

Sstar 音频调试指导手册

Version 2.4

© 2020 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.

REVISION HISTORY

| Revision No. | Description | Date |
|--------------|------------------------------------------------------------------------------------------------|------------|
| 1.0 | <ul style="list-style-type: none">• Create this document | 08/14/2019 |
| 2.0 | <ul style="list-style-type: none">• Fix bugs and add a amplifier distortion case | 09/12/2019 |
| 2.1 | <ul style="list-style-type: none">• Update case | 09/26/2019 |
| 2.2 | <ul style="list-style-type: none">• Update ANR/EQ/AGC path | 02/11/2020 |
| 2.3 | <ul style="list-style-type: none">• Add attention of EQ | 05/19/2020 |
| 2.4 | <ul style="list-style-type: none">• Add more common case analysis | 07/21/2020 |

TABLE OF CONTENTS

| | |
|-----------------------------------------------|-----------|
| REVISION HISTORY | i |
| TABLE OF CONTENTS..... | ii |
| 1. MI Audio 内部流程介绍 | 1 |
| 1.1. MI Audio 内部框图介绍 | 1 |
| 1.1.1 Ai 数据流向分析 | 1 |
| 1.1.2 Ao 数据流向分析 | 1 |
| 1.1.3 确认原始数据和算法处理正常 | 2 |
| 2. MI Audio 相关算法介绍 | 3 |
| 2.1. AEC (Acoustic Echo Cancellation) | 3 |
| 2.1.1 AEC 算法的简单介绍..... | 3 |
| 2.1.2 AEC 算法的参数说明..... | 3 |
| 2.2. ANR (Acoustic Noise Reduction) | 4 |
| 2.2.1 ANR 算法的简单介绍 | 4 |
| 2.2.2 ANR 算法的参数说明 | 4 |
| 2.3. EQ (Equalizer) | 6 |
| 2.3.1 EQ 算法的简单介绍 | 6 |
| 2.3.2 EQ 算法的参数说明 | 6 |
| 2.4. AGC (Automatic Gain Control) | 7 |
| 2.4.1 AGC 算法的简单介绍 | 7 |
| 2.4.2 AGC 算法的参数说明 | 7 |
| 3. 如何确认客户设备的结构好坏 | 11 |
| 3.1. 测试音档介绍 | 11 |
| 3.2. 确定 speaker 和 microphone 的增益 | 11 |
| 3.2.1 确认 speaker 的增益..... | 11 |
| 3.2.2 确认 microphone 的增益 | 11 |
| 3.3. 内部隔离 | 12 |
| 3.4. 回音损耗 | 12 |
| 3.5. 频响特性 | 12 |
| 4. 音效调整 | 14 |
| 4.1. 音效参数调整 | 14 |
| 4.1.1 确定 mic 和 speaker 的增益..... | 14 |
| 4.1.2 调整 AO 的 EQ 参数, 修正 speaker 的频响特性 | 14 |
| 4.1.3 AEC 频段和强度调整..... | 19 |
| 4.1.4 ANR 参数调整 | 19 |
| 4.1.5 AGC 参数调整 | 20 |
| 5. 音效调整结构设计建议以及硬件选型 | 22 |
| 5.1. Speaker 和 microphone 的选型建议 | 22 |
| 5.2. 结构设计建议 | 22 |
| 6. 基本调试流程 | 23 |
| 7. 常见案例 | 34 |

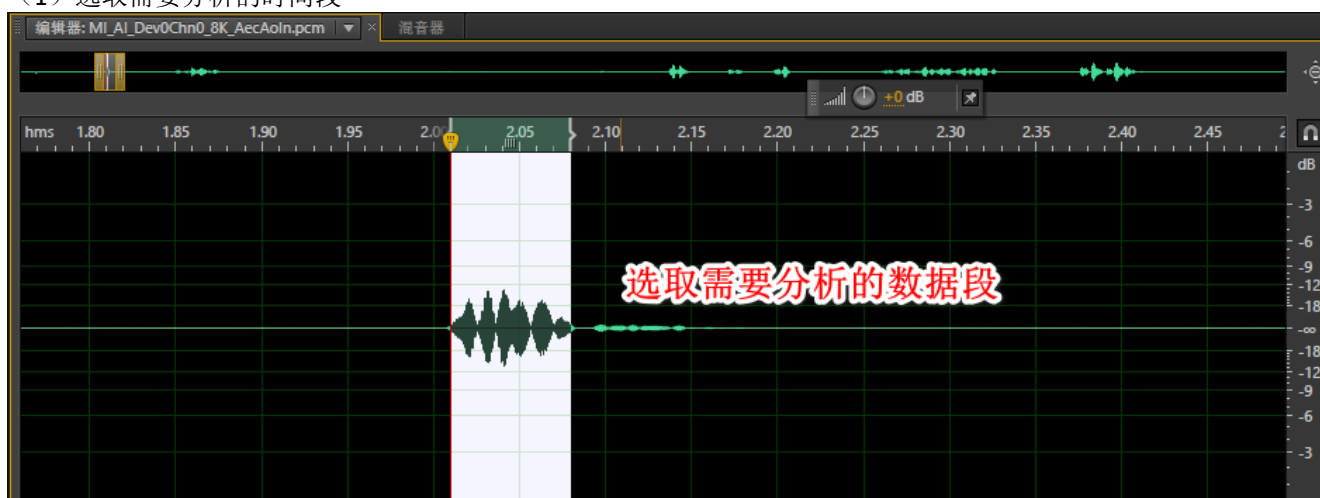
8. 注意事项 37

1. ADOBE AUDITION 的常用操作

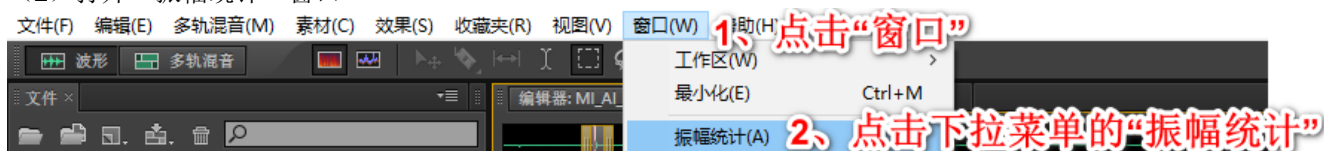
1.1. 查看增益

在对讲的调试过程中经常需要查看一段时间内的信号大小。下面以图为例，说明如何通过 Adobe audition 来查看信号的大小。

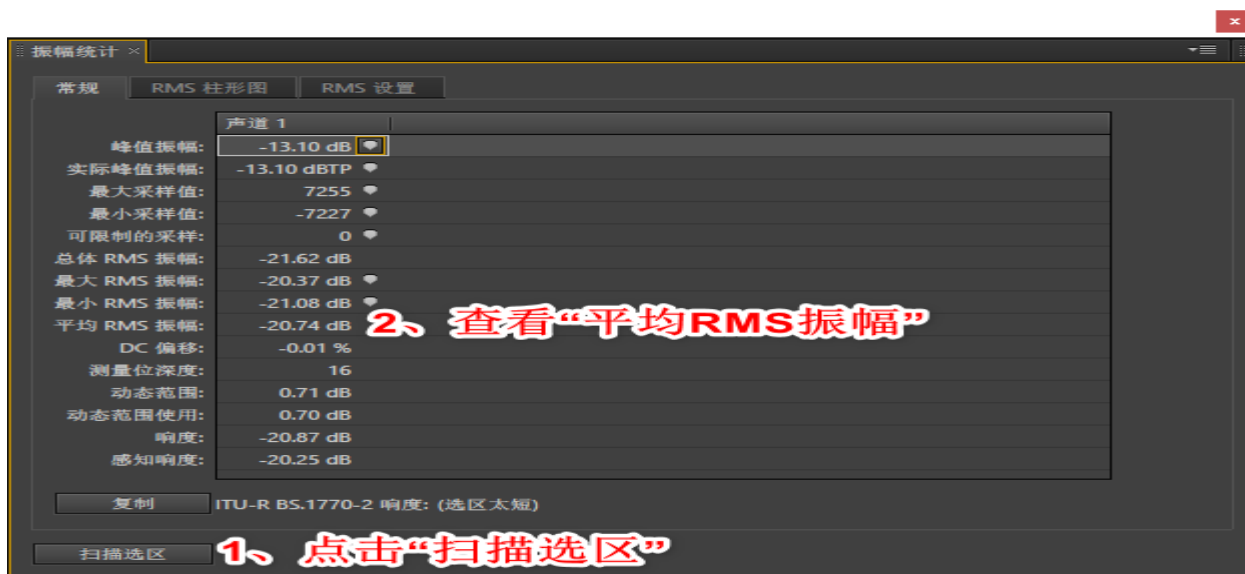
(1) 选取需要分析的时间段



(2) 打开“振幅统计”窗口



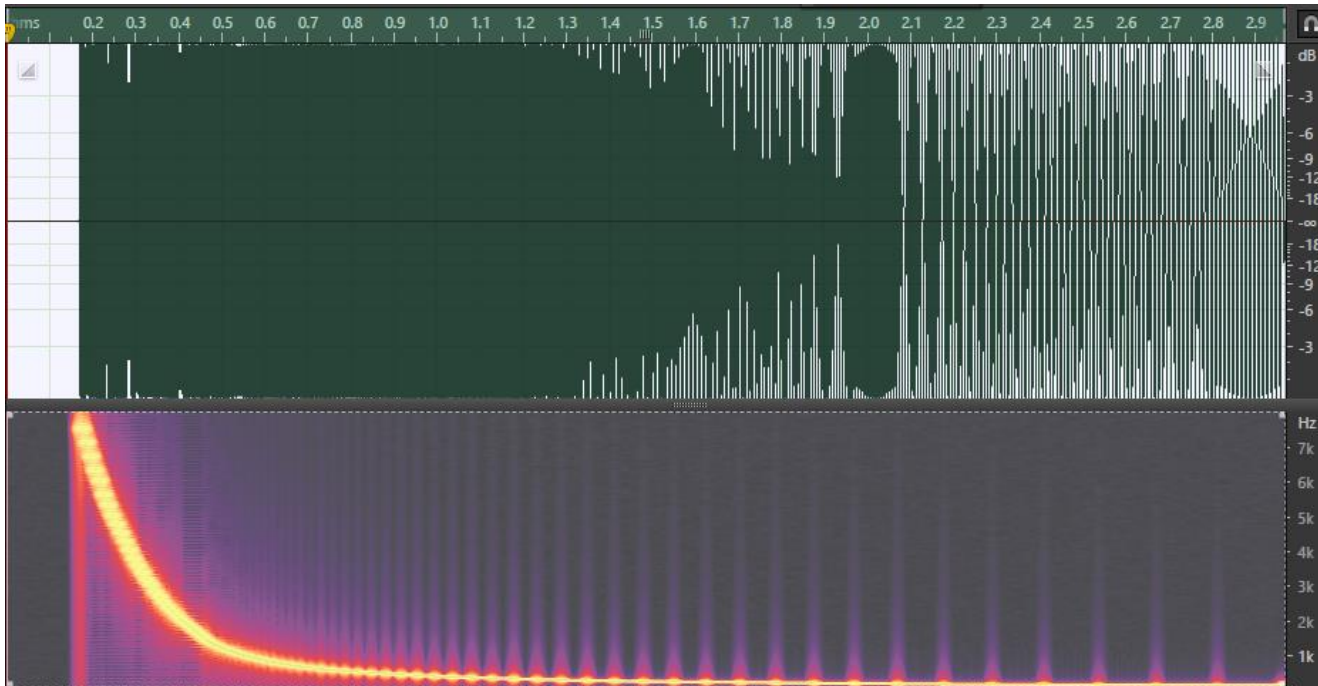
(3) 查看选区的平均振幅



1.2. 查看频响特性

在对讲的调试过程中，经常需要对设备的频响特性做调整，下面将以图例讲述如何通过 adobe audition 来查看音档的频响特性。当我们得到当前设备的频响曲线后，则可以通过 EQ 去调整。

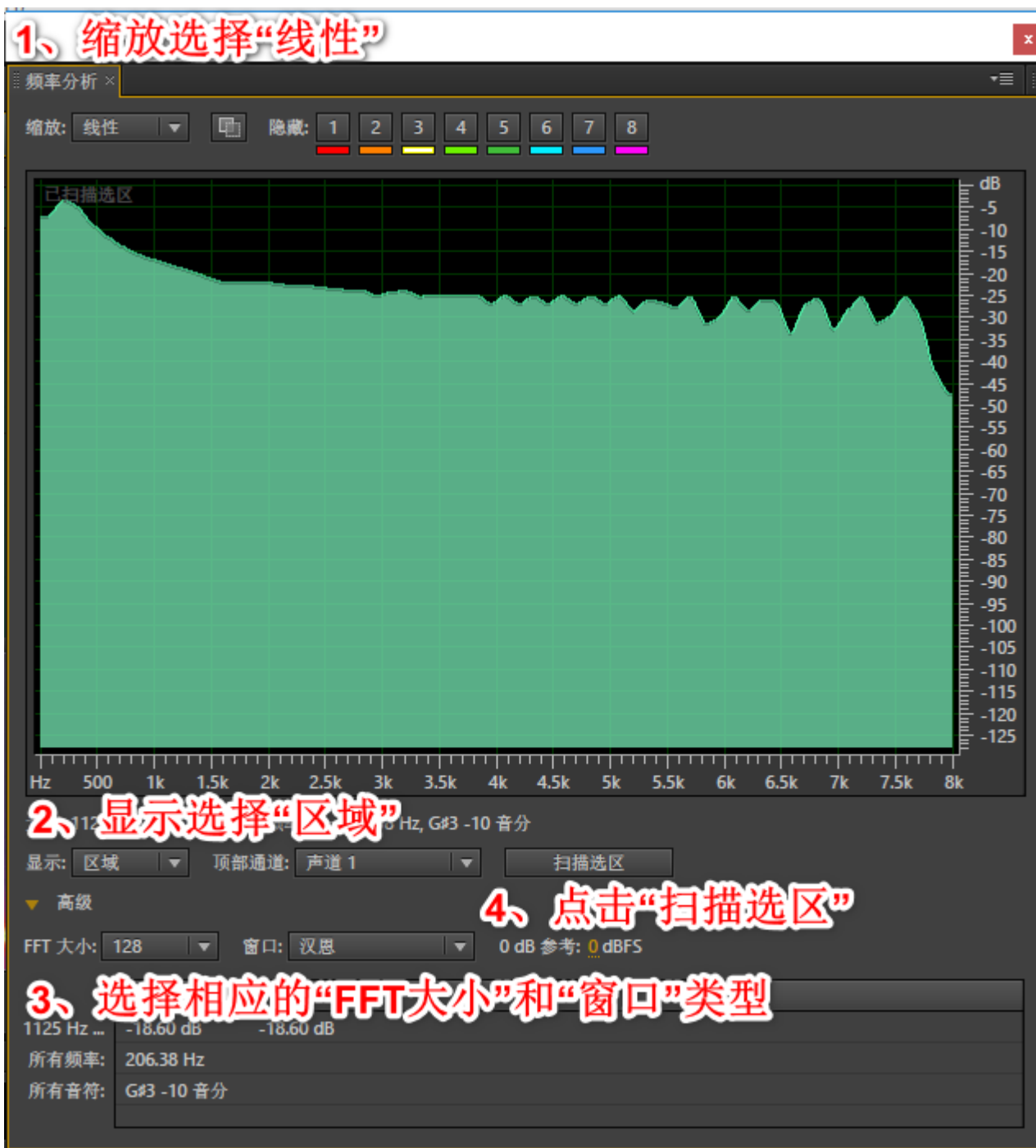
(1) 选取需要分析的数据选区

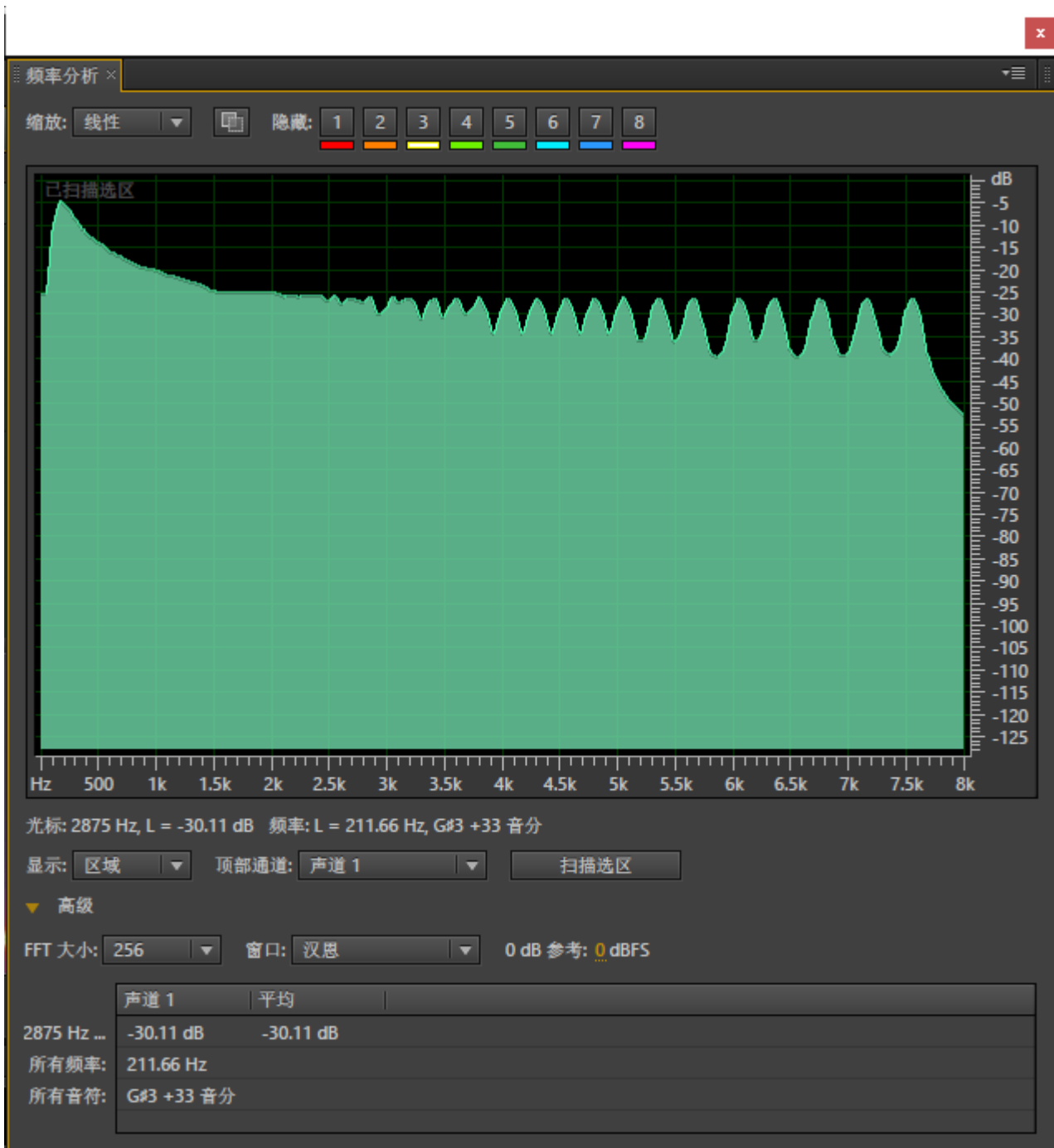


(2) 打开“频率分析”窗口



(3) 点击“扫描选区”



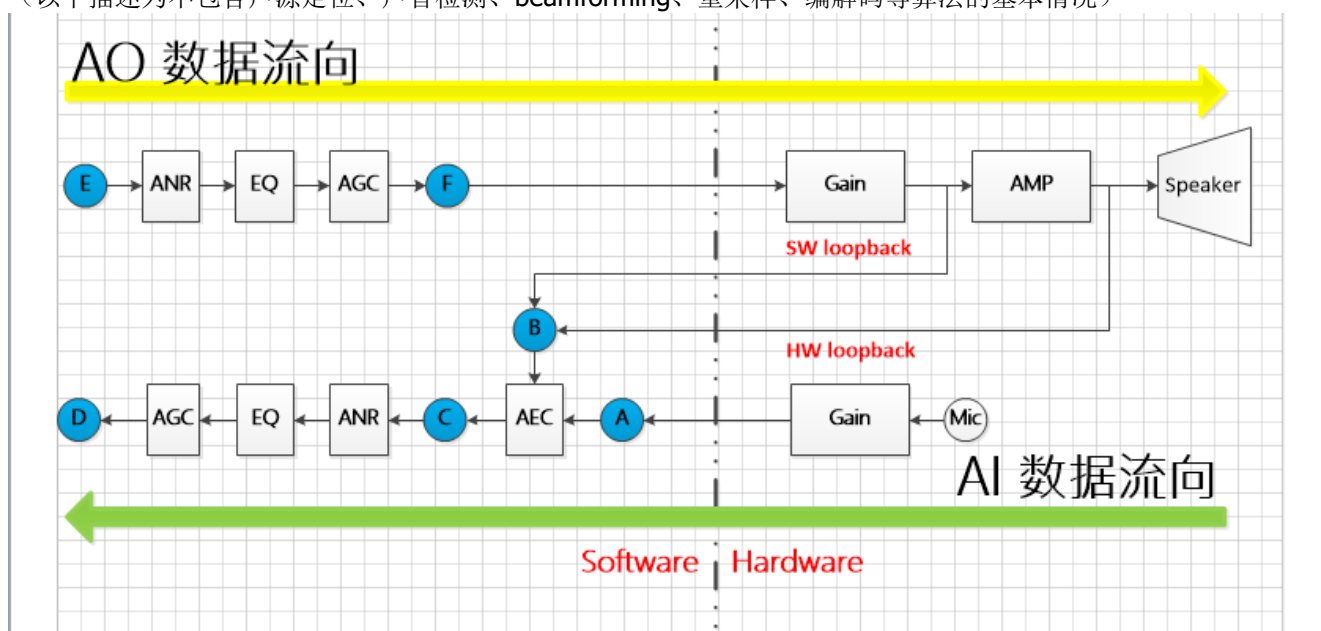


2. MI AUDIO 内部流程介绍

2.1. MI Audio 内部框图介绍

下图为 MI Audio 的基本数据流向：

（以下描述为不包含声源定位、声音检测、beamforming、重采样、编解码等算法的基本情况）



2.1.1 Ai 数据流向分析

上图中绿色箭头方向为 AI 数据的流向，数据由麦克风采集，经过芯片内部的增益，到达 MI 层（A 点），此处（A 点）的数据为原始的 PCM 数，接着和 AO 端的数据（B 点）一起送入 AEC 算法进行处理，AEC 算法处理后的数据（C 点）会继续送到后面的 APC 算法（APC 为 ANR、EQ、HPF、AGC 算法的集合）进行处理，处理后的数据即为最终的数据（D 点）。

SW loop back 与 HW loop back 的区别：

SW loop back 和 HW loop back 的主要区别在于送回 AEC 算法的 AO 数据差异。SW loop back 用于处理 AEC 算法的 AO 数据是只经过了芯片内部增益的数据，好处在于不需要外部电路进行特定分压设计，且只需占用一个音频输入通道，与 HW loop back 相比，对于外部电路的功放芯片引起的失真，SW loop back 没法较好地进行处理。而 HW loop back 用于处理 AEC 算法的 AO 数据为外部电路功放芯片处理后的数据，可以较好地应对上述情况，但需要对外部电路进行特定的分压设计，多占用一个音频输入通道，以及需要确保 loop back 回来的峰值不能超过芯片内部量程（3.3V）。HW loop back 必须要有两个音频输入通道才能使用。

（HW loop back 只有两通道的 Amic 才能使用）

2.1.2 Ao 数据流向分析

上图中黄色箭头方向为 AO 数据的流向，待播放 PCM 数据（E 点）由应用通过 MI 接口送入后，经过 APC 算法（ANR、EQ、AGC 处理后（F 点），经过芯片内部的增益、外部电路的功放芯片，再由喇叭输出。

2.1.3 确认原始数据和算法处理正常

上图中所有蓝色点的数据都可以通过 SDK 提供的 debug 手段 dump 出来（因为此功能仅仅是为了 debug 用，若有出现文件名有误，SDK 暂时不做修正）。SDK 提供的 dump 手段，需要在应用起来之前，通过 export 一系列的环境变量来指定文件的保存路径以及需要 dump 的数据位置。

指定文件保存的路径：

`export MI_AI_DUMP_PATH=xxx`，指定 AI 的 dump 路径为 xxx

`export MI_AO_DUMP_PATH=xxx`，指定 AO 的 dump 路径为 xxx

指定 dump 的数据位置：

AI:

`export MI_AI_DUMP_PCM_ENABLE=1`，指定 dump AI 的原始 pcm（即 A 点处的数据），文件名为 MI_AI_Dev*Chn*_K_Src.pcm。

`export MI_AI_DUMP_AEC_ENABLE=1`，指定 dump AI AEC 前后的数据（即 A、B、C 点处的数据），文件名为 MI_AI_Dev*Chn*_K_AiIn.pcm、MI_AI_Dev*Chn*_K_AoIn.pcm 和 MI_AI_Dev*Chn*_K_AecOut.pcm。（当 AEC 开启时，MI_AI_Dev*Chn*_K_AiIn.pcm 和 MI_AI_Dev*Chn*_K_Src.pcm 的内容是一致的。）

`export MI_AI_DUMP_VQE_ENABLE=1`，指定 dump AI APC 前后的数据（C、D 点处的数据），文件名为 MI_AI_Dev*Chn*_K_VqeIn.pcm 和 MI_AI_Dev*Chn*_K_VqeOut.pcm（当 AEC 开启时，MI_AI_Dev*Chn*_K_VqeIn.pcm 和 MI_AI_Dev*Chn*_K_AecOut.pcm 的内容是一致的。当没有开启 AEC 时，MI_AI_Dev*Chn*_K_Src.pcm 和 MI_AI_Dev*Chn*_K_VqeIn.pcm 的内容是一致的。）

AO:

`export MI_AO_DUMP_PCM_ENABLE=1`，指定 dump 最终送给驱动的数据（即 F 点处的数据），文件名为 MI_AO_Dev*_K_Dst.pcm。

`export MI_AO_DUMP_VQE_ENABLE=1`，指定 dump AO APC 前后的数据（即 E、F 点处的数据），文件名为 MI_AO_Dev*_K_VqeIn.pcm 和 MI_AO_Dev*_K_VqeOut.pcm。（当 AO 端只有 APC 算法开启时，MI_AO_Dev*_K_VqeOut.pcm 和 MI_AO_Dev*_K_Dst.pcm 的内容一致。）

(*为应用具体使用的 device id、channel id、采样率)

建议只要 dump 自己想要的节点处的音档，不要所有都同时 dump，不然可能会引起一直在写文件导致 SDK 丢帧。

3. MI AUDIO 相关算法介绍

3.1. AEC (Acoustic Echo Cancellation)

3.1.1 AEC 算法的简单介绍

AEC 即回声消除，通过算法来消除由扬声器设备发出的直接或间接耦合回麦克风的聲音。

3.1.2 AEC 算法的参数说明

下面为 MI API 开给应用设定 AEC 参数的数据结构：

```
typedef struct MI_AI_AecConfig_s
{
    MI_BOOL bComfortNoiseEnable;
    MI_S16 s16DelaySample;
    MI_U32 u32AecSupfreq[6];
    MI_U32 u32AecSupIntensity[7];
    MI_S32 s32Reserved;
} MI_AI_AecConfig_t;
```

| 参数 | 作用 | 限制/注意事项/详细说明 |
|---------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bComfortNoiseEnable | AEC 加入舒适噪声使能 | 默认情况设置 FALSE 。 说明：当使能该参数时，AEC 算法的处理结果会加入一些背景噪声，使处理后声音不会过于空洞。此项主要是为了解决 AEC 消得太干净，导致 ANR 收敛需要太长时间（具体现象为语音之间存在噪声被拉起后又被压制的现象）。 |
| s16DelaySample | 左右声道间的回声延迟样本 | 默认情况设置 0 。 说明：由于麦克风和喇叭的放置位置、麦克风间的距离等会造成左右声道接收到回声的时间点不一致，两个声道之间回声延迟有差异。此值表示左声道比右声道提早多少个采样点收到回声。 注意：此参数只适用于双声道做 AEC 的情况，且不建议使用。 |
| u32AecSupfreq | AEC 处理的频段 | 说明：该数组将当前采样率的最高频率分成 8 个频段来处理。以下数组元素和频率范围的转换公式： $\frac{\text{residual echo frequency range}}{\frac{\text{sample rate}}{2}} \times \text{point number}(128)$ 假设当前采样率 16K，以{4, 6, 36, 49, 50, 51}为例，当前采样率为 16K，最大的采样频率为 8K，第一个频段 F1 = (0~4) * 8000 / 128 = 0~250Hz，第二个频段 F2 = (4~6) * 8000 / 128 = 250~375Hz，第三个频段 F3 = (6~36) * 8000 / 128 = 375~2250Hz，第四个频段 F4 = (36~49) * |

| | | |
|--------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | $8000 / 128 = 2250 \sim 3062\text{Hz}$, 第五个频段 F5 = (49~50) * 8000 / 128 = 3062~3125Hz, 第六个频段 F6 = (50~51) * 8000 / 128 = 3125~3187Hz, 第七个频段 F7 = (51~128) * 8000 / 128 = 3187~8000Hz。 注意：数组要求每一个元素都必须比前一个元素大。参数范围是[1,127]。 |
| u32AecSupIntensity | AEC 处理的强度 | 说明：此数组代表频段的回声消除强度，各元素和 u32AecSupfreq 划分出来的频率范围一一对应。 注意：参数的范围是[0,15]，强度越大，消除的细节越多，声音越不自然。反之，强度越小，细节保留越多，但会回声消除不够干净。（若客户对消除效果不满意，只能让客户来权衡，再做微调） |

3.2. ANR (Acoustic Noise Reduction)

3.2.1 ANR 算法的简单介绍

ANR 即声音降噪，通过算法来消除环境中持续存在的噪声（比如白噪声等），**没法很好地处理突然出现的噪声。**

3.2.2 ANR 算法的参数说明

下面为 MI API 开给应用设定 ANR 参数的数据结构：

因 NR 算法有更新，请按照当前 SDK 版本对应的 NR 数据结构来相对应。

```
typedef struct MI_AUDIO_AnConfig_s
{
    MI_AUDIO_AlgorithmMode_e eMode;
    MI_U32    u32NrIntensity;
    MI_U32    u32NrSmoothLevel;
    MI_AUDIO_NrSpeed_e eNrSpeed;
}MI_AUDIO_AnConfig_t;
```

| 参数 | 作用 | 限制/注意事项/详细说明 |
|----------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eMode | ANR 算法的模式 | 说明：此参数表示 ANR 算法的模式，不同的模式对应不同的应用场景。 E_MI_AUDIO_ALGORITHM_MODE_DEFAULT ：此模式为语音模式，会使用算法中为语音数据调教的默认降噪参数。 E_MI_AUDIO_ALGORITHM_MODE_USER ：此模式全部使用应用下的降噪参数。 E_MI_AUDIO_ALGORITHM_MODE_MUSIC ：此模式为音乐模式，会使用算法中为音乐数据调教的降噪参数。 （推荐使用此模式） |
| u32NrIntensity | ANR 算法的强度 | 推荐值：20 说明：该参数表示 ANR 算法的降噪强度，该值越大，降噪 |

| | | |
|------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | 效果越好，但细节损失越多。反之，降噪强度越小，细节保留越多，但噪声也没法去除干净。（此值也需要权衡和微调。） 注意：参数范围为[0,30]。 |
| u32NrSmoothLevel | ANR 算法的平滑程度 | 推荐值：10 说明：该参数表示 ANR 算法处理的平滑程度，值越大降噪效果越平滑。平滑程度越小在频域上损失的细节越多。 注意：参数范围为[0,10]。 |
| eNrSpeed | ANR 算法的收敛速度 | 推荐值：E_MI_AUDIO_NR_SPEED_MID 说明：此值表示 ANR 算法的收敛速度，收敛速度越快损失的细节越多。反之，收敛速度越慢，参考的时间越长，细节保留越多，但也会延长降噪起作用的时间。当当前的收敛速度已设置为 E_MI_AUDIO_NR_SPEED_HIGH，但仍然收敛得比较慢时，可以打开 AEC 的 bComfortNoiseEnable，让 AEC 的处理结果主动加入噪声，来辅助 ANR 的收敛。 |

```
typedef struct MI_AUDIO_AnConfig_s
{
    MI_AUDIO_AlgorithmMode_e eMode;
    MI_U32    u32NrIntensityBand[_NR_BAND_NUM - 1];
    MI_U32    u32NrIntensity[_NR_BAND_NUM];
    MI_U32    u32NrSmoothLevel;
    MI_AUDIO_NrSpeed_e eNrSpeed;
}MI_AUDIO_AnConfig_t;
```

| 参数 | 作用 | 限制/注意事项/详细说明 |
|--------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eMode | ANR 算法的模式 | 说明：此参数表示 ANR 算法的模式，不同的模式对应不同的应用场景。 E_MI_AUDIO_ALGORITHM_MODE_DEFAULT：此模式为语音模式，会使用算法中为语音数据调教的默认降噪参数。 E_MI_AUDIO_ALGORITHM_MODE_USER：此模式全部使用应用下的降噪参数。 E_MI_AUDIO_ALGORITHM_MODE_MUSIC：此模式为音乐模式，会使用算法中为音乐数据调教的降噪参数。（ 推荐使用此模式 ） |
| u32NrIntensityBand | ANR 的频段划分 | 说明：u32NrIntensityBand 需要与 u32NrIntensity 搭配使用，与 AEC 的 u32AecSupfreq 和 u32AecSupIntensity 作用一致，可以将整个频段划分为 7 段来使不同降噪强度作用到不同的频段上。 |
| u32NrIntensity | ANR 算法的强度 | 推荐值：20 |

| | | |
|------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | 说明：该参数表示 ANR 算法的降噪强度，该值越大，降噪效果越好，但细节损失越多。反之，降噪强度越小，细节保留越多，但噪声也没法去除干净。（此值也需要权衡和微调。） 注意：参数范围为[0,30]。 |
| u32NrSmoothLevel | ANR 算法的平滑程度 | 推荐值：10 说明：该参数表示 ANR 算法处理的平滑程度，值越大降噪效果越平滑。平滑程度越小在频域上损失的细节越多。 注意：参数范围为[0,10]。 |
| eNrSpeed | ANR 算法的收敛速度 | 推荐值：E_MI_AUDIO_NR_SPEED_MID 说明：此值表示 ANR 算法的收敛速度，收敛速度越快损失的细节越多。反之，收敛速度越慢，参考的时间越长，细节保留越多，但也会延长降噪起作用的时间。当当前的收敛速度已设置为 E_MI_AUDIO_NR_SPEED_HIGH，但仍然收敛得比较慢时，可以打开 AEC 的 bComfortNoiseEnable，让 AEC 的处理结果主动加入噪声，来辅助 ANR 的收敛。 |

3.3. EQ (Equalizer)

3.3.1 EQ 算法的简单介绍

EQ 即均衡器，该算法用于提高或降低某些频段的能量。

3.3.2 EQ 算法的参数说明

下面为 MI API 开给应用设定 EQ 参数的数据结构：

```
typedef struct MI_AUDIO_EqConfig_s
{
    MI_AUDIO_AlgorithmMode_e eMode;
    MI_S16    s16EqGainDb[129];
}MI_AUDIO_EqConfig_t;
```

| 参数 | 作用 | 限制/注意事项/详细说明 |
|-------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| eMode | EQ 算法的模式 | 说明：该值表示 EQ 算法的模式 E_MI_AUDIO_ALGORITHM_MODE_DEFAULT：默认模式，完全不吃应用下的参数，全频段放大。 E_MI_AUDIO_ALGORITHM_MODE_USER：此模式全部使用应用下的降噪参数。（当客户需要自己调教时使用此模式） |
| s16EqGainDb | EQ 算法的均衡器数组 | 说明：该数组表示各个频段的增益设置。 该参数把当前采样率对应的最大采样频率分成 129 份，可单独对某个频段进行能量的调节。 假设当前的采样率为 16K，则最大的采样频率为 8K，每一 |

| | | |
|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>份对应的频段大小为 $F = 8000 / 129 \approx 62\text{Hz}$。此数组元素对应的频段为 0~62Hz, 62~124Hz, 124~192Hz,, 7938~8000Hz。</p> <p>注意：该参数的为范围是[-50,20]dB。</p> <p>当需要调整的增益变化很大时，往往没法在一个频段内完成增益的变化，此时需要将附近相邻的频段也设置为同样的参数，来达到调整该频段的目的。</p> |
|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

3.4. AGC (Automatic Gain Control)

3.4.1 AGC 算法的简单介绍

AGC 即自动增益控制，此算法可以对输入的数据进行增益控制，当信号过小时，会对信号进行增益，当信号过大时，会对信号进行衰减。

3.4.2 AGC 算法的参数说明

下面为 MI API 开给应用设定 AGC 参数的数据结构：

因为 AGC 算法更新，数据结构会有所改动，以下通过蓝色和红色进行区分。

```
typedef struct AgcGainInfo_s{
    MI_S32    s32GainMax;
    MI_S32    s32GainMin;
    MI_S32    s32GainInit;
}AgcGainInfo_t;

typedef struct MI_AUDIO_AgcConfig_s
{
    MI_AUDIO_AlgorithmMode_e eMode;
    AgcGainInfo_t stAgcGainInfo;
    MI_U32    u32DropGainMax;
    MI_U32    u32AttackTime;
    MI_U32    u32ReleaseTime;
    MI_S16    s16Compression_ratio_input[5/7];
    MI_S16    s16Compression_ratio_output[5/7];
    MI_S32    s32TargetLevelDb/s32DropGainThreshold
    MI_S32    s32NoiseGateDb;
    MI_U32    u32NoiseGateAttenuationDb;
}MI_AUDIO_AgcConfig_t;
```

| 参数 | 作用 | 限制/注意事项/详细说明 |
|-------|-----------|--------------------------------------------------------------------------------------------------------------------|
| eMode | AGC 算法的模式 | <p>说明：此参数表示 AGC 算法的模式，不同的模式对应不同的应用场景。</p> <p>E_MI_AUDIO_ALGORITHM_MODE_DEFAULT：此模式为语音模式，会使用算法中为语音数据调教的 AGC 参数。</p> |

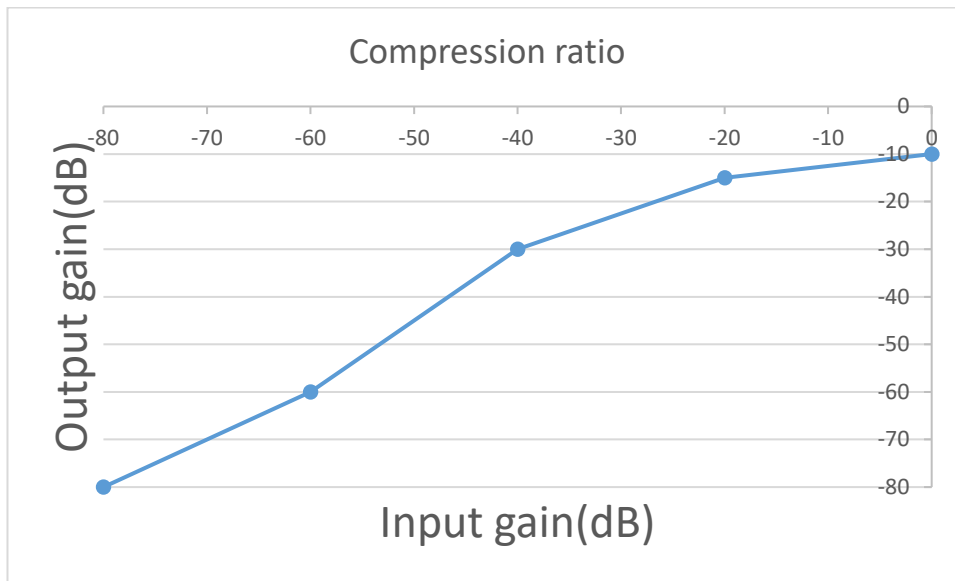
| | | |
|----------------------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | <p>E_MI_AUDIO_ALGORITHM_MODE_USER: 此模式全部使用应用下的 AGC 参数。(当客户需要自己调教时使用此模式)</p> <p>E_MI_AUDIO_ALGORITHM_MODE_MUSIC: 此模式为音乐模式, 会使用算法中为音乐数据调教的 AGC 参数。</p> |
| s32GainMax | AGC 算法调节增益的最大值(dB) | <p>说明: 此值为 AGC 算法增益可调节的最大值。</p> <p>注意: 该参数的范围为[0,60]dB。</p> |
| s32GainMin | AGC 算法调节下降的最小值(dB) | <p>说明: 此值为 AGC 算法增益可调节的最小值。</p> <p>注意: 该参数的范围为[-20,30]dB。</p> |
| s32GainInit | AGC 算法调节增益的初始值(dB) | <p>说明: 此值为 AGC 算法增益的初始值。</p> <p>注意: 该参数的范围为[-20,60]dB。</p> |
| u32DropGainMax | AGC 算法增益下降的最大值(dB) | <p>说明: 此值为 AGC 算法增益调节超过目标值(s32TargetLevelDb/s32DropGainThreshold)时, 可下降的最大值, 防止输出饱和。(当出现此情况时, AGC 算法会在 0dB~u32DropGainMax dB 的范围内衰减, 并不是只要一超过就会衰减 u32DropGainMax dB, 且调整后的增益还会受到 stAgcGainInfo 的限制)</p> <p>注意: 该参数的范围为[0,60]dB。该值不宜设置太小, 否则当输入信号远高于目标值时, 增益没法快速的降下来, 导致出现削顶。同时, 该值也不宜设置过大, 在输入信号远高于目标值 s32TargetLevelDb/s32DropGainThreshold 时, 信号的增益会瞬间下降太大, 导致帧与帧之间没法接起来, 可能导致出现明显的噗噗声。</p> |
| u32AttackTime | AGC 算法增益下降的时间 | <p>说明: 此值为 AGC 算法增益下降的时间(多少个时间单位下降 0.5dB), 一个时间单位为 16ms。此值越小, 增益下降越快。此值尽量设小, 否则当输入信号高于目标值 s32TargetLevelDb, 没法快速地降下来。此参数主要配合曲线来工作。</p> <p>注意: 该值的范围为[1,20]</p> |
| u32ReleaseTime | AGC 算法增益上升的时间 | <p>说明: 此值为 AGC 算法增益上升的时间(多少个时间单位上升 0.5dB), 一个时间单位为 16ms。此值越小, 增益上升越快。此值尽量避免设置太小, 否则当经过长时间的增益积累, 突然出现一个较大的输入信号, 则会导致声音爆掉。此值设置太大也会导致信号拉不起来。此值主要配合曲线来工作。</p> <p>注意: 该值的范围为[1,20]</p> |
| s16Compression_ratio_input | AGC 算法的增益设置曲线 | <p>说明: 由 s16Compression_ratio_input 和 s16Compression_ratio_output 组成一条具有 4/6 段斜率的增益调节曲线。</p> |

| | | |
|---------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| s16Compression_ratio_output | | 注意：两个数组的元素的参数范围为[-80,0]dBFS。 建议将 compression ratio output 的最大值设定为-3dBFS 及以下，以防削顶。 注：更改的内容只是增加了可以调节的段数。 |
| s32TargetLevelDb/s32DropGainThreshold | AGC 算法的增益目标值 | 说明：AGC 算法允许输出的增益最大值，大于此值算法将会对信号进行衰减，以防爆音。 注意：此值的参数范围为[-80,0]dB。 |
| s32NoiseGateDb | AGC 算法处理的噪声门限值 | 推荐值：-80 说明：AGC 算法处理的噪声门限值，但输入信号的增益低于此值时，AGC 会认为是噪声，会配合 u32NoiseGateAttenuationDb 的设置，对噪声进行衰减。若想保留噪声的话，需要将此值设定到-80，且同时将-80dBFS 到实测噪声门限值之间的这段曲线的斜率设置为 1。建议尽量使用曲线来处理噪声。 注意：此参数范围为[-80,0]dBFS 当喇叭存在对方传过来的回声时，当对方 AEC 实在没法把回声处理掉，可使用 s32NoiseGateDb 和 u32NoiseGateAttenuationDb 配合，砍掉对方传回来的回声，同时也会对人声有影响（需要权衡）。 |
| u32NoiseGateAttenuationDb | AGC 算法处理的噪声衰减 | 推荐值：0 说明：AGC 算法处理的噪声衰减的增益，此值会配合 s32NoiseGateDb 来处理低于门限值的噪声。不推荐使用此值来对噪声进行衰减，防止输出信号听起来像被中断，当使用此值时，会导致说话的开头和结尾被消掉，使声音听起来不够自然。 注意：此参数范围为[0,100]dB |

s16Compression_ratio_input 和 s16Compression_ratio_output 是需要根据实际的场景来进行设置（各个产品的器件、结构、适配的机型各不相同，需要具体抓音档来分析），没有默认推荐值。

下面按以 s16Compression_ratio_input = {-80,-60,-40,-20,0}; s16Compression_ratio_output = {-80,-60,-30,-15,-10};这组参数进行讲解。如下面的折线图所示，在输入增益为-80~0dB 划分为四段斜率，-80dB~-60dB 范围内保持原来的增益，斜率为 1，-60dB~-40dB 范围内需要稍微提高增益，斜率为 1.5，-40dB~-20dB 范围内斜率为 1.25，-20dB~0dB 范围内斜率为 0.25。即输入增益在-80~-60dB 范围内的信号，输出增益为-80~-60dB; 输入增益在-60~-40dB 范围内的信号，输出增益为-60~-30dB; 输入增益在-40~-20dB 范围内的信号，输出增益为-30~-15dB; 输入增益在-20~0dB 范围内的信号，输出增益为-15~-10dB;

根据所需的曲线转折点对 s16Compression_ratio_input 和 s16Compression_ratio_output 设置，若不需要那么多段曲线，则将数组不需要的部分填 0。



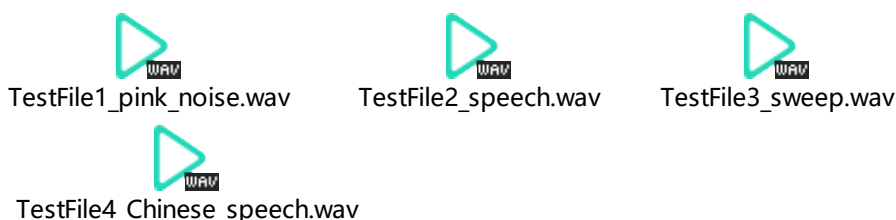
(此图仅为了说明曲线的参数及其意义，并无实际参考价值)

4. 如何确认设备的结构好坏

当拿到客户设备时，如何确认客户设备结构设计的好坏？主要由以下两点来确认：**内部隔离**和**回声损耗(Echo Return Loss (ERL))**。（若这两项没法达到要求，则说明结构设计不太合理，要么建议可以修改结构，要么只能降低对效果的要求，尽量接近）

注：首先必须先确认客户使用的是 HW loop back 还是 SW loop back，若是使用 HW loop back 必须先对 chn1 做衰减处理（-6，-3，0dB），保证 chn0 不会因为 chn1，而导致失真。以下所有步骤均在不开启任何音频算法的前提下操作。

4.1. 测试音档介绍



TestFile1 为一段噪声，用于辅助确认 microphone 和 speaker 的增益。

TestFile2 为一段对话语音，主要是外国友人的声音，频段比较全，消除难度相对比 TestFile4 高，用于调整 AEC 参数。

TestFile3 为一段频率逐渐增加的声音，用于扫频，用于检查 mic 和 speaker 的频率响应特性。

TestFile4 为一段对话语音，主要是国人的声音，用于调整 AEC 参数。

4.2. 确定 speaker 和 microphone 的增益

需要测量**内部隔离**和**回声损耗**，首先第一步是根据客户的需求（**mic 的收音距离**和**speaker 的音量**），通过测量定好 microphone 和 speaker 的增益。此步骤为后续调音的**最为重要**的一步，若客户需要更改这两个增益，所有的算法相关参数都需要重新调整。

4.2.1 确认 speaker 的增益

- (1) 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
- (2) 用待测设备播放 TestFile1，调整 speaker 的增益，在保证没有失真的前提下，使之达到客户的需求。

4.2.2 确认 microphone 的增益

- (1) 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
- (2) 将分贝仪和待测设备放置在同一位置，在距离待测设备一米处，放置一台播放设备，用播放设备播放 TestFile1，并调节播放设备的增益，使分贝仪达到 70dbA。（没有两台机器请跳转到(4)）
- (3) 使用待测试设备进行收音，dump **A 点**处的数据（或 demo 在不打开任何算法的情况下录制的最终数据），调整 microphone 增益，使 RMS 达到-25 DBRMS 或以上。（[使用 Adobe audition 查看 RMS 的方法](#)）
- (4) 确定 speaker 和 microphone 的增益后，待测设备使用[确认 speaker 的增益](#)得到的增益播放 TestFile2，同时使用调整好的 microphone 增益录制音档，确认 microphone 录制的音档，没有失真。若有失真请继续调整 speaker 或 microphone 的增益（减小 speaker 的增益或减小 microphone 的增益）。**但有时候会因为结构、speaker、功放等因素，无论你怎么调整录回来的音档都会有某些频段存在失真，此时只能从音档上分析是什么原因造成的，尽量去调整，找到原因。**
- (5) 若结构设计不合理或者 microphone 选型灵敏度太低没法达到-25 DBRMS，则以开启 AEC 后 microphone 录回来的回音没有爆音的情况下，将 microphone 的增益调整到足够大。

4.3. 内部隔离

内部隔离越大越好，至少需要确保大于 6dB。以下为测量内部隔离的步骤：

- (1) 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
- (2) 用待测设备自录自播测试音档 TesFile1（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据），查看平均 RMS 振幅，记为 AVGRMS1。
- (3) 用黏土堵住机壳上的 microphone 孔，待测设备自录自播测试音档 TesFile1（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据），查看平均 RMS 振幅，记为 AVGRMS2。
- (4) 内部隔离为 AVGRMS1 - AVGRMS2。

4.4. 回音损耗

回音的损耗同样也是越大越好，至少需要确保这个值为一个正值。以下为测量回音损耗的步骤：

- (1) 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
- (2) 用工具查看测试音档 TestFile1 的平均 RMS 振幅，记为 AVGRMS1。
- (3) 用待测设备自录自播测试音档 TesFile1（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据），查看平均 RMS 振幅，记为 AVGRMS2。
- (4) $ERL = AVGRMS1 - AVGRMS2$ 。
- (5) 将 microphone 和 speaker 的增益设置为 0dB（注意 speaker 的增益包含功放的增益，若功放已有做放大，AO 的 gain 则设置成衰减使整体接近 0dB），用测试音档 TestFile3 代替 TestFile1，自播自录，dump 下音档（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据）。
- (6) 确认该音档在任何频段都不能有失真。

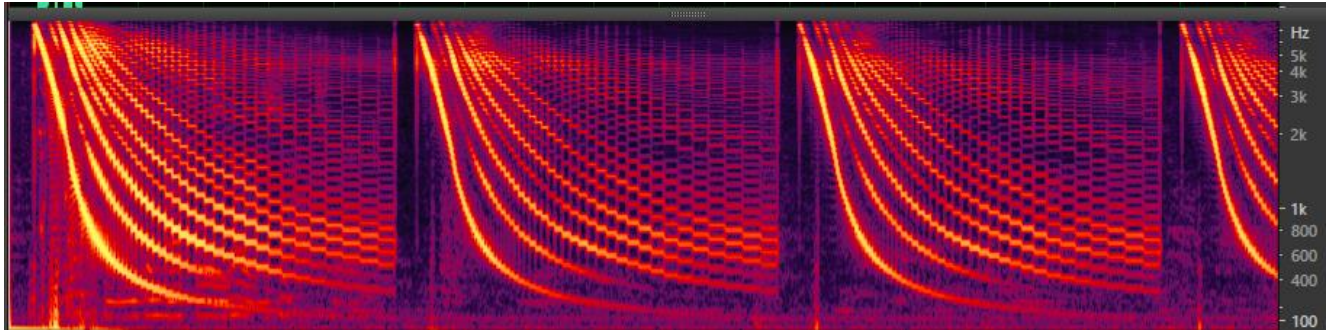
4.5. 频响特性

如何初步确定设备的频响特性的好坏？如何分析导致频响特性差的原因？

- (1) 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
- (2) 将 mic 和 speaker 的增益都设置为 0dB（注意 speaker 的增益包含功放的增益，若功放已有做放大，AO 的 gain 则设置成衰减使整体接近 0dB），自播自录扫频音档 TestFile3，查看收音的音档。

若发现收音音档的频响特性比较差时，可以按以下步骤来确认问题点：

- (1) 先把机壳拆开，确认 mic 和 speaker 是否有固定牢固，重复扫频操作，若频响特性变好，则确定为 mic 和 speaker 没有固定牢的原因。
- (2) 使用高保真音响播放测试音档 TestFile3，用标准麦克风和设备的 mic 同时拾音，对比两者的频响特性，确认是否为 microphone 的频响特性问题（因 mic 的集成度较高，一般频响特性没有太大的问题）。
- (3) 把 speaker 拿到机壳外，重复扫频操作，若频响特性变好，则确定为机构问题。
- (4) 当把 speaker 拿到机壳外，使用[标准麦克风](#)（或上面测量频响特性没有太大问题的 mic），重复扫频动作，若只有小部分频段失真，则可以在确定增益后使用 AO 端的 EQ 来对失真的频段进行修复。
- (5) 当把 speaker 拿到机壳外，使用[标准麦克风](#)（或上面测量频响特性没有太大问题的 mic），重复扫频动作，若基本上是全频段失真如下图。首先尝试下降 speaker 的增益，若发现已经降到很低了，还是会有全频段的失真，这时需要请硬件同事帮忙确认功放电路设计是否存在问题，可以修改电路最好。若没法通过修改电路来改善，则一直降 speaker 的增益，自录自播扫频音档，找到一个最大且不失真的增益（这一步也可通过用示波器来量功放的输出端波形来确定，此方法更加精确，但需要拆卸设备的外壳以及硬件同事的协助），以此增益来作为最大的输出增益。若在此增益上觉得 speaker 出来的声音大小不满足需求，可以使用 AO 端的 AGC 将输出信号拉高，并限制在 -3~-1dB 以下，来提高声音的大小。



5. 音效调整

当基础的 microphone 和 speaker 增益确认后，且频响特性没有太大的问题，就可以进行音效的调整。

5.1. 音效参数调整

5.1.1 确定 mic 和 speaker 的增益

如果在步骤[确认 microphone 的增益](#)因为结构设计不合理或者 microphone 的选型灵敏度太低等原因没法达到 -25DBRMS 的情况下，请按以下步骤来确定增益，若能达到 -25DBRMS，则请沿用前面选定的 microphone 和 speaker 增益。

- (1) 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
- (2) 将 microphone 和 speaker 的增益（注意 speaker 的增益包含功放的增益，若功放已有做放大，AO 的 gain 则设置成衰减使整体接近 0dB）都设定为 0dB。
- (3) 播放测试音档 TestFile3，自录自播，查看录制下来的音档（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据），确认录到的幅值比较小。（此时某些频段有失真不用太过关心，下文有讲解处理的方式）。
- (4) 播放测试音档 TestFile2，自录自播，将 microphone 增益在保证不失真的情况下调整到最大（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据）。
- (5) 调整 speaker 到符合客户要求的增益，在调整 speaker 增益后，microphone 增益也要跟随一起调整，大体遵循加一减一，speaker 加 1dB，microphone 相应要降 1dB。

在第(4)步时，可能会因为结构设计不合理，造成 mic 用很小的增益都会失真，或者说在保证不失真的前提下 mic 输出的声音振幅很小，请按以下步骤进行尝试调整：

当 microphone 增益很小就出现了失真，可以根据步骤[内部隔离](#)，确认机构是否合理。若确定为机构问题，但不想修改机构，则需要调小 speaker 的增益，根据新的 speaker 增益进行步骤[确定 mic 和 speaker 的增益](#)来确定新的 microphone 增益。或者通过调整 speaker 端来尝试消除此失真，步骤如下：

1. 寻找一个比较安静的环境，保证环境噪音低于 40dbA。
2. 将 microphone 和 speaker 的增益都设定为当前增益。
3. 播放测试音档 TestFile3，自录自播（dumpA 点处的数据，或者使用 demo 在关闭所有算法的情况下录制的的数据）。通过录制的音档查看，寻找到有失真的地方，应用工具的频率分析，找出增益比其他频段要高的频段，通过设置 AO 的 EQ 将这些频段做衰减（[如何调整 AO 的 EQ table?](#)，此方法不适用于因谐振产生的失真），重复步骤(1)和(2)，若对这些频段调整 EQ 后还有失真的情况，则只能加大到衰减值继续尝试。

当 speaker 端使用 EQ 修正 speaker 的频率响应，效果仍然没有较大改善时，可以考虑降低 microphone 的增益，以及打开 microphone 端的 ANR 和 AGC 来对数据进行调整，在 AEC 处理后的数据中找到消不掉的回声，用 ANR 和 AGC 对回声进行衰减。

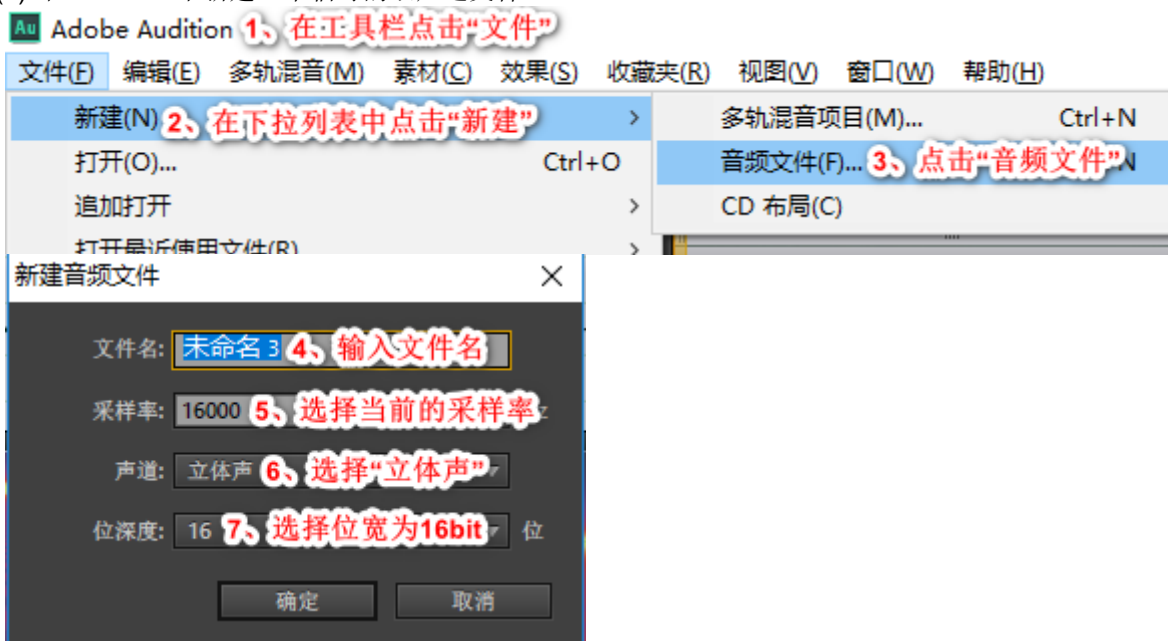
当进行以上操作得到的 microphone 数据仍然很小声的时候，只能尝试在拉大 microphone 增益，让回声爆掉的同时，加大 AEC 的消回声处理强度。（或建议改做单向对讲）（若使用 HW loopback，则转为 SW loopback）

5.1.2 调整 AO 的 EQ 参数，修正 speaker 的频响特性

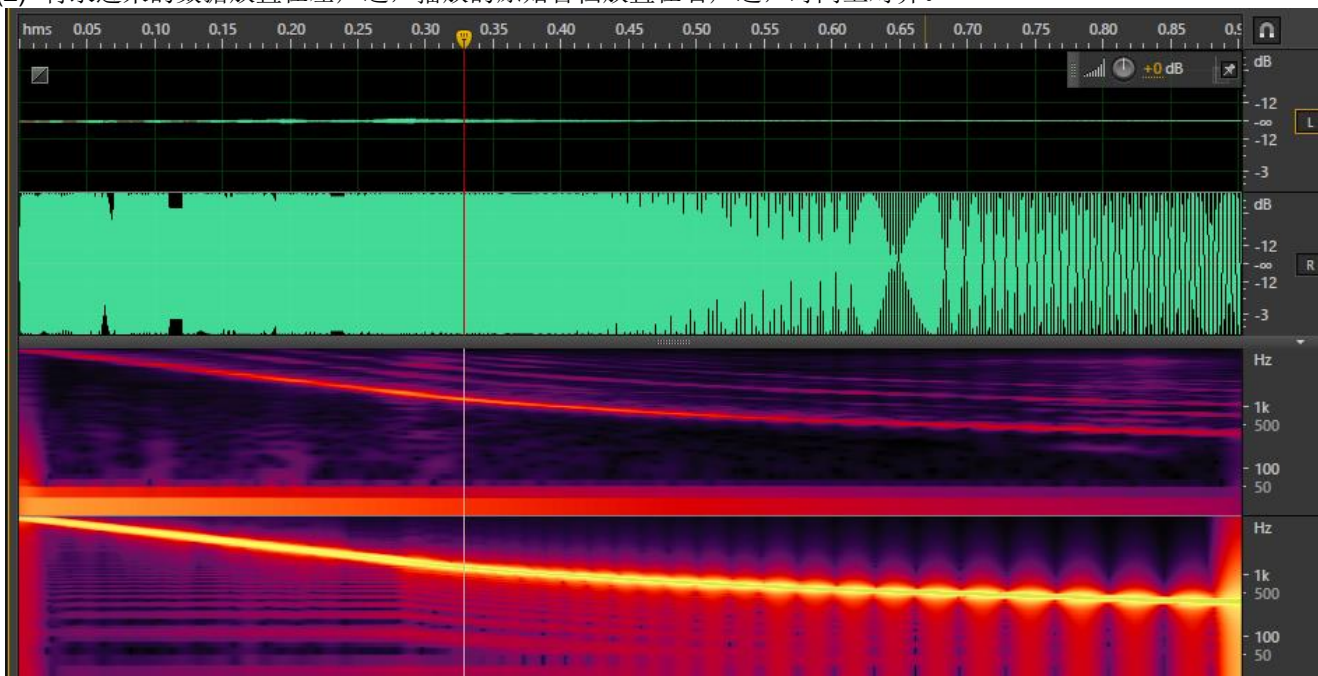
只有当 speaker 的频响特性非常差，或者说有部分固定频率的回声消不掉时，才要使用 AO 的 EQ。

如何得到 AO EQ 的调整 table 呢？当只有少部分特定频段会失真时，可以直接通过 dump 到的音档来确认是哪些频段，然后在 AO EQ 的 table 上直接填写适当的衰减来修正这些失真。但是当录制的音档频响特性较差时（当失真是由倍频引起时不建议使用），可以考虑以下方法来调整 AO EQ 的 table。（此调整是基于当前 AI/AO 的增益）

(1) 在 audition 中新建一个临时的双声道文件。

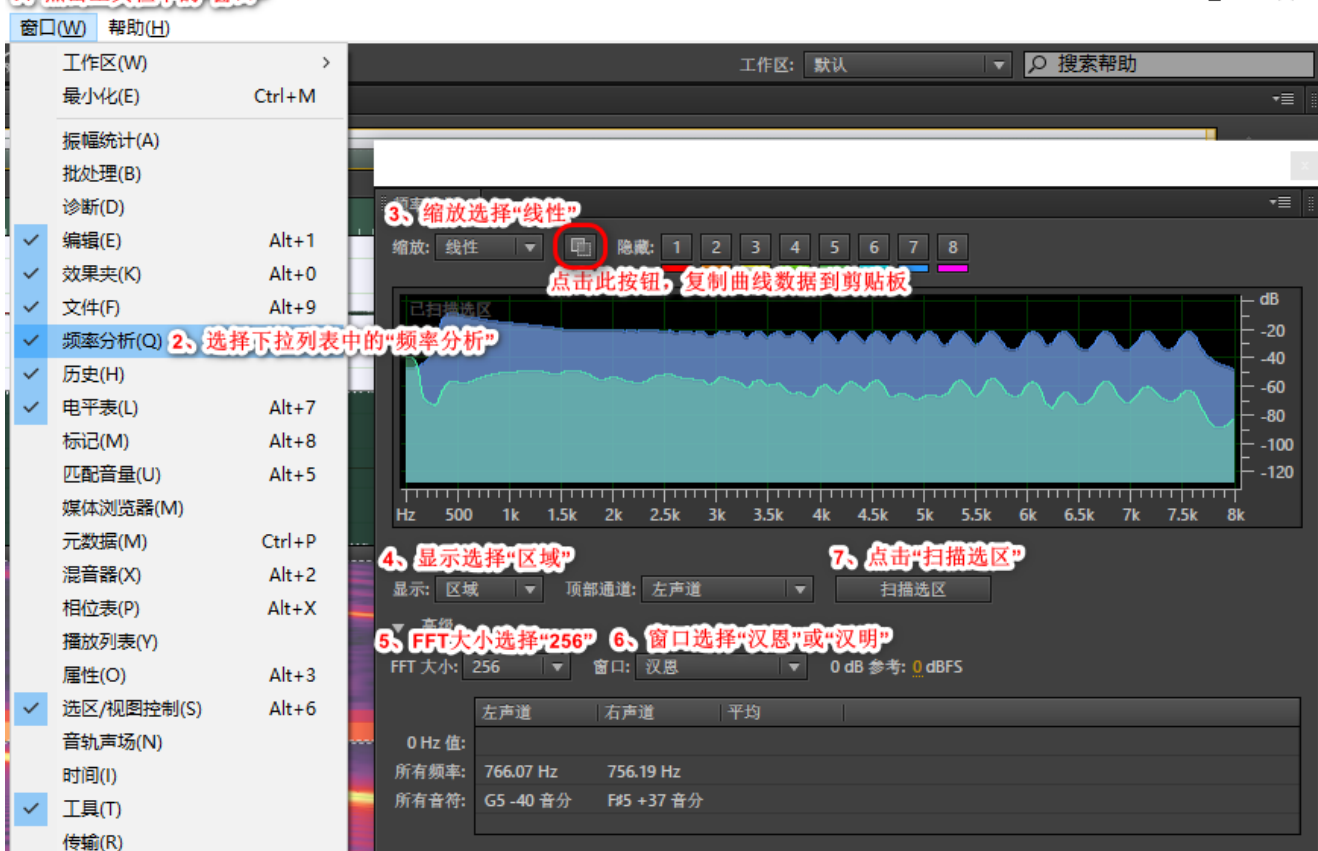


(2) 将录进来的数据放置在左声道，播放的原始音档放置在右声道，时间上对齐。



(3) 打开上述音档的频率分析，得到频响特性曲线数据。

1、点击工具栏中的“窗口”



(4) 打开 Excel，粘帖上面的曲线数据，并删除 D 栏（平均）。

| | A | B | C | D |
|----|--------|--------|--------|--------|
| 1 | Freq | 左声道 | 右声道 | 平均 |
| 2 | 0 | -39.53 | -47.37 | -42.6 |
| 3 | 62.5 | -45.47 | -46.7 | -46.06 |
| 4 | 125 | -69.22 | -44.11 | -49.66 |
| 5 | 187.5 | -73.1 | -39.43 | -45.28 |
| 6 | 250 | -74.13 | -24.79 | -30.79 |
| 7 | 312.5 | -62.95 | -11.46 | -17.45 |
| 8 | 375 | -57.42 | -7.85 | -13.84 |
| 9 | 437.5 | -57.71 | -8.4 | -14.39 |
| 10 | 500 | -58.65 | -9.4 | -15.39 |
| 11 | 562.5 | -58.05 | -10.73 | -16.71 |
| 12 | 625 | -56.06 | -11.55 | -17.52 |
| 13 | 687.5 | -54.27 | -12.4 | -18.35 |
| 14 | 750 | -52.88 | -13.27 | -19.2 |
| 15 | 812.5 | -51.9 | -14.12 | -20.03 |
| 16 | 875 | -51.27 | -14.47 | -20.37 |
| 17 | 937.5 | -51.1 | -14.78 | -20.67 |
| 18 | 1000 | -50.64 | -15.42 | -21.29 |
| 19 | 1062.5 | -50.09 | -16.09 | -21.94 |
| 20 | 1125 | -50 | -16.37 | -22.21 |
| 21 | 1187.5 | -49.69 | -16.96 | -22.79 |
| 22 | 1250 | -49.99 | -17.48 | -23.3 |
| 23 | 1312.5 | -51.03 | -17.9 | -23.73 |
| 24 | 1375 | -51.85 | -18.61 | -24.44 |
| 25 | 1437.5 | -51.7 | -19.57 | -25.38 |
| 26 | 1500 | -49.96 | -19.84 | -25.59 |
| 27 | 1562.5 | -49.32 | -19.83 | -25.57 |
| 28 | 1625 | -50.09 | -19.83 | -25.59 |

(5) 统计录进来的音档和播放音档在各个频段相差的增益值，以及相差增益值的平均值。

D2 : *fx* =B2-C2

| | A | B | C | D |
|----|-------|--------|--------|----------|
| 1 | Freq | 左声道 | 右声道 | -37.0908 |
| 2 | 0 | -39.53 | -47.37 | 7.84 |
| 3 | 62.5 | -45.47 | -46.7 | 1.23 |
| 4 | 125 | -69.22 | -44.11 | -25.11 |
| 5 | 187.5 | -73.1 | -39.43 | -33.67 |
| 6 | 250 | -74.13 | -24.79 | -49.34 |
| 7 | 312.5 | -62.95 | -11.46 | -51.49 |
| 8 | 375 | -57.42 | -7.85 | -49.57 |
| 9 | 437.5 | -57.71 | -8.4 | -49.31 |
| 10 | 500 | -58.65 | -9.4 | -49.25 |
| 11 | 562.5 | -58.05 | -10.73 | -47.32 |
| 12 | 625 | -56.06 | -11.55 | -44.51 |
| 13 | 687.5 | -54.27 | -12.4 | -41.87 |
| 14 | 750 | -52.88 | -13.27 | -39.61 |
| 15 | 812.5 | -51.9 | -14.12 | -37.78 |
| 16 | 875 | -51.27 | -14.47 | -36.8 |

D1 : *fx* =AVERAGE(D2:D129)

| | A | B | C | D | E |
|----|-------|--------|--------|----------|---|
| 1 | Freq | 左声道 | 右声道 | -37.0908 | |
| 2 | 0 | -39.53 | -47.37 | 7.84 | |
| 3 | 62.5 | -45.47 | -46.7 | 1.23 | |
| 4 | 125 | -69.22 | -44.11 | -25.11 | |
| 5 | 187.5 | -73.1 | -39.43 | -33.67 | |
| 6 | 250 | -74.13 | -24.79 | -49.34 | |
| 7 | 312.5 | -62.95 | -11.46 | -51.49 | |
| 8 | 375 | -57.42 | -7.85 | -49.57 | |
| 9 | 437.5 | -57.71 | -8.4 | -49.31 | |
| 10 | 500 | -58.65 | -9.4 | -49.25 | |
| 11 | 562.5 | -58.05 | -10.73 | -47.32 | |
| 12 | 625 | -56.06 | -11.55 | -44.51 | |
| 13 | 687.5 | -54.27 | -12.4 | -41.87 | |
| 14 | 750 | -52.88 | -13.27 | -39.61 | |
| 15 | 812.5 | -51.9 | -14.12 | -37.78 | |
| 16 | 875 | -51.27 | -14.47 | -36.8 | |
| 17 | 937.5 | -51.1 | -14.78 | -36.32 | |
| 18 | 1000 | -50.64 | -15.42 | -35.22 | |

此时得到相差增益值的平均值为-37.0908dB

(6) 用各个相差增益值减去平均值。

| | | | | | |
|----|--------|--------|--------|----------|----------|
| 1 | Freq | 左声道 | 右声道 | -37.0908 | |
| 2 | 0 | -39.53 | -47.37 | 7.84 | 44.9308 |
| 3 | 62.5 | -45.47 | -46.7 | 1.23 | 38.3208 |
| 4 | 125 | -69.22 | -44.11 | -25.11 | 11.9808 |
| 5 | 187.5 | -73.1 | -39.43 | -33.67 | 3.4208 |
| 6 | 250 | -74.13 | -24.79 | -49.34 | -12.2492 |
| 7 | 312.5 | -62.95 | -11.46 | -51.49 | -14.3992 |
| 8 | 375 | -57.42 | -7.85 | -49.57 | -12.4792 |
| 9 | 437.5 | -57.71 | -8.4 | -49.31 | -12.2192 |
| 10 | 500 | -58.65 | -9.4 | -49.25 | -12.1592 |
| 11 | 562.5 | -58.05 | -10.73 | -47.32 | -10.2292 |
| 12 | 625 | -56.06 | -11.55 | -44.51 | -7.4192 |
| 13 | 687.5 | -54.27 | -12.4 | -41.87 | -4.7792 |
| 14 | 750 | -52.88 | -13.27 | -39.61 | -2.5192 |
| 15 | 812.5 | -51.9 | -14.12 | -37.78 | -0.6892 |
| 16 | 875 | -51.27 | -14.47 | -36.8 | 0.2908 |
| 17 | 937.5 | -51.1 | -14.78 | -36.32 | 0.7708 |
| 18 | 1000 | -50.64 | -15.42 | -35.22 | 1.8708 |
| 19 | 1062.5 | -50.09 | -16.09 | -34 | 3.0908 |
| 20 | 1125 | -50 | -16.37 | -33.63 | 3.4608 |
| 21 | 1187.5 | -49.69 | -16.96 | -32.73 | 4.3608 |
| 22 | 1250 | -49.99 | -17.48 | -32.51 | 4.5808 |
| 23 | 1312.5 | -51.03 | -17.9 | -33.13 | 3.9608 |
| 24 | 1375 | -51.85 | -18.61 | -33.24 | 3.8508 |
| 25 | 1437.5 | -51.7 | -19.57 | -32.13 | 4.9608 |
| 26 | 1500 | -49.96 | -19.84 | -30.12 | 6.9708 |
| 27 | 1562.5 | -49.32 | -19.83 | -29.49 | 7.6008 |
| 28 | 1625 | -50.09 | -19.83 | -30.26 | 6.8308 |

(7) 排除掉太大的值后，转换为整数，填入到 AO EQ 的 table 中。

如图的 F 列即为填入 AO EQ 的数据。

| F2 | | =INT(E2) | | | | |
|----|--------|----------|--------|----------|----------|-----|
| | A | B | C | D | E | F |
| 1 | Freq | 左声道 | 右声道 | -37.0908 | | |
| 2 | 0 | -39.53 | -47.37 | 7.84 | 44.9308 | 44 |
| 3 | 62.5 | -45.47 | -46.7 | 1.23 | 38.3208 | 38 |
| 4 | 125 | -69.22 | -44.11 | -25.11 | 11.9808 | 11 |
| 5 | 187.5 | -73.1 | -39.43 | -33.67 | 3.4208 | 3 |
| 6 | 250 | -74.13 | -24.79 | -49.34 | -12.2492 | -13 |
| 7 | 312.5 | -62.95 | -11.46 | -51.49 | -14.3992 | -15 |
| 8 | 375 | -57.42 | -7.85 | -49.57 | -12.4792 | -13 |
| 9 | 437.5 | -57.71 | -8.4 | -49.31 | -12.2192 | -13 |
| 10 | 500 | -58.65 | -9.4 | -49.25 | -12.1592 | -13 |
| 11 | 562.5 | -58.05 | -10.73 | -47.32 | -10.2292 | -11 |
| 12 | 625 | -56.06 | -11.55 | -44.51 | -7.4192 | -8 |
| 13 | 687.5 | -54.27 | -12.4 | -41.87 | -4.7792 | -5 |
| 14 | 750 | -52.88 | -13.27 | -39.61 | -2.5192 | -3 |
| 15 | 812.5 | -51.9 | -14.12 | -37.78 | -0.6892 | -1 |
| 16 | 875 | -51.27 | -14.47 | -36.8 | 0.2908 | 0 |
| 17 | 937.5 | -51.1 | -14.78 | -36.32 | 0.7708 | 0 |
| 18 | 1000 | -50.64 | -15.42 | -35.22 | 1.8708 | 1 |
| 19 | 1062.5 | -50.09 | -16.09 | -34 | 3.0908 | 3 |
| 20 | 1125 | -50 | -16.37 | -33.63 | 3.4608 | 3 |
| 21 | 1187.5 | -49.69 | -16.96 | -32.73 | 4.3608 | 4 |
| 22 | 1250 | -49.99 | -17.48 | -32.51 | 4.5808 | 4 |
| 23 | 1312.5 | -51.03 | -17.9 | -33.13 | 3.9608 | 3 |
| 24 | 1375 | -51.85 | -18.61 | -33.24 | 3.8508 | 3 |
| 25 | 1437.5 | -51.7 | -19.57 | -32.13 | 4.9608 | 4 |
| 26 | 1500 | -49.96 | -19.84 | -30.12 | 6.9708 | 6 |
| 27 | 1562.5 | -49.32 | -19.83 | -29.49 | 7.6008 | 7 |
| 28 | 1625 | -50.09 | -19.83 | -30.26 | 6.8308 | 6 |

5.1.3 AEC 频段和强度调整

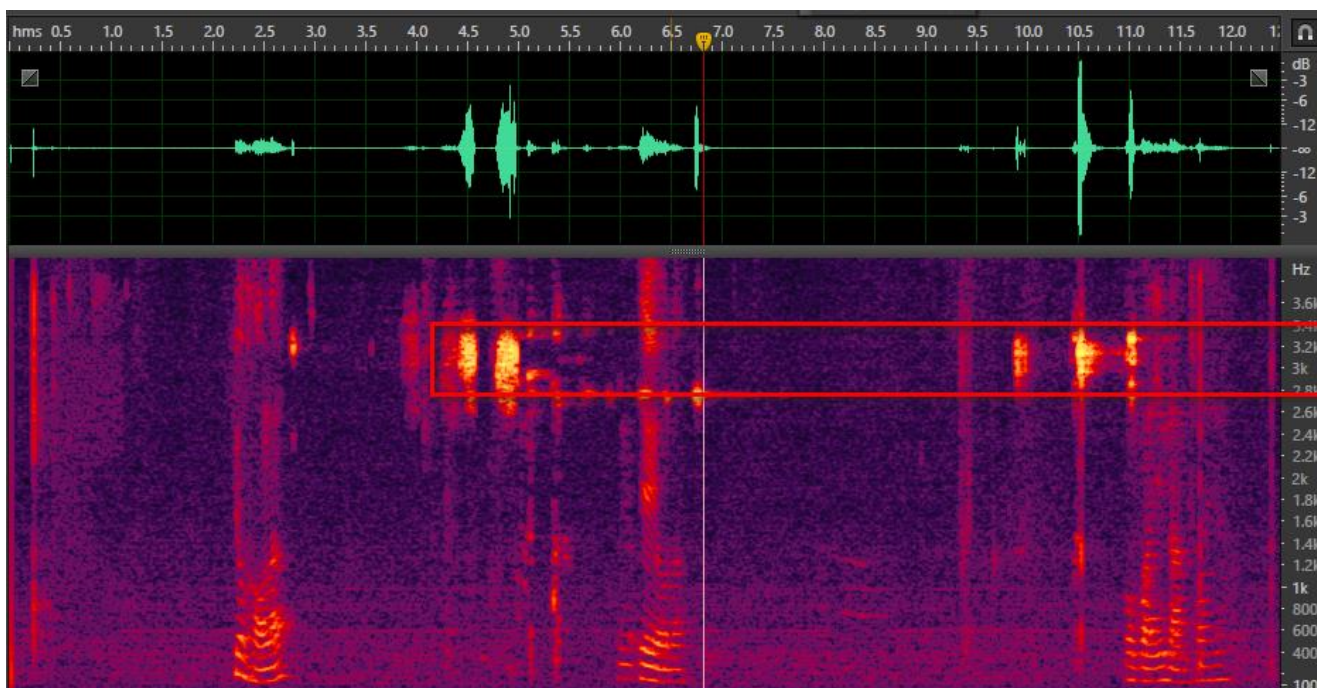
当我们确定好 speaker 和 microphone 的增益后，就需要对 AEC 的频段和强度进行调整。

- (1) 将 AEC 的频段划分和强度设置成以下参数，并进行单向对讲（可用自录自播 TestFile2 进行模拟）。
`MI_U32 u32AecSupfreq[6] = {20,40,60,80,100,120};`
`MI_U32 u32AecSupIntensity[7] = {4,4,4,4,4,4,4};`
- (2) dump 得到 AEC 处理后的音档，查看音档是否有消不干净的回声，有则找到回声对应的频段，对上面的频段划分进行调整，将回声的频率范围调整成一个频段，设置到 u32AecSupfreq 数组中，同时调整 u32AecSupIntensity 数组，增强对此频段的消回声强度。
- (3) 当从 dump 得到的 AEC 处理后的音档听到有数据被误砍了，可以将 AecAIn、AecAoIn、AecOut 三个音档放到 Adobe audition 里面对比，看被误砍到的频段是否对应的时间点 AecAoIn 中是否有较大的能量，若没有则调整 u32AecSupIntensity 数组，降低该频段的 AEC 消除强度。

例子：如下图所示，通过音档 AecOut（听音档、看频谱）我们发现在 2.8K 到 3.4K 的频段回声比较严重，我们可以将 2.8K~3.4K 划分为一个 AEC 消除的一个段，调整该范围的 AEC 消除强度。2800/31.25≈89，3400/31.25≈108，故我们可以将 AEC 的参数调整为

```
MI_U32 u32AecSupfreq[6] = {20,40,60,89,108,120};
```

```
MI_U32 u32AecSupIntensity[7] = {4,4,4,4,6,4,4}; (AEC 的消除强度则需要根据实际情况来尝试)
```



注：当有某些固定的频段的回声一直消不掉时，可以使用 mic 端的 EQ 对这些频段来做衰减，来消掉这些回声。

5.1.4 ANR 参数调整

当我们录进来的声音因为环境或硬件问题存在持续的噪声时，则可以使用 ANR 来进行降噪，但是 ANR 没法处理那些突然出现的噪声。测试 ANR 最近先不要说话，让 ANR 收一段音作为参考。

- (1) 将 ANR 的默认参数按以下设置，在 Vqe 设置中只使能 AEC 和 ANR，抓取 APC 算法前后的音档。

```
eMode = E_MI_AUDIO_ALGORITHM_MODE_MUSIC;
```

```
u32NrIntensity = 20;
```

```
u32NrSmoothLevel = 10;
```

```
eNrSpeed = E_MI_AUDIO_NR_SPEED_MID;
```

或

```
eMode = E_MI_AUDIO_ALGORITHM_MODE_MUSIC;
u32NrIntensityBand[6] = {20,40,60,80,100,120};
u32NrIntensity[7] = {20,20,20,20,20,20,20};
u32NrSmoothLevel = 10;
eNrSpeed = E_MI_AUDIO_NR_SPEED_MID;
```

- (2) 查看 APC 算法处理后的音档。
- (3) 若觉得 ANR 算法收敛的时间过长（即音档中噪声从开始到稳定下来的时间），则增大 eNrSpeed。当把 eNrSpeed 设置到最大时，收敛时间仍然没法接受，可将 AEC 的 bComfortNoiseEnable 打开，让 AEC 加入噪声辅助 ANR 收敛。
- (4) 若 ANR 算法收敛后，仍觉得噪声没法接受，则增大 u32NrIntensity。
- (5) 当遇到环境中只有某些频段噪声比较大，可以使用带频率划分的 NR 参数（u32NrIntensityBand），单独针对某个频段进行降噪强度的调整（需当前 SDK 版本支持）。

5.1.5 AGC 参数调整

AGC 的参数设置完全依赖 AEC, ANR, EQ 等算法的处理效果来进行调整，使用音频分析工具来看录制下来的音档，根据客户的需求对各个增益范围进行处理。

注：当使用 AGC 后，当开启 AGC 后输出信号没有任何变化时，请检查 Gain info、release time、attack time、曲线斜率、noise gate 是否设置正确。若开启 AGC 后，。

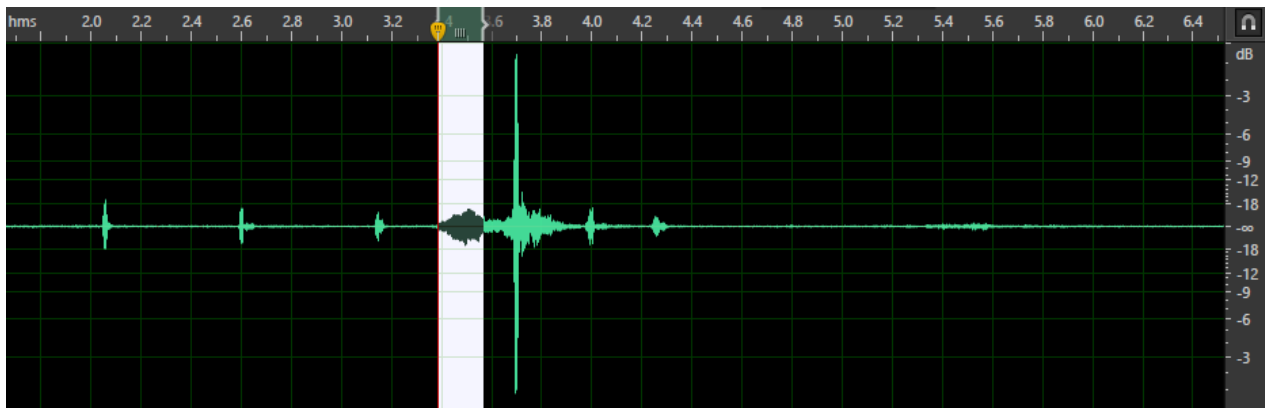
- (1) 使用一下参数作为 AGC 起始参数：

```
compression_ratio_input[AGC_CR_NUM] = {-80,-60,-40,-20,0};
compression_ratio_output[AGC_CR_NUM] = {-80,-40,-20,-10,-5};
user_mode = 1;
gain_max = 30;
gain_min = 0;
gain_init = 0;
drop_gain_max = 36;
attack_time = 1;
release_time = 3;
noise_gate_db = -80;
noise_gate_attenuation_db = 0;
target_level_db = -5;
```

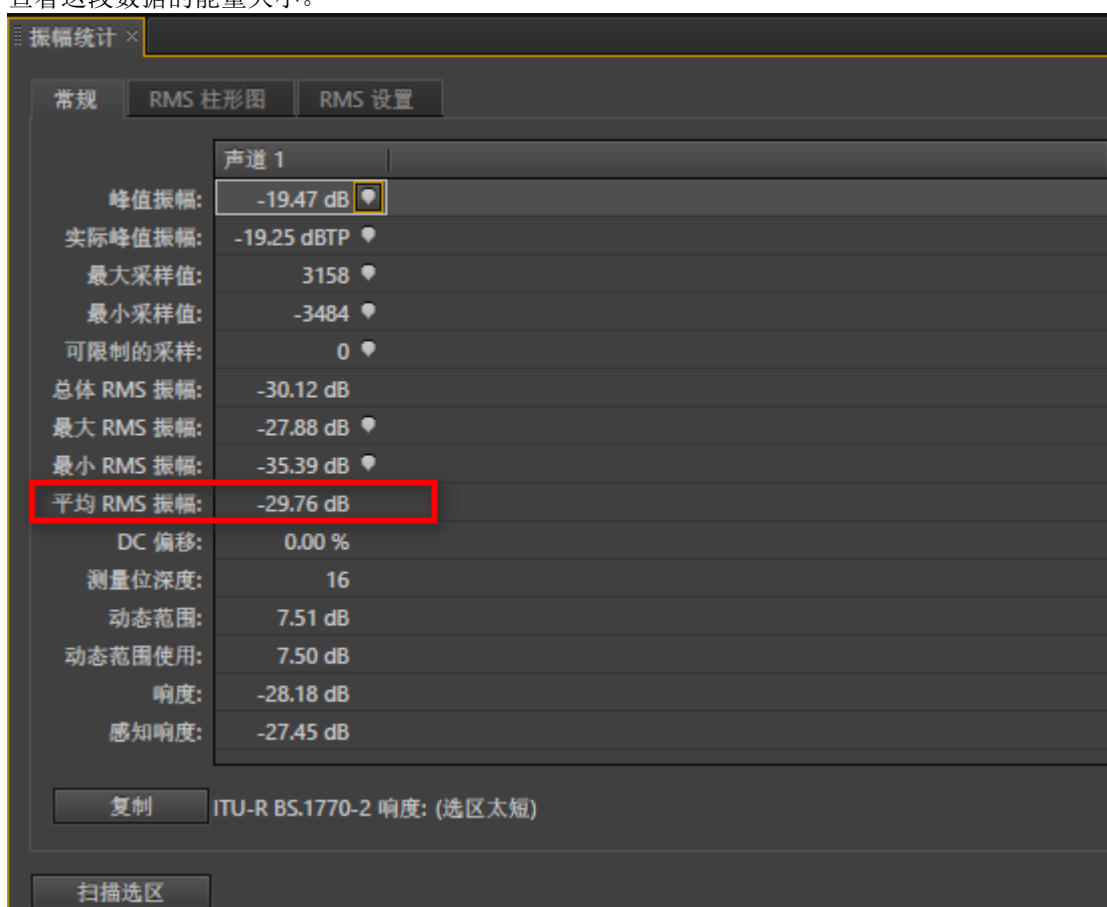
- (2) 将 Vqe 的 AEC,ANR,(EQ, 若有使用 EQ)使能，dump 出 APC 算法处理后的数据查看，分析出要增强的声音和要抑制的声音分别在哪个 dB 范围内（[查看增益](#)），根据应用需求调整曲线的斜率，即想要增强的部分，增加曲线的斜率，想要抑制的部分，减小曲线的斜率，来初步调整曲线。
- (3) 将 Vqe 的 AEC,ANR,(EQ, 若有使用 EQ), AGC 使能，dump 出 APC 算法处理后的数据查看，是否符合预期。
- (4) 若数据存在大量削顶时，可加大 drop_gain_max、减小对应曲线的斜率、减小 target_level_db、以及增加 release_time 等方式来缓解。
- (5) 若数据的增益没有达到要求，可增加对应曲线的斜率、减小 release_time、增大 gain_max 等方式解决。
- (6) 若数据没有任何变化，请检查参数是否设置正确。

例子：下面图例说明怎么将调整 AGC 参数将语音信号的增益拉起来。

当我们需要拉起某个能量范围的信号时，需先从音档中选出需要调整的数据



查看这段数据的能量大小。



假设我们想把这个信号拉起来，可以通过调整 AGC 的曲线的来达到这个目的。由上图得到的平均振幅，可以知道该信号在曲线的-40dB~-20dB 的段上，那最简单的做法就是将该端的斜率拉起来。这样相当于-40dB~-20dB 的信号拉到-20dB 到-5dB。

```
compression_ratio_input[AGC_CR_NUM] = {-80,-60,-40,-20,0};
```

```
compression_ratio_output[AGC_CR_NUM] = {-80,-40,-20,-5,-5};
```

6. 音效调整结构设计建议以及硬件选型

6.1. Speaker 和 microphone 的选型建议

- (1) microphone 最好选择灵敏度在-35dB 以下的，但不建议太小，以免达不到收音距离的要求。
- (2) 必须选择带有外壳的 speaker。

6.2. 结构设计建议

- (1) Microphone 和 speaker 之间的距离最好在 4~10cm，距离越远效果越好。
- (2) 禁止在结构中使用裸露的 speaker，speaker 外壳前后之间的紧密连接有助于降低声音从后腔到扬声器前部的传播。
- (3) Speaker 的外壳必须牢牢地固定在空腔内，可以避免扬声器产生咔哒咔哒的杂音。
- (4) 当将 speaker 安装到结构内时，可使用高密度泡沫橡胶（防震，隔音）来使 speaker 更加紧密牢固。
- (5) 结构上 speaker 位置的开孔至少要达到 speaker 面积的 20%以上，且开孔应距离 speaker 1~2mm。
- (6) 可在 microphone 后面加上泡沫橡胶，来减少 speaker 和 microphone 的直接耦合。更进一步可以将麦克风和泡沫橡胶装到一个独立的外壳内。
- (7) Microphone 必须对准机壳上的开孔。

7. 基本调试流程

以下记录一次简单的调试过程，好让读者了解大概调整的流程，以下参数仅适用用某款机型，不具有参考意义。

第一步，确定 AO 的增益。

根据 [3.2.1 确认 speaker 的增益](#)，在设备上使用 demo 播放 TestFile1 和 TestFile2，根据客户的要求选择一个足够大声的增益。（因此机型客户没有给出具体要求 speaker 要多响，这里自行先选择一个合适的增益）。此处选择 0dB 作为 AO 的增益。

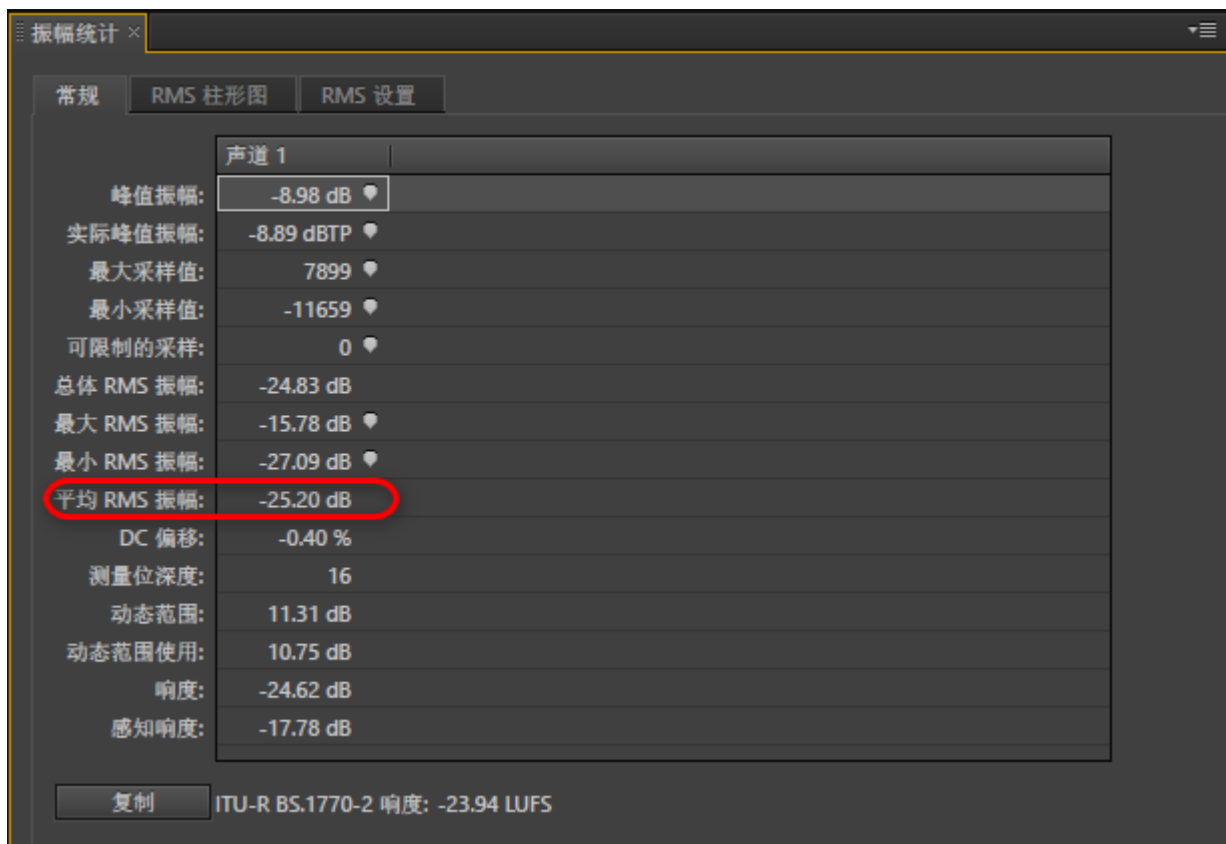
第二步，确定 AI 的增益。

用上述选定的初始增益，跟据 [3.2.2 确认 microphone 的增益](#)，将设备放置和分贝仪一同放置在桌上，在距离设备一米处，放置一个可播放音频的设备，使用播放设备播放 TestFile1，调整播放设备的音量，使分贝仪能达到 70dBA。此时逐步调节 microphone 的增益，使收进来的声音足够大声，最好收进来的声音能到-25DBRMS。由于手上的机型 mic 的灵敏度比较低，故 mic 增益设定到 54dB 才能勉强达到接近-25DBRMS。

第三步，测试结构的内部隔音效果。

用上述选定的初始增益，按照 [3.3.内部隔离](#) 的步骤测量，内部的隔音效果如何。自录自播 TestFile1，得到平均 RMS 振幅，记为 AVGRMS1，堵住 mic 孔，重复上面的操作，记为 AVGRMS2，内部隔音 = AVGRMS1 - AVGRMS2。下图分别为自录自播 TestFile1 的平均 RMS 振幅和堵住 mic 孔自录自播 TestFile1 的平均 RMS 振幅。





由此可见内部的隔音效果满足要求。

第四步，测试结构的回声损耗。

按照 [3.4.回音损耗](#) 的步骤测量，回声损耗的效果如何。TestFile1 音档本身的平均 RMS 振幅记为 AVGRMS1，自录自播 TestFile1，得到平均 RMS 振幅，记为 AVGRMS2。下图分别为 AVGRMS1 和 AVGRMS2。由此可见该机构对回声几乎没有损耗，设计较差。

The image displays two screenshots of the 'Amplitude Statistics' (振幅统计) window, showing audio measurement data for 'Channel 1' (声道 1). The 'Average RMS Amplitude' (平均 RMS 振幅) is highlighted with a red circle in both screenshots.

Top Screenshot Data:

| 声道 1 | 声道 2 |
|-----------------------------|------|
| 峰值振幅: -4.89 dB | |
| 实际峰值振幅: -4.47 dBTP | |
| 最大采样值: 17360 | |
| 最小采样值: -18661 | |
| 可限制的采样: 0 | |
| 总体 RMS 振幅: -18.28 dB | |
| 最大 RMS 振幅: -14.60 dB | |
| 最小 RMS 振幅: -21.44 dB | |
| 平均 RMS 振幅: -18.37 dB | |
| DC 偏移: -0.01 % | |
| 测量位深度: 16 | |
| 动态范围: 6.83 dB | |
| 动态范围使用: 4.65 dB | |
| 响度: -18.20 dB | |
| 感知响度: -13.59 dB | |

复制 ITU-R BS.1770-2 响度: -18.88 LUFS

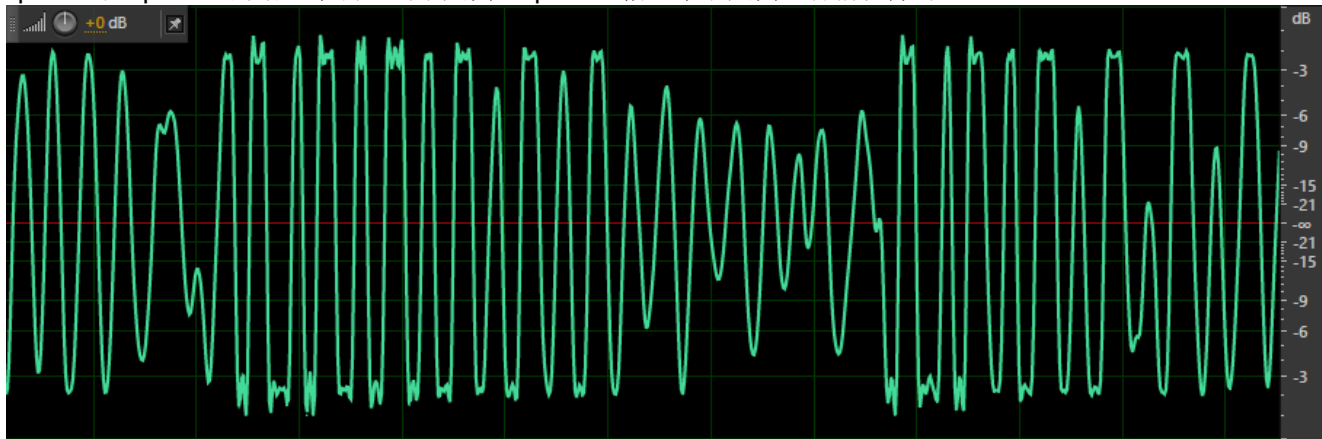
Bottom Screenshot Data:

| 声道 1 | 声道 2 |
|-----------------------------|------|
| 峰值振幅: -4.34 dB | |
| 实际峰值振幅: -4.34 dBTP | |
| 最大采样值: 19876 | |
| 最小采样值: -18237 | |
| 可限制的采样: 0 | |
| 总体 RMS 振幅: -17.81 dB | |
| 最大 RMS 振幅: -16.16 dB | |
| 最小 RMS 振幅: -36.16 dB | |
| 平均 RMS 振幅: -18.19 dB | |
| DC 偏移: 0.17 % | |
| 测量位深度: 16 | |
| 动态范围: 20.00 dB | |
| 动态范围使用: 19.00 dB | |
| 响度: -17.70 dB | |
| 感知响度: -9.70 dB | |

复制 ITU-R BS.1770-2 响度: -15.87 LUFS

第五步，调整 AI 和 AO 的增益。

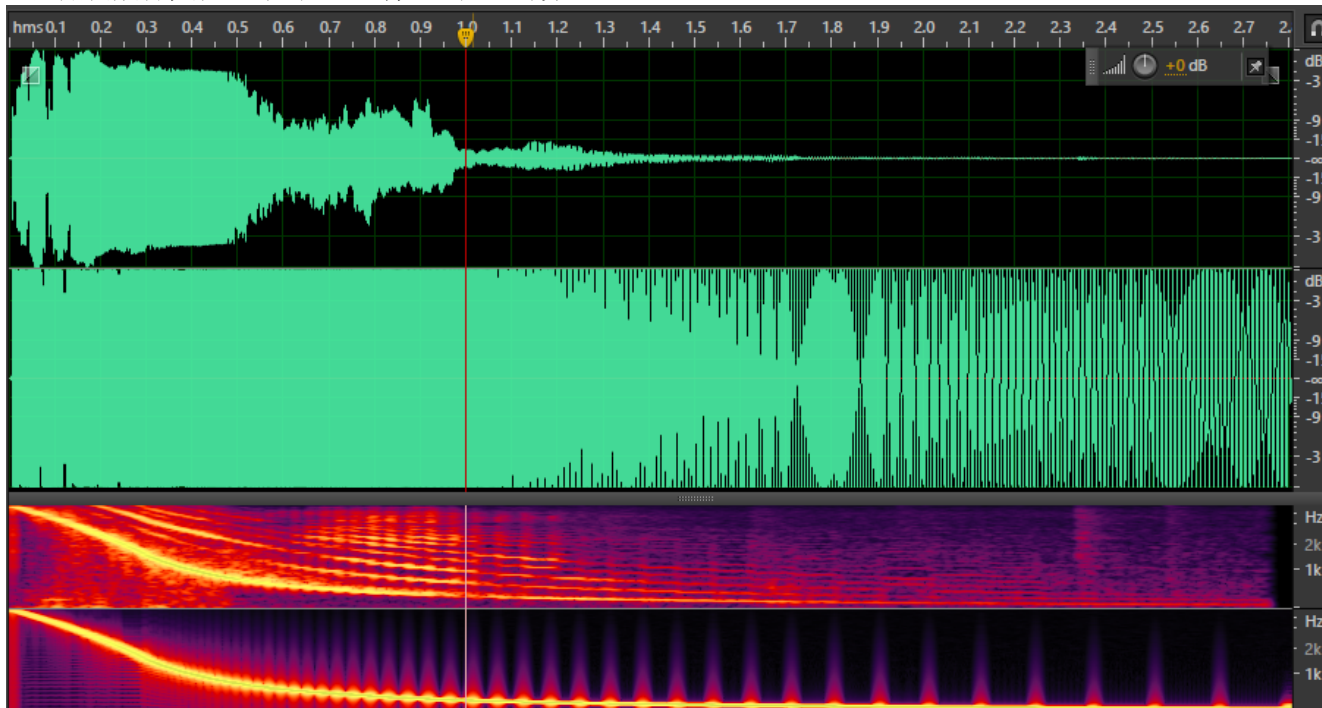
微调 AI 和 AO 的增益，使自录自播 TestFile2，尽量不会出现失真。若有出现失真，这可以通过减小 microphone 和 speaker 的增益来调整到不失真（speaker 播出来就失真这种情况除外）。



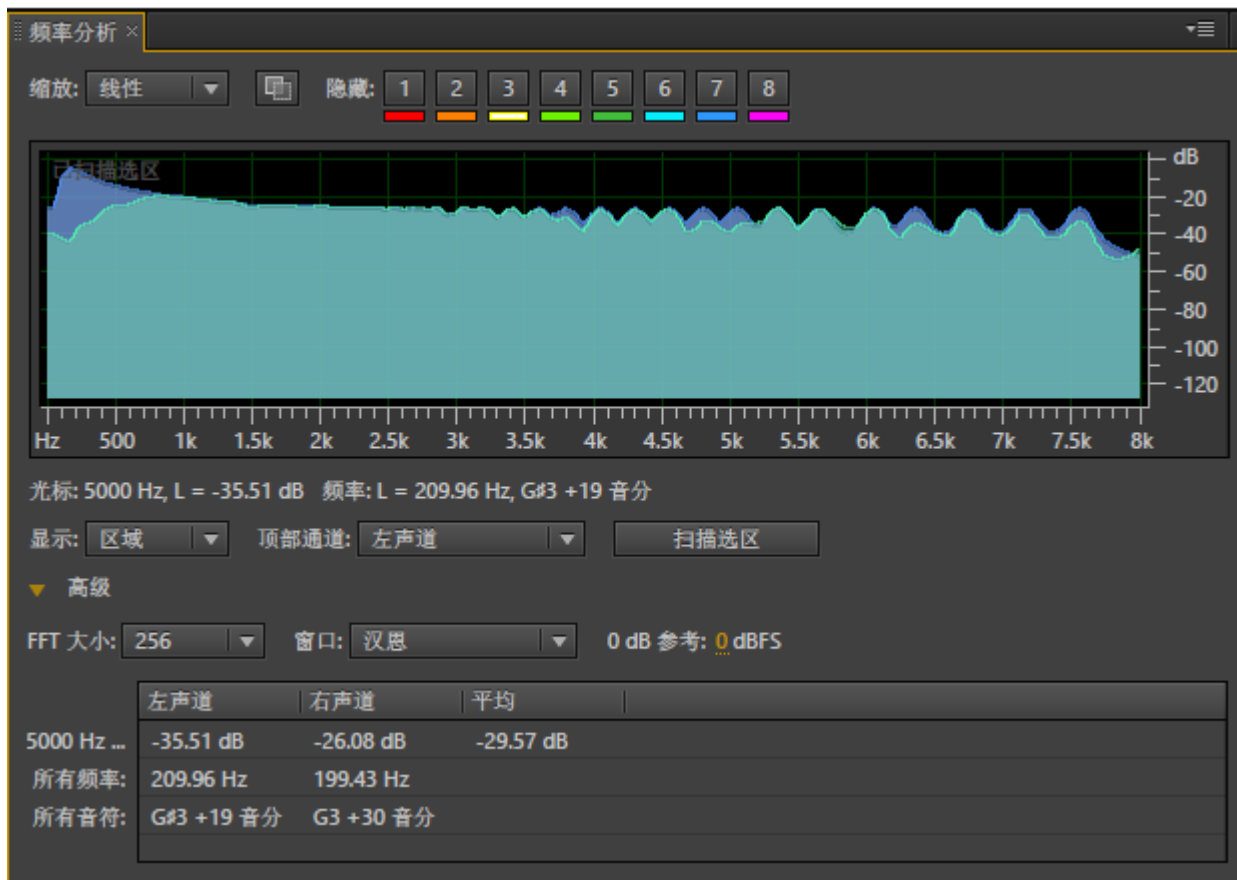
由上图看出部分波形已经削顶了，我们需要对 microphone 和 speaker 的增益进行调整。用分贝计在距离一米处进行测试，发现 speaker 播放 TestFile2 时，已经达到约 75dBA，这个值已经是比较大的了，对 speaker 增益进行调整，调整到分贝计显示 65dBA 左右。经过调整后确定 microphone 和 speaker 增益为 48dB 和 -9dB。

第六步，对 speaker 进行扫频，修正 speaker 的 EQ。

首先我们在上面确定的增益基础上，自录自播扫频音档 TestFile3。创建一个双声道的文件，将录制的放在左声道，原始扫频音档放置到右声道，将左右声道对齐。



点击频率分析，设置相应的参数得到该音档的频响特性。

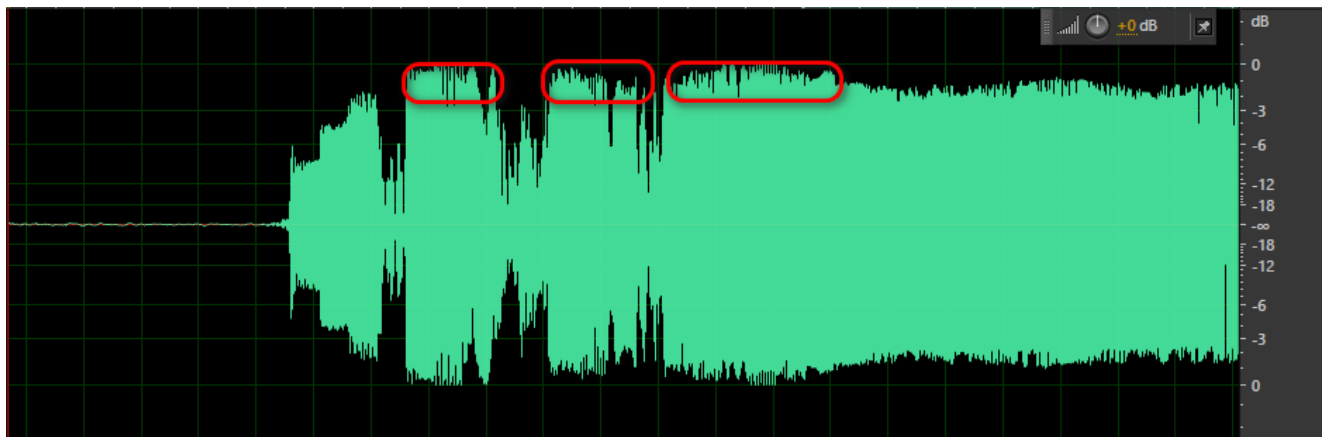


得到曲线的信息，复制到 Excel 中，计算出 AO 的 EQ table。

```
<EqGainDb>
-12,-31,-37,-28,-23,-19,-12,-9,-7,-5,
-2,0,1,2,2,2,3,3,2,2,
3,3,2,2,2,3,3,3,2,2,
2,3,3,3,2,2,3,2,3,3,
3,4,3,3,3,3,4,3,3,3,
3,3,3,3,3,3,2,2,1,-2,
-3,-4,-1,1,2,2,2,2,2,2,
4,3,2,1,-2,-6,-4,-2,0,-8,
-7,-2,5,4,4,3,2,2,2,2,
3,4,6,5,4,3,2,2,-1,-5,
-6,-5,-3,1,2,0,2,1,0,1,
1,1,0,-1,-1,-1,-1,-1,-2,-4,
-4,-4,-5,-6,-4,-2,2,7,0
</EqGainDb>
```

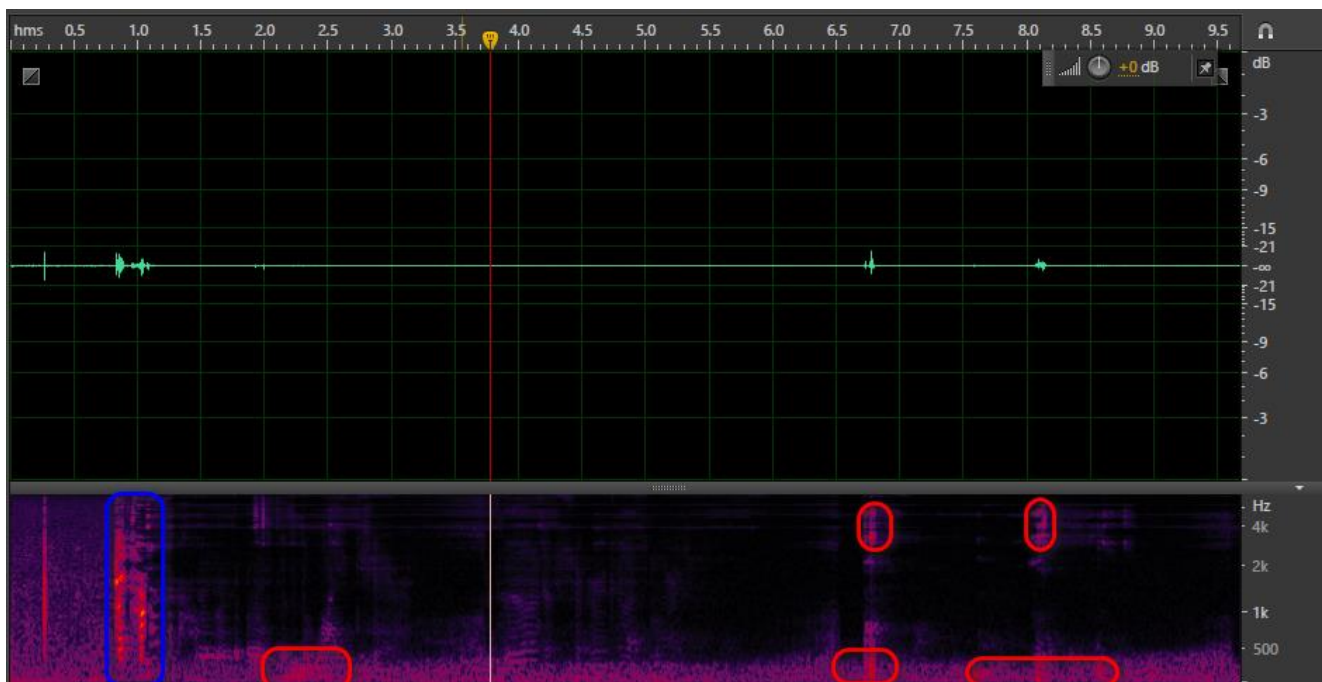
填入到 AO 的 EQ table 中。

开启 AO 的 EQ，同样自录自播 TestFile3。发现以下频段的增益太大了，对相应的频段继续做衰减。

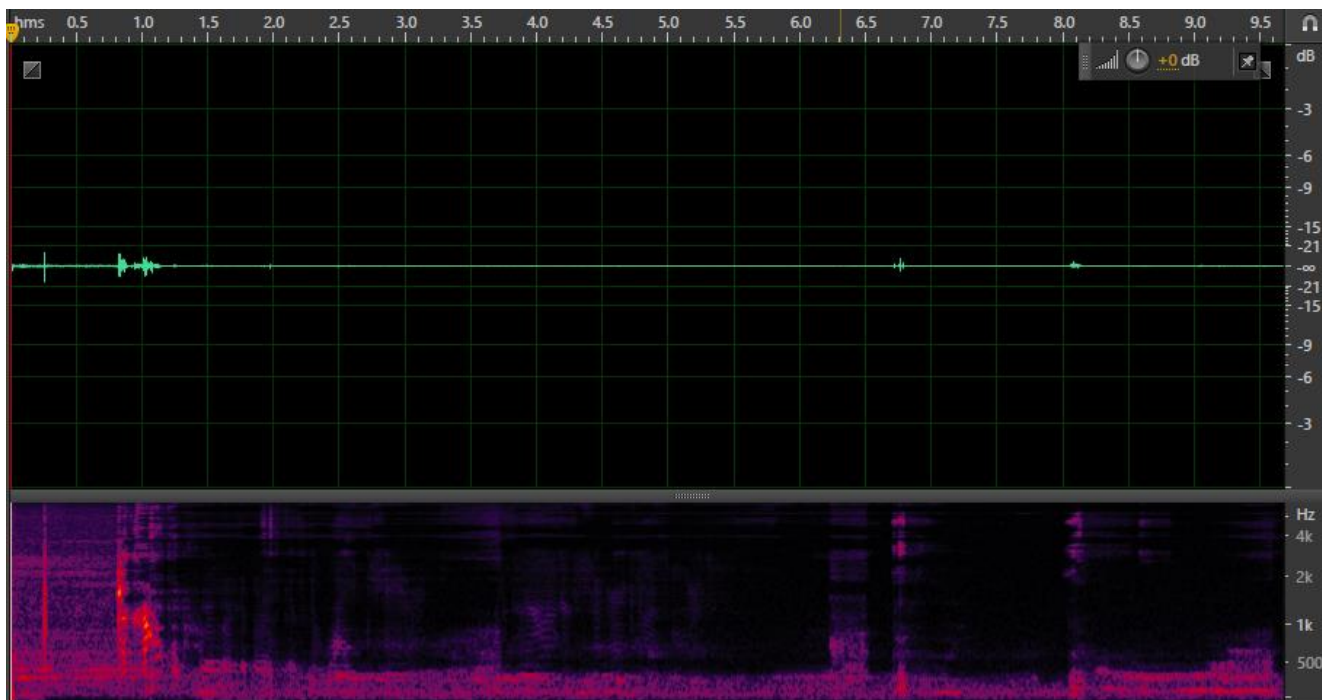


第七步，调整 AEC 参数。

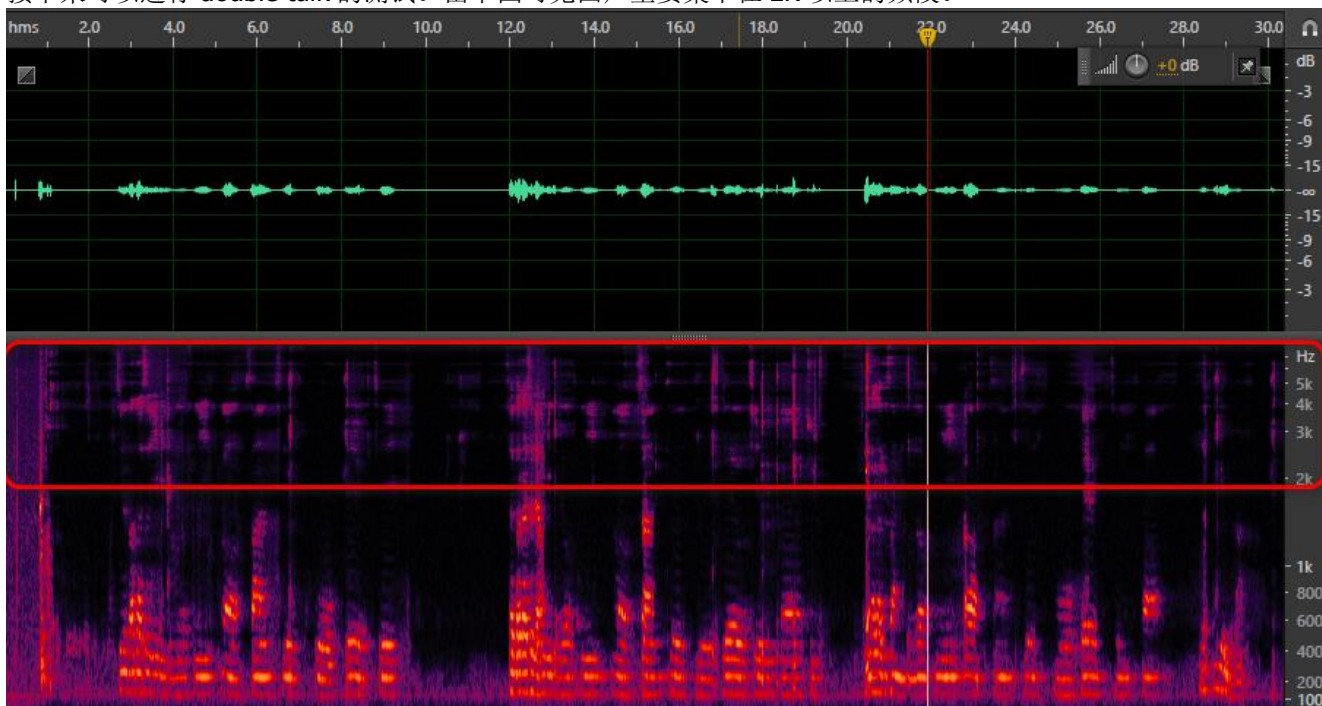
首先我们填入 AEC 的初始化参数，设置环境变量，先进行 single talk 测试，自录自播 TestFile2，录出 AEC 的相关音档。由 AecOut 的音档可以看出消不掉基本只有红色标记的两个频段，蓝色标记处为 AEC 算法还没收敛，不用管，接着调整 AEC 的参数，对这两个频段的 AEC 强度。



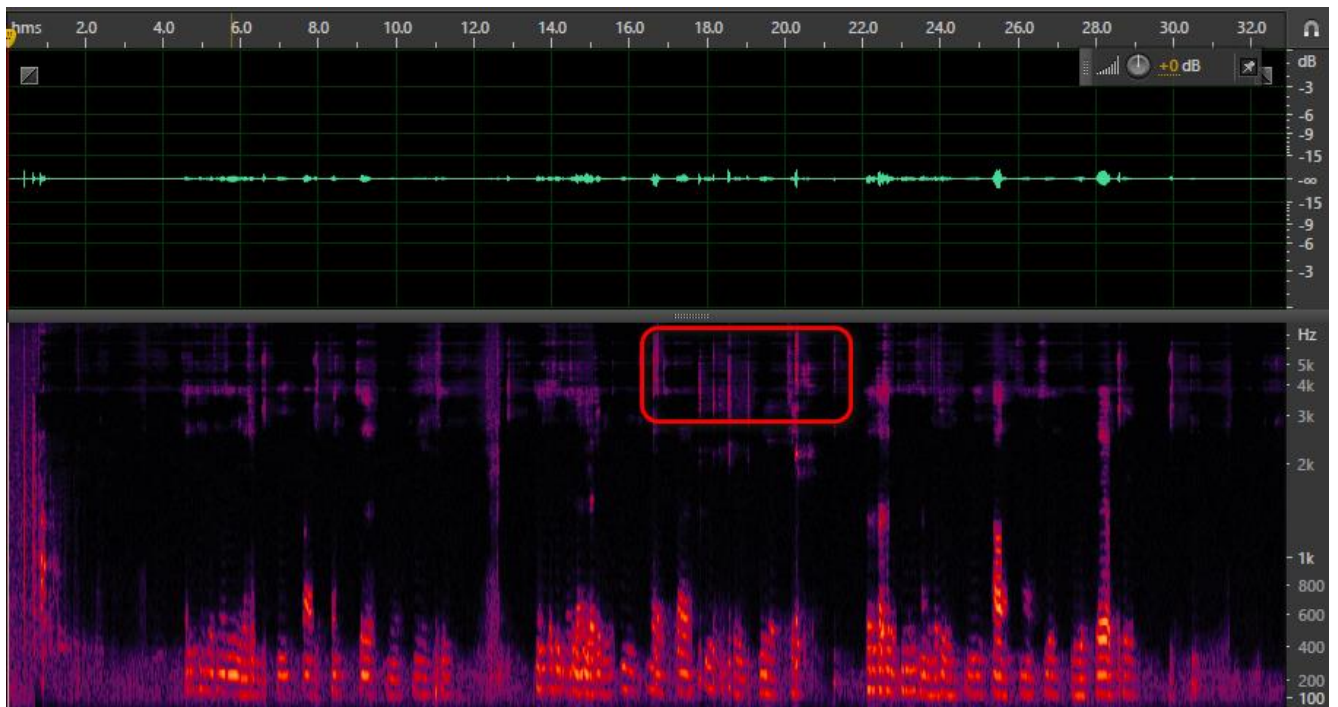
调整参数后，录制的 AecOut 音档如下：



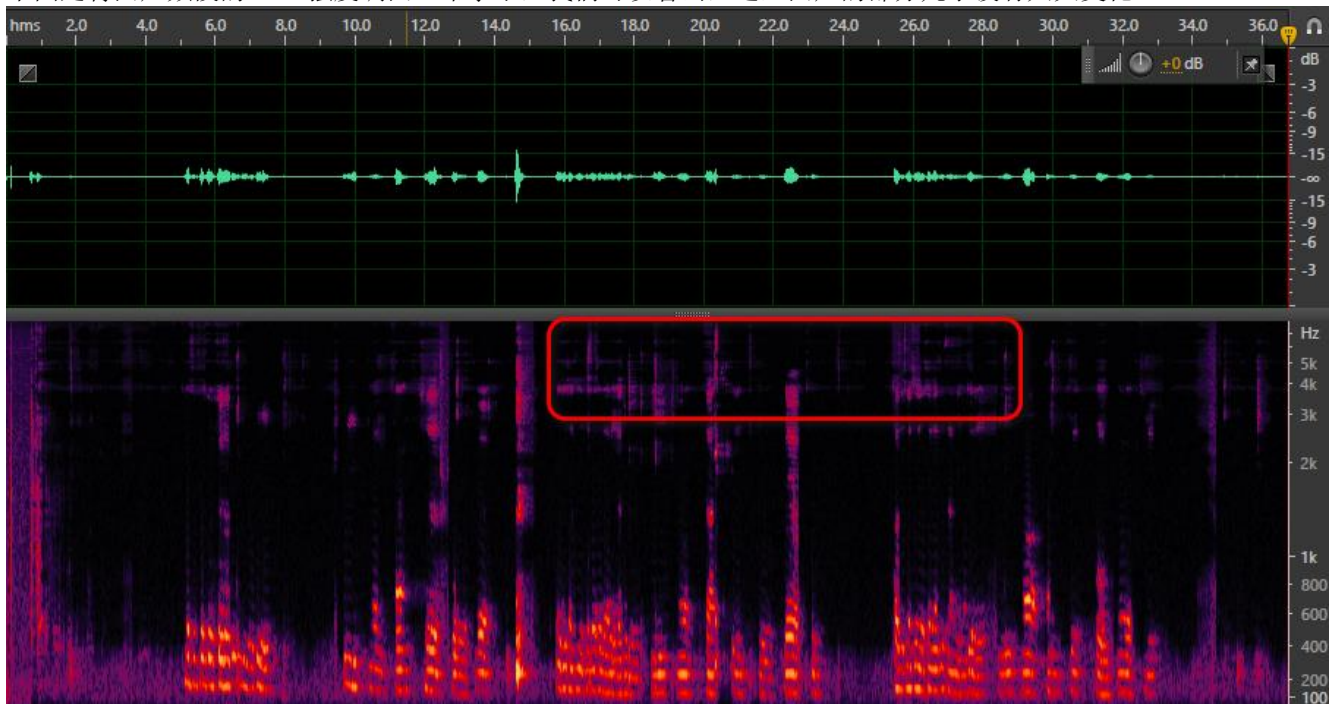
接下来可以进行 double talk 的测试。由下图可见回声主要集中在 2K 以上的频段。



继续调整参数，将以上频段加强。后面发现即使我们怎么调 AEC 的强度此处的回声都没法消掉，即使将 AEC 的强度开到了最大，仍然没法消掉这个部分的回声。只能将参数调整至 microphone 收音仍然比较大，且回声较小的情况，后面利用 EQ 来尝试消掉。



下图是将回声频段的 AEC 强度调回正常水平，我们可以看出，这些回声的部分几乎没有太大变化。



第七步，调整 AI NR 参数。

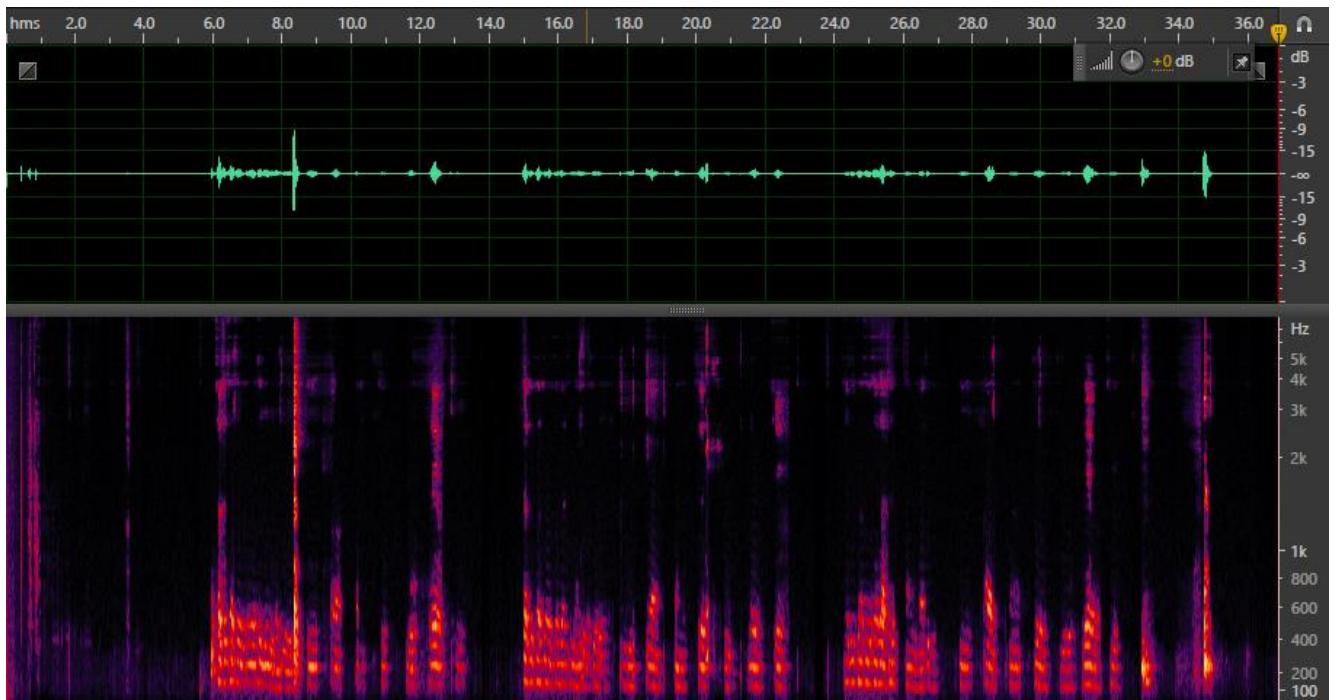
此处基本上我们将默认参数填入即可，当发现因电路和测试环境等因素，发现降噪效果不理想时，再做微调。

eMode: E_MI_AUDIO_ALGORITHM_MODE_MUSIC

u32NrIntensity: 20

u32NrSmoothLevel: 10

eNrSpeed: E_MI_AUDIO_NR_SPEED_MID



第八步，调整 AI EQ 参数。

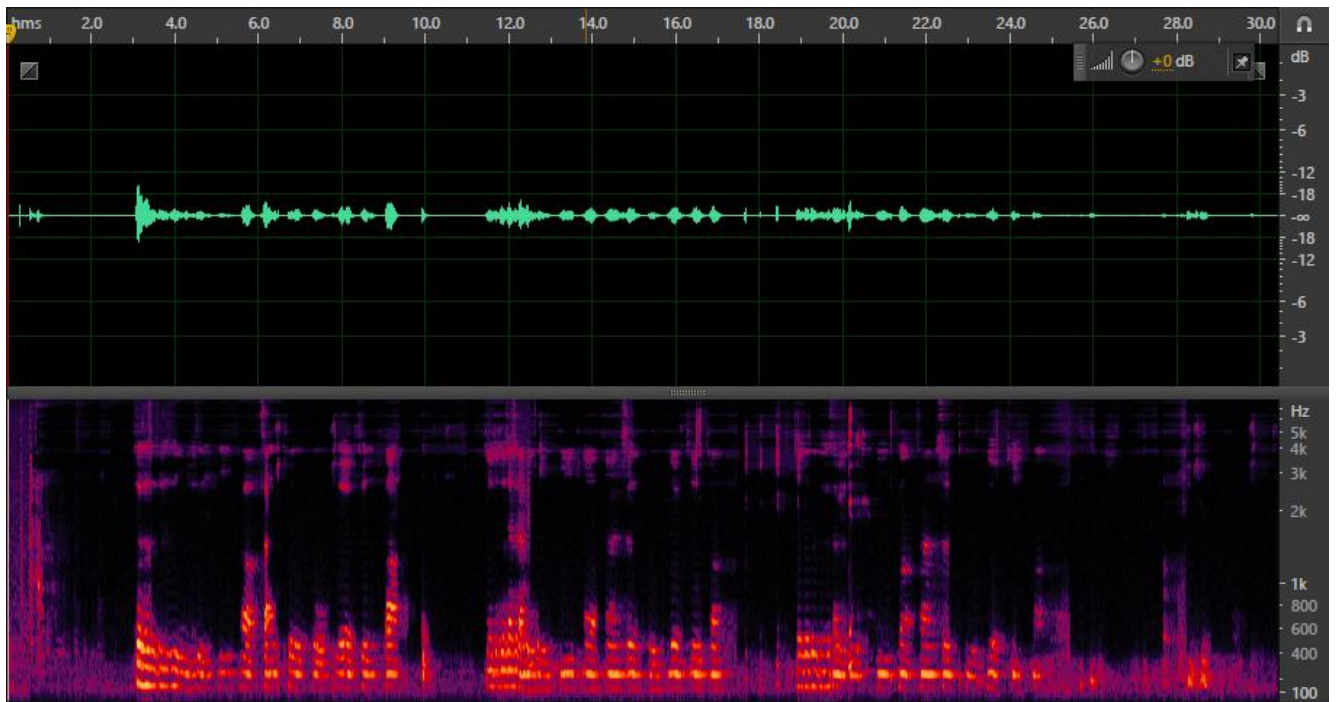
除了在某些频段的噪声消不掉，其他情况不建议直接使用 AI 的 EQ 来进行调整增益的处理。

第九步，调整 AI AGC 参数。

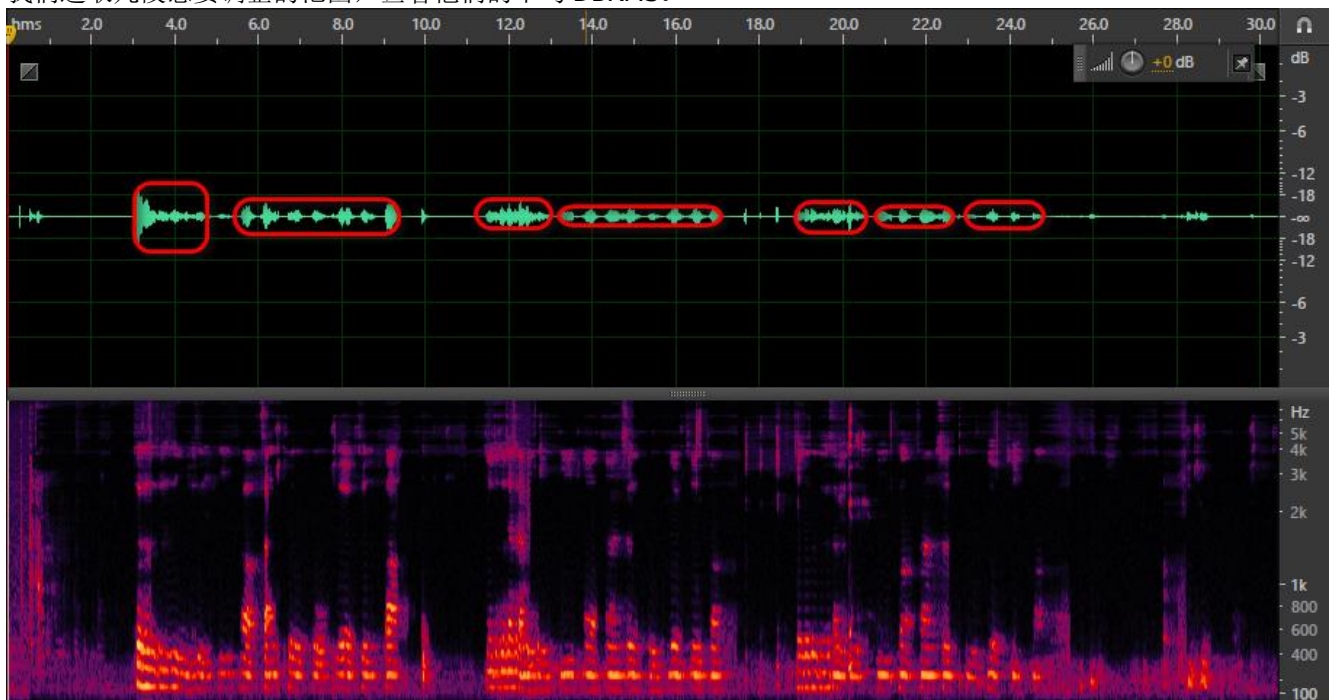
跟 AEC 一样，我们先填入起始参数再来进行微调。起始参数如下：

```
compression_ratio_input[AGC_CR_NUM] = {-80,-60,-40,-20,0};
compression_ratio_output[AGC_CR_NUM] = {-80,-40,-20,-10,-5};
user_mode = 1;
gain_max = 30;
gain_min = 0;
gain_init = 0;
drop_gain_max = 36;
attack_time = 1;
release_time = 3;
noise_gate_db = -60;
noise_gate_attenuation_db = 0;
target_level_db = -5;
```

接着我们来分析 AEC 处理后的音档



我们选取几段想要调整的范围，查看他们的平均 DBRMS。



再更改默认参数曲线中的 output 参数，调整到想到的 dB 值。调整如下：



以上突然飙高的地方均为喇叭失真导致的突然出现且没法消除的回声被 AGC 拉高了，这个除了改喇叭改结构没有其他的办法。

8. 常见案例

(1) 功放电路失真

功放电路失真的案例，主要在于如何来确定问题点在功放电路上，以及确认问题后相应的解决方法。当我们拿到客户的机器，来进行初步的扫频分析时（将 mic 和 speaker 的增益设定为 0dB，自播自录扫频音档），若此时看到的频响特性非常差，几乎所有频段都有谐振，且能量也比较高。这时就需要怀疑功放电路在当前的状态下已经出现了非常严重的失真，在确定 mic 频响特性没有太大问题的前提下，可通过更换喇叭来进行测试。若更换了几个喇叭后，对比音档频响特性均没有较大的改善。这时则需要请硬件同事测量功放电路的输入和输出信号，来进行确定。若 speaker 增益已经衰减得比较大，功放的输入信号正常幅值较小，但功放的输出信号仍然出现失真，就要请硬件同事确认是否为功放放大倍数太大，或者是功放的供电电压太小没法满足放大的需要，出现削顶。这时若能修改功放电路解决最好。但在不可抗力因素下，比如客户已量产、硬件修改对客户影响太大或修改难度大等原因，我们只能在不失真的前提下，尽量提高对讲时的信号幅值来弥补声音小的缺陷。首先请硬件同事帮忙测量在播放扫频音档时，speaker 增益最大下到多少可以保证功放电路的输出信号不失真，以此增益作为软件可调整的最大增益值，同时为了使 speaker 输出的声音足够大，使用 AO 端的 AGC 尽量对输出数据拉大，并限制在 -3~-1dB 以下（留有余量，避免功放失真）。

(2) Speaker 端源数据爆音处理

在对讲时发现即将播放的数据（即 far end 传过来的原始数据）会经常爆掉的情况下，可以对传过来的数据进行相应的处理来减缓这种情况，可以对数据进行多点平均的做法来将一些爆掉的点拉下来，多点平均（将当前采样点的数值与前面多个点进行平均，至于多少个点做一次平均请按具体情况来调整，爆得越严重要求的点数越多，计算公式： $y = x(n)/n + x(n-1)/n + x(n-2)/n + \dots + x(0)/n$ ）的方法相当于做了一次低通滤波，可以有效地消除锯齿，带来的副作用就是周期短的高频信号会被滤掉。多点平均的做法会把音质变糊，其他情况不建议使用。

(3) 对方 speaker 播出来的语音间，有噪声被拉起后，又被压掉

当遇到对方 speaker 播出来的语音之间底噪忽大忽小的情况时，首先需要先确认是哪一方的原因。我们可以通过 dump 出 AI 的 VqeIn 和 VqeOut 音档来确定，用 adobe audition 听音档 VqeOut 是否有底噪忽大忽小的现象。

若 VqeOut 音档听起来正常，则是对方的原因，需要对方来解。

若 VqeOut 音档听起来就已经存在忽大忽小的噪声，而 VqeIn 音档中没有，那基本可以肯定是 Vqe 的处理过程导致的。该现象可能有以下几个原因：

1、语音间隔短，导致 NR 收敛慢。

解决办法：

加快 NR 的收敛速度，eNrSpeed = E_MI_AUDIO_NR_SPEED_HIGH。

使能 AEC 的舒适噪声，bComfortNoiseEnable = TRUE。

2、EQ 调整不当。

解决方法：

若从 VqeOut 音档中看出噪声是在固定频段，可通过调整相应频段的 EQ，来压掉噪声。

若 VqeIn 音档听起来已经有忽大忽小的噪声，那可以 dump 出 Aec 的音档，对比 AecAiIn 和 AecOut 音档，确认是否是 Aec 过程产生的还是环境中的声音本来就是这样。若只有 AecOut 音档有，则基本可以确认是 AEC

消不干净导致的，查看“噪声”（回声）处在哪个频段，将该频段的 AEC 消除强度加大到可以消掉此回声。
解决方法：调整 AEC 的 u32AecSupfreq 和 u32AecSupIntensity 来处理。

(4) 我方 speaker 播放有“尾音”。

这种情况主观表现为我方 speaker 播放出来有尾音。首先需要先 dump 出 AO 的 Vqeln 音档来分析，若 Vqeln 的数据就已经有“尾音”，则是对方 AEC 消不干净导致的。我们可以通过以下几个方法来处理：

- 1、调整对方的 AEC 参数。
- 2、若“尾音”的频段比较固定，我们可以通过调整 EQ 来砍“尾音”所处的频段。
- 3、若“尾音”频段不固定，则只能分析尾音的能量大小，在“尾音”和正常声音能量相差较大时，可通过 AGC 来处理（仅适用于对方是固定的机子，不存在我方需要对很多不同机子的情况，且很可能会误伤到正常的语音信号）。我们可以分析尾音的信号能量大小（[查看增益](#)），然后调整 ACG 的曲线（s16Compression_ratio_input 和 s16Compression_ratio_output）或噪声门限值（s32NoiseGateDb 和 u32NoiseGateAttenuationDb）将“尾音”压掉。

(5) 单向对讲对方消不干净

我方能听到回声，原因是对方消不干净。这种情况下，我们要先定位是什么原因导致对方 AEC 消不干净。一般主要是以下几个因素导致对方消不干净：

- 1、对方的 speaker 或 mic 增益太大
解决方法：降低对方的 speaker 或 mic 的增益，降低我方的 mic 增益
- 2、对方结构较差，内部串音严重。可以通过以下两种方式来确认。一、用黏土堵住对方的收音孔，在对方处说话，我方是否能听到对方的声音。若能听到，则说明对方的结构气密性较差。二、用黏土将对方 mic 的背面完全堵住，在我方说话，看回音是否有非常大的改善，若有非常大的改善，则说明是对方结构的问题。
解决方法：对方调整结构
- 3、对方 AEC 强度调整不恰当
解决方法：对方调整 AEC 强度

注：若对方传回来的回音，频段固定，且能量也不是特别大时，可以通过 AO 端的 EQ 来砍对方消不干净的回音。或频段不固定，但回声和正常语音能量相差较大，可通过 AO 端的 AGC 曲线（s16Compression_ratio_input 和 s16Compression_ratio_output）或噪声门限值（s32NoiseGateDb 和 u32NoiseGateAttenuationDb）将回声压掉。（同样仅适用于只配对一台设备的情况，且可能误伤正常语音信号）

(6) 单向对讲我方消不干净

我方消不干净原因与对方消不干净的原因大同小异，完全可以同样的思路来处理。

- 1、mic/speaker 的增益太大导致声音爆掉。
解决方法：降低我方的 mic/speaker 的增益，降低对方 mic 的增益
- 2、我方结构较差，内部串音严重。可以通过以下两种方式来确认。一、用黏土堵住我方的收音孔，在我方处说话，对方是否能听到我方的声音。若能听到，则说明我方的结构气密性较差。二、用黏土将我方 mic 的背面完全堵住，在对方说话，看回音是否有非常大的改善，若有非常大的改善，则说明是我方结构的问题。
解决方法：我方调整结构
- 3、我方 AEC 参数没调好
分析 Aec 的相关音档 AecAiln、AecAoln、AecOut，看消不干净的回声主要在哪些频段，调整频段参数和消除强度（u32AecSupfreq 和 u32AecSupIntensity）

注：若我方 AEC 处理完仍有回声，频段固定，且能量也不是特别大时，可以通过 AI 端的 EQ 来砍消不干净的回音。或频段不固定，但回声和正常语音能量相差较大，可通过 AI 端的 AGC 曲线

(s16Compression_ratio_input 和 s16Compression_ratio_output) 或噪声门限值 (s32NoiseGateDb 和 u32NoiseGateAttenuationDb) 将回声压掉。(同样仅适用于只配对一台设备的情况, 且可能误伤正常语音信号)

(7) 双向对讲, 对方听到的声音忽大忽小

双向对讲时, 对方听到的声音忽大忽小。首先要先确定是哪一方的问题, 可以通过 dump 我方 AI 的 AEC、VQE 音档, 通过分析 VqeOut 来确认是否是我方引起的问题, 若 VqeOut 音档没有问题, 则是对方的原因, 需要对方解。若是我方导致的问题则可以通过以下几个因素来进行排查:

1、对方回过来的回声太大, 导致我方 AEC 把 mic 的信号消得比较多。

解决方法:

调整我方的 AEC 参数, 通过分析 AEC 音档 AecAIn、AecAoIn、AecOut, 判断是哪些频段 AEC 开得太强了, 重新调整 AEC 的频段和消除强度 (u32AecSupfreq 和 u32AecSupIntensity), 或降低 speaker 的增益。

2、我方 mic 端使能了 AGC, 但 AGC 参数没调好。若关掉 AGC, 则不会忽大忽小, 则说明是 AGC 参数没调好。

解决方法:

结合音档调整 AGC 参数, 主要需要关注曲线 (s16Compression_ratio_input 和 s16Compression_ratio_output)、噪声门限值 (s32NoiseGateDb 和 u32NoiseGateAttenuationDb) 以及 u32DropGainMax 等设定。

(8) 双向对讲, 我方听到的声音忽大忽小

双向对讲, 我方听到的声音忽大忽小, 首先要先确定是哪一方的问题, 可以通过 dump 我方 AO 的 VqeIn 和音档 VqeOut 来确认是否是我方引起的问题, 若 VqeIn 音档就已经忽大忽小, 则是对方处理的问题。若 VqeIn 音档没有忽大忽小, 而 VqeOut 有忽大忽小的现象, 则可以通过以下方向来排查:

1、我方 speaker 端使能了 AGC, 但 AGC 参数没调好。若关掉 AGC, 则不会忽大忽小, 则说明是 AGC 参数没调好。

解决方法:

结合音档调整 AGC 参数, 主要需要关注曲线 (s16Compression_ratio_input 和 s16Compression_ratio_output)、噪声门限值 (s32NoiseGateDb 和 u32NoiseGateAttenuationDb) 以及 u32DropGainMax 等设定。

2、结构、喇叭等原因。比如喇叭没装稳, 喇叭某些频段打不出来等。

(9) 双方同时说话漏字

双方同时说话漏字, 主要由以下原因造成 (双方都有可能):

- 1、漏字方 AEC 强度开得太强
- 2、其中一方回声消不干净, 导致另一方 AEC 误砍语音信号
- 3、mic/speaker 增益开得太大, 影响 AEC 处理结果
- 4、EQ/AGC 等算法参数设置不合理

(10) 喇叭声音破音、颤音

- 1、结构或喇叭原因
- 2、降低我方 speaker 的增益、降低对方 mic 的增益

9. 注意事项

- (1) 在 speaker 增益固定后，按当前的增益播放 1KHz0dB 的正弦波音档或扫频音档 TestFile3，请硬件同事帮忙测量送到 speaker 端信号的峰峰值，根据 speaker 的阻抗，请硬件同事计算此增益下，功率是否超过了 speaker 本身的额定功率，以防烧坏喇叭。