

论文题目 钢琴音色识别与电子合成系统的设计与实现

---

专业学位类别 工 程 硕 士

---

学 号 201291221016

---

作 者 姓 名 程美芳

---

指 导 教 师 程红蓉 副教授

---



分类号\_\_\_\_\_密级\_\_\_\_\_

UDC<sup>注1</sup>\_\_\_\_\_

# 学 位 论 文

钢琴音色识别与电子合成系统的设计与实现

(题名和副题名)

程美芳

(作者姓名)

指导教师

程红蓉

副教授

电子科技大学

成 都

周泽彬

高 工

江联重工股份有限公司

南 昌

(姓名、职称、单位名称)

申请学位级别 硕士

专业学位类别 工程硕士

工程领域名称

软 件 工 程

提交论文日期 2014.03

论文答辩日期 2014.05.16

学位授予单位和日期

电子科技大学

2014 年 6 月 29 日

答辩委员会主席

评阅人

注 1: 注明《国际十进分类法 UDC》的类号。



# **DESIGN AND REALIZATION OF PIANO TIMBRE RECOGNITION AND ELECTRONIC SYNTHESIS SYSTEM**

A Master Thesis Submitted to  
University of Electronic Science and Technology of China

Major: **Master of Engineering**

Author: **Cheng Meifang**

Advisor: **Cheng Hongrong**

School : **School of Information and Software Engineering**



## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：\_\_\_\_\_ 日期：\_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日

## 论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：\_\_\_\_\_ 导师签名：\_\_\_\_\_

日期：\_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日





## 摘要

乐器仿真是跨软件工程和音乐理论双领域的交叉学科研究方向，长期以来，受到专业知识的限制，音乐专业人士不能采用软件工程及数学方法对乐器声学原理进行深入的剖析，而软件工程人员对音乐知识了解相对较少，因此涉及音乐的软件工程研究一直相对薄弱。实际上，在现代信号处理、模式识别的技术支撑下，可以揭示音乐背后的数学秘密，提高人类对音乐的理解。软件工程技术在音乐领域的应用，将给人类带来全新的听觉感受，符合国家近期提出的文化强国理念，在人文、科技领域都具有研究意义。

本论文正是把软件工程和音乐理论相结合，以电子合成钢琴音色为目的，在剖析“十二平均律”、“钢琴弦振动方程”的数学基础上，对钢琴音色识别与电子合成方法进行了详尽的讨论和研究。主要内容为：

1. 对比了软件工程中的生命周期法、快速原型法、面向对象法，认为只有快速原型法才能够降低技术风险，缩短工期。选择 MATLAB 语言来构建快速原型，用 C#.net 来开发成品软件。

2. 应用短时傅里叶变换的频谱分析方法，提取到钢琴音色特征矩阵，研究发现弦类乐器音色特征矩阵不同于管类、打击类乐器；接着对音强衰减变化和时间的关系进行分析，获得了音强包络曲线，发现钢琴作为一种击弦类乐器，音强衰减曲线不同于弹拨弦、拉弦类乐器。利用音色特征矩阵和音强包络曲线可以区分钢琴和其它乐器的差异，从而实现钢琴乐音的识别。

3. 本软件对钢琴音色特征矩阵进行矩阵计算，实现了钢琴音色中的 25 次倍频谐波的电子合成，音色仿真效果优于市面上采用“柯西函数五次倍频”原理的电子琴。同时引入了音强包络曲线，使仿真效果更加接近于击弦类乐器。

4. 设计了三层架构的中间件发音技术，成品软件在发音时无需重新进行矩阵运算，降低了 CPU 负荷，提高了发音效率，具有音色纯净、音强稳定、声音文件体积小的优点。成品软件中不含数学模型和发音参数，即使成品软件被逆向工程，也不会泄漏研究成果。

**关键词：**机器听觉，傅里叶变换，频谱分析，钢琴音色合成

## ABSTRACT

Simulation of musical instruments is a cross disciplinary in both the fields of cross software engineering and music theory. In the past, due to the lack of professional knowledge, music professionals are unable to apply software engineering and mathematical methods to analyze musical acoustics principles at an in-depth level. On the other hand, software engineering researchers generally have very limited knowledge about music, and therefore, the study of software engineering related to music has been relatively weak all time through. In fact, under the technical support of signal processing and pattern recognition, it is now possible to reveal the mathematical secrets behind music, and thus improve our understanding of music. Applications of the software engineering technology in the field of music will bring new experience of hearing to human. It is in line with the idea of "Development of Cultural Power" proposed by the Chinese government, having great significance in the cultural and academic fields.

The purpose of this study is to synthesize the piano timbre electrically by integrating software engineering and music theories. On the mathematical basis of the "Twelve-tone Equal Temperament" and the "Piano String Vibration Equation", this study conducted in-depth discussions and research about the piano timbre recognition and electronic synthesis methods. The main contents include:

1. Based on the comparison among the life cycle method, the rapid prototype method and the object oriented method in software engineering, this study suggests that only the rapid prototype method may minimize the technical risk and shorten the construction period. MATLAB was used to build the rapid prototype and C#.net was used to develop the software product.

2. The spectrum analysis method of short-time Fourier transform was applied to extract the characteristic matrix of piano timbre. The findings show that the timbre characteristic matrix of string instruments is different from that of tube or percussion instruments. Then, the relationship between the intensity attenuation and time was analyzed to obtain the amplitude envelope curve. It suggests that, as a hammered string instrument, the intensity attenuation curve of piano is different from that of the plucked string instruments and spring instruments. The timbre characteristic matrix and the

amplitude envelope curve can be used to distinguish the differences between piano and other musical instruments, so as to realize the recognition of piano sound.

3. This software performed matrix calculation on the timbre characteristic matrix of piano, and realized electronic synthesis of 25 harmonics for the piano timbre. The eventual simulation effect is better than the electronic organ based on "Cauchy function of the 5 harmonics" principle. Meanwhile, with the introduction of the amplitude envelope curve, the simulation effect is more close to real hammered string instruments.

4. This software designed the middleware sound production technology with a three-layer structure. The finished software does not need to re-perform matrix operations for sound production, which reduces the load of CPU and improves the production efficiency. Thus, it is featured with pure timbre, stable amplitude, and small file size. The finished software does not contain any mathematical models or vocal parameters, so even if reverse engineering is conducted, the research achievements will not be disclosed.

**Keywords::** machine hearing, Fourier transform, spectrum analysis, piano timbre synthesis

## 目 录

第一章 绪 论 .....	1
1.1 研究背景与意义 .....	1
1.2 国内外研究现状及趋势 .....	2
1.2.1 机器听觉中的频谱特征检测方法 .....	2
1.2.2 机器听觉中的音色特征识别方法 .....	4
1.2.3 机器听觉中的乐曲识别分类方法 .....	6
1.3 论文的主要内容和组织结构 .....	8
第二章 相关理论与技术 .....	10
2.1 钢琴的物理结构分析 .....	10
2.2 钢琴乐音的音高 .....	12
2.2.1 十二平均律简介 .....	12
2.2.2 十二平均律的数学原理 .....	13
2.2.3 钢琴键盘的音高定义 .....	15
2.3 钢琴乐音的音长 .....	15
2.4 钢琴乐音的音强 .....	16
2.5 钢琴乐音的音色 .....	17
2.5.1 纯音与复音的音色区别 .....	18
2.5.2 钢琴弦的振动方程 .....	18
2.5.3 钢琴弦振动的泛音特征分析 .....	20
2.6 小结 .....	20
第三章 制定开发方案 .....	22
3.1 软件开发方法选优 .....	22
3.1.1 面向对象法 .....	22
3.1.2 生命周期法 .....	23
3.1.3 快速原型法 .....	23
3.2 软件开发目标 .....	24
3.3 制定技术路线 .....	24
3.4 小结 .....	26
第四章 钢琴乐音特征的数字化识别 .....	27
4.1 基于短时傅里叶变换的频谱分析 .....	27

4.2 音色特征矩阵的提取.....	35
4.3 音强包络函数的提取.....	39
4.4 乐音特征数字化识别.....	41
4.5 小结.....	42
<b>第五章 基于 MATLAB 的钢琴乐音仿真.....</b>	<b>43</b>
5.1 正弦波的电子合成.....	43
5.2 包络函数的应用.....	44
5.3 钢琴乐音的电子合成.....	51
5.4 快速原型的测试工作.....	53
5.5 小结.....	55
<b>第六章 基于 C#.NET 的钢琴演奏软件的开发.....</b>	<b>56</b>
6.1 钢琴音色合成方案整体设计.....	56
6.2 中间件音色合成方案的优点.....	57
6.3 成品软件的开发.....	58
6.3.1 发音函数的开发.....	58
6.3.2 键盘响应函数的开发.....	59
6.3.3 虚拟键盘击打动画的开发.....	61
6.3.4 键盘与琴键联动的开发.....	62
6.4 成品软件的测试工作.....	64
6.5 小结.....	65
<b>第七章 总结与展望.....</b>	<b>66</b>
7.1 总结.....	66
7.2 展望.....	66
<b>致 谢.....</b>	<b>68</b>
<b>参考文献.....</b>	<b>69</b>



## 第一章 绪 论

钢琴音色电子识别与电子合成是跨软件工程和音乐理论双领域的交叉学科研究方向，在软件工程领域涉及到信号处理、矩阵计算、模式识别和信号仿真等知识，在音乐理论方面涉及到基本乐理、十二平均率、钢琴结构、钢琴弦振动方程等知识。长期以来，受到专业知识的限制，音乐专业人士不能采用软件工程理论及数学方法对音乐领域中的声学原理进行深入的剖析，而软件工程人员对音乐知识了解相对较少，因此涉及音乐的软件工程研究一直相对薄弱。实际上，在现代先进的信号处理、模式识别技术的帮助下，人们可以透过音乐表象看到音乐本质，提高人们对音乐的理解<sup>[1]</sup>。

### 1.1 研究背景与意义

当前机器听觉的学术研究包括三大研究领域：音频信号记录与分析、语音信号处理与识别、音乐信号识别和仿真。

其中，音频信号记录与分析的研究进行较早，1877 年，托马斯·阿尔瓦·爱迪生发明了留声机，开创了机器记录音频信号的新时代。1584 年，中国明代音乐家朱载堉在《律学新说》一书中提出“新法密率”，推算出以比率“二的十二分之一次方”将八度音划分十二等分的算法，即“十二平均律”，同时朱载堉制造了新法密率律管及新法密率弦乐器，是世界上最早的十二平均律乐器。十二平均律实现了对乐器的校音和改良，在演奏时，利用十二平均律改良的乐器，可以很方便地实现转调，乐器能够和其它乐器协同演奏，十二平均律的对乐器的改良，使乐器的表现力大为增强<sup>[2]</sup>。

语音信号处理与识别也是机器听觉的一个传统的研究热点，语音识别技术和语音合成技术实现人类和机器的语音交流，这两项技术使机器首次拥有了听、讲的语音能力。语音识别技术所涉及的技术有：听觉器官工作原理、发声器官工作原理、振动波信号处理、统计学、概率论、人工智能模式识别等等。语音合成技术以 TTS(Text To Speech)技术为代表，能够将 Text 格式的文字转换为人类可以接受的口语表达。语音信号处理与识别已经进入了初步的商业应用，苹果公司的 siri 语音机器人在强大的云计算技术的支撑下，以实现了和人类无障碍的语音交流。国内的科大讯飞、百度语音、搜狗语音输入法、紫冬口译等系统也都实现了产业化，为现代智能手机提供了语音识读能力，使智能手机成为人类的工作助手。

随着信息科学技术的迅速发展, 音乐信号识别和仿真也逐渐成为机器听觉领域新的研究热点, 但是国内外研究相对较少, 采用音乐信号识别和仿真理论研制的产品也较少。双排键电子琴是这一领域的最高端商业化产品的代表, 双排键电子琴能够模仿真实乐器的音色, 从而实现“一个人的乐队”, 其工作原理为商业机密, 受到专利保护。本论文的研究也属于音乐信号识别和仿真的学术领域, 将实现从纷杂的环境声音中找到钢琴的乐音信号, 解析钢琴的频谱特征, 实现钢琴乐音信号的电子合成。将利用音乐信号识别时提取的音色特征, 控制扬声器发出和钢琴音色完全相似的声音<sup>[3]</sup>。

音乐信号识别和仿真的研究能推动传统乐器音色改良、电子琴研制、自动作曲机器人、机器人乐队等技术发展。计算机技术在音乐领域的应用, 将给人类带来全新的音乐享受, 符合国家近期提出文化强国理念, 在人文、科技领域都具有重大的研究意义。

## 1.2 国内外研究现状及趋势

20 世纪 80 年代末, 日本率先在机器听觉上获得了进展, 大阪大学建立了一个机器听觉系统<sup>[4]</sup>, 能够仿照人耳工作机理, 实现从环境噪音中感知音乐, 系统功能较为简单。1993 年, Hawley<sup>[5]</sup>发表了一系列研究成果, 较早涉及到了计算机听觉领域。Douglas Nunn<sup>[6]</sup>进行了机器听觉的研究, 发表论文介绍机器听觉系统的识别原理, 应用了对声音信号启发式处理方法, 能初步理解声音信号。东京大学 Kashino<sup>[7]</sup>分析了人类听觉神经工作机制, 研发了机器听觉的新技术, 开发了一个机器听觉系统。1995 年, Kashino 又对研究发的机器听觉系统加以改进<sup>[8]</sup>, 获得了更好的识别效果。麻省理工学院的 Keith Martin<sup>[9]</sup>对多声部钢琴乐曲进行了一系列的研究, 开发了一个机器听觉系统, 能够识别多声部钢琴乐曲。后来 Davies<sup>[10]</sup>建立了一个机器听觉系统, 和 Martin 研发的机器听觉系统功能类似, 实验效果相近。

在机器听觉的研究过程中, 乐器演奏的音色信号的频谱分析一直是研究热点, 这需要选用频谱特征提取技术, 不同的技术应用领域和分析效果不尽相同。

### 1.2.1 机器听觉中的频谱特征检测方法

#### 1. 时域检测法

时域(Time Domain)是描述数学函数值或物理信号值对应时间变化的关系。例如一个数字信号的时域波形图可以视为信号数值随着时间变化而发生着变化。在



分析研究时域问题时，以时间作基本自变量，即以时间做为横坐标，函数值或信号值做为纵坐标，从而分析两者的数学关系。

单个乐音信号中包含有基音和谐波两种分量，音乐理论中乐音的泛音是指振动物理学中的谐波，基波频率两倍的谐波被称为一次泛音，基波频率三倍的谐波被称为二次泛音。特定乐器发出乐音时，把两种分量周期性叠加，两种分量相位关系稳定，谐波分量和基音分量呈整倍数关系。时域检测法种类较多，具有代表性的方法有短时幅度差方法<sup>[11]</sup>、短时自相关法<sup>[12]</sup>等。时域检测法能够分离基音和谐波两种分量，以便后续研究，实现乐音的识别。

## 2. 频域检测法

频域(Frequency Domain)检测法是指在对数学函数值或数字信号进行分析时，重点分析函数值或信号值和频率之间关系，而不是研究函数值或信号值与时间之间的关系。采用频域来分析研究问题时，通常取一个较短的时间片断中的数据，以频率变化作基本研究自变量，即频率做为横坐标，函数值或信号值做为纵坐标，从而分析两者的数学关系。时域信号经过傅立叶变换后获得了频域信号，横轴及自变量是频率，纵轴表达了某一个频率信号的振动幅度。

频域检测法种类较多，使用最广泛的方法有谐波峰值法<sup>[11]</sup>，谐波峰值法的信号分析技术核心是“快速傅里叶变换”(FFT)。谐波峰值法用“快速傅里叶变换”处理信号，从而获得频率谱，频率谱中功率峰值呈离散关系，在频率谱图中功率最大的峰值为乐音的基频，其它功率较小的峰值为该乐音的谐波(泛音)。

## 3. 时频域检测法

时频域检测法具有“时域检测法”、“频域检测法”两种方法的特点，可以同时分析声音信号在时域中以及在频域中的变化关系，在时变信号分析上优势显著。时频域检测法中最具代表性的是小波分析法。小波分析法同时实现了“时域”、“频域”的双重变换，在对声音信号进行低频区域分析，可以获取良好的频率分辨率，而在声音信号的高频区域分析，可以获取良好的时间分辨率<sup>[13]</sup>。所以小波分析是信号分析中的“放大镜”，利用小波分析技术可以观察信号特征的细节，小波分析具有恒 Q 性质<sup>[14]</sup>。小波分析技术同时还是一种具有特殊功能的滤波器，对声音信号进行小波变换，其实质是把声音信号分解成高频、低频两种分量。如果用小波分析技术对声音信号进行  $1/2$  的减样，可以把声音信号表达成更平滑的信号样本，对平滑后的声音信号样本再次使用小波分析技术减样处理，能得到更加平滑的信号样本，多次叠代后，最终能成功滤除高次泛音(谐波)，这样声音信号中仅有被小波技术放大的基频分量，这样乐器发出的乐音信号的基音频率被小波分析技术成功识别<sup>[15]</sup>。而采用其它识别方法，往往被大量的谐波能量所

干扰，无法找到准确的基音频率，谐波越丰富的乐音，例如钢琴乐音，越难精确地找到基音频率。

面对复杂的声音信号，纯粹使用“时域检测法”无法完全掌握其信号特点，不能获得精确的结论。如果能采用多分辨率小波辨析算法和传统的傅里叶分析法(FFT 算法)相结合，能够全面掌握复杂信号的特点。“Mallat”<sup>[16]</sup>是常用的多分辨率小波辨析算法代表，可用“Mallat 算法”分解处理声音信号，该算法具有“低频小波系数”的分析能力，能够重建声音信号的低频区，然后检测低频区峰值序列，从而确定声音信号的周期特点<sup>[17]</sup>。再结合能够掌握声音信号的频谱特征的“FFT 算法”，这样就可以全面解析复杂声音信号<sup>[18]</sup>。应用多分辨率小波辨析算法的核心问题是如何选取小波函数，核心的小波函数  $\psi(x)$  是某种光滑函数  $\theta(x)$  的一阶导数<sup>[16]</sup>，才能提高小波函数的辨析能力。

#### 4. 倒谱域检测法

“倒谱域检测法”能够解析获得复杂声音信号的倒谱特征，“倒谱域检测法”是一种基本周期检测算法，把信号的卷积关系转变为求和关系，随后进行分离处理。声音信号取对数的反变换后获得倒谱集，利用倒谱集解析复杂声音信号的特征，理解声音的结构，既不属于时域分析法，也不属于频域分析法，因此被划分为独立的“倒谱域检测法”<sup>[12]</sup>。

“倒谱域检测法”也可归类于声音信号的周期检测方法之一，“倒谱域检测法”提取到声音信号的倒谱特征，从而检测声音信号的基本周期<sup>[12]</sup>。“倒谱域检测法”获得第一个峰值即可能确定为声音的基频<sup>[15]</sup>。

### 1.2.2 机器听觉中的音色特征识别方法

乐器音色识别是机器听觉的一个重要研究趋势，然而从机器听觉的学术角度分析，乐器音色识别与语音识别难度大致相当，但研究乐器识别的学者需要同时精通音乐、计算机两方面的学科知识，因此研究人群小，乐器音色识别比语音识别研究进展较慢，成就少，参考文献少。随着人类社会的物质生产能力的提高，人类社会摆脱了温饱问题，越来越多的学者投入重视文化、精神领域的研究。音乐在人们社会生活中的特殊作用，乐器乐音识别、乐器乐音仿真、乐器电子化会成为音乐领域内软件工程研究趋势。

解析乐器演奏的乐音时，采用不同的数学解析模型，实现乐音的模式识别<sup>[19]</sup>，识别精度完全不同。或许一种数学模型只对一类乐器非常有效，而对其它乐器模式识别效果不理想<sup>[20]</sup>。因此针对不同的乐器，其声学上的数字特征完全不同，采用的数学模型应该不同。调研相关资料，总结常用于乐器音色特征识别的

数学方法分为以下几类<sup>[21]</sup>:

1. 时域特征识别法: 把乐音信号从发音开始至结束全过程作为整体, 对整个声音信号进行整体分析, 获取特征, 可了解声音的起始时间、振幅与时间的变化曲线、声音的结束时间<sup>[21]</sup>;

2. 能量特征识别法: 重点解析乐器乐音信号中的各种能量分布, 例如对声音信号的总能量统计, 对各频率能量统计, 对噪音能量统计, 对超过人耳识别范围的无效声学能量统计<sup>[22]</sup>, 获得各能量分布的数学规律;

3. 波谱特征识别法: 此类方法从乐器乐音信号中取一个短时片断, 采用傅丽叶变换分析法中的短时傅丽叶变换(STFT)计算乐音特征, 找到声音信号中的波谱质心, 测量波谱差幅、计算波谱斜度<sup>[21]</sup>, 以上规律可以用于区分不同乐器的波谱差异;

4. 人类感知识别法: 建立和人类听觉神经系统相似的仿真模型, 模仿人类感知能力来解析乐器乐音信号特征, 例如确定人类感觉上的音量、刺耳度、声学敏感度、MEL 倒谱系数等<sup>[22]</sup>, 可用接近于人类感知方式来区分不同乐器对人耳带来的感受差异。

近年来, 关于乐器乐音信号特征抽取的研究已有一些研究成果。Eronen<sup>[23]</sup>对 16 种西方管弦乐器乐音信号进行采样分析, 实现了倒谱系数特征比较, 在此基础上尝试实现乐器的自动识别, 实验结果是对乐器族的最高识别率为 77%。Krishna<sup>[24]</sup>在乐器乐音识别时提取并比较了 LPCC 和 MFCC 等特征, 采用了 K 近邻法和高斯混合模型实现了对乐器乐音特征实现了自动分类, 实验结果是对 14 种乐器的最高识别率为 90%。Christopher J C<sup>[25]</sup>采用“支持向量机”模型和“高斯混合”模型来分类 9 种乐器的演奏片断, 并用 MFCC 系数特征组合的傅里叶变换和线性预测参数 LPCC 和倒谱系数来识别乐器, 实验结果获得 70% 的乐器个体的识别率。Martin<sup>[26]</sup>从 27 种乐器的乐音中提取一系列人类听觉系统所能感知到的乐音特征, 把这些特征应用于乐器分类, 最终对乐器感知差异识别率达到了 71%, 其研发的系統在处理混响和噪音音节时具有良好的稳定性。Eronen A<sup>[27]</sup>在 30 种乐器的乐音中提取了一系列人类听觉系统所能感知到的特征, 用于分类乐器, 最终对个体乐器识别率达到 85%。Agostini<sup>[28]</sup>提取了各种乐器的波谱特征, 对 27 种乐器分类, 实验结果对乐器族识别率为 96%, 同时对个体乐器的识别率达到了 92%。Antti Eronen<sup>[29]</sup>使用基频、时域、波谱等其它特征, 采用多元正态分析的方法, 实验结果对乐器族识别率达到了 90%, 对个体乐器的识别率达到了 80%。Fragoulis D K 等人<sup>[30]</sup>采用神经网络“ARTMAP”来实现对钢琴、小号、长笛、萨克斯和吉他 5 类乐器发出的单音符信号分类, 取得了较好的分类效果。Bertrand David<sup>[31]</sup>在

实验中获得了 27 种乐器的 1007 段音频信号，采用数学方法提取频谱特征，并用于乐器音色识别和分类，实验结果说明二次分类的效果较好，对乐器族的识别错误率为 3.13%，对个体乐器的识别错误率为 7.19%。

### 1.2.3 机器听觉中的乐曲识别分类方法

乐器音色识别大多是从乐器单个乐音的音色、频率、波谱特征入手，可以用于简单的乐器分类。但是乐器分类也可以从乐曲片断整体模式识别入手，例如击打类乐器和弹拨类乐器在弹奏技巧上完全不同，如果综合考虑乐器演奏片断的所有特征，可以用于对乐器分类和识别，即不仅仅采用单个乐音特征的分析方法，而是对乐器的演奏片断实现综合识别。乐器识别是计算机智能模式识别的一种。模式识别指对测试集类的样本划分为两类或多类，分类方法往往通过比较被分类测试的数据集和系统已认知的数据集的相似性来实现模式识别<sup>[32]</sup>。已知类别的数据作为训练集，被分类测试的数据称为测试集，通过对训练集的学习，最后实现对测试集的成功分类<sup>[32]</sup>。因此音色识别也可以看成是乐器模式识别的简单方法。

在进行乐器乐曲片断模式识别时，机器听觉系统首先对已知的乐器的乐曲片断的演奏风格进行主观描述，并数字化特征，提取特征向量，由此建立一个庞大的乐器演奏风格样本数据库，还可以利用已知特征对机器听觉系统进行训练。当机器听觉系统听到未知乐器乐曲片断时，和样本库中所有已知乐器的特征向量矩阵逐个对比，采用矩阵计算的方式，判断当前乐器演奏风格和何种已知乐器的演奏风格最为接近，以矩阵相似度差异作为识别与分类的数学依据，最终可判定当前乐器的种类。目前常用的乐器乐曲模式识别法有三种：

#### 1. 高斯混合模型模式识别法

高斯模型是采用高斯概率密度函数，即使用正态分布曲线来分析乐曲的演奏特征，来精确地量化提取到的乐器乐曲特征，采用概率统计的方式描述乐曲特征<sup>[33]</sup>，从而把未知乐器的乐曲特征和已知乐器区分。因为高斯模型较容易实现，能够实时响应分类需求，所需的存储空间小，运算量小。这一技术已在语音识别领域中广泛应用，语音识别系统采用概率统计的方式来判定语音片断信号和已知语言片断特征最为相似，从而语音识别系统能够实现对未知语音片断的整体理解，这类方法需要较大的语音知识库来支撑。语音识别和乐曲识别的需求相近，因此理论上，如果能建立大规模的乐曲知识库，高斯模型也能适用于乐器演奏乐曲的分类识别。

#### 2. 支持向量机分类模式识别法

支持向量机可应用于非线性、小样本及高维的模式识别的场合，还能完成函数拟合任务。支持向量机(SVM, Support Vector Machine)理论能够广泛应用于多种机器学习领域<sup>[34]</sup>。

支持向量机分类模式识别法建在统计学基础之上，采用了结构风险最小原理和统计学习理论的 VC 维理论，能在少量的样本信息支持下，对训练样本达到较高的学习精度，实现较好的模型复杂性和较好的学习能力，能尽可能无错误地识别样本。在需要样本数量和良好的识别效果两者之间寻找到最佳折衷，并具有较好推广能力<sup>[34]</sup>。

### 3. 最近邻分类模式识别法

最近的邻(KNN, K-nearest neighbor)分类算法，操作简单，计算速度快，在理论上已发展完善<sup>[35]</sup>。该算法的思路是：一个样本在特征空间中和周边  $k$  个距离最近样本相似，这些样本归入同一个类别，成为样本集<sup>[36]</sup>。在 KNN 算法中，特征空间中分布的所有样本都是已正确分类的对象，所以把未知类别样本按特征坐标放置在特征空间中的正确位置上，依据未知类别样本最邻近的已知样本，来推定未知类别样本的归属<sup>[37]</sup>。KNN 方法只要考察特征空间中的最近的相邻样本即可，所以作为机器学习算法，思路清晰简单，被学术界称为最简单的机器学习算法。

KNN 算法只考察周围若干距离最近样本的类别，通过已知分类样本的分类推定未知分类样本的类别，在原理上不依赖判断类域的方法来确定未知分类样本的类别，因此应对于具有交叉类域待定样本时具有较好的识别效果<sup>[38]</sup>。有些未知分类样本的特征和多个已知分类样本重叠较多时，KNN 算法也能达到很好的分类效果，这一特性优于其它模式识别分类方法<sup>[39]</sup>。

### 4. 神经网络分类模式识别法

人工神经网络(Artificial Neural Networks)模型建立了神经元的拓扑结构，具有一定的学习能力。目前，已研发 40 多种人工神经网络，

Back Propagation 神经网络(BPNN)是一种多层前馈网络，能按误差逆传播算法进行训练，能够求解许多模糊工程问题，目前应用较为广泛。BPNN 能够储存并学习大量输入、输出的映射关系，通过训练学习，无需告知输入和输出的变换过程，就能够扩展推理未学习输入数据大致的输出值。理论上，如果给 BPNN 大量的具有钢琴特征的特征，BPNN 通过训练后，能够判定一个新声音样本是否可以归类至钢琴乐音类别组。

李春杰<sup>[40]</sup>应用了神经网络对乐器乐曲进行模式识别，在实验中采集了 44 组钢琴乐曲样本，30 组长笛乐曲样本，42 组小提琴乐曲样本，36 组琵琶乐曲样本，

26 组贝司乐曲样本、48 组吉他乐曲样本、28 组小号乐曲样本、44 组二胡乐曲样本以及 34 组萨克斯乐曲样本<sup>[40]</sup>。首先解析乐曲，获得了数据特征，将 166 组乐曲样本数据分别提取三色、亮度、不规则度和奇偶度等特征向量，并将特征向量归一化。神经网络参数如下：输入层神经元数 6 个，竞争层神经元为双层结构  $6 \times 9$ ，输出层神经元数为 9 个，输出层的神经元和乐器呈一一对应的关系<sup>[40]</sup>。选择最优化学习速率，训练迭代次数选择为 1000；初始学习率为  $\eta(0)=0.02$ ；相对学习率为  $m=0.3$ ；最大学习率为  $\eta_0=0.1$ ；窗口宽度为  $\varepsilon=0.2$ 。在 9 个类别乐曲中随机选取若干样本组做为训练样本，将处理好的训练样本特征向量输入神经网络，进行 1000 次迭代训练，获得训练好的神经网络。利用训练好的神经网络对 166 组乐曲样本进行模式识别和分类，获得了较好的分类效果<sup>[40]</sup>。

### 1.3 论文的主要内容和组织结构

乐器特征的分析与数字化是乐器识别及电子合成的核心工作，本论文将以钢琴音乐信号识别和合成为例，采用数学方法对钢琴音乐的信号特征识别与分析，对提取出来的钢琴音乐信号特征数字化，实现钢琴音色识别与仿真，最后应用软件工程的理论开发软件，实现钢琴音乐的电子合成。

本论文的组织结构为：

第一章为“绪论”，介绍了钢琴音色电子识别和合成的研究背景和意义。调研了机器听觉相关的研究现状，认为钢琴音色稳定，谐振成分复杂，应用短时傅里叶变换识别效果好，开发难度小。还可以通过乐曲识别的方法，检测钢琴乐音片段振动能量的衰减变化，来提高识别率。

第二章为“钢琴乐音的物理学分析”，调查了钢琴的物理结构，并分析了十二平均律、钢琴弦振动方程的数学原理，对钢琴发声的物理学原理进行了分析，调查了钢琴的结构特点及演奏特点，总结了钢琴乐音的音高、音长、音强、音色的特点。认识到钢琴弦特定的振动方程产生了钢琴特定的音色，和其它乐器比较，钢琴乐音中存在能量较大的泛音，所以钢琴音色更显丰满，被称为“乐器之王”。

第三章为“制定开发方案”，对比软件工程中的生命周期法、快速原型法、面向对象法，认为只有快速原型法能降低本软件工程的成本，缩短工期。最终选择 MATLAB 语言来快速构建原型，用 C#.net 来开发成品软件。

第四章“钢琴乐音特征的数字化识别”，选择了频域检测方法中的短时傅里叶变换研究法对钢琴乐音音色频谱进行分析。用 MATLAB 编写了傅里叶频谱分析程序，提取到钢琴发音的音色矩阵。还发现击弦类乐器声音振动能量衰减变化曲线

和其它乐器不同，所以还提取了音强包络函数，可用于识别击弦类乐器。

第五章“基于 MATLAB 的钢琴乐音仿真”，对钢琴音色特征矩阵进行矩阵计算，实现了钢琴音色中的 25 次倍频谐波的电子合成，同时引入了音强变化包络曲线，实现了钢琴乐音的仿真。

第六章“基于 C#.NET 的钢琴演奏软件的开发”，.设计了三层架构的中间件发音技术，成品软件在发音时无需重新进行矩阵运算，提高了工作效率。

第七章“总结与展望”，总结了本论文的研究成果，指出了不足之处，展望了后续研究的改良方向。

## 第二章 相关理论与技术

声音是一种常见的物理现象。物体机械振动形成声波，借助空气媒介，传至人类的耳膜，引起神经反射，被人类感知为声音，人脑又对声音信号快速翻译成人类可以理解的语言、音乐和各种已知的声音。人类能感知的声音在每秒振动数为20-20000次左右，即20 Hz-20000 Hz的频率范围。

音乐中使用的乐音通过长期改良、调整，已形成了固定组合，能够表达作曲者的感情，传递作曲者的思想，形成音乐文化。考虑到人耳对声波频率的感知能力，音乐乐音被限定在27-4100 Hz的频率范围内，超过这个频率范围，人类不能清晰地感知，对表现塑造音乐形象的意义不大。乐器的乐音音高特征对应振动频率，乐音音长特征对应振动的持续时间长短，乐音的音强特征对应振动的振幅，乐音的音色特征对应着振动的频谱分布。因此乐音特征由振动波的频率、时长、振幅、频谱四种物理量所决定。乐音四种物理量是乐器具有完美表现与感染效果不可缺失的四大元素，所以对乐音的研究，应首先从物理学入手。

物理学研究发现，声音的音高和声波的振动频率关系密切，琴弦振动时，观察振动的波形图，两个相邻同方向峰值之间的间隔时间称为振荡周期，琴弦振动周期越短，则振动频率越高，表现为产生的声音音高越高。

乐音的发音时间的时间长短被称为音长，在乐理知识中，音符是记录不同音长的符号，音符使音乐更具有节奏感。乐音的音长以全音符为代表，其它音符都按和全音符的时间比值来命名，如二分音符是全音符发音时长的二分之一，四分音符是全音符发音时长的四分之一。在乐理知识中，通常以四分音符为一拍，八分音符为半拍。

音强就是人耳感知的声波响度，也就是声音音量的强弱，音强的变化使音乐更具有表现力，表达了演奏者的情绪。例如在弹奏钢琴时按琴键的力度越大，弦槌敲击琴弦的力度也加大，琴弦发出的乐音音量越大，则称为音强越高。

乐器识别的基础任务就是通过识别获得这四个参数的信息，通过音高识别，判定乐器所按的键名；通过音长识别，判定乐器的发声音符；通过音强识别，判定乐器的按键力度；通过音色识别，判定乐器的种类。

### 2.1 钢琴的物理结构分析

普通钢琴的物理结构如图2-1所示。钢琴结构图各部位的标准英文名称如下：



frame (1); lid, front part (2); capo bar (3); damper (4); lid, back part (5); damper mechanism (6); sostenuto rail (7); pedal mechanism, rods (8, 9, 10); pedals: right (sustain/damper), middle (sostenuto), left (soft/una-corda) (11); bridge (12); hitch pin (13); frame (14); sound board (15); string (16)

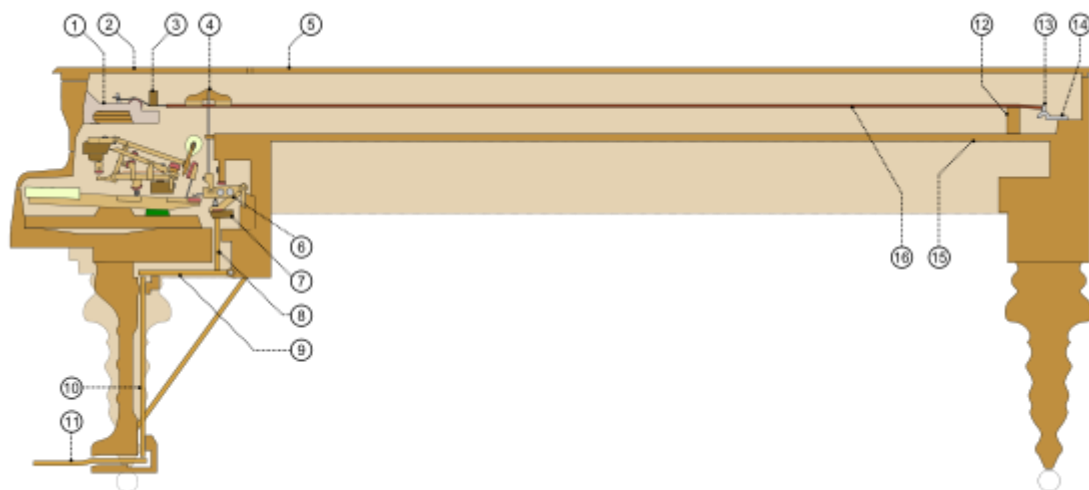


图 2-1 标准钢琴结构图

以演奏专用的大型三角钢琴为例，其物理结构特点归纳如下：

- 1、拥有88个琴键，其中52个白色琴键，36个黑色琴键。第1号琴键为A<sub>2</sub>，第88号琴键为c<sub>5</sub>。
- 2、第1号琴键A<sub>2</sub>为最低音，发出乐音的基频为27.5Hz，第88号琴键c<sub>5</sub>为最高音，发出乐音的基频为4186.01Hz。
- 3、钢琴上存在着一个国际基准音，为第49号琴键a<sub>1</sub>，其发出乐音的基频为440 Hz，该国际基准音是调校钢琴音准的重要标识。
- 4、钢琴的各琴键乐音基频按“十二平均律”制定。音阶关系按半音频率比值的1.0594631的规则排布。
- 5、高档钢琴琴弦由优质碳素钢制作，化学性能稳定，物理性能良好，
- 6、钢琴的琴弦按单、双、三分列为88组，低、中、高音交叉排列，高音区和中音区采用裸弦，低音区采用铜丝缠绕弦，低音和中音区交界处的琴弦也采用铜丝缠绕弦。
- 7、普通钢琴在结构上包含有三大部分：外壳、音源、机械部分。机械部分又包含了键盘机械、击弦机械和踏板机械。
- 8、击弦机械包含有弦槌、联动器、转击器等其它零件。
- 9、音源由弦、共鸣、结构支撑三个部分组成，共鸣系统由共鸣体、振动波传

导机构、共鸣辅助机构三个部分组成，含有音板、弦码、肋木等零件。结构支撑系统中的铁板也具有共鸣效力。

## 2.2 钢琴乐音的音高

在一个乐音成分中，包含了许多频率的振动波，而音高通常指最低频率的基频数值。在乐理中，规定乐器所有音符音高的规则叫做律制。律制首先规定了一个标准音的振动频率，以标准音为基准，应用特定数学运算法则，用倍频的方法从标准音的频率算出其他音的频率。在现代键盘乐器制作中，各键乐音的音高必须遵循十二平均律的标准。现代钢琴正是根据十二平均律来定音的，十二平均律方便了乐器移调，让音乐家能够更加随心所欲的操纵钢琴，提升了演奏技巧，完美地表现钢琴乐曲的感染力。

### 2.2.1 十二平均律简介

中国古代制定音律时所采用的律制是“三分损益法”，公元前7世纪，春秋时期《管子·地员篇》最早记录了这一方法。“三分损益法”依照一定的长度比例，从某一标准音的管长或弦长推算其余一系列音律的管长或弦长。“三分损益法”包含“三分损一”、“三分益一”两种算法。“三分损一”是指将标准音管长度作3等分而减去其1份；而“三分益一”则是指将原有长度作3等分而增添其1份。两种算法交替运用、连续运用，由此产生各音律。“三分损益法”对中国古代管乐器影响深远。

公元前6世纪，古希腊哲学家、科学家毕达哥拉斯在音乐史上首次提出了“五度相生律”，“五度相生律”是根据弦乐器的复合音的“第二分音”和“第三分音”的纯五度关系，即由某一基音向上推一个纯五度，产生次一律，再由次一律再向上推一个纯五度，产生再次一律，这种方法称为“五度相生律”。“五度相生律”对西方的弦乐器发展影响深远。中国的曾侯乙编钟的音符排列未采用“三分损益法”而具有“五度相生律”雏形，说明中国在早期也有对“五度相生律”的音乐律制进行探索。

在音乐发展史，公认现代音乐律制“十二平均律”由明朝音乐理论家、音乐数学家朱载堉发明。朱载堉(1538-1601年)字伯勤，号句曲山人。为明仁宗后裔、郑恭王朱厚烷之子，其潜心研究音乐，发表了多篇音乐论著。

公元1584年(万历十二年)，朱载堉完成了《律学新说》一书，首次提出了“十二平均律”的理论。在此之前，从春秋时期起，管弦乐器都是采用“三分损益法”

来确定管或弦的长度,规定了各个乐音音高关系。“三分损益法”计算获得的十二个律,相邻两律间的频率差不同,所以也称为“十二不平均律”。比基音高八度的乐音,近似比基音音高高一倍,不可能是正好的整一倍。如基音“do”的相对频率被视为“一”,高八度的“do”音的相对频率不为整倍数“二”,而是略高于“二”,其倍数存在着一定的误差,导致乐器不适宜进行“变调”演奏,也不便于实现和声演奏。“十二平均律”的发明则彻底取消了“三分损益法”计算产生的倍数差异。使乐器的十二律不能周而复始、循环演奏的难题得到了完美解决,“十二平均律”推动了世界音乐文化史的发展。欧洲人发现并采用十二平均律改良乐器,滞后于朱载堉半个世纪,52年后(1636年),Pere Marin Mersenne<sup>[41]</sup>发表了著作《谐声通论》,也独立提出了十二平均律。

“十二平均律”亦称为“十二等程律”,即把一组八度音程按频率等分成十二个音程的律制,相邻两律之间的振动频率之比相等,每一等份的音程称为一个“半音”,在乐理上称为“小二度”,乐理上的一个“大二度”则是两等分<sup>[42]</sup>。在数学上存在一个巧合,“十二平均律”将八度音程均分成12等分,“纯五度音程”两音频率比为“2的7/12次方”,接近于“1.5”,所以人耳一般无法分辨“五度相生律”和“十二平均律”在五度音程上的音高差别。

中西音乐文化在律制的选择多多少少存在着一些相似性,中国音乐界早期采用“三分损益法”律制,西方音乐界早期采用了“五度相生律”,两个古律制到现代的“十二平均律”都有许多数学上的相似之处,使得中西不同文化背景的人群产生音乐感受习惯的共通性。这些音乐律制都存在着数学规则,表明了数学至始至终就存在音乐的灵魂中,数学是开启音乐研究大门的钥匙。

### 2.2.2 十二平均律的数学原理

现代键盘乐器普遍采用十二平均律作为确定各键音高的方法。乐曲的曲调由音阶组成,音阶由乐音组成,不同的乐音其绝对音高不同。键盘乐器的乐音是靠琴弦振动发出的,琴弦振动的频率,决定了的琴弦发音的绝对音高,不同的乐音则由不同的振动频率来表现,所以绝对音高不同。人类选取一些特殊的振动频率作为特定音高来搭建音乐体系。

十二平均律确定了乐器各乐音的基频,在乐理中,“八度”音程在基频频率上差异等同于物理学上倍频概念。在乐理知识中称两个同名音符音高相差为八度,在物理学上恰指两个同名音符的基本频率相差1倍<sup>[43]</sup>。

钢琴也是遵循十二平均律制制作的乐器。钢琴上每相邻的两个琴键(包括黑、

白琴键)的频率差异称为半音。半音是十二平均律体系中最小的频率差异距离,即最小的音高距离。将一个八度音程十二等分为十二个半音,其数学意义如下:八度音程指的是从基频至两倍频率的范围。因此在八度音程中分为十二等分乃是把这个频率范围分为十二个等比级数,其结果就是每个后续音的基频为前一个音的  $f_x = f_a \times (\sqrt[12]{2})^n$  (n取1),即1.059463倍<sup>[44]</sup>。

1939年,英国伦敦国际会议决定440Hz为第一国际标准音高。该会议还规定钢琴键是49A(a1)的基频为440Hz;又规定每相邻半音的频率比值为  $2^{1/12} = 1.059463$ ,根据这些标准,就可以计算出钢琴上每琴键乐音的基本频率。按照  $f_{a1} = 440\text{hz}$ ,比a1低个八度的a的频率值  $f_a = f_{a1}/2 = 220\text{hz}$ ,在a至a1八度内的音x的频率  $f_x = f_a \times (\sqrt[12]{2})^n$ ,其中指数n为音x至音a间隔的半音数目<sup>[45]</sup>。

计算过程举例如下,与琴键“a1”右边相邻琴键“#a1”的频率是  $440 \times 1.059463 = 466.16372\text{Hz}$ ;接着计算可得,“#a1”右边相邻“b1”的频率是  $493.088321\text{Hz}$ ;c2的频率是  $523.25099\text{Hz}$ ……。同法计算,与“a1”左边相邻的“#g1”的频率是  $440 \div 1.059463 = 415.030473\text{Hz}$ ……。

从数学理论上来看,十二平均律的设计是完美的,可以应用于乐器的各个音域。但在实践中,人类发现十二平均律在中音音域类匹配人对音阶感觉。当频率大于1500Hz后,人类感觉上的音高较十二平均律计算值略高,为了人类感知钢琴声音的和谐,高音域类和低音域类各琴键的音高不是完全按照十二平均律来定音的,所以钢琴调音师不能完全被十二平均律的仪器所替代。在中音区区域,完全按照“十二平均律”来调整各琴键的音高;在低音区域,倾向于采用“五度相生律”的方法调音,即半音音程变小;在低音区区域,倾向于采用“纯律”来调整音高,半音音程变宽。

从钢琴的调音曲线(如图2-2所示)上看,高音区各音符音高偏高,各音符低音区偏低,这是为了保证人类听觉系统感知的乐音更加和谐,但也具有数学、物理学上的因素。钢琴在低音域内半音减小,高音域内半音增大,从物理意义上说,主要是琴弦两端的约束造成的,分数的分子和分母同时增大或较小,会引起比值减小或增大,意味着过长的琴弦和过短琴弦会引起半音音程发生变化。人类听觉器官中重要振动元件“耳膜”表现也与琴弦特点类似,在低音区域内,耳膜对音高变化感知敏感,在低音区域内,耳膜对音高变化迟顿。因此从自然现象的角度来看,钢琴的调音曲线即符合了钢琴的振动数学和物理学原理,又符合人类耳朵

的敏感特性。因此在钢琴音色识别时，应将钢琴化分成高音、中音、低音三个区域，分别提取音色特征，分别加以识别。

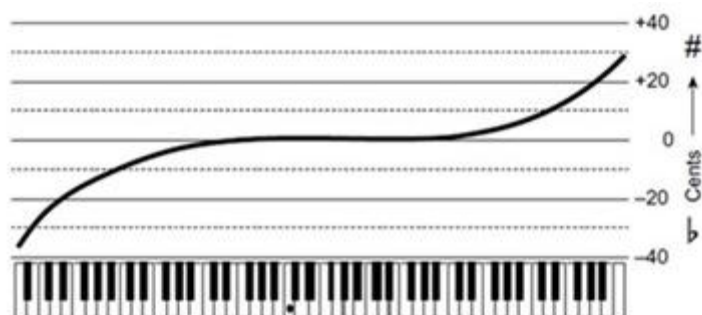


图 2-2 钢琴音高的调音曲线

### 2.2.3 钢琴键盘的音高定义

钢琴键盘上的每个琴键对应着不同基频的乐音，无论白键、黑键，钢琴上相邻两个键的音高相差半音。每七个白键(含白键间的黑键)分为一组，键盘由多个白键组构成。每一个“七白键组”的音高频率高于左侧的“七白键组”，而低于右侧的“七白键组”。另外两个白键连接排列，两个白键的音程距离为半音，如果两个白键中有黑键，两个白键的音程距离为全音，白键和夹在其中的黑键的音程距离为半音。

在乐理中，“音列”按照高低顺序形成排列组合，由此形成了“乐音体系”，“音列”排列成“音级”，“音级”分为“基本音级”、“变化音级”，“基本音级”有独立名称，钢琴键盘上的52个白键反复使用7个“基本音级”。相邻的同名音级的音程距离为八度(Octave)。钢琴键盘上的36个黑键使用“变化音级”，“变化音级”对“基本音级”进行升高半音或降低半音。所以一个八度中包含了12个音符，音符名称分别为：“C”、“#C”、“D”、“#D”、“E”、“F”、“#F”、“G”、“#G”、“A”、“#A”、“B”，这个现象和“十二相生律”对八度音程的十二等分的概念是匹配的。

## 2.3 钢琴乐音的音长

音长指单个乐音发出声音的持续时间，其实质是琴弦的振动持续时间，因此对音长的识别则是实现对琴弦的振动波持续时间的识别。但是人类的音乐存在着鲜明的节奏感，音乐理论规定了音长，限制了乐音的持续时间，以保证乐音的持续时间存在着严格的比例关系。

在乐曲中，音符被按照一定的时长组织起来，形成了节奏，一系列发音的乐音组织成为了节奏型(Rhythm Pattern)音乐。节奏型音乐存在着许多风格，如进行

曲、舞曲、玛祖卡舞曲等。因此节奏的识别是十分必要的，能够确定音乐的基本风格。

乐曲中的节奏概念有两个专有名词：拍(Beat)、小节(Measure)。例如，音乐节奏被定义为“2/4”节拍标识，即指一个“四分音符”为一个完整的“音乐节拍”，所以“八分音符”的发音时间应为半拍，“八分音符”乐音的持续的时间应为“四分音符”乐音的 1/2 半；而“十六分音符”乐音应持续发音 1/4 拍。所以对音长的识别，其实质是检测声音的振动时长，从而实现了识别乐曲的节奏模式。在乐曲中，小节是音乐乐句的基础，包含了若干数量的节拍数。

对于一首乐曲，作曲者并没有硬性规定音符演奏绝对时长，乐谱是通过如下规定，来决定每个乐音的演绎方法。

#### 1. 单音符的发音时长等级的划分

按照乐音持续时长不同，音符分为“全音符”、“二分音符”、“四分音符”、“八分音符”、“十六分音符”、“三十二分音符”等。乐曲的节奏风格由上述音符决定。

#### 2. 每个音乐小节所包含的拍数

每个乐曲定义了音乐小节中包含的拍数，同时也定义了音乐小节中拍的强弱关系。演奏者可以从乐谱的拍号定义中获得拍数、强弱关系的信息，从而真实再现乐曲的节奏。

#### 3. 每分钟应当演奏多少拍数

每分钟演奏多少拍数，决定了乐曲节奏快慢，这可以通过计时器检测获得。

根据上述介绍三个和音长有关的参数，可以明确确定一首乐曲的音符节奏模式。通过计时器的应用，同时识别各音的音长，可以捕获并记录以上三个参数，从而再现一首乐曲的节奏风格。

## 2.4 钢琴乐音的音强

音强又称响度、音量，是指人类听觉神经系统对声音对耳膜声压大小的主观感受。人类的听觉神经感知声音时，人耳对不同频率的声音的敏感性不同，但是假设一个发声体振动频率不变，人耳所能感知音强由发声体振动振幅大小决定。振动的能量最大，振幅越大，则人耳感知到的音强越大。和人耳工作原理不同，声压级能用于客观标定声音能量大小，声强的单位为分贝(dB)。音乐理论中对乐音的声音的强弱变化划分为“ppp 至 fff”七个级别。

对钢琴的琴弦而言，即便敲击力度、位置、方式完全相同，音量却不一定相同。因为琴弦振幅还受到琴弦的张紧程度、弦工艺质量，甚至钢琴的共鸣腔结构

的影响。演奏大厅内的三角钢琴为了保证演奏音量宏大，整体尺寸设计较大，通常为9英尺以上(274 cm)，这样确保了四个条件；1)获得了足够尺寸的共振板，大型三角钢琴会选用三平方米以上的共振板；2)可采用张紧程度高、质量重、尺寸长的优质琴弦；3)可以有更大的空间布置击打琴弦的键盘机构；4)可使用硬度较高、质量较大的弦槌，保证琴弦振幅。

根据钢琴弦的振动原理，钢琴低音区的键A2(27.5 Hz)至A25(110 Hz)必须使用长、粗、质量大的琴弦，使用大型弦槌，形成较大的冲击力，击打琴弦的有效部位，才能获得与中音区相匹配的音量。这一现象除了和钢琴弦振幅物理学原理有关外，其实还和人的听觉神经系统的敏感性有关。声学博士韩宝强先生在《音的历程》书中指出：“频率20 Hz、响度为80 db的纯音与频率为1000 Hz，响度为10 db的纯音听起来一样响”，这是因为人类听觉神经系统对不同频率的振动波敏感性不同导致的。其它声学研究还发现，人类听觉系统对声音的频率在“1000 Hz-6000 Hz”范围内特别敏感。这一现象在钢琴乐音合成时，应当加以注意，对不同频率段的音符的振幅强度应当分别处理，争对人类听觉神经系统的特性，以保证高音、中音、低音区域不同乐音的音强更加和谐。

## 2.5 钢琴乐音的音色

钢琴做称为“乐器之王”，低音域音色厚重，中音域音色自然，高音域音色透亮，具有独特的魅力，是音乐家最喜爱的乐器，用途最为广泛。

前人发现，天然乐器发出的声音一定是复音，复音在振动波成分上有基音和大量泛音，而不同频率的基音和泛音的音强不同，组成了极其复杂的频谱结构。不同的乐器音色不同，其实质是不同的乐器发出的振动波有着不同的频谱结构。所以两种不同的乐器发出乐音，即使音高、音长、音强完全相同，人类还是能快速区分乐音差异，确定乐器的种类。这个原因是同一类乐器的频谱结构十分稳定，人类在生活中早已将各种乐器的频谱结构牢记在大脑里，而不同种类的乐器的频谱结构完全不同，当人类听到一个新的乐音，会把这个乐音的频谱结构和脑海里所有已知乐器的频谱结构做对比，从而判定未知乐音类似于何种乐器。

人类只要通过简单音乐训练，即可从区分乐器的音色差异，对于人类来说，能从众多乐器声音、环境噪声中快速分辨钢琴乐音。但是应用计算机机器听觉技术，实现音色自动识别，难度则较大，这涉及到了信号处理、模式识别和人工智能的问题。要实现机器听觉对乐器音色的识别，仍然要从声波的数学原理入手，从而发现各种乐器音色在数学表达形式上的差异，利用这些差异，才能够快速识

别和区分乐器音色。

### 2.5.1 纯音与复音的音色区别

纯音(Pure Tone)是指发声体作简谐振动,对外产生的单一频率的乐音,纯音通常只能在实验室人工制造的环境中产生,完美的纯音大多产生于人工制造的电子设备,在自然界中唯有音叉能产生近似的纯音,大多数乐器不能发出纯音,只能发出复音。复音(Complex Tone)是发声体在形成复杂振动时,发出的声音中包含有许多频率,绘制频谱图,可以发现复音中频谱结构复杂,最低的频率称为基频(First Partial),该声音成分为基音(Fundamental Tone)。在频谱结构中高于基频的其它频率对应的声音成分称为泛音(Over Tones)。当泛音的频率恰好为基频的整数倍时,称为谐频。乐器发出的复音中振动能量通常分配给最低的频率,所以人类感知的乐音音高往往由基频决定。

自然乐器产生的音都是复音,在同一个时间内只有一个复音的乐曲被称为“单音旋律”,而在同一个时间内产生多个复音的乐曲被称来“多音旋律”。可以假想“多音旋律”中存在着多个通道,每一个通道中能够发出一个乐器的声音,所以多通道技术能够产生多乐器合奏的效果。在手机铃声中有一类“16和弦”,则是指铃声中存在了16个通道,每一个通道能够发出一种乐器的声音,从而形成16种乐器合奏的效果。从电子合成的难度来看,应从纯音的仿真入手,然后实现复音,最后建立多通道,每个通道仿真一种乐器的复音,最终实现多音旋律。由易至难,开发电子乐队软件。

### 2.5.2 钢琴弦的振动方程

钢琴独特的复音音色是由钢琴弦振动产生的,受到了钢琴弦振动物理学原理的数学定义。所以要分析钢琴的音色特征,必须从钢琴弦振动特征入手。物理学原理指出,两端被牢牢固定的钢琴弦可以容纳一组正弦波的往复运动。这一组正弦波在琴弦上各自形成驻波。驻波的存在条件必须满足钢琴弦弦长是驻波的整数倍。即第一级驻波的波长恰好等于钢琴弦弦长,其它驻波的频率恰好是第一级驻波的整数倍。除此外其它频律的波是行波,没有办法稳定在钢琴弦弦上持续振动,因此很快会消失。前人在研究中推导了钢琴弦振动方程来描述钢琴弦的振动过程,钢琴弦振动方程决定了琴弦上的所有驻波成分。

钢琴演奏时,表演者敲击键盘,键盘机械把人力能量传递至弦槌,弦槌快速打击琴弦,假设弦槌对琴弦敲击部位较窄,时间短促,宽度、位置固定<sup>[46]</sup>。因此,



设琴弦长度为 $l$ ，弦槌宽度为 $2\delta$ ，敲击点位置为 $x = x_0$ ，则敲击区间 $X$ 为琴弦上 $(x_0 - \delta \leq X \leq x_0 + \delta)$ 区域。钢琴弦在振动时，初速度运动为 $v_0$ ，钢琴弦振动加速度为 $a$ ，钢琴弦振动时间为 $t$ ，假设在敲击瞬间时， $x = x_0$ 点获得的初速度最大，而在 $x = x_0 - \delta$ 和 $x = x_0 + \delta$ 两点速度设为0，这样弦槌离开后，钢琴弦物理运动为自由振动，因此前人推导的弦振动方程如下<sup>[47]</sup>：

$$\left\{ \begin{array}{l} \partial^2 u / \partial t^2 = a^2 (\partial^2 u / \partial x^2) \\ u|_{x=0} = u|_{x=l} = 0, (0 < x < l, t > 0) \\ u|_{t=0} = \varphi(x) = 0 \quad (0 < x < l, t > 0) \\ \frac{\partial u}{\partial t}|_{t=0} = \begin{cases} v_0 \cos \frac{x-x_0}{2\delta} \pi, & (x_0 - d \leq x \leq x_0 + d) \\ 0, & 0 \leq x \leq x_0 - d, x_0 + d \leq x \leq l \end{cases} \end{array} \right. \quad (2-1)$$

根据多个边界条件，可知本征函数为 $\sin(n\pi x/l)$ ，钢琴弦振动的一般公式可表达如下：

$$u(x, t) = \sum_{n=1}^{\infty} (C_n \cos \frac{n\pi a}{l} t + D_n \sin \frac{n\pi a}{l} t) \sin \frac{n\pi}{l} x \quad (2-2)$$

把式（2）带入初始条件来确定系数 $C_n$ 和 $D_n$

$$\varphi(x) = \sum_{n=1}^{\infty} C_n \cdot \sin \frac{n\pi}{l} x = 0 \quad (0 < x < l) \quad (2-3)$$

$$\psi(x) = \sum_{n=1}^{\infty} D_n \cdot \frac{n\pi a}{l} \cdot \sin \frac{n\pi}{l} x = \begin{cases} v_0 \cos \frac{x-x_0}{2\delta} \pi, & (x_0 - \delta \leq x \leq x_0 + \delta) \\ 0, & (0 < x < x_0 - \delta, x_0 + \delta < x < l) \end{cases} \quad (2-4)$$

式（3）和式（4）的左侧部分是右侧部分中函数的傅里叶级数，由此可求得系数如下：

$$C_n = \frac{2}{l} \int_0^l \varphi(x) \sin \frac{n\pi x}{l} dx = 0 \quad (2-5)$$

$$D_n = \frac{2}{n\pi a} \int_0^l \psi(x) \sin \frac{n\pi x}{l} dx = \frac{8v_0 d}{n^2 \pi^2 a} \cdot \frac{1}{1 - \frac{4d^2 n^2}{l^2}} \cdot \sin \frac{n\pi x_0}{l} \cdot \cos \frac{n\pi d}{l} \quad (2-6)$$

所以钢琴弦振动的方程式为：

$$u(x,t) = \frac{8v_0 d}{n^2 \pi^2 a} \sum_{n=1}^{\infty} \left( \frac{1}{n} \frac{1}{1 - \frac{4\delta^2 n^2}{l^2}} \sin \frac{n\pi x_0}{l} \cos \frac{n\pi d}{l} \sin \frac{n\pi a t}{l} \sin \frac{n\pi x}{l} \right) \quad (2-7)$$

### 2.5.3 钢琴弦振动的泛音特征分析

钢琴弦振动方程决定了钢琴弦振动的基频能量最大，所以人类听觉系统所感受最显著的频率是基频。即基音振动频率决定钢琴琴键对应的音高。钢琴弦振动方程还决定着在基频的倍频处，还会产生一定能量的振动，这些低能量的倍音人耳也能感受一些，这样人类在主观上形成了对钢琴音色的主观体验。各种乐器能拥有不同的音色，其实都是因为基音和倍音能量分配不同而形成的不同的主观体验。所以要实现一种乐器的电子合成，其实就是要实现基音和倍音能量分配的仿真。人类的听觉系统有特定的敏感范围，所以无须对全部频率上的振动能量分配比例进行忠实记录，只要以人类的听觉神经系统的主观感受为评价标准，一旦通常专业训练的音乐人士认为电子合成的乐音效果和真实乐器的乐音无法区分，即达到了完美电子合成乐音的目标。

从钢琴弦振动方程可知，钢琴琴弦振动综合波形图简单叠加了基频和所有谐波，即简单叠加了所有驻波的波形图，且 $n$ 次谐波的幅度有差异，钢琴弦振动综合波形图中的正弦、余弦函数所组成所有谐波能量与基音能量按 $1/n$ 成比例衰减，而弹拨类弦乐器，如吉它弦的振动所有谐波能量与基音能量按 $1/n^2$ 成比例衰减。人类听觉的感知能力有限，太低能量的振动无法感知，钢琴泛音能量比其它乐器高，所以人类能感知钢琴琴弦的泛音比其它弦乐器多得多，在人类的听觉系统中，钢琴琴弦振动含有了丰富的人类可感知到的泛音，所以人类认为钢琴比弹拨类乐器音色更加丰满动听。

## 2.6 小结

本章首先从十二平均律入手，综述了东西方对乐律的研究历史，剖析了十二平均律的数学模型。接着对钢琴发声的物理学原理进行分析，调查了钢琴乐音的音高、音长、音强、音色的特征，分析了钢琴的结构特点及演奏特点。

在此基础上，尝试对钢琴音乐信号进行识别，首先分析了前人推导的钢琴弦振动方程的过程，认识到钢琴弦特定的振动方程产生了钢琴特定的音色，而且影响了各音符的音强、音长程度。从钢琴弦振动方程可获知，钢琴琴弦振动产生了大量的驻波，而弹拨类弦乐器驻波更少。所以钢琴乐音中存在能量更大的泛音，

所以钢琴音色更显丰满。

综上所述，钢琴乐音的音高、音长、音强、音色具有独特特征，其发音原理遵循物理学的基本规律，由十二平均律和钢琴弦振动方程两个数学模型所决定，从两个数学模型入手可以提取钢琴乐音的特征，为后续研究提供了数学依据。

## 第三章 制定开发方案

本课题的需求分析过程较简单，客户提出的需求目标为：实现对钢琴乐音的音色识别，并电子合成钢琴乐音。要实现这一需求目标，要从钢琴乐器声学理论入手，了解钢琴乐音的发声原理，提取钢琴乐音的各种特征，最后实现钢琴乐音的仿真。本次软件工程的验收标准是仿真音质要达到音乐专业人士的认可。

软件工程的目的是：在甲方给定需求、进度、成本前提下，乙方开发出具有可靠性、有效性、适用性、可理解性、可重用性、可修改性、可维护性、可移植性、可追踪性、可互操作性及甲方完全满意的软件产品。追求这些目标有助于降低软件开发风险，提高软件产品的质量，加快研发进度，减少维护的工作量。

### 3.1 软件开发方法选优

在 20 世纪 60 年代发生了软件危机，为了提高软件开发绩效，1968 年和 1969 年召开的两次 NATO 会次，提出了“软件工程”的概念，由此发展了适宜软件工程的多种软件开发方法。目前软件工程中常用的软件开发方法有生命周期法、快速原型法和面向对象法等。

#### 3.1.1 面向对象法

软件工程中的面向对象开发方法具有如下特点：1.在软件工程中提出了对象概念，一个软件工程项目被分解出多个对象，对象融合了属性和方法为一体，把庞大的工程分解为多个对象取代了传统的功能分解；2.把所有对象都按属性和方法划分，具有相同属性和方法的对象被归入同一个类；3.把多个属性或方法相近类组成层次结构，下层的类自动拥有上层类的属性和方法，实现子类从父类的继承；4. 对象之间通过消息技术相互联系，对象的私有信息被封装在自身结构中，不能被外界访问，实现封装性<sup>[48]</sup>。

如果熟知各种乐器的特征差异，可以采用面向对象软件工程方法开发一个“综合乐器”仿真系统，首先建立一个乐器基类，其特征为振动发音。在乐器基类划分弦乐器类、管乐器类、鼓乐器类、簧乐器类，各子类的振动部位不同，振动方程不同。然后在弦乐器类下再划分击弦类、弹拨弦类、拉弦类，每一个子类对应一种乐器对象，音色不同，音强衰减曲线不同，但是可以继承父类“弦乐器类”的所有特征，所以采用面向对象的方法能有效地缩减“综合乐器”软件工程的开发工作量。

然而本论文只研究一种大型三角钢琴，在研究之前，不知其发音特征，不知该发音特征和其它乐器的关系，无法划分“对象类”的层级，无法建立“对象类”的继承关系，因此只要实现一种乐器的仿真需求，而不知道其发音特征时，选用面向对象软件工程开发方法没有任何优势。

### 3.1.2 生命周期法

生命周期法(也称为结构化系统开发法、瀑布模型法)是目前最流行的开发方法，涉及到了软件工程学和系统工程学的理论，采用结构化方法对系统进行分析和设计，象瀑布一样，自顶向下完成软件工程。

生命周期法将软件工程划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护六个阶段<sup>[49]</sup>，这六个阶段构成信息系统的生命周期，六个阶段划分清晰，每一阶段工作量小，风险可控。所以生命周期法在开发需求明确、大而全、技术路线成熟的管理信息系统时，具有突出的优势。

本项目涉及的知识领域跨越多个学科，开发有难度，无成熟的技术路线可参考，最大风险是不能预先确定完美再现钢琴音色的技术方法。传统的生命周期法要求按六个阶段自上而下、相互衔接的固定次序来完成软件工程，如果一开始选择了播放钢琴录音文件来合成钢琴乐曲的技术方法，直到测试阶段提交软件给客户时，客户才能发现其合成效果单调而无感情。因为录音这一技术方法过于简单，无改善音色的扩展空间，最终软件可能被客户拒绝，将导致工期延误，资金损耗等一系列巨大损失，所以考虑到音色仿真上存在的技术风险，本项目不适采用生命周期法。

### 3.1.3 快速原型法

快速原型法(Prototyping)是在软件工程的前期，采用第四代程序语言(4th Generation Language, 4GL)快速搭建一个简单系统模型，能够在最短的时间内实现向客户演示系统模型。在后续客户交互过程中，不断完善系统。

快速原型法的优点克服时间周期法投资风险大的缺点，在项目开发前期快速制作低成本的简单原型，当客户不满意而又无改良的方法，就采用抛弃策略，更换技术方法重新制作原型；如果客户可以接受，则采用补充策略，继续完善原型，最终达到客户的要求。

快速原型法的缺点是如果大量地修改，或是添加大量的补丁程序，可能使软件产品质量低下。在本软件工程中，采用的方法是先用 Matlab 语言开发快速原型，找到客户满意的钢琴音色仿真的数学方法后，丢弃快速原型，再用 C#.net 语言重

新开发界面良好、整体质量优秀的软件系统<sup>[50]</sup>。

Matlab 是一种数学建模能力强、交互式好、可即时修改观察运行结果的语言<sup>[51]</sup>。开发的快速原型虽然界面差、程序零散，但是 Matlab 便于调试和数学分析，因此顺利地实现了钢琴音色的特征提取和仿真，通过多次交互和数学模型的改良，仿真效果最终获得了音乐专业人士的认可。

C#.net 编程语言从 C，C++ 和 Java 发展而来，是事件驱动，面向对象的可视化编程语言，可以开发界面漂亮的应用程序。而且 C#.net 反编译效果优于 Matlab，可以保护最终的研究成果。C#.net 编译的程序执行效率也优于 Matlab，可以跨平台在 PC 机、智能手机、嵌入式设备中运行，拓宽钢琴音色的电子合成的应用领域。

### 3.2 软件开发目标

本软件要求在现代信号处理、模式识别的技术支撑下，揭示音乐背后的数学秘密，提高人类对音乐的理解。软件工程技术在音乐领域的应用，将给人类带来全新的听觉感受，符合国家近期提出的文化强国理念，在人文、科技领域都具有研究意义。

本软件要求软件工程和音乐理论相结合，以电子合成钢琴音色为目的，在剖析“十二平均律”、“钢琴弦振动方程”的数学基础上，对钢琴音色识别并实现电子合成。最终目标是开发一个软件，能在电脑上仿真钢琴的界面，在屏幕上绘制钢琴键盘，演奏者将能够用鼠标或键盘操作，弹奏钢琴乐曲。

要求仿真效果接近于真实钢琴的声音，但是要求干净无背景噪音，而仿真的钢琴乐音和其它乐器的乐音有明显差异，具有良好的区分度。

### 3.3 制定技术路线

面对本项目，大多数软件工程人员首先想到的技术方法是制作钢琴的录音文件，在电子合成时播放录音文件即可，所以首先开发了采用录音技术的快速原型系统。很快发现该方法技术不可行：

1. 程序员在制作原型时发现：录音中带有环境噪声和电气噪声，噪声随机分布在各频率上，虽然有频段去噪、小波去噪等技术，但在不知钢琴发音特征的情况下，去噪过程破坏了钢琴音色。而如果已知钢琴发音的特征，就可直接电子合成钢琴音，不必利用录音文件。

2. 音乐专业人士在快速原型交互中指出：钢琴弹奏时可以采用不同的力度，使用延时踏板。而录音文件中的振幅和时长被固化了，而简单播放单个音符的录

音文件拼接成乐曲，呆板而失去生命。就象一些利用单个汉字的录音文件拼接成机器人语言，语音单调而无感情。只有对数字化的钢琴音色特征再加工，才可以自由改变振幅和时长，合成的音乐才具有灵性。

面对本软件的需求，软件工程人员还会想到利用 MIDI 技术来实现钢琴乐曲的合成，MIDI(Musical Instrument Digital Interface, 乐器数字接口)技术最早出现于 1980 年，当时 MIDI 技术仅是为了解决电子化乐器之间的通信问题，在通讯中仅仅传递音符和控制信息，未包含声音的波形信号，所以体积小，传递快。MIDI 音色库接收到音符和控制信息，产生相应乐器的振荡波形信号，从而发出所需要乐器声音。一个音质优秀的 MIDI 音色库所需要存储空间通常很大，例如象牙钢琴音色库体积达 9 张 DVD 的容量，一个音符在 256 级力度下表现不同，在延时踏板下表现不同，在弱音踏板下也表现不同，MIDI 音色库对这些表现具有差异的乐音忠实地采样，完整地记录原始波形，所以音质优秀的 MIDI 音色库体积都很大，导致电子合成的软硬件成本高。市面上也有一些经过简化的、携带性好的小型音色库，一个经过专业训练的音乐人士可以听出其音色和真实乐器存在着一定的差别。综上所述，利用优秀的 MIDI 音色库来合成钢琴音色的成本太高，而利用小型 MIDI 音色库则效果不佳。

也有一些公司开发了软波表技术，用软件算法代替声卡上的硬件音色库。软波表通过 CPU 的实时运算来回放乐器音效，不同公司的软波表效果不同，差异较大。软波表的优点是能够产生高品质的音色、价格低廉和便于升级。本论文拟采用类似软波表的技术来实现钢琴音色的高效、低成本地合成。而他人研发的软波表中的电子合成算法是商业机密，不可能免费获得，未授权不能投入商业活动中，因此需要自行研究。

通过快速原型法的多次尝试，和客户细致交流，在软波表技术基础上确定了一个可行的研究方案：

首先从十二平均律入手，调研东西方对乐律原理的研究历史，剖析十二平均律的数学模型。接着对钢琴发音的物理学原理进行分析，调查钢琴发音的音高、音长、音强、音色的特征，分析了钢琴的结构特点及演奏特点。在此基础上，尝试对钢琴乐音特征进行分析，首先在前人的研究基础上，对钢琴弦振动方程做了一些研究工作。从数学角度，认识到钢琴特定的振动方程产生了基本音频振动以及许多高次谐波振动，这些振动混合形成了钢琴特定的音色。

随后，调研机器听觉在音乐领域的研究进展，分析对比时域检测方法、频域检测方法、时频域检测方法、倒谱域检测方法。最终确定采用频域检测方法中的短时傅里叶变换研究法对钢琴频谱进行分析，通过精确调音，获得了钢琴 440Hz

的国际标准音录音文件，编写傅里叶变换分析程序，获得钢琴频谱图。把钢琴频谱图与校音器发出的纯正 440Hz 正弦波对比，发现了钢琴频谱图充满了大量的倍频泛音，钢琴特定的泛音的组合决定了钢琴特定的音色，钢琴频谱中充满了大量倍频振动，这是钢琴音色浑厚浓郁的原因。随后发现可采用矩阵记录钢琴音色中的倍频振动幅度的关系，从而实现钢琴音色的描述和识别。在此基础上，将音色特征矩阵转化为各倍频的物理振幅，最终实现钢琴音色的仿真。

综上所述，确定本次研究的技术路线如图 3-1 所示。

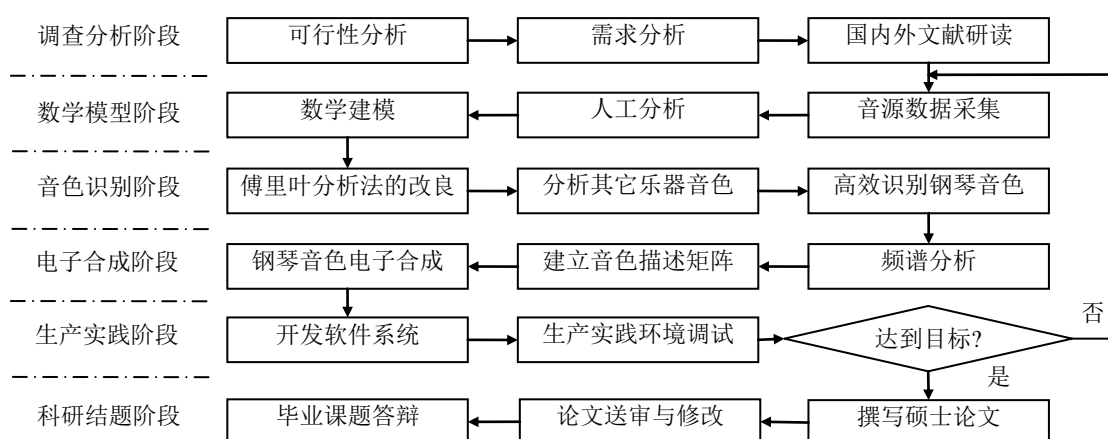


图 3-1 技术路线流程图

### 3.4 小结

本项目存在着最大的技术风险是不能事先确定仿真钢琴音色的有效方法，争对这一风险，对比软件工程开发方法中的生命周期法、快速原型法及面向对象法，认为唯有快速原型法能有效地降低开发风险，缩短工期。本项目拟采用 MATLAB 语言来快速构建原型，待用户对电子合成的音色满意后，再改用 C#.net 来重新开发成品软件。针对快速原型法，制定了高效的技术路线，能够根据最终电子合成的乐音是否达到用户要求，选择是否要返回快速原型，对快速原型进行完善，这一技术路线保证了后续研究能够顺利进行。



## 第四章 钢琴乐音特征的数字化识别

在前期研究中，调研了机器听觉在音乐领域的研究进展，分析对比了时域检测方法、频域检测方法、时频域检测方法、倒谱域检测方法。本论文选择了频域检测方法中的短时傅里叶变换研究法对钢琴乐音音色频谱进行分析。

### 4.1 基于短时傅里叶变换的频谱分析

傅里叶分析广泛应用于各种工程领域，以及各种科学研究领域，是一种分析状态平稳的数字信号稳态特性的方法。如果能假设信号存在短时平稳特征，可以采用短时傅里叶分析方法。短时傅里叶分析方法也可称为时间依赖傅里叶变换，该方法是频域检测方法的一种，用稳态的频谱分析方法，处理非平稳信号。

钢琴演奏时，急板乐曲节拍速度较快，节拍为 240 拍/min 左右。可以假设钢琴乐音信号在 10ms 这样短的时间片段中，频谱结构在短时中稳定，所以采用短时傅里叶分析的短时平稳钢琴乐音可行。在这种情形下，短时傅里叶分析法可以分辨的乐音最低频率为 16Hz，而钢琴键盘能发出的最低频率约为 27.5Hz，说明整个钢琴发音频率段都处在短时傅里叶分析法可以分析的区域内。

人类的耳朵是一个天然的傅里叶分析器，不同频率的乐音会引发听觉神经系统产生复杂的反应，所以人耳能够感知一个乐音的复杂频谱结构，从而人类具有对乐器音色的感知能力。因此，采用频谱结构分析的方法，仿真了人类听觉系统的工作原理，是一种处理音频信号的自然方法。

钢琴的“a1”（对应钢琴键是 49A）的基频为 440Hz，除了发出 440Hz 的振动波外，同时还伴随产生了更高频率的泛音，由此形成了钢琴独特的音色，但是该键发音时，主要振动能量集中于 440Hz。振动能量最强的频率让听觉神经产生最强的刺激反应，所以人耳在确定乐器音高时，自然会以最大的振动能量所处的频率为判断依据。如果钢琴的“a1”振动最强的频率不是 440Hz 频点，则需要对该琴弦校音了。最优秀的钢琴调音师裸耳能判定 0.5Hz 频率误差。在本研究中，作者未能达到钢琴调音师的专业水平，无法裸耳准确判定基频频率。为了能够精确测定钢琴频谱，顺利开展频谱分析的后续研究，必须借助专业电子频率计。在实验器材中，增购了一台“musedo MT-70”校音器，如图 3-1 所示。



图 4-1 musedo MT-70 操作面板

该设备具有校音器、节拍器、定音器三合一功能，校音原理采用十二平均律，信号检测速度 $<20\text{ms}$ ，校音精度 $\pm 3\text{cent}$ (音分)，节拍精度 $<0.5\%$ 。该设备可以精确测定各种乐器发音的基频，误差为 $\pm 3\text{cent}$ ，并且可以发出精确频率的正弦波讯响。该设备拾音方式振动传感器、麦克风，通过实验在吉他、小提琴校音时两种拾音方式均可使用，但是在钢琴、洞箫、笛子校音时，振动传感器捕获不到正确的音频振动频率，仅能采用内置麦克风的方式，方法是将校音器放在乐器前方 $0.5\text{m}$ 至 $1\text{m}$ 的范围内。使用内置麦克风的方式，容易受到环境噪声的干扰，必须在夜深人静或密闭安静的环境中进行实验。

cent(音分)是为提高测量声音频率高低的准确度，计量上将每个“半音”等分为 $100\text{cent}$ ，以利计算声高误差率，十二平均律半音的音程距离相差 $1.0594630941(\sqrt[12]{2})$ 倍。将半音音程的距离再分成 $100$ 个单位，即 $100\text{cent}$ 。每 $1$ 音分之间的距离为 $1.0005777895(\sqrt[100]{1.0594630941})$ 倍。

例如高于 $440\text{Hz}$ 的半音为 $466.1638(440 \times 1.0594630941)\text{Hz}$ ，

两者之间的 $3\text{cent}$ 误差为 $0.7849(\frac{3 \times (466.1638 - 440)}{100})\text{Hz}$ ，

因此借助电子仪器，可以精确调整琴弦声高，以接近优秀调音师调校效果。

首先使用MATLAB按照傅里叶变换的数学原理编写FFT()函数，程序如下：

```
function data_sjf = fft(data_sj, nnum, varargin)
if isempty(data_sj)
    data_sjf = data_sj;
```

```
        data_sjf.Domain = 'Frequency';
    return
end
if nargin == 3
    rnfeq = 1;
else
    rnfeq = 0;
end
if nargin<2
    nnum = [];
end
if isempty(nnum)
    nnum = size(data_sj,'Num');
end
Nume = size(data_sj,'Nume');
if length(nnum)~=Nume
    if length(nnum)==1
        nnum = nnum*ones(1,Nume);
    else
        ctrlMsgUtils.error('Ident:dataprocess:fftLength')
    end
end
if strcmpi(data_sj.Domain,'frequency')
    ctrlMsgUtils.error('Ident:dataprocess:fftDataDomain')
end
y = data_sj.OutputData;
u = data_sj.InputData;
ts = data_sj.Ts;
for pkxe = 1:length(ts)
    if isempty(ts{pkxe})
        ctrlMsgUtils.error('Ident:dataprocess:fftSampling')
    end
end
ss = data_sj.TimeUnit;
if isempty(ss), ss= 's';end
for pkxe = 1:length(y)
```

```

Num = nnum(pkxe);
if floor(Num)~=Num || Num<=0
    ctrlMsgUtils.error('Ident:dataprocess:fftLength2')
end
Ts = ts{pkxe};
if isreal(data_sj) && ~rnfeg
    if fix(Num/2)==Num/2
        nnum1 = Num/2+1;
        freq{pkxe} = (0:Num/2)'/Ts*2*pi/Num;
    else
        nnum1 = (Num+1)/2;
        freq{pkxe} = (0:(Num-1)/2)'/Ts*2*pi/Num;
    end
else
    if fix(Num/2)==Num/2
        freq{pkxe} = [(-Num/2+1:-1),(0:Num/2)']'/Ts*2*pi/Num;
    else
        freq{pkxe} = [(-(Num+1)/2+1:-1),(0:(Num-1)/2)']'/Ts*2*pi/Num;
    end
end
Y1 = fft(y{pkxe},Num,1)/sqrt(min(Num));
U1 = fft(u{pkxe},Num,1)/sqrt(min(Num));
if length(y{pkxe})==1 || length(u{pkxe})==1;
    if isempty(Y1), Y1 = zeros(size(U1,1),0);end
    if isempty(U1), U1 = zeros(size(Y1,1),0);end
end
if isreal(data_sj) && ~rnfeg
    Y{pkxe}= Y1(1:nnum1,:);
    U{pkxe}= U1(1:nnum1,:);
else
    if fix(Num/2)==Num/2
        Y{pkxe} = Y1([(Num/2+2:Num),1+(0:Num/2)],:);
        U{pkxe} = U1([(Num/2+2:Num),1+(0:Num/2)],:);
    else
        Y{pkxe} = Y1([(Num+1)/2+1:Num),1+(0:(Num-1)/2)],:);
        U{pkxe} = U1([(Num+1)/2+1:Num),1+(0:(Num-1)/2)],:);
    end
end

```

```

        end
    end
    unit{pkxe} = ['rad/',ss];
end
data_sjf = data_sj;
data_sjf.Name = "";
data_sjf.Notes = {};
data_sjf.UserData = [];
data_sjf.InputData = U;
data_sjf.OutputData = Y;
data_sjf.SamplingInstants = freq;
data_sjf.Domain = 'Frequency';
data_sjf.Tstart = unit;
data_sjf = timemark(data_sjf,'c');

function Y = fft(NX, n_um, N_dim)
if nargin < 3
    N_dim = distributedutil.Sizes.firstNonSingletonDimension(size(NX));
else
    N_dim = distributedutil.CodistParser.gatherIfCodistributed(N_dim);
end
if nargin < 2 || (nargin > 1 && isempty(n))
    n_um = size(NX,N_dim);
else
    n_um = distributedutil.CodistParser.gatherIfCodistributed(n_um);
end
if ~isa(NX, 'codistributed')
    Y = fft(NX, n_um, N_dim);
    return;
end
codistributed.pVerifyUsing1d('fft', NX);
xDist = getCodistributor(NX);
if xDist.Dimension ~= N_dim
    localY = fft(getLocalPart(NX),n_um,N_dim);
    sz = xDist.Cached.GlobalSize;
    sz(N_dim) = n_um;

```

```

        codistr = codistributor1d(xDist.Dimension, xDist.Partition, sz);
        Y = codistributed.pDoBuildFromLocalPart(localY, codistr);
    else
        Y = dist_fft(NX,n_um,N_dim); %fft on codistributed data
    end

function Y = dist_fft(NX,n_um,N_dim)
if N_dim == 1
    Y = redistribute(NX,codistributor('1d', 2));
else
    Y = redistribute(NX,codistributor('1d', N_dim-1));
end
Y = fft(Y,n_um,N_dim);
if size(NX,N_dim) == n_um
    Y = redistribute(Y, getCodistributor(NX));
else
    Y = redistribute(Y,codistributor('1d',N_dim));
end

```

然后编写乐音的wav格式文件傅里叶变换频谱分析程序如下：

```

% 本程序功能是读入WAV波表文件
% 并转换为分贝频谱
clear all
close all
clear command windows
CDIR=dir('test/*.wav');
% 读取test目录下所有的WAV格式的声波文件
YB_dir_cell=struct2cell(CDIR)';
% 对矩阵转置是让文件名能按列排列
% 转换成元胞类型(cell)，从原始结构体(struct)
file_name=YB_dir_cell(:,1);
% 第一列是文件名
% 通过wavread函数读取wav文件数据，
% 并将其存储在DATA_Y和DATA_F中。
Ngh=5000;

```

```

fngbber=size(file_name,1);
% 求取表格行数，即文件个数
for N=1:1:fngbber
    CFN=char(file_name(N,1));
    % 将cell格式转换为string格式
    CFN=strcat('test/',CFN);
    % 校正CFN变量中记录的文件路径
    %[Y,Fs]=wavread(CFN,65535);
    % 读取当前声波文件前65535个点
    [Y,Fs]=wavread(CFN,Ngh);
    % 读取当前声波文件前65535个点
    DATA_Y(N,:)=Y(:,1);
    % 记录数据
    DATA_F(N,:)=Fs;
    % 记录采样率
end
%3、对数据做FFT运算，并绘制其频谱图。
for N=1:1:fngbber
    yy = fft(DATA_Y(N,:),Ngh);
    % 对数据做Ngh点FFT运算
    FFT_Y(N,:)=20*log10(abs(yy));
    % 对计算结果取模转化为分贝
    figure;
    %绘图
    ff = linspace(0,DATA_F(N),Ngh);
    % 在0-采样率之间取Ngh个频率点
    plot(ff,FFT_Y(N,:), 'r');
    % 绘制频谱
    grid on;
    axis([21 5001 -41 91]);
    % 调整修饰绘图，并存储绘图区域。
    set(gcf,'Position',[1 1 700 300]);
    % 设置绘图区域的大小
    tt = strcat(file_name(N,:), '-频谱分析图');

```

```

title(tt);
% 设置绘图区域标题
xlabel('Frequency (Hz) ', 'FontSize', 9, 'FontName', '宋体');
% 设置绘图区域x坐标轴标注字体 频率
ylabel('Amplitude (dB) ', 'FontSize', 9, 'FontName', '宋体');
% 设置绘图区域y坐标轴标注字体 振幅
% 绘图区域 Note = strcat('m=', Modindex(N,:));
% 绘图区域 text(1800,55,Note,'FontSize',14);
% 绘图区域 在400, 300位置添加标注
set(gca, 'FontSize', 9);
% 设置绘图区域坐标轴字体
set(gcf, 'visible', 'on');
% 设置绘图区域显示图片
% 绘图区域 tt=strrep(file_name(N,1), '.wav', '');
% 绘图区域 tt=str(N);
% 绘图区域 %tt = strcat(file_name(N,:), '.emf');
tt=ngh2str(N);
saveas(gcf, tt, 'emf');
% 存储绘图区域为emf格式

```

End

利用MT-70发出440Hz频率标准正弦波乐音，以采样频率44100Hz，16bit，录制0.5s，保存文件名为“440Hz.wav”，采用傅里叶变换程序分析该音频文件前部短时片断，获得该乐音频域波形，如图4-2。该图表明正弦波的振动能量集中于440Hz，在频谱图中只形成一个振动频率。

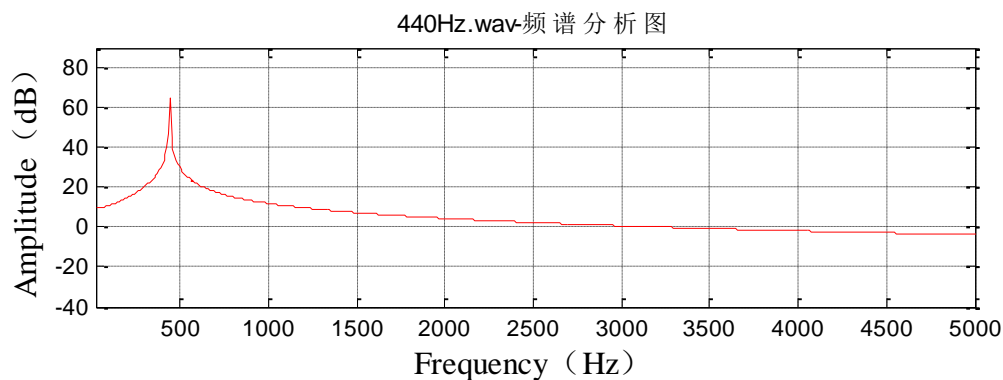


图 4-2 440Hz 正弦波频谱分析图



在专业录音棚内设置一架立式钢琴，利用MT-70校正“a1”(第49号琴键)对应的琴弦，使之发出440Hz的国际标准音。把录音话筒放置在钢琴正前方3米处采样，请演奏者正常弹奏“a1”(第49号琴键)，不踩任何踏板，录制声音直至声音自然消失，录制文件取名为“la.wav”。采用傅里叶变换程序分析后，获得频域波形，这是典型的钢琴单音奏响时域频域波形，如图4-3所示。对比图4-3与图4-2可见，钢琴频谱中充满了大量倍频振动，这是钢琴音色浑厚浓郁的原因。钢琴单音的振动能量分散于多个频点上，其中基频440Hz能量最高，在2倍频880Hz，3倍频1320Hz、4倍频1760Hz、5倍频2200Hz等处都具有较大的振动能量。从钢琴弦振动的方程来解析，钢琴弦整体振动产生的最低的音是基音，在发出基音的同时，钢琴弦1/2、1/3、1/4等各部分也是同时振动，产生了泛音。泛音的组合决定了特定的音色，并能使人明确地感到基音的响度。乐器和自然界里所有的音都有泛音。通过本实验可知，钢琴单音音高由振动能量最大的基频决定，而人耳感知的钢琴音色受到复杂频谱结构中泛音的影响。

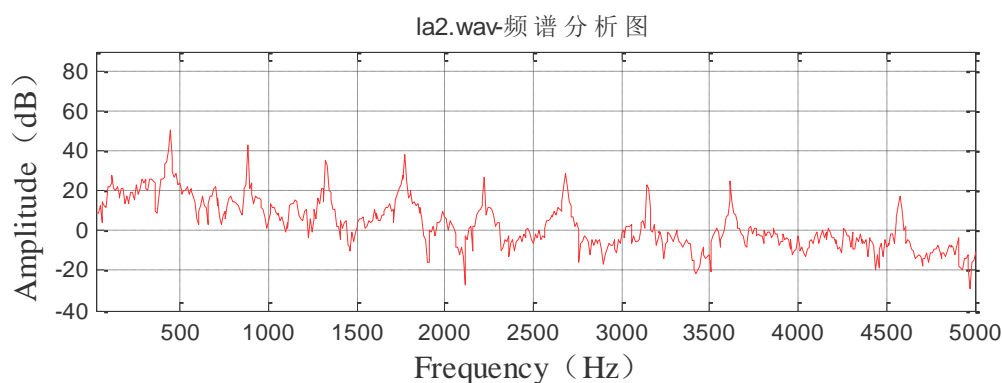


图 4-3 440Hz 钢琴单音频谱分析图

以上实验说明钢琴频谱图充满了大量的倍频泛音，钢琴特定的泛音的组合决定了钢琴特定的音色，钢琴频谱中充满了大量倍频振动，这是钢琴音色浑厚浓郁的原因，所以钢琴能成为“乐器之王”。

## 4.2 音色特征矩阵的提取

以上“钢琴振动方程”、“短时傅里叶分析”可知在钢琴乐音中存在着复杂频谱结构，其中高频的谐波振动频率是乐音基本频率的整倍数。而人耳是通过确定复杂频谱结构中的基频来判断乐音音高。

一些研究者发现一些特定函数波形类似于钢琴波形。可以选取的特定函数有 sinc 函数、高斯函数、柯西函数。研究者还发现可以用这些特定函数来近似表达钢琴乐音的频域波形。对比各函数的仿真效果，其中加权柯西函数能够描述基频和多个倍频信息，通常采用5次倍频的方法，仿真效果最接近钢琴真实音色。令  $Y(j\omega)$  为离散时间信号  $y(n)$  的离散傅里叶变换。则  $Y(j\omega)$  可以由如下算式表达：

$$Y_i(j\omega) = S_i(j\omega) \cdot F_i(j\omega)$$

$$S_i(j\omega) = \begin{cases} A_i \frac{\alpha_i}{\alpha_i^2 + (\omega - \omega_i)^2} & , \quad \omega_i - \alpha_i \leq \omega \leq \omega_i + \alpha_i \\ A_i \frac{2}{|\omega - \omega_i|} & , \quad \text{其它情况} \end{cases}$$

上式中， $\omega_i$  为指定音的基频或者倍频， $A_i$  为基频点或者倍频点的振幅， $\alpha_i$  用来调节基频或者倍频附近波形宽度， $F_i$  为关于正弦和余弦的函数。

采用被正弦余弦加权的柯西函数重构钢琴音色时，选取恰当参数，可以在频域及时域上构造出一个与实际演奏采集波形相当近似的波形。在频域上，基频到五次倍频及其附近区域内的波形很好的描绘了钢琴音高、时值、强度及其音色这四个音乐基本要素。而基于加权柯西函数的频域波形构造，可以通过反 FFT 变换准确模拟钢琴音的音色、音高和强度。通过计算机模拟出钢琴上 88 个音的演奏效果，从而可以完成钢琴乐曲的计算机演奏。

虽然一些研究者尝试用特殊函数对钢琴音色加以描述，在适当的参数支持下，特殊函数可以模拟出较好的钢琴音色。但是在本论文中发现，为了便于模型的推导与简化，许多研究者都是取5次倍频为建模计算样本，以国际标准音440Hz为例，5次倍频仅为2200Hz，即函数模拟的钢琴的国际标准音的泛音范围为440Hz - 2200Hz。而人耳的频率范围为 20Hz-20000Hz左右，因此大部分音乐爱好者经过钢琴音色识别训练后，还是可以发现函数模拟的钢琴泛音不足，音色不够丰满，表达力不足。另外函数模拟钢琴时，需要临时进行复杂的函数计算，如果是由MCU(单片机)来承担电子合成工作，则MCU负荷较大。

通过本文大量实验研究，认为完全可以采用一种更加高效的钢琴音电子合成方法。建立一个矩阵，记录一个钢琴音各倍频点的相对振幅比例关系，在电子合成时，只需把振动总能量按相对振幅的比例分配到各倍频点上，即可再现钢琴音色。如果矩阵记录50个倍频点的振幅比例，以国际标准音440Hz为例，50次倍频为22000Hz，即已超越了人耳识别范围，这样再挑剔的人类耳朵也无法发现电子合成

的钢琴音泛音不足。

采用傅里叶分析方法，分析钢琴国际标准音440Hz的各倍频点的相对振幅，因为只要获得振动能量分配的比例关系，因此无需变换为db(分贝)单位。应用Matlab开发乐音“频谱特征提取模块”，编程如下：

```
% 对钢琴原声文件“la.wav”作傅里叶分析，
% 只要对比相对值，无需变换为分贝单位
% 对钢琴录音文件左右声道的信号直接相加，以简化研究工作量
[y1,fb,bit]=wavread('la.wav');
% 读取声波格式的录音文件
y0=y1(:,2)+y1(:,1);
% 左右声道的信号直接相加
[n1,n2]=size(y0);
t=(1:n2)/fb;
figure;
plot(w,fy(1:2^15+1));
grid on;
set(gcf,'Position',[0 0 600 200]);
% 设置绘图区域的大小
tt = '440Hz钢琴音各频点振幅';
title(tt);
% 设置绘图区域标题
axis([21 5001 -51 1201]);
% 设置绘图区域坐标范围
xlabel('频率（Hz）','FontSize',9,'FontName','宋体');
% 设置绘图区域x坐标轴标注字体
ylabel('相对振幅','FontSize',9,'FontName','宋体');
% 设置绘图区域y坐标轴标注字体
set(gca,'FontSize',9);
% 设置绘图区域坐标轴字体
set(gcf,'visible','on');
% 显示绘图区域图片
saveas(gcf,'440Hz钢琴乐音各频点振幅','emf');
```

```
% 存储绘图区域图标为emf格式
```

```
sound(y0,44.1e3)
```

```
%发出录音原声
```

上述代码运行后，获得440Hz钢琴乐音各频点振幅图，如图3-4所示。

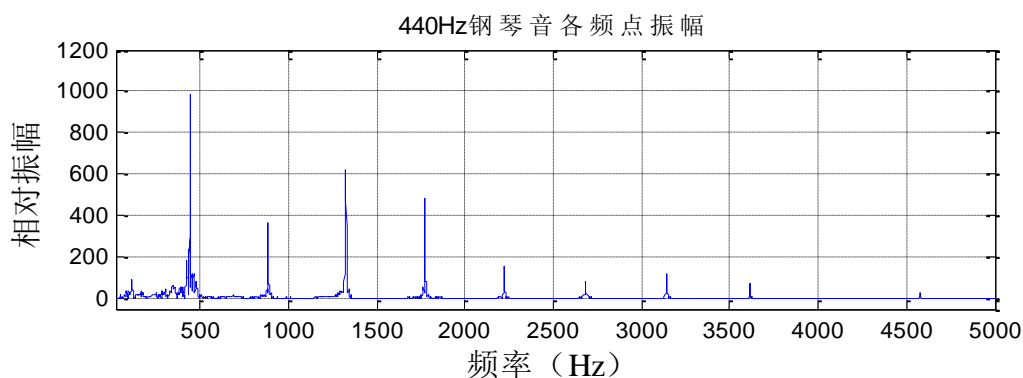


图 4-4 钢琴乐音各频点振幅图

随后，获取各倍频点相对振幅描述二维矩阵如下(节取26行)，第一列为440Hz倍频点，第二列为相对振幅。

440*1,	987.8;
440*2,	368.6;
440*3,	620.2;
440*4,	483.9;
440*5,	156.7;
440*6,	83.62;
440*7,	120.1;
440*8,	70.73;
440*9,	5.348;
440*10,	24.41;
440*11,	27.35;
440*12,	21.3;.;
440*13,	10.31;
440*14,	6.477;
440*15,	15.91;
440*16,	3.495;
440*17,	2.546;
440*18,	0.4751;

```

440*19,    0.8858;
440*20,    0.3792;
440*21,    0.6012;
440*22,    0.4224;
440*23,    0.1538;
440*24,    0.1454;
440*25,    0.2032;
440*26,    0.0483;

```

.....

从描述矩阵也可以看出，以基频振动为1的话，前17个倍频点相对基频振动振幅在1/1000以内，对音色都有一定的影响，因此采用柯西函数重构钢琴音色时，采样前5个倍频点，确实不能如实表现出钢琴丰富的泛音。在后期仿真实验中，也发现矩阵法将比函数建模的方式获得更好的音色效果。

观察矩阵可知前17个倍频点相对基频振幅在1/1000以内。因此得出一个结论：如果对电子合成钢琴音色要求较高，可以发出50个倍频点的谐振，这已经超过了人耳的感知范围(20Hz-20000Hz)，钢琴音色再现将无可挑剔。如果要求对电子合成钢琴的速度要求较高，可以发出前17个倍频点的谐振，此时钢琴音色仿真效果已很理想，是效果和效率的一个均衡点。

### 4.3 音强包络函数的提取

通过听取大量击弦类乐器的乐音演奏，可以发现该类乐器的乐音音强不同于拉弦类、吹奏类乐器。击弦类乐器的音强在短时间内存在由强至弱的快速变化，而拉弦类、吹奏类乐器的音强可以保持一段时间。如果要完美合成钢琴乐音，应该提取音强和时间关系的包络函数，让电子合成音遵循包络函数的变化过程。应用Matlab程序开发乐音“振幅特征提取模块”。

```

[y1,fb,bit]=wavread('la.wav');
y0=y1(:,2)'+y1(:,1)';
[n1,n2]=size(y0);
t=(1:n2)/fb;
figure;
plot(t,y0);
grid on;
set(gcf,'Position',[0 0 600 200]);

```

```

% 设置绘图区域的大小
tt = '440Hz钢琴波形图';
title(tt);
% 设置绘图区域标题
axis([0 2 -0.5 0.5]); %x坐标 坐标范围
xlabel('时长 (s)', 'FontSize', 9, 'FontName', 'Times New Roman');
% 设置绘图区域x坐标轴标注字体
ylabel('振幅', 'FontSize', 9, 'FontName', 'Times New Roman');
% 设置绘图区域y坐标轴标注字体
set(gca, 'FontSize', 9);
% 设置绘图区域坐标轴字体
set(gcf, 'visible', 'on');
% 显示绘图区域图片
saveas(gcf, '440Hz钢琴波形图', 'emf')
% 将绘图区域图样保存成图片

```

应用该“振幅特征提取模块”分析录音获得的440HZ钢琴乐音(未踩延时踏板)的振幅特征，如钢琴单音振幅波形图(图3-5)所示，波形的长尾巴通过功放技术放大并监听，判定为录音室的环境噪声和话筒的电气噪声，所以与钢琴弦振动无关。可见钢琴弦振动的振幅波形是从弱激增至强，再从强缓慢衰退为弱，直至波形消失。

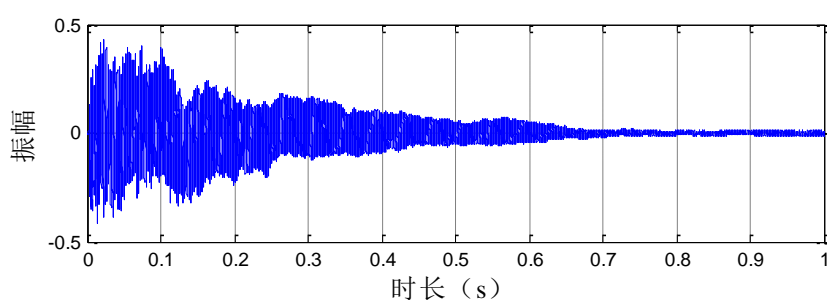


图 4-5 钢琴乐音振幅波形图

由此可见，钢琴电子合成模块的另一个功能重点在于能否在时间上准确分配振动能量，仿真钢琴弦受敲击发音的振幅能量在时间上分布特征。在仿真阶段，尝试了一些包络函数，试图模拟钢琴弦振幅特征。多次实验发现，

$$f(x) = \frac{x^2}{e^x} \quad (x \geq 0)$$

包络函数和真实曲线较匹配(图3-6)。在电子合成时,保证了前段振动能量上升较快,随着时间的延长,振动能量逐渐下降,如果检测到延时踏板被踏下的状态,则减缓振幅下降趋势。包络函数技术能够保证乐器演奏的单个音符自身变化柔和,音符间切换平顺。每个音符振动能量衰减类似于钢琴、吉他这些击弦、弹拨类乐器,为后续的钢琴琴声电子合成建立了良好的基础。

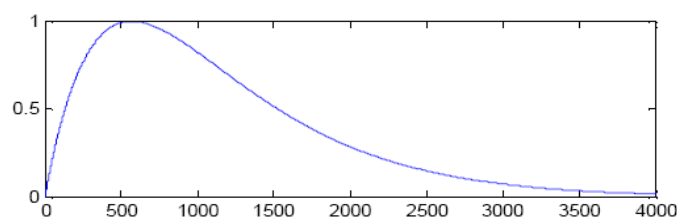


图 4-6 钢琴乐音振幅包络函数

#### 4.4 乐音特征数字化识别

综上所述,乐器乐音特征数字化识别流程图见图4-7。

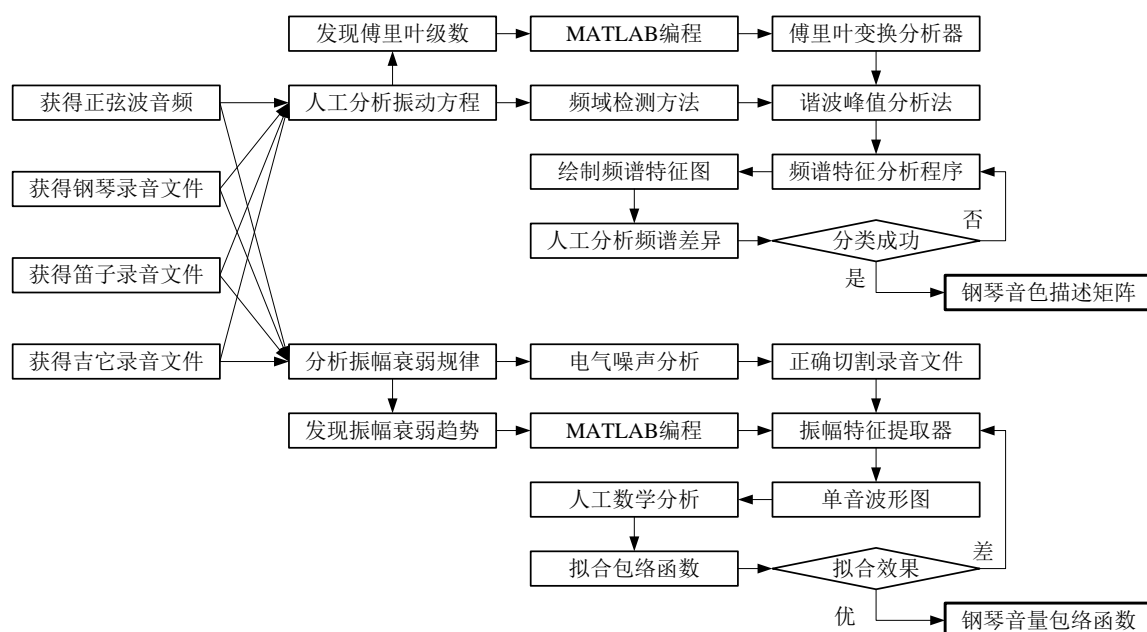


图 4-7 乐音特征的数字化识别流程图

采用自开发的傅里叶级数频谱分析代码,分别对正弦波、钢琴、笛子、吉它乐音频谱分析,确定了不同的乐器在基音各倍频点振幅各不相同,提取的音色描述矩阵可以用于各种乐器的乐音分类识别。说明频谱描述矩阵可以有效地区分不

同乐器的音色。

同时采用自开发的振幅特征提取代码，分别对正弦波、钢琴、笛子、吉它乐音振幅和时间关系分析，确定了不同乐器的包络函数各不相同。利用好包络函数，可以模拟出管类乐器、弦类乐器的发音时音强振幅特征，甚至可以模拟出击打类弦乐器和弹拨类弦乐器的细微差异。

## 4.5 小结

本章在综述钢琴音色模拟的研究进展时，发现一些研究者采用了加权柯西函数表达了钢琴5倍频谐振，实现了钢琴音色的电子合成，此法效果极高，被广泛应用于简单电子琴产品。但是后续实验也发现，人耳的频率范围为20Hz-20000Hz，一些经过钢琴训练的专业人士可以发现5倍频仿真的钢琴电子合成音泛音域不足，缺乏表现力。

本章随后采用傅里叶分析方法获得钢琴50个倍频点的振幅相对比例值，并保存成描述钢琴音色的二维矩阵。观察矩阵可知前17个倍频点相对基频振幅在1/1000以内。因此如果对电子合成钢琴音色要求较高，可以发出50个倍频点的谐振，这已经超过了人耳的感知范围，音色再现无可挑剔。如果要求对电子合成钢琴的速度要求较高，可以发出17个倍频点的谐振，此时钢琴音色仿真效果已很理想，是效果和效率的一个均衡点。

本章还分析了各种乐器发音时振幅和时长的关系，提出了可利用包络函数来进一步仿真击打弦乐器、弹拨弦乐器、吹奏管乐器的发音时振动能量衰减的细微差异，可使电子合成的音乐更加逼真。



## 第五章 基于 Matlab 的钢琴乐音仿真

采用MATLAB语言构建快速原型，尝试对钢琴乐音的仿真。首先从最简单的正弦波乐音的仿真入手，建立一个正弦波音乐的演奏系统，然后逐步提高难度，并在客户互动中，改良音色，最后实现仿真钢琴乐音的目标。

### 5.1 正弦波电子合成

以《东方红》简谱的第一小节为例，该小节简谱见图5-1，首先采用十二平均律计算出该小节每个乐音的基频，在MATLAB中生成振动幅度相对值为1，抽样频率为8kHz的正弦信号表达这些乐音，用MATLAB的“sound”函数可以把电子合成的音乐信号矩阵转变音箱的振动信号，从而实现了乐音的仿真。

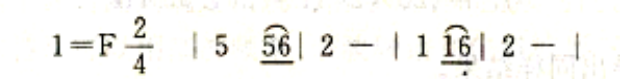


图 5-1 东方红第一小节简谱

由图5-1可知《东方红》的曲调为1=F，说明简谱音符“1”对应的频率为349.23Hz，据此可以计算出其他乐音的频率，例如简谱音符“5”对应的频率为

$$f_5 = 349.23 \times 2^{7/12} = 523.25 \text{ Hz},$$

以此类推计算出第一小节各乐音对应的频率如表5-1所示。

表 5-1 东方红第一小节频率表

音符	5	5	6	2	1	1	6̇	2
频率(Hz)	523.25	523.25	587.33	392	349.23	349.23	293.66	392

然后，根据节拍标识，可知本曲每小节有两拍，由此确定每个乐音的持续时间。设一拍的时间是0.5s，因此第一小节的各乐音的音长如表5-2所示。

表 5-2 东方红第一小节节拍表

音符	5	5	6	2	1	1	6̇	2
时间(s)	0.5	0.25	0.25	1	0.5	0.25	0.25	1

在MATLAB中设定仿真该乐曲所用的抽样频率为  $f_s = 8000 \text{ Hz}$ ，也就是所在1s内要仿真产生8000个数据点，一个乐音占用的点数多少，即表示出该乐音的音长。用一个矩阵来存储电子合成的音乐信号，在用MATLAB的“sound”函数播放即可[20]。

根据以上乐理分析，编写MATLAB电子合成正弦波音乐程序：

```
%抽样频率为8000Hz，存入cypl变量中
cypl=8000;
%把各个乐音对应的频率存入矩阵fdfh
fdfh=[523.25 523.25 587.33 392 349.23 349.23 293.66 392];
%把各个乐音的抽样点数存入矩阵yfsc，即表达了各音符的演奏时长
yfsc=cypl*[1/2,1/4,1/4,1,1/2,1/4,1/4,1];
%N的数值表示这小节音乐的总抽样点数
N=length(yfsc);
%用cyxy向量来储存抽样点
cyxy=zeros(1,N);
yydxh=1;
for yydxh=1:N
    %利用循环产生抽样数据，yydxh表示乐音编号
    t=1/cypl:1/cypl:yfsc(yydxh)/cypl;
    %以上指令产生第yydxh个乐音的抽样点
    cyxy(n:n+yfsc(yydxh)-1)=sin(2*pi*fdfh(yydxh)*t);
    %以上指定是抽样点对应的振幅值
    n=n+yfsc(yydxh);
end
sound(cyxy,8000);
%利用“sound”函数播放音乐
wavwrite(cyxy,'dfhtys.wav')
%将乐音矩阵存成WAV格式的音乐文件
plot(cyxy);
%绘制波形图
```

该程序段以完美的正弦波演奏了东方红第一小节，并生成了一个“dfhtys.wav”音乐文件。正弦波演奏乐曲听起来十分纯静和柔和，接近于箫声，但因为无任何泛音成份，因此人耳可以感知该音乐的音色和钢琴音色相差甚远。

## 5.2 包络函数的应用

在以上代码演奏乐曲时，不同音符之间有时会发出有“啪”的杂音，这是因

为上一个音符和下一个音符的相位经常错位，不能平滑过渡，相位产生了突变，形成高频分量，发出杂音。在电子合成乐器中通常采用包络技术来消除此类音符过渡杂音。常见的包络函数为指数衰减，最简单衰减方式是对每个音乘以  $e^{-\alpha t}$  因子，在实验中首先加的是  $e^{-1.5t}$  的衰减，不管是什么音符，都采用相同的衰减速度，但是通过人耳判听，可以发现噪音减弱但并没有消除，所以播放的音乐效果不是很好，同时还感觉音乐起伏性不强。于是尝试用不同速度的衰减算法，根据乐音音长标识来确定衰减速度的快慢，所以音长越长的乐音衰减越慢，音长越短的乐音衰减越快。在上一个正弦波音乐程序的基础上加上包络函数的变换，编写程序如下：

```
% 设置抽样频率为8000Hz，存入变量cypl
cypl=8000;
fdfh=[523.25 523.25 587.33 392 349.23 349.23 293.66 392];
% 设置第一小节各乐音对应的频率
yfsc=cypl*[1/2,1/4,1/4,1,1/2,1/4,1/4,1];
% 按节拍不同，设置各乐音的抽样点数，即演奏时长
N=length(yfsc);
% N为这第一小节音乐的总抽样点数
cyxy=zeros(1,N);
% 用cyxy向量来储存各乐音抽样点矩阵
yydxh=1;
for yydxh=1:N
    % 利用循环产生抽样数据，yydxh表示乐音编号
    t=1/cypl:1/cypl:yfsc(yydxh)/cypl;
    % 产生第yydxh个乐音的抽样点
    G=zeros(1,yfsc(yydxh));
    % G为存储包络函数数据的矩阵
    G(1:yfsc(yydxh))=exp(1:(-1/yfsc(yydxh)):1/8000);
    % 产生各音符包络点
    cyxy(yydxh:yydxh+yfsc(yydxh)-1)=
        sin(2*pi*fdfh (yydxh)*t).*G(1:yfsc(yydxh));
    % 给第yydxh个乐音加上相应的包络
    yydxh=yydxh+yfsc(yydxh);
end
```

```

sound(cyxy,8000);
%播放包络后的正弦波音乐
wavwrite(cyxy,'dfhbl1.wav')
%将包络后的正弦波音乐保存成dfhbl1.wav波表文件
plot(cyxy);
%绘制波形图

```

以上代码演奏了东方红乐曲的第一节，采用指数衰减的包络函数技术来减轻各音符的过渡杂音，乐音衰减快慢取决于演奏时长，衰减速度不一，音乐听起来不单调，具起伏变化，更加活泼一些，图5-2是指数衰减包络方法效果示意图。但是音乐专业人士在试听乐曲片断时，仍然指出，可以听见音符和音符之间发生了突变，还存在因为相位突变产生的轻微杂音，两音符变化过程过渡得较为生硬，需要对原型进一步优化。

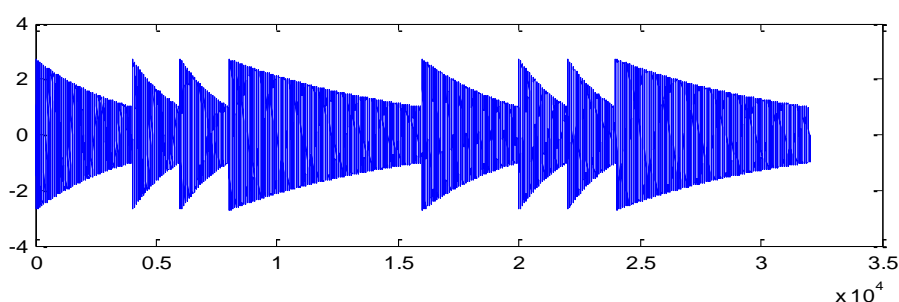


图 5-2 指数衰减包络效果示意图

随后尝试四段直线包络方法，能保证前一个音符能量消失后，才发出后续音符，这样就能有效消除，音符交替时的轻微杂音。四段直线包络方法实现了每个乐音都经过“冲激”、“衰减”、“持续”、“消失”四个音强变化阶段。由图5-3可以看出四段直线段构成一次包络，因此只要确定了每段线段的前后端点坐标，即可用端点坐标求出直线方程，直线方程可以用斜截式通式写出，该四段直线包络技术将作用于每个音符，所以可用简单的循环，逐个对原有音符的正弦波信号进行包络，最后完成对所有音符的包络。代码如下：

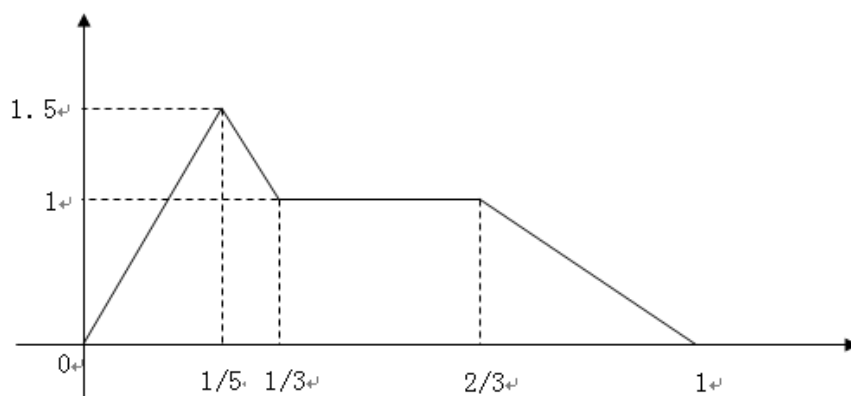


图 5-3 四段直线包络方法

```

%设置抽样频率为8000Hz，存入变量cypl
cypl=8000;
fdfh=[523.25 523.25 587.33 392 349.23 349.23 293.66 392];
%第一小节各个乐音对应的频率表
yfsc=cypl*[1/2,1/4,1/4,1,1/2,1/4,1/4,1];
%第一小节各个乐音的抽样数据
N=length(yfsc); %第一小节乐曲的总抽样点数
cyxy=zeros(1,N); %用cyxy向量来储存抽样数据
yydxh=1;
for yydxh=1:N
    %利用循环产生每个音符的抽样数据，yydxh表示乐音编号
    t=1/cypl:1/cypl:(yfsc(yydxh))/cypl;
    %产生第yydxh个乐音的抽样数据
    P=zeros(1,yfsc(yydxh));
    %P为存储乐音包络数据的矩阵
    Lzx=(yfsc(yydxh))*[0 1/5 332/999 332/499 1];
    %四段包络直线端点对应的横坐标
    sdx=[0 1.5 1 1 0];
    %四段包络直线端点对应的纵坐标
    s=1;
    b=1:1:yfsc(yydxh);
    %产生四段包络直线抽样数据
    for dx=1:4
        P(s:Lzx(dx+1)-1)=(sdx(dx+1)-sdx(dx))/(Lzx(dx+1)-;

```

```

Lzx(dx))*(b(s:Lzx(dx+1)-1)-Lzx(dx+1)*ones(1,Lzx(dx+1)-s))+;
sdx(dx+1)*ones(1,Lzx(dx+1)-s);
% 四段包络直线方程斜截通式
s=Lzx(dx+1);
end
cyxy(n:n+yfsc(yydxh)-1)=sin(2*pi*fdfh (yydxh)*t).*P(1:yfsc(yydxh));
% 给第yydxh个乐音加上包络
n=n+yfsc(yydxh);
end
sound(cyxy,8000);
% 按8000播放包络后的音乐信号
wavwrite(cyxy,'dfhbl2.wav')
plot(cyxy);
set(gcf,'Position',[1 1 700 300]);
% 设置绘图区域的大小
set(gca,'FontSize',12);
% 设置绘图区域坐标轴字体为12号字
set(gcf,'visible','on');
% 设置绘图区域显示图片
saveas(gcf,'dfh_4qbl.emf','emf');
% 设计绘图区域的图样存储绘图图为emf格式

```

四段直线法包络效果见图4-4，可以看到前一个乐音一直衰减到0，后一个乐音从0开始增加，前后两音的过渡平滑，因此彻底消除了音符之间的轻微杂音。

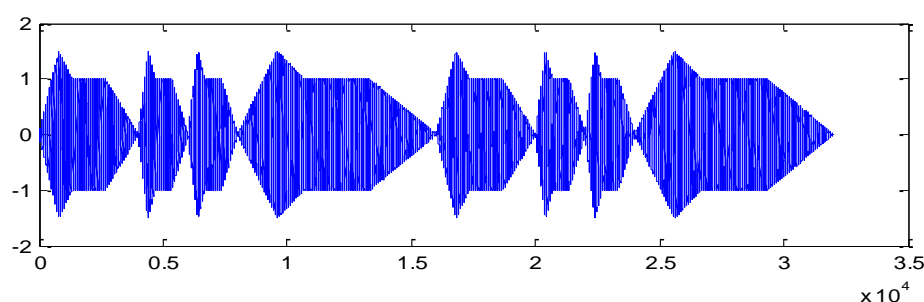


图 4-4 四段直线法包络效果示意图

在快速原型使用四段直线包络法演奏正弦波乐曲时，音乐专业人士在细心品味后指出四段直线法包络算法虽然使音符过渡平顺，但是各音符在发声时略显生

硬，音量变化不符合击弦类乐器的现象。因此尝试对四段直线包络技术进一步改良。随后录制440Hz钢琴国际标准音，录音过程1s，编写MATLAB代码对音量变化进行分析。发现了真实的钢琴包络函数接近于函数  $f(x) = x^2 / e^x \ (x \geq 0)$ 。所以四段直线法包络算法在直线包络仿真过程中略为生硬，不能完美表现钢琴乐音从击弦发音到声强消失的柔和变化的自然过程。

设计新的音符包络函数为  $f(x) = x^2 / e^x \ (x \geq 0)$ ，因为该函数的形状极其柔和，前段振动能量上升较快，保证了电子合成器高效发音，随着时间的延长，振动能量逐渐下降，在当前音符发音时间结束时，振动能量恰好降为0，为演奏下个音符做好准备，避免了音符切换时相位差异发生的杂音。这个包络曲线可以通用于每个音符中，曲线长度由音符音长控制，可任意设定，最大振幅由弹奏时的音强决定，也可任意设定。编写代码如下。

```

cyl = 8000;
% 设置抽样频率为8000Hz，存入变量cyl
T = 1/cyl;
% 根据抽样频率设置抽样周期
jpsc = 0.5;
% 根据乐曲的节拍规定，设置每一拍持续时间
jpsc1 = 0:T:4*jpsc;
% 表示全音符音长
jpsc2 = 0:T:2*jpsc;
% 表示二分音符音长
jpsc4 = 0:T:1*jpsc;
% 表示四分音符音长
jpsc8 = 0:T:0.5*jpsc;
% 表示八分音符音长
dgyy1 = sin(2*pi*523.25*jpsc4) .* zdybl(length(dgyy1));
dgyy2 = sin(2*pi*523.25*jpsc8) .* zdybl(length(dgyy2));
dgyy3 = sin(2*pi*587.33*jpsc8) .* zdybl(length(dgyy3));
dgyy4 = sin(2*pi*392.00*jpsc2) .* zdybl(length(dgyy4));
dgyy5 = sin(2*pi*349.23*jpsc4) .* zdybl(length(dgyy5));
dgyy6 = sin(2*pi*349.23*jpsc8) .* zdybl(length(dgyy6));

```

```

dgyy7 = sin(2*pi*293.66*jpsc8) .* zdybl(length(dgyy7));
dgyy8 = sin(2*pi*392.00*jpsc2) .* zdybl(length(dgyy8));
yy = [dgyy1 dgyy2 dgyy3 dgyy4 dgyy5 dgyy6 dgyy7 dgyy8];
%-----
sound(yy,8000);
%播放自定义函数包络后的正弦波音乐
plot(yy);
set(gcf,'Position',[1 1 700 300]);
% 设置绘图区域的大小
set(gca,'FontSize',12);
% 设置绘图区域坐标轴字体
set(gcf,'visible','on');
% 设置绘图区域显示图片
saveas(gcf,'dfhzdybl.emf','emf');
% 存储绘图区域的图样为emf格式
wavwrite(wave,'dfhzdybl.wav')
% 将包络过的乐音集生成一个wav音乐文件
return

% 建立一个自定义包络函数
function shap_yy = zdybl(yylen)
yyt = 1:yylen;
shap_yy = yyt /exp(yyt /yylen*7);
shap_yy = shap_yy /max(shap_yy);
return

```

程序运行后，采用自定义函数包络效果示意图见图5-5。通过对比实验，该自定义包络函数试听效果优于四段直线算法，音符间切换平顺，单个音符自身变化柔和。每个音符振动能量衰减类似于钢琴、吉他这些击弦、弹拨类乐器，该快速原型在音强模拟上获得了用户较好评价。



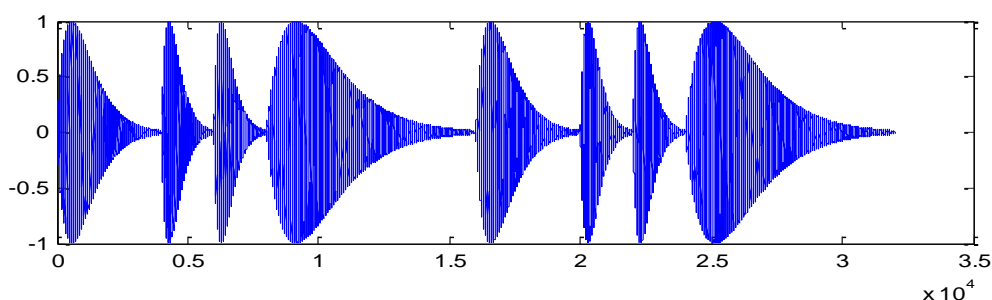


图 5-5 自定义函数的包络效果示意图

### 5.3 钢琴乐音的电子合成

以上研究表明，利用钢琴音色描述矩阵可以电子合成钢琴乐音。如果对电子合成钢琴音色的要求很高，可以发出50个倍频点的谐振，这已经超过了人耳的感知范围，音色再现无可挑剔。如果要求对电子合成钢琴的速度要求较高，可以发出17个倍频点的谐振，此时钢琴音色仿真效果已很理想，是效果和效率的一个均衡点。

利用Matlab编写函数piano()，该函数直接内置了钢琴音色描述矩阵，计算过程简捷，仿真了钢琴音色中的25次谐波，能够快速产生一个完美的钢琴单音。

```
%本函数输出钢琴波形数据
function Y=piano(F,phy,t)
f=F.*(1:30);
%产生各个谐波频率
%fft分析，第一列为频率，
%第二列为各谐波分量
Af=[440*1,987.8;440*2,368.6;440*3,620.2;440*4,483.9;440*5,156.7;440*6,83.62
;440*7,120.1;440*8,70.73;440*9,5.348;440*10,24.41;440*11,27.35;440*12,21
.3;440*13,10.31;440*14,6.477;440*15,15.91;440*16,3.495;440*17,2.546;440
*18,0.4751;440*19,0.8858;440*20,0.3792;440*21,0.6012;440*22,0.4224;440*
23,0.1538;440*24,0.1454;440*25,0.2032;440*26,0.0483];
[n1,n2]=size(t);
y=zeros(25,n2);
for i=1:25
    %计算至25次谐波
```

```

        y(i,:)=1.5e-4.*Af(i,2).*exp(-2.5.*(t-phy))...
        .*heaviside(t-phy).*sin(2.*pi.*f(i).*(t-phy));
    end
    Y=sum(y);

```

利用Matlab编写主程序，该程序调用上文介绍的钢琴音色合成函数 piano(F,phy,t)，以钢琴的音色演奏《致爱丽丝》片段，并生成一个WAV文件，达到了良好的仿真效果。

```

%简谱表
%----1---2---3---4---5---6---7----
fl=[262. 294. 330. 349. 392. 440. 496.];
flt=[277. 311. 0.00 370. 415. 466. 0.00];
fm=[523. 578. 659. 698. 784. 880. 988.];
fmt=[554. 622. 0.00 740. 831. 932. 0.00];
fh=[1046 1175 1318 1397 1568 1760 1976];
fht=[1109 1245 0.00 1480 1661 1865 0.00];
%简谱序列：
pu=[3 2 3 2 3 7 2 1 6];
[m,n]=size(pu);
%频率序列：
F=zeros(1,n);
F=[fh(3),fht(2),fh(3),fht(2),fh(3),fm(7),fh(2),fh(1),fm(6)];
%时间轴位置，即相移
Phy=[0:1/3:8/3];
t=0:1/44.1e3:5;
[t1,t2]=size(t);
y=zeros(n,t2);
for i=1:n
    y(i,:)=piano(F(i),Phy(i),t);
end
Y=sum(y);
sound(Y,44.1e3)
wavwrite(Y,44.1e3,'ailisi.wav')

```

## 5.4 快速原型的测试工作

目前国内软件公司通常采用生命周期法，严格按照制定计划、需求分析、软件设计、程序编写、软件测试和运行维护六个阶段来完成。最终客户要等到第五个阶段软件测试才能见到系统实物。

最终客户通常对计算机知识了解不多，在前四阶段看到几万字的软件开发文档时，没见到实物，觉得文档描述一切都还行，能满足工作需要，因此在前四个阶段的甲、乙双方交流时，都没有提出较尖锐的意见。而在第五阶段时，客户第一次见到系统实物时，期望越高，失望越大，往往一下子提出许多意见，最后导致软件工程验收不通过。

此类现象特别容易发生在软件工程师不熟悉的特殊专业领域内，因为软件工程师对特殊的专业知识不熟悉，无法正确理解客户的需求描述，所以可能从需求分析阶段就出现了方向性错误，投入了大量人力物力，直到软件测试阶段才发现错误，延误软件工期，造成了经济损失。

快速原型法即在系统开发初期，用最短的时间向最终客户提供一个简单模型，提供给客户测试，由客户评价模型优缺点。软件公司根据意见，修改模型，尽快把新模型提交给客户测试，快速原型法改变了甲乙双方的交互方式，能及早发现理解错误，保证软件开发的方向和进度。

本系统是音乐领域的专业系统，不是常规的管理系统，软件公司的经验不足。而且验收的唯一标准是钢琴仿真效果要达到音乐专业人士认可，因此，为了降低软件工程开发风险，必须采用快速原型法。

就是在系统开发之初，尽快给用户构造一个新系统的模型（原型），反复演示原型并征求用户意见，开发人员根据用户意见不断修改完善原型，直到基本满足用户的要求后，原型阶段结束，进入成品软件开发阶段，这种软件开发方法就是快速原型法。

快速原型法的每一次原型开发，都快速经历了需求分析、详细设计、代码实现、测试阶段各项工作，但是各项工作被融合在一起，才能在很短的时间内快速提交新的原型。快速原型螺旋式重复四个阶段的工作，不断完善系统功能，不断逼近客户的需求，所以无法象“生命周期法”那样能够精确地分出“需求分析、详细设计、代码实现、测试”四个阶段。所以快速原型法只能划分为“快速原型、成品软件”两大阶段。

但是测试工作是快速原型阶段中的最重要工作，每一次测试工作，都确定下

一个原型进一步完善的方向。在本系统的快速原型阶段，经历了许多次测试工作。

例如，本次设计从制定计划至提交第一个原型仅用了 7 天时间，第一个原型应用了录音技术，录制了 88 个钢琴乐音，应用了小波降噪技术抑制了录音时的环境噪音和电气噪音，最后通过播放降噪后的录音文件来实现钢琴乐音的仿真。在测试时，客户对第一个原型的评价结论为“1.声音不干净，有高频段的噪音。2.仿真效果呆板，无灵气，不能表现钢琴的真实魅力”。而录音技术无进一步完善的潜力，所以第一个原型很快就被抛弃。如果采用生命周期法，由于最终客户在“制定计划、需求分析、软件设计、程序编写”前四个阶段听不到仿真效果，而对着几万字的开发文档，可能会觉得录音技术是一种优秀的方法。而软件工程师对音乐文化又不熟悉，也可能觉得录音技术是正确的方向。直到第五阶段，提交软件时测试时，软件公司才发现录音技术根本达不到客户的要求，此时造成的经济损失和工期延误十分巨大，所以本软件工程采用快速原型法是正确的选择。

本设计的第二个原型使用了 14 天时间，采用了频谱分析法，应用了傅立叶变换的技术实现了对钢琴的频谱结构进行解析，在第二个原型中仅仿真了钢琴频谱分析中的 7 倍频。客户对第二个原型评价为“对高音区仿真效果较好，低音区似乎仿真的不够饱满。”通过分析发现人类的耳朵可以感知到 20Hz 到 20kHz 的音域范围，因此 7 倍频不能充分表现低音区的频谱结构，所以计划提高基频的仿真倍数。

本设计的第三个原型使用了 7 天时间，全面分析了钢琴各乐音的频谱结构，实现 17 倍频、25 倍频、50 倍频的仿真三个级别可调整，其中 17 倍频是表达了钢琴频谱结构的最重要振动能量，计算速度极快。50 倍频完全表达了钢琴频谱结构的复杂性，即使表现低音区的泛音，也超过人耳的感知范围，仿真效果无可挑剔。而 25 倍频是一种折衷方案，对钢琴的常见音符的表现比较完美。客户对第三个原型评价为“高、中、低仿真效果较理想，但是整个发音过程，象是电子琴没有音量变化。客户的测试结论决定着下一个原型的研发方向，因此重新制定了新的软件开发计划。

本设计的第四个原型使用了 7 天时间，我们发现了乐器仿真时，除了要实现音色仿真，还需要实现包络曲线的仿真。发现了音符之间存在突变杂音，设计了较简单指数衰减包络技术和四段直线包络技术，最后选用四段直线包络技术来近似表达钢琴包络曲线，客户对第四个原型评价为“不象是击弦类乐器而象是吹奏类乐器”。因此又重新启动新的软件开发计划。

本设计的第五个原型使用了 14 天时间，收集了大量的击弦类乐器和吹奏类乐

器的录音文件，分析各种乐器乐音的振动能量在时间上衰减的规律。设计了一个复杂的曲线自定义包络函数，仿真效果通过了客户评价。

至此快速原型阶段结束，启动了成品软件开发阶段。

## 5.5 小结

本章选用了Matlab作为快速原型的开发工具，尝试对钢琴音色电子合成，发挥了Matlab的强大的数学分析能力。

本章首先采用了Matlab语言编写正弦波发音程序，演奏了《东方红》乐曲第一小节，发现音符之间存在突变杂音，使用指数衰减包络技术、四段直线包络技术减轻了音符切换的杂音，但是随后发现这种方法导致发音生硬、不自然。编写程序分析了钢琴单音，获得了振幅波形图，针对这个振幅波形图，设计了一个自定义钢琴包络曲线函数，模拟了钢琴琴弦被敲击发音的全过程。新函数包络技术保证了前段振动能量上升较快，随着时间的延长，振动能量逐渐下降，在当前音符发音时间结束时，振动能量恰好降为0，为演奏下个音符做好准备，有效地消除了音符切换时的相位差异产生的杂音。因此，仿真效果优于四段直线算法，单个音符自身变化柔和，音符间切换平顺，振动能量衰减类似于钢琴、吉他这些击弦、弹拨类乐器。

本章还利用了钢琴音色描述矩阵实现了钢琴音色的电子合成，采用了Matlab语言编写函数piano()，该函数直接内置了钢琴音色描述矩阵，计算过程简捷，仿真了钢琴音色中的25次谐波，能够快速产生一个完美的钢琴单音。

## 第六章 基于 C#.net 的钢琴演奏软件的开发

快速原型以完美仿真为第一目的，对发音速度和界面都不做严格要求。在对快速原型测试时发现，虽然MATLAB矩阵计算能力很强大，但要临时展开傅里叶级数计算，计算数十个倍频点的振动能量，最终合成效果良好的乐音，整个过程也会有明显的滞后感。为了能够流畅的完成钢琴乐曲的演奏，只能采用先把整个钢琴乐曲的音频WAV文件生成完毕后，再播放WAV文件。所以无法采用Matlab实现直接演奏的交互式程序，而且MATLAB制作可视化界面的能力低下，不可能作为最终产品的开发语言。所以在开发即时交互发音的钢琴演奏软件时，选用了C#.net作为开发工具。

### 6.1 钢琴音色合成方案整体设计

钢琴乐音合成的核心工作是傅里叶级数生成器能够发出逼真的电子合成音，但是傅里叶级数生成和多频率乐音混音过程需要完成复杂的矩阵计算，消耗大量的CPU资源。C#.net的矩阵计算能力还不如Matlab的效率，在常见的家用计算机的配置下，Matlab在仿真钢琴音色时都出现了滞后现象，所以C#.net也无法胜任临时电子合成钢琴音色的任务。因此设计了中间件层，能够有效地提高发音效率，钢琴音色合成方案见图6-1。

如果最终音箱输出的钢琴音色不理想，则回到快速原型，重新调试最优参数。通过中间件重新制作88个WAV文件，提交给成品软件发音。由于中间件的存在，随时可以优化WAV样本文件，而成品软件无须修改发音部分的源码，实现了界面制作和音色合成的模块分离。因此，在后续的音色改良过程中，成品软件无须代码升级，维护工作量极小。

本论文最关键的成果在于：

1. 快速原型中的傅里叶级数的数学分析结论；
2. 中间件中通过大量调试获得的最优发音参数。

在本论文三层设计架构中，发布最终成品软件时，无需附带快速原型与中间件。所以即便意图侵权者对最终软件逆向工程，也无法获得以上两个重要信息，对88个WAV文件分析也无法获得两个重要信息。快速原型、中间件、成品软件分离设计最大程度地保护了关键成果。

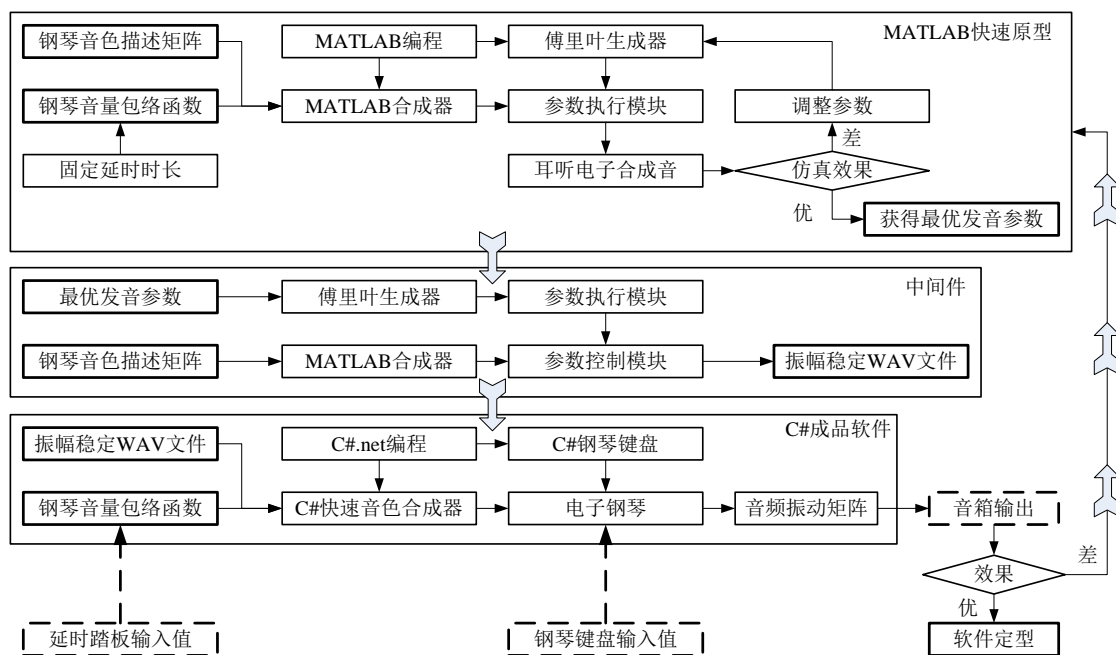


图 6-1 高效的钢琴音色合成方案

## 6.2 中间件音色合成方案的优点

本技术方法和对钢琴直接录音成 WAV 文件的技术方法相比较，虽然后期方法类似，都是播放 WAV 文件。但是中间件生成的 WAV 文件和自然录音获得的 WAV 文件相比较，具有较多优点：

### 1) 音色纯静

目前播放 WAV 文件的技术很成熟，比临时动态计算钢琴音色的频谱分布的效率高，演奏乐曲时更加流畅。电子合成的钢琴乐音无环境噪音和录音话筒的电气噪音，无需引入有损音色的降噪技术，因此电子合成钢琴乐音更加纯静。

### 2) 音强稳定

录音获得的钢琴单音，由钢琴弦振动方程所决定，振幅能量存在由强至弱的变化，无法二次加工。而电子合成的钢琴单音振幅稳定，没有强弱变化，可以在后期加工中加入包络函数以控制振幅变化，所以可以再现击打力度和延时踏板的差异变化，表现乐曲时更具有感情。

3) 声音文件体积小，电子合成的钢琴单音的频谱范围可以控制恰好落在人耳的敏感识别范围内，因此仅存储有效信号，节省了声音文件的存储体积。而录音获得的钢琴单音文件中带有环境噪音和人耳无法听见的超声波信号，因此声音文件

较大。

## 6.3 成品软件的开发

应用C#.net开发成品软件界面见图6-2。



图 6-2 成品软件界面

### 6.3.1 发音函数的开发

采用C#.net编写播放WAV文件的发音函数，代码如下：

```
private Microsoft.DirectX.DirectSound.Device wavdev
= new Microsoft.DirectX.DirectSound.Device();
private string fpath;
//C#.net自定义的发声函数
private void playwav(string filename)
{
    filepath = Application.StartupPath + "\\\\WAVFILE\\"+filename+".wav";
    //给定WAV声波文件的路径，便于播放声音
    try
    {
```



```

Microsoft.DirectX.DirectSound.SecondaryBuffer snd
    = new Microsoft.DirectX.DirectSound.SecondaryBuffer(filepath, wavdev);
    snd.Play(0, Microsoft.DirectX.DirectSound.BufferPlayFlags.Default);
    //通过c#.net调用directx类完成WAV声波文件的播放
    filepath = null;
}
catch (Exception ex)
{
    label101.Text = ex.ToString();
}
}

```

### 6.3.2 键盘响应函数的开发

采用C#.net编写的键盘响应函数代码如下，帮助操作者利用物理键盘实现钢琴的仿真演奏：

```

private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Q && q == 1)
    {
        //对程序下方的黑白钢琴键盘实现更换图片的动画
        this.pictureBox_k.Image = Image.FromFile(Application.StartupPath +
            "\\pic\\Keyw2.jpg");
        //以上程序实现了键盘上按键的图片更换
        this.pictureBoxq.Image = Properties.Resources.k;
        //让picturebox装载白色琴键新图片
        pictureBoxphoto.Visible = true;
        //让picturebox显示新图片
        this.pictureBoxphoto.Image = Properties.Resources.k;
        //设置状态位，敲击一次键盘，键盘动画只能响应一次
        q = 0;
        //自定义播放匹配钢琴声音
        playsound("15");
    }
    if (e.KeyCode == Keys.W && w == 1)

```

```
{
//对程序下方的黑白钢琴键盘实现更换图片的动画
this.pictureBox_1.Image = Image.FromFile(Application.StartupPath
    + "\\pic\\Keyw2.jpg");
//以上程序实现了键盘上按键的图片更换
this.pictureBoxw.Image = Properties.Resources.l;
//让picturebox装载白色琴键新图片
pictureBoxphoto.Visible = true;
//让picturebox显示新图片
this.pictureBoxphoto.Image = Properties.Resources.l;
//设置状态位，敲击一次键盘，键盘动画只能响应一次
w = 0;
//自定义播放匹配钢琴声音
playsound("16");
}
if (e.KeyCode == Keys.E && keye == 1)
{
this.pictureBox_e.Image = Image.FromFile(Application.StartupPath
    + "\\pic\\Keyb2.jpg");
this.pictureBoxe.Image = Properties.Resources.e;
//让picturebox装载黑色琴键新图片
pictureBoxphoto.Visible = true;
//让picturebox显示新图片
this.pictureBoxphoto.Image = Properties.Resources.e;
//设置状态位，敲击一次键盘，键盘动画只能响应一次
keye = 0;
//自定义播放匹配钢琴声音
playsound("17");
}
if (e.KeyCode == Keys.R && r == 1)
{
this.pictureBox_r.Image = Image.FromFile(Application.StartupPath +
    "\\pic\\Keyw2.jpg");
\\加载白色的钢琴琴键图像
```

```

this.pictureBoxr.Image = Properties.Resources._7;
pictureBoxphoto.Visible = true;
this.pictureBoxphoto.Image = Properties.Resources._7;
r = 0;
playsound("18");
}
... 后续键盘匹配程序略

```

### 6.3.3 虚拟键盘击打动画的开发

采用C#.net编写的键盘击打后弹性动画变化的函数代码如下，能够改善操作者交互使用的体验：

```

//本程序绘制琴键和字母键抬起响应
private void Form1_KeyUp(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.PageDown)
    {
        //下方的钢琴键盘图片更换
        this.pictureBox_pagd.Image = Image.FromFile(Application.StartupPath
            + "\\pic\\Keyw2.jpg");
        //键盘上的图片更换
        this.pictureBoxpagd.Image = Image.FromFile(Application.StartupPath
            + "\\pic\\3_+_+.jpg");
        //大照片的PictureBox不显示
        pictureBoxphoto.Visible = false;
        //更改标志位
        pagd = 1;
    }
    if (e.KeyCode == Keys.End)
    {
        this.pictureBox_end.Image = Image.FromFile(Application.StartupPath
            + "\\pic\\B_K_P_2.jpg");
        //加载新图片
        this.pictureBoxend.Image = Image.FromFile(Application.StartupPath
            + "\\pic\\2++.bmp");
        //加载新图片
    }
}

```

```

pictureBoxphoto.Visible = false;
//图片不可见
end = 1;
}
if (e.KeyCode == Keys.Delete)
{
    this.pictureBox_del.Image = Image.FromFile(Application.StartupPath
        + "\\pic\\W_K_22.jpg");
    //加载新图片
    this.pictureBoxdel.Image = Image.FromFile(Application.StartupPath
        + "\\picture\\1++.bmp");
    //加载新图片
    pictureBoxphoto.Visible = false;
    //图片不可见
    del = 1;
}
if (e.KeyCode == Keys.PageUp)
{
    this.pictureBox_pagu.Image
        = Image.FromFile(Application.StartupPath
            + "\\pict\\B_K_P_222.JPG");
    //加载新图片
    this.pictureBoxpagu.Image = Image.FromFile(Application.StartupPath
        + "\\pict\\6_+_+.jpg");
    //加载新图片
    this.pictureBoxphoto.SendToBack();
    pagu = 1;
}
..... 后续键盘键盘动画程序略

```

#### 6.3.4 键盘与琴键联动的开发

采用C#.net编写的物理键盘和琴键联动的弹性动画变化的函数代码如下，能够改善操作者交互使用的体验：

```
//设置钢琴各琴键状态标识位
```

```
private int q=1, w=1, keye=1, r=1, t=1, y=1, u=1, i=1, o=1,
    p=1,d1=1,d2=1,d3=1,d4=1,d5=1,d6=1,d7=1,d8=1,d9=1,d0=1,
    a=1,s=1,d=1,f=1,g=1,h=1,j=1,k=1,l=1,z=1,x=1,c=1,v=1,
    b=1,n=1,m=1,n0=1,n1=1,n2=1,n3=1,n4=1,n5=1,n6=1,n7=1,
    n8=1,n9=1,ins=1,hom=1,pagu=1,del=1,end=1,
    pagd=1,exe=1,dec=1,mul=1,div=1,add=1,sub=1;
//处理单击事件，单击键盘，黑白钢琴键盘响应
private void pictureBox_d1_Click(object sender, EventArgs e)
{
    playsound("1");
    //发出简谱“1”的钢琴乐音
}
private void pictureBox_d2_Click(object sender, EventArgs e)
{
    playsound("2");
    //发出简谱“2”的钢琴乐音
}
private void pictureBox_d3_Click(object sender, EventArgs e)
{
    playsound("3");
    //发出简谱“3”的钢琴乐音
}
private void pictureBox_d4_Click(object sender, EventArgs e)
{
    playsound("4");
    //发出简谱“4”的钢琴乐音
}
private void pictureBox_d5_Click(object sender, EventArgs e)
{
    playsound("5");
    //发出简谱“5”的钢琴乐音
}
..... 后续键盘键盘联动程序略
```

## 6.4 成品软件的测试工作

请音乐专业人士对成品软件的仿真效果进行评价，成品软件比较顺利地通过验收。这属于黑盒测试，客户已知产品的任务目标，按照预定的目标要求对产品各项功能进行测试，检测软件各项功能是否都达到了要求。

同时我们也用 MATLAB 语言开发频谱、波形分析程序，从数学原理上对软件仿真效果进行全面的白盒测试。白盒测试的前提条件是已知产品的工作原理和工作步骤，按步骤检测中间过程的每一个环节是否达到软件详细开发说明书所制定的验收标准。

经过小波降噪的钢琴录音文件的频谱结构见图 6-3；本系统电子合成乐音的频谱结构见图 6-4。测试表明，本系统电子合成乐音的频谱结构和钢琴真实录音文件的频谱结构十分相近，而且电子合成乐音干净没有任何噪声。

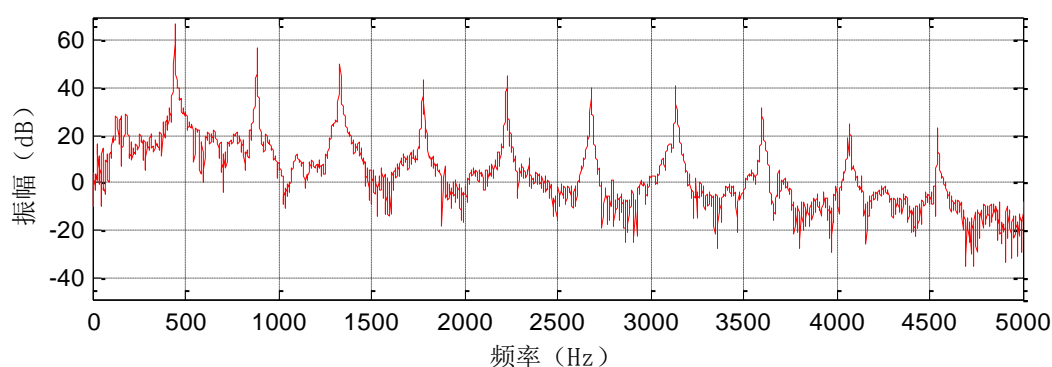


图 6-3 小波降噪后的钢琴录音文件的频谱结构

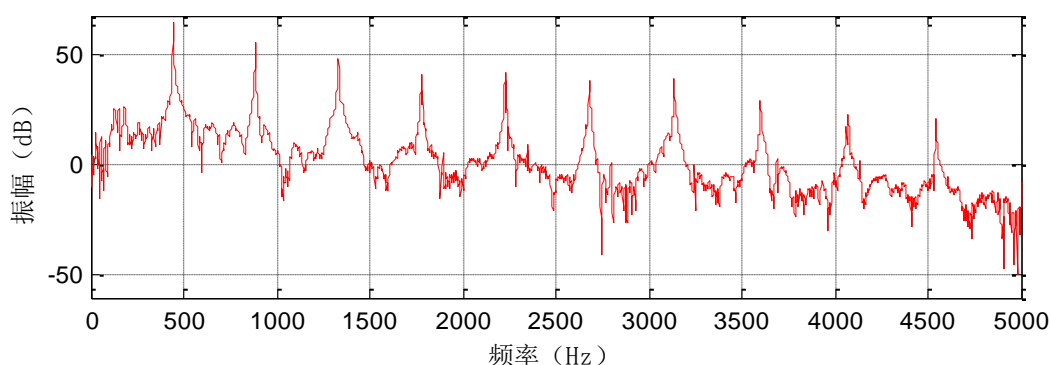


图 6-4 电子合成乐音的频谱结构

经过小波降噪的弹拨弦乐器吉它录音文件的频谱结构见图 6-5；经过小波降噪的拉弦乐器小提琴录音文件的频谱结构见图 6-6。测试表明电子合成的击弦乐器钢琴乐音频谱和其它弦乐器有明显的差异。

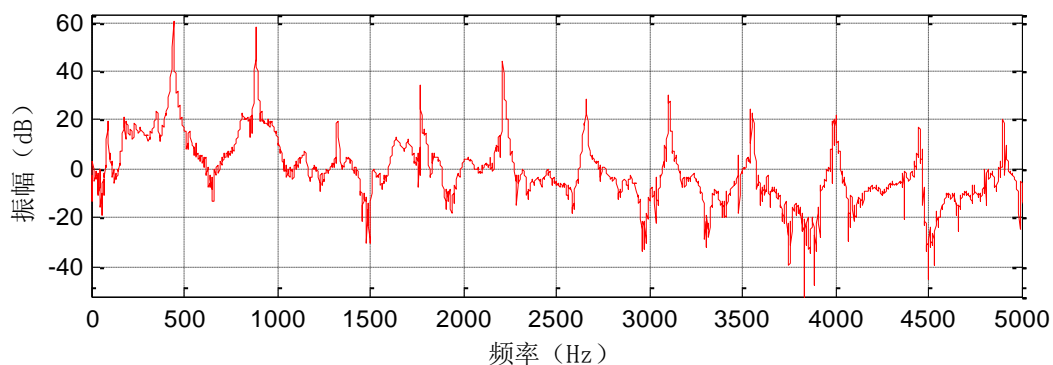


图 6-5 降噪后的吉它录音文件的频谱结构

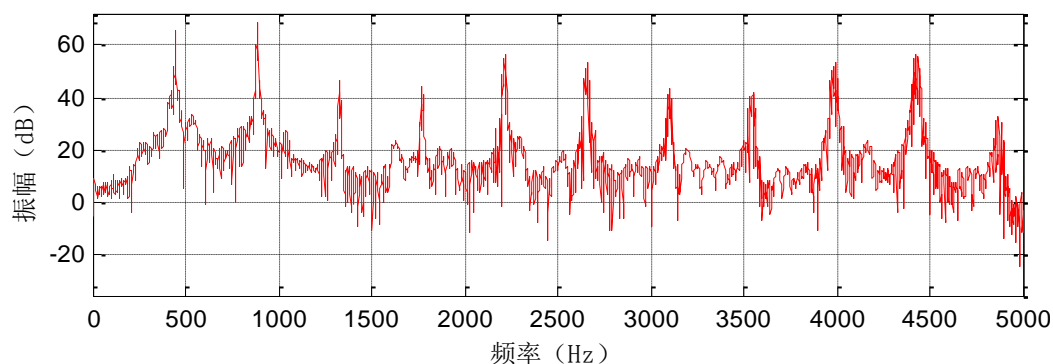


图 6-6 降噪后的小提琴录音文件的频谱结构

限于篇幅，其它白盒测试工作不做过多的描述。

## 6.5 小结

为了提高成品软件的钢琴音色仿真效率，设计了中间件层，由Matlab开发的快速原型生成完美的钢琴音色WAV文件后，提供给C#.net的成品软件播放。为由于采用中间层隔离了数学模型和成品软件，所以发布的成品软件中不含有本论文最重要的知识成果，所以不怕逆向工程反编译分析，因此能够有效的保护研究成果。

## 第七章 总结与展望

### 7.1 总结

本论文以实现钢琴音色电子合成为目的，把软件工程和音乐理论相结合，在剖析十二平均律、钢琴弦振动方程的数学基础上，对钢琴乐音特征电子识别和电子合成方法进行了详尽的讨论、研究和实验。主要成果为：

1. 本软件工程最大技术风险是不能事先确定钢琴音色的仿真方法，本论文对比软件工程中的生命周期法、快速原型法和面向对象法三种软件开发方法，认为只有快速原型法才能有效降低技术风险，缩短工期，提高软件质量。所以最终选择 MATLAB 语言来快速构建原型，用 C#.net 来开发成品软件。本论文在降低软件工程技术风险上的尝试，可供其它交叉领域软件工程参考。

2. 基于短时傅里叶变换的频谱分析，提取到钢琴音色特征矩阵，研究发现弦类乐器音色特征矩阵不同于管类、打击乐器；接着对钢琴音强变化和时间的关系进行分析，获得了音强变化包络曲线，发现钢琴作为一种击弦类乐器，音强变化包络曲线不同于弹拨弦、拉弦类乐器。利用音色特征矩阵和音强变化包络曲线可以区分钢琴和其它乐器的差异，从而实现钢琴乐音的识别。

3. 本论文开发的软件对钢琴音色特征矩阵进行矩阵计算，实现了钢琴音色中的 25 次倍频谐波的电子合成，仿真效果优于市面上采用“柯西函数五次倍频”原理的电子琴。同时引入了音强变化包络曲线，在能量衰减上和其它弦乐器表现不同，实现了钢琴乐音的仿真。

4. 本论文设计了三层架构的中间件发音技术，成品软件在发音时无需重新进行矩阵运算，降低了 CPU 工作压力，提高了发音效率，同时还具有音色纯静、音强稳定、声音文件体积小的优点。成品软件中不含数学模型和发音参数，即使成品软件被逆向工程，也不会泄漏研究成果。

### 7.2 展望

软件工程技术在音乐领域的应用，将给人类带来全新的音乐享受，符合国家近期提出文化强国理念，在人文、科技领域都具有研究意义。所以应当加强科研力度，增加科研投资，以进一步推动音乐领域的软件研发。

由于时间、经费的限制，本次研究存在一些不足之处，可供同类软件工程借鉴，本项目也将进一步完善，对后续研究展望如下：



1. 目前市面上低价位电子琴大多采用“柯西函数”的方法,虽然这些电子琴只计算了 5 次倍频,仿真效果一般,但是计算量小,无需额外的音色库物理存储空间,在商业应用中性价比高。相对而言,本软件的音色仿真效果好于这些电子琴,但是也存在两个缺点。快速原型部分矩阵计算量较大,对单片机(MCU)的运算能力要求高,不适在单片机结构的低价位电子琴上直接应用。而成品软件部分虽然无需矩阵计算,但是需要预存 88 个 WAV 格式的波表文件,以支撑发音功能,约需要 880kb 的永久存储空间,而低价位的电子琴单片机构造中无法提供这样的永久存储空间。因此在后续研究中,拟对 WAV 格式的文件采用压缩存储的方式,目前有 ZIP 无损压缩技术,或 MP3 有损压缩技术可选择,估计仅需要 440 kb 的存储空间,能降低电子琴的硬件成本。

2. 本论文的成品软件采用 C#.net 开发,具有一定的跨平台工作能力,可以工作于 PC 机、WinCE 嵌入系统的智能终端以及缓存空间较大的电子琴产品,但是在市场调研时也发现,目前基于 android 系统的智能终端在市场上占有较大的比重,android 嵌入系统采用的主流开发语言为 JAVA,本论文的成品软件无法直接在 android 嵌入系统上工作。在后续研究中,拟将成品软件移植到 JAVA 平台上。由于本软件具有三层架构的优势,由中间件隔离了数学模型和成品软件。因此只要重新开发 JAVA 语言的成品软件,无须改写快速原型、中间件,移植工作量很小。

## 致 谢

首先必须向我的论文指导老师程红蓉副教授送上我最最真挚的感谢，从本论文的选题、开题报告到现在的论文结构、观点甚至于个别由于我粗心导致的错别字和格式问题都细心一一指出并给予指导意见。程老师一丝不苟的高尚品德深深的影响了我，再次献上我真诚的感谢！

其次要感谢这两年来学校的培养，以及培养点的关心和指导，给予了我取得发展更大进步空间的机会。

最后要感谢的是我的同学和朋友，在这篇论文中给予了我很多电脑的技术和音乐理论方面的支持和帮助，让我少走了很多的弯路。

## 参考文献

- [1] 彭志敏.新音乐分析教程[M].湖南文艺出版社,2004:2-40.
- [2] 黄允箴,王璨,郭树芸,等.中国传统音乐导学[M].上海:上海音乐学院出版社,2006:1-40.
- [3] 郑建鸥.关于节奏学的研究[J].西南师范大学学报(人文社会科学版),2000,26(2):114-115.
- [4] D. Byrd, T. Crawford. Problems of Music Information Retrieval in the Real World[J].Information Processing and Management,2002,38(2):249-272.
- [5] Z. N. Li, T. D. Zeng, T. Fan, et al. Source Separation Method of Machine Faults Based on Post-nonlinear Blind Source Separation[J]. Proceedings of the World Congress on Intelligent Control and Automation,2008:1734-1737
- [6] D. J. E Nunn., A. Purvis, P. D. Manning. Source Separation and Transcription of Polyphonic Music[C]. Proceedings of the International Colloquium on New Music Research[M],Gent, Belgium,1994:10-15.
- [7] Kashino, Tanaka. A Sound Source Separation System with the Ability of Automatic Tone Modeling[C]. Proceeding of the International Computer Music Conference,1993:248-255.
- [8] Kashino, Nakadai, Kinoshita, et al. Application of Bayesian Probability Network to Music Scene Analysis[C].Proceedings of the International Computer Music Conference,1993:21-26.
- [9] Martin. A Blackboard System for Automatic Transcription of Simple Polyphonic Music[C].MIT Media Laboratory Perceptual Computing Section Technical Report,1996:369-399.
- [10] A. B. Aicha, S. B. Jebara. Comparison of Three Methods of Eliminating Musical Tones in Speech Denoising Subtractive Techniques[C]. Proceedings of the IEEE International Conference on Electronics, Circuits and Systems,2006:652-655.
- [11] 杨行峻,迟惠生.语音信号数字处理[M].北京:电子工业出版社,1995:44-53.
- [12] 顾亦然,秦军,王锁萍,等.几种音乐识别算法的比较[J].南京邮电大学学报,1998,18(2):36-40.
- [13] J. C. Brown, M. S. Puckette. An Efficient Algorithm for the Calculation of Aconstant Q Transform[J]. J A coust Soc Am,1992,92(5):2698-2701.
- [14] Z. Z. Ding, L. R. Dai. A Study of Constant Q Transform in Music Signal Analysis[J].Technical Acoustics ,2005 , 24(4) :259-263.
- [15] 李功宇. 弦乐和弦声信号时频特征的小波包分析[J]. 昆明理工大学学报(理工版),2003,28(4):36-39.
- [16] A. L. Chen, M. Chang, J. Chen, et al. Query by Musica Segments:an Efficent Approach for Song Retrieval[C]. In:Proceedings of IEEE International Conference on Multimedia and

- Expo,2000,2:873-876.
- [17] 郭任亮.MATLAB 7.x 数字信号处理[M].北京:人民邮电出版社,2006:41-53.
- [18] 王琦.MATLAB 基础与应用实例集粹.北京:人民邮电出版社,2007:32-55.
- [19] Bozena Kostek. Musical instrument classification and duet analysis employing music information retrieval techniques[J]. Proceedings of the IEEE, 2004, 92( 4) : 712-729.
- [20] Herrera P, Peeters G, Dubnov S. Automatic classification of musical instrument sounds[J]. Journal of new music research, 2003, 32( 1) : 3-21.
- [21] Arie A. Livshin, Xavier Rodet. Musical instrument identification in continuous recordings[J]. Proceedings of the 7th international conference on digital audio effects ( DAFX 04) , 2004: 5-8.
- [22] Wold E, Blum T, Keislar D, et al. Content — based classification, search, and retrieval of audio[J]. IEEE Multimedia, 1996, 3( 3) : 27 -36.
- [23] Eronen A. Comparison of features for musical instrument recognition[J]. IEEE workshop on the application of signal processing to audio and acoustics, 2001: 19-22.
- [24] Krishna A G, Sreenivas T V. Music instrument recognition: from isolated notes to solo phrases[J]. Proceedings of the IEEE international conference on acoustics, speech, and signal processing, 2004, 4: 265-268.
- [25] Christopher J C, Burges. A tutorial on support vector machines for pattern recognition[J]. Journal of data mining and knowledge discovery, 1998, 2( 2) : 121-167.
- [26] Keith Dana Martin. Sound - source recognition: A theory and computational mode[J]. PhD Thesis, Massachusetts institute of technology, 1999.
- [27] Eronen A, Klapuri A. Musical instrument recognition using cepstral coefficients and temporal features[J]. Proc of the IEEE international conference on acoustics, speech and signal processing, 2000, 2: 753 - 756.
- [28] Agostini G, Longari M, Pollastri E. Music instrument timbres classification with spectral features[J]. IEEE 4th workshop on multimedia signal processing, 2001: 97 - 102.
- [29] Antti Eronen. Automatic musical instrument recognition[J]. Master' s thesis, Tampere university of technology, 2001.
- [30] Fragoulis D K, Avaritsiotis J N, Papaodysseus C N. Timbre recognition of single notes using an ARTMAP neural network. Proceedings of the 6th IEEE[J] international conference on electronics, circuits and Systems, 1999, 2: 1009 - 1012.
- [31] Bertrand David, Gael Richard. Efficient musical instrument recognition on solo performance music using basic features[J]. Proceedings of the 25th international conference: Metadata for

- audio, 2004: 2 - 5.
- [32] Martin K D, kim E Youngmoo. Musical instrument identification: a pattern recognition approach[J]. Journal of the acoustical society of America, 1998, 104( 3) : 1768 — 1776.
- [33] Marques J, Moreno P J. A study of musical instrument classification using gaussian mixture models and support vector machines[J]. Technical Report, Cambridge research labs, 1999.
- [34] 吴飞, 庄越挺, 潘云鹤. 基于增量学习支持向量机的音频例子识别与检索[J]. 计算机研究与发展, 2003, 40( 7) : 950 - 955.
- [35] P.Nangtin, P.Kumhom, K.Chomnongthai. Video-based obstacle tracking for automatic train navigation [C]. 2005 International Symposium on Intelligent Signal Processing and Communication Systems, Hong Kong, 2005, 21-24.
- [36] 江新姿,高尚. 基于 K 均值与蚁群混合聚类的图像分割[J]. 计算机与数字工程,2011,(06):39-41.
- [37] 林宇洪. 木材供应链追溯 RFIC 卡的设计[J]. 西北林学院学报, 2013, 28(5): 175-179.
- [38] 李翠,冯冬青.基于改进 K\_均值聚类的图像分割算法研究[J].郑州大学学报(理学版),2011,43(01):1671-6841.
- [39] 王易循. 基于 K 均值聚类分割彩色图像算法的改进[J]. 计算机应用与软件,2010,27(08) .
- [40] 李春杰. 基于 LVQ 神经网络的乐器识别方法[J]. 科技信息, 2013, (2): 153.
- [41] 李礼. 新疆维吾尔族民族乐器音色库的采样与构建[D]. 新疆师范大学硕士研究生论文, 2009.
- [42] 席春梅. 新型键盘电子乐器的研究与设计实现[D]. 青海师范大学 , 2010.
- [41] 高雪松. 基于 MIDI 的乐器控制系统和音符的自动识别方法研究[D]. 南京理工大学, 2012.
- [42] 沈骏, 胡荷芬. 中国民族乐器的特征值提取和分类[J]. 计算机与数字工程, 2012, 40(9):119-121.
- [43] 芮瑞, 鲍长春. 基于非线性动力学的乐器分类方法[J]. 电子学报, 2012, 40(7):1481-1488.
- [44] J Garcia A Barbedo , G Tzanetakis. Musical instrument classification using individual partials [ J ] . IEEE Transactions on Audio,Speech, and Language Processing, 2011, 19( 1) : 111- 122.
- [45] M Hall, E Frank, et al. The Weka data mining software: Anupdate [ J ] . ACM SIGKDD Explorations Newsletter, 2009, 11(1) : 10- 18.
- [46] 李时中.谈钢琴弦列最佳弦点的理论与实践.乐器,1992,35(1):14-22.
- [47] T. Liu, S. Q. Sun, Y. H. Pan, et al. Emotional Recognition for Chime Bell Music[C].Conference Proceedings-IEEE International Conference on Systems, Man and Cybernetics,2004,1:568-573.
- [48] Mohammad S Sharawi, Dennis M. Akos, Daniel N Aloï. GPS C/N0 estimation in the presence of interference and limited quantization levels [J]. IEEE transactions on aerospace and

electronic systems(S0018-9251), 2007, 43(1): 227-238.

- [49] 林宇洪, 沈嵘枫, 邱荣祖. 南方林区林产品运输监管系统的研发[J]. 北京林业大学学报, 2011, 33(5): 130-135.
- [50] 沙丹, 许建修. 基于 MATLAB 及 VB 的城市交通仿真研究[J]. 上海师范大学学报: 自然科学版, 2010, 39(1): 36-36.
- [51] 周爱莲, 李旭宏, 毛海军. 基于 MATLAB 的静止图像车牌定位研究 [J]. 交通运输工程学报, 2010,(1): 61-61.