

**实验报告**

|  |  |
| --- | --- |
| 课程名称： | 信号分析与处理综合实验 |
| 实验名称： | 数学音效处理 |
| 姓名学号： | 戴斯成（3220103149）  冯静怡（3220104119）  汤懿轩（3220103872）  何翌（20171234） |
| 指导教师： | 孙晖 |
| 学 院： | 电气工程学院 |

2024 年 6月 13日

1. **实验目的**
2. 加深对数字滤波器基本原理的理解，掌握数字滤波器时域、频域特性的分析方法。
3. 了解数字信号处理技术在音效处理中的应用，实现混响和均衡器的设计。
4. 为了探寻数字音效处理系统可能的发展方向，本实验将在树莓派硬件平台上，通过软硬件结合的方式，实现实时的数字音效处理系统。本实验将会涉及到信号分析与处理中有关数字滤波器的内容，不仅有助于加深对数字滤波器编程实现方式的理解，还能提高理论与实际相结合的能力和兴趣。
5. **主要仪器设备与软件**
6. 树莓派（ Raspberry Pi )5套件1套（主板、外壳、带开关电源适配器、散热片、有源 USB 集线器、键盘、鼠标、64G Micro SD 卡）。
7. HDMI 接口显示器1台。
8. Python 3版本软件1套。
9. 音箱（带功放）1台。
10. 音源（如手机等）1台。
11. **实验原理**
12. **背景知识**

自然界的声音是一个随时间变化的连续信号，人耳区别声音的不同依靠音调、响度和音色，也就是声音的3个要素。

**音调**是指声音的调子，反映人耳对声音频率的感受，由发声体振动的频率决定，振动频率越高，发出声音的音调越高。

**响度**是人耳对声音强弱程度的感觉。响度的大小不仅与发声体的振幅有关，还与距发声体的远近有关。我们常用音量来描述响度的级别，以分贝为单位。在这种尺度中， 0dB 约为人耳能探测到的最弱的声音，一般的话音约为60dB，而使人耳感到疼痛的声音大约为140dB。

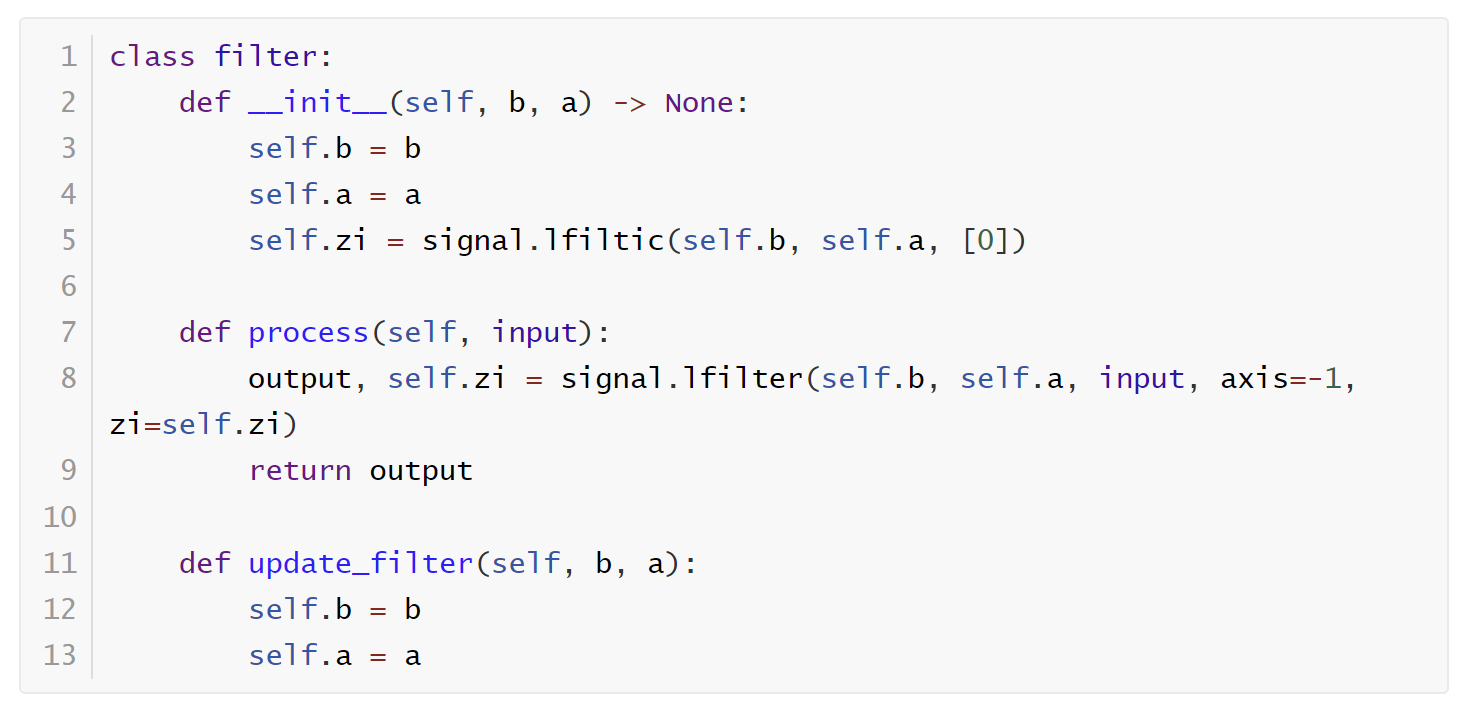
**音色**是人在感觉上区别具有同样响度和音调的两个声音所有不同的特性，由发声体本身的材料、结构等因素决定；反映在音频信号中，由信号的谐波分量即泛音决定。不同的谐波具有不同的幅值和相位偏移，由此产生各种音色效果，谐波分量越丰富，音色就越有明亮感和穿透力。

数字音频技术是指把模拟信号通过采样、量化和编码转换成数字信号，然后进行存储、传输和处理，在重放时再还原为模拟的音频信号。数字音效处理是指通过数字化的手段，经数字信号处理器处理数字音频信号。随着数字信号处理技术的发展以及人们对音乐欣赏的要求越来越高，数字音效处理已经成为数字音频技术中最为活跃的领域之一。

数字音效处理包含音频控制和音频变换两大方面，前者是指音频的动态范围控制，使其效果更能被人耳所接受，典型的如去噪、压限等；后者则涉及音频的时域或频域变换，产生混响、回声、均衡等特殊效果。

1. **滤波器设计**

各个基础滤波器的设计都可以基于scipy.signal库，应用其中的butter等iir滤波器方法获得系统函数的分子与分母的系数列表b和a，最后通过lfilter方法进行数据处理。基本代码处理如下图所示



根据系统函数建立滤波器对象filter。输入数据通过filter类下的process函数对输入数据进行处理。如果在音频处理过程中需要更新系统函数，则通过update\_filter(b,a)函数更新滤波器系统函数，从而达到更新参数的目的。

1. **树莓派蓝牙连接与音频并行处理**

使用Linux音频架构PulseAudio对蓝牙音频读入数据进行处理。

**PulseAudio** 是一个跨平台的声音服务器，主要用于 Linux 操作系统。它的设计目的是在桌面和移动操作系统上提供高级音频功能。PulseAudio 作为 ALSA (Advanced Linux Sound Architecture) 的一个抽象层，支持多种音频后端，如 OSS (Open Sound System) 和 JACK (Jack Audio Connection Kit)。

PulseAudio 的一些主要功能和特点包括：

**网络透明性**：支持音频流通过网络传输，使得声音可以在一台计算机上播放，但实际输出到网络上的另一台计算机上。

**模块化设计**：通过加载不同的模块来扩展其功能，例如用于处理不同的音频设备，或用于音频流的路由和混合。

**支持多种音频源和接收设备**：能够同时处理来自多个音频源的音频流，并发送到多个音频接收设备。

PulseAudio中有如下概念：

**Sink（接收端）：**Sink 是 PulseAudio 中的音频接收端，负责音频输出。它通常表示一个物理设备，如扬声器、耳机或 HDMI 音频输出等。Sink 处理来自应用程序的音频数据，并将其输出到相应的物理设备。

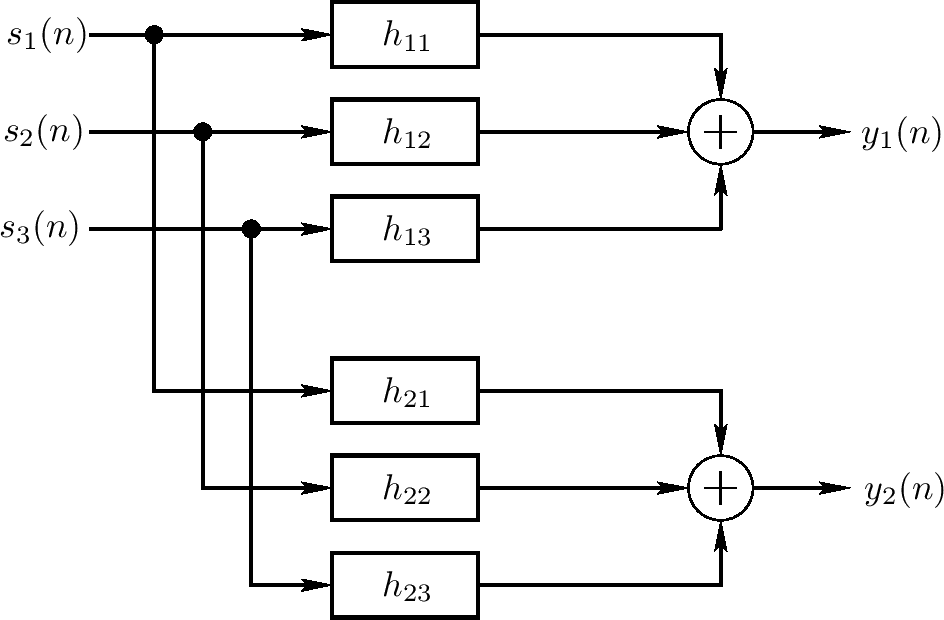
**Source（源）：**Source 是 PulseAudio 中的音频源端，负责音频输入。它通常表示一个物理设备，如麦克风、音频接口的输入端口等。Source 从物理设备获取音频数据，并将其提供给应用程序。

**Sink Input（接收输入）：**Sink Input 是发送到 Sink 的音频流。它代表一个应用程序或音频源正在向 Sink 传输音频数据。Sink Input将应用程序的音频数据发送到 Sink 进行输出。

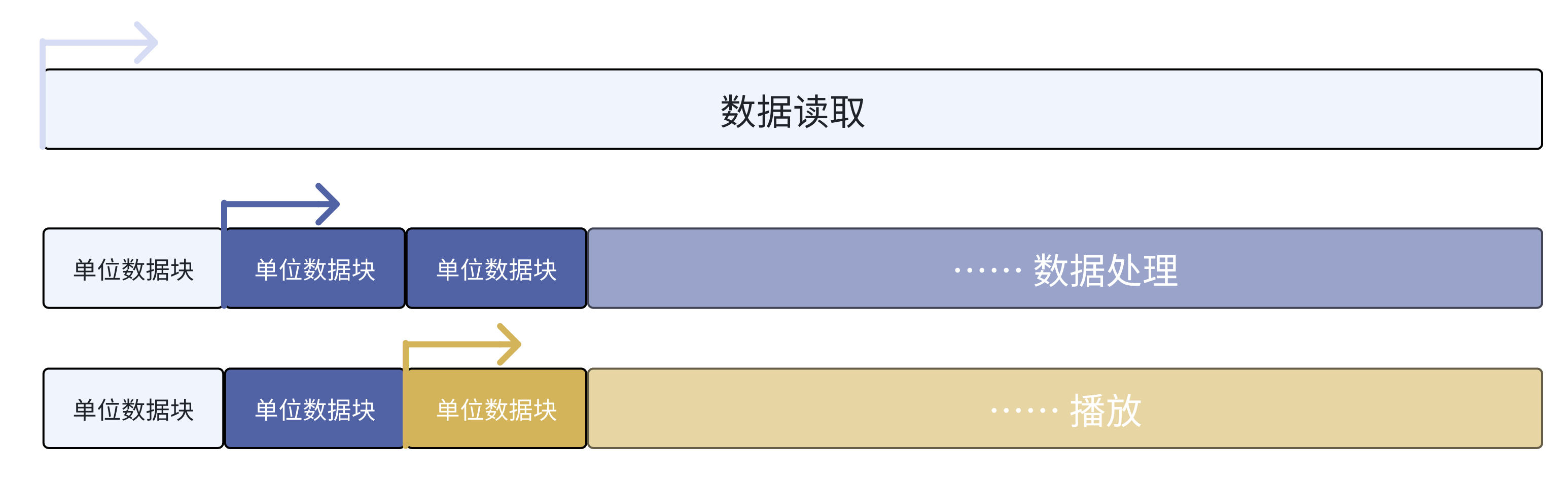
**Source Output（源输出）：**Source Output 是从 Source 获取的音频流。它代表一个应用程序或音频目标从 Source 获取音频数据。Source Output 从 Source 获取音频数据并将其传输给应用程序。

在本实验中，我们主要使用pulseAudio中虚拟设备模块(module-null-sink)，对蓝牙音频输入（sink-input）重定向至新创建的null-sink之中，从而能够通过python中的pyaudio模块得到蓝牙的数据。

1. **实验内容与步骤**
2. **混响音效**

****

1. **均衡器设计**
2. 利用 Python 在树莓派中编程设计低通、高通、带通滤波器，利用 Python 在树莓派中编程设计不同频段的一阶、二阶滤波器，并将低通、高通、带通滤波器并联形成图形均衡器，调节参数，比较幅频特性。读入音频信号，调整各滤波器参数，将均衡器处理后的信号输出，比较处理前后的音效。
3. 设计尖峰滤波器和陷波滤波器，利用 Python 在树莓派中编程设计级联结构参数均衡器，调节中心频率和增益，比较幅频特性。读入音频信号，调整各滤波器参数，将均衡器处理后的信号输出，比较处理前后的音效。
4. **音频数据实时读取、处理、播放**



1. **UI界面设计与功能结合**

（1）环境配置和库安装。因PyQt6对python版本有要求，要想在3.12版本现状，需要修复一个补丁，并在PyCharm中拓展工具PyUIC和QtDesigner。

（2）在QtDesigner中设计各个窗口的界面，并对按键和滑轨加入信号与槽功能。

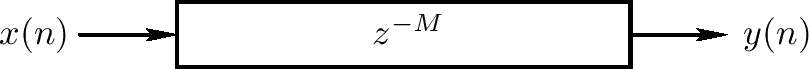
（3）保存设计好的.ui，用PyUIC生成相应的python代码，因其结构冗余，要想之后与后端结合方便，还需重新整理分类。

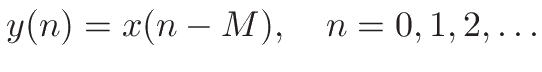
（4）将各个二级面板做入主面板中，利用print确保各按键正常工作。

（5）完成基础功能后，美化完善。

1. **效果预设**
2. **实验数据记录、处理和分析**
3. **混响音效**
4. Delay Line（延迟线）

混响最基本的思想从延迟线的概念中产生。延迟线，即将原有的声音延迟几秒之后和新加入的声音同时播放。理论图如下图所示。

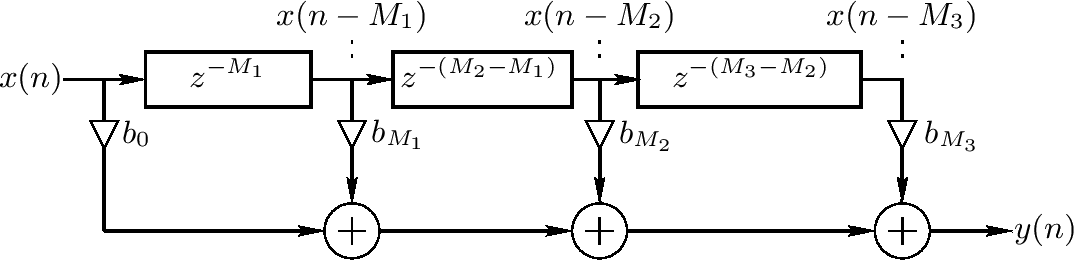




这样子，产生了最基本的回声效果，但并不能产生非常丰富的声音效果。

1. TDL(Tapped Delay Line)

在DelayLine的基础上，我们通过如下方式，获得了更加复杂的回声效果。

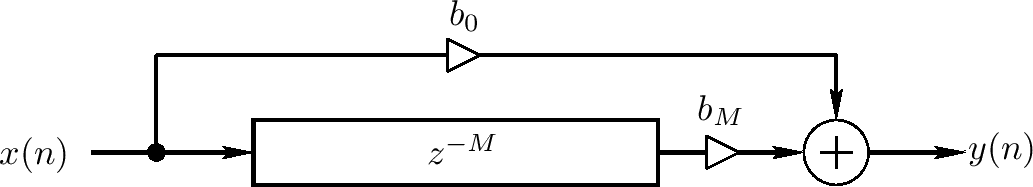


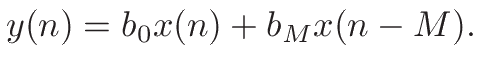
在声音不同的时间点上抽出声音信号，乘上不同的系数，再相加就能够模拟现实生活中有多个回声壁并进行反射的效果。但是，这样子产生的回声数量非常有限。

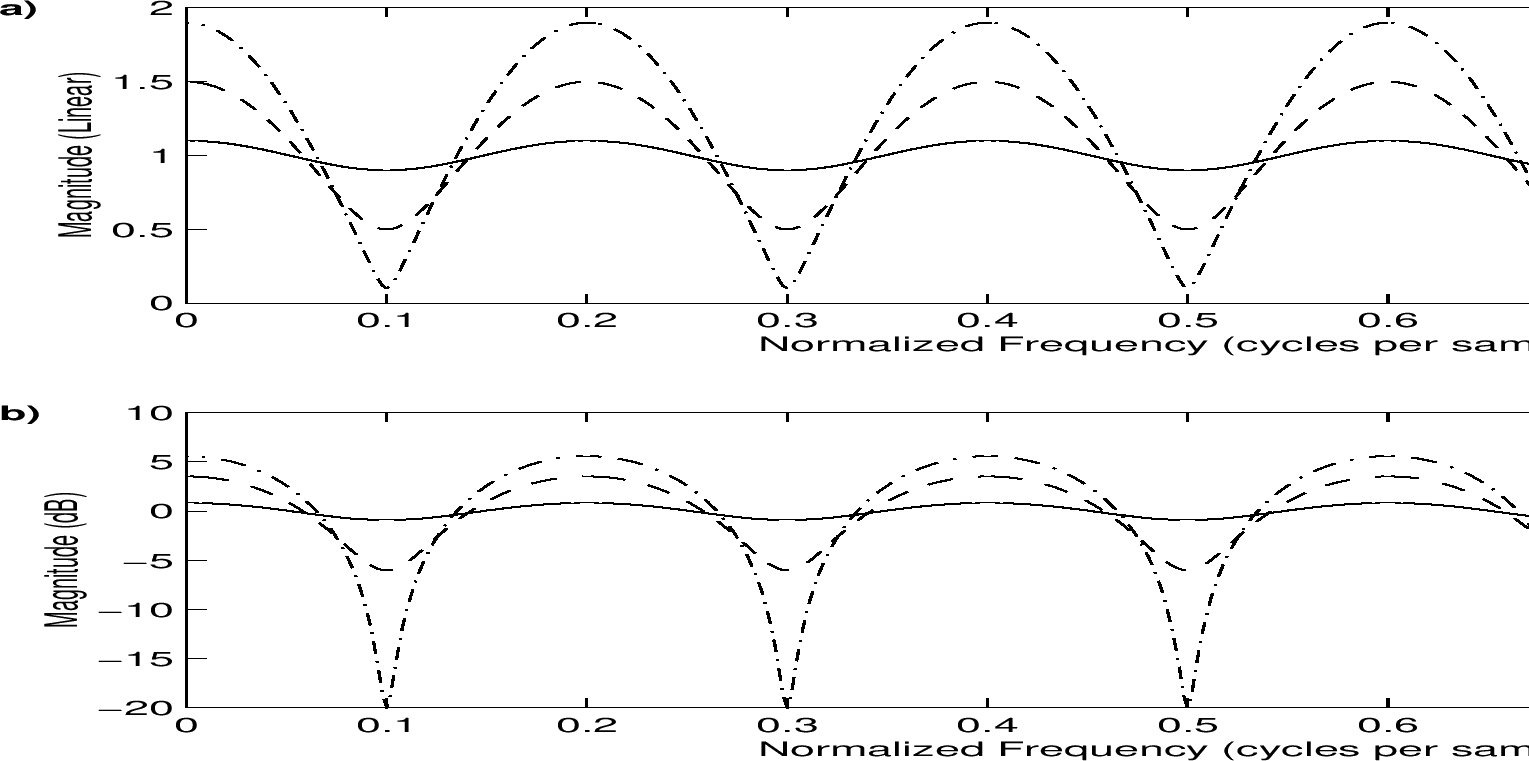
1. Comb Filter

在以上滤波器的基础上，我们引入了作为滤波器最为基本的梳状滤波器。

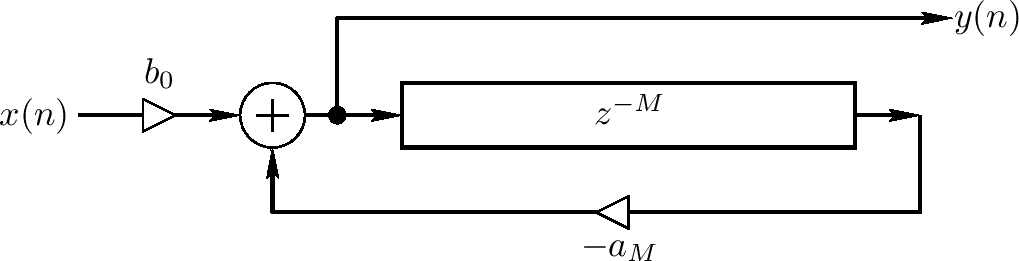
1. 前置梳状滤波器

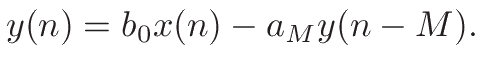


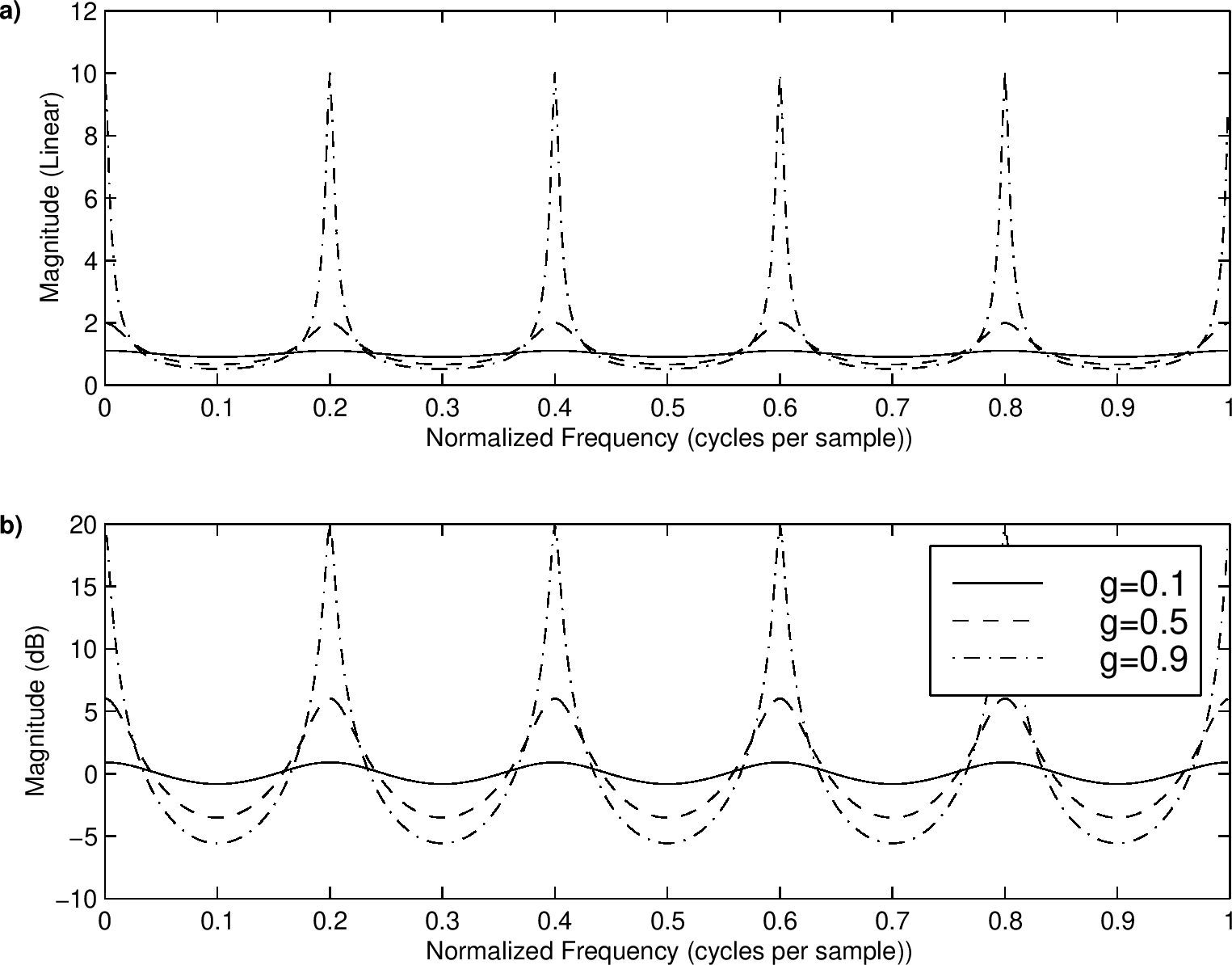




1. 后置梳状滤波器

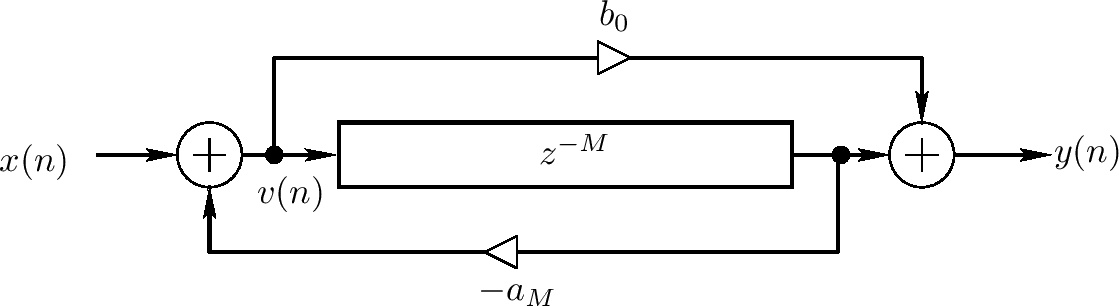


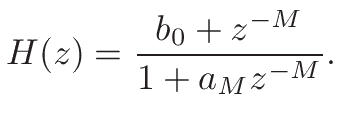


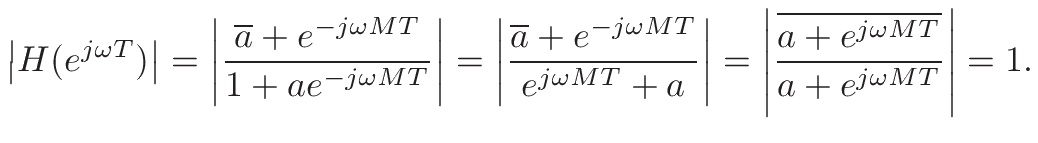


后置梳状滤波器模拟了输出量在经过反射之后重新进入人耳的过程，这样子能大大增加回声的数量从而丰富声音输出的效果。

1. Allpass Filter







全通滤波器通过前置梳状滤波器和后置梳状滤波器的组合，得到一个在全频范围内都为1的滤波器模型。

每个全通滤波器可以将前一级的每个非零输入样本扩展为一个无限的全通脉冲响应。也因此，Schroeder全通滤波器有时被称为脉冲扩展器或脉冲扩散器。虽然这不是物理上的漫反射模型，但单个反射会被扩展为多个反射，这在质量上符合预期效果。理论上，全通滤波器是完全“无色”的，但实际上，当它们的脉冲响应非常短（少于10毫秒）时，才真正无色。如果脉冲响应时间更长，它们听起来会像反馈梳状滤波器，即会引入类似的频率响应特性。

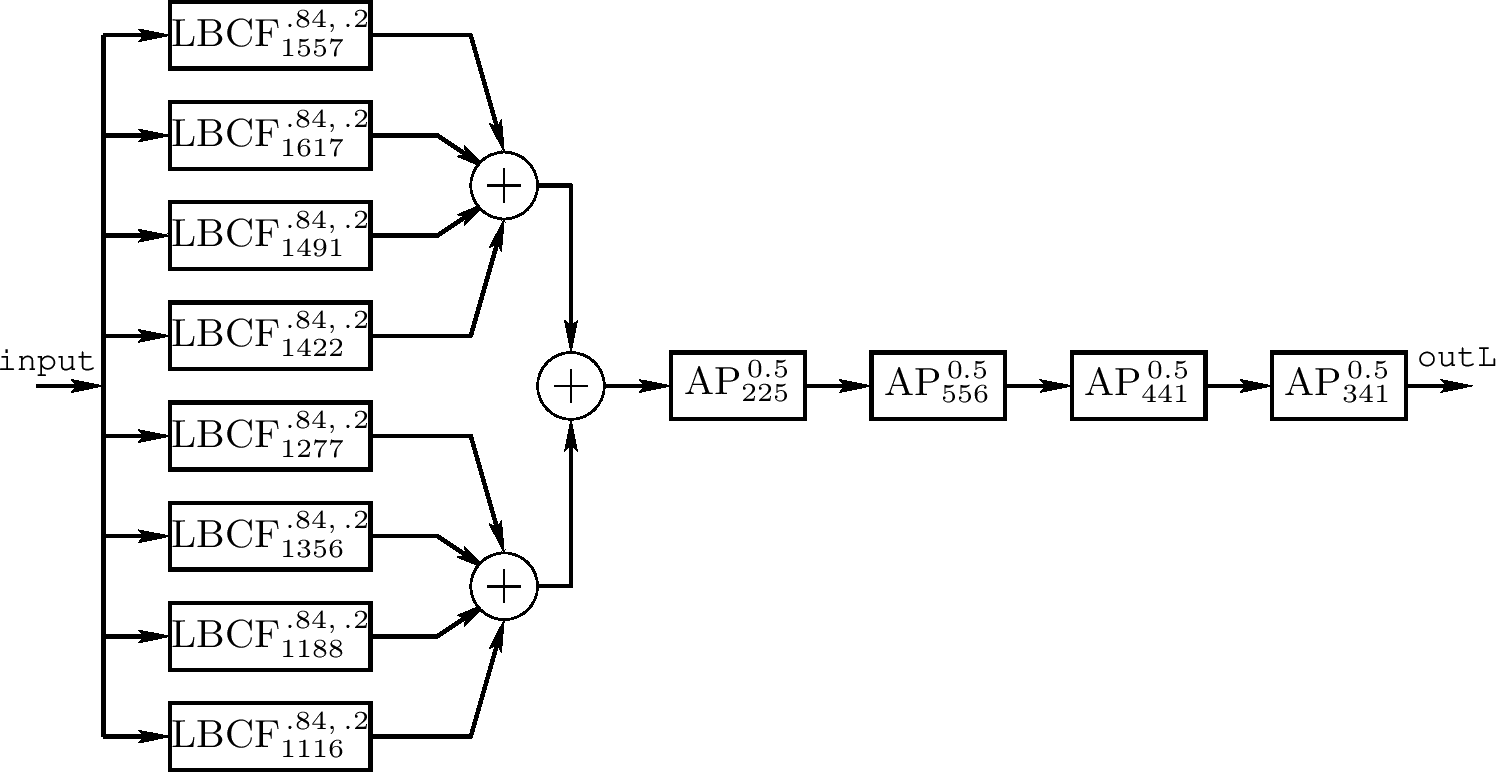
对于稳态音调（如连续的正弦波），全通滤波器在所有频率上提供相同的增益，不会像梳状滤波器那样在特定频率上有峰值和谷值。

Schroeder 提出了一种设置全通滤波器的方法，使用不同长度的延迟线，并确保**这些长度是互质的**（即没有公因数），从而避免任何周期性的共振。具体的长度按下式分布：

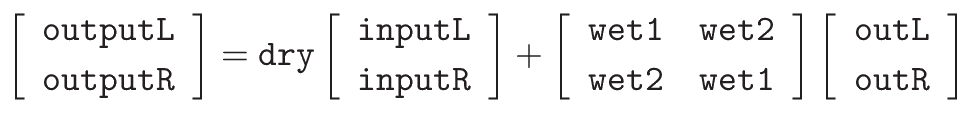
每个滤波器的延迟时间是前一个的三分之一，最长的约为100毫秒。这种配置可以确保每个全通滤波器的脉冲响应时间逐步缩短，并且通过串联5个这样的全通滤波器，可以达到每秒大约810次回响的密度，这接近理想的每秒1000次回响密度。这个设计保证了混响的平滑和自然，使得混响效果更加逼真。

在全通滤波器的基础上，添加并列的梳状滤波器，能够模拟一对平行墙壁的效果，提供一种符合心理声学上的波动。

设计混响模型，本实验中模型参考Freeverb混响模型中的参数设置：



以上为左声道的模型，右声道在左声道的基础上增加23个延迟点从而增加空间感。



整体效果如上所示，dry设定了原声的比例，wet设置了混响后声音的比例强度。

上式中LBCF为Lowpass-Feedback-Comb-Filter，由下面的低通滤波器，

和反馈梳状滤波器组成，

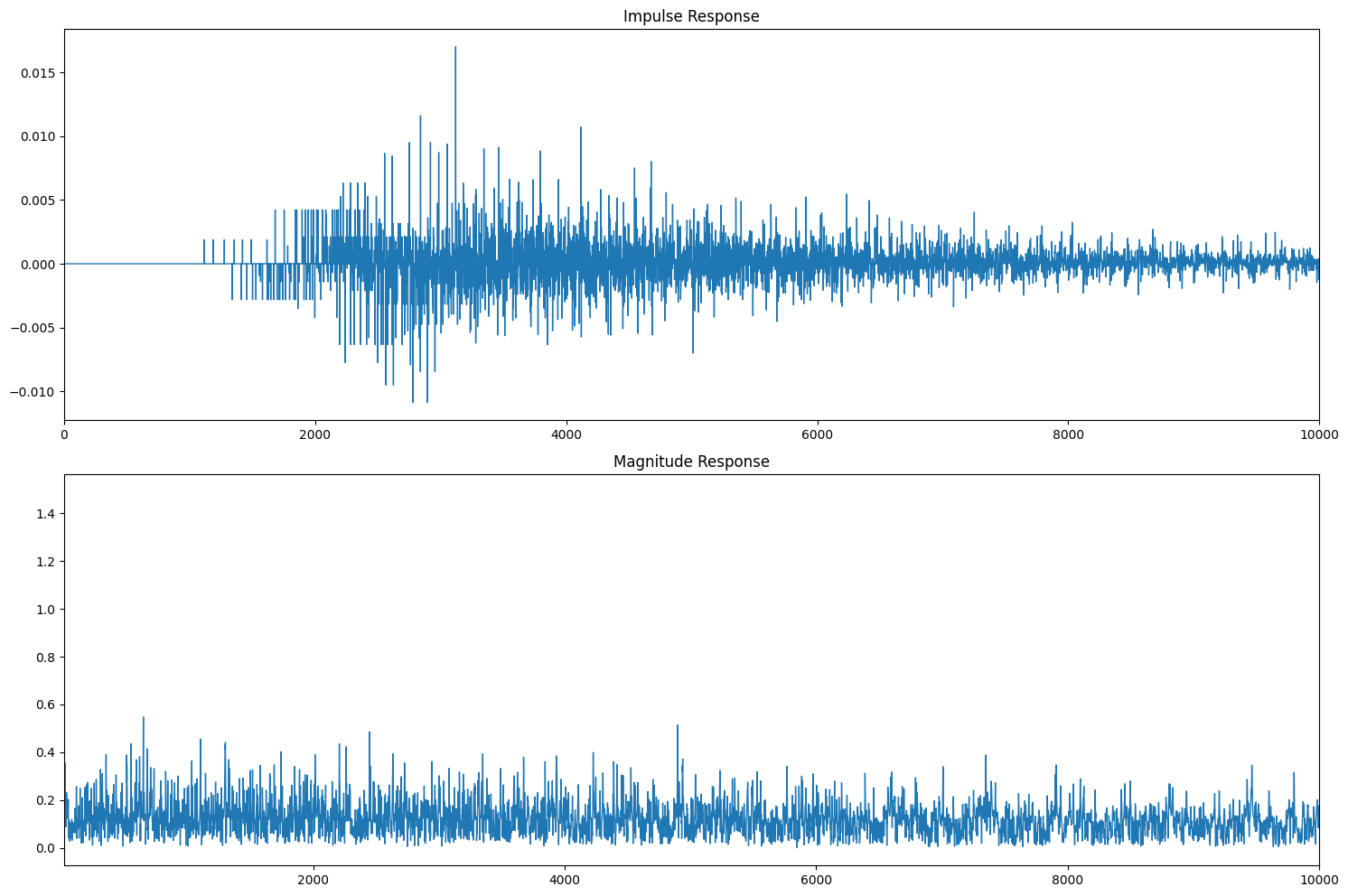
其中在 Freeverb 的 comb 部分中，初始“阻尼”参数d设置为：

低通缩放因子f在源代码中称为反馈，初始设置为：

增加房间大小参数（通常通过 GUI 滑块控制）会增加f，从而增加混响时间。

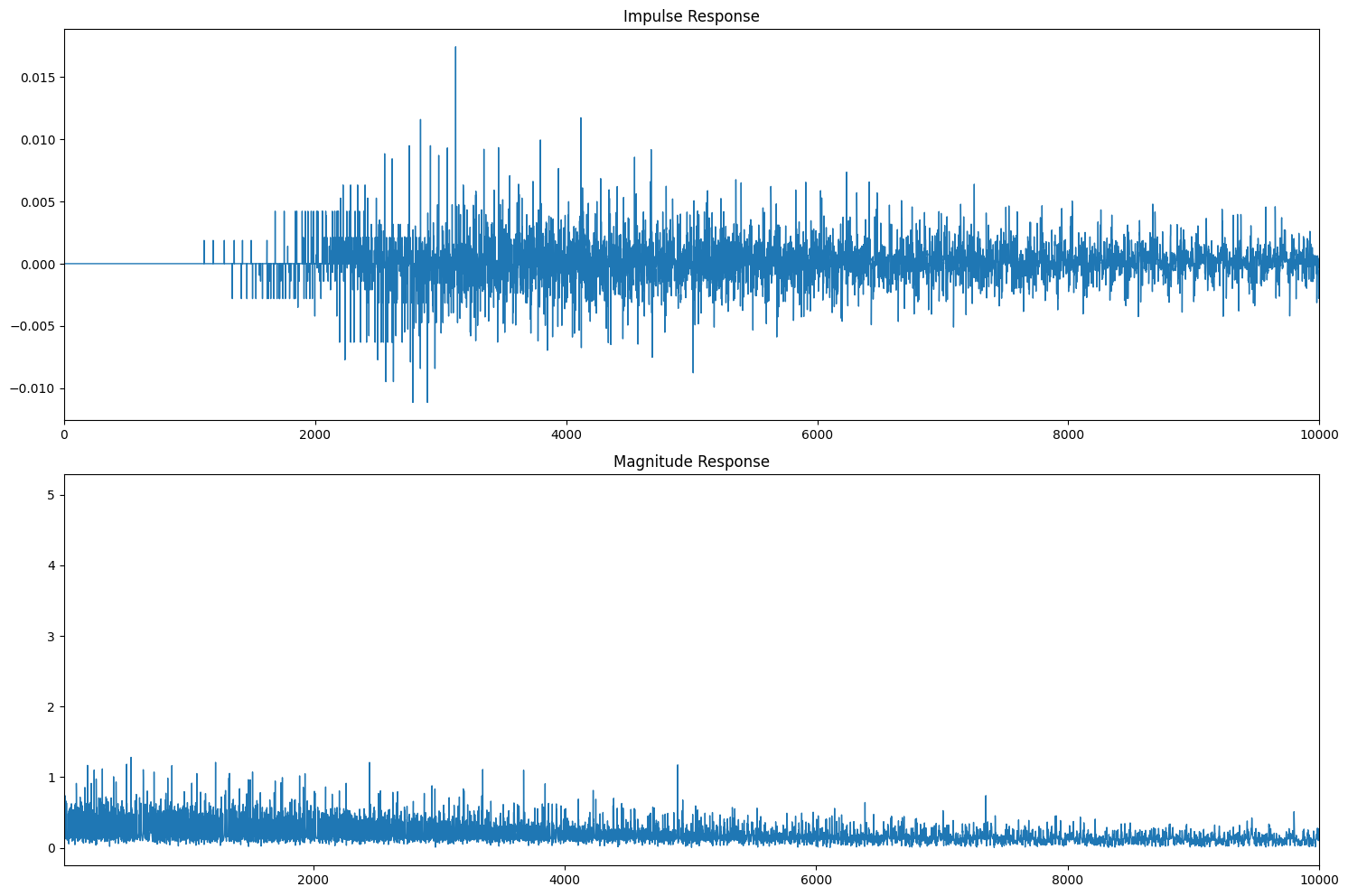
为了保证直流稳定性，要求 ( f < 1 )，因此房间大小必须小于 1.0714，GUI 滑块的最大值通常设为 1（即 ( f = 0.98 )）。

默认参数下，单位冲激响应如下图所示：



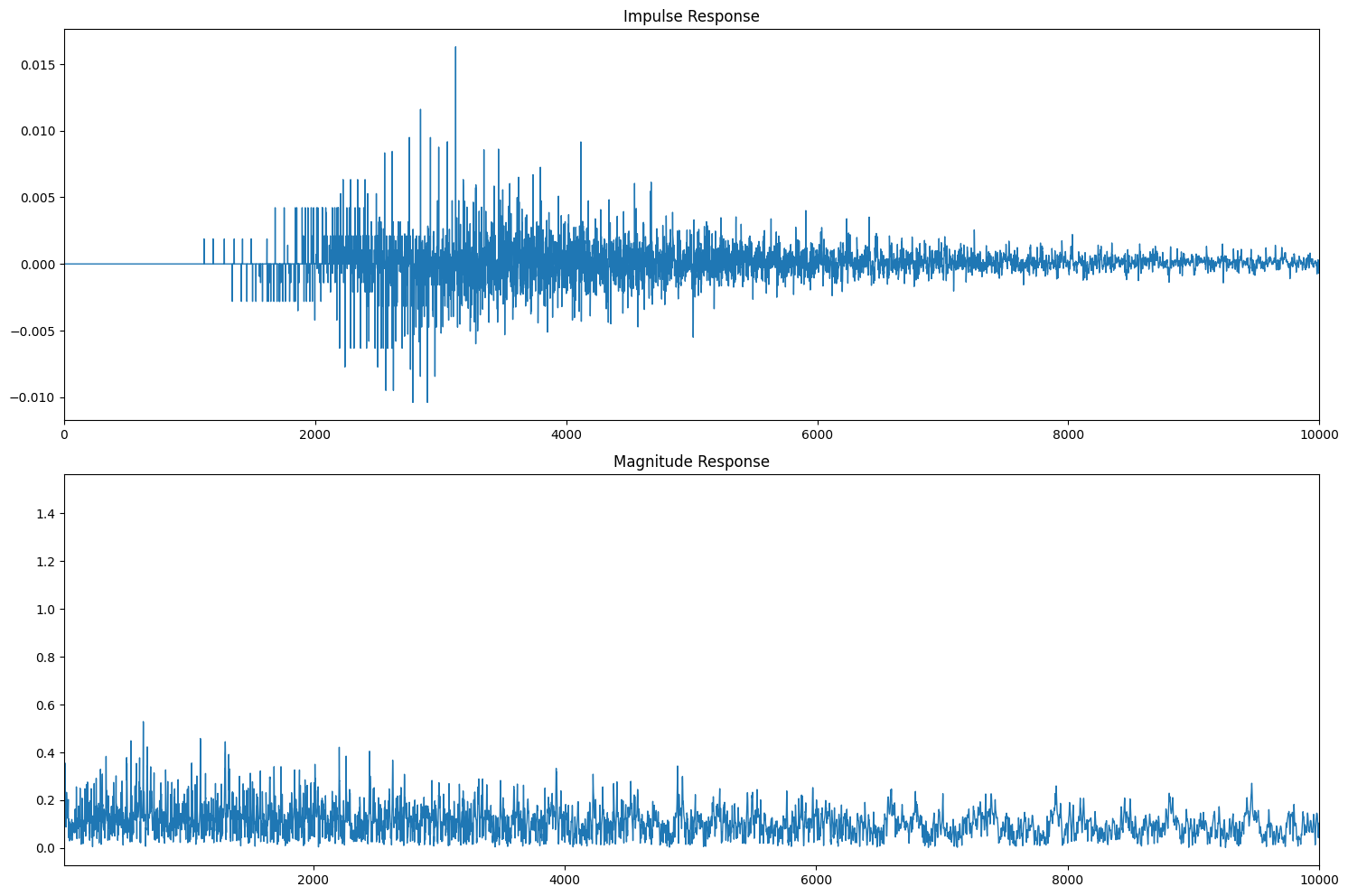
dry=0,damp=0.5,wet=0.33,roomsize=0.5

调整roomsize为0.99，单位冲激响应基本不变，但是幅度响应有一定变化，，可见roomsize改变了声音的延迟效果：



roomsize=0.99

调整damp为0.99，信号衰减很快：

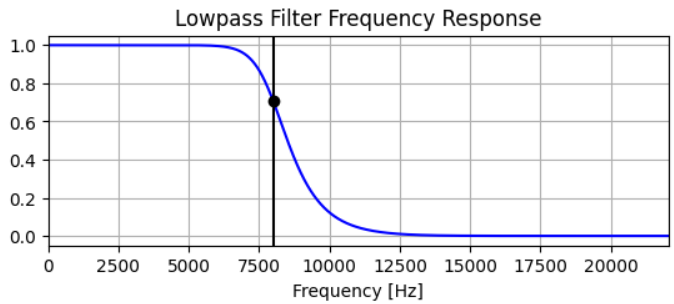


damp=0.99

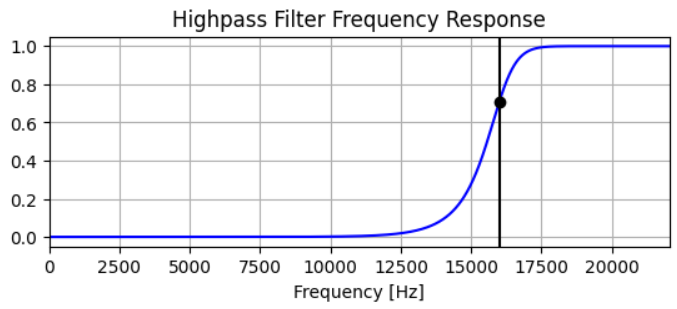
1. **均衡器设计**

我们的所有均衡器相关的滤波器均写在newEQ.py文件中。

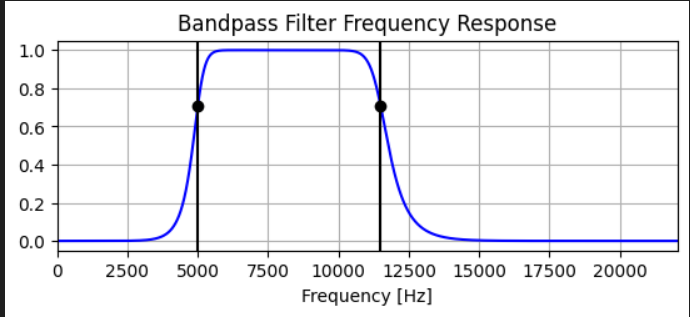
1. 图形均衡器（截止频率固定，但可改变各频段增益）
2. 低通滤波器



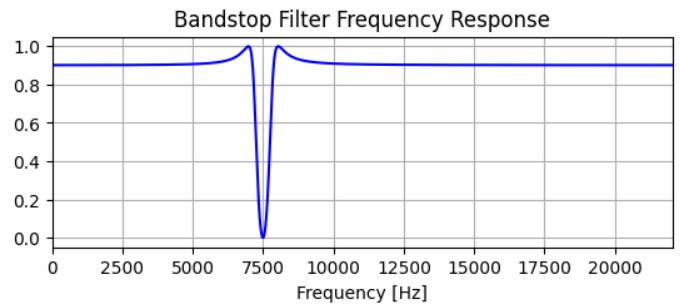
1. 高通滤波器

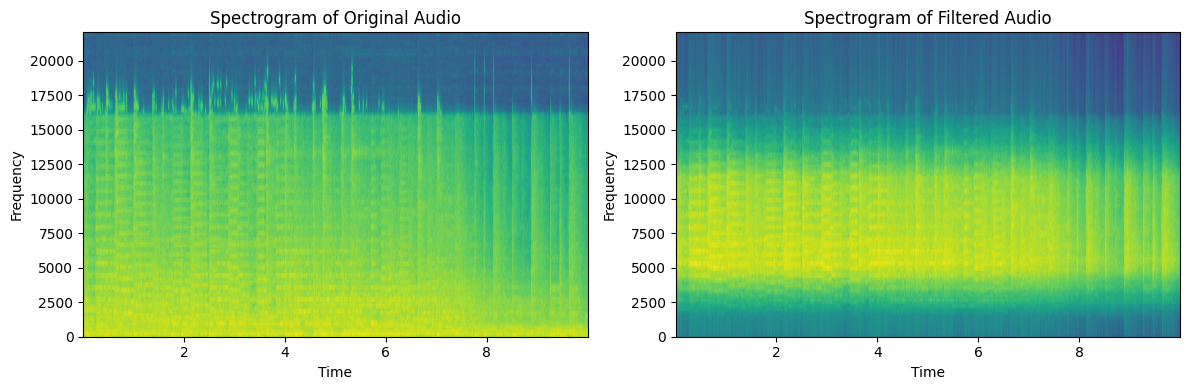


1. 带通滤波器

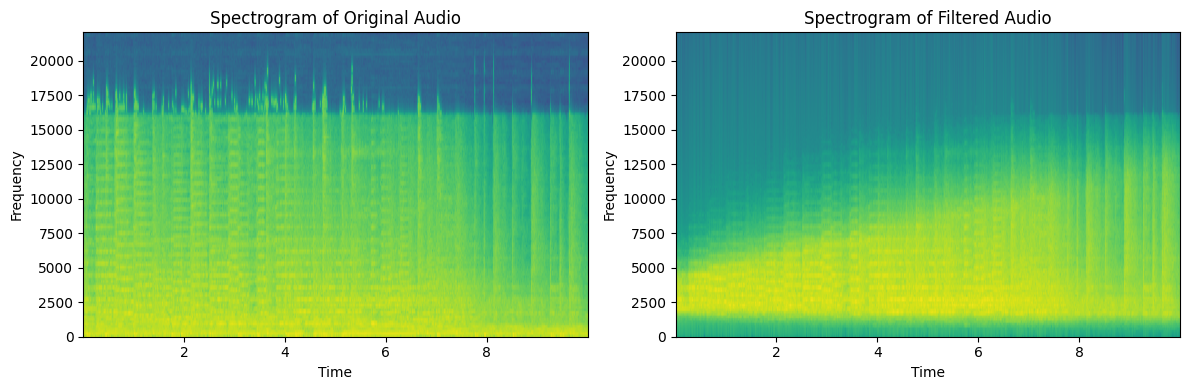


1. 带阻滤波器（最后没有用到，应用了切比雪夫滤波器，其他的都是巴特沃斯滤波器模型）

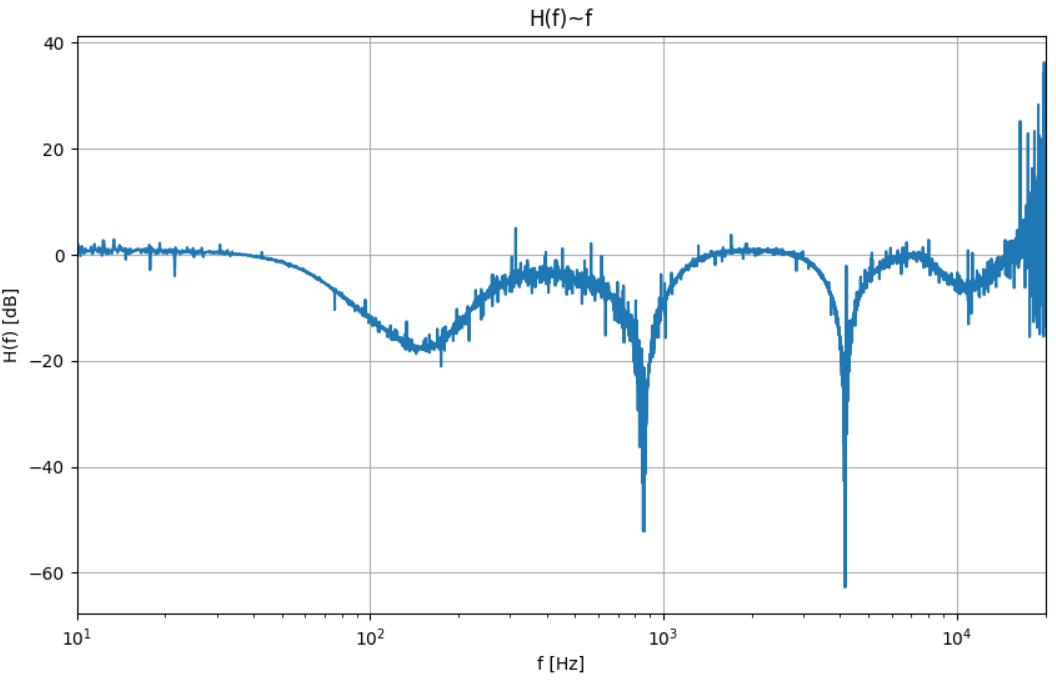


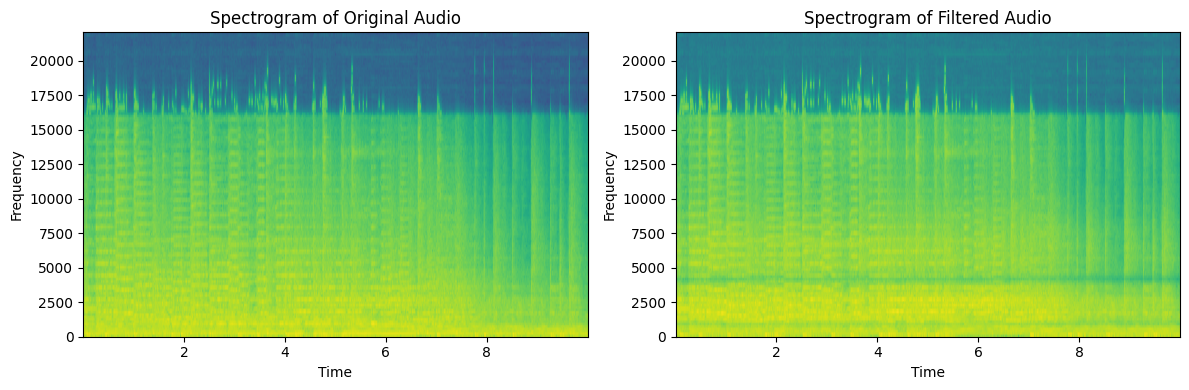
应用带通滤波器后10s的音频频谱图变化如下：  


其中左图是原音频数据，右图是处理后的数据（横坐标为时间，自动坐标为频率，颜色越亮代表幅值越大，深蓝色代表幅值为0）

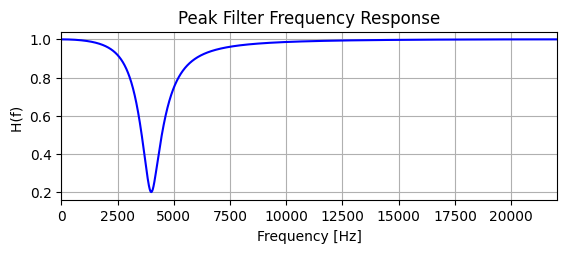
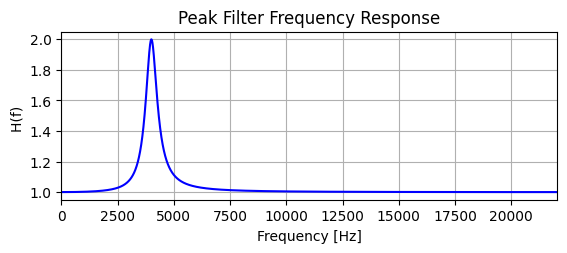
修改newEQ.py中滤波器类的格式，以固定长度chunk=1024为处理单元，增加update函数，方便后续实时更新。实时处理效果如下（频率上限不断增加）：  


组合（多个带通滤波器，与低通、高通滤波器各一个并联）后，获得新的图形均衡器的系统函数曲线（fft（应用滤波器后的音频数据）÷fft（原数据），因此有一定偏差）：

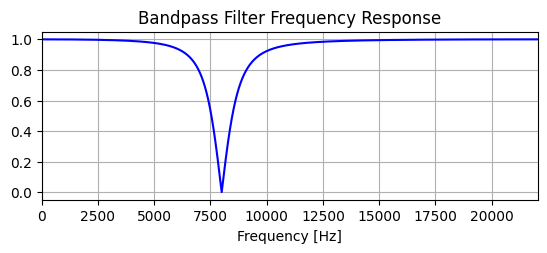


观察处理前后的音频频谱：  


1. 参数均衡器
2. 尖峰滤波器



1. 陷波滤波器



不同于尖峰滤波器，陷波滤波器模型只能是中心频率处增益为0，也可以看成一种特殊的尖峰滤波器。但是要注意尖峰滤波器的增益不能设置为0，因为其系统函数的算法不同于陷波滤波器。

组合（最后我们只应用了尖峰均衡器的级联，因为如果增益设置为0.001时可以近似起到陷波滤波器的效果）后，得到参数均衡器。

1. **音频数据实时读取、处理、播放**
2. **UI界面设计与功能结合**
3. UI界面
4. 主面板



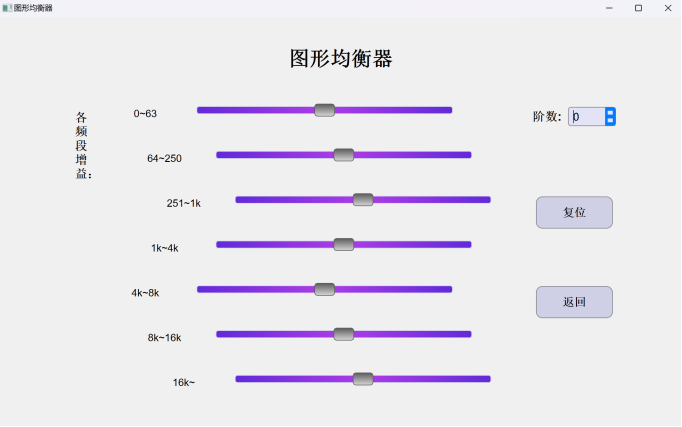
主界面分为两类：音频处理和模式选择。因为用的是蓝牙传输，而不是读取文件，所以设计的播放和暂停按键只是摆设，实质是手机控制播放。模式选择里的三个按键对应后端的三个音频处理模式。

1. 混响面板

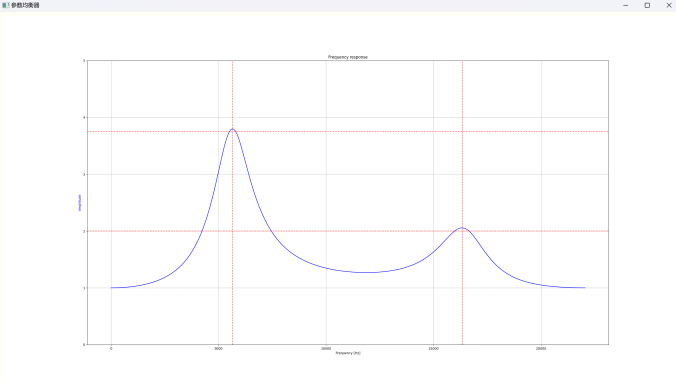


鼠标移动到滑轨按钮区域会有样式变化，移动后滑轨也会有明显变化，当鼠标释放后会返回对应值，并且只有当鼠标控制按钮移动时才会改变数据，避免误触。切换另外两种模式时，滑轨被禁用。

1. 图形均衡器面板



1. 参数均衡器面板



图像根据参数均衡器处理完的数据流，运用了freqz库分割成离散的点，在频域上绘制曲线，比较巧妙的设计是右键撤回。

1. 专属模式面板

将参数写在对应按键里，播放预设模式。

1. 功能结合

最大的难点是如何将pyaudio的stream方法中具有固定callback参数类型的回调函数的参数与外部的数据更新进行结合。我们的思路是首先在mainwindow的主程序中开启音频读取、处理与播放的pulsemodule线程，将多个队列（传递更新参数数据的元组）作为参数传进线程中。其次在pulsemodule中将整个线程封装成一个类，将传递进来的队列作为类的自由变量。这样就相当于self.q始终是刚开始传递进来的队列名，而这个队列就像一个隧道不断为callback回调函数传递主程序中更新的参数数据，从而实现了实时的参数更新与效果改变。

最终我们的效果能够达到改变一个参数正式启用一个模型（三种混响，两种均衡器），并且响应时间基本可以忽略。

1. **效果预设**
2. **总结与展望、心得**

**戴斯成：**

总的来说，我们这个实验完成度还是非常高的，基本上完全实现了实验指导书上的基础功能。尤其是混响和并行处理两个环节，我们在学长们的基础上做了很多拓展，并且突破了过去延时八秒的问题。这个问题主要是因为书上的并行处理方式是基于多进程的。首先，进程之间数据并不互通，只能借助队列等方式进行传输，因此为了节约成本，只能像书上那样每个进程都要完成读取、处理、播放2s的数据。这样就会造成一个问题：每个2s都与先前的数据隔离，就会导致处理的音频效果产生失真。我本来想为什么不能将2s缩减到零点几秒，现在答案很明显，因为2s的时候，单位音频段（chunk）较长，人耳可能几乎听不出这些数据缺失产生的影响；但是一旦将单位音频段时间压缩后，这样的失真就会非常明显了。这也证明了这种并行处理的思路是不可行的。因此我们最后采用了上述的三线程（算上UI其实是四线程）的方式，成功解决了过去那种模式的不足（当然这对树莓派的算力要求更高，我们用的是树莓派5）。

当然这次实验由于经验的不足、多数组员python基础薄弱，我们的成果还有许多亟待改进的地方，包括线程安全、编程规范等等。另外我们的并行处理环节是双声道在同一个线程中处理的，如果有更充裕的时间我们也可以将其改进为单声道的模式，或许能更进一步地提高运算速度。其次，我们最初希望实现的音轨分离和基于ai的效果器设计也由于时间关系未能进行，这些都是我们现在这个项目可以进行拓展的地方。

我在这次的小组实验中收获颇多，这是我第一次参与一个较大的项目，更是我第一次做团队项目的组长。一方面我自己的python编码技术获得了质的飞跃，另一方面也对信号处理本身有了更加深入的理解，当然last but not least，和小组成员一起实现了本来我一个人完全无法想象的任务，更是获得了满满的成就感。相信未来面对这种团队大工程，我不再会束手无策、焦头烂额、毫无头绪的迎接它了，而是会有更多的经验和自信去完成。

**汤懿轩：**

最初开始设计UI想着怎么直接怎么来，用pygame这个库做出了最初稿，虽然功能基本能实现，但是界面很丑，而且可能运行缓慢，影响音频的实时性，幸亏及时转战PyQt6。除了之前所说环境配置的问题，更大的问题是python和PyQt6的学习，由于pygame浪费了很多时间，连续学了一个星期，算是把QtDesigner的所有功能都理解得很透彻，最重要的功能是信号与槽，这是前端和后端结合的关键。第二个问题是参数均衡器界面设计，需要将曲线绘制和鼠标点击结合在一起，用到了matplotlib这个十分强大的绘图库。由于我电脑python版本是3.12，而PyQt6建议3.5~3.9，想要下载PyQt6需要安装一个补丁，可能是这个原因让我的代码无法和matplotlib结合，就算是从官网上复制下来的代码也无法显示图像。于是就用PyQt5写了参数均衡器，我的电脑上就能正常显示了。第三个问题是如何把这个界面变成二级面板。因为这个界面不是用QtDesigner设计的，所以格式不一样，在改完以后发现PyQt6和5并不完全兼容，正因如此，无法成功显示，而改成PyQt6后参数均衡器界面又无法显示。正当我一筹莫展之时，组长刚好去树莓派上测试代码，发现树莓派上是可以正常显示的，万幸。那么基础功能基本都实现了，后期除了美化还有一些些小小的升级，比如滑轨禁用、鼠标释放才返回数据等等。后来小组进度较快，又增加了一个专属模式界面，在前几个界面的美学基础上，增加了插图背景。

稍有遗憾的是作为信号处理的实验，并没有做太多和信号直接相关的内容，虽然理解了混响和均衡器的原理，并成功地将代码和按键控制结合起来，但整体上未能深入探索信号处理的复杂性和多样性。不过还是掌握了一项全新的技能，一个良好的UI界面，不仅需要功能齐全，还要在视觉上带来愉悦，朴素而不失雅致。在后期优化的过程中，大大提升了我的美学修养，或许能在日后精进我的摄影技术。

总的来说，这是一次很有意义的实验，虽说实验内容已经连续好几年了，但对于我们来说是创新性的，当音频随着按键实时改变时，内心也是百感交集，我也没辜负小组成员的心血，将后端完美地呈现出来。