

消人声之术

张 翔

(版权所有，欢迎交流传播)

写在前面的：

这是一篇关于消除歌曲中主唱部分声音信号方法的简单讨论。两个星期前刚刚考完 DSP 考试，百无聊赖之中打算做几首伴奏唱会儿卡拉 OK 发泄一下，结果发现网络上的伴奏文件质量一如既往的不尽如人意，于是打开 Cubase 试着用 DSP 的思路解决一下做伴奏的问题。而后经历了试验、请教教授、阅读文献，希望写下一些东西为朋友们无论在方法上还是思路上提供一点参考。在中国海洋大学的时候非常理解学生演出中“一伴奏难求”的痛苦，正如某位挚友所说：“这可是大学阶段演出最尖锐的难题啊！”笑谈归笑谈，如果有一个比较简单可行的方法去除一般歌曲中的人声只保留伴奏部分，对于不论是学生还是一般家庭娱乐都肯定都是非常好的消息。然而，科技的发展并没有那么快，虽然我相信这个课题会在不久的将来得到完美解决，但到现在为止，完美的解决方案依然在各种实验室处于研究阶段。

本文将在第一部分给出一个比较可行的去人声保留伴奏的方法，这一方法其实已经存在多年，本文只是对其进行一个简单的介绍。这一方法的确适合相当一部分歌曲的伴奏制作，但并不一定适合所有歌曲。而在第二第三部分将给出这种方法的原理解释和讨论，并且对世界上这一课题的研究状态给出简单介绍，希望对有兴趣从事此课题研究的朋友提供一个简单的概观。

1. 一种去除人声保留伴奏的方法：反相相加+修正

1.1 方法一

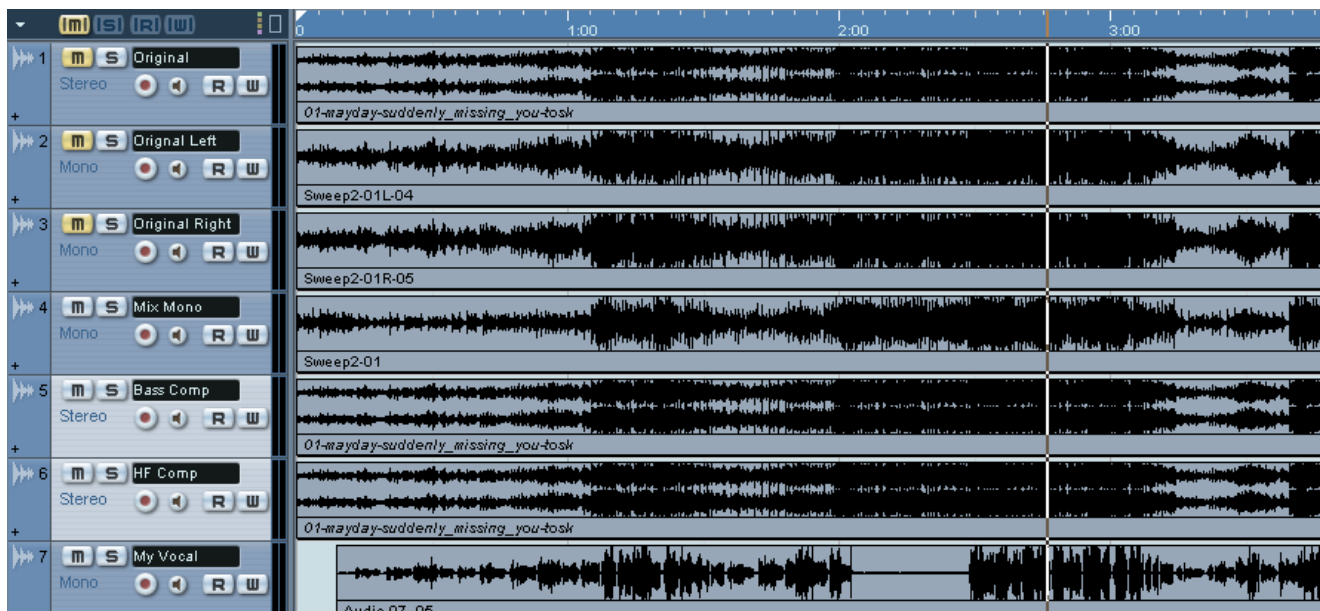


图 1.1

图 1.1 中显示的是这种方法所使用的几个音轨。这里使用的平台是 Cubase。Track1 为原始 Stereo 音乐信号（双声道），Track2 和 Track3 为分离出来的原始立体声信号的左右声道(都为 Mono)，其中对右声道进行了反相（Phase Reverse）处理。而 Track4 为 Track2 和 Track3 的缩混结果（由于 Track2 和 Track3 都为 Mono,因此缩混出来的 Track4 也为 Mono）。Track5 和 Track6 都为原始 Stereo 信号的拷贝留作伴奏修正时使用。Track7 为本人狼嚎的一段人声，作为验证伴奏质量和微调时使用的素材。

其实这种方法相当简单，相信已经有无数的朋友正在使用。下边是这种做法的流程，原理将在下一节进行讨论：

第一步：把原始音频 Stereo 信号的左右声道剥离开来成为两路独立的 Mono 信号(Track2 和 Track3)；

第二步：对分开的两路 Mono 信号中的一路进行反相(Phase Reverse)处理（这里的“相”指的是相位，有兴趣的朋友可以参考以下连接：<http://baike.baidu.com/view/132570.html?wtp=tt> 以及 <http://en.wikipedia.org/wiki/Phase>）。在 Cubase 中，Phase Reverse 功能可以在 Audio-Process-Phase Reverse 中找到，在 Cool Edit, Audition, Sonar, Logic, Samplitude 等软件中也有相应功能，只需要在 Google 中键入相应混音软件名称和关键词“Phase Reverse”就可以找到反相功能所在的目录。

第三步：将以上两路 Mono 信号缩混成为一路 Mono 信号（Track4）。

第四步：这一步是此方法中最关键的一步。把原始信号复制两份到两个新的音轨中（上边的 Track5 和 Track6 且全部为 Stereo）。对其中一轨应用高通滤波器（截止频率 $\geq 1500\text{Hz}$ ），另外一轨应用低通滤波器（截止频率 $\leq 150\text{Hz}$ ）。注意：这两个截止频率并不是一定的，而是需要根据不同歌曲仔细调节。上边的例子以五月天的《突然好想你》为例，说明一下如何选择这两个截止频率和相应的 Q 值。

低频补偿轨(Track5)：

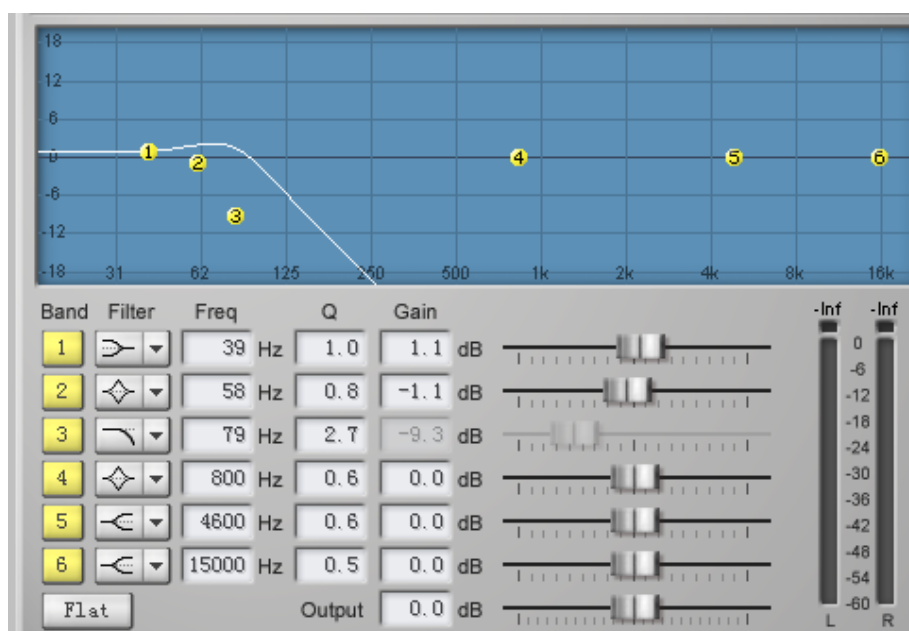


图 1.2

我们对 Track5 应用了一个低通滤波器（Band3），将期 Q 值设定为 2.7，而后将频率点设定为 79Hz。这时其实 Band1 和 Band2 的设定对 EQ 的频响曲线也是有影响的。值得注意的是，当 Q 值越大，在衰减曲线变陡的同时，在截止频率点上会出现一个增益的奇点（图 1.2 中 Band2 点处）。这是由于 Two-Pole 滤波器的频响特性决定的，而一般软件 EQ 中的低通滤波器都是采用的这种算法（参考 https://ccrma.stanford.edu/~jos/fp/Resonator_Bandwidth_Terms_Pole.html）。为了减小这个奇点过度增益，我们需要在保证低通滤波器的 Q 值不过大的同时，适当调整 Band1 和 Band2 的 Q 值和频点，来在一定程度上对 Band3 这个低通滤波器进行一定修正。

相应的，我们对 Track5 应用一个高通滤波器，些时同样会出现相关的奇点问题（参考：https://ccrma.stanford.edu/~jos/fp/Two_Zero.html），依照上面的方法调整好此高通滤波器。

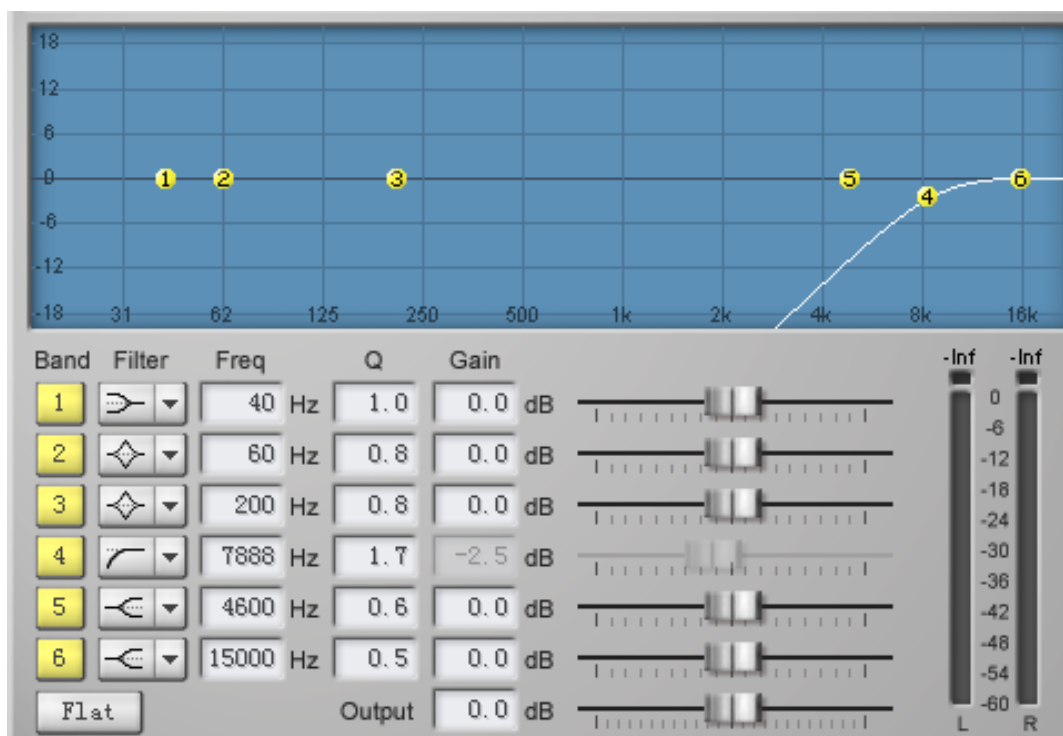


图 1.3

调节这两个高、低频补偿轨的思路：

在调节高、低频补偿轨的时候，首先需要把相应轨道中的人声部分的频率尽量排除掉，而只保留比人声频率高和低的两部分频率来对上边的 Track4 进行一定补偿。截止频率的选择最好是刚刚听不到原唱的声音。下一节将解释这么做的原因。

简单说明：我们都知道人声的主频范围大概在 100-1kHz 左右，而每个人发声时产生的泛音频率的频率及其能量分量又各不相同，根据不同的人的特点，这一频率范围可能会变窄也可能会变宽。换句话说：一首歌曲中主唱歌手声音所在的频率范围是因人而异的，甚至是因曲子 and 混音师而异（唱片的混音师的听音习惯不同，在混音时对人声采用的激励量也不同，造成泛音的分布情况会各不相同）。此外，还有一个相当重要的因素需要注意，一般软件 EQ 的增益衰减值都可以做到大于上下 18dB，如果所使用的 EQ 的增益衰减值小于这个值，建议使用处理动态范围更大的 EQ 或使用此 EQ 的大动态档

进行处理。之所以使用两个单独的音轨分别使用两个 EQ 是因为, EQ 是一串 Biquad 滤波器的组合, 这其中既有上边提到过的 two-pole 也有好多例如 one-pole, one-zeros 等等的滤波器。使用越少的滤波器越不容易造成对周围滤波器的影响。如果在一个 EQ 中使用所谓的“带切” (Band Block) 滤波器, 理论上讲可行, 但操作起来, 人声部分频率的衰减效果和人声外频率的保留效果都不尽如人意, 而且很有可能造成一些令人讨厌的“奇点”, 增加调节难度。

第五步: 录制一段人声进去跟着伴奏进行演唱 (Track7)。此时再次调节 Track5 和 Track6 的 EQ 设定, 即使单独听伴奏时会有一定原唱声音若隐若现, 如果加上伴奏后这些声音基本听不到了, 那么结果也是可以接受的。

1.2 方法二:

采用上边的思路, 为了节省系统资源, 我们减少使用的音轨数目, 而使用 Routing (路由) 的方法 (估计在读到刚才的方法时已经有朋友大骂道这种方式太浪费系统资源了, 呵呵)。

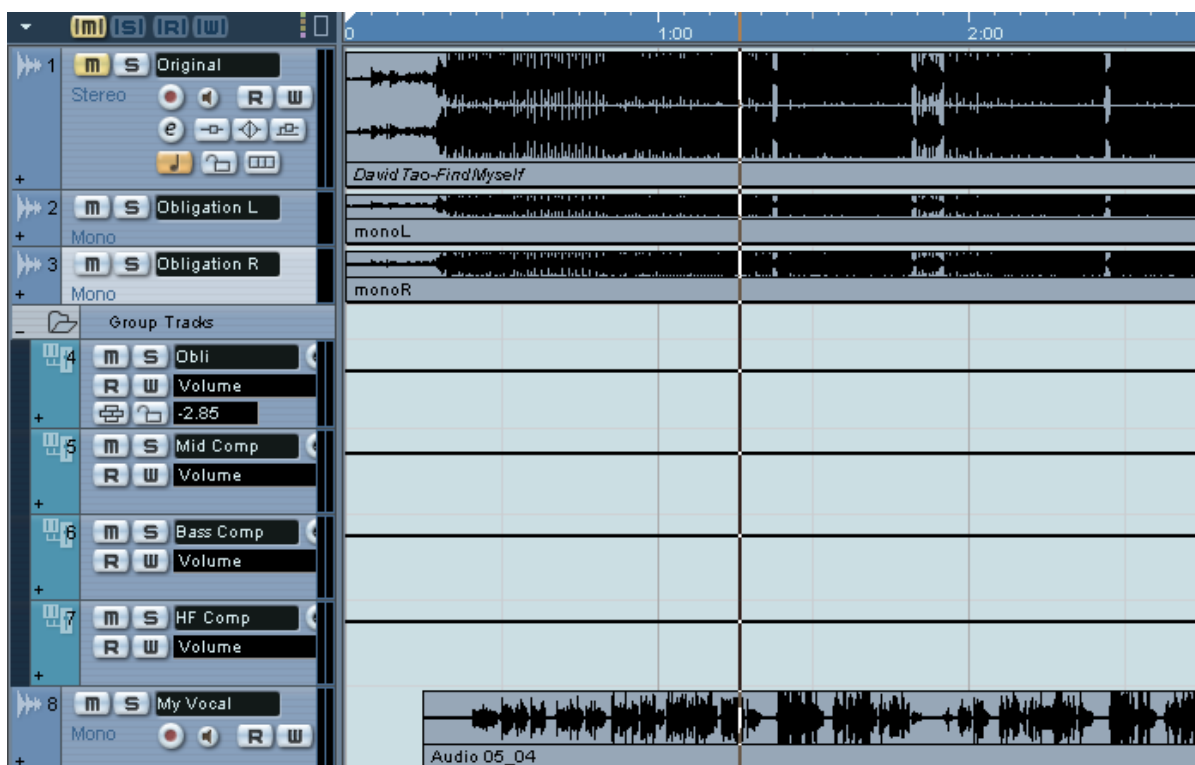


图 1.4

图 1.4 的思路和方法一中完全一样, 只不过不需要大量的音轨拷贝。我们把原始信号 (Track1) 分开为两路 Mono 信号 (Track2 和 Track3, 其中 Track3 已经进行反相处理), 然后将 Track1 发送到两路 Aux 中 (Track5 和 Track6) 然后对这两路 Aux 加载 EQ 进行高低频的补偿, 而后将 Track2 和 Track3 发送到另外一路 Aux (Track4) 作为主伴奏 Mono 信号, 再将 Track4 这一路 Aux 发送到 Track5 这一路新的 Aux 里, 并对 Track5 进行带通滤波, 从而对伴奏的中频部分进行一定补偿 (方法一中没有对伴奏的

中频进行补偿)。得到的结果和方法一是相同的,但这种方法会大大减少系统资源的使用,降低运算时间。

1.3 小结

对低频的补偿几乎在每首歌曲中都要使用到,下边会对这一点的原因进行分析。而高频和中频的补偿并不一定每首歌曲都会用到,可以根据自己的耳朵来决定对于特定的歌曲是否需要使用中频和高频的补偿。

建议不论使用什么音频处理平台,可以建立一个工程模板,在制作新伴奏的时候只需要把四轨信号贴到工程模板中就可以进行处理,而不需要每一次都进行路由的设置和EQ等效果器的调节,减少批量处理时的工作量。

图 1.5 是这一思路的信号框图。

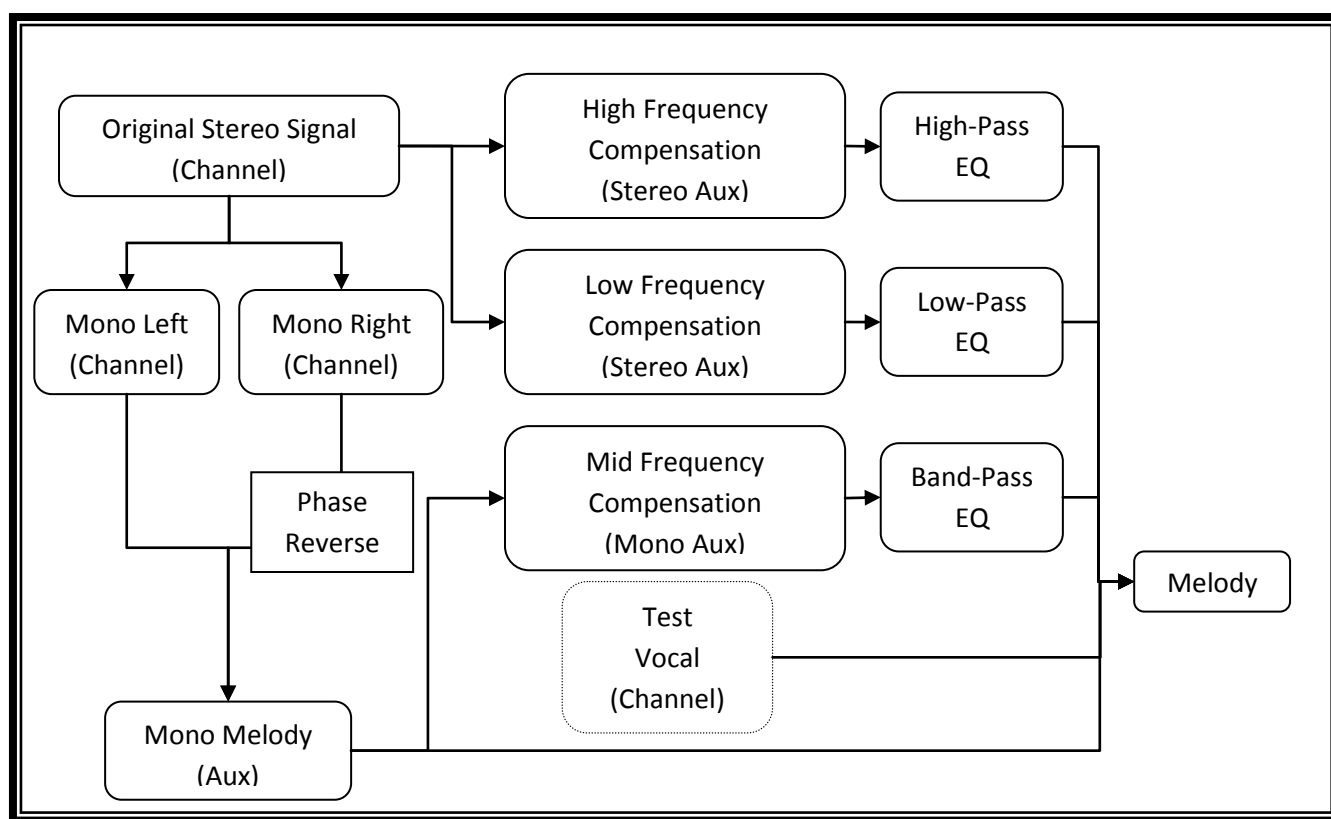


图 1.5

2. 对于“反相相加+修正”方法的原理及问题的讨论

2.1 反相相加的原理

其实太多人早就明白这么做可以去除人声。而且网上已经存在好多关于这个方法的讨论，包括在 Audition 平台上的效果器插件可以直接实现左右声道的相减去除人声。在我们把耳机使用的 1/8 寸双声道接口虚接时产生的人声消失现象也是因为电路中产生了左右声道信号的相减运算。

有过混音经验的朋友一般都知道，主唱人声的 Pan 在绝大多数时间里都位于 0 度，也就是正中间。这说明在原始 Stereo 信号中，左声道和右声道里人声部分的信号的频率、增益值和延迟都是完全一样的。正如我们把两个正弦波 $\text{Signal1} = \cos(\theta)$ 和 $\text{Signal2} = \cos(\theta + \pi)$ 相加，其中 Signal2 是 Signal1 相位移动 180 度得到的结果，如图 2.1，得到的结果就是它们相互抵消。而频域中，每个频点（Bin）是相互独立的（参考离散傅利叶变换：

http://en.wikipedia.org/wiki/Discrete_Fourier_transform），因此我们在把原始信号的左右声道相减之后（把一个声道反相与另外一个声道相加，相当于把原始信号相减），会把左右声道中频率、相位、增益值完全相同的信号相互抵消掉。但同时，频率相同但相位、增益值不同的信号会受到部分抵消或部分增益（波的干涉参考 http://en.wikipedia.org/wiki/Interference_%28wave_propagation%29），没有办法完全抵消，因此在直接把左右声道相减以后会出现一些奇怪的声音，这些声音就是未完全抵消的结果。（这一思路同样被应用在音频连线的平衡接法中：把 A 信号进行反相得到信号 B，二者共用一条地线作为电压参考，在接收端对 A、B 进行相减运算，这样不仅得到了二倍的动态范围，也使得 AB 信号中相同的信号成分相互抵消，从而达到屏蔽线路噪声的效果）

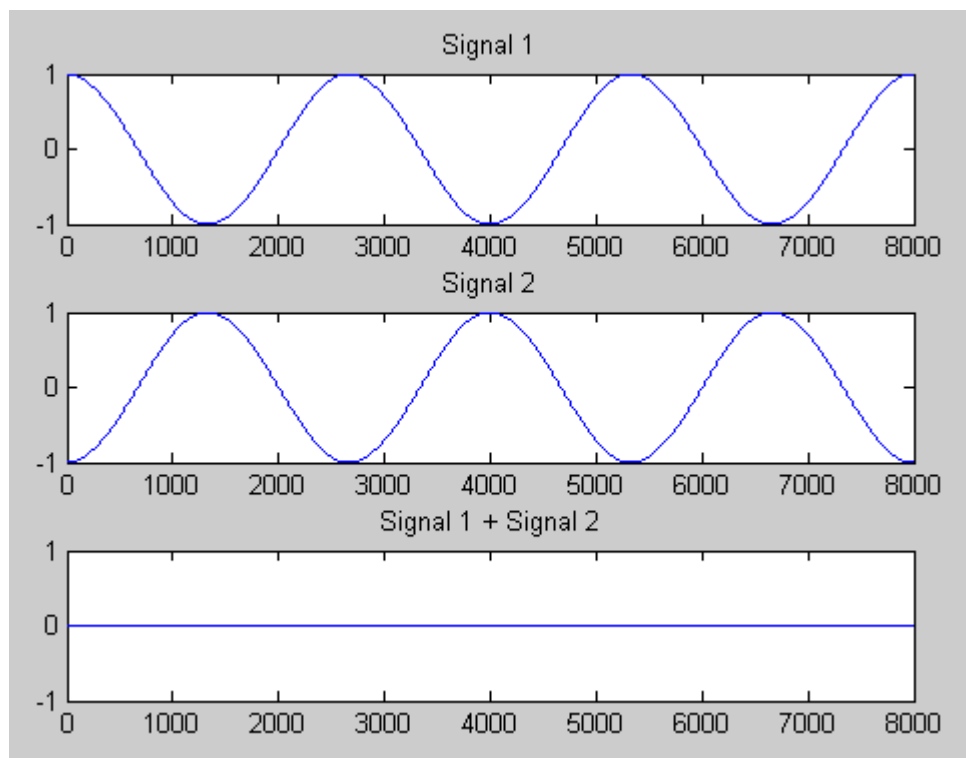


图 2.1

2.2 对一些频率进行修正的原因

有过混音经历的朋友都知道，对于某一些乐器和声音我们会对其进行一些相应的处理。比如合声会进行反相然后分配到左右声道，比如贝斯和底鼓的 **Pan** 几乎一定是 0 度（由于人对次低频的方位感受力不强，即使给它一个非 0 的 **Pan** 也不容易感觉到具体位置），再比如伴奏乐器的 **Pan** 我们会根据需要分配到左或右一定度数来保证声场的宽度等等。因此我们在用上边方法制作伴奏的时候：“合声反相分配到左右声道”就造成了我们得到的 **Mono** 伴奏信号中合声部分往往被加强了；而同时，贝斯和底鼓的声音被相应减弱了（好在贝斯和底鼓一般会多少加一点 **Stereo** 混响或延迟，因此在左右相减以后不至于完全消除掉，关于混响和延迟的影响在下边会具体讨论到。正是 **Stereo** 混响和延迟造成了这种做伴奏的方法对于相当一部分音乐不能适用）；还是同时，一些 **Pan** 被设定为 0 度的乐器的高频部分（直接影响音乐的空间深度和音色）就会被抵消掉或产生 2.1 中最后提到的噪声。另外，我们左右相减得到的伴奏是 **Mono** 信号，声场太窄，因此需要加入一些原始信号对它进行一定声场的拓宽，而使用原始信号的同时，我们又不希望原始信号中原唱部分频率的信号加入到我们的伴奏中。

鉴于这些原因，我们使用了一些低频和高频的补偿。而补偿中，最关键的问题就是调节 **EQ** 时频点和 **Q** 值的选择。由于歌曲与歌曲的特性差别很大，因此这一步最好手动加耳朵听来完成。

而中频部分的补偿是由于有些混音者在混音时把中频部分的声音大部分让给了人声，在原曲配上原唱时也许很好听，但做成伴奏以后会明显感觉伴奏声音很干。因此，我们把做好的 **Mono** 伴奏的中频部分根据需要进行很小的一点增益或衰减往往可以使伴奏更加好听。中频的补偿或衰减其实更大程度上取决于左右声道相加后得到的 **Mono** 伴奏音乐的干净程度。如果我们得到的这个 **Mono** 信号中人声已经被绝大部分消除掉，对其中频进行增益是肯定没有问题的，但如果这个信号中人声部分还依然存在不少，那么只能对其进行相应的衰减。对于这个问题，下边还会进行更进一步的讨论。这也是此种方法不适合于所有歌曲的一个原因。

2.3 加入人声后调整的原因

有过混音经验的朋友都知道，混音时只混伴奏不加入声的话，混音工作其实是没有意义的。一首歌曲只有在每个部分都完整的时候才可以把大家混在一起，相互在各个参数上齿合协作才能够达到你中有我我中有你的美感。那么为什么在做伴奏的时候也需要加入人声进行最后一步调整呢？主要是由于我们的耳朵有一些比较奇怪的生理特点，往往我们可以对其加以利用。这里只讨论几个我们在用这种方法制作伴奏时会用到的。

首先，在声功率相同时，人耳对不同频率的声音，感受的灵敏度不同（相同声功率下人耳感觉各个频率的响度不同，人耳的等响曲线：http://me.seu.edu.cn/jmp/jpkc2006/kczd/kc/chap_14/14-2-2-1.htm），刚好不巧的是，歌曲中人声在的频率范围正好是我们人耳比较灵敏的频段，换句话说，左右声道相减以后，只要人声消除得不是特别干净，那么听起来肯定会相当刺耳。

第二，人耳对声音有掩蔽效应(参考：《声音听觉论》<http://www.yoeki.cn/text/js001.htm>)，掩蔽其实涉及数个变量的综合结果，其掩蔽包含了时域、频域和强弱三种。时域的掩蔽更多应用在 **PA** 系统设计中（这里不作讨论），而频域的掩蔽以及强弱上的掩蔽对我们制作伴奏就很有用。频域和不同声压级的掩蔽效应简单总结为：功率相近时低频纯音可以掩蔽高频纯音，处于中等强度时的纯音的最

有效的掩蔽出现在它的频率附近；低频的纯音可以有效地掩蔽高频的纯音，而反过来则作用很小；较弱的声音(被掩蔽音)的听觉感受会被较强的声音(掩蔽音)影响。

因此，在加入人声演唱以后，往往很多去得不干净的原唱声或噪声其实可以被人声有效遮蔽掉，虽然单听伴奏时人声去除得并不太干净甚至有些不希望有的杂音，但加上人唱时已经无法明显感觉到它们了。

2.4 “反相相加+修正”方法存在的问题

其实相信很多朋友都有过使用这种方法没办法消去人声或结果不理想的经验。必需承认，这种方法其实只是种相当简单的应急之策，很难做出非常完美的伴奏。但对于很多对音质要求不是太高的场合，比如小形广场演出，校园演出等等，如果制作者选到了可以应用此方法的歌曲，而且认真做过上边几步，应该可以得到基本满足需要的伴奏。

无法应用这种方法的几种情况及原因。

1) 其实反相相加+修正”的方法有一个应用前提，我们处理的对象必需在混音时人声信号的 Pan 处于正中 0 度位置，而且其它部分的声音信号是立体声分部而非左右声道完全相同的单声道或等同单声道分布。因为此时我们才可以把左右声道中相同的人声部分相互减掉而保留不同的其它部分的信号。因此，如果我们所拿到的原始音乐不满足上边的前提，那么用这种方法去人声是肯定不可行的。（结果往往是要么人声根本去不掉，要么一下把所有声音全部消掉）

2) 这一条所述的，是无法使用“反相相加+修正”方法的最主要的原因。现代混音的过程种很少会不对人声进行任何处理。一般情况下，人声这轨信号上很有可能使用了混响、延迟和激励这三种效果。而现在使用混响和延迟效果时很少人有会使用左右声道完全相同的参数设置，因为如果使用这样的设置会使得得到的结果单调或听起来不真实。因此，比如我们在使用延迟时左声道在 1s, 2s, 4s 处有延迟声，而右声道在 0.5s, 1.5s, 3.5s 处有延迟，而即使延迟信号发生的点是完全一样的，其左右声道的延迟效果湿声的比例也会不一样。那么得到的声道相减的结果肯定也不会完全相互抵消。混响其实就是时间非常非常短的延迟（混响效果器的基本原理请参考胡戈《混响大法》），而软件混响效果器中类似于 diffusion 和 stereo width 之类的参数，本身就会影响到左右延迟点位置和混入量的相似程度。混响效果器中，早期反射声的设定延迟时间一般在 50 个毫秒以内，这种延迟凭听觉是感觉不到的，因此得到的结果是：你感觉主唱的人声并没有给去掉，只是低频部分被衰减了。事实其实是这样：主唱的干声的确被去掉了，而你听到的没去掉那些人声只是通过混响效果器以后的湿声，由于左右声道的早期反射声设定值不一样，但都小于 20 个毫秒，因此这些湿声无法相互抵消。（听起来低频被衰减，是因为混响的湿声一般会经过一个高通滤波器，被去除一定低频成分。其实混响效果器同时也会去除不少高频成分，但人耳朵对 8000Hz 以上高频实在不太敏感，加上低频对高频本身就有的掩蔽效应，因此可能不太会注意到高频其实也一定程序上被衰减了）另外，激励效果器的使用会使很多本来不在人声主频范围内的泛音频率的增益值变得非常大，在调节 EQ 时可能需要把非常宽的一个频带衰减下去。

这些都成为无法在一些歌曲中使用本文所述方法制作伴奏的客观原因。但是好消息是，现在的混音工作者的技术水平越来越高，乱用混响乱用延迟的人越来越少。因此，比起老歌来讲，很多新流行音乐中的人声反而更容易用“反相相加+修正”的方法去除掉。

3) 在混音的过程中，混音师会使用大量的滤波器（一切对原始信号进行的非线性改变的过程都可以看作一个滤波的过程）。而对于类似于 EQ 这样的多个带通、低通、高通小滤波器组成的滤波器组，在信号被处理的过程中会引起相应的相位的变化。这里的相位变化并不是前边我们提到的反相 180 度这么极端，而是指不同的频率的相位变化是不同的。而具体如何不同则取决于信号所通过滤波器的相位响应，平时我们更多关心的是频率响应中的增益响应这一部分，而相位响应是频率响应的另一个组成部分。（滤波器的频率响应基本知识参考：Julius O Smith:

https://ccrma.stanford.edu/~jos/filters/Frequency_Response_Analysis.html）这些相位的变化的具体数值，估计没有哪个混音师会记得住、算得出，因为他们所使用的效果器的厂商也不可能在产品说明书上标出此产品中每个滤波器的相位响应。（毕竟人对相位的轻微不同不是很敏感，而针对人能不能感受到轻微的相位不同这个问题，学术界依然在争论不休）然而，这些轻微的相位变化虽然对我们的听音影响不大，但对制作伴奏的影响却非常大。因为如果左右声道主唱人声的相位一致程度不够，得到的左右相减的结果就没办法把人声干净的去掉。

解决这个问题有一个比较土的办法，我们可以使用一个可以调节 Phase 的插件，在混成伴奏单声道前，加入到被分开并一路反相过的 L 或 R 单声道上，轻微调整这一路混入前的 Phase，看是否会在某一点上（非 180 度和 0 度）可以消去更多的人声。注意，不要用“Phaser”一类的动态相位调整器。大家都知道，类似哇音之类的效果是通过 Phaser 制作出来的，在哇音效果中我们需要被处理轨的信号相位一直变化，而我们这里需要的是对一轨声道的 Phase 进行恒定的改变。这一方法其实也是缓兵之计，因为当左右声道的相位变化因频率而不同时，给出单一的相位变化量其实并不是一个能够解决整个人声频带的办法。只能对最刺耳的那个频带上消得不干净的人声给与一定的处理。

4) 随着国内外录音棚硬件水平的突飞猛进，录音师们有时会用 Pair 或者 Pair + Mono 的方法录制人声，简言之就是说得到的主唱的人声信号是一路立体声或者一路立体声加一路单声道的信号，而有可能不是传统的只一路单声道。（参考：“Professional Microphone Techniques” by David Miles Huber and Philip Williams, Mix Books, 1998）此时，在声压级和延迟两个参数之间可能立体声的两路信号达不到完全的参数平衡，因此反相相加得到的结果也会不能完全抵消一部分主唱的声信号。但对于无法使用这种方法制作伴奏的原因而言，这种情况比起以上几种情况所占的比重要小得多。

3. 对于去除人声保留伴奏技术的概况介绍

提到去除人声的技术就不得不提到 2004 年的一场比赛：ISMIR 2004 Melody Extraction Contest (2004), organized by the Music Technology group of the Universitat Pompeu Fabra。本文提到的只是一种比较通用的小把戏，避开了这一过程中最难的问题，而在这场比赛中，各国的音频信号处理界的精英从理论和实验两个方面实打实地从一段音频信号中抽出相应的主旋律成分，把主旋律和伴奏分成两个独立的信号。

其实想到去除人声，最容易想到的就是根据人声主频及其泛音成分的变化找到相应的信号分量，将其提取出来。而这场比赛中的选手们也是根据这一思路展开自己的项目。有兴趣的朋友可以在网上搜索相应资料进行研究学习。

选手们使用的一个很关键的算法叫做 **Correlation**（参考 http://en.wikipedia.org/wiki/Correlation_and_dependence），中文翻译为“相关”。**Correlation** 这种运算是用来判断两个信号的相似程度，它广泛应用于 GPS，红外测距、统计、信息处理、图像处理等技术中。**Correlation** 和 **Convolution**（卷积）在数学上很相似，它把一个信号和另外一个信号进行相应运算以后得到一个曲线，这个曲线会在某几个点上有突然的峰值，而处于峰值处时，两个信号会有较大的相似性。采用这一方法，我们就可以利用人对音乐旋律的判断来找到音乐中主旋律所在的变化频率的信息。因此就会看到有某些提取人声的方法中，需要人跟随音乐旋律唱出一段，而后经过运算，计算机比较精确地把和人所唱的内容相似的频率部分剥离开来。

2.4 中的第 3 点其实会在很大程度上影响到我们对音频信号的处理。这也就是为什么如果有可能，尽量使用 **Matlab** 或者 **C++** 之类的工具处理音频而不是使用 **Protools** 之类的音频处理平台。相位的变化在这些平台里边是一个隐藏的数据，这个数据对于人听音并不构成太大影响，但对于信号处理却是不能忽略的。有能力的朋友如果兴趣可以试验一下 **Matlab** 里 **filtfilt** 这个函数，因为它可以设计一些相位响应为零（**Constant**）的滤波器，从而避免了相位变化对我们信号处理过程中造成的影响。

做过此类研究的教授还有 **Jean Marc Jot**，**Jean Laroche**，以及 **Mike Goodwin**，有兴趣的朋友可以在 **Google Scholar** 里搜索到他们写过的相关文章和做过的研究。

写在后面的:

我实验了三首歌曲，第一首是五月天的《突然好想你》，第二首是庾澄庆的《春泥》，第三首是陶吉吉的《找自己》，第一首歌的人声去除程度非常好，第二首非常不好，第三首正好处于二者之间，但经过调整以后基本可以达到使用需要。当然在伴奏乐器比较少的安静一些的音乐中，如果人声去得不是很干净，听者会很容易听出来那些残留的声音。相对于此，如果伴奏音乐比较复杂比较吵，人声即使去得不是很干净也可以在加入自己演唱以后基本听不出残留的原唱的声音。其实当伴奏乐器比较少时，我们完全可以自己用 **MIDI** 软件根据原谱来制作一些简单的伴奏（现在网络上更不乏各路高人自己制作的简单翻版伴奏），当伴奏音量起来以后再使用本文提到的方法处理后面的部分。

写到这里，希望对此感兴趣的朋友们或多或少能够得到一些有用的信息。毕竟并非人人都是从事音频信号处理专业的，但用信号处理的思路的确可以在一定程度上解决一些实际问题。希望这篇文章能够为国内不太多的音频信号处理爱好者们提供一些灵感（如果能够为某位朋友的硕士论文或者博士论文提供灵感的话我可真就万分荣幸了）。当然，最希望的还是能够为我无数热爱音乐的朋友们简单介绍这个还算勉强可以用的上的自己 **DIY** 伴奏的方法。如果有人拿去做卡拉 **OK** 卖给大公司挣钱养家的话，这也算是为伟大祖国的繁荣稳定尽了点薄力了^_^。本想做出一个可以比较方便地使用这一方法制作伴奏的小软件，无奈自己时间太紧，而且 **C++** 水平不是太高，因此这项任务就交给其他热爱音乐的人们去完成吧。

张翔

2009 年 12 月 18 日 @ Home