分。采用减法去除了梳状滤波器的频谱效应,同时保持了回声和衰减的特性。

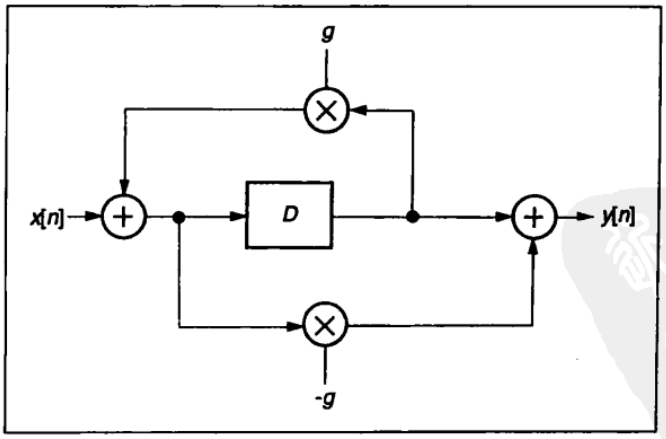


图 10.24 简单全通滤波器结构示意图

一般来说,在一个全通滤波器中的相移(用角度表示)是延时的对数函数。也就是说,一个 100 毫秒的延时相对于一个低频波形来说只是很小的一部

分——仅仅造成几度的相移。但是对于 10kHz 的信号，相同的 100 毫秒的延时相对于频率周期来说却可以造成一个完整的 360 度的相移。

两个属性决定了全通滤波器。翻转频率(turnover frequency)描述的是相移达到 180 度时的频率。全通滤波器中的转换宽度(transition width)是指从 0 度到 360 度相移转换的尖锐度。一个全通滤波器中的转换宽度与带通滤波器中的 Q(峰值比)值相似(见第 5 章对 Q 值的解释)。

全通滤波器在音乐中的应用是多种多样的。在一些普遍应用中，全通滤波器可以用来补偿在其他滤波器中产生的相移(Meyer 1984)。比如，一些音频设备生产厂家对其早期的数字音频录音机进行改进翻新后，使用全通滤波器对原有未进行调整的录音机中潜在的相位畸变进行补偿。另一个应用是在一些合成器中，通过全通滤波器产生时间变化，频率相关相移，从而实现更加丰富的各种电子声。也就是说，我们可以通过这种方法产生所谓的合唱效果(chorus effect)——通过延时和相移的组合应用。也许全通滤波器最主要的应用还是在混响器中，参见在第 11 章中的论述。

卷积(Convolution)

卷积(Convolution)是一种在数字音频信号处理中的基本运算(Rabiner and Gold 1975; Dolson 1985; Oppenheim and Schafer 1975; Oppenheim and Willsky 1983)。每个人都对由它所产生的效果非常熟悉，即使他们从来没有听说过卷积这个概念。例如任何一个滤波器都可以用输入信号来卷积它的冲激响应，以产生滤波后的输出信号。(可以参考前面我们给出的冲激响应定义或 IR。)卷积常常隐含在我们所熟悉的一些术语中如滤波、调制、混响或是交叉合成中。由于卷积的应用越来越普遍，因此在这一节中我们对其进行较详细的探讨。

对给定音频信号进行任意冲激响应的卷积可以产生各种各样的音频效果。比如我们可以制作一个较为复杂的混响器，它其实就是一种复杂的滤波器，通过获得房间的冲激响应，对任意的输入信号与获得的房间冲激响应进行卷积来实现，然后将卷积得到的声音信号与原始声音信号进行混合，就会得到类似于原始输入信号在这个房间中直接进行播放所得到的声音效果。

除了上述的混响效果之外，通过对任意音频处理器(如传声器、扬声器、滤波器、畸变、效果等)相应的冲激响应和音频信号进行卷积可以得到音频信号通过这些处理器后相应的效果特性。

这也使得卷积在音频领域有着重要的作用，其中一个主要的应用就是：通

过对两个任意声音信号的卷积得到交叉合成。通过交叉合成得到的产物可能与原始声音所具有的特性完全不同。如果输入信号是一件乐器发出的声音，结果可能听上去像是乐器在“演奏”其他乐器的声音(如一系列的钟演奏出锣的声音)。在本章节的最后，我们会对音乐信号进行卷积的意义进行更多细节的讨论，同时给出一些使用中的应用指导。

卷积运算(The Operation of Convolution)

为了了解卷积，首先让我们来看一个最简单的例子：信号 a 与单位冲激进行卷积，单位冲激我们用 $unit[n]$来表示。在前面介绍过，一个单位冲激是指在 n 个时间点上的数字序列，在时间 $n=0$ 时，$unit[n]=1$，而在其他时间 n 不等于 0 时，$unit[n]=0$。对 $a[n]$和 $unit[n]$的卷积定义如下：

$$output[n]=a[n]*unit[n]=a[n]$$

这里“*”符号代表卷积，输出的一系列结果与原始信号 $a[n]$是相同的(见图 10.25a)。因此对信号进行单位冲激的卷积，也可以称之为卷积关系中的恒等运算(identity operation)，这是因为任何与 $unit[n]$进行卷积的函数所得到的结果没有发生任何变化。

与具有比例缩放和延时的单位冲激的卷积 (Convolution by Scaled and Delayed Unit Impulses)

下面两个简单的例子充分地告诉我们，在任意卷积运算中样本幅度的变化情况。如果我们对 $unit[n]$通过一个常数 c 进行比例缩放，卷积运算如下：

$$output[n]=a[n]*(c\times unit[n])$$

结果为：

$$output[n]=c\times a[n]$$

换句话说，我们对信号 a 进行了以常数 c 为缩放比例的恒等运算(图 10.25b)。

如果我们对信号 a 与一个具有 t 样本时延的单位冲激进行卷积，此时冲激

出现在样本 $n-t$ 的位置而不再是 $n=0$ 的位置上了，因此卷积运算如下：

$$output[n]=a[n]*unit[n-t]$$

结果为：

$$output[n]=a[n-t]$$

也就是信号 a 进行了以 n 和 t 之差时延的恒等运算(图 10.25c)。

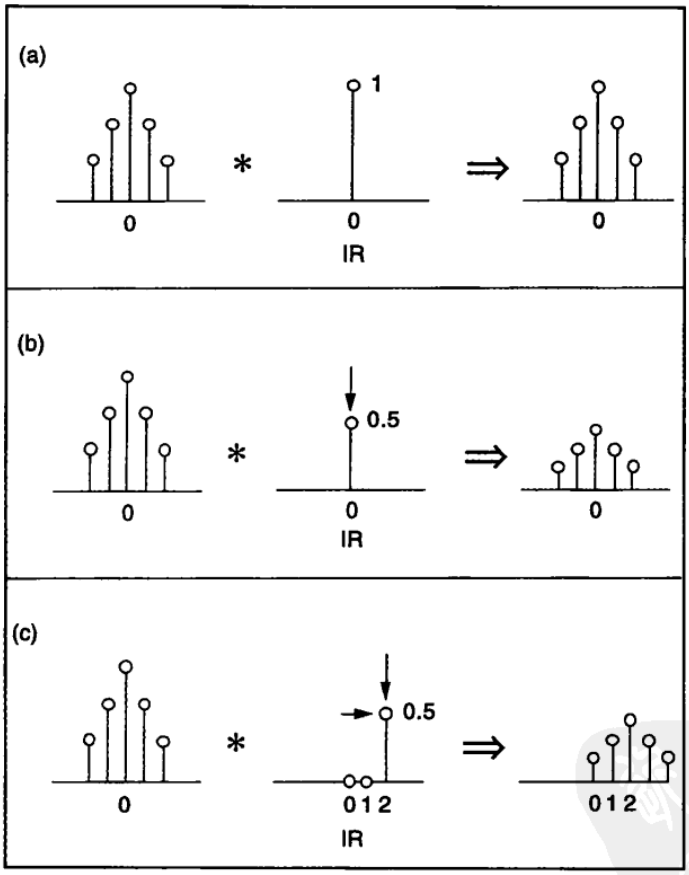


图 10.25　样本卷积范例。(a)对信号进行单位冲激的卷积称为恒等运算；(b)对信号进行以常数 0.5 为缩放比例的单位冲激的卷积；(c)对信号与一个具有一定样本时延的单位冲激进行卷积。

将以上两个结论放在一起，我们可以得出任意一个样本序列函数与具有比例缩放和延时的单位冲激函数卷积后的结果。另外，对一个具有两个有一定空间分布的冲激信号 a 与任意函数 b 进行卷积，结果相当于在信号 a 的两个冲激

位置上分别对信号 b 进行了具有比例缩放和延时的单位冲激函数卷积(图 10.26a)。因此可以看出通过卷积可以产生回声效果。当信号 a 中的两个冲激比较接近时,在两个位置上信号 b 的卷积结果将叠加在一起(图 10.26b),这样就产生了时间融合(time-smearing)效果。当时间融合比较密集(每秒几百个冲激)并且分布比较随机,就产生了混响的效果。

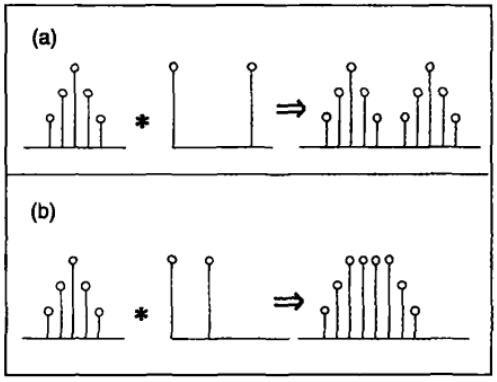


图 10.26 卷积的时域效果。(a)对信号与空间分离较大的两个冲激信号进行卷积产生回声效果;(b)对信号与空间分离较小的两个冲激信号进行卷积产生混响效果。

因此,对于一个输入序列信号 $a[n]$与任意一个函数 $b[n]$进行卷积,只要将 $b[n]$的拷贝放置到信号 $a[n]$中的每一点的位置上,同时以信号 $a[n]$每一点上的数值来进行比例缩放即可。对 a 和 b 的卷积就是对比例缩放和延时后的函数进行求和计算(如图 10.27 所示)。

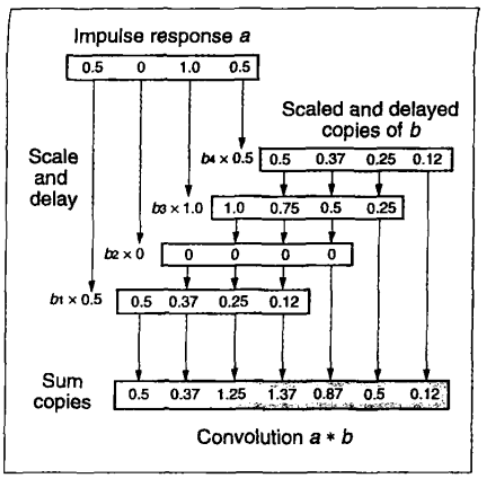


图 10.27 对两个分别具有 4 个样本值的信号 a 和 b 进行直接卷积就是对 b 信号的每个延时副本按照 a 样本的值进行比例缩放,卷积结果 c 就是这些延时并缩放后的 b 信号副本进行求和的结果,结果 c 序列的长度为 7 个样本。
Impulse response =冲激响应
Scale and delay=比例缩放和延时
Scaled and delayed=比例缩放后和延时后的副本
Sum copies=副本相加
Convolution=卷积

卷积的数学定义(Mathematical Definition of Convolution)

两个有限样本序列卷积的数学定义如下：

$$a[n] * b[N] = output[k] = \sum_{n=0}^{N-1} a[n] \times b[k-n]$$

其中 N 代表 a 样本序列的长度，k 的范围就是整个 b 序列的长度。信号 $a[n]$ 的每一个样本都可以看作是一个延时的 $b[n]$ 副本的权重函数，然后这些被加权的和被延时的拷贝进行叠加完成卷积运算。对这个方程进行计算的传统方法就是对每一个 k 上的值进行求和，这被称为直接卷积(direct convolution)。在卷积的中间部分，有 n 个副本相叠加，因此这种卷积方式所得到的结果在后来往往要进行比例缩放(规格化)。

由直接卷积产生的输出序列的长度可表示如下：

$$\text{length}(output) = \text{length}(a) + \text{length}(b) - 1$$

在典型的滤波器应用中，a 是一个冲激响应，与信号 b 相比，它的长度相当短。比如一个宽带平滑滤波器，冲激响应持续时间小于 1 毫秒。

卷积与乘法的比较(Comparison of Convolution with Multiplication)

卷积中包括乘法，但是对两个信号的卷积与两个信号相乘是不同的。一个信号 a 与另一个信号 b 相乘代表着信号 a 中的每一个样本与信号 b 中相应的样本进行乘法运算。即：

$output[1] = a[1] \times b[1]$,
$output[2] = a[2] \times b[2]$,
etc.

而对于卷积来说，信号 a 中的每一个样本都与信号 b 中的每一个样本相乘，得到一个以信号 b 样本长度为长度的每一个信号 a 样本的阵列，卷积是对这些阵列进行求和。另外比较一下信号与单位冲激的卷积(前面所述)和信号与单位冲激的乘法。与卷积明显不同的是，信号 $a[n]$ 与单位冲激 $unit[n]$ 相乘后，其结果除了 $output[0]$ 之外，所有 $output[n]$ 的值都被设为 0，而 $unit[n]$ 等于 1。

卷积定律(The Law of Convolution)

信号处理中的一个基本定律就是两个波形的卷积等于它们对应频谱的相乘,反过来也成立。也就是说,两个波形相乘等于它们对应频谱的卷积。另外一种描述的方式如下:

时域的卷积等于频域相乘,反之亦然。

卷积定律有着很重要的意义,特别是在音频领域中,对两个音频信号的卷积等同于用一个声音信号的频谱对另一个声音信号的频谱进行滤波处理。相反的,将两个音频信号相乘〔如实现振幅调制(amplitude modulation)或环形调制(ring modulation);见第 6 章〕就等于对它们的频谱进行卷积。频谱的卷积意味着输入信号 a 对应的离散频谱中的每一个点与信号 b 频谱中的每一个点进行卷积。卷积并不区分输入序列代表的是样本还是频谱,对于卷积算法来说它们都是离散的序列。

卷积定律意味着每一个时刻信号都对一个声音信号的包络重新整形,信号对其频谱包络和重新整形后的声音的频谱包络进行卷积。换句话说就是,每个时域内的变换必将导致相应的频域内的变换,反之亦然。

卷积与滤波的关系(Relationship of Convolution to Filtering)

卷积与滤波直接相关。复习一下 FIR 滤波原理的通用方程式:

$$y[n]=(a\times x[n])\pm(b\times x[n-1])\pm\cdots(i\times x[n-\mathrm{j}])$$

在此我们可以将系数 a、b、$\cdots i$ 作为矩阵 $h(i)$中的元素,矩阵 $h(i)$中的每一个元素就是矩阵 $x[j]$中相对应元素的乘法系数。按照这种思路,前面介绍的 IR 滤波器通用方程式就可以以卷积的形式表述如下:

$$y[n]=\sum_{m=0}^{N-1}h[m]\times x[n-m]$$

其中 N 代表序列 h 的样本长度,n 的范围为整个 x 序列的长度。请注意系数 h 在卷积方程中扮演着冲激响应的角色。事实上,FIR 滤波器的冲激响应可以直接从这些系数中的值得到,因此 FIR 滤波能够被看作是卷积,反之亦然。

既然 IIR 滤波也可看作是卷积,就有理由来考虑在它的系数和它的冲激响应之间是否也存在着直接的联系。答案是否定的。但是现在已经有了一些用

于设计 IIR 滤波器的数学算法来近似地模拟给定的冲激响应。参见 Rabiner and Gold(1975, p. 265)。

快速卷积(Fast Convolution)

直接卷积是一种众所周知的运算量大的算法，它需要进行 N^2 次运算，N 是最长的输入序列的长度。因此当简单的方法能够满足需要时，直接卷积就很少用于窄带滤波器或混响器中(这两种设备都具有较长的冲激响应)。

在许多实际应用中，卷积是采用一种被称为快速卷积(Fast Convolution)的方法来实现的(Stockham 1969)。对于长序列来说，快速卷积利用了两个 N 点的离散傅里叶变换(discrete fourier transforms, DFTs)等同于两个 N 点序列卷积的离散傅里叶变换的事实。因为 DFT 可以通过使用快速傅里叶变换(fast fourier transform, FFT)算法来进行快速的计算，这就使得卷积运算速度得到巨大的提升(在第 13 章和附录中对 DFT 和 FFT 进行了说明)。在进行 FFTs 之前，两个序列通过添补 0 的手段将两个序列的长度都拉长到与卷积输出序列长度相同的长度。这种处理叫做添 0(zero-padding)，在第 13 章和附录中也进行了讨论。卷积的结果也可以通过应用 FFT 逆变换来重新合成。图 10.28 描述了快速卷积的整体方案。

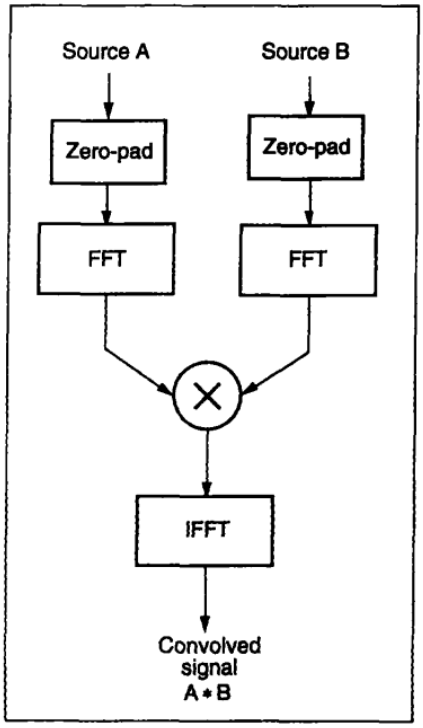


图 10.28　快速卷积方案
Source=源　Zero-pad=添 0
Convolved signal=卷积后的信号

这就是说我们可以使用 FFTs 来代替直接卷积进行卷积运算，并且对于数值较大的 N 也有着快速的运算速度。事实上，快速卷积只需要进行 $N \times \log_2(N)$ 次运算。举个例子来说明，假设使用直接卷积对两个采样频率为 48kHz 时间为 2 秒的声音进行卷积运算，那么需要进行 $96\,000^2$ 或 9 216 000 000 次运算。而对于同样的两个声音信号采用快速卷积算法只需要不超过1 500 000次运算，整整提高了 6 100 倍。换句话说，对于一个给定的微处理器，使用快速卷积花 1 秒钟计算的工作，换成直接卷积的话则需要 101 分钟来完成。

对于实时应用，或多或少地直接输出是必需的，此时也可以采用分段式卷积来实现，也就是说，在一定的时间内只有很少的样本需要卷积处理。分段式和非分段式卷积产生的结果是相同的。参见 Rabiner and Gold(1975)和 Kunt (1981)关于分段式卷积技术标准的解释。其中 Rabiner and Gold 还讨论了关于实时卷积器的执行。

卷积在音乐中的重要性(Musical Significance of Convolution)

很多的声音转换都可以被解释为卷积处理过程，包括将在后面三个部分进行讨论的滤波、时间效果以及调制。

作为卷积的滤波(Filtering as Convolution)

滤波是一种频谱相乘的典型应用，因此我们可以通过对输入信号和所需滤波特性的冲激响应进行卷积来实现各种滤波效果。但是，卷积还可以超越简单滤波而达到交叉合成——通过一个声音对另一个声音进行滤波处理。让我们命名两个信号为 a 和 b 以及它们相应分析后的频谱为频谱 a 和频谱 b，如果我们将频谱 a 中的每一个点与频谱 b 中相应的点进行乘法运算，然后重新合成结果频谱，这样我们就可以得到信号 a 和信号 b 在时域中卷积后的波形。比如我们对两个萨克斯的音调进行卷积，每个音调都具有平缓的音头，将它们按照一定的程度进行混合，两个音调听起来就像被同时演奏一样。与简单的混合不同，卷积的滤波效果衰减了这两个音调中的金属共鸣。另一种效果，在此例中比较不明显，它就是时间融合，我们接下来进行讨论。

卷积的时域效果(Temporal Effects of Convolution)

在卷积中也包括时域效果，如回声、时间融合和混响(Dolson and Boulanger 1985；Roads 1993 a)。这些效果依赖于被卷积信号的特性而表现得不明

显或很明显。

对于输入中一个信号的冲激响应的卷积将得到另一个信号的一个副本。因此如果我们对任意一个声音信号与一个由两个间隔 500 毫秒的冲激响应组成的信号进行卷积时，结果将会是第一个声音信号的清晰回声。

对于一个房间来说，房间的各个表面都对应着不同的反射效果——反射面，因此存在着许多的房间冲激响应。当这个房间的冲激响应与任意一个声音信号进行卷积后，结果听上去就像是这个声音在此房间中进行重放所得到的效果一样，这是因为它反映出了房间反射面的情况。

如果冲激响应序列中峰值排列较密集，副本之间就会形成时间融合效果（见前面的图 10.26b）。时间融合将信号中尖锐的瞬态变化抹平，同时模糊了声音中清晰的音头。图 10.29 显示出对一个牛铃声进行自身卷积所产生的时间融合效果。

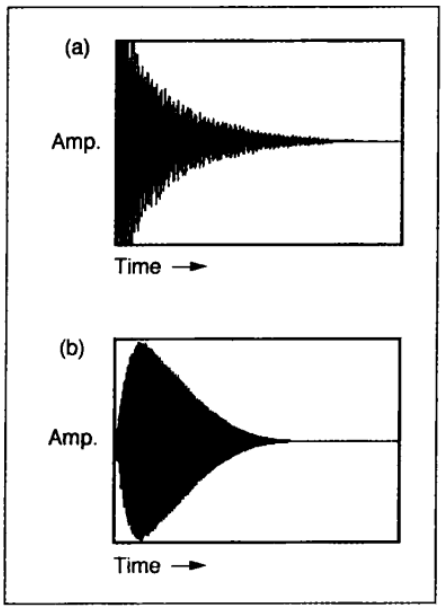


图 10.29　时间融合效果示意图。(a)带有尖锐音头的原始牛铃打击声音波形；(b)通过自卷积得到的结果，可以看到音头部分产生了时间融合效果。
Amp.=振幅　　Time=时间

结合时间融合和回声效果，我们不难解释为什么包含很多尖锐峰值的噪声信号经过卷积后成为了混响效果。如果噪声信号的幅度包络有尖锐的音头和指数型的衰减，卷积后的结果会是一种最接近自然的混响包络。为了丰富混响效果，我们可以在卷积之前或之后对噪声进行滤波处理。如果噪声具有对数型的衰减，那么第二个声音将会在衰减之前暂缓出现。

作为卷积的调制(Modulation as Convolution)

振幅和环形调制(见第 6 章)都可以被看作时域波形的乘法运算。根据卷积定律时域相乘等于频域卷积的描述,对于边带的卷积会导致这些调制效果。重新看图 10.26,我们想象去掉时域中的冲激,卷积在频域中以线性方式进行。相同的规则也可适用——但是这种算法与采用复数算法有着重要的区别。举个例子来说,FFT 产生了每一个频谱分量上的一个复数。附录 A 解释了这个过程,同时在主点位置也就是 0Hz 处产生的对称的频谱分量的拷贝(振幅等分)出现在频域的负半轴上。这部分负半轴上频谱相当小,因为它只有在 FFT 区域内才有意义。

图 10.30 显示出发生环形调制的频谱卷积过程。其中图 10.30a 所示为 100Hz 正弦曲线经 FFT 后的频谱,图 10.30b 为 1 000Hz 的正弦曲线经 FFT 后的频谱,图 10.30c 揭示了卷积后的结果。在−100 和 100 位置的脉冲被延时和比例缩放至 1 和−1kHz 的区域内,其中频率 900 和 1.1kHz 代表两个输入信号频率的和与差,这就是典型的环形调制。

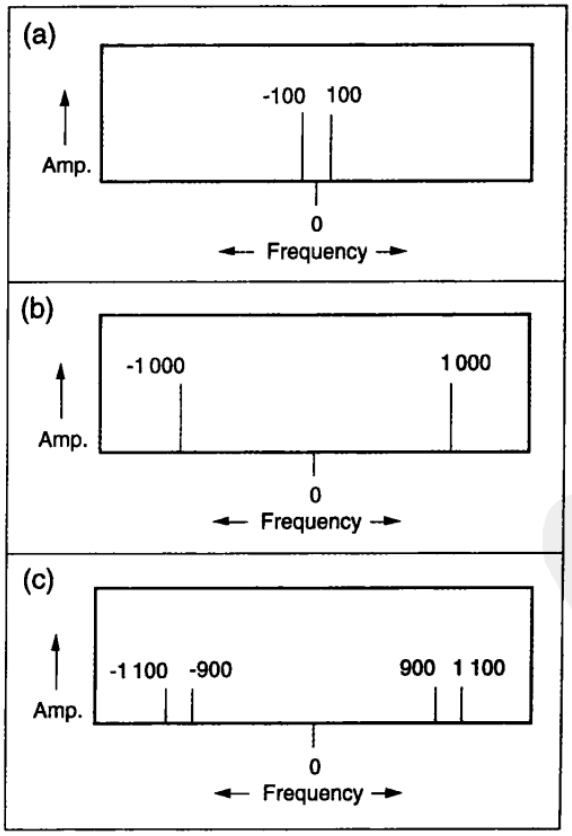


图 10.30 卷积的环形调制示意图。这些图像中的频谱都位于 FFT 区域内,并且可以看到都呈现出对称性。其中(a)为 100Hz 正弦曲线经 FFT 后的频谱;(b)为 1 000Hz 的正弦曲线经 FFT 后的频谱;(c)为(a)和(b)卷积后的结果。
Amp.=振幅
Frequency=频率

颗粒和脉冲的卷积(Convolution with Grains Pulsars)

声音转换中的一个特殊类型是声音信号与一些声音颗粒云(clouds)的卷积〔见第 5 章对异步颗粒合成(asynchronous granular synthesis)的描述〕。在这种情况下,这些声音颗粒本身不会被听到,但是他们往往被作为不同寻常的滤波器或人造空间的"虚拟的冲激响应"来使用(Roads 1992b)。

声音信号与这些颗粒卷积时会根据这些颗粒云的分布和输入信号的特点而产生变化非常大的结果。对于一个具有尖锐上升沿的输入信号来说,如果与之进行卷积的颗粒云分布比较稀疏并且每个颗粒的持续时间很短,那么结果将会产生输入信号回声的统计分配,即在颗粒位置上的信号回声效果,如图10.31所示。如果颗粒云的分布比较密集,那么就会产生较密集的回声,最终融合为无规律的混响效果。如果颗粒的持续时间较长,那么结果将会导致声音融合的加强,并且会产生具有尖锐上升沿的信号。当输入信号具有较为平缓的上升沿时,如连奏的萨克斯音调,结果将会导致在音调上出现类似于时间变化的滤波效果,这主要依赖于在颗粒中波形的频谱(关于此技术的更多细节见 Roads 1993a)。

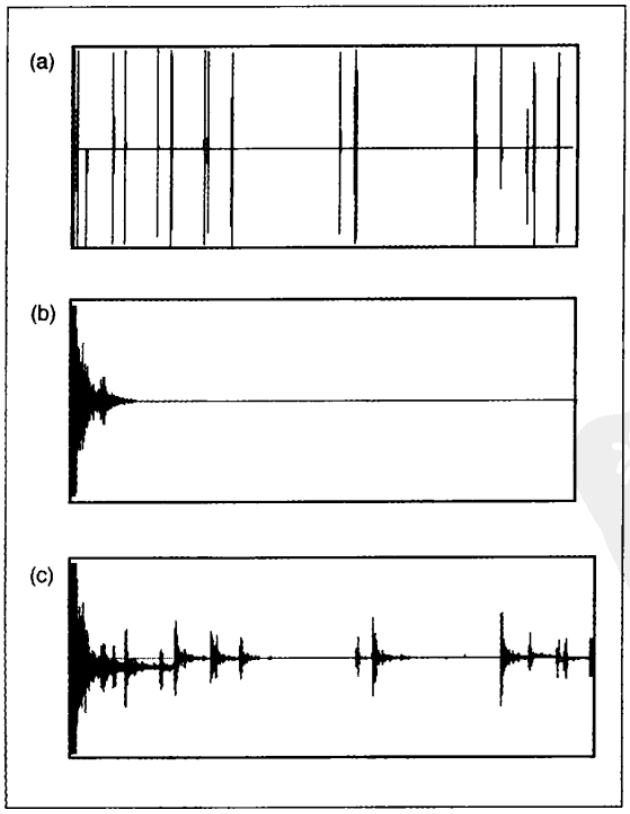


图 10.31　与颗粒信号的卷积。(a)分散出现、持续时间为 0.5 毫秒的颗粒云;(b)小手鼓的打击声;(c)对(a)和(b)卷积的结果产生了铃鼓打击声在相应颗粒出现位置上的重复。注意(a)中第二个颗粒使得卷积结果能量向负半轴的瞬间偏移。

另一类的声音合成是通过对样本声音和一种称之为脉冲星(pulsars)的可变波形冲激颗粒进行卷积产生的。这些脉冲星颗粒连续散落在次声波与声波频率之间,结果导致类似于鼓点一样的一系列节奏效果。关于此技术的细节可参见 Roads(1994)。

线性和循环卷积(Linear versus Circular Convolution)

直接卷积是一种线性卷积。如前面所述,对于输入信号直接卷积结果的长度定义如下:

$$\text{length}(output)=\text{length}(a)+\text{length}(b)-1$$

给定一个具有 1024 样本长度的输入信号 a 和一个具有 512 样本长度的冲激响应 b,直接卷积后输出结果的长度为:

$$\text{length}(a)+\text{length}(b)-1=1535 \text{ 样本}$$

这是因为信号 a 中的每一个样本都要与信号 b 中的每一个样本进行卷积,包括中 a 信号中的第 1024 个样本,共冲激响应扩展了 511 个样本。

循环卷积是一种不规则的卷积,主要应用于通过 FFT 实现的卷积中。每个 FFT 取 N 个样本作为它的输入(其中 N 是两个输入信号序列中较长的序列值)。快速卷积产生 N 个样本作为它的输出。那么对于线性卷积来说这种扩展是如何发生的呢?

在快速卷积中,这些扩展点被直接"环绕"到 1024 个点系列的开始位置,就像是一个循环列表从开始位置到结束位置进行接合。结果在开始位置和结束位置的卷积都包括了无效数据。幸运的是通过指定 FFT 窗口大小(windows size),等于或大于所期望的输出序列的长度,就可以容易地避免在循环卷积中出现的上述畸变(见第 13 章关于窗口大小的讨论)。这可以通过设定 FFT 窗口大小等于两个能量最接近并且比最长的输入序列中 N 个样本都大的尺寸来实现。其他的样本点都添为 0。

去卷积(Deconvolution)

不幸的是,一旦两个信号进行了卷积,那么对它们进行分离或去卷积处理希望完美地得到这两个原始信号的方法还没有实现。假如已知一个信号

的频谱，可以通过对这个卷积后的信号进行滤波处理来去除它的频谱，但是由于时间融合（如回声和包络整形）在卷积中造成的其他失真仍然会保留下来。

由于语言信号的特殊性质，通过去卷积可以对两个语言信号的声音激励（声带脉冲）和共鸣（声腔共振）实现近似的分离。其中，包括自回归和同形去卷积（Rabiner and Gold，1975）。第 13 章介绍了自回归分析，它与第 5 章中的线性预测编码（linear predictive coding）有着紧密的联系。同形去卷积是一种对数倒频谱分析技术，在第 12 章中进行了描述（见 Galas and Rodet1990）。

固定时延效果（Fixed Time Delay Effects）

时间延时是一种通用的音频信号处理技术。一个数字延时单元（digital delay unit）或数字延时线（digital delay line，DLL），是通过对输入信号样本流取出并存储它们到存储器中，并保留一定的周期后再进行输出的装置。然后将延时后的信号与未延时的原始信号进行混合，能够得到各种各样的音频处理效果，下面我们进行说明。

延时线（DDL）与 FIR 低通滤波器和梳状滤波器的比较（Comparison of DDL with FIR Lowpass and Comb Filters）

图 10.32 是一个简单的数字延时线流程的示意图。参见图 10.9 所示的 FIR 低通滤波器和图 10.19 所示的 FIR 梳状滤波器，可以看到这些流程中的相似之处。它们之间的主要区别不是结构流程上的不同，而是延时的引入。对于低通滤波器来说，延时是一个样本，因此结构流程具有对连续样本进行平均的效果。对于梳状滤波器来说，有效的延时在 0.1 到 1 毫秒之间。而对于数字延时线来说，延时时间可以大于 1 毫秒。

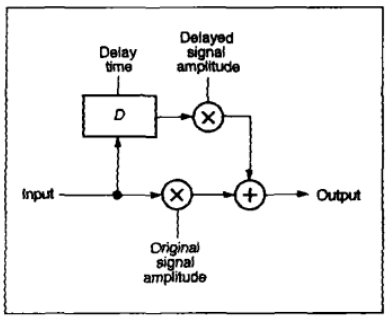


图 10.32　数字延时线流程示意图。注意此图结构与图 10.9 和图 10.19 的相似性。
Input=输入　Output=输出
Delay time=延时时间
Delayed signal amplitude=延时后信号的振幅
Original signal amplitude=原始信号的振幅

延时线的实现(Implementation of a Delay Line)

在信号处理器中,被称为循环队列(circular queue)的数据结构是一种最有效的延时线实现方式,如图 10.33 所示。这个队列可以看作是包含音频样本存储定位的简单顺序列表。在每一个样本周期,延时程序将旧样本读出并且通过在同一个位置上写入一个新的样本进行对旧样本的替换。读/写指针行进到队列中下一个包含旧样本的位置(见第 2 章对队列和指针的解释)。当指针到达了序列的结尾(end)处,会再一次跳到起始(first)的位置,这就是其被称为循环队列这个术语的原因。

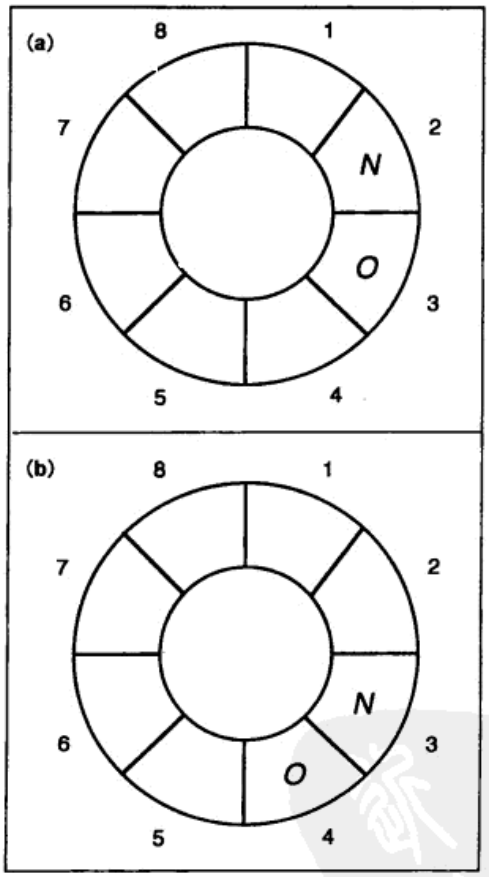


图 10.33 通过循环队列操作实现延时线示意图。其中 N 是序列中最新的样本,O 是最老的样本。(a)"之前的状态",循环序列在时间 t 时刻的指针位置;(b)"之后的状态",序列在时间 $t+1$ 时刻的指针位置,通过在时间 t 时刻将最老的样本读出并且用一个新进入的样本进行替代来指示空间保留。

迄今为止,我们描述了一个与队列长度相对应的具有固定持续间隔的延时。这种延时有一个读指针——在信号处理术语中称为 tap,并且读指针总是处于相同位置写指针的前面。通过读指针在队列中任意位置的读取,我们可以实现延时间隔短于队列的长度,其中包括了延时变化的时间消耗。这些可能性

就实现了我们在后面将要讨论的可变时间的延时。

从理论上来说，多抽头延时线(multitap delay line)具有多个读指针。图 10.34 显示出一个循环队列的多抽头延时线的实现。在每一个样本周期，一个新的样本都会被写入到标记为 N 的存储位置上，同时两个样本从标记为 Tap1(代表一个样本的延时)和标记为 Tap2(代表三个样本的延时)的位置上读出。然后所有的指针增量行进到下一个位置，为下一个周期做准备。

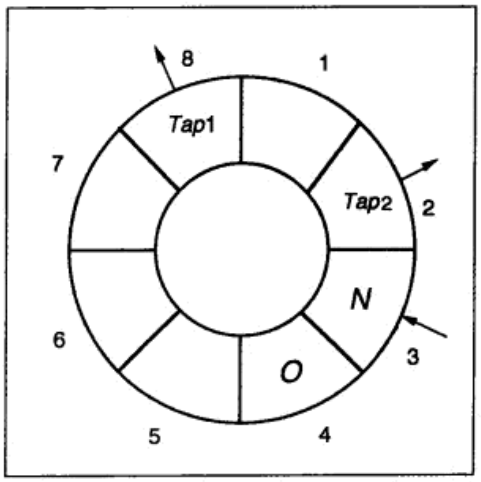


图 10.34　在一个循环队列中实现的两抽头延时线。Tap1 和 Tap2 分别为两个读出抽头，按照位置 O(老样本)和 N(新样本)进行循环操作。在每一个延时周期期间新样本都会被写入 N 所占据的位置。

固定延时效果(Fixed Delay Effects)

为了简单化，首先有必要先区分固定和可变延时效果。在固定延时单元中，当声音信号通过时，延时时间不发生变化。在可变延时单元中，延时时间则是不断地在变化；它是通过在每个样本周期变化读指针实现的。在此我们先讨论固定延时的情况，后面我们再介绍可变延时的内容。

固定的音频延时根据它们所产生的相应的听觉效果可以归结到三种时间范围内：

- 短延时(小于 10 毫秒)
- 中延时(大约 10 到 50 毫秒)
- 长延时(超过 50 毫秒)

短延时产生的听觉感受主要是体现在频率域的不规则效果。比如说，一个延时一个样本或几个样本长度的信号与原始信号混合，所产生的效果与 FIR 低通滤波器所产生的效果是一样的；当延时时间处于 0.1 到 10 毫秒之间时，此时将会出现梳状滤波器的效果。

中延时能够增强“单薄”的信号。比如说，中延时主要用于流行音乐中来增

强人声、鼓和合成器的声轨。中延时可以为信号产生一个“周围环境气氛”效果，从而给出一种感觉上的响度提升，而事实上相应的客观振幅并没有提升（请注意，这里响度是描述听觉的术语，而振幅则是描述物理的测量）。延时时间在10到50毫秒之间会与原始信号进行融合产生“复制（doubling）”效果，通过使用很少的时变调移和对信号在与原始信号混合之前的延时可以增强这种复制效果。

长延时（超过50毫秒）会产生离散的回声效果——声音听上去是原始声音的重复。在自然界中，从声源发出的声波，与反射面碰撞后反射回来，经过足够长的时间再次到达听音者，并使得听音者听到一个相对于原始声音离散的复制声音的现象就是回声。因为声音在20摄氏度的环境中每秒可以传播1 100英尺（344米），延时1毫秒意味着延时的声音与原始的声音到达听音者之间相差1英尺（0.3米）。为了产生离散的回声效果，需要至少50毫秒的延时时间，这意味着声源和听音者距离反射面至少25英尺（8米），或是声源从发出声音到经反射面反射后再到达听音者的总距离为50英尺（16米）左右，如图10.35所示。

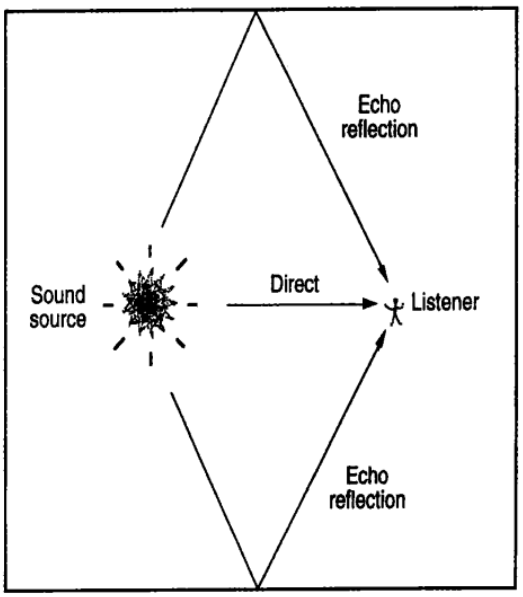


图10.35 由于直达声和反射声的混合而产生的回声效果。
Sound source=声源 Echo reflection=回声反射
Direct=直达声 Listener=听音者

延时和声音定位（Delays and Sound Localization）

声音定位（Localization）指的是通过人耳对一个发声声源位置进行判断的能力。在多声道系统中，延时可以起到声音定位提示的作用。举例来说，如果

两个扬声器被馈送入相同幅度的声音信号，同时听音者位于听声范围的中间位置，那么“声象”(“sound image”)会出现在听声范围的中间位置；如果一个短延时(0.2 到 10 毫秒)处理加入到右边扬声器的声音信号中，此时的声象会向左扬声器位置移动(Blauert 1983)。这就证明我们的耳朵能够使用延时信息来进行声源定位。通过实验证明，多重回声能够重建特殊空间中声源发声的声场印象。第 11 章对声音定位进行了详细的描述。

可变时间延时效果(Variable Time Delay Effects)

由于延时线中延时时间的变化导致通过它的音频信号产生的效果称为可变时间延时效果。其中两个最显著的效果就是镶边(flanging)和相变(phasing 或相移，phase shifting)，它们在 20 世纪 60 年代和 70 年代流行音乐中得到了广泛的应用。虽然技术上相似，但是所产生的效果却不同。

镶边(Flanging)

电子镶边效果来源于自然的声学现象，当一个宽带噪声的原始信号与其延时后的信号进行混合时就会产生镶边的听觉效果。Bilsen and Ritsma(1969)指出这种效果最早是由 Christian Huygens 在 1693 年发现的。吉他演奏家及录音革新家 Les Paul 是第一个将镶边效果作为声音效果用于录音室的人。他的 1945 镶边系统由两个开盘录音机组成，其中一个具有变速控制装置(Bode 1984)。在 20 世纪 60 年代，在录音室中采用两台模拟磁带录音机和调音台来实现镶边效果。磁带录音机被馈送入相同的信号，当两台录音机以相同的速度播放声音时，录音工程师用手放在其中一台录音机的带盘边缘，使其走带速度比另一台稍微滞后一些，然后监听两台录音机输出的混合信号(如图 10.36 所示)。之所以使用两台录音机是为了同步通过监听镶边录音机重放磁头所产生的整体延时。在 38cm/s 的磁带速度上，由于典型的模拟磁带录音机录音磁头和重放磁头之间的距离在磁带上所产生的固定延时大约为 35 毫秒(通过设定录音和重放磁头来产生准确的延时)。因此如图 10.36 所示左边的磁带录音机也可以看作是实现固定延时线的替代品。

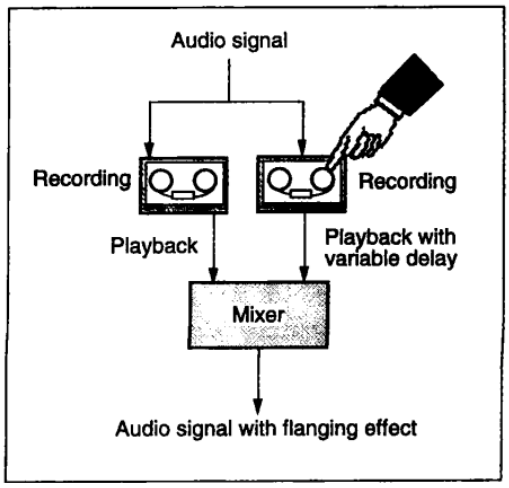


图 10.36 使用两台模拟磁带录音机实现的磁带振铃效果。通过手指对第二台磁带录音机磁带盘的按压造成其回放转速变化来实现的。
Audio signal=音频信号 Recording=录音 Playback=重放
Playback with variable delay=带有可变延时的重放 Mixer=调音台
Audio signal with flanging effect=带有镶边效果的音频信号

镶边总体规律如下描述：

镶边＝信号＋延时后的信号

这里延时时间是不断变化的。

电子镶边是通过使用连续变化的延时线来实现相同效果的(Factor and Katz 1972)。代替了盘带边缘的手工按压，通过工作范围为 0.1 到 20Hz 的低频振荡器(通常产生正弦波和三角波)来实现在电子镶边器中的延时时间变化。

镶边也被称为扫频式梳状滤波器效果(swept comb filter effect)。在镶边效果中，对频谱从上到下扫频时会出现一些听觉零点。滤波器峰值出现在 $1/D$ 的整数倍位置上，其中 D 为延时时间。如果原始信号的振幅与延时信号的振幅相等，此时镶边深度达到最大。

以上所描述的流程结构等同于一个前向或具有时变延时的 FIR 梳状滤波器。事实上，现在许多的镶边实现就是使用 IIR 或具有时变延时的递归反馈梳状流程结构来实现的，如图 10.37 所示。一般可以通过切换正负反馈进行比较来确定哪一种方式对于镶边是更有效的。

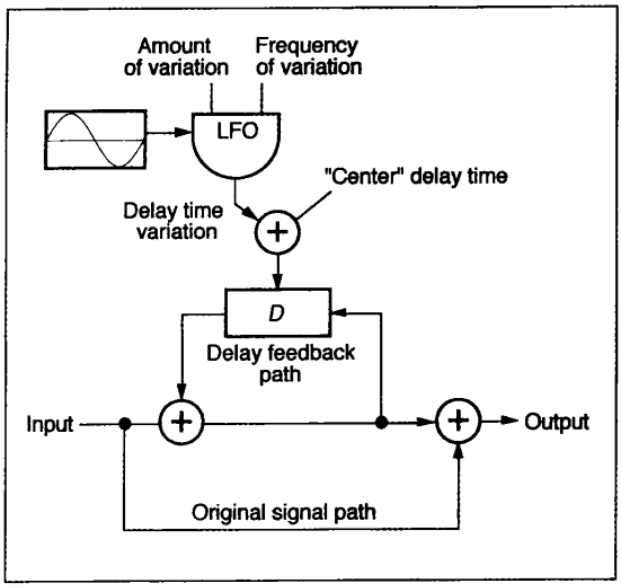


图 10.37　带有反馈回路的镶边流程示意图，将原始信号与延时后的信号进行混合实现。一个低频振荡器(LFO)用来提供在“中心”延时时间左右范围的延时时间变化。另外还可以在反馈回路中的反馈路径和原始信号路径上加入更多的乘法器来实现更复杂的效果，这样就可以调整两个信号之间的比例大小或实现反馈回路中的反相。
Amount of variation=变化量　　Frequency of variation=频率变化
Delay time variation=延时时间变化　　“center” delay time=“中心”延时时间
Delay feedback path=延时反馈通路　　Original signal path=原始信号通路
Input=输入　　Output=输出

相变(Phasing)

相变与镶边的效果类似，但是这种通过梳状滤波器扫频所产生的“搅拌”声却是镶边效果所不具备的。在相变效果中，一个频谱分量丰富的信号往往被送入到一系列的全通滤波器中(Hartman 1978，Beigel 1979，Smith 1984)，这些全通滤波器具有平坦的频率响应(也就是说在任何频率点上都没有衰减)但是它们会对信号的相位进行改变。一个低频振荡器能够被用来实现每一个滤波器所带来的相位改变量。滤波器的输出与原始信号按照统一的增益进行混合后，就像镶边一样，一系列的扫频梳状滤波效果就产生了。

那么在相变和镶边之间有什么不同呢？镶边效果导致在频谱上会出现完整的峰值和零点，并且它们以相同的频率间隔分布。相反，相变响应中的峰值和峰谷的个数与使用的滤波器的个数相对应，峰谷之间的间隔、深度以及宽度都是变化的。

相变会产生各种各样的声音效果，Chamberlin(1985)就曾经做过这样的一个试验，将一个 1kHz 正弦音调的信号送入到四个具有相同截止频率及相同转换带宽的全通滤波器组中，如果截止频率从 10 到 100Hz 扫频，音调就会产生连

续的相位移动，这时就会出现正弦波频率的暂时降低。如果截止频率反向扫频，此时会出现正弦波频率的暂时升高。如果正弦波信号被一个具有丰富谐波分量的信号所代替，由于这些谐波频率会随着截止频率扫频的变化而变化，这些暂时性的频率漂移会造成听觉“涟漪(ripple)”效果。

合唱效果(Chorus Effects)

对于合唱效果，音乐家和音频工程师一直在进行着研究探索，并且对它产生的效果具有浓厚的兴趣。当只有一件乐器进行声音演奏时(可以是任何的电声音色)，是否具有一种方法对这个信号进行处理从而产生这个声音的合唱效果呢？如果存在这种方法，那么这种效果还需要在这些模拟的各个声音中产生细微的差别，包括微小的延时、基波频率的细微变化(可以得到拍音效果)和不同步的揉音。因此对于合唱效果来说，并不能通过单一的一种算法来实现，往往是通过许多不同的算法集合来达到合唱效果。

合唱效果发生器的出现最早可以追溯到 20 世纪 40 年代的早期，当时 Hanert 将一些电动机械式延时线组合在一起应用于电子音乐创作中(Hanert 1944，1945，1946)。这些成果被集成在 Hammond organs(电子琴的一种)中并实现了合唱音调效果(choral tone effect)(Bode 1984)。到 20 世纪 50 年代韦恩(W. Wayne)制造了一种纯电子合唱音调调制器(choral tone modulator)用于 Baldwin 电子琴(Wayne 1961)。

在数字系统中，一种类型的合唱效果可以通过将信号送入到多抽头延时线中实现，其中的延时时间可以在一个较小的范围内连续变化，这些变化会造成失谐和时变复制效果。这就等同于将信号通过一系列的并行镶边器中，尽管在镶边器中的延时要比用于合唱效果的延时要小得多。

通过使用负反馈(将延时声音的反相信号返送回来)可以大大丰富这种技术的类型，比如使用镶边实现。这意味着如图 10.37 所示的镶边器的反馈回路中实现了相位翻转。负反馈相对于正反馈降低了谐振的风险和系统过载的风险。

另一种合唱效果技术是将输入信号按照倍频程带宽进行分割，然后将它们分别送入到不同的频谱或频率移位器中。频率移位器可以看作是对频谱中每一个分量的频率都加上一个常数。例如频率移位器的量是 10Hz，那么 220Hz 就变为 230Hz，440Hz 就变为 450Hz，880Hz 就变为 890Hz，以此类推。很明显，频率移位器破坏了各个分量之间的谐波关系。另外频率移位器的量是在一个很小的范围内随机变化的，因此频率移位器可以被看作是一个时变延时线。Chamberlin (1985)指出这种类型的设计用于模拟大型合唱效果是最佳的选择。

在并行使用一些全通滤波器时，通过使用低频准随机信号来改变滤波器的

截止频率也可以实现合唱的效果(Chamberlin 1985)。

变速/变调(Time/Pitch Changing)

一些声音转换过程中同时存在着时域和频域的合成处理。这包括一对具有关联性的处理技术,称为时间压缩/扩展(time compression/expansion)和变调(pitch shifting)。因为这两种处理技术常常同时使用,因此我们在这一部分同时对变速/变调(time/pitch changing)进行说明。这些处理技术中包含两个方面,一个是声音的持续时间发生拉伸或缩短而音调保持不变,即变速不变调;另一个是声音的音调可以上下变化而持续时间不发生变化,即变调不变速。

通过对选择的、与前后相连的素材进行变速/变调处理可以感受到非常明显的效果。为了保证原始声音的特性,在仅对信号的稳态部分进行处理时,其中最重要的就是保留完整的音头和瞬态特性。例如,对于拉伸语言信号来说,通过对元音的拉伸比对辅音的拉伸更加能够增强语言的可懂度和自然度。

不同等级的变速/变调可以通过以下的几种方式实现:颗粒时域技术、实时调谐、相位声码技术、小波变换和线性预测编码技术。下面的章节将对这些技术逐一进行解释。为了避免与本书中其他章节中的重复,这里仅对这些技术中最主要的部分进行介绍。

通过时间颗粒化实现变速/变调 (Time/Pitch Changing by Time-Granulation)

所谓时间颗粒化(time-granulation)指的是将音频样本流分解为持续时间很短的、称为颗粒(grains)单元的过程。它与许多声音分析算法中的加窗(windowing)处理是等效的(见第 13 章)。这些颗粒可以简单地通过对时间线上的样本流按照规则的空间连续间隔进行分段切割,或是从重叠的间隔和包络中抽取,目的是通过这些颗粒相加能够重建原来的波形。在时间颗粒处理中,每一个颗粒的持续时间可以在最短 1ms 到最长 200ms 甚至更长之间变化。在第 5 章中有关于颗粒细节的描述。

电动式时间颗粒化(Electromechanical Time-granulation)

英国物理学家 Dennis Gabor(1946)制造了世界上最早的一台电动式变速/变调效果器。一个德国公司 Springer 在磁带的基础上也制作了相似的设备用

于电子音乐演播室中(Springer 1955;Morawaska-Büngler 1988)。这个设备被叫做 Tempophon,并在 H. 艾默特(Herbert Eimert)1963 年的电子音乐作品《Aikichi Kuboyama 的墓志铭》(*Epitaph für Aikichi Kuboyama*, Wergo 60014)中用来处理人声。(见 Fairbanks、Everitt 与 Jaeger 1954 年关于相似设备的描述)。这些设备的基本原理都是对录制的声音进行时间颗粒化。通过这些早期工具的操作说明可以解释当时的数字化方式。

在电动式变速/变调效果器中,一个旋转磁头(采样磁头)旋转穿过声音录音介质(胶片或磁带)。采样磁头的旋转方向与磁带运行的方向一致,因为磁头只与穿过的磁带有很短时间的接触,因此,结果就好像是在磁带上对声音进行等间隔的"采样(sampling)",而这些采样的片断就是声音颗粒。

在 Gabor 的系统中,这些颗粒被重新组合成连续的音频流并送入到另一台录音机中。当这个第二台录音机对此音频流进行重放时,结果就产生了具有不同时基的基本连续的信号了。例如,通过降低采样磁头的旋转速度来达到缩短原始信号的持续时间。这意味着重新采样的样本中包含了一系列以前就被分割的颗粒,如图 10.38(a)所示。对于时间扩展,相应的通过增加采样磁头的旋转速度,采样得到原始信号的更多复制样本来实现。当这些样本被作为一个连续的信号进行重放时,这些复制样本就实现了延伸重放持续时间的效果,如图 10.38(b)所示。而原始音频样本中的频率分量,也就是音调,并没有在重建音频样本序列中发生变化。

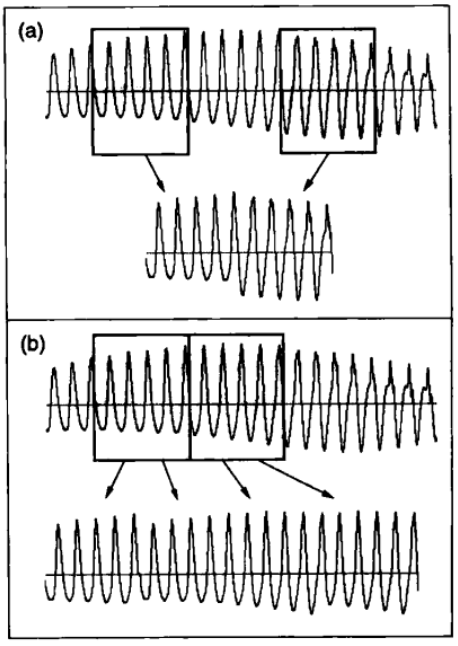


图 10.38 时间颗粒化。(a)通过对分离的颗粒进行抽取实现时间缩短;(b)通过对每个颗粒进行两次复制实现时间延展。在两种情况中,信号中的原始频率分量都得到了保留。

对于变调不变速的效果，目的是仅仅改变原始信号的重放速率和使用刚刚描述的时间缩放变化来调整信号的持续时间。比如说，对原始信号进行向上八度的移调，此时可以通过采用两倍的重放速度并且利用上面所讲的时间颗粒化来进行两倍时间扩展，这样来恢复原始信号的持续时间长度，实现变调不变速。

数字化时间颗粒化(Digital Time-granulation)

在伊利诺伊大学实验音乐工作室的开发研究小组最早进行了时间颗粒化的数字化实现(Otis，Grossman，and Cuomo 1968)。它通过模拟旋转磁头的采样效果来实现，同时也指出了在它最基本的形式中这种基本方法存在的缺点。其中最主要的问题就是采样样本颗粒波形的起始位置和结束位置与下一个连续样本颗粒相连接的位置可能存在电平不匹配的现象。这就导致了在连续颗粒连接位置出现瞬态跳变，效果如图 10.39 所示。正是由于这样的接合处瞬态跳变，在电动式和部分数字式时间颗粒化设备中都存在着周期咔咔声。

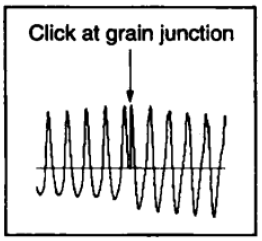


图 10.39　当两个颗粒任意接合时，前面颗粒的结束位置与后面颗粒的起始位置不能完全匹配，导致在接合处产生瞬时咔咔声。
Click at grain junction=颗粒连接处的咔咔声

Lee(1972)开发的 Lexicon“变语”(Lexicon Varispeech)系统是一个与模拟磁带录音机相连的数字时间压缩扩张器。Lee 的设计中采用了一个电路来实现接合位置的电平匹配用以减小咔咔声。Jones 与 Parks(1988)最近的研究成果也展示了如何使用略带重叠的平滑颗粒包络在颗粒与颗粒之间建立无缝过渡，来实现信号的平滑过渡重建。

正如电动式变调/变速效果器一样，对声音的持续时间进行加倍意味着每一个颗粒都将被复制成两个，为了减半持续时间，在重放之前必须对其中的一个复制品进行去除，这样就可以实现通过复制(扩展持续时间)或删除(压缩持续时间)颗粒来改变时间缩放时，保持颗粒中的频率分量内容不发生变化。

为了对采样信号进行一个八度的向上移调而不改变它的持续时间，此时重放采样速率加倍，每个颗粒被复制一次用来恢复原始信号的持续时间。为了对采样信号进行一个八度的向下移调而不改变它的持续时间，此时重放采样速率减半，颗粒也间隔删除用来恢复原始信号的持续时间。

以上我们讨论了对音调、时间加倍、减半的处理，但是这些处理并不仅仅局限于这两种方式中。对频率和时间缩放可以按任意比例进行变化，它是通过改

变相应比例的颗粒复制或删除的采样速率来实现的。

通过调谐器实现变速/变调(Time/Pitch Changing with a Harmonizer)

调谐器是一种实时转换设备,它可以实现对输入信号的变调不变速。Eventide H910 Harmonizer 开发于 20 世纪 70 年代中期,完全基于时域技术,是第一个面世的这种类型的数字设备(Bode 1984)。接下来将要进行描述的是在 20 世纪 80 年代早期,Publison 在法国开发的一个采样效果处理器,并由 Bloom(1985)加以改进。

调谐器的基本思路是通过将经过采样率为 SR_{in} 采样的输入信号放置到一个随机存储器中,然后以 SR_{out} 的采样率将信号读出来实现的。其中 SR_{in}/SR_{out} 的比值决定了音调的变化。

为了保持连续的输出信号,样本必须被重复(向上移调)或跃过(向下移调)。因为输出地址指针重复超过输入地址指针(对于向上移调)或被循环的输入地址指针超过(对于向下移调),输出地址必须间断性地跳到存储器中的一个新的位置。为了使得接合处咔咔声不可闻,对输入信号的周期性(音调)进行计算后,来精确地决定上述的跳跃。当确定了接合的位置之后,一个平滑的淡出包络将上一个接合的信号结束位置的振幅缓变为 0,相应的下一个接合的信号开始位置的振幅由一个淡入包络从 0 缓变为最大。

对于这种基本方案还可以增加一些细节来提升它的性能。其中一个是在系统输入部分增加一个噪声门,来避免与输入信号相关联的背景噪声不会在移调过程中也产生移调效果。

一个简单调谐器的声音质量取决于输入信号的自然特性和所要求处理而采用的移调的比例。小比例的移调处理会产生更少的边带效果。当使用一些临界素材,如人声等,一些商业设备在进行处理时,会产生一些不需要的边带效果(如在接合点的频率上产生嗡嗡声)。

通过相位声码器实现变速/变调 (Time/Pitch Changing with the Phase Vocoder)

相位声码器(PV)是一种通过快速傅里叶变换——通常采用重叠方式——对输入声音信号片断进行缩短。关于相位声码器进一步的细节描述可参考第 5 章、第 13 章和附录。使用 FFTs 可以产生一系列的频谱帧,这些频谱帧捕捉了声音信号在整个时间段中的频域变化过程。基于这些数据,原始的声音就可以使用加法合成来重新合成出来;每一个正弦波振荡器的频率都对应一个分析

后的频率分量。这种通过加法重新合成的结果往往被看作是原始信号的模拟产物。

重叠相加转换(Overlap-add Transformations)

相位声码器的主要构成就是通过将分析数据转换为重新合成所需的数据,从而产生原始信号的变化。其中一个最常用的转换就是时间压缩/扩展。它可以通过两种方式实现,取决于使用哪种类型的相位声码器。其中使用重叠相加进行重新合成(在第 13 章和附录中有详细描述)时,时间扩展是通过将用于重新合成的这些重叠帧开始的时间移动得更远来实现的,而时间压缩则是通过将这些重叠帧开始的时间移动得更近来实现的。正如 Dolson (1986)指出的,对于时间和音调的改变,相位声码器主要采用整数变换比例,而对于那些平滑变换,相位声码器应该与相位值进行相乘,其中相位值与时基变换中使用的常数相同(Arfib 1991)。

通过对重新合成中的频率进行缩放实现音调转换是比较简单的事情。尤其是对于语音信号,一个固定的缩放系数不仅可以改变音调同时也改变了共振峰频率。当向上移调一个八度或更高时,会明显地降低语音的可懂度。因此 Dolson(1986)建议采用频率缩放修正来对用于转换的频谱信号的原始频谱包络进行加强。比如说,如果原始信号频谱只到 5kHz,进行转换后会将 5kHz 以上的频谱信号切除掉,不管这些频率分量在整个的包络上如何的延展。

追踪相位声码器转换(Tracking Phase Vocoder Transformations)

另外一种对分析后的声音信号进行时基变化的方法是采用追踪相位声码器或 TPV(见第 13 章)。TPV 将一系列的频谱帧转化为一组振幅和频率包络方程用于每一个分析后的频率分量。这些方程在计算机存储器中以阵列的形式表示,通过编辑这些振幅和频率方程可以实现独立的移调或声音片断持续时间的扩展(Portnoff 1978, Holtzman 1980, Gordon and Strawn 1985)。比如进行时间的扩展可以通过在振幅和频率阵列中对当前存在的点与点之间内插新的点来实现。为了实现缩放因子 n 的持续时间变换,那么在幅度和频率阵列中每次只需读出第 n 个值即可。事实上这也会对采样频率进行改变(如图 10.40 所示)。Maher(1990)讨论了由于这种简单的内插所带来的一些畸变,同时提出对于这种“包络卷曲”的补偿措施。

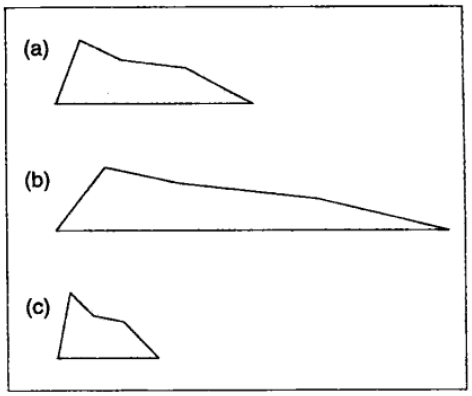


图 10.40 追踪相位声码器包络实现变速调整。其中垂直方向代表振幅大小，水平方向代表持续时间。(a)原始信号；(b)时间拉伸；(c)时间紧缩。

为了只变调不变速，那么信号就必须与频率值相乘，频率值通过需要的缩放因子来为每一个频率方程进行设定。比如说，将一个声音向上移调一个大二度，那么每个频率分量都应该乘以百分之 11.892，这样 1kHz 就变成了 1 189.2Hz了。当然也可以进行有选择性的移调，如通过只改变基波频率而不改变其他频率分量来实现。

通过小波变换实现变速/变调 (Time/Pitch Changing with the Wavelet Transform)

正如前面在相位声码器中描述的一样，通过小波变换实现变速/变调的第一步也是进行一种类型的频谱分析(Kronland-Martinet 1988；Kronland-Martinet and Grossmann 1991；Vetterli 1992)。在第 13 章对小波变换的基本概念进行了解释。小波变换与用于 FFTs 中的加窗片断相类似，但是在小波变换中每一个小波的持续时间是由它的频率分量所决定的：频率越高，小波对应的持续时间就越短。这意味着小波变换中时间分辨率在高频时更大(意味着它具有准确定位事件起始时间的能力)。

与傅里叶变换方式一样，小波变换也是将声音样本按照时间定位，分隔成为一系列独立的分量。这些分量通过分析获取到的幅度和相位值进行分类。为了调整音调和时基，在重新合成之前必须要改变这些分析数据。

以固定比例因子移调时，必须乘以一个相位值，这个相位值是根据这个固定比例因子来对小波进行分析后得到的(Kronland-Martinet and Grossmann 1991)。为了拉伸或缩短时基而保持音调不变，在重新合成时，对小波中重叠的点进行拉伸和压缩即可。

通过线性预测编码实现变速/变调
(Time/Pitch Changing with Linear Predictive Coding)

在第 5 章介绍了线性预测编码(linear predictive coding,LPC)———一种通过减法分析/重新合成来产生语言、歌唱、类似乐器音色及人工合成的声音(Cann 1979-1980,Moorer 1979a,Dodge and Jerse 1985,Dodge 1989,Lansky 1989,Lansky and Steiglitz1981)。LPC 分析将输入信号模拟为一个激发函数(excitation funtion)(比如人的声带、簧片或琴弦产生的)和一系列时变共鸣(time-varying resonances)(例如人的声道、萨克斯的管体或小提琴箱)。这些共鸣由模拟激励响应的时变滤波器来完成。关于更多的 LPC 频谱分析的内容,见第 13 章。

LPC 并不是一种完美的分析/重新合成的方式。它最初被设计作为一种有效的语音信号编码方案,来完成低带宽情况下的通信要求。用于音乐目的是它功能的扩展,但是由于在分析过程中细节的丢失,使得重新合成的声音总带有人为的特点(Moorer 1979a)。如果这种不足能够被接受,那么 LPC 将可以并且被继续用于大量的作曲创作中去。

LPC 对分析结果进行编码,这些分析结果由一序列持续时间短的帧构成,每一帧中都获取了给定时间声音片断中的滤波系数、音调和语音/非语音数据。参见第 5 章中对帧数据的解释。用于音乐应用时,作曲家对这些帧进行编辑,然后将它们转换为原来的声音。第 5 章中的图 5.38 显示出了在 LPC 帧中的序列数据。

为了实现变速/变调,对这些帧进行编辑然后通过使用这些编辑后的帧来进行重新合成。LPC 中分析的帧往往以规则的间隔进行计算,从每秒 50 到 120 次。例如通过编辑指令,可以对帧的持续时间进行变化,来实现单帧持续时间从 10 到 100 毫秒之间的扩展。在重新合成时也可以仅仅对音调柱进行编辑来独立实现音调的改变。因此持续时间和音调可以独立地被改变。除变速/变调之外,LPC 数据还可以采用其他方法进行编辑来产生原始分析后的声音中剧烈的变化。(见 Cann1979-1980 年与 Dodge1985 年对 LPC 数据的编辑实例)。通过 LPC 实现的变速/变调的音乐应用也可以在保罗·兰斯基(Paul Lansky)和查尔斯·道奇(Charles Dodge)的作品中找到实例。

结论(Conclusion)

近些年来,对信号处理的研究一直持续不断地发展,大量相关书籍层出不

穷。紧跟技术发展步伐的方法就是经常阅读浏览一些期刊杂志，如《IEEE 信号处理学报》(*IEEE Transactions on Signal Processing*)、《音频工程学会学报》(*Journal of the Audio Engineering Society*)以及《计算机音乐杂志》(*Computer Music Journal*)。

当前的信号处理技术与目前可利用的技术紧密联系。更小、运算速度更快、价格更低廉的系统被不断地部署，这意味着以前被大机构采用的方式也被个人工作站或被现场表演所使用。

大多数信号处理都关注于整体的操作，而很少考虑被处理声音本身的特性。然而，与之相反的趋势，是信号响应算法的崛起。一个典型的例子是延长稳态信号但保留未触及的瞬时信号的时间延展算法；另一个是将输入信号延迟并向前预测宽幅振荡的压缩算法。当分析与信号处理融为一体时，可以想象精密而复杂的信号响应算法的应用将更加普遍。

第 11 章　声音空间化和混响
(Sound Spatialization and Reverberation)

今天的声音空间艺术呈现出同 19 世纪的管弦乐作曲艺术相似的地位。配置空间关系就是安排、规划声音：定位声源和声音的运动。给声音加上混响，我们会令听众陶醉于美妙的境界中。

声音的空间化包括两个方面：虚拟的和现实的。在工作室的虚拟现实空间中，作曲家们通过利用延迟、滤波器、调整相位和混响等手段使声音空间化——从虚拟的环境中产生声音的幻觉。有时这些虚拟空间呈现出一些从建筑学上无法实现的特性，比如一个连续变化的回声模式。在音乐厅的现实环境中，声音可以被一个多声道的声音系统从多个位置重放出来：四周、上方、下方以及观众当中。

声音的结构或空间化已经逐渐成为作曲的一个重要元素。在被近距离扩音和那些远距离混响的声音中的戏剧化的并置方面起重要作用的乐曲中，可以看到电影对声音空间的应用趋势。一些作曲家在麦克风的运用技术上，采用一种类似于电影摄像角度、镜头透视（广度）和场景深度，并且对声音进行空间处理。这让我们想起让·克劳德·瑞塞特的《萨德》（*Sud*）（1985，Wergo 2013-50）。

这一章从简述声音在三维空间中的分布开始，接着描述了数字混响艺术——这是在今后的几年中将继续被拓展的空间化领域。最后，通过综述模拟特定空间环境方面的研究扩展了前面的讨论。建议在阅读本章前，了解并熟悉第 5 章和第 10 章中介绍的滤波器部分的内容。

声音的空间化（Sound Spatialization）

声音的运动利用空间创造出富有表现力的效果，这可以成为作曲中一个重要的建构元素。作曲家通过赋予每个声音一个特定的空间位置，能清晰地给出每个声音的空间定位。听众周围真实的、物理的声音舞台可以被看成是一个场景，有背景和前景，有静止的和运动的音源。这个声音舞台可在回放声音时建立或是由音乐会中的指挥掌控（Harada et al. 1992）。

运动声音源的数字仿真提出了特殊的问题。在很多音乐会上，观众被许多扬声器所围绕。如何令听众产生声音在大厅移动、远离或逼近自己的真实的幻觉呢？在只有两个扬声器或耳机的欣赏条件下，产生声源在空间中自由运动的幻觉就更加困难。

最常见的空间幻觉是水平声象移动——从扬声器到扬声器的横向的声象移动——以及混响——为一个声音添加一个密集的、散开的回声模式，使之听上去像是处于一个更大的空间中。竖向声象平移（上下来回的移动）也能在电子音乐中创造出显著的效果。（见 1973 年 Gerzon 关于“有高度的声音”的录制和回放的讨论。）

音乐的空间化：背景(Spatialization in Music: Background)

“无论是否在一边，使用多少个扬声器，是否左右转动，固定还是运动，声音和声音组应怎样分布到空间中去，所有这些对于了解这个作品来说都是决定性的。”(卡尔海茵茨·施托克豪森(Karlheinz Stockhausen)于1958年描述他的乐曲《青年之歌》(*Song of the Youths*)

音乐作曲中的空间化技术已不是什么新技术了。在16世纪，与威尼斯的圣马可大教堂有关系的作曲家(特别是阿德里安·维拉尔特和他的学生加布里埃利)的作品中在两个或多个教堂音乐中采用了空间唱和。在这些作品中，最初的唱诗声是从大厅的一边传来，而后另一边传来呼应的唱诗。方形教堂中两个相对的管风琴促进了这个布局。沃尔夫冈·莫扎特曾为两个空间上分离的管弦乐队写了乐曲(K. 239和K. 286)，而埃克托·柏辽兹和古斯塔夫·马勒曾为多个管弦乐队和合唱队(其中一些乐队与合唱队被置于舞台后面)创作乐曲。尽管有过这些实验，关于乐曲中空间化技术的资料却是直到电子时代才有的。

扬声器的发明可以与电灯泡的发明相媲美。突然之间，我们可在大大小小的空间中以任何角度或强度发出声波能量。但是扬声器的使用——在剧院、体育场、火车站以及家用收音机上——在清晰度和功能性上做得还不够。直到第二次世界大战后期在电子音乐中，才开始对通过扬声器播放的声音的审美特性进行挖掘。

电子音乐中空间化的实例
(Examples of Spatialization in Electronic Music)

有很多关于电子音乐和计算机音乐空间化项目的著名例子值得在这里列举。

■ 卡尔海茵茨·施托克豪森(Karlheinz Stockhausen)的《青年之歌》于1956年在一场音乐会上被播出，在当时西德广播电台的听众席上有超过五组的扬声器(Stockhausen 1961)。他完成于1961年的作品《接触》是第一部从四声道磁带里播放出的电子音乐作品，是用德律风根(Telefunken)T9型磁带录音机录制的(Stockhausen 1968)。

■ 1958年，埃德加德·瓦雷兹(Edgard Varèse)著名的磁带录制的音乐作品《电子音诗》和伊安尼斯·克赛纳基斯(Iannis Xenakis)的《具象PH》，都是通过一个安装在菲利浦展厅曲面墙上的11声道音响系统的425个扬声器播

放出的，这是在布鲁塞尔世界展览会上由克赛纳基斯和勒科尔比西耶（Le Corbusier）设计的。

■ 在第 70 届大阪世界博览会上，施托克豪森通过分布在德国展团的网格球顶内表面上的扬声器播放他的电子音乐（Stockhausen 1971）。

■ 在同一个博览会上，伊安尼斯·克赛纳基斯在日本钢铁展团里，用一套分布于听众周围、头顶及座位下的有 800 个扬声器的系统，演示了他 12 声道的电声乐曲《希比基-哈纳玛》（Matossian，1986）。一套 12 声道的音响播放系统，使他在巴黎旧克鲁尼博物馆内放映的声光展示作品《克吕尼的建筑多面体》（Xenakis 1992）充满生机。

■ 作曲家塞尔瓦托·马尔蒂拉诺创建了一套叫做“萨尔-玛（Sal-Mar）结构”的复杂的数字设备，用它去控制一个自定义模拟音响合成器，并将声音从音乐厅的天花板上悬下的处于多种不同高度的 250 个薄扬声器发出（Martirano 1971）。

■ 在一个在舞台上有很多扬声器的管弦乐队上方分布声音的想法在“GMEB（布尔日实验音乐小组）失音患者”中得以实现，并在 1973 年的音乐会上首次被听到（Clozier 1993）。

■ 洪荒之声（Acousmonium）的第一场音乐会——集合了多种由布尔日实验音乐组设想的“声音投射器”（图 11.1）——在 1974 年巴黎 Espace Cardin 艺术中心举行（Bayle 1989，1993）。

■ 在 20 世纪 80 年代中期彼埃尔·布莱兹（Pierre Boulez）的作品《应答圣歌》中使用钢构架将扬声器置于观众头顶上方的演出中，使用乔瓦尼·迪朱尼奥 4 倍的电子合成器实现了对空间的控制（Asta et al. 1980；Boulez and Gerzso 1988）。

■ 在 1987 年，研究者在佛罗伦萨卢恰诺·贝里奥（Luciano Berio）的坦普锐耳（Tempo Reale）工作室开发了一个叫做“Trails”的基于计算机的声音分布系统，这个系统发出的声音能够达到 32 个声道，混合了预编程和实时的空间模式（Bernardini and Otto 1989）。

还有很多其他的声音空间化系统也被开发出来了，包括爱德华·科布兰的 16 声道“混合”IV 型合成器（Kobrin 1977）（图 11.2），SSSP 声音分配系统（Federkow，Buxton，and Smith 1978），AUDIUM 装置（Loy 1985b），Hans Peter Haller 的被彼埃尔·布莱兹和路易吉·诺诺使用的 Halaphon（Haller 1980），由里昂的格拉姆（GRAME）工作室开发的用计算机控制的“交响曲”空间效果器，以及在斯坦福大学由博西（Marina Bosi）（1990）实现的全数字空间化设备等。

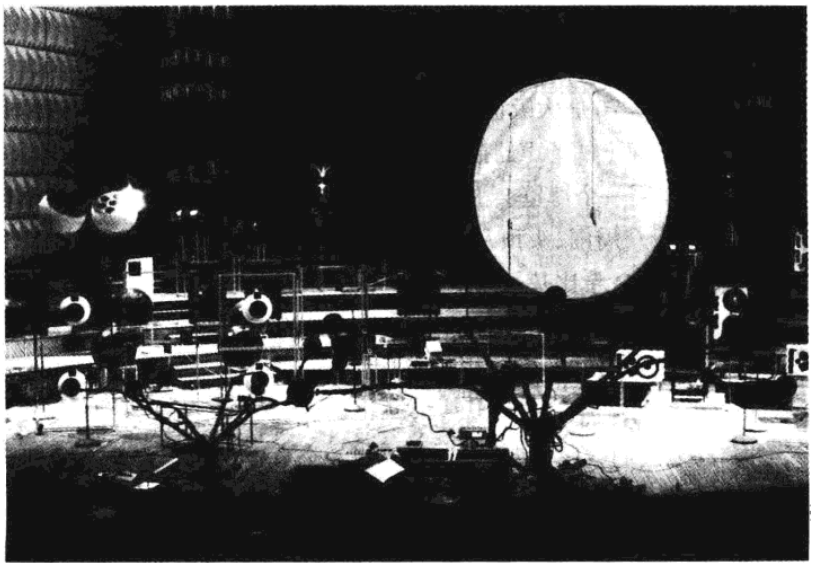


图 11.1 “洪荒之声”——一个由音乐研究小组(GRM,一个以自然声音和工业声音合成电子乐的始祖团体)所设计的多声道空间效果演奏装置——1980 年安装在位于巴黎的法国国家广播电台奥利维尔·梅西昂音乐大厅里。用 80 个扬声器发出声音,这些扬声器发出的声音都来自一个 48 声道的混频器,“洪荒之声”实现了可与一支管弦乐队相媲美的复杂声象。它令一个作曲家为“洪荒之声”系统空间电子乐曲演奏而“重写管弦乐谱”。(由 L. Ruszka 拍摄,由弗朗索瓦·贝勒和 GRM 提供。)

在演奏中增强空间分布(Enhancing Spatial Projection in Performance)

即使没有特别精确的声音分布系统,电声音乐的音乐厅仍可通过努力增强演奏的空间质量。如图 11.3 所示的一些标准的配置。

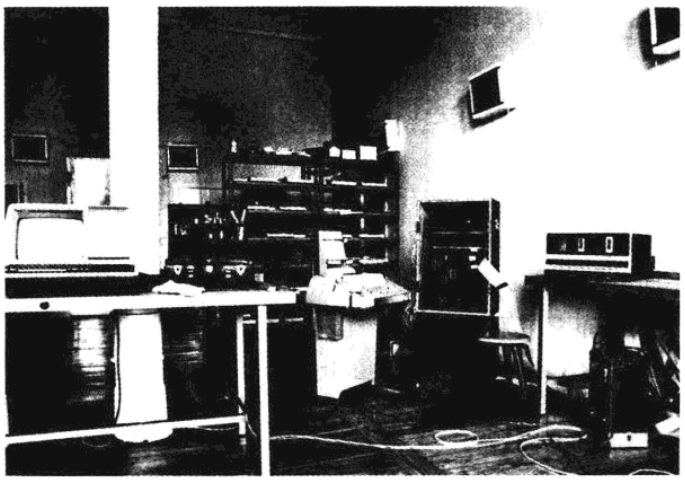


图 11.2 Edward Kobrin 的“混合”IV 型合成器工作室,1977 年建于柏林,是一个由计算机控制的 16 声道空间化系统。扬声器安装在墙上。

1. 条件允许时，使用至少一个四向声的声音系统（用四个扬声器的四声道放大系统），置于听众的四周[图 11.3(b)]。

2. 当在这个四向声的系统上播放双声道录音带时，前后两对声道接线时，后面一对的左右与前面的相反。这样，当一个声音在前方从左边平移到右边时，它在后面也从右边平移到左边，增加了空间的生动感。

3. 为了增加更大的空间清晰度，可将扬声器放在相对角落中一个较高的位置处。这叫做全向声或是“有高度的声音”回放（Gerzon 1973）。当一个声音从左边平移到右边时，用这样的配置同样还会产生竖向方向的移动[图 11.3(c)]。

4. 当有放大的乐器声或是演唱声时，将独立的放大器和扬声器配件连同那些使得特定乐器更清楚的效果（例如均衡音量）一起提供给每一个表演者。为使声音舞台中的每一个乐器在原地，并避免声音与表演者分离，扬声器应该放在表演者的旁边（Morrill 1981b）。在声音与表演者分离时，乐器的声音提供给远离表演者的总的声音增强系统。因为听众对声音源的想象是由第一个到达他们耳朵的声音所支配的（这就是“优先效应”：Durlach and Colburn1978），对一个表演者在演奏声学乐器时所做的任何全局放大，都要被延迟 5 到 40 毫秒，以使局部放大器作为声音源形成第一印象（Vidolin 1993）。（当然，有时作曲家想要使一种乐器的声音环绕整个大厅，或是将它合并到一个事先录制好的声音源中，那是另一种情况了。）

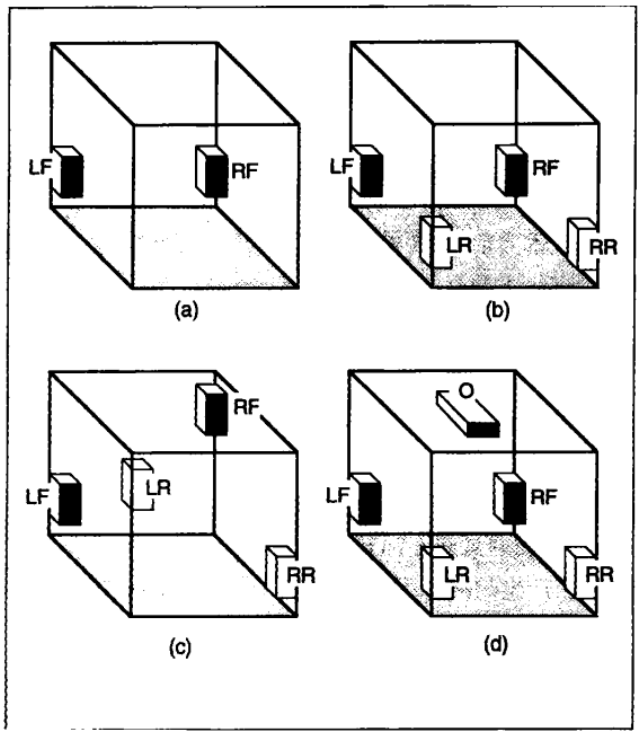


图 11.3　为电子音乐和计算机音乐的空间化所选择的扬声器配置。(a)基本立体声，LF＝左前，RF＝右前；(b)四声道的，RR＝右后，LR＝左后；(c)四声道全向声。右前和左后的扬声器置于人耳高度以上，这样使得声音在水平方向平移的同时也在竖向方向上平移；(d)五扬声器的配置中有一个向下发声的竖向扬声器。

5. 另一种不同的方法是在台上装配一个由不同扬声器组成的“管弦乐队”(Gmebaphone/“洪荒之声”所使用的方法)。这种方法实现了一个空间源的多样性和与声学乐器的管弦乐队有关的音速差异。

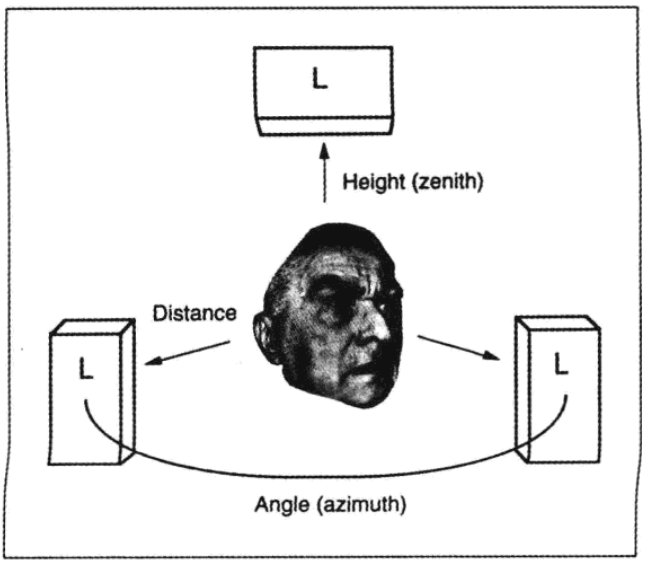


图 11.4 专心的听众能够通过水平角度、高度和距离等暗示信号来定位一个声音源。
L=扬声器 Height(zenith)=高度(顶点) Distance=距离 Angle(azimuth)=角度(方位)

精确控制空间上的幻觉需要了解定位理论——人们如何感觉到声音的方向,这是下一部分的主题。

定位暗示(Localization Cues)

在钻研声音空间化技术之前,重要的是了解听众如何弄清声音是从现场哪里发出来的基本原理。这一主题是心理声学的一个已被广泛研究的领域,被称为声音定位。而定位是需要三维暗示信号的(图 11.4;定位),即

方位或水平角暗示信号;

距离(对固定声源来说)或速度(对运动声音来说)暗示信号;

顶点(高度)或竖向角暗示信号。

要决定一个声音的方位,听众要用到三个暗示信号:

- 当声音来自一边时,到两耳的不同到达时间
- 由头部“阴影效应”产生的两耳听到的高频声音在振幅上的区别
- 声音从外耳(耳廓)、肩膀和胸部发出不对称的反射而产生的频谱暗示

距离的暗示有三个:

- 当直达信号的强度按距离的平方衰减时,直达信号与混响信号的比率

- 随着距离的增加，高频成分的损失
- 随着距离的增加，细节（更细微的声音的缺少）的损失

当声音和听众之间的距离改变的时候，声音速度的暗示表现为音调的改变，这便称为多普勒频移效应（Dopller shift effect），后面会解释。

顶点的主要暗示是一个由离开耳廓和肩膀的声音反射产生的频谱的变化。

模拟方位暗示（Simulating the Azimuth Cue）

人类很容易定位一个头部高度发出的强高频声响。由单扬声器扩音的声源，所有的声音信号都应发自那个扬声器。但当这个音源从一个扬声器转移到另一个时，目的扬声器方向上的振幅增大了，同时在最初的源扬声器方向的振幅减弱了。

在有很多扬声器被等距地摆放为一个环绕在观众四周的圆形演奏中，无论扬声器的总数量如何，一个空间位置的算法仅需要计算在某一时刻相邻两个扬声器的振幅。要在两个扬声器 A 和 B 之间一个精确的点 P 定位一个声音源，首先应找到从 AB 的中点测量到的声源角（θ）（图 11.5）。

可能会出现很多不同的平移曲线，每一条曲线显示出了声音运动的略有差别的空间形象。下面我们来讨论两条平移曲线：线性功率和恒定功率。对于一个均匀的平移来说，这些曲线假定了一个听众坐在两个扬声器的中点。

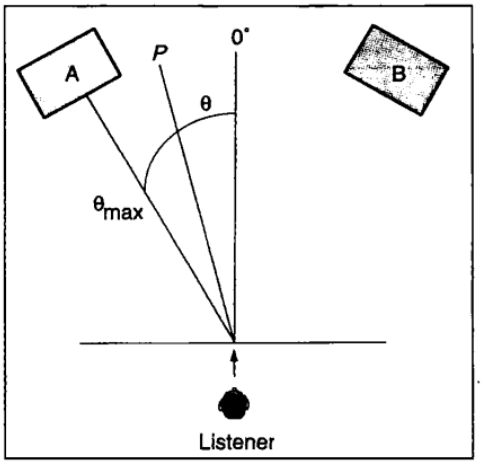


图 11.5　要在两个扬声器 A 和 B 之间的点 P 处定位一个声音源，应确定从 A 和 B 的中点测量到的声源角（θ）。在中间时 θ 等于 0 度。角 θ_{max} 是最大角，一般来说是正负 45 度。使用本文给出的公式可得出发送到两个扬声器的信号振幅。
Listener=听众

当听众坐在偏离中心的位置时，就会在声象上有一个方位偏移量。对功率来说，仅需要使用 θ 值的索引进行查表操作，曲线便能够预先计算出来了。

线性声象平移(Linear Panning)

最简单的定位公式是一个简单的线性关系：

$$A_{amp}=\theta/\theta_{max}$$
$$B_{amp}=1-(\theta-\theta_{max})$$

这种平移类型的问题在于它产生了一个“中间的洞”的现象，因为耳朵趋向于听到终点(扬声器)的信号要比中间位置的信号更强(图 11.6)。这是由声音强度法则规定的，它指出，感觉到的声音的响度是与它的强度成正比的。声音的强度如下所给出：

$$I=\sqrt{A_{amp}^2+B_{amp}^2}$$

在转移的中心处(如，当 $\theta=0$ 时，$A_{amp}=B_{amp}=0.5$)，这变成了

$$\sqrt{0.5^2+0.5^2}=\sqrt{0.25+0.25}=\sqrt{0.5}=0.707$$

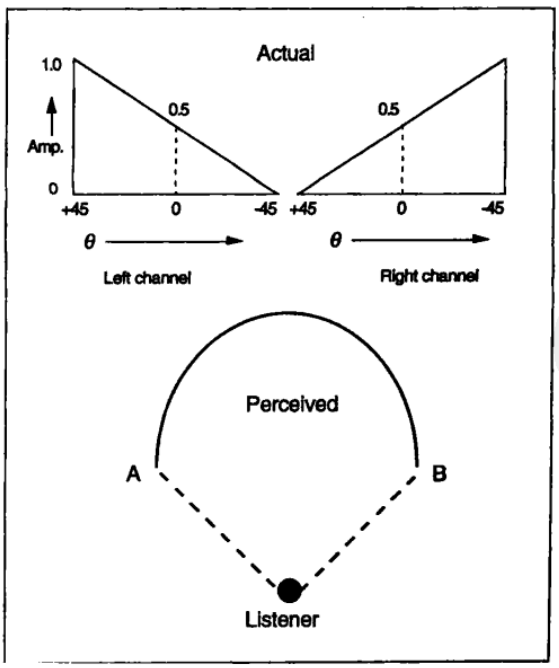


图 11.6 由于强度的减弱而感觉线性平移曲线是在中间后退的。图的上端所示的是每个声道的振幅曲线，下端所示的是感觉的声音轨迹。
Actual=实际的声音
Left channel=左声道
Right channel=右声道
Perceived=感觉到的声音
Listener=听众

因此强度从大小为 1 的边上的起点开始，到中间部分就降至 0.707。这就是 3dB 的差别。对人耳来说，对强度比对振幅更敏感，对中心位置的声音的感觉似乎就更微弱，就像是声音已远离听众一样。

恒定功率声象平移(Constant Power Panning)

恒定功率的声象平移按正弦曲线控制从两个扬声器发射出的声音振幅(Reveillon 1994)。这就产生了以更恒定的响度平移的感觉。

$$A_{amp}=\frac{\sqrt{2}}{2}\times(\cos\theta+\sin(\theta))$$

$$B_{amp}=\frac{\sqrt{2}}{2}\times(\cos\theta-\sin(\theta))$$

这个平移的中间位置，$A_{amp}=B_{amp}=0.707$，因此

$$I=\sqrt{0.707^2+0.707^2}=\sqrt{0.5+0.5}=\sqrt{1}=1$$

然后一个恒定的强度就被保留了下来。

图 11.7 表示的是恒定强度的声象平移。感觉上的平移看上去像是在两个扬声器绕听众以恒定的距离旋转。

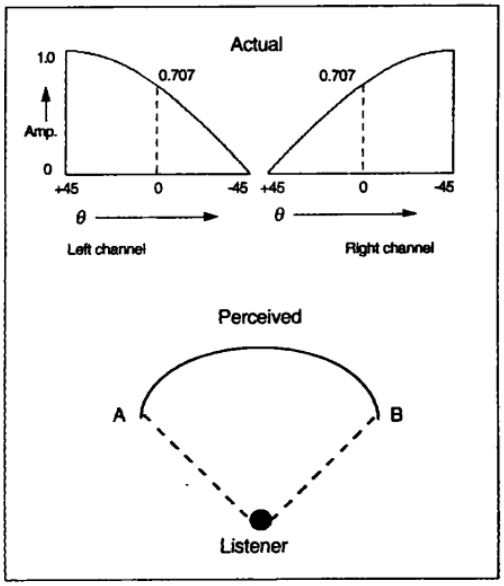


图 11.7　一条恒定功率平移曲线维持了感觉上的距离和中间部分的强度。每一个声道的振幅曲线见图的上半部分所示；感觉上的声音轨迹见图的下半部分所示。
Actual=实际的声音
Left channel=左声道
Right channel=右声道
Perceived=感觉到的声音
Listener=听众

反射(Reflections)

当声音在音乐厅里从一个扬声器平移到另一个扬声器时,在大厅里产生的反射为声音源的定位提供了更多的暗示信号。(一些大厅里的特定位置会混淆方向感,但这是很个别的情形。)因此,为增强定位的效果,作曲家可对来自于"无方向"的声道(例如,来自没被发射出的主声源的声道)的信号添加一点延迟。这些声道延迟模拟了大厅的反射;它们告诉耳朵声音源的方向在别处。理想状态下,反射模式应随着声音的平移而改变。

表 11.1 每个单位时间里声波传播的距离

时间 (单位:毫秒)	总距离 (单位:米)	波长频率 (单位:赫兹)
1.0	0.34	1000
3.4	1	340
6.8	2	168
34	10	34
68	20	16.8
100	34	10
340	100	3.4
680	200	1.68
1000	340	1

注意:上表还列出了相应的波长。用上表相应的"总距离"作为音源到反射表面到听众的距离,来计算一次反射所延迟的时间。设声音的速度为 340 米/秒。

延迟的时间和声音在感觉上的距离之间的关系,参见表 11.1。表中为声音在每个单位时间内移动的距离。为了严谨起见,在表 11.1 中加入了第三栏,表示与所给距离相应的波长。例如,像第三行所示,一个 168Hz 的声调(大约是 E 音)在空间中两米处成形。

模拟距离暗示(Simulating Distance Cues)

为使声音后退,可以降低它的振幅、使用低通滤波器、添加回声、或混响等。前两个暗示模拟发生在一个大的户外场地,能让我们通过声音的强度和在高频率下空气吸收的滤除效果来感觉声音的距离。

回声和反射暗示模型模拟了类似音乐厅的封闭空间中的情形。为模拟在一个房间内的准确距离,最简单的方法是保持反射常数的值不变,然后测量与

想得到的距离成反比的直达信号(图 11.8)。这个技术的延伸是根据比直达信号衰减更慢的函数缩小/放大反射信号。

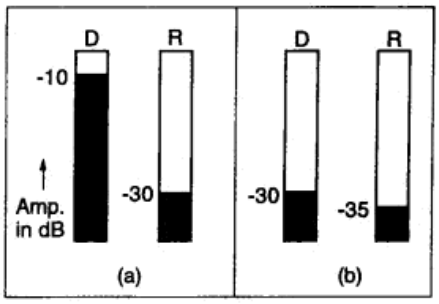


图 11.8　模拟一个向远离听众方向移动的声音的标准指示器;D=直达的,R=反射的;(a)近处的声音,其中直达的声音比混响声音振幅高得多;(b)远处的声音,其总体的振幅更低,直射声音与反射声音之间的比率降低了。
Amp.=振幅　　in dB=单位: 分贝

局部混响和全局混响(Local and Global Reververation)

另一个距离暗示是局部混响与全局混响间的关系,这可用一套多扬声器的系统演示出来。全局混响由所有的扬声器均等发出,而局部混响则是在相邻的两个扬声器之间进行的。因此,一个声音会同时拥有短而弱的全局混响,和来自多扬声器设备中的一组扬声器的长而强的局部混响。这会模拟两个扬声器之间一个大空间的情形。

局部混响和全局混响之间的区别在于能够消除在直达信号和全局混响信号的振幅相等的距离上所产生的掩蔽效应。这个掩蔽效应消除了方位暗示。消除这个效应的一个方法是将混响分为局部和全局的分量,并根据如下关系使局部混响同距离一起增加:

$$local\_reverberation \cong 1-(1/distance)$$

局部混响≅1−(1/距离)。

随着距离的增加,这个关系便趋向于 1。因此,当声音源与听众的距离很近时,所有声道发出的混响分布都同样好。随着声音源的移走,混响信号集中到声音源的方向。

速度暗示或多普勒频移效应(The Velocity Cue or Doppler Shift)

对静止声音的基本定位暗示可被扩展应用到对运动声音源的模拟上。这

是通过声音源的速度这一暗示实现的，也就是“多普勒频移”，它最初是由物理学家多普勒(C. Doppler)提出的(1842)。第一次对计算机音乐进行多普勒频移的模拟是由乔宁(John Chowning)完成的(1971)。

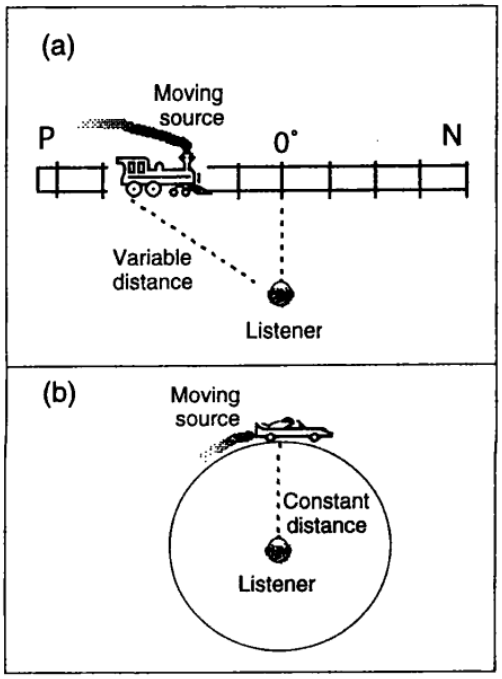


图 11.9 一个向听众运动的声音具有正的(P)径向速度。远离的声音有负的(N)径向速度。(b)声源做圆周运动时与听众的距离始终不变，并且有大小为零的径向速度。
Moving source=运动的声源 Variable distance=变化的距离 Listener=听众
Constant distance=恒定的距离

多普勒频移是在声音源和听众二者相对运动时所产生的音调的改变。一个通俗的例子是随着一辆火车高速驶来然后开过去，站在火车道旁边所听到的声音。随着火车的驶近，声音的波阵面很快便到了我们所在的位置，使得音调升高；而当火车驶过后，我们听到的音调又变低了。

多普勒频移是音源相对于听众的径向速度的暗示。径向运动是相对中心点—听众的运动[图 11.9(a)]。径向速度不同于角速度，一个声音如果要获得一个角速度，它必须是以听众为圆心做圆周运动[图 11.9(b)]。这样从音源到听众的距离是恒定不变的(也就是说，径向速度为零)，所以不存在多普勒效应。如果听众的位置保持固定，那么多普勒频移效应可按下式表达：

$$new\_pitch = original\_pitch \times [vsound/(vsound - vsource)]$$

新音调＝原音调×[声速/(声速－源速)]

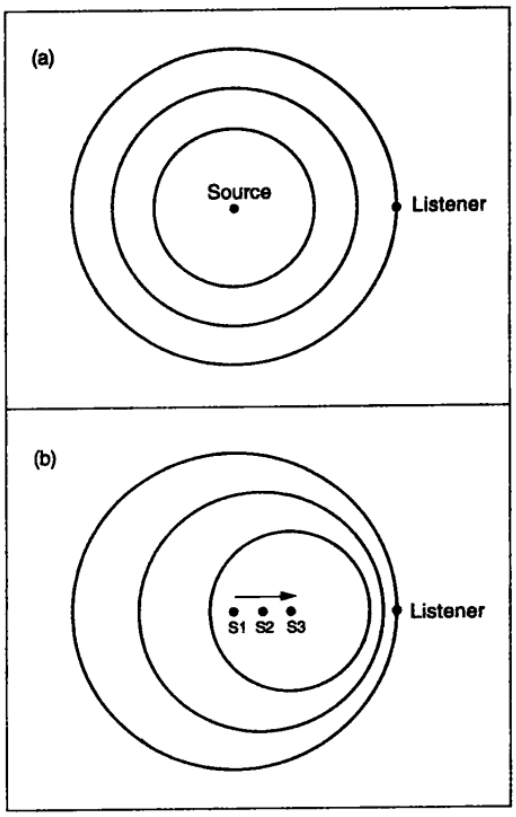


图 11.10　多普勒频移波阵面模式。(a)静态声音，波阵面以恒定间隔到达，所以不存在音调的改变；(b)S1、S2 和 S3 代表一个移动的声源连续的位置，音调频移上升。
Source=音源　Listener=听众

原音调是声音源的原始基准音调，声速是声音的速度(约 344 米/秒或 1 100英尺/秒)，源速是声音源相对于听众的速度。如果源速为正，则声音是向听众移近的，音调的频移是提高的；而当它的值为负时，音调的频移是下降的。

多普勒频移中产生的音调的改变可被解释为随着声音源向听众移近，波阵面之间间隔的缩短。图 11.10(a)描述了一个静态声音以一个恒定率或是恒定音调发出波阵面。图 11.10(b)表示的是向听众移近的声音源。点 S1、S2 和 S3 代表一个运动的声音源连续的位置。随着声音的临近，波阵面也聚拢得更近，产生一个提高的音调变化。

在一个给定时刻，多普勒效应按相同的对数间隔移动所有的频率。例如，一个以 20 米/秒(大约 45 千米/小时)的速度移动过来的声音，被提升了大约一个小二度的音程(百分之 6.15)。这百分之 6.15 的偏移对一个 10kHz 的成分来说是 615Hz，而一个百分之 6.15 的偏移对一个 100Hz 的成分来说

仅为6.15Hz。因此多普勒效应在一段声音中保留了以对数缩放的内谐波关系。这与调制中产生的线性频率偏移相反。一个线性频率偏移的例子是给所有成分都增加 50Hz。一个从 100Hz 偏移到 150Hz 的音调形成了稍大于纯五度的音程,但是在 10kHz 的范围内,50Hz 的偏移是很难察觉到的。线性频率偏移在一段声音中破坏了原有的内谐波关系。(见第 6 章)

模拟高度(顶点)暗示[Simulating Altitude(Zenith)Cues]

音源从高处降下来的效果会是很富有表现力的。自 20 世纪 70 年代,人们发现置于人耳高度的常规立体声系统即可形成竖向的声音幻觉。这一发现促进了商业上可行的竖向空间化系统的发展。这一效果可在许多音像制品中听到。

总的来说,"三维声(音)"系统是基于从外耳(耳廓)和肩膀反射的高频声音(大于 6kHz 的)为竖向定位提供了一个关键暗示信号的这项研究之上的。耳廓表面和肩膀充当了反射体,产生了在频谱上相当于梳状滤波器效果的短时间延迟(Bloom 1977; Rodgers 1981; Kendall and Martens 1984; Kendall, Martens, and Decker 1989)。

顶点暗示可以用电子手段模拟,给人一种声音从高处发出的感觉。这是利用从头和肩膀反射形成的频谱中的变化,通过对输入信号滤波完成的。滤波器是根据你想要模拟的音源的位置设置的。这一过滤的频率响应被称为头部相关传递函数(HRTF, head-related transfer function)(Begault 1991)。图 11.11 画出了高于、位于和低于耳朵高度声音的有代表性的几个头部相关传递函数。

应用中,如果是在前后都有扬声器的情况下分布声音,会极大地增强竖向空间效果。从前往后或是从后往前地平移声音,并应用头部相关传递函数的效果,所听到的声音是随着它的平移经过听众的头顶。就像所有的空间效应一样,宽带脉冲声音的竖向平移要比有平滑包络的低频声音移动得更好。

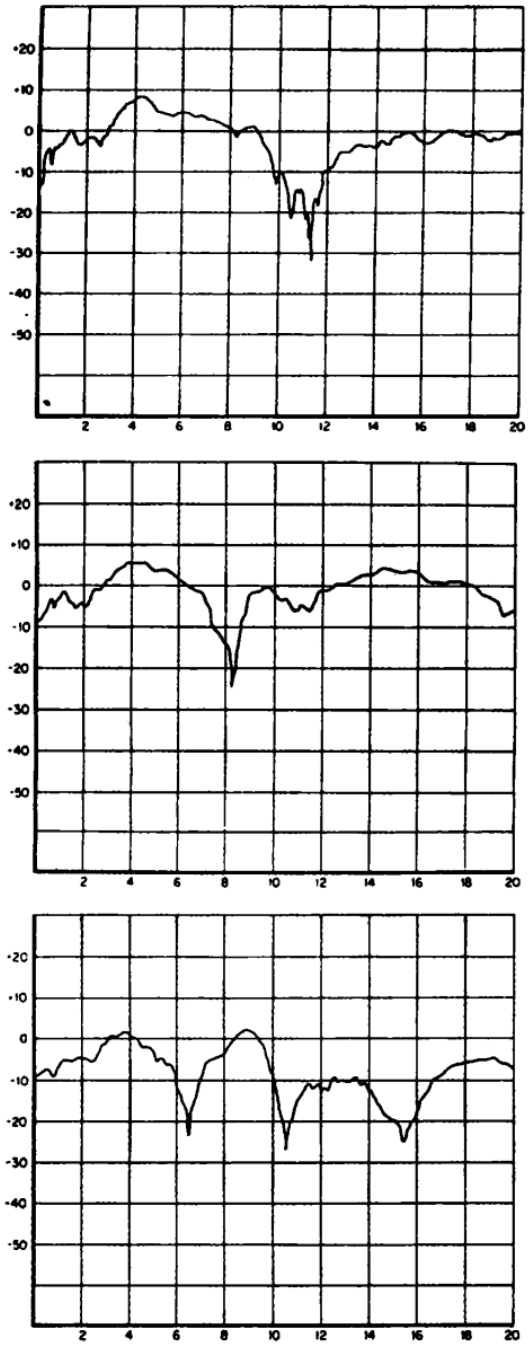


图 11.11　在不同高度，90 度方向（直接进到左耳）所听到的声音的 HRTF 频谱。（上）位于人耳高度上方 15 度；（中）人耳高度；（下）低于人耳高度。（出自 Rodgers 1981；经音响工程师协会的允许发表。）

竖向声音幻觉的问题(Problems with Vertical Sound Illusions)

如图 11.12 所示,在模拟的竖向平面中分布声音时带来的一个问题是,不同的人头部相关传递函数不同(Begault 1991; Kendall, Martens, and Decker 1989)。当对一个人使用了错误的头部相关传递函数时,竖向移动效果便减弱了。对在回放过程中实时完成滤波的家庭收听环境中,解决这个问题的一个办法是提供多个不同的头部相关传递函数和测试信号,这样可以对不同的人预先调整系统以匹配他们耳朵的反射。

竖向幻觉的强度依赖于所使用的扬声器质量以及听众与扬声器的接近程度。例如,从一个小型近场监听器收听时,你必须保持在直达声路径的范围内,否则竖向幻觉会大大降低。因此在一个音乐会的环境中,在听众的头上方悬挂真正的扬声器要比依靠虚拟音源的微弱的幻觉更有实用价值[见图 11.3(d)]。

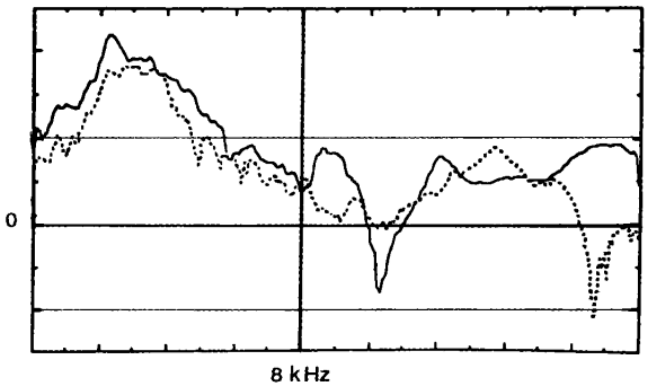


图 11.12 HRTF 对不同的两个人的频谱。左耳,声源在人耳高度。特性曲线从 1kHz 到 18kHz。竖线表示的是 8kHz 的位置。两个头部相关传递函数在高于 8kHz 后的差别是很明显的。横线表示 20dB 的差别。

双耳立体声(Binaural Sound)

在心理声学研究中,双耳原指这样一个独特的收听环境——将收听对象置于一间消声室里,由机器控制他的头部以保持静止不动,并将取样管塞入耳道中。这些环境条件是为了分析在一个受控的环境中听觉系统反应的变化(Durlach and Colburn 1978; Colburn and Durlach 1978; Buser and Imbert 1992)。由于这种实验的困难,很多研究都选择使用耳机。在其他的实验里,一个在耳中放有麦克风的仿真头代替了真人作为实验对象。

这项研究的副产品是一类双耳立体声唱片,用有两个麦克风放在里面的仿

真头或是类似的结构做成，需要通过耳机收听。这种类型的节目在无线电广播产品中很受欢迎，因此导致了双耳路新系统的出现，包括用计算机进行水平和竖向平移控制的混合控制台。

双耳立体声研究的一个成果，是仅通过滤波就可以在双声场中一个特定的位置创造声音源的幻觉。“双声场”指的是通过耳机感觉到的空间，包括头的上方和后方。这些技术使用了之前谈过的头部相关传递函数。详情可见 Blauert (1983)，Durlach and Colburn(1978)，and Begault(1991)。

声波幅射(Sound Radiation)

我们用对声波幅射的说明为所讨论的定位知识做出结论。每一个产生声音的系统都有它特有的辐射(方向)图。这种三维的模式描绘出了各个方向的设备所分布的声音的振幅。传统的声学仪器中，辐射(方向)图是依赖于频率的(Fletch and Rossing 1991)。确切地说，它是随着发射出的频率变化的。辐射(方向)图是确定声音源特性和位置的一个要素。

扬声器系统有它们自己的辐射(方向)图，用技术规格进行描述的话可称为分布类型。一个前发声的扬声器分布图显示出了扬声器保持一个线性录放幅频响应的宽度与高度范围。

在这个意义上，听众会觉察出一个真实的小提琴声和回放小提琴录音之间的区别，因此长久以来声学研究中的方向之一已经集中到模拟乐器的辐射(方向)图，将它们分布到球形的多扬声器设备中(Bloch et al. 1992)。在计算机的控制下，这种系统还可被用于合成，例如给每个声音片断它们自己的辐射(方向)图。

旋转式扬声器(Rotating Loudspeakers)

由一个转动的扬声器发出的声音的辐射产生一个明显的空间效应。一个扬声器的物理旋转使呆板、平稳的声音生动起来。

旋转式扬声器：背景(Rotating Loudspeakers: Background)

最初的旋转式扬声器设备是莱斯利音调箱(Leslie Tone Cabinet)，它将一个输入信号接到两个分离的旋转设备：一个针对高频的可旋转的扬声器和一个针对低频的可旋转的隔音板(对一个固定的低音调扬声器进行阻断和非阻

断)。一个对马达转速的遥控器使得音乐制作者可以控制转动的速度。莱斯利音调箱的共鸣扬声器可使得它立刻可以被识别。

莱斯利音调箱是设计用来丰富由类似于著名的哈蒙德 B3(Hammond B3)型电子琴之类的电子元件发出的固定声音的,它们通常一起使用。但是音乐制作人们和音响工程师们发现任何声音都可以用这种方法丰富,包括语音和电吉他。

在 20 世纪 50 年代,在瑞士格拉夫萨诺(Gravesano)的赫尔曼·谢尔欣实验室工作的工程师开发了一种可以在水平和纵向方向旋转的球状扬声器(图 11.13)(Loescher 1959,1960)。他们的目标是减小由普通扬声器的特性造成的"指向性声音波束"。正如一位设计者所说:

"在纵面和水平面的双轴旋转使单个扬声器的获得倾斜的旋转平面并得到最好的效果。声场在实际应用中变得具有同质性,在声音重现过程中呈现出令人惊讶的丰富和平滑,一般在重现时有的刺耳声完全不见了。"(Loescher 1959)

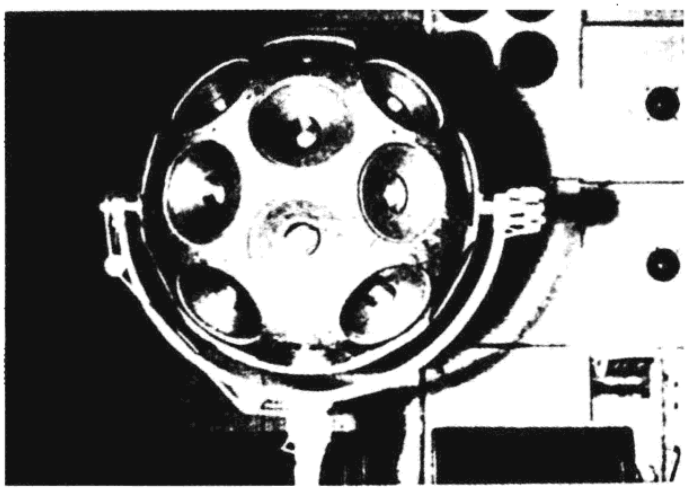


图 11.13 旋转的球状扬声器,1959 年在格拉夫萨诺的实验室开发制成。

卡尔海茵茨·施托克豪森在他的装置"接触"(Kontakte,1960)和"颂歌"(Hymnen,1967)(图 11.14)中,用手转动安装在转盘上的扬声器旋转以发出声音。之后,西德广播电台(WDR)的工程师制造了一个马达驱动的旋转式声音系统,将施托克豪森系统用于音乐会演出(Morawska-Büngler 1988)。

图 11.14　卡尔海茵茨·施托克豪森(K. Stockhausen)和旋转扬声器设备(1960)。围绕扬声器的手动转盘安装了四个麦克风。之后的一套设备是马达驱动的。(照片版权属于 WDR,德国科隆。)

旋转式扬声器的模拟(Simulation of Rotating Loudspeakers)

旋转有多重效应,包括多普勒频移颤音,时变滤波,相位偏移,空气扰动引起的失真和邻近表面的回声反射——更不用提使用的放大器和扬声器的转移特性了。例如莱斯利音调箱使用真空管。如果需要,可以使该电子器件出现“overdrive”失真特性。使用数字信号处理方法,难以令人信服地模拟出这些复杂的声学和电子学效应。尽管如此,还是有很多合成器和效果单元提供模拟旋转扬声器的程序。这些程序会随着更复杂的算法的发展不断改进。

混响(Reverberation)

混响是一种自然发生的听觉效果,我们经常会在大教堂、音乐厅和其他拥有高屋顶与反射表面的地方听到。在这些地方发出的声音与千百万个从房顶、墙面或者地板反射回来的声音相混合并得到增强。到达我们耳朵的声音中,有很多都是被多种不同的表面接连反射,所以我们先听到原始声音,然后才能听到混响。人的耳朵可以分辨出原始声音和反射声,因为反射声一般都具有减弱

的振幅、轻微的延迟以及空气和反射面对高频的吸收所产生的低通滤波器效应(图 11.15)。无数的回声在我们的耳中融合,形成一个相对于源声延迟的声学"声晕"。

在音乐厅中对一个乐器录音的麦克风会被来自于音乐厅的反射声所围绕。特别是在麦克风具有全指向性时,这个效果就更明显。如果在很小的录音棚中进行录音,人们往往需要增加混响,因为没有混响的声音会使人觉得"干瘪",缺乏"空间感"或"纵深感"。

某些合成声音缺乏或不具有内在的空间性。这些在声音学上已经"死掉"了的信号能够通过空间平移、回声和反射处理被加强。

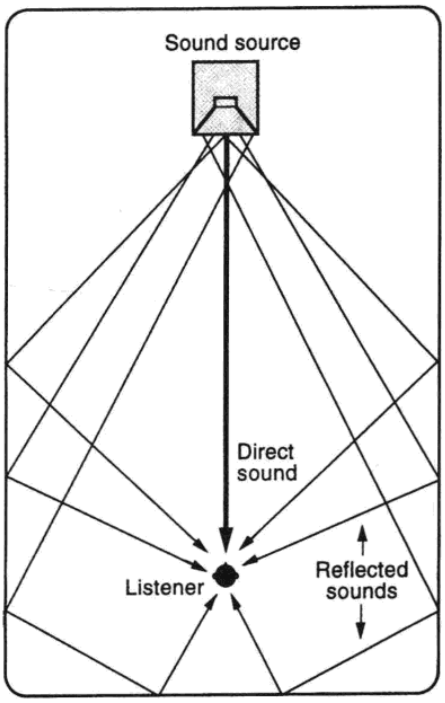


图 11.15 混响产生于声音在空间里的表面反射。图中的黑线是直达声音的路径;其他的线表示的是由于反射而具有较长路径的反射声,它们比源声到达得晚。
Direct sound=直达声音
Reflected sounds=反射声
Listener=听众
Sound Source=声源

但是空间感对于声音来说并不仅仅是装饰性的花样。空间深度在一个音乐结构中可以被用来区分前景元素与背景元素。此外,混响不是一个单一的效果,而是具有丰富的混响属性,这种多样性与多种自然空间和综合的反射物体相对应。没有一种单一形式的混响(自然的或人造的)对于音乐来说是理想的。多数的电子混响器可以模拟若干形态的混响。一些还尝试(通常粗糙地)模拟音乐厅效果,而其他的则创建出奇异的空间景象,这些效果无法在真实的音乐厅中产生。

混响特性(Properties of Reverberation)

贵族音乐沙龙和音乐厅在古代就产生了,但是直到 19 世纪晚期,它们的基本声学属性在科学观点上还无法得到正确解释。对于反射空间的研究是从华莱士·萨拜因(Wallace Sabine, 1868—1919)开始的。他在 1900 年指导建立了波士顿著名的交响乐厅。波士顿交响音乐厅是第一个根据严格的科学的声学定律建成的表演空间。

萨拜因发现房间的混响取决于声音的音量、几何学以及表面的反射率(Sabine 1922)。毫无疑问,有反射面的大房间拥有更长的回音时间;而具有吸收面的小屋子具有短的回音时间。光滑的,坚硬的表面,例如玻璃、铬金属和大理石具有很好的反射所有频率声音的能力;而吸收性表面,例如厚窗帘、泡沫塑料和厚毛毯具有吸收高频声音的特点。

房间内表面的几何学特性决定了声音反射的角度。非平行的墙面可以将声波发散为复杂散射形态,不规则的细碎表面,例如石膏墙面、锯齿状墙面、圆柱墙面和雕像等可以将反射散开,创造出一个更生动、更复杂的混响效果。

萨拜因还发现大厅中的空气潮湿程度也对混响的时间长度有影响,潮湿的空气在某一个潮湿度以上可以对高频的声音产生吸收作用。

房间的脉冲响应(Impulse Response of a Room)

一个测量房间混响的方法是发出一个很短的脉冲,然后在一个时间段内画出房间的混响频率图。这个图形根据脉冲的频谱校正之后,就可用于表明这个房间的脉冲反应。就像在第 10 章中所提到的那样,电路也有脉冲反应,这样就使得脉冲反应测量成为设计电路和音乐厅的最常用的工具。自然混响的脉冲反应曲线与图 11.16 中所示的图形相类似。

混响的形状符合一个准指数曲线,该曲线在半秒内达到波峰,然后多多少少地进行缓慢的衰减。

一般来说,波峰之间的不规则时间间隔,对于一个音乐厅来说是合乎情理的。规则间隔的波峰表示大厅中的频率共鸣,这种频率共鸣是令人厌烦的。

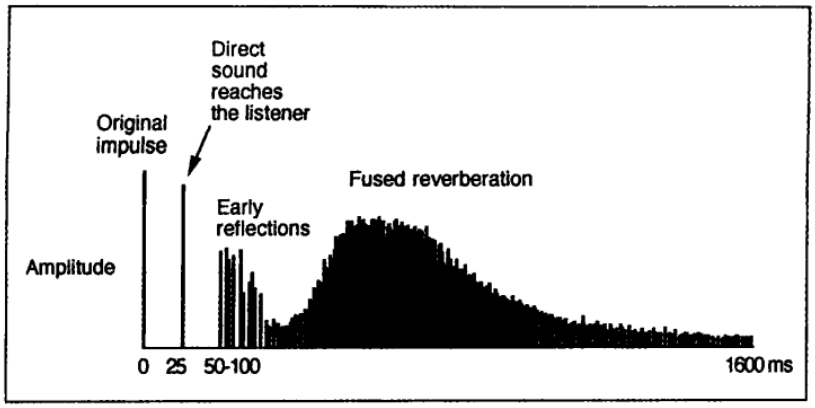


图 11.16 一个混响大厅的脉冲反应曲线。其中的混响的成分被分别表示为“预延迟”(表示为声源到达听者前的 25 毫秒延迟)、早期反射和混合混响。
Amplitude=振幅 Original impulse=原始脉冲
Direct sound reaches the listener=直达声音到达听众 Early reflections=早期反射
Fused reverberation=混合混响

混响时间(Reverberation Time)

另一个重要的测量混响的参数为“混响时间”或 RT60。RT60 代表混响从它的波峰频率(1/1 000 的波峰能量)衰减 60dB 所需要的时间。对于一个音乐厅来说,典型的 RT60 为 1.5 到 3 秒。在图 11.17 中的 RT60 为 2.5 秒。

人工混响:背景(Artificial Reverberation: Background)

录音中最早的人造混响的尝试是将声音发射到一个声学回音室中,然后将源声与混响信号相混合。一些大型的录音工作室现在依然使用一个房间作为回音室。他们将扬声器放在反射房间的一端,再将一个高质量的麦克风放在房间的另一端。将要混响的声音通过扬声器播放出来,然后由麦克风拾取(图 11.18)。一个回音室提供一个独特的声学环境,这需要一个特定的房间、扬声器和麦克风。所有这些条件都配合的很好的话,可以产生出完美的混响。回音室的一个缺点就是(除了建造这一空间的可行性之外)不能很大幅度地改变混响。

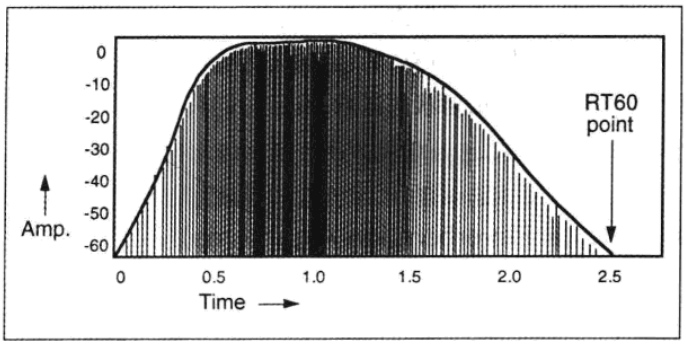


图 11.17　根据混响的波峰衰减 60dB 的时间点来测量混响时间。
Amp.=振幅　Frequency=频率　RT 60 point=RT 60 点

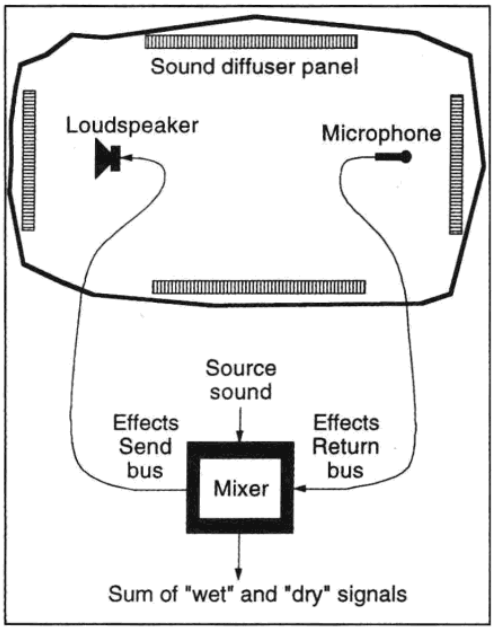


图 11.18　为了实现一种环境声音效果，声音可以通过回音室中的扬声器发出。反射出来的非直达的声音通过房间另一端的麦克风接收。理论上讲，房间应该是不规则的形状。为了使房间的反射最大化并具有随机性，房间应该安装声音散射板。这些声音散射板上面有很多距离不同的凹洞。当一个声波到达面板的时候，就会以不同的延迟时间进行反射，不同的凹洞会产生不同的延迟时间。这些散射效果可以消除平行墙面所产生的驻波（房间的共鸣频率）。
Sound diffuser panel=声音散射板　Loudspeaker=扬声器　Microphone=麦克风
Source sound=声源　Effects Send bus=效果输出总线　Mixer=混音器
Effects Return bus=效果回馈总线　Sum of "wet" and "dry" signals = "干的" 和 "湿的" 信号的汇总

另一个更加常用的增加混响的方法是运用一个"混响单元"或"混响器"。在数字混响器发明之前，在 20 世纪 70 年代中期，混响器是一个精巧的包含有两个引脚（输入与输出）的机电装置，而混响体则像一个长金属弹簧或者金属盘。需要进行混响的声音通过混响体的输入端传输进来，混响体通过其内置的

信号混响/反射器，将无数个回声与源声混合，再通过输出端输出。这样就可以使源声放大和混合，使它们具有“彩色”的人造混响效果。最好的金属板状混响器可以产生相对纯净和散射状的混响，但是它们只能提供几秒钟的 RT60 时间和一个固定的混响模式。

数字混响算法(Digital Reverberation Algorithms)

数字混响利用时间延迟、滤波器和混合来实现一个房间的声音散射幻象。从信号处理的观点来看，一个混响器就是一个能模拟一个房间的脉冲响应的滤波器。贝尔实验室的施罗德(Manfred Schroeder 1961，1962，1970)是第一个通过计算机实现人造混响算法的人。他的混响程序要占用当时的巨型计算机几个小时的运算时间。现代混响单元很紧凑并支持实时运算。在前面板上的控制钮和按键可以让音乐家实现各种不同的效果。大多数的混响器可以通过 MIDI 控制(见第 21 章)。

混响的组成部分(Parts of Reverberation)

混响效果正如前面的图 11.16 中所示的那样，可以被分解为三个部分。

- 直达声(无反射)沿着直线传播，是第一个到达听者的耳朵的声音。
- 早期离散反射声紧接着直达声到达听者。
- 融合的混响包含有成千上万种距离相近的回音，但是需要更多的时间出现然后衰减。

商业混响单元通常可以让人单独地控制这些部分。在这些单元中，源声与混响的平衡通常被称为干/湿比(混响被称为“湿的”)，而在早期反射之前的延迟被称为预延迟。

有效模拟自然混响效果需要高回音密度。一些早期的数字混响器最多只能提供每秒 30 个回音，而一个真实的音乐厅却具有每秒 1000 个回音的回音密度。今天的很多混响器可以使用户根据效果的要求来调节回音密度，完成从离散回音到浓重的、融合的混响效果。

一个音乐厅的离散早期反射可以被模拟为“分接延迟线”。简单地说就是一个延迟单元，可以被“间断地”置于若干个点上，反射若干个版本的输入信号，每个位置具有不同的延迟量。(关于分接延迟线的解释参见第 10 章。)

一个美妙动听的融合混响需要比延迟线更丰富的回音密度。确实存在很多实现不同混响效果的算法，但是它们通常都是施罗德的原始公式的变型，这个原始公式将在下面讲到。

单元混响器(Unit Reverberators)

施罗德称基本构造单元为单元混响器,它们有两种形式:递归梳状滤波器和全通滤波器。这两种形式在第 10 章中都有详细的论述。

递归梳状滤波器(Recursive Comb Filters)

就像第 10 章中说到的,一个无限脉冲响应(IIR)的递归梳状滤波器含有一个反馈循环,在这个循环中,输入信号被采样 D 所延迟,而且以一个振幅或增益系数 g 放大,然后接收并加入最近的输入信号[图 11.19(a)]。

当延迟 D 很小(例如小于 10 毫秒),那么梳状滤波器就会成产生一个谱线的效果。也就是说,它产生了波峰,并含有一点输入信号的频率响应。当 D 大于 10 毫秒,它会产生一系列的衰减回音,如图 11.19(b)所示的那样。这些回音呈指数级衰减,所以如果想获得最大数量的回音(最长的衰减时间),那么 g 就要被设为接近 1.0。输入端的声音通过梳状滤波器衰减 60dB 的时间按照下列的方程式来计算(Moore 1990):

decay_time=(60/−loopGain)× loopDelay

环状增益是增益 g 的 dB(分贝)表示=$20 * \log_{10}(g)$,而环状衰减是衰减 D 的秒数表示=D/R,其中 R 表示的是采样率。所以如果 $g=0.7$,那么环状增益=−3dB。

全通滤波器(Allpass Filters)

全通滤波器将稳态信号的所有频率以相同频率输出(见第 10 章)。但是它们会将瞬时信号通过引入频率相关衰减而“扭曲”。当衰减时间足够长时(在 5 和 100 毫秒之间),图 11.20(a)中所示的全通滤波器会就会像图 11.20(b)中所示的那样有一个脉冲反应:产生一系列的指数级衰减回音脉冲,就像应用了长时间延迟的梳状滤波器那样。脉冲之间均匀的间隔说明,当一个短的、瞬时的声音被引入时,滤波器就会以与该滤波器延迟时间相同的周期产生混响。

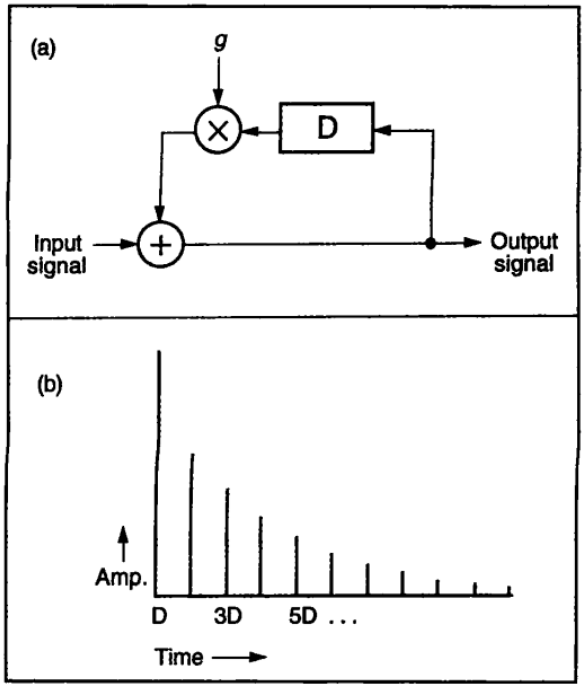


图 11.19 一个有混响作用的递归梳状滤波器。(a)以 D(延迟采样的数量)和 g(反馈的数量)为系数的梳状滤波器电路;(b)作为一系列回音的脉冲响应。
Input signal=输入信号 Output signal=输出信号 Amp.=振幅 Time=时间

这就解释了为什么当全通滤波器对待尖锐和瞬时的声音时不是“无色的”。

混响排秩(Reverberation Patches)

我们已经确定了递归梳状滤波器和全通滤波器都可以生成一系列的衰减回音。对于效果丰富的混响,非常必要将一定数量的单元混响器交叉连接起来,以创建出足够的回音密度来使回音相混合。当单元混响器并联的时候,它们的回音会增加;当它们串联的时候,每一个单元所产生的回音都会成为下一个单元的回音触发信号,创造出更加浓密的回音。这个串联所产生的回音的数量是每个单元所产生的回音数量的总和。

在施罗德的设计中,梳状滤波器以并联的形式相互连接以减少异常声音。例如,一个通过梳状滤波器的频率可能会被另一个滤波器所削弱。全通滤波器通常以串联的形式相连接。由于它们所具有的相位扭曲的特点,并联全通滤波器会由于相位抵消效应而导致不均匀的振幅反馈。

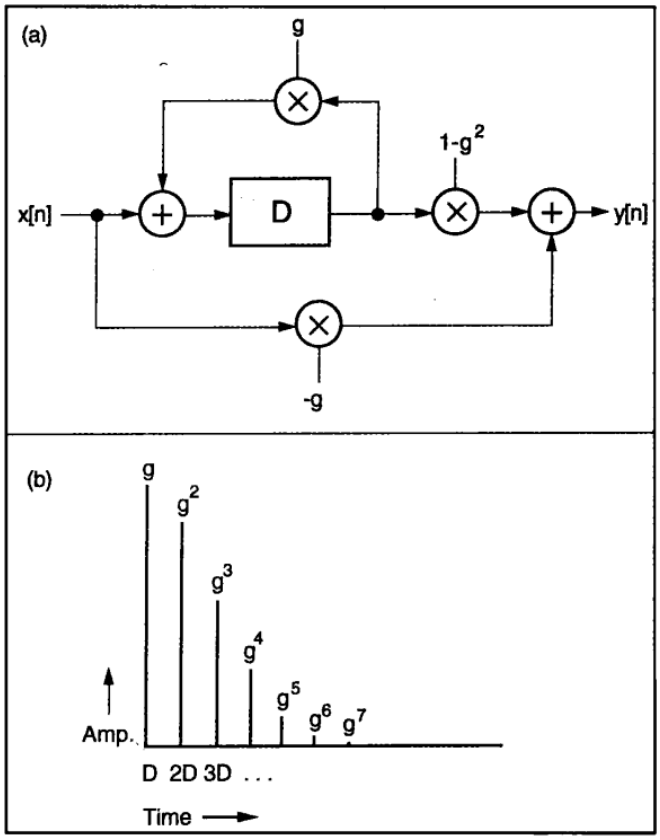


图 11.20　一个先序全通网络。(a)通过添加-g 倍的输入信号，而得到输出端的延迟，一个梳状滤波器变为一个全通滤波器；(b)全通滤波器的脉冲响应是一系列指数级衰减的回音脉冲。这使得脉冲滤波器成为混响器的有用组成部分。
Amp.=振幅　Time=时间

图 11.21 显示了施罗德提出的两种混响器。在图 11.21(a)中，并联的梳状滤波器形成了一个序列的回音，并且将其提供给两个串联的全通滤波器。在图 11.21(b)中，五个全通滤波器使得回音密度在每个单元中都得到增加。如果每个全通滤波器都生成四个可听到的回音，那么 5 号全通滤波器的输出端最后将有 1 024 个回音。

这种类型的数字混响系统的声音特征依赖于延迟时间 D(这些时间 D 决定了回音的间隔时间)和放大系数 g(这些系数 g 决定了衰减或混响的时间)。延迟时间也被称为循环时间。

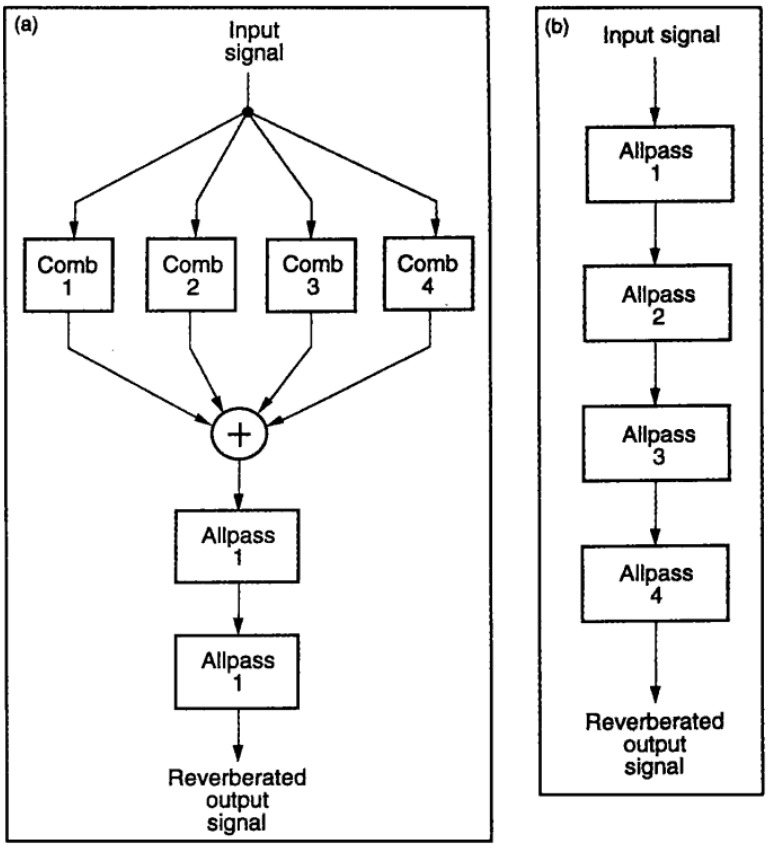


图 11.21 施罗德设计的原始混响器:(a)并联的梳状滤波器将信号传送给两个全通滤波器;(b)五个全通滤波器组成一个串联。
Input signal=输入信号 Comb=梳状滤波器 Allpass=全通滤波器
Reverberated output signal=混响的输出信号

对于自然声响的混响,选择合适的不同延迟时间是很重要的,相对来说这比其他因素都要重要(它们不具有相同的约数)(Moorer1977,1979c)。为什么是这样呢?对于两个梳状滤波器,它们的第一个延迟时间为 10 毫秒,那么第二个延迟时间就为 12.5 毫秒,它们的延迟时间线的长度在采样率为 40kHz 时,分别为 800 个单位与 1 000 个单位。由于这两个延迟时间的长度都可以被 200 整除,那么由这两个单元所组成的混响器不具有一个平滑的衰减。基于 200 毫秒的倍数,回音在这个点上会得到加强,产生强烈的不连续的回音或者有规律的"跳动"衰减。当延迟时间被调整为 10.025 和 24.925 毫秒时,它们的延迟时间线长度分别为 799 个单位和 997 个单位。那么现在回音的第一次重合在(799 个单位×997 个单位)/40kHz= 19.91 秒前不会出现。(见 Moorer 1979c 年关于如何调整这些参数的讨论。)

大家在这时也许会发现,更短的延迟时间与更小的声音空间相关。对于一

个巨大的音乐厅，使用图 11.21a 中的梳状滤波器组成的混响器，以 50 毫秒为延迟时间，最长的延迟时间与最短的比值为：1.7：1。对于一个很小的瓷砖房间，梳状滤波器的延迟时间可以被设定为 10 毫秒左右，全通滤波器具有相对较短的 5 毫秒或更短的循环时间。全通滤波器的混响时间应很短（低于 100 毫秒），因为其用途是增加整体混响密度，而不是增加持续时间。

对早期反射声的模拟（Simulation of Early Reflections）

施罗德（Schroeder）的混响算法可以被称为“分接循环延迟”模型（TRD）。就像前面所解释过的，混响器通常由梳状滤波器和全通滤波器所组成，它们可以生成足够的回音密度，从而创造出真实的“全局混响”的模拟。“分接循环延迟”模型虽然是有效的，但是它只能生成普通的全局混响，而且不能生成具有精密声学特性的现场真实演奏的空间效果。

在 1970 年，施罗德将他的最初的混响器公式扩展到“分接循环延迟”，以此来模拟早期反射，这些早期反射是音乐厅的融合的混响声音之前的重要声音（见第 10 章中关于多抽头延迟线）。这个设计适用于绝大多数的商业混响器，图 11.22 中展示的就是这个设计。这样，在模拟一个特定的音乐厅时，对基础 TRD 模型改进的最直接的方法就是将测量过的大厅的早期反射移植到通用全局混响器（Moorer 1979）。对于这个公式的更进一步的扩展是使用全局混响的低通滤波器，这个低通滤波器是根据所测量的大厅的声音吸收特性所设计的。

设计混响器的另一个很重要的考虑因素是声音进入人的两个耳朵时，每个耳朵所接收的声音应该是“相互不连贯的”。也就是说，混响算法应该对每一个通道的处理具有微小的不同（不相关性）。

虚拟混响效果（Fictional Reverberation Effects）

电子音乐合成师在合成音乐的拓展上所取得的成功要远远大于对于真实空间的混响效果的模拟。一个混响师可以使用很多不真实存在的“虚构的”空间效果，这些效果在真实的环境是不存在的。一个很常用的例子就是“门”混响器，它可以快速地增加回音的密度，然后突然中断。“门”混响器在 20 世纪 80 年代被用来模拟小军鼓，后来被流行音乐广为使用。

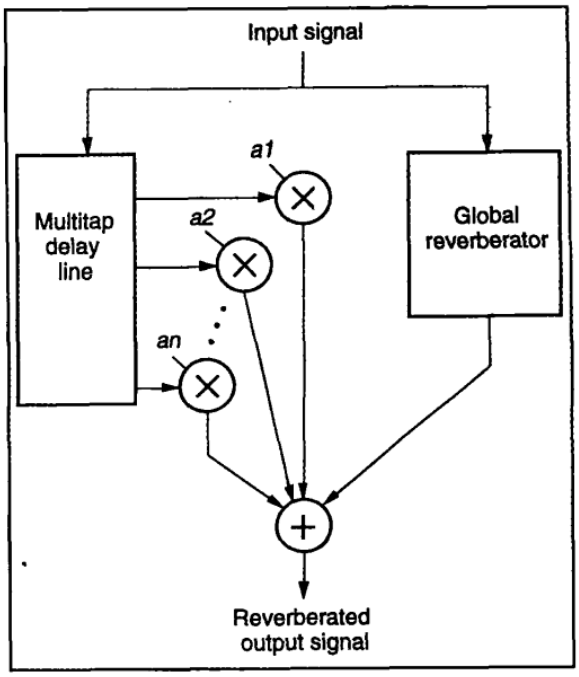


图 11.22 在施罗德最新的设计中，多分接延迟线模拟出了音乐厅中的早期混响声。
Input signal=输入信号 Multitap delay line=多分接延迟线 Global reverberator=全局混响器
Reverberated output signal=混响的输出信号

表 11.2 典型混响器的参数

参数	描述
混响类型	选项有“大厅”“房间”“板”“门廊”
大小	设置单位混响的延迟时间
预延迟	控制效果的触发时间
输入端延迟	在合成之前产生效果(湿声先于干声)
混响时间	设置延迟时间
散射	决定回音密度
混合	设备输出端的输入声音与混响声音的比率
高通滤波器	只对高音进行混响，创造“咝咝声”效果
低通滤波器	只对低音进行混响，创造“消音器”效果

其他的效果例如“咝咝声”混响器，含有一个高通滤波器来生成混响声，而与它相反的，另一个“消声器”混响器含有一个陡峭的低通滤波器。通过控制混响器的参数，人们可以创造出怪异的混合体，例如在很小的房间中实现很长的混响时间。表 11.2 列出的就是很多商业混响器的参数。

在这章的后面会谈到混响器的卷积部分，提供了另一种非现实混响器。它使用了第 5 章中所讲的异步颗粒合成技术。

声音空间建模(Modeling Sound Spaces)

对混响器的研究还在进行。在前几章中关于混响的算法是我们现在要讲的设计的前提。这章将要解释若干个近年来出现的实现更真实的混响技术，这些技术包括基本施罗德公式的扩展、几何学模型、卷积混响器、波导混响器以及多流混响器。

其中的一些技术属于混响的物理建模(见第 7 章声音合成中的物理模型理论的应用的介绍)。这些包含大量数学的方法构建了真实的声波扩散模型。它们不但可以建立更加真实的模型，而且可以为想象空间的模拟提供了可行性。在这个讨论中，我们引入了多个空间，这些空间的特性和几何形状可以随着时间变化——例如一个弹性的音乐厅，它可以在乐句当中进行扩展和收缩——或者是一个不可能存在的空间，例如具有很长混响时间的壁橱。这样，这些技术所能产生的效果不仅仅是真实的混响，而是具有戏剧性的空间变化。

施罗德混响算法的扩展(Extensions to Schroeder Reverberation Algorithms)

在标准的施罗德混响算法中，全通滤波器可以生成一系列指数延迟的回音，一个对于施罗德模型的扩展就是用“振荡全通”滤波器来代替施罗德设计的常规的全通滤波器。这样一来，全通滤波器对于脉冲的反应就成为了一个具有阻尼正弦曲线振幅的脉冲序列，

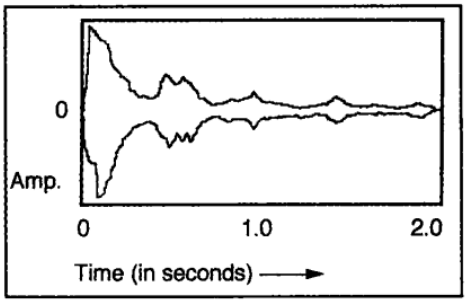


图 11.23　摆动全通滤波单元混响器对于刺激的反应图。
Amp.=振幅
Time=时间
in seconds=单位：秒

这就为“良好听音”空间构建了具有轻微波动混响模式的模型。

声音空间的几何模型(Geometric Modeling of Sound Spaces)

与 TRD 实现不同的另一种方法就是利用电脑辅助设计(CAD)系统来建立

物理模型，这个物理模型要具有房间的几何学特性。扬声器将声音透过“声学窗户”投射到一个虚拟的房间中，使之产生环绕效果。

在 1983 年穆尔的设计中，每一个声音源都是一个矢量，这个矢量有可调节的位置、方向性、数量和散射性。从将声源矢量向房间中投影开始，由电脑追踪声音反射的路径（Moore 1983）。在一个完备的几何模型中，混响算法可以将反射模式建模所产生成为上百条模拟声音的射线。根据模型的精细程度不同，这种方法可能是非常耗费运算量的。为了提高运算效率，穆尔使用了一个几何学方法来建立模型，这个模型只对模拟房间的早期反射进行建模。他使用了施罗德的标准 TDR 模型来生成全局混响。

穆尔在 1979 年就指出了用简约的几何学方法来为混响建模所产生的一个问题，就是这种方法无法生成真实音乐厅中的漫射效果。漫射的产生原理是由于没有哪个物体的表面是 100％光滑和反射的，所以当声音从这些表面扩散，它们的部分能量就会被反射点所吸收，有一些方法试图针对声音漫反射进行单独建模来改进追踪模型，这就需要在每个反射点上引入随机散射函数。“波导网络”混响（后面会讨论到）就是另一个针对漫反射进行建模的方法。

卷积混响（Reverberation via Convolution）

一个很精确但很耗费运算时间的模拟给定空间混响的方法，就是将空间的脉冲响应与发生混响的信号相卷积（见第 10 章以及 Smith 在 1985 年对于卷积的文章）。我们可以将混响器看作为一种滤波器，它的脉冲响应的长度（采样）是与模拟的大厅的混响时间（采样）相关的。一个房间的脉冲响应是房间的非常短暂的爆破声的反馈的集合。然后这个采样集与要产生混响的信号进行卷积。

第 10 章区分开了“直接”与“快速”卷积。直接卷积对于混响来说并不实用，因为它要非常大量的运算时间。举例来说，一个采样率在 48kHz 脉冲响应时间在 3 秒的声音，每一个采样信号的每一个输入信号的通道都必须相加在一起，这就是 48 000×3 次运算，对于一个 1 秒钟的输入声音来说，这个变换需要如下运算量：

144 000	×	48 000	=	6 912 000 000
相乘/加		采样数		相乘/加
每个采样		每秒		每秒每通道

所以，要对一个 1 秒钟的立体声声音通过卷积的方法来实现混响的话，大

概需要 13.824×10^{12} 次相乘/加。这么庞大的运算要想产生实时效果就需要昂贵的超级电脑的计算性能。对于一个实际应用性能为 100×10^{6} 相乘/加每秒的信号处理引擎，例如一个个人电脑的插件板，这样的计算会用掉 2 分 80 秒的时间来计算，与实时相比就是 138∶1 的比率。

所以可以用于实用的卷积混响就是使用快速卷积。这种快速卷积利用了快速傅里叶变换(FFT)所提供的加速效果。参见第 10 章的快速卷积；附录中解释了 FFT。

颗粒混响(Granular Reverberation)

“滚滚雷声在云层之间产生了回音；如果一个云层是由水滴所组成的。那么每一个水滴都可以反射声音，那么云中产生巨大混响的原因就不言自明了。”(约翰·贺希尔(John Herschel)爵士，在 Tyndall 中引述 1875)

这段描写说明了混响效果可以用一个任意的输入声音与云层中的声粒子相卷积来产生。

据推测，大气中的云层可以产生混响效果，20 世纪的法国声学家 Arago，Mathieu 和 Prony 在他们的关于声速的实验中观察到了在晴朗的天空中加农炮的声音通常都是单独且短暂的。然而，当天空中有大片的乌云覆盖时，加农炮的声音经常伴随着一个长时间持续的类似于雷声的“混响”(Tyndall 1875)。(见 Uman 1984 年对于雷的声学分析。)

倘若人们了解卷积，那么他们就不难理解声音粒子与云层声音的卷积所产生的漫无目标的类似于大气混响的“断续时间”效果。断续时间由不同密度的云层声音颗粒开始，颗粒由“异步颗粒合成”(AGS)技术产生，参见第 5 章。AGS 在统计学的时间/频率平面上发射粒子。在卷积中，大量的粒子可以被认为是积云云层的脉冲响应。由每个粒子形成的虚拟的“反射”及时地散射输入的声音；也就是说，它增加了多个间隔不规则的延迟。如果每个粒子都是一个单一的采样脉冲，那么它的回音就是原始输入声音的复制。然而由于每个粒子可能都会含有上百个采样，每个回音在局部都是时间抵消的。

时间散射效果可以被分为两个基本的类型，这取决于输入声音的强度。如果输入的声音由一个强起音开始，那么每个粒子都会产生那个冲击的回音，如果粒子云不是连续的，那么这些回音就会在时间中不均匀地分布。如果输入的声音又一个很柔和的冲击，那么时间阶段发射本身也会是柔和的，但是会带有某种“色彩”奇特的混响。这种奇特混响的“色彩”和回音是由粒子的频谱决定的，这个频谱是由每个粒子的时间长度、包络和波形决定的。参见第 5 章中关于粒子参数的内容。

波导混响(Waveguide Reverberation)

“波导”是一种声音传播的媒介运算模型。物理学家已经使用波导网络很长时间了,他们用它来描述一个共鸣空间的波的行为(Crawford 1968)。波在其中传输(Smith 1985 c,1987 a,b;Garnett and Mont-Reynaud 1988;第 7 章介绍了更多关于声音合成中的波导应用)。每个延迟线包含在一个方向传播的波,当它到达线的终点时反射回中心交汇处。通过将一定数量的波导连接在一起形成网络,人们可以建立声学媒介的模型,例如一个音乐厅的反射模式。

在波导混响中,每一个波导延迟线的长度都是不同的,以此来模拟音乐厅中的不同回音时间。能量在多个波导的相交处被散射,产生了融合的反射效果,这就形成了典型的融合混响声音(图 11.25)。在一个闭合的网络中,当一个信号被引入进来,它就可以在网络中自由的循环而不损失能量。

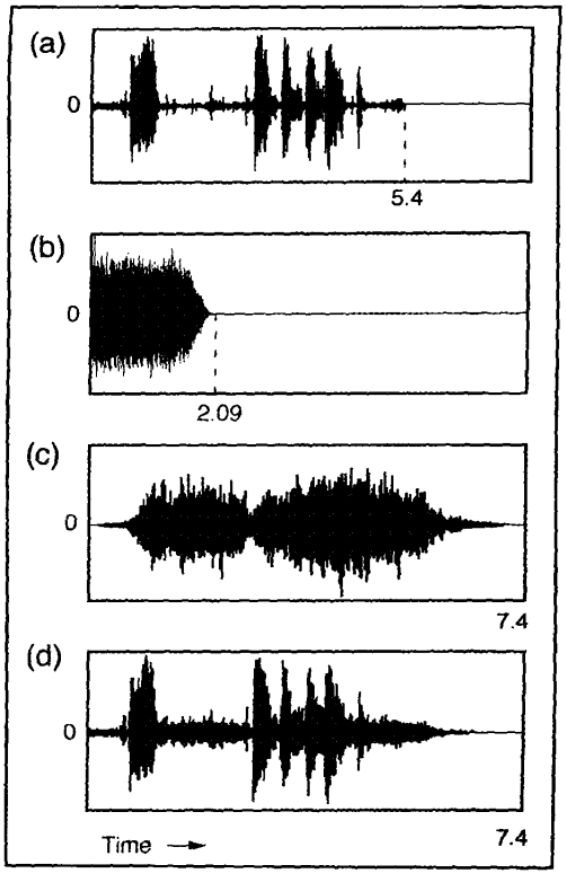


图 11.24 颗粒卷积的混响 (a)语音输入:“Moi, Alpha Soixante”;(b)粒子脉冲响应,每 9 毫秒包含 1 000 个正弦粒子,粒子中心在 14 000Hz,带宽为 5 000Hz;(c)(a)和(b)的卷积;(d)(a)和(c)以 5∶1 的比例相混合,创建语音的混响。
Time=时间

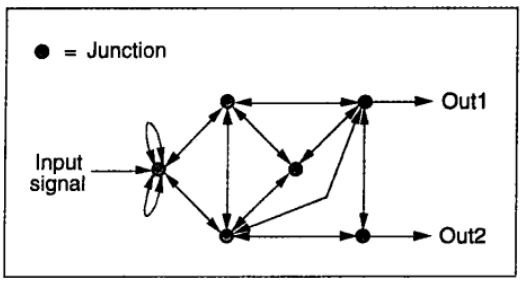


图 11.25　一个三端口 6 节点的波导网络。这个波导将能量传播到输出端外，这就意味着这是一个开放的网络，会逐渐丢失能量，这和一个混响大厅相同。
Input signal=输入信号　　Junction=连接　　Out=输出

为了保持一个混响效果，人们必须使用丢失很少振幅能量的网络，以此来获得想要的混响时间。信号的输入和输出可以在网络的任何一个地方实现。

波导网络为高效的混响模型。一个具有 N 个连接的网络需要 N 次乘法和 $2N$-1 次加法来得到输出样本。连接的数量 N 取决于被建模的系统。一个共鸣箱的模型需要 8 个交叉点，而一个复杂房间的混响模型需要上百个连接点，因为信号发散的任何一个地方都需要一个连接点。

波导网络的结构保证了网络中不会有任何的数值溢出或者振荡。此外，声音射线散射的重要性质可以通过一个波导网络来模拟出来(Moorer 1979)，而简单的几何模型却不能。“移动墙”效果可以通过平滑地改变延迟线长度来得到。

多流混响(Multiple-stream Reverberation)

多流混响可以被看作是两种方法的妥协，即详细但计算量大的方式来实现混响(例如几何建模或用卷积实现混响)与“快捷的但为全局 TRD 模型方法两者的妥协。多流混响将混响的信号划分为若干流，每个流都是对来自虚拟房间的很小的不同部分的混响建模。每个流都是通过一个根据房间的不同部分而调准的 TRD 网络(梳状和全通滤波器)实现的。

20 世纪 80 年代由美国西北大学开发的“立体声混响器”系统，使用了多流技术，并且将其他两种处理器与其相结合：(1)一个房间反射的模型；(2)方位提示是由人的耳朵、肩和人体上部对声音级别的反映形成的(Kendall and Martens 1984；Kendall et al. 1986；Kendall，Martens and Decker 1989)。第一个和第二个反射决定了每个独立混响流的延迟时间。然后，在对每个流单独地实现混响效果后，一个“导向器”将每个流滤波，根据它在虚拟的三维空间中的位置来生成一个额外的声音定位暗示(图 11.26)。

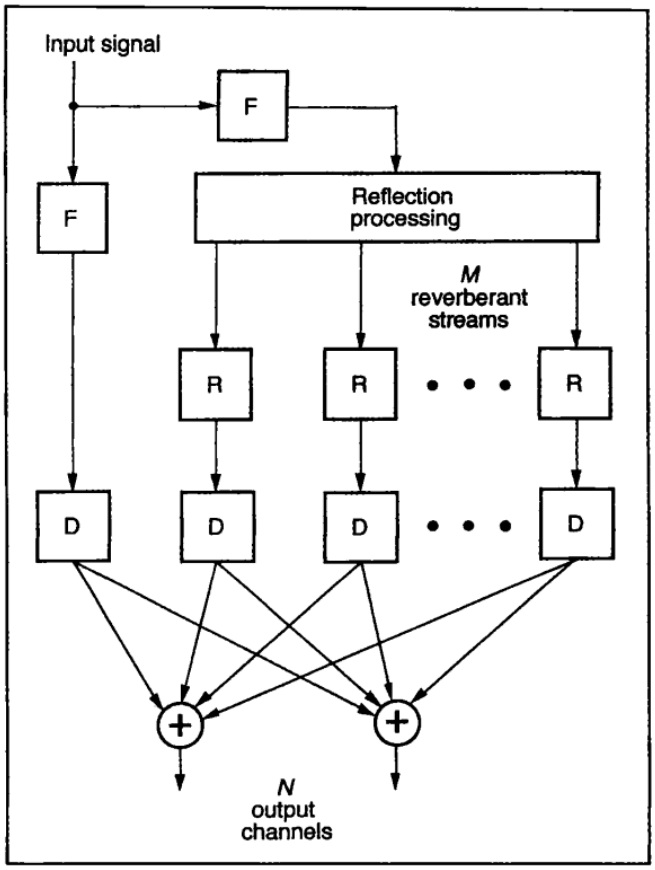


图 11.26 由 Kendall, Martens 和 Decker 所绘制的"空间混响"简化图。这个系统通过把 M 个局部混响器的贡献加在一起,构造一个空间的模型。这个空间最终产生 N 个输出通道。F 是一个预滤器,它可以加入由于距离和空气收缩产生的频谱改变。R 是局部混响流,它为整个房间的子空间混响建立了模型。D 是一个定向器,它可以根据声音在虚拟空间中的位置过滤筛选声音。实现这个系统需有两个独立的反射处理器,和一些混响流的互馈器。

Input signal=输入信号
Reflection processing=反射处理
M reverberant streams=M 个混响流
N output channels=N 个输出通道

使用这个系统的用户可以设定虚拟空间的声学特性,例如空间维度、声响位置、听者位置、墙体的声音吸收等。为了模拟一个房间的混响模式,每个主要方向的混响都被处理为一个单独的流,一个实现方案中最多有 18 个流(Kendall, Martens and Decker 1989)。如图 11.26 所示,混响流的数量与最后用于投射声音的输出通道数量是相互独立的。

独立的混响流的概念也在 20 世纪 80 年代早期麻省理工学院的四声道混响研究中出现(Stautner and Puckette 1982)。在这个研究中,扬声器输出与声源的输入通道有空间相关性。例如,从一个左前扬声器发射出来一个直达声音可能被听成两个相邻的扬声器的声音的混响,而最终似乎来自相反的右后扬

声器。

结论(Conclusion)

用电子手段实现声音的空间化广泛存在于音乐制作中，从拾音技术、信号处理到声音通过扬声器或者耳机投射。随着心理声学中关于空间感知的理论知识的发展，空间化系统也已经变得更加复杂。很多的工作室拥有多种空间效果处理器。一般而言，在一个特定的时候使用不同的设备能获得不同的音频质量。

对于自然效果的模拟，例如扬声器旋转和混响仅是对真实世界的近似。当一个扬声器旋转时，我们仍然不是完全都知道其中声音的变化。混响效果最好的音乐厅具有非常真实而且丰富的混响效果，这是人造混响无法实现的。但是尽管最好的感受是在这些音乐厅中获得，它们自然的混响效果也要通过录音技术与扩音器才得以传达和重现。

合成空间化处理器的最主要的优点在于它的灵活性——很多不同种类的混响和空间化处理。它甚至可以实现超自然的效果，而这些效果在真实的世界中是不存在的。它们从一种空间特效到另一种空间特效的转换可以在音乐线上实时完成。

作为结语，需要谈一下对立体声媒介的评论。在 20 世纪 30 年代出现的立体声录音与播放是一个很大的进步。但由于在广播和销售市场上双声道媒体持续占据主导地，使得实现更复杂的空间化处理非常困难，而且也不现实。大规模发布真正的多声道音频媒体才有可能极大地促进音乐空间化领域的进步。

PDG

第四部分　声音分析
(Sound Analysis)

PDG

第四部分概述(Overview to Part Ⅳ)

声音分析在计算机音乐中具有深远的意义——计算机通过“听”,使其能够识别、理解其所听到的内容并做出音乐性的反应。对以下诸多音乐应用来说,声音分析是第一步:

- 分析/再合成,声音被分析、改变,经音乐家处理后再合成。
- 实时响应(Roads 1985b, 1986b)。
- 建立可以根据声音的声学属性检索的声音数据库(Feiten and Ungvary 1990)。
- 根据空间分析调谐声音补强系统从而增加表演和听音空间。
- 修复旧唱片重新发行(Borish 1984, Lagadec and Pelloni 1983, Moorer and Berger 1986)。修复可以是普通的识别如消除咔嗒与哼鸣声,亦或可以神奇似地将人声从乐队背景中分离出来。一个著名的例子是对歌剧演员恩里科·卡鲁索(Enrico Caruso)演唱的修复。在几乎保证人声完整无损的前提下,用一个高保真音质的乐队替代了破旧录音中的交响乐队(Miller 1973, Stockham, Cannon, and Ingebretson 1975)。
- 降低采样数字音频存储量的数据压缩算法(Stautner1983, Moorer 1979b, Pohlmann 1989a)。
- 将声音转换为普通的音乐记谱(Moorer 1975, Piszczalski and Galler 1977, Chafe et al. 1982, Foster et al. 1982, Haus 1983),或出于音乐分析的目的将声音转换成声谱图的形式(Potter and Teaney1980, Cogan 1984, Waters and Ungvary 1990, Lundén and Ungvary1991)。
- 基于音乐演奏声音本身,而不仅仅是低质的乐谱,来发展音乐理论。

第四部分中的一些历史段落讲述了百年来声音分析的历程。以前,不运用计算机来进行声音分析是非常困难的,需要极大的努力。20 世纪 60 年代声音分析成为可应用的科学工具。只有到 20 世纪 80 年代晚期,当理论成果与廉价

且功能强大的硬件结合在一起的时候，声音分析在音乐应用中才变得普遍起来。如今，音乐家可以将物理、工程学及人工智能等一系列成果应用于对音乐信号的分割、分析、识别、理解和实时转译的工作中。

第四部分的构成(Organization of Part Ⅳ)

第四部分将声音分析分为三个部分：音高、节奏(第 12 章)与频谱(第 13 章)。这些章节既涵盖时域也包括频域方法。第 13 章最后部分检查系统综合了数字声音处理算法、高级控制结构以及决策机制——智能信号处理领域的内容。

音高、节奏和频谱仅是声音分析中最为确定的范畴。正涌现出的技术将声音分解为激磁或共振分量或将混合在某信号中的不同声音源分离；其他方法包括定点共振峰、搜寻隐蔽调制，或衍生无序驱动函数(Bernardi，Bugna，and De Poli 1992，Pressing，Scallan，and Dicker 1993)；以及其他无数可以想象的可能性的出现，使得声音分析成为开放的探索领域。

第 12 章　音高及节奏识别 (Pitch and Rhythm Recognition)

声音分析已经成为未来计算机音乐发展背后的焦点，并在交互式作曲（可响应的乐器）、伴奏系统和新的声音转化方法等应用中扮演着关键角色。由于人类听觉知识的匮乏和计算机能力的局限，从计算效率或直观需求上看，现代声音分析方法呈现出混杂的趋势。尽管如此，基于详细的人类感知模型的方法正在浮出水面，本章及下一章将阐述声音分析艺术背后的基础概念。第 12 章主要讲述对音高、节奏的机器分析——两个年轻的领域，每个领域都尚未出现标准，新方法不断涌现。由此，本章不会试图涵盖所有可能的方法，例如，神经网络在音高、节奏分析方面的应用似乎大有前途（D'Autilia and Guerra 1991，Desain and Honing 1989，Todd and Loy 1991）。这里我们的目标是呈示基础问题，审视以往的策略，并对这些技术在音乐上的应用进行介绍。看起来未来的工作将会建立在本章所呈现方法的基础之上。

在直接切入当前的音高、节奏分析主题之前，我们推荐预习一下对声音属性的探究历史，即如下的背景部分。

音高、节奏和波形分析：背景 (Pitch, Rhythm, and Waveform Analysis: Background)

对音乐声音特征的描述和度量可以追溯到古代。古吠檀多（正统印度教教徒）经文有关音乐的文献中体现出八度均等的概念，并将一个八度分成 22 个施鲁提（shruti）音程（Framjee 1958，Danielou 1958）。这种被希腊人称为恩哈默尼科斯（Enarmonikos）的施鲁提（shruti）音阶，被古希腊看作是一切音阶的基础。毕达哥拉斯（Pythagoras，公元前 580—公元前 500）记载了音高与绳索长度部分间的对应关系，致使他以算术比例来描述音程和音阶。希腊人也发展出一系列作为中世纪欧洲音乐节奏基础的节奏套路或者“模型”。尽管之后音乐记谱法缓慢地发展起来，但这几乎不能作为精确声学测量的基础。

在声音放大器、振荡器、示波器这些电子设备发明之前，声学测量被局限在最基础的声音特性层面。1636 年，伽利略（Galileo，1564—1642）和梅森（Mersenne，1588—1648）用试验方法将音高归于波形的频率。梅森和伽桑狄（Gassendi，1592—1655）第一次尝试测定声波的传导速度。大约 1 700 年，索沃尔（Sauver，1653—1716）发明了一种对声学振动进行计数的方法。他采用术语泛音（les harmoniques）描述高频分音伴随基频出现这一现象。

小号与琉特琴演奏家肖尔（John Shore）于 1711 年发明了以某一固定音高振动的音叉，并幽默地称之为“音高餐叉”。1830 年索沃尔开发出了一种通过旋转的带锯齿的轮子测量音高的方法。索沃尔将一个簧片压在不同的轮子上，

根据旋转速度和锯齿数量测量声音精确的频率(Beranek 1949)。在巴黎圣路易斯(Île Saint Louis),一个实验室工作的德裔声学家鲁道夫·凯尼格(Rudolf Koenig,1842—1914)通过 154 个音叉的共振拍音来测量声音的音高,制造了一个精确的,可听音域范围的音调计(tonometer)。

第一个精确测量声波强度的仪器是拉库尔(La Cour,1878)的音轮(phonic wheel)和以伟大的英国声学家劳德·雷利(Lord Rayleigh,1842—1919)命名的雷利声盘(Rayleigh disk,1882)。第一个电子声级计直到德福雷斯特(Lee DeForest,1873—1961)发明三极真空管两年后,才由 G. W. 皮尔斯(G. W. Pierce)在 1908 年建造。

早期的声音图像(Early Images of Sound)

波形可被听到却不能被看到是早期声学家们进行声音研究时面临的一个难题。他们别出心裁地发明了各种观察声音的方法。其中之一包含调节本生灯(译注:煤气灯)的声音并观察火焰效果的变化。第一次史料记载的试图对声音火焰的分析是由希根斯(Higgens)博士于 1777 年完成的(Tyndall 1875)。鲁道夫·凯尼格(Rudolf Koenig)建造了他称为感压焰(manometric flames)的产生声音图像的精密仪器(图 12.1)。(详情见 Mayer1878, Poynting and Thomson 1900, Beranek 1949。)

通过在本生灯周围放置共振管,约翰·廷德耳(John Tyndall ,1820—1893)让火焰"歌唱"了起来。他也描述了被其称为易感裸焰(sensitive naked flames)——不被管子包围的试验。廷德耳根据"尾"、"翼"和"叉"的形状分析声音火焰的模式。用来描绘声音波形的其他介质有:经声音调制的烟雾和高压喷水。

更多声音波形的直接映像出现于 19 世纪中期。韦斯登示振器(Wheatstone Kaleidaphone, 1827)将振动投射到屏幕上。这使得利萨如(Lissajous, 1857)建立起他自己的,同时表示两个振动信号频率差及相位差的利萨如模型。斯柯特-凯尼格(Scott-Koenig)声波记振仪是一个位于声学号筒尾部的振动膜。附着在振膜上的是一个记录针,它在由旋转滚筒驱动的熏纸上映描出振膜的振动(图 12.2)。D. C. 米勒(D. C. Miller)的声波显示仪(Phonodeik,1916)是在时域显示波形方面的一次重大进步,因为它记录在速度为 13.3 米/秒的光学胶片上。

早期声音记录器(Early Sound Recorders)

第一台声音记录器源自图形方式捕捉声音的成果。在声波记振仪的启发下,爱迪生(Thomas Alva Edison)的早期留声机(Phonograph, 1878)将声音刻

在锡薄筒上，可以支持后来声音的回放。一年后，爱迪生换用了蜡筒。若干研究者研制出了诸多将声音波形刻录在留声机滚筒上从而给声波摄影的方法(Miller 1916)。另一个记录设备，埃米尔·贝利纳(Emile Berliner)的留声机(Gramophone)系统(1887)使用了旋转的涂漆的碟，最终成为大家选择的介质。钢丝留声电话机(Poulson Telegraphone，1900)是第一个使用磁信号的录音系统。在留声电话机中，金属丝由一个卷轴经过录音磁头绕向另一个旋转的卷轴。1924 年斯蒂尔(Stille)发明的叫做磁录声机(Magnetophon)的记录系统使用磁带作为其存储介质。当然磁介质的变迁是围绕着数字计算机技术的发展为中心的。能够存储声学信号这一事实——即使只暂存于随机读写存储器中——导致声音分析的真正进步。

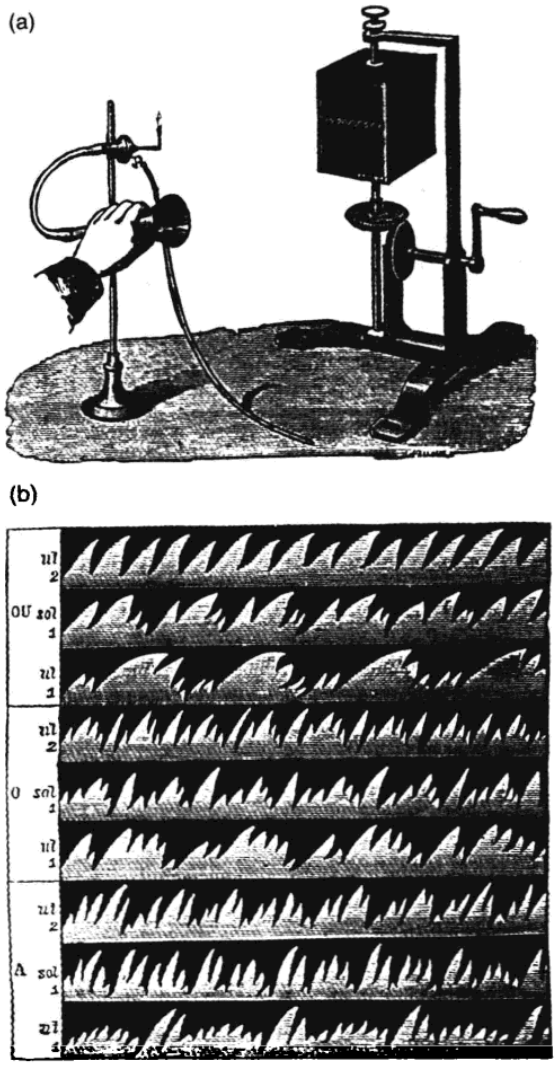


图 12.1　用于分析波形的感压焰。(a)仪器。经吹嘴拾取到的声音调制盒子里的本生灯焰。当旋转盒子时，根据输入声音的音高和频谱，盒外的镜子将火焰投射为边缘参差不齐或锯齿带；(b)由鲁道夫·凯尼格(R. Koenig)制作的，演唱音高分别为 C1(各组最下行)、G1(各组中间行)和 C2(各组最上一行)的法文元音[OU]、[O]和[A]的感压焰图。(出自 Tyndall 1875。)

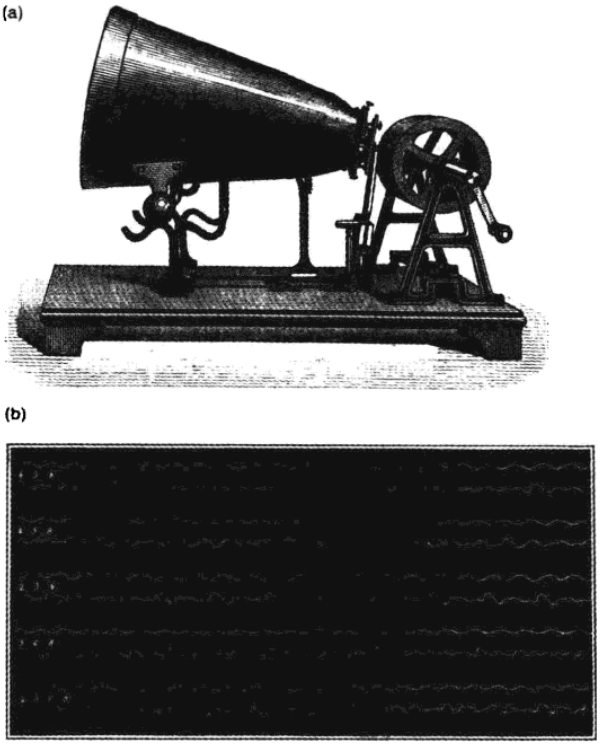


图 12.2 鲁道夫·凯尼格(Rudolf Koenig)版本的用于记录声音波形图像的声波记振仪。(a)仪器;(b)记录结果。

MIDI 系统中的音高、节奏识别 (Pitch and Rhythm Recognition in MIDI Systems)

音高、节奏识别始于以下任何一个出发点:分析原始声音波形或者解析 MIDI 信息流(第 21 章)。显然,后者要容易些。当音乐家演奏一个 MIDI 录入器材如 MIDI 键盘或 MIDI 单簧管时,音高和事件检测是由录入器材本身以电机学方式处理完成的。录入器材内的微处理器时刻监测着琴键、按钮及该乐器其他控制界面的状态。当音乐家演奏时,这些控制器的状态会改变,微处理器侦测这些事件。它为每个事件产生一个随演奏控制变化而变化的,包含开始和结束时间的 MIDI 音符信息。这些来自控制器的信息可以通过 MIDI 线传输给正在电脑上运行的分析软件。这些软件只需解析那些 MIDI 信息就可以获得音高和时间信息。从那里它们可以直接进行更高级别的分析。

对控制器来说,尚存在许多难度很大的音高识别问题。弦乐器给音高识别器制造了严重的麻烦,需要将不同策略合而为一的设计(组合声学及电机传感

器)。并且,如何根据脑电波传感器发出的信号产生出一个"音高"呢? 看来只有更复杂的设计才有可能实现。

本章有关音高识别部分的中心内容是:基于声音波形的分析。MIDI 系统仅当数据流源自音高——MIDI 转换器(PMC)(pitch-to-MIDI converter)时面对这一问题。PMC 试图根据接受到的声音信号产生相应的 MIDI 音高数值(Fry 1992)。节奏识别部分仍旧始自声音波形的分析,但继而转移到同样可以应用于 MIDI 系统的节奏跟踪和记谱等方面。

音高侦测的问题(The Pitch Detection Problem)

"在感知宽度上,耳朵远远超越眼睛;前者范围超过十一个倍频程,而后者可能只比一个倍频程多一点点。"约翰·廷德耳(John Tyndal, 1875)

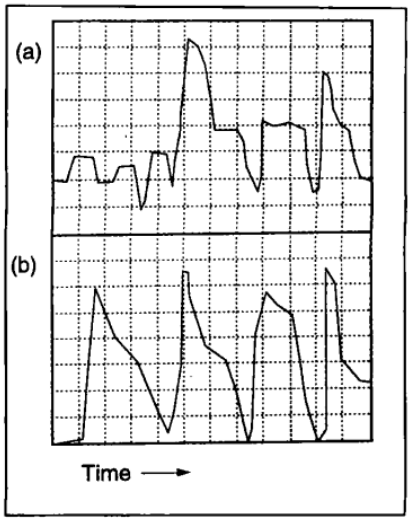
我们可以将音高侦测器(pitch detector)简称 PD,或音高估量器(pitch estimator)定义为:将声音信号作为输入源,试图判定该信号基频周期(fundamental pitch period)的软件算法或硬件设施。亦即其试图找出一个频率,它是人类听觉对该信号认同的音高(假设有这样一个音高存在的话)。一部分原因是由于很多声音的音高概念是模棱两可的,也因为人类音高感知的机理未全部揭开,PD 仅对一些有限的声音有效。试图对诸如吊镲的捶击声、短暂脉冲、低沉的隆隆声或复杂混合在一起的一片声音进行音高识别是没有意义的。事实上,如果我们仔细分析对传统乐器音高的追踪结果,可以发现它们的音高从来没有完美地稳定过,并且充斥着细微变化。在很多音乐应用中,例如现场演奏,PD 的任务是忽略那些细微变化,定位出中央音高。因此我们要求 PD 做的是一件天生就很困难的事。它应该精确,但不能太精确,就像人类的听觉一样。

在音高识别之外,还存在着如何在音乐语境和作品分析当中来理解音高的巨大空间。关于这个层面上的分析在本章讨论之外,不过我们将简要地在后续有关音乐语汇的部分中有所讨论。

音高侦测的应用(Applications of Pitch Detection)

音高侦测在音乐中的应用十分广泛。早期应用来自音乐学家们对捕捉世界各地音乐文化中那些绚丽旋律的需求,比如印度歌者。那些精致的微分音旋律不能由普通的音乐记谱来描述。这类设备之一西格记谱仪(Seeger Melograph),每 4 毫秒就对一个精度为 3/100 倍频程的带通滤波器输出进行扫描,

以找到最大值。第一个最大值被假设为包含基频。在一些处理后，旋谱仪绘制出一个二阶图或叫做旋谱图(melogram)(图 12.3)，表示相对时间的基频、振幅变化(Seeger 1951，Moorer 1975)。西格记谱仪通过计算机技术得到进一步升级，用于对不同旋律运动观念的界定(Gjerdingen 1988)。



Time=时间

图 12.3 与旋谱仪相仿的，对印度歌者声音长度为 2 秒钟的跟踪记录图表。横坐标为时间。(a)对基频的跟踪；(b)振幅跟踪。(出自 G jergingen 1988。)

另一个音高测定的应用是在声音转换方面，声音编辑软件中大都包含对音高移位和时长编辑进行指导的音高测定程序。另一个基于录音室的应用是在记谱方面，如将诸如独奏萨克斯这样的声学乐器用音乐记谱程序记录下来。高级程序如从音高侦测开始，区分同时发声的两个语声(Maher 1990)。在音乐会中，PD 可以帮助合成器跟随乐器演奏家或歌者的表演。随着演奏家的演奏被话筒拾取，信号被送入一个根据所演奏声音产生 MIDI 音符信息的音高侦测器。如果在音高侦测器与合成器之间加上电脑的话，表演模式可以更复杂。在这种情况下，计算机上运行的软件可以指导合成器配和声，或根据演奏者的音高进行变奏。计算机可以让合成器不发声，直到被演奏家演奏的暗示性线索激活。

音高侦测的困难(Difficulties in Pitch Detection)

人类对音高的感知是非常复杂的现象(Goldstein 1973，Moorer 1975，Hermes 1992)。我们的耳朵甚至可以对噪声信号产生音高感。我们可以跟踪同时出现的不同音高(否则和声与复调将成为不可辨认的)，同时还可以识别轻微但富有表现的音高游离(揉音、装饰音、微分音程)。不过，人耳可被诱导从而听到不存在的音高(例如，隐含在泛音列中的基频——一种从任何小扬声器中

都可以听到的效果），并且幻听出音高轨迹〔例如——谢波德音（Shepard tones）表现为持续上行或下行的声音〕。很多声音不能唤起音高感受。我们识别音高的机制还未被完全了解，因为，同时涉及认知处理、主观因素如训练和精通以及内耳的生理结构。

一些 PD 试图模拟出人类音高感知机制的理论模型，但大多数实用设备根据它们的计算效率首先选择较简单的技术。对于实时工作状态下鉴别演奏音高的 PD 来说，效率是至关重要的。在任何情况下，没有百分之百准确的 PD。即使输入信号通过各种方法被限制后，一些高强度计算方法（通常是非实时的）还是可靠的。

触发瞬变值（Attack Transients）

PD 面临的第一个难题就是搜寻出声音的触发瞬变值。通过对许多乐器触发的细节分析揭示混杂及易变的波形。如果触发阶段存在基频，其很可能被噪声或非谐和分音掩盖。一些乐器需要 100 毫秒或更多时间才到达音高稳定期，这个不稳定阶段迷惑了 PD（Fry 1992）。

低频（Low Frequencies）

基于频谱分析的音高侦测器通常有处理低频音高的困难，致使时域 PD 的应用成为必须（Lyon and Dyer 1986）。所有 PD 都有实时鉴别低频音高的问题，为决定基频音高周期，在开始分析前，至少获得三个周期的稳定波形采样。对于低频音高而言，例如 55Hz，采样 3 个周期需要 54 毫秒。如果加上触发瞬变值和音高侦测算法本身所需的时间，那么可被感知的延迟现象是不可避免的。

高频（High Frequencies）

高频也能给实时 PD 带来困扰。随着频率的增高，表示音高周期的样本数就越少。时域可决定音高的分辨率直接受音高周期长度与延迟样本数量的影响（Amuedo 1984）。

短视的音高跟踪（Myopic Pitch Tracking）

所有的 PD 都是从持续 20 毫秒到 50 毫秒长的时间颗粒开始进行分析的，由此其进行分析的时间是非常有限的。相对的，人类的听觉感知并非如此时间

化的。预期形成音高感知，即我们根据音乐语境(music context)估测音高。由于 PD 只对本地细节进行处理，它们有可能短视地跟踪一些非有意产生的无关细节，例如音符开始时的不稳定或额外颤音。

声学环境(Acoustical Ambience)

乐器或人声所在的听音声学环境影响音高侦测的准确率。离声源过近的话筒和狭小的录音空间可能夸大意外的演奏或演唱噪音，例如琴弓声、琴键声或呼吸声，扰乱 PD 听到的信号。相反，“浸泡”在混响或延迟效果中的音符，被相继出现的音符涂抹得面目全非。假设所需分析在非实时条件下进行，并且试图将背景声移除，这些对 PD 也许有帮助。(见 Beachamp, Maher, and Brown 1993 及有关频域音高侦测的描述。)

音高侦测方法(Pitch Dection Methods)

大部分 PD 算法出自语音识别和语音合成研究。从目前已发明的各种复杂方法可以折射出重要的自然属性问题(Gold 1962, Noll 1967, Schafer and Rabiner 1970, Moorer 1973, Rabiner et al. 1976, Hess 1983, Amuedo 1984, Fry 1992, Hermes 1992, Hutchins and Ku 1982, Hutchins, Parola, and Ludwig 1982, Beauchamp, Maher, and Brown 1993)。我们可以将大部分音高侦测方法分为五类：时域(time-domain)、自相关(autocorrelation)、自适应滤波器(adaptive filter)、频域(frequency-domain)和人耳模型(models of the human ear)，将在接下来的部分中介绍。

时域基频周期音高侦测(Time-domain Fundamental Period Pitch Detection)

基频周期法将输入信号看作随时域(TD)波动的振幅。试图发现波形中的重复模式，将其作为线索判断周期。也许“周期侦测器”是描述这类音高侦测器更恰当的术语(Moorer 1975)。

一类音高侦测器通过寻找波形中重复的零交点(zero-crossings)发现周期。零交点是波形振幅由正变负或反之变化的那一点。例如一个正弦波在中间和结尾处跨越零振幅临界值。通过测量零交点间的时间间隔并与相继间隔进行比较，PD 推断出基频(图 12.4)。零交点侦测的另一种变形是测量峰值间的距离(Hermes 1992)。通常零交点和峰值侦测器相对简单且便宜，但与更缜密的

方法相比，它们也具有较差的准确度(Voelkel 1985，Hutchins and Ku 1982)。这是因为其他非音高频点的频率同样也产生跨越零点的波形或产生峰值。图 12.4b 中，作为例子，为跟踪明显可见的基频，PD 必须忽略三个或四个非常快的，由高频分音在每个主零交点产生的低振幅零交点。

滤波器预处理应该可以改善时域 PD 的准确度。库恩(Kuhn)(1990)将输入信号通过一组滤波器的方法改进了基本的零交点方法。然后算法将检查滤波器输出的振幅，仅对滤波后有实际意义的两个最低滤波器的输出实施零交点侦测。最后，对语音和演唱信号而言，电喉音仪(electroglottograph)或喉谱仪(laryngograph)有不少成功的应用。这些方法要求演唱者佩带一个感知声带所产生脉冲的领圈。可是该方法对非语音(口哨)无效，且可能对一些鼻音元音判断失误(Hermes 1992)。和其他所有实时 PD 相同，其同样面临对音符触发的处理问题(Fry 1992)。

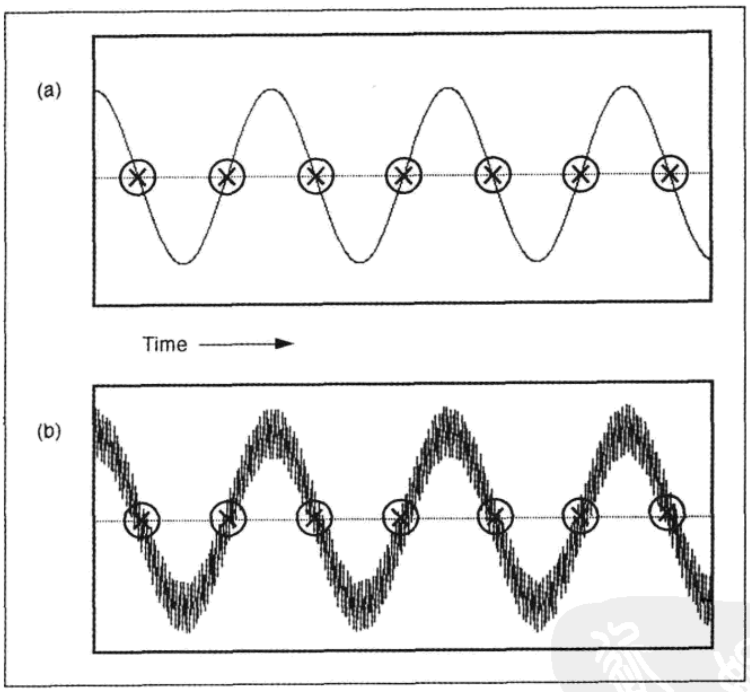


图 12.4　零交点音高侦测器。(a)通过测量零交点(标记为×的地方)间的时间间隔，我们获得一个信号最低周期的线索；(b)假设 PD 忽略由高频分音产生的零交点变体，对强基频信号而言，不管叠置在信号上的高频分音出现与否，该方法同样适用。
Time=时间

自相关音高侦测(Autocorrelation Pitch Detection)

相关(Correlation)函数比较两个信号。相关程序的宗旨是发现两信号的"相似性"(在精确的数学意义上)。相关函数是点继点地进行信号比较,由此相关函数的输出本身就是一个信号。如果相关函数为1,表明该点上两个信号绝对相关。如果为0,则两信号为不相关。

自相关(Autocorrelation)法将信号与它相继一定间隔的另一版本的自身进行比较,互相关(cross-correlation)法是在一定延迟或滞后(lags)时间范围内与另一个不同信号进行比较。对延迟版信号进行比较的目的是为了发现重复模式——信号周期的标识。这是我们在此将关注的周期侦测方法。

自相关音高侦测器将部分输入信号放置于缓冲区中(Moorer 1975, Rabiner 1977, Brown and Puckette 1987)。随着更多相同信号的输入,侦测器试图将进入的波形与存储的波形进行模式配比。如果在假定错误范围准则内侦测器发现一个匹配,这表明一个周期,侦测器测量两个模式出现间的时间间隔,从而估量出周期。图12.5为自相关音高侦测器的图解。

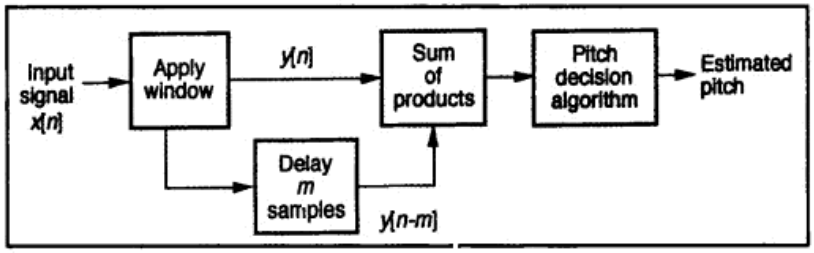


图12.5 自相关图解。输入信号被取样窗分割,取样窗中的片断与比其晚一个采样、两个采样直至 m 个采样的片断进行比较。最大相关被评估为主要基频音高。
Input signal=输入信号 Apply window=实施窗取 Delay m samples=延时m个样本
Sum of products=积之和 Pitch decision algorithm=音高决定算法 Estimated pitch=估测音高

自相关算法多种多样(Moorer 1975)。对已知延迟或滞后(lag)时间,典型的自相关函数如下所示:

$$autocorrelation[lag]=\sum_{n=0}^{N} signal[n]\times signal[n+lag]$$

其中 n 是输入样本指数,且 $0<lag\leqslant N$。不同次序上的第 n 个信号,与经滞后样本延迟后的同一信号的值相等,两者之积构成的和数列的次数决定 $autocorrelation\ [lag]$ 的大小。自相关输出表明不同滞后时间的量。

正弦波的自相关性表示出这一原则,见图12.6,情况(a),$lag=0$,两个函

数是一样的。因此，由正弦波乘方规定的自相关函数为 1。自相关函数如图 12.6 底部所示。现在假设正弦波延迟四分之一周期，如情况(b)所示，该周期内 *signal* [*n*]与 *signal* [*n*+ *lag*]的积为 0。情况(c)中，延迟为半个周期，相关性为-1。情况(d)，延迟为四分之三周期，相关性为 0。最后情况(e)，延迟为整个周期，因此相关性为 1。由此我们可以看出，正弦波的自相关仍是正弦波，它在输入正弦波的周期的整数倍处取最大值。

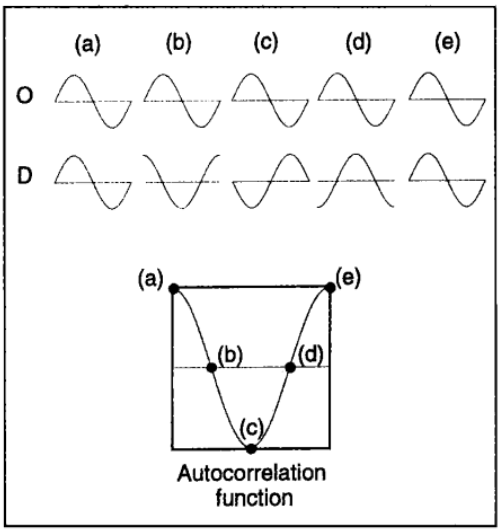


图 12.6　正弦波的自相关本身就是正弦波形。O 表示原始信号，D 表示延迟信号。文字解释情况(a)至(e)。自相关函数如底部图示。
Autocorrelation function= 自相关函数

对更复杂的信号而言，PD 程序在自相关中搜寻循环的峰值，指出(有可能隐含掉)输入波形中的周期(图 12.7)。

通过自相关进行音高侦测对中低频率最有效。这是为什么其在语音识别等有限音域应用中非常流行的原因。在音乐应用中，音域更广阔，对每秒钟输入信号的直接相关计算，需要数百万次乘法与加法运算。另一种计算自相关信号的方法是按照一定规则将信号分割成片断，然后对每个片断施加快速傅里叶变换，这将比直接运算明显地提高速度。有关该算法的细节请参考 Rabiner and Gold(1975)。

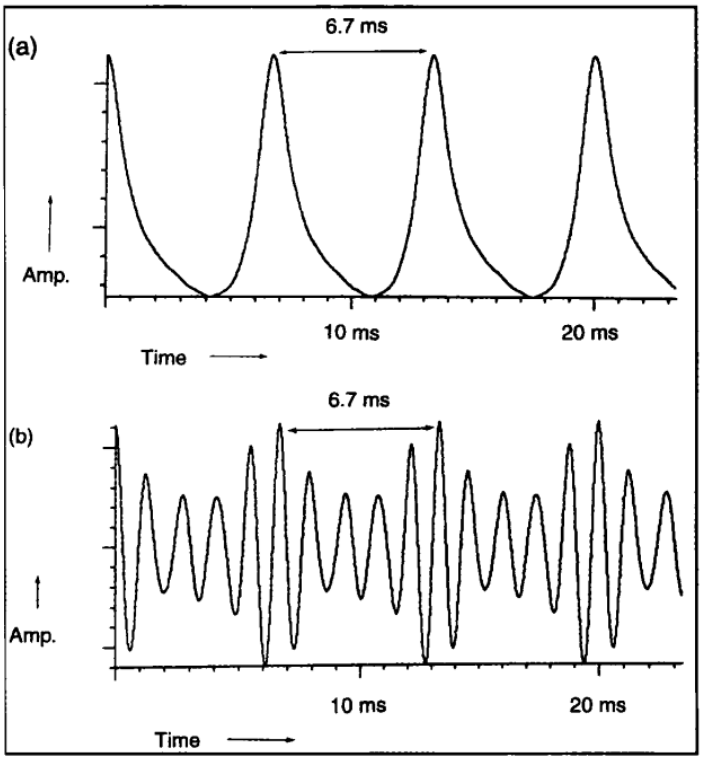


图 12.7 周期性信号的自相关函数本身也是时间的周期性函数。(a)具有五个泛音某信号的自相关,包含基频周期为 6.7 毫秒或 149Hz(接近音高 D3)。自相关是周期性的,但其泛音振幅与输入不同。请注意峰值与基频的关系;(b)只有第五、第六和第七共三个泛音的信号的自相关。自相关是周期性的,其周期与波形缺省基频(隐含音高)周期同为 6.7 毫秒(出自 Moorer 1975)。
Amp.=振幅 Time=时间

自适应滤波音高侦测器(Adaptive Filter Pitch Detectors)

正如名字中所蕴涵的那样,自适应滤波器是根据输入信号,以自我调协的方式工作的。基于自适应滤波器(adaptive filter)的音高侦测方法是将输入信号送入窄带宽的带通滤波器中。滤波后与未滤波信号都被送入差动检波器(difference detector)电路。差动检波器电路的输出被反馈给带通滤波器以控制其中央频率(图 12.8)。这种控制迫使带通滤器将频率集中到输入信号上来。聚合检测度量滤波器输出 $y(n)$与输入 $x(n)$的差。当差接近 0 时,系统作出一个音高判断。

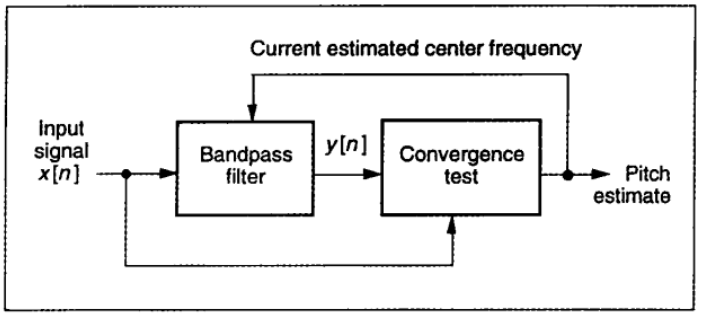


图 12.8　基于自适应滤波器的音高侦测器示意图。请注意由估算音高反馈给滤波器的连接。
Input signal=输入信号　Current estimated center frequency=当前所估测的中央频率
Bandpass filter=带通滤波器　Convergence test=聚合检测　Pitch estimate=估测音高

另一种自适应滤波器技术是最佳梳滤法(optimum comb method)(Moorer 1973)。该方法寻找一个可以最小化其输入信号的梳状滤波器(第 10 章介绍梳状滤波器)。为使输入信号最小化,梳状滤波器的凹间必须调谐为输入信号的主要频率。由此,通过找到最佳梳状滤波器,找到主要音高。此方法主要适用于基频很强且泛音分布规律的声音。

有关更多自适应滤波器的内容请参考 Lane (1990)、Hush et al. (1986)和 Hutchins (1982—1988)。

频域音高侦测(Frequency-domain Pitch Detection)

频域(FD)音高侦测方法将输入信号分解为构成全部频谱的频率。频谱展示出信号所包含的各频率分量的强度。目的是将显要频率或"音高"从频谱中分离出来。

典型的 FD 方法利用短时傅里叶变换(STFT)分析连续的输入信号片断(更多有关傅里叶分析请参考第 13 章和附录)。FD 音高侦测器在频谱中搜寻与显著频率一致的峰值。当找到峰值后,音高侦测器必须决定哪个频率是基频(通常被感知为音高),哪些只是泛音或外部分音(Kay and Marple 1981)。快速实时的 FD 音高侦测器可能简单地选择最强频率为音高,更复杂的侦测器会审视隐含基频的泛音关系。基频可以不是最强分量,但由于多个泛音的强化,其有可能是最显著感知音高。

基于 STFT 的音高侦测器有这样一个问题:STFT 将音域带宽分成等距的道(channel)或线(bin),道与邻道间相隔 nHz。由于人类音高感知基本上是对数型的,这意味着对低音的追踪精确度要小于高音。例如,分辨率为 20Hz 的分析可以解析 10 至 20Hz 间的微分音,但低于中央 C 时需要提供小于半音的分辨率。(在 1 万至 2 万 Hz 之间,20Hz 可以被解析为一个微分音,而在中央 C

以下音区，20Hz 则已经超过一个半音——校注）频谱最低区域精确的音高分辨率需要大量的分析道。正如第 13 章中所述，增加分析道的代价是时间效率的损失。选择其他方法或许更适合低频的音高跟踪（见第 13 章有关这些问题的进一步讨论）。

追踪相位声码器法分析（Tracking Phase Vocoder Analysis）

与固定频道的 STFT 相反，追踪相位声码器法（tracking phase vocoder）（TPV）容许改变频率（McAulay and Quatieri 1986，亦请参考第 13 章）。TPV 开始于 STFT 产生的数据，之后，产生一套轨迹（track），每个轨迹代表频谱中的一个显著分音。轨迹可以及时改变频率，贯穿固定分析频带进行插值。这种跟踪处理暗含着数据量的减少，因为只跟踪最显著的分音，TPV 削弱了外部噪声和环境声，产生了一个“消毒”版的输入信号。

马赫耳（Maher，1990）、比彻姆（Beauchamp）与布朗（Brown，1993）开发出一种由 TPV 输出开始的 FD 音高侦测器。此系统在被跟踪的频率中扫描，并且通过各种方法将它们与假想基频的泛音频率进行比较。总差异最小的假想者被估量为基频音高。

图 12.9 为系统产生的三个标绘图。在图 12.9a 中，系统精确地跟踪计算机合成版本的 J. S. 巴赫的《第三组曲》（*Partita* Ⅲ）。图 12.9b 表明当面对小提琴演奏的录音版本时系统的降格表现。音符间的小误差是系统被弓子噪音混淆时的结果。图 12.9c 展示了当对高混响环境中录制的小提琴进行分析时，由于“和弦效果”（新音符出现时，旧音符仍然在发声），系统表现出进一步降级。

作为对这种系统的改进，发明者们对经 TPV“消毒”后的同一小提琴录音进行相同算法分析。随着数据量的减少，TPV 从录音中去掉了噪声和摩擦声，包括弓擦声及混响。当 PD 对这个重新合成版本进行分析时，准确性高了许多。

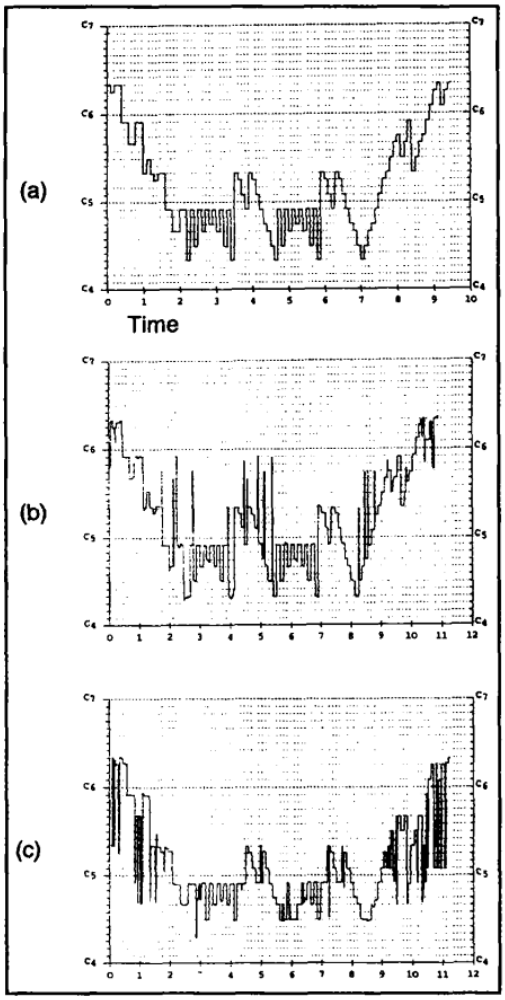


图 12.9　对 J. S. 巴赫第三组曲前八小节通过频域音高跟踪得到的音高侦测图。纵坐标是以半音为单位的 12 平均律音节，从 C_4 到 C_7。横坐标为时间。(a)计算机合成音高；(b)录音室录音；(c)带混响录音。(出自 Beauchamp、Maher 和 Brown 1993。)
Time=时间

普言谱(对数倒频谱)分析(Cepstrum Analysis)

在语音研究领域，一个普遍应用的 FD 音高侦测方法就是对数倒频谱(cepstrum)技术，此技术原先应用于语音分析(Noll 1967，Schafer and Rabiner 1970)。普言谱(对数倒频谱)经常与第 5 章中介绍过的，线性预测编码(linear predictive coding)(LPC)共同使用。通过将频谱(spectrum)一词英文的前四个字母，按照相反顺序书写，得到了普言谱(cepstrum)这一专有名词。一种关于普言谱的简单描述就是将某个具有强音高感觉的分量从其他频谱中分离出来。很多人声及乐器

声音的频谱可以被看作是激励(excitation)(以声音音高为代表的原始振动脉冲)与共振(resonances)(由乐器腔体或发声道产生的声音中被滤的部分)的和,这是一个合理的模型。(第7章,物理仿真部分介绍了有关激励/共振的概念)

技术上讲,普言谱是对数傅里叶频谱的傅里叶逆变换,是离散傅里叶变换(discrete Fourier transform)的输出的对数(以10为底)的绝对值。(附录介绍有关离散傅里叶变换及其反转。)

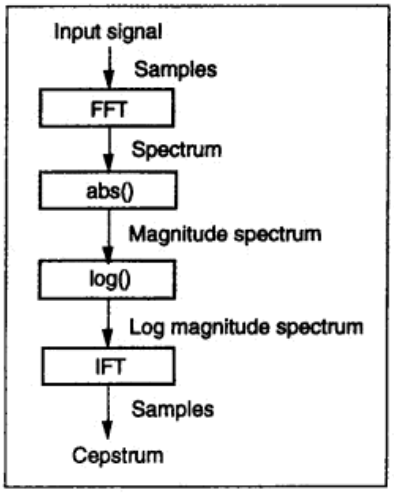


图 12.10 普言谱计算流程示意图
Input signal=输入信号
Samples=样本
FFT=快速傅里叶变换
Spectrum=频谱
abs()=取绝对值
Magnitude spectrum=幅度谱
log()=取对数
Log magnitude spectrum=对数幅度谱
IFT=傅里叶逆变换
Cepstrum=普言谱(对数倒频谱)

普言谱计算的结果,像输入信号本身一样,是一个时间序列。如果输入信号有一个强基频音高周期,在普言谱中将显示为一个峰值。通过测量时间0至峰值时间的时差,找到该音高的基频周期(图12.11)。

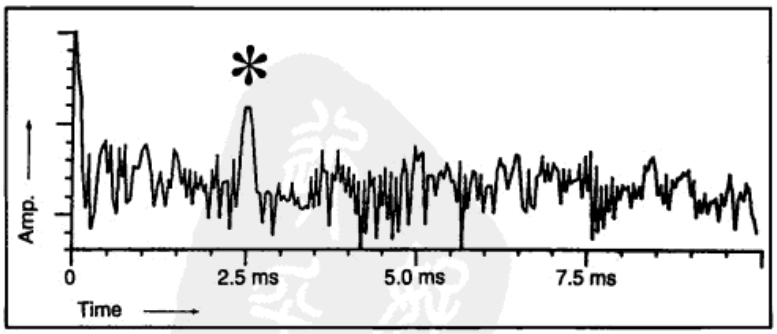


图 12.11 混响大厅中音高为396Hz的小号独奏的普言谱图。由星号标注的峰值表明信号的周期,大约2.52毫秒,与被侦测音高相当。请注意,即使在混响存在的情况下,普言谱中的峰值仍旧很清晰地出现。(出自 Moorer 1975。)
Amp.=振幅 Time=时间

普言谱分析如何在语音分析中发挥作用的呢?普言谱用于将两个叠生频谱分离:自声门发出的脉冲(声带)激励和发声道共振。激励可被视为准周期脉

冲序列。对这些脉冲的傅里叶变换是一个以原频率泛音相间的线性频谱(见图 12.12 中的起伏摆动线)。取对数操作不影响频谱的整体构成。倒傅里叶变换产生脉冲的另一个准周期波形。与此相对照的,反应发声管道(作用像滤波器)的频谱是一个缓慢的频率变异函数,如图 12.12 中粗线所示。取对数及傅里叶逆操作产生一个其中只有很少样本,并有显著振幅的波形,通常少于基频音高周期。其关系如下所示:脉冲响应衰减为 $1/n$,则普言谱衰减为 $1/n^2$。由此,普言谱将脉冲响应聚集为普言谱波形开始的短脉动,其将音高聚集为以基频频率为周期的一系列峰值(见图 12.11)。

普言谱计算有很多应用因为其倾向于从激励中清理出脉冲响应。换言之,普言谱倾向于反卷积(deconvolue)两个卷积(convolved)频谱(Smith 1981)。(参见第 10 章中有关卷积的内容。)之所以我们称之为"倾向于"是因为对音乐信号的解卷积(deconvolution)很少完美无缺。普言谱操作中的取对数运算倾向于聚集频谱中这两个准分离分量。这里不再介绍更复杂的运算,也不再介绍这些功能可以被滤除以便使普言谱中包含无论与音色还是音高联合的频谱信息。(细节请参考 Noll 1967, Schafer and Rabiner 1970, Rabiner and Gold 1975, Rabiner et al. 1976。)

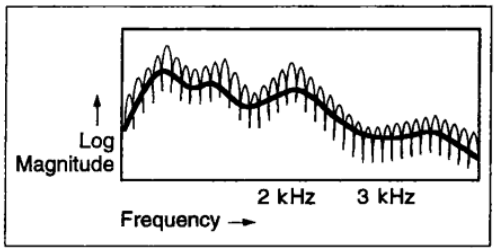


图 12.12　普言谱对声带脉冲响应和发声道脉冲响应的分离。对数操作将细摆线(相当于激励)与粗长起伏频谱(对应于脉冲响应或共振)分离。
Log Magnitude=对数振幅　Frequency=频率

基于人耳模型的音高侦测器(PDs Based on Models of the Ear)

经数十年的系统研究,听觉科学集中于对人类听觉系统结构的细节了解上。声音分析领域有这样一种倾向,将听觉知识与超级计算机技术相挂钩(Hermes 1992, Slaney and Lyon 1992),从而获得对声音微观结构新的洞察力。这些模型的应用之一就是音高侦测。现在的音高侦测器已经综合了基于已知的人类听觉系统结构模型的感知理论的算法。利克里德(Licklider)的音高感知理论预示了该领域的现代应用(Licklider 1951, 1959)。

图 12.13 展示了这样的音高侦测器的总体结构,其中分为三个子模型:外耳

及中耳模型、耳蜗模型及中枢神经系统模型。第一步是基于外耳和中耳响应的滤除预处理阶段。下一阶段通过带通滤波器的平均值将输入信号转换成频域表征。接下来是换能阶段，将基底膜能量转换为神经触发可能与随后发生的少许时域尖峰信号(Meddis, Hewitt, and Schackleton 1990)。至此，其过程基于很完善设立的、简化的科学化数据。接下来的阶段是最冒险的部分，模拟中枢神经系统处理输入的尖峰信号。目标是测量尖峰信号间的周期并估测其最大频次间隔或音高。最后阶段与时域和自相关音高侦测器类似。综合频域与时域的优势在于当频域的频道被转换成时域尖峰信号时，不和谐"污物"被滤除了。

复音音高侦测(Polyphonic Pitch Detection)

当有音高感的声音与噪声或多个音高声音同时出现时，所有音高侦测的困难都被放大了。这是复音记谱(polyphonic transcription)(根据声学信号产生书面乐谱)面临的一个难题。大多数人类音高感知理论主要讨论有关单个音高的听觉问题。对人类复音听觉的机理还知之甚少。

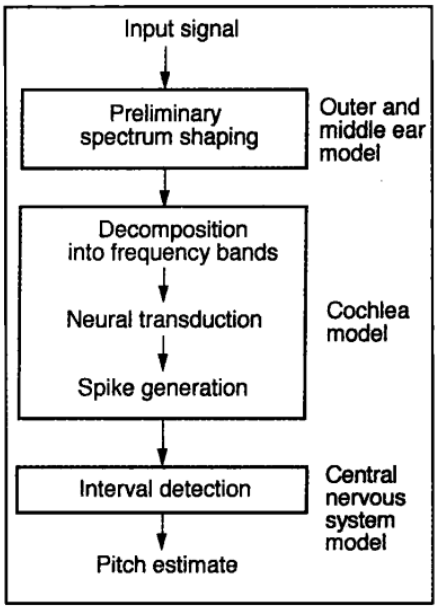


图 12.13 基于人类听觉系统模型的音高侦测器示意图。
Input Signal=输入信号
Preliminary spectrum shaping=频谱预塑型
Outer and middle ear model=外耳及中耳模型
Decomposition into frequency bands=分解为频带
Neural transduction=神经换能
Spike generation=产生尖峰脉动
Cochlea model=耳蜗模型
Interval detection=时值侦测
Central nervous system model=中枢神经系统模型
Pitch estimate=音高估测

复音音高侦测通常在查询和决定机制中应用频域分析技术。主要目标是从包含很多振幅峰值的频谱中筛选出旋律线，其中那些峰值有可能是基频音高也有可能是强泛音。为决定哪些峰值可能是基频音高，分析必须从不同角度检查数据并为不同因子赋予不同权重，在结果中进行评估(Moorer 1975, Maher 1990)。经常采用由人工智能派生出的技术，如在整个显著频率清单中进行预

期驱动(expectation-driven)搜索。当它们使用了分析方面的知识指导查询策略的时候,系统被称为预期驱动型(Moorer 1975, Terhardt 1982, Chafe et al. 1982, 1985, Foster et al. 1982, Strawn 1980, 1985a, b; Maher 1990)。(参见第 13 章有关信号理解系统方面的内容。)由于额外的数据收集、查询和复杂的抉择算法,复音音高侦测所需的计算时间远比单音情况下大得多。

音乐语境分析(Analysis of Musical Context)

在很多表演环境中有必要从单纯的音高侦测迈进至音高分析——通常意义上的旋律和声分析。更确切地说,音高分离已经出现,它们的音乐意义是什么,它们包含哪些含义?这一任务的另一个名称是音乐语境分析(analysis of musical context),实例之一是判断某调性音乐的调性和谱号(Chafe et al. 1982, Holtzman 1977)。从该分析开始,进一步的目标可能是出于记谱目的为每个音符制定相应的音符名(例如升 F 或降 G)。

在交互表演系统中,计算机应当对人的表演做出相应的反应,由此,必须很快地辨别音乐语境。已经研究出各种快速分析和弦及旋律的算法,其通常为使用该系统的作曲家要求的风格而定制(Chabot, Dannenberg, and Bloch 1986, Roads 1985b; Rowe 1992a, b)。在快速算法之外是计算机辅助音乐风格分析的广阔领域,有关该领域的话题超出了本书的内容范畴。

节奏识别(Rhythm Recognition)

演奏出普通乐谱上的节奏是音乐院校训练获得的基本技能之一。与之相关的就是识别演奏的节奏并将它们转记为乐谱。对于这些技能而言,从初学者到大师需要经过长期的训练过程。音乐节奏的记谱看起来是机械的计数工作,应该很容易教会机器。结果发现其远比最初想象复杂得多。此外,节奏听写技能已经被简化,因其是基于对韵律相关节奏的识别之上的。很多节奏没有规则的拍子,且任何节奏性分组(包括那些没有简单韵律关系的)都能以韵律性结构呈现。由此,节奏识别的总体问题仍悬而未决(一个很好的有关音乐节奏理论的概论见 Yeston 1976,其中引述了许多自远古时代的早期理论家。)

声学信号节奏的机器识别将输入样本列入由单个声音事件组成的表。给这些事件赋予音符时值(二分音符、四分音符等),然后将其分组为更大的音乐单位:符干组、二连音(三连音等)、小节或许还有同样决定着拍子的乐句。这些任务天生就是困难重重的,部分原因是音乐记谱具有模糊性,因此人们对乐谱的演奏不

是完美精确的。即同样或很相似的节奏可以按照不同方式记谱。与音高侦测相同，节奏识别必须忽略“无意义”的偏差以便抽离出“实质”节奏。例如，其必须知道一个轻微顿奏的全音符，不是一个用连线连接起来的二分—四分—八分—十六分—三十二分音符的集合。这涉及音序器中的量化(quantization)问题，但面对声学信号时要困难得多，因为需要由系统找到音符列表，并且在一开始时速度是未知的。

试图将音乐按照节奏乐句划分的系统，一开始就受制于这样一个事实，即“乐句”是基于音乐语汇与风格的概念。此外，音乐专家们对某些音乐的乐句结构也不是总能达成共识的。

节奏识别方法的多样性恰似音高侦测的翻版，但有一个很重要的不同。音高侦测植根于长期以来对语音及信号处理等更广阔领域的研究，而节奏识别研究只针对音乐领域。[赛福瑞吉(Selfridge)和内塞尔(Neisser)1960 年有关计算机解析莫尔斯电码的研究是很少有的特例之一]。结果是研究及标准更少。除此之外，不同的音乐任务和音乐风格需要不同的对待，其中不仅只是节奏识别的问题，还有很多其他问题。

节奏识别的应用(Applications of Rhythm Recognition)

对声学信号源的节奏识别在很多应用中有用武之地，包括跟踪实时表演的速度、判断拍号以及作为音乐自动记谱的组成部分之一，在音乐学及音乐表演研究中也有应用。

速度跟踪(Tempo-tracking)算法试图跟着声学信号“打拍子”，可以被变异成自由或突变速度的函数。在音乐会中，当计算机伴奏系统试图跟随演奏家或歌唱家的表演时，这是很有用的。

根据特定音乐应用将音符表解析成不同级别的节奏单元，直至达到所需为止。交互式即兴系统可能只搜寻很少的节奏型或可以触发它的线索。其内存是短期的，当未发现期待音型时将继续往下寻找，丢弃先前的输入。伴奏程序则一直试图将输入节奏型与内存中的乐谱相比较。其试图牢固地锚住拍子以便紧紧地“跟着它”。印刷乐谱的记谱系统必须时刻保持与输入信号一致。其努力地发现拍号、估测出小节边界，且为每个音符赋予适合的时值。将声学信号全自动地记谱为印刷乐谱的系统是真正意义上的人工智能问题，因系统必须使用一套分析方法，且从各个层面在多重臆测中选择。不仅必须精确地描绘时值、休止，对一些特殊情况如多连音、倚音、装饰音和连线，也必须按照自然记谱方式来生成。同时进行的音高和振幅分析应该可以帮助节奏分析系统做出适合的音符分配。该领域中还遗留着许多问题，尤其是对复音的记谱。

节奏识别级别(Levels of Rhythm Recognition)

节奏分析可以发生在三个级别：

- 低级：事件检测
- 中级：转记成乐谱
- 高级：风格分析

低级阶段，输入信号流必须被转换成数字形式，之后将其分割为开始和结束时间片断表，以便抽象出音乐事件。中级阶段，输入信号流已经被分割和编码，这与使用键盘发出的 MIDI 数据的情况一样。这时的任务是从分割的数据中将音符表转换成有意义的乐谱。音符分配和分组是这个级别最主要的子任务。高级节奏分析涉及作曲理论及分割分析范畴，取决于具体应用内容。因为音乐可以按无数的方法划分成高级结构(Roads 1985d，e)，我们这里只讨论前两个级别的内容。

事件检测(Event Detection)

低级节奏分析集中解决事件检测——从音频流中分离出单个时间并决定它们的时值。

振幅阈限(Amplitude Thresholding)

对于非混响房间录制的单音简单音乐，可以通过时域技术如振幅阈限(amplitude thresholding)(Foster et al. 1982，Schloss 1985)解决时间侦测问题。该方法中，系统扫描输入波形寻找时间的振幅包络，特别是明显的触发和衰减曲线。如果看到有超过既定振幅阈限的触发包络出现，则表明某时间的开始。通过高通滤波器得到暂态值(transients)(锐利突起和衰减发生点)可以提高该方法的效率。

然而振幅图容易成为时间突发和时值的误导暗示。一些音乐本身就很难只靠时域技术来划分。例如连弓的弦乐触发、被延留音符、混响模糊了的新音符或像和弦这样的复音信号。这些情况中，连续振幅包络可能密封了许多事件，甚至遮蔽了那些起重要节奏作用的重音事件(Foster et al. 1982)。如踩有延音踏板的颤音琴，其振幅图就不能作为音符触发时间的清晰向导(图12.14)。这种情况下，音高和频谱变化是新事件的绝佳线索。

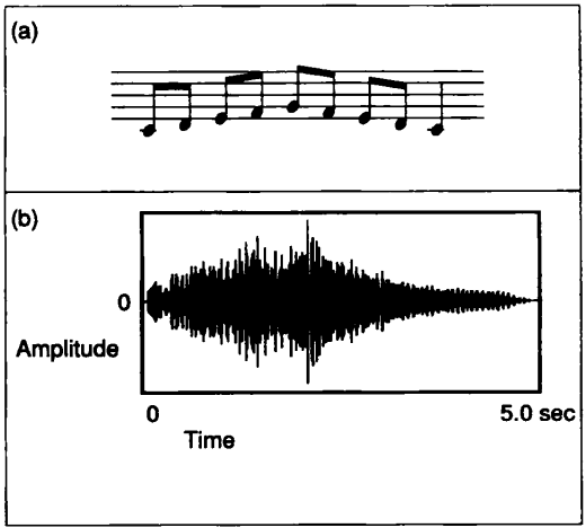


图 12.14 时域事件检测的问题。(a)一系列音符;(b)踩着延音踏板的颤音琴演奏的这些音符产生的时域信号。
Amplitude=振幅 Time=时间

因此时域与频域相结合或许是最有效的方法(Chafe et al. 1985, Piszczalski and Galler 1977, Piszczalski et al. 1981, Foster et al. 1982)。例如简单振幅阈限方法失败时,基于自回归(autoregression)(AR)模型的频域事件分割器可以成功地工作(Makhoul 1975, Foster et al. 1982)。自回归侦测信号周期中的变化,使其对音高变化敏感。但是重复的等音将不被 AR 模型识别,AR 和振幅阈限一起工作将非常有效,因为 AR 是频率敏感,阈限是振幅敏感(见第 13 章有关 AR 技术的介绍)。

区分复音音乐中的音乐声部(Separating Voices in Polyphonic Music)

在复音音乐中区分出每个声源或声部是困难的。超出了相当级别的难度,目前是不可能实现的。还没有人试图将每个乐器的每个音符从交响乐队全奏中分离出来。在具备足够处理能力的前提下,对少量各自差异的乐器而言,问题容易驾驭一些(Moorer 1975, Foster et al. 1982, Wold 1987)。除提及过的方法外,复音声源分离中使用如下方法:

- 通过滤波将乐器分离(比如短笛和大号)。
- 如果在多轨录音中声源是很宽的公布的话,将空间声场分布作为线索。
- 将输入信号与频谱模板(spectrum templates)(某已知乐器频谱样式)相比较,以便将某音色分离出来;该模板很可能基于某乐器的物理仿真模型(Wold 1987)。

■ 在频谱中找到共有的揉音和震音模式(频率和振幅调制),它们表明哪些分音是由哪些演奏乐器产生的。在心理声学研究中,该方法被称为源相干判据(source coherence criteria)(Chafe and Jaffe 1986)。

■ 鉴别每个乐器特质的触发样式,因为即使是和弦,乐器也很少精确地同步开始。

综合各种方法进行事件识别的系统,当进行某一特定尝试时,问题接踵而至。当系统采用了多个技术时,需要有判断不同技术方法所得结果权重的手段,同时得出某一答案。更多相关内容参见第 13 章有关信号理解系统的部分。

记谱(Transcription)

"任何已知的音符数据流在原则上都拥有无穷可能,但这种不确定性对听者来说很少凸显出来。"(H. C. Longuet-Higgins 1976)

记谱——中级节奏识别——开始于离散事件被组合在列表中。基于 MIDI 的节奏识别从这一点开始着手。记谱包括以下子任务:速度跟踪、节奏赋值、音符分组、决定拍号、设定小节边界以及可能的基础乐曲结构的清理。这里我们将分别对待,但实际上它们是相互关联影响的。

记谱的最终目的未必是将其打印出来,也可能是将分析数据送入交互式作曲程序、演奏系统、音乐学分析程序或音乐倾听模型。因为目的不同,对乐谱的分析很可能就不同。

速度跟踪(Tempo Tracking)

速度跟踪试图发现"拍子"——以相同时值划分时间间隔的一种被感知的脉动。商业的记谱软件通过让音乐家跟随软件产生的节拍机声音而解决了这一问题,但这里要讨论的是没有节拍声音参考情况下的,更困难的速度跟踪,即对真正音乐表演演奏的跟踪(Rowe 1975, Pressing and Lawrence 1993)。

速度跟踪的第一步是测量事件间的时间长度。测量可被用做设置分等级的度量格(metrical grid)。拍子通常是被测量时值的共同特征。该过程听起来简单,但速度变异使栅格偏离,使最开始估测基础脉冲困难重重。同时,如果有切分节奏模式,速度跟踪器必须设法知道那些不在拍子上的重音音符并没改变拍子。

有一个方法可以简化该任务的复杂程度,在有限的时值窗中搜寻,比如 5 秒钟一次(Miller, Scarborough, and Jones 1992)。对过往拍子进行衰减记忆的历史机制方法与此类似(Dannenberg and Mont-Reynaud 1987, Allen and Dannenberg 1990)。短记忆忽略过往事件,容许强速度起伏,但似乎不很稳定,

长记忆稳定些，但代价是忽略了强速度变化。

图 12.15 为并行采用的两种方法速度跟踪器示意图。图左上部分是对"重要事件"的提取，这些事件在音乐中起结构性作用。这种应用受启发于易识别的节奏或旋律重音常发生在重拍这样重要的结构性位置上，由此锚定点之间的关系通常是简单的。由于不是所有情况下都如此，图 12.15 右上部分说明了跟踪速度起伏的单独处理方法，该过程在连续时值中查找重复的模式，并不断统计最共通时值。重要时值间的关系通常比较简单，锚点与锚点时值间的关系也是如此。通过这两种方法的结合，速度跟踪决定选择一个合理的当前速度假设。该方式的适应性在出现切分——在错位拍出现的锚定点，而重要时值继续跟踪速度的情况中体现出来。相反，当锚定点给出强烈暗示，主速度将顺应而进行调整。

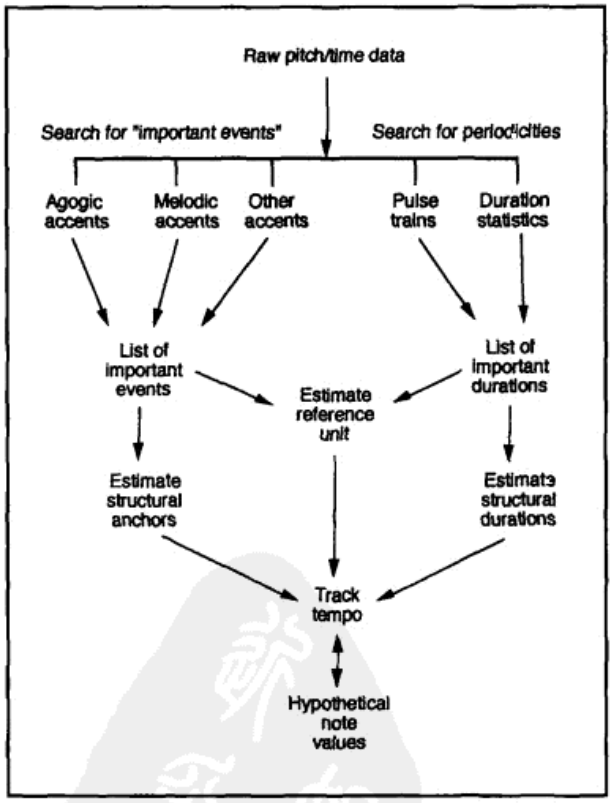


图 12.15 芒特-瑞诺德(Mont-Reynaud)的速度跟踪器。内容见文本。
Raw pitch/time data=原始音高流/时间数据 Search for "important events"=寻找"重要事件"
Search for periodicities=寻找周期 Agogic accents=离拍重音 Melodic accents=旋律重音
Other accents=其他重音 Pulse trains=脉冲队列 Duration statistics=时值统计
List of important events=重要事件列表 Estimate reference unit=估测参考单元
List of important durations=重要时值列表 Estimate structural anchors=估测结构锚定点
Estimate structural durations=估测结构时值 Track tempo=跟踪速度
Hypothetical note values=假定音符值

指定音符时值(Note Duration Assignment)

假定拍子很稳定,可以给每个事件指定一个与律动相关的时值。如果演奏很机械的话,这将很容易做到。但富于表情的演奏中,那些想象中的等时值音符,呈现出相当大的变化。在音乐演奏中有很多将重要音符时值伸展了的离拍重音。

另一类速度跟踪方法基于贯联性策略(D'Autilia and Guerra 1991, Rowe 1992a, b)。在这些系统中,节点网络表示相互结合事件间的时间间隔。这改变它们的数值以便使它们之间互为简单有理数倍。在理想的条件下,这些数值定义了度量格。

为使推断律动相关时值的工作容易些,分析程序可能会对音符时值进行量化(quantize),即将它们近似为诸如八分音符或十六分音符这样的律动相关时值。实用记谱程序通常在记谱前向演奏者要求暗示,比如要求他们设定代表量化栅格的最小演奏时值。即便如此,德塞恩(Desain)和霍宁(Honing,1992c)的比较研究表明,商用记谱软件中采用的简单量化栅格方法,可以导致有缺陷的结果。来自他们论文的图 12.16 表明了当程序以六十四分音符为栅格量化三连音时的情况。问题之一是标记有 A 的音符比 B 演奏得短,即使它们被记成另一种近似谱。还有其他的量化方法,包括那些基于贯联性模型的方法,但似乎每种都各有其局限(见第 16 章图 16.10)。

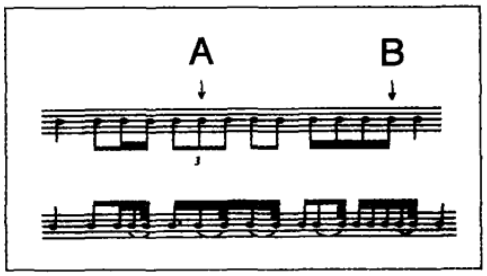


图 12.16　量化的危害。(a)正确的音乐记谱;(b)以六十四分音符为量化栅格的商用音乐编辑软件转记的乐谱。

节奏型分组(Grouping into Patterns)

下一步识别是将列表中的音符细分成组或节奏型。图 12.17a 为分组处理的起点:一些没有任何小节线或拍号迹象的音符时值。程序如何像图 12.17b 中所示那样,推断出应该在第一、第七和第十四个音符处插入小节线?其如何决定第二、第三和第四个音符是三连音?

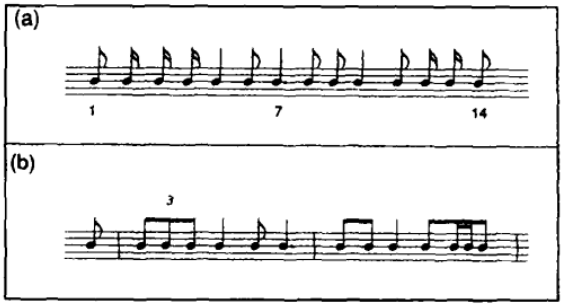


图 12.17 节奏分组问题。(a)节奏分析器得到的一组音符;(b)对(a)的似是而非的转记。

不同的应用程序按照不同的原则将音符分组。一个记谱系统可能按符干分组音符,例如一组八分音符。试图模仿人类听觉模型的程序可能会尝试建立乐句的层次体系。根据假设的拍号按小节将音符分组,我们将在下一部分讨论相关内容。

节奏型识别目前为查询和比较技术所垄断(Rowe 1975, Mont-Reynaud 1985b, Nont-Reynaud and Goldstein 1985)。一些节奏分析的准语法理论,如勒戴尔(Lerdahl)和杰肯道夫(Jackendoff)(1983),希根斯(Longuet-Higgins)(1976, 1987)以及龙盖特-西金斯(Longuet-Higgins)和李(Lee)(1983),作为分析算法的指导发挥着作用。例如罗森塞尔(Rosenthal)(1988)引用勒戴尔(Lerdahl)和杰肯道夫(Jackendoff)理论中的五条规则,提出了应用在简单音乐节奏中的逐步预案。这里我们将其作为典型的分组原则列举如下:

1. 由重音开始分组。
2. 不构成单独事件的组。
3. 短时值事件倾向于与后继的长时值事件分在一组。
4. 分组线将长时值事件与后续短时值事件分割开。
5. 尽可能地按照等时值在同级别上进行分组。

应该被强调的是,这些理论衍生自书面的,而且是不一定被演奏过的音乐。因此在实际操作中,这些算法通常会被来自试验的经验规则所修正。更复杂的规则,举例而言,可能出于解析相互竞争的节奏推论的目的,将音高和振幅模式计算在内(Katayose and Inokuchi 1989, Katayose et al. 1989)。

作为基于规则的模式分类器的另一种选择,采用神经网络模型的联系法(connectionist methods)曾被应用(Desain and Honing 1989, 1992b, 1992c; Linster 1992)。

估测拍号及小节线(Estimating Meter and Measure Boundaries)

拍号是两个时间级别的比值。其一是拍子的周期(例如每四分音符等于一

秒钟），其二是基于一个固定拍数的更大的周期——小节。拍号通常强加给拍子重音结构，一个倾向于形成小节的结构。决定拍号可被分成以下两个问题：首先是根据可以被整数 n（如二倍、三倍、四倍、五倍）整除的循环节奏型找到被感拍号（perceived meter）。这通常是音乐倾听的模型和交互式作曲程序的目的。其次是估测整个音乐的确切拍号（如是$\frac{2}{4}$而不是$\frac{4}{4}$），在记成可印刷乐谱时将面临该问题。

由于节奏关系的不确定性，估测被感拍号和以小节划分音乐的任务不是一帆风顺的（Rosenthal 1992）。罗森塞尔（Rosenthal）的策略是配制多重专门任务，各任务分别收集音符分布、音符时值、重音、音高特征和节奏型等不同分析结果。每个任务提出一种假设，然后管理程序负责在多重假设中进行抉择。其通过标记某任务比其他更可靠（由此获得更大权重），当该任务同意某推测时，看起来应该就是正确的那一个。米勒（Miller）、斯卡博拉夫（Scarborough）和琼斯（Jones）（1992）把基于规则和贯联性策略的拍号估测方法进行了比较。基于规则的方法略显古板，且优劣是可预见的。贯联性策略，由于其灵活性，可以处理基于规则方法无法完成的情况，例如对变速的估测。但有时贯联性策略会做出荒唐的估测，表明了预测和转译贯联性分析器输出结果的普遍困难。

对精确拍号的估测相当困难，部分原因是很多拍号听起来都一样。假设根据不同情况调整速度的话，某旋律可以按照$\frac{1}{2}$、$\frac{2}{2}$、$\frac{2}{4}$、$\frac{4}{4}$、$\frac{4}{8}$、$\frac{8}{8}$等拍演奏，且听起来一样。给某节奏指定拍号需要了解该音乐创作风格等相关知识。例如，18 世纪维也纳时期创作的音乐很可能遵守惯例，这为选择拍号划定了界限。目前程序能完成的最好情况通常是根据音乐风格做出一个猜测。对于经常变换拍号的现代音乐而言困难显然更大。再一次重申，商业记谱软件中，音乐家可以预制拍号，因此程序不会面临这样的困难。

恢复（Recovery）

狂热的演奏、节奏的模糊性、音头不清晰的低振幅段落，或某些特殊段落的识别漏洞等情况都可以迷惑节奏识别器。因此，任何实用节奏识别器必须试图从混淆处平稳地恢复，就像音乐家那样重新聆听。这个主题很复杂，且恢复策略取决于表演任务，因此我们这里仅将其作为问题提出来。正如艾伦（Allen）和丹南伯格（Dannenberg）（1990）强调的，如果系统主张演奏的多重诠释，这在根本上有可能减少完全混乱情况的出现。

结论(Conclusion)

教会机器如何听音乐的困难过程告诉我们,人类的感知系统是如此的敏锐和强大(Carterette and Friedman 1978, Buser and Imbert 1992)。人类从声源中分离出声音(方向上和音色上),跟随复杂和声、复调和复合节奏几乎立即从两三个音符片断唤起对整个音乐的记忆等能力——使得目前这一代机器识别器黯然失色。

机器识别音高和节奏的方法仍然在发展中,工作在非实时状态下的语音基音侦测器"非常可靠"(Hermes 1992),尽管对于音乐而言,本身可靠的音高侦测器更难一些,即使我们将复音音乐排除在外(几乎所有音乐!)。

当提供节拍机参考拍,且预制量化范围的话,节奏值侦测是可操作的。但在没有这些明显暗示的情况下,节奏识别不能一贯地得出合理结论。成功系统只可以解决最常用的节奏规则。即便我们开发出新的解析规则,然而对音乐而言,任何一个规则都有例外。因此太多音乐超出了机器节奏识别器能驾驭的复杂程度。更多了解人类的节奏感知似乎是这一领域的关键。

第 13 章　频谱分析 (Spectrum Analysis)

“那些富于创造力的音乐家们，如果同样深谙有关纯科学方法以及他们艺术作品中的材料，不就能成为更优秀的大师吗？如果了解了各成分的自然属性以及它们产生的效果，他们不能够用更优异的技能混合声音的颜色吗？”(D. C. Miller, 1916)

正如一个影像可以被描述为是颜色(可见电磁波频谱中的频率)的混合一样，一个声音目标可以被描述为声学振动元素的聚合。解析声音的方法之一是考量各成分的分布，每个成分代表某空气压力的变化比率。测量这些成分的均衡关系叫做频谱分析(spectrum analysis)。

实际工作中对频谱的定义是“以频率函数的形式测量信号能量分布”。由于不同的分析技术各自测量“频谱”，其结果多少带有趋异性，所以这是目前最贴切的定义。除个别的测试情况外，频谱分析的实践不是精密科学〔详细内容请参考马泼尔(Marple)1987〕。其典型结果为实际声谱的近似值，由此称之为频谱估量(spectrum estimation)或许更恰当。

频率分析的应用很广泛。本章内容涉猎广泛但不可能涵盖所有可能的方法。鉴于该内容的技术性，本章我们的主要目的是不时地将深奥的概念还原为音乐实践中的术语。作为对本章内容的补充，附录更详细地介绍了傅里叶分析。

频谱分析的应用(Applications of Spectrum Anaylsis)

频谱图揭示人声、乐器和合成声音的微观结构(Moorer, Grey, and Snell 1977, Moorer, Grey, and Strawn 1978, Piszczalski 1979a, b; Dolson 1983, 1986, Stautner 1983, Strawn 1985a, b)。由此它们是声学家和心理声学家们必需的工具(Risset and Wessel 1982)。

音乐学家们正日益转向语图和其他声音分析技术以便研究电子音乐的表演和结构(Cogan 1984)。进而扩展为音乐的自动转记——从声音到乐谱——不是以普通音乐记谱就是以图形方式(Moorer 1975, Piszczalski and Galler 1977, Chafe et al. 1982, Foster et al. 1982, Haus 1983, Schloss 1985)。

实时频谱分析是交互式音乐系统的某种“耳朵”。频谱分析揭示乐器和人声音色之特质频率的能量，由此可以帮助鉴别音色并将它们从同时演奏的复音色声源中分离出来(Maher 1990)。如第 12 章中所述，频谱分析的结果常常对音高和节奏识别很有价值。

但音乐家们不仅仅想分析声音，他们希望编辑分析数据和对原声音进行变体再合成。越来越多的声音转化技术起始于分析，包括时间压缩和扩展、频率位移、卷积(convolution)(滤除和混响效果)，以及多种两个声音间的交互合成

混血。基于频谱分析的技术可以通过对被分析音色的再合成实现“自然”与“合成”音色间的连续变换(Gordon and Grey 1977, Risset 1985a, b; Serra 1989)。有关分析与再合成的更多内容请参考第 4 章和第 5 章。

频谱图(Spectrum Plots)

现存有多种测量和绘制频谱的方法。这里集中讨论两个基本类别:静态(static)(好比频谱的相片)和时变(time-varying)(好比频谱的电影)。

静态频谱图(Static Spectrum Plots)

静态图捕捉声音的静止画面。这些声音快照投射出相对频率的振幅二维图,分析测量了被分析片断时间内每个频率的平均能量。时间段或取样窗可以小到瞬间,大至几秒钟或更长不等(稍后我们将讨论有关权衡取样窗的问题)。

离散谱(discrete spectrum)或线状谱(line spectrum)是静态频谱图之一,其中每个垂直线代表每个频率分量。对大部分泛音,最清晰的分析是音高同步(pitch-synchronous)。该类分析测量那些可以预先决定音高的乐音的泛音振幅。图 13.1a 所示为稳定状态小号的线状谱,使用音高同步技术测量。注意在该测量时刻,第三泛音振幅高于基频。

图 13.1b 所示为按对数振幅(分贝)显示的另一小号频谱。对数振幅将该图压缩在一个垂直窄带中。通过描绘峰值的轮廓可以看出总体的共振峰形状。

图 13.1c 所示为“啊”声的连续型(continuous)频谱,其中通过图形插值的方法将分析器测量的离散点连接起来。单个正弦分量被隐含,但频谱的总体形状更为清晰。

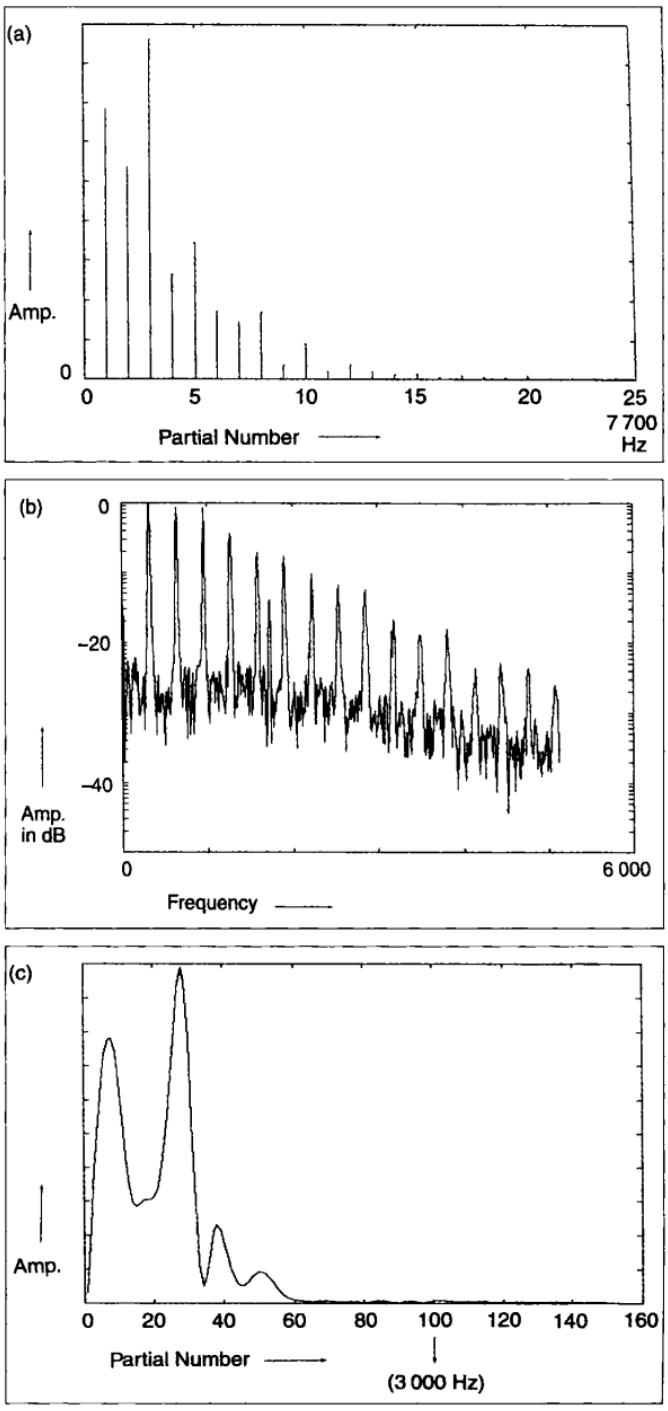


图 13.1 静态频谱图。(a)延留阶段小号相对频率的振幅直线谱。每条直线代表基频 309Hz 某一泛音的强度。以线性关系显示振幅;(b)以压缩的垂直窄带对数关系(分贝)图来表示振幅的(a)中小号的频谱;(c)连续型频谱,线性关系表示振幅,展示出"啊"声共振峰轮廓。〔刊出图形授权自皮奇阿利(A. Piccialli),那不勒斯大学物理系。〕

Amp.=振幅
Amp. in dB=按分贝显示的振幅
Partial Number=分音序号
Frequency=频率

幂频谱(Power Spectrum)

从振幅频谱可以得到幂频谱(power spectrum)。物理学家将幂定义为信号振幅的平方。由此,幂频谱即是振幅频谱的平方。以幂的形式表示频谱胜于振幅的形式,因其与人类听觉更相关。另一种测量是幂频谱密度(power spectrum density)或 PSD,适用于像噪声这样的连续频谱。对 PSD 的简单定义是特定带宽内的幂频谱(Tempelaars 1977)。

时变频谱图(Time-varying Spectrum Plots)

即使是独奏乐器的频谱细节也是不断变化的,因此静态、恒定频谱图只可以表示声音演化中的某一段。时变频谱描绘某事件时值内的频率变化。其可被绘制为按时间变化的三维频谱图(图 13.2)。那些图实质上一个接一个地将一系列静态频谱图连在一起。

图 13.3 展示了时变频谱分析的另外两种格式。图 13.3a 是瀑布图(waterfall display)的照片——时间轴实时运动的频谱图。瀑布图这一术语得自该类频谱图按照液体描述的方式表现频率能量的上升和下降波动。图 13.3b 描绘的是人声旋律。

另一种表示时变频谱的方法是绘制语图(sonogram)或声谱图(spectrogram)——语音分析中的常用工具,原被称为可视语言(visible speech)(Potter 1946)。语图以相对时间轴的频率展示信号,纵坐标为频率,时间在横坐标,用轨迹的明暗度表示频谱中频率的振幅。高强度频率分量以深色表示,低强度频率分量以浅色表示(图 13.4)。我们将在以后详细介绍语图。

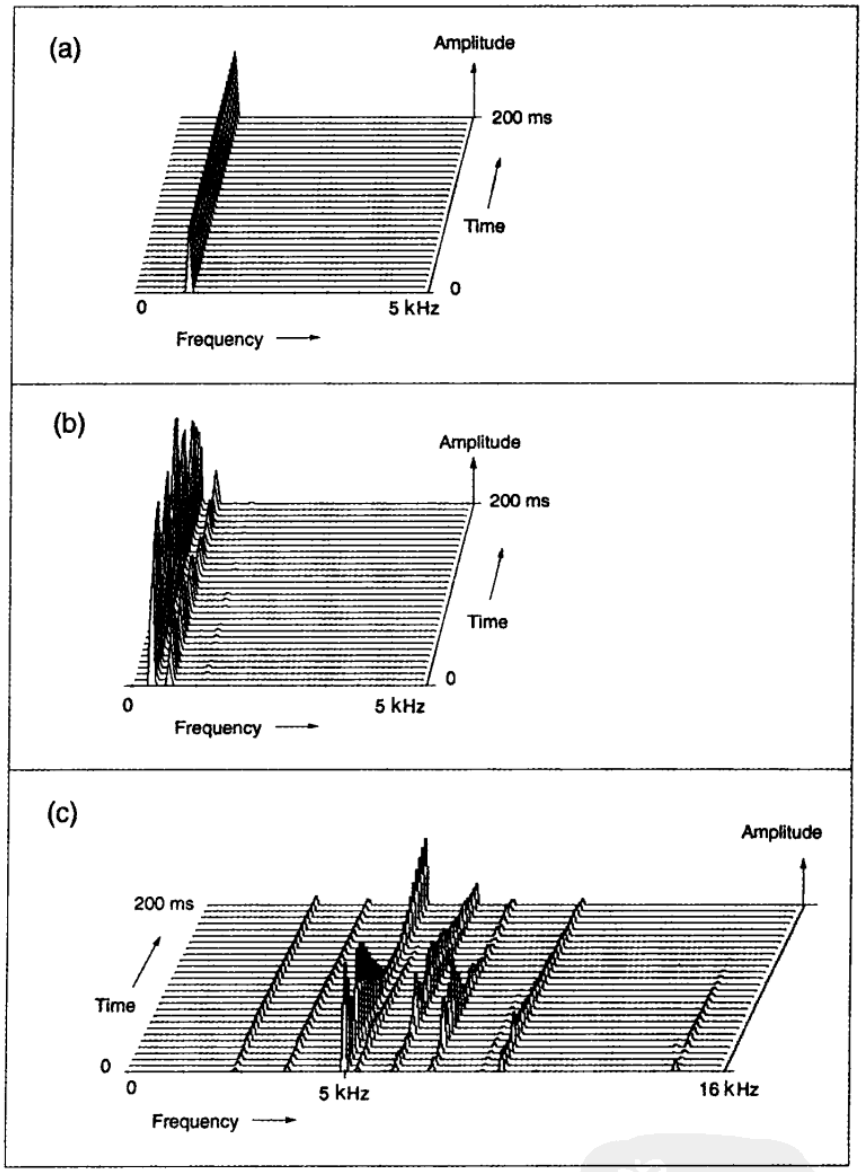
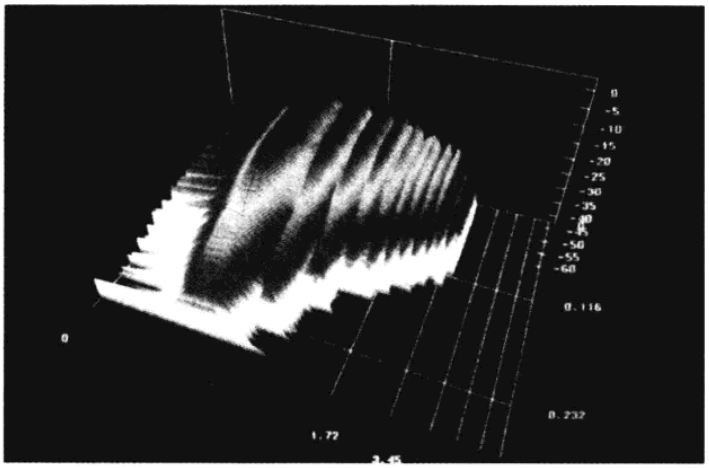


图 13.2 以线性关系表示振幅的时变频谱图。时间从前向后。(a)1kHz 正弦波;(b)音高为 E4 的花舌长笛;(c)敲击一次的三角铁。
Amplitude=振幅 Frequency=频率 Time=时间

(a)



(b)

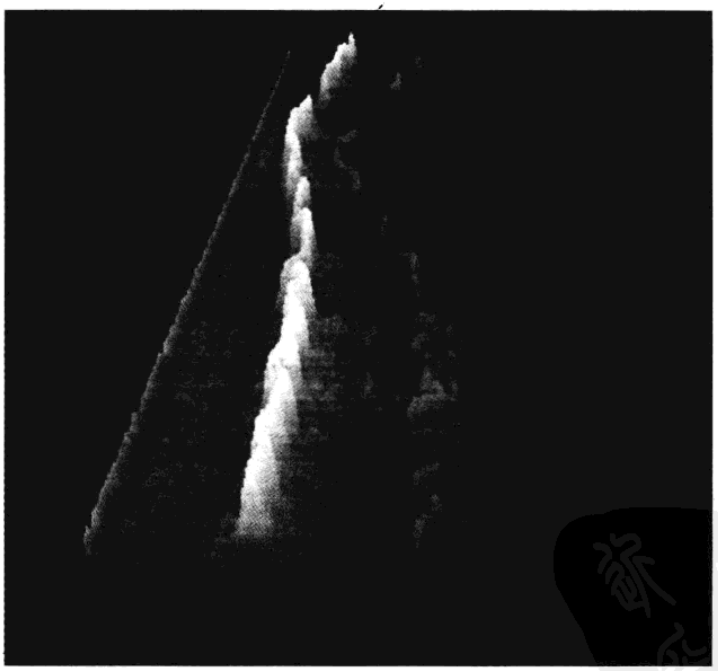


图 13.3　实时瀑布图的照片。(a)合成小号音色。时间由后向前，最近时间在最前。频率按对数关系从左至右展示。基频大约是 1kHz。振幅在纵坐标以对数关系(分贝)表示。(b)人声旋律。时间由远至读者，最近时间在最前。左边是低频。〔图片授权自伯克利，加利福尼亚大学、新音乐和艺术科技中心，皮弗斯(A. Peevers)〕

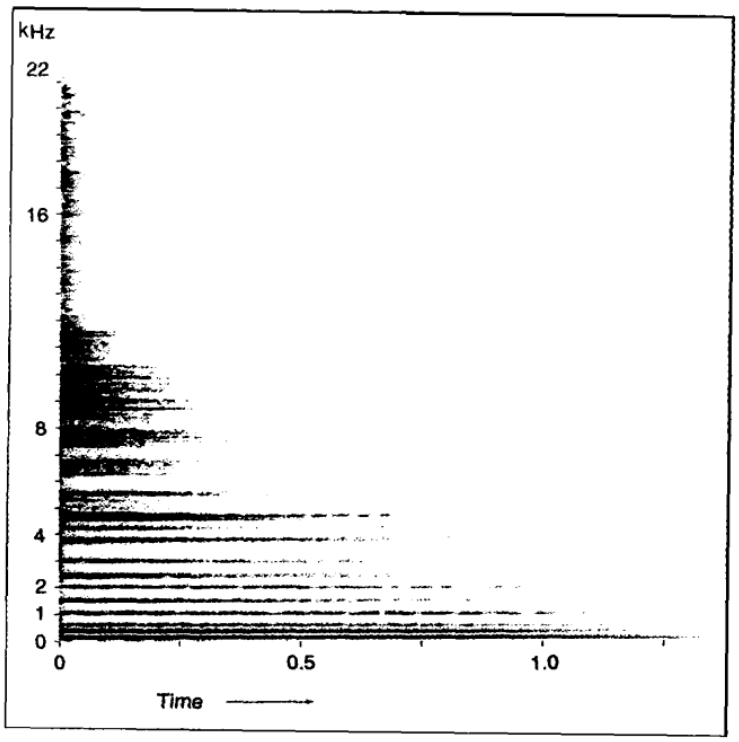


图 13.4 敲击锣声的语图。纵坐标为频率，横坐标为时间轴。该语图使用输入数据的1 024点及汉明窗(Hamming Window)。频率分辨率为43Hz，时间分辨率为1毫秒。分析带宽为0至22kHz，测量动态范围从－10分贝至－44.5分贝，以线性关系绘制振幅。
Time=时间

频谱分析方法背后的模型(Models Behind Spectrum Analysis Methods)

“对于分析、合成某种声音而言似乎都没有具普遍意义或最佳的范例。一方面细察声音——准周期的、不谐和分量总和、噪声、快速或慢速演化；同时也研究声音的哪些特征与人耳有关。”(Jean-Claude Risset 1991)

没有哪一个频谱分析方法可以理想地适用于所有音乐应用。傅里叶分析——最通行的方法——实际上是由许多还在发展中的不同技术构成的一个家族。许多非傅里叶方法被持续开发出来，我们将在后面介绍。

每一个声音分析技术都可以被看作是将输入数据装在假设的模型中。基于傅里叶分析的方法将输入声音定型为谐波相关的正弦波集合——其可能是也可能不是正弦波。其他技术将输入信号定型为由共振滤波的激励信号，或按指数衰减的正弦波或方波，或为非谐波关系的正弦波的集合，或为附加着噪声的共振峰组或为代表传统乐器演奏行为的一组方程。能想到的其他方法不胜

枚举。在后面的详介中我们可以看出，不同方法实施中的偏差通常可以被认为是被分析过程与假设模型间匹配度的结果。因此根据特定的音乐应用内容选择相应的分析方法是很重要的。

频谱和音色(Spectrum and Timbre)

术语“音色”是一系列现象的载体。像“洪亮”(sonority)和“理想音色(klangideal)(Apel 1972)”这些模糊的术语，或许某一天会被精确描述声音质量的词汇所替代。对音乐音色的分类是一门古老的科学。早期中国文人发明了精妙的、描述音色的记录法，包括对乐器发生源的分类〔金属、石头、陶、皮革、丝线、木、葫芦和竹子(译注：金、石、土、革、丝、木、匏、竹)〕，并且精细记录下演奏古琴丝弦时的不同“指触”(触发型、拨和揉音)(Needham, Ling, and Girdwood-Robinson 1962)。实际上，古琴演奏的主要技巧之一就是在相同音高上演奏出不同音色。

频谱和音色是相互关联的概念，但它们并不相同。频谱是一个可以被描述为频率函数的能量分布的物理特征，而怎样测量这些能量恰好是另一个问题！心理声学家们用“音色”一词代表将声音分类归属的感知装置。根据这一定义，音色至少与声音信号的感知有关。显然在传统乐器和人声领域讨论音色是最容易的，几乎所有以往的研究者们都专注于此。只有很少在此范围外进行声音全域分类的尝试，最勇敢的应属皮埃尔·舍费尔(Pierre Schaeffer)的研究(1977，也可参见 Schaeffer, Reibel, and Ferreyra 1967)。

通常音色将在乐器上以不同音高、响度和时值演奏的声音集合成一组。无论演奏哪个音符，例如，我们总能得知是一架钢琴在演奏。人类感知可以将相同音高、响度和时值演奏的每个不同乐器音色区分开。没有人在区分同音高、响度和时值的马林巴与小提琴音色上有很大困难。当然单个乐器也可以产生许多音色，好比不同强度吹奏萨克斯管可以获得不同的洪亮度。

很多因素影响着音色感知，包括声音的振幅包络(特别是触发形状)、基于揉音和震音的波动、共振峰结构、被感知的响度、时值以及时变频谱包络(time varying spectral envelope)(随时间变化的频率内容)(Schaeffer 1977, Risset 1991, McAdams and Bregman 1979, McAdams 1987, Gordon and Grey 1977, Grey 1975,1978,Barrière 1991,还可参见第 23 章)。

在辨别乐器声源音色时，对感知而言音色的触发阶段远比稳定阶段(延留)重要(Luce 1963, Gery 1975)。传统乐器家族中如簧片乐器、铜管乐器、弦乐器和打击乐器，各自有独特的触发“签名”，其对识别那些由它们演奏出来的音符

十分重要。

振幅和时值对音色感知有影响。例如，60dB 的长笛音色频谱中的频率比例关系可能与放大到 120dB 时一样，但我们听起来后者只是个吵闹的汽笛。同样，时长 30 毫秒和历时 30 秒的音色也许有着同样的周期波形，但听众在判断它们是不是同一声源时或许会有困难。

这里要说明的观点是频谱不是感知音色的唯一线索。通过仔细地检视时域的波形而不进行细节的频谱分析，也可以发现很多关于声音音色的内容(Strawn 1985b)。

频谱分析：背景(Spectrum Analysis: Background)

18 世纪，科学家和音乐家们清晰地发现许多音乐声音有围绕根音的泛音振动这一特点，但他们没有以系统地方式分析这些泛音的技术。牛顿于 1781 年发明了“光谱”用来描述透过玻璃棱镜的表示不同频率的离散颜色带。

1822 年法国工程师让-巴普蒂斯 约瑟夫(Jean-Baptiste Joseph)和傅里叶男爵(Baron de Fourier，1768—1830)发表了其里程碑式的论文《热分析原理》(*Analytical Theory of Heat*)。在这篇论文中，他发现了复杂振动可以被分析为许多同时存在的简单信号集合的理论，特别是傅里叶证明了任何周期函数可以被描述为无限正弦和余弦项之和。由于傅里叶分析中正弦频率间的整数倍关系，其被称作谐波分析(harmonic analysis)(关于傅里叶分析简史请参考附录)。1843 年，纽伦堡工学院(Polytechnic Institute of Nürnberg)的格奥尔格·欧姆(Georg Ohm，1789—1854)第一个将傅里叶理论应用于声学信号(Miller 1935)。不久，德国科学家亥姆霍兹(H. L. F. Helmholtz，1821—1894)推测乐器音色很大程度上取决于乐器音色静止状态时的傅里叶泛音列。亥姆霍兹(Helmholtz)基于机械声学共鸣器发明了一个谐波分析的方法。

将亥姆霍兹的术语 Klangfarbe(“声音颜色”)翻译为声音着色(clang-tint)的英国物理学家约翰·廷德耳(John Tyndall)将音色描述为“两个或更多个音的混合”，并完成了将声音信号视觉化的富有想象力的试验，如“歌唱的火焰”和“歌唱的喷泉”(Tyndall 1875)。

机械频谱分析(Mechanical Spectrum Analysis)

手动的机械波形分析仪开发于 19 世纪晚期和 20 世纪早期(Miller 1916，也可参阅附录)。巴克豪斯(Backhaus)(1932)开发出每次单个泛音的分析系

统。由可调节的带通滤波器和连接在其输入端的炭精话筒组成。滤波器输出被送入放大器，其输出依次与一只笔和磁鼓记录器相连接。巴克豪斯将滤波器调整到某一泛音频率，然后指挥演奏家演奏一个音符。随着音乐家的演奏，巴克豪斯用曲柄旋转磁鼓的同时，笔在卷动的纸上描绘出该频率的滤波器输出。结果轨迹被用来描绘单个泛音的举止。迈尔(Meyer)和布赫曼(Buchmann)(1931)开发过类似的系统。

20 世纪 40 年代示波器的兴起带动了新研究的浪潮。科学家们将示波屏上的波形拍照，之后手绘出它们的轮廓至机械傅里叶分析仪。

理论上的飞跃出现在诺伯特·维纳(Norbert Wiener)的经典论文《广义谐波分析》(*Generalized Harmonic Analysis*)(Wiener 1930)中，将强调谐波分量的傅里叶分析推广到连续光谱范畴。在其他结果中，维纳表明，以白光类推，白噪声是由等量的所有频率构成。布莱克曼(Blackman)和图基(Tukey)(1958)描述了通过采样数据施行的维纳法实践。20 世纪 50 年代早期出现计算机之后，布莱克曼-图基法是最通行的频谱分析方法，直到 1965 年，往往归功于库利(Cooley)和图基的快速傅里叶变换(fast Fourier transform)的应用之前。(见 Singleton 1967，Rabiner and Gold 1975，及附录中更多有关 FFT 历史的内容。)

大多数前计算机时期的分析，如米勒(Miller)(1916)和霍尔(Hall)(1937)，都平均看待乐器音色的时变特性。与亥姆霍兹(Helmholtz)的研究一样，这些研究假设稳定阶段的频谱(音符的持续或“延留”)在音色感知中扮演重要角色。正如前文所述，现在已经知道乐音触发阶段的前半秒钟比稳定阶段在识别乐器音色时更重要。

加布尔(Dennis Gabor)在声音分析领域先锋性的贡献未发生即时影响(1946，1947)，但现在被视为根基，主要因为他提出了一个分析时变信号的方法。在加布尔的理论中，声音可以同时在时域和频域被分析为其称之为量子(quanta)的单元——现根据所使用分析系统被称为颗粒(grains)，小波(wavelets)或取样窗(windows)。我们将在本章以后的内容中讨论小波分析及取样窗。

基于计算机的频谱分析(Computer-based Spectrum Analysis)

早期计算机乐器音色分析需要勇敢的尝试。模数转换器还很稀有，计算机很匮乏，理论不成熟，并且必须在穿孔卡片上从零开始编写分析程序(图 13.5)。在对抗这些不利障碍的背景下，相对建立模拟模型而言，发展于 20 世纪 60 年代的基于计算机的分析与合成，获得了更多细节上的结果。在贝尔电话实验室(Bell Telephone Laboratories)，马克斯·马修斯(Max Mathews)和让-克劳德·里塞(Jean-Claude Risset)使用音高-同步(pitch-synchronous)程序

分析了铜管音色(Mathews, Miller, and David 1961, Risset 1966, Risset and Mathews 1969)。音高同步分析将输入波形分成伪周期段(pseudoperiodic segments)。分析系统估测每个伪周期片段的音高。根据估测的音高周期调整分析区间(analysis segment)的大小。之后通过分析区间计算出谐波傅里叶频谱,仿佛声音是周期性的,好像音高贯穿全部分析区间是一个准常量。程序可以产生已知基频所有泛音的振幅函数。卢斯(Luce)在麻省理工学院的博士研究(1963)中采用了另一种音高同步方法进行乐器音色的分析/再合成。

图 13.5 詹姆士·比彻姆(James Beauchamp)1966 年在伊利诺伊大学(University of Illinois)进行声音分析试验。

几年后,伦敦电子音乐工作室(Electronic Music Studios)的彼得·金诺维夫(Peter Zinovieff)和他的同事们开发了一种模拟——数字混合型的音乐声音实时傅里叶分析/再合成系统(Grogorno 1984)。

外差滤波器分析(Heterodyne Filter Analysis)

进一步的计算机音乐音色分析与外差滤波器(heterodyne filters)有关(Freedman 1965, 1967, Beauchamp 1969, 1975, Moorer 1973, 1975)。外差滤波器法对解析已知基频的泛音(或准泛音)有用,这意味着基频是在分析前期被估测出的。外差滤波器将输入波形乘以频率为泛音频点的正弦和余弦波,在很短时间内相加得出结果,以便获得振幅和相位数据。

图 13.6a 所示为外差法的实施。输入信号与分析用的正弦波相乘。在图

13.6a 中，两个信号的频率恰好匹配，所以能量都为正，表明在分析频点处有很强的能量。图 13.6b 中，两个频率不同，因此我们得到了一个沿振幅轴基本对称的波形。当外差滤波器在很短时间内把该波形相加时，其基本上被其自身抵消了。

经过 20 世纪 70 年代一段时间的实验后，外差法的局限开始逐渐被熟知。穆勒(Moorer)指出外差滤波器法被短触发时间(小于 50 毫秒)及大于 2 个百分点(大约四分之一音)的音高变化(例如滑奏、滑音、揉音)所迷惑。虽然比彻姆(Beauchamp)(1981)采用了可以跟踪频率变化轨迹的跟踪(tracking)版外差滤波器(与后面要讨论的声码器神似)，外差法已经被其他方法取代了。

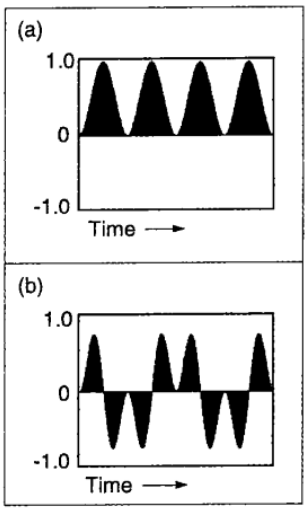


图 13.6　外差滤波器分析。(a)输入信号(100Hz 正弦波)与分析信号(100Hz 正弦波)相乘的积。结果全部为正，表明 100Hz 处有强能量；(b)输入信号(200Hz 正弦波)与分析信号(100Hz 正弦波)相乘的积。结果能量离散在正负两端，表明输入信号在 100Hz 处没有强能量。
Time=时间

相位声码器传奇(The Saga of the Phase Vocoder)

最流行的频谱分析/再合成技术之一就是相位声码器(phase vocoder，PV)。贝尔实验室的弗拉纳根(Flanagan)和戈尔登(Golden)于 1966 年开发出首个 PV 程序。本来是用于减少语音带宽的编码方法。与压缩音频数据相去甚远，PV 产生了数据爆炸！因为分析数据流远大于原始信号数据。

PV 需要强计算能力。早期的实施需要太久的计算时间，因此，很多年 PV 没能付诸在实际应用中。在麻省理工学院工作的波特诺夫(Portnoff；1976，1978)开发出了一种相对高效的 PV，证明通过 FFT 其便可行。他以语音变形为试验，如压缩和扩展时间。这导致了穆勒有关计算机音乐中 PV 应用这一里程碑式论文的出现(Moorer 1978)。

20 世纪 70 至 80 年代期间，基于计算机的频谱分析滋生出重要的洞察乐器和人声微结构的能力(Moorer，Grey，and Snell 1977，Moorer，Grey，and

Strawn 1978, Piszczalski 1979a, b; Dolson 1983, Stautner 1983, Strawn 1985b)。20 世纪 90 年代频谱分析从深奥的专业技术演化为音乐家们录音间里的常用工具——分析、记谱和声音转换。下一部分讨论各种形式的频谱分析,包括短时傅里叶变换和相位声码器。然后介绍扩展傅里叶分析,包括 Q 值滤波器组和小波变换。虽然傅里叶方法主要占据着频谱分析领域,但近年来其他方法也逐渐成长。所以我们在本章后面介绍这些"非傅里叶技术"。(有关频谱分析技术方面的概观请参考轶事体的文献,Robinson 1982。)

短时傅里叶频谱(The Short-time Fourier Spectrum)

傅里叶变换(Fourier transform)是一个将任何续时(模拟)波形制定为相应的,各自具有专门振幅和相位的正弦波单元的、无限傅里叶级数和的数学计算过程。换言之,FT 将其输入的内容转换为相应的频谱表示法。为将傅里叶分析移植到采样、有限时值、时变信号的可操作世界中来,研究者们将 FT 铸型为短时傅里叶变换(short-time Fourier transform),缩写为 STFT(Schroeder and Atal 1962; Flanagan 1972; Allen and Rabiner 1977, Schafer and Rabiner 1973b)。

窗口化输入信号(Windowing the Input Signal)

作为频谱分析的前期准备,STFT 将时间窗(time windows)强加给输入信号(图 13.7)。就是通过取样窗(window)函数将输入信号肢离成有时间边界的"短时"(摘要)片断。取样窗(window)不是别的什么,其是为频谱分析设计的一种特殊包络。取样窗(window)的时值一般在 1 毫秒至 1 秒钟之间,且片断间不时迭盖。通过分别对窗取(windowed)后的频谱片断进行分析,可以获得组成时变频谱的一系列测量。

窗口化(windowing)过程是"短时傅里叶变换"中"短时"的来源。不幸的是,窗口化(windowing)具有歪曲频谱测量的副作用。因为频谱分析仪没有纯粹地对输入信号进行测量,而结果是输入信号与取样窗(window)之积。频谱的计算结果是输入信号和窗取信号(window signals)频谱的卷积。我们将在以后讨论其含意〔第 10 章有关于卷积的解释。附录对窗口化(windowing)有更详尽的讨论〕。

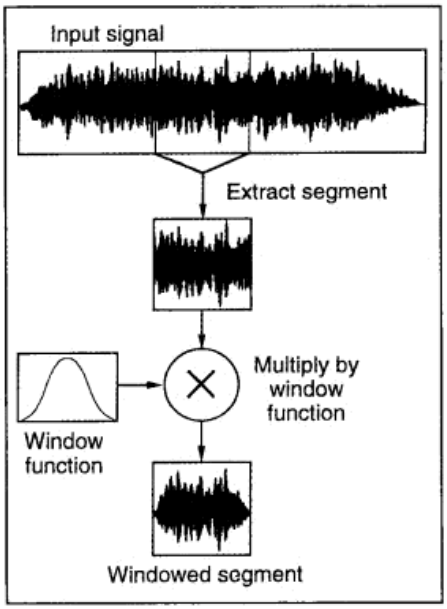


图 13.7　窗口化输入信号。
Input signal=输入信号
Extract segment=提取片断
Window function=窗函数
Multiply by window function=乘以窗函数
Windowed segment=窗取片断

STFT 的操作(Operation of the STFT)

窗口化(windowing)之后，STFT 对各窗取片断施行离散傅里叶变换(discrete Fourier transform，DFT)。这里有关 DFT 我们需要介绍的是，其是一种能处理离散时间或采样信号的傅里叶变化算法。它的输出是一个离散频率频谱，即对一套特定的等间频率能量的度量。(参见附录中介绍 DFT 的内容。)

在历史部分提到过的快速傅里叶变换或 FFT，仅仅是 DFT 的一个高效应用。由此，大多数实践应用中，STFT 对每个窗取片断(windowed segment)施行 FFT。图 13.8 为 STFT 示意图。

FFT 产生的每块数据被称为帧(frame)，类比自电影中的连续帧。每帧包含两方面内容：(1)描绘每个被分析频率分量的幅度谱(magnitude spectrum)；(2)表示每个频率分量最初相位的相位谱(phase spectrum)。图 13.1—13.4 中所有的图都是幅度谱。

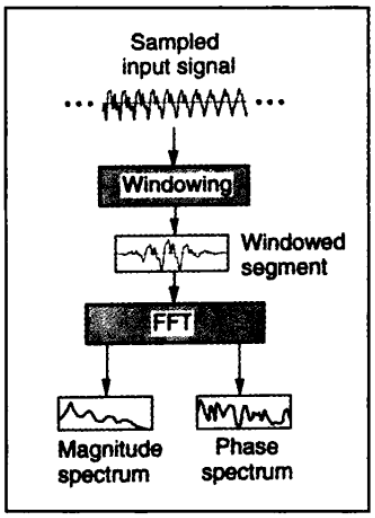


图 13.8 短时傅里叶变换(STFT)概观。
Sampled input signal=采样的输入信号
Windowing=窗口化
Windowed segment=窗取片断
Magnitude spectrum=幅度谱
Phase spectrum=相位谱

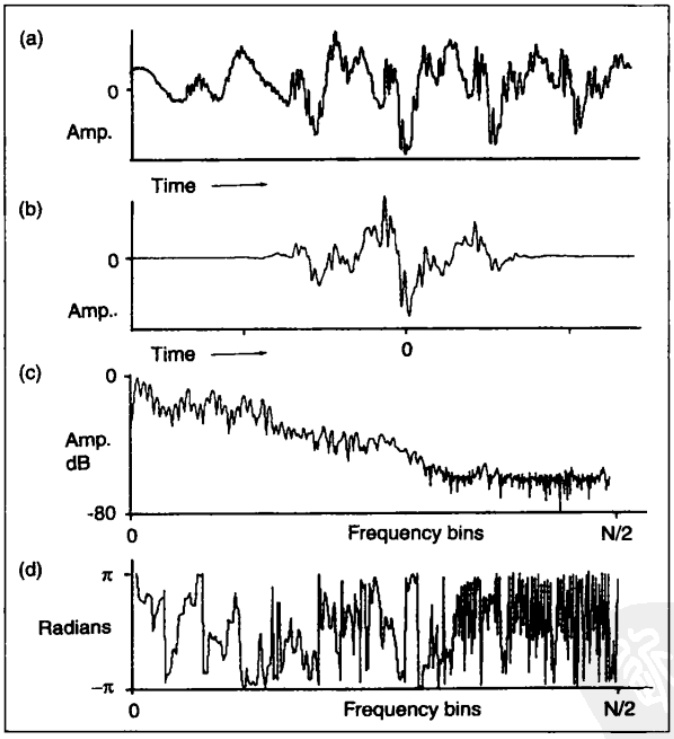


图 13.9 STFT 信号。(a)输入波形;(b)窗取片断;(c)范围从 0 至 −80dB 的幅度谱;(d)范围从 −π 到 π 的相位谱。(出自 Serra 1989。)
Amp.=振幅 Radians=弧度 Frequency bins=频线

我们可以分别将这两种频谱视为沿横坐标每个频率分量有一条垂直线的直方图。在幅度谱(magnitude spectrum)中垂直线表示振幅,在相位谱中表示开始相位(−π 到 π 之间)(图 13.9)。幅度谱相对容易读懂。当相位谱被标准

化在$-\pi$到π之间时被称为闭环相位(wrapped phase)表示法。对很多信号而言,其开环相位谱用肉眼看起来像随机函数。一个开环相位(unwrapped phase)图也许在视觉上更有意义。附录解释了闭环及开环相位的概念。

总之,对输入样本数据流应用 STFT 的结果是构成时变频谱的一系列帧。

根据分析数据进行迭加再合成(Overlap-add Resynthesis from Analysis Data)

为再合成原始的时域信号,通过对每帧进行离散傅里叶逆变换(inverse discrete Fourier transform,IDFT)来重建源自频谱分量的窗取波形片断。IDFT 取幅度谱和相位谱中的每个元素进行计算,产生一个与分析窗包络相同的时域信号。

然后,通过把这些再合成的取样窗迭盖和相加,典型地在它们的-3dB 点(-3dB points)(关于该术语请参见第 5 章),可以获得一个与原来很近似的信号。图 13.10 是迭加处理的示意图。(附录对 IDFT 及迭加再合成都有更详尽的介绍。)

我们使用"紧密逼近"这一概念,作为将 STFT 在实际中的实施与其数学理论相比较的一种方式。理论上,由 STFT 再合成是恒等运算,通过样本再现输入样本(Portnoff 1976)。如果实践中也是恒等运算,我们可以通过 STFT 无限次复制信号也不会产生损失。但是,即使执行得很好的 STFT 也会损失一小部分信息。也许执行 STFT 后听不到这种损失。

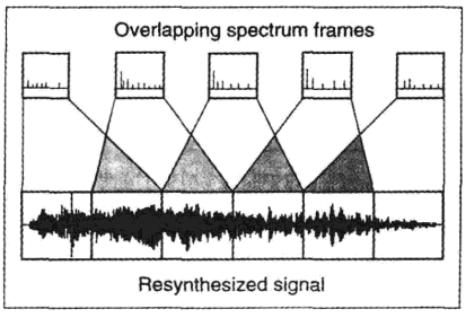


图 13.10　迭加再合成。灰色区域表示迭盖的频谱帧。注意:为清晰起见,我们只展示了五帧。声音分析实践中每秒钟大于 100 帧是很典型的应用。
Overlapping spectrum frames=迭盖频谱帧　　Resynthesized signal=被再合成的信号

迭加再合成的局限(Limits of Overlap-add Resynthesis)

站在音乐转换的角度看,通过简单迭加(OA)进行再合成的应用是有限的。原因是迭加处理是针对取样窗之和完全为常数的情况设计的。正如艾伦(Al-

len)和瑞毕耐(Rabiner,1977)所指出的,任何 OA 最后的阶段都会扰乱理想求和准则的加法或乘法转换,其带来的副作用很可能会被听到。例如通过拉伸取样窗间的距离来扩展时间,可能会引至梳状滤除或反响效果,取决于分析所使用的频率通道(frequency channels)或线(bins)的数量。以语音或歌唱为声源,很多转换的结果只能产生用途有限的,如机器人般的,或高频共振类声音。

减少这些不期望的后生现象的方法之一是在分析阶段保证大量的前后取样窗间的迭盖,这将在下一部分中阐述。"改良版迭加(improved overlap-add)"再合成是处理这类问题的另一种途径(George and Smith 1992;亦请参考本章下面的内容)。

为什么使用迭盖取样窗?(Why Overlapping Windows?)

在 STFT 中迭盖分析窗背后的动机可能会产生歧义。毕竟理论上我们分析任何长度的片断都可以从分析数据中准确地再合成该片断。显然,我们可以使用 30 分钟长的取样窗来分析完整的斯特拉文斯基的《春之祭》(*Le sacre du printemps*),然后从这个分析中重建整个作品。在这种情况下,为什么还麻烦地将分析打散成小的迭盖片断?

原因是多方面的。对时长 30 分钟,采样速率为 44.1kHz 的单声道声音的分析结果将是一个超过 7 900 万点的频谱。对这个庞大频谱的视觉观察可以最终告诉我们 30 分钟时值内出现的所有频率,但不能告诉我们它们精确的出现时间;该过程信息深埋在幅度谱和相位谱的数学组合中,视觉无法察觉。对视觉观察频谱有益是采用窗取法的第一个原因。通过把分析限制在短片断(通常小于十分之一秒)上,每个分析描绘很少的点,我们可以更精确地知道何时出现了哪些频率。

采用短时包络(short-time envelopes)的第二个原因是节省内存。以一口吞下 30 分钟长这么一大块声音的分析为例,假设用 16 比特样本,当计算机进行 FFT 计算的时候,随机读写存储器(RAM)至少需要 7 900 万个 16 比特字长的空间来容纳输入信号。通过将输入信号分成微小的片断,每次对每个小片断的 FFT 运算都变得容易起来。

采用短时取样窗的第三个原因是为了更快地获得结果。以《春之祭》为例,不得不等 30 分钟,只为了读取输入信号,加之还有对至少 79 00 万个点的输入信号进行 FFT 所需要的时间。窗取输入信号可以让我们在读取输入信号的几毫秒后就获得最初结果,为实时频谱分析的应用提供了出路。

这三个原因解释了窗取的目的,但为什么将取样窗迭盖?正如我们先前解释过,平滑的铃状取样窗(bell-shaped windows)最小化窗取中的失真。那么当然,类铃声包络状的取样窗必需有些迭盖以便无缝记录信号。但通常完美求和

准则甚至要求更大量的迭盖以达到令人满意的结果。为什么呢？增大迭盖系数等同于频谱过采样(oversampling the spectrum)，其保证对抗诸如时间扩展和交叉合成(cross-synthesis)转换中出现的混淆后生现象。当分析目的是对输入信号施行转换时，建议迭盖系数为 8 或更大。

后面我们将讨论取样窗的选择和其长度设置等基本准则。附录有更详尽的关于窗口化方面的内容。下面我们将介绍一个迭加再合成模型的替代(alternative)方法。

振荡器组再合成(Oscillator Bank Resynthesis)

正弦加法再合成(Sinusoidal additive resynthesis，SAR)(或振荡器组再合成)与迭加法不同，SAR 采用了一组由横越帧边界的振幅和频率包络驱动的振荡器(图 13.11)，这个方法要胜于每帧将正弦波相加(例如：OA 再合成模型)。这意味着在装入这些包络之前必须预先转换分析数据。幸运的是，将分析数据(幅度和相位)转换为合成数据(振幅和频率)需要很少的计算时间。

SAR 模型的优点是在音乐性的转换中，包络远比帧式频谱数据流健全得多。在众多的限制中，可以拉伸、缩短、重新置率或位移包络而不必担心再合成处理中的后生现象，并可无视 OA 模型的理想求和准则。SAR 的不利之处在于计算效率不如 OA 法。

相位跟踪声码器可以被看作 SAR 法，因其同样为正弦波加法合成构建频率包络。我们将在之后的相位声码器部分介绍该方法的详细内容。

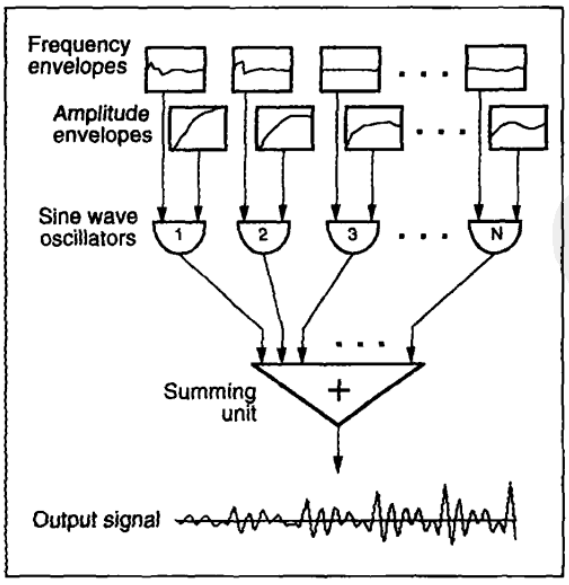


图 13.11　振荡器组再合成。分析数据被转换成一系列连续的振幅和频率包络。根据声音的复杂度增减再合成所需的振荡器数量。
Frequency evnelopes=频率包络
Amplitudes evnelopes=振幅包络
Sine wave oscillators=正弦波振荡器
Summing unit=加法单元
Output signal=输出信号

分析速率(Analysis Frequencies)

可以将 STFT 视为加载于窗取信号上的等频率间隔的滤波器组。频率按整数倍(泛音)相间为

$$\frac{sampling\ frequency}{N}$$

其中 N 为分析片断的大小(从后面的内容我们将看到 N 通常大于分析样本数;这里我们假设它们等长)。因此如果采样频率为 50kHz,取样窗长度为 1 000个样本,分析速率将从 0Hz 开始相间 50 000/1 000=50Hz。位于 0Hz 的分析器测量信号的直流(direct current)或直流偏置(DC offset),一个可以将整个信号移动高于或低于零振幅中央点的常数。

音频信号的带宽被限制为采样速率的一半(这里是 25kHz),所以我们只关心一半的分析线(bin)。(正如前文所述,谱线、频线是信号处理的特有用语,即频率通道)。STFT 的有效频率分辨率是等间分布于音频带宽中的 $N/2$ 线,从 0Hz 开始,结束于尼奎斯特频点。在我们的例子中,可用音频频线数为 500,相互间被 50Hz 隔开。

时间/频率不定性(Time/Frequency Uncertainty)

所有窗取频谱分析都受制于一个基本的时间和频率分辨率间的不定性原理(uncertainty principle),这一点首先被 20 世纪初的量子物理学家认识到,如维尔纳·海森堡(Werner Heisenberg)(Robinson 1982)。该原理意思是如果我们想要得到时域的高分辨率(我们想精确地知道一个事件何时发生),则要牺牲掉频率分辨率。换言之,我们可以得到某事件发生的精确时间,却无法告知其包含的确切频率。反之,如果我们想要频率维度的高分辨率(我们想得知某分量的精确频率),则需要牺牲掉时间分辨率,更确切地说,我们只能通过一段长时值间隔查明频率内容。领会这种关系以便解析傅里叶分析结果是很重要的。

周期意味着无限(Periodicity Implies Infinitude)

傅里叶分析始于这样一个抽象前提,如果一个信号只包含一个频率,那么该信号一定是时值无限长的正弦曲线。频率的纯正——绝对周期——意味着无限。

一旦某个因素限制了这个正弦波，傅里叶分析能够说明它的唯一方法就是，将信号看作是很多无限长度的正弦波之和，它们恰好以相互抵消这样的方式，导致结果表现为一个有限时值的正弦波！虽然对频率的这一特性记述使数学整齐干净，但与我们最基础的声音经验不符。正如加布尔(Gabor1946)指出的，如果频率的概念只用来关注无限长信号的话，那么改变频率这一概念将不可能！

我们依然可以通过思想实验法理解抽象傅里叶表达的实质，通过声音编辑器，想象放大到数字系统时域的极限。我们可以看到最短"瞬间"上的单个采样点(图 13.12a 中标有 O 的矩形)。我们精确地知道该样本发生在何时，那么我们具有高的时间分辨率，但我们无法得知其是哪个波形的一部分，它可以是系统尼奎斯特范围内任一频率波形的一部分。当我们缩小视图(图 13.12b)，我们就有更多供分析的样本，且由此我们更确定它们可能代表的频率。但由于傅里叶分析一次计算整个分析片断的频谱，长片断的频谱图遗留下这样的不确定性：某特定频率到底是何时发生的。再次重申，频率精度是以牺牲时间精度为代价的。

滤波器设计提供更多线索。回顾第 10 章，延时阶段(delay stage)的数量影响滤波器的锐利度。为分离出一个非常窄的频带，比如某单一频率分量，我们需要特别锐化滤波器的响应。这意味着必须向前检视过往已久的信号以便提取出纯频率。另一种说法是，这样的滤波器具有一个长的脉冲响应特性曲线。(参见第 10 章有关脉冲响应特性曲线的解释。)

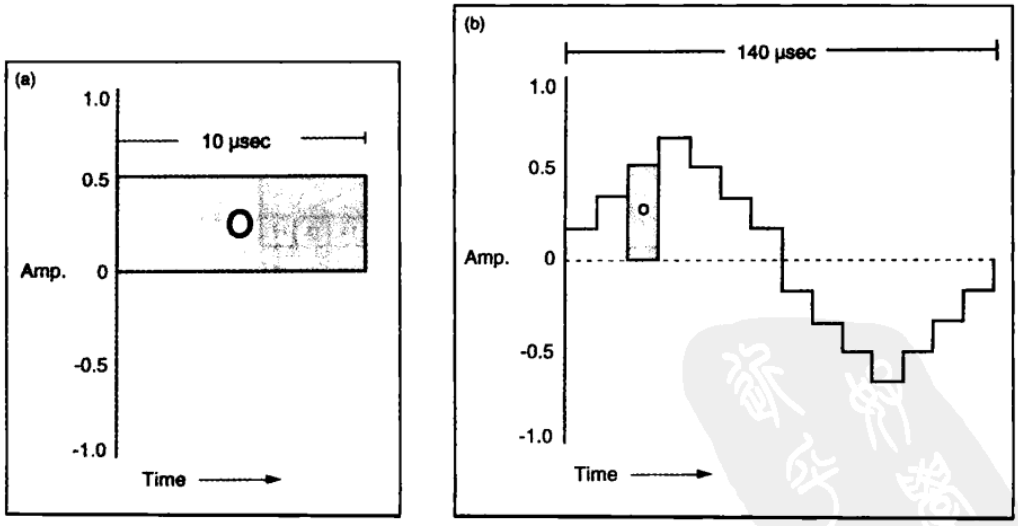


图 13.12　短时间量程时的频率不确定性。(a)方框 O 表示放大至高时间分辨率系统中的样本周期(10 微秒的采样周期意味着采样速率为 100kHz)。在这个时间分辨率下没能显示任何频率信息，我们丧失了得知其为哪个较大波形之一部分的判断力。由此根据一个或几个样本进行的频率估测被限制在粗略的猜测范围内；(b)放大到 140 微秒量程给出有关总体波形和当前频率周期的好得多的图像。
Amp.=振幅　　Time=时间

时间/频率抵换(Time/Frequency Tradeoffs)

FFT 将听阈频率分成 $N/2$ 个频线(frequency bins),其中 N 为分析窗(analysis window)的样本宽度。由此频线数量和分析窗宽度之间存在一个抵换关系(图 13.13)。例如,如果 N 等于 512 个样本,那么分析频率被限制在 256 以内。假设采样速率为 44.1kHz,我们得到 256 个分析线均匀分布于 0Hz 至尼奎斯特频点 22.05kHz 之间。提高采样速率只加宽可测量带宽,并不增加分析的频率分辨率。

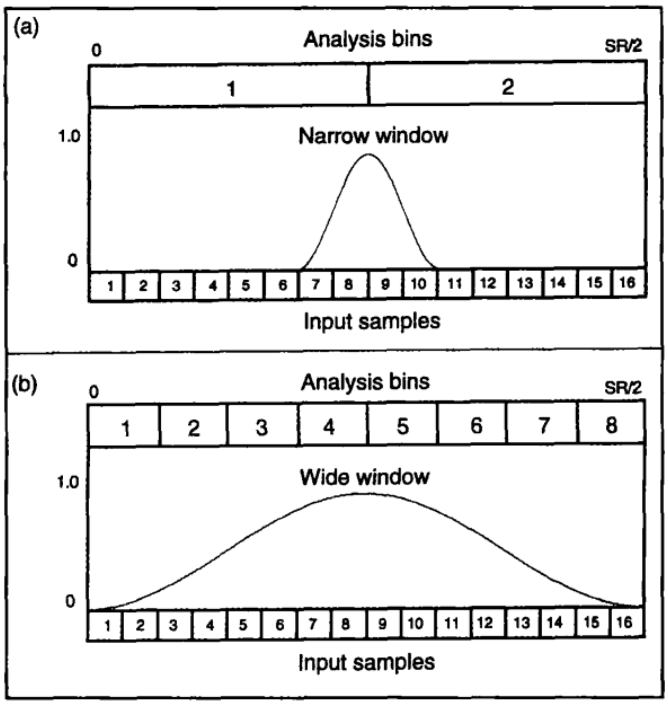


图 13.13 窗幅与频率分析线数量的关系。(a)四个样本的窄分析窗仅能解决两个频率;(b)宽一些的 16 个样本的分析窗将频谱分成 8 个谱线。
Analysis bins=分析谱线 Narrow window=窄窗 Input samples=输入样本 Wide window=宽窗

表 13.1 所示为时间与频率分辨率之间的关系。如果我们想要高时间精度(比如 1 毫秒或大约 44.1kHz 采样速率时 44 个样本),我们必须满足于 44/2 或 22 个频线。0 至 22.05kHz 的音频带宽除以 22 频线,我们获得 22 050/22 或 1kHz 的频率分辨率。亦即如果我们想以 1 毫秒为单位得知事件发生的具体时间,那么我们的频率分辨率被限制在大约 1kHz 宽的频带内。通过牺牲更多的时间分辨率,将分析间隔增宽至 30 毫秒,就可以按照 33Hz 的带宽描绘频率。对于高频率分辨率(1Hz),必须将时间间隔拉伸至 1 秒钟(44 100 个样本)!

表 13.1　窗取频谱分析中的时间与频率分辨率

时间长度 取样窗(单位:毫秒)	频率分辨率 (分析带宽)(单位:Hz)
1	1 000
2	500
3	330
10	100
20	50
30	33
100	10
200	5
300	3
1 000	1
2 000	0.5
3 000	0.3

由于窗取 STFT 分析的限制,研究者们在检视时域和频域分析的混成体,多重分辨率分析(multiresolution analysis),或非傅里叶方法试图解决两个维度都高分辨率的问题,稍后我们将讨论这些方法。

分析线间的频率(Frequencies in between Analysis Bins)

STFT 只能知道均间分布在音频带宽内的频率离散集。频率间隔取决于分析窗的长度。该长度对分析的"基频周期"有效。这样的模型可以很好地工作于谐波或准谐波声音,其中泛音与分析线接近对齐。但对于那些落在 STFT 分析线之间的频率会出现什么情况呢? 诸如非谐波声音锣或军鼓这样的噪声。

让我们称分析频率为 f。当 f 与某分析通道(analysis channel)的中央一致,其所有能量都集中于该通道中,由此被精确地测量。当 f 接近但非与中央精确一致时,能量分散到所有其他分析通道中,但保留集中接近 f。图 13.14 所示为某频率从 2Hz 延伸至 3Hz 的三张截图,可被延及其他频率范围。线间分量渗漏至所有频线的现象是 STFT 产生的频谱评估不可靠性的著名根源。当线间分量超过一个时,频率和振幅描绘中都有可能出现拍音效果(beating effects)(周期性的抵消和增长)。其结果表现为:分析显示出了输入信号中非物理存在的频率分量脉动能量。

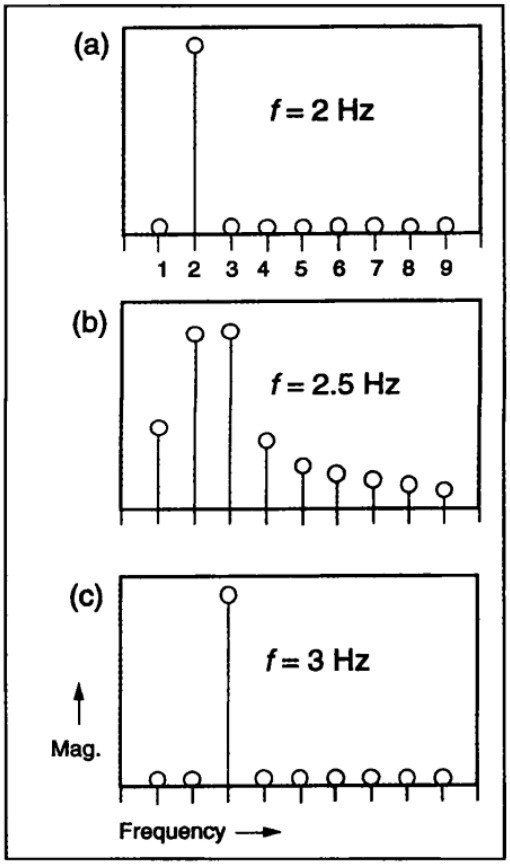


图 13.14 从 2Hz 改变至 3Hz 声音的三张 STFT“截图”。这里 STFT 分析线相间 1Hz。当输入频率为 2.5Hz 时，其落在分析器均间频线的中间，且能量分布横跨全部频谱。（出自 Hutchins 1984。）
Mag.=幅度量 Frequency=频率

杂波干扰的显著性(Significance of Clutter)

如果信号直接再合成自分析数据，额外的频率分量和拍音效果不会构成问题；它们是再合成中得到解决的良性 STFT 分析后生现象。拍音效果仅是 STFT 在频率维度表示时变频谱的方式。再合成中，某些分量建设性地相加，某些破坏性地（相互抵消）相加，因此再合成的结果是原来的近似值。（再次，理论上其为恒等，但实践应用中悄悄混进了小错误。）

拍音和其他异常对信号的直接再合成无害，但它们使得对频谱的视觉观察变得难以理解，或使之变形。由于这个原因，分析的后生现象被称为杂波干扰（clutter）。多尔森（Dolson，1983）和斯特劳（Strawn，1985a）鉴定了乐器音色分析中杂波干扰的意义。戈赞（Gerzon，1991）提出“超级解析”频谱分析器理论，

用以在时间和频率两方面都提高分辨率，代价是提高杂波干扰，其中，戈赞表明，存在一些可感知的显著性。

另类再合成技术(Alternative Resynthesis Techniques)

这里简要总结一下标准再合成技术之外的两种另类方法的优点：第一种是提供改良分辨率和更强变换的适应性方法；第二种方法大大加快了再合成速度。

基于合成的分析/迭加(Analysis-by-synthesis/overlap-add，ABS/OLA)在迭加再合成中通过并用一个错误分析程序改善了 STFT(George and Smith 1992)。该程序将原始信号与再合成进行比较。当错误超出了规定阈限，程序在分析帧中调整振幅、频率和相位以便更接近原型。这种适应性操作也许会重复出现直到信号接近被恰好重建。ABS/OLA 法的结果是可以处理触发暂态，非谐波频谱以及揉音这样的效果，比单一的迭加方法更精确。它也允许更强的音乐变换。稍后我们可以看到，一个叫做追踪相位声码器的方法具有类似的好处。

“FFT^{-1}”法是为实时操作而优化自迭加和振荡器组再合成法的特殊混成法。之所以如此命名该方法是因为其实现自倒 FFT，可被缩写为 FFT^{-1}。其开始于原先计算出的振荡器组再合成数据。然后通过高效数据缩减和优化步骤算法，将这些数据转换为迭加模型，从而大大提高再合成速度，细节请参阅罗德特(Rodet)和德哈雷(Depalle，1992)以及法国专利 900935。

语图表示法(The Sonogram Representation)

语图(sonogram)、声谱仪(sonograph)或声谱图(spectrogram)是语音研究领域著名的频谱显示技术，几十年来用于语音分析。语图展示几秒钟声音频谱的总体面貌。这使得观测者可以看到诸如音节或音素开始、共振峰和主要转变等总体特征。受过训练的观测者可以阅读语音语图。参阅科根(Cogan 1984)作为音乐分析中使用语图的实例。语图表示法同样也被当作频谱编辑的接口(Eckel 1990；见第 16 章)。

原始的语图是巴克豪斯(Backhaus，1932)系统，在有关频谱分析背景的早些篇幅中描述过，亦请参阅凯尼格(Koenig et al.，1946)。凯声谱仪(Kay Sonograph)是 20 世纪 50 年代标准的语图制作设备。它包含许多模拟的窄带宽滤波器和一个在卷动纸上打印深色条带的记录系统。条带按照每个滤波器输出能量的大小成比例增长浓密度。现在，语图通常是 STFT 实现的。

图 13.4 所示为语图，从时间和“频率＋振幅”这两个维度表示一个声音信

号。纵坐标所示为频率(高频位于语图上端),灰色梯度表示振幅,其中深色代表更高强度。

语图参数(Sonogram Parameters)

现代语图参数与那些 STFT 的一样,除几个显示参数之外。调整这些参数致使输出图像有很大不同。

1. 振幅范围和所用的线性或对数的比例关系。

2. 频率范围和所用的线性或对数的比例关系。

3. 分析窗前移时间,亦称跃幅(hop size)(以样本为单位)或取样窗迭盖系数(window overlap factor)。这决定输出图形相继栏之间的时间距离。(我们将在相位声码器部分讨论该参数的细节。)

4. 被分析样本数量和 FFT 分析窗大小,时间和频率分辨率取决于这些参数。

5. 显示频道的数量,决定输出图像的行数且与频率维度的范围和显示比例有关;这不能超过窗幅规定的分辨率。

6. 取样窗类型——参阅相位声码器中有关部分的讨论及附录。

参数 4 包括两个参数,FFT 窗幅通常大于实际分析声音样本,差异被取值为零的样本所填补(见相位声码器分析参数部分)。这些参数对显示结果有着最戏剧性的影响。短取样窗结果是纵向显示图,表明精确的事件发生时间,但模糊了频率可读性(图 13.15a)。中等长度取样窗对时间和频率特征都有很好的解析,标示出共振频率之所在(图 13.15b)。长取样窗产生水平方向显示图,单个频带变得清晰可见,但它们的时间位置模糊在横坐标中(图 13.15c)。

语音语图必须进行调整以便适应更严格的音乐要求。音乐的语图趋向于比语音语图长,包括片断或整个作品。音乐的动态范围远大于语音。同时正如兰登(Lundén)和温格瓦里(Ungvary,1991)所指出的,语音语图趋向于声谱的精确物理表达,而音乐家们对和我们听觉相一致的感知层面更感兴趣。随后将介绍的耳蜗图(cochleagram display)可能是更精确的感知图。有关站在精确度立场上对传统语图的批判性分析,请见 Loughlin,Atlas,Pitton(1992)。

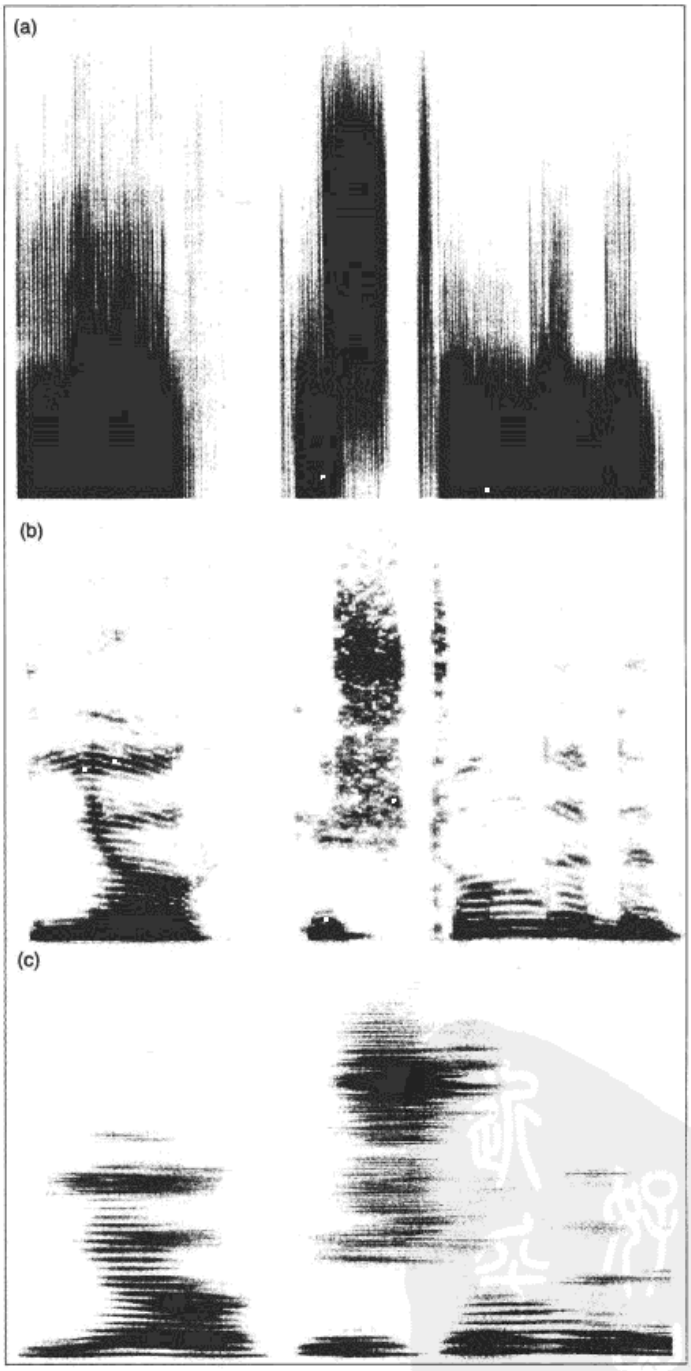


图 13.15　语图分析和显示中的时间频率权衡。所有图都以 44.1kHz 为采样速率显示语音。(a)分析窗长度为 32 个样本，时间分辨率为 0.725 毫秒，频率分辨率为 1378Hz；(b)分析窗长度为 1024 个样本，时间分辨率为 23.22 毫秒，频率分辨率为 43.07Hz；(c)分析窗长度为 8192 个样本，时间分辨率为 185.8 毫秒，频率分辨率为 5.383Hz。〔语图由伊克尔(Gerhard Eckel)通过他的频谱绘制程序 SpecDraw 提供。〕

相位声码器(The Phase Vocoder)

相位声码器声作为声音分析工具表现得越来越时兴,被打包在几个广为分布的软件包中。(Gordon and Strawn 1985 and Moore 1990 中包含实用相位声码器带注解的代码。)可以将 PV 视为将窗取输入信号穿过均间分布于音频带宽的并联带通滤波器。这些滤波器在每个频带上测量正弦信号的振幅和相位。通过后继操作(附录中有讨论),这些值可以被转换成两个包络:一个为正弦的振幅;另一个为正弦的频率。这与我们先前讨论过的振荡器组再合成情况相同。各式各样的 PV 执行提供修改这些包络的工具,允许对所分析声音进行音乐性的转换。

理论上,通过相位声码器进行分析和再合成是样本及样本的克隆(Portnoff 1976)。实践中,通常会有微小的信息丢失,也许在单一分析/再合成过程中听不出来。在任何情况下,音乐家对 PV 的使用无可避免地包括再合成前对分析数据的修改。作曲家在输出结果中搜寻的不是输入信号的克隆体,而是留有声源同一性感觉的音乐化变体。那就是说,如果输入信号是说话声音,即使转换后,通常希望声音听起来像说话。也可以通过 PV 进行根本变形,破坏输入信号的同一性,但更高效的失真算法很容易得到,例如第 6 章讨论的调制。

有关第一个声码器请参阅第 5 章。更多关于 PV,包括实用执行描述,参见 Portnoff 1976, 1978, 1980, Holtzman 1978, Moorer 1978, Moore 1990, Dolson 1983, 1986, Gordon and Strawn 1985, Strawn 1985b; Strawn 1987, Serral989, Depalle and Poirot 1991, Erbe 1992; Walker and Fitz 1992; Beauchamp 1993。

相位声码器参数(Phase Vocoder Parameters)

特定 PV 分析的质量取决于使用者选择的参数设置。这些参数必须根据所分析声音的自然属性和所期望的结果类型进行调整。PV 的主要参数如下:

1. 帧长(Frame size)——每一次分析的输入样本数。

2. 取样窗类型(Window type)——从标准类型中选择某一形状(稍后讨论)。

3. FFT 大小(FFT size)——实际馈送给 FFT 算法的样本数;通常接近帧长的平方,其中 FFT 大小由点(points)表示,如"1024 点 FFT"(等于"1024 样本 FFT")。

4. 跃幅(Hop size)或迭盖系数——某帧前进至下一帧的时间。

现在我们依次讨论这些参数，接下来的部分将给出这些参数设置的粗浅常识。

帧长(Frame Size)

出于两个原因帧长(以样本为单位)很重要。首先是帧长决定“时间/频率”分辨率权衡的一个方面。更大的帧长，更多的频线，但是更低的时间分辨率，反之亦然。如果我们试图按高频率分辨率分析低音区声音，大帧长是不可避免的。因为 FFT 在帧内计算平均频谱内容，当频谱被绘制或转换时，帧内任何频谱变化的发生时间将丢失(如果声音被简单地再合成，该临时信息将被重建)。对高频声音而言，短帧长更适宜，同时时间分辨率也更精确。

第二个重要原因是长 FFTs 比短 FFTs 计算起来要慢得多。单凭经验的方法，计算时间呈如下比例关系 $N\times \log_2(N)$，其中 N 为输入信号长度(Rabiner and Gold 1975)，计算一个 32 768 点 FFT 要比例如 64 点 FFT 花费上千倍的时间。在实时系统中，长 FFT 或许潜在着过于繁重的负担。

取样窗类型(Window Type)

大多数 PV 提供使用标准取样窗家族类型之一的选项，包括加重平衡，汉宁修匀〔或汉(Hann)；见 Marple 1087〕，高斯截短曲线，布莱克曼-哈里森和凯塞尔(Harris 1978，Nuttall 1981，亦参见附录)。所有类型都是铃类形状且在一般音乐分析/再合成中都表现得相当好。对于分析来说精确度很重要(例如给乐器音色建立系统目录)对分析窗的选择也许更苛刻。这是因为窗口化导致失真，且每种取样窗以略微不同的方式使分析图“搀杂”。更多有关取样窗请参阅附录。

FFT 大小和零填充(FFT Size and Zero-padding)

FFT 大小的选择取决于计划施加给输入信号的转换类型。安全获得交叉合成则是接近平方亦即帧长的两倍。例如 128 个样本帧长将令使 FFT 大小为 256。FFT 中另外 128 个样本被设为零——一个称作零填充的操作(见附录)。

跃幅(Hop Size)

跃幅是每次进行新频谱测量时，分析器沿输入波形前跳的样本数(图 13.16)。更短的跃幅则相继取样窗间更多的迭盖。由此有些 PV 将这个参数指定为迭盖系数，用来描述有多少相互覆盖的取样窗。不管如何规定它，跃幅通常是帧长的一个区间。为保证精确再合成，一定数量的迭盖(例如 8 次)是必须的。当分析数据将用来转换时，甚至更多迭盖才能改善精确度，但计算负担也成正比增加。

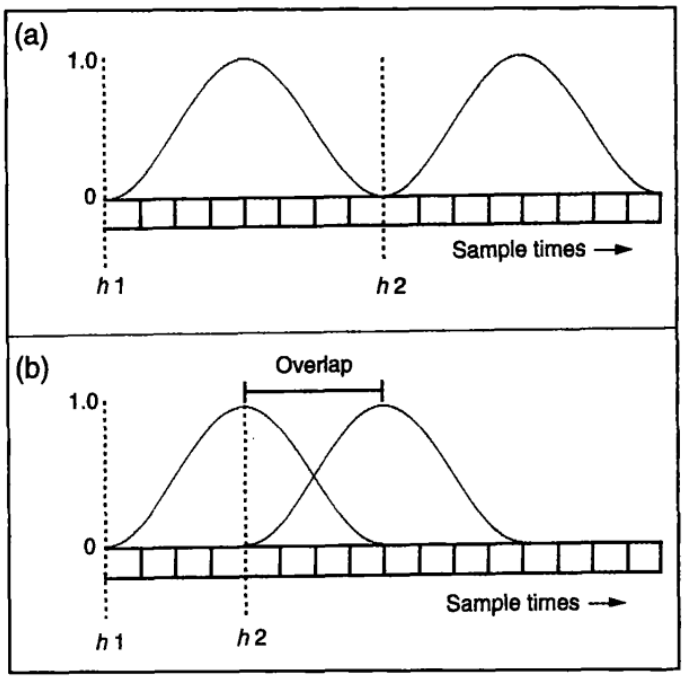


图 13.16 长度为 8 个样本的分析窗的不同跃幅。h1 和 h2 是每个取样窗的开始时间。(a)跃幅=取样窗大小时无迭盖取样窗；(b)当跃幅小于窗幅时的迭盖取样窗。这里跃幅为 4 个样本。
Sample times=采样时间 Overlap=迭盖

典型参数值(Typical Parameter Values)

不存在对所有声音都理想的 PV 参数设置。但当参数设置在一定范围内时，可以合理保真地分析和再合成多种传统乐器声音。这里是一些可以作为更“协调”的分析开端的 PV 参数经验值：

1. 帧长——长到足够可以捕获有益最低频的四个周期(Depalle and Poirot

1991)。这对声音拉伸来说尤其重要;过小的帧长意味着个别音高脉冲被分离,会改变音高,尽管共振峰被保留。

2. 取样窗类型——三角形除外的任何一种标准类型。

3. FFT 大小——在样本中为帧长的一倍。

4. 跃幅——如果要在时间上对分析数据进行失真处理,推荐跃幅为帧长的八分之一(8 次迭盖)。一般来说,最小技术准则是所有取样窗相加为一个常数,亦即所有数据的权重相等。这典型地暗指被选特定取样窗类型在－3 分贝处迭盖,由此可以获得跃幅。

闭窗(Window Closing)

"一次不够"(S. J. Marple 1987)

任何特定的窗幅设置结果都可表现为频谱分析趋向被窗幅定义的周期的泛音。落在与给出窗幅相联系的频线之外的频率分量将被错误评估。由此,有些频谱分析步骤按照不同窗幅设置让同一信号重复地穿过分析器,由高时间分辨率、低频率分辨率开始,逐渐过渡到低时间分辨率、高频率分辨率的过程叫做闭窗(译注:泛指时间、频率从某分辨率过渡到其他分辨率的过程)(Marple 1987)。

有些 STFT 分析器试图估量信号音高以便决定最佳窗幅,正如前文提及,如果被分析声音具有基本的泛音结构,音高同步分析就可以很好地发挥作用。

追踪相位声码器(Tracking Phase Vocoder)

当前很多 PV 都被称作追踪相位声码器(TPVs),因为它们随时间跟随或追踪频谱中的最显著峰值(Dolson 1983, McAulay and Quatieri 1986, Quatieri and McAulay 1986, Serra 1989, Maher and Beauchamp 1990, Walker and Fitz 1992)。与再合成频率受制于分析窗泛音的普通相位声码器不同,TPV 跟随改变频率。峰值跟踪的结果是一组驱动再合成阶段正弦振荡器组的振幅和频率包络。

跟踪操作只跟随最显著的频率分量。对这些分量而言,与等间滤波器组(传统 STFT 的执行)相比分析结果更精确。另一个好处是,跟踪操作为这些分量产生频率和振幅包络,在信号转换情况下比迭盖帧更坚实有效。不利条件是分析质量可能要比常规 STFT 更依赖于恰当的参数设置。

TPV 的操作(Operation of the TPV)

TPV 执行以下步骤：

1. 通过使用者指定的帧长、取样窗类型、FFT 大小和跃幅计算 STFT；

2. 以分贝为单位得到平方量频谱；

3. 找到频谱中峰值的线数；

4. 计算每个频率峰值的量和相位；

5. 通过将前一帧与当前帧的峰值进行匹配，将每个峰值指派给频率轨迹(frequency track)(参见后面有关峰值跟踪的内容)；

6. 对分析参数施加任意期望修正；

7. 如果需要加法再合成，为每个频率轨迹生成一个正弦波，将所有正弦波分量相加以产生输出信号；通过一帧帧的插值，计算出每个正弦分量的瞬时振幅、相位和频率(或使用先前介绍过的另类再合成方法)。

峰值跟踪(Peak Tracking)

追踪相位声码器跟随频谱中最显著频率的行迹。与声音分析的其他方面类似，峰值跟踪的确切方法应该随声音的不同而不同。当跟踪算法被调谐为被分析声音类型——语音、谐和频谱、柔和的非谐和频谱、噪声等时最有效。本部分更多地将跟踪过程简要解释为分析参数设置的向导。

峰值跟踪的第一阶段是峰值识别。一个设置最小峰值高度(minimum peak height)的简单控制，使识别过程聚焦在频谱中的最重要陆标上(图 13.17a)。其他的算法试图应用一组在时间上提前的频率向导(frequency guides)(图 13.17b)。向导只是臆测，之后算法将决定哪个向导可被确认为频率轨迹。算法将继续按照当前频率值查找峰值。各种情况如下所示：

- 如果找到相匹配的，延续该向导。
- 如果在一帧内向导不能延续，将被认为是“休眠”。
- 数帧(可以由使用者指定)后向导未苏醒——向导被删除。其也许可能接通向导滞后(guide hysteresis)，继续跟踪那些少许落在指定振幅范围(amplitude range)以下的向导。滞后缓解轻微落在范围外，被峰值跟踪器削除置零，而后渐强进入范围(Walker and Fitz 1992)，重复“转换”的向导带来的可被听到的问题。在滞后的协同下向导被按照其实际值合成，可能低于振幅范围，而不是零振幅。
- 如果向导间有冲突，最近的向导胜出，“失败者”在一个使用者指定的频

带，最大峰值偏差(maximum peak deviation)内查找另一个峰值。

■ 如果存在未被当前向导计算在内的峰值，那么开始一个新向导。

窗口化处理或许会危及跟踪的准确度，特别是对快速变化如触发波形的瞬间。使用翻转时间顺序的锐利触发进行声音处理对跟踪算法很有帮助(Serra 1989)。这使泛音跟踪器有机会在遭遇触发混乱之前锁定它们的稳定频率轨迹，从而降低失真度。再合成前数据可以翻转回正常顺序。

接下来的内容讨论步骤 6，修改 TPV 分析包络。

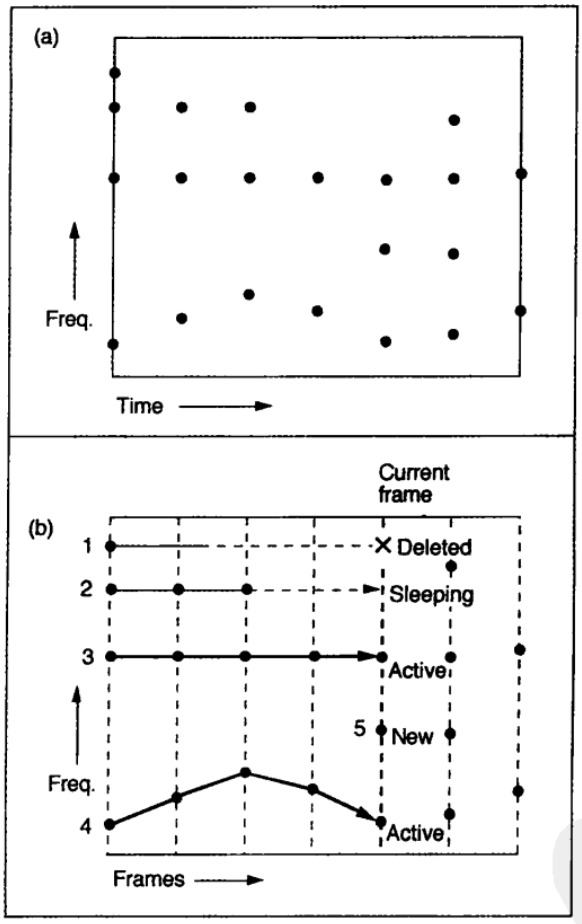


图 13.17　峰值识别和跟踪。(a)分离一组频谱峰值；(b)适配频率向导和峰值。顶端的向导 1 三帧后没苏醒，因此被删除了。向导 2 仍在休眠。向导 3 和向导 4 是活跃者。向导 5 从一个新峰值开始。

Freq.=频率　Time=时间　Curent frame=当前帧　Deleted=被删除的　Active=活跃的　New=新的　Frames=帧

编辑分析包络(Editing Analysis Envelopes)

改变再合成参数产生声音转换。比如通过修改回放跃幅,可以实现时间伸展和缩减效果。然而由于潜在的正弦模型,当对复杂触发或嘈杂声施行时间伸展时,单独的正弦波浮现出来而丢失了嘈杂感。稍后将讨论解决这一问题的塞拉(Serra,1989)频谱建模合成。

为制造高级音乐转换,必须编辑 TPV 产生的分析数据——频率、振幅和相位曲线(Moorer 1978, Dolson 1983, Gordon and Strawn 1985)。通过自动数据削减和图形编辑程序,繁复的处理过程已很自动化了(见第 4 章加法合成数据削减及第 6 章频谱编辑器部分)。第 4 章表 4.1 列出了通过修改 PV 频谱数据可以产生的音乐效果。

通过相位声码器进行交叉合成(Cross-synthesis with the Phase Vocoder)

另一种减少编辑量的声音转换可能性是交叉合成(cross-synthesis)。交叉合成不是一种技术,其有多种类型。最常见的类型是用一个频谱的幅度函数(magnitude functions)控制另外一频谱的幅度函数,即声音 A 中各频率分量的强度按比例决定声音 B 中相应频率分量的强度,该类交叉合成是通过将频谱 A 中每点与频谱 B 中相应点相乘执行的。这种交叉合成的另一类型是卷积滤除(filtering by convolution)(更多有关卷积的内容请参见第 10 章)。音乐上,当被滤除声音具有宽广的带宽,如噪声声源时使用交叉合成最有效。使用双输入相位声码器,交叉合成基本上是自动的(Depalle and Poirot 1991)。另一类交叉合成是使用某声音的幅度函数和另一声音的相位函数产生杂交声音效果(Boyer and Kronland-Martinet 1989)。

对 PV 交叉合成的音乐指导方针与快速卷积交叉合成的方针相同。更多的关于这些指导方针的内容请参见第 10 章。

相位声码器的计算代价(Computational Cost of the Phase Vocoder)

相位声码器是音乐家们可用的,比较消耗计算资源的操作之一,特别当执行跟踪的时候。即使内核采用了高效 FFT 算法,追踪相位声码器也会占用相当大的计算机资源。PV 同时产生很大的分析数据量,某些情况中可能比被分析样本数据大数倍。或许通过技术防护的应用可以减少计算量和节省空间。例如 TPV 生成的包络可以按低采样速率计算,由于控制函数趋向于比音频采

样速率改变慢得多，所以不会影响到声音质量。再合成之前，通过插值它们可以被恢复到原来的采样速率，也可以应用其他数据缩减(data reduction)方法；请参见第 4 章有关数据缩减的讨论。

再合成精确度(Accuracy of Resynthesis)

所有基于傅里叶的再合成精确度都局限于分析过程的分辨率。数字取舍、窗口化、峰值跟踪、包络函数的欠采样(undersampling)带来的小失真，都会给分析的其他方面带来误差。一个良好运转的 PV，当分析参数被熟练的工程师设置得当，且分析数据未被改变，感知上的错误可以忽略不计。

另外，跟踪 PV 在其跟踪构建中解释分析数据流，丢弃所有对跟踪没贡献的信息。这样筛选可能会遗漏声音能量的有意义成分，特别是噪声、暂态能量。实际上它们从原始频谱中减去 TPV 频谱(Serra 1996)，并可以将这个残留或差值认为是分析/再合成误差。一般将再合成、准谐波成分视为信号的“干净”的部分，将误差或噪声分量视为信号“脏”的部分。对很多声音而言(那些有快速暂态值，如敲击吊镲)，误差完全可闻。也就是说，“干净”信号听起来是不自然的“被消毒”或正弦性；“脏”信号，当分开听时，丢失了粗砾感。(见非谐和与噪声声音的分析。)

为效率的缘故，一些 PV 具有舍弃相位信息的选项，只保留振幅和频率数据，这样做的结果缩减了数据且相应节省了计算时间，但同时降低了再合成精度。例如没有适当的相位数据，一个再合成波形虽然包含基本的频率内容，但不会像原始波形(Serra 1989)。在某些恒稳态声音中，重新整理的相位或许不可闻，但对暂态和准恒稳态音色的高保真复制而言，相位数据帮助按照适当的顺序重新组装那些短时存在的分量和变化分量，因此是有价值的。

有问题的声音(Problem Sounds)

PV 最擅长处理谐和、静态的或平缓变化的音色。对这些声音的转换如时间拉伸和压缩可以产生自然声音效果。但对某些声音，用 PV 技术进行修改存在着固有的困难，包括嘈杂的声音，诸如刺耳的或喘气的人声、发动机、任何以几毫秒的时间比例不断变化的声音和包含室内噪声的声音。对这些类型声音的转换可能会有回声、颤振(flutter)、不需要的共振和非预期的染色混响效果。这些大都归咎于分析数据被转换时出现的相位失真。

分析不谐和及嘈杂的声音(Analysis of Inharmonic and Noisy Sounds)

实验表明追踪相位声码器可以分析和再合成很多不谐和的声音,包括鸟鸣(Serra and Smith 1990)和有音高的打击乐声音(锣、马林巴、木琴等)的音色。但由于 TPV 是基于傅里叶分析的,其必须将嘈杂的和不谐和信号翻译成周期性的正弦函数集合,特别对嘈杂信号,站在存贮和计算角度,这可能是很消耗的操作。例如为合成一个简单的噪声带,需要不断变化混合很多正弦波,而存贮这些正弦波的控制函数则需要占据相当大的空间。在有些 TPV 中,其量超过原始声音样本 10 倍字节之多,再合成这些正弦波需要极其大量的计算。此外,由于 TPV 允许的转换基于正弦模型,对嘈杂声音的操作结果通常是丢失了它们嘈杂特质的正弦簇。

确定信号加随机信号技术(Deterministic Plus Stochastic Techniques)

为更好地处理这类信号,TPV 被扩展致使其在音乐应用中更有效。塞拉(1989) 为频谱建模合成(spectral modeling synthesis,SMS)中的不谐和正弦模型加上了经过滤波的噪声。(见第 4 章及 Serra and Smith1990。)如图 13.18 所示,SMS 将分析数据简化为确定(deterministic)部分(原始声音的显著窄带分量)和随机(stochastic)部分。确定部分跟踪频谱中最显著的频率。SMS 将这些被跟踪的频率再合成为正弦波。跟踪只对最显著的频率分量进行,放弃信号中的其他能量。由此 SMS 也分析余差(residue)〔或余量(residual)〕,其乃是原始频谱与确定部分的差,并被用来合成信号的随机部分。余差被一组简化的频率包络分析和模拟。可以将再合成想象为将白噪声通过由这些包络控制的滤波器。然而在执行中,SMS 使用随机相位值的正弦波,等同于我们解释的受滤噪声。

SMS 方法使用频谱包络和正弦波,而不用滤波器组,使得编辑随机部分变得更容易,以便转换声音。包络的图形化操作对音乐家而言更感性,改变滤波器参数带来技术的复杂化。SMS 的一个问题是,确定和随机部分在感知方面上的连接很薄弱;分开编辑这两部分可能会导致它们之间感知融合度的遗失。

恒定Q值滤波器组分析(Constant Q Filter Bank Analysis)

许多频谱分析方法可以被划定在恒定 Q 值滤波器技术的规范下——20 世

纪 70 年代以来应用于音频分析中(Petersen 1980, Petersen and Boll 1983, Schwede 1983, Musicus, Stautner, and Anderson 1984)。这个家族中有听觉变换(auditory transform)(Stautner 1983)和有界 Q 值频率变换(bounded-Q frequency transform)(Mont-Reynaud 1985a; Chafe et al. 1985)。下一部分讨论的小波变换也可以被归类于恒定 Q 值技术。

回顾第 5 章 Q 值可以被定义为带通滤波器的中央频率与带宽的比值。在恒定 Q 值滤波器中,每个滤波器有着相近或相同的 Q 值。因此高频滤波器的带宽远比低频滤波器的宽,因为像音乐时值一样,恒定 Q 值分析器工作于对数频率比例。例如,一个三分之一八度滤波器组是一个恒定 Q 值设备。

恒定 Q 值与传统傅里叶分析(Constant Q versus Traditional Fourier Analysis

恒定 Q 值滤波器组对数的频率分析与普通的傅里叶分析器不同。傅里叶分析将频谱分割为一套等间的频线,其中线数是输入样本的一半(对真实信号,负频率分量复制正频率分量)。在傅里叶分析中,线的宽度是一个尼奎斯特采样率除以线数的常数。例如,对于采样速率为 48kHz 的 1024 点 FFT 而言,频线宽度为 24 000/1 024 或 23.43Hz。

当 FFT 的结果被翻译成对数比例(如音乐中的八度)时,很明显低音区的分辨率最差。区分 E1(41.2 Hz)和 F1(43.65Hz)这两个音程关系为半音的低频音高需要长时间的取样窗(举例来说 2^{14} 或 16384 个样本)。但是对高频使用同样的分辨率就是浪费,因为在 1 万到 2 万 Hz 所构成的八度范围内,人类要分辨相差 2.45Hz 的两个乐音有困难。因此我们听到的对数频率体系与 FFT 分析的线性频率比例之间存在着失谐现象。这个问题由类似恒定 Q 值转换这样的方法解决,其带宽随频率按比例变化。也就是说,低频时分析带窄而高频时宽(图 13.19)。因此在恒定 Q 值分析中分析窗长度根据被分析频率而变化。宽窗幅分析低频,窄窗幅分析高频。

恒定 Q 值滤波器组无法避免时间和频率间的不确定关系,前文有所讨论,但时间不确定性只集中在低八度区,那里分析带窄,且由此取样窗和滤波器脉冲响应长。因为暂态声音(触发)趋向于包含高频分量,恒定 Q 值响应具有定位低音区之高频时间的优势。

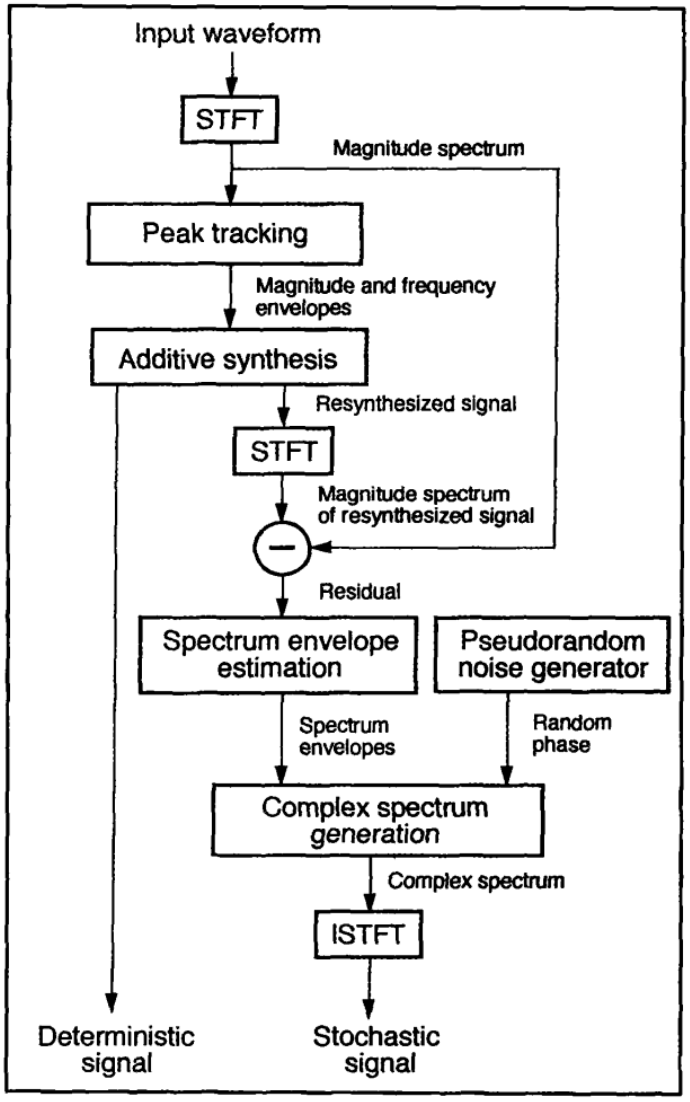


图 13.18 塞拉(X. Serra)的频谱建模合成技术。确定部分遵循严格的正弦加法合成方法。信号的随机部分得自确定部分(准谐波)的再合成与 STFT 输入波形之差。系统简化每个余差分量通过为其适配包络。包络表示法使音乐家们更容易编辑随机部分。随机部分的再合成之后与随机相位分量——等同于受滤白噪声一起使用这些包络。

Input waveform=输入波形 Magnitude spectrum=幅度谱 Peak tracking=峰值跟踪
Magnitude and frequency evnelopes=幅度及频率包络 Additive synthesis=加法合成
Re synthesized signal=再合成信号 Magnitude spectrum of resynthesized signal=再合成信号的幅度谱
Residual=余差 Spectrum envelope estimation=频谱包络估测
Pseudorandom noise generator=伪随机噪声发生器 Spectrum envelopes=频谱包络
Random phase=随机相位 Complex spectrum generation=复杂频谱生成
Complex spectrum=复杂频谱 Deterministic signal=确定信号 Stochastic signal=随机信号

恒定 Q 值技术的另一个吸引人的特色是，人耳具有同恒定 Q 值响应类似的频率响应，特别是高于 500Hz(Scharf 1961，1970)。也就是说，听觉系统采用类似由频率决定带宽的滤波器组分析。这些被称作临界频带(critical bands)的规整听觉频带是如此基本的自然属性(更多有关临界频带的内容请参见第 23 章)。图 13.20 所绘为用在所谓听觉变换中的具有 23 个带通滤波器的滤波器组的中央频率和带宽，其基于斯汤纳(Stautner，1983)的临界频带数据的近似值。为增加频率分辨率，斯汤纳还使用过 79 至 3177Hz 间 42 个滤波器的版本。

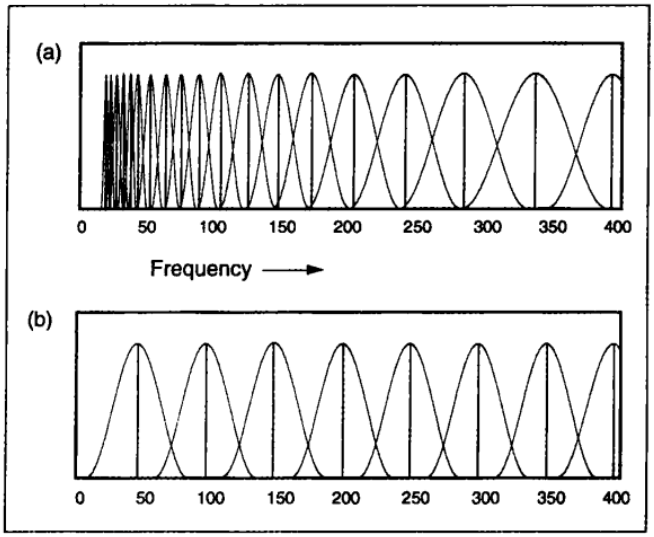


图 13.19　恒定 Q 值与傅里叶技术的滤波器间距。(a)仅使用 43 个滤波器(显示了 19 个)从 20Hz 至 21kHz 范围内，恒定 Q 值方法获得 1/4 八度频率分辨率；(b)傅里叶滤波器间距，每 46Hz 一个频带。使用大约 12 倍的滤波器数(512 个或被显示的 8 个)，傅里叶方法仍然无法和恒定 Q 值方法一样拥有相同的低频分辨率。傅里叶方法在整个听阈频带中具有 46Hz 的分辨率，即使人耳根本无法精确地分辨最高音区的音高差异。
Frequency= 频率

恒定 Q 值分析的施行(Implementation of Constant Q Analysis)

直接施行恒定 Q 值分析的方法是，使用滤波器带宽与其中央频率成比例变化的滤波器组(Stautner 1983)。通过测量几十个这样滤波器的输出，我们可以相当精确地估量输入信号的频谱。直接方式的主要问题是，其无法拥有 FFT 计算的高效率优势。因此研究主体集中在建立一个基于传统 FFT 分析产生的数据之上的恒定 Q 值分析上(Nawab，Quatieri and Lim 1983)，或集中在诸如“频率卷积(frequency warping)”，一个实施 FFT 的固定滤波器(Musicus 1984)这样的方法上。

恒定 Q 值算法也许不如那些基于快速傅里叶变换的高效，但对数分布的分

析通道意味着恒定 Q 值法的通道数或许少些，从而维持与 STFT 相同的分辨率感受。STFT 分析中典型的通道数在几百和几千之间。覆盖相同音域所需的恒定 Q 值滤波器通道数一般小于 100。

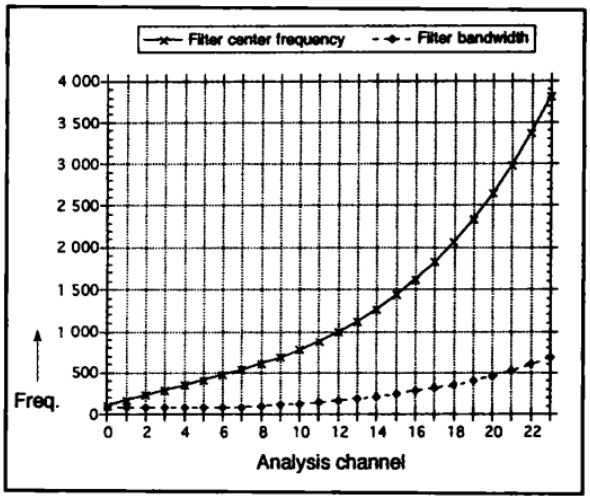


图 13.20 斯汤纳开发的所谓听觉变换系统的中央频率和带宽。图中所示为中央频率为 99 至 3 806Hz 之间，且带宽从 80 至 700Hz 之间的 23 个分析通道数据，其类似人耳的临界频带响应。

Filter center frequency=滤波器中央频率 Filter bandwidth=滤波器带宽 Freq.=频率
Analysis channel=分析通道

恒定 Q 值滤波器组带来的另一个结果是具有可翻转性。恒定 Q 值滤波器组的存在使再合成法失去了存在的必要性。有些恒定 Q 值滤波器组已经提供了这种功能，有些则没有。

小波分析(Analysis by Wavelets)

小波变换(wavelet transform，WT)原为物理及声学所用，由马赛大学(University of Marseille)的科学家们发明(Dutilleux，Grossmann，and Kronland-Martinet 1988，Kronland-Martinet and Grossmann 1991，Evangelista 1991，Boyer and Kronland-Martinet 1989，Kronland-Martinet 1988 Strang 1989，Kussmaul 1991，Vetterli 1992)。小波是一个形似具有平缓触发和衰减的正弦曲线的信号。“小波”及其法文同义词“ondelette”这一术语并不新鲜，20 世纪早期被物理学家们用来描述原子处理发射出的能量包(Carwford 1968，Robinson 1982)。

从音乐的角度来看，WT 可被看成是恒定 Q 值滤波器的一个特殊情况范

例。小波将“短时”概念或“颗粒”表示法注入到恒定 Q 值滤波器模型中。WT 表示及控制声音，将其映射到一个时间-频率栅格(grid)或平面(plane)上。这个格上的每个矩形表示它的不定积(uncertainty product)。格中央是发生事件和频谱质心的平均时。这样的栅格也暗含恒定 Q 值法，但很少被显而易见地使用。在通过 WT 进行的音乐分析中，根据分析目的进行栅格的设定，同时根据再合成的目的进行栅格的失真。

小波理论中，每个输入信号可以被表示为拥有精确开始时间、时值、频率和初始相位的小波集合。典型原生态的音乐小波具有高斯包络(见第 5 章)，但其他类型的小波包络可以被定义。因此小波与第 5 章中讨论的颗粒相像，也与本章先前讨论过的短时傅里叶变换的窗取片断相像。小波独特的地方是，无论其包含什么频率，其总是封装常数个循环。这表明小波窗大小根据被分析频率扩展或缩减(图 13.21)。这个拉伸和收缩在文献中被引用为伸缩(dilation)，且经常被特指为 1/频率的系数。

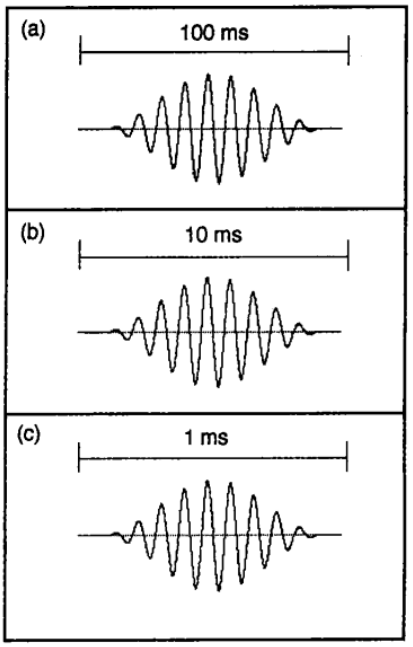


图 13.21　不同频率基本小波的伸缩。高频时小波时值收缩以便波形周期数保持不变。(a)100Hz小波；(b)1kHz 小波；(c)10kHz 小波。

膨胀窗幅的执行是这样的，高频时 WT 用频率分辨率换取时间分辨率，低频时用时间分辨率换取频率分辨率。因此 WT 在精确侦测高频暂态信号发生时间的同时，低频声谱同样也解决得很好。

小波分析的操作(Operation of Wavelet Analysis)

WT 将输入信号乘以一个分析小波的栅格，其中栅格被频率轴及另一个轴

上的时间伸缩系数界定限制(图 13.22)。这个乘法操作相当于一个滤波器组。实际上,一种看待小波的方式就是其每一个所代表的是一个带通滤波器的脉冲响应。

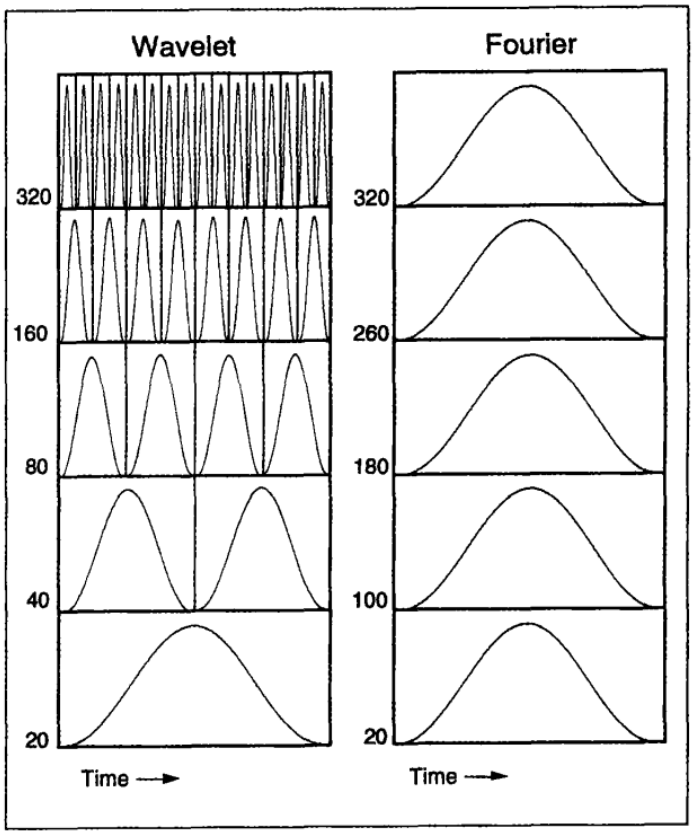


图 13.22 相同时间/频率区域中,小波对短时傅里叶表示法。左图中小波栅格在频谱高频范围中拥有很好的时间分辨率,而短时傅里叶的分辨率是常量。
Wavelet=小波 Fourier=傅里叶 Time=时间

脉冲响应的伸缩与频率轴反转对应。由此,每个小波的时值对应一个滤波器的中央频率,更长的小波,其中央频率更低。

WT 窗取输入信号的同时在每个分析小波的频率上测量输入信号的能量。结果是另一个栅格,每个单元〔在每个核函数(Kernel)中〕上的能量是对原始信号时间—频率能量的反射。WT 的输出是,与短时傅里叶分析中一样,一个包括两部分的频谱,一部分代表某已知频率的幅度,另一部分表示相位。

分析栅格的频率轴呈典型的对数型。这意味着每个分析小波与其他小波的频率之间为对数的音乐时值关系如五度、三度或无论哪个,取决于系统设置的方式。对数关系的使用并不是强制性的,由于 WT 可与任意频率比例关系相匹配。当然,小波的时值是根据它们的频率改变的。

小波变换的直接计算是繁重的，类似于加载离散傅里叶变换的直接计算（见附录）。目前已经提出了许多有关削减小波变换计算量的提议（Dutilleux，Grossmann，and Kronland-Martinet 1988，Mallat 1989，Evangelista 1991）。有关这些算法的细节请参考文献中的记载。

小波图（Wavelet Display）

小波研究的一个副产品是马塞（Marseille）团队使用的一种显示方法，如图 13.23 所示。这可以被看作是翻了个儿的按时间投射的传统频谱图。另一种方法将其视为语图的一种：按水平为时间，低频在底部，高频在顶部绘制。

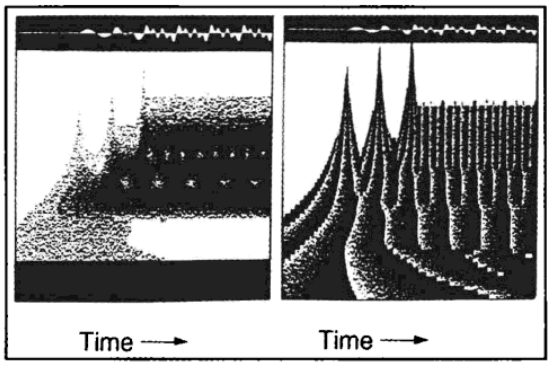


图 13.23　小波图示的三个迭盖正弦波。小波图包含两个部分：模数（modulus）〔或幅度量（magnitude）〕，显示在图左，还有相位显示。两部分都按时间从左至右显示。纵坐标按对数频率显示，每个顶部都有一个普通时域的参考波形图。（a）在模数中，暗部指示出能量，注意高频“指针”显示出每个正弦波的发生时间；（b）相位图基本上直接显示波形的偏移，U 状“山峰”跟随波形的峰值。任何变化显示为无序表面，同样，“指针”表明即时变化。（出自 Arfib 1991。）
Time=时间

语图和小波图之间的差异在于它们投射的时间定位模式不同。短的小波按时间定位侦测暂态量。这些小波位于频率—时间平面图三角形的顶端（图 13.24a）。长的小波侦测低频，它们位于三角形底部，随时间延展（模糊）开。这个三角形是小波的时域影响（domain of influence in time）。频域影响（domain of influence for frequencies）是常量水平条带，如声谱图中所示（图 13.24b）。条带颜色越深，该频率范围的幅度越强。

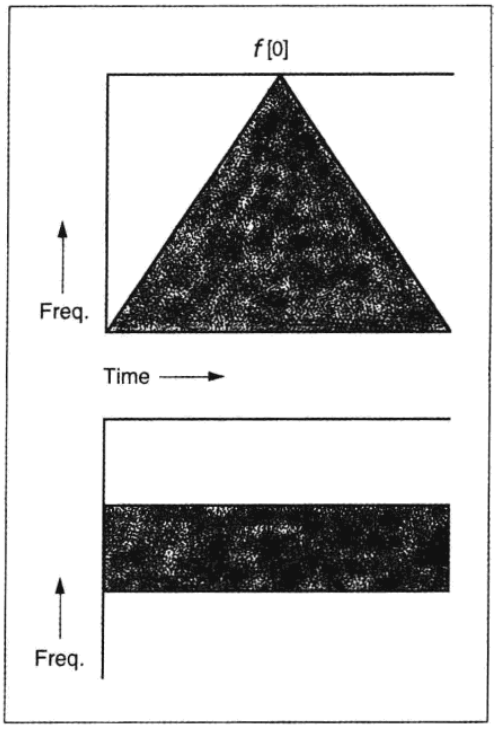


图 13.24 小波影响域。(a)时间;(b)频率。解释请参看正文。
Freq.=频率 Time=时间

当然,此图示技术只是诸多投射 WT 产生的数据的方法之一。在马塞(Marseille)的图形中,模数(幅度)和相位都被绘制。相位频谱有时被叫做阶谱(scalagram)。仅当超过指定数量阈限时才被绘制,以避免产生不可靠的判断。

如果频率栅格与音乐的音程相对齐,当输入信号包含该音程时,图中投射出深色的指示。图 13.25 所示为被设置来进行八度侦测用的 WT。图中的四处八度音程以深三角形的方式呈现。在本例中,可以说正进行分析的小波是两个频率相隔一个八度的简单小波之和。

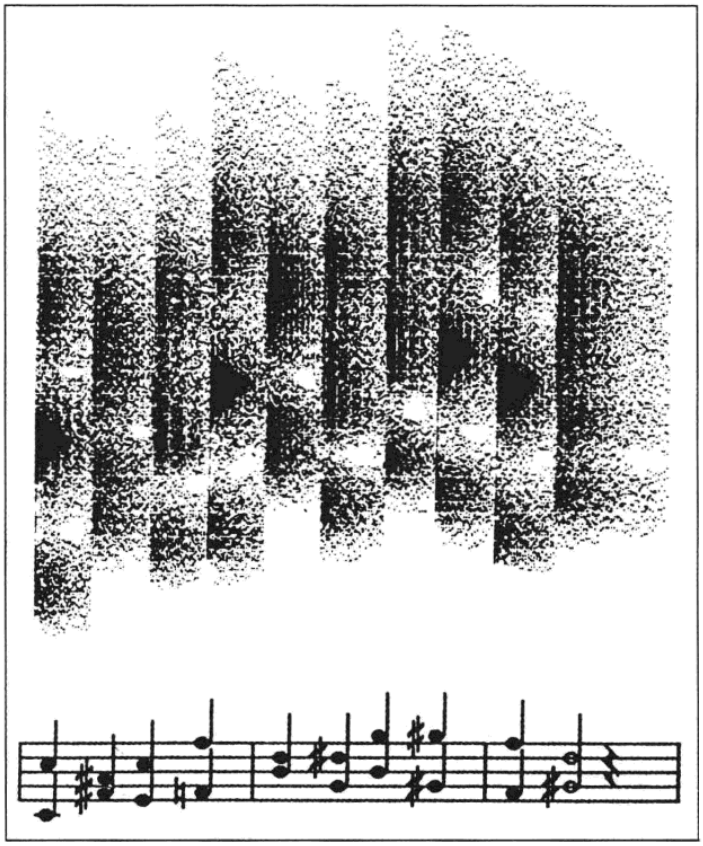


图 13.25　与下面音乐记谱相应的小波变换模数。深色三角形标志着最多出现的八度。（出自 Kronland-Martinet and Grossman 1991。）

小波再合成(Wavelet Resynthesis)

如同 STFT 中一样，小波再合成可以按两种方式实施：迭加(overlap-add)及加法(additive)。各个方法使其适合某类型的转换。在迭加情况下，我们需要与迭盖小波数量相当的振荡器。加法再合成中，振荡器数是固定的，因为每个频率分量被指派给其自己的振荡器。

小波声音转换(Sound Transformation with Wavelets)

已经出现了各种不同的基于小波分析/再合成的声音转换(Boyer and Kronland-Martinet 1989)，显然其中之一是再合成中通过抑制某些频道的方式执行某种滤波处理。对数关系相间的频道使得从声音中提取和弦更容易。举例来说，当这个技术被应用到语音上时，会产生一种某人讲话呈“谐和”的

感觉。另一种效果是一种交叉合成，使用一个声音的振幅分量与另外一个声音的相位分量制造出一个混生声音。

另一类型的转换包括改变频率栅格的几何分布，比如再合成中于每个频率上加上或乘以一个比例系数，时间压缩/扩展(Time compression/expansion)效果也同样可能(卷积时间格)。在频率和时间卷积中，相位分量都必须乘以与音高或时间操作相同的比例系数(无论被修改的是哪个)。〔这被称作相位展开(phase unwrapping)，有关小波变换中的相位展开请参见阿菲伯(Arfib 1991)，附录中介绍了相位声码器中的相位展开处理〕。克罗兰-马提内(Kronland-Martinet 1998)描述了一种再合成中基于波形塑型相位值的音高位移方法。

谐和频谱中噪声的梳状小波分离
(Comb Wavelet Separation of Noise from Harmonic Spectrum)

梳状小波变换(comb wavelet transform)开发于那不勒斯大学，用来在准周期信号中将暂态的，不带音高的声音和有音高的声音区分开来(Evangelista 1992; Piccialli et at. 1992)。梳状 WT 开始自声音的窗取片断，基频音高周期被估测出，且一个梳状滤波器与该片断相契，峰值与基频的泛音相对齐。梳状滤波器筛选出谐和频谱中的能量，之后对这个“干净”的谐和信号实施小波分析。当倒 WT 被从原始信号中减去时，余差信号或“脏”的部分保留下来(图 13.26)。“脏”的部分包括触发暂态和赋予该声音个性和特点的细节。

一旦“干净”和“脏”的部分被分离，可以通过将一个声音的“脏”的部分移植到另一个声音的“干净”部分上的方式进行一种交叉合成。这类分离在概念上类似于——虽然执行上不同于前面介绍过的，塞拉(1989)用在频谱建模合成中的技术。

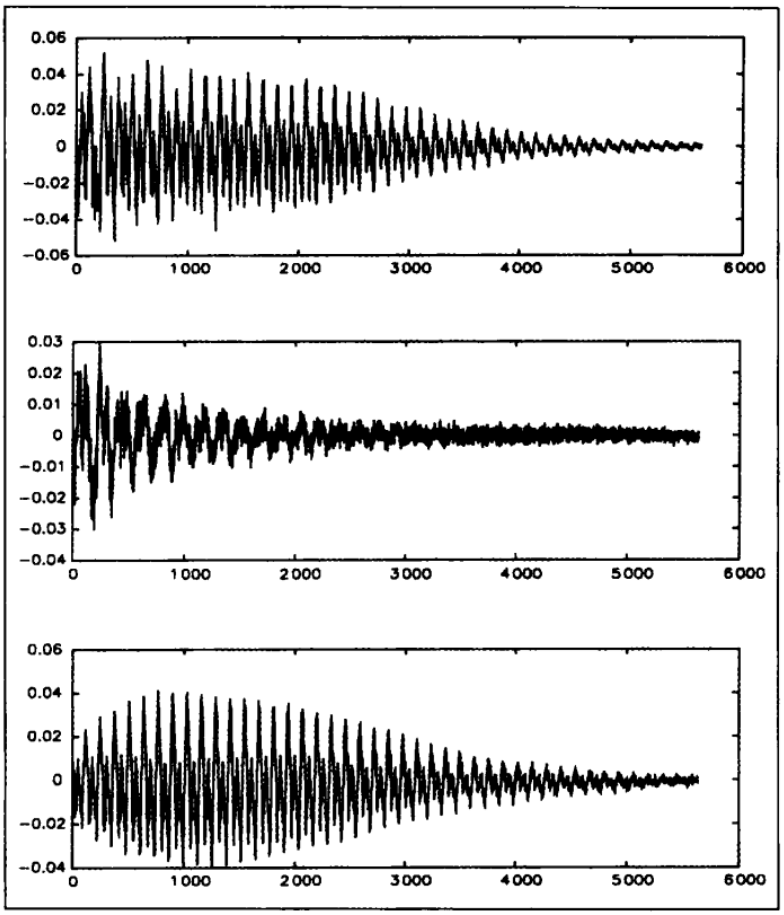


图 13.26　谐和频谱中噪声的小波分离。振幅(垂直方向)-时间(水平方向)图。最上面为原始吉他音色。中间部分为从梳状小波变化而来的嘈杂余差，其中包括富有特色的音符触发部分。底部所示为合成自梳状小波方法准谐和部分的再合成〔图片授权自那不勒斯大学，吉安保罗·埃万杰利斯塔(Gianpaolo Evangelista)。〕

小波分析与傅里叶方法的比较
(Comparison of Wavelet Analysis with Fourier Methods)

传统傅里叶方法测量时值保持为常量的取样窗中的平均能量，无论被分析的是哪个频率分量。这往往导致对高频暂态发生时间的观察发生移位(delocalize)。相反，WT 提供检视音乐信号的多重分辨率，因为细微瞬时分析是由短的高频小波完成的，而细微频率分析使用长的低频小波。对于“慢的”(低频)小波而言，镲的敲击声持续为不可见，可将被很“快”的小波迸发侦测到。由此，WT 更适宜研究音乐信号的暂态或发生时间。如图 13.27 所示，在高频区 WT 图显示出非常的时间敏感力。

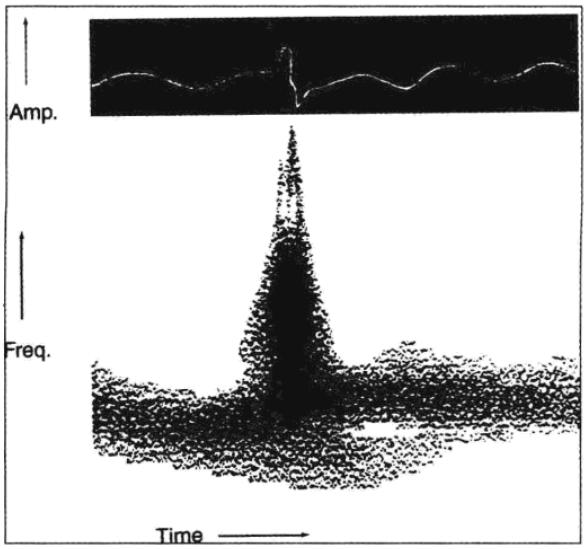


图 13.27 由小波进行的暂态侦测。上图所示为时域信号中的一个短时脉冲波形干扰。下图为小波表示图。高频小波正好指向干扰时间，对低频小波（底部的水平带）是不可见的。（出自 Kronland-Martinet 1988。）
Amp.=振幅 Freq.=频率 Time=时间

在注重计算效率的应用中，对于类似分辨率来说，基于 FFT 的方法比小波和其他恒定 Q 值方法具有优势。虽然已经开发出类似严格对数频率栅格的 WT 优化方法(Dutilleux, Grossmann, and Kronland-Martinet 1988)。更多有关快速小波技术亦请参见 Shensa(1992)。

维格纳分布信号分析(Signal Analysis with the Wigner Distribution)

维格纳分布(Wigner distribution, WD)首次应用于 20 世纪 30 年代，用于解决量子物理学的问题(Wigner 1932)。在声学应用中，本质上 WD 的目的不是声音分析，而是系统分析。换言之，输入 WD 的可以不必是声音，而是音箱、换能器或声音电路的响应。WD 将描述出这个系统相对时间的频率分布。站在理论的角度，WD 是其他基于傅里叶方法的直系亲属，如语图。更多有关 WD 数学方面的内容，请参考 Janse, Kaizer(1983, 1984); Preis et al. (1987); Gerzon(1991)。

维格纳分布图释义(Interpreting Wigner Distribution Plots)

典型 WD 的输入既不是被测量系统的脉冲响应也不是振幅频率响应(参见

第 5 章有关振幅频率响应的定义)。其输出的是频率时间图。工程学测量诸如群延迟、瞬时频率和功率、瞬态畸变及频谱可以得自 WD 图,其可以二维或三维形式来展示。以二维图来说,特定频率横切片之下的区域给出该频点的频率响应(幅度平方)值(图 13.28a)。横切片的重心(center of gravity)(所有区域可以被集中在垂直轴向上产生相同"重量"的那一点)给出该频率的群延迟。在图 13.28a 中表示为一个黑点。同样地,特定时间纵切片下的区域产生该时刻信号包络的瞬时功率(instantaneous power)(图 13.28b),其实该切片的重心等同于瞬时频率(instantaneous frequency)(图 13.28b 中的黑点)。本例中,x 轴和 y 轴均为对称图,因此重心位与中央。在真实信号中,它们随着信号的不同而不同。当按时间变化绘制瞬时功率和瞬时频率,可以揭示出施加在信号上的振幅调制和频率调制的效果。

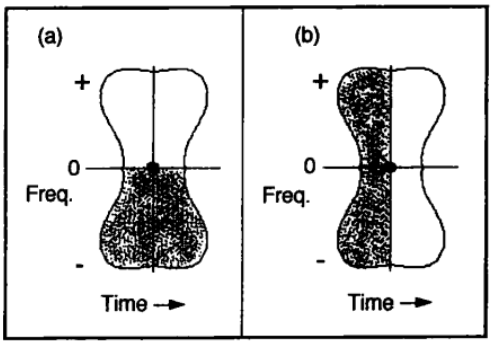


图 13.28　维格纳分布图释义。解释见正文。
Freq.=频率　　Time=时间

Janse, Kaizer(1983, 1984)提出了三维图示和释义 WD 的指导方针。在实践中,他们将理想系统(例如理想滤波器)与真实世界的设备如音箱进行比较。

维格纳分布的局限(Limits of the Wigner Distribution)

实践中,WD 是基于样本和窗取数据的,且有时被称作伪维格纳分布(pseudo-Wigner distribution)(Janse and Kaizer 1983)。正如在其他分析技术中一样,已知失真会由采样和窗口化引入。这些是相对次一级的效果。

WD 的主要问题是,其是非线性的。也就是说,两信号之和的 WD 不等于它们各自 WD 的和。例如,经过 WD 的单一 100Hz 正弦曲线显示为一个独立的频率分量,单一 300Hz 正弦曲线时也一样。但如果我们将 100Hz 和 300Hz 这两个正弦曲线之和送进 WD 运行的话,我们看到了第三个 200Hz 的分量——两频率差。这种混乱呈示出输入端中不存在的频率。这种混淆给视觉检视音乐信号 WD 图带来困难。

WD对人类声音听觉的现实意义有限。它的图以图形方式刻画可感知的相位失真(见第10章)。从图13.29中我们可以看出这一点。图13.29的x轴所示时间为0至5毫秒。y轴所示为时间-频率失真,范围从-6.25至+6.25kHz,正频率是负频率的反相图像。沿x轴某些频率的拉伸清晰地表明由频率决定的群延迟的效果。有关如何计算这些图的细节参见Janse and Kaizer(1984)及Preis et al.(1987)。

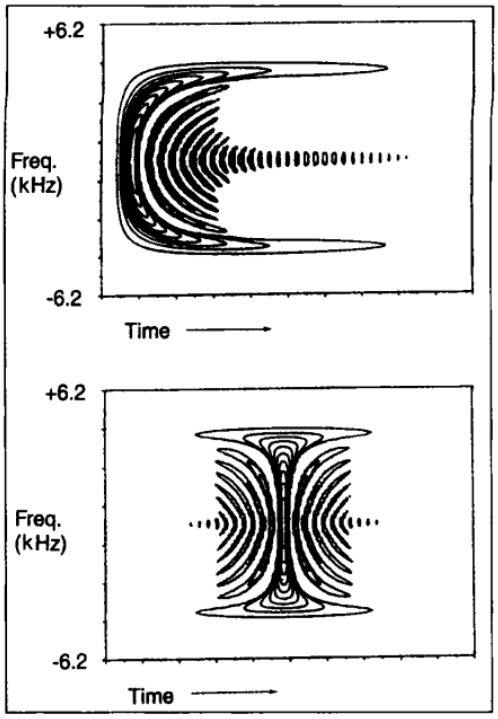


图13.29 维格纳分布图与两个低通滤波器的比较。历时5毫秒。(a)相位失真滤波器。沿时间轴频率的拉伸是相位失真的明显迹象。两个滤波器间有着强烈的可闻差异;(b)线性相位(未失真的)滤波器。(出自Preis et al. 1987。)
Freq.=频率 Time=时间

非傅里叶声音分析(Non-Fourier Sound Analysis)

本部分探讨一些传统傅里叶频谱分析的问题,以及简要地考察一些另类方法,包括自回归分析、信号源/参数分析,以及根据正弦波之外的其他正交函数进行的分析。

审视傅里叶频谱分析(Critiques of Fourier Spectrum Analysis)

对于有限长度信号而言，基于传统傅里叶方法的频谱分析具有根本上的局限。首先，频率分辨率有限(不能区分两个相近的频率)，特别是在小样本的情况下；第二，FFT 中隐含着以窗口化副作用形式出现的频谱领域的“遗漏”(Gish 1978，Kay and Marple 1981)。傅里叶分析天生是分析嘈杂声音的低效分析法，因其假设该声音是由具谐和关系的正弦波构成。傅里叶方法中内在的周期性假设当分析复杂暂态现象时会导致误差。

为减少 FFT 方法的局限，很多另类频谱分析方法被提出来。图 13.30 显示了从三个正弦和一个经滤波的噪声带这样的输入可以获得的方法多样性及其结果的差异性，如图 13.30a 所示。傅里叶方法展示于图 13.30b、图 13.30c 和图 13.30g。它们无法解析正弦曲线或甚至无法将其从噪声中分离出来。一个像图 13.30k 这样的技术可以精确地测量这三个正弦曲线，但其后将噪声带描绘成五个正弦曲线的和！显然，没有始终都是“最好”的频谱测量技术，一切取决于我们正在寻求的内容。

自回归频谱分析(Autoregression Spectrum Analysis)

自回归(AR)、线性预测编码(LPC)和最大熵法(MEM)组成一个家族，其本质上采用相同的技术，即根据输入信号的频谱设计滤波器(Makhoul 1975，Burg 1976，Atal and Hanauer 1971，Flanagan 1972；Markel and Gray 1976，Cann 1978，1979，1980，Moorer 1979a；Dodge 1985，Lansky 1987，Lansky and Steiglitz 1981，Hutchins 1986a)。因为这样才可能将它们应用为频谱分析方法。这里我们将这三种方法皆视为 AR 范畴下。(第 5 章中介绍过带编辑的实用 LPC 音乐系统。)

AR 法与傅里叶法相比其优势之一是它们可以从很少量的输入数据中估测出频谱，因为它们具有改良的时间/频率分辨率的潜质。但 AR 执行的频谱分析形式不能直接与傅里叶分析进行比较。AR 模型假设频谱是一个激励信号(如声带发出的声门脉冲)加载于共鸣器(声道的其他部分)上的结果。AR 估测共振的总体频谱形状而不是许多独立频率的能量。图 13.30d 显示了这个效果。

AR 法获得几个输入样本然后以最近的样本作为参考。其试图从以滤波器系数为权重的过往样本和中“预测”出这个样本。作为这种预测的副作用，AR 算法适合输入信号频谱的倒滤波器。正是这个副作用具有音乐意义。当倒滤波器自身被反转——一个平常的步骤——结果滤波器的响应是对输入信号频谱的一种估量。

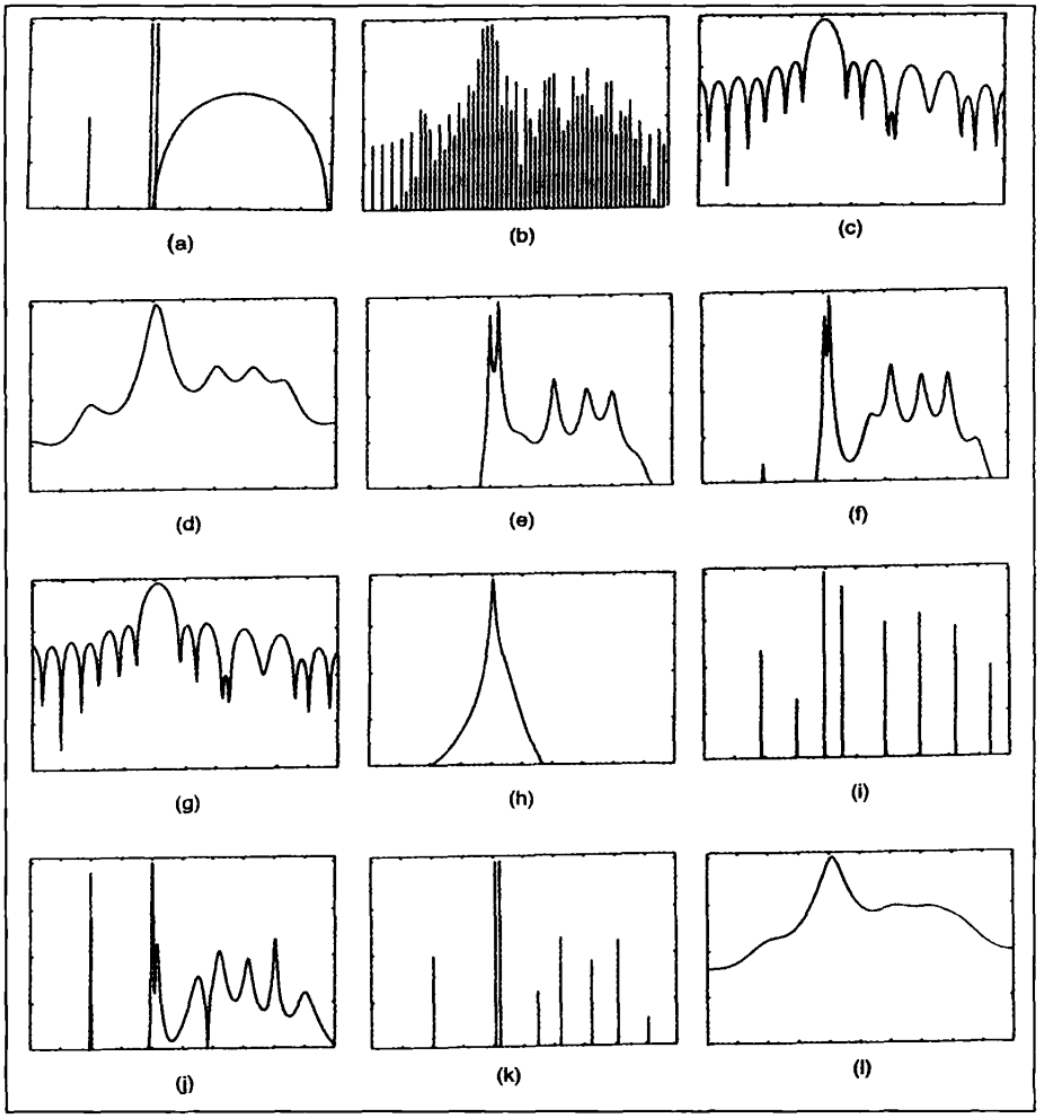


图 13.30 对一个输入声音的不同频谱测量方法。在描述中,"PSD"意思是幂频谱密度。所有情况中水平刻度都是频率,从 0 至采样速率的一半。纵向刻度是振幅,从顶端的 0dB 至底部的 -40dB,线性绘制。(a)输入信号源,包括三个正弦和一个噪声带;(b)双零填充 FFT 周期图;(c)布莱克曼-图基 PSD;(d)自回归 PSD 与耶尔-瓦克尔(Yule-Walker)法;(e)自回归 PSD 与博格(Burg)法;(f)自回归 PSD 与最小平方法;(g)移动平均数 PSD;(h)ARMA PSD 与扩展的耶尔-瓦克尔(Yule-Walker)法;(i)皮萨仁苛(Pisarenko)谐波分解;(j)普郎尼(Prony)PSD;(k)特殊普郎尼与希尔德布兰德(Hildebrand)法;(l)卡彭(Capon)或最大概似法。

AR 法根据如下公式预测信号的第 t 个值：

$$signal[t]=\sum_{i=1}^{p}\{coeff[i]\times signal[t-i]\}-noise[t]$$

换言之，预计值 $Signal\ [t]$ 是通过对预测滤波器系数的 p 值与已知 *Signal* 的 p 值的卷积运算求得的。（第 10 章中有关于卷积的介绍）。p 的选择是一个复杂问题。p 的值过小，产生过于平滑的频谱；选择过大的 p 带来伪峰值。因此必须根据具体应用调整该参数（Kay and Marple 1981）。现存有交互式的 p 选择法。当 p 从小值增大时可以测量预测的吻合度。当吻合度不再改进时，停止对吻合度的测量。

通常 *noise*[t]被假定为一个白噪声驱动的信号，经滤波为产生与输入信号相匹配的频谱。一些采用线性回归（linear regression）方法的算法可以从数据块中计算滤波器系数——由此得名“自回归”。这个过程由记载于工程学文献中的矩阵操作完成（Burg 1976，Makhoul 1975，Markel 1972，Markel and Gray 1976，Bowen and Brown 1980）。参见 Kay 和 Marple（1981）有关这些方法的比较。

自回归移动平均值分析（Autoregressive Moving Average Analysis）

AR 法对于圆顺的，带有明显峰值但无深空的连续频谱而言是一个有效的模型。因此其对诸如鼻音元音——频谱中有空洞——或打击性脉冲（军鼓、吊镲等）这样的声音不能很好地建模，其预测误差大。对这类声音而言，一个被称作自回归移动平均值（autoregressive moving average）（ARMA）的 AR 法的一般化方法或许是更好的选择。ARMA 通过结合过往输入信号和过往输出值得出输出样本。因此，ARMA 滤波器既有极点又有零点，潜在地比 AR 法更精确。当然，站在计算角度 ARMA 滤波器比 AR 耗费的资源要多得多。

源信息和参数分析（Source and Parameter Analysis）

一些分析种类，著名的 AR 法、普言谱（对数倒频谱）分析（描述见第 12 章）和第 7 章中介绍的物理模型（physical model）法，其目标不仅仅是简单地得出信号中现有的频率，而是重新获得源信息，如再合成该声音时需要的激励和共振参数。该方法对诸如军鼓敲击、捶击吊镲等这些具有相当音乐意义的声音很有用。这类声音携带着大量的源信息，比如它们的尺寸、质量、几何形状以及制造它们的材质。实际上，这些技术背后的科学动机是将它们应用于将信号自噪

声中分离，或解析混合信号(Kashino and Tanaka 1993)。

参数估计(Parameter Estimation)

所有声音分析都是一种形式的参数估计(parameter estimation)，其试图按照参数设置分析输入声音，该参数设置是以某特定合成方法近似模拟出该声音所要求的(Tenney 1965, Justice 1979, Mian and Tisato 1984)。例如，我们可以将傅里叶分析看作是一种正弦波形再合成的参数估计方法，因其计算所有趋近输入声音所需的频率、振幅和相位。

理论上，参数估计可以应用于任何一种合成技术中。实践中，不能保证任意一种合成方法对某特定输入信号的模拟都获得成功。无数为频率调制合成而开发的参数估计分析的尝试，都是对原始输入声音的大致近似。不存在通用的分析/再合成技术，某些技术不是为产生某些特定类型声音而设计的。

某些类型的参数估计采用适应性(adaptive)信号处理算法，其试图通过调整仿真模型的参数，最小化输入信号与仿真结果之间的误差。在实时系统中，测量和调整必须在信号样本周期时间内完成，迫使对理想数学算法进行折中。

第 7 章中有呈现关于物理模型合成的源信息分析，因此我们推荐读者参考那部分内容。

其他函数分析(Analysis by Other Functions)

傅里叶法将正弦波形相加以再造一个特定输入信号，但正弦波仅是用来分解然后复制一个特定输入函数的大批函数之一。沃尔什函数(Walsh functions)(方波)和复指数(complex exponentials)(带衰减振幅包络的正弦曲线)就是这些基本单元中的两个成员。可以被考虑的其他函数不计其数，但由于这两个具有特殊属性且已经被应用于音乐中，我们接下来对其进行讨论。

沃尔什函数(Walsh Functions)

沃尔什分析的主要优势在于其基础单元——二元脉冲或方波——对于数字系统而言，好像执行起来很自然，比如表面上比正弦波更自然。沃尔什分析的不利因素在于其将信号分解成与频域无直接关系的叫做序列(sequencies)的集合。由于第 4 章中对沃尔什函数有更详细地介绍，我们推荐读者参考该部分内容。

普郎尼法(Prony's Method)

阻尼正弦曲线是被称为普郎尼法(Prony's method)中的基础单元(Kay and Marple 1981, Marple 1987, LaRoche and Rodet 1989)。关于阻尼正弦曲线,我们指的是具有锐利触发的正弦曲线,但突然被削弱,通常按指数式衰减。该技术是以 Gaspard Riche Baron de Prony 命名的,他原先发明了一个分析各种气体膨胀的方法(Prony 1795)。该技术的现代版本进化自普郎尼原来的方法,且与先前讨论过的 AR 法有类似之处。

普郎尼法现在是由一些相关技术组成的一个家族,将输入信号建模为阻尼正弦曲线加上噪声的结合体(Kay and Marple 1981)。像 AR 技术一样,普郎尼法根据过往输入样本估计一组系数。但取代 AR 法中驱动一个滤波器,普郎尼法中的系数驱动趋近输入信号的频率、阻尼系数、振幅和一组被阻尼的正弦曲线的相位。通过对普郎尼法产生的输出信号进行 FFT 将普郎尼法变成一种频谱分析技术。与 AR 技术相比,普郎尼法的优势之一是其产生相位信息从而使再合成更精确。有关该方法的算法介绍请参见 Marple(1987)。

在计算机音乐中,普郎尼法被应用于 CHANT 合成系统(d'Alessandro and Rodet 1989,参见第 7 章)的分析平台以及 LaRoche(1989a, b)中介绍的的一个试验性的分析/再合成系统中。拉罗什(LaRoche)用其分析和再合成阻尼打击乐声音,如钟琴、颤音琴、马林巴、钢琴低音区和锣声。根据拉罗什(LaRoche)的描述,对钢琴高音区和吊镲等声音的分析结果低于预期。

在普郎尼法与傅里叶分析的比较中,拉罗什(LaRoche)注意到,一般来说,普郎尼法比傅里叶分析更"敏感"。使用者必须细致地调整分析参数,然则结果频谱估计可能与真正的频谱具有很少的共同之处(LaRoche 1989a)。相反,傅里叶方法中的主要参数是取样窗。傅里叶分析的结果可能会不完善和不够精确,但绝不会完全地令人不知所云。

当普郎尼法的参数被很好地设置,其对不谐和分音的计算有问题且会将间距很近的多重正弦分量分解。相反,傅里叶分析武断地将频谱分成等间谐和分音,且将间距很近的正弦分量总括为声谱中的一个总体的类共振峰值。普郎尼法被限制在一次分析 50 个分音以下,因为该点之下用于计算它的多项式不会收敛为一个解式。且普郎尼法比傅里叶分析计算起来更密集。小节一下,在普郎尼法中,如果预先谨慎设置得当的话,我们便有了这样一个可以精确解析某些类型信号的分析方法,特别是对包含一些正弦分量的打击乐音色而言。

听觉模型(Auditory Models)

我们可以将声音分析方法分为两极:试图模拟已知人类听觉系统行为的方法,和那些不这样做的方法。前者为听觉模型(auditory models);后者为受数学启发的技术如维格纳分布(Wigner distribution)。听觉模型通常始于频谱分析,但该环节的输出仅是根据听觉机构的计算模型或多或少进行的精密后加工的起点(Mellinger 1991)。

听觉模型的目标有两部分:(1)更符合我们感知的、更清晰的、对音乐信号的审视;(2)通过在模拟试验中使用模型以便更深地了解人类听觉机制。这里我们简要地介绍两种听觉模型:分别名为耳蜗图(cochleagram)和相关图(correlogram)。

耳蜗图(Cochleagrams)

耳蜗(cochlea)是内耳中蜗牛状的小器官,其将接收到的振动转换为传输给大脑的神经脉冲(见第 23 章)。沿耳蜗长度上的各个区域对广泛围绕特属该区域的中央频率的振动作出响应。听觉科学家们已经测量出沿耳蜗长度神经的平均触发率,且已明确它们与被耳朵感知到的不同频率有关。

耳蜗响应接收信号的软件模型被称为耳蜗图(cochleagram)(Slaney and Lyon 1992)。与声谱图将频率映射到纵坐标不同,耳蜗图将耳蜗区域(cochlear place)映射到纵坐标上。也就是说,其代表耳蜗不同部分对接收声音的响应。当以低分辨率绘制耳蜗图时,其看起来像语图,但对触发有增强地体现。语图和耳蜗图更重要的差异可以通过图 13.31 观察到。这个放大了的耳蜗高分辨率图揭示出语音信号中每个声门脉冲的时间。由此,耳蜗图提供了一个对低级别时间(触发)和频谱都可以进行研究的方法。

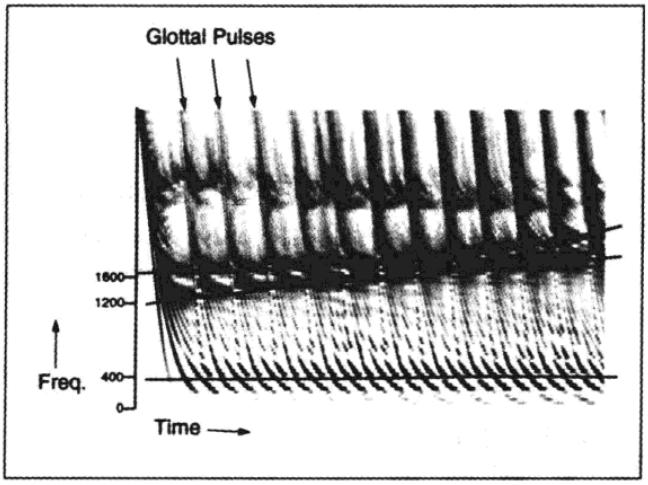


图 13.31　美语双元音"ree"的耳蜗展开图。横坐标表示最先的三个共振峰踪迹。纵坐标表明声门脉冲，由于通过耳蜗的自然延迟而略微倾斜。（出自 Slaney and Lyon 1992。）
Glottal Pulses=声门脉冲　　Freq.=频率　　Time=时间

相关图(Correlograms)

相关图(Correlograms)是由利克里德(Licklider;1951，1959)于 20 世纪 50 年代早期提出的，但从计算的立场出发仅在 20 世纪 90 年代才成为可实用的方法。相关图由耳蜗模型开始，接下来是耳蜗图每通道产生信号的自相关(Slaney and Lyon 1992)。根据应用的不同，自相关是"一帧一帧"或基于窗取每秒 30 至 120 次。

结果为一个展示频率、时间和自相关延迟函数的三维图。相关图是"随时间"移动的图像映像。斯莱尼(Slaney)的相关图可以在视频媒介或以运行在个人计算机上的数字影片的形式观看(Slaney and Lyon 1991a,b)。

延耳蜗的位置被投射到纵坐标轴，其中高频在图像的上方。横坐标轴显示自相关延迟。与传统的语图一样，深色区域表示强振幅。具有强音高感受和泛音结构的声音表现为以自相关滞后时间为间隔的垂直线，那里同一时期有大量触发着的耳蜗细胞(有关相关图与音高侦测的应用请参考 Slaney and Lyon 1992)。当音高升高，主要垂直线向间隔左边移动表明周期更短。水平带代表强能量频带，如共振峰。噪声、无音高声音仅显示为水平带，没有垂直线。

回顾第 12 章正弦音的自相关其自身就是一个正弦波，带有以基频为周期的子泛音，即 f、$f/2$、$f/3$ 周期为间隔的峰值。与之类似，通过相关图的单一正弦音显示为根据最左边基频周期的"虚拟"子泛音的一系列垂直线。我们没必要听到这些子泛音，它们是自相关函数周期搜寻这一自然属性的后生现象。

图 13.32 所示为相关影片中的三帧画面，分别取自 0 秒、600 毫秒和 2 秒钟。本例我们看到的是一个敲击的钟琴，开始有很多泛音，声音很丰满。不同泛音按照不同速率衰减，正如第二帧中所示。最后一帧中只剩下两个分量。

相关图的优势在于其展示时间变化的同时显示出音高和共振信息。水平或延迟间隙表示音高，同时纵向维度表示频谱。相关图的计算是一个高强度的计算操作。最近，相关图被用作再合成的基础(Slaney, Naar, and Lyon 1994)。

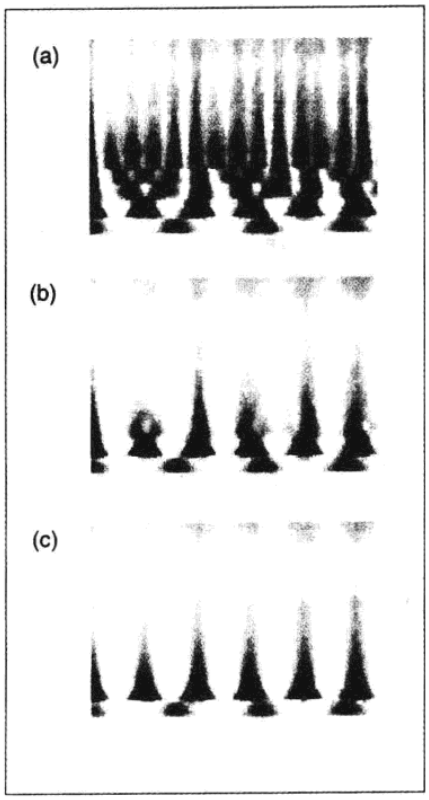


图 13.32 敲击钟琴的相关图。(a)触发；(b)600 毫秒；(c)2 秒钟；U 状的曲线，特别是出现在(a)中的那些，结果自时间栅格的连续分区——正如你在观察频带的波形峰值一样，带有位于底部的低频(且因此峰值间的周期更长)。(出自 Slaney and Lyon 1992。)

言号解读系统(Signal-understanding Systems)

在信号分析中看到综合低级别信号处理工具与来自人工智能研究的软件技术这样的应用正成为司空见惯的事(Nii et al. 1982, Roads 1985d; Oppenheim and Nawab 1992)。这样做的目的是为了将表面的信号流分析引向深层的信号解读(signal-understanding, SU)。当然广义地讲，很明显有太多类型和层面的“解读”，但出于某特定目标而言，如果一个系统可以将音乐信号识别为一个音乐单元或单元集合，且与高于声学层面的音乐概念分析相关联的话，我

们可以说该系统可以解读音乐信号。

我们可以将音乐解读系统分为两类：试图模拟受过训练的人的听觉的（包括人类听觉系统模型）和不进行人类技能效仿的。前者包括实时富有表现力的伴奏、乐器音色分类（声源分离）和复音声源的音乐转记系统；后者包括乏味地分析数据压缩以及从背景噪声中分离出音乐的任务。一个音乐信号解读系统也许打包了很多层面的专门技能。这里我们只有触及全局总体问题和举出典型事例的篇幅。

模式识别（Pattern Recognition）

与那些通过普遍的数学操作将声音数据从一种表示形态转换成另外一种的纯数字的信号处理方法不同，信号解读系统采用预期驱动模式识别（expectation-driven pattern recognition）找出和鉴别音乐的标志（Mont-Reynaud and Goldstein 1985，Chafe et al. 1982，1985，Foster，et al. 1982，Strawn 1980，1985a，b；Dannenberg and Mont-Reynaud 1987）。我们之所以称之为“预期驱动”因为它们被编程用来搜寻典型表征。例如，在一个自动记谱系统中，其由声学信号源开始，分析算法首先在其中查找音符。将音符分割之后，其也许试图从先前乐器分析得出的频谱模版表中鉴别出它们的音色，或许试图根据节奏分组规则惯例将音符划分为如连音和小节这样更大的音乐单元。

低级别的模式识别处理往往依靠来自于人类听觉和心理学研究的线索。使用这些线索，他们可能也可能不试图效仿整个人类听觉机制和音乐认知。高级别模式识别大概更多遵循常规的更取决于文化因素的风格规则。一个为维也纳十二平均律音乐（Viennese twelve-tone music）严格语法开发的音高分类系统，应用在具有细微音律差异的传统印度歌唱上很可能会失败。

控制结构和策略（Control Structure and Strategy）

普通信号处理中，分析策略应该没有变化。例如，每个短时傅里叶分析遵循相同的操作步骤。反之，一个信号解读系统或许不得不设计一个周期性评估的初始策略，如果需要，很可能进行路线修正和采用不同的方法。由此，分析系统的分析控制和策略是其设计的中心问题。这决定着如何在系统不同的分析代理中进行任务分配以及它们之间如何进行内部交流。有时一个叫做黑板（blackboard）的公共存储区，被用来发表不同代理相互竞争的分析策略结果。该信息可以被其他代理使用或由决策管理器从提供给它的不同臆测中进行挑选（Mont-Reynaud 1985b）。

分析系统中不同级别和组成之间的交互是其运行中至关重要的因素(Minsky 1981, Rosenthal 1988)。例如,假设中级节奏分析可以从前一侦测事件中建立一个测量语法关系的话,这个知识可以被告知低级别的事件侦测器接下来有可能出现的事件。再例如,有关复音演奏的乐器频谱知识,可以对试图从音乐中分离出单个声部线条的系统执行有所帮助;另外,Maher(1990)指出多重策略并行出现的问题。

除了像记谱或数据压缩这样明确定义的任务以外,创造高级别音乐分析程序的空间十分广阔(Brinkman 1990, Castine 1993)。这样的系统可以辅助或者接管音乐学家或音乐理论家们的一些烦琐工作。最终,这些程序应该能够足够好地解读作曲结构或为之创造变体。如果那样的任务所需要的音乐知识未被事先编程,一个学习子系统必须被合并到系统中。

信号解读系统实例(Examples of Signal-understanding Systems)

信号解读系统开始自詹姆士·穆勒(J. A. Moorer)在斯坦福大学(Stanford University)进行的划时代的研究,创造一个"音乐记谱仪"(Moorer 1975)。图 13.33 是穆勒策略的示意图。图 13.34 是原乐谱与系统记谱的比较。穆勒的自动音乐记谱工作很快由皮什扎尔斯基(Piszczalski)和加勒(Galler)(1977)跟进进行研究。

另一个信号解读的有限事例是对诸如相位声码器(参见先前有关相位声码器的讨论)这样的系统产生的"爆炸性信息"的解读。相位声码器产生的数据流(每个分析通道的振幅和频率包络)可以比原始信号占用很多倍的存储空间,手动地编辑和解释这些数据是单调乏味的。我们可以应用通过模式识别方法进行数据压缩的算法以便让使用者以简化的形式操纵这些数据,而不明显丧失保真度(Strawn 1980,1985b)。为完成这样的任务,系统必须懂得包络的哪些特征对人类听觉来说是重要的,以及哪些不重要。

20 世纪 80 年代,在斯坦福大学(Standford University),另一个自动音乐记谱系统被开发出来(Chowning et al. 1984, Chowning and Mont-Reynaud 1986)。系统分析录制的音乐(主要来自 18 世纪的旋律)并且试图以该时代典型方式进行自动记谱。演奏背离了原始乐谱,而记谱系统的目标是恢复乐谱原貌,不是再现真实的演奏。这需要低级别的分析本领和对 18 世纪记谱习惯知识的了解。低级别和高级别的综合运作是信号解读系统的特征。

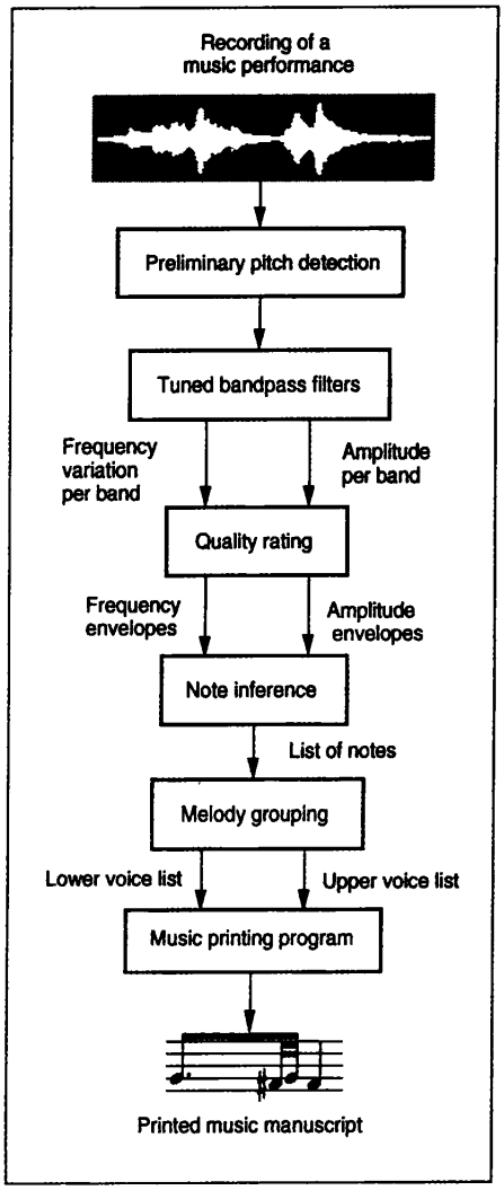


图 13.33　詹姆士·穆勒(James A. Moorer)开发的音乐自动记谱策略。
Recording of a music performance=一段音乐演奏录音
Preliminary pitch detection=音高预侦测
Tuned bandpass filters=受调谐带通滤波器
Frequency variation per band=每带频率变化
Amplitude per band=每带振幅
Quality rating=质量评价
Frequency envelopes=频率包络
Amplitude envelopes=振幅包络
Note inference=音符推论
List of notes=音符列表
Melody grouping=旋律分组
Lower voice list=低声部列表
Upper voice list=高声部列表
Music printing program=音乐打印程序
Printed music manuscript=打印的音乐原稿

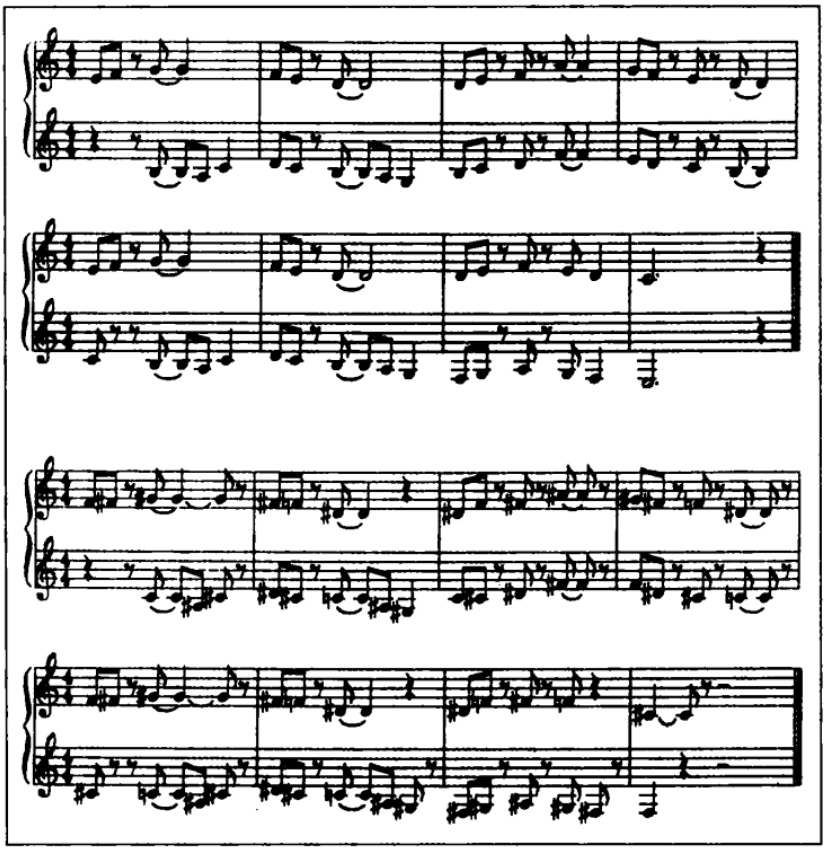


图 13.34 原始乐谱与摩尔系统完成的声学演奏记谱之比较。长音符的时值被低估了，倒数第二小节丢失了一个音符。最明显的改变是由于吉他被错定高了一个小二度。照本宣科的计算机从头至尾忠实地汇报出乐谱走高了一个小二度。

WABOT-2(图 13.35)是信号解读系统的一个生动示例，一个由早稻田大学(Waseda University)(东京)大批师生建造，后被住友商事(Sumitomo Corporation)在日本重新实施的机器人(Matsushima et at. 1985, Roads 1986b)。在 1985 和 1986 年筑波世博会(Tsukuba World Expo)期间向上百万参观者展览了该机器人。WABOT-2 能够理解语音信号、音乐信号以及视觉乐谱。它可以对用日文说出的点歌要求作出回应，且可以看乐谱。记住放在计算机眼面前的乐谱的同时，WABOT-2 计划着它的表演。它也可以为一个唱歌的人伴奏。如果歌唱者走调或不稳，机器人会对风琴音高和伴奏节奏作出调整以便试图配合歌唱者(Roads 1986b)。

图 13.35　WABOT-2，一个开发于 1985 年，日本早稻田大学（Waseda University）的音乐机器人，进一步工程化由住友商事（Sumitomo Corporation）完成。该机器人可以听懂语音要求（日语）且可以读乐谱，使用风琴给歌唱者伴奏。机器人可以跟随歌唱者的表演（音高及速度），并且调整其自身的演奏以适应歌唱者。

结论(Conclusion)

第 12 和第 13 章中所讨论方法的多样性表明每种声音分析方法都有其各自的长处及弱点。表 13.2 是一个建议表，根据音乐目的检索的与之相匹配的分析方法。

表 13.2　频谱分析的应用

特征检索	可能的分析方法
共振峰结构	宽频道带宽的声谱图（300Hz 左右）、线性预测编码
总体时变音乐频谱	窄带宽 FFT 声谱图（<50Hz）、三维投射的“瀑布图”
事件触发时间（分割）	振幅阈限、高通滤除、自回归分割、小波相位图
刮擦声，暂态声	小波分析投射的相位图中的突变性
将谐波部分与信号中噪声部分分离	梳状小波变换、频谱建模合成
特定音程事例	栅格与检索音程对齐的小波分析
音高侦测	基频周期法、自相关、自适应滤波器法、频率维度的方法、音言谱（对数倒频谱）法
频谱感知	基于临界带宽的恒定 Q 值分析、相关图、其他听觉模型
相位失真	维格纳分布（Wigner distribution）

声音分析的音乐蕴涵目前显然受到欢迎。对于音乐家来说声音分析意味着两件事：首先，我们测量了乐器，其让我们检视声音的细微结构。这使得将有兴趣的成分（振幅包络、激励和共振信号、音高、节奏和频谱）分离出来以及按照意愿重新编排，对其实施显微外科手术成为可能。其次，音乐机器有了耳朵。其可以被编程来对演奏模式和暗示线索作出响应，或以音乐的方式对采样的声学信号进行处理。

频谱估算是一个迅速发展的领域；可以说其还处在原始成长期，至今很少系统能够综合几种技术来适应被测量信号或以不同视角测量同一信号。倘若输入参数被仔细设定的话，标准技术可以恰当地发挥作用。但目前很多技术从零开始，与人类听者不同，它们不能把先前收集到的知识应用到后续分析中。未来的系统或许在这方面会更智能化。

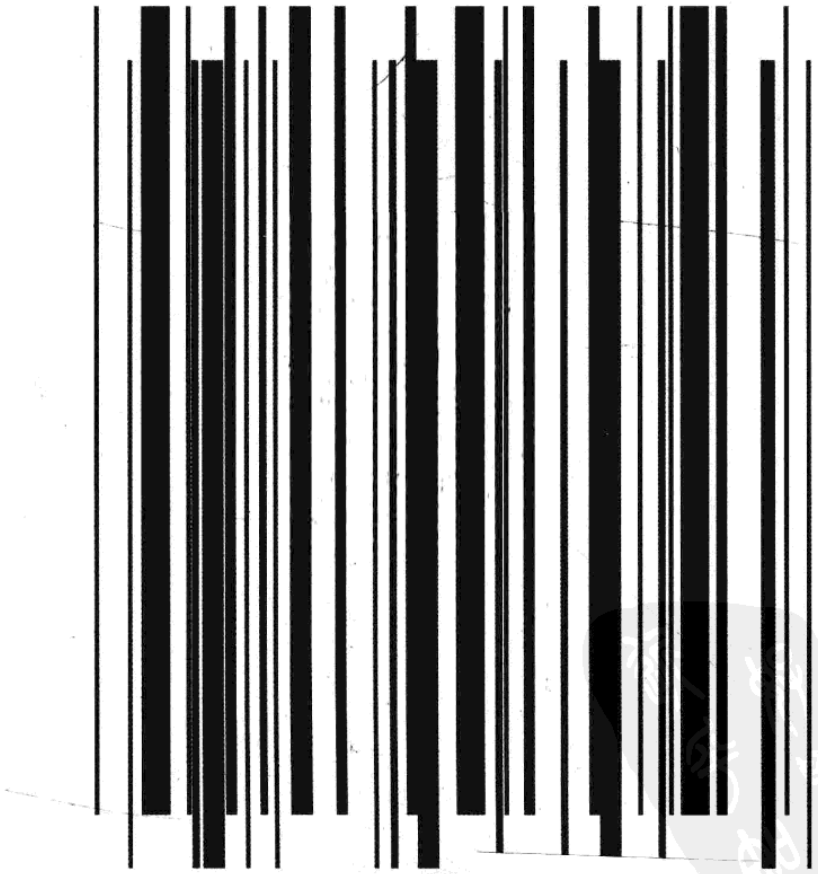
创造敏感的音乐分析工具从来都不是容易的。例如，如何告诉计算机在保留钢琴家的演奏榔头声和呼吸声的同时，将“噪声”从差的钢琴音乐录音中移除？在处理“降噪”过程中我们马上会面临欣赏趣味问题。比起经降噪、解卷积、再合成和人工混响重现的消毒般的质量，有些人更喜欢旧时代的声音以及老录音的真实性。一个“新的改良的”赝品不一定是美学道路上的进步。

贯穿本章的主题是频谱分析方法的多样性。在选择一个分析方法之前，我们必须知道我们期待的是哪种测量。为合理地设置分析参数，最好预先知道被分析的是哪种类型的声音。各分析方法都建立在其背后的声音制作模型的基础之上。因此在阐释一个分析时我们必须总是将该模型的局限性考虑在内。

The Computer Music Tutorial
Curtis Roads

〔美〕柯蒂斯·罗兹等著

计算机音乐教程 上册



人民音乐出版社
PEOPLE'S MUSIC PUBLISHING HOUSE

The Computer Music Tutorial

Curtis Roads

[美] 柯蒂斯·罗兹著

计算机音乐教程

上册

定价：182.00元
（上、下册）
ISBN 978-7-103-03702-7
9 787103 037027 >

中译本译著委员会名单

主　　任　肯尼斯·费尔兹 (Kenneth Fields)

专家小组　张小夫　杜晓十　刘　健　吴粤北

韩宝强　黄枕宇　金　平

审　　订　齐　刚　李斯心

译　　者　李斯心　李岳凌　齐　刚　陈　泱

姜　浩　杨仁瑛　张睿博　程伊兵

胡　泽　黄志鹏　常　炜

译文统筹　金　平

特约编辑　张志羽

责任编辑　任　云

中译本序

我们知道，20世纪下半叶在音乐领域发生的最重要的事件之一就是电子音乐的兴起。作为音乐艺术与科学技术碰撞的结晶，电子音乐已经先后经历了初创阶段的“具体音乐”、早期发展阶段的“磁带音乐”、中期发展阶段的“电子声学音乐”和整体数字化的“计算机音乐”四个不同发展阶段的演进，完成了从“模拟技术”到“数字技术”划时代的过渡与整合，形成了电子音乐在当代“专业化”、“社会化”和“个人化”三个不同层面多元发展的格局，迎来了半个多世纪以来最辉煌的发展时期。

半个多世纪以来，科学技术一日千里的进步为电子音乐的发展提供了广阔的空间，国际电子音乐学界因之诞生并积累了一批高水准的学术论著。电子音乐大师柯蒂斯·罗兹(Curtis Roads)先生所著《计算机音乐教程》堪称其中的佼佼者。此书中译本的问世，对于正在蓬勃发展而教材建设则刚刚起步的中国电子音乐学科而言可谓是雪中送炭。本书在北京的出版也是2009年中国电子音乐学界的一件大事。

《计算机音乐教程》是柯蒂斯·罗兹先生历经十余年完成的一部完整、系统地论述计算机音乐的名著。它涵盖了计算机音乐的各个方面——包括数字音频、合成技术、信号处理、声音分析、算法作曲、数字乐器接口和心理声学等，同时也对计算机音乐的诸多术语和概念做了详尽的说明和阐释。该书内容丰富，层次清晰，理论价值高，是国际电子音乐学界公认的权威著作。此书中译本的诞生不仅会对中国的电子音乐、计算机音乐基础理论建设和学术研究产生重要的影响，同时也定将对中国各音乐艺术院校飞速发展的电子音乐、计算机音乐教学工作和学科建设发挥不可替代的推动作用。

2006年北京国际电子音乐节的主题是“语言”，它在学科建设上的意义是既现实而又影响深远的。毋庸讳言，当前我国电子音乐学界对众多电子音乐学术概念，以及大量电子音乐术语的理解和运用还存在着言人人殊的现象，甚至存在某些误区；而我们将权威名著《计算机音乐教程》的英文原版翻译成中文，不啻是在中国电子音乐学科领域梳理、统一学术概念和音乐术语这一艰巨工程的先导。

在该教程中文译本问世之际，我们首先要感谢人民音乐出版社高瞻远瞩的学术眼光与不单纯计较经济效益的果断投入。本书的翻译和定稿工作凝聚了国内众多专家学者呕心沥血的付出与满怀历史责任感的努力，这是尤其值得我们铭记的。我们中国有“前人栽树，后人乘凉”的说法，本书中译本的翻译出版正是这种甘心“为人作嫁衣”的宝贵精神的体现。开拓者们的艰辛劳动和丰硕成果让我们肃然起敬。

我很高兴将此书推荐为中央音乐学院中国现代电子音乐中心的指定教材，也愿意推荐给我们中国电子音乐学界的各位同行们。

中央音乐学院电子音乐中心主任，
中国电子音乐学会会长
张小夫

序:新音乐与科学
(Foreword:New Music and Science)

随着计算机和数字设备的应用,音乐的创造过程与社会科技资源的紧密连结已到了前所未有的程度。出于创作的需要,作曲家通过在声音生成、声音处理以及从微观到宏观形式的各个层面的创作中广泛应用计算机,从而促使科学与作曲这两个领域彼此依赖而密不可分。科学与技术极大地丰富了当代音乐,反之亦然,即具有特殊的音乐价值的课题有时也会直接提出或引发科学技术方面的课题。音乐与科学有着各自发展的动力,却又相互依靠,由此显示出一种独特的互惠关系。

科技施用于音乐并非新奇之事,然而计算机系统的迅猛发展已经使音乐发展到一个新的水平。现代计算机系统的内涵已远远超出其作为物理机器的固有性质。"计算"的特性之一就是可编程性和由此涉及的程序语言。高级程序设计语言代表了几个世纪以来人类关于思维方式的思想,它是各门学科对计算机加以利用的手段。

程序设计所伴随的心智过程以及对细节的严密把握与音乐创作颇有异曲同工之妙,难怪作曲家们成了实际使用计算机最早的艺术家。有那么一些动机促使人们把某些必要的科学知识和概念同音乐的意识熔于一炉,并在看似音乐以外的领域获得新的能力。在这些动机之中有两个一直很有召唤力:(1)计算机合成声音的通行法则;(2)涉及音乐结构和作曲过程的编程能力。

声音合成(Sound Synthesis)

尽管传统乐器确实已经构成了一个丰富的声音空间,但是几十年来作曲家的听觉想象力一直在魔法般地召唤各种声音,这些声音建立在对自然声的改换和推想的基础之上,无法用原声乐器或电子模拟乐器来实现。由计算机控制的扬声器是现存最通用的合成工具。任何声音,从最简单的到最复杂的,凡是通过扬声器能发出的,都可以借此工具合成出来。计算机声音合成是人类听觉想

象与听觉真实之间的一座桥梁。它意味着作曲家会拥有更为广阔的声音空间，这对作曲家们显然具有很大的吸引力。

虽然，由工具引起的对于创造声音的制约已经消除，但是仍然存在一个巨大的障碍，这就是作曲家缺乏有效驾驭计算机进行声音合成的必要知识，对此，他们必须加以克服方能充分利用计算机。从某种程度上来说，与计算机相关的技术知识相对容易获取，而涉及最多的还是声音的物理描述以及人对声音的感知。说来有些奇怪，这些必需的知识绝大多数并不存在于科学查询时人们特别预期的那些领域之中，例如物理声学与心理生物学，这些学科并不能为作曲家提供终须涉及的细节层面上的精确内容或数据。在过去，科学的数据和结论被尝试用于复制自然声，并将其作为获取声音信息的一般方式。但是，音乐家与音乐科学家很快就发现这些数据和结论大部分都不足以为用。要做声音合成，即使是接近最简单的自然声的声音合成，也要求必须具备有关声音各种要素之间瞬间演变的详细知识。

尽管如此，物理学、心理学、计算机科学以及数学这些学科为我们提供了强大的工具和概念。当这些概念同音乐知识以及敏感的听觉相结合的时候，音乐家、科学家和技术人员就有机会合力一处，在细节层次上共同创造声音的各种新概念和挖掘声音的物理学与心理物理学性质，这对作曲家是有益的，可以满足其听觉与想象力的苛求。

正如本书所展现的，一些成果已经问世：对音色更加深刻的理解为作曲家在工作中提供了色彩大为丰富的声音调色板；新的有效合成技术被发现和开发出来，它们是基于人对声音的感知特征模型，而非声音的物理特征模型；用于对合成声与(或)数码录制声进行编辑和混音的强大软件也已被开发出来；一些知觉融合(perceptual fusion)的试验已经发展到声源识别与听象(auditory image)这样一类新颖而有益于音乐的研究领域；最后，专用的计算机合成器正在设计构造之中。这些实时演奏系统融合了许多知识与技术上的进步。

编程与作曲(Programming and Composition)

由于计算机程序语言在设计上的一个基本假设便是通用性，因此任何已知高级语言的实际应用范围都极广，这其中显然也包括音乐。已经出现了许多程序，它们适应不同的音乐用途并且以不同的语言编写而成。这些程序中最有用，同时也是作曲家体验最多的就是声音合成与声音处理程序，以及合成程序所需的能够把一段乐曲的音乐描述转化为物理描述的程序。

掌握一定程度的编程能力对作曲家大有裨益，因为这是全面了解计算机系

统的关键。虽然计算机系统由许多极其复杂的程序构成,并以非专家难以理解的技术写成,但是,为了有效使用计算机,具备编程能力可以使作曲家理解一个系统的整体运行。编程能力也赋予了作曲家在处理层上的某种自主性,而在处理层——合成这里的自主恰恰是作曲家最为渴求的。与传统管弦乐曲创作的情形类似,作曲家在跟音色和微联接(microarticulation)有关的乐音合成中所做的选择也往往是非常主观的。这个选择的过程由于作曲家获得了自由改变声音合成算法的能力而大为强化。

音乐结构的编程是程序设计能力所能提供的另一个机会。如果乐曲的构造过程能够在一个大致明确的意义上公式化,那么在此范围内,音乐就可能以程序的形式实现。例如,建立在某些循环过程上的音乐结构,就可借助编程得以适当的实现。

但是,作曲家接触程序设计语言的概念,对于他们的编程能力很少有切实的影响。尽管一个程序要实现的功能会影响到选择编写这个程序所使用的语言种类,同样真实的是,一种程序语言也能影响到一个程序的功能设计观念。在更广的意义上,程序设计观念能够为编程环境以外的人提出一些意想不到的功能,这在音乐创作中是极其重要的,因为程序设计观念与音乐听觉想象力的结合使想象力自身的疆域得以延伸。也就是说,程序语言并不仅仅是一种用来完成预想任务或作用的工具,它还是一个广阔的结构平台,想象力可在此与之交相作用。

包含了物理学和心理物理学概念的计算机声音合成虽说是得自于对自然声的分析,而当它与音乐结构的高级编程相结合的时候,其含义就远远超越了自然声音色。与传统乐器的振动模式基本不受作曲影响的情况不同,计算机合成允许作曲家对音乐的微结构进行创作。

另外,在计算机处理的背景下,音乐的微结构不一定采取预定的形式——这与具体乐器的特定发音方式(演奏法)有关。确切地说,微结构可以服从于作品整体一致的思维方法,并像作品的其他方面一样,作曲家可以在自己的想象力中自由地决定微结构。

约翰·乔宁(John Chowning)

前　言(Preface)

音乐是变化的:新的音乐形式层出不穷,音乐家对音乐的重新诠释向旧音乐类型不断注入新的活力。各种音乐文化的浪潮前赴后继,散播着新风格的共鸣。音乐演奏与作曲的技术亦蜿蜒其中。音乐生产领域被不断地再开发,与其一同跃进的是音乐技术也在持续演变。每一种音乐都有一个乐器的家族与之相随,所以,即使仅限于原声乐器,我们现在也会有几百种乐器可供挑选。

在20世纪,电子学把乐器设计这条小溪变成了沸腾的激流。电气(声)化把吉他、低音提琴、钢琴、管风琴和鼓变成了工业社会的民间乐器。模拟合成器扩张了乐音的调色板,并由此发动了一轮对声音素材的实验。但是,模拟合成器由于缺乏可编程能力、精确性、记忆力和理解力,因此有其局限性。而凭借上述这些能力,数字计算机提供了远为丰富的功能来处理声音的色彩。它会聆听、分析,并老练地对音乐表情动作做出反应。它能让音乐家们编辑音乐或根据逻辑规则来作曲,然后把结果打印成乐谱。计算机还可进行人机互动教学,利用声音和图像表现音乐的每个方面。在计算机音乐的专业研究之外,各种新的音乐应用将会自行不断地发展下去。

与正在发生的变化随之而来的,是音乐家面临的挑战,即如何理解计算机这一媒介的发展潜能,并跟上它的最新发展。《计算机音乐教程》的应运而生,是为了满足对一本标准、全面的计算机音乐理论与实践基础知识课本的需求。作为对参考书籍《计算机音乐基础》(*Foundations of Computer Music*,MIT Press,1985)和《音乐机器》(*The Music Machine*,MIT Press,1989)的补充,本书提供了计算机音乐领域高级研究的必要背景。《计算机音乐基础》和《音乐机器》两本书都是文集,而这本《计算机音乐教程》包括了所有可用于教学的新材料。

受众群(Intended Audience)

本书不仅是为音乐专业的学生,也是为以计算机音乐为研究方向的工程师

和科学家编撰的。本书的很多部分打开了技术的“黑匣子”，揭示了软件和硬件的内部运行机制。为什么这些技术信息与音乐家有关呢？我们的目标是让音乐家更好地掌握和使用音乐技术，而不是要把他们变成工程师。技术上无知的音乐家有时候对这个快速进化的工具的潜在可能性抱有过于狭隘的观念，他们可能还在受过去年代某些过时观念的制约。因为缺乏基础知识，他们在盲目实践中浪费时间，不知道如何将想法变成实用的成果。因此，本书的目的之一就是给予那些想要最终建立和经营私人或公共机构计算机音乐工作室的众多音乐家以相关的知识，使他们在这一领域能有独立的判断力。

对于一些音乐家来说，本书可以作为专业技术研究的一个向导。他们中的一些人将以新的技术进步推动计算机音乐领域的发展。这一点理应不会引起跟随着这个领域发展的任何人的惊奇。历史一再证明，一些音乐技术方面最重要的进步是由技术上见识广博的音乐家构想出来的。

跨学科精神(Interdisciplinary Spirit)

计算机音乐的知识基础来自于以下多个学科领域：作曲、声学、心理声学、物理学、信号处理、声音合成、音乐演奏、计算机科学以及电子工程。因此，一个成熟的计算机音乐教学法必须体现各学科间的合作精神。在本书中，音乐上的应用旨在体现对技术观念的表达，而技术步骤的讨论则穿插着对其音乐性价值的评述。

继承(Heritage)

我们工作的目标之一始终是传达一种计算机音乐的继承意识。概述(Overview)和背景(Background)部分将现时图景置于历史的背景之下。大量的参考文献为读者继续研究提供了源头，同时强调了这些被引用的概念背后的先行者们。

概念和术语(Concepts and Terms)

每一种音乐设备和软件包都采用一套不同的规程——专用术语、标记系统、指令语法和界面分布等。这些不同的规程均建立在本书所解释的诸多基础

中译本序

我们知道，20世纪下半叶在音乐领域发生的最重要的事件之一就是电子音乐的兴起。作为音乐艺术与科学技术碰撞的结晶，电子音乐已经先后经历了初创阶段的“具体音乐”、早期发展阶段的“磁带音乐”、中期发展阶段的“电子声学音乐”和整体数字化的“计算机音乐”四个不同发展阶段的演进，完成了从“模拟技术”到“数字技术”划时代的过渡与整合，形成了电子音乐在当代“专业化”、“社会化”和“个人化”三个不同层面多元发展的格局，迎来了半个多世纪以来最辉煌的发展时期。

半个多世纪以来，科学技术一日千里的进步为电子音乐的发展提供了广阔的空间，国际电子音乐学界因之诞生并积累了一批高水准的学术论著。电子音乐大师柯蒂斯·罗兹(Curtis Roads)先生所著《计算机音乐教程》堪称其中的佼佼者。此书中译本的问世，对于正在蓬勃发展而教材建设则刚刚起步的中国电子音乐学科而言可谓是雪中送炭。本书在北京的出版也是2009年中国电子音乐学界的一件大事。

《计算机音乐教程》是柯蒂斯·罗兹先生历经十余年完成的一部完整、系统地论述计算机音乐的名著。它涵盖了计算机音乐的各个方面——包括数字音频、合成技术、信号处理、声音分析、算法作曲、数字乐器接口和心理声学等，同时也对计算机音乐的诸多术语和概念做了详尽的说明和阐释。该书内容丰富，层次清晰，理论价值高，是国际电子音乐学界公认的权威著作。此书中译本的诞生不仅会对中国的电子音乐、计算机音乐基础理论建设和学术研究产生重要的影响，同时也定将对中国各音乐艺术院校飞速发展的电子音乐、计算机音乐教学工作和学科建设发挥不可替代的推动作用。

2006年北京国际电子音乐节的主题是“语言”，它在学科建设上的意义是既现实而又影响深远的。毋庸讳言，当前我国电子音乐学界对众多电子音乐学术概念，以及大量电子音乐术语的理解和运用还存在着言人人殊的现象，甚至存在某些误区；而我们将权威名著《计算机音乐教程》的英文原版翻译成中文，不啻是在中国电子音乐学科领域梳理、统一学术概念和音乐术语这一艰巨工程的先导。

在该教程中文译本问世之际，我们首先要感谢人民音乐出版社高瞻远瞩的学术眼光与不单纯计较经济效益的果断投入。本书的翻译和定稿工作凝聚了国内众多专家学者呕心沥血的付出与满怀历史责任感的努力，这是尤其值得我们铭记的。我们中国有“前人栽树，后人乘凉”的说法，本书中译本的翻译出版正是这种甘心“为人作嫁衣”的宝贵精神的体现。开拓者们的艰辛劳动和丰硕成果让我们肃然起敬。

我很高兴将此书推荐为中央音乐学院中国现代电子音乐中心的指定教材，也愿意推荐给我们中国电子音乐学界的各位同行们。

中央音乐学院电子音乐中心主任，
中国电子音乐学会会长
张小夫

序:新音乐与科学
(Foreword:New Music and Science)

随着计算机和数字设备的应用,音乐的创造过程与社会科技资源的紧密连结已到了前所未有的程度。出于创作的需要,作曲家通过在声音生成、声音处理以及从微观到宏观形式的各个层面的创作中广泛应用计算机,从而促使科学与作曲这两个领域彼此依赖而密不可分。科学与技术极大地丰富了当代音乐,反之亦然,即具有特殊的音乐价值的课题有时也会直接提出或引发科学技术方面的课题。音乐与科学有着各自发展的动力,却又相互依靠,由此显示出一种独特的互惠关系。

科技施用于音乐并非新奇之事,然而计算机系统的迅猛发展已经使音乐发展到一个新的水平。现代计算机系统的内涵已远远超出其作为物理机器的固有性质。"计算"的特性之一就是可编程性和由此涉及的程序语言。高级程序设计语言代表了几个世纪以来人类关于思维方式的思想,它是各门学科对计算机加以利用的手段。

程序设计所伴随的心智过程以及对细节的严密把握与音乐创作颇有异曲同工之妙,难怪作曲家们成了实际使用计算机最早的艺术家。有那么一些动机促使人们把某些必要的科学知识和概念同音乐的意识熔于一炉,并在看似音乐以外的领域获得新的能力。在这些动机之中有两个一直很有召唤力:(1)计算机合成声音的通行法则;(2)涉及音乐结构和作曲过程的编程能力。

声音合成(Sound Synthesis)

尽管传统乐器确实已经构成了一个丰富的声音空间,但是几十年来作曲家的听觉想象力一直在魔法般地召唤各种声音,这些声音建立在对自然声的改换和推想的基础之上,无法用原声乐器或电子模拟乐器来实现。由计算机控制的扬声器是现存最通用的合成工具。任何声音,从最简单的到最复杂的,凡是通过扬声器能发出的,都可以借此工具合成出来。计算机声音合成是人类听觉想

象与听觉真实之间的一座桥梁。它意味着作曲家会拥有更为广阔的声音空间，这对作曲家们显然具有很大的吸引力。

虽然，由工具引起的对于创造声音的制约已经消除，但是仍然存在一个巨大的障碍，这就是作曲家缺乏有效驾驭计算机进行声音合成的必要知识，对此，他们必须加以克服方能充分利用计算机。从某种程度上来说，与计算机相关的技术知识相对容易获取，而涉及最多的还是声音的物理描述以及人对声音的感知。说来有些奇怪，这些必需的知识绝大多数并不存在于科学查询时人们特别预期的那些领域之中，例如物理声学与心理生物学，这些学科并不能为作曲家提供终须涉及的细节层面上的精确内容或数据。在过去，科学的数据和结论被尝试用于复制自然声，并将其作为获取声音信息的一般方式。但是，音乐家与音乐科学家很快就发现这些数据和结论大部分都不足以为用。要做声音合成，即使是接近最简单的自然声的声音合成，也要求必须具备有关声音各种要素之间瞬间演变的详细知识。

尽管如此，物理学、心理学、计算机科学以及数学这些学科为我们提供了强大的工具和概念。当这些概念同音乐知识以及敏感的听觉相结合的时候，音乐家、科学家和技术人员就有机会合力一处，在细节层次上共同创造声音的各种新概念和挖掘声音的物理学与心理物理学性质，这对作曲家是有益的，可以满足其听觉与想象力的苛求。

正如本书所展现的，一些成果已经问世：对音色更加深刻的理解为作曲家在工作中提供了色彩大为丰富的声音调色板；新的有效合成技术被发现和开发出来，它们是基于人对声音的感知特征模型，而非声音的物理特征模型；用于对合成声与(或)数码录制声进行编辑和混音的强大软件也已被开发出来；一些知觉融合(perceptual fusion)的试验已经发展到声源识别与听象(auditory image)这样一类新颖而有益于音乐的研究领域；最后，专用的计算机合成器正在设计构造之中。这些实时演奏系统融合了许多知识与技术上的进步。

编程与作曲(Programming and Composition)

由于计算机程序语言在设计上的一个基本假设便是通用性，因此任何已知高级语言的实际应用范围都极广，这其中显然也包括音乐。已经出现了许多程序，它们适应不同的音乐用途并且以不同的语言编写而成。这些程序中最有用，同时也是作曲家体验最多的就是声音合成与声音处理程序，以及合成程序所需的能够把一段乐曲的音乐描述转化为物理描述的程序。

掌握一定程度的编程能力对作曲家大有裨益，因为这是全面了解计算机系

统的关键。虽然计算机系统由许多极其复杂的程序构成,并以非专家难以理解的技术写成,但是,为了有效使用计算机,具备编程能力可以使作曲家理解一个系统的整体运行。编程能力也赋予了作曲家在处理层上的某种自主性,而在处理层——合成这里的自主恰恰是作曲家最为渴求的。与传统管弦乐曲创作的情形类似,作曲家在跟音色和微联接(microarticulation)有关的乐音合成中所做的选择也往往是非常主观的。这个选择的过程由于作曲家获得了自由改变声音合成算法的能力而大为强化。

音乐结构的编程是程序设计能力所能提供的另一个机会。如果乐曲的构造过程能够在一个大致明确的意义上公式化,那么在此范围内,音乐就可能以程序的形式实现。例如,建立在某些循环过程上的音乐结构,就可借助编程得以适当的实现。

但是,作曲家接触程序设计语言的概念,对于他们的编程能力很少有切实的影响。尽管一个程序要实现的功能会影响到选择编写这个程序所使用的语言种类,同样真实的是,一种程序语言也能影响到一个程序的功能设计观念。在更广的意义上,程序设计观念能够为编程环境以外的人提出一些意想不到的功能,这在音乐创作中是极其重要的,因为程序设计观念与音乐听觉想象力的结合使想象力自身的疆域得以延伸。也就是说,程序语言并不仅仅是一种用来完成预想任务或作用的工具,它还是一个广阔的结构平台,想象力可在此与之交相作用。

包含了物理学和心理物理学概念的计算机声音合成虽说是得自于对自然声的分析,而当它与音乐结构的高级编程相结合的时候,其含义就远远超越了自然声音色。与传统乐器的振动模式基本不受作曲影响的情况不同,计算机合成允许作曲家对音乐的微结构进行创作。

另外,在计算机处理的背景下,音乐的微结构不一定采取预定的形式——这与具体乐器的特定发音方式(演奏法)有关。确切地说,微结构可以服从于作品整体一致的思维方法,并像作品的其他方面一样,作曲家可以在自己的想象力中自由地决定微结构。

约翰·乔宁(John Chowning)

前　言(Preface)

音乐是变化的:新的音乐形式层出不穷,音乐家对音乐的重新诠释向旧音乐类型不断注入新的活力。各种音乐文化的浪潮前赴后继,散播着新风格的共鸣。音乐演奏与作曲的技术亦蜿蜒其中。音乐生产领域被不断地再开发,与其一同跃进的是音乐技术也在持续演变。每一种音乐都有一个乐器的家族与之相随,所以,即使仅限于原声乐器,我们现在也会有几百种乐器可供挑选。

在20世纪,电子学把乐器设计这条小溪变成了沸腾的激流。电气(声)化把吉他、低音提琴、钢琴、管风琴和鼓变成了工业社会的民间乐器。模拟合成器扩张了乐音的调色板,并由此发动了一轮对声音素材的实验。但是,模拟合成器由于缺乏可编程能力、精确性、记忆力和理解力,因此有其局限性。而凭借上述这些能力,数字计算机提供了远为丰富的功能来处理声音的色彩。它会聆听、分析,并老练地对音乐表情动作做出反应。它能让音乐家们编辑音乐或根据逻辑规则来作曲,然后把结果打印成乐谱。计算机还可进行人机互动教学,利用声音和图像表现音乐的每个方面。在计算机音乐的专业研究之外,各种新的音乐应用将会自行不断地发展下去。

与正在发生的变化随之而来的,是音乐家面临的挑战,即如何理解计算机这一媒介的发展潜能,并跟上它的最新发展。《计算机音乐教程》的应运而生,是为了满足对一本标准、全面的计算机音乐理论与实践基础知识课本的需求。作为对参考书籍《计算机音乐基础》(*Foundations of Computer Music*, MIT Press, 1985)和《音乐机器》(*The Music Machine*, MIT Press, 1989)的补充,本书提供了计算机音乐领域高级研究的必要背景。《计算机音乐基础》和《音乐机器》两本书都是文集,而这本《计算机音乐教程》包括了所有可用于教学的新材料。

受众群(Intended Audience)

本书不仅是为音乐专业的学生,也是为以计算机音乐为研究方向的工程师

和科学家编撰的。本书的很多部分打开了技术的“黑匣子”，揭示了软件和硬件的内部运行机制。为什么这些技术信息与音乐家有关呢？我们的目标是让音乐家更好地掌握和使用音乐技术，而不是要把他们变成工程师。技术上无知的音乐家有时候对这个快速进化的工具的潜在可能性抱有过于狭隘的观念，他们可能还在受过去年代某些过时观念的制约。因为缺乏基础知识，他们在盲目实践中浪费时间，不知道如何将想法变成实用的成果。因此，本书的目的之一就是给予那些想要最终建立和经营私人或公共机构计算机音乐工作室的众多音乐家以相关的知识，使他们在这一领域能有独立的判断力。

对于一些音乐家来说，本书可以作为专业技术研究的一个向导。他们中的一些人将以新的技术进步推动计算机音乐领域的发展。这一点理应不会引起跟随着这个领域发展的任何人的惊奇。历史一再证明，一些音乐技术方面最重要的进步是由技术上见识广博的音乐家构想出来的。

跨学科精神(Interdisciplinary Spirit)

计算机音乐的知识基础来自于以下多个学科领域：作曲、声学、心理声学、物理学、信号处理、声音合成、音乐演奏、计算机科学以及电子工程。因此，一个成熟的计算机音乐教学法必须体现各学科间的合作精神。在本书中，音乐上的应用旨在体现对技术观念的表达，而技术步骤的讨论则穿插着对其音乐性价值的评述。

继承(Heritage)

我们工作的目标之一始终是传达一种计算机音乐的继承意识。概述(Overview)和背景(Background)部分将现时图景置于历史的背景之下。大量的参考文献为读者继续研究提供了源头，同时强调了这些被引用的概念背后的先行者们。

概念和术语(Concepts and Terms)

每一种音乐设备和软件包都采用一套不同的规程——专用术语、标记系统、指令语法和界面分布等。这些不同的规程均建立在本书所解释的诸多基础

概念之上。面对大量不相兼容而又不断变化的技术环境，对于一本教科书来说，传授基础概念比起详细讲解一个指定语言的特性、软件的运用或者合成器来似乎更为适当。因此，本书并不打算教读者如何操作某一设备或者软件——因为那是每个系统所提供的文档的目标。不过，本书将使读者的此类学习变得轻松许多。

用本书教学(Use of This Book in Teaching)

《计算机音乐教程》是作为一本综合性教材而编写的，旨在提出一种国际视野下的平衡的观点。本书设计定位为核心教材，同时应该可以容易适应多种教学情况。在理想的条件下，本书应作为学生的指定读本，它结合学生们所在的音乐工作室环境，在这样的环境中，学生有充裕的时间检验本书中的各种观点。每一个工作室都有其偏爱的设备(计算机、软件、合成器等)，那么，与这些设备相关的工具手册、工作室的实用指导会和本书一道，共同达到良好的教学效果。

导读(Roadmap)

《计算机音乐教程》分为七个部分，每一部分包括了若干章节。第一部分基础概念，介绍数字音频和计算机技术。熟练掌握这几个章节可以对理解本书以后的内容有所帮助。

第二部分集中在数字声音合成。第 3 章到第 8 章介绍了主要的合成方法，包括对实验性和商用性都有效的方法。

第三部分混音和信号处理，包括四章内容，让这些有时令人难以捉摸的主题不再显得神秘，包括混音、滤波、延迟效应、混响和空间操控。

第四部分的主题是声音分析，占有支配地位，它是很多音乐应用如声音转化、交互式演奏与音乐录制的关键，包括音高、节奏和频谱的计算机分析。

第五部分主要介绍计算机音乐系统有关音乐家界面的重点主题。其中，第 14 章的主题是演奏者可操控的物理设备，第 15 章研究了解释演奏者姿态的软件。第 16 章是对音乐编辑系统的一个概观。音乐语言是第 17 章的主题。第五部分的最后两章(第 18、19 章)介绍了算法作曲(Algorithmic composition)体系的方法及表示法。

第六部分解密计算机音乐系统，第 20 章考察数字信号处理器的内部构建，第 21 章讨论流行的 MIDI 接口协议，最后，第 22 章介绍了计算机、输入设备及

数字信号处理硬件之间的互连。

第七部分是由约翰·戈登(John Gordon)撰写的一个独立章节(第 23 章),介绍心理声学(psychoacoustics),也就是探讨听觉,即人类感知。有关心理声学基本概念的知识在以下几个方面有助于计算机音乐的研究,包括声音设计、混音以及解释信号分析程序的输出。

本书的最后部分是一个技术性附件,为读者介绍傅里叶分析的历史、数学原理及整体设计,特别是快速傅里叶变换,这也是计算机音乐系统普遍使用的工具。

作曲(Composition)

本书虽然涵盖的范围很广,但是不可能把作曲艺术的介绍集中压成一个单独的部分。相反,读者将看到很多对作曲家的引用以及与技术讨论交织在一起的音乐实践内容。第 18、19 章提出了算法作曲背后的技术原则,但这只是一个庞大的(而且事实上是开放的)学科的一个方面,没有必要一定把它当作整体上计算机音乐作曲的代表。

我们在其他的出版物里面对于作曲的实践给予了整体介绍,《作曲家和计算机》(*Composers and the Computer*)特别关注一些音乐家(Roads 1985a)。在我担任《计算机音乐杂志》(*Computer Music Journal*)编辑期间,我们出版了很多有关作品评论、采访和作曲家的文章。其中,包括一本由十四位作曲家参与写作的《作曲论文集》(*Symposium on Composition*)(Roads 1986a)和一份特刊——《计算机音乐杂志》[5 (4) 1981],其中一些文章被再次收入发行很广的《音乐机器》(MIT Press 1989)一书中。特刊 11(1)1987 的特色在于计算机音乐作曲里的微分音。很多其他的期刊和书籍也收入了作曲类刊物中有关电子与计算机音乐的内容丰富的文章。

参考文献与索引(References and Index)

在一本涵盖如此丰富的不同主题的书中,为读者进一步的学习研究提供指针是绝对必要的。本书后面的部分包括了大量的引用和多达一千三百余条的参考文献列表。作为对读者提供的深层服务,我们投入了大量时间以保证人名和主题词索引的广泛详尽。

数学和代码形式(Mathematics and Coding Style)

由于本教材主要为音乐读者而编写,我们用通俗的语言阐述技术理念。本书尽可能少地使用数学符号,这保持了范例的简洁性。数学标记在需要的时候表示为运算符、优先关系符以及明确的指定分组符以保证易读。这是重要的,因为传统数学符号的用法乍看起来有时含义模糊,或者作为算法描述不够完善。同样的原因,本书经常以长变量名称取代那些证明式中偏爱用的单个字符变量。除了一些简单的 Lisp 语言范例,为了可读性的缘故,代码范例是以类 Pascal 语言的伪代码书写的。

附录主要汇集了专业的材料和数学公式,因此,我们在那里又回到了传统的数学符号。

欢迎校正与评论(Corrections and Comments Invited)

作为一本大型图书的第一版,而且是在一个新的领域,某些疏漏恐在所难免。我们欢迎指正和建议,我们也会进一步探寻历史的资料。

致　谢(Acknowledgments)

此书编写历时数年之久。1980 至 1986 年为初稿完成阶段,时任麻省理工学院计算机音乐研究助理和麻省理工学院出版社《计算机音乐杂志》的编辑。初稿之后的修改编撰蒙众多友人鼎立支持才得以完成,借此对他们一并表示感谢。

1988 年受奥尔多·皮奇阿利(Aldo Piccialli)邀请,我前往那不勒斯费德里克二世大学(Università di Napoli Federico Ⅱ)物理系做访问学者,本书第三部分(混音与信号处理)的主要章节和第四部分(声音分析)得以在此期间补充撰写。在此,要特别感谢皮奇阿利教授对第 13 章(频谱分析)和附录(傅里叶分析)所做的详细点评,并感谢他对信号处理理论的慷慨赐教。

在伊万·齐尔品(Ivan Tcherepnin)的推荐下,1989 年我在哈佛大学音乐系执教,其间,从音乐系学习作曲的学生那里反馈回有关第二部分(声音合成)的宝贵意见。另外,还要感谢康拉德·卡明斯(Conrad Cummings)教授和加里·纳尔逊(Gary Nelson)教授提供机会,1990 年我得以在奥柏林音乐学院(Oberlin Conservatory of Music)执教,并将所编内容以讲座形式呈现,这使得我在编写中做到条分缕析。

1991 年,在东京国立音乐大学(Kunitachi College of Music)计算机音乐和音乐技术中心,利用空暇时间,我开始着手于第五部分(音乐家界面)的编写。借此对中心主任科妮莉亚·科利尔(Cornelia Colyer)、国立大学主席海老泽敏(Bin Ebisawa)及日本文化部作曲委员会表示感谢。在杰勒德·佩普(Gerard Pape)、伊恩内斯·克赛纳基斯(Iaanis Xenakis)以及巴黎第八大学(University of Paris Ⅷ)音乐系赫拉西奥·瓦奇万(Horacio Vaggione)教授的帮助下,1993 年和 1994 年我在 UPIC 工作室(Les Ateliers UPIC)对撰写的全部内容首次授课。

约翰·斯特朗(John Strawn)是曾经与我在《计算机音乐杂志》共事的编辑,几年以来对本书撰写做出了巨大贡献。在斯坦福大学攻读博士期间,他主笔了第 1 章和第 3 章的部分内容,之后又以极其认真的态度审阅了大部分章节的文稿。在这场马拉松般的编写过程中,通过电子邮件向约翰详细咨询了无数

个细节问题。在此，特别对其表示感谢，感谢他与我们一起分享他那丰富的音乐知识和专业技术知识，以及敏锐的智慧。

柯蒂斯·阿博特(Curtis Abbott)和约翰·戈登(John Gordon)也欣然参与本书创作，撰写了两章精彩内容，我非常高兴将其归入本书当中。还要感谢麻省理工学院电机工程和计算机科学系的菲利普·格林斯潘(Phillip Greenspun)，他负责执笔了六页内容，建立了附录(傅里叶分析)重要内容的框架并且对初稿进行了仔细的审阅。

另外，许多热心肠人对此书的编写群策群力，无论从信息、文献、照片的提供还是到章节文稿内容的审读。在此，我由衷地向那些提出若干建议、批评指正和为此书做出贡献的人表示感谢，他们是：玛丽-吉恩·阿德里安(Jean-Marie Adrien)、吉姆·艾肯(Jim Aiken)、克拉伦斯·巴洛(Clarence Barlow)、弗朗索瓦·贝勒(François Bayle)、詹姆士·比彻姆(James Beauchamp)、保罗·贝尔格(Paul Berg)、尼古拉·贝尔纳迪尼(Nicola Bernardini)、彼得·贝尔斯(Peter Beyls)、杰克·比斯威尔(Jack Biswell)、汤姆·布吕姆(Thom Blum)、理查德·布朗热(Richard Boulanger)、大卫·布里斯托(David Bristow)、威廉·巴克斯顿(William Buxton)、温迪·卡洛斯(Wendy Carlos)、勒内·科塞(René Caussé)、泽维尔·沙博(Xavier Chabot)、约翰·乔宁(John Chowning)、科妮莉亚·科利尔(Cornelia Colyer)、K. 康克林(K. Conklin)、康拉德·卡明斯(Conrad Cummings)、詹姆士·达索(James Dashow)、菲利普·德帕勒(Philippe Depalle)、马克·多尔森(Mark Dolson)、乔瓦尼·德·波立(Giovanni De Poli)、格哈德·埃克尔(Gerhard Eckel)、威廉·埃尔德里奇(William Eldridge)、吉安保罗·埃万杰利斯塔(Gianpaolo Evangelista)、艾什·弗拉曼-法尔马伊安(Ayshe Framan-Farmaian)、阿德里安·弗里德(Adrian Freed)、克里斯多佛·弗赖依(Christopher Fry)、盖伊·加尼特(Guy Garnett)、约翰·W. 戈登(John W. Gordon)、库尔特·赫布尔(Kurt Hebel)、亨克让·霍宁(Henkjan Honing)、戈特弗里德·迈克尔·凯尼格(Gottfried Michael Koenig)、保罗·兰斯基(Paul Lansky)、奥托·拉斯克(Otto Laske)、大卫·卢因(David Lewin)、D. 加雷斯·洛伊(D. Gareth Loy)、马克斯·V. 马修斯(Max V. Mathews)、斯蒂芬·麦克亚当斯(Stephen McAdams)、丹尼斯·米勒(Dennis Miller)、迭戈· 明恰基(Diego Minciacchi)、伯纳德·蒙-雷诺(Bernard Mont-Reynaud)、罗伯特·穆格(Robert Moog)、穆尔(F. R. Moore)、詹姆士·穆勒(James A. Moorer)、彼德·奈(Peter Nye)、罗伯特·J. 欧文斯(Robert J. Owens)、艾伦·皮弗斯(Alan Peevers)、奥尔多·皮奇阿利(Aldo Piccialli)、斯蒂芬·卜普(Stephen Pope)、爱德华·L. 波林(Edward L. Poulin)、米勒·帕克特(Miller Puckette)、弗朗索瓦·雷韦永(François

Reveillon)、托马斯·雷亚(Thomas Rhea)、让-克劳德·里塞(Jean-Claude Risset)、克雷格·罗兹(Craig Roads)、泽维尔·罗代(Xavier Rodet)、约瑟夫·罗斯坦(Joseph Rothstein)、威廉·朔特施塔德(William Schottstaedt)、玛丽-埃莱娜·塞拉(Marie-Hélène Serra)、约翰·斯内尔(John Snell)、斯汤纳(John Stautner)、莫顿·萨博特尼克(Morton Subotnick)、玛莎·斯威茨奥夫(Martha Swetzoff)、斯坦·腾佩拉斯(Stan Tempelaars)、丹尼尔·泰鲁吉(Daniel Teruggi)、伊雷纳·塔诺斯(Iréne Thanos)、巴里·杜亚士(Barry Truax)、阿尔维斯·维多林(Alvise Vidolin)、迪安·瓦尔拉夫(Dean Wallraff)、大卫·韦克斯曼(David Waxman)、厄尔林·沃尔德(Erling Wold)和伊恩内斯·克赛纳基斯(Iannis Xenakis)。

此外,还要感谢麻省理工学院出版社人员——珍妮特·费希尔(Janet Fisher)经理——《计算机音乐杂志》的出版商。在过去14年中,没有他们的鼎立支持,此书编写近乎不能。

同时,麻省理工学院出版社的主任弗兰克·乌尔班诺夫斯基(Frank Urbanowski)和执行编辑特里·埃林(Terry Ehling)对此书付出了无限的耐心和友善的支持,对此表示衷心感谢。同时,感谢大卫·安德森(David Anderson)、桑德拉·明基宁(Sandra Minkkinen)、德博拉·康托尔-亚当斯(Deborah Cantor-Adams)和克里斯·马洛伊(Chris Malloy)对此书精心的编辑和付出的辛劳。

谨以此书献给我的母亲:玛乔丽·罗兹(Marjorie Roads)。

对于本书的中文版本,我要感谢以下人员为本书提供了堪误,他们是:Keiji Hirata、Takafumi Hikichi、詹姆斯·麦卡特尼(James McCartney)和格雷厄姆·哈德菲尔德(Graham Hadfield)。

凡　例(Notes)

1. 由于本书由多人翻译，又是一部大型的译著，所以全书文字体例的统一就显得十分的重要。

特别是有关“主题词英汉对照表”的确定，涉及了国内整个计算机音乐学科的重大理论课题，目前本书所确定“主题词”(关键词)和“人名英汉对照表”的中译名经过了译者与业内专家学者反复和认真的论证，但难免不妥之处，有待在今后的修订中不断完善。

2. 本书各章节内，所涉及的观点、学术成果、音响制品在原著作中均在括号内简略标明著述人(制作公司)的名字，出处及年份，读者可以依据这些信息，在书后的参考文献(References)中，查到相关的论文的标题及其他更为详细的资料。本书对此种情况采取外文不译出、原文照录的方式，读者可以根据需要，通过这些信息在参考文献(原外文)中，以外文检索方式查到相关的内容。部分外文人名可以在本书“人名英汉对照表”中查到中文的译名。一般不在括号引用的这类信息仍应译出中文。

一般有如下几种情况(举例说明)：

1) Oppenheim 1992＝表示(论点)出自于奥本赫姆 1992 年的论文，并可据此在书后的参考文献(References)中查到更多信息。

2) Rodet and Cointe 1984＝表示(论点)出自于罗代和考因特 1984 年合作的论文。

3) Friberg et al. 1991＝表示(论点)出自于弗雷伯格等人合作的论文。

4) Stockhausen 1971b＝表示(论点)出自于斯托克豪森 1971 年排序 b 的论文。

5) Convolution，Roads1993a＝表示盘旋结构由罗兹在 1993 年他的排序 a 的论文中表述。

6) 1985，wergo campact disc 2010-50＝表示沃格唱片于 1985 年出品的激光唱片，2012-50 是唱片编号。

读者可自行以此类推。

3. 本书的参考文献(References)照录排在全书的最后,不另行翻译,只作读者查阅参考之用。

4. 本书日文人名和部分外文人名不译出,原文照录。

5. 本书正文标题均中英文对照排出。

6. 本书的正文小四号字,标题分大(四号)、中(小四号)、斜体(小四号)分层排出(均加粗),目录标题规格同原著。

7. 本书原文中斜排的关键词或人名,采取中文加括号内注原文(正体)的形式呈现。

如:原文"... are said to be in the *stopband* of a filter."

应作:"……称其处于滤波器阻带(stopband)中。"

8. 世纪、年份开用阿拉伯数字表示,不用"1930 年代",而用"20 世纪 30 年代"表示。

9. 本书"部分"的序号采用汉字数字,如:第一部分、第二部分等。

10. 本书"章"的序号均采用阿拉伯数字,如:第 1 章、第 2 章等。

11. 本书有关术语、作品名等词汇或词组原则上在其后标注原文;其中外文术语不加粗,外文的作品名称和书名变斜体,不加粗。

如:原文"... including *Prozession* and *Kurzwellen* (Stockhausen 1968)."

应作:"…… 包括《行进》(*Prozession*)、《短波》(*Kurzwellen*) (Stockhausen 1968)。"

但是,如果相近文本(同一章节)内再次出现或多次出现该词语时,则可视情况省略原文标注。

12. 照片版权不可少,可原文照录。

13. 人名从"人名英汉对照表",一般日本人名不译,"人名英汉对照表"以外的人名也可不译。

14. 产品名一般不译。

15. 本书所有插图中的标示文字均采用在该图注文下分别译出的方式,而不在图中译出,以保持图的原貌。如:"Play=播放","Record=录音","Track=音轨","Live performance on a musical keyboard=音乐键盘上的现场演奏"等。

图书在版编目(CIP)数据

计算机音乐教程/(美)罗兹等著；李斯心等译.—北京：人民音乐出版社，2011.5
ISBN 978-7-103-03702-7

Ⅰ.计… Ⅱ.①罗…②李… Ⅲ.计算机应用-音乐制作-教材 Ⅳ.J619-39

中国版本图书馆CIP数据核字(2010)第035671号

特约编辑：张志羽
责任编辑：任 云

著作权合同登记
图字：01-2003-8896号

The Computer Music Tutorial

本书由Massach Usetts Institute of Technology授权出版

人民音乐出版社出版发行
(北京市东城区朝阳门内大街甲55号邮政编码：100010)
Http://www.rymusic.com.cn
E-mail:rmyy@rymusic.com.cn
新华书店北京发行所经销
北京美通印刷有限公司印刷
787×1092毫米 16开 4插页 72.25印张
2011年5月北京第1版 2011年5月北京第1次印刷
印数：1-1,500册 (上、下册) 定价：182.00元