[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
H04M 9/08
H04B 3/20

[12]发明专利申请公开说明书

[21]申请号 98813859.X

[43]公开日 2001年3月7日

[11]公开号 CN 1286862A

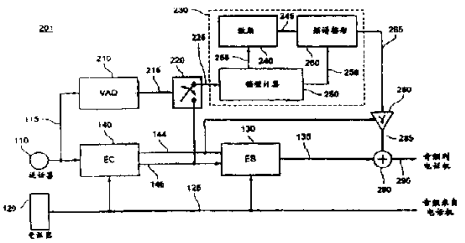
[22]申请日 1998.12.29 [21]申请号 98813859.X
[30]优先权
[32]1998.1.9 [33]US [31]09/005,145
[86]国际申请 PCT/US98/26238 1998.12.29
[87]国际公布 WO99/35813 英 1999.7.15
[85]进入国家阶段日期 2000.9.5
[71]申请人 艾利森公司
地址 美国北卡罗莱纳州
[72]发明人 E·D·罗梅斯博格 L·S·布勒鲍姆
C·N·S·古鲁帕兰
[74]专利代理机构 中国专利代理(香港)有限公司
代理人 王 勇 李亚非

权利要求书6页 说明书55页 附图页数5页

[54]发明名称 用于在通信系统中提供舒适噪声的方法和设备

[57]摘要

公开了在回声抑制系统中用于参数模拟背景噪声和产生舒适噪声 的方法和设备。按照示范性实施例,背景噪声模型是基于一组噪声模 型参数,而噪声模型参数又基于在回声抑制系统中对实际背景噪声的 计量。该示范性实施例包括自回归,自回归移动平均和频域模型。示 范性一阶自回归移动平均模型包括固定单一零点和可变单一极点。该 单一零点和单一极点在产生的模拟噪声中足够提供适合的频谱斜 率,和单一零点保证了模型是无条件稳定的。也公开了利用各种回声 抑制系统的示范性参数噪声模型的综合。



ISSN1008-4274

权　利　要　求　书

1. 一种配置用于衰减通信信号中回声成分的回声抑制装置，该设备包括：

一个噪声模拟处理器，配置以根据所述通信信号产生至少一个噪声模型参数，所述至少一个噪声模型参数限定了所述通信信号的噪声成分的一个参数模型；和

一个噪声发生处理器，配置以根据所述至少一个噪声模型参数提供模拟出的噪声样本。

2. 按照权利要求 1 的回声抑制装置，其中所述回声抑制装置衰减所述通信信号的回声和噪声成分两者以提供一个输出信号，其中所述模拟出的噪声样本被加入到所述输出信号中以替代被衰减的噪声成分。

3. 按照权利要求 1 的回声抑制装置，其中所述通信信号的噪声成分被模拟成一个自回归随机过程。

4. 按照权利要求 1 的回声抑制装置，其中配置所述噪声模拟处理器以通过根据所述通信信号样本帧计算一个自相关向量来产生所述至少一个噪声模拟参数。

5. 按照权利要求 4 的回声抑制装置，其中配置所述噪声模拟处理器以循环地平滑对所述通信信号几个帧计算出的自相关向量。

6. 按照权利要求 4 的回声抑制装置，其中配置所述噪声模拟处理器以非循环地平滑对所述通信信号几个帧计算出的自相关向量。

7. 按照权利要求 4 的回声抑制装置，其中配置所述噪声模拟处理器以根据所述自相关向量计算一组滤波器系数，所述滤波器系数为所述自回归处理限定了一个频谱整形滤波器。

8. 按照权利要求 7 的回声抑制装置，其中所述滤波器系数是使用一个 Levinson-Durbin 算法计算的。

9. 按照权利要求 8 的回声抑制装置，其中配置所述噪声发生处理器以产生一个零平均伪随机序列，所述序列具有与自相关向量零延迟系数成正比的方差，和对使用所述频谱整形滤波器的所述序列滤波以提供所述模拟出的噪声样本。

10. 按照权利要求 3 的回声抑制装置，其中配置所述噪声模拟处

理器以通过使用回归预定算法根据所述通信信号的样本计算一组滤波器系数来产生所述至少一个噪声模拟参数，所述滤波器系数为所述自回归处理限定了一个频谱整形滤波器。

11. 按照权利要求 10 的回声抑制装置，其中所述自适应算法是一个最小均方算法。

12. 按照权利要求 10 的回声抑制装置，其中所述自适应算法是一个标准化最小均方算法。

13. 按照权利要求 10 的回声抑制装置，其中配置所述噪声发生处理器以产生一个零平均伪随机序列，所述序列具有与所述自适应算法残余误差信号能量电平成正比的方差，和对使用所述频谱整形滤波器的所述序列滤波以提供所述模拟出的噪声样本。

14. 按照权利要求 1 的回声抑制装置，其中所述通信信号的噪声成分被模拟成为一个自回归移动平均过程。

15. 按照权利要求 14 的回声抑制装置，其中根据与环境相关的预先信息固定在频谱整形滤波器中限定所述自回归移动平均过程的一组零点，在该环境中将实现所述回声抑制装置。

16. 按照权利要求 15 的回声抑制装置，其中配置所述噪声模拟处理器以通过根据所述通信信号的样本帧计算一个自相关向量来产生所述至少一个噪声模拟参数。

17. 按照权利要求 16 的回声抑制装置，其中在计算出所述自相关向量之前由一个中间滤波器对所述样本帧滤波，所述中间滤波器包括对应所述频谱整形滤波器一组固定零点的一组固定极点。

18. 按照权利要求 16 的回声抑制装置，其中配置所述噪声模拟处理器以循环地平均对所述通信信号样本帧所计算出的自相关向量。

19. 按照权利要求 16 的回声抑制装置，其中配置所述噪声模拟处理器以非循环地平滑对所述通信信号样本帧所计算出的自相关向量。

20. 按照权利要求 16 的回声抑制装置，其中配置所述噪声模拟处理器以根据所述自相关向量为所述频谱整形滤波器计算一组极点。

21. 按照权利要求 20 的回声抑制装置，其中所述极点使用一个

Levinson-Durbin 算法计算的。

22. 按照权利要求 20 的回声抑制装置，其中配置所述噪声发生处理器以产生一个零平均伪随机序列，所述序列具有与自相关向量零延迟系数成正比的方差，和对使用所述频谱整形滤波器的所述序列滤波以产生所述模拟出的噪声样本。

23. 按照权利要求 15 的回声抑制装置，其中配置所述噪声模拟处理器以通过使用回归预定算法根据所述通信信号的样本为所述频谱整形滤波器计算一组极点来产生所述至少一个噪声模拟参数。

24. 按照权利要求 23 的回声抑制装置，其中所述自适应算法是一个最小均方算法。

25. 按照权利要求 23 的回声抑制装置，其中所述自适应算法是一个标准化最小均方算法.

26. 按照权利要求 23 的回声抑制装置，其中配置所述噪声发生处理器以产生一个零平均伪随机序列，所述序列具有与所述自适应算法残余误差信号能量电平成正比的方差，和对使用所述频谱整形滤波器的所述序列滤波以提供所述模拟出的噪声样本。

27. 按照权利要求 14 的回声抑制装置，其中所述自回归移动平均过程是一个一阶过程，和其中限定所述一阶过程的一个频谱整形滤波器包括一个单一固定零点和一个单一可变极点。

28. 按照权利要求 27 的回声抑制装置，其中根据与环境相关的预先信息设置所述单一固定零点，在该环境中将实现所述回声抑制装置。

29. 按照权利要求 27 的回声抑制装置，其中所述单一固定零点被设置得近似-1。

30. 按照权利要求 27 的回声抑制装置，其中所述单一固定零点被设置为-13/16。

31. 按照权利要求 27 的回声抑制装置，其中所述单一固定极点由α给出，和其中所述频谱整形滤波器的增益与 1-α成正比。

32. 按照权利要求 27 的回声抑制装置，其中配置所述噪声模拟处理器以使用一个自适应算法根据所述通信信号样本计算所述单一可变极点。

33. 按照权利要求 32 的回声抑制装置，其中所述自适应算法是

一个最小均方算法。

34. 按照权利要求 32 的回声抑制装置，其中所述自适应算法是一个标准化最小均方算法。

35. 按照权利要求 32 的回声抑制装置，其中配置所述噪声发生处理器以产生一个零平均伪随机序列，所述序列具有与自相关向量零延迟系数成正比的方差，和对使用所述频谱整形滤波器的所述序列滤波以产生所述模拟出的噪声样本。

36. 按照权利要求 1 的回声抑制装置，其中所述通信信号噪声成分的所述模型取决于所述通信信号样本的线性正交变换。

37. 按照权利要求 1 的回声抑制装置，其中所述通信信号噪声成分的所述模型是一个频域模型。

38. 按照权利要求 37 的回声抑制装置，其中配置所述噪声模拟处理器以通过根据所述通信信号样本帧计算频谱幅度的一个向量产生所述至少一个噪声模拟参数。

39. 按照权利要求 38 的回声抑制装置，其中通过计算所述通信信号的所述样本帧的付氏变换导出所述频谱幅度向量。

40. 按照权利要求 38 的回声抑制装置，其中配置所述噪声模拟处理器以循环地平滑对所述通信信号几个帧所计算出的频谱幅度向量。

41. 按照权利要求 38 的回声抑制装置，其中配置所述噪声模拟处理器以非循环地平滑对所述通信信号几个帧所计算出的频谱幅度向量。

42. 按照权利要求 38 的回声抑制装置，其中配置所述噪声发生处理器以产生相位值伪随机序列并根据所述序列和所述频谱幅度向量计算所述模拟出的噪声样本。

43. 按照权利要求 42 的回声抑制装置，其中通过计算一个复合向量的反付氏变换导出所述模拟出的噪声样本，在所述复合向量中的每个复合样本包括来自所述频谱幅度向量的一个频谱幅度和来自所述伪随机相位值序列的一个伪随机相位。

44. 按照权利要求 1 的回声抑制装置，其中配置所述噪声发生处理器以从所述通信信号样本缓存器中伪随机地选择单一样本，并使用由所述至少一个噪声模拟参数限定的一个频谱整形滤波器对所述伪

随机地选择出的单一样本序列滤波。

45. 一种回声抑制装置，包括：

一个回声抑制器，配置用于衰减通信信号的回声和噪声成分；

一个样本缓存器，用于存储所述通信信号的样本帧；和

一个噪声发生处理器，配置用于从所述缓存器中伪随机地选择单一样本，以便提供具有等于所述样本帧功率电平的功率电平的白噪声样本的序列。

46. 按照权利要求 45 的回声抑制装置，其中所述噪声发生处理器对所述白噪声样本序列滤波以提供被加入到所述回声抑制器输出中的舒适噪声。

47. 一种回声抑制装置，包括：

一个回声抑制器，配置用于衰减通信信号的回声和噪声成分；

一个噪声模拟和发生处理器，配置用于模拟所述通信信号的噪声成分并根据所述模型对所述回声抑制装置提供舒适噪声；

一个话音激活检测器，配置用于提供所述通信信号是否包括一个话音成分的一个指示；和

一个噪声电平估计器，配置用于计算所述通信信号噪声电平的估计值并提供所述通信信号能量电平是否小于所述估计值的一个指示，

其中所述噪声模拟和发生处理器配置用于只当所述话音激活检测器指示在通信信号中没有话音成分和所述噪声电平估计器指示所述能量电平小于所述估计值时更新所述模型。

48. 一种回声抑制装置，包括：

一个回声抑制器，配置用于衰减通信信号的回声和噪声成分，其中所述回声抑制器消除所述通信信号落入衰减窗口内的一部分；和

一个舒适噪声发生器，配置用于对所述回声抑制装置提供舒适噪声，其中所述舒适噪声发生器的一个输出被限制到所述衰减窗口中以提供一个被加入到所述回声抑制器音频输出中的受限舒适噪声输出。

49. 一种回声抑制装置，包括：

一个回声抑制器，配置用于衰减通信信号的回声和噪声成分，其中所述回声抑制器将所述通信信号落入衰减窗口内的一部分乘以一

个比例因子；和

一个舒适噪声发生器，配置用于对所述回声抑制装置提供舒适噪声，其中所述舒适噪声发生器的一个输出被限制到所述衰减窗口中以提供一个受限舒适噪声输出，和其中所述受限舒适噪声输出根据所述比例因子成比例以提供被加入到所述回声抑制器音频输出中的受限和成比例的舒适噪声输出。

说　明　书

用于在通信系统中提供舒适噪声的方法和设备

本发明背景技术

本发明涉及通信系统，更具体地涉及在双向通信链路中的回声抑制。

在许多通信系统中，例如陆地线路和无线电话系统中，经常在两个系统用户之间通过双向通信链路传送话音信号。在这种系统中，近端用户的语音一般由在通信链路一端的近端送话器检测，然后在通信链路上发送给远端受话器以再现和呈现给远端用户。相反地，远端用户的语音由远端送话器检测，然后通过通信链路发送给近端受话器以再现和呈现给近端用户。在通信链路的任何一端，由接近的送话器检测到的受话器输出可以不注意地被发送回通信链路，产生在用户看来不可接受的破坏性反馈或回声。

因此，为避免传输这种不需要的回声信号，送话器的音频输入应当与受话器的输出尽可能隔离。利用常规电话手机可以容易地实现必要的隔离，在常规电话中手机送话器位于靠近用户口部而手机受话器基本上覆盖用户的耳朵。可是，由于便携电话的物理尺寸已经减少，和由于免提受话器电话已经变得更普遍，制造商已经发展了配置，在新配置中从受话器到送话器的路径不被用户的头或身体阻碍。结果，在现代系统中对更高级回声抑制技术的需要已经变得重要了。

在免提汽车电话情况下尤其体现了这种需要，其中封闭的车内环境可以引起受话器信号的多次反射可以被耦合回到高增益免提送话器中。例如开关车窗或用户在驾驶中移动其头部等的用户在车内的移动和相对方向的改变和回声信号的强度使汽车环境中回声抑制任务进一步复杂。另外，最新开发的数字电话通过话音编码器处理语音信号，该编码器产生明显的信号延迟和产生非线性信号失真。这种延长的延迟将扩大从用户观点出发的信号回声问题，和额外的非线性失真使网络设备进行的回声抑制更困难。

电话制造商已经开发了大量各种回声抑制装置。图 1A 表示了示范性回声抑制系统 100。如图所示，示范性系统 100 包括一个送话器 110，一个受话器 120 和一个回声抑制器 130。送话器的一个音频输出

115 被耦合到回声抑制器 130 的一个音频输入上，而回声抑制器 130 的一个音频输出 135 用作电话（未示出）的近端音频输入。另外，来自电话的远端音频输出 125 被耦合到受话器 120 的音频输入和回声抑制器的基准输入上。

在工作中，回声抑制器 130 处理送话器信号 115 以将音频输出信号 135 提供给远端电话用户。更具体地，回声抑制器 130 根据远端音频信号衰减送话器信号 115，以便从受话器 120 到送话器 110 的音频回声不被返回到远端电话用户。

一般地，回声抑制器 130 是一个非线性削波型抑制器或一个线性比例型抑制器。削波型抑制器一般通过消除落入特定值范围内（即，在特定削波窗口内）的信号部分衰减送话器输出信号。另一方面，比例型抑制器通过将该信号乘以适当的比例因子衰减送话器输出信号 115。现在开发的混合抑制器结合了削波和比例两方面，例如通过将落入特定衰减窗口的送话器信号部分按比例衰减。无论如何，一般按照远端音频信号 125 的幅度直接或间接调节衰减水平（即，削波窗口和/或比例因子），以便送话器输出 115 被衰减到只有远端用户讲话的程度。

公知的中心削波器的常规削波型抑制器由例如 Rasmusson 等人的 1995 年 12 月 12 日所公开的题目为“回声消除系统和使用回声估计调整误差信号的方法”的美国专利第 5475731 号所描述。公知的 AC 中心削波器的另一个形式的削波型抑制器由 1996 年 12 月 31 日申请的题目为“用于通信系统中噪声和回声抑制的 AC 中心削波器”联合未审结专利美国专利申请第 08/775797 号描述。示范性比例型削波器在 Genter 的题目为“回声消除器处理技术和处理”的 1994 年 2 月 1 日公开的美国专利第 5283784 号中描述。在此称为 AC 中心衰减器的先进混合抑制器在题目为“改善通信系统中回声抑制的方法和设备”的相同日申请的联合未审结美国专利申请第 09/005149 号中描述。这些和其它削波、比例和混合型抑制器的先进控制在相同日申请的题目为“在通信系统中控制回声抑制的方法和设备”的联合未审结美国专利申请第 09/005144 号中描述。每个上面指认的专利以及上面指认的每个联合未审结专利申请在此整体引用作为参考。

图 1A 的回声抑制器 130 也可以与线性回声消除器合并以提供更

复杂的回声抑制系统。图 1B 表示了一个示范性系统 101，包括送话器 110、受话器 120 和图 1A 的回声抑制器 130，和一个音频回声消除器 140。如图所示，送话器 110 的音频输出 115 被耦合到音频回声消除器 140 的音频输出，而音频回声消除器 140 的控制和音频输出 144、145 分别耦合到回声抑制器 130 的控制和音频输出。回声抑制器 130 的音频输出 135 用作电话（未示出）的近端音频输入，而来自电话的远端音频输出 125 被耦合到受话器 120 的音频输入和音频回声消除器 140 和回声抑制器 130 的基准输入。

在工作中，音频回声消除器 140 动态地模拟从受话器 120 到送话器 110 的音频路径，并试图从送话器输出信号 115 中消除任何被送话器 110 拾音的受话器声音。用于模拟音频回声路径的普通算法包括众所周知的最小均方（LMS）算法和派生的例如标准化最小均方（NLMS）。以最小均方为基础的示范性消除器在上面引用的 Rasmusson 等人的美国专利第 5475731 中描述了。另外，以先进的标准化最小均方为基础的消除器在联合未审结美国专利申请第 08/8527291997 年 5 月 7 日申请的题目为“用于通信系统中的改善的回声消除器”中描述，该申请在此整体引用作为参考。

控制输出或控制尺度 144 表示由音频回声消除器 140 实现的消除瞬时电平，并被回声抑制器 130 用于确定将任何残留回声成分抑制到特定目的电平所需要的额外衰减电平。如同在图 1A 的系统 100 中，回声抑制器 130 可以是一个削波抑制器、一个定比抑制器或一个混合抑制器。按照例如上述引用的专利和专利申请中所描述的调节控制尺度 144。另外，当跟随回声消除器 140 时，回声抑制器 130 可以是一个简单的开关，该开关选择性地在适当时间对音频输出 135 进行静音（例如，在近端话音激活检测器表示送话器信号 1115 不包含近端语音期间）。

注意在图 1A 和 1B 的两个示范性系统 100、101 中，回声抑制器 130 衰减整个音频信号。因此，除了衰减回声外，回声抑制器 130 也衰减可能出现的任何背景噪声和/或近端语音。实际上，当回声抑制器 130 激活时背景噪声可以被抑制到远端用户可能错误地认为通话已经中断的程度。因此，为改善对于远端用户的通信质量，当回声抑制器 130 激活时现在的系统通常在电话音频信号 135 中加入舒适噪声。

例如，某些系统以伪随机数发生器（PRNG）产生的白噪声代替静音音频信号，其中根据对实际背景噪声能量的估计设置各种噪声样本。另外，上面所引用的 Genter 的美国专利第 5283784 号描述了一种类似方案，在该方案中噪声样本被频带限制到电话系统带宽并被存储在只读存储器（ROM）表中。然后在需要时从表中选择样本产生舒适噪声。而另一个方案由 1995 年 1 月 19 日申请的题目为“在免提蜂窝无线通信系统中用于回声衰减的方法和设备”美国专利申请第 08/375144 描述，该申请在此被完全引用作为参考。在存储器中存储了一组实际背景噪声的样本，并通过输出连续存储的以分组中随机起点开始的样本段产生舒适噪声。

尽管上述系统提供了某些优点，仍没有提供在频谱和幅度两方面与实际环境噪声紧密和协调匹配的舒适噪声。例如，通过产生白噪声产生的舒适噪声的频谱成分根据定义在音频频带上是统一的，而汽车背景噪声一般偏向频带的低端。也由于车与车之间频谱斜率不同和取决于流行的驾驶条件，只在 ROM 中存储一个示范性倾斜频谱还不够。另外，通过重复输出实际噪声样本段产生的舒适噪声包括明显的周期成分和因此通常听起来就象包含附加的失真音调。

因此，利用常规噪声发生技术，由于仅当回声抑制器 130 激活时选择性地加入和替换了舒适噪声，远端用户感觉到在所传输的背景噪声特性和内容上的连续变化。在背景噪声中感觉到的这种改变可能令人讨厌甚至不可忍受。例如，有着现在数字蜂窝电话中相对较长的延迟，实际背景噪声与模拟舒适噪声之间的不同经常被感觉为低音回声。因此，需要改善的方法和设备以产生回声抑制系统中的舒适噪声。

本发明概述

本发明通过提供在回声抑制系统中用于参数模拟音频背景噪声的方法和设备来满足上述和其它需要。通过与生成白噪声样本或重复输出所存储的噪声样本系列来产生舒适噪声的常规系统对比，本发明教导可以根据一组噪声模型参数有效地产生更高质量的舒适噪声（即，更紧密匹配实际系统环境噪声的舒适噪声），而该组参数又以对实际系统噪声的测量为根据。

按照本发明，在语音未激活期间（即只出现噪声）计算噪声模型

参数和在语音激活期间冻结参数。然后正流行的噪声模型参数被用于产生高质量的舒适噪声，无论何时由回声抑制器对噪声进行静音或衰减都以该舒适噪声替代实际噪声。由于该舒适噪声在特性和电平方面与实际背景噪声紧密匹配，远端用户感觉到信号的连续性和不受常规系统引入的人为噪声打扰。

按照第一示范性实施例，参数噪声模型以送话器输出的音频样本帧的自相关函数为依据。在几个样本帧上对自相关函数进行平滑和求平均，和将流行的平滑自相关函数用于计算全极点整形滤波器系数。然后整形滤波器被用于合成舒适噪声，该舒适噪声以具有与平滑自相关函数的第一零延迟成分成正比的变化的白噪声激励为依据。

按照第二示范性实施例，参数噪声模型以自相关函数和一组固定滤波器系数为基础，两者被组合应用以产生一个自回归移动平均（ARMA）整形滤波器。具体地，由送话器输出的音频样本帧通过根据固定滤波器系数构成的全极点滤波器，而滤波后的样本用于计算自相关函数。自相关函数在几个帧上被平滑，而流行的被平滑自相关函数用于计算全极点滤波器的系数。该全极点滤波器与根据固定滤波器系数派生的全零点滤波器级联以产生自回归移动平均频谱整形滤波器。然后该频谱整形滤波器被用于合成舒适噪声，该舒适噪声以具有与自相关函数的第一零延迟成分成正比的变化的白噪声样本激励为依据。

按照第三示范性实施例，参数噪声模型以离散频谱幅度向量为根据。根据由送话器输出的音频样本帧计算频谱幅度。具体地，频谱幅度作为音频样本帧的离散付氏变换来计算。另一个方式，当使用频域音频回声抑制器处理送话器信号时，直接从音频回声抑制器输出的音频样本帧中提取频谱幅度。在任一情况下，都在几个帧上平滑频谱幅度向量，而必要时流行和平滑的幅度向量被用于合成舒适噪声。具体地，均匀随机相位的激励被应用于流行的平滑幅度向量，而用离散付氏反变换将产生的复合频谱样本转化到时域。

按照第四示范性实施例，参数噪声模型包括一个具有单一固定零点和单一可变极点的自回归移动平均频谱整形滤波器。在语音非激活期间使用标准化最小均方算法计算频谱整形滤波器的可变极点，以按照频谱整形滤波器相反方向循环调节自适应滤波器。然后使用流行频

谱整形滤波器合成舒适噪声，该舒适噪声以具有等于实际系统噪声能量的白噪声样本激励为基础，其中通过从实际系统噪声样本的缓存随机选择的单个样本产生白噪声样本。有利地，频谱整形滤波器的单一固定零点和单一可变极点足以在产生的舒适噪声中提供适当的频谱斜率，和单一固定零点保证自适应反向滤波器无条件地稳定。结果，示范性实施例即稳固又不复杂。

上述技术可以有利地应用于需要或希望噪声模拟的任何通信系统中。例如，按照本发明的示范性回声抑制装置包括一个噪声模拟处理器和一个噪声发生器。该装置配置以衰减通信信号的回声成分，而噪声模拟处理器配置以产生一个或多个以通信信号为基础的噪声模拟参数。该噪声模拟参数限定了通信信号噪声成分的参数模型，而噪声发生处理器配置得提供以噪声模拟参数为基础的模拟噪声样本。在示范性实施例中，模拟出的噪声样本被加入到回声抑制装置的输出中以替代被衰减的噪声成分。

按照本发明的另一个方式的回声抑制装置包括一个回声抑制器，一个样本缓存器和一个噪声发生处理器。该回声抑制器配置以衰减通信信号的回声和噪声成分，而样本缓存器被用于存储通信信号的样本帧。该噪声发生处理器配置以伪随机地从缓存器中选择单一样本以便提供白噪声样本序列，该白噪声样本具有等于样本帧功率电平的功率电平。在示范性实施例中，对白噪声样本序列滤波以提供舒适噪声，该舒适噪声被加入到回声抑制器的输出中。

按照本发明的另一个改型回声抑制装置包括一个回声抑制器，一个噪声模拟和发生处理器，一个话音激活检测器和一个噪声电平估计器。该回声抑制器配置以衰减通信信号的回声和噪声成分，而噪声模拟和发生处理器配置以模拟通信系统的噪声成分并以该模型为基础为回声抑制装置提供舒适噪声。该话音激活检测器提供通信信号是否包括话音成分的指示，而噪声电平估计器计算通信信号的噪声电平和提供通信信号能力电平是否小于计算出的估计值的指示。按照本发明，仅当话音激活检测器指示在通信信号中没有话音成分和噪声电平估计器指示通信信号的能量电平小于计算出的噪声电平估计值时更新噪声模型。结果，不可能根据包括语音的通信信号部分错误地更新噪声模型。

按照本发明的又一个改型回声抑制装置包括一个回声抑制器和一个舒适噪声发生器。该回声抑制器配置以衰减通信信号的回声和噪声成分，而该舒适噪声发生器配置以为回声抑制装置提供舒适噪声。在回声抑制器中，通信信号落入衰减窗口中的部分被乘以一个比例因子。因此，舒适噪声发生器的输出被限制在衰减窗口中和基于比例因子成比例，以提供适当受限和成比例的舒适噪声输出，该舒适噪声输出被加入到回声抑制器的音频输出中。

下面参照附图中所示的说明性示例详细解释本发明的上述和其它特征。本领域技术人员将理解所提供的实施例是为说明和理解的目的，和在此可以推导出大量等同实施例。

附图简介

图 1A 表示实施本发明技术的示范性回声抑制系统。

图 1B 表示实施本发明技术的另一种形式的回声抑制系统。

图 2A 表示按照本发明包括一个噪声模拟和发生处理器的示范性回声抑制系统。

图 2B 表示按照本发明包括一个噪声模拟和发生处理器的另一种形式的回声抑制系统。

图 3 表示按照本发明的示范性一阶自回归移动平均噪声发生滤波器。

图 4 表示与图 3 的示范性噪声发生滤波器结合使用的示范性一阶噪声模拟训练处理器。

图 5 表示示范性回声抑制系统，其中按照本发明的噪声模拟发生处理器与示范性混合残留回声抑制器集成为一体。

本发明详细说明

图 2A 表示示范性回声抑制系统 200，其中本发明的舒适噪声部分与图 1A 的回声抑制配置被集成为一体。除了送话器 110、受话器 120 和回声抑制器 130 外，示范性回声抑制系统 200 包括一个话音激活检测器 210、一个第一开关 220、一个噪声模拟和发生处理器 230（包括一个激励块 240、一个模型计算块 250 和一个频谱整形块 260）和一个第二开关 270。

送话器 110 的音频输出 115 被耦合到回声抑制器 130 的音频输入和话音激活检测器 210 的音频输入。送话器 110 的音频输出也被耦合

到第一开关 220 的第一极。话音激活检测器 210 的二进制输出 215 被耦合到第一开关 220 的接入输入，而来自第一开关 220 第二极的音频输出 225 被耦合到模型计算块 250 的音频输入。

模型计算块 250 的激励参数输出 255 被耦合到激励块 240 的控制输入，而由激励块 240 输出的激励信号 245 被耦合到频谱整形块 260 的激励输入。模型计算块 250 的整形参数输出 256 被耦合到频谱整形滤波器 260 的控制输入，而频谱整形块 260 的模拟噪声输出 265 被耦合到第二开关 270 的第一输入极。

来自回声抑制器 130 的音频输出 135 被耦合到第二开关 270 的第二输入极，而第二开关 270 的输出极向电话（未示出）提供一个音频输入 275。来自电话的音频输出 125 被耦合到受话器 120 的音频输入和回声抑制器 130 的基准输入。

在工作中，话音激活检测器（VAD）210 输出指示送话器输出信号 115 中存在或不存在语音的一个二进制标志（例如，1=话音，0=没有话音）。实施话音激活检测器的方法是众所周知的。例如，欧洲电信标准协会（ETSI）文献 GSM-06.82 描述了适合于本发明目的的一种实施。

当话音激活检测器 210 指示在送话器信号 115 中没有语音出现时（即，仅出现噪声），送话器信号 115 通过第一开关 220 被连接到噪声模拟和发生处理器 230 的音频输入（更具体地，到模型计算块 250 的音频输入），而噪声模拟和发生处理器 230 使用音频信号 115、225 计算和/或更新参数噪声模型。可是，当话音激活检测器 210 指示在送话器信号 115 中出现语音时，第一开关 220 被断开，噪声模型参数被冻结，而噪声模拟和发生处理器 230 使用流行的参数噪声产生舒适噪声 265 的样本。

在图 2A 的配置中，第二开关 275 用于选择性地以抑制器输出 135 的舒适噪声 265 替代电话的近端音频信号 275。换句话说，当回声抑制器 130 被激活并衰减送话器信号 115 的噪声成分时，舒适噪声信号 275 被传递给远端用户。相反，来自回声抑制器 130 的音频输出 135 被传递给远端用户。在另一个方式的配置中，第二开关 270 被一个加法装置替代，和舒适噪声 265 的比例形式被加入到回声抑制器输出 135 中以提供补偿回声抑制器 130 产生的噪声衰减的舒适噪声。换句

话说，由于回声抑制器 130 变得更活跃而且将背景噪声衰减更大程度，所加入的舒适噪声电平被增加，和反之相同。

图 2B 表示示范性回声抑制系统 201，其中图 2A 的该系统 200 的舒适噪声部分与图 1B 的回声抑制配置合并。如图所示，示范性系统 201 包括图 1B 的送话器 110、受话器 120，回声抑制器 130 和音频回声消除器 140，以及图 2A 的话音激活检测器 210、第一开关 220 和噪声模拟和发生处理器 230。示范性系统 201 也包括一个γ衰减块 280 和一个加法装置 290。

送话器 110 的音频输出 115 被耦合到音频回声消除器 140 的音频输入和话音激活检测器 210 的音频输入。音频回声消除器 140 的控制输出 144 被耦合到残余回声抑制器 130 的控制输入和γ衰减块 280 的控制输入。音频回声消除器 140 的音频输出 145 被耦合到回声抑制器 130 的音频输入和第一开关 220 的第一极。话音激活检测器 210 的二进制输出 215 被耦合到开关 220 的接入输入，和来自开关 220 第二极的音频输出 225 被耦合到噪声模拟和发生处理器 230 的音频输入。

上面针对图 2A 的实施例描述了噪声模拟和发生处理器 230 的内部连接。噪声模拟和发生处理器 230 模拟出的噪声输出 265 被耦合到γ衰减块 280 的信号输入，而γ衰减块 280 的调节后噪声输出 285 被耦合到加法装置 290 的第一相加输入。回声抑制器 130 的音频输出被耦合到加法装置 290 的第二相加输入，而加法装置 290 的输出 295 用作电话（未示出）的音频输入。来自电话的音频输出 125 被耦合到受话器 120 的音频输入和音频回声消除器 140 和回声抑制器 130 的基准输入。

在工作中，话音激活检测器 210 通常起针对图 2A 的上述作用。更具体地，当话音激活检测器 210 指示在送话器信号 115 中没有出现语音时，对回声抑制器 130 的音频输入 145 被通过第一开关 220 连接到噪声模拟和发生处理器 230 的音频输入。该噪声模拟和发生处理器 230 然后使用音频信号 145、225 来计算和/或更新参数噪声模型。可是，当话音激活检测器 210 指示在送话器信号 115 中出现语音时，第一开关 220 被断开，噪声模型参数被冻结，和噪声模拟和发生处理器 230 使用流行噪声模型产生舒适噪声 265.

如图所示，按照控制尺度 144 通过γ衰减块 280 将舒适噪声样本

265 改变比例和/或削波，以提供具有与非线性回声抑制器 130 所衰减的噪声电平相匹配的电平的调节后舒适噪声样本 285. 调节后舒适噪声样本 285 通过加法装置 290 被加入到抑制器输出 135，而产生的音频信号 295 被传递给远端用户。另一个方式，图 2A 的第二开关 275 可以替换为γ衰减块 280 和加法装置 290，以提供抑制器音频信号 135 与舒适噪声样本 265 之间的简单切换。

在示范性系统 200 中，图 2A 和 2B 的由噪声模拟和发生处理器 230 提供的参数噪声模型 201 一般包括两个部分。即，一个频谱整形滤波器和一个激励信号。该频谱整形滤波器和激励信号通过频谱整形块 260 和激励块 240 使用频谱整形参数（例如，滤波器系数）255 和由模型计算块 250 提供的至少一个激励参数 256 分别实现。激励和频谱整形参数被存储在静态存储器中并被用于必要时产生舒适噪声样本 265（即，当非线性回声抑制器 130 被激励时）。

最好，模型参数可以一帧一帧地或一个样本一个样本地更新，例如根据实施的音频回声消除器 140 的特定类型。在帧方式的实施中，噪声模型参数被使用已知技术在几个更新周期上平滑，以防止周期到周期之间突然的用户可觉察的改变。当例如噪声模型参数根据包含话音成分的音频样本被错误地更新时（即当话音激活检测器 210 将话音和噪声误认为仅仅是噪声时）可以产生这种突然的改变。可是注意，平滑的程度实际上与快速适应特性和背景噪声电平改变的需要相平衡。

按照第一示范性实施例，舒适噪声 265 被模拟为具有由下式给出的频谱整形滤波器的自回归（AR）随机过程：

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

其中 G 是增益常数而 a_i，i=1 到 N 是滤波器系数。

在第一实施例中，通过计算音频信号 225 样本帧自相关函数 $\underline{r}$ 的前面 N+1 个系数确定第 N 个滤波器系数 a_i，然后使用这些值形成矩阵关系 $\underline{R}\,\underline{a}=\underline{r'}$，其中 $\underline{R}$ 是一个 N 乘 N 矩阵具有由 $R_{ij}=r_{|i-j|}$ 给出的 i 行和 j 列单元，和 $\underline{a}=[a_1, a_2, \cdots\cdots, a_N]^T$ 是未知滤波器系数 a_i 的列向量，和 $\underline{r'}=[r_1, r_2, \cdots\cdots, r_N]^T$ 是最后 N 个自相关系数的列向量。本领域

技术人员将确认存在许多从矩阵关系派生出未知滤波器系数 a_i 的方法。例如，在示范性实施例中，公知的 Levinson-Durbin 算法用于循环地推导未知滤波器系数 a_i。

最好，自相关 r 提供了整套的噪声模型参数，描述了频谱整形滤波器和激励信号两者。具体地，频谱整形滤波器由上面公式中所示的 a_i 系数限定，而激励信号用具有与零延迟自相关值 R_0 成正比的方差的零平均随机序列实现。几个自相关函数 r 的值被循环或非循环地平滑。无论何时需要舒适噪声综合，自相关函数 r 平滑后的形式被保持在噪声模拟和发生处理器 230 内的静态存储器中并被用于计算滤波器系数 a_i。

在另一个方式的实施例中，根据样本方式计算自回归模型的滤波器系数 a_i。具体地，使用公知的自适应算法例如最小均方（LMS）或递进最小平方（RLS）一个样本一个样本地直接更新或调整滤波器系数 a_i。因此，N 个滤波器系数 a_i 限定了上面的频谱整形滤波器，并使用另一个方式的方差 σ_e^2 模拟激励信号，该方差与自适应算法所提供的残余误差信号功率成正比。

按照另一个示范性实施例，舒适噪声 265 可以用一个自回归移动平均（ARMA）随机过程模拟，该过程具有如下给出的频谱整形滤波器：

$$H(z) = G\frac{1-\sum_{i=1}^{M} b_i z^{-i}}{1-\sum_{i=1}^{N} a_i z^{-i}}$$

其中 G 是增益常数而 a_i（i=1 到 N）和 b_i（i=1 到 M）是滤波器系数。

最好，与自回归模型相比，自回归移动平均模型足够灵活以使用低阶频谱整形滤波器紧密配合复杂的背景噪声频谱。可是，由于根据音频信号 225 动态地估计滤波器系数 b_i 的常规方法相对复杂和可能不稳定，根据与背景噪声环境总性能相关的先验信息为滤波器系数 b_i 设置固定值，回声抑制系统将工作在该环境中（例如，对于车载免提辅助应用的汽车噪声环境）。给出了系数 b_i 的固定值，音频信号 225 被首先如下滤波：

$$G(z)=\frac{1}{1-\sum_{i=1}^{M}b_i z^{-i}}$$

这样作消除了总模型的移动平均部分的影响，并允许使用针对第一示范性实施例描述的技术模拟剩余自回归部分。对于以帧为基础的系统，自回归移动平均模型由固定滤波器系数 b_i 和自相关函数 r 限定。在此情况下，激励信号是用具有与零延迟自相关值 R_0 成正比的方差的随机序列实现的。下面参照图 3-4 详细描述了示范性基于样本的自回归移动平均模型。

最好，本发明的参数模拟技术不限于时域的参数组。另一个方式，可以使用线性正交变换将时域音频样本帧转换为可以构成参数组的另一个域。这种正交变换的示例包括离散付氏变换，离散余弦变换和离散子波变换，本领域技术人员可以确认许多其它的变换。在本发明的示范性实施例中，由一组 N 个频谱幅度限定的一个频域参数模型，如下给出：

$$\underline{M}=\left|\mathrm{DFT}\{X_n\}\right|_{\omega(i),\ i=1\cdots N}$$

其中ω（i），i=1 到 N 是离散频率点，而向量 X_n 代表音频输出信号 225 样本帧。“DFT”运算代表公知的离散付氏变换，并在实际中使用低复杂实施方式例如也是公知的快速付氏变换（FFT）实现。在该实施例中，频谱整形参数被包括在幅度向量 M 中，该向量只在正频率上被评价，因为频谱对实数 X_n（例如音频信号样本）在ω=0 上对称。

离散频率ω（i）可以是快速付氏变换输出中离散频率的全体或仅仅是子集。如果需要，直接在幅度向量 M 上进行多帧平滑。通过将伪随机相位值应用于频谱放大产生激励信号。统一的伪随机噪声发生器用于产生固定范围[0，2π]内的相位值。时域舒适噪声然后通过将具有伪随机相位值的幅度向量 M 进行反向快速付氏变换产生时域舒适噪声。也注意到，这种类型的模型对于频域回声抑制器也非常有用。在此情况下，幅度向量 M 可以被直接从抑制器输出的频域样本中提取。

如上说明，本发明的另一个示范性实施例利用了基于样本的自回归移动平均舒适噪声模型。更具体地，该示范性实施例利用了具有单一固定零点和单一可变极点的一阶自回归移动平均模型。现在的实验性研究已经表明，这种一阶自回归移动平均模型为具有最小模型复杂

性的舒适噪声提供了足够的频谱精确度。实际上，一阶模型提供了在最先进的基于线性预测编码（LPC）的话音编码器中的十阶自回归模型相等的性能。

按照实施例，在频谱整形滤波器中单一固定零点接近 Z=-1，以衰减高频音频成分。当必要时，单一可变极点则被用于提供频谱偏移或倾斜（回忆，例如，汽车背景噪声的频谱一般向低端频率偏移）。因此，本发明教导了可用于密切配合实际背景噪声的频谱内容的相对简单和容易实现的一阶滤波器。另外，在频谱整形滤波器中的固定零点意味着在自适应滤波器中的固定极点，该滤波器用于在模拟阶段获得频谱整形滤波器的系数。在自适应滤波器中的单一固定极点反过来意味着自适应滤波器无条件地稳定。因此，该实施例也非常稳固。

图 3 和 4 分别表示了按照示范性实施例的一阶频谱整形滤波器 300 和补充的标准化最小均方自适应滤波器 400。一阶频谱整形滤波器 300 对应图 2A 和 2B 中的频谱整形块 260，而补充的自适应滤波器 400 对应图 2A 和 2B 的模型计算块 250。

在图 3 中，一阶频谱整形滤波器包括一个第一延迟块 310，一个 $13/16^{ths}$ 增益块 320，一个加法装置 330，一个 1-α增益块 340，一个第二延迟块 350 和一个α增益块 360。模拟图 2A 和 2B 的激励信号 245 的一个白噪声激励信号被耦合到加法装置 330 的第一相加输入和第一延迟块 310 的一个输入。第一延迟块 310 的输出 315 被耦合到 $13/16^{ths}$ 增益块 320 的输入，而 $13/16^{ths}$ 增益块 320 的输出 325 被耦合到加法装置 330 的第二相加输入。加法装置 330 的输出 335 被耦合到 1-α增益块 340 的输入和第二延迟块 350 的输入。第二延迟块 350 的输出 355 被耦合到α增益块 360 的输入，而α增益块 360 的输出 365 被耦合到加法装置 330 的第三相加输入。1-α增益块 340 的输出用作模拟背景噪声，对应图 2A 和 2B 的舒适噪声 265。

在图 4 中，补充一阶自适应滤波器 400 包括一个第一延迟块 410，一个第一乘法器 420，一个第二延迟块 430，一个第一加法装置 440，一个第二乘法器 450，一个第二加法装置 460，一个 $13/16^{ths}$ 增益块 470，一个第三延迟块 480 和一个标准化增益块 490。对应图 2A 和 2B 的有色背景噪声信号被耦合到第二加法装置 460 的相加输入和第一延迟块 410 的输入。第一延迟块 410 的输出 415 被耦合到第一乘法器 420

的第一输入和第二乘法器450的第一输入。第一乘法器420的输出425被耦合到第二加法装置460的相减输入，而第二乘法器450的输出455被耦合到第一加法装置440的第一输入。

第一加法装置440的输出445被耦合到第二延迟块430的输入，而第二延迟块430的输出α被耦合到第一乘法器420的第二输入和第一加法装置440的第二输入。第二延迟块430的输出α用作图3的频谱整形滤波器300的滤波器整形参数（即作为单一滤波器系数）并对应图2A和2B的滤波器整形参数256。

第二加法装置460的输出465被耦合到第三延迟块480的输入和标准化增益块490的输入。第三延迟块480的输出485被耦合到$13/16^{ths}$增益块470的输入，而$13/16^{ths}$增益块470的输出475被耦合到第二加法装置460的第二相减输入。标准化增益块490的输出495被耦合到第二乘法器450的第二输入。

在工作中，图4的自适应滤波器400利用标准化最小均方算法将实际背景噪声225进行白化。产生的滤波器参数α则被用于图3的相反频谱整形滤波器300以产生基于白噪声激励信号245的模拟舒适噪声265。

按照实施例，通过对每个单独样本使用随机指针从实际噪声样本缓存器读取单个噪声样本产生白噪声激励信号245。以此方式产生的激励信号245产生具有等于实际背景噪声功率电平的白噪声样本。最好，即使当实际噪声样本缓存器由于错误而包含语音成分时激励信号245的白化度也不受影响。由于话音激励检测器210有时可能错误地指示没有语音（例如，当由于在免提应用中汽车加速或减速引起背景噪声电平改变时），该特征是重要的。

如图3和4所示，噪声发生和自适应滤波器300、400的固定零点/极点被设置在Z=-13/16处。这样做，在自适应滤波器400中将高频提升限制到20dB，因此避免了实际中的溢出问题。另外，当与单一可变极点结合时，由噪声发生滤波器300提供的高频的固定20dB衰减足够提供40dB的频谱斜率（实验研究已经表明该频谱斜率对于汽车免提应用环境的实际背景噪声是典型的）。

与采用全部零点滤波器相反，在自适应滤波器400中包含固定极点改善了模型的精确调整能力，以匹配实际背景噪声的频谱内容。另

外，在频谱整形滤波器 300 中包含 1-α增益放大器将滤波器 300 的 DC 增益固定在 29/16，实验研究已经表明当话音激励检测器 210 的判决由噪声电平估计器量化时（如同参照图 5 在下面所详细描述的），该增益提供了在模拟的和实际噪声之间匹配的良好电平。通过对比，实验研究也已经表明在输入和输出信号中包含相同的功率产生了模拟的噪声，当通过噪声电平估计器对话音激励检测器 210 的判决进行量化时该噪声听起来比实际噪声安静。

尽管实际上实际背景噪声电平可以快速改变，实际背景噪声的频谱形状一般更逐渐地变化。因此，自适应滤波器 400 配置得以便舒适噪声 265 的频谱也逐渐变化。具体地，对标准化块 490 选择相对小的更新增益常数以便调整不会出现太块。实验研究已经表明 4 分母增益乘法器提供了跟踪速度与平滑之间的良好折中。通过使用音频信号 225 中能量的按块测量标准化更新增益（如图 4 所示），调整速度可以独立于背景噪声电平。

除了使舒适噪声 265 特性中感觉到的不连续性最小外，自适应滤波器 400 相对小的更新增益也提供了对基于包含话音的音频样本块错误调整的免疫性。可是注意，由于只要话音激励检测器 210 指示只有噪声，白噪声激励信号 245 就更新，背景噪声电平中的改变几乎马上被引入。结果，图 3 和 4 的示范性实施例很快跟踪背景噪声电平，同时保持舒适噪声 265 频谱形状稳定。

注意，对自回归模型使用最小均方算法计算基于音频样本块的频谱整形系数需要与对样本块计算自相关系数然后对自回归模型使用 Levinson 算法计算滤波器系数同样数量的 DSP 周期。可是，Levinson 算法也需要额外的 DSP 周期在几个块或帧上进行平滑。因此，由于平滑功能在最小均方算法中是固有的，最小均方方法导致 DSP 周期纯节省。

也注意，假设为最小均方算法，从十阶自回归模型进行到一阶自回归模型导致 DSP 周期节省大约百分之八十（开销的复杂性呈现与阶数的严格正比）。可是，对自回归移动平均模型增加另外的极点，和对标准化最小均方算法增加另外的乘法，增加了对所需要的 DSP 周期的数量。但是，从利用最小均方算法的十阶自回归模型进行到利用标准化最小均方算法的一阶自回归移动平均模型导致 DSP 周期至少节省

百分之五十。

最好，图 3 和 4 的示范性实施例可以被引入到任何图 1A，1B，2A 和 2B 所示的回声抑制系统 100，101，200，201 中。换句话说，该示范性实施例可以与任何类型的回声消除器和/或任何类型回声抑制器集成为一体。图 5 表示了示范性系统 500，其中示范性实施例被与图 1B 和 2B 中表示的回声抑制系统类型集成。

如图所示，示范性系统 500 包括送话器 110，受话器 120，音频回声消除器 140，和图 1B 的非线性回声抑制器 130 以及话音激活检测器 210，第一开关 220，模型计算块 250，频谱整形块 260 和图 2B 的加法装置 290。回声抑制系统 500 也包括一个噪声电平估计器 510，一个样本缓存器 520，一个削波器 530，一个第一乘法器 540，一个第二加法装置 550，一个第二乘法器 560 和一个包络检测器 570。

送话器 110 的音频输出 115 耦合到话音激活检测器 210 的音频输入和音频回声消除器 140 的音频输入。话音激活检测器 210 的二进制输出 215 耦合到噪声电平估计器 510 的控制输入，而噪声电平估计器 510 的二进制输出 515 耦合到开关 220 的接入输入。音频回声消除器 140 的音频输出 145 耦合到噪声电平估计器 510 的音频输入、开关 220 的第一极和非线性回声抑制器 130 的音频输入。音频回声消除器 140 的第一控制尺度 144a 耦合到非线性回声抑制器 130 的第一控制输入和第二加法装置 550 的相减输入。音频回声消除器 140 的第二控制尺度 144b 耦合到第二乘法器 560 的第一输入，而开关 220 的第二极输出 225 耦合到模型训练块 250 的音频输入和样本缓存器 520 的排队输入。

一个随机指针 515 耦合到样本缓存器 520 的控制输入，而样本缓存器 520 的控制输入起对频谱整形滤波器 260 的激励输入 245 的作用。模型训练块 250 的滤波器系数输出 256 为频谱整形滤波器 260 提供了控制输入，由频谱整形滤波器 260 输出的模拟汽车噪声耦合到削波器 530 的噪声输入。削波器 530 的削波后噪声输出耦合到第一乘法器 540 的第一输入，而第一乘法器 540 的成比例的噪声输出 545 耦合到第一加法装置 290 的第一输入。非线性回声抑制器 130 的音频输出 135 耦合到第一加法装置 290 的第二输入，而第一加法装置 290 的输出 295 用作电话（未示出）的音频输入。

来自电话的音频输出耦合到包络检测器 570 的基准输入，音频回声消除器 140 和受话器 120。包络检测器 570 的输出 575 耦合到第二乘法器 560 的第二输入，而第二乘法器 560 的输出 565 耦合到削波器 530 的控制输入。常数 1 耦合到第二加法装置 550 的相加输入，而第二加法装置 550 的输出 555 耦合到第一乘法器 140 的第二输入。

在工作中，残余回声抑制器 130 和音频回声消除器 140 一般起针对图 1B 和 2B 所描述的作用。通过举例，图 5 的残余抑制器 130 表示为上面引用的美国专利申请第 09/005149（相同日期同时申请的“用于改善通信系统中的回声抑制的方法和设备”）中描述的混合抑制器类型。

一般，AC 中心衰减器 130 利用适当的衰减因子α将音频输入信号 145 缩小一部分。更具体地，AC 中心衰减器 130 将落入由窗口尺寸Δ定义的衰减窗口内的音频信号 145 缩小一部分。衰减窗口的中心随音频信号 145 的幅度移动，和衰减器 130 提供了具有信号失真最小的良好残余抑制。

图 5 的音频回声消除器 140 可以是例如上面引用的未审结美国专利申请第 08/852729（1997 年 5 月 7 日申请的“一种用于通信系统中的改善的回声消除器”）中描述的类型。这种回声消除器可以动态地计量其正消除的回声电平，和因此提供了对 AC 中心衰减器 130 的控制尺度 144a，144b。

如图所示，第一控制尺度 144a 直接用作衰减因子α。第二控制尺度 144b 被乘以远端音频信号 125 的包络（通过乘法器 560 和包络检测器 570），而产生的控制信号 565 用作窗口尺寸Δ。回声消除器 140 和 AC 中心衰减器 130 的详细工作和综合在上面参考的美国专利申请中描述，并在此为简短而省略。

图 5 的舒适噪声方面一般如同参照图 2B，3 和 4 的描述。通常，音频信号 145 在没有语音时间段期间被传递到样本缓存器 520，而训练处理器 260（即自适应滤波器 400）处理样本缓存器 520 的内容，以对噪声发生处理器 260（即，频谱整形滤波器 300）提供整形参数 256（即，滤波器系数α）。在语音时间段期间，冻结整形参数 256，和噪声发生处理器 260 对激励信号 245 滤波以提供舒适噪声样本 265。

如上所述，通过随机指针从样本缓存器 520 中随机选择样本产生激励信号 245。激励信号 245 因此包括具有等于实际背景噪声功率的白噪声样本。注意，由于样本缓存器 520 不受整个系统的帧尺寸的限制（例如，在许多 TDMA 电话应用中 160 个样本），图 5 的配置可以在基于样本和基于帧的通信系统中实现。

也注意，在图 5 中话音激活检测器 210 受噪声电平估计器 510 所提供的判决的限制。换句话说，只在下面时间更新噪声模型 a）话音激活检测器 210 指示没有语音和 b）噪声电平估计器 510 指示音频信号 145 的能量少于在音频信号 145 中的噪声电平估计值。以此方式限制话音激活检测器判决减少了噪声模型被根据包含话音的样本块进行错误调整的可能性。

可是，现在，以此方式限制话音激活检测器 210 也导致所模拟出的舒适噪声往往具有低于实际背景噪声功率电平。换句话说，由于在噪声模型被训练之前噪声电平估计器 510 要求音频信号 145 中的能量少于流行噪声电平估计值，用于训练噪声模型的实际噪声被朝着低端偏置。可是，这可以通过如同上面针对图 3 和 4 所描述的在噪声发生处理器 260（即，在频谱整形滤波器 300）中提供适当的补偿增益因子来弥补。

本领域技术人员将理解可以使用已知技术计算噪声电平估计值。此外，下面将通过示范性伪码描述用于计算噪声电平估计值的新颖方法。可是，由于噪声电平估计器 510 的具有工作对目前要求的本发明不重要，在此省略了详细描述。

由于 AC 中心衰减器 130 不衰减音频信号 145 落入由Δ定义的衰减窗口外的部分，模拟出的噪声信号 265 通过如图 5 所示的削波器 530 限制在±Δ。还由于音频信号 145 落入衰减窗口内被乘以衰减因子α，在该窗口内的模拟出的噪声通过第一乘法器 540 和第二加法装置 500 乘以 1-α。产生的削波后和成比例的噪声信号 545 因此有与 AC 中心衰减器 130 消除的实际噪声相同的特性和电平。如图所示，削波后和成比例的噪声信号 545 然后被加入到 AC 中心衰减器输出 135 中以提供所需要的近端音频信号 295。

为降低不包括一个音频回声消除器前端的回声抑制系统（即，对于如图 1A 和 2A 所示的这种系统）的复杂性，控制信号 144a、144b

可以设置为常数并以相同方式实现舒适噪声特性。还对于在音频回声消除器 140 与残余回声抑制器 130 之间插入了一个噪声抑制器的系统，可以通过向噪声模拟和发生处理器（即，向图 5 中的第一开关 220 第一极）提供噪声抑制器的音频输出而非回声消除器的音频输出 145 来适当地调节舒适噪声的电平和特性。

注意，当使用纯削波型残余抑制器时，第一控制信号 144a 不是必须的。在此情况下，使用第二控制信号 144b 和削波器 530 调节舒适噪声电平，而第一乘法器 540 和第二加法装置 550 是不需要的。反之，当使用纯比例型残余抑制器时，第二控制信号 144b 不是必须的。因此，使用第一控制信号 144a、第一乘法器 540 和第二加法装置 550 调节舒适噪声电平，不需要削波器 530。

为进一步说明本发明的各个特征和优点，下面通过伪码描述类似图 5 的回声抑制系统。由于是使用 32 位数字信号处理器实现的所写出的伪码模拟示范性系统。本领域技术人员将理解该伪码实际上是示范性的和可以使用各种硬件配置实现该实施例。

```
**********************************************************************
%AEC 和 ANLP 模拟程序用于 MATLAB
%在运行该程序之前，设置下列变量:
%inFile=输入文件名称，left=远端，right=近端。
%outFile=输出文件名称，left=ANLP 输出，right=AEC 输出。
%所有文件使用 DAT-Link 的原始格式。
%estnoise.m 包括估计噪声功能。
%词表:
%EC=回声消除器=线性回声抑制器
%AEC=音频回声消除器=受话器-回声消除器
%NLP=非线性处理=残余回声抑制器=AC 中心衰减器
%ANLP=音频非线性处理
%VAD=话音激活检测器

%用分数表示的最大位置值。
ONE=32767/32768;
```

```
%读取包含远端和近端信号的文件。
FidIn=fopen ( inFile,  'r' ) ;
If  fidIn== -1
   Error ( [ 'Eroor opening file' inFile])
End
[Lrmatrix, wordCount]=fread(fidIn, [2, inf], 'int16');
fclose(fidIn);

%在更新综合期间的样本数量。
FRAME-SIZE=160;
%给出的帧尺寸越大对双方谈话和近端噪声的坚固性越大
%其往往趋于零。
%较大则改善了检测收敛的能力，因为相关更新与帧尺寸成比例增长
而非相关（噪声）随帧%尺寸平方根增长.
%较小则改善对变化的响应时间（回声路径，单一≥双方谈话）和加速
收敛。
%较小也改善丢弃元音声音的能力.
%160 用于容易移植每帧 160 个样本的 TDMA 电话。
%产生的 20ms 帧对于将语音分成静态信号段是近似最佳的。
NFRAMES=floor((wordCount/2)/FRAME-SIZE);%处理的帧数量。
Clear wordCount
Nsamples=NFRAMES*FRAME-SIZE;用于调试的样本形式向量

%
%从分开的文件加载送话器（上行）和受话器（下行）VAD 输出。如果
没有找到每个文件，%运行近端和远端声音文件上的 C 可执行文件，
并在具有相同前缀饿文件中存储 VAD 输出。
%
fidIn = fopen([inFile '_up_vad'],'r');
if fidIn == -1
  disp(['File = ' inFile '_up_vad not found -- creating ...'])
```

```
  fidOut =fopen('vad_in.raw','w');
  fwrite(fidOut, LRmatrix(2,:), 'int16');     % 上行音频
  fclose(fidOut);

  !nrsim -s=f vad_in.raw junk vad_out.bit
  !rm vad_in.raw
  !rm junk.flt

  fidIn = fopen('vad_out.bit','r');
  if fidIn == -1
    error('Error opening file = vad_out.bit')
  end
  micVad = fread(fidIn, NFRAMES, 'int8');
  !rm vad_out.bit
  fclose(fidIn);

  fidOut =fopen([inFile '_up_vad'],'w');
  fwrite(fidOut, micVad, 'int8');
  fclose(fidOut);
else
  micVad = fread(fidIn, NFRAMES, 'int8');
  fclose(fidIn);
end

fidIn = fopen([inFile '_down_vad'],'r');
if fidIn == -1
  disp(['File = ' inFile '_down_vad not found -- creating ...'])
  fidOut =fopen('vad_in.raw','w');
  fwrite(fidOut, LRmatrix(1,:), 'int16');     % 下行音频
  fclose(fidOut);

  !nrsim -s=f vad_in.raw junk vad_out.bit
  !rm vad_in.raw
  !rm junk.flt

  fidIn = fopen('vad_out.bit','r');
  if fidIn == -1
    error('Error opening file = vad_out.bit')
  end
  speakerVad = fread(fidIn, NFRAMES, 'int8');
  !rm vad_out.bit
  fclose(fidIn);

  fidOut =fopen([inFile '_down_vad'],'w');
  fwrite(fidOut, speakerVad, 'int8');
  fclose(fidOut);
else
  speakerVad = fread(fidIn, NFRAMES, 'int8');
  fclose(fidIn);
end
```

%将输入乘比例使用-1 到 1 的范围。

Lrmatrix=Lrmatrix/32768;

%在一个样本帧上累计右移位数值的比特数量。

```
FRAME-BITS=ceil(log2(FRAME-SIZE));
%将能量量化成 32 位的比例因子（Z8.23 格式 w/FRAME-SIZE=160）
ENERGY-SCALE=2^(31-FRAME-BITS);
%在基准延迟线中的抽头数量.
%它必须足够长使高延迟抽头几乎与远端信号不相关而只具有近端能量.
AEC-REF-TAPS=512;

%在 FIR 回声估计滤波器中的抽头数量。
AEC-COEF-TAPS=256;

%用于计量近端能量的更新向量高延迟部分中的抽头数量.
AEC-NEAR-TAPS=128;

%用于捕获汽车噪声样本的向量长度。
COMFORT-NOISE-SIZE=128;

%计算在更新向量峰值/基线比的阈值用于确定最大更新增益。对于在[-1, 1]范围内均匀的
%噪声，所希望的峰值更新幅度是 g*FRAME-SIZE/3，同时所希望非相关插头更新向量的
%RMS 是 g*sqrt（FRAME-SIZE）/3，其中 g 是回声路径增益。因此，最大更新向量峰值/
%基线比是 sqrt（FRAME-SIZE）。
%可是，话音没有类似噪声的恒定包络。因为 FRAME-SIZE 远远小于更新向量长度，语音
%的一个突发有时在更新向量区内，其中峰值不在计量基线的区域中计量。因此，实际峰
%值/基线比可以非常高（>300）。
%将阈值设置太低会引起由于即使对噪声或双方谈话条件下高能量远端信号的高增益更新
%产生的不稳定.
```

%将阈值设置太低会引起由于只对大的回声路径变化高能量远端信号的高增益更新产生的
%慢衰减。
%经验化地确定该阈值作为一种折中。
AEC-MAX-GAIN, THRESH=16;

%对于噪声的峰值/RMS 与帧尺寸无关。用于抵制近端话音/噪声和远端周期性信号（单音
%或元音）的阈值被以经验导出。
AEC-BASELINE-THRESH=5.5;

%为状态尺度预先计算用作乘法器的常量。
AEC-STATUS-GAUGE-SCALER=floor（ 32768/..... （ AEC-MAX-GAIN-THRESH — AEC-BASELINE-THRESH））/32768;

%产生 FIR 系数的增益轮廓。该轮廓大致与汽车内系数的希望范围匹配。这样，包含周期
%性成分（元音）的更新被强迫服从合适的指数衰落特性和使发散最小。在较高延迟插头
%上的较低增益也减少了更新噪声对系数的影响。高轮廓的总效果是允许较高更新增益而
%没有不稳定。
%由于所有系数具有 16 位动态范围，较高延迟插头也具有作为该轮廓结果的较好量化。该
%轮廓通过在右移位中间时刻计算 FIR64 个插头实现。

```
profile=ones(AEC_COEF_TAPS,1);
for k=2:(AEC_COEF_TAPS/64),
  profile((k*64-63):(k*64))=ones(64,1)*2^(1-k);
end

% 分配调试向量以加速执行.
aecUpdateFactor = zeros(1,NFRAMES);
aecChanGainHist = zeros(1,NFRAMES);
aecSpeedHist = zeros(1,NFRAMES);
aecVoiceGainHist = zeros(1,NFRAMES);
aecVoiceGainBaseHist = zeros(1,NFRAMES);
aecNearRatioHist = zeros(1,NFRAMES);
aecNearGainHist = zeros(1,NFRAMES);
aecEchoGainHist = zeros(1,NFRAMES);
aecInNoiseHist = zeros(1,NFRAMES);
aecInEchoNoiseHist = zeros(1,NFRAMES);
aecInVoiceHist = zeros(1,NFRAMES);
aecInEchoVoiceHist = zeros(1,NFRAMES);
anlpInVoiceHist = zeros(1,NFRAMES);
anlpInNoiseHist = zeros(1,NFRAMES);
anlpDeltaHist = zeros(1, NSAMPLES);
anlpGainHist = zeros(1,NFRAMES);

% 初始化变量
aecRef = zeros(1,AEC_REF_TAPS);                                  % 使用最新值:
aecCoef = zeros(AEC_COEF_TAPS,1);                                % 注释
aecInNoise = FRAME_SIZE;                                         % 注释
aecInEchoNoise = aecInNoise;                                     % 注释
anlpInNoise = aecInNoise;                                        % 注释
aecChanGain = ONE;                                               % 注释
aecVoiceGain = ONE;                                              % 注释
aecVoiceGainBase = aecVoiceGain;                                 % 注释
aecNearGain = aecVoiceGain;                                      % 注释
aecEchoGain = aecVoiceGain;                                      % 注释
anlpComfortNoiseInOld = 0;                                       % 注释
anlpComfortNoiseOutOld = 0;                                      % 注释
anlpComfortNoise = zeros(1,COMFORT_NOISE_SIZE);                  % 注释
anlpArCoef = 0.75;                                               % 注释
aecNearRatio = 0;                          % 只初始化记录
aecInNoiseStateVars     = [aecInNoise 0 0];
aecInEchoNoiseStateVars = [aecInEchoNoise 0 0];
anlpInNoiseStateVars    = [anlpInNoise 0 0];
anlpSeed = 1;
anlpArGain = 1 - anlpArCoef;
anlpRefEnvelope = 0;
anlpOutLast = 0;
anlpNearSpeechCount = 0;
```

```
anlpNearSpeechFlag = 0;

%显示(['aecCoef(1) = ' dec2hex(aecCoef(1)*32768+(aecCoef(1)<0)*65536)])
%显示(['aecCoef(2) = ' dec2hex(aecCoef(2)*32768+(aecCoef(2)<0)*65536)])
%显示(['aecInNoise = ' dec2hex(aecInNoise*2^31)])
%显示(['anlpInNoise = ' dec2hex(anlpInNoise*2^31)])
%显示(['aecChanGain = ' dec2hex(aecChanGain*32768)])
%显示(['aecVoiceGain = ' dec2hex(aecVoiceGain*32768)])
%显示(['aecVoiceGainBase = ' dec2hex(aecVoiceGainBase*32768)])
%显示(['aecEchoGain = ' dec2hex(aecEchoGain*32768)])
%显示(['anlpComfortNoiseInOld = ' dec2hex(anlpComfortNoiseInOld*32768)])
%显示(['anlpComfortNoiseOutOld = ' dec2hex(anlpComfortNoiseOutOld*32768)])
%显示(['anlpArCoef = ' dec2hex(anlpArCoef*2^31)])
%显示(['anlpArGain = ' dec2hex(anlpArGain*32768)])

fidOut = fopen(outFile, 'w');

for frame = 1:NFRAMES,

frame                                    %显示帧数量以指示程序。
%AEC  帧前部分
%
%由于在用于更新 FIR 系数的基准向量插头与用于近端能量相关的插头之间有一个间隙，
%不需要对基准向量的每个插头计算更新向量。因此，更新向量由特别用于两个目的的子
%向量代表。
%清除在一帧上积累的更新子向量。
aecUpdate=zeros(AEC-COEF-TAPS, 1);          %用于 FIR 系数更新
aecUpdateNear=zeros(AEC-NEAR-TAPS, 1);      %用于近端计量


%清除其它帧积累
aecEchoEstEnergy=0;
%重新设置块浮点变量。
aecShiftPending=0;
aecErrorShift=0;

%使上行和下行链路 PCM 音频样本进入缓存器。
downlinkAudio=LRmatrix(1, (frame-1)*FRAME-
```

```
SIZE+1:frame*FRAME_SIZE);
uplinkAudio=LRmatrix(2,(frame-1)*FRAME-
SIZE+1:frame*FRAME_SIZE);

%在一个帧上积累 AEC 近端输入能量
aecInEnergy=sum(uplinkAudio.^2);
%量化能量为 32 位。
aecInEnergy=floor(aecInEnergy*ENERGY-SCALE)/ENERGY-SCALE;

%AEC 样本部分
for k=1:FRAME-SIZE,
%将远端（受话器）样本移位到基准延迟线和计算 FIR 输出。
%在 DSP 中，两个操作在一个指令中。
aecRef=[downlinkAudio(k) aecRef(1:AEC-REF-TAPS-1)];
%测试码开始
%下列码快速逼近注释，比特-精度码。
aecEchoEst=aecRef(1:AEC-COEF-TAPS)*(aecCoef.*profile);
aecEchoEst=max(min(round(aecEchoEst*32768)/32768,ONE),-1);
%测试码结束
aecEchoEst=0;
for m=(AEC-EOEF-TAPS/64):-1:2;
aecEchoEst=aecEchoEst+aecRef(m*64-63:m*64)*aecCoef(m*64-
63:m*64);
aecEchoEst=max(min(aecEchoEst,ONE),-1)2;
%量化 S.15 格式
aecEchoEst=floor(aecEchoEst*2^31)/2^31;
结束
aecEchoEst=aecEchoEst+aecRef(1:64)*aecCoef(m*64);
aecEchoEst=max(min(aecEchoEst,ONE),-1);
%量化 S.15 格式
%当将具有恰好 0.5 的分数的负数舍入时加入 2^(-17)强制 1 补数浮
点起与 2 补数的作用。
```

```
aecEchoEst=round(aecEchoEst*32768+2^(-17))/32768;
aecEchoEst=max(min(aecEchoEst,ONE),-1);
%在一个帧上积累回声估计能量。
%为改善小信号性能和进行相同方式的计量作为其它能量积累，在环
路通路之间保留了整
%40 位累加器。
aecEchoEstEnergy=aecEchoEstEnergy+aecEchoEst^2;

%计算 AEC output=near-end(microphohe)input-回声估计值。
uplinkAudio(k)=max(min(uplinkAudio(k)-aecEchoEst,ONE),-1);
%累计系数更新=误差相关（uplinkAudio（k））和基准。使用块浮点
表达式，其中 aecErrorShift
%是指数而 aecUpdate/aecUpdateNear（）是尾数。
T=uplinkAudio(k)*2^aecErrorShift;
%量化 S.15 格式 T = floor（T*32768）/32768;
if  aecShiftPending,
ASM= -1
aecErrorShift=aecErrorShift -1;
aecShiftPending=0;
else
ASM=0;
end
%计算用于更新 FIR 系数的区间。
aecUpdate = aecUpdate + T*aecRef(1:AEC-COEF-TAPS)';
%计算用于计量近端能量的区间。
aecUpdateNear=aecUpdateNear + .....
        T * aecRef(AEC-REF-TAPS  -  AEC-NEAR-TAPS + 1:AEC-
REF-TAPS)';
%量化 S.15 格式
%当将具有恰好为 0.5 的分数的负数舍入时加入 2^(-17)强制 1 补数
浮点起 2 补数作用。
```

```
aecUpdate     = round(aecUpdate     * 32768 + 2^(-17))/32768;
aecUpdateNear = round(aecUpdateNear * 32768 + 2^(-17))/32768;
aecUpdate     = max(min(aecUpdate    ,ONE),-1);
aecUpdateNear = max(min(aecUpdateNear,ONE),-1);
aecUpdate     = aecUpdate     * 2^ASM;
aecUpdateNear = aecUpdateNear * 2^ASM;
```

```
%在可能的右位移之后量化 S.15 格式。
aecUpdate=floor(aecUpdate*32768)/32768;
aecUpdateNear=floor(aecUpdateNear*32768)/32768;
%找到更新向量的峰值平方（假设在前面 128 个插头中）。
%目的是获得峰值的绝对值，但是在 DSP 中峰值平方占较少周期，甚至具有帧结尾的平方
%根。
aecUpdatePeak2=max(aecUpdate(1:128).^2);
%标志指示更新是否需要在下个循环中除以 2。
aecShiftPending=aecUpdatePeak2>0.25;        %0.25=0.5^2
end
%量化能量为 32 位。
aecEchoEstEnergy=floor(aecEchoEstEnergy*ENERGY-
SCALE)/ENERGY-SCALE;
%
%AEC 帧后的部分
%
aecOut=uplinkAudio;                    %将输出保留在文件用于调试.
%在一个帧上积累 AEC 输出能量。
aecOutEnergy=sum(uplinkAudio.^2);
%量化能量为 32 位。
aecOutEnergy=floor(aecOutEnergy*ENERGY-SCALE)/ENERGY-
SCALE;
%对于更新向量每个元素实际基准能量是不同的。aecUpdate（1）将使用来自 aecRef（1）
%的能量，aecUpdate（2）将使用来自 aecRef（2）的能量，等等。为减少复杂性，使用单
%一数字表示基准能量。
```

```
%当基准能量被用于计量信道回声增益以确定适应速度或标准化更新（NLMS），使用太小
%值将不会引起不稳定性。在此实现的折中方案使用端点的最大值，其中 profile=1.
aecRefEnergy=max(sum(aecRef(1:FRAME-SIZE).^2),.....sum(aecRef(64:FRAME-SIZE+63).^2));
%量化 Z8.7 格式
aecRefEnergy=floor(aecRefEnergy*128)/128;
%计量更新基线作为高延迟单元的 RMS，其中误差与远端信号之间的相关性希望为 0.
%在最低有效位中加 1 以保证量化后 aecUpdateBase 较大和 aecPeakToBase 较小。这避免了
%对较高回声相关性的错误印象。
%在底层操作产生与顶层相同的结果后在最低有效位中加入 1，除了所有截短位等于零的
%少有情况。
aecUpdateBase=sum(aecUpdteNear.^2)/AEC-NEAR-TAPS;
%对 Z.31 格式量化平方的中间结果
aecUpdateBase = floor(aecUpdateBase*2^31)/2^31;
aecUpdateBase = sqrt(aecUpdateBase);

%量化 Z.15 格式
aecUpdateBase=floor(aecUpdateBase*32768+1)/32768;
%查找更新向量的峰值幅度。
aecupdatePeak=sqrt(aecUpdatePeak2);
%计算更新峰值/基线比。
aecPeakToBase=aecUpdatePeak/aecUpdateBase;
%从 4 分数为量化 Z11.4 格式足够了。
aecPeakToBase=floor(aecPeakToBase*16)/16;
%根据更新峰值基线比计算状态尺度（范围=[0, 1]）。即使峰值/基线比随帧尺寸和基线阈
%值的改变可能改变，用于下行数据流处理的该尺度保持相同。
%对于近端话音/噪声或远端周期信号（元音），尺度<0.1
```

%例如，对于具有频率 941Hz 和 1209Hz 的 DTMF 单音的第一帧被发现产生%aecPeakToBase=3.5，因此 aecStatusGauge=0.

%对于双方谈话，尺度<0.3

%对于远端单方谈话：

%不论噪声如何当消除器不收敛时，尺度=1

%如果近端安静，尺度=1 直到接近完全收敛.

%由于消除器收敛，只有能量比近端噪声高电平的残余回声使尺度=1.

%因此，近端噪声很少引起更新，不减少更新增益.

aecStatusGauge=(aecPeakToBase-AEC-BASELINE-THRESH)* … AEC-STATUS-GAUGE-SCALER;

aecStatusGauge=max(min(aecStatusGauge,ONE),0);

%对 Z.15 格式量化

```
aecStatusGauge = floor(aecStatusGauge*32768)/32768

% 估计在AEC输入处的噪声帧能量.
[confirmedNoVoiceFlag, aecInNoiseStateVars] = estnoise(aecInEnergy, ...
  (micVad(frame)==0) & (aecStatusGauge==0), aecInNoiseStateVars);
aecInNoise = aecInNoiseStateVars(1);
aecInNoise                                    % 显示状态
aecInNoiseHist(frame) = aecInNoise;           % 保留用于调试
```

%估计在 AEC 输入处的回声增益(信道增益).

%只在远端信号语音期间更新估计值(speakerVad(frame)==1)&(aecStatusGauge>0.3)，当该比值正确时不管基准计量包括话音和噪声同时 AEC 输入计量包括仅话音(aecInVoice>aecInNoise*8)，和当基准能量不受量化的明显影响(aecRefEnergy>=10/128).瞬时增益计量不非常准确，因为时间为校准和频谱改变。

因此，使用一个平均处理.回声消除器滤波器系数的标准是信道增益的非常好的长期估计值.可是，它不能足够快地跟踪正改变的回声路径.

由于信道增益估计值也控制线性回声消除器的自适应速度，如果估计值尽可能与线性回声消除器无关可以改善稳定性.因此，使用下列方案：

如果瞬时计量(aecDhanGainTrial)少于估计值(aecGhanGain)，该估

计值呈指数下降.

为加速自适应,估计值被直接设置为瞬时计量,如果误差大于 50%,和估计值呈指数下降如果误差小于 12.5%, 即 (aecChanGain-aecGhanGainTrial)/4>aecChanGain/32 aecChanGain-aecChanGainTrial>aecChanGain/8=aecChanGain*0.125

如果瞬时计量大于估计值,如果近端话音在计量中不占明显优势(aecChanGainTrial<2), 估计值呈指数上升.使用相对比较例如(aecChanGainTrial<2*aecChanGain)将无效,因为当实际信道增益快速跳动时没有更新出现.使用下降时间常数 1/8 的上升时间常数有助于在瞬时计量中容忍近端话音.很难区别近端语音和当信道回声增益变坏的情况.上升时间常数在此确定了跟踪速率,它按照经验被确定.

```
aecInVoice = max(0, aecInEnergy-aecInNoise);
aecInVoiceHist(frame) = aecInVoice;
if (speakerVad(frame) == 1) & (aecStatusGauge > 0.3) & ...
   (aecInVoice > aecInNoise*8) & (aecRefEnergy >= 10/128),
  % 量化aecInVoice为4Z8.20格式,以用作被除数获得所需要的比例
  % desired scale for the quotient.
  aecChanGainTrial = floor(aecInVoice *2^20)/2^20 / aecRefEnergy;
  % 量化和限制系数为Z2.13格式
  aecChanGainTrial = min(4*ONE, floor(aecChanGainTrial *2^13)/2^13);
  % 计算系数的平方根
  aecChanGainTrial = sqrt(aecChanGainTrial);
  % 量化根为Z1.14格式
  aecChanGainTrial = floor(aecChanGainTrial *2^14)/2^14;
  if (aecChanGainTrial < aecChanGain/2),
    aecChanGain = aecChanGainTrial;
  elseif (aecChanGain > aecChanGainTrial),
    aecChanGain = aecChanGain - ...
                  max((aecChanGain-aecChanGainTrial)/4, aecChanGain/32);
  elseif (aecChanGainTrial < 2*ONE),
    aecChanGain = min(ONE, aecChanGain + max(aecChanGain/256, 2^(-11)));
  end
  % 量化为Z.15格式
  aecChanGain = floor(aecChanGain*32768)/32768;
  % 保留调整
  aecChanGainHist(frame) = aecChanGainTrial;
else
  aecChanGainHist(frame) = ONE;
end
aecChanGain                                    % 显示状态
aecSpeedHist(frame) = aecChanGain;             % 保留调整
```

%再次确定更新

%只要增益小于或等于 1 使用 NLMS 产生自适应速率常数(与远端信号

幅度无关）

%使用 max 运算产生更快收敛，然后在分母上加 1，因为产生的增益较高。

%使用 AEC 基准和近端输入能量的最大值限制标准化增益，当有近端噪声和/或话音时．不使用 AEC 输出能量而使用 AEC 近端输入能量，因为在收敛后不需要高增益，和因为稳定性不是通过使用基于 AEC 输出的另一个参数控制 AEC 更新增益来改善。aecInEnergy 将小于没有近端话音或噪声的 aecRefEnergy，以便避免送话器过载（由于送话器增益是按照最大近端语音电平设置的）。如果保证远端信号语音具有低近端噪声（aecInEnergy/16>=aecOutEnergy）能量乘数被设置为 8．反之，被设置为 16．因此，当汽车安静时自适应更快。太小的能量乘数引起较小稳定性，在自适应中产生过大行动在回声消除器输出中产生尖峰信号。过大行动也阻碍了具有正改变回声路径的远端信号语音与近端语音之间的区别。太大乘数在混乱后立即增加了回声增益。在分子中，使用信道回声增益作为乘数优化信道自适应速率。没有该乘数，自适应或对于高信道增益慢于需要或对于低信道增益不稳定。。使用自适应滤波器系数标准而非基于信道增益的能量产生长期更精确和连续的估计值，但是在回声路径改变的短期内速率和稳定性必须折中。

```
if (aecInEnergy/16>=aecOutEnergy),
aecDenom=max(1,8*max(aecRefEnergy,aecInEnergy));
%为调整，设置因子等于负尺度值
aecUpdatePactor(frame)= - aecStatusGauge;
else
aecDenom=max(1,16*max(aecRefEnergy,aecInEnergy));
%为调整，设置因子等于尺度值
aecUpdateFactor(frame)=aecS tatuGauge;
end
%量化 Z12.3 格式.
aecDenom=floor(8*aecDenom)/8
aecNumer=aecChanGain*aecStatusGauge;
%量化 22Z8.3 格式。
```

```
aecNumer=floor(aecNumer*2^18)/2^18
aecUpdateGain=min(aecNumer/aecDenom, ONE);
%量化 Z.15 格式
aecUpdateGain=floor(aecUpdateGain*32768)/32768;
%使用自适应增益使更新向量与系数向量相加。
%在用作 FIR 系数使用之前 aecCoef 被乘以包络。
aecCoef=aecCoef + (aecUpdate*2^(-
aecErrorShift)*aecUpdateGain);
%量化 S.15 格式
%加 2^(-17)以强制 1 补数浮点起 2 补数浮点的作用，当舍入具有恰好
0.5 的分数的负数时。
    aecCoef = round(aecCoef * 32768 + 2^(-17))/32768;
    aecCoef = max(min(aecCoef,ONE),-1);

%噪声抑制器将在此处理 uplinkAudio。
%通过只在一个子程序中使用 aecUpdate 和 aecUpdateNear，它们可
以在临时存储器中，可%重叠使用。
%
%ANLP 帧前部分
%
%在 ANLP 输入处积累能量，该处通过噪声抑制器被连接到 AEC 输出。
anlpInEnergy=sum(uplinkAudio.2^);
%量化能量为 32 位。
anlpInEnergy=floor(anlpInEnery*ENERGY-SCALE)/ENERGY-SCALE;
%在 ANLP 输入处估计噪声帧能量。
%由于回声估计值来自受话器信号而使用 speakerVad。
[confirmedNoVoiceFlag, anlpInNoiseStateVars] = estnoise(anlpInEnergy, ...
  (micVad(frame)==0) & (aecStatusGauge==0), anlpInNoiseStateVars);
anlpInNoise = anlpInNoiseStateVars(1);
anlpInNoiseHist(frame) = anlpInNoise;          % 保留调整

%当确认没有话音时计算舒适噪声。
if confirmedNoVoiceFlag,
anlpComfortNoise=uplinkAudio(1:COMFORT-NOISE-SIZE);
%在一阶 ARMA 噪声模型形式中使用 NLMS 算法估计 anlpArCoef:
```

```
%   (1-anlpArCoef) *(1-0.8125*z^-1)/(1-anlpArCoef*z^-1)
%该 ARMA 模型将用于滤波白噪声以获得听起来象汽车噪声的噪声。
%NLMS 算法努力使下列表达式最小:
%   error=uplinkAudio*(1-anlpArCoef*z^-1)/(1+0.8125*z^-1)
%误差与-1 之间的相关性是对系数估计值的更新。

%anlpInEnergy 用于标准化更新增益以提供与电平无关的自适应速率。计算环路外的非零分%母以节省 MIPS。
anlpDenom = 4 * max(1/128, floor(anlpInEnergy*128)/128);
error = 0;
for i=2:FRAME_SIZE,
  error = - 0.8125 * error;
  % 量化5S4.31格式 (10*uplinkAudio的最大值)
  error = floor(error * 2^31)/2^31;
  error = error + uplinkAudio(i-1:i) * [-anlpArCoef; 1];
  quotient = uplinkAudio(i-1:i-1) * floor(error * 2^12)/2^12 / anlpDenom;
  quotient = max(min(quotient,ONE),-1);
  % 量化S.15格式
  quotient = floor(quotient * 32768)/32768;


    anlpArCoef = anlpArCoef + quotient;
    anlpArCoef = max(min(anlpArCoef,ONE),-1);
  end
  anlpArGain = 1 - anlpArCoef;
  % 标绘舒适噪声响应频率用于调整.
%   plot((0:8191)/8192*4000,...
%        20*log10(abs(freqz(anlpArGain*[1 0.8125],[1 -anlpArCoef],8192))))
%   axis([0 4000 -40 10])
%   pause(1)
end

% 在AEC输入处估计回声的噪声帧能量.
% 由于回声估计值来自受话器信号使用speakerVad.
[confirmedNoVoiceFlag, aecInEchoNoiseStateVars] = estnoise(...
  aecEchoEstEnergy, (speakerVad(frame)==0) & (aecStatusGauge==0), ...
  aecInEchoNoiseStateVars);
aecInEchoNoise = aecInEchoNoiseStateVars(1);
aecInEchoNoiseHist(frame) = aecInEchoNoise;                    % 保留调整
test1Hist(frame) = aecInEchoNoiseStateVars(2);                 % 保留调整
test2Hist(frame) = aecInEchoNoiseStateVars(3);                 % 保留调整
aecInEchoNoiseStateVars                                        % 显示状态

% 估计话音能量估计值. 不使它们低于零.
aecInEchoVoice = max(0, aecEchoEstEnergy - aecInEchoNoise);
aecInEchoVoiceHist(frame) = aecInEchoVoice;
aecNearVoice = max(0, aecInVoice - aecInEchoVoice);
anlpInVoice = max(0, anlpInEnergy - anlpInNoise);
anlpInVoiceHist(frame) = anlpInVoice;
```

%向上渗漏线性回声消除器基线回声增益估计值。
%将其用于与瞬时回声增益比较以检测近端语音和近端语音期间的ANLP增益。
%大概内瞬时回声增益变差时它快得足以跟踪。该渗漏作为快速跟踪避免在具有变化回声路径远端单方谈话期间近端语音检测失败和在远端话音/噪声期间近端话音/噪声失真最小之间的折中。当远端单方谈话在回声路径已经改变后开始时，aecNearGain可能下降，上升一些，然后下降。如果aecNearGain上升明显高于aecVoiceGainBase，将检测近端语音。如果这些发生，在aecNearGain上的渗漏可以增加防止此事发生。经验性地找出5/4096的渗漏常数提供足够的跟踪速度。可是，增加渗漏减少了在双方谈话期间近端语音的检测。
aecVoicegainBase=min(aecVoicegainBase + 1/8192,ONE);
%再次计算线性回声消除器话音，排除该信道。
%涉及的信道增益将使近端语音检测不可靠。
%计算线性回声消除器输入和输出的话音能量比的根。
%原始计量不以话音是否来自近端，远端或两者为条件。
%在ANLP输入处提取线性回声消除器输出，因为该信号已经通过了噪声抑制器，使计量更精确。靠对能量的计量包括回声中受话器的回声失真。比较回声消除器输入和输出的能量也避免当比较回声消除器输出与基准时出现的时间排队（由于信道延迟）。
%只有当不被量化破坏（aecInVoice>8/2^7），当出现话音（aecInVoice>aecInNoise*8），和当残留回声可计量（anlpInVoice>anlpInNoise/2）时，计量增益。
%远端信号中的周期性成分不减少计量的精确度。
if
(aecInVoice>max(8/2^7,aecInNoise*8))&...(anlpInVoice>anlpInNoise/2),
aecVoiceGain=min(ONE,sqrt(anlpInVoice/(follr(aecInVoice*2^7)/2^7)));
%量化Z.15格式
aecVoicegain=floor (aecVoicegain*32768)/32768;
%对总话音计量近端话音比。

%这产生很少的由于改变具有远端单方谈话的回声路径引起的近端话音失误指示，因为它只取决于回声估计值的能量，而不依赖在输出处回声被消除得如何好。可是，就象 aecVoiceGain，当消除器是总的未训练时近端话音失误指示也是可能的。

由于话音计量是能量单位常用比值包括一个平方根。可是，通过直接使用能量比值可以获得近端话音与不良消除器训练之间更大的差别。这也节省 MIPS。

```
aecNearRatio=min(ONE,aecfNearVoice/(floor(aecInvoice*2^7)/
2^7));
%量化 Z.15 格式
aecNearRatio=floor(aecfNearratio*32768)/32768;
```

%在远端语音期间计量线性回声消除器基线回声增益估计值。无论何时存在远端单方谈话都计量（aecStatusGauge>0.1）以保证参数跟踪真实回声增益（不是噪声）。反之，可能听到回声。当话音增益低于基线估计值更新基线回声增益估计值，因为这指示远端单方谈话的高可能性。通过使用当它高于基线时通过使用渗漏和不使基线跟踪话音，基线对噪声和近端语音非常牢固。

```
if (aecStatusGauge>0.1),
aecVoiceGainBase=min(aecVoicegainBase,aecvoiceGain);
end
```

%在近端语音期间计量线性回声消除器话音增益，和在远端单方谈话期间（采用话音增益的最小值）重新设置。这将用于对话音增益基线的比较以检测双方谈话。近端语音可能出现的唯一可使用指示符是回声消除器尺度低（aecStatusgauge<0.3）和回声消除器输出包含足够能量，该能量可能不只是残余回声（anlpInVoice>anlpInNoise*8）。记住由于远端语音中的周期性成分该尺度低。因此，需要额外装置区别因回声路径改变或受话器失真话音增益突然变坏的情况和有近端语音的情况。当在远端单方谈话期间话音增益突然变坏时，该持续时间往往很短。因此，滤除在该参数中的增量通常拒绝了远端单方谈话事件。滤波器的时间常数是远端单方谈话拒绝与双方谈话确认速度之间的折中，它被以经验确定。

```
aecNearGainLast = aecNearGain;
aecNearGain = min(aecNearGain, aecVoiceGain);
if (aecStatusGauge < 0.3) & (anlpInVoice > anlpInNoise*8),
  aecNearGain = aecNearGain + (aecVoiceGain-aecNearGain)/8;
  % 量化为Z.15格式
  aecNearGain = floor(aecNearGain*32768)/32768;
end
```

%使用近端语音三个检测方法的任何一个。每个方法在某些条件下对近端语音更敏感。只有当总话音比绝对电平或拒绝只出现噪声的条件的噪声平台足够大时才出现所有方法。

%方法 1

%(aecInechoVoice<aecInVoice/4)等效于(aecNearRatio>=0.75),但是前者对量化不敏感。在 aecNearRatio 上的尖峰信号对于远端单方谈话低电平是常有的。因此,预置必须高以避免错误检测。对该检测方法需要的唯一限制是状态尺度=0 和话音增益基线低于 0.4625=15155/32768.后者阈值是在当消除器基本为训练时避免错误检测与消除器开始训练后尽快检测双方谈话之间的经验折中。

%方法 2

%(aecInVoice>32/2^7&anlpInVoice)拒绝低电平话音。这样做,检测对近端话音与总话音的低比率更敏感。再次,仅当状态尺度=0 有助于拒绝远端单方谈话时接受该条件。通过随 aecVoiceGainBase 改变阈值优化近端语音的敏感度。比例因子和偏移是当消除器基本未训练时避免错误检测与消除器开始训练后对近端语音尽可能敏感之间的经验折中。

%方法 3

%当回声抑制突然和连续变坏时比较 aecNearGain 与 aecVoiceGainBase 检测近端语音.这是因为 aecVoiceGainBase 不反映突然的变化,而 aecNearGain 使用滤波器保证检测中的连续性.因为滤波器被用于计算 aecNearGain,条件更放松,其允许对近端语音更大的敏感度。这种条件包括使状态尺度<0.3 而非=0,和没有对话音增加最小绝对值电平。当回声路径已经改变而远端单方谈话开始时,aecNearGain 可能下降但仍然在短时间保持高于 aecVoiceGainBase。为避免这种情况下的音频回声,只有当 aecNearGain 大于其最后值时检测近端语音。

```
if ((aecStatusGauge == 0) & (aecInEchoVoice < aecInVoice/4) & ...
    (aecVoiceGainBase < 15155/32768)) | ...
   ((aecStatusGauge == 0) & (aecInVoice > 32/2^7) & ...
    (anlpInVoice > anlpInNoise*8) & ...
    (aecNearRatio - aecVoiceGainBase*1.5 >= 9830/32768)) | ...
   ((aecNearGain - aecVoiceGainBase >= 6554/32768) & ...
    (aecNearGain > aecNearGainLast)),
```

%近端语音已经检测。

%如果该帧开始了近端语音的新周期（暂停时间已经过去而包含话音的最新帧只是回声），则设置线性回声消除器回声增益等于基线回声增益估计值。这只在近端语音周期开头处进行，以便在近端语音期间残余回声抑制器具有连续衰减而非因为基线向上渗漏而增加失真，同时不进行更新。

```
if (anlpNearSpeechCount==0)&( anlpNearSpeechflag==0),
aecEchoGain=aecVoiceGainBase;
end
```

%设置表示包含话音的最后一帧的标志。在不检测话音帧期间将保留设置。

```
anlpNearSpeechflag=1;
```

%重新开始近端语音暂停时间计数器。当非零时，在双方谈话期间或当话音能量低得不能检测时，将越过 anlpNearSpeechflag 由近端语音残余回声抑制器使失真最小。

%如果仅当远端单方谈话被检测时该计数器开始，在双方谈话期间将有更少的近端语音中断。可是，近端人员说话后远端人员每次开始说话将听到回声，即使在长中止后。

%当有从近端语音到远端单方谈话的快速转变时（例如当近端人员在远端人员语音中间谈话时），暂停时间将产生远端人员听到回声（在-25dB）的短时间段。这种人工产物是值得的，因为暂停时间获得的双方谈话期间的中断明显减少。

```
anlpNearSpeechCount=25;          %  25*20ms=500ms 暂停时间
end
end
```

%如果回声估计话音能量至少是近端话音能量的 15/16，假设该帧只包含远端回声语音，并清除近端语音标志。如果近端语音暂停时间计数

器已经超时，残余回声抑制器马上到远端单方谈话模式。反之，当近端语音暂停时间计数器超时时，残余回声抑制器马上到远端单方谈话模式。注意当该计数器正工作时可以再次检测近端语音，然后一旦计数器超时残余回声抑制器保持近端语音模式。

%该检测方案比较 32 位数字，而不使用基于能量电平、状态尺度或根据前面帧的计量的任何限制（除了噪声估计）。因此，该方案相当敏感而牢固。如果该方案检测远端语音失败，当近端人员说话后远端人员开始说话时，将听到回声，即使长中断后。

```
if (aecInEchoVoice> (aecInVoice - aecInVoice/16)),
anlpNearSpeechFlag=0;
end
```

%如果近端语音暂停时间计数器已经超时，和包含话音的最新帧只是回声，则设置线性回声消除器的回声增益再次等于最新计量出的线性回声消除器话音增益。假设没有出现近端语音，所以残余回声抑制器试图抑制所有话音。这种实验在清除 anlpNearSpeechFlag 之后，以便 aecEchoGain 马上反映该判决。

```
if (anlpNearSpeechCount == 0) & (anlpNearSpeechFlag == 0),
  aecEchoGain = aecVoiceGain;
end

aecVoiceGainHist(frame) = aecVoiceGain;              % 保留调整
aecVoiceGainBase                                     % 回声状态
aecVoiceGainBaseHist(frame) = aecVoiceGainBase;      % 保留调整
aecEchoGain                                          % 回声状态
aecEchoGainHist(frame) = aecEchoGain;                % 保留调整
aecNearRatioHist(frame) = aecNearRatio;              % 保留调整
aecNearGainHist(frame) = aecNearGain;                % 保留调整
anlpNearSpeechCount                                  % 回声状态
anlpNearSpeechFlag                                   % 回声状态
```

%计算环路回声增益直到 ANLP。当需要满足对该系统的总环路回声抑制目标时 ANLP 将衰减。

```
%? 贯穿音量控制。
%aecChangain 是从受话器到送话器          -- 信道。
%aecEchoGain 是从线性回声消除器的输入到输出。
aecLoopEchoGain= aecChangain* aecEchoGain;          %在此
插入音量增益。
%量化为 Z.15 格式
```

```
aecLoopEchoGain = floor(aecLoopEchoGain*32768)/32768
```

%设置 ANLP 窗口以捕获所希望的残余回声，但是不再有了。这使近端话音和噪声最小。由于 AEC 改善了其回声增益 ANLP 窗口尺寸缩小。音量控制增益不再使用，因为 ANLP 的检测器包络输入在音量控制之后到达。回声增益估计 RMS 方面的计量，但 ANLP 需要抑制包括峰值的全部残余回声。因此，使用峰值/RMS 因子乘数（=3）。anlpWindowGain 的动态范围是二，因为当线性回声消除器基本未训练时 anlpWindowGain 需要至少是二以捕获在该窗口内的回声。

```
anlpWindowGain=min(2*ONE, 3* aecChangain* aecEchoGain);
%量化为 Z1.14 格式
anlpWindowGain=floor(anlpWindowGain*16384)/16384;
```

%如果 VAD 指示在受话器上只有噪声，则以通过模式输入 ANLP（增益=1）。该 VAD 允许高质量的近端单方谈话，因为没有近端语音失真或噪声。当只有噪声时 VAD 经常告知有话音，所以需要其它装置使在此情况下的近端语音失真和噪声最小。当由低话音电平时 VAD 也经常告知只有噪声。此问题与受话器上的噪声电平成某种比例。可是，在环路中具有线性回声消除器提供了足够的回声抑制，使这些情况下的回声不发声。

```
if speakerVad(frame)==0,                    %近端单方谈话条件
anlpEchoGain=ONE;
```

%由于 speakerVad=1，受话器最可能具有语音，但不一定。如果近端语音被检测，假设双方谈话。在此情况下，设置总环路回声抑制目标为-25dB（1843/32768），和设置 ANLP 增益为在环路其余部分中不提供的所需要的回声抑制。ANLP 增益越高（抑制越低）和当用户将音量从最大刻度下调时失真被减少。当训练线性回声消除器时，ANLP 增益在此模式中一般高于-10dB，所以噪声掩盖没有改善声音质量。远端用户在双方谈话期间将听到一些回声，但是这种人造产物比中断或近端话音的高失真更可取。

```
elseif     (anlpNearSpeechCount>0)|(anlpNearSpeechFlag==1),
%双方谈话
anlpEchoGain=1843/32768/aecLoopEchoGain;
```

%由于 seakerVad=1 和近端语音没有检测，假设有远端单方谈话。总环路回声抑制目标是-56dB（52/32768）以便当两端具有安静背景时回声几乎不发声。如同在双方谈话模式，ANLP 增益被设置为所需要的环路其余部分不提供的回声抑制，和当用户将音量从最大刻度下调时和当线性回声消除器训练时 ANLP 增益更高和减少失真。可是，衰减到回声不发声而低于噪声平台所需要的增益可能更高，所以两个增益的较高者用于 ANLP 使失真最小。没有舒适噪声，用相同因子衰减回声和噪声两者将不改变信噪比；所以噪声掩盖不起作用。舒适噪声使 ANLP 输入和输出上的噪声为相同电平。因此，ANLP 可以回声衰减到低于噪声平台的可发声阈值，而没有不必要的失真。因为这些变量是能量单位的，采用噪声/话音比的平方根。将所需要的回声/噪声定量乘以实际噪声/回声比值使噪声析出。剩下的是所需要的实际回声比值，它是掩盖回声所需要的增益。

```
else                                        %远端单方谈话条件
ECHO-TO-NOISE-GOAL=1/8;                     %-18dB
if (anlpInVioce>anlpInNoise),
anlpEchoGin=max(52/32768/aecLoopEchoGain,...
      ECHO-TO-NOISE-GOAL*sqrt(anlpInNoise/ anlpInVioce));
else
%在嘈杂条件下，该增益不超过 ECHO-TO-NOISE-GOAL，尽管理论上它可以用于低能量话音，因为当噪声电平快速下降时噪声估计太大（汽车减速）。这使当 ANLP 回声增益太高时将听到回声。
anlpEchoGain=max(52/32768/aecLoopEchoGain,  ECHO-TO-NOISE-GOAL);
end
end
anlpEchoGain= min(ONE, anlpEchogain);
%量化为 Z.15 格式
anlpEchoGain=floor(anlpEchoGain*32768)32768
anlpGainHist(frame)=anlpEchoGain;           %保留调整
%递减近端语音暂停计数器，如果需要，以便独立于受话器 VAD 工作.
anlpNearSpeechCount=max(0, anlpNearSpeechCount - 1);
```

```
%
%ANLP 样本部分
%
%如果 ANLP 回声增益是 1，ANLP 不激活—跳过以节省 MIPS。
if (anlpEchoGain==ONE),
%保持包络检测器运行。
for k=1:FRAME-SIZE,
anlpRef=aecRef(FRAME-SIZE + 1);
anlpRefEnvelope=max(abs(anlpRef),255/256*
anlpRefEnvelope);
anlpRefEnvelope=floor(anlpRefEnvelope*2^31)/2^31;
end
%将用于 AC 中心衰减器的变量更新成为与处理整个帧产生的变量相
同。
anlpOutLast=uplinkAudio(FRAME-SIZE);
else
for k=1:FRAME-SIZE,
%ANLP 远端基准=AEC 基准。使用 aecEchoEst 替代或额外使用不给出
更好的结果，因为 aecEchoEst 与残余回声很不相关。由于优先于包
含在 anlIn 中最早的回声，AEC 基准同样有效。在峰值检测器中的长
时间常数是 ANLP 的关键，而它使 anlpRef 与残余回声的接近延迟匹
配不必要。进入 AEC 基准延迟线的偏移可以改变以补偿由于超采样、
少采样、缓存器，和/或最小信道延迟引起的回声路径中固定延迟。
anlpRef=aecRef（FRAME-SIZE - k + 1）;
%包络检测（峰值检测）anlpRef 信号。
%峰值检测器中的指数延迟模拟了汽车中的反射延迟。设置时间常数
以处理预计的最慢延迟反射条件。
%少于 255/256 的极点导致回声通过。
%大于 255/256 的极点导致近端的过度失真。
anlpRefEnvelop=max（abs（anlpRef），255/256* anlpRefEnvelop）
% anlpRefEnvelop 对于存储是 32 位。
% 16 位左右的 anlpRefEnvelop 将不衰减到低于 512/2^15。当低于
```

```
256/2^15 时截短到 16 位的 anlpRefEnvelop 每个样本衰减 1 位，而
这太快了（导致回声通过）。
%量化为 Z.31 格式
anlpRefEnvelope=floor(anlpRefEnvelop*2^31)/2^31;
%ANLP 的增量值是由 AEC 控制的增益并限制为 1。
anlpDelta=min(ONE,...
      anlpWindowGain*floor(anlpRefEnvelope*32768)/32768);
%量化为 Z.15 格式
anlpDelta=floor(anlpDelta*32768)/32768;
anlpDeltaHist((frame - 1)*FRAME-SIZE + k)= anlpDelta
%保留调整
%
%执行 AC 中心衰减器
%
%ANLP 输入通过噪声抑制器连接到 AEC 输出。
anlpIn=uplinkAudio(k);
%如果该输入低于窗口
if (anlpOutLast - anlpIn)>=anlpDelta,
%使用所有窗口外信号并衰减窗口内信号。
anlpOutLast= anlpIn + anlpDelta - ...
      anlpEchoGain*anlpDelta;
%此外如果输入高于该窗口，
else if (anlpIn - anlpOutLast)>= anlpDelta,
%使用所有窗口外信号并衰减窗口内信号。
anlpOutLast= anlpIn + anlpDelta - ...
      anlpEchoGain*anlpDelta;
%此外如果输入在该窗口内，
else
%衰减该信号。
anlpOutLast= anlpEchoGain*( anlpIn - anlpOutLast) +
anlpOutLast;
end
```

```
%
anlpOutLast=floor(anlpOutLast*32768)/32768;                %保留
到下次.
%
%加入舒适噪声以至 ANLP 输出噪声具有与输入到 ANLP 的汽车噪声输
入相同的电平和相似频谱。
%
%使用来自汽车的捕获噪声帧的随机样本。
%即使从汽车捕获的音频偶然包含话音，这产生与汽车噪声相同功率
的白噪声。
anlpSeed=rem(48271*anlpSeed,2147483647);
anlpComfortNoiseIn=anlpcomfortNoise(1 +...
   rem(anlpSeed,COMFORT-NOISE-SIZE));
%使用上面讨论的 ARMA 模型滤除白噪声。
%假设 anlpArGain=1 - anlpArCoef，下列是等效式，
% anlpComfortNoiseOut= anlpArCoef *anlpComfortNoiseOutOld
+ ...
%                    anlpArGain*(anlpComfortNoiseIn          +
0.8125*anlpComfortNoiseInOld);
ma=  anlpComfortNoiseIn  +  0.8125*anlpComfortNoiseInOld;
%移动平均
maDiff=anlpComfortNoiseInOld - ma;
%量化为 S1.15 格式
%舍入是必须的以避免在舒适噪声上的偏移。
%当舍入具有刚好 0.5 分数的负数时，加入 2^(-17)强制 1 的补码浮
点起 2 的补码相同的作用.
maDiff=round(maDiff*32768 + 2^(-17))/32768;
anlpComfortNoiseOut=ma + anlpArCoef* maDiff;
%量化为 S1.15 格式
%舍入是必须的以避免在舒适噪声上的偏移。
anlpComfortNoiseOut=round(anlpComfortNoiseOut*32768 + 2^(-
17))/32768;
```

```
anlpComfortNoiseOut=max(-1 ,min(ONE, anlpComfortNoiseOut));
%为下个环路延迟变量
anlpComfortNoiseInOld=anlpComfortNoiseIn;
anlpComfortNoiseOutOld= anlpComfortNoiseOut;
%限制舒适噪声到窗口尺寸.
anlpComfortNoiseOut=min(anlpdelta,...
        max(-anlpdelta,anlpComfortNoiseOut));
%将舒适噪声乘以比例以便 ANLP 输出噪声等于 ANLP 输入噪声电平.
anlpComfortNoiseOut    =    anlpComfortNoiseOut    *(ONE    -
anlpEchoGain);
%量化为 S.15 格式
anlpComfortNoiseOut=
floor(anlpComfortNoiseOut*32768)/32768;
%将舒适噪声加入到 ANLP 输出信号中.
uplinkAudio(k)=max(-1,min(ONE,anlpOutLast                    +
anlpComfortNoiseOut));
end
end
%将在整数帧中的帧上采集的输出保留在文件中.
```

```
  fwrite(fidOut, [uplinkAudio; aecOut]*32768, 'int16');

%显示(['aecInEnergy = ' dec2hex(aecInEnergy * ENERGY_SCALE)])
%显示(['aecErrorShift = ' dec2hex(aecErrorShift+(aecErrorShift<0)*65536)])
%显示(['aecUpdate(1) = ' dec2hex(aecUpdate(1)*32768+(aecUpdate(1)<0)*65536)])
%显示(['aecUpdate(2) = ' dec2hex(aecUpdate(2)*32768+(aecUpdate(2)<0)*65536)])
%显示(['aecUpdateNear(1) = ' ...
%      dec2hex(aecUpdateNear(1)*32768+(aecUpdateNear(1)<0)*65536)])
%显示(['aecUpdateNear(2) = ' ...
%      dec2hex(aecUpdateNear(2)*32768+(aecUpdateNear(2)<0)*65536)])
%显示(['aecUpdatePeak2 = ' dec2hex(aecUpdatePeak2*2^31)])
%显示(['aecOutEnergy = ' dec2hex(aecOutEnergy * ENERGY_SCALE)])
%显示(['aecRefEnergy = ' dec2hex(aecRefEnergy*128)])
%显示(['anlpInEnergy = ' dec2hex(anlpInEnergy * ENERGY_SCALE)])
%显示(['aecUpdateBase = ' dec2hex(aecUpdateBase*32768)])
%显示(['aecUpdatePeak = ' dec2hex(aecUpdatePeak*32768)])
%显示(['aecPeakToBase = ' dec2hex(aecPeakToBase*16)])
%显示(['aecStatusGauge = ' dec2hex(aecStatusGauge*32768)])
%显示(['aecInNoise = ' dec2hex(aecInNoise*2^31)])
%显示(['anlpInNoise = ' dec2hex(anlpInNoise*2^31)])
%显示(['anlpComfortNoise(1) = ' ...
%      dec2hex(anlpComfortNoise(1)*32768+(anlpComfortNoise(1)<0)*65536)])
%显示(['anlpComfortNoise(2) = ' ...
%      dec2hex(anlpComfortNoise(2)*32768+(anlpComfortNoise(2)<0)*65536)])
%显示(['anlpArCoef = ' dec2hex(anlpArCoef*2^31)])
%显示(['anlpArGain = ' dec2hex(anlpArGain*32768)])
%显示(['aecChanGainTrial = ' dec2hex(aecChanGainTrial*?)])
```

```
%显示(['aecChanGain = ' dec2hex(aecChanGain*32768)])
%显示(['aecVoiceGainBase = ' dec2hex(aecVoiceGainBase*32768)])
%显示(['aecVoiceGain = ' dec2hex(aecVoiceGain*32768)])
%显示(['aecEchoGain = ' dec2hex(aecEchoGain*32768)])
%显示(['aecDenom = ' dec2hex(aecDenom*8)])
%显示(['aecNumer = ' dec2hex(aecNumer*2^18)])
%显示(['aecUpdateGain = ' dec2hex(aecUpdateGain*32768)])
%显示(['aecCoef(1) = ' dec2hex(aecCoef(1)*32768+(aecCoef(1)<0)*65536)])
%显示(['aecCoef(2) = ' dec2hex(aecCoef(2)*32768+(aecCoef(2)<0)*65536)])
%显示(['anlpWindowGain = ' dec2hex(anlpWindowGain*16384)])
%显示(['anlpEchoGain = ' dec2hex(anlpEchoGain*32768)])
%显示(['anlpEchoGain = ']); 20*log10(anlpEchoGain)
%显示(['anlpComfortNoiseInOld = ' dec2hex(anlpComfortNoiseInOld*32768)])
%显示(['anlpComfortNoiseOutOld = ' dec2hex(anlpComfortNoiseOutOld*32768)])
%显示(['uplinkAudio(1) = ' ...
%       dec2hex(uplinkAudio(1)*32768+(uplinkAudio(1)<0)*65536)])
%显示(['uplinkAudio(2) = ' ...
%       dec2hex(uplinkAudio(2)*32768+(uplinkAudio(2)<0)*65536)])
%暂停
%标绘([abs(aecUpdate)/aecUpdateBase; ...
%       zeros(AEC_REF_TAPS - AEC_REF_TAPS - AEC_NEAR_TAPS, 1); ...
%       abs(aecUpdateNear)/aecUpdateBase])
%轴线([0 AEC_REF_TAPS 0 aecPeakToBase+0.1])
%x标签 ('Update element'),y标签 ('Normalized update magnitude'),暂停
%标绘(20*log10(abs(aecCoef .* profile))),轴线([0 AEC_COEF_TAPS -100 1]);
%x标签 ('Coefficient number'),y标签 ('Magnitude in dB'),暂停

end

fclose(fidOut);

清除  AEC_MAX_GAIN_THRESH AEC_BASELINE_THRESH AEC_STATUS_GAUGE_SCALER ONE
清除  AEC_COEF_TAPS AEC_NEAR_TAPS AEC_REF_TAPS
清除  FRAME_SIZE FRAME_BITS ENERGY_SCALE COMFORT_NOISE_SIZE
清除  confirmedNoVoiceFlag
清除  LRmatrix downlinkAudio uplinkAudio aecRef aecUpdate aecUpdateNear aecOut
清除  frame m k i anlpSeed aecEchoEst aecShiftPending aecErrorShift ASM T
清除  aecRefEnergy aecInEnergy aecOutEnergy anlpInEnergy aecEchoEstEnergy
清除  anlpIn anlpRef anlpRefEnvelope anlpDelta anlpOutLast
清除  aecUpdatePeak2 aecUpdatePeak aecUpdateBase aecPeakToBase
清除  aecStatusGauge aecDenom aecNumer aecUpdateGain aecLoopEchoGain
清除  aecInNoiseStateVars aecInEchoNoiseStateVars anlpInNoiseStateVars
清除  aecInVoice anlpInVoice aecInEchoVoice aecNearVoice
清除  aecNearRatio aecNearGainLast anlpEchoGain
清除  aecChanGainTrial anlpComfortNoiseIn anlpComfortNoiseOut
清除  aecPeakGain anlpWindowGain anlpDenom anlpArGain error quotient ma
清除  fidIn fidOut anlpNearSpeechCount anlpNearSpeechFlag ECHO_TO_NOISE_GOAL
```

%根据一个信号的帧能量估计噪声帧能量。

% 基于标准的噪声估计用 32 比特的变量提供更宽的动态范围，因此即使在非常低的噪声电平时（例如 test_65.raw 的远端）该运算也保持不变。然而，基于标准的算法要求平方根和 32 比特平方运算（更高的 MIPS（百万条指令／秒））。该基于能量的算法已经为低噪声电平的量化进行优化，并且它进行一个可以接收的工作—尤其考虑噪声

电平估计对于非常低的噪声电平（话音始终远大于噪声）并不重要。还说明可以为基于标准和基于能量的算法使自适应速度相同。在此所有的调整操作使用移位—仅为基于标准的算法移位一个以上的比特。

% 目标是在信号中没有语音时快速更新估计，否则慢速更新估计。需要恒定自适应以在汽车噪声快速变化时予以跟踪。话音激活检测器（VAD）输出对于确定何时快速更新是一个好的开始，但当确实存在远高于噪声的话音低电平时它有时指示没有话音。NoVoiceFlag 是由线性回波消除器的状态标准限定后的 VAD 输出以提高可靠性。然而，状态标准并不区分噪声和近端话音、双方讲话或音调。因此，该算法必须容许快速更新周期内的一些话音。状态标准顺利工作的时间是在远端单方讲话时，在此对于正确估计噪声以避免可以听到的回波是最重要的。在此使用以降低错误偏差（主要出现在近端话音过程中）的技巧是根据比较结果而不是线性滤波能量信号以固定速率提高和降低估计。使用提高时间常量 1/4 的降低时间常量使估计偏向于噪声电平，而不考虑某些噪声。通过固定增量而非固定时间常量提高和降低将导致与噪声电平成正比的时间常量。

%很难讲出上升噪声电平和话音之间的差别。在此提高常量确定跟踪速率。然而，提高必须很慢以在话音过程中最小化误差增长。执行状态机以在某些条件下允许噪声估计更快的增加。如果状态机观察到在无话音的过程中在至少 900 毫秒的时间内信号能量大于噪声估计的八倍，假定噪声电平已经增加，并设置噪声电平使其等于测试周期的测试噪声估计。当输入噪声由来自远端的噪声控制时，由于 AMPS 带内控制信道或者由于不良的射频条件的消隐将致使噪声电平临时下降。状态机在消隐之后试图恢复噪声估计。首先确认能量下降在 5 和 25 个帧长度之间。然后确认能量返回到原始水平。如果能量保持在围绕原始噪声电平的窗口内，需要较短的确认周期。如果能量跳变得远高于原始水平，则可能出现噪声，并需要一个较长的确认周期以确保噪声电平未下降。

%NoiseStateVars 矩阵的定义：

（1）= 噪声估计

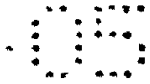
（2）= 消隐之前的噪声估计或测试噪声估计

（3）=状态变量/计数器

```
function[confirmedNoVoiceFlag,noiseStateVars]=…
  estnoise(inEnergy, noVoiceFlag, noiseStateVars)
%以与父程序相同的方式定义所需常量.
%更新综合周期内的抽样数。
FRAME_SIZE = 160;
%向右移位在一个抽样帧上的累加值的比特数。
FRAME_BITS=ceil(log2(FRAME_SIZE));
%将能量量化成 32 比特的比例因子（Z8.23 格式 w/FRAME_SIZE=160）
ENERGY_SCALE = 2^(31-FRAME_BITS);
%如果父程序已经初始化噪声估计,
if noiseStateVars(1)==FRAME_SIZE,
   %采取行动以加速呼叫开始的自适应;
   % 设置标记以便舒适噪声将用该电平初始化;
   confirmedNoVoiceFlag=1;
   %开始于等于该能量的噪声估计。
   NoisestateVars(1)=inEnergy;
  Else
   %更新噪声估计。
   % 通过缺省，清除标志以便指示没有舒适噪声训练。
   ConfirmedNoVoiceFlag=0;
   %在白噪声上 Matlab 中执行下述偏向测试:
% >>n=160*3000;noise=zeros(1,m);
% >>seed=1;for i=1:n,[noise(i) seed]=noisegen(seed);end
% >>m=3000;for i=1:m,noiseEnergy(i)=sum(noise(160*(i-1)+1:160*i).^2);end
% >>ne=100;for i=1:m,if ne>noiseEnergy(i),
% >>ne=ne-ne/16;else,ne=ne+ne/64;end,neh(i)=ne;end
% >>plot(neh)              % neh 是噪声估计历史记录.
% >>axis([0 20 0 100])     % 注意neh在10帧之后调整.
% >>1/(sum(neh(11:m))/(m-10)/(sum(noiseEnergy)/m)-1)
% ans = -14.2082

   %这表示噪声估计具有大约 - 1/14 的偏向因子。
   % 为了简化运算，为了比较从噪声估计中减去其值的 1/16 以恢复偏向。
   % 在汇编时，为了比特的正确性在减法之前向右移位。
```

```
noiseBiased=noiseStateVars(1)-noiseStateVars(1)/16;
%量化成 32 比特
noiseBiased=cell(noiseBiased*ENERGY_SCALE)/ENERGY_SCALE
;
%如果 VAD 和状态标准指示没有话音,
if noVoiceFlag,
  %快速调整噪声估计。
  % 如果噪声估计太高,
  if noiseBiased > inEnergy,
    %降低噪声估计。
% 为加速跟踪下降的噪声电平，如果噪声估计高至 8 倍，设法立即设置噪声估计使其等于信号能量，然而，这使得在激活时偏向极其相反，并产生错误检测问题。该时间常量太短以致于使噪声估计反而工作得很好。
% 为了最小化偏差，在噪声估计等于信号能量时不要降低。
% 除了零点之外，量化致使最小的改变 1/ENERGY_SCALE.
% 在汇编中，为了比特的正确性在右移之前取反。
noiseStateVars(1)=noiseStateVars(1)-
noiseStateVars(1)/16;
%为舒适噪声训练设置标记。输入能量低于噪声估计的要求提高了在话音期间不改变舒适噪声的可能性。
ConfirmedNoVoiceFlag=1;
Else
%提高噪声估计。
% 使用最小增量以避免停滞在零点附近。
% 改为不使用相加 min(noiseStateVars(1)/64,1/ENERGY_SCALE)，因为它导致较慢的自适应以致瞬间增加的噪声。
NoiseStateVars(1)=noiseStateVars(1)+noiseStateVars(1)/64+
……1/ENERGY_SCALE;
End
Else
%缓慢提高噪声估计。
```



% 因为正确跟踪递减噪声电平的重要性，在话音期间不停止自适应。噪声的估计过高导致话音能量的估计过低。由于较低的远端话音能量，这对 NLP 输入的影响比对线性回波消除器的影响更明显。因而，结果是不充分的回波抑制。所以最好过低估计噪声，而不要过高估计噪声。

```
% 如果噪声估计过高，
if noiseBiased > inEnergy,
%降低噪声估计。
% 除了在零点之外，量化致使最小的改变 1/ENERGY_SCALE.
% 在汇编中，为了比特的正确性在右移之前求反。
NoiseStateVats(1)=noiseStateVars(1)-
noiseStateVars(1)/64;
Else
%提高噪声估计。
% 乍看起来好象正确跟踪递减的噪声电平仅需要在话音期间降低。然而，这将致使偏向变得过于相反。根据经验将提高速率确定成最快的可能速率，而不允许话音期间过多的错误变化。因为在暂停之间话音可以持续若干秒，这变得非常慢。
% 不使用最小增量，否则对于话音期间的低噪声电平，上升沿将过大。
NoiseStateVars(1)=noiseStateVars(1)+noiseStateVars(1)/1
024;
End
 End
 %量化成 32 比特。
NoiseStateVars(1)=floor(noiseStateVars(1)*ENERGY_SCALE)
/ENERGY_SCALE;
%用于 AMPS 消隐和噪声跳变跟踪的状态机。
% 不要在初始化时执行。
% 如果状态机处于空闲状态，
if noiseStateVars(3)==0,
 % 如果信号能量已经明显低于噪声估计，并且如果 inEnergy 的量
```

```
化并不给出错误结果，
    if noiseStateVars(1)>max(8*inEnergy, 8/ENERGY_SCALE),
      %为状态机存储噪声估计。
      NoiseStateVars(2)=noiseStateVars(1);
      %开始状态机以查找消隐。
      NoiseStateVars(3)=1;
    %如果 VAD 和状态标准指示没有话音，信号能量明显高于噪声估
计，并且如果噪声估计的量化并不给出错误结果，
      elseif noVoiceFlag & …
            (  inEnergy  >  max(8*noiseStateVars(1)  ,
8/ENERGY_SCALE)),
      %初始化测试噪声估计。
      NoiseStateVars(2)=inEnergy;
      %开始状态机以查找噪声跳变。
   NoiseStateVars(3)=-1;
   End
   %否则如果状态机在查找噪声跳变，
   elseif noiseStateVars(3) < 0,
   %如果 VAD 和状态继续指示没有话音，并且如果信号能量保持明显
高于噪声估计，
   if noVoiceFlag & (inEnergy > 8*noiseStateVars(1)),
     %偏向测试噪声估计就象正规的噪声估计一样。
     % 在汇编中，为了比特的正确性在相减之前右移。
     NoiseBiased=noiseStateVars(2)-noiseStateVars(2)/16;
     %量化成 32 比特。
     NoiseBiased=cell(noiseBiased*ENERGY_SCALE)/ENERGY_SCA
LE;
     %如果测试噪声估计过高，
     if noiseBiased > inEnergy,
      %降低测试噪声估计。
      % 量化导致最小变化 1/ENERGY_SCALE.
      % 在汇编中，为了比特正确性在右移之前求反。
```

```
      NoiseStateVars(2)=noiseStateVars(2)-
noiseStateVars(2)/16;
   Else
      %提高测试噪声估计
      NoiseStateVars(2)=noiseStateVars(2)+noiseStateVars(2)
/64;
   End
    %量化成 32 比特
    noiseStateVars(2)=floor(noiseStateVars(2)*ENERGY_SCALE)
/ENERGY_SCALE;
    %递减也用作一个计数器的状态变量。
    NoiseStateVars(3)=noiseStateVars(3)-1;
   %如果在一个足够长的周期内信号能量保持明显高于噪声估计。
   % (测试跟踪 s_top10_1.raw 的回波需要 45 个帧)
   if noiseStateVars(3)==-45,
     %噪声估计跳转到测试噪声估计。
     NoiseStateVars(1)=noiseStateVars(2);
     %将状态机重置回空闲状态。
     NoiseStateVars(3)=0;
    End
   Else,
   %否则状态机在查找消隐。
   Else,
   % 递增状态变量。
   NoiseStateVars(3)=noiseStateVars(3)+1;
   %状态 1-100 计算所谓消隐周期内的帧数。
   If noiseStateVars(3) < 101,
     %如果能量再次变高,
     if  8*inEnergy >noiseStateVars(2),
      %如果消隐小于 5 帧,
      if noiseStateVars(3)-1<5,
        %检测是错误的或者不值得恢复噪声估计。使状态机返回空闲
```

状态。

```
        NoiseStateVars(3)=0
      Else
        %话音已经出现或者在消隐之前噪声返回。
        % 将状态变量设置为 101 以启动下一程序段。
        NoiseStateVars(3)=101;
       End
      %如果低能量帧的计算过长,
      elseif noiseStateVars(3)-1==25,
      %消隐并未出现—而是噪声电平降低。
      % 使状态机返回空闲状态。
      NoiseStateVars(3)=0;
     End
     %状态机已经检测到所谓消隐周期的结束。
     % 不考虑状态机是否在计算话音或噪声帧,首先检测能量是否又降低。
     Elseif 8*inEnergy < noiseStateVars(2),
      %消隐并未出现—而是噪声电平降低。
      % 使状态机返回空闲状态。
      NoiseStateVars(3)=0;
      %状态 101 - 200 计算所谓消隐周期之后话音或噪声帧的个数。
      Elseif noiseStateVars(3)<201,
       %如果能量变得很高,
       if inEnergy >noiseStateVars(2)*8,
       %假设这是话音。
       % 将状态变量设置为 201 以启动下一程序段。
       noiseStateVars(3)=201;
       %如果在最大和最小能量接近于所存储的噪声估计时存在足够的帧数,
       elseif noiseStateVars(3)-101==10,
       确认消隐。
       将噪声估计恢复成消隐之前的值。
```

```
        NoiseStateVars(1)=noiseStateVars(2);
        %使状态机返回空闲状态。
        NoiseStateVars(3)=0;
        End
      %状态 201 - 300 计算所谓消隐周期之后话音帧的个数。
      % 如果在最小能量接近于所存储的噪声估计时存在足够的帧
数,
      elseif noiseStateVars(3)-201==50,
      %确认消隐。
      % 将噪声估计恢复成消隐之前的值。
      NoiseStateVars(1)=noiseStateVars(2);
      %使状态机返回空闲状态.
     NoiseStateVars(3)=0;
     End
   End
End
**********************************
```

本领域技术人员应当理解，本发明不限于为说明目的已经在此描述的特定示范性实施例。例如，所公开实施例的各种功能块实际上是概念化的。这些块功能的实际实施可以使用各种技术完成。此外，可以使用几个标准数字信号处理芯片实现每个示范性系统，例如单一应用性能的集成电路，或适当配置的计算机。还注意，尽管已经在在音频回声消除环境下描述了示范性实施例，本发明的教导同样可以应用于网络回声消除的环境中（例如，近端用户是陆地线路用户而远端用户是移动用户的场合）。此外，本发明的某些方面一般可以应用于通信系统而不限于回声抑制系统。因此，本发明的范围由附带的权利要求书而非前面的说明书所限定，所有与本发明思想相一致的等同发明均包含在其中。

说 明 书 附 图

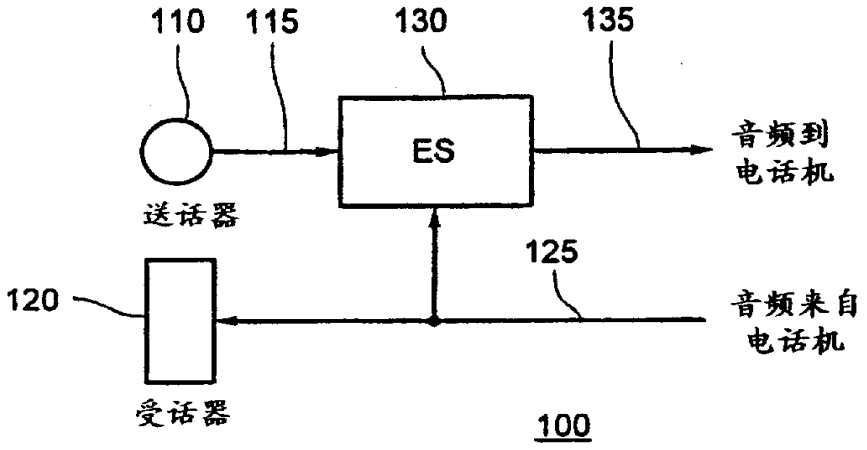


图 1A

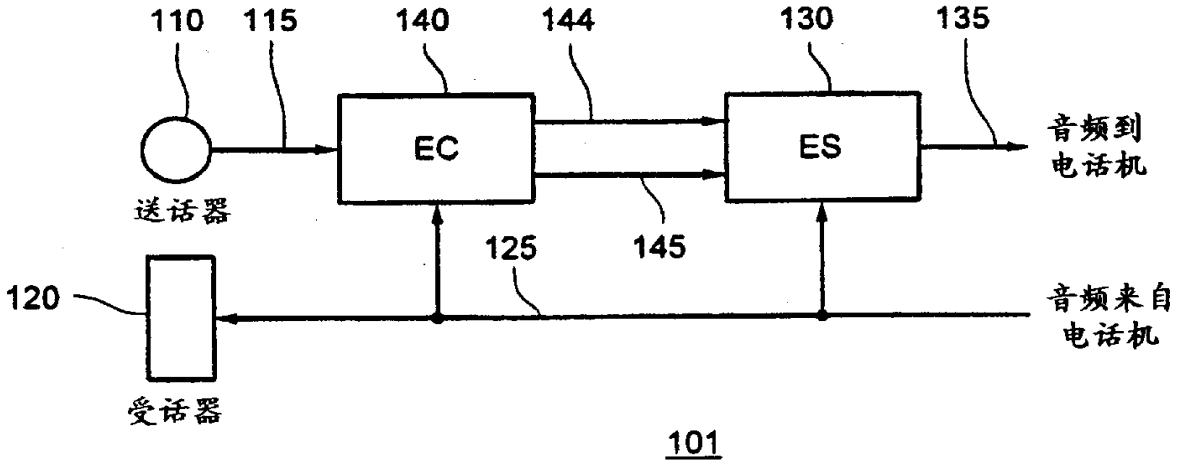


图 1B

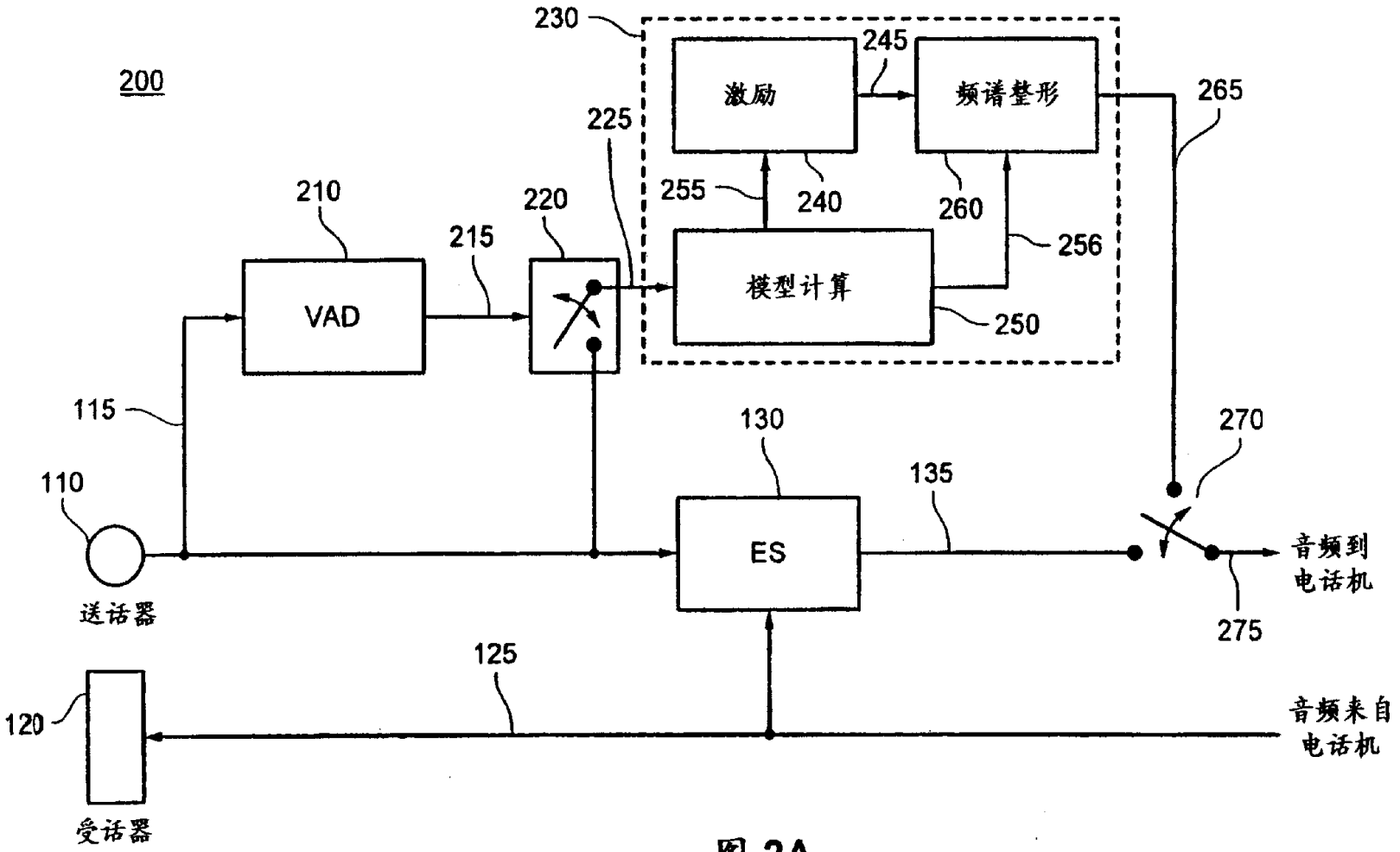

200
230
激励
245
频谱整形
265
225
255
240
260
256
210
220
215
VAD
模型计算
250
115
130
270
110
135
ES
音频到
电话机
送话器
275
125
120
音频来自
电话机
受话器

图 2A

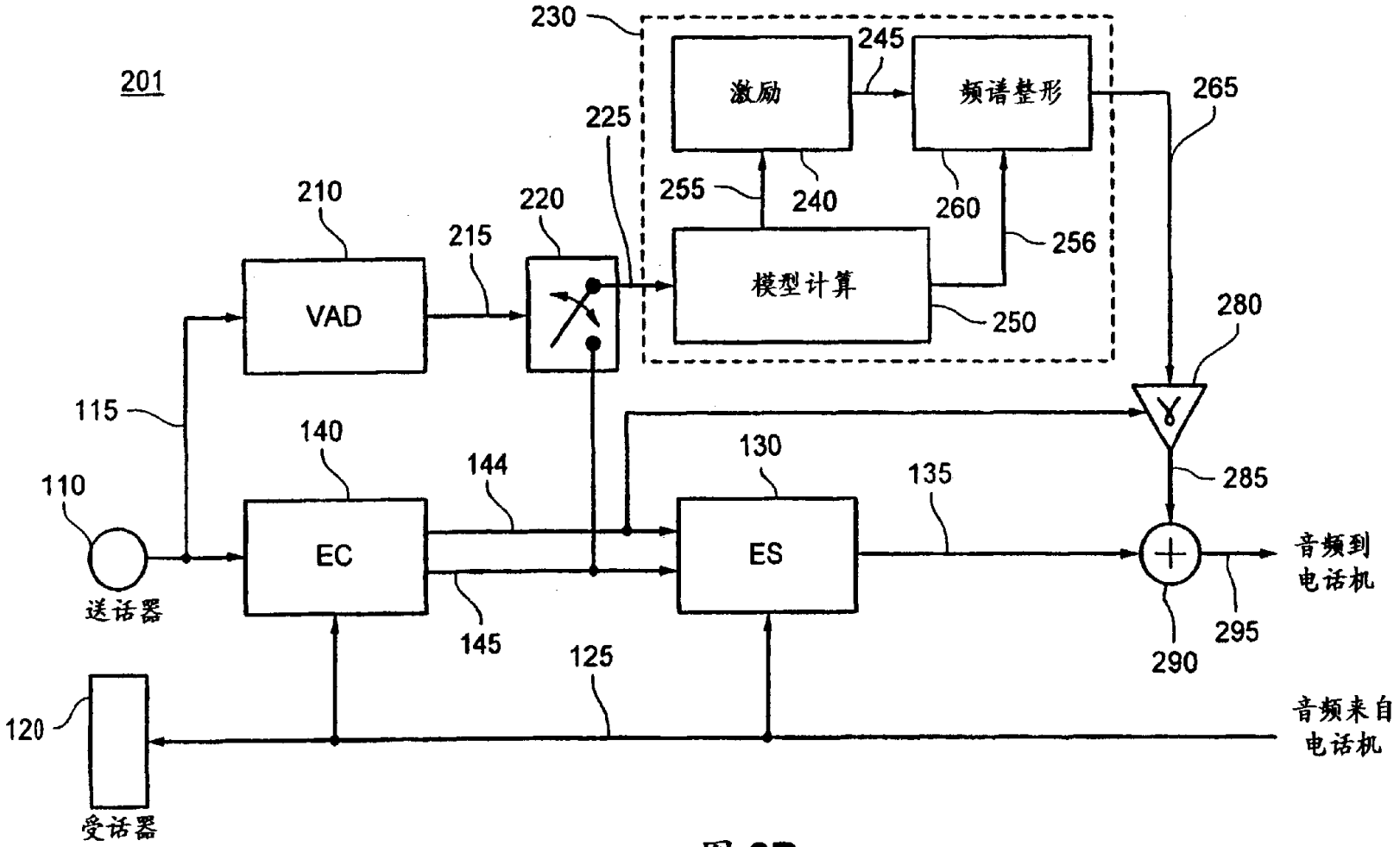

201
230
激励
245
频谱整形
265
225
255
240
260
256
210
220
215
VAD
模型计算
250
280
115
140
130
144
135
285
110
EC
ES
音频到
电话机
送话器
145
295
290
125
120
音频来自
电话机
受话器

图 2B

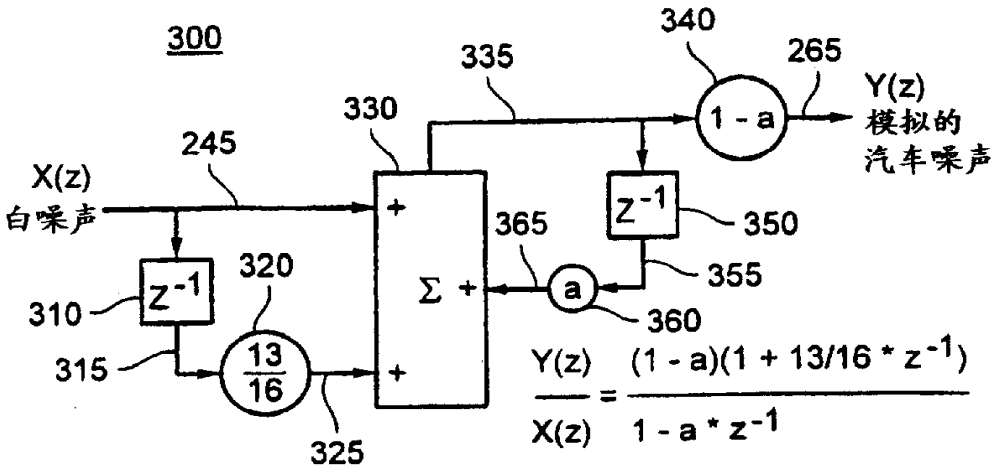

300
340
335
265
330
1 - a
Y(z)
模拟的
汽车噪声
245
X(z)
白噪声
+
Σ +
+
365
Z-1
350
355
320
310
Z-1
a
360
315
13/16
325
Y(z) (1 - a)(1 + 13/16 * z-1)
X(z) 1 - a * z-1

图 3

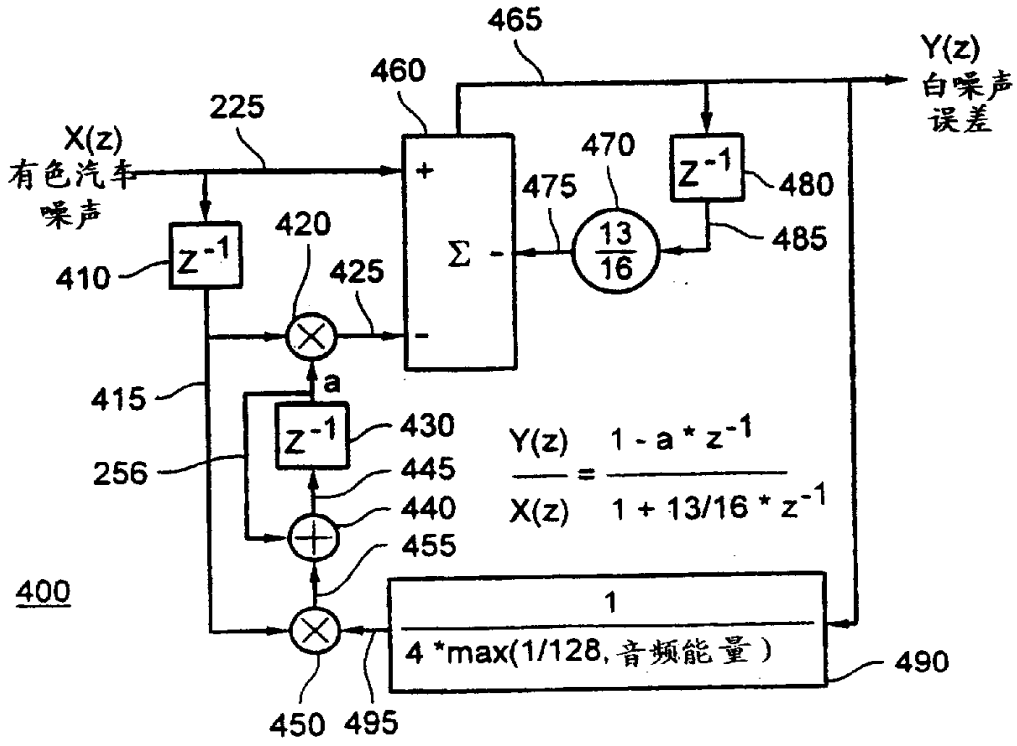

465
460
Y(z)
白噪声
误差
225
X(z)
有色汽车
噪声
+
Σ -
-
470
475
Z-1
480
13/16
485
420
410
Z-1
425
a
415
Z-1
430
256
445
440
455
400
1
4 *max(1/128, 音频能量)
490
450
495
Y(z) 1 - a * z-1
X(z) 1 + 13/16 * z-1

图 4

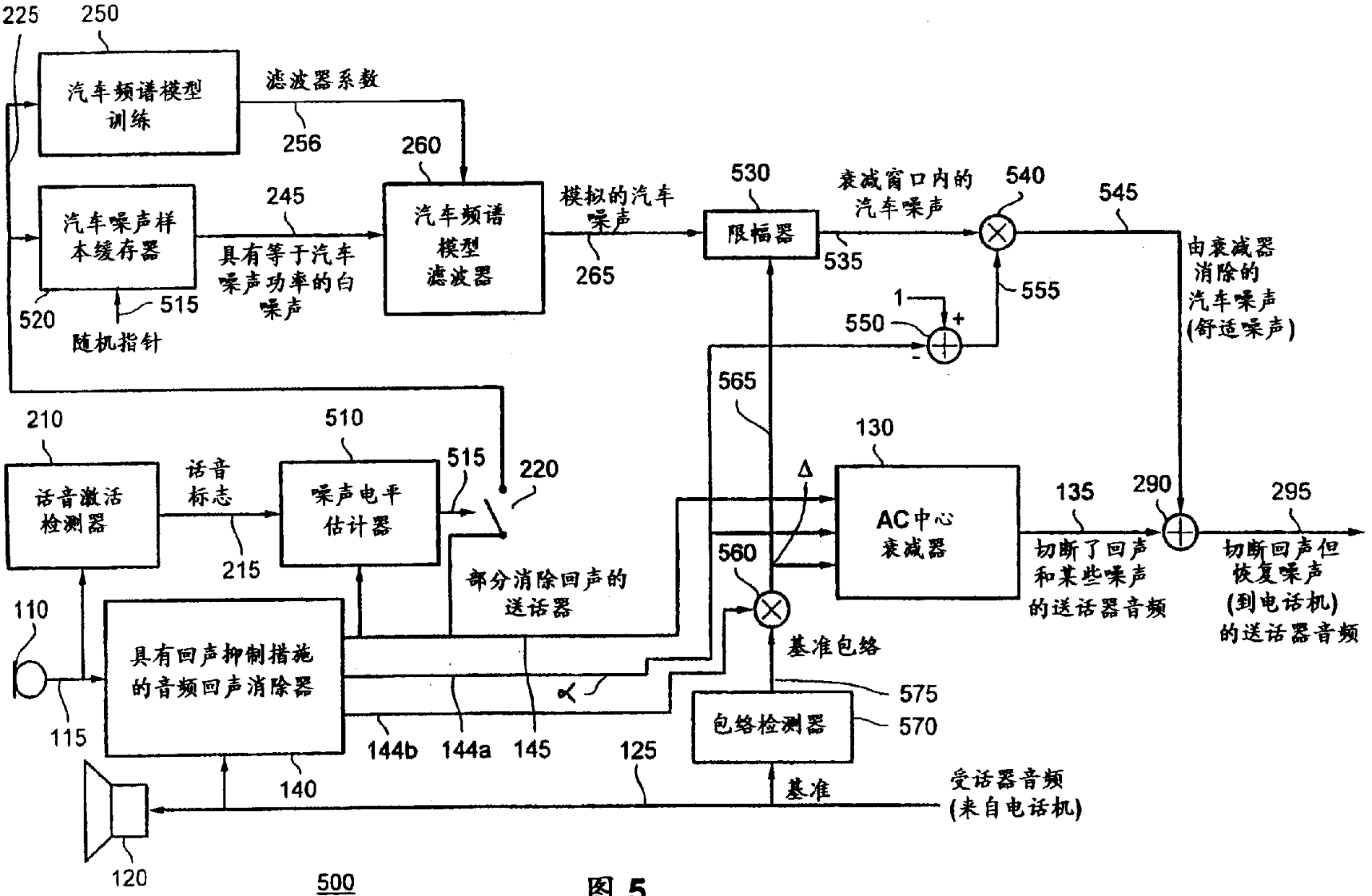

225
250
汽车频谱模型训练
滤波器系数
256
260
汽车噪声样本缓存器
245
具有等于汽车噪声功率的白噪声
汽车频谱模型滤波器
模拟的汽车噪声
265
530
限幅器
535
衰减窗口内的汽车噪声
540
545
由衰减器消除的汽车噪声(舒适噪声)
555
1
550
+
-
520
515
随机指针
565
210
话音激活检测器
话音标志
215
510
噪声电平估计器
515
220
部分消除回声的送话器
130
Δ
AC中心衰减器
135
290
295
切断了回声和某些噪声的送话器音频
切断回声但恢复噪声(到电话机)的送话器音频
560
基准包络
110
具有回声抑制措施的音频回声消除器
575
包络检测器
570
115
144b
144a
145
125
基准
受话器音频(来自电话机)
140
120
500

图 5