

高等院校通信与信息专业规划教材

语音信号处理实验教程

梁瑞宇 赵力 魏昕 编著



机械工业出版社

本书为《语音信号处理（第3版）》的配套实验教程。该教材阐述了语音信号处理的基本理论，并基于 MATLAB 软件介绍了语音信号处理的实现方法和关键技术。本书共分13章，内容包括：MATLAB 基础教程、语音信号处理基础实验、语音信号分析实验、语音信号特征提取实验、语音增强实验、语音编码实验、语音合成与转换实验、语音隐藏实验、声源定位实验、语音识别实验、说话人识别实验、语音情感识别实验、实用语音信号处理平台。

本书可作为高等院校的教材用书或教学参考用书，同时也可供从事语音信号处理等领域的工程技术人员参考。

本书提供 MATLAB 仿真程序，读者可登录机械工业出版社教育服务网（www.cmpedu.com）注册并审核通过后下载。

图书在版编目（CIP）数据

语音信号处理实验教程/梁瑞宇，赵力，魏昕编著. —北京：机械工业出版社，2016.2

高等院校通信与信息专业规划教材

ISBN 978-7-111-53071-8

I. ①语… II. ①梁… ②赵… ③魏… III. ①语音信号处理-高等学校-教材 IV. ①TN912.3

中国版本图书馆 CIP 数据核字（2016）第 037809 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：李馨馨

责任校对：张艳霞

责任印制：

印刷厂印刷（×××装订）

2016 年 3 月第 1 版·第 1 次印刷

184mm×260mm·19 印张·466 千字

0001— 册

标准书号：ISBN 978-7-111-53071-8

定价： 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

网络服务

服务咨询热线：(010)88379833

机工官网：www.cmpbook.com

读者购书热线：(010)88379649

机工官博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

封面防伪标均为盗版

金书网：www.golden-book.com

前 言

语音信号处理是研究用数字信号处理技术对语音信号进行处理的一门学科，它是在多门学科基础上发展起来的综合性技术，涉及语音学、语言学、生理学及认知科学、数字信号处理、模式识别和人工智能等许多学科领域。同时语音信号处理也是目前发展最为迅速的信息科学技术之一，其研究涉及一系列前沿课题。

根据教育部加强学生实践能力培养的要求，本书立足于语音信号基本理论，辅以 MATLAB 实现，使读者在学习理论知识的同时，能快速实践，提高学习兴趣，增强解决问题的能力。

本书是机械工业出版社出版的《语音信号处理（第3版）》的配套实验教材（也可独立作为实验教材使用）。作者基于多年从事语音信号处理研究的成果，在阐述理论知识的同时，配以 MATLAB 程序，并加上详细的注释。教材的编写由浅及深，验证性实验与设计性实验并重，同时适合初学者和有一定基础的研究者使用。

本书共分 13 章，第 1 章是 MATLAB 基础教程；第 2 章是语音信号处理基础实验；第 3 章是语音信号分析实验；第 4 章是语音信号特征提取实验；第 5 章是语音增强实验；第 6 章是语音编码实验；第 7 章是语音合成与转换实验；第 8 章是语音隐藏实验；第 9 章是声源定位实验；第 10 章是语音识别实验；第 11 章是说话人识别实验；第 12 章是语音情感识别实验；第 13 章介绍了本研究团队开发的实用语音信号处理平台。

本书主要面向信号与信息处理、电路与系统、通信与电子工程、模式识别与人工智能、计算机信息处理等学科有关专业的高年级学生和研究生，也可以作为从事语音信号处理这一领域科研工作的技术人员参考书。

本书的参考学时为本科生 32 学时、研究生 40 学时，可以根据不同的教学要求对其内容进行适当取舍，灵活安排讲课学时数。

本书主要由梁瑞宇、赵力、魏昕编写，其中，1~9 章由梁瑞宇编写，12~13 章由赵力编写，10~11 章由魏昕编写，全书由梁瑞宇统稿。广州大学的邹采荣教授主审了全书，并提出很多宝贵意见，在此表示诚挚的感谢。

语音信号处理是一门理论性强、实用面广、内容新、难度大的交叉学科，同时这门学科又处于快速发展之中，尽管作者在编写过程中始终注重理论紧密联系实际，力求以尽可能简明、通俗的语言，深入浅出、通俗易懂地将这门学科介绍给读者，但因作者水平有限，缺点错误在所难免，敬请广大读者批评指正。

编 者

目 录

前言

第 1 章 MATLAB 基础教程	1
1.1 MATLAB 简介	1
1.1.1 功能和特点	1
1.1.2 通用操作界面窗口	1
1.1.3 文件格式	3
1.2 MATLAB 数值计算	3
1.2.1 数据类型	3
1.2.2 变量	4
1.2.3 矩阵和数组	4
1.2.4 字符串	6
1.2.5 数学函数	6
1.2.6 关系逻辑	7
1.3 MATLAB 绘图功能	7
1.3.1 二维曲线的绘制	7
1.3.2 MATLAB 的三维图形绘制	12
1.3.3 立体图形与图轴的控制	13
1.4 MATLAB 的特殊图形绘制	13
1.4.1 条形图	13
1.4.2 直方图	13
1.4.3 饼图	14
1.4.4 对数坐标和极坐标图	14
1.4.5 对话框	14
1.4.6 句柄图形	15
1.5 MATLAB 程序设计	15
1.5.1 M 文件	15
1.5.2 程序流程控制	16
1.5.3 函数调用和参数传递	18
1.6 MATLAB 设计实例——FFT 频谱分析	19
1.6.1 FFT 基础	19
1.6.2 基于 FFT 的信号频谱分析	19
第 2 章 语音信号处理基础实验	24
2.1 语音采集与读写实验	24

2.1.1	实验目的	24
2.1.2	实验原理	24
2.1.3	实验步骤及要求	28
2.1.4	思考题	29
2.1.5	参考例程	29
2.2	语音编辑实验	30
2.2.1	实验目的	30
2.2.2	实验原理	30
2.2.3	实验步骤	32
2.2.4	思考题	33
2.2.5	参考例程	34
2.3	声强与响度实验	34
2.3.1	实验目的	34
2.3.2	实验原理	35
2.3.3	实验步骤及要求	38
2.3.4	思考题	38
2.3.5	参考例程	38
2.4	语音信号生成的数学模型	39
2.4.1	实验目的	39
2.4.2	实验原理	39
2.4.3	实验步骤及要求	43
2.4.4	思考题	43
2.4.5	参考例程	43
2.5	语音信号的预处理	45
2.5.1	实验目的	45
2.5.2	实验原理	45
2.5.3	实验步骤及要求	49
2.5.4	思考题	50
2.5.5	参考例程	50
第3章	语音信号分析实验	51
3.1	语音分帧与加窗	51
3.1.1	实验目的	51
3.1.2	实验原理	51
3.1.3	实验步骤及要求	53
3.1.4	思考题	54
3.1.5	参考例程	54
3.2	短时域分析	54
3.2.1	实验目的	54
3.2.2	实验原理	54

3.2.3	实验步骤及要求	56
3.2.4	思考题	58
3.2.5	参考例程	58
3.3	短时频域分析	60
3.3.1	实验目的	60
3.3.2	实验原理	60
3.3.3	实验步骤及要求	62
3.3.4	思考题	62
3.3.5	参考例程	62
3.4	倒谱分析与 MFCC 系数	62
3.4.1	实验目的	62
3.4.2	实验原理	63
3.4.3	实验步骤及要求	67
3.4.4	思考题	69
3.4.5	参考例程	69
3.5	线性预测分析	70
3.5.1	实验目的	70
3.5.2	实验原理	70
3.5.3	实验步骤及要求	75
3.5.4	思考题	77
3.5.5	参考例程	77
3.6	线谱对转换实验	78
3.6.1	实验目的	78
3.6.2	实验原理	78
3.6.3	实验步骤及要求	82
3.6.4	思考题	83
3.6.5	参考例程	83
第 4 章	语音信号特征提取实验	84
4.1	语音端点检测实验	84
4.1.1	实验目的	84
4.1.2	实验原理	84
4.1.3	实验步骤及要求	90
4.1.4	思考题	94
4.1.5	参考例程	94
4.2	基音周期检测实验	95
4.2.1	实验目的	95
4.2.2	实验原理	96
4.2.3	实验步骤及要求	100
4.2.4	思考题	102

4.2.5 参考例程	102
4.3 共振峰估计实验	103
4.3.1 实验目的	103
4.3.2 实验原理	103
4.3.3 实验步骤及要求	107
4.3.4 思考题	109
4.3.5 参考例程	110
第5章 语音增强实验	111
5.1 基于自适应滤波器法的语音降噪实验	111
5.1.1 实验目的	111
5.1.2 实验原理	111
5.1.3 实验步骤及要求	115
5.1.4 思考题	116
5.1.5 参考例程	116
5.2 基于谱减法的语音降噪实验	116
5.2.1 实验目的	116
5.2.2 实验原理	117
5.2.3 实验步骤及要求	118
5.2.4 思考题	120
5.2.5 参考例程	120
5.3 基于维纳滤波的语音降噪实验	121
5.3.1 实验目的	121
5.3.2 实验原理	121
5.3.3 实验步骤及要求	123
5.3.4 思考题	125
5.3.5 参考例程	125
5.4 基于小波分解的语音降噪实验	126
5.4.1 实验目的	126
5.4.2 实验原理	126
5.4.3 实验步骤及要求	129
5.4.4 思考题	130
5.4.5 参考例程	130
第6章 语音编码实验	132
6.1 PCM 编解码实验	132
6.1.1 实验目的	132
6.1.2 实验原理	132
6.1.3 实验步骤及要求	134
6.1.4 思考题	135
6.1.5 参考例程	135

6.2	LPC 编解码实验	136
6.2.1	实验目的	136
6.2.2	实验原理	136
6.2.3	实验步骤及要求	138
6.2.4	思考题	139
6.2.5	参考例程	139
6.3	ADPCM 编解码实验	141
6.3.1	实验目的	141
6.3.2	实验原理	141
6.3.3	实验步骤及要求	143
6.3.4	思考题	144
6.3.5	参考例程	144
第7章	语音合成与转换实验	146
7.1	帧合并实验	146
7.1.1	实验目的	146
7.1.2	实验原理	146
7.1.3	实验步骤及要求	152
7.1.4	思考题	153
7.1.5	参考例程	153
7.2	基于线性预测的语音合成实验	154
7.2.1	实验目的	154
7.2.2	实验原理	154
7.2.3	实验步骤及要求	155
7.2.4	思考题	156
7.2.5	参考例程	156
7.3	基于共振峰检测和基音参数的语音合成实验	157
7.3.1	实验目的	157
7.3.2	实验原理	157
7.3.3	实验步骤及要求	159
7.3.4	思考题	160
7.3.5	参考例程	160
7.4	语音信号的变调与变速实验	162
7.4.1	实验目的	162
7.4.2	实验原理	162
7.4.3	实验步骤及要求	166
7.4.4	思考题	167
7.4.5	参考例程	167
第8章	语音隐藏实验	170
8.1	LSB 语音信息隐藏实验	170

8.1.1	实验目的	170
8.1.2	实验原理	170
8.1.3	实验步骤及要求	172
8.1.4	思考题	173
8.1.5	参考例程	173
8.2	回声法语音信息隐藏实验	174
8.2.1	实验目的	174
8.2.2	实验原理	174
8.2.3	实验步骤及要求	176
8.2.4	思考题	177
8.2.5	参考例程	178
第9章	声源定位实验	180
9.1	简单房间回响模型	180
9.1.1	实验目的	180
9.1.2	实验原理	180
9.1.3	实验步骤及要求	182
9.1.4	思考题	183
9.1.5	参考例程	183
9.2	基于广义互相关的声源定位实验	185
9.2.1	实验目的	185
9.2.2	实验原理	185
9.2.3	实验步骤及要求	188
9.2.4	思考题	189
9.2.5	参考例程	189
9.3	基于空间谱估计的声源定位实验	191
9.3.1	实验目的	191
9.3.2	实验原理	191
9.3.3	实验步骤及要求	199
9.3.4	思考题	199
9.3.5	参考例程	200
第10章	语音识别实验	202
10.1	基于动态时间规整 (DTW) 的孤立字语音识别实验	202
10.1.1	实验目的	202
10.1.2	实验原理	202
10.1.3	实验步骤及要求	206
10.1.4	思考题	207
10.1.5	参考例程	207
10.2	基于隐马尔可夫模型 (HMM) 的孤立字语音识别实验	208
10.2.1	实验目的	208

10.2.2	实验原理	208
10.2.3	实验步骤及要求	213
10.2.4	思考题	217
10.2.5	参考例程	218
第 11 章	说话人识别实验	219
11.1	基于矢量量化 (VQ) 的说话人识别实验	219
11.1.1	实验目的	219
11.1.2	实验原理	219
11.1.3	实验步骤及要求	222
11.1.4	思考题	224
11.1.5	参考例程	224
11.2	基于高斯混合模型 (GMM) 的说话人识别实验	225
11.2.1	实验目的	225
11.2.2	实验原理	225
11.2.3	实验步骤和实验结果	227
11.2.4	思考题	229
11.2.5	参考例程	230
第 12 章	语音情感识别实验	231
12.1	基于 K 近邻分类算法的语音情感识别实验	231
12.1.1	实验目的	231
12.1.2	实验原理	231
12.1.3	实验步骤及要求	235
12.1.4	思考题	236
12.1.5	参考例程	236
12.2	基于神经网络的语音情感识别	240
12.2.1	实验目的	240
12.2.2	实验原理	240
12.2.3	实验步骤及要求	247
12.2.4	思考题	249
12.2.5	参考例程	249
12.3	基于支持向量机的语音情感识别	250
12.3.1	实验目的	250
12.3.2	实验原理	250
12.3.3	实验步骤及要求	254
12.3.4	思考题	255
12.3.5	参考例程	255
12.4	基于特征降维的语音情感识别	256
12.4.1	实验目的	256
12.4.2	实验原理	256

12.4.3 实验步骤及要求	262
12.4.4 思考题	263
12.4.5 参考例程	263
第 13 章 实用语音信号处理平台	266
13.1 基于 MFC 的语音信号处理软件平台	266
13.1.1 基于 MFC 的语音信号处理软件平台	266
13.1.2 语音开发软件平台	269
13.2 基于嵌入式 Linux 的音频驱动程序移植	272
13.2.1 高级 Linux 声音架构	272
13.2.2 Platform 功能和数据解析	273
13.2.3 Codec 功能和数据解析	274
13.2.4 WM8960 驱动移植	275
13.3 实时语音信号处理硬件平台	281
13.3.1 平台架构与资源	281
13.3.2 基于 QT 的语音信号处理	282
参考文献	290

第 1 章 MATLAB 基础教程

1.1 MATLAB 简介

1.1.1 功能和特点

MATLAB 是 MathWorks 公司开发的科学与工程计算软件，广泛应用于自动控制、数学运算、信号分析、计算机技术、图像信号处理、财务分析、航天工业、汽车工业、生物医学工程、语音处理和雷达工程等行业，是国内外高校和研究部门科学研究的重要工具。

MATLAB 的特点包括：

1) 功能强大。MATLAB 的数值运算主要针对矩阵设计和优化，包括加、减、乘、除、函数运算等多种数学运算；自带符号工具箱，可以解决在数学、应用科学和工程计算领域中常常遇到的符号计算问题；大量针对各专业应用的工具箱的提供，使 MATLAB 适用于不同领域；MATLAB 的 Notebook 为用户提供了强大的文字处理功能，允许用户从 Word 访问 MATLAB 的数值计算和可视化结果。

2) 人机界面友好，编程效率高。MATLAB 的语言规则与笔算式相似，命令表达方式与标准的数学表达式非常相近。MATLAB 的程序设计基于解释方式实现，算式无需编译即可得出结果，并能针对错误即时反馈，便于编程者改正。

3) 强大而智能化的作图功能。MATLAB 包含多种坐标系，可以绘制三维坐标中的曲线和曲面，使工程计算的结果可视化，使原始数据的关系清晰化。

4) 可扩展性强。MATLAB 软件包括基本部分和工具箱两大部分，工具箱可以任意增减，具有良好的可扩展性。

5) Simulink 动态仿真功能。MATLAB 的 Simulink 提供了动态仿真的功能，用户通过绘制框图来模拟一个线性、非线性、连续或离散的系统，并由 Simulink 实现仿真和系统分析。

1.1.2 通用操作界面窗口

MATLAB R2010a 版的 MATLAB 工作界面如图 1-1 所示。界面的上层铺放着几个最常用的部分：指令窗（Command Window）、当前目录（Current Directory）浏览器、工作内存（Workspace）浏览器、历史指令（Command History）窗等。

1. 指令窗

指令窗是进行各种 MATLAB 操作的最主要窗口。在窗口内，使用者可键入各种 MATLAB 指令、函数、表达式。窗口可显示除图形外的所有运算结果，并在运行错误时，给出相关的出错提示。指令窗常用命令如表 1-1 所示。

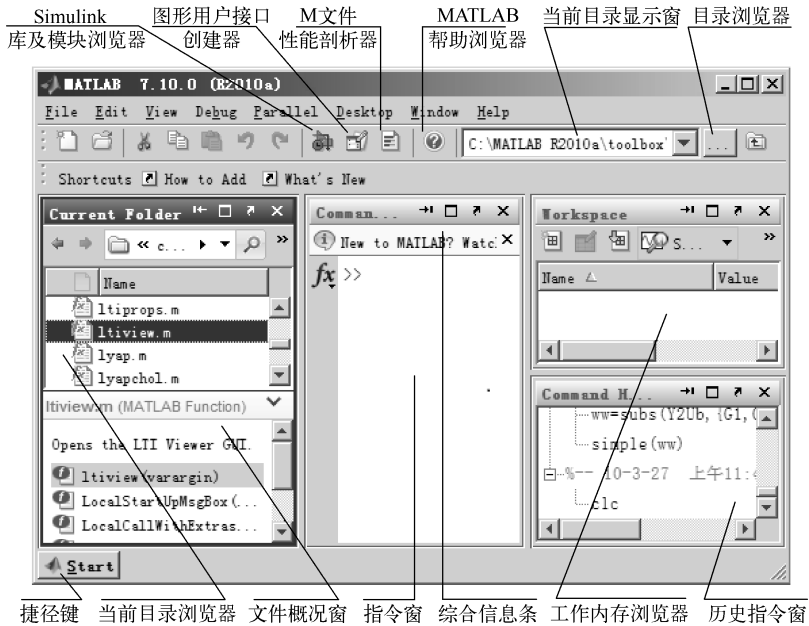


图 1-1 MATLAB 工作界面概貌

表 1-1 指令窗常用命令

命令名	作用
↑/↓	向前/向后调回已输入的命令
close all	关闭所有的图形窗口
clc	清除命令窗口中的内容
clear	从工作空间清除所有的变量
save	保存工作区间里的变量到磁盘文件
load	导入磁盘里的变量文件到工作空间
format type	输出数据格式显示控制命令
addpath	将一个新目录名添加到 MATLAB 的搜索路径里
rmpath	将某个目录从 MATLAB 搜索路径中清除
help <函数名>	在命令窗口显示 MATLAB 函数的帮助
whos	列出当前工作区间的所有变量，并显示变量的大小、类型及其所占用的存储空间

2. 当前目录浏览器

该浏览器用来展示 MATLAB 的子目录、M 文件、MAT 文件和 MDL 文件等。界面上的 M 文件，可直接进行复制、编辑和运行；界面上的 MAT 数据文件，可直接送入 MATLAB 工作内存；该界面上的子目录，可进行 Windows 平台的各种标准操作。

此外，在当前目录浏览器正下方，还有一个“文件概况窗”，用于显示所选文件的概况信息，如 M 文件所包含的内嵌函数和其他子函数。

3. 工作内存浏览器

窗口里罗列了 MATLAB 工作空间中所有变量的变量名、大小、字节数，可对变量进行观察、图示、编辑、提取和保存。

4. 历史指令窗

该窗记录已经运作过的指令、函数、表达式，及它们运行的日期、时间。窗口中的所有指令、文字都允许复制、重运行及用于产生 M 文件。

5. 捷径键

捷径键用于引出通往 MATLAB 所包含的各种组件、模块库、图形用户界面、帮助分类目录、演示算例等的捷径，以及向用户提供自建快捷操作的环境。

1.1.3 文件格式

1. 程序文件

程序文件即 M 文件，其文件的扩展名为 .m，包括主程序和函数文件。MATLAB 的各工具箱中的函数大部分是 M 文件，M 文件通过 M 文件编辑/调试器生成。

2. 数据文件

数据文件即 MAT 文件，其文件的扩展名为 .mat，用来保存工作空间的数据变量，数据文件可以通过在命令窗口中输入“save”命令生成。

3. 可执行文件

可执行文件即 MEX 文件，其文件的扩展名为 .mex，由 MATLAB 的编译器对 M 文件进行编译后产生，其运行速度比直接执行 M 文件快得多。

4. 图形文件

图形文件的扩展名为 .fig，可以在“File”菜单中创建和打开，也可由 MATLAB 的绘图命令和图形用户界面窗口产生。

5. 模型文件

模型文件扩展名为 .mdl，是由 Simulink 工具箱建模生成的。

1.2 MATLAB 数值计算

1.2.1 数据类型

MATLAB 中共有 15 种基本数据类型和 2 种自定义类型。MATLAB 数据类型关系如图 1-2 所示。

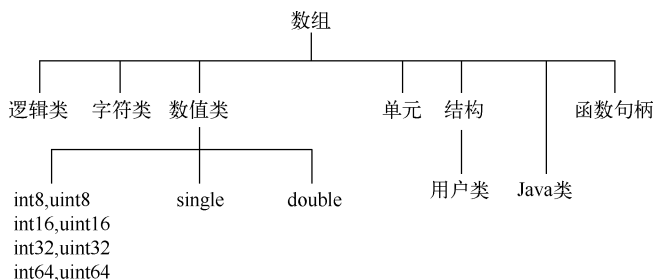


图 1-2 MATLAB 数据类型关系

MATLAB 的实数数据表达方式有两类：带小数点的形式直接表示和科学计数法，如 -2.567 、 $2.56e-56$ 。

MATLAB 的复数数据用特殊变量“i”和“j”来表示，这里 i 和 j 代表 $\sqrt{-1}$ 。复数可表示为 $z = a + b * i$ 或 $z = a + b * j$ 。

1.2.2 变量

MATLAB 中所有的变量都是用矩阵形式来表示的，即所有的变量都表示一个矩阵或者一个向量。其命名规则如下：

- 1) 变量名对大小写敏感。
- 2) 变量名必须以英文字母开头，其长度不能超过 63 个字符。
- 3) 变量名可以包含下连字符、数字，但不能包含空格符、标点。
- 4) 关键字不能作为变量名。

此外，当 MATLAB 启动时，MATLAB 的一些特殊变量会驻留在内存中。常用的特殊变量如表 1-2 所示。

表 1-2 特殊变量表

特殊变量	取值	特殊变量	取值
pi	圆周率 π	i 或 j	$i = j = \sqrt{-1}$
eps	计算机的最小数	flops	浮点运算数
inf	无穷大，如 $1/0$	NaN 或 nan	非数，如 $0/0$ 、 ∞/∞ 、 $0 \times \infty$
nargin	函数的输入变量数目	nargout	函数的输出变量数目
realmin	最小的可用正实数	realmax	最大的可用正实数

1.2.3 矩阵和数组

MATLAB 最基本也是最重要的功能就是进行实数或复数矩阵的运算。矩阵元素都是用方括号 ([]) 括住，每行内的元素间用逗号或空格隔开，行与行之间用分号或回车键隔开，元素可以是数值或表达式。矩阵的常用生成方式如表 1-3 所示。

表 1-3 矩阵生成方式

生成方式	实例	结果
显式元素列表输入	<code>c = [1 2;3 4;5 3 *2]</code>	3×2 矩阵
from:step:to 方式	<code>c = 1:0.5:2</code>	<code>c = [1 1.5 2]</code>
linspace 或 logspace	<code>c = linspace(1,6,3)</code>	<code>c = [1 3.5 6]</code>
矩阵生成函数	如表 1-4 所示	
MAT 数据文件加载	load 命令	

MATLAB 提供了很多能够产生特殊矩阵的函数，各函数的功能如表 1-4 所示。

表 1-4 矩阵生成函数

函数名	功能	实 例	
		输 入	结 果
zeros(m,n)	产生 $m \times n$ 的全 0 矩阵	zeros(2,3)	ans = 0 0 0 0 0 0
ones(m,n)	产生 $m \times n$ 的全 1 矩阵	ones(2,3)	ans = 1 1 1 1 1 1
rand(m,n)	产生均匀分布的随机矩阵 (元素值小于 1.0)	rand(2,3)	ans = 0.9501 0.6068 0.8913 0.2311 0.4860 0.7621
randn(m,n)	产生正态分布的随机矩阵	randn(2,3)	ans = -0.4326 0.1253 -1.1465 -1.6656 0.2877 1.1909
magic(N)	产生 N 阶魔方矩阵 (矩阵 的行、列和对角线上元素的和 相等)	magic(3)	ans = 8 1 6 3 5 7 4 9 2
eye(m,n)	产生 $m \times n$ 的单位矩阵	eye(3)	ans = 1 0 0 0 1 0 0 0 1

矩阵和 multidimensional array 都是由多个元素组成的，每个元素通过下标来标识。矩阵中的元素可以用全下标方式标识，即由行下标和列下标表示，一个 $m \times n$ 的 a 矩阵的第 i 行第 j 列的元素表示为 $a(i,j)$ 。矩阵运算有明确而严格的数学规则，矩阵运算规则是按照线性代数运算法则定义的。数组运算是按数组的元素逐个进行的。矩阵或数组的基本运算如表 1-5 所示。

表 1-5 矩阵和数组运算对比表

数组运算		矩阵运算	
命 令	含 义	命 令	含 义
A + B	对应元素相加	A + B	与数组运算相同
A - B	对应元素相减	A - B	与数组运算相同
S * B	标量 S 分别与 B 元素的积	S * B	与数组运算相同
A .* B	数组对应元素相乘	A * B	内维相同矩阵的乘积
S ./ B	S 分别被 B 的元素左除	S \ B	B 矩阵分别左除 S
A ./ B	A 的元素被 B 的对应元素除	A / B	A 的逆阵与 B 相乘
B \ A	结果一定与上行相同	B \ A	A 左除 B
A .^ S	A 的每个元素自乘 S 次	A ^ S	A 矩阵为方阵时，自乘 S 次
A ^ S	S 为小数时，对 A 各元素分别求非整数幂，得出矩阵	A ^ S	S 为小数时，方阵 A 的非整数乘方
S .^ B	分别以 B 的元素为指数求幂值	S ^ B	B 为方阵时，标量 S 的矩阵乘方
A .'	非共轭转置	A'	共轭转置
exp(A)	以自然数 e 为底，分别以 A 的元素为指数求幂	expm(A)	A 的矩阵指数函数
log(A)	对 A 的各元素求对数	logm(A)	A 的矩阵对数函数
sqrt(A)	对 A 的各元素求平方根	sqrtm(A)	A 的矩阵平方根函数
f(A)	求 A 各个元素的函数值	funm(A, 'f')	矩阵的函数运算

常用的矩阵运算函数如表 1-6 所示。

表 1-6 常用矩阵运算函数

函数名	功能
det(X)	计算方阵行列式
rank(X)	求矩阵的秩, 得出的行列式不为零的最大方阵边长
inv(X)	求矩阵的逆阵, 当方阵 X 的 det(X) 不等于零, 逆阵 X^{-1} 才存在。X 与 X^{-1} 相乘为单位矩阵
[v,d] = eig(X)	计算矩阵特征值和特征向量。如果方程 $Xv = vd$ 存在非零解, 则 v 为特征向量, d 为特征值
diag(X)	产生 X 矩阵的对角阵
[l,u] = lu(X)	方阵分解为一个准下三角方阵和一个上三角方阵的乘积。l 为准下三角阵, 必须交换两行才能成为真的下三角阵
[q,r] = qr(X)	$m \times n$ 阶矩阵 X 分解为一个正交方阵 q 和一个与 X 同阶的上三角矩阵 r 的乘积。方阵 q 的边长为矩阵 X 的 n 和 m 中较小者, 且其行列式的值为 1
[u,s,v] = svd(X)	$m \times n$ 阶矩阵 X 分解为三个矩阵的乘积, 其中 u, v 为 $n \times n$ 阶和 $m \times m$ 阶正交方阵, s 为 $m \times n$ 阶的对角阵, 对角线上的元素就是矩阵 X 的奇异值, 其长度为 n 和 m 中的较小者

1.2.4 字符串

在 MATLAB 中, 字符串是作为字符数组来引入的。一个字符串由多个字符组成, 用单引号 (') 来界定。字符串是按行向量进行存储的, 每一字符 (包括空格) 以其 ASCII 码的形式存放。常用的字符串函数如表 1-7 所示。

表 1-7 常用的字符串函数

函数名	功能
length(x)	用来计算字符串的长度 (即组成字符的个数)
double(x)	查看字符串的 ASCII 码储存内容
char(x)	将 ASCII 码转换成字符串形式
class(x) 或 ischar(x)	判断某一个变量是否为字符串
strcmp(x,y)	比较字符串 x 和 y 的内容是否相同
findstr(x,x1)	寻找在某个长字符串 x 中的子字符串 x1, 返回其起始位置
deblank(x)	删除字符串尾部的空格
str2mat 或 strvcat	字符串拼接函数, 不必考虑每个字符串的字符数是否相等
eval	字符串执行函数
disp	显示字符串内容

1.2.5 数学函数

MATLAB 中数学函数对数组的每个元素进行运算。数组的基本数学函数如表 1-8 所示。

表 1-8 基本数学函数

函 数 名	含 义	函 数 名	含 义
abs	绝对值或者复数模	rat	有理数近似
sqrt	平方根	mod	模除求余
real	实部	round	4 舍 5 入到整数
imag	虚部	fix	向最接近 0 取整
conj	复数共轭	floor	向最接近 $-\infty$ 取整
sin	正弦	ceil	向最接近 $-\infty$ 取整
cos	余弦	sign	符号函数
tan	正切	rem	求余数留数
asin	反正弦	exp	自然指数
acos	反余弦	log	自然对数
atan	反正切	log10	以 10 为底的对数
atan2	第四象限反正切	pow2	2 的幂
sinh	双曲正弦	bessel	贝赛尔函数
cosh	双曲余弦	gamma	伽马函数
tanh	双曲正切		

1.2.6 关系逻辑

MATLAB 含有基本的关系操作和逻辑操作，如 $<$ 、 $<=$ 、 $>$ 、 $>=$ 、 $==$ 、 $\sim =$ 、 $\&$ 、 $|$ 、 \sim 、 xor 等。此外，MATLAB 还包含一些特有的关系逻辑函数，如表 1-9 所示。

表 1-9 关系逻辑函数

函 数 名	功 能
all(A)	判断 A 的列向量元素是否全非 0，全非 0 则为 1
any(A)	判断 A 的列向量元素中是否有非 0 元素，有则为 1
isequal(A,B)	判断 A、B 对应元素是否全相等，相等为 1
isempty(A)	判断 A 是否为空矩阵，为空则为 1，否则为 0
isfinite(A)	判断 A 的各元素值是否有限，是则为 1
isinf(A)	判断 A 的各元素值是否无穷大，是则为 1
isnan(A)	判断 A 的各元素值是否为 NAN，是则为 1
isnumeric(A)	判断数组 A 的元素是否全为数值型数组
isreal(A)	判断数组 A 的元素是否全为实数，是则为 1
isprime(A)	判断 A 的各元素值是否为质数，是则为 1
isspace(A)	判断 A 的各元素值是否为空格，是则为 1
find(A)	寻找 A 数组非 0 元素的下标和值

1.3 MATLAB 绘图功能

1.3.1 二维曲线的绘制

1. 基本绘图命令 plot

plot 命令是 MATLAB 中最简单而且使用最广泛的一个绘图命令，用来绘制二维曲线。

语法：

```
plot(x)           % 绘制以 x 为纵坐标的二维曲线
plot(x,y)        % 绘制以 x 为横坐标、y 为纵坐标的二维曲线
```

表 1-10 给出了绘制二维、三维图形的一般步骤。

表 1-10 绘制二维、三维图形的一般步骤

步 骤	内 容
1	曲线数据准备：对于二维曲线，横坐标和纵坐标数据变量；对于三维曲面，矩阵参变量和对应的函数值
2	指定图形窗口和子图位置：默认时，打开 Figure No. 1 窗口或当前窗口、当前子图；也可以打开指定的图形窗口和子图
3	设置曲线的绘制方式：线型、色彩、数据点形
4	设置坐标轴：坐标的范围、刻度和坐标分格线
5	图形注释：图名、坐标名、图例、文字说明
6	着色、明暗、灯光、材质处理（仅对三维图形使用）
7	视点、三度（横、纵、高）比（仅对三维图形使用）
8	图形的精细修饰（图形句柄操作）：利用对象属性值设置；利用图形窗工具条进行设置

说明：

- 步骤 1 和 3 是最基本的绘图步骤，如果利用 MATLAB 的默认设置通常只需要这两个基本步骤就可以基本绘制出图形，而其他步骤并不完全必需。
- 步骤 2 一般在图形较多的情况下，需要指定图形窗口、子图时使用。
- 除了步骤 1、2、3 的其他步骤用户可以根据需要改变前后次序。

2. 多个图形绘制的方法

(1) 指定图形窗口

如果需要多个图形窗口同时打开，可以使用 figure 语句。

语法：

```
figure(n)       % 产生新图形窗口
```

(2) 同一窗口显示多个子图

如果需要在同一个图形窗口中布置几幅独立的子图，可以在 plot 命令前加上 subplot 命令来将一个图形窗口划分为多个区域，每个区域一幅子图。

语法：

```
subplot(m,n,k)   % 使(m×n)幅子图中的第k幅成为当前图
```

说明：将图形窗口划分为 $m \times n$ 幅子图， k 是当前子图的编号，“，”可以省略。子图的序号编排原则是：左上方为第 1 幅，先向右后向下依次排列，子图彼此之间独立。

(3) 同一窗口多次叠绘

为了在一个坐标系中增加新的图形对象，可以用“hold”命令来保留原图形对象。

语法：

```
hold on         % 使当前坐标系和图形保留
hold off        % 使当前坐标系和图形不保留
```

hold % 在以上两个命令中切换

说明：在设置了“hold on”后，如果画多个图形对象，则在生成新的图形时保留当前坐标系中已存在的图形对象，MATLAB 会根据新图形的大小，重新改变坐标系的比例。

(4) 双纵坐标图

语法：

plotyy(x1,y1,x2,y2) % 以左、右不同纵轴绘制两条曲线

说明：左纵轴用于 (x1, y1) 数据，右纵轴用于 (x2, y2) 数据来绘制两条曲线。坐标轴的范围、刻度都自动产生。

3. 曲线的线型、颜色和数据点形

plot 命令还可以设置曲线的线段类型、颜色和数据点形等，如表 1-11 所示。

表 1-11 线段、颜色与数据点形

颜色		数据点间连线		数据点形	
类型	符号	类型	符号	类型	符号
黄色	y (Yellow)	实线 (默认)	-	实点标记	.
品红色 (紫色)	m (Magenta)	点线	:	圆圈标记	o
青色	c (Cyan)	点画线	-.	叉号形 ×	x
红色	r (Red)	虚线	--	十字形 +	+
绿色	g (Green)			星号标记 *	*
蓝色	b (Blue)			方块标记 □	s
白色	w (White)			钻石形标记 ◇	d
黑色	k (Black)			向下的三角形标记	v
				向上的三角形标记	^
				向左的三角形标记	<
				向右的三角形标记	>
				五角星标记 ☆	p
				六连形标记	h

语法：

plot(x,y,s)

说明：x 为横坐标矩阵；y 为纵坐标矩阵；s 为类型说明字符串参数，s 字符串可以是线段类型、颜色和数据点形三种类型的符号之一，也可以是三种类型符号的组合。

4. 设置坐标轴和文字标注

(1) 坐标轴的控制

用坐标控制命令 axis 来控制坐标轴的特性，表 1-12 列出其常用控制命令。

表 1-12 常用的坐标控制命令

命令	含义
axis auto	使用默认设置
axis tight	把数据范围直接设为坐标范围
axis off	取消轴背景
axis on	使用轴背景

(续)

命 令	含 义
axis ij	矩阵式坐标, 原点在左上方
axis xy	普通直角坐标, 原点在左下方
axis([xmin, xmax, ymin, ymax])	设定坐标范围, 必须满足 $xmin < xmax$, $ymin < ymax$

(2) 分格线

语法:

```
grid on           % 显示分格线
grid off         % 不显示分格线
grid             % 在以上两个命令间切换
```

说明: 不显示分格线是 MATLAB 的默认设置。分格线的疏密取决于坐标刻度, 如果要改变分格线的疏密, 必须先定义坐标刻度。

(3) 坐标框

语法:

```
box on           % 使当前坐标框呈封闭形式
box off         % 使当前坐标框呈开启形式
box             % 在以上两个命令间切换
```

说明: 在默认情况下, 所画的坐标框呈封闭形式。

5. 文字标注

(1) 添加图名

语法:

```
title(s)        % 书写图名
```

说明: s 为图名, 为字符串, 可以是英文或中文。

(2) 添加坐标轴名

语法:

```
xlabel(s)       % 横坐标轴名
ylabel(s)       % 纵坐标轴名
```

(3) 添加图例

语法:

```
legend(s, pos)  % 在指定位置建立图例
legend off      % 擦除当前图中的图例
```

说明: 参数 s 是图例中的文字注释, 如果有多个注释则可以用 's1', 's2', ... 的方式。用 legend 命令在图形窗口中产生图例后, 还可以用鼠标对其进行拖拉操作, 将图例拖到合适的位置。

(4) 添加文字注释

语法:

```
text(xt,yt,s)
```

```
% 在图形的(xt,yt)坐标处书写文字注释
```

6. 特殊符号

表 1-13 给出了图形标识用的希腊字母、数学符号和特殊字符。

表 1-13 图形标识用的希腊字母、数学符号和特殊字符

类 别	命 令	字 符	命 令	字 符	命 令	字 符	命 令	字 符
希腊字母	\alpha	α	\ eta	η	\ nu	ν	\ upsilon	υ
	\beta	β	\ theta	θ	\ xi	ξ	\ Upsilon	Υ
	\epsilon	ϵ	\ Theta	Θ	\ Xi	Ξ	\ phi	ϕ
	\gamma	γ	\ iota	ι	\ pi	π	\ Phi	Φ
	\Gamma	Γ	\ zeta	ζ	\ Pi	Π	\ chi	χ
	\delta	δ	\ kappa	κ	\ rho	ρ	\ psi	ψ
	\Delta	Δ	\ mu	μ	\ tau	τ	\ Psi	Ψ
	\omega	ω	\ lambda	λ	\ sigma	σ		
	\Omega	Ω	\ Lambda	Λ	\ Sigma	Σ		
数学符号	\approx	\approx	\oplus	\equiv	\neq	\neq	\leq	\leq
	\geq	\geq	\pm	\pm	\times	\times	\div	\div
	\int	\int	\exists	∞	\infty	∞	\in	\in
	\sim	\cong	\forall	\sim	\angle	\angle	\perp	\perp
	\cup	\cup	\cap	\cap	\vee	\vee	\wedge	\wedge
	\surd	\surd	\otimes	\otimes	\oplus	\oplus		
箭头	\uparrow	\uparrow	\downarrow	\downarrow	\rightarrow	\rightarrow	\leftarrow	\leftarrow
	\leftrightarrow	\leftrightarrow	\updownarrow	\updownarrow				

如果需要对文字进行上下标设置,或设置字号大小,则必须在文字标识前先使用表 1-14 中所示的设置值。

表 1-14 文字设置

命 令	含 义
\fontname{s}	字体的名称, s 为 Times New Roman、Courier、宋体等
\fontsize{n}	字号大小, n 为正整数, 默认为 10 (points)
\s	字体风格, s 可以为 bf(黑体)、it(斜体一)、sl(斜体二)、rm(正体)等
^ s	将 s 变为上标
_ s	将 s 变为下标

7. 交互式图形命令

(1) ginput 命令

ginput 命令是从图上获取数据。

语法:

```
[x,y] = ginput(n)
```

```
% 用鼠标从图形上获取 n 个点的坐标(x,y)
```

说明：参数 n 应为正整数，是通过鼠标从图上获得数据点的个数； x 、 y 用来存放所取点的坐标。

(2) gtext 命令

gtext 命令是把字符串放置到图形中鼠标所指定的位置上。

语法：

```
gtext( s ) % 用鼠标把字符串放置到图形上
```

说明：如果参数 s 是单个字符串或单行字符串矩阵，那么一次鼠标操作就可把全部字符以单行形式放置在图上；如果参数 s 是多行字符串矩阵，那么每操作一次鼠标，只能放置一行字符串，需要通过多次鼠标操作，把一行一行的字符串放在图形的不同位置。

1.3.2 MATLAB 的三维图形绘制

1. 绘制三维线图命令 plot3

plot3 是用来绘制三维曲线的，它的使用格式与二维绘图的 plot 命令很相似。

语法：

```
plot3(x,y,z, s) % 绘制三维曲线
plot3(x1,y1,z1, s1, x2,y2,z2, s2, ...) % 绘制多条三维曲线
```

说明：当 x 、 y 、 z 是同维向量时，则绘制以 x 、 y 、 z 元素为坐标的三维曲线；当 x 、 y 、 z 是同维矩阵时，则绘制三维曲线的条数等于矩阵的列数。 s 是指定线型、色彩、数据点形的字符串。

2. 绘制三维网线图和曲面图

(1) meshgrid 命令

为了绘制三维立体图形，MATLAB 的方法是将 x 方向划分为 m 份，将 y 方向划分为 n 份，meshgrid 命令是以 x 、 y 向量为基准，来产生在 $x-y$ 平面的各栅格点坐标值的矩阵。

语法：

```
[X,Y] = meshgrid(x,y)
```

说明： X 、 Y 是栅格点的坐标，为矩阵； x 、 y 为向量。

(2) 三维网线图

语法：

```
mesh(z) % 画三维网线图
mesh(x,y,z,c)
```

说明：当只有参数 z 时，以 z 矩阵的行下标作为 x 坐标轴，把 z 的列下标当作 y 坐标轴； x 、 y 分别为 x 、 y 坐标轴的自变量；当有 x 、 y 、 z 参数时， c 是指定各点的颜色矩阵，当 c 省略时默认颜色矩阵是 z 的数据。如果 x 、 y 、 z 、 c 四个参数都有，则应该都是维数相同的矩阵。

(3) 三维曲面图

语法：

```
surf(z) % 画三维曲面图
```

```
surf(x,y,z,c)
```

说明：参数设置与 mesh 命令相同。

1.3.3 立体图形与图轴的控制

1. 网格

如果要使被遮盖的网格也能呈现出来，可用“hidden off”命令。

语法：

```
hidden off           % 显示被遮盖的网格
hidden on           % 隐藏被遮盖的网格
```

2. 改变视角

三维图形的观测角度不同则显示也不同，如果要改变观测角度，可用“view”命令。

语法：

```
view([az,el])       % 通过方位角和俯仰角改变视角
view([vx,vy,vz])    % 通过直角坐标改变视角
```

说明：az 表示方位角，el 表示俯仰角；vx、vy、vz 表示直角坐标。

1.4 MATLAB 的特殊图形绘制

1.4.1 条形图

条形图常用于对统计的数据进行作图，特别适用于少量且离散的数据。

语法：

```
bar(x,y,width,'参数') % 画条形图
bar3(x,y,z,width,'参数') % 画三维条形图
```

说明：x 是横坐标向量，默认值是 1:m，m 为 y 的向量长度；y 是纵坐标，可以是向量或矩阵，当是向量时每个元素对应一个竖条，当是 m×n 的矩阵时，将画出 m 组竖条，每组包含 n 条；width 是竖条的宽度，默认宽度是 0.8，如果宽度大于 1，则条与条之间将重叠；' 参数 有 grouped（分组式）和 stacked（累加式），默认为 grouped。bar3 命令的格式也相同，y 必须是单调增加或减小，默认时为 1:m；' 参数 除了 grouped 和 stacked 之外还有 detached（分离式）。

1.4.2 直方图

语法：

```
hist(y,m)           % 统计每段的元素个数并画出直方图
hist(y,x)
```

说明：m 是分段的个数，默认时为 10；x 是向量，用于指定所分每个数据段的中间值；y 可以是向量或矩阵，如果是矩阵则按列分段。

1.4.3 饼图

饼图是用于显示向量中的各元素占向量元素总和的百分比，可以用 pie 和 pie3 命令分别绘制二维和三维饼图。

语法：

```
pie(x,explode',label')           % 画二维饼图
pie3(x,explode',label')         % 画三维饼图
```

说明：x 是向量；explode 是与 x 同长度的向量，用来决定是否从饼图中分离对应的一部分块，非零元素表示该部分需要分离；'label' 是用来标注饼图的字符串数组。

1.4.4 对数坐标和极坐标图

1. 对数坐标图形

对数坐标图形有 semilogx、semilogy 和 loglog 命令。

语法：

```
semilogx(x,y',参数)             % 绘制 x 为对数坐标的曲线
semilogy(x,y',参数)            % 绘制 y 为对数坐标的曲线
loglog(x,y',参数)              % 绘制 x、y 都为对数坐标的曲线
```

说明：参数和 plot 命令一样，只是坐标不同。

2. 极坐标图

极坐标图由 polar 命令来实现。

语法：

```
polar(theta,radius',参数)       % 绘制极坐标图
```

说明：theta 为相角；radius 为与原点的距离。

1.4.5 对话框

1. 输入信息对话框

输入对话框为用户的输入信息提供了界面，使用 inputdlg 命令创建。

语法：

```
answer = inputdlg(prompt,title,lineno,defans,addopts) % 创建输入对话框
```

说明：answer 返回用户的输入，为元胞数组；prompt 为提示信息字符串，用引号括起来，为元胞数组；title 为标题字符串，用引号括起来，可以省略；lineno 用于指定输入值的行数，可以省略；defans 为输入项的默认值，用引号括起来，是元胞数组，可以省略；addopts 指定对话框是否可以改变大小，取 on 或 off，默认时为 off，表示不能改变大小，为有

模式对话框（有模式对话框是指在对话框关闭之前，用户无法进行其他程序的运行），如果为 on 则可以改变大小，自动变为无模式对话框。

2. 输出信息对话框

语法：

```
msgbox(message,title,icon,icondata,iconcmap,CreateMode) % 创建消息框
```

说明：message 为显示的信息，可以是字符串或数组；title 为标题，是字符串，可省略；icon 为显示的图标，可取值为“none”（无图标）、“error”（出错图标）、“help”（帮助图标）、“warn”（警告图标）或“custom”（自定义图标），也可省略；当使用“custom”时，用 icondata 定义图标的数据，用 iconcmap 定义图标的颜色映象；CreateMode 为对话框的产生模式，可省略，取值为“modal”（有模式）、“replace”（无模式，可代替同名的对话框）、“non-modal”（默认为无模式）。

1.4.6 句柄图形

1. 句柄图形体系

句柄图形是一种面向对象的绘图系统，又称为低层图形。句柄图形体系由若干个图形对象组成，如图 1-3 所示。

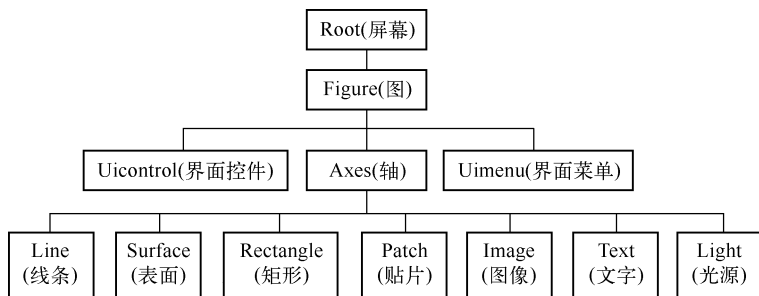


图 1-3 句柄图形体系

2. 对象句柄的获取

MATLAB 提供了三个获取当前对象句柄的命令，分别是 gcf、gca、gco。

语法：

```
gcf % 获取当前图形窗口句柄
gca % 获取当前坐标轴句柄
gco % 获取被鼠标最近点击对象的句柄
```

1.5 MATLAB 程序设计

1.5.1 M 文件

M 文件有两种形式：M 脚本文件和 M 函数文件。MATLAB 的 M 文件是通过 M 文件编辑/

调试器 (Editor/Debugger) 窗口来创建的。

M 脚本文件的特点:

1) 脚本文件中的命令格式和前后位置, 与在命令窗口中输入的没有任何区别。

2) MATLAB 在运行脚本文件时, 只是简单地按顺序从文件中读取一条条命令, 送到 MATLAB 命令窗口中去执行。

3) 与在命令窗口中直接运行命令一样, 脚本文件运行产生的变量都是驻留在 MATLAB 的工作空间 (workspace) 中, 可以很方便地查看变量, 除非用 clear 命令清除; 脚本文件的命令也可以访问工作空间的所有数据, 因此要注意避免变量的覆盖而造成程序出错。

M 函数文件的特点:

1) 第一行总是以 “function” 引导的函数声明行。函数声明行的格式:

```
function [输出变量列表] = 函数名(输入变量列表)
```

2) 函数文件在运行过程中产生的变量都存放在函数本身的工作空间。

3) 当文件执行完最后一条命令或遇到 “return” 命令时, 就结束函数文件的运行, 同时函数工作空间的变量就被清除。

4) 函数的工作空间随具体的 M 函数文件调用而产生, 随调用结束而删除, 是独立的、临时的, 在 MATLAB 运行过程中可以产生任意多个临时的函数空间。

1.5.2 程序流程控制

1. for ... end 循环结构

语法:

```
for 循环变量 = array
    循环体
end
```

说明: 循环体被循环执行, 执行的次数就是 array 的列数, array 可以是向量也可以是矩阵, 循环变量依次取 array 的各列, 每取一次循环体执行一次。

2. while ... end 循环结构

语法:

```
while 表达式
    循环体
end
```

说明: 只要表达式为逻辑真, 就执行循环体; 一旦表达式为假, 就结束循环。表达式可以是向量也可以是矩阵, 如果表达式为矩阵, 则当所有的元素都为真时才执行循环体, 如果表达式为 nan, MATLAB 认为是假, 不执行循环体。

3. If...else...end 条件转移结构

语法:

```
if 条件式 1
    语句段 1
elseif 条件式 2
    语句段 2
...
else
    语句段 n + 1
end
```

说明：当有多个条件时，当条件式 1 为假时判断 elseif 的条件式 2，如果所有的条件式都不满足，则执行 else 的语句段 n + 1，当条件式为真时执行相应的语句段；If...else...end 结构也可以是没有 elseif 和 else 的简单结构。

4. switch...case 开关结构

语法：

```
switch 开关表达式
case 表达式 1
    语句段 1
case 表达式 2
    语句段 2
...
otherwise
    语句段 n
end
```

说明：

1) 将开关表达式依次与 case 后面的表达式进行比较，如果表达式 1 不满足，则与下一个表达式 2 比较，如果都不满足则执行 otherwise 后面的语句段 n；一旦开关表达式与某个表达式相等，则执行其后面的语句段。

2) 开关表达式只能是标量或字符串。

3) case 后面的表达式可以是标量、字符串或元胞数组，如果是元胞数组则将开关表达式与元胞数组的所有元素进行比较，只要某个元素与开关表达式相等，就执行其后的语句段。

5. try... catch... end 试探结构

语法：

```
try
    语句段 1
catch
    语句段 2
end
```

说明：首先试探性地执行语句段 1，如果在此段语句执行过程中出现错误，则将错误信息赋给保留的 lasterr 变量，并放弃这段语句，转而执行语句段 2，当执行语句段 2 又出现错

误时，终止该结构。

6. 流程控制语句

(1) break 命令

break 命令可以使包含 break 的最内层的 for 或 while 语句强制终止，立即跳出该结构，执行 end 后面的命令，break 命令一般和 if 结构结合使用。

(2) continue 命令

continue 命令用于结束本次 for 或 while 循环，只结束本次循环而继续进行下次循环。

(3) return 命令

return 命令是终止当前命令的执行，并且立即返回到上一级调用函数或等待键盘输入命令，可以用来提前结束程序的运行。

注意：如果程序进入死循环，则按〈Ctrl + break〉键来终止程序的运行。

(4) pause 命令

pause 命令用来使程序暂停运行，等待用户按任意键继续。

语法：

```
pause           % 暂停  
pause(n)       % 暂停 n 秒
```

(5) keyboard 命令

keyboard 命令用来使程序暂停运行，等待键盘命令，执行完自己的工作后，输入 return 语句，程序就继续运行。

(6) input 命令

input 命令用来提示用户应该从键盘输入数值、字符串和表达式，并接受该输入。

1.5.3 函数调用和参数传递

1. 子函数

在一个 M 函数文件中，可以包含一个以上的函数，其中只有一个是主函数，其他则为子函数。

1) 在一个 M 文件中，主函数必须出现在最上方，其后是子函数，子函数的次序无任何限制。

2) 子函数不能被其他文件的函数调用，只能被同一文件中的函数（可以是主函数或子函数）调用。

3) 同一文件的主函数和子函数变量的工作空间相互独立。

4) 用 help 和 lookfor 命令不能提供子函数的帮助信息。

2. 私有函数

私有函数是指存放在 private 子目录中的 M 函数文件，具有以下性质：

1) 在 private 目录下的私有函数，只能被其父目录的 M 函数文件所调用，而不能被其他目录的函数调用，对其他目录的文件私有函数是不可见的，私有函数可以和其他目录下的函数重名。

- 2) 私有函数父目录的 M 脚本文件也不可调用私有函数。
- 3) 在函数调用搜索时, 私有函数优先于其他 MATLAB 路径上的函数。

3. 函数的参数

函数的输入输出参数的个数可以通过变量 nargin 和 nargs 获得, nargin 用于获得输入参数的个数, nargs 用于获得输出参数的个数。

语法:

nargin	% 在函数体内获取实际输入变量的个数
nargs	% 在函数体内获取实际输出变量的个数
nargin('fun')	% 在函数体外获取定义的输入参数个数
nargs('fun')	% 在函数体外获取定义的输出参数个数

1.6 MATLAB 设计实例——FFT 频谱分析

1.6.1 FFT 基础

离散傅里叶变换 (DFT), 是傅里叶变换在时域和频域上都呈现离散的形式, 将时域信号的采样变换为在离散时间傅里叶变换频域的采样。在形式上, 变换两端 (时域和频域上) 的序列是有限长的, 而实际上这两组序列都应当被认为是离散周期信号的主值序列。即使对有限长的离散信号作 DFT, 也应当将其看作经过周期延拓成为周期信号再作变换。在实际应用中通常采用快速傅里叶变换 (FFT) 以高效计算 DFT。DFT 的定义如下:

$$X(k) = \sum_{j=1}^N x(j) w_N^{(j-1)(k-1)} \quad (1-1)$$

$$x(j) = (1/N) \sum_{k=1}^N X(k) w_N^{-(j-1)(k-1)} \quad (1-2)$$

其中, $w_N = e^{(-2\pi i)/N}$ 。

在 MATLAB 中使用 FFT 函数对信号进行频谱分析, 其语法如下:

X = FFT(x);	% 计算向量 x 的快速傅里叶变换
X = FFT(x,N);	% 计算向量 x 的 N 点快速傅里叶变换
x = IFFT(X);	% 计算向量 x 的快速傅里叶反变换
x = IFFT(X,N)	% 计算向量 x 的 N 点快速傅里叶反变换

1.6.2 基于 FFT 的信号频谱分析

1. 设计要求

- 1) 用 MATLAB 产生正弦波及白噪声信号, 并显示各自时域波形图。
- 2) 进行 FFT 变换, 显示各自频谱图。
- 3) 做出两种信号的均方根图谱, 功率图谱, 以及对数方均根图谱。
- 4) 用 IFFT 傅里叶反变换恢复信号, 并显示时域波形图。

2. 设计参考例程

```
clc
clear all
% ***** 1. 正弦波 ***** %
fs = 100; % 设定采样频率
N = 128;
n = 0:N-1;
t = n/fs;
f0 = 10; % 设定正弦信号频率
% 生成正弦信号
x = sin(2 * pi * f0 * t);
figure(1);
subplot(231);
plot(t,x); % 作正弦信号的时域波形
xlabel( '时间/s' );
ylabel( '幅值' );
title( '时域波形' );
grid;
% 进行 FFT 变换并做频谱图
y = fft(x,N); % 进行 FFT 变换
mag = abs(y); % 求幅值
f = (0:length(y)-1) * fs/length(y); % 进行对应的频率转换
subplot(232);
plot(f,mag); % 作频谱图
axis([0,100,0,80]);
xlabel( '频率/Hz' );
ylabel( '幅值' );
title( '幅频谱图' );
grid;
% 求均方根谱
sq = abs(y);
subplot(233);
plot(f,sq);
xlabel( '频率/Hz' );
ylabel( '均方根谱' );
title( '均方根谱' );
grid;
% 求功率谱
power = sq.^2;
subplot(234);
```



```
plot(f,power);
xlabel( 频率/Hz );
ylabel( 功率谱 );
title( 功率谱 );
grid;
% 求对数谱
ln = log(sq);
subplot(235);
plot(f,ln);
xlabel( 频率/Hz );
ylabel( 对数谱 );
title( 对数谱 );
grid;
% 用 IFFT 恢复原始信号
xifft = ifft(y);
magx = real(xifft);
ti = [0:length(xifft) - 1]/fs;
subplot(236);
plot(ti,magx);
xlabel( 时间/s );
ylabel( 幅值 );
title( IFFT 后的信号波形 );
grid;

% ***** 2. 白噪声 ***** %
fs = 50; % 设定采样频率
t = -5:0.1:5;
x = rand(1,100);
figure(2);
subplot(231);
plot(t(1:100),x); % 作白噪声的时域波形
xlabel( 时间(s) );
ylabel( 幅值 );
title( 时域波形 );
grid;
% 进行 FFT 变换并做频谱图
y = fft(x); % 进行 FFT 变换
mag = abs(y); % 求幅值
f = (0:length(y) - 1) * fs/length(y); % 进行对应的频率转换
subplot(232);
plot(f,mag); % 作频谱图
```



```
xlabel( 频率/Hz );
ylabel( 幅值 );
title( 幅频谱图 );
grid;
% 求均方根谱
sq = abs( y );
subplot(233);
plot( f, sq );
xlabel( 频率/Hz );
ylabel( 均方根谱 );
title( 均方根谱 );
grid;
% 求功率谱
power = sq.^2;
subplot(234);
plot( f, power );
xlabel( 频率/Hz );
ylabel( 功率谱 );
title( 功率谱 );
grid;
% 求对数谱
ln = log( sq );
subplot(235);
plot( f, ln );
xlabel( 频率/Hz );
ylabel( 对数谱 );
title( 对数谱 );
grid;
% 用 IFFT 恢复原始信号
xifft = ifft( y );
magx = real( xifft );
ti = [ 0; length( xifft ) - 1 ] / fs;
subplot(236);
plot( ti, magx );
xlabel( 时间/s );
ylabel( 幅值 );
title( IFFT 后的信号波形 );
grid;
```

3. 实验结果

实验结果如图 1-4 和图 1-5 所示。

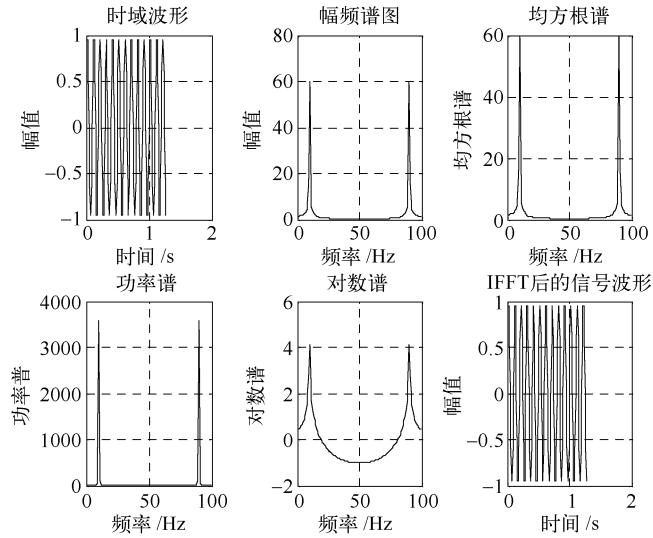


图 1-4 正弦信号的频谱分析结果

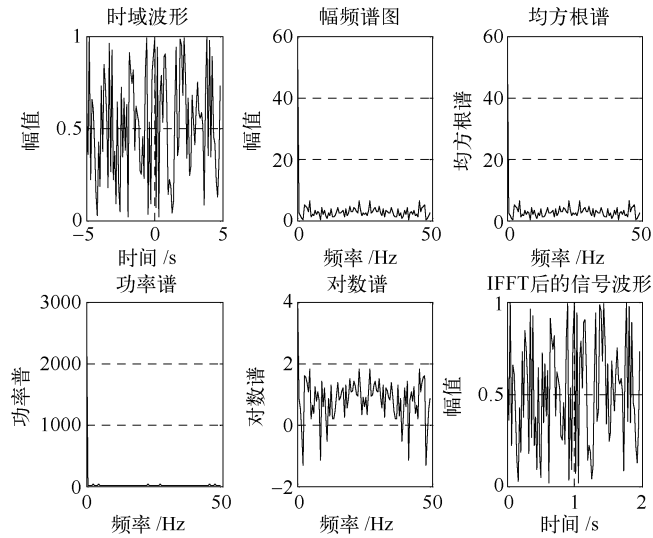


图 1-5 白噪声的频谱分析结果

第2章 语音信号处理基础实验

2.1 语音采集与读写实验

2.1.1 实验目的

- 1) 了解 MATLAB 采集语音信号的原理及常用命令。
- 2) 熟练掌握基于 MATLAB 的语音文件的创建、读写等基本操作。
- 3) 学会使用 plot 命令来显示语音信号波形，并掌握基本的标注方法。

2.1.2 实验原理

1. 语音信号特点

20 世纪 90 年代以来，语音信号采集与分析在实用化方面取得了许多实质性的研究进展。其中，语音识别逐渐由实验室走向实用化。一方面，对声学语音学统计模型的研究逐渐深入，鲁棒的语音识别、给予语音段的建模方法及隐马尔可夫模型与人工神经网络的结合成为研究的热点。另一方面，为了语音识别实用化的需要，讲者自适应、听觉模型、快速搜索识别算法以及进一步的语音模型研究等课题备受关注。

通过对大量语音信号的观察和分析发现，语音信号主要有下面两个特点：

- 1) 在频域内，语音信号的频谱分量主要集中在 300 ~ 3400 Hz 的范围内。利用这个特点，可以用一个防混叠的带通滤波器将此范围内的语音信号频率分出，然后按 8 kHz 的采样率对语音信号进行采样，就可以得到离散的语音信号。
- 2) 在时域内，语音信号具有“短时性”的特点，即在总体上，语音信号的特征是随着时间而变化的，但在一段较短的时间间隔内，语音信号保持平稳。在浊音段表现出周期信号的特征，在清音段表现出随机噪声的特征。

2. 语音信号采集的基本原理

为了将原始模拟语音信号变为数字信号，必须经过采样和量化两个步骤，从而得到时间和幅度上均为离散的数字信号语音。采样是信号在时间上的离散化，即按照一定时间间隔 Δt 在模拟信号 $x(t)$ 上逐点采取其瞬时值。采样时必须要注意满足奈奎斯特定理，即采样频率 f_s 必须以高于受测信号的最高频率两倍以上速度进行取样，才能正确地重建信号。

在 Windows 环境下，学生可以使用 Windows 自带的录音机录制语音文件，图 2-1 是基于 PC 的语音信号采集过程，声卡可以完成语音波形的 A/D 转换，获得 WAV 文件。通过 Windows 录制的语音信号，一方面可以为后续实验储备原始语音，另一方面可以与通过其他方式录制的语音进行比对，比如使用 MATLAB 自带的 wavrecord 函数进行录制。

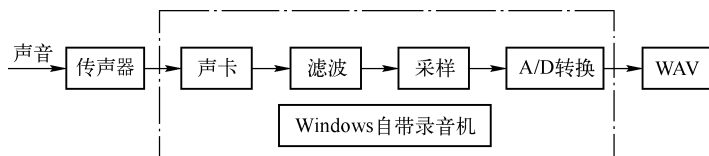


图 2-1 PC 语音信号采集原理图

3. 基于 MATLAB 的语音信号采集与读写方法

MATLAB 将声卡作为对象处理，其后的一切操作都不与硬件直接相关，而是通过对该对象的操作来作用于硬件设备（声卡）。操作时首先要对声卡产生一个模拟输入对象，并给模拟输入对象添加一个通道设置采样频率，然后就可以启动设备对象，开始采集数据，采集完成后停止对象并删除对象。

常用的相关 MATLAB 函数包括 wavrecord、wavread、wavwrite、wavplay 等，下面分别介绍其用法。

(1) wavrecord 函数

● 功能说明：

使用基于 PC 的音频输入设备进行录音。

函数语法：

```

y = wavrecord(n, Fs)
y = wavrecord(..., ch)
y = wavrecord(..., 'dtype')
  
```

● 参数解析：

$y = \text{wavrecord}(n, F_s)$ ——以 F_s 采样率 (Hz) 记录 n 个声音信号采样点。默认的 F_s 为 11025Hz。

$y = \text{wavrecord}(\dots, ch)$ ——设置音频设备的通道数 ch 。 ch 可以是 1 或 2，分别代表单声道或立体声。默认的通道数 ch 为 1。

$y = \text{wavrecord}(\dots, 'dtype')$ ——指定数据类型 $dtype$ 来记录声音。表 2-1 为数据类型 $dtype$ 与采样点的位数、数据范围的关系。

表 2-1 dtype 参数说明

数据类型	采样位数	输出的 y 数据范围
'double'	16	$-1.0 \leq y < 1.0$
'single'	16	$-1.0 \leq y < 1.0$
'int16'	16	$-32768 \leq y \leq 32767$
'uint8'	8	$0 \leq y \leq 255$

● 参考例程：

```

Fs = 11025; % 采样率为 11025Hz
y = wavrecord(5 * Fs, Fs, 'int16'); % 录制 5 s 的数据
wavplay(y, Fs); % 播放录制的音频
  
```

(2) wavwrite 函数

● 功能说明:

写入 wav 声音文件。

● 函数语法:

```
wavwrite(y,filename)
wavwrite(y,Fs,filename)
wavwrite(y,Fs,N,filename)
```

● 参数解析:

wavwrite (y, filename) ——将保存在变量 y 的数据保存到 wav 文件 filename 中。默认的采样率为 8000 Hz, 采样位数为 16 位。y 的各列为各个通道。因此立体声数据是两列的矩阵。

wavwrite (y, Fs, filename) ——按给定的采样率 Fs, 将保存在变量 y 的数据保存到 wav 文件 filename 中。缺省的采样位数为 16 位。

wavwrite (y, Fs, N, filename) ——按给定的采样率 Fs 和采样位数 N, 将保存在变量 y 的数据保存到 wav 文件 filename 中。N 可选值为 8, 16, 24 及 32。输入的数据范围: y 的取值范围与采样位数 N 以及 y 的数据类型有关, 关系如表 2-2 所示。

表 2-2 y 的取值范围与采样位数 N 及 y 数据类型的关系

N	y 的数据类型	y 的取值范围	输出格式
8	uint8	$0 \leq y \leq 255$	uint8
8	single 或 double	$-1.0 \leq y < 1.0$	unit8
16	int16	$-32768 \leq y \leq 32767$	int16
16	single 或 double	$-1.0 \leq y < 1.0$	int16
24	int32	$-2^{23} \leq y \leq 2^{23}-1$	int32
24	single 或 double	$-1.0 \leq y < 1.0$	int32
32	single 或 double	$-1.0 \leq y \leq 1.0$	single

● 参考例程:

load handel.mat%载入 MATLAB 自带的示例音频数据文件

```
hfile = 'Data_wavread.wav' ; %准备写的音频数据文件
wavwrite(y,Fs,hfile) %将 y 以 Fs 采样率写到文件中
```

(3) wavread 函数

● 功能说明:

读取 wav 声音文件。

● 函数语法:

```
y = wavread(filename)
[y,Fs] = wavread(filename)
[y,Fs,nbits] = wavread(filename)
[y,Fs,nbits,opts] = wavread(filename)
[...] = wavread(...,fmt)
```

- 参数解析:

`y = wavread(filename)` ——载入由 `filename` 字符串指定的 wav 文件, `y` 为返回采样点的数据。若文件名 `filename` 不包含扩展名, `wavread` 函数将添加 `.wav` 扩展名。

`[y, Fs] = wavread(filename)` ——返回文件的采样率 `Fs` (Hz)。

`[y, Fs, nbits] = wavread(filename)` ——返回每次采样的位数 `nbits` (位)。

`[y, Fs, nbits, opts] = wavread(filename)` ——返回包含 wav 文件额外信息的结构体 `opts`。`opts` 具体的字段与文件有关。`opts` 典型的两个字段为 `fmt` 和 `info`, 分别代表声音格式信息和描述标题作者等信息的文本。

`[...] = wavread(...,fmt)` ——用于指定文件中采样数据 `y` 的格式。`fmt` 可以是表 2-3 中的任意类型。默认为 `double` 类型, 即归一化的双精度采样。

输出数据的范围:

返回的 `y` 的范围与指定的数据格式 `fmt` 有关, 如表 2-3 所示。

表 2-3 输出数据格式

位 数	MATLAB 数据类型	数据范围
8	uint8	$0 \leq y \leq 255$
16	int16	$-32768 \leq y \leq 32767$
24	int32	$-2^{23} \leq y \leq 2^{23}-2$
32	single	$-1.0 \leq y < 1.0$
$N < 32$	Double	$-1.0 \leq y < 1.0$
$N = 32$	Double	$-1.0 \leq y \leq 1.0$ 注意存储在 wav 文件中的 32 位的数据采样率, 数据可能超过 -1.0 或 1.0

- 参考例程:

```
load handel.mat           % 载入 MATLAB 自带的示例音频数据文件
hfile = 'Data_wavread.wav'; % 准备写的音频数据文件
wavwrite(y,Fs,hfile)     % 将 y 以 Fs 采样率写到文件中
clear y Fs               % 清除载入的 y 以及 Fs
[y,Fs,nbits,readinfo] = wavread(hfile); % 从声音文件中载入数据,y 为从声音文件载入的数据,Fs
                                % 为采样率,nbits 为采样使用的位数。
sound(y,Fs);             % 播放声音
pause(4)                 % 暂停 4 s,与后面的声音分隔开
```

(4) wavplay 函数

- 功能说明:

在基于 PC 的音频输出设备上播放录制的声音。

- 函数语法:

```
wavplay(y,Fs)
wavplay(y,Fs,mode)
```

- 参数解析:

`wavplay(y, Fs)` ——在基于 PC 的音频输出设备上, 播放保存在向量 `y` 中的音频信号。`Fs` 是采样率 (Hz)。默认的 `Fs` 为 11025 Hz。`wavplay` 支持单通道或双通道 (立体声) 的音频信

号。播放立体声 y 必须是两列的矩阵。

`wavplay(y, Fs, mode)`——指定在命令行下如何交互播放音频。`mode` 的可选值为：
'sync' (默认)：直到声音播放结束才返回到命令窗口；'async'：声音一播放就返回命令窗口，当音频正在播放时，以 `async` 模式再次调用 `wavplay`，`wavplay` 将阻塞命令行直至上一个播放结束。信号 y 可以是表 2-4 中所列四种数据类型之一。

表 2-4 y 的数据类型

数据类型	量 化	数据类型	量 化
双精度 (默认)	每次采样 16 位	16 位无符号整型	每次采样 16 位
单精度	每次采样 16 位	8 位无符号整型	每次采样 8 位

• 参考例程：

```
load chirp;           % 载入 chirp.mat 到 y 和 Fs 中
y1 = y;Fs1 = Fs;     % 备份 y 和 Fs 变量到 y1 和 Fs1
load gong;           % 载入 gong.mat 到 y 和 Fs 中
wavplay(y1,Fs1,'async') % 播放 chirp 文件中的音频
wavplay(y,Fs)        % 上述文件播放完后,才播放 gong 文件的音频
```

2.1.3 实验步骤及要求

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

1) 编写 MATLAB 程序实现录制语音信号“你好，欢迎”，并保存为 `C2_1_y_1.wav` 文件，要求采样频率为 16000 Hz，采样精度 16 bit。

2) 使用 `wavread` 函数读取 `C2_1_y_1.wav` 文件，并使用 `plot` 函数显示出来。要求：横轴和纵轴带有标注。横轴的单位为秒 (s)，纵轴显示归一化后的数值。图 2-2 为参考图例。

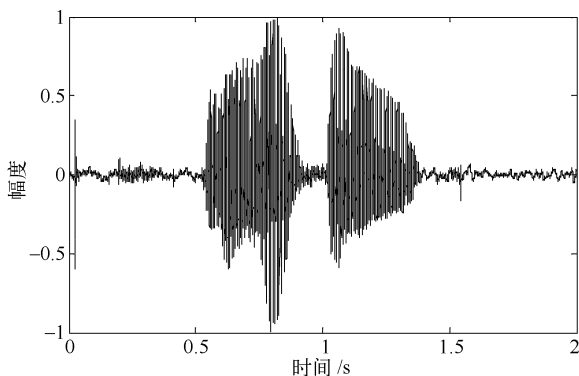


图 2-2 ‘你好，欢迎’ 时域图

3) 使用 `wavplay` 函数播放录制的语音信号，并改变播放的采样频率为原始采样频率的

倍数，体验效果。

2.1.4 思考题

1. 分析并解释实验要求3)的现象原理。

2. 自行录制一段语音，并存储为 wav 文件。要求：存储为 wav 文件时，分别以采样频率、2 倍采样频率和 1/2 采样频率存为三个 wav 文件，并将 plot 函数结合 subplot 函数在一幅图上显示 3 个波形。横轴和纵轴带有标注。横轴的单位为秒 (s)，纵轴显示的为归一化后的数值。

2.1.5 参考例程

```
% 语音采样与显示
fs = 8000; % 采样频率
duration = 2; % 时间长度
n = duration * fs;
t = (1:n)/fs;
fprintf( 'Begin by pressing any key %gseconds:\n' ,duration );pause
fprintf( 'recording... \n' );
y = wavrecord( n,fs,'double' );
ymax = max( abs( y ) ); % 归一化
y = y/ymax;
fprintf( 'Finish\n' );
fprintf( 'Press any key to play audio:\n' );pause
wavplay( y,fs );
wavwrite( y,fs,'original' );
wavwrite( y,fs/2,'halfsam' );
wavwrite( y,fs * 2,'doublesam' );
[ y1,fs1 ] = wavread( 'halfsam' );
t1 = (1:length( y1 ))/fs1;
[ y2,fs2 ] = wavread( 'doublesam' );
t2 = (1:length( y2 ))/fs2;
figure(1);
subplot(311)
axis([0 3 -1 1]);
plot( t,y );
xlabel( '时间/s' );
ylabel( '幅度' );
title( '(a)初始采样率' );
subplot(312)
axis([0 3 -1 1]);
plot( t1,y1 );
xlabel( '时间/s' );
ylabel( '幅度' );
```



```

title( (b)1/2 采样率 );
subplot(313)
axis([0 3 -1 1]);
plot(t2,y2);
xlabel( 时间/s );
ylabel( 幅度 );
title( (c)2 倍采样率 );

```

2.2 语音编辑实验

2.2.1 实验目的

- 1) 掌握语音信号线性叠加的方法，编写 MATLAB 程序实现非等长语音信号的叠加。
- 2) 熟悉语音信号卷积原理，编写 MATLAB 程序实现两语音卷积。
- 3) 熟悉语音信号升采样/降采样方法，并编写 MATLAB 程序实现。

2.2.2 实验原理

1. 信号的叠加

两个信号 x_1 和 x_2 ，通过短信号的补零使两语音信号有相同的长度，叠加信号为 $x_{new} = x_1 + x_2$ 。

实验中常通过生成随机信号的方法来叠加白噪声，随机信号的生成函数为 randn，其使用说明如下。

- 功能说明：
用于产生正态分布的随机信号。
- 函数语法：

```

Y = randn
Y = randn( n )
Y = randn( m, n )
Y = randn( [ m n ] )

```

- 参数解析：

$Y = \text{randn}$ ——返回一个伪随机数，其值来自于均值为 0，标准差为 1 的正态分布。

$Y = \text{randn}(n)$ ——返回一个 $n * n$ 的矩阵，其元素值如上描述。

$Y = \text{randn}(m, n)$ 或 $Y = \text{randn}([m \ n])$ ——返回一个 $m * n$ 的矩阵，元素值如上描述。

2. 信号的卷积

两序列 x_1 和 x_2 的卷积 y 定义为

$$y(n) = \sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k) = x_1(n) * x_2(n) \quad (2-1)$$

卷积运算满足交换律，即

$$y(n) = \sum_{k=-\infty}^{\infty} x_2(k)x_1(n-k) = x_2(n) * x_1(n) \quad (2-2)$$

注意：产生的序列 $y(n)$ 长度为 x_1 和 x_2 长度之和减 1。

MATLAB 中自带的卷积函数为 `conv`。

- 功能说明：

表示卷积和多项式乘法。

- 函数语法：

$$w = \text{conv}(u, v)$$

- 参数解析：

此函数将矢量 u 和 v 进行卷积运算。从代数上讲，卷积是与多项式相乘一致的操作。这些多项式的系数就是 u 和 v 的元素。

3. 信号采样频率的变换

采样率变换是多采样率信号处理的基础，主要由两个操作组成：抽取和内插。

抽取就是把原采样序列 $x(n)$ 每隔 $D-1$ 点取一个值，形成一个新的序列：

$$x_D(m) = x(mD) \quad (2-3)$$

其中， D 为正整数。为了避免抽取序列频谱的混叠，通常需要在抽取前将信号通过一个抗混叠滤波器。

内插器和抽取器作用相反，它在两个原始序列的样点之间插入 $I-1$ 个值。设原始序列为 $x(n)$ ，则内插后的序列 $x_I(m)$ 为：

$$x_I(m) = \begin{cases} x\left(\frac{m}{I}\right), & m = 0, \pm I, \pm 2I \dots \\ 0, & \text{others} \end{cases} \quad (2-4)$$

内插之后还要通过低通滤波器，抑制混叠信号。

MATLAB 中自带函数 `resample` 能实现采样率的变换。

- 功能说明：

对时间序列进行重采样。

- 函数语法：

$$y = \text{resample}(x, p, q)$$

$$y = \text{resample}(x, p, q, n)$$

$$y = \text{resample}(x, p, q, n, \text{beta})$$

$$y = \text{resample}(x, p, q, b)$$

$$[y, b] = \text{resample}(x, p, q)$$

- 参数解析：

$y = \text{resample}(x, p, q)$ ——采用多相滤波器对时间序列进行重采样，得到的序列 y 的长度为原来的序列 x 的长度的 p/q 倍， p 和 q 都为正整数。此时，默认采用使用 FIR 方法设计的抗混叠的低通滤波器。

$y = \text{resample}(x, p, q, n)$ ——采用 chebyshev IIR 型低通滤波器对时间序列进行重采样，滤波器的长度与 n 成比例， n 默认值为 10。

$y = \text{resample}(x, p, q, n, \text{beta})$ —— beta 为设置低通滤波器时使用 Kaiser 窗的参数，默认值为 5。

• 参考例程：

```
% 对简单的线性序列进行为原采样率 3/2 倍的重采样
fs1 = 10; % 初始采样率单位为 Hz
t1 = 0:1/fs1:1; % 时间向量
x = t1; % 定义一个线性序列
y = resample(x,3,2); % 进行重采样
t2 = (0:(length(y) - 1)) * 2/(3 * fs1); % 新的时间向量
plot(t1, x, 'k', t2, y, 'o', -0.5:0.01:1.5, -0.5:0.01:1.5, 'r');
legend('original', 'resampled'); xlabel('Time')
```

2.2.3 实验步骤

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

1) 录制或从 wav 文件中读取一段语音，并归一化。然后生成一段随机信号（长度与语音信号相同），归一化后幅度乘以 0.01。最后线性叠加两段语音，并用 plot 函数显示三种信号。要求：横轴和纵轴带有标注。横轴的单位为秒（s），纵轴显示的为归一化后的数值。图 2-3 为参考图例。

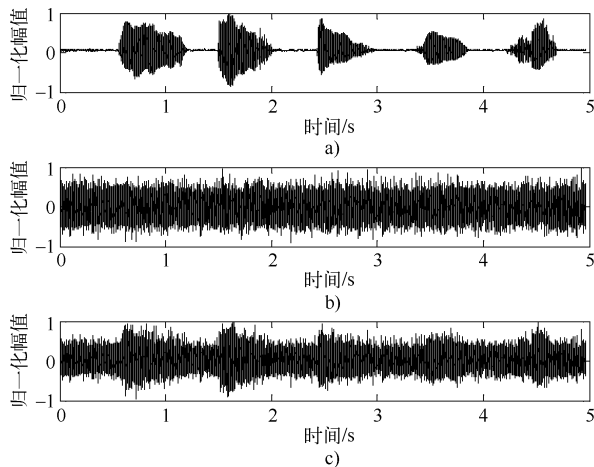


图 2-3 语音叠加示例

a) 原始信号 b) 随机序列 c) 线性叠加

2) 将录制或读取的语音信号与随机信号进行卷积，并用 plot 函数显示该信号，并对比线性叠加信号的区别。然后使用 wavplay 函数播放两种信号，并比较区别。图 2-4 为参考图例。

3) 改变录制或读取的语音信号的采样频率，使用 plot 函数进行显示，如图 2-5 所示。

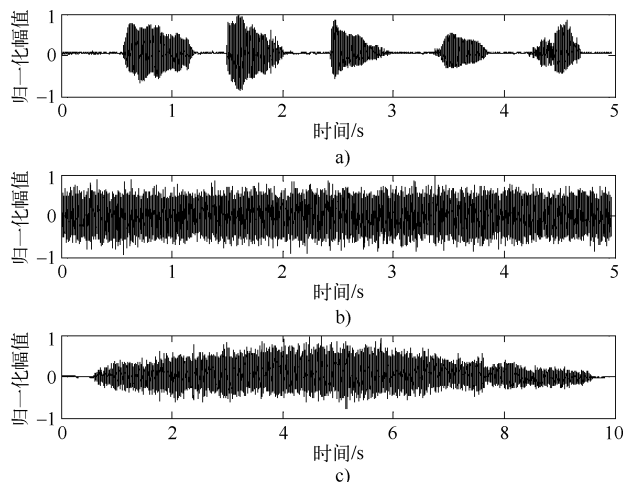


图 2-4 信号卷积示例

a) 原始信号 b) 随机序列 c) 信号卷积

然后采用 `wavplay` 函数播放，比较采样频率改变对语音信号的影响。

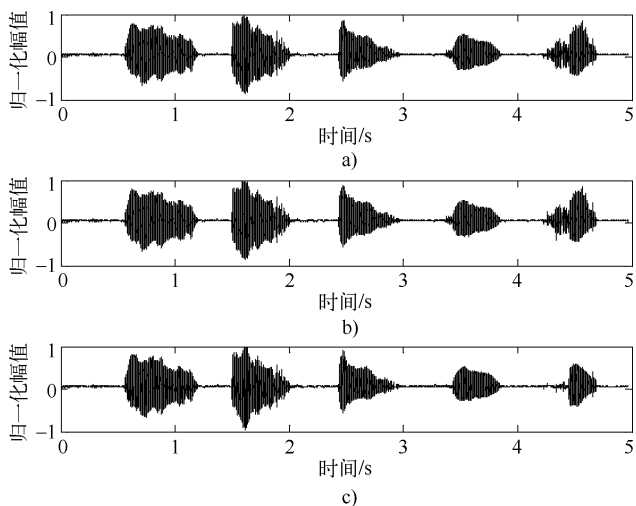


图 2-5 采样频率改变示例

a) 原始信号 b) 2 倍采样率 c) 1/2 采样率

2.2.4 思考题

1. 编写 MATLAB 函数实现任意长度的两个信号的线性叠加，生成信号的长度为两个叠加信号的最大长度。
2. 改变实验要求 1) 的随机信号幅度，重复实验并观察叠加信号的波形，感知语音质量，总结规律。
3. 自行编写 `conv` 函数，并编程同 MATLAB 函数进行比较。

2.2.5 参考例程

```
% 卷积函数
function a = my_conv(b,c)
bs = size(b);
cs = size(c);
i = any(bs - cs);
if i
    error(' error! ')
end
i = any( ~(bs - 1) );
if ~i
    error(' error! ')
end
ko = 0;
if bs(1) > bs(2)
    b = b' ;
    c = c' ;
    ko = 1;
end
bs = size(b);
cs = size(c);
ss = 2 * bs(2) - 1;
a = zeros(1,ss);
for i = 1 : cs(2)
    q = zeros(1,i - 1);
    p = zeros(1,ss - cs(2) + 1 - i);
    ba = [ q,c,p];
    ma = b(i) * ba;
    a = a + ma;
end
if ko
    a = a' ;
end
end
```

2.3 声强与响度实验

2.3.1 实验目的

- 1) 掌握语音声强的计算方法，并编写 MATLAB 程序实现。

- 2) 了解语音响度的含义, 编写 MATLAB 程序实现响度计算。
- 3) 了解等强度的概念, 并学会绘制等响度曲线。

2.3.2 实验原理

1. 声压与声强

(1) 声压

声压是定量描述声波的最基本的物理量, 它是由于声扰动产生的逾量压强, 是空间位置 and 时间的函数。由于声压的测量比较易于实现, 而且通过声压的测量也可以间接求得质点振速等其他声学参量, 因此, 声压已成为人们最为普遍采用的定量描述声波性质的物理量。

(2) 有效声压

通常讲的声压指的是有效声压, 即在一定时间间隔内将瞬时声压对时间求方均根值得。设语音长度为 T , 离散点数为 N , 则有效声压的计算公式为

$$p_e = \sqrt{\frac{1}{T} \sum_{n=1}^N x^2 \Delta t} = \sqrt{\frac{1}{N \Delta t} \sum_{n=1}^N x^2 \Delta t} = \sqrt{\frac{1}{N} \sum_{n=1}^N x^2} \quad (2-5)$$

其中, x 表示语音信号的采样点。只要保证所取的点数 N 足够大, 即可保证计算准确性。

(3) 声压级

声音的有效声压与基准声压之比, 取以 10 为底的对数, 再乘以 20, 即为声压级, 通常以符号 L_p 表示, 单位为 dB。

$$L_p = 20 \lg \frac{p_e}{p_{ref}} \text{ (dB)} \quad (2-6)$$

式中, p_e 为待测声压的有效值; p_{ref} 为参考声压, 在空气中参考声压一般取 2×10^{-5} Pa。

(4) 声强

在物理学中, 声波在单位时间内作用在与其传递方向垂直的单位面积上的能量称为声强。日常生活中能听到的声音其强度范围很大, 最大和最小之间可达 10^{12} 倍。

(5) 声强级

用声强的物理学单位表示声音强弱很不方便。当人耳听到两个强度不同的声音时, 感觉的大小大致上与两个声强比值的对数成比例。因此, 用对数尺度来表示声音强度的等级, 其单位为分贝 (dB)。

$$L_I = 10 \lg(I/I_0) \text{ (dB)} \quad (2-7)$$

在声学中用 1×10^{-12} W/m² 作为参考声强 (I_0)。

(6) 声压与声强的关系

对于球面波和平面波, 声压与声强的关系是:

$$I = P^2 / \rho \cdot c$$

式中, ρ 为空气密度; c 为声速。在标准大气压和 20℃ 的环境下, $\rho \cdot c = 408$ Pa · s/m。该数值为国际单位值, 也叫瑞利, 称为空气对声波的特性阻抗。

2. 响度

响度描述的是声音的响亮程度, 表示人耳对声音的主观感受, 其计量单位是宋。定义为声压级为 40 dB 的 1 kHz 纯音的响度为 1 Son (宋)。人耳对声音的感觉, 不仅和声压有关,

还和频率有关。声压级相同，频率不同的声音，听起来响亮程度也不同。如空压机与电锯，同是 100 dB 声压级的噪声，听起来电锯声要响得多。按人耳对声音的感觉特性，依据声压和频率定出人对声音的主观响度感觉量，称为响度级，单位为方，符号 phon。根据国际协议规定，0 dB 声级的 1000 Hz 纯音的响度级定义为 0 phon。其他频率声音的声级与响度级的对应关系，要从等响度曲线才能查出。

3. 等响度曲线

人耳对不同频率的纯音有不同的敏感度。1 kHz 的纯音在 0 dB HL 时就可被察觉；而 20Hz 的纯音则在 70 dB HL 时才可能被察觉。这种对不同频率的声音有不同的敏感也表现在听阈以上的情况。所以当不同频率的声音有同样响度的时候，它们的强度并不一定是一样的。这样就产生了等响度曲线，即把不同频率和不同强度的纯音和 1 kHz 的纯音做等响度的配对。

对于等响度曲线的研究，最早可追溯到 1927 年 Kingsbury 的工作，由于他是对单耳听觉条件下的等响度曲线进行的测量，因此受到了一定限制。虽然等响曲线的测量可以在自由声场、扩散声场或耳机听音情况下进行，但大多数发表的等响曲线都是在双耳听音或相对自由声场条件下得到的。1961 年，Robinson 和 Dadson 的研究成果被国际标准化组织所采纳，并被制定成 ISO/R266。2003 年，Suzuki 和 Takeshima 根据新近的研究数据对标准等响度曲线进行了重新修订，公布了 ISO226—2003 版等响度曲线。

根据 ISO226—2003 标准，等响度曲线的定义如下。

假设频率为 f 的纯音的响度级为 L_N ，则其声压级 L_p 为

$$L_p = \left(\frac{10}{\alpha_f} \cdot \lg A_f - L_U + 94 \right) (\text{dB}) \quad (2-8)$$

这里，

$$A_f = 4.47 \times 10^{-3} \times (10^{0.0025L_N} - 1.15) + [0.4 \times 10^{\left(\frac{T_f + L_U}{10} - 9\right)}] \alpha_f$$

T_f 为听力阈值； α_f 为响度感知指数； L_U 为以 1000Hz 为标准所计算的线性传输函数的幅值。参数的具体数值见表 2-5。

注：式 (2-4) 的适用范围为 20 ~ 80phon (5 ~ 12.5 kHz) 或 90phon (20 Hz ~ 4 kHz)。

表 2-5 等响度曲线参数表

频率/Hz	α_f	L_U/dB	T_f/dB
20	0.532	-31.6	78.5
25	0.506	-27.2	68.7
31.5	0.48	-23.0	59.5
40	0.455	-19.1	51.1
50	0.432	-15.9	44
63	0.409	-13.0	37.5
80	0.387	-10.3	31.5
100	0.367	-8.1	26.5

(续)

频率/Hz	α_f	L_U/dB	T_f/dB
125	0.349	-6.2	22.1
160	0.33	-4.5	17.9
200	0.315	-3.1	14.4
250	0.301	-2.0	11.4
315	0.288	-1.1	8.6
400	0.276	-0.4	6.2
500	0.267	0	4.4
630	0.259	0.3	3
800	0.253	0.5	2.2
1000	0.25	0	2.4
1250	0.246	-2.7	3.5
1600	0.244	-4.1	1.7
2000	0.243	-1.0	-1.3
2500	0.243	1.7	-4.2
3150	0.243	2.5	-6.0
4000	0.242	1.2	-5.4
5000	0.242	-2.1	-1.5
6300	0.245	-7.1	6
8000	0.254	-11.2	12.6
10000	0.271	-10.7	13.9
12500	0.301	-3.1	12.3

根据计算公式，可以绘制出等响度曲线。图 2-6 即为按照式 (2-8) 绘制的等响度曲线。

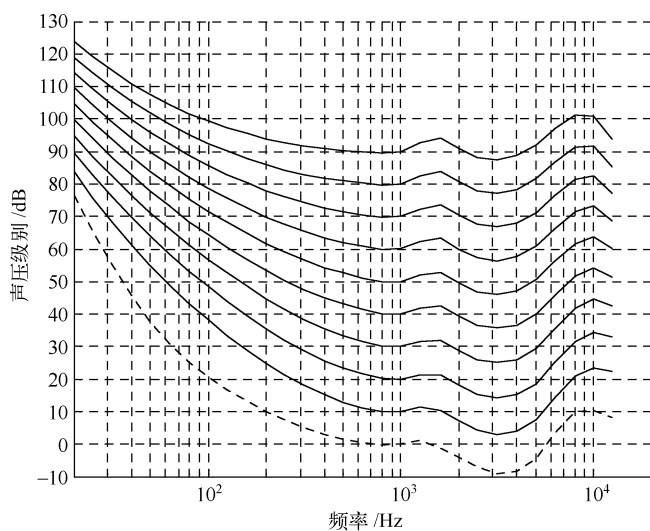


图 2-6 等响度曲线

曲线从低到高分别为 0phon 到 90phon 的响度曲线。从图上可知，当声压级在 80 dB 以上

时, 各个频率的声压级与响度级的数值就比较接近了, 这表明当声压级较高时, 人耳对各个频率的声音的感觉基本是一样的。

2.3.3 实验步骤及要求

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

(1) 编写计算有效声压的函数

录制或从 wav 文件中读取一段语音, 调用该函数计算该语音的有效声压 (要求每隔 30 ms 计算一次声压级), 并调用 plot 函数进行显示。要求: 横轴和纵轴带有标注。

函数定义如下:

函数格式:

$$\text{spl} = \text{SPLCal}(x, \text{fs}, \text{flen})$$

输入参数: x 为输入的语音信号, 这里用 x 的值直接替代瞬时声压的值, 理论上瞬时声压值跟 x 的值是成正比的, 后期会对其进行修正; fs 为采样率, 单位为 Hz; flen 为一帧信号的时间长度, 单位为 ms。

输出参数: spl 为函数输出的声压级值。

(2) 编写函数实现式 (2-4)

要求可以输入任意非负响度级时, 可得到该响度级对应的声压级曲线。然后仿照图 2-6 中的任意一条曲线, 使用 plot 函数完成曲线的显示。

函数定义如下:

函数格式:

$$[\text{spl}, \text{freq}] = \text{iso226}(\text{phon})$$

输入参数: phon 为输入的响度级。

输出参数: spl 为函数输出的声压级值, freq 是对应的频率值。

2.3.4 思考题

1. 查阅资料并解答, 实际环境中如何获得参考声压 p_{ref} 。
2. 修改有效声压计算的时间间隔, 并比较声压曲线的不同。
3. 编写 MATLAB 函数实现图 2-6 的绘制。

2.3.5 参考例程

```
function spl = SPLCal(x, fs, flen)
```

```
% SPLCal 函数用于计算一帧语音信号的声压级值
```

```
% 输入参数 x 为输入的语音信号
```

```
% 这里用 x 的值直接替代瞬时声压的值, 理论上瞬时声压值与 x 的值是成正比的
```

```

% 后期会对其进行修正
% 输入参数 fs 为采样率,单位为 Hz
% flen 为一帧信号的时间长度,单位为 ms
% spl 为函数输出的声压级值

% 输入的语音帧的长度
Length = length(x);
% 每一帧信号的离散点数
M = flen * fs/1000;
if Length ~ = M
    error( '输入信号长度与所定义帧长不等!' );
end
%% ----- 计算有效声压 -----
% 根据定义计算有效声压,pa = sqrt((x(1)^2 + x(2)^2 + ... + x(M)^2)/M)
% 单位为 Pa
pp = 0;
for i = 1:M
    pp = pp + x(i)^2;
end
pa = sqrt(pp/M);
%% ----- 计算声压级 -----
% 声压级值 spl = 20 * log10(pa/p0),单位为 dB
% 基准声压 p0,单位为 Pa
p0 = 2 * 10^-5;
spl = 20 * log10(pa/p0);
end

```

2.4 语音信号生成的数学模型

2.4.1 实验目的

- 1) 了解语音信号产生的基本原理。
- 2) 熟悉常见的语音信号产生模型。
- 3) 能编程实现语音信号产生模型,并分析其时频特性。

2.4.2 实验原理

从人类的发音器官的机理来看,发不同性质的声音时,声道的情况是不同的。此外,声门和声道的相互耦合会形成语音信号的非线性特性。但语音信号特性随着时间变化是很缓慢的,所以可以作出一些合理的假设,将语音信号分为一些相继的短段进行处理,在这些短段中可以认为语音信号特性是不随时间变化的平稳随机过程。通过对发音器官和语音产生机理

的分析, 语音生成系统理论上分成三个部分, 在声门 (声带) 以下, 称为“声门子系统”, 它负责产生激励振动, 是“激励系统”; 从声门到嘴唇的呼气通道是声道, 是“声道系统”; 语音从嘴唇辐射出去, 因此嘴唇以外部分称为“辐射系统”。

1. 激励模型

激励模型一般分成浊音激励和清音激励来讨论。发浊音时, 由于声带不断张开和关闭, 将产生间歇的脉冲波。这个脉冲波的波形类似于斜三角形的脉冲, 如图 2-7a 所示。它的数学表达式如下:

$$g(n) = \begin{cases} (1/2) [1 - \cos(\pi n/N_1)], & 0 \leq n \leq N_1 \\ \cos[\pi(n - N_1)/2N_2], & N_1 \leq n \leq N_1 + N_2 \\ 0, & \text{其他} \end{cases} \quad (2-9)$$

式中, N_1 为斜三角波上升部分的时间; N_2 为其下降部分的时间。单个斜三角波波形的频谱 $G(e^{j\omega})$ 的图形如图 2-7b 所示。由图可见, 这是一个低通滤波器, 其 z 变换的全极模型为

$$G(z) = \frac{1}{(1 - e^{-cT}z^{-1})^2} \quad (2-10)$$

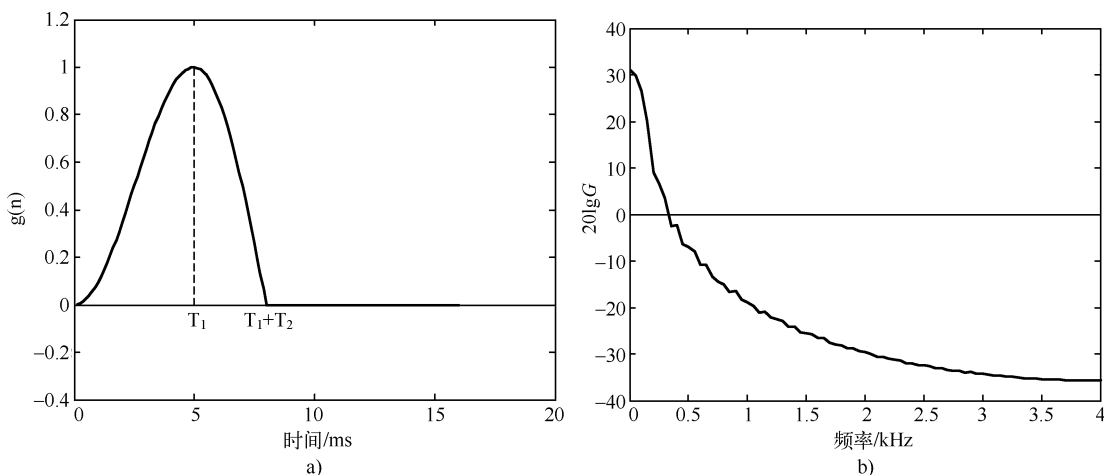


图 2-7 激励模型响应

a) 时域波形 b) 频域波形

这里, c 是一个频率常数。由于 $G(z)$ 是一个二极点的模型, 因此斜三角波形可视为加权的单位脉冲串激励上述单个斜三角波模型的结果。而单位脉冲串及幅值因子则可表示成下面的 z 变换形式:

$$E(z) = \frac{A_v}{1 - z^{-1}} \quad (2-11)$$

所以, 整个浊音激励模型可表示为

$$U(z) = G(z)E(z) = \frac{A_v}{1 - z^{-1}} \cdot \frac{1}{(1 - e^{-cT}z^{-1})^2} \quad (2-12)$$

即浊音激励波是一个以基音为周期的斜三角脉冲串。

发清音时, 无论是发阻塞音或摩擦音, 声道都被阻碍形成湍流。所以, 可把清音激励模

拟成随机白噪声。实际情况一般使用均值为0、方差为1，并在时间或/和幅值上为白色分布的序列。

2. 声道模型

共振峰模型把声道视为一个谐振腔，共振峰就是这个腔体的谐振频率。由于人耳听觉的柯替氏器官的纤毛细胞是按频率感受而排列其位置的，所以这种共振峰的声道模型方法是非常有效的。一般来说，一个元音用前三个共振峰来表示就足够了；而对于较复杂的辅音或鼻音，大概要用到前五个以上的共振峰才行。

从物理声学观点，可以很容易推导出均匀断面的声管的共振频率。一般成人的声道约为17cm长，因此算出其开口时的共振频率为

$$F_i = \frac{(2i-1)c}{4L} \quad (2-13)$$

这里， $i=1,2,\dots$ 为正整数，表示共振峰的序号； c 为声速； L 为声管长度。由此可算的元音 $e[\partial]$ 的前三共振峰为 $F_1=500\text{ Hz}$ ， $F_2=1500\text{ Hz}$ ， $F_3=2500\text{ Hz}$ 。另外，除了共振峰频率之外，共振峰模型还包括共振峰带宽和幅度等参数。

对于级联型的共振峰模型来说，声道可以视为一组串联的二阶谐振器。从共振峰理论来看，整个声道具有多个谐振频率和反谐振频率，所以可被模拟为一个包含零极点的数学模型；但对于一般元音，则用全极点模型即可，其传输函数可表示为

$$V(z) = \frac{G}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (2-14)$$

式中， N 是极点个数； G 是幅值因子； a_k 是常系数。此时可将它分解为多个二阶极点的网络的串联，即

$$V(z) = \prod_{i=1}^M \frac{a_i}{1 - b_i z^{-1} - c_i z^{-2}} \quad (2-15)$$

式中，

$$\begin{aligned} c_i &= -\exp(-2\pi B_i T) \\ b_i &= 2\exp(-\pi B_i T) \cos(2\pi F_i T) \\ a_i &= 1 - b_i - c_i, \\ G &= a_1 \cdot a_2 \cdot a_3 \cdots a_M \end{aligned} \quad (2-16)$$

并且 M 是小于 $(N+1)/2$ 的整数。若 z_k 是第 k 个极点，则有 $z_k = e^{-B_k T} e^{-2\pi F_k T}$ ， T 是取样周期。根据式(2-15)，则第一共振峰的二阶谐振器的幅频特性如图2-8所示。

3. 辐射模型

从声道模型输出的是速度波 $u_L(n)$ ，而语音信号是声压波 $p_L(n)$ ，二者之倒比称为辐射阻抗 Z_L 。其既表征口唇的辐射效应，又包括圆形的头部的绕射效应等。从理论上推导这个阻抗是有困难的，但是如果认为口唇展开的面积远小于头部的表面积，则可近似地看成平板开槽辐射的情况。此时，辐射阻抗的公式如下：

$$Z_L(\Omega) = \frac{j\Omega L_r R_r}{R_r + j\Omega L_r} \quad (2-17)$$

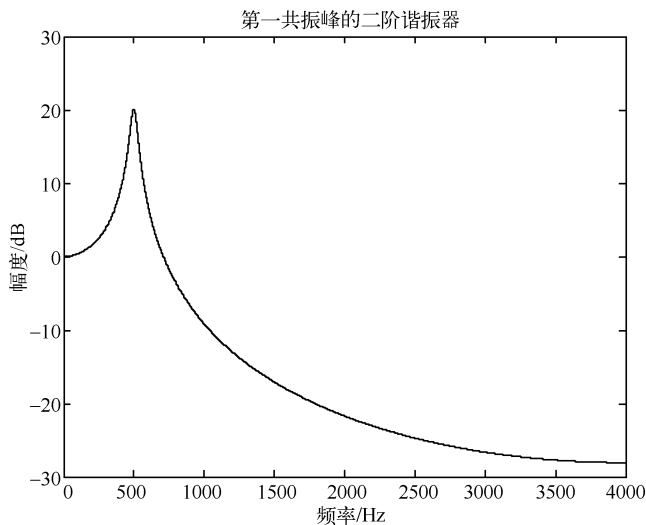


图 2-8 二阶谐振器

式中, $R_r = \frac{128}{9\pi^2}$, $L_r = \frac{8a}{3\pi c}$ 。这里, a 是口唇张开时的开口半径; c 是声波传播速度。

由辐射引起的能量损耗正比于辐射阻抗的实部, 所以辐射模型是一阶类高通滤波器。不考虑冲激脉冲串模型 $E(z)$, 斜三角波模型是二阶低通, 而辐射模型是一阶高通。因此, 在实际信号分析时, 常用所谓“预加重技术”, 即在取样之后, 插入一个一阶的高通滤波器, 从而只剩下声道部分, 便于声道参数分析。在语音合成时再进行“去加重”处理, 就可以恢复原来的语音。常用的预加重因子为 $[1 - (R(1)z^{-1}/R(0))]$ 。这里, $R(n)$ 是信号 $s(n)$ 的自相关函数。通常对于浊音来说, $R(1)/R(0) \approx 1$; 而对于清音来说, 该值可取得很小。

4. 语音信号的数字模型

综上所述, 完整的语音信号数字模型可用激励模型、声道模型和辐射模型的串联来表示, 如图 2-9 所示。传输函数 $H(z)$ 可表示为:

$$H(z) = A \cdot U(z)V(z)R(z) \quad (2-18)$$

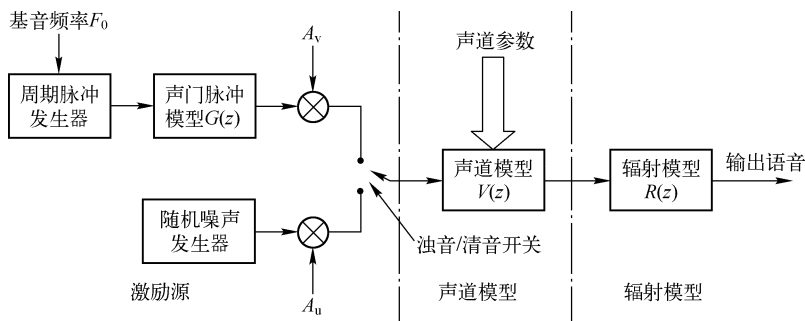


图 2-9 语音信号产生的数字模型

这里, $U(z)$ 是激励信号, 浊音时 $U(z)$ 是声门脉冲即斜三角形脉冲序列的 z 变换; 在清音的情况下, $U(z)$ 是一个随机噪声的 z 变换。 $V(z)$ 是声道传输函数, 既可用声管模型, 也可以共振峰模型等来描述。实际上就是全极点模型

$$V(z) = \frac{1}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (2-19)$$

而 $R(z)$ 可表示一阶高通的形式

$$R(z) = R_0(1 - z^{-1}) \quad (2-20)$$

应该指出, 式(2-14)所示模型的内部结构并不和语音产生的物理过程相一致, 但这种模型和真实模型在输出处是等效的。1) 这种模型是“短时”的模型, 因为一些语音信号的变化是缓慢的, 例如元音在 10 ~ 20 ms 内其参数可假定不变。而声道转移函数 $V(z)$ 是一个参数随时间缓慢变化的模型; 2) 这一模型认为语音是声门激励源激励线性系统——声道所产生的; 实际上, 声带—声道相互作用的非线性特征还有待研究; 3) 模型中用浊音和清音进行简单划分的方法是有缺陷的, 对于某些音是不适用的, 例如浊音当中的摩擦音。这种音要有发浊音和发清音的两种激励, 而且两者不是简单的叠加关系。对于这些音可用一些修正模型或更精确的模型来模拟。

2.4.3 实验步骤及要求

1. 实验步骤

运行 MATLAB → 新建 m 文件 → 编写 m 程序 → 编译并调试。

2. 实验要求

1) 编写激励模型函数。参考式(2-9) ~ (2-12), 编写激励模型函数, 实现时域和频域曲线。

2) 编写共振峰模型函数。根据实验原理介绍的共振峰频率组 ($F_1 = 500$ Hz、 $F_2 = 1500$ Hz、 $F_3 = 2500$ Hz), 参考级联型共振峰模型, 编写类似图 2-8 的二阶谐振器曲线。

2.4.4 思考题

已知英文字母 a' 的前三共振峰频率分别为 730 Hz、1090 Hz、2440 Hz, 试根据图 2-9 编写程序实现字母 a' 的发音。

2.4.5 参考例程

```
% 根据共振峰合成发音函数
function y = SpeechModelGen(len, pitch, sampleRate, f)
% SpeechModelGen(len, pitch, sampleRate, f) 用来生成元音信号
% 其中, len 为生成语音长度, pitch 是基音, sampleRate 是采样率, f 是共振峰频率数组

if nargin < 4; f = [730 1090 2440]; end; % 默认发音 a'
if nargin < 3; sampleRate = 16000; end;
if nargin < 3; pitch = 100; end;
f1 = f(1); f2 = f(2); f3 = f(3);
% 生成指定基音和采样率的声门脉冲, 长度为 len
y = zeros(1, len);
```

```
points = 1 : sampleRate/pitch : len;
indices = floor( points );

% 使用三角近似,并保持幅度不变
y( indices ) = ( indices + 1 ) - points;
y( indices + 1 ) = points - indices;

% 使用二阶滤波器(250Hz)模拟声门传输函数
a = exp( -250 * 2 * pi/sampleRate );
y = filter( [ 1 ], [ 1, 0, -a * a ], y );
% y = filter( [ 1 ], [ 1, -2 * a - 1, a * a + 2 * a, -a * a ], y );

% 根据指定的共振峰频率和带宽(50 Hz)建模语音信号中的共振峰
% 第一共振峰
if f1 > 0
    cft = f1/sampleRate;
    bw = 50;
    q = f1/bw;
    rho = exp( -pi * cft / q );
    theta = 2 * pi * cft * sqrt( 1-1/(4 * q * q) );
    a2 = -2 * rho * cos( theta );
    a3 = rho * rho;
    y = filter( [ 1 + a2 + a3 ], [ 1, a2, a3 ], y );
end;

% 根据指定的共振峰频率和带宽(50 Hz)建模语音信号中的共振峰
% 第二共振峰
if f2 > 0
    cft = f2/sampleRate;
    bw = 50;
    q = f2/bw;
    rho = exp( -pi * cft / q );
    theta = 2 * pi * cft * sqrt( 1-1/(4 * q * q) );
    a2 = -2 * rho * cos( theta );
    a3 = rho * rho;
    y = filter( [ 1 + a2 + a3 ], [ 1, a2, a3 ], y );
end;

% 根据指定的共振峰频率和带宽(50 Hz)建模语音信号中的共振峰
% 第三共振峰
if f3 > 0
    cft = f3/sampleRate;
    bw = 50;
```

```

q = f3/bw;
rho = exp(-pi * cft / q);
theta = 2 * pi * cft * sqrt(1-1/(4 * q * q));
a2 = -2 * rho * cos(theta);
a3 = rho * rho;
y = filter([1 + a2 + a3],[1, a2, a3],y);
end;

```

2.5 语音信号的预处理

2.5.1 实验目的

- 1) 掌握消除趋势项、直流项的方法。
- 2) 掌握语音信号预加重的原理及方法。
- 3) 掌握语音信号预滤波的目的和方法。
- 4) 编程实现上述方法，并测试。

2.5.2 实验原理

1. 消除趋势项和直流分量

在采集语音信号数据的过程中，由于测试系统的某些原因在时间序列中会产生一个线性的或者慢变的趋势误差，例如放大器随温度变化产生的零漂移，传声器低频性能的不稳定或传声器周围的环境干扰，总之使语音信号的零线偏离基线，甚至偏离基线的大小还会随时间变化。零线随时间偏离基线被称为信号的趋势项。趋势项误差的存在，会使相关函数、功率谱函数在处理计算中出现变形，甚至可能使低频段的谱估计完全失去真实性和正确性，所以应该将其去除。数据采集仪中信号放大器受环境温度、湿度、电磁场、噪声、振动冲击等外部环境的干扰，采集到的振动信号往往与真实信号发生了偏离，使信号信噪比降低、甚至信号完全失真。一般情况下测量被测物体的加速度比测量位移和速度方便得多。但由于信号中含有长周期趋势项，在对数据进行二次积分时得到的结果可能完全失真，因此消除长周期趋势项是振动信号预处理的一项重要任务。直流分量的消除比较简单，即减去语音信号的平均项即可。而对于线性趋势项或多项式趋势项，常用的消除趋势项的方法是用多项式最小二乘法。

设实测语音信号的采样数据为 $\{x_k\} (k=1, 2, 3, \dots, n)$ ， n 为样本总数，由于采样数据是等时间间隔的，为简化起见，令采样时间间隔 $\Delta t=1$ 。用一个多项式函数 \hat{x}_k 表示语音信号中的趋势项：

$$\hat{x}_k = a_0 + a_1 k + a_2 k^2 + \dots + a_m k^m = \sum_{j=0}^m a_j k^j (k \in [1, n]) \quad (2-21)$$

为了确定各待定系数 a_j ，令函数 \hat{x}_k 与离散数据 x_k 的误差二次方和为最小，即

$$E = \sum_{k=1}^n (\hat{x}_k - x_k)^2 = \sum_{k=1}^n \left(\sum_{j=0}^m a_j k^j - x_k \right)^2 \quad (2-22)$$

对 E 求偏导, 得

$$\frac{\partial E}{\partial a_i} = 2 \sum_{k=1}^n k^i \left(\sum_{j=0}^m a_j k^j - x_k \right) = 0 \quad i \in [0, m] \quad (2-23)$$

依次对 a_i 求偏导, 可得 $m+1$ 元线性方程组

$$\sum_{k=1}^n \sum_{j=0}^m a_j k^{j+i} - \sum_{k=1}^n x_k k^i = 0 \quad i \in [0, m] \quad (2-24)$$

通过解方程组求出 $m+1$ 个待定系数 a_j 。各式中, m 为设定的多项式阶次。

当 $m=0$ 时求得的趋势项为常数, 有

$$\sum_{k=1}^n a_0 k^0 - \sum_{k=1}^n x_k k^0 = 0 \quad (2-25)$$

解方程得

$$a_0 = \frac{1}{n} \sum_{k=1}^n x_k \quad (2-26)$$

由此可知, 当 $m=0$ 时的趋势项为信号采样数据的算术平均值, 即是直流分量。消除常数趋势项的计算公式为

$$y_k = x_k - \hat{x}_k = x_k - a_0 \quad (2-27)$$

当 $m=1$ 时为线性趋势项, 有

$$\begin{cases} \sum_{k=1}^n a_0 k^0 + \sum_{k=1}^n a_1 k - \sum_{k=1}^n x_k k^0 = 0 \\ \sum_{k=1}^n a_0 k + \sum_{k=1}^n a_1 k^2 - \sum_{k=1}^n x_k k = 0 \end{cases} \quad (2-28)$$

解方程组得

$$\begin{cases} a_0 = \frac{2(2n+1) \sum_{k=1}^n x_k - 6 \sum_{k=1}^n x_k k}{n(n-1)} \\ a_1 = \frac{12 \sum_{k=1}^n x_k k - 6(n-1) \sum_{k=1}^n x_k}{n(n-1)(n+1)} \end{cases} \quad (2-29)$$

消除线性趋势项的计算公式为

$$y_k = x_k - \hat{x}_k = x_k - (a_0 + a_1 k) \quad (2-30)$$

当 $m \geq 2$ 时为曲线趋势项。在实际语音信号数据处理中, 通常取 $m=1 \sim 3$ 来对采样数据进行多项式趋势项消除的处理。

在 MATLAB 里自带有消除线性趋势项的函数 `detrend`。

函数格式:

$$y = \text{detrend}(x)$$

输入参数: x 是带有线性趋势项的信号序列。

输出参数: y 是消除趋势项的序列。

2. 数字滤波器

在采集语音信号时, 交流隔离不好常会将工频 50 Hz 的交流声混入到语音信号中, 因此

需要采用低通滤波器滤除工频干扰；此外，由于基音的频率较低，通常位于 60 ~ 450 Hz 之间。因此，在基音提取算法中，为了抗干扰，常设计低通滤波器来提取低频段信号。

常用的经典 IIR 数字滤波器包含巴特沃斯滤波器、切比雪夫 I 型滤波器、切比雪夫 II 型滤波器和椭圆滤波器四类。基于 MATLAB 的数字滤波器设计步骤如下。

(1) 根据设计指标确定滤波器参数

滤波器的设计指标包括：通带截止频率 W_p 和阻带截止频率 W_s ，其取值范围为 0 至 1 之间，当其值为 1 时代表采样频率的一半。通带和阻带区的波纹系数分别是 R_p 和 R_s 。

不同类型（高通、低通、带通和带阻）滤波器对应的 W_p 和 W_s 值遵循以下规则：

- 高通滤波器： W_p 和 W_s 为一元矢量且 $W_p > W_s$ 。
- 低通滤波器： W_p 和 W_s 为一元矢量且 $W_p < W_s$ 。
- 带通滤波器： W_p 和 W_s 为二元矢量且 $W_p < W_s$ ，如 $W_p = [0.2, 0.7]$ ， $W_s = [0.1, 0.8]$ 。
- 带阻滤波器： W_p 和 W_s 为二元矢量且 $W_p > W_s$ ，如 $W_p = [0.1, 0.8]$ ， $W_s = [0.2, 0.7]$ 。

常用的滤波器函数为：

- 巴特沃斯滤波器： $[n, Wn] = \text{buttord}(W_p, W_s, R_p, R_s)$ 。
- 切比雪夫 I 型滤波器： $[n, Wn] = \text{cheb1ord}(W_p, W_s, R_p, R_s)$ 。
- 切比雪夫 II 型滤波器： $[n, Wn] = \text{cheb2ord}(W_p, W_s, R_p, R_s)$ 。
- 椭圆滤波器： $[n, Wn] = \text{ellipord}(W_p, W_s, R_p, R_s)$ 。

其中， n 代表滤波器阶数； Wn 为滤波器的截止频率（无论高通、带通、带阻滤波器在设计中最终都等效于一个低通滤波器）。

(2) 采用 MATLAB 函数设计数字滤波器

数字滤波器设计函数包括：

- 巴特沃斯滤波器： $[n, Wn] = \text{butter}(n, Wn', \text{ftype}')$ 。
- 切比雪夫 I 型滤波器（通带等波纹）： $[n, Wn] = \text{cheby1}(n, R_p, Wn', \text{ftype}')$ 。
- 切比雪夫 II 型滤波器： $[n, Wn] = \text{cheby2}(n, R_s, Wn', \text{ftype}')$ 。
- 椭圆滤波器（阻带等波纹）： $[n, Wn] = \text{ellip}(n, R_p, R_s, Wn', \text{ftype}')$ 。

这里， ftype 的取值包括 `high`，`low`，`stop`，分别指代高通、低通和带阻（通）。

参考例程：设计一个椭圆带通滤波器，通带衰减 3 dB，阻带衰减 80 dB，通带频率为 [45 55] Hz，阻带频率为 [40 60] Hz，并画出滤波器的幅频曲线和相频曲线。

参考程序：

```
% 数字滤波器设计
clc
clear all
fs = 200;
% 把截止频率转成弧度表示
wp = [45 55] * 2/fs;
ws = [40 60] * 2/fs;
rp = 3;
```

```

rs = 80;
Nn = 512;
[n, wn] = ellipord(wp, ws, rp, rs);
[b, a] = ellip(n, rp, rs, wn);
freqz(b, a, Nn, fs);

```

幅频曲线和相频曲线如图 2-10 所示。

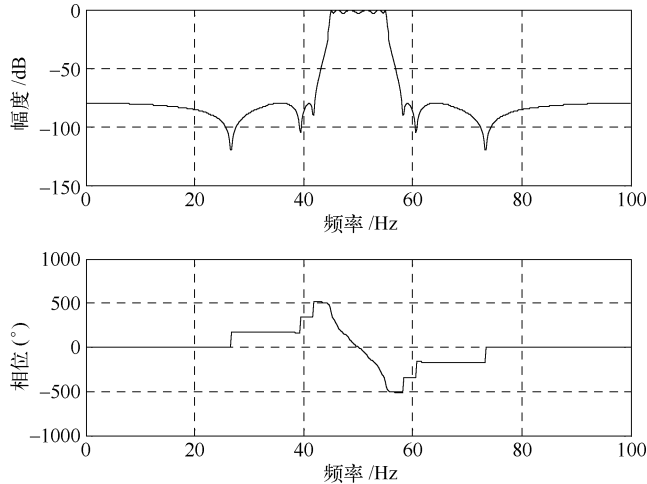


图 2-10 滤波器的幅频曲线和相频曲线

3. 预加重与去加重

语音和图像信号低频段能量大，高频段信号能量明显小；而鉴频器输出噪声的功率谱密度随频率的平方而增加（低频噪声小，高频噪声大），造成信号的低频信噪比很大，而高频信噪比明显不足，使高频传输困难。调频收发技术中，通常采用预加重和去加重技术来解决这一问题。预加重是发送端对输入信号高频分量的提升；去加重是解调后对高频分量的压低。

对于语言和音乐来说，其功率谱随频率的增加而减小，其大部分能量集中在低频范围内，这就造成消息信号高频端的信噪比可能降到不能容许的程度。但是由于消息信号中较高频率分量的能量小，很少有足以产生最大频偏的幅度，因此产生最大频偏的信号幅度多数是由信号的低频分量引起的。平均来说，幅度较小的高频分量产生的频偏小得多。因为调频系统的传输带宽是由需要传送的消息信号（调制信号）的最高有效频率和最大频偏决定的，所以调频信号并没有充分占用给予它的带宽。然而，接收端输入的噪声频谱却占据了整个调频带宽，也就是说，鉴频器输出端的噪声功率谱在较高频率上已被加重了。

为了抵消这种不希望有的现象，在调频系统中普遍采用一种叫作预加重和去加重的措施，其中心思想是利用信号特性和噪声特性的差别来有效地对信号进行处理。在噪声引入之前采用适当的网络（预加重网络），人为地加重（提升）发射机输入调制信号的高频分量。然后在接收机鉴频器的输出端，再进行相反的处理，即采用去加重网络把高频分量去加重，恢复原来的信号功率分布。在去加重过程中，同时也减小了噪声的高频分量，但是预加重对噪声并没有影响，因此有效地提高了输出信噪比。很多信号处理都使用这个方法，对高频分量电平提升（预加重），然后记录（调制、传输），播放（解调）时对高频分量衰减（去加重）。录音带系统中

的杜比系统是个典型的例子。假设信号高频分量为 10，经记录后，再播放时，引入的磁带本底噪声为 1，那么还原出来信号高频段信噪比为 10:1；如果在记录前对信号的高频分量提升，假设提升为 20，经记录后再播放时，引入的磁带本底噪声为 1，此时依然是 10:1 的信噪比，但是此时的高频分量是被提升了的，在对高频分量进行衰减的同时，磁带本底噪声也被衰减，如果将信号高频分量衰减还原到原来的 10，则本底噪声就会被降低到 0.5。

常用的所谓“预加重技术”是在取样之后，插入一个一阶的高通滤波器，具体效果如图 2-11 所示。常用的预加重因子为 $[1-R(1)z^{-1}/R(0)]$ 。这里， $R(n)$ 是信号 $s(n)$ 的自相关函数。通常对于浊音， $R(1)/R(0) \approx 1$ ；而对于清音，则该值可取得很小。在语音播放时再进行“去加重”处理，即预加重的反处理。

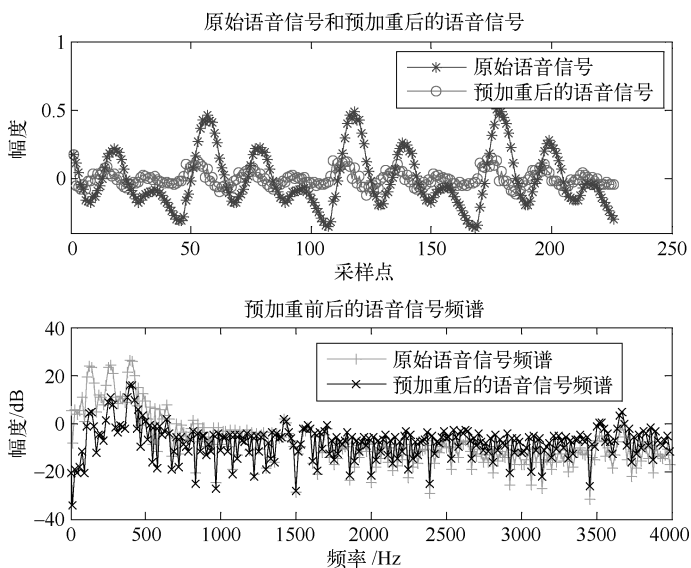


图 2-11 语音信号处理预加重效果图

2.5.3 实验步骤及要求

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

(1) 测试 MATLAB 的去线性趋势项函数 `detrend` 的作用

根据消除多项式趋势项的原理，编写消除多项式趋势项的函数 `detrendN`，并仿真测试，参考语音信号 `C2_5_y_1.wav`。

函数定义如下：

函数格式：

$$y = \text{detrendN}(x, fs, m)$$

输入参数： x 是带有趋势项的信号序列； fs 是采样频率； m 是多项式的阶次。

输出参数: y 是消除趋势项的序列。

仿真效果如图 2-12 所示。

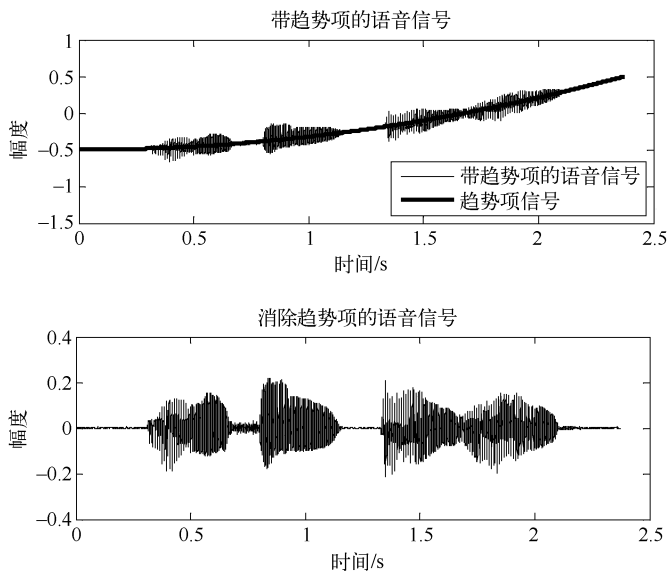


图 2-12 消除趋势项效果图

(2) 设计低通滤波器

参考例程, 设计一个低通滤波器, 并绘出其幅频和相频曲线。设计指标为: $f_s = 8000$ Hz, $W_p = 60$ Hz, $W_s = 50$ Hz, $R_p = 3$ dB, $R_s = 80$ dB。

(3) 编写函数

编写预加重函数实现类似图 2-12 的效果, 参考语音信号 C2_5_y_3. wav。

2.5.4 思考题

- 1) 使用实验要求 2) 设计的低通滤波器来滤去语音信号中的低频成分, 并用 FFT 分析其频谱变化情况。
- 2) 编写去加重函数, 并绘制如图 2-12 所示的图例来说明其效果。

2.5.5 参考例程

```
% 消除多项式趋势项函数
function [y,xtrend] = detrendN(x,fs,m)
x = x(:); % 把语音信号 x 转换为列数据
N = length(x); % 求出 x 的长度
t = (0:N-1)/fs; % 按 x 的长度和采样频率设置时间序列
a = polyfit(t,x,m); % 用最小二乘法拟合语音信号 x 的多项式系数 a
xtrend = polyval(a,t); % 用系数 a 和时间序列 t 构成趋势项
y = x - xtrend; % 从语音信号 x 中清除趋势项
```

第3章 语音信号分析实验

3.1 语音分帧与加窗

3.1.1 实验目的

- 1) 了解语音信号分帧与加窗的重要性和必要性。
- 2) 掌握常用的窗函数和加窗分帧处理的原理。
- 3) 能编程实现分帧函数，并恢复。

3.1.2 实验原理

1. 语音分帧

贯穿于语音分析全过程的是“短时分析技术”。因为，语音信号从整体来看其特性及表征其本质特征参数均是随时间而变化的，所以它是一个非平稳态过程，不能用处理平稳信号的数字信号处理技术对其进行分析处理。但是，由于不同的语音是由人的口腔肌肉运动构成声道某种形状而产生的响应，而这种口腔肌肉运动相对于语音频率来说是非常缓慢的，所以从另一方面看，虽然语音信号具有时变特性，但是在一个短时间范围内（一般认为在10~30ms的短时间内），其特性基本保持不变即相对稳定，因而可以将其看作是一个准稳态过程，即语音信号具有短时平稳性。所以任何语音信号的分析处理必须建立在“短时”的基础上，即进行“短时分析”，将语音信号分为一段一段来分析其特征参数，其中每一段称为一“帧”，帧长一般即取为10~30ms。这样，对于整体的语音信号来讲，分析出的是由每一帧特征参数组成的特征参数时间序列。

分帧示意图如图3-1所示。一般每秒的帧数约为33~100帧，视实际情况而定。分帧虽然可以采用连续分段的方法，但一般要采用如图3-1所示的交叠分段的方法，这是为了使帧与帧之间平滑过渡，保持其连续性。前一帧和后一帧的交叠部分称为帧移。帧移与帧长的比值一般取为0~1/2。

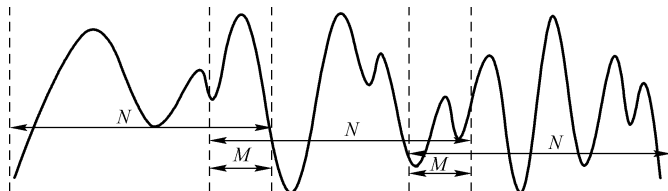


图3-1 分帧示意图 (N 为帧长, M 为帧移)

2. 窗函数

分帧是用可移动的有限长度窗口进行加权的方法来实现的，这就是用一定的窗函数

$w(n)$ 来乘 $s(n)$ ，从而形成加窗语音信号 $s_w(n) = s(n) * w(n)$ 。窗函数 $w(n)$ 的选择（形状和长度），对于短时分析参数的特性影响很大。为此应选择合适的窗口，使其短时参数更好地反映语音信号的特性变化。选择的依据有两类：

1) 窗口的形状。一个好的窗函数的标准是：在时域因为是语音波形乘以窗函数，所以要减小时间窗两端的坡度，使窗口边缘两端不引起急剧变化而平滑过渡到零，这样可以使截取出的语音波形缓慢降为零，减小语音帧的截断效应；在频域要有较宽的 3 dB 带宽以及较小的边带最大值。

2) 窗口的长度。如果长度很大，则它等效于很窄的低通滤波器，语音信号通过时，反映波形细节的高频部分被阻碍，短时能量随时间变化很小，不能真实地反映语音信号的幅度变化；反之，长度太小时，滤波器的通带变宽，短时能量随时间有急剧的变化，不能得到平滑的能量函数。通常认为在一个语音帧内应包含 1 ~ 7 个基音周期。然而不同人的基音周期变化很大，从女性和儿童的 2 ms 到老年男子的 14 ms（即基音频率的变化范围为 500 ~ 70 Hz），所以 N 的选择比较困难。通常在 10 kHz 取样频率下， N 折中选择为 100 ~ 200 点为宜（即 10 ~ 20 ms 持续时间）。

在语音信号数字处理中常用的窗函数有三种：

1) 矩形窗：

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases} \quad (3-1)$$

2) 汉明窗：

$$w(n) = \begin{cases} 0.54 - 0.46 \cos[2\pi n / (N-1)], & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases} \quad (3-2)$$

3) 海宁窗：

$$w(n) = \begin{cases} 0.5(1 - \cos(2\pi n / (N-1))), & 0 \leq n \leq N-1 \\ 0, & \text{其他} \end{cases} \quad (3-3)$$

三种窗函数对应的 MATLAB 函数为 `boxcar(N)`、`hanning(N)` 和 `hamming(N)`。其中， N 代表窗口长度。下面以矩形窗为例，编程观察其时域波形，如图 3-2 所示。

参考例程如下：

```
clc
clear all
close all
N = 32; nn = 0:(N-1);
subplot(311);
w = boxcar(N); stem(nn, w)
xlabel( '点数' ); ylabel( '幅度' ); title( ' (a) 矩形窗 ' )
subplot(312);
w = hamming(N); stem(nn, w)
xlabel( '点数' ); ylabel( '幅度' ); title( ' (b) 汉明窗 ' )
subplot(313);
w = hanning(N); stem(nn, w)
xlabel( '点数' ); ylabel( '幅度' ); title( ' (c) 海宁窗 ' )
```

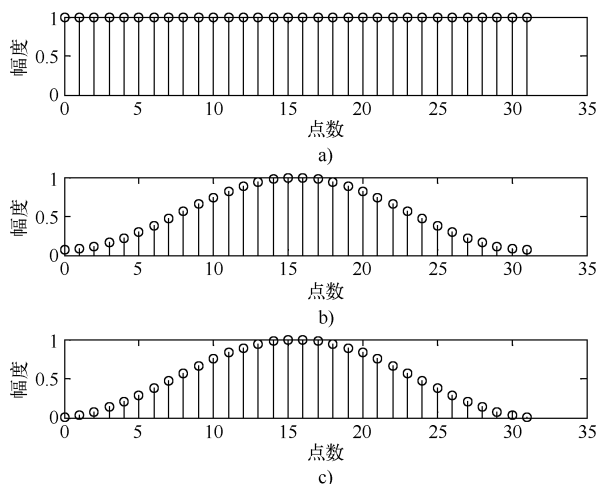


图 3-2 窗函数时域波形

a) 矩形窗 b) 汉明窗 c) 汉宁窗

3.1.3 实验步骤及要求

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

1) 根据语音分帧的思想, 编写分帧函数。函数定义如下:
函数格式:

```
frameout = enframe(x, win, inc)
```

输入参数: x 是语音信号; win 是帧长或窗函数, 若为窗函数, 帧长便取窗函数长; inc 是帧移。

输出参数: $frameout$ 是分帧后的数组, 长度为帧长和帧数的乘积。

根据分帧后的语音, 绘制连续四帧语音信号 (不用窗函数), 效果如图 3-3 所示。

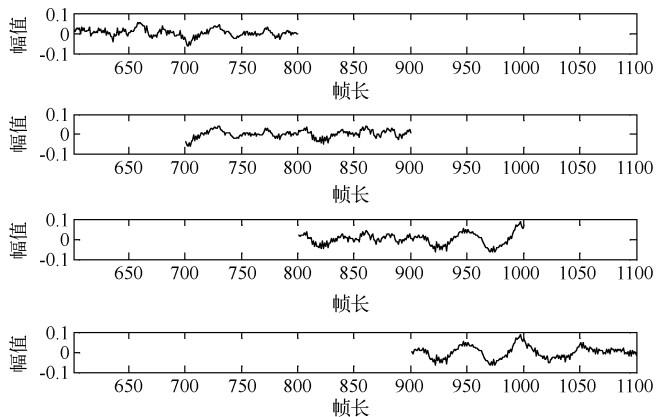


图 3-3 连续四帧语音信号

2) 编程实现矩形窗、汉明窗和汉宁窗,效果如图 3-2 所示。

3.1.4 思考题

编程实现将分帧加窗后的语音信号恢复成原始分帧前的信号。

3.1.5 参考例程

```
% 分帧函数
function frameout = enframe(x, win, inc)

nx = length(x(:)); % 取数据长度
nwin = length(win); % 取窗长
if (nwin == 1) % 判断窗长是否为 1, 若为 1, 即表示没有设窗函数
    len = win; % 是, 帧长 = win
else
    len = nwin; % 否, 帧长 = 窗长
end
if (nargin < 3) % 如果只有两个参数, 设帧 inc = 帧长
    inc = len;
end
nf = fix((nx - len + inc)/inc); % 计算帧数
frameout = zeros(nf, len); % 初始化
indf = inc * (0:(nf - 1)).'; % 设置每帧在 x 中的位移量位置
inds = (1:len); % 每帧数据对应 1:len
frameout(:) = x(indf(:, ones(1, len)) + inds(ones(nf, 1), :)); % 对数据分帧
if (nwin > 1) % 若参数中包括窗函数, 把每帧乘以窗函数
    w = win(:); % 把 win 转成行数据
    frameout = frameout .* w(ones(nf, 1), :); % 乘窗函数
end
```

3.2 短时域分析

3.2.1 实验目的

- 1) 了解语音信号短时域分析的原理。
- 2) 掌握短时域分析的一些参数计算方法。
- 3) 根据原理能编程实现短时域分析的参数计算。

3.2.2 实验原理

语音信号的时域分析就是分析和提取语音信号的时域参数。语音信号本身就是时域信号,因而时域分析是最早使用,也是应用最广泛的一种分析方法,这种方法直接利用语音信

号的时域波形。时域分析通常用于最基本的参数分析及应用,如语音的分割、预处理、分类等。语音信号的时域参数有短时能量、短时过零率、短时自相关函数和短时平均幅度差函数等。这些最基本的短时参数在各种语音信号数字处理技术中都有重要的应用。

1. 短时能量与短时平均幅度

设第 n 帧语音信号 $x_n(m)$ 的短时能量用 E_n 表示, 则其计算公式如下:

$$E_n = \sum_{m=0}^{N-1} x_n^2(m) \quad (3-4)$$

E_n 是一个度量语音信号幅度值变化的函数, 但它有一个缺陷, 即它对高电平非常敏感 (因为它计算时用的是信号的平方)。为此, 可采用另一个度量语音信号幅度值变化的函数, 即短时平均幅度函数 M_n , 它定义为

$$M_n = \sum_{m=0}^{N-1} |x_n(m)| \quad (3-5)$$

M_n 也是一帧语音信号能量大小的表征, 它与 E_n 的区别在于计算时小取样值和大取样值不会因取平方而造成较大差异, 在某些应用领域中会带来一些好处。

2. 短时过零率

短时过零率表示一帧语音中语音信号波形穿过横轴 (零电平) 的次数。对于连续语音信号, 过零即意味着时域波形通过时间轴; 而对于离散信号, 如果相邻的取样值改变符号则称为过零。因此, 过零率就是样本改变符号的次数。

定义语音信号 $x_n(m)$ 的短时过零率 Z_n 为

$$Z_n = \frac{1}{2} \sum_{m=0}^{N-1} |\text{sgn}[x_n(m)] - \text{sgn}[x_n(m-1)]| \quad (3-6)$$

式中, $\text{sgn}[\cdot]$ 是符号函数, 即

$$\text{sgn}[x] = \begin{cases} 1, & (x \geq 0) \\ -1, & (x < 0) \end{cases} \quad (3-7)$$

在实际中求过零率参数时, 需要注意的一个问题是, 如果输入信号中包含有 50 Hz 的工频干扰或者 A/D 转换器的工作点有偏移 (这等效于输入信号有直流偏移), 往往会使计算的过零率参数很不准确。为了解决前一个问题, A/D 转换器前的防混叠带通滤波器的低端截止频应高于 50 Hz, 以有效地抑制电源干扰。对于后一个问题除了可以采用低直流漂移器件外, 也可以在软件上加以解决, 这就是算出每一帧的直流分量并予以滤除。

这里应注意, 在 MATLAB 编程中, 实际上求短时平均过零率并不是按照上述公式计算, 而是使用了另外一种方法。按上述过零的描述, 即离散信号相邻的取样值改变符号, 那它们的乘积一定为负数, 即

$$x_i(m) * x_i(m+1) < 0 \quad (3-8)$$

3. 短时自相关

自相关函数具有一些性质 (如它是偶函数; 假设序列具有周期性, 则其自相关函数也是同周期的周期函数等)。对于浊音语音可以用自相关函数求出语音波形序列的基音周期。此外, 在进行语音信号的线性预测分析时, 也要用到自相关函数。

语音信号 $x_n(m)$ 的短时自相关函数 $R_n(k)$ 的计算式如下:

$$R_n(k) = \sum_{m=0}^{N-1-k} x_n(m)x_n(m+k) \quad (0 \leq k \leq K) \quad (3-9)$$

这里 K 是最大的延迟点数。

短时自相关函数具有以下性质：

- 1) 如果 $x_n(m)$ 是周期的（设周期为 N_p ），则自相关函数是同周期的周期函数，即 $R_n(k) = R_n(k + N_p)$ 。
- 2) $R_n(k)$ 是偶函数，即 $R_n(k) = R_n(-k)$ 。
- 3) 当 $k=0$ 时，自相关函数具有最大值，即 $R_n(0) \geq |R_n(k)|$ ，并且 $R_n(0)$ 等于确定性信号序列的能量或随机性序列的平均功率。

4. 短时平均幅度差

短时自相关函数是语音信号时域分析的重要参量。但是，计算自相关函数的运算量很大，其原因是乘法运算所需要的时间较长。利用快速傅里叶变换（FFT）等简化计算方法都无法避免乘法运算。为了避免乘法，一个简单的方法就是利用差值。为此常常采用另一种与自相关函数有类似作用的参量，即短时平均幅度差函数。

平均幅度差函数能够代替自相关函数进行语音分析的原因在于：如果信号是完全的周期信号（设周期为 N_p ），则相距为周期的整数倍的样点上的幅值是相等的，差值为零。即：

$$d(n) = x(n) - x(n+k) = 0 \quad (k=0, \pm N_p, \pm 2N_p, \dots) \quad (3-10)$$

对于实际的语音信号， $d(n)$ 虽不为零，但其值很小。这些极小值将出现在整数倍周期的位置上。为此，可定义短时平均幅度差函数：

$$F_n(k) = \sum_{m=0}^{N-1-k} |x_n(m) - x_n(m+k)| \quad (3-11)$$

显然，如果 $x(n)$ 在窗口取值范围内具有周期性，则 $F_n(k)$ 在 $k = N_p, 2N_p, \dots$ 时将出现极小值。平均幅度差函数和自相关函数有密切的关系，两者之间的关系可由下式表达：

$$F_n(k) = \sqrt{2}\beta(k) [R_n(0) - R_n(k)]^{1/2} \quad (3-12)$$

式中， $\beta(k)$ 对不同的语音段在 0.6 ~ 1.0 之间变化，但是对一个特定的语音段，它随 k 值变化并不明显。

显然，计算 $F_n(k)$ 只需加、减法和取绝对值的运算，与自相关函数的加法与乘法相比，其运算量大大减少，尤其在用硬件实现语音信号分析时有很大好处。为此，平均幅度差已被用在许多实时语音处理系统中。

3.2.3 实验步骤及要求

1. 实验步骤

运行 MATLAB → 新建 m 文件 → 编写 m 程序 → 编译并调试。

2. 实验要求

1) 为了显示方便，编程实现 FrameTimeC 函数，函数功能为计算分帧后每帧语音中点处对应的时间。函数定义如下：

函数格式：

```
frametime = FrameTimeC( frameNum, framelen, inc, fs)
```

输入参数: frameNum 是帧的个数; framelen 是帧长; inc 是帧移; fs 是采样频率。

输出参数: frametime 是分帧后每帧对应的时间。

2) 编程实现短时能量、短时平均幅度和短时过零率, 显示例图如图 3-4 所示。参考测试语音为 C3_2_y. wav。

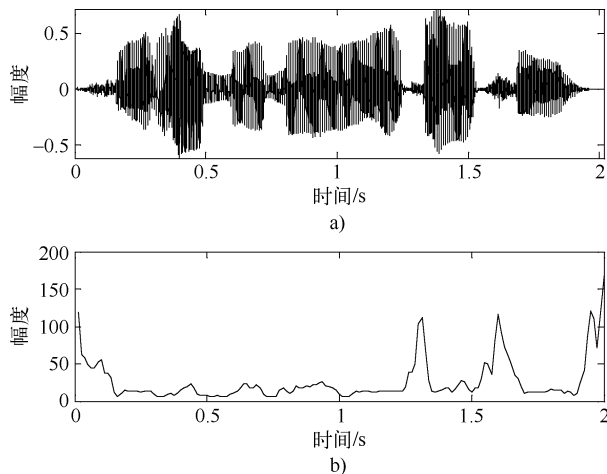


图 3-4 短时过零率的显示例图

a) 语音波形 b) 短时过零率

每个参数的函数定义格式为:

```
funcvalue = funcname( x, win, inc)
```

其中 x 为语音信号, win 为窗函数或帧长, inc 为帧移, funcvalue 为 [1, 帧数] 的向量。

3) 编程实现短时自相关和短时平均幅度差, 显示例图如图 3-5 所示。参考测试语音为 C3_2_y. wav。

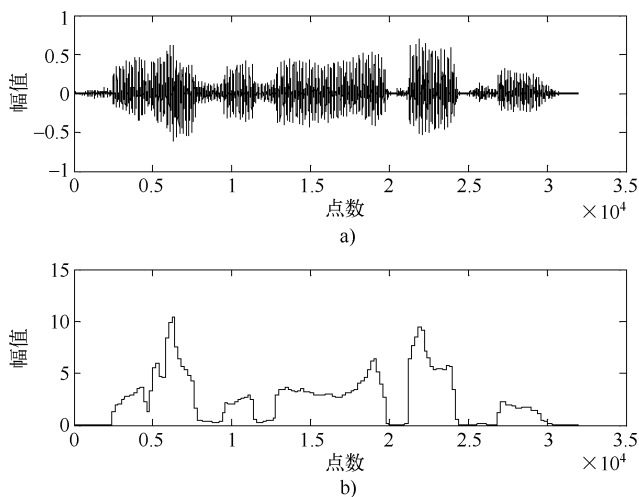


图 3-5 短时自相关的显示例图

a) 语音波形 b) 短时自相关

每个参数的函数定义格式为:

$$\text{funcvalue} = \text{funcname}(x)$$

其中 x 为语音信号, funcvalue 为[帧长,帧数]的矩阵。

这里显示的语音信号,是语音帧拼接而成,即没有去掉交叠项。

3.2.4 思考题

编程比较不同的窗函数对短时域参数估计的影响。

3.2.5 参考例程

```
% 短时时域分析参数计算并显示
clear all;clc;close all;

[x,Fs] = wavread('C3_2_s_1.wav'); % 读入数据文件

wlen = 200;inc = 100; % 给出帧长和帧移
win1 = boxcar(wlen); % 给出海宁窗
win2 = hanning(wlen); % 给出海宁窗
win3 = hamming(wlen); % 给出海宁窗
N = length(x); % 信号长度
time = (0:N-1)/Fs; % 计算出信号的时间刻度
En1 = STEn(x,win1,inc); % 短时能量
En2 = STEn(x,win2,inc); % 短时能量
En3 = STEn(x,win3,inc); % 短时能量
Mn1 = STMn(x,win1,inc); % 短时平均幅度
Mn2 = STMn(x,win2,inc); % 短时平均幅度
Mn3 = STMn(x,win3,inc); % 短时平均幅度
Zcr1 = STZcr(x,win1,inc); % 短时过零率
Zcr2 = STZcr(x,win2,inc); % 短时过零率
Zcr3 = STZcr(x,win3,inc); % 短时过零率
% 此处和上述3个参数不同,返回的不是向量而是矩阵,因为一帧信号得到的不是一个数值
X1 = enframe(x,win1,inc); % 分帧
xn1 = X1(:);
Ac1 = STAc(X1);
Ac1 = Ac1(:);
Amdf1 = STAmdf(X1);
Amdf1 = Amdf1(:);

X2 = enframe(x,win2,inc); % 分帧
xn2 = X2(:);
Ac2 = STAc(X2);
Ac2 = Ac2(:);
Amdf2 = STAmdf(X2);
Amdf2 = Amdf2(:);
```

```
X3 = enframe(x, win3, inc); %分帧
xn = X3(:);
Ac3 = STAc(X3);
Ac3 = Ac3(:);
Amdf3 = STAmdf(X3);
Amdf3 = Amdf3(:);

fn = length(En1); %求出帧数

figure
subplot 311; plot(Mn1', B)
title( (a)短时幅度(矩形窗) );
ylabel( 幅值 ); xlabel( 帧数 );
subplot 312; plot(Mn2', B)
title( (b)短时幅度(汉宁窗) );
ylabel( 幅值 ); xlabel( 帧数 );
subplot 313; plot(Mn3', B)
title( (b)短时幅度(汉明窗) );
ylabel( 幅值 ); xlabel( 帧数 );

figure
subplot 311; plot(En1', B)
title( (a)短时能量(矩形窗) );
ylabel( 幅值 ); xlabel( 帧数 );
subplot 312; plot(En2', B)
title( (b)短时能量(汉宁窗) );
ylabel( 幅值 ); xlabel( 帧数 );
subplot 313; plot(En3', B)
title( (b)短时能量(汉明窗) );
ylabel( 幅值 ); xlabel( 帧数 );

figure
subplot 311; plot(Zcr1', B)
title( (a)短时过零率(矩形窗) );
ylabel( 幅值 ); xlabel( 帧数 );
subplot 312; plot(Zcr2', B)
title( (b)短时过零率(汉宁窗) );
ylabel( 幅值 ); xlabel( 帧数 );
subplot 313; plot(Zcr3', B)
title( (b)短时过零率(汉明窗) );
ylabel( 幅值 ); xlabel( 帧数 );

figure
subplot 311; plot(Ac1', B)
```

```

title( (a)短时自相关图(矩形窗) );
ylabel( 幅值 );xlabel( 点数 );
subplot 312;plot( Ac2', B )
title( (b)短时自相关图(汉宁窗) );
ylabel( 幅值 );xlabel( 点数 );
subplot 313;plot( Ac3', B )
title( (b)短时自相关图(汉明窗) );
ylabel( 幅值 );xlabel( 点数 );

```

```

figure
subplot 311;plot( Amdf1', B )
title( (a)短时幅度差(矩形窗) );
ylabel( 幅值 );xlabel( 帧数 );
subplot 312;plot( Amdf2', B )
title( (b)短时幅度差(汉宁窗) );
ylabel( 幅值 );xlabel( 帧数 );
subplot 313;plot( Amdf3', B )
title( (b)短时幅度差(汉明窗) );
ylabel( 幅值 );xlabel( 帧数 );

```

3.3 短时频域分析

3.3.1 实验目的

- 1) 了解短时傅里叶变换的原理，并编程实现短时傅里叶函数。
- 2) 了解语谱图的意义和表现方法，并编程实现。

3.3.2 实验原理

1. 短时傅里叶变换

语音信号是一种典型的非平稳信号，但是其非平稳性是由发音器官的物理运动过程而产生的，此过程与声波振动的速度相比较缓慢，可以假定在 10 ~ 30 ms 这样的短时间内是平稳的。傅里叶分析是分析线性系统和平稳信号稳态特性的强有力手段，而短时傅里叶分析，也叫时间依赖傅立叶变换，就是在短时平稳的假定下，用稳态分析方法处理非平稳信号的一种方法。

设语音波形时域信号为 $x(l)$ 、加窗分帧处理后得到的第 n 帧语音信号为 $x_n(m)$ ，则 $x_n(m)$ 满足下式：

$$x_n(m) = w(m)x(n+m) \quad 0 \leq m \leq N-1 \quad (3-13)$$

设离散时域采样信号为 $x(n)$ ， $n=0,1,\dots,N-1$ ，其中 n 为时域采样点序号， $N-1$ 是信号长度。然后对信号进行分帧处理，则 $x(n)$ 表示为 $x_n(m)$ ， $m=0,1,\dots,N-1$ ，其中 n 是帧序号， m 是帧同步的时间序号。信号 $x(n)$ 的短时傅里叶变换为

$$X_n(e^{j\omega}) = \sum_{m=0}^{N-1} x_n(m) e^{-j\omega m} \quad (3-14)$$

定义角频率 $w = 2\pi k/N$ ，则得离散的短时傅里叶变换 (DFT)，它实际上是 $X_n(e^{jw})$ 在频域的取样，如下所示：

$$X_n(e^{j\frac{2\pi k}{N}}) = X_n(k) = \sum_{m=0}^{N-1} x_n(m) e^{-j\frac{2\pi km}{N}} \quad (0 \leq k \leq N-1) \quad (3-15)$$

在语音信号数字处理中，都是采用 $x_n(m)$ 的离散傅里叶变换 $X_n(k)$ 来替代 $X_n(e^{jw})$ ，并且可以用高效的快速傅里叶变换 (FFT) 算法完成由 $x_n(m)$ 至 $X_n(k)$ 的转换。当然，这时窗长 N 必须是 2 的倍数 2^L (L 是整数)。

2. 语谱图表示与实现方法

一般定义 $|X_n(k)|$ 为 $x(n)$ 的短时幅度谱估计，而时间处频谱能量密度函数 (或功率谱函数) $P(n, k)$ 为

$$P(n, k) = |X_n(k)|^2 \quad (3-16)$$

则 $P(n, k)$ 是二维的非负实值函数，并且不难证明它是信号 $x(n)$ 的短时自相关函数的傅里叶变换。用时间 n 作为横坐标， k 作为纵坐标，将 $P(n, k)$ 的值表示为灰度级所构成的二维图像就是语谱图。如果通过变换 $10 \log_{10} P(n, k)$ 后，得到语谱图就是采用 dB 进行表示的。将经过变换后的矩阵精细图像和色彩的映射后，就可得到彩色的语谱图，如图 3-6 所示。

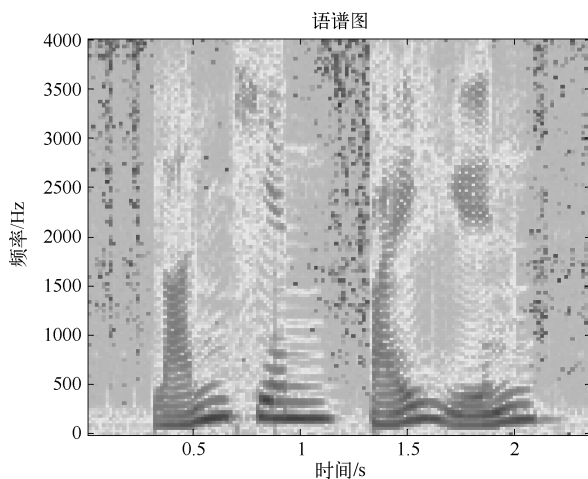


图 3-6 语谱图显示

需要用到的 MATLAB 函数主要有两个：

1) 频谱图显示函数：`imagesc (t, f, L)`。其中 t 是时间坐标， f 是频率坐标， L 则是从功率谱值经伪彩色映射后的彩色电平值，即 $10 \log_{10} P(n, k)$ 。

2) 伪彩色映射函数：`colormap (MAP)`。其中 MAP 是所采用的伪彩色映射矩阵，默认值为 `JET`，即大值为红色，小值为蓝色。可以通过 `MAP = colormap` 获得当前的伪彩色映射矩阵，它可以是一个任意行的矩阵，但其必须有且只有三列，并分别表示红色、绿色和蓝色的饱和度。

语谱图中的花纹有横杠、乱纹和竖直条等。横杠是与时间轴平行的几条深黑色带纹，它们是共振峰。从横杠对应的频率和宽度可以确定相应的共振峰频率和带宽。在一个语音段的语谱图中，有没有横杠出现是判断它是否是浊音的重要标志。竖直条 (又叫冲直条) 是语谱图中出现与时间轴垂直的一条窄黑条。每个竖直条相当于一个基音，条纹的起点相当于声门脉冲的起点，条纹之间的距离表示基音周期。条纹越密表示基音频率越高。

3.3.3 实验步骤及要求

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

1) 根据短时傅里叶变换的原理, 编写其函数。函数定义如下:

函数格式:

```
d = STFT(x, win, nfft, inc)
```

输入参数: x 是语音信号; win 是帧长或窗函数, 若为窗函数, 帧长便取窗函数长; inc 是帧移; $nfft$ 是快速傅里叶的点数。

输出参数: d 是得到的语谱图矩阵。

2) 根据语谱图的显示原理, 编程实现语谱图的计算和显示, 显示效果如图 3-6 所示。

3.3.4 思考题

编程实现将语谱图矩阵还原为原始的语音信号。

3.3.5 参考例程

```
% 语谱图实现函数
function Y = NSpectrogram(x, win, inc, fs)
wlen = length(win);
y = enframe(x, win, inc); % 分帧
fn = size(y, 2); % 帧数

W2 = wlen/2 + 1;
n2 = 1:W2;
freq = (n2-1) * fs/wlen; % 计算 FFT 后的频率刻度
Y = fft(y); % 短时傅里叶变换
Yn = Y(n2, :);
clf % 初始化图形
% 画出语谱图
frameTime = FrameTimeC(fn, wlen, inc, fs); % 计算每帧对应的时间
imagesc(frameTime, freq, 20 * log10(abs(Yn) + eps)); % 画出 Y 的图像
axis xy; ylabel('频率/Hz'); xlabel('时间/s');
title('语谱图');
colormap(jet)
```

3.4 倒谱分析与 MFCC 系数

3.4.1 实验目的

1) 了解语音信号倒谱分析的意义。

- 2) 掌握语音信号倒谱和复倒谱分析的原理。
- 3) 编程实现倒谱和复倒谱计算函数。

3.4.2 实验原理

1. 倒谱分析

根据对语音产生的机理的研究可知, 语音信号 $x(n)$ 可看作是声门激励信号 $x_1(n)$ 和声道冲激响应信号 $x_2(n)$ 的卷积, 即

$$x(n) = x_1(n) * x_2(n) \quad (3-17)$$

为了便于处理各卷积信号, 系统通常采用同态处理的方法进行解卷积, 即将卷积关系变为求和关系, 分离参与卷积的各个信号。一般同态系统可分解为三个部分, 如图 3-7 所示。

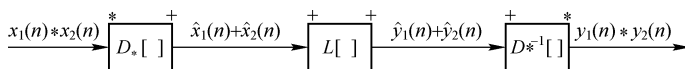


图 3-7 同态系统的组成

如图 3-7 所示, 系统包含两个特征子系统 (取决于信号的组合规则) 和一个线性子系统 (取决于处理的要求)。图中, 符号 $*$ 、 $+$ 和 \cdot 分别表示卷积、加法和乘法运算。

第一个子系统 $D_*[]$ 完成将卷积性信号转化为加性信号的运算, 即对于信号 $x(n) = x_1(n) * x_2(n)$ 进行如下运算处理:

$$\begin{cases} Z[x(n)] = X(z) = X_1(z) \cdot X_2(z) \\ \ln X(z) = \ln X_1(z) + \ln X_2(z) = \hat{X}_1(z) + \hat{X}_2(z) = \hat{X}(z) \\ Z^{-1}[\hat{X}(z)] = Z^{-1}[\hat{X}_1(z) + \hat{X}_2(z)] = \hat{x}_1(n) + \hat{x}_2(n) = \hat{x}(n) \end{cases} \quad (3-18)$$

第二个子系统是一个普通线性系统, 满足线性叠加原理, 用于对加性信号进行线性变换。由于 $\hat{x}(n)$ 为加性信号, 所以第二个子系统可对其进行需要的线性处理得到 $\hat{y}(n)$ 。

第三个子系统是逆特征系统 $D_*^{-1}[]$, 它对 $\hat{y}(n) = \hat{y}_1(n) + \hat{y}_2(n)$ 进行逆变换, 使其恢复为卷积性信号, 处理如下:

$$\begin{cases} Z[\hat{y}(n)] = \hat{Y}(z) = \hat{Y}_1(z) + \hat{Y}_2(z) \\ \exp \hat{Y}(z) = Y(z) = Y_1(z) \cdot Y_2(z) \\ y(n) = Z^{-1}[Y_1(z) \cdot Y_2(z)] = y_1(n) * y_2(n) \end{cases} \quad (3-19)$$

由此可知, 通过第一个子系统 $D_*[]$, 可以将 $x(n) = x_1(n) * x_2(n)$ 变换为 $\hat{x}(n) = \hat{x}_1(n) + \hat{x}_2(n)$ 。此时, 如果 $\hat{x}_1(n)$ 与 $\hat{x}_2(n)$ 处于不同的位置并且互不交替, 那么适当地设计线性系统, 便可将 $x_1(n)$ 与 $x_2(n)$ 分离开来。

在 $D_*[]$ 和 $D_*^{-1}[]$ 系统中, $\hat{x}(n)$ 和 $\hat{y}(n)$ 信号也是时域序列, 但是它们与 $x(n)$ 和 $y(n)$ 所处的离散时域不同, 称为复倒频谱域。 $\hat{x}(n)$ 是 $x(n)$ 的复倒频谱域, 简称复倒谱。其表达式如下:

$$\hat{x}(n) = Z^{-1}[\ln Z[x(n)]] \quad (3-20)$$

在绝大多数数字信号处理中, $X(z)$, $\hat{X}(z)$, $Y(z)$, $\hat{Y}(z)$ 的收敛域均包含单位圆, 因而 $D_*[]$ 和 $D_*^{-1}[]$ 系统有如下形式:

$$D_*[\] : \begin{cases} F[x(n)] = X(e^{j\omega}) \\ \hat{X}(e^{j\omega}) = \ln[X(e^{j\omega})] \\ \hat{x}(n) = F^{-1}[\hat{X}(e^{j\omega})] \end{cases} \quad (3-21)$$

$$D_*^{-1}[\] : \begin{cases} \hat{Y}(e^{j\omega}) = F[\hat{y}(n)] \\ Y(e^{j\omega}) = \exp[\hat{Y}(e^{j\omega})] \\ y(n) = F^{-1}[Y(e^{j\omega})] \end{cases} \quad (3-22)$$

设 $X(e^{j\omega}) = |X(e^{j\omega})| e^{j\arg[X(e^{j\omega})]}$, 则对其取对数得

$$\hat{X}(e^{j\omega}) = \ln |X(e^{j\omega})| + j\arg[X(e^{j\omega})] \quad (3-23)$$

如果只考虑 $\hat{X}(e^{j\omega})$ 的实部, 得

$$c(n) = F^{-1}[\ln |X(e^{j\omega})|] \quad (3-24)$$

式中, $c(n)$ 是 $x(n)$ 对数幅值谱的逆傅里叶变换, 称为倒频谱, 简称倒谱。

由于浊音信号的倒谱中存在着峰值, 出现位置等于该语音段的基音周期, 而清音的倒谱中则不存在峰值。由这个特性可以进行清浊音的判断, 并且可以估计浊音的基音周期。

MATLAB 工具箱为倒谱计算提供了三个函数:

1) cceps 函数——计算复倒谱

调用格式:

$$[\text{xhat}, \text{nd}] = \text{cceps}(x)$$

说明: x 是被测信号序列; xhat 是实信号序列 x 的复倒谱; nd 是为了保证频率 π 处具有零相位特性而对信号 x 所做的单位圆延迟。

2) rceps 函数——计算实倒谱

调用格式:

$$[\text{xh}, \text{yh}] = \text{rceps}(x)$$

说明: x 是被测信号序列; xh 是实信号序列 x 的实倒谱; yh 是最小相位重构序列。

3) icceps 函数——计算逆复倒谱

调用格式:

$$\text{xh} = \text{icceps}(\text{xhat}, \text{nd})$$

说明: xh 是复倒谱 xhat 的逆变换; nd 为所要去除的时间延迟。

2. 离散余弦变换

离散余弦变换 (Discrete Cosine Transform, DCT) 具有信号谱分量丰富、能量集中, 且不需要对语音相位进行估算等优点, 能在较低的运算复杂度下取得较好的语音增强效果。

设 $x(n)$ 是 N 个有限值的一维实数信号序列, $n=0, 1, \dots, N-1$, DCT 的完备正交归一函数是:

$$\begin{cases} X(k) = a(k) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \\ x(n) = \sum_{k=0}^{N-1} a(k) X(k) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \end{cases} \quad (3-25)$$

式中, $a(k)$ 的定义为

$$a(k) = \begin{cases} \sqrt{1/N}, & k=0 \\ \sqrt{2/N}, k \in [1, N-1] \end{cases} \quad (3-26)$$

式中, $n=0, 1, \dots, N-1$; $k=0, 1, \dots, N-1$ 。

将式(3-25)略作变形, 可得到 DCT 的另一表示形式

$$X(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} C(k)x(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad k=0, 1, \dots, N-1 \quad (3-27)$$

式中, $C(k)$ 是正交因子。

$$C(k) = \begin{cases} \sqrt{2}/2, & k=0 \\ 1, & k \in [1, N-1] \end{cases} \quad (3-28)$$

则 DCT 的逆变换为

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C(k)X(k) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad n=0, 1, \dots, N-1 \quad (3-29)$$

在 MATLAB 工具箱中有 `det` 和 `idct` 函数, 其使用方法如下:

1) `det` 函数——离散余弦变换

调用格式:

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

说明: x 是原始信号; X 是离散余弦变换后的序列; N 是离散余弦变换的长度。

2) `idct` 函数——离散余弦逆变换

调用格式:

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

说明: x 是原始信号; X 是离散余弦变换后的序列; N 是离散余弦变换的长度。

3. 美尔频率倒谱系数

美尔频率倒谱系数 (Mel-Frequency Cepstral Coefficients, MFCC) 的分析是基于人的听觉特性机理, 即根据人的听觉实验结果来分析语音的频谱。因为人耳所听到的声音的高低与声音的频率并不成线性正比关系, 所以用 Mel 频率尺度更符合人耳的听觉特性。美尔频率尺度的值大体上对应于实际频率的对数分布关系, 其与实际频率的具体关系可用下式表示:

$$F_{Mel}(f) = 1125 \ln(1 + f/700) \quad (3-30)$$

式中, F_{Mel} 是以美尔 (Mel) 为单位的感知频率; f 是以 Hz 为单位的实际频率。临界频率带宽随着频率的变化而变化, 并与 Mel 频率的增长一致, 在 1000 Hz 以下, 大致呈线性分布, 带宽为 100 Hz 左右; 在 1000 Hz 以上呈对数增长。类似于临界频带的划分, 可以将语音频率划分成一系列三角形的滤波器序列, 即美尔滤波器组, 如图 3-8 所示。

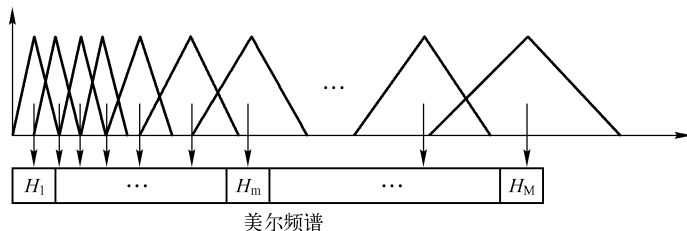


图 3-8 美尔频率尺度滤波器组

在语音的频谱范围内设置若干带通滤波器 $H_m(k)$, $0 \leq m \leq M$, M 为滤波器的个数。每个滤波器具有三角形滤波特性, 其中心频率为 $f(m)$, 在 Mel 频率范围内, 这些滤波器是等带宽的。每个带通滤波器的传递函数为

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \quad (3-31)$$

其中, $\sum_m^{M-1} H_m(k) = 1$ 。

美尔滤波器的中心频率 $f(m)$ 定义为

$$f(m) = \frac{N}{f_s} F_{Mel}^{-1} \left(F_{Mel}(f_h) + m \frac{F_{Mel}(f_h) - F_{Mel}(f_l)}{M+1} \right) \quad (3-32)$$

其中, f_h 和 f_l 分别为滤波器组的最高频率和最低频率; f_s 为采样频率; 单位为 Hz。 M 是滤波器组的数目; N 为 FFT 变换的点数, 式中 $F_{Mel}^{-1}(b) = 700(e^{\frac{b}{1125}} - 1)$ 。

在 MATLAB 中, melbankm 函数可用于计算 Mel 滤波器组。函数定义如下:
调用格式:

$$h = \text{melbankm}(p, n, fs, fl, fh, w)$$

输入参数: fs 是采样频率; fl 是设计的滤波器的最低频率; fh 是设计的滤波器的最高频率 (fl 和 fh 都需要用 fs 进行归一化); p 是设计的 Mel 滤波器的个数; n 是一帧 FFT 后数据的长度; w 是窗函数 ('t' 代表三角窗; 'n' 代表汉宁窗; 'm' 代表汉明窗)。

输出参数: h 是滤波器的频域响应, 是一个 $p \times (n/2 + 1)$ 的数组, p 为滤波器个数, 每个滤波器的响应曲线长 $n/2 + 1$, 相当于取正频率的部分。

4. MFCC 系数的计算

MFCC 特征参数提取原理框图如图 3-9 所示。

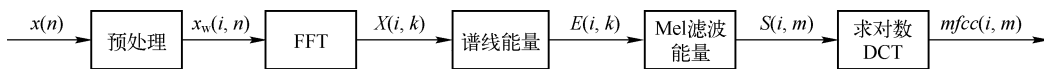


图 3-9 MFCC 特征参数提取原理框图

(1) 预处理

预处理包括预加重、分帧、加窗函数。

预加重: 在第 1 章中已指出声门脉冲的频率响应曲线接近于一个二阶低通滤波器, 而口腔的辐射响应也接近于一个一阶高通滤波器。预加重的目的是为了补偿高频分量的损失, 提升高频分量。预加重的滤波器常设为

$$H(z) = 1 - az^{-1} \quad (3-33)$$

式中, a 为一个常数。

分帧处理: 由于一个语音信号是一个准稳态的信号, 把它分成较短的帧, 在每帧信号中可将其看作稳态信号, 可用处理稳态信号的方法来处理。同时, 为了使一帧与另一帧之间的参数能较平稳地过渡, 在相邻两帧之间互相有部分重叠。

加窗函数：加窗函数的目的是减少频域中的泄漏，将对每一帧语音乘以汉明窗或海宁窗。语音信号 $x(n)$ 经预处理后为 $x_i(m)$ ，其中下标 i 表示分帧后的第 i 帧。

(2) 快速傅里叶变换

对每一帧信号进行 FFT 变换，从时域数据转变为频域数据：

$$X(i, k) = \text{FFT}[x_i(m)] \quad (3-34)$$

(3) 计算谱线能量

对每一帧 FFT 后的数据计算谱线的能量：

$$E(i, k) = [X_i(k)]^2 \quad (3-35)$$

(4) 计算通过美尔滤波器的能量

把求出的每帧谱线能量谱通过美尔滤波器，并计算在该美尔滤波器中的能量。在频域中相当于把每帧的能量谱 $E(i, k)$ （其中 i 表示第 i 帧， k 表示频域中的第 k 条谱线）与美尔滤波器的频域响应 $H_m(k)$ 相乘并相加：

$$S(i, m) = \sum_{k=0}^{N-1} E(i, k) H_m(k), 0 \leq m < M \quad (3-36)$$

(5) 计算 DCT 倒谱

序列 $x(n)$ 的 FFT 倒谱 $\hat{x}(n)$ 为

$$\hat{x}(n) = \text{FT}^{-1}[\hat{X}(k)] \quad (3-37)$$

式中， $\hat{X}(k) = \ln\{\text{FT}[x(n)]\} = \ln\{X(k)\}$ ，FT 和 FT^{-1} 表示傅里叶变换和傅里叶逆变换。序列 $x(n)$ 的 DCT 为

$$X(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} C(k)x(n) \cos\left[\frac{\pi(2n+1)k}{2N}\right], k = 0, 1, \dots, N-1 \quad (3-38)$$

式中，参数 N 是序列 $x(n)$ 的长度； $C(k)$ 是正交因子，可表示为

$$C(k) = \begin{cases} \sqrt{2}/2 & k=0 \\ 1 & k=1, 2, \dots, N-1 \end{cases} \quad (3-39)$$

在式 (3-37) 中求取 FFT 的倒谱是把 $X(k)$ 取对数后计算 FFT 的逆变换。而这里求 DCT 的倒谱和求 FFT 的倒谱相类似，把美尔滤波器的能量取对数后计算 DCT：

$$\text{mfcc}(i, n) = \sqrt{\frac{2}{M}} \sum_{m=0}^{M-1} \log[S(i, m)] \cos\left[\frac{\pi n(2m-1)}{2M}\right] \quad (3-40)$$

式中， $S(i, m)$ 是由式 (3-36) 求出的美尔滤波器能量； m 是指第 m 个美尔滤波器（共有 M 个）； i 是指第 i 帧； n 是 DCT 后的谱线。

3.4.3 实验步骤及要求

1. 实验步骤

运行 MATLAB → 新建 m 文件 → 编写 m 程序 → 编译并调试。

2. 实验要求

1) 根据倒谱计算的原理，编程实现 MATLAB 的 `rceps` 函数，函数定义为

$$\text{xh} = \text{Nrceps}(x)$$

然后调用 MATLAB 的 `cceps` 函数和 `icceps` 函数以及设计的 `rceps` 函数对一段较短语音进

行复倒谱的计算与恢复，图例如图 3-10 所示。

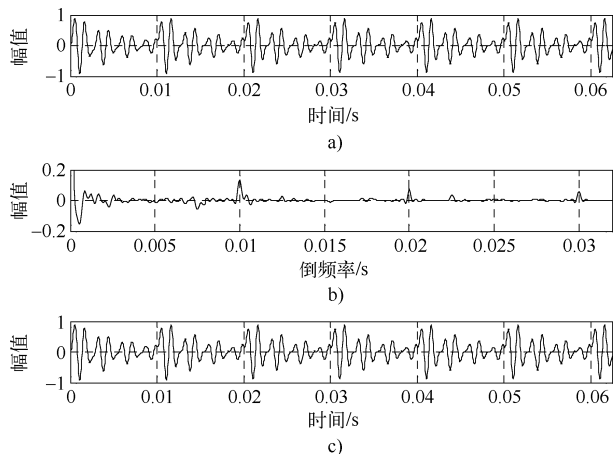


图 3-10 倒谱程序例图

a) 信号波形 b) 信号倒谱图 c) 还原信号

2) 调用 MATLAB 的 DCT 函数求序列 $x(n) = \cos(2\pi fn/f_s)$ 的 DCT 系数 (其中, $n \in [0, 1000)$, $f_s = 1000$ Hz, $f = 50$ Hz), 然后仅用幅值大于 5 的系数进行信号重建, 并比较重建前后信号的差异。

3) 调用 MATLAB 的 melbankm 函数设计 24 个美尔滤波器, 其中采样频率为 8000 Hz, 最低频率参数为 0, 最高频率参数为 0.5 (即 $8000 \text{ Hz} * 0.5 = 4000 \text{ Hz}$), 使用三角函数, 图例如图 3-11 所示。

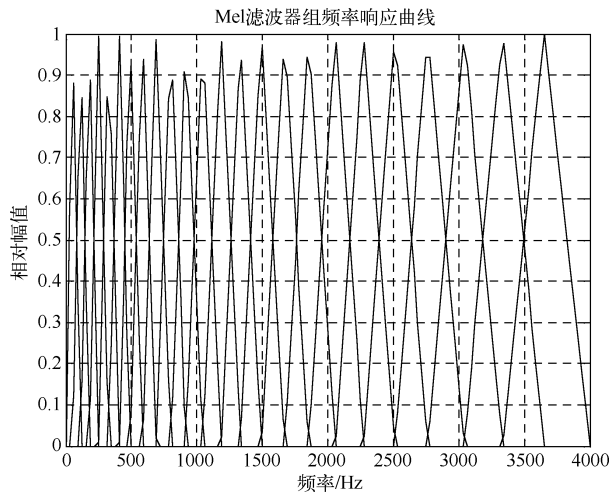


图 3-11 Mel 滤波器组频率响应曲线

4) 根据 MFCC 系数计算流程, 编写 MFCC 计算函数, 并用来计算一段语音的 MFCC 系数。函数定义如下:

$$ccc = \text{Nmfcc}(x, fs, p, \text{frameSize}, \text{inc});$$

其中, x 是输入语音序列; Mel 滤波器的个数为 p ; 采样频率为 fs ; frameSize 为帧长和 FFT 点数; inc 为帧移; ccc 为 MFCC 参数。

3.4.4 思考题

编写离散余弦变换和离散余弦逆变换函数，并与 MATLAB 自带函数进行比较，函数名称为 Ndet 和 Nidct。

3.4.5 参考例程

```
% MFCC 系数计算函数
function ccc = Nmfcc(x,fs,p,frameSize,inc)
% x 是输入语音序列,Mel 滤波器的个数为 p,采样频率为 fs,frameSize 为帧长和 FFT 点数,inc 为帧移;ccc 为 MFCC 参数

% 按帧长为 frameSize,Mel 滤波器的个数为 p,采样频率为 fs
% 提取 Mel 滤波器参数,用汉明窗函数
bank = melbankm(p,frameSize,fs,0,0.5',m);
% 归一化 Mel 滤波器组系数
bank = full(bank);
bank = bank/max(bank(:));

% DCT 系数,12 * p
for k = 1:12
    n = 0:p - 1;
    dctcoef(k,:) = cos((2 * n + 1) * k * pi/(2 * p));
end

% 归一化倒谱提升窗口
w = 1 + 6 * sin(pi * [1:12] ./12);
w = w/max(w);

% 预加重滤波器
xx = double(x);
xx = filter([1 -0.9375],1,xx);

% 语音信号分帧
xx = enframe(xx,frameSize,inc);
n2 = fix(frameSize/2) + 1;
% 计算每帧的 MFCC 参数
for i = 1:size(xx,1)
    y = xx(i,:);
    s = y .* hamming(frameSize);
    t = abs(fft(s));
    t = t.^2;
    c1 = dctcoef * log(bank * t(1:n2));
    c2 = c1 .* w;
    m(i,:) = c2;
end
```

```

end

% 差分系数
dtm = zeros( size( m ) );
for i = 3 : size( m, 1 ) - 2
    dtm( i, : ) = -2 * m( i - 2, : ) - m( i - 1, : ) + m( i + 1, : ) + 2 * m( i + 2, : );
end
dtm = dtm / 3;
% 合并 MFCC 参数和一阶差分 MFCC 参数
ccc = [ m dtm ];
% 去除首尾两帧, 因为这两帧的一阶差分参数为 0
ccc = ccc( 3 : size( m, 1 ) - 2, : );

```

3.5 线性预测分析

3.5.1 实验目的

- 1) 了解线性预测分析在语音信号处理中的重要性和必要性。
- 2) 掌握线性预测分析的基本思想。
- 3) 掌握 MATLAB 进行线性预测分析的流程。

3.5.2 实验原理

1. 语音信号线性预测分析

图 3-12 是简化的语音产生模型, 将辐射、声道以及声门激励的全部效应简化为一个时变的数字滤波器来等效, 其传递函数为

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}}$$

这种表现形式称为 p 阶线性预测模型, 这是一个全极点模型。

此时, $s(n)$ 和 $u(n)$ 间的关系可以用差分方程

$$s(n) = \sum_{i=1}^p a_i s(n-i) + Gu(n) \quad (3-41)$$

表示, 称系统

$$\hat{s}(n) = \sum_{i=1}^p a_i s(n-i) \quad (3-42)$$

为线性预测器。 $\hat{s}(n)$ 是 $s(n)$ 的估计值, 它由过去 p 个值线性组合得到的, 即由 $s(n)$ 过去的值来预测或估计当前值 $s(n)$ 。式中, $a_i (i=1, 2, \dots, p)$ 是线性预测系数。线性预测系数可以通过在某个准则下使预测误差 $e(n)$ 达到最小值的方法来决定, 预测误差的表示形式如下:

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^p a_i s(n-i) \quad (3-43)$$

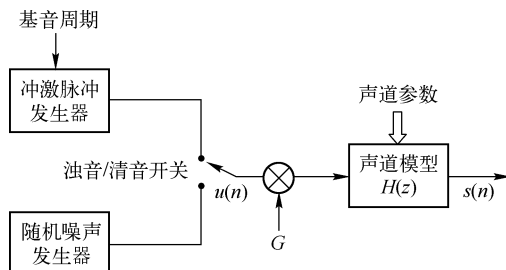


图 3-12 简化的语音产生模型

预测的二次方误差为

$$E = \sum_n e^2(n) = \sum_n [s(n) - \hat{s}(n)]^2 = \sum_n \left[s(n) - \sum_{i=1}^p a_i s(n-i) \right]^2 \quad (3-44)$$

为使 E 最小, 求 E 对 a_i 的偏导为 0, 即

$$\frac{\partial E}{\partial a_j} = 0 \quad (1 \leq j \leq p) \quad (3-45)$$

则有,

$$\frac{\partial E}{\partial a_j} = 2 \sum_n s(n)s(n-j) - 2 \sum_{i=1}^p a_i \sum_n s(n-i)s(n-j) = 0 \quad (1 \leq j \leq p) \quad (3-46)$$

定义 $\varphi(j, i) = \sum_n s(n-i)s(n-j)$, 则式 (3-46) 可简化为

$$\varphi(j, 0) = \sum_{i=1}^p a_i \varphi(j, i) \quad (1 \leq j \leq p) \quad (3-47)$$

联立式 (3-44)、式 (3-46) 和式 (3-47), 可得最小方均误差表示为

$$E = \varphi(0, 0) - \sum_{i=1}^p a_i \varphi(0, i) \quad (3-48)$$

因此, 最小误差由一个固定分量 $\varphi(0, 0)$ 和一个依赖于预测系数的分量 $\sum_{i=1}^p a_i \varphi(0, i)$ 构成。为求解最佳预测器系数, 必须首先求出 $\varphi(j, i)$ ($i, j \in [1, p]$), 然后可按照式 (3-47) 进行求解。很显然, $\varphi(j, i)$ 的计算及方程组的求解都是十分复杂的。

2. 线性预测分析的自相关解法

为了有效地进行线性预测分析, 求得线性预测系数, 有必要用一种高效的方法来求解线性方程组。虽然可以用各种各样的方法来解包含 p 个未知数的 p 个线性方程, 但是系数矩阵的特殊性质使得解方程的效率比普通解法的效率要高得多。自相关法是经典解法之一, 其原理是在整个时间范围内使误差最小, 即设 $s(n)$ 在 $0 \leq n \leq N-1$ 以外等于 0, 等同于假设 $s(n)$ 经过有限长度的窗 (如矩形窗、海宁窗或汉明窗) 的处理。

通常, $s(n)$ 的加窗自相关函数定义为

$$r(j) = \sum_{n=0}^{N-1} s(n)s(n-j) \quad 1 \leq j \leq p \quad (3-49)$$

同式 (3-47) 比较可知, $\varphi(j, i)$ 等效为 $r(j-i)$ 。但是由于 $r(j)$ 为偶函数, 因此 $\varphi(j, i)$ 可表示为

$$\varphi(j, i) = r(|j-i|) \quad (3-50)$$

此时式 (3-47) 可表示为

$$\sum_{i=1}^p a_i r(|j-i|) = r(j) \quad 1 \leq j \leq p \quad (3-51)$$

则最小均方误差改写为

$$E = r(0) - \sum_{i=1}^p a_i r(i) \quad (3-52)$$

展开式 (3-50), 可得方程组为

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p-1) \\ r(1) & r(0) & r(1) & \cdots & r(p-2) \\ r(2) & r(1) & r(0) & \cdots & r(p-3) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(p) \end{bmatrix} \quad (3-53)$$

式 (3-53) 左边为相关函数的矩阵, 以对角线为对称, 其主对角线以及和主对角线平行的任何一条斜线上所有的元素相等。这种矩阵称为托普利兹 (Toeplitz) 矩阵, 而这种方程称为 Yule - Walker 方程。对于式 (3-53) 的矩阵方程无需像求解一般矩阵方程那样进行大量的计算, 利用托普利兹矩阵的性质可以得到求解这种方程的一种高效方法。

这种矩阵方程组可以采用递归方法求解, 其基本思想是递归解法分布进行。在递推算法中, 最常用的是莱文逊 - 杜宾 (Levinson - Durbin) 算法 (如图 3-13 所示)。

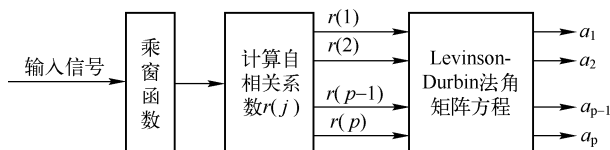


图 3-13 自相关解法

算法的过程和步骤为:

① 当 $i=0$ 时, $E_0 = r(0)$, $a_0 = 1$;

② 对于第 i 次递归 ($i=1, 2, \dots, p$):

$$k_i = \frac{1}{E_{i-1}} \left[r(i) - \sum_{j=1}^{i-1} a_j^{i-1} r(j-i) \right] \quad (3-54)$$

$$ia_i^{(i)} = k_i \quad (3-55)$$

对于 $j=1$ 到 $i-1$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \quad (3-56)$$

$$E_i = (1 - k_i^2) E_{i-1} \quad (3-57)$$

③ 增益 G 为

$$G = \sqrt{E_p} \quad (3-58)$$

通过对式 (3-54) ~ 式 (3-56) 进行递推求解, 可获得最终解为

$$a_i = a_j^{(p)} \quad 1 \leq j \leq p \quad (3-59)$$

由式 (3-57) 可得

$$E_p = r(0) \prod_{i=1}^p (1 - k_i^2) \quad (3-60)$$

由式 (3-60) 可知, 最小均方误差 E_p 一定大于 0, 且随着预测器阶数的增加而减小。因此每一步算出的预测误差总是小于前一步的预测误差。这就表明, 虽然预测器的精度会随着阶数的增加而提高, 但误差永远不会消除。由式 (3-60) 还可知, 参数 k_i 一定满足

$$|k_i| < 1, \quad 1 \leq i \leq p \quad (3-61)$$

由递归算法可知, 每一步计算都与 k_i 有关, 说明这个系数具有特殊的意义, 通常称之为反射系数或偏相关系数。可以证明, 它就是多项式 $A(z)$ 的根在单位圆内的充分必要条件,

因此它可以保证系统 $H(k)$ 的稳定性。

在 MATLAB 工具箱中有 lpc 函数，是用莱文逊 - 杜宾的自相关方法计算预测系数的。其函数说明如下：

名称：lpc

功能：是 MATLAB 中自带的用莱文逊 - 杜宾自相关法计算线性预测系数。

调用格式：

$$[\text{ar}, \text{r}] = \text{lpc}(\text{x}, \text{p})$$

说明：输入参数 x 是一帧数据； p 是线性预测阶数。输出参数 ar 是按下式计算得到的预测系数 $\{a_i\}$ ($i = 1, 2, \dots, p$)：

$$A(z) = 1 + \sum_{i=1}^p a_i z^{-i} = \sum_{i=0}^p a_i z^{-i} \quad a_0 = 1 \quad (3-62)$$

求得的预测系数 a_i 有 $p+1$ 个，且第 1 个 a_0 永远为 1； e 是预测计算中的最小方均方程。

3. 线性预测的其他参数

用线性预测分析法求得的是一个全极点模型的传递函数。在语音产生模型中，这一全极点模型与声道滤波器的假设相符合，而形式上是一自回归滤波器。用全极点模型所表征的声道滤波器，除预测系数 $\{a_i\}$ 外，还有其他不同形式的滤波器参数。这些参数一般可由线性预测系数推导得到，但各有不同的物理意义和特性。在对语音信号做进一步处理时，为了达到不同的应用目的，往往按照这些特性来选择某种合适的参数来描述语音信号。

(1) 预测误差及其自相关函数

由式 (3-43) 可知，预测误差为

$$e(n) = s(n) - \sum_{i=1}^p a_i s(n-i) \quad (3-63)$$

而预测误差的自相关函数为

$$R_e(m) = \sum_{n=0}^{N-1-m} e(n)e(n+m) \quad (3-64)$$

(2) 反射系数和声道面积

反射系数 $\{k_i\}$ 在低速率语音编码、语音合成、语音识别和说话人识别等许多领域都是非常重要的特征参数。由式 (3-56) 可得

$$\begin{cases} a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \\ a_{i-j}^{(i)} = a_{i-j}^{(i-1)} - k_i a_j^{(i-1)} \end{cases} \quad j = 1, \dots, i-1 \quad (3-65)$$

进一步推导，可得

$$a_j^{(i-1)} = (a_j^{(i)} + a_j^{(i)} a_{i-j}^{(i)}) / (1 - k_i^2) \quad j = 1, \dots, i-1 \quad (3-66)$$

由线性预测系数 $\{a_i\}$ 可递推出反射系数 $\{k_i\}$ ，即

$$\begin{cases} a_j^{(p)} = a_j & j = 1, 2, \dots, p \\ k_i = a_i^{(i)} & \\ a_j^{(i-1)} = (a_j^{(i)} + a_j^{(i)} a_{i-j}^{(i)}) / (1 - k_i^2) & j = 1, \dots, i-1 \end{cases} \quad (3-67)$$

反射系数的取值范围为 $[-1, 1]$ ，这是保证相应的系统函数稳定的充分必要条件。从声

学理论可知,声道可以被模拟成一系列截面积不等的无损声道的级联。反射系数 $\{k_i\}$ 反映了声波在各管道边界处的反射量,有

$$k_i = \frac{A_{i+1} - A_i}{A_{i+1} + A_i} \quad (3-68)$$

式中, A_i 是第 i 节声管的面积函数。式 (3-68) 经变换后,可得声管模型各节的面积比为

$$\frac{A_i}{A_{i+1}} = \frac{(1 - k_i)}{(1 + k_i)} \quad (3-69)$$

(3) 线性预测的频谱

由式 (3-40) 可知,一帧语音信号 $x(n)$ 模型可化为一个 p 阶的线性预测模型。当 $z = e^{j\omega}$ 时,能得到线性预测系数的频谱 (令 $G=1$):

$$H(e^{j\omega}) = \frac{1}{1 - \sum_{n=1}^p a_n z^{-j\omega n}} \quad (3-70)$$

根据语音信号的数字模型,在不考虑激励和辐射时, $H(e^{j\omega})$ 即 $X(e^{j\omega})$ 的频谱的包络谱。线性预测系数的频谱勾画出了 FFT 频谱的包络,反映了声道的共振峰的结构。

(4) 线性预测倒谱

语音信号的倒谱可以通过对信号做傅里叶变换,取模的对数,再求傅里叶逆变换得到。由于频率响应 $H(e^{j\omega})$ 反映声道的频率响应和被分析信号的谱包络,因此用 $\log |H(e^{j\omega})|$ 做傅里叶逆变换求出的线性预测倒谱系数 (Linear Prediction Cepstrum Coefficient, LPCC), 其可看做是原始信号短时倒谱的一种近似。

通过线性预测分析得到的合成滤波器的系统函数为 $H(z) = 1 / \left(1 - \sum_{i=1}^p a_i z^{-i}\right)$, 其冲激响应为 $h(n)$ 。下面求 $h(n)$ 的倒谱 $\hat{h}(n)$, 首先根据同态处理法,有

$$\hat{H}(z) = \log H(z) \quad (3-71)$$

因为 $H(z)$ 是最小相位的,即在单位圆内是解析的,所以 $\hat{H}(z)$ 可以展开成级数形式,即

$$\hat{H}(z) = \sum_{n=1}^{+\infty} \hat{h}(n) z^{-n} \quad (3-72)$$

也就是说, $\hat{H}(z)$ 的逆变换 $\hat{h}(n)$ 是存在的。设 $\hat{h}(0) = 0$, 将式 (3-72) 两边同时对 z^{-1} 求导,得

$$\frac{\partial}{\partial z^{-1}} \log \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}} = \frac{\partial}{\partial z^{-1}} \sum_{n=1}^{+\infty} \hat{h}(n) z^{-n} \quad (3-73)$$

得到

$$\sum_{n=1}^{+\infty} n \hat{h}(n) z^{-n+1} = \frac{\sum_{i=1}^p i a_i z^{-i+1}}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (3-74)$$

有

$$\left(1 - \sum_{i=1}^p a_i z^{-i}\right) \sum_{n=1}^{+\infty} n \hat{h}(n) z^{-n+1} = \sum_{i=1}^p i a_i z^{-i+1} \quad (3-75)$$

令式 (3-75) 等号两边 z 的各次幂前系数分别相等, 得到 $\hat{h}(n)$ 和 a_i 间的递推关系:

$$\hat{h}(1) = a_1 \quad (3-76)$$

$$\hat{h}(n) = a_n + \sum_{i=1}^{n-1} \left(1 - \frac{i}{n}\right) a_i \hat{h}(n-i) \quad 1 < n \leq p \quad (3-77)$$

$$\hat{h}(n) = \sum_{i=1}^p \left(1 - \frac{i}{n}\right) a_i \hat{h}(n-i) \quad n > p \quad (3-78)$$

按式 (3-76) ~ 式 (3-78) 可直接从预测系数 $\{a_i\}$ 求得倒谱 $\hat{h}(n)$ 。这个倒谱系数是根据线性预测模型得到的, 又利用线性预测中声道系统函数 $H(z)$ 的最小相位特性, 因此避免了一般同态处理中求复对数的麻烦。

3.5.3 实验步骤及要求

1. 实验步骤

运行 MATLAB → 新建 m 文件 → 编写 m 程序 → 编译并调试。

2. 实验要求

1) 根据莱文逊 - 杜宾自相关法求线性预测系数的原理, 编写 MATLAB 函数, 并与 MATLAB 自带的 LPC 函数进行比较, 测试效果如图 3-14 所示, 测试语音文件为 C3_5_y.wav。

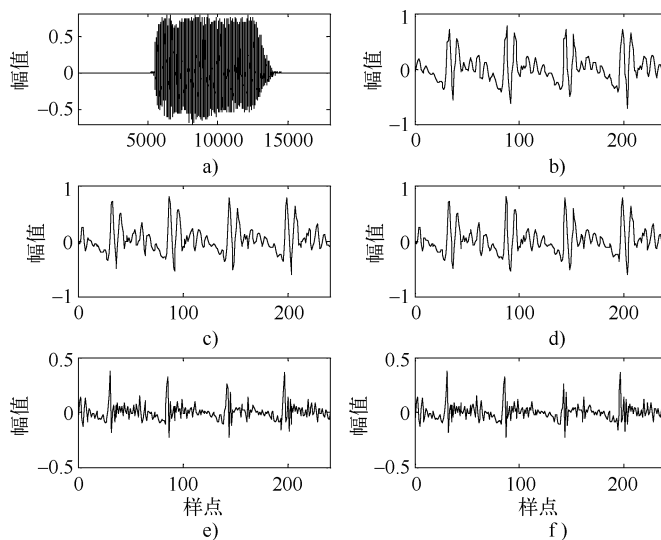


图 3-14 线性预测系数对比

a) 元音/a/波形 b) 一帧数据 c) LPC 预测值 d) lpc_coeff 预测值 e) LPC 预测误差 f) lpc_coeff 预测误差

函数定义如下:

名称: lpc_coeff

功能: 用莱文逊 - 杜宾自相关法计算线性预测系数。

调用格式:

$$[ar, G] = \text{lpc_coeff}(s, p)$$

说明: 输入参数 s 是一帧数据; p 是线性预测阶数。输出参数 ar 是按式 (3-57) 计算

得到的预测系数 $\{a_i\}$ ($i=1,2,\dots,p$), 共得 p 个预测系数; G 是按式 (3-58) 计算得到的增益系数。

2) 编写求取 LPC 预测系数的复频谱函数, 并与 FFT 频谱进行对比, 测试效果如图 3-15 所示, 测试语音文件为 C3_5_y.wav。LPC 预测系数的复频谱函数的定义如下:

名称: lpcff

功能: 计算线性预测系数的复频谱。

调用格式:

$ff = \text{lpcff}(ar, np)$

说明: ar 是线性预测系数, np 是 FFT 阶数; 输出 ff 是线性预测系数的复频谱。

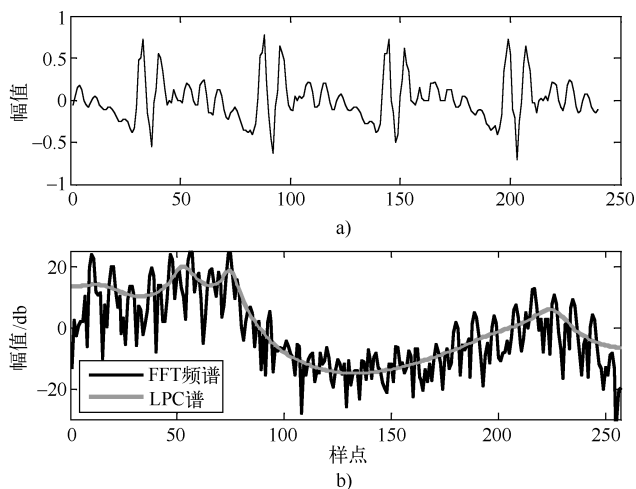


图 3-15 LPC 预测系数的复频谱与 FFT 频谱

a) 一帧语音信号的波形 b) FFT 频谱和 LPC 谱的比较

3) 编程实现由预测系数求取线性预测倒谱系数, 并比较与 rceps 函数的区别, 测试效果如图 3-16 所示, 测试语音文件为 C3_5_y.wav。

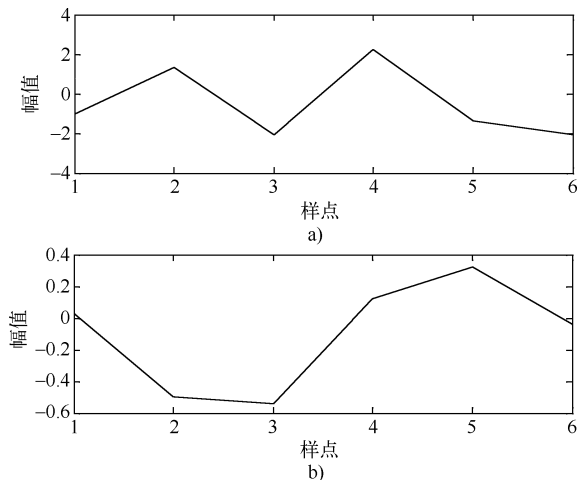


图 3-16 LPCCC 比较

a) 线性预测系数求 LPCCC b) 直接求 LPCCC

名称: lpc_lpcem

功能: 由预测系数 a_i 求 LPCC。

调用格式:

$$\text{lpc} = \text{lpc_lpcem}(\text{ar}, \text{n_lpc}, \text{n_lpc})$$

说明: 输入参数 ar 是线性预测系数; n_lpc 是预测系数的长度; n_lpc 是 LPC 倒谱的长度; 输出参数 lpc 是线性预测倒谱系数。

3.5.4 思考题

编程实现预测系数与反射系数间的转换和反射系数与声管面积间的转换。函数定义如下。

(1) 由预测系数求反射系数

名称: lpcar2rf

功能: 已知预测系数计算出反射系数

调用格式:

$$\text{rf} = \text{lpcar2rf}(\text{ar})$$

说明: ar 是预测系数; rf 是反射系数。

(2) 由反射系数求出预测系数

名称: lpcrf2ar

功能: 已知反射系数计算出预测系数

调用格式:

$$[\text{ar}, \text{arp}, \text{aru}, \text{g}] = \text{lpcrf2ar}(\text{rf})$$

说明: rf 是反射系数; ar 是预测系数; arp 是压力传递函数; aru 是体积速度的传递函数; g 是增益。

(3) 由反射系数求声管面积比

名称: lpcrf2ao

功能: 已知反射系数计算出归正化的声管面积。

调用格式:

$$\text{ao} = \text{lpcrf2ao}(\text{rf})$$

说明: rf 是反射系数; ao 是声管面积比。

(4) 由声管面积求反射系数

名称: lpcao2rf

功能: 已知归正化的声管面积比计算出反射系数。

调用格式:

$$\text{rf} = \text{lpcao2rf}(\text{ao})$$

说明: ao 是声管面积比; rf 是反射系数。

3.5.5 参考例程

```
% 从 LPC 计算线性预测倒谱系数
function lpc = lpc_lpcem(ar, n_lpc, n_lpc)
```

```

lpcc = zeros(n_lpcc,1);
lpcc(1) = ar(1);           % 计算 n = 1 的 lpcc
for n = 2:n_lpc           % 计算 n = 2, ..., p 的 lpcc
    lpcc(n) = ar(n);
    for l = 1:n-1
        lpcc(n) = lpcc(n) + ar(l) * lpcc(n-1) * (n-1)/n;
    end
end

for n = n_lpc + 1:n_lpcc % 计算 n > p 的 lpcc
    lpcc(n) = 0;
    for l = 1:n_lpc
        lpcc(n) = lpcc(n) + ar(l) * lpcc(n-1) * (n-1)/n;
    end
end

lpcc = -lpcc;

```

3.6 线谱对转换实验

3.6.1 实验目的

- 1) 了解线谱对的定义和特点。
- 2) 掌握线性预测与线谱对转换的基本思想。
- 3) 能编程实现线性预测系数与线谱对的相互转换。

3.6.2 实验原理

1. 线谱对的定义与特点

根据线性预测的原理可知, $A(z) = 1 - \sum_{i=1}^p a_i z^{-i}$ 为线性预测误差滤波器, 其倒数 $H(z) = 1/A(z)$ 为线性预测合成滤波器。该滤波器常被用于重建语音, 但是当直接对线性预测系数 a_i 进行编码时, $H(z)$ 的稳定性就不能得到保证。由此引出了许多与线性预测等价的表示方法, 以提高线性预测的鲁棒性, 如线谱对 (Line Spectrum Pair, LSP) 就是线性预测的一种等价表示形式。LSP 最早由 Itakura 引入, 但是直到人们发现利用 LSP 在频域对语音进行编码, 比其他变换技术更能改善编码效率时, LSP 才被重视。由于 LSP 能够保证线性预测滤波器的稳定性, 其小的系数偏差带来的谱误差也只是局部的, 且 LSP 具有良好的量化特性和内插特性, 因而已经在许多编码系统中得到成功的应用。LSP 分析的主要缺点是运算量较大。

LSP 作为线性预测参数的一种表示形式, 可通过求解 $p+1$ 阶对称和反对称多项式的共轭复根得到。其中, $p+1$ 阶对称和反对称多项式表示如下:

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1}) \quad (3-79)$$

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1}) \quad (3-80)$$

其中,

$$z^{-(p+1)} A(z^{-1}) = z^{-(p+1)} - a_1 z^{-p} - a_2 z^{-p+1} - \dots - a_p z^{-1} \quad (3-81)$$

可以推出:

$$P(z) = 1 - (a_1 + a_p)z^{-1} - (a_2 + a_{p-1})z^{-2} - \cdots - (a_p + a_1)z^{-p} + z^{-(p+1)} \quad (3-82)$$

$$Q(z) = 1 - (a_1 - a_p)z^{-1} - (a_2 - a_{p-1})z^{-2} - \cdots - (a_p - a_1)z^{-p} - z^{-(p+1)} \quad (3-83)$$

$P(z)$ 、 $Q(z)$ 分别为对称和反对称的实系数多项式,它们都有共轭复根。可以证明,当 $A(z)$ 的根位于单位圆内时, $P(z)$ 和 $Q(z)$ 的根都位于单位圆上,而且相互交替出现。如果阶数 p 是偶数,则 $P(z)$ 和 $Q(z)$ 各有一个实根,其中 $P(z)$ 有一个根 $z = -1$, $Q(z)$ 有一个根 $z = 1$ 。如果阶数 p 是奇数,则 $P(z)$ 有两个根 $z = -1$, $z = 1$, $Q(z)$ 没有实根。此处假定 p 是偶数,这样 $P(z)$ 和 $Q(z)$ 各有 $p/2$ 个共轭复根位于单位圆上,共轭复根的形式为 $z_i = e^{\pm j\omega_i}$,设 $p(z)$ 的零点为 $e^{\pm j\omega_i}$, $Q(z)$ 的零点为 $e^{\pm j\theta_i}$,则满足

$$0 < \omega_1 < \theta_1 < \cdots < \omega_{p/2} < \theta_{p/2} < \pi \quad (3-84)$$

其中, ω_i 和 θ_i 分别为 $P(z)$ 和 $Q(z)$ 的第 i 个根。

$$\begin{aligned} P(z) &= (1 + z^{-1}) \prod_{i=1}^{p/2} (1 - z^{-1} e^{j\omega_i})(1 - z^{-1} e^{-j\omega_i}) \\ &= (1 + z^{-1}) \prod_{i=1}^{p/2} (1 - 2\cos\omega_i z^{-1} + z^{-2}) \end{aligned} \quad (3-85)$$

$$\begin{aligned} Q(z) &= (1 - z^{-1}) \prod_{i=1}^{p/2} (1 - z^{-1} e^{j\theta_i})(1 - z^{-1} e^{-j\theta_i}) \\ &= (1 - z^{-1}) \prod_{i=1}^{p/2} (1 - 2\cos\theta_i z^{-1} + z^{-2}) \end{aligned} \quad (3-86)$$

式中, $\cos\omega_i$ 和 $\cos\theta_i$ ($i = 1, 2, \dots, p/2$)是LSP系数在余弦域的表达; ω_i 和 θ_i 则是与LSP系数对应的线谱频率(Line Spectrum Frequency, LSF)。

由于LSP参数成对出现,且反应信号的频谱特性,因此称为线谱对。LSF就是线谱对分析所要求解的参数。

LSP参数的特性包括:

- 1) LSP参数都在单位圆上且降序排列。
- 2) 与LSP参数对应的LSF升序排列,且 $P(z)$ 和 $Q(z)$ 的根相互交替出现,这可使与LSP参数对应的LPC滤波器的稳定性得到保证。上述特性保证了在单位圆上,任何时候 $P(z)$ 和 $Q(z)$ 不可能同时为零。
- 3) LSP参数具有相对独立的性质。如果某个特定的LSP参数中只移动其中任意一个线谱频率的位置,那么它所对应的频谱只在附近与原始语音频谱有差异,而在其他LSP频率上则变化很小。这样有利于LSP参数的量化和内插。
- 4) LSP参数能够反映声道幅度谱的特点,在幅度大的地方分布较密,反之较疏。这样就相当于反映出了幅度谱中的共振峰特性。

按照线性预测分析的原理,语音信号的谱特性可以由LPC模型谱来估计,将式(3-79)和式(3-80)相加,可得

$$A(z) = \frac{1}{2} [P(z) + Q(z)] \quad (3-87)$$

此时,功率谱可以表示为

$$|H(e^{j\omega})|^2 = \frac{1}{|A(e^{j\omega})|^2} = 4 |P(e^{j\omega}) + Q(e^{j\omega})|^{-2}$$

$$= 2^{-p} \left[\sin^2(\omega/2) \prod_{i=1}^{p/2} (\cos\omega - \cos\theta_i)^2 + \cos^2(\omega/2) \prod_{i=1}^{p/2} (\cos\omega - \cos\omega_i)^2 \right]^{-1} \quad (3-88)$$

由此可见, LSP 分析是用 p 个离散频率的分布密度来表示语音信号谱特性的一种方法, 即在语音信号幅度谱较大的地方 LSP 分布较密, 反之较疏。

5) 相邻帧 LSP 参数之间都具有较强的相关性, 便于语音编码时帧间参数的内插。

2. LPC 到 LSP 参数的转换

在进行语音编码时, 要对 LPC 进行量化和内插, 就需要将 LPC 转换为 LSP 参数, 为计算方便, 可将 LSP 参数无关的两个实根去掉, 得到如下多项式:

$$\begin{aligned} P'(z) &= \frac{P(z)}{(1+z^{-1})} = \prod_{i=1}^{p/2} (1 - z^{-1}e^{j\omega_i})(1 - z^{-1}e^{-j\omega_i}) \\ &= \prod_{i=1}^{p/2} (1 - 2\cos\omega_i z^{-1} + z^{-2}) \end{aligned} \quad (3-89)$$

$$\begin{aligned} Q'(z) &= \frac{Q(z)}{(1-z^{-1})} = \prod_{i=1}^{p/2} (1 - z^{-1}e^{j\theta_i})(1 - z^{-1}e^{-j\theta_i}) \\ &= \prod_{i=1}^{p/2} (1 - 2\cos\theta_i z^{-1} + z^{-2}) \end{aligned} \quad (3-90)$$

从 LPC 到 LSP 参数的转换过程, 其实就是求解式 (3-79) 和式 (3-80) 等于零时的 $\cos\omega_i$ 和 $\cos\theta_i$ 的值, 可采用下述几种方法求解:

(1) 代数方程式求解

由式 (3-89) 可知, 等式右边可进一步表示为

$$\begin{aligned} 1 - 2\cos\omega_i z^{-1} + z^{-2} &= 2z^{-1}(0.5z - \cos\omega_i + 0.5z^{-1}) \\ &= 2z^{-1}[0.5(z + z^{-1}) - \cos\omega_i] \end{aligned} \quad (3-91)$$

令 $z = e^{j\omega}$, 则由 $e^{j\omega} = \cos\omega + j\sin\omega$, 可得 $z + z^{-1} = 2\cos\omega = 2x$ 。因此, 式 (3-79) 和式 (3-80) 就是关于 x 的一对 $p/2$ 次代数方程式, 其系数取决于 $a_i (i=1, 2, \dots, p)$, 且 a_i 是已知的, 可以用牛顿迭代法来求解。

(2) 离散傅里叶变换方法

对 $P'(z)$ 和 $Q'(z)$ 的系数求离散傅里叶变换, 得到 $z_k = \exp\left(-\frac{jk\pi}{N}\right)$, ($k=0, 1, \dots, N-1$) 各点的值, 搜索最小值的位置, 即是零点所在。由于除了 0 和 π 之外, 总共有 p 个零点, 而且 $P'(z)$ 和 $Q'(z)$ 的根是相互交替出现的, 因此只要很少的计算量即可解得, 其中 N 的值取 64 ~ 128 就可以。

(3) 切比雪夫多项式求解

用切比雪夫多项式估计 LSP 系数, 可直接在余弦域得到。当 $z = e^{j\omega}$ 时, $P'(z)$ 和 $Q'(z)$ 可写为

$$P'(z) = 2e^{-j\frac{p}{2}\omega} C(x) \quad (3-92)$$

$$Q'(z) = 2e^{-j\frac{p}{2}\theta} C(x) \quad (3-93)$$

此处,

$$C(x) = T_{\frac{p}{2}}(x) + f(1)T_{\frac{p}{2}-1}(x) + f(2)T_{\frac{p}{2}-2}(x) + \dots + f\left(\frac{p}{2}-1\right)T_1(x) + f\left(\frac{p}{2}\right)/2 \quad (3-94)$$

其中, $T_m(x) = \cos mx$ 是 m 阶的切比雪夫多项式; $f(i)$ 是由递推关系计算得到的 $P'(z)$ 和 $Q'(z)$ 的每个系数。由于, $P'(z)$ 和 $Q'(z)$ 是对称和反对称的, 所以每个多项式只计算前 5 个系数即可。用下面的递推关系可得

$$\begin{cases} f_1(i+1) = a_{i+1} + a_{p-i} - f_1(i) \\ f_2(i+1) = a_{i+1} - a_{p-i} + f_2(i) \end{cases} \quad i = 0, 1, \dots, p/2 \quad (3-95)$$

其中, $f_1(0) = f_2(0) = 1.0$ 。

多项式 $C(x)$ 在 $x = \cos \omega$ 时的递推关系是

$$\begin{aligned} & \text{for } k = \frac{p}{2} - 1 \quad \text{to } 1 \\ & \quad \lambda_k = 2x\lambda_{k+1} - \lambda_{k+2} + f\left(\frac{p}{2} - k\right) \\ & \quad \text{end} \\ & C(x) = x\lambda_1 - \lambda_2 + f\left(\frac{p}{2}\right)/2 \end{aligned}$$

其中, 初始值 $\lambda_{\frac{p}{2}} = 1$, $\lambda_{\frac{p}{2}+1} = 0$ 。

(4) 其他方法

将 $0 \sim \pi$ 之间均分为 60 个点, 以这 60 个点的频率值代入式 (3-79) 和式 (3-80), 检查它们的符号变化, 在符号变化的两点之间均分为 4 份, 再将这三个点频率值代入式 (3-79) 和式 (3-80), 符号变化的点即为所求的解。这种方法误差略大, 计算量较大, 但程序实现容易。

3. LSP 参数到 LPC 的转换

LSP 系数被量化和内插后, 应再转换为预测系数 $a_i (i = 1, 2, \dots, p)$ 。已知量化和内插的 LSP 参数 $q_i (i = 1, 2, \dots, p)$, 可用式 (3-79) 和式 (3-80) 来计算 $P'(z)$ 和 $Q'(z)$ 的系数 $p'(i)$ 和 $q'(i)$ 。其中, $p'(i)$ 可通过以下的递推关系获得:

$$\begin{aligned} & \text{for } i = 1 \quad \text{to } p/2 \\ & \quad p'(i) = -2q_{2i-1}p'(i-1) + 2p'(i-2) \\ & \quad \text{for } j = i-1 \quad \text{to } 1 \\ & \quad \quad p'(j) = p'(j) - 2q_{2i-1}p'(j-1) + p'(j-2) \\ & \quad \text{end} \\ & \text{end} \end{aligned}$$

其中, $q_{2i-1} = \cos \omega_{2i-1}$, 初始值 $p'(0) = 1$, $p'(-1) = 0$ 。把上面递推关系中的 q_{2i-1} 替换为 q_{2i} , 就可以得到 $q'(i)$ 。

一旦得出系数 $p'(i)$ 和 $q'(i)$, 就可以得到 $P'(z)$ 和 $Q'(z)$, $P'(z)$ 乘以 $(1+z^{-1})$ 得到 $P(z)$, $Q'(z)$ 乘以 $(1-z^{-1})$ 得到 $Q(z)$, 即

$$\begin{cases} p_1(i) = p'(i) + p'(i-1), i = 1, 2, \dots, p/2 \\ q_1(i) = q'(i) + q'(i-1), i = 1, 2, \dots, p/2 \end{cases} \quad (3-96)$$

最后得到预测系数为

$$a_i = \begin{cases} 0.5p_i(i) + 0.5q_1(i) & i = 1, 2, \dots, p/2 \\ 0.5p_i(p+1-i) - 0.5q_1(p+1-i) & i = p/2 + 1, \dots, p \end{cases} \quad (3-97)$$

3.6.3 实验步骤及要求

1. 实验步骤

运行 MATLAB→新建 m 文件→编写 m 程序→编译并调试。

2. 实验要求

根据 LPC 到 LSP 参数和 LSP 参数到 LPC 的转换原理，编写 MATLAB 函数，并基于测试语音 C3_6_y.wav 比较转换前后的线性预测谱。函数定义如下：

名称：lpctolsf

功能：把 LPC 系数转换成线谱频率 LSF。

调用格式：

lsf = lpctolsf(a)

说明：a 是预测系数，当预测为 p 阶时，要输入 p + 1 个值；lsf 是 LSP 的参数，将得到 p 个数值，奇数项是 $P(z)$ 的根，偶数项是 $Q(z)$ 的根，所得 lsf 是角频率。

名称：lsftolpc

功能：将线谱频率 LSF 转换为 LPC 系数。

调用格式：

a = lsftolpc(lsf)

说明：lsf 是 LSP 的参数，预测为 p 阶时要输入 p 个值，奇数项是 $P(z)$ 的根，偶数项是 $Q(z)$ 的根，lsf 单位是角频率；a 是预测系数，将输出 p + 1 个 a 值。

仿真效果如图 3-17 所示。

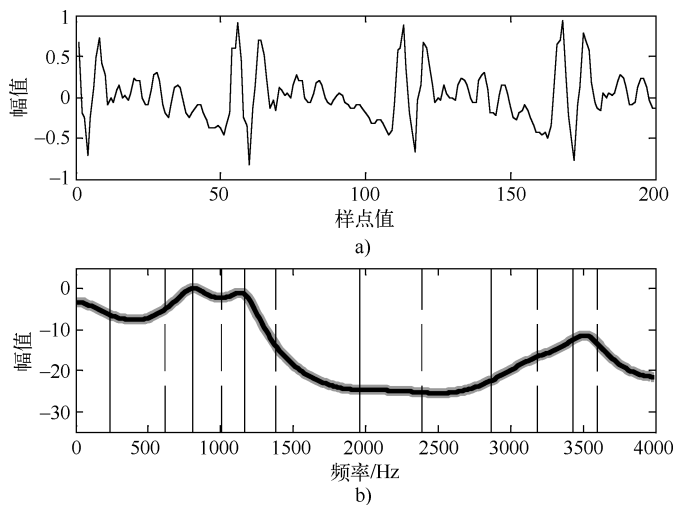


图 3-17 仿真效果图

a) 语音信号 C3_6_y.wav 的一帧波形图 b) 语音信号的 LPC 谱和线谱对还原 LPC 的频谱

3.6.4 思考题

添加适当噪声，观察参数转换的效果。

3.6.5 参考例程

```

%将线谱频率 LSF 转换为 LPC 系数
function a = lsftolpc(lsf)
% 如果线谱频率 lsf 是复数,则返回错误信息
if( ~ isreal(lsf) ),
    error(' Line spectral frequencies must be real.' );
end
% 如果线谱频率 lsf 不在 0 ~ pi 范围,则返回错误信息
if( max(lsf) > pi || min(lsf) < 0 ),
    error(' Line spectral frequencies must be between 0 and pi.' );
end
lsf = lsf( : );           % 将 lsf 转换为列向量
p = length(lsf);        % lsf 阶次为 p
% 用 lsf 形成零点
z = exp(j * lsf);
rP = z(1:2:end);        % 把奇次 z(1)、z(3)到 z(p-1) 赋给 rP
rQ = z(2:2:end);        % 把偶次 z(2)、z(4)到 z(p) 赋给 rQ
% 考虑共轭复根
rQ = [rQ; conj(rQ)];    % 把 rQ 的共轭复根赋上
rP = [rP; conj(rP)];    % 把 rP 的共轭复根赋上
% 构成多项式 P 和 Q,注意必须是实系数
Q = poly(rQ);
P = poly(rP);
% 考虑 z = 1 和 z = -1 以形成对称和反对称多项式
if rem(p,2),
% 如果是奇数阶次,则 z = +1 和 z = -1 都是 Q1(z) 的根
    Q1 = conv(Q,[1 0 -1]);
    P1 = P;
else
% 如果是偶数阶次,z = -1 是对称多项式 P1(z) 的根,z = 1 是反对称多项式 Q1(z) 的根
    Q1 = conv(Q,[1 -1]);
    P1 = conv(P,[1 1]);
end
% 按式(4-5-8)由 P1 和 Q1 求解 LPC 系数
a = .5 * (P1 + Q1);
a(end) = [];           % 最后一个系数是 0,不返回

```