译注: 本人是边学边译,对专业术语不太了解,英语功底也较差, 请不吝赐教, 以便我可以改进, 谢谢

1 简介

V. Verfaille, M. Holters and U. Zo¨lzer

# 1.1. 基于MATLAB的数字音效(Digital audio effects, DAFX)

音效被参与音乐制作的所有个人所使用，从音乐家的特殊演奏技术开始，包括特殊麦克风技术的使用，传送到效果处理器以合成，记录，产生和广播音乐信号。本书将涵盖几种类别的声音或音频效果及其对声音修改的影响。数字音效(缩写为DAFX)是根据某些声音控制参数, 对输入的音频信号或声音进行修改，并提供输出信号或声音的软硬件（见图1.1）。输入和输出信号通过扬声器或耳机进行监听，并对信号进行某种形式的视觉表示，例如时间信号，信号电平及其频谱。根据声学标准，声音工程师或音乐家为他希望达到的声音效果调整他的控制参数。输入和输出信号均为数字格式，代表模拟音频信号。数字音频效果的主要目标是修改输入信号的声音特性。控制参数的设置通常由声音工程师，音乐家（表演者，作曲家或数字乐器制造商）或简单地由音乐收听者完成，但也可以属于数字音频效果的信号处理链中特定级别的一部分。

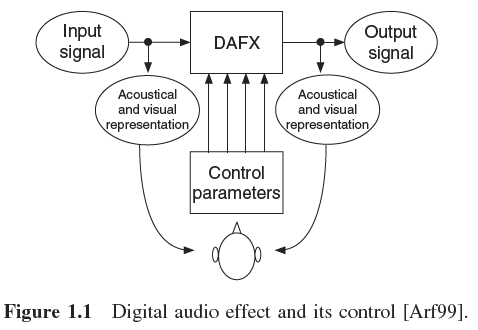


图1.1 数字音效和控制[Arf99]

本书的目的是描述以下类型的数字音频效果：

* 物理和声学效果：我们简要介绍一下物理背景和说明。我们描述了产生声音效果的模拟装置或设备。
* 数字信号处理：我们对底层算法进行了正式描述，并显示了一些实现示例。
* 音乐应用程序：我们指出了一些应用程序，并提供了CD或网络上可用的声音示例的引用。

在每个效果描述的开头将介绍数字音效的物理和声学现象，然后解释实现该效果的信号处理技术，一些音乐应用程序以及效果参数的控制。

在作为介绍的本章中，我们接下来介绍一些词汇说明，然后概述数字音频效果的分类。然后，我们解释一些简单的数字信号处理基础知识，并展示如何编写使用MATLAB仿真工具(http://www.mathworks.com)或免费软件仿真工具(http://www.octave.org)进行音频效果处理的仿真软件。 MATLAB的数字音频效果实现距离在个人计算机上实时运行或允许对其参数的实时控制还有很长的路要走。不过，使用MATLAB进行信号处理算法（尤其是声音效果算法）的编程非常容易，并且可以很快地学习。

## 声音效果，音频效果和声音变换

使用“效果”一词，体现的是观察一种现象的客观性。实际上，“效果”表示在人的思想中产生的印象，是由原因引起的感知变化。这个词的两种用法表示相关但略有不同的方面：“声音效果”和“音频效果”。请注意，在本书中，我们将专门讨论后者。

“声音效果”一词通常用于描述各种earcones（icons for the ear, 译注: 10.1.1.472.2278这篇论文里面将earcones标注为抽象的音乐声<abstract musical sounds>, 还有个相关的概念auditory icons标记为a sound caricature of the intended action the user is supposed to take or has taken>），特殊声音，这些声音在生产模式下具有很强的标志性，因此很容易识别。声音效果数据库提供自然的（录制的）和处理后的声音（来自声音合成和音频效果），这些声音对感知产生特定的影响，用于在各种情况下模拟动作，交互或情感。例如，它们用于电影配乐，卡通和音乐作品。

另一方面，“音频效果”一词描述的是这样一种工具, 它可以对声音进行变换以改变我们对这个声音的感知。

我们可以将这两种含义理解为“效果”含义的转移：从对变化本身的感知到用于实现这种感知变化的信号处理技术。这种转变反映了目标（感知到的东西）和制造目标的工具（信号处理技术）之间的语义混淆。 “声音效果”确实涉及主观观点，而“音频效果”使用与主题相关的术语（效果）来谈论客观事实：产生声音变换的工具。

从历史上看，可以说，当在精致的声音模型上使用此表达时，首先出现音频效果，然后出现声音转换。实际上，利用分析/变换/合成方案的技术, 包含了在精致声音的模型上执行变换的步骤。这从技术方面明显地区分了“音频效果”和”音频转换”, 前者使用声音的简单表示（样本）来执行信号处理，而后者使用复杂的技术来执行增强的信号处理。

音频效果最初表示为基于简单操作的简单处理系统，例如通过延迟线调制的随机控制进行合唱；用延迟线产生回声；用非线性处理造成的失真。可以认为音频效果是在声音的”表面”进行处理，因为声音由波形样本表示（不是高级声音模型），并且仅通过延迟线，滤波器，增益等进行处理。说是” 表面”, 不是指声音被改变的多少（实际上可以进行深度修改；比如失真），而是指我们在表述并准确/精确地处理声音和模型参数方面走了多远。

另一方面，声音变换表示基于分析/变换/合成模型的复杂处理系统。例如，我们想到具有基本频率跟踪能力的相位声码器(phase vocoder)，源滤波器模型或正弦加残差加性模型(sinusoidal plus residual additive model)。它们被认为可以提供更深层次的修改，例如高质量的能保留共振峰的音高转换(pitch-shifting with formant preservation)，音色变形(timbre morphing), 具有启动/音高/声像保留的时间缩放(time-scaling with attack, pitch and panning preservation)。对控制参数的这种深层次的操纵反过来又使声音修改听起来非常微妙。

但是，随着时间的流逝，实践模糊了音频效果和声音转换之间的界限。实际上，几种分析/转换/合成方案可以简单地执行我们认为是音频效果的各种处理。另一方面，为了达到控制频率范围和幅度增益的能力，同时注意限制相位调制，常规的音频效果（例如滤波器）在设计方面已经取得了长足的发展。另外，一些通常被认为是简单处理的音频效果实际上需要复杂的处理。例如，混响系统通常被认为是简单的音频效果，因为它们最初是使用带有延迟线的简单操作开发的，即使它们应用了复杂的声音转换。出于所有这些原因，人们可能会认为术语“音频效果”，“声音转换”和“音乐声音处理”都指的是相同的想法，即对声音应用信号处理技术以修改声音感知效果. 或换句话说，是将一种声音转换为另一种在感觉上不同的声音。尽管不同的术语通常可以互换使用，但为了保持一致，我们在整本书中都使用“音频效果”。

# 1.2. 数字音效分类

数字音频效果主要由作曲家，表演者和声音工程师使用，但通常从设计它们的DSP工程师的角度进行描述。因此，它们在软件文档和教科书中的分类和文档都依赖于基础技术。如果我们观察不同社区中发生的情况，则存在其他常用的分类方案。这些包括按信号处理分类[Orf96，PPPR96，Roa96，Moo90，Zol102]，按控件类型分类[VWD06]，按感知分类[ABL + 03]以及按声音和音乐计算分类[CPR95]等。仔细观察以便比较这些分类，我们观察到了很大的差异。原因是每种分类的引入都是为了最好地满足特定受众的需求；然后，它依赖于一系列功能。从逻辑上讲，此类功能与给定社区相关，但对于其他社区可能毫无意义或晦涩难懂。例如，信号处理技术很少根据修改后的感知特征来呈现，而是从声学维度(acoustical dimensions)来呈现。相反，作曲家通常依赖于感知或认知特征，而不是声学尺寸，甚至更少依赖信号处理方面。

一种跨学科的音频效果分类方法[VGT06]旨在促进从事音频效果或使用音频效果的研究人员和创作者之间的交流。这涉及到各个学科：从声学和电气工程到心理声学，音乐认知和心理语言学。接下来的小节通过对音乐中的沟通链路(Communication chain)的描述，介绍了数字音频效果的各种观点。从这个角度出发，描述了三种特定学科的分类：基于底层技术，控制信号和感知特性，然后引入跨学科分类，以将不同层面的领域相关的表示法链接起来。应该指出的是，提出的分类不是严格的分类，因为它们既不是穷举也不是互斥的：一个效果可以属于一个以上的类别，这取决于其他参数，例如控制类型，产生的假象，使用的技术等

## 音乐(生命周期)中的沟通链路(communication chain)

尽管需求和观点多种多样，但音频效果的实际用户（作曲家和表演者）仍主要使用技术术语。 这种技术分类可能是最严格和最系统的分类，但不幸的是，它仅涉及所使用的技术，而忽略了我们对由此产生的音频效果的理解，而这在音乐似乎方面更有意义。

我们来看看音乐声创作过程中的沟通链路[Rab，HMM04]。根据莫利诺的著作[Mol75], 沟通链路的概念, 已经从语言学和符号学 [Nat75] 中改编而来, 并用在音乐上。 通过三重符号学这种适应性方案，通过诸如声音之类的物理上的中性轨迹(trace)，可以区分作曲家（生产者）和听者（接收者）之间的三个音乐交流层面。

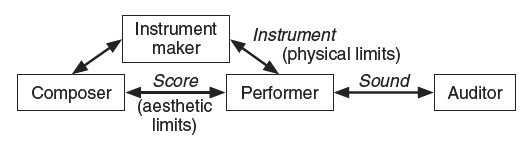


图1.2音乐中的沟通链：作曲家，表演者和乐器制造商也是听众，但与听众场景不同。

如图1.2所示，为了研究音频效果的所有可能的观察角度，我们根据绘制了完整的沟通链路。

在此过程中, 我们列出了, 以各种方式参与音乐概念、创作和感知过程的所有参与者，他们分别是乐器制作者，作曲家，表演者和听众。

* 诗意层面(poietic level)涉及音乐信息的概念和创作，乐器制造者、作曲家和表演者以不同的方式和阶段参与其中。
* 中性层面(physical level)是物理“轨迹”（乐器，声音或乐谱）。
* 审美层面(aesthetic level)对应于听众对音乐信息的感知和接受。

对于音频效果，乐器制造者是设计者（实现信号处理），表演者是效果的使用者（音乐家，声音工程师）。在家庭工作室和特定音乐流派（例如混合音乐创作）的背景下，作曲家，表演者和乐器制作者（音乐技术人员）通常是截然不同的人，需要彼此有效地沟通。但是，链路中的所有演员也是听众，他们可以分享他们所听到的内容以及他们对其的解释和描述。

因此，我们将把感知和认知的观点视为不同领域分类的跨学科网络的入口点。我们还考虑家庭工作室的特殊情况，其中表演者也可以是他自己的声音工程师，设计或设置其处理链，并进行母带制作（performs the mastering）。同样，电声音乐作曲家通常将此类任务与其他编程和演奏技巧结合在一起。他们构思自己的处理系统，控制并演奏乐器。尽管在这两种情况下，所有制作任务均由单个多学科的艺术家执行，但横向分类仍有助于更好地从技术到感性的角度来理解,音频效果的不同描述层面之间的关系。

## 1.2.1. 基于底层技术分类

从“乐器制造商”（DSP工程师或软件工程师）的角度出发，第一个分类重点是用于实现音频效果的底层技术。 实际上，许多数字音频效果都是在模仿它的模拟原型(analog ancestors)。 同样，一些模拟音频效果只是模仿其他已经存在的模拟音频效果。 当然，在某些时候，模拟和/或数字技术也被创造性地使用，以提供新的效果。 我们可以按年代顺序区分以下模拟技术：

* 机械/声学（例如乐器和室内声学效果），例如为其声学特性设计或选择一个特定的房间，音乐被修改并塑造成作曲家和表演者的想要的效果。 通过机电手段，留声机产品可以用于通过改变磁盘旋转速度来对声音进行时间缩放和音高偏移。
* 机电（例如，使用留声机唱片(vinyl)）
* 电磁（例如，磁带镶边(flanging, 译注:参考<http://blog.sina.com.cn/s/blog_448565dc0100gcov.html>)和时间缩放(time-scaling)），最初是在将拇指按在磁带录音机的边缘上时产生镶边(flanging)的，现在可以用具有不同延迟的数字梳状滤波器来模拟。电磁手段的另一个例子是在1950年代初，作曲家和工程师Pierre Schaeffer进行了不带音调偏移的时间缩放效果（即，音色没有“太差劲”）。
* 电子（例如滤波器，声码器，环形调制器）。电子方式包括环形调制，环形调制指的是两个信号的相乘，其名称来自最初用于实现此效果的二极管的环形模拟电路。

数字效果模仿了机电、电气或电子效果的声学或感知特性， 包括滤波，哇音效果(wah-wah)，声码器效果，混响，回声和Leslie效果(译注:参考<http://kingfour.com/leslie.html>)。

最近，电子和数字声音处理和合成技术允许创建新的前所未有的效果，例如机器人化(robotization)，频谱全景化(spectral panoramization)，通过自适应时间缩放和音高转换实现的韵律变化等。 当然，模仿与创造性使用技术之间的界限并不明确。 声码器效果, 最开始是使用滤波器组控制频谱包络, 来对语音进行编码，但后来用于音乐目的，特别是在音乐声中增加人声方面。 数字合成的对应部分是通过系统地创造性地使用（LPC，相位声码器）来模仿声学特性。

数字音频效果的分类可以在实现技术的基础上进行组织，如本书所建议的：

* 过滤器和延迟（重采样）
* 调制器和解调器
* 非线性处理
* 空间效应
* 时间段处理
* 时频处理
* 源过滤器处理
* 自适应效果处理
* 谱域处理
* 时间和频率扭曲
* 虚拟模拟效果
* 自动混音
* 音源分离

数字音频效果的另一种分类是基于进行信号处理的域（即时间、频率或时频），以及处理是按采样还是按块的：

* 时域：
  + 按块处理, 使用重叠相加（overlap-add , OLA）技术（例如基本OLA，同步OLA，音高同步OLA）
  + 按样本处理（滤波器，使用延迟线，增益，非线性处理，重采样和插值）
* 频域（带块处理）：
  + 具有傅立叶逆变换的频域合成（例如，具有或不具有相位解包(phase unwrapping)的相位声码器(phase vocoder)）
* 时域综合（使用振荡器组）
  + 时域和频域（例如，相位声码器加LPC）。

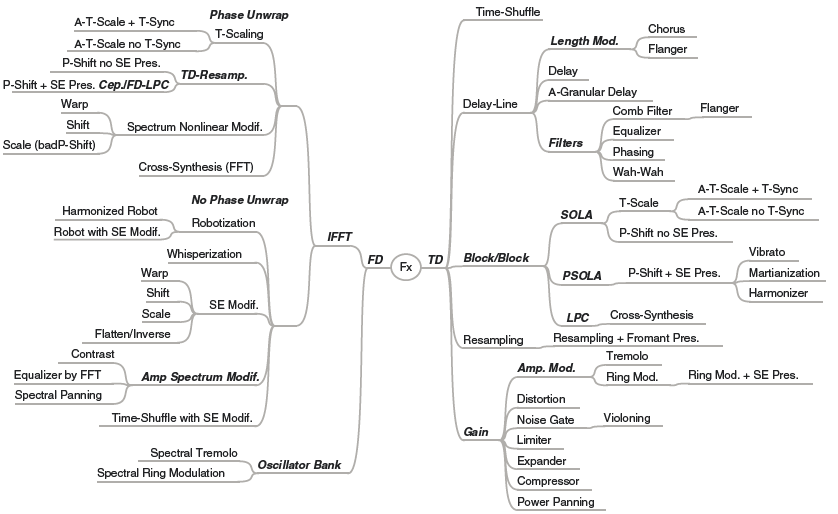


图1. 3 一种可以用来设计多效果系统音效的技术分类。 “ TD”代表“时域”，“ FD”代表“频域”，“ t-scale”代表“时间缩放”，“ p-shift”代表“音高移位”，“+”代表“ 和”, “ A-”表示自适应控制，“ SE”表示“频谱包络”，“ osc”表示“振荡器”，“ mod.”表示“调制”，“ modif.”表示“修改”。粗斜体词表示技术方面，而常规字体词表示音频效果。

这种基于基础技术的分类的优势在于，软件开发人员可以轻松地看到各种效果在技术和实现方面的相似性，从而简化了对多效果系统的理解和实现, 如图1.3。它还很好地概述了效果所涉及的技术领域和信号处理技术。但是，有几个音频效果出现在图表的两个位置属于多个类（再次说明了为什么该图表不是真正的分类），因为它们可以使用来自不同领域的技术来实现。例如，可以使用时间分段处理以及时频处理来执行时间缩放。进一步，具有时间同步的自适应时间缩放[VZA06]够可以使用SOLA(synchronized OLA)逐块或时域处理来实现，也可以使用基于逐块频域分析来加IFFT合成的相位声码器来实现。

根据用户的专业知识（DSP程序员，电声作曲家），这种分类可能不是最容易理解的，因为这种类型的分类没有显式说明是如何影响感知特性的，而感知特性是所有听众的通用词汇。在分类中引入声音的感知属性的另一个原因是，当用户可以在效果的各种实现之间进行选择时，他们也可以从某种效果的多个可选方案中选择。例如，使用时间换算，重采样不会保留音调或共振峰；带循环缓冲区的OLA增加了窗口调制，声音听起来更加粗糙和过滤；相位声码器听起来有点混响，除了启动(attack)外，“正弦噪声”加性模型听起来不错，“正弦瞬态噪声”加性模型可以保留启动(attack)，但不能保留多通道声音的空间图像等。因此，为了选择一种技术时，用户必须了解每种技术方案. 因此，将实现技术关联到感知特征的需求变得很清楚，接下来将进行讨论。

## 1.2.2. 基于感知特性分类

使用感知分类，可以根据感知属性对音频效果进行分类, 该效果主要通过数字处理改变（还有“音乐手势”的例子）：

响度：与强弱（dynamics）、nuances和phrasing（连奏和弹拨, legato and pizzicato）、重音、震音（Tremolo）有关

时间：与持续时间、节奏和节奏变化有关（加速，减速）

音高：由高度和色度(chroma)组成，与旋律、音调、和声相关并表现为这些形式； 有时与滑弦（Glissandi）有关

空间听觉：与声源定位（距离，方位角，仰角），运动（多普勒）和方向性以及房间效果（混响，回声）有关

音色：由短期时间特征（例如瞬变和起伏）和长期时间特征（共振峰（音色）和频谱属性（纹理（texture），谐波））组成，两个方面都包括亮度（或频谱高度）、 音质、音质变态； 相关的音乐手势包含各种演奏模式、装饰（ornamentation ）和特殊效果，例如抖音（Vibrato，揉弦）、颤音（trill）、颤舌音（flutter tonguing）、连奏（legato）、拨弹（pizzicato）、和弦音符（harmonic notes）、多音（multiphonics）等。

我们认为这种分类对于音乐家和听众来说是最自然的分类，因为这种感知属性通常可以在乐谱中清楚地识别出来。 它已用于对基于内容的转换[ABL + 03]以及自适应音频效果[VZA06]进行分类。 因此，我们现在通过重点介绍每种感知属性的心理声学基础知识，来讨论这些感知属性的更详细概述。 我们还命名了常用的数字音频效果，特别强调音色，因为这种更复杂的感知属性提供了最大范围的声音可能性。 我们还强调了感知属性（或高级特征）与其物理意义（信号或低级特征）之间的关系，后者通常更易于计算。

响度(Loudness)：响度是声音随时间变化的感知强度(intensity)。它的计算模型对关键频带中的能量进行时间和频率积分[ZS65，Zwi77] 。由RMS（均方根， root mean square）计算出的声强级(intensity level)是其物理意义。使用加法分析和瞬态检测，我们提取了谐波含量，瞬态（transient ）和残差（residual）的声强级。我们通常使用对数刻度，即分贝：强度为I时，响度为LdB =20log10I。通过将声音强度级别乘以10，响度将增加20 dB。响度的音乐意义被称为强弱（dynamic），它对应的标度范围从pianissimo（pp）到fortissimo（ff），两个连续音调之间有3 dB的间隔。震音（Tremolo）描述了具有特定频率和深度的响度调制。常用的响度效果会改变声音强度：音量变化，震音（tremolo），压缩器（compressor），扩展器（Expander），噪声门（noise gate）和限制器（limiter）。

* 震音（Tremolo）是声音强度级别的正弦振幅调制，调制频率在4到7 Hz之间（在抖音（vibrato）的5.5 Hz频率调制附近）。
* 压缩器和扩展器使用非线性函数修改强度级别(Intensity level)；它们是最早的自适应效果之一。前者压缩强度水平，从而产生更多的打击乐声音，而后者则具有相反的效果，并用于扩展声音的强弱范围。
* 通过特定的非线性函数，我们可以获得噪声门和限制器效应。噪声门过滤掉响度非常低的声音，这对于避免背景噪声在包含延迟的系统中回环传播(circulate)特别有用。限制强度级别可以保护硬件。
* 其他形式的响度效果包括自动混音器和自动音量/增益控制，有时会用到噪声传感器(noise-sensor)。

时间和节奏：时间是通过两个非常复杂的属性来感知的：声音的持续和间隙时间，以及基于重复和模式推断的节奏[DH92]。节拍(beat)可以用自相关(autocorrelation)技术提取，模式(pattern)可以用定量(quantification)技术提取[Lar01]。

* 时间缩放(time scale)用于将信号时间匹配到给定时间，从而影响节奏。重采样(resampling)可以实现时间缩放，但是导致不必要的音高偏移(pitch shifting)。时间缩放比率通常是恒定的，大于1时就是时间扩展（或时间拉长，时间扩张：声音会变慢），小于1时, 就是时间压缩（或时间收缩：声音会加速）。
* 三种逐块处理技术可以避免这种(音高偏移)的情况：相位声码器(phase vocoder)[Por76，Dol86，AKZ02a]，SOLA [MC90，Lar98]和加性模型(additive model)[MQ86，SS90，VLM97]。
* 相位声码器技术的时间缩放包括使用不同的分析和合成步长增量。相位声码器使用可短时傅立叶变换（STFT）[AR77]。在分析步骤中，以RA样本的步长增量执行窗口输入块的STFT。在合成步骤中，傅立叶逆变换提供输出块，将这些窗口加窗，重叠，然后加上RS采样步长增量。必须适当地选择相位声码器的步长增量，以提供信号[All77，AR77]的完美重构。合成STFT的每个频率桶 (bin) 都需要相位计算。相位声码器技术可以对任何类型的声音进行时间缩放，但如果不加注意，则会增加相位.
* 峰值锁相技术可以解决此(增加相位)问题[Puc95，LD97]。 SOLA技术的时间缩放是通过对时间颗粒或块进行复制或抑制来实现的，其中重叠颗粒的音高同步，以避免相位抵消引起的低频调制。音高同步意味着SOLA技术只能正确处理单声道声音。在谈论SOLA技术时，我们指的是所有的同步和重叠添加技术：SOLA，TD-PSOLA，TF-PSOLA，WSOLA等。
* 使用加性模型进行时间缩放会缩放部分频率及其幅度的时间轴。加性模型可以处理谐波以及非谐音，同时具有良好的频谱线分析(spectral line analysis)质量。

音高：和声(Harmonic sound)的音高由基波(harmonics)的频率和振幅给定；其对应的物理概念是基频(fundamental frequency)。音高的属性是音高（高频/低频）和色度(chroma)或音色(color)[She82]。音乐声音可以是完美的和声（例如管乐器），接近和声的（例如弦乐器）或非和声的（例如打击乐器，钟声）。和谐度也与音色有关。感知音调的心理声学模型同时使用声音[dCO4]的频谱信息（频率）和周期性信息（时间）。音调以准对数梅尔音阶(quasi-logarithmic mel scale)表示，该音阶通过对数赫兹音阶(log-Hertz scale)近似。当将基本频率乘以2（相同的色度(chroma)，使音高加倍）时，音阶将加一个八度。音调随着时间的演绎, 在单个声音上被称为旋律(melody), 有多个声音时称为和声(harmony)。和声(Harmonic sound)的音调可以移动，从而改变调号。

* 音高变换是双重变换的, 包括时间缩放和在频域(时频表示的频率轴)上的缩放。音高偏移率(pitch-shifting ratio)大于1时音变高；低于1时变低。这可以通过一个包括时间缩放和重新采样的组合来实现。为了保留音色和频谱包络[AKZ02b]，相位声码器将信号分解为每个分析块的信号源和滤波器：对共振峰进行预校正（在频域上实现[AD98]），对源信号进行重新采样（在时域上实现）和连续块间的相位处理（在频域中）。
* PSOLA技术会保留频谱包络线[BJ95，ML95]，并通过使用不同于分析步长的合成步长来执行音高移位。
* 加性模型通过将每个部分的频率乘以变调比来缩放频谱。然后从频谱包络线线性内插振幅。诸如铃铛之类的非谐音的音调也可以通过振铃调制来执行。使
* 用音高转换效果，可以获得和音效果器(harmonizer )和自动调谐效果。和声包括将声音与几种变音版本混合以获得和弦。当受输入音高和旋律上下文控制时，称为智能(smart)和声[AOPW99]或智能(intelligence)和声。自动调谐包括对单声道信号进行音高偏移，以使音高匹配音阶[ABL + 03]。

Spatial Hearing: Spatial hearing has three attributes: the location, the directivity, and the room effect. The sound is localized by human beings with regards to distance, elevation and azimuth, through interaural intensity (IID) and inter-aural time (ITD) differences [Bla83], as well as through filtering via the head, the shoulders and the rest of the body (head-related transfer function, HRTF). When moving, sound is modified according to pitch, loudness and timbre, indicating the speed and direction of its motion (Doppler effect) [Cho71]. The directivity of a source is responsible for the differences in transfer functions according to the listener position relative to the source. The sound is transmitted through a medium as well as reflected, attenuated and filtered by obstacles (reverberation and echoes), thus providing cues for deducing the geometrical and material properties of the room. Spatial effects describe the spatialization of a sound with headphones or loudspeakers.

e.g., constant power panoramization with two loudspeakers or headphones [Bla83], vector-based amplitude panning (VBAP) [Pul97] or Ambisonics [Ger85] with more loudspeakers

The position in the space is simulated using intensity panning , delay lines to simulate the precedence effect due to ITD, as well as filters in a transaural or binaural context [Bla83].

The Doppler effect is due to the behaviour of sound waves approaching or going away; the sound motion throughout the space is simulated using amplitude modulation, pitch-shifting and filtering [Cho71, SSAB02]. Echoes are created using delay lines that can eventually be fractional [LVKL96]. The room effect is simulated with artificial reverberation units that use either delay-line networks or all-pass filters [SL61, Moo79] or convolution with an impulse response. The simulation of instruments’ directivity is performed with linear combination of simple directivity patterns of loudspeakers [WM01]. The rotating speaker used in the Leslie/Rotary is a directivity effect simulated as a Doppler [SSAB02].

空间听觉：空间听觉具有三个属性：位置、方向性和室内效果。通过听觉强度（interaural intensity ，IID）和听觉时间（inter-aural time ，ITD）差异[Bla83]，以及通过头部、肩膀和其他部位（与头部有关的传递函数，HRTF）的滤波，人类可以在距离，高度和方位方面对声音进行定位。移动时，声音的音高、响度和音色的变化，会反映出其运动的速度和方向（多普勒效应）[Cho71]。根据收听者相对于源的位置，源的方向性导致了传递函数的差异。声音通过介质传输，并被障碍物反射（混响和回声）、衰减和过滤，这可以用来推断房间的几何形状和材料特性。空间效应描述了耳机或扬声器对声音的空间化作用。空间中的位置使用声强定位（声强绘像, intensity panning）来模拟，并使用基于听觉时间的延迟线来模拟优先效应（哈斯效应）， 还会到用 经耳或双耳场景滤波函数。声强定位的例子有：使用两个扬声器或耳机[Bla83]进行恒定功率全景化，使用更多扬声器的基于矢量的振幅定位（VBAP）[Pul97]或Ambisonics [Ger85]）。

多普勒效应是由于声波接近或消失的行为所致。使用幅度调制，音高偏移和滤波来模拟整个空间中的声音运动[Cho71，SSAB02]。回波是使用延迟线创建的，该延迟线最终可能是分数[LVKL96]。使用人工混响单元模拟房间效果，该混响单元使用延迟线网络或全通滤波器[SL61，Moo79]或具有冲激响应的卷积。仪器指向性的仿真是通过扬声器[WM01]的简单指向性模式的线性组合来执行的。 Leslie / Rotary中使用的旋转扬声器是模拟为多普勒[SSAB02]的指向性效果。

Timbre: This attribute is difficult to define from a scientific point of view. It has been viewed for a long time as “that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar” [ANS60]. However, this does not take into account some basic facts, such as the ability to recognize and to name any instrument when hearing just one note or listening to it through a telephone [RW99]. The frequency composition of the sound is concerned, with the attack shape, the steady part and the decay of a sound, the variations of its spectral envelope through time (e.g., variations of formants of the voice), and the phase relationships between harmonics. These phase relationships are responsible for the whispered aspect of a voice, the roughness of low-frequency modulated signals, and also for the phasiness10 introduced when harmonics are not phase aligned. We consider that timbre has several other attributes, including:

The brightness or spectrum height, correlated to spectral centroid11 [MWdSK95], and com- puted with various models [Cab99]

The quality and noisiness, correlated to the signal-to-noise ratio (e.g., computed as the ratio between the harmonics and the residual intensity levels [ABL+03]) and to the voiciness

(computed from the autocorrelation function [BP89] as the second-highest peak value of the normalized autocorrelation)

The texture, related to jitter and shimmer of partials/harmonics [DT96] (resulting from a statistical analysis of the partials’ frequencies and amplitudes), to the balance of odd/even harmonics (given as the peak of the normalized autocorrelation sequence situated half way between the first- and second-highest peak values [AKZ02b]) and to harmonicity

The formants (especially vowels for the voice [Sun87]) extracted from the spectral envelope, the spectral envelope of the residual and the mel-frequency critical bands (MFCC), perceptual correlate of the spectral envelope.

Timbre can be verbalized in terms of roughness, harmonicity, as well as openness, acuteness and laxness for the voice [Sla85]. At a higher level of perception, it can also be defined by musical aspects such as vibrato [RDS+99], trill and Flatterzunge, and by note articulation such as appoyando, tirando and pizzicato.

Timbre effects is the widest category of audio effects and includes vibrato, chorus, flanging, phasing, equalization, spectral envelope modifications, spectral warping, whisperization, adaptive filtering and transient enhancement or attenuation.

Vibrato is used for emphasis and timbral variety [MB90], and is defined as a complex timbre pulsation or modulation [Sea36] implying frequency modulation, amplitude modulation and

10 Phasiness is usually involved in speakers reproduction, where phase inproperties make the sound poorly spatialized. In the phase vocoder technique, the phasiness refers to a reverberation artifact that appears when neighboring frequency bins representing the same sinusoid have different phase unwrapping.

11 The spectral centroid is also correlated to other low-level features: the spectral slope, the zero-crossing

rate, the high frequency content [MB96].

sometimes spectral-shape modulation [MB90, VGD05], with a nearly sinusoidal control. Its modulation frequency is around 5.5 Hz for the singing voice [Hon95]. Depending on the instruments, the vibrato is considered as a frequency modulation with a constant spectral shape (e.g., voice, [Sun87], stringed instruments [MK73, RW99]), an amplitude modulation (e.g., wind instruments), or a combination of both, on top of which may be added a complex spectral-shape modulation, with high-frequency harmonics enrichment due to non-linear properties of the resonant tube (voice [MB90], wind and brass instruments [RW99]).

A chorus effect appears when several performers play together the same piece of music (same in melody, rhythm, dynamics) with the same kind of instrument. Slight pitch, dynamic, rhythm and timbre differences arise because the instruments are not physically identical, nor are perfectly tuned and synchronized. It is simulated by adding to the signal the output of a randomly modulated delay line [Orf96, Dat97]. A sinusoidal modulation of the delay line creates a flanging or sweeping comb filter effect [Bar70, Har78, Smi84, Dat97]. Chorus and flanging are specific cases of phase modifications known as phase shifting or phasing.

Equalization is a well-known effect that exists in most of the sound systems. It consists in modifying the spectral envelope by filtering with the gains of a constant-Q filter bank. Shifting, scaling or warping of the spectral envelope is often used for voice sounds since it changes the formant places, yielding to the so-called Donald Duck effect [AKZ02b].

Spectral warping consists of modifying the spectrum in a non-linear way [Fav01], and can be achieved using the additive model or the phase vocoder technique with peak phase-locking [Puc95, LD97]. Spectral warping allows for pitch-shifting (or spectrum scaling), spectrum shifting, and in-harmonizing.

Whisperization transforms a spoken or sung voice into a whispered voice by randomizing either the magnitude spectrum or the phase spectrum of a short-time Fourier transform

[AKZ02a]. Hoarseness is a quite similar effect that takes advantage of the additive model to modify the harmonic-to-residual ratio [ABL+03].

Adaptive filtering is used in telecommunications [Hay96] in order to avoid the feedback loop effect created when the output signal of the telephone loudspeaker goes into the microphone. Filters can be applied in the time domain (comb filters, vocal-like filters, equalizer) or in the frequency domain (spectral envelope modification, equalizer).

Transient enhancement or attenuation is obtained by changing the prominence of the transient compared to the steady part of a sound, for example using an enhanced compressor combined with a transient detector.

Multi-Dimensional Effects: Many other effects modify several perceptual attributes of sounds simultaneously. For example, robotization consists of replacing a human voice with a metallic machine-like voice by adding roughness, changing the pitch and locally preserving the formants. This is done using the phase vocoder and zeroing the phase of the grain STFT with a step increment given as the inverse of the fundamental frequency. All the samples between two successive non overlapping grains are zeroed12 [AKZ02a]. Resampling consists of interpolating the wave form, thus modifying duration, pitch and timbre (formants). Ring modulation is an amplitude modulation without the original signal. As a consequence, it duplicates and shifts the spectrum and modifies pitch and timbre, depending on the relationship between the modulation frequency and the signal fundamental frequency [Dut91]. Pitch-shifting without preserving the spectral envelope modifies

12 The robotization processing preserves the spectral shape of a processed grain at the local level. However, the formants are slightly modified at the global level because of overlap-add of grains with non-phase-aligned grain (phase cancellation) or with zeros (flattening of the spectral envelope).

both pitch and timbre. The use of multi-tap monophonic or stereophonic echoes allow for rhythmic, melodic and harmonic constructions through superposition of delayed sounds.

Summary of Effects by Perceptual Attribute: For the main audio effects, Tables 1.1, 1.2, and 1.3 indicate the perceptual attributes modified, along with complementary information for programmers and users about real-time implementation and control type. When the user chooses an effect to modify one perceptual attribute, the implementation technique used may introduce artifacts, implying modifications of other attributes. For that reason, we differentiate the perceptual attributes that we primarily want to modify (“main” perceptual attributes, and the corresponding dominant modification perceived) and the “secondary” perceptual attributes that are slightly modified (on purpose or as a by-product of the signal processing).

Table 1.1 Digital audio effects according to modified perceptual attributes (L for loudness, D for duration and rhythm, P for pitch and harmony, T for timbre and quality, and S for spatial qualities). We also indicate if real-time implementation (RT) is not possible (using “ ”), and the built-in control type (A for adaptive, cross-A for cross-adaptive, and LFO for low-frequency oscillator).

Effect name Perceptual Attributes RT Control

Main Other

Effects mainly on loudness (L)

compressor, limiter, expander, noise gate L T A

gain/amplification L

normalization L –

tremolo L LFO

violoning (attack smoothing) L T A

Effects mainly on duration (D)

time inversion D P,L,T –

time-scaling D

time-scaling with formant preservation D –

time-scaling with vibrato preservation D –

time-scaling with attack preservation D – A

rhythm/swing change D T – A

Effects mainly on pitch (P)

pitch-shifting without formant preservation P T

pitch-shifting with formant preservation P

pitch change P A

pitch discretization (auto-tune) P T A

harmonizer/smart harmony P A

(in-)harmonizer P A

Effects mainly on spatial aspects (S)

distance change

S

L,T

directivity S P,T

Doppler effect S L,P

echo S L

granular delay S L,D,P,T A

reverberation S L,D,T

panning (2D, 3D) S

spectral panning S L,T

rotary/Leslie S P,T LFO

Table 1.2 Digital audio effects that mainly modify timbre only.

Effect name Perceptual Attributes RT Control

Main Other

Effects mainly on timbre (T)

Effects on spectral envelope:

filter T L

arbitrary resolution filter T L

comb filter T L,P

resonant filter T L,P

equalizer T L

wah-wah T L,P

auto-wah (sensitive wah) T L,D,P LFO

envelope shifting T L

envelope scaling T L

envelope warping T L

spectral centroid change T L

Effects on phase:

chorus T random

flanger T P LFO

phaser T P LFO

Effects on spectral structure:

spectrum shifting T P

adaptive ring modulation T P A

texture change T

Effects on spectrum & envelope:

distortion T L,P

fuzz T L,P

overdrive T L,P

spectral (in-)harmonizer T

mutation T L,P cross-A

spectral interpolation: T L,P cross-A

vocoding T L,P cross-A

cross-synthesis T L,P cross-A

voice morphing T L,P cross-A

timbral metamorphosis T L,P cross-A

timbral morphing T L,P cross-A

whispering/hoarseness T L –

de-esser T L A

declicking T L –

denoising T L

exciter T L

enhancer T L

Table 1.3 Digital audio effects that modify several perceptual attributes (on purpose).

Effect name Perceptual Attributes RT Control Main Other

Effects modifying several perceptual attributes

spectral compressor L,T

gender change P,T L A

intonation change L,P A

martianisation P,T L A

prosody change L,D,P A

resampling D,T L,P –

ring modulation P,T

robotization P,T L

spectral tremolo L,T D LFO

spectral warping T,P L

time shuffling L,D,P,T –

vibrato L,P T,D LFO

By making use of heuristic maps [BB96] we can represent the various links between an effect and perceptual attributes, as depicted in Figure 1.4, where audio effects are linked in the center to the main perceptual attribute modified. Some sub-attributes (not necessarily perceptual) are introduced. For the sake of simplicity, audio effects are attached to the center only for the main modified perceptual attributes. When other attributes are slightly modified, they are indicated on the opposite side, i.e., at the figure bounds. When other perceptual attributes are slightly modified by an audio effect, those links are not connected to the center, in order to avoid overloading the heuristic map, but rather to the outer direction. A perceptual classification has the advantage of presenting audio effects according to the way they are perceived, taking into account the audible artifacts of the implementation techniques. The diagram in Figure 1.4, however, only represents each audio effect in its expected use (e.g., a compressor set to compress the dynamic range, which in turn slightly modifies the attacks and possibly timbre; it does not indicate all the possible settings, such as the attack smoothing and resulting timbral change when the attack time is set to 2s for instance). Of course, none of the presented classifications is perfect, and the adequacy of each depends on the goal we have in mind when using it. However, for sharing and spreading knowledge about audio effects between DSP programmers, musicians and listeners, this classification offers a vocabulary dealing with our auditory perception of the sound produced by the audio effect, that we all share since we all are listeners in the communication chain.

## 1.2.3. 跨学科分类

Before introducing an interdisciplinary classification of audio effects that links the different layers of domain-specific descriptors, we recall sound effect classifications, as they provide clues for such interdisciplinary classifications. Sound effects have been thoroughly investigated in electroa- coustic music. For instance, Schaeffer [Sch66] classified sounds according to: (i) matter , which is constituted of mass (noisiness; related to spectral density), harmonic timbre (harmonicity) and grain (the micro-structure of sound); (ii) form, which is constituted of dynamic (intensity evolu- tion), and allure (e.g., frequency and amplitude modulation); (iii) variation, which is constituted of melodic profile (e.g., pitch variations) and mass profile (e.g., mass variations). In the context

amplification Timbre compressor Timbre expander Timbre noise gate

Timbre limiter Loudness Timbre contrast

nuance change Rhythm

Time

timbral metamorphosis Loudness

Pitch

gender change Pitch spectral envelope warping vocoder effect

Pitch

timbre morphing cross synthesis Loudness

Timbre spectral tremolo tremolo

Timbre granular delay echo Room reverberation distance

height

panning

Timbre spec. panning azimuth

Formants

hybridization spectral interpolation mutation

scaling Voice quality

spec. env. modifications shifting

warping

Loudness

Pitch Doppler

3D binaural

3D transaural

Localization

Space

harmonics generator Pitch subharmonics generator Pitch scaling Pitch

Pitch

directivity change Directivity Directivity

Timbre

Spectrum shifting

Harmonicity

Formants

Localization Leslie / Rotary Pitch

Formants no formant preservation pitch-shifting

spectral warping

detune Harmonicity

autotune harmonizer Pitch

Time resampling

Time prosody change

Loudness intonation change

Timbre

vibrato preservation

formants preservation time-scaling

Audio Effects

ring modulation spectral ring modulation

SSB modulation Formants Harmonicity

warping inharmonizer Harmonicity denoising

declicking

enhancer Brightness distorsion Pitch fuzz

Quality vibrato Pitch

robotization Pitch

whisperization voice quality hoarseness

tremolo preservation attack preservation

Pitch resampling

Duration

Time

martianization Pitch

centroid change Brightness comb filter Pitch

swing change Rhythm Timbre time-shuffling Timbre inversion

resonant filter Pitch

Filter telephon effect

chorus, flanger, phase, wah-wah

Figure 1.4 Perceptual classification of various audio effects. Bold-italic words are perceptual attributes (pitch, loudness, etc.). Italic words are perceptual sub-attributes (formants, harmonicity, etc.). Other words refer to the corresponding audio effects.

of ecological acoustics, Schafer [Sch77] introduced the idea that soundscapes reflect human activ- ities. He proposed four main categories of environmental sounds: mechanical sounds (traffic and machines), human sounds (voices, footsteps), collective sounds (resulting from social activities) and sounds conveying information about the environment (warning signals or spatial effects). He considers four aspects of sounds: (i) emotional and affective qualities (aesthetics), (ii) function

and meaning (semiotics and semantics), (iii) psychoacoustics (perception), (iv) acoustics (physical characteristics). That in turn can be used to develop classification categories [CKC+04]. Gaver [Gav93] also introduced the distinction between musical listening and everyday listening. Musi-

cal listening focuses on perceptual attributes of the sound itself (e.g., pitch, loudness), whereas everyday listening focuses on events to gather relevant information about our environment (e.g., car approaching), that is, not about the sound itself but rather about sound sources and actions producing sound. Recent research on soundscape perception validated this view by showing that people organize familiar sounds on the basis of source identification. But there is also evidence that the same sound can give rise to different cognitive representations which integrate semantic

features (e.g., meaning attributed to the sound) into physical characteristics of the acoustic signal [GKP+05]. Therefore, semantic features must be taken into consideration when classifying sounds, but they cannot be matched with physical characteristics in a one-to-one relationship.

Similarly to sound effects, audio effects give rise to different semantic interpretations depending on how they are implemented or controlled. Semantic descriptors were investigated in the context of distortion [MM01] and different standpoints on reverberation were summarized in [Ble01]. An

interdisciplinary classification links the various layers of discipline-specific classifications ranging from low-level to high-level features as follows:

• Digital implementation technique

• Processing domain

• Applied processing

• Control type

• Perceptual attributes

• Semantic descriptors.

It is an attempt to bridge the gaps between discipline-specific classifications by extending previous research on isolated audio effects.

Chorus Revisited. The first example in Figure 1.5 concerns the chorus effect. As previously said, a chorus effect appears when several performers play together the same piece of music (same in melody, rhythm, dynamics) with the same kind of instrument. Slight pitch, dynamic, rhythm and timbre differences arise because the instruments are not physically identical, nor are perfectly tuned and synchronized. This effect provides some warmth to a sound, and can be considered as an effect on timbre: even though it performs slight modifications of pitch and time unfolding, the resulting effect is mainly on timbre. While its usual implementation involves one or many delay lines, with modulated length and controlled by a white noise, an alternative and more realistic sounding implementation consists in using several slightly pitch-shifted and time-scaled versions of the same sound with refined models (SOLA, phase vocoder, spectral models) and mixing them together. In this case, the resulting audio effect sounds more like a chorus of people or instruments playing the same harmonic and rhythmic patterns together. Therefore, this effect’s control is a random generator (white noise), that controls a processing either in the time domain (using SOLA

Semantic Descriptors

Perceptual Attribute

Effect Name

Control Type

Applied Processing

Processing Domain

Digital Implementation Technique

Figure 1.5 Transverse diagram for the chorus effect.

or a delay line), in the time-frequency domain (using the phase vocoder) or in the frequency domain (using spectral models).

Wah-Wah Revisited. The wah-wah is an effect that simulates vowel coarticulation. It can be implemented in the time domain using either a resonant filter or a series of resonant filters to simulate several formants of each vowels. In any case, these filters can be implemented in the time domain as well as in the time-frequency domain (phase vocoder) and in the frequency domain (with spectral models). From the usual wah-wah effect, variations can be derived by modifying its control. Figure 1.6 illustrates various control types for the wah-wah effect. With an LFO, the control is periodic and the wah-wah is called an “auto-wah.” With gestural control, such as a foot pedal, it becomes the usual effect rock guitarists use since Jimmy Hendrix gave popularity to it. With an adaptive control based on the attack of each note, it becomes a “sensitive wah” that moves from “a” at the attack to “u” during the release. We now can better see the importance of specifying the control type as part of the effect definition.

Semantic Descriptors

Perceptual Attribute

Effect Name

Control Type

Applied Processing

Processing Domain

Digital Implementation Technique

Figure 1.6 Transverse diagram for the wah-wah effect: the control type defines the effect’s name, i.e., wah-wah, automatic wah-wah (with LFO) or sensitive wah-wah (adaptive control).

Comb Filter Revisited. Figure 1.7 depicts the interdisciplinary classification for the comb filter. This effect corresponds to filtering a signal using a comb-shaped frequency response. When the signal is rich and contains either a lot of partials, or a certain amount of noise, its filtering gives rise to a timbral pitch that can easily be heard. The sound is then similar to a sound heard through the resonances of a tube, or even vocal formants when the tube length is properly adjusted. As any filter, the effect can be implemented in both the time domain (using delay lines), the time-frequency domain (phase vocoder) and the frequency domain (spectral models). When controlled with a LFO, the comb filter changes its name to “phasing,” which sounds similar to a plane landing, and has been used in songs during the late 1960s to simulate the effects of drugs onto perception.

Cross-synthesis revisited. The transverse diagram for cross-synthesis shown in Figure 1.8 consists in applying the time-varying spectral envelope of one sound onto the source of a second sound, after having separated their two source and filter components. Since this effect takes the whole spectral envelope of one sound, it also conveys some amplitude and time information, resulting in modifications of timbre, but also loudness, and time and rhythm. It may provide the

Semantic Descriptors

Perceptual Attribute

Effect Name

Control Type

Applied Processing

Processing Domain

Digital Implementation Technique

Figure 1.7 Transverse diagram for the comb-filter effect: a modification of the control by adding a LFO results in another effect called “phasing.”

Semantic Descriptors

Perceptual Attribute

Control Type

Applied Processing

Processing Domain

Digital Implementation Technique

Figure 1.8 Transverse diagram for the cross-synthesis effect.

illusion of a talking instrument when the resonances of a human voice are applied onto the source of a musical instrument. It then provides a hybrid, mutant voice. After the source-filter separation, the filtering of the source of sound A with the filter from sound B can be applied in the time domain as well as in the frequency and the time-frequency domains. Other perceptually similar effects on voice are called voice morphing (as the processing used to produce the castrato’s voice in the movie Farinelli’s soundtrack), “voice impersonator” (the timbre of a voice from the database is mapped to your singing voice in real time), the “vocoder effect” (based on the classical vocoder), or the “talk box” (where the filter of a voice is applied to a guitar sound without removing its original resonances, then adding the voice’s resonances to the guitar’s resonances; as in Peter Frampton’s famous “Do you feel like I do”).

Distortion revisited. A fifth example is the distortion effect depicted in Figure 1.9. Distortion is produced from a soft or hard clipping of the signal, and results in a harmonic enrichment of a sound. It is widely used in popular music, especially through electric guitar that conveyed it from the beginning, due to amplification. Distortions can be implemented using amplitude warping (e.g., with Chebyshev polynomials or wave shaping), or with physical modeling of valve amplifiers. Depending on its settings, it may provide a warm sound, an aggressive sound, a bad quality sound, a metallic sound, and so on.

Semantic Descriptors

Perceptual Attribute

Effect Name

Control Type

Applied Processing

Processing Domain: Time

Digital Implementation Technique

Figure 1.9 Transverse diagram for the distortion effect.

Equalizer revisited. A last example is the equalizer depicted in Figure 1.10. Its design consists of a series of shelving and peak filters that can be implemented in the time domain (filters), in the time-frequency domain (phase vocoder) or in the frequency domain (with spectral models). The user directly controls the gain, bandwidth and center frequency in order to apply modifications of the energy in each frequency band, in order to better suit aesthetic needs and also correct losses in the transducer chain.

We illustrated and summarized various classifications of audio effects elaborated in different disciplinary fields. An interdisciplinary classification links the different layers of domain-specific features and aims to facilitate knowledge exchange between the fields of musical acoustics, signal processing, psychoacoustics and cognition. Besides addressing the classification of audio effects, we further explained the relationships between structural and control parameters of signal process- ing algorithms and the perceptual attributes modified by audio effects. A generalization of this

Semantic Descriptors

Perceptual Attribute

Effect name

Control Type

Applied Processing

Processing Domain

Digital Implementation Technique

Figure 1.10 Transverse diagram for the equalizer effect.

classification to all audio effects would have a strong impact on pedagogy, knowledge sharing across disciplinary fields and musical practice. For example, DSP engineers conceive better tools when they know how it can be used in a musical context. Furthermore, linking perceptual features to signal processing techniques enables the development of more intuitive user interfaces providing control over high-level perceptual and cognitive attributes rather than low-level signal parameters.

# 1.3. 数字信号处理基础g

The fundamentals of digital signal processing consist of the description of digital signals – in the context of this book we use digital audio signals – as a sequence of numbers with appropriate number representation and the description of digital systems , which are described by software algorithms to calculate an output sequence of numbers from an input sequence of numbers. The visual representation of digital systems is achieved by functional block diagram representations or signal flow graphs. We will focus on some simple basics as an introduction to the notation and refer the reader to the literature for an introduction to digital signal processing [ME93, Orf96, Zo¨l05, MSY98, Mit01].

## 1.3.1. 数字信号

The digital signal representation of an analog audio signal as a sequence of numbers is achieved by an analog-to-digital converter (ADC). The ADC performs sampling of the amplitudes of the analog signal x(t) on an equidistant grid along the horizontal time axis and quantization of the amplitudes to fixed samples represented by numbers x(n) along the vertical amplitude axis (see Fig. 1.11). The samples are shown as vertical lines with dots on the top. The analog signal x(t) denotes the signal amplitude over continuous time t in micro seconds. Following the ADC, the digital (discrete time and quantized amplitude) signal is a sequence (stream) of samples x(n) represented by numbers over the discrete time index n. The time distance between two consecutive samples is termed sampling interval T (sampling period) and the reciprocal is the sampling frequency

fs = 1 fs = 1

x (n) y (n )

0.05

x (t )

0.05

x (n)

0.05

y (n )

0.05

y (t )

0 0 0 0

0.05

0.05

0.05

0.05

0 500

t in s 

0 10 20

n 

0 10 20

n 

0 500

t in s 

Figure 1.11 Sampling and quantizing by ADC, digital audio effects and reconstruction by DAC.

fS 1/T (sampling rate). The sampling frequency reflects the number of samples per second in Hertz (Hz). According to the sampling theorem, it has to be chosen as twice the highest frequency fmax (signal bandwidth) contained in the analog signal, namely fS > 2 fmax. If we are forced to use a fixed sampling frequency fS , we have to make sure that our input signal to be sampled has a bandwidth according to fmax fS /2. If not, we have to reject higher frequencies by filtering with a lowpass filter which only passes all frequencies up to fmax. The digital signal is then passed to a DAFX box (digital system), which in this example performs a simple multiplication of each sample by 0.5 to deliver the output signal y(n) 0.5 x(n). This signal y(n) is then forwarded to a digital-to-analog converter DAC, which reconstructs the analog signal y(t). The output signal y(t) has half the amplitude of the input signal x(t).

Figure 1.12 shows some digital signals to demonstrate different graphical representations (see M-file 1.1). The upper part shows 8000 samples, the middle part the first 1000 samples and the lower part shows the first 100 samples out of a digital audio signal. Only if the number of samples inside a figure is sufficiently low, will the line with dot graphical representation be used for a digital signal.

% Author: U. Z¨olzer

M-file 1.1 (figure1\_03.m)

[x,FS,NBITS]=wavread(’ton2.wav’);

figure(1) subplot(3,1,1);

plot(0:7999,x(1:8000));ylabel(’x(n)’); subplot(3,1,2); plot(0:999,x(1:1000));ylabel(’x(n)’); subplot(3,1,3); stem(0:99,x(1:100),’.’);ylabel(’x(n)’);

xlabel(’n \rightarrow’);

0.5

0

 0.5 0 1000 2000 3000 4000 5000 6000 7000 8000

0.4

0.2

0

 0.2

 0.4

0.05

0 100 200 300 400 500 600 700 800 900 1000

0

 0.05

0 10 20 30 40 50 60 70 80 90 100

n 

Figure 1.12 Different time representations for digital audio signals.

Two different vertical scale formats for digital audio signals are shown in Figure 1.13. The

quantization of the amplitudes to fixed numbers in the range 1−6 32768 .. . 32767 is based on a

16-bit representation of the sample amplitudes which allows 2 quantized values in the range

215 ... 215 1. For a general w-bit representation the number range is 2w−1 ... 2w−1 1. This representation is called the integer number representation. If we divide all integer numbers by

the maximum absolute value, for example 32768, we come to the normalized vertical scale in Figure 1.13, which is in the range 1 .. . 1 Q, where Q is the quantization step size. It and can be calculated by Q 2−(w−1), which leads to Q 3.0518 10−5 for w 16. Figure 1.13 also displays the horizontal scale formats, namely the continuous-time axis, the discrete-time axis and the normalized discrete-time axis, which will be used normally. After this narrow description we

can define a digital signal as a discrete-time and discrete-amplitude signal, which is formed by sampling an analog signal and by quantization of the amplitude onto a fixed number of amplitude values. The digital signal is represented by a sequence of numbers x(n). Reconstruction of analog signals can be performed by DACs. Further details of ADCs and DACs and the related theory can be found in the literature. For our discussion of digital audio effects this short introduction to digital signals is sufficient.

Signal processing algorithms usually process signals by either block processing or sample- by-sample processing . Examples for digital audio effects are presented in [Arf98]. For block processing, samples are first collected in a memory buffer and then processed each time the buffer is completely filled with new data. Examples of such algorithms are fast Fourier transforms (FFTs) for spectra computations and fast convolution. In sample processing algorithms, each input sample is processed on a sample-by-sample basis.

Vertical axis format

32767

32766

Normalized vertical axis format

1

1

0 t Continuous

1 time axis

32767

32768

1

0 1 2 3 4 5 6 7 8 9 10 11

0 1 2 3 4 5 6 7 8 9 10 11

Discrete time axis

Normalized discrete time axis

Figure 1.13 Vertical and horizontal scale formats for digital audio signals.

A basic algorithm for weighting of a sound x(n) (see Figure 1.11) by a constant factor a demonstrates a sample-by-sample processing (see M-file 1.2). The input signal is represented by a vector of numbers [x(1), x(2),..., x(length(x))].

% Author: U. Z¨olzer

M-file 1.2 (sbs\_alg.m)

% Read input sound file into vector x(n) and sampling frequency FS [x,FS]=wavread(’input filename’);

% Sample-by sample algorithm y(n)=a\*x(n) for n=1:length(x),

y(n)=a \* x(n);

end;

% Write y(n) into output sound file with number of

% bits Nbits and sampling frequency FS wavwrite(y,FS,Nbits,’output filename’);

## 1.3.2. 数字信号的频谱分析

The spectrum of a signal shows the distribution of energy over the frequency range. The upper part of Figure 1.14 shows the spectrum of a short time slot of an analog audio signal. The frequencies range up to 20 kHz. The sampling and quantization of the analog signal with sampling frequency of fS 40 kHz leads to a corresponding digital signal. The spectrum of the digital signal of the same time slot is shown in the lower part of Figure 1.14. The sampling operation leads to a replication of the baseband spectrum of the analog signal [Orf96]. The frequency contents from 0 Hz up to 20 kHz of the analog signal now also appear from 40 kHz up to 60 kHz and the folded version of it from 40 kHz down to 20 kHz. The replication of this first image of the baseband spectrum at

40 kHz will now also appear at integer multiples of the sampling frequency of fS = 40 kHz. But

notice that the spectrum of the digital signal from 0 up to 20 kHz shows exactly the same shape as the spectrum of the analog signal. The reconstruction of the analog signal out of the digital signal is achieved by simply lowpass filtering the digital signal, rejecting frequencies higher than fS /2 20 kHz. If we consider the spectrum of the digital signal in the lower part of Fig. 1.14 and reject all frequencies higher than 20 kHz, we come back to the spectrum of the analog signal in the upper part of the figure.

0

20

40

60

80

0 4000 8000 12000 16000 20000

24000 28000 32000 36000 40000

f in Hz 

44000 48000 52000 56000 60000

0

20

40

60

80

0 4000 8000 12000 16000 20000

24000 28000 32000 36000 40000

f in Hz 

44000 48000 52000 56000 60000

Figure 1.14 Spectra of analog and digital signals.

Discrete Fourier transform

The spectrum of a digital signal can be computed by the discrete Fourier transform (DFT) which is given by

N −1

X(k) = DFT[x(n)] = x(n)e− k = 0, 1,...,N − 1. (1.1)

n=0

The fast version of the above formula is called the fast Fourier transform (FFT). The FFT takes N consecutive samples out of the signal x(n) and performs a mathematical operation to yield N samples X(k) of the spectrum of the signal. Figure 1.15 demonstrates the results of a 16-point FFT applied to 16 samples of a cosine signal. The result is normalized by N according to X=abs(fft(x,N))/N;.

The N samples X(k) XR(k) j XI (k) are complex-valued with a real part XR(k) and an

imaginary part XI (k) from which one can compute the absolute value

|X(k)|= ,X2 (k) + X2(k) k = 0, 1,...,N − 1 (1.2)

which is the magnitude spectrum, and the phase

ϕ(k) arctan XI (k)

XR(k)

k = 0, 1,...,N − 1 (1.3)

which is the phase spectrum. Figure 1.15 also shows that the FFT algorithm leads to N equidistant frequency points which give N samples of the spectrum of the signal starting from 0 Hz in steps

of fS up to N−1 fS . These frequency points are given by k fS , where k is running from 0, 1, 2,...,

N N N

N − 1. The magnitude spectrum |X(f)| is often plotted over a logarithmic amplitude scale

1

0.5

0

0.5

1

1

0 2 4 6 8 10

n 

12 14 16

0.5

0

0 2 4 6 8 10 12 14 16

k 

1

0.5

0

0 0.5 1 1.5 2 2.5 3 3.5

f in Hz 

x 104

Figure 1.15 Spectrum analysis with FFT algorithm: (a) digital cosine with N = 16 samples,

(b) magnitude spectrum |X(k)| with N = 16 frequency samples and (c) magnitude spectrum |X(f)|

from 0 Hz up to the sampling frequency fS = 40 000 Hz.

according to 20 log X(f ) which gives 0 dB for a sinusoid of maximum amplitude 1. This

normalization is equivalent to 20 log10 X(k) . Figure 1.16 shows this representation of the example from Fig. 1.15. Images of the baseband spectrum occur at the sampling frequency fS and multiples of fS . Therefore we see the original frequency at 5 kHz and in the first image spectrum the folded frequency fS − fcosine = 40 000 Hz − 5000 Hz = 35 000 Hz.

20

0

20

40

0 0.5 1 1.5 2 2.5 3 3.5

f in Hz 

x 104

Figure 1.16 Magnitude spectrum |X(f)| in dB from 0 Hz up to the sampling frequency

fS = 40000 Hz.

Inverse discrete Fourier transform (IDFT)

Whilst the DFT is used as the transform from the discrete-time domain to the discrete-frequency domain for spectrum analysis, the inverse discrete Fourier transform (IDFT) allows the transform from the discrete-frequency domain to the discrete-time domain. The IDFT algorithm is given by

N −1

x(n) IDFT[X(k)] X(k)e n 0, 1,...,N 1. (1.4)

N

k=0

The fast version of the IDFT is called the inverse Fast Fourier transform (IFFT). Taking N complex- valued numbers with the property X(k) X∗(N k) in the frequency domain and then performing the IFFT gives N discrete-time samples x(n), which are real-valued.

Frequency resolution: zero-padding and window functions

To increase the frequency resolution for spectrum analysis we simply take more samples for the FFT algorithm. Typical numbers for the FFT resolution are N 256, 512, 1024, 2048, 4096 and 8192. If we are only interested in computing the spectrum of 64 samples and would like to increase the frequency resolution from fS /64 to fS /1024, we have to extend the sequence of 64 audio samples by adding zero samples up to the length 1024 and then perform an 1024-point FFT. This technique is called zero-padding and is illustrated in Figure 1.17 and by M-file 1.3. The upper left

2.5

2

1.5

1

0.5

0

0.5

1

1.5

8 samples

0 2 4 6

8-point FFT

10

8

6

4

2

0

0 2 4 6

2.5

2

1.5

1

0.5

0

0.5

1

1.5

8 samples  zero-padding

0 5 10 15

n 

16-point FFT

10

8

6

4

2

0

0 5 10 15

k 

0 frequency in Hz fS

Figure 1.17 Zero-padding to increase frequency resolution.

part shows the original sequence of eight samples and the upper right part shows the corresponding eight-point FFT result. The lower left part illustrates the adding of eight zero samples to the origi- nal eight-sample sequence up to the length N 16. The lower right part illustrates the magnitude spectrum X(k) resulting from the 16-point FFT of the zero-padded sequence of length N 16. Notice the increase in frequency resolution between the eight-point and 16-point FFT. Between each frequency bin of the upper spectrum a new frequency bin in the lower spectrum is calculated. Bins k 0, 2, 4, 6, 8, 10, 12, 14 of the 16-point FFT correspond to bins k 0, 1, 2, 3, 4, 5, 6, 7 of the eight-point FFT. These N frequency bins cover the frequency range from 0 Hz up

to N−1 fS .

%Author: U. Z¨olzer

M-file 1.3 (figure1\_17.m)

x1=[-1 -0.5 1 2 2 1 0.5 -1]; x2(16)=0;

x2(1:8)=x1;

subplot(221);

stem(0:1:7,x1);axis([-0.5 7.5 -1.5 2.5]);

ylabel(’x(n) \rightarrow’);title(’8 samples’); subplot(222);

stem(0:1:7,abs(fft(x1)));axis([-0.5 7.5 -0.5 10]); ylabel(’|X(k)| \rightarrow’);title(’8-point FFT’);

subplot(223);

stem(0:1:15,x2);axis([-0.5 15.5 -1.5 2.5]);

xlabel(’n \rightarrow’);ylabel(’x(n) \rightarrow’); title(’8 samples+zero-padding’);

subplot(224);

stem(0:1:15,abs(fft(x2)));axis([-1 16 -0.5 10]); xlabel(’k \rightarrow’);ylabel(’|X(k)| \rightarrow’); title(’16-point FFT’);

The leakage effect occurs due to cutting out N samples from the signal. This effect is shown in the upper part of Figure 1.18 and demonstrated by the corresponding M-file 1.4. The cosine spectrum is smeared around the frequency. We can reduce the leakage effect by selecting a window function like Blackman window and Hamming window

wB (n) = 0.42 − 0.5 cos(2πn/N) + 0.08 cos(4πn/N), (1.5)

wH (n) = 0.54 − 0.46 cos(2πn/N) (1.6)

n = 0, 1,.. . N − 1.

and weighting the N audio samples by the window function. This weighting is performed according to xw w(n) x(n)/ k w(k) with 0 n N 1 and then an FFT of the weighted signal is performed. The cosine weighted by a window and the corresponding spectrum is shown in the

middle part of Figure 1.18. The lower part of Figure 1.18 shows a segment of an audio signal weighted by the Blackman window and the corresponding spectrum via a FFT. Figure 1.19 shows further simple examples for the reduction of the leakage effect and can be generated by the M-file 1.5.

1

0.5

0

0.5

1

4

2

0

 2

(a) Cosine signal x (n )

0 200 400 600 800 1000

(c) Cosine signal xw (n ) = x (n )  w (n) with window

0

 20

 40

 60

 80

0

 20

 40

 60

(b) Spectrum of cosine signal

0 2000 4000 6000 8000 10000

(d) Spectrum with Blackman window

 4

1

0.5

0

 0.5

1

0 2000 4000 6000 8000

(e) Audio signal xw (n) with window

 80

0

 20

 40

 60

 80

0 2000 4000 6000 8000 10000

(f) Spectrum with Blackman window

Figure 1.18 Spectrum analysis of digital signals: take N audio samples and perform an N point discrete Fourier transform to yield N samples of the spectrum of the signal starting from 0 Hz

over k fS where k is running from 0, 1, 2,...,N − 1. For (a) –(d), x(n) = cos(2 · π · 1 kHz · n).

%Author: U. Z¨olzer

M-file 1.4 (figure1\_18.m)

x=cos(2\*pi\*1000\*(0:1:N-1)/44100)’;

figure(2) W=blackman(N);

W=N\*W/sum(W); % scaling of window f=((0:N/2-1)/N)\*FS;

xw=x.\*W; subplot(3,2,1);plot(0:N-1,x);

axis([0 1000 -1.1 1.1]);

title(’a) Cosine signal x(n)’)

subplot(3,2,3);plot(0:N-1,xw);axis([0 8000 -4 4]); title(’c) Cosine signal x\_w(n)=x(n) \cdot w(n) with window’)

X=20\*log10(abs(fft(x,N))/(N/2));

subplot(3,2,2);plot(f,X(1:N/2)); axis([0 10000 -80 10]);

ylabel(’X(f)’);

title(’b) Spectrum of cosine signal’)

Xw=20\*log10(abs(fft(xw,N))/(N/2)); subplot(3,2,4);plot(f,Xw(1:N/2)); axis([0 10000 -80 10]);

ylabel(’X(f)’);

title(’d) Spectrum with Blackman window’)

s=u1(1:N).\*W;

subplot(3,2,5);plot(0:N-1,s);axis([0 8000 -1.1 1.1]); xlabel(’n \rightarrow’);

title(’e) Audio signal x\_w(n) with window’)

Sw=20\*log10(abs(fft(s,N))/(N/2));

subplot(3,2,6);plot(f,Sw(1:N/2)); axis([0 10000 -80 10]);

ylabel(’X(f)’);

title(’f) Spectrum with Blackman window’) xlabel(’f in Hz \rightarrow’);

M-file 1.5 (figure1\_19.m)

%Author: U. Z¨olzer

x=[-1 -0.5122 1 0.5 -1];

w=blackman(8); w=w\*8/sum(w); x1=x.\*w’;

x2(16)=0;

x2(1:8)=x1;

subplot(421);

stem(0:1:7,x);axis([-0.5 7.5 -1.5 2.5]);

ylabel(’x(n) \rightarrow’); title(’a) 8 samples’); subplot(423);

stem(0:1:7,w);axis([-0.5 7.5 -1.5 3]);

ylabel(’w(n) \rightarrow’);

title(’b) 8 samples Blackman window’);

subplot(425);

stem(0:1:7,x1);axis([-0.5 7.5 -1.5 6]);

ylabel(’x\_w(n) \rightarrow’); title(’c) x(n)\cdot w(n)’);

subplot(427);

stem(0:1:15,x2);axis([-0.5 15.5 -1.5 6]);

xlabel(’n \rightarrow’);ylabel(’x\_w(n) \rightarrow’); title(’d) x(n)\cdot w(n) + zero-padding’);

subplot(222);

stem(0:1:7,abs(fft(x1)));axis([-0.5 7.5 -0.5 15]); ylabel(’|X(k)| \rightarrow’);

title(’8-point FFT of c)’);

subplot(224);

stem(0:1:15,abs(fft(x2)));axis([-1 16 -0.5 15]); xlabel(’k \rightarrow’);ylabel(’|X(k)| \rightarrow’); title(’16-point FFT of d)’);

(a)

2

1

0

1

0 1 2 3 (b) 4 5 6 7

3

2

1

0

1

0 1 2 3 (c) 4 5 6 7

6

4

2

0

0 1 2 3 (d) 4 5 6 7

6

4

2

0

0 5 10 15

n 

8-point FFT of (c)

15

10

5

0

0 1 2 3 4 5 6 7

16-point FFT of (d)

15

10

5

0

0 5 10 15

k 

Figure 1.19 Reduction of the leakage effect by window functions: (a) the original signal,

(b) the Blackman window function of length N 8, (c) product x(n) wB (n) with 0 n N 1,

(d) zero-padding applied to x(n) wB (n) up to length N 16 and the corresponding spectra are shown on the right side.

Spectrogram: time-frequency representation

A special time-frequency representation is the spectrogram which gives an estimate of the short- time, time-localized frequency content of the signal. To obtain the spectrogram the signal is split into segments of length N, which are multiplied by a window and an FFT is performed (see Figure 1.20). To increase the time-localization of the short-time spectra an overlap of the weighted segments can be used. A special visual representation of the short-time spectra is the spectrogram in Figure 1.21. Time increases linearly across the horizontal axis and frequency increases across

the vertical axis. So each vertical line represents the absolute value |X(f)| over frequency by a

k k k

|X(k )| |X(k )| |X(k )|

DFT DFT DFT

N = 8 N = 8 N = 8

n

Figure 1.20 Short-time spectrum analysis by FFT.

Figure 1.21 Spectrogram via FFT of weighted segments.

Time

grey-scale value (see Figure 1.21). Only frequencies up to half the sampling frequency are shown. The calculation of the spectrogram from a signal can be performed by the MATLAB function B = specgram(x,nFFT,Fs,window,nOverlap).

Another time-frequency representation of the short-time Fourier transforms of a signal x(n) is the waterfall representation in Figure 1.22, which can be produced by M-file 1.6 that calls the waterfall computation algorithm given by M-file 1.7.

%Author: U. Z¨olzer

M-file 1.6 (figure1\_22.m)

[signal,FS,NBITS]=wavread(’ton2’); subplot(211);plot(signal); subplot(212);

waterfspec(signal,256,256,512,FS,20,-100);

M-file 1.7 (waterfspec.m)

function yy=waterfspec(signal,start,steps,N,fS,clippingpoint,baseplane)

% Authors: J. Schattschneider, U. Z¨olzer

% waterfspec( signal, start, steps, N, fS, clippingpoint, baseplane)

%

% shows short-time spectra of signal, starting

% at k=start, with increments of STEP with N-point FFT

% dynamic range from -baseplane in dB up to 20\*log(clippingpoint)

% in dB versus time axis echo off;

if nargin<7, baseplane=-100; end if nargin<6, clippingpoint=0; end if nargin<5, fS=48000; end

if nargin<4, N=1024; end % default FFT if nargin<3, steps=round(length(signal)/25); end if nargin<2, start=0; end

windoo=blackman(N); % window - default windoo=windoo\*N/sum(windoo); % scaling

% Calculation of number of spectra nos n=length(signal);

rest=n-start-N; nos=round(rest/steps);

if nos>rest/steps, nos=nos-1; end

% vectors for 3D representation x=linspace(0, fS/1000 ,N+1); z=x-x;

cup=z+clippingpoint; cdown=z+baseplane;

signal=signal+0.0000001;

% Computation of spectra and visual representation for i=1:1:nos,

spek1=20.\*log10(abs(fft(windoo.\*signal(1+start+....

....i\*steps:start+N+i\*steps)))./(N)/0.5); spek=[-200 ; spek1(1:N)]; spek=(spek>cup’).\*cup’+(spek<=cup’).\*spek; spek=(spek<cdown’).\*cdown’+(spek>=cdown’).\*spek; spek(1)=baseplane-10;

spek(N/2)=baseplane-10; y=x-x+(i-1);

if i==1,

p=plot3(x(1:N/2),y(1:N/2),spek(1:N/2),’k’); set(p,’Linewidth’,0.1);

end

pp=patch(x(1:N/2),y(1:N/2),spek(1:N/2),’w’,’Visible’,’on’); set(pp,’Linewidth’,0.1);

end; set(gca,’DrawMode’,’fast’);

axis([-0.3 fS/2000+0.3 0 nos baseplane-10 0]); set(gca,’Ydir’,’reverse’);

view(12,40);

Signal x (n )

1

0.5

0

0.5

1 0 1000 2000 3000 4000 5000 6000 7000 8000

n 

Waterfall representation of short-time FFTs

0

 50

0

1000

2000

3000

100

0 5

10 15 20

f in kHz 

4000 n

5000

6000

7000

Figure 1.22 Waterfall representation via FFT of weighted segments.

## 1.3.3. 数字系统

A digital system is represented by an algorithm which uses the input signal x(n) as a sequence (stream) of numbers and performs mathematical operations upon the input signal such as additions, multiplications and delay operations. The result of the algorithm is a sequence of numbers or the output signal y(n). Systems which do not change their behavior over time and fulfill the superposition property [Orf96] are called linear time-invariant (LTI) systems. The input/output relations for a LTI digital system describe time-domain relations which are based on the following terms and definitions:

• Unit impulse, impulse response and discrete convolution;

• Algorithms and signal flow graphs.

For each of these definitions an equivalent description in the frequency domain exists, which will be introduced later.

Unit impulse, Impulse response and Discrete convolution

• Test signal: a very useful test signal for digital systems is the unit impulse

1 for n = 0 0 for n /= 0,

which is equal to one for n = 0 and zero elsewhere (see Figure 1.23).

(1.7)

1 0 1 2 3

x (n ) = d(n)

n

y (n) = h (n )

n

1 0 1 2 3 4

Figure 1.23 Impulse response h(n) as a time-domain description of a digital system.

Impulse response: if we apply a unit-impulse function to a digital LTI system, the digital system will lead to an output signal y(n) h(n), which is called the impulse response h(n) of the digital system. The digital LTI system is completely described by the impulse response, which is pointed out by the label h(n) inside the box, as shown in Figure 1.23.

Discrete convolution: if we know the impulse response h(n) of a digital system, we can calculate the output signal y(n) from a freely chosen input signal x(n) by the discrete convolution formula given by

y(n) =

k=.−∞

x(k) · h(n − k) = x(n) ∗ h(n), (1.8)

which is often abbreviated by the second term y(n) x(n) h(n). This discrete sum for- mula (1.8) represents an input–output relation for a digital system in the time domain. The computation of the convolution sum formula (1.8) can be achieved by the MATLAB function y=conv(x,h).

Algorithms and signal flow graphs

The above given discrete convolution formula shows the mathematical operations which have to be performed to obtain the output signal y(n) for a given input signal x(n). In the following we will introduce a visual representation called a signal flow graph which represents the mathematical input/output relations in a graphical block diagram. We discuss some example algorithms to show that we only need three graphical representations for the multiplication of signals by coefficients, delay and summation of signals.

• A delay of the input signal by two sampling intervals is given by the algorithm

y(n) = x(n − 2) (1.9)

and is represented by the block diagram in Figure 1.24.

• A weighting of the input signal by a coefficient a is given by the algorithm

y(n) = a · x(n) (1.10)

and represented by a block diagram in Figure 1.25.

1 1

1 0 1 2

x (n)

n

Figure 1.24

y (n) = x (n 2)

Delay of the input signal.

n

1 0 1 2 3

1

n

1 0 1 2

a

x (n ) y (n) = a·x (n)

a

n

1 0 1 2

Figure 1.25 Weighting of the input signal.

• The addition of two input signals is given by the algorithm

y(n) = a1 · x1(n) + a2 · x2(n) (1.11) and represented by a block diagram in Figure 1.26.

The combination of the above algorithms leads to the weighted sum over several input samples, which is given by the algorithm

1 1 1

y(n) = 3 x(n) + 3 x(n − 1) + 3 x(n − 2) (1.12)

and represented by a block diagram in Fig. 1.27.

a .x (n )

1 0 1 2

1 1

n

y(n ) = a . x (n) + a .x (n)

n

1 0 1 2

1 1 2 2

n

1 0 1 2

Figure 1.26 Addition of two signals x1(n) and x2(n).

1

1 0 1 2

x (n)

n

x (n 1)

x (n 2)

\_1 3

1\_ x (n 1)  1\_ x (n 2)

1/3 1/3 1/3

3 3 n

1 0 1 2 3

Figure 1.27 Simple digital system.

Transfer function and frequency response

So far our description of digital systems has been based on the time-domain relationship between the input and output signals. We noticed that the input and output signals and the impulse response of the digital system are given in the discrete time domain. In a similar way to the frequency-domain

description of digital signals by their spectra given in the previous subsection we can have a frequency-domain description of the digital system which is represented by the impulse response h(n). The frequency-domain behavior of a digital system reflects its ability to pass, reject and enhance certain frequencies included in the input signal spectrum. The common terms for the frequency-domain behavior are the transfer function H (z) and the frequency response H (f ) of the digital system. Both can be obtained by two mathematical transforms applied to the impulse response h(n).

The first transform is the Z-Transform

X(z) =

n=.−∞

x(n) · z−n (1.13)

applied to the signal x(n) and the second transform is the discrete-time Fourier transform

∞

X(ejω) =

n=.−∞

x(n) · e−jωn, (1.14)

with ω = 2πf/fS (1.15)

applied to the signal x(n). Both are related by the substitution z ejω. If we apply the Z-transform to the impulse response h(n) of a digital system according to

∞

H (z) =

n=.−∞

h(n) · z−n (1.16)

we denote H (z) as the transfer function. The transfer function is of special interest as it relates the Z-transforms of input signal and output signal of the described system by

Y (z) = H (z) · X(z). (1.17)

If we apply the discrete-time Fourier transform to the impulse response h(n) we get

∞

H (ejω) =

n=.−∞

h(n) · e−jωn. (1.18)

Substituting (1.15) we define the frequency response of the digital system by

∞

Causal and stable systems

H (f ) =

n=.−∞

h(n) · e−j2πf/fSn. (1.19)

A realizable digital system has to fulfill the following two conditions:

Causality: a discrete-time system is causal , if the output signal y(n) 0 for n < 0 for a given input signal x(n) 0 for n < 0. This means that the system cannot react to an input before the input is applied to the system.

• Stability: a digital system is stable if

∞

n=.−∞

|h(n)| < M2 < ∞ (1.20)

holds. The sum over the absolute values of h(n) has to be less than a fixed number M2 < ∞.

The stability implies that the transfer function (Z-transform of impulse response) and the frequency response (discrete-time Fourier transform of impulse response) of a digital system are related by the substitution z ejω. Realizable digital systems have to be causal and stable systems. Some Z-transforms and their discrete-time Fourier transforms of a signal x(n) are given in Table 1.4.

Table 1.4 Z-transforms and discrete-time Fourier transforms of x(n).

Signal Z-transform Discrete-time

Fourier transform

x(n) X(z) X(ejω)

x(n − M) z−M · X(z) e−jωM · X(ejω) δ(n) 1 1

δ(n − M) z−M e−jωM

x(n) · ejω0 n X(e−jω0 · z) X(ej(ω−ω0 ))

IIR and FIR systems

IIR systems: A system with an infinite impulse response h(n) is called an IIR system. From the block diagram in Figure 1.28 we can read the difference equation

y(n) = x(n) − a1y(n − 1) − a2y(n − 2). (1.21)

The output signal y(n) is fed back through delay elements and a weighted sum of these delayed outputs is summed up to the input signal x(n). Such a feedback system is also called a recursive system. The Z-transform of (1.21) yields

Y (z) = X(z) − a1z−1Y (z) − a2z−2Y (z) (1.22)

X(z) = Y (z)(1 + a1z−1 + a2z−2) (1.23)

x (n ) y (n)

a1

a2

y (n 1) = xH1(n)

y (n2) = xH2(n)

Figure 1.28 Simple IIR system with input signal x(n) and output signal y(n).

and solving for Y (z)/X(z) gives transfer function

Y (z) 1

H (z) = X(z) = 1 + a1z−1 + a2z−2 . (1.24)

Figure 1.29 shows a special signal flow graph representation, where adders, multipliers and delay operators are replaced by weighted graphs.

x (n ) y (n)

z 1

y (n 1) = xH1(n)

X (z ) Y (z )

z 1

XH1(z ) = z 1Y(z )

a1

a2

z 1

) = xH2(n)

a1

a2

z 1

) = z 2Y(z )

Figure 1.29 Signal flow graph of digital system in Figure 1.28 with time-domain description in the left block diagram and corresponding frequency-domain description with Z-transform.

If the input delay line is extended up to N 1 delay elements and the output delay line up to

M delay elements according to Figure 1.30, we can write for the difference equation

M N −1

y(n) = − . ak y(n − k) + . bk x(n − k), (1.25)

k=1

the Z-transform of the difference equation

k=0

M N −1

Y (z) =− . ak z−kY (z) + . bk z−kX(z), (1.26)

x (n)

x (n 1)

x (n N 1)

b0 b1 b2

bN 2

bN 1

y (n)

aM

aM1

aM 2

a1

y (n M )

y (n 1)

Figure 1.30 IIR system.

and the resulting transfer function

H (z) =

N −1

k=0

bkz−k

. (1.27)

1 M

k=1

akz−k

The block processing approach for the IIR filter algorithm can be performed with the MATLAB/Octave function y = filter(b, a, x), where b and a are vectors with the filter coefficients as above and x contains the input signal.

A sample-by-sample processing approach for a second-order IIR filter algorithm is demon- strated by M-file 1.8.

% Author: U. Z¨olzer

M-file 1.8 (DirectForm01.m)

% Impulse response of 2nd order IIR filter

% Sample-by-sample algorithm clear

% Coefficients for a high-pass a=[1, -1.28, 0.47];

b=[0.69, -1.38, 0.69];

% Initialization of state variables xh1=0;xh2=0;

yh1=0;yh2=0;

% Input signal: unit impulse N=20; % length of input signal x(N)=0;x(1)=1;

% Sample-by-sample algorithm for n=1:N

y(n)=b(1)\*x(n)+b(2)\*xh1+b(3)\*xh2 - a(2)\*yh1 - a(3)\*yh2; xh2=xh1;xh1=x(n);

yh2=yh1;yh1=y(n); end;

% Plot results subplot(2,1,1)

stem(0:1:length(x)-1,x,’.’);axis([-0.6 length(x)-1 -1.2 1.2]); xlabel(’n \rightarrow’);ylabel(’x(n) \rightarrow’); subplot(2,1,2)

stem(0:1:length(x)-1,y,’.’);axis([-0.6 length(x)-1 -1.2 1.2]); xlabel(’n \rightarrow’);ylabel(’y(n) \rightarrow’);

Computation of the frequency response based on the coefficients of the transfer function

H (z) = B(z) can be made with the MATLAB/Octave function freqz(b, a), while the poles

and zeros can be determined with zplane(b, a).

FIR systems: A system with a finite impulse response h(n) is called an FIR system. From the block diagram in Figure 1.31 we can read the difference equation

y(n) = b0x(n) + b1x(n − 1) + b2x(n − 2). (1.28)

x (n ) y (n )

b0

x (n 1) = xH1(n )

b1

x (n 2) = xH2(n )

b2

Figure 1.31 Simple FIR system with input signal x(n) and output signal y(n).

The input signal x(n) is fed forward through delay elements and a weighted sum of these delayed inputs is summed up to the input signal y(n). Such a feed-forward system is also called a non-recursive system. The Z-transform of (1.28) yields

Y (z) = b0X(z) + b1z−1X(z) + b2z−2X(z) (1.29)

= X(z)(b0 + b1z−1 + b2z−2) (1.30)

and solving for Y (z)/X(z) gives transfer function

H (z) Y (z) b0 b1z−1 b2z−2. (1.31)

X(z)

A general FIR system in Figure 1.32 consists of a feed-forward delay line with N 1 delay elements and has the difference equation

N −1

y(n) = bk x(n − k). (1.32)

k=0

The finite impulse response is given by

N −1

h(n) = bk δ(n − k), (1.33)

k=0

x (n )

x (n 1)

x (n 2)

x (n N 1)

Figure 1.32 FIR system.

which shows that each impulse of h(n) is represented by a weighted and shifted unit impulse. The Z-transform of the impulse response leads to the transfer function

N −1

H (z) = bkz− . (1.34)

k=0

The time-domain algorithms for FIR systems are the same as those for IIR systems with the exception that the recursive part is missing. The previously introduced M-files for IIR systems can be used with the appropriate coefficients for FIR block processing or sample-by-sample processing.

The computation of the frequency response H (f ) H (f ) ej∠H(f ) ( H (f ) magnitude response, ϕ ∠H (f ) phase response) from the Z-transform of an FIR impulse response according to (1.34) is shown in Figure 1.33 and is calculated by the following M-file 1.9.

0.3

0.2

0.1

0

0.1

(a) Impulse Response h (n)

0.7

0.6

0.5

0.4

0.3

0.2

0.1

0

(b) Magnitude Response |H (f )|

1

0.5

0

0.5

1

1 0 1 2 3 4 5

n 

(c) Pole/Zero plot

0

0.5

1

1.5

0 10 20 30 40

f in kHz 

(d) Phase Response  H (f )

1 0 1 2

Re(z )

2 0 10 20 30 40

f in kHz 

Figure 1.33 FIR system: (a) impulse response, (b) magnitude response, (c) pole/zero plot and

(d) phase response (sampling frequency fS = 40 kHz).

M-file 1.9 (figure1\_33.m)

function magphasresponse(h)

% Author: U. Z¨olzer FS=40000;

fosi=10;

if nargin==0

h=[-.1 .15 .3 .15 -.1];

end hmax=max(h); hmin=min(h); dh=hmax-hmin;

hmax=hmax+.1\*dh; hmin=hmin-.1\*dh;

N=length(h);

% denominator polynomial:

a=zeros(1,N);

a(1)=1;

subplot(221) stem(0:N-1,h)

axis([-1 N, hmin hmax])

title(’a) Impulse Response h(n)’,’Fontsize’,fosi); xlabel(’n \rightarrow’,’Fontsize’,fosi)

grid on;

subplot(223) zplane(h,a)

title(’c) Pole/Zero plot’,’Fontsize’,fosi); xlabel(’Re(z)’,’Fontsize’,fosi)

ylabel(’Im(z)’,’Fontsize’,fosi)

subplot(222) [H,F]=freqz(h,a,1024,’whole’,FS); plot(F/1000,abs(H))

xlabel(’f in kHz \rightarrow’,’Fontsize’,fosi); ylabel(’|H(f)| \rightarrow’,’Fontsize’,fosi); title(’b) Magnitude response |H(f)|’,’Fontsize’,fosi);

grid on;

subplot(224) plot(F/1000,unwrap(angle(H))/pi)

xlabel(’f in kHz \rightarrow’,’Fontsize’,fosi) ylabel(’\angle H(f)/\pi \rightarrow’,’Fontsize’,fosi) title(’d) Phase Response \angle H(f)’,’Fontsize’,fosi);

grid on;

# 1.4 总结

In this first chapter we introduced definitions and classifications of audio effects, to provide an overview of the territory to be explored. Then, some basic concepts of digital signals, their spectra and digital systems have been introduced. The description is intended for persons with little or no knowledge of digital signal processing. The inclusion of MATLAB M-files for all stages of processing may serve as a basis for further programming in the following chapters. As well as showing simple tools for graphical representations of digital audio signals we have calculated the spectrum of a signal x(n) by the use of the FFT M-file

Xmagnitude=abs(fft(x)) Xphase=angle(fft(x)).

Time-domain processing for DAFX can be performed by block-based input–output computations which are based on the convolution formula (if the impulse response of a system is known) or difference equations (if the coefficients a and b are known). The computations can be done by the following M-files:

• y=conv(h,x) %length of output signal l\_y = l\_h +l\_x -1 y=filter(b,a,x) %l\_y = l\_x

These M-files deliver an output vector containing the output signal y(n) in a vector of correspond- ing length. Of course, these block processing algorithms perform their inner computations on a sample-by-sample basis. Therefore, we have also shown an example for the sample-by-sample programming technique, which can be modified according to different applications:

• y=dafxalgorithm(parameters,x)

% Sample-by sample algorithm y(n)=function(parameters,x(n)) for n=1:length(x),

y(n)=....do something algorithm with x(n) and parameters; end;

That is all we need for DAFX exploration and programming, good luck!

# 参考文献

[ABL+03] X. Amatriain, J. Bonada, A. Loscos, J. L. Arcos and V. Verfaille. Content-based transformations.

J. New Music Research, 32(1): 95– 114, 2003.

[AD98] D. Arfib and N. Delprat. Selective transformations of sound using time-frequency representations: An application to the vibrato modification. In 104th Conv. Audio Eng. Soc., Amsterdam, 1998.

[AKZ02a] D. Arfib, F. Keiler and U. Zo¨lzer. DAFX–Digital Audio Effects , First edition, Time-Frequency Processing, pp. 237– 97. U. Zo¨lzer ed., J. Wiley & Sons, Ltd, 2002.

[AKZ02b] D. Arfib, F. Keiler and U. Zo¨lzer. DAFX - Digital Audio Effects , First edition, Source-filter pro- cessing, pp. 299– 372. U. Zo¨lzer ed., J. Wiley & Sons, Ltd, 2002.

[All77] J. B. Allen. Short term spectral analysis, synthesis and modification by discrete fourier transform.

IEEE Trans. on Acoustics, Speech, and Signal Processing , 25(3): 235– 8, 1977.

[ANS60] ANSI. USA Standard Acoustic Terminology . American National Standards Institute, 1960. [AOPW99] S. Abrams, D. V. Oppenheim, D. Pazel and J. Wright. Higher-level composition control in lusic

sketcher: Modifiers and smart harmony. In Proc. Int. Computer Music Conf. (ICMC’99), Beijing, pp. 13– 6, 1999.

[AR77] J. B. Allen and L. R. Rabiner. A unified approach to short-time fourier analysis and synthesis.

Proc. IEEE , 65(11): 1558– 64, 1977.

[Arf98] D. Arfib. Different ways to write digital audio effects programs. In Proc. DAFX-98 Digital Audio Effects Workshop, pp. 188– 191, Barcelona, November 1998.

[Arf99] D. Arfib. Visual representations for digital audio effects and their control. In Proc. DAFX-99 Digital Audio Effects Workshop, pp. 63– 68, Trondheim, December 1999.

[Bar70] B. Bartlett. A scientific explanation of phasing (flanging). J. Audio Eng. Soc., 18(6): 674– 5, 1970. [BB96] T. Buzan and B. Buzan. Mind Map Book . Plume, 1996.

[BJ95] R. Bristow-Johnson. A detailed analysis of time-domain formant-corrected pitch-shifting algorithm.

J. Audio Eng. Soc., 43(5): 340– 52, 1995.

[Bla83] J. Blauert. Spatial Hearing: the Psychophysics of Human Sound Localization. MIT Press, 1983. [Ble01] B. Blesser. An interdisciplinary synthesis of reverberation viewpoints. J. Audio Eng. Soc., 49(10):

867– 903, 2001.

[BP89] J. C. Brown and M S. Puckette. Calculation of a narrowed autocorrelation function. J. Ac. Soc. of America, 85: 1595– 601, 1989.

[Cab99] D. Cabrera. PsySound: a computer program for psychoacoustical analysis. In Proc. Australian Ac.

Soc. Conf., Melbourne, pp. 47– 53, November 1999.

[Cho71] J. Chowning. The simulation of moving sound sources. J. Audio Eng. Soc., 19(1): 1– 6, 1971.

[CKC+04] P. Cano, M. Koppenberger, O. Celma, P. Herrera and V. Tarasov. Sound effects taxonomy man- agement in production environments. In Int. Conf. Audio Eng. Soc., London UK, 2004.

[CPR95] A. Camurri, G. De Poli and D. Rocchesso. A taxonomy for sound and music computing. Computer Music J ., 19(2): 4– 5, 1995.

[Dat97] J. Dattoro. Effect design, part 2: Delay-line modulation and chorus. J. Audio Eng. Soc., pp. 764– 88, 1997.

[dC04] A. de Cheveigne´. Pitch, Pitch perception models. C. Plack and A. Oxenham eds, Springer-Verlag, Berlin, 2004.

[DH92] P. Desain and H. Honing. Music, Mind and Machine: Studies in Computer Music, Music Cognition, and Artificial Intelligence. Thesis Publishers, 1992.

[Dol86] M. Dolson. The phase vocoder: a tutorial. Computer Music J ., 10(4): 14– 27, 1986.

[DT96] S. Dubnov and N. Tishby. Testing for gaussianity and non linearity in the sustained portion of musical sounds. In Proc. Journe´es Informatique Musicale (JIM’96), 1996.

[Dut91] P. Dutilleux. Vers la machine a` sculpter le son, modification en temps-re´el des caracte´ristiques fre´quentielles et temporelles des sons . PhD thesis, University of Aix-Marseille II, 1991.

[Fav01] E. Favreau. Phase vocoder applications in GRM tools environment. In Proc. of the COST-G6 Work- shop on Digital Audio Effects (DAFx-01), Limerick , pp. 134– 7, 2001.

[Gav93] W. W. Gaver. What in the world do we hear? An ecological approach to auditory event perception.

Ecological Psychology , 5(1): 1– 29, 1993.

[Ger85] M. A. Gerzon. Ambisonics in multichannel broadcasting and video. J. Audio Eng. Soc., 33(11), 1985.

[GKP+ 05] C. Guastavino, B. F. Katz, J.-D. Polack, D. J. Levitin and D. Dubois. Ecological validity of sound-

scape reproduction. Acta Acust. United Ac., 91(2): 333– 341, 2005.

[Har78] W. M. Hartmann. Flanging and phasers. J. Audio Eng. Soc., 26: 439– 43, 1978. [Hay96] S. Haykin. Adaptive Filter Theory , Third edition Prentice Hall, 1996.

[HMM04] D. Hargreaves, D. Miell and R. MacDonald. What do we mean by musical communication, and why it is important?, introduction of “Musical communication (part 1)” session, ICMPC CD-ROM. In Proc. Int. Conf. Music Perc. and Cog ., 391– 394, 2004.

[Hon95] H. Honing. The vibrato problem, comparing two solutions. Computer Music J ., 19(3): 32– 49, 1995.

[Lar98] J. Laroche. Time and pitch scale modification of audio signals. In M. Kahrs and K. Brandenburg, eds, Applications of Digital Signal Processing to Audio & Acoustics, pp. 279– 309. Kluwer Academic Publishers, 1998.

[Lar01] J. Laroche. Estimating tempo, swing and beat locations in audio recordings. In Proc. IEEE Workshop on Applications of Digital Signal Processing to Audio and Acoustics , pp. 135– 8, 2001.

[LD97] J. Laroche and M. Dolson. About this phasiness business. In Proc. Int. Computer Music Conf. (ICMC’97), Thessaloniki , pp. 55– 8, 1997.

[LVKL96] T. I. Laakso, V. Va¨lima¨ki, M. Karjalainen and U. K. Laine. Splitting the unit delay. In IEEE Signal Proc. Mag ., pp. 30– 60, 1996.

[MB90] R. C. Maher and J. Beauchamp. An investigation of vocal vibrato for synthesis. Appl. Acoust ., 30: 219– 45, 1990.

[MB96] P. Masri and A. Bateman. Improved modelling of attack transients in music analysis-resynthesis.

In Proc. Int. Computer Music Conf. (ICMC’96), Hong Kong , pp. 100– 3, 1996.

[MC90] E. Moulines and F. Charpentier. Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Com., 9(5/6): 453– 67, 1990.

[ME93] C. Marvin and G. Ewers. A Simple Approach to Digital Signal Processing . Texas Instruments, 1993. [Mit01] S.K Mitra. Digital Signal Processing–A Computer-Based Approach, Second edition. McGraw-Hill,

2001.

[MK73] M. Mathews and J. Kohut. Electronic simulation of violin resonances. J. Ac. Soc. of America, 53(6): 1620– 6, 1973.

[ML95] E. Moulines and J. Laroche. Non-parametric technique for pitch-scale and time-scale modification.

Speech Com., 16: 175– 205, 1995.

[MM01] A. Marui and W. L. Martens. Perceptual and semantic scaling for user-centered control over distortion-based guitar effects. In 110th Conv. Audio Eng. Soc., Paris, France, 2001. Preprint 5387.

[Mol75] J. Molino. Fait musical et se´miologie de la musique. Musique en Jeu, 17: 37– 62, 1975. [Moo79] J. A. Moorer. About this reverberation business. Computer Music J ., 3(2): 13– 8, 1979.

[Moo90] F. R. Moore. Elements of Computer Music. University of California, San Diego, Prentice Hall Inc., 1990.

[MQ86] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation.

IEEE Trans. Acoust., Speech Signal Proc., 34(4): 744– 54, 1986.

[MSY98] J. McClellan, R. Schafer and M. Yoher. DSP FIRST: A Multimedia Approach. Prentice-Hall, 1998.

[MWdSK95] S. McAdams, S. Winsberg, G. de Soete and J. Krimphoff. Perceptual scaling of synthesized musical timbres: common dimensions, specificities, and latent subject classes. Psychol. Res., 58: 177– 92, 1995.

[Nat75] J.-J. Nattiez. Fondements d’une Se´miologie de la Musique. U. G. E., Coll. 10/18, Paris, 1975. [Orf96] S.J. Orfanidis. Introduction to Signal Processing . Prentice-Hall, 1996.

[Por76] M. Portnoff. Implementation of the digital phase vocoder using the fast Fourier transform. IEEE Trans. Acoust. Speech Signal Proc., 24(3): 243– 8, 1976.

[PPPR96] G. De Poli, A. Picialli, S. T. Pope, and C. Roads (eds). Musical Signal Processing . Swets & Zeitlinger, 1996.

[Puc95] M. S. Puckette. Phase-locked vocoder. In Proc. IEEE ASSP Conf. on Appl. of Signal Proc. Audio Acoust. (Mohonk, NY), 1995.

[Pul97] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. J. Audio Eng.

Soc., 45(6): 456– 66, 1997.

[Rab] C. A. Rabasso´. L’improvisation: du langage musical au langage litte´raire. Intemporel: Bulletin de la Socie´te´ Nationale de Musique, 15.

[RDS+99] S. Rossignol, P. Depalle, J. Soumagne, X. Rodet and J.-L. Collette. Vibrato: Detection, estima-

tion, extraction, modification. In Proc. COST-G6 Workshop on Digital Audio Effects (DAFx-99), Trondheim, 1999.

[Roa96] C. Roads. The Computer Music Tutorial . MIT Press, 1996.

[RW99] J.-C. Risset and D. L. Wessel. Exploration of Timbre by Analysis and Synthesis , pp. 113– 69.

D. Deutsch, Academic Press, 1999.

[Sch66] P. Schaeffer. Traite´ des Objets Musicaux . Seuil, 1966.

[Sch77] R. M. Schafer. The Tuning of the World . Knopf: New York, 1977.

[Sea36] C. E. Seashore. Psychology of the vibrato in voice and speech. Studies Psychol. Music, 3, 1936. [She82] R. Shepard. Geometrical approximations to the structure of musical pitch. Psychol. Rev ., 89(4):

305– 33, 1982.

[SL61] M. R. Schrœder and B. Logan. “Colorless” artificial reverberation. J. Audio Eng. Soc., 9: 192– 7, 1961.

[Sla85] W. Slawson. Sound Color . University of California Press, 1985.

[Smi84] J. O. Smith. An allpass approach to digital phasing and flanging. In Proc. Int. Computer Music Conf. (ICMC’84), Paris, pp. 103– 8, 1984.

[SS90] X. Serra and J. O. Smith. A sound decomposition system based on a deterministic plus residual model. J. Ac. Soc. of America, Sup. 1 , 89(1): 425– 34, 1990.

[SSAB02] J. O. Smith, S. Serafin, J. Abel and D. Berners. Doppler simulation and the Leslie. In Proc. Int.

Conf. on Digital Audio Effects (DAFx-02), Hamburg , pp. 13– 20, 2002.

[Sun87] J. Sundberg. The Science of the Singing Voice. Northern Illinois University Press, 1987.

[VGD05] V. Verfaille, C. Guastavino and P. Depalle. Perceptual evaluation of vibrato models. In Colloq.

Interdisc. Musicol., Montre´al (CIM’05), 2005.

[VGT06] Vincent Verfaille, Catherine Guastavino and Caroline Traube. An interdisciplinary approach to audio effect classification. In Proc. 9th Int. Conf. Digital Audio Effects (DAFx-06), Montreal, Canada, pp. 107– 13, 2006.

[VLM97] T. Verma, S. Levine and T. Meng. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In Proc. Int. Computer Music Conf. (ICMC’97), Thessaloniki , 164– 167, 1997.

[VWD06] V. Verfaille, M. M. Wanderley and Ph. Depalle. Mapping strategies for gestural control of adaptive digital audio effects. J. New Music Res., 35(1): 71– 93, 2006.

[VZA06] Vincent Verfaille, U. Zo¨lzer and Daniel Arfib. Adaptive digital audio effects (A-DAFx): A new class of sound transformations. IEEE Trans. Audio, Speech and Lang. Proc., 14(5): 1817– 1831, 2006.

[WM01] O. Warusfel and N. Misdariis. Directivity synthesis with a 3D array of loudspeakers - Application for stage performance. In Proc. of the COST-G6 Workshop on Digital Audio Effects (DAFx-01), Limerick , pp. 232– 6, 2001.

[Zo¨l02] U. Zo¨lzer (ed). DAFX–Digital Audio Effects , First edition. J. Wiley & Sons, Ltd, 2002.

[Zo¨l05] U. Zo¨lzer. Digital Audio Signal Processing , Second edition. John Wiley & Sons, Ltd, 2005. [ZS65] E. Zwicker and B. Scharf. A model of loudness summation. Psychol. Rev ., 72: 3– 26, 1965. [Zwi77] E. Zwicker. Procedure for calculating loudness of temporally variable sounds. J. Ac. Soc. of America,

62(3): 675– 82, 1977.