# 中国传媒大学

# 电子设计竞赛设计报告


题　　目：　流行音乐的人声去除
姓　　名：　任骋翔（201010013337）
学　　院：　信息工程学院
专　　业：　通信工程
指导教师：　杜伟韬

# 摘要

流行音乐是人们喜爱的一种音乐类型，它一般由主唱的人声和伴奏混缩而成。将流行音乐中主唱的声音去除在诸如卡拉 OK 等领域都有着广泛的应用。本设计用三种方法实现了流行音乐人声的去除，分别是带阻滤波、双声道抵消和盲信号分离法，并对它们各自的优缺点进行了评价。



**关键词：滤波器　　盲信号分离　　短时傅里叶变换　　时频掩蔽**

# 目录

# 1 设计背景与要求

## 1.1 背景

流行音乐是人们喜爱的一种音乐类型。因此将流行音乐中主唱的声音去除在诸如卡拉 OK 等领域都有着广泛的应用。

这个设计中，用作测试的音频文件为近期流行的电视节目“我是歌手”中林志炫现场演唱的《没离开过》片段，长度约 12 秒，采样率 44100Hz，16bit 量化，由于找不到 WAV 版本的原始音频，此音频 192kbps 的 MP3 文件格式转换得到。

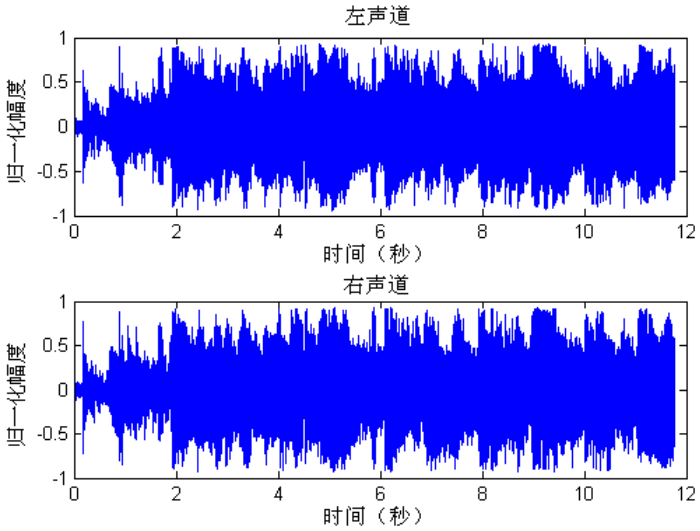



图 1-1 设计所用的测试音频波形

## 1.2 设计要求及目标功能

1) 设计算法实现滤掉流行音乐中的人声
2) 利用 MATLAB 的滤波器设计工具箱中的滤波器函数，设计 FIR 数字滤波器
3) 在设计中应用短时傅里叶变换（*STFT*）以及语谱图，并理解它们的含义

# 2 方案设计

## 2.1 带阻滤波

最简单的人声去除方法是带阻滤波，人声的频率范围是 300Hz 到 3400Hz。如果用一个带阻滤波器滤掉这一频带的信号，就可以达到人声去除的目的。

考虑到 FIR 滤波器种种优良特性[4]，我们选择它来进行滤波。软件滤波使得极高阶的滤波器可以轻松实现，故在这个设计中，我们将 FIR 滤波器的阶数定为 1000 阶，而阻带设为 300Hz-3400Hz。

这种方法虽然可以抑制人声，但是同频段内的其他乐器声音也被抑制了，所以效果并不是很理想。

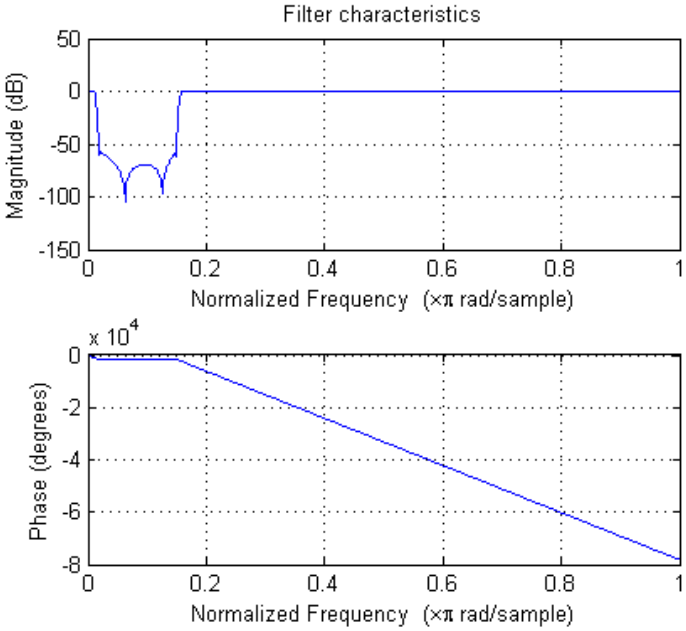



图 2-1 滤波器的幅频特性与相频特性

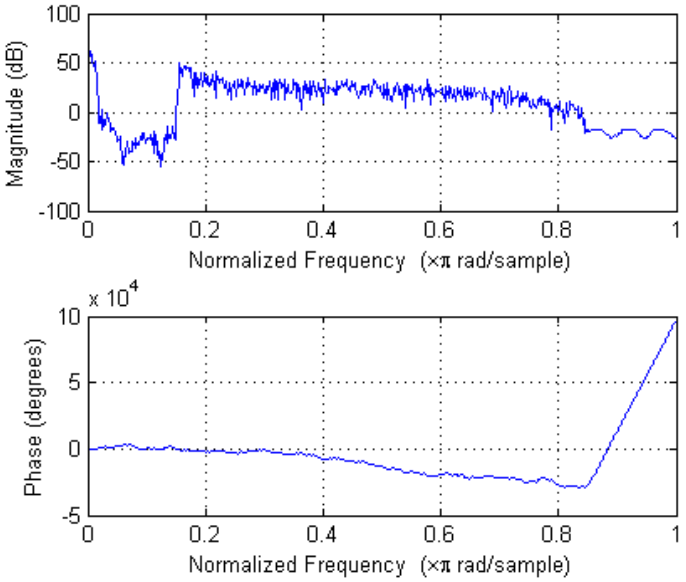



图 2-2 信号滤波后的频谱图

## 2.2 双声道抵消（*Stereo Cancellation*）

双声道抵消，即两个声道的音频在频域上做差。用公式表达为：

$$x_{out}(n) = IDFT\{DFT[x_{left}(n)] - DFT[x_{right}(n)]\}$$

其中 $x_{left}(n), x_{left}(n)$ 分别为原始音频的左右声道。这种方法在大多数情况下都可以适用，因为一般主唱的声音都分布在中间位置，因此频率成分在两个声道当中都是相同的。这种方法比滤波的效果好了很多，不过还会造成一些两个声道上都相同的音乐成分衰减，比如说鼓声和一些低音。

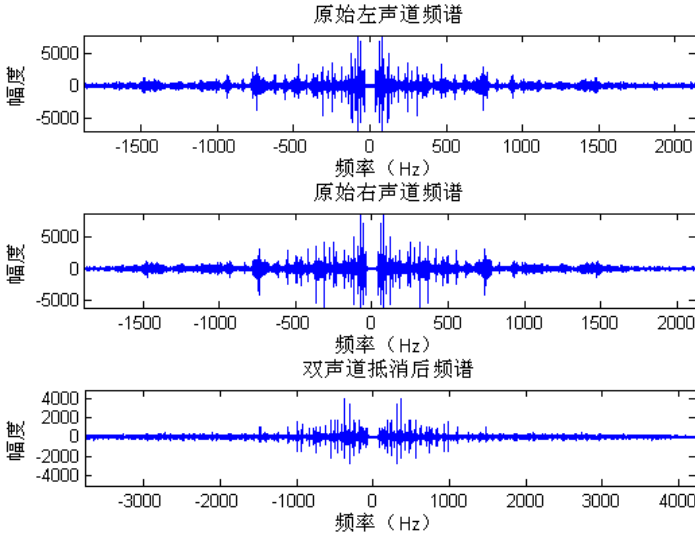



图 2-3 两路信号与抵消后的频谱

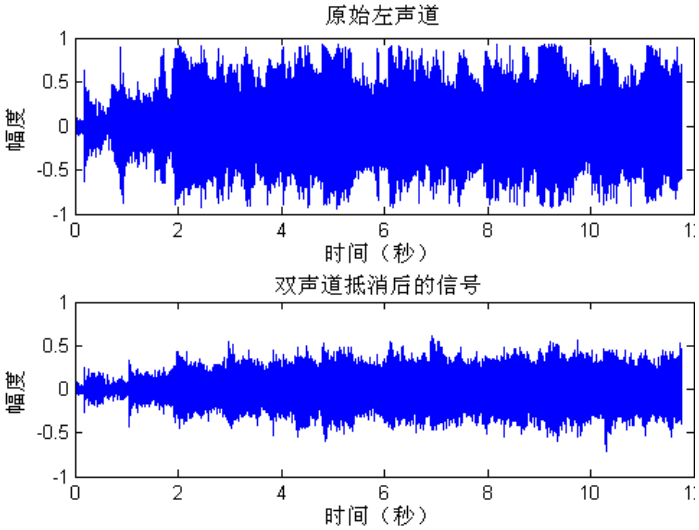



图 2-4 左声道原信号与抵消后的对比

## 2.3 盲信号分离法

### 2.3.1 方法描述

流行音乐一般是用混缩（*mix*）的方法制作的，缩混也就是混音，在音乐的后期制作中，把伴奏和人声等各个音轨进行后期的效果处理，调节音量然后最终缩混导出一个完整的音乐文件。我们在听混缩后的音乐时，大多数时间可以分辨出这几个音轨。

盲信号分离指的是从多个观测到的混合信号中分析出没有观测的原始信号。盲信号的“盲”字强调了两点：1)原始信号并不知道；2)对于信号混合的方法也不知道。[3]

本设计需要做的是将人声信号从混缩当中分离出来，即音频信号盲分离(*Audio Blind Separation*)。我们令混缩后的两路音频信号为 $\mathrm{out}_L[k], \mathrm{out}_R[k]$，而混缩之前的信号表示为 $in_i^L[k], in_i^R[k]$，同时令提取出来的信号表示为 $s_i^L[k], s_i^L[k]$，那么有:

$$\begin{pmatrix} \mathrm{out}_L[k] \\ \mathrm{out}_R[k] \end{pmatrix} = \begin{pmatrix} \sum_i in_i^L[k] \\ \sum_i in_i^R[k] \end{pmatrix}$$

我们希望得到：

$$\begin{pmatrix} s_i^L[k] \\ s_i^R[k] \end{pmatrix} \approx \begin{pmatrix} in_i^L[k] \\ in_i^L[k] \end{pmatrix}$$

设计将用到时频掩蔽 TFM（*Time-Frequency Masking*）算法来生成目标信号。

### 2.3.2 短时傅里叶变换（*STFT*）

本设计通过对原始信号的短时傅里叶变换来，以便后续的时频分析。

首先，对于混缩后的音频（即我们分析的原始音频），以左声道为例，设分帧数为 $P$，每一帧的点数为 $N$，帧叠为 $M$，分帧之后我们会得到以下矩阵（每一行代表一帧）：

$$\begin{pmatrix} out_L[0]\ldots out_L[N-1] \\ out_L[P-M]\ldots out_L[(P-M)+N-1] \\ \vdots \\ out_L[P-M]\ldots out_L[(P-M)+N-1] \end{pmatrix}$$

每一帧都各自乘以窗函数，防止频谱泄露，然后做离散傅里叶变换（DFT）运算。由于 DFT 具有对称性，所以只取前 $N/2+1$ 个系数，得到以下矩阵[1]：

$$\begin{pmatrix} DFT_0(out_L)[0]\ldots DFT_0(out_L)[N/2] \\ DFT_1(out_L)[0]\ldots DFT_1(out_L)[N/2] \\ \vdots \\ DFT_{P-1}(out_L)[0]\ldots DFT_{P-1}(out_L)[N/2] \end{pmatrix}$$

对于右声道的处理同理。在本设计中，每一帧的 DFT 点数 $N=8192$ 点，帧叠 $M=6144$ 点（$3N/4$），即帧间的偏移为 2048 点。对于本设

计分析的音频片段，帧数 $P=250$。此外，在做 STFT 时，本设计选用的窗函数是汉明窗。根据 STFT 的结果我们画出左右两个声道的语谱图，如图 2-5 所示：

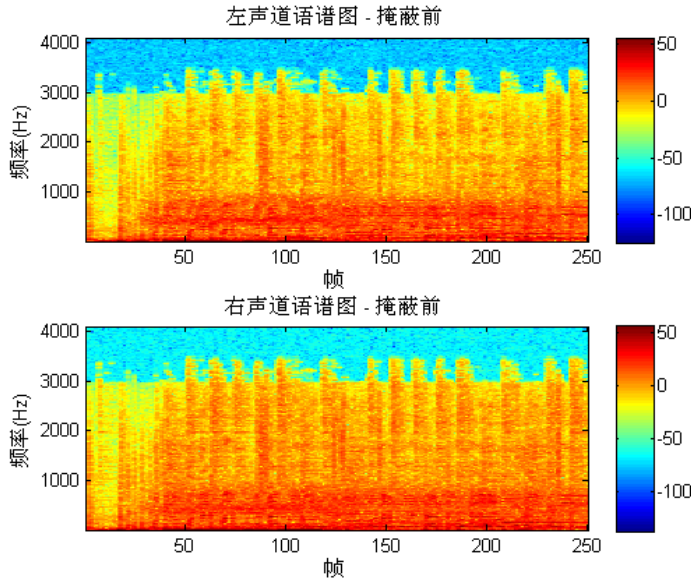



图 2-5 原始音频两个声道的语谱图

### 2.3.3 音轨识别

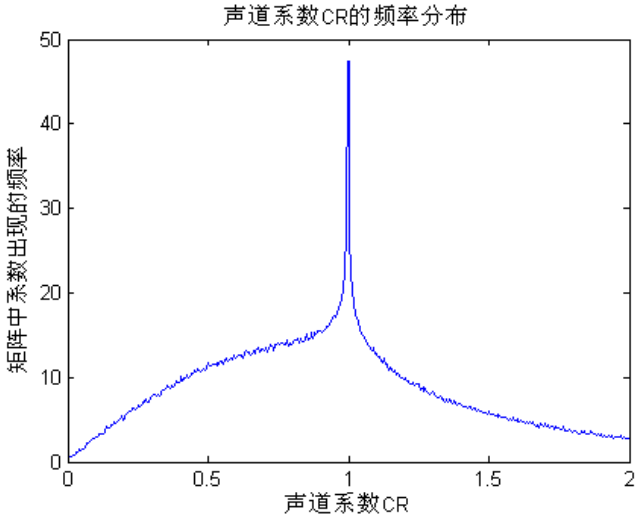
一旦我们得到了两个声道的语谱图，接下来要做的就是比较两个声道的相似性。我们的目的是获取两个声道共有的音轨，即人声轨。我们引入声道系数 CR，令第 $p$ 帧第 $f$ 个频率点的声道系数为左右声道 DTF 幅值（不为零时）之比的模：

$$CR(p,f)=\left|\frac{DFT_p(out_i^L)[f]}{DFT_p(out_i^R)[f]}\right|$$

因为人声轨实为单声道，在两个声道都相同，故对于我们需要的分离的信号有：

$$\frac{DFT_p(s_i^R)[f]}{DFT_p(s_i^L)[f]}=\frac{DFT_p(in_i^R)[f]}{DFT_p(in_i^L)[f]}=const,\forall f\in 0\ldots\frac{N}{2}$$

通过计算，我们得到了每一个频点的声道系数 CR，并绘制出了系数的频率分布图。




通过这张图，我们可以清楚地发现，系数在 1 处的出现频率最高，说明有一个音轨均等地加在了两个声道上。

### 2.3.4 二进制时频掩蔽（*Binary Time-Frequency Mask*）

前面所做的工作已经将单声道轨（人声轨）辨别了出来，下面就需要将这一音轨用算法掩蔽。这里采用的方法为二进制掩蔽法（*Binary Time-Frequency Mask*），即用 0 值代替人声轨上的幅值，而非人声轨的幅值不变，对于人声轨的判定，我们给定一个阈值。二进制时频掩蔽可以表示为：

$$Ma(p,f)=\begin{cases}0, & th_{\min}<CR(p,f)<th_{\max}\\ 1, & else\end{cases}$$

$$\mathrm{DFT}_p(s_i^L)[f]=Ma(p,f)\mathrm{DFT}_p(out_i^L)[f]$$

$$\mathrm{DFT}_p(s_i^R)[f]=Ma(p,f)\mathrm{DFT}_p(out_i^R)[f]$$

对于阈值，需要人为选定，对于林志炫的《没离开过》，我们选择 $th_{\min}=0.68, th_{\max}=1.32$。

掩蔽后的音频的语谱图如图 2-6 所示，我们可以发现一些样点已经被挖掉了（变成了蓝色）。

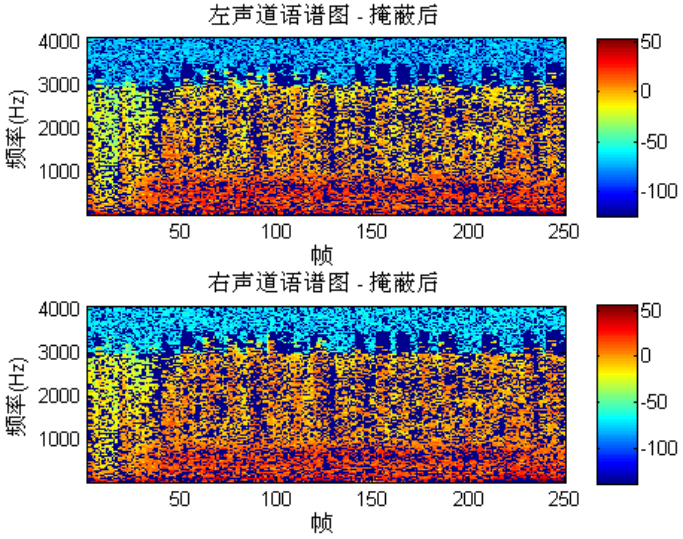



图 2-6 掩蔽后两个声道的语谱图

### 2.3.5 信号恢复——短时傅里叶逆变换（*ISTFT*）

二进制掩蔽之后，就需要对信号在时域进行恢复。对每个声道做短时傅里叶逆变换。变换后得到的两个声道与原声道波形的对比分别如图 2-7 和图 2-8 所示。这是我们得到的就是一个去除了人声的立体声音频了。

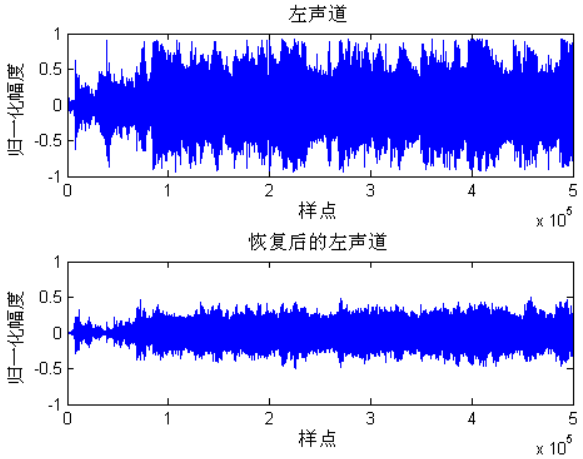



图 2-7 左声道原波形和恢复后波形对比

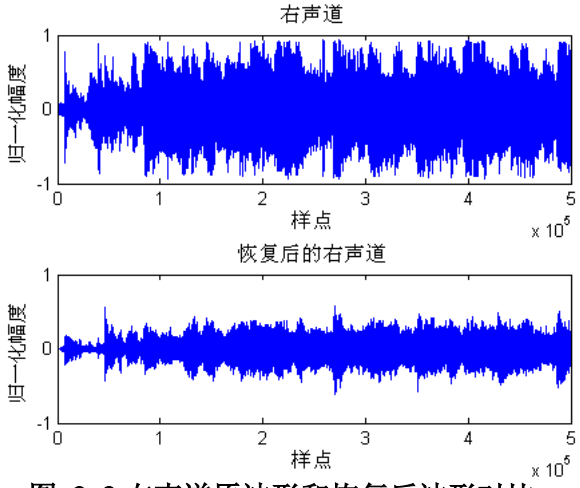



图 2-8 右声道原波形和恢复后波形对比

### 2.3.6 评价

通过试听，用时频掩蔽处理后的音频里人声去除得比前两种方法更加的彻底，并且得到的音频是立体声音频。不过这种方法也有不足，音频听起来有严重的数码感，计算量庞大，处理时间长。

同时这个方法还有很大的优化余地，比如将二进制掩蔽替换为带权值的掩蔽方法等，见[1]。可见对于音乐中人声的去除，盲信号分离法的有着很大的优势。

## 3 后记

这是我第一次以英文文献为基础做的一次设计。我在努力查找中文文献，最多也就前两个老掉牙的方法。无意间还翻到水木上有人的吐槽“音乐信号处理这个方面的资料国内几乎没有，而在国外却很多也都是公开的。你在 Google 上可以找到很多该领域世界著名教授的个人网站，这些教授的人格高尚令人十分敬佩，几乎把他们的所有论文和教材发布在它们的个人主页上供全世界的任何人免费下载……”

这个设计确实很有意思，主要拿第三个盲信号分离说吧，这种提取音轨的方法是 2006 年的一篇论文上提出的，盲信号分离法处理音频是当下研究的热门，自己能看看研究的前沿并且亲自动手试试很有一种自豪感。当然，我这个设计很大程度上参考了两位哥伦比亚大学研究生的课程设计项目[2]。虽然做的是修修补补的工作，但杜老师要求的滤波器设计和 STFT 也算过了手，顺便还练了练文章的排版。

感谢杜老师写的那几篇通俗的教程，对于我们这样 DSP 学得似懂非懂的学生帮助很大，也给我们提供了找 MATLAB 参考代码的好去处。

# 参考文献

[1] MarC Vinyes, Jordi Bonada, Alex Loscos. “Demixing Commercial Music Productionsvia Human -Assisted Time-Frequency Masking” Presented at the Audio Engineering Society, Paris, France, 2006.

[2] Jaime Peretzman,Shrivathsa Bhargav, “ELEN E4810 - Digital Signal Processing – Project: Removing vocals from commercial tracks”, http://shrivathsa.com/fall2007/ELEN_E4810-DSP/project/4810project.htm, 2007

[3] 维基百科中文版，盲信号分离,http://zh.wikipedia.org/wiki/%E7%9B%B2%E4%BF%A1%E5%8F%B7%E5%88%86%E7%A6%BB, 2013.4.6

[4] 杜伟韬, MATLAB 信号处理仿真,http://ecdav.cuc.edu.cn/wiki/index.php/MATLAB_%E4%BF%A1%E5%8F%B7%E5%A4%84%E7%90%86%E4%BB%BF%E7%9C%9F, 2013.4.3

[5] Dan Ellis, A Phase Vocoder in Matlab, http://labrosa.ee.columbia.edu/matlab/pvoc/ 2013.4.3