



## [12] 发明专利说明书

专利号 ZL 02105511.4

[45] 授权公告日 2005 年 9 月 21 日

[11] 授权公告号 CN 1220334C

[22] 申请日 1992.6.11 [21] 申请号 02105511.4  
分案原申请号 92104618.9

[30] 优先权

[32] 1991.6.11 [33] US [31] 07/713,661

[71] 专利权人 高通股份有限公司

地址 美国加利福尼亚州

[72] 发明人 保罗·E·雅各布  
威廉·R·加德纳 冲·U·李  
克莱恩·S·吉豪森  
S·凯瑟琳·兰姆 民昌·蔡

审查员 凌 林

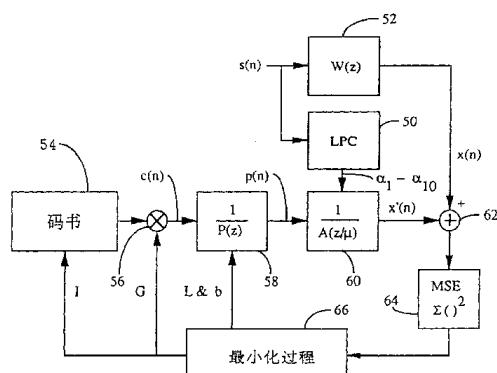
[74] 专利代理机构 上海专利商标事务所有限公司  
代理人 钱慰民

权利要求书 2 页 说明书 48 页 附图 22 页

[54] 发明名称 可变速率声码器

[57] 摘要

本发明揭示了一种为多帧语音信号中一帧语音的背景噪声生成当前估算的方法和设备。该方法和设备计算一帧语音信号的能量电平；获得前一帧语音信号的背景噪声估算；并根据能量电平和前一帧语音信号的背景噪声估算生成当前背景噪声估算。



1. 一种为多帧语音信号中一帧语音的背景噪声生成当前估算的方法，其特征在于，包括以下步骤：

计算一帧语音信号的能量电平；

获得前一帧语音信号的背景噪声估算；和

根据所述能量电平和前一帧语音信号的所述背景噪声估算，生成当前背景噪声估算。

2. 如权利要求 1 所述的方法，其特征在于，所述生成步骤包括限制当前背景噪声估算相对于前一帧语音信号之背景噪声估算的增量，使其低于一预定值。

3. 如权利要求 2 所述的方法，其特征在于，所述限制步骤包括将当前背景噪声估算限制为小于或等于所述能量电平的值。

4. 如权利要求 2 所述的方法，其特征在于，所述增量由一增量百分数构成。

5. 如权利要求 2 所述的方法，其特征在于，所述增量由一绝对增量构成。

6. 一种语音处理器，用于处理包含多个帧的语音信号，其特征在于，所述语音处理器包括：

计算装置（202），用于计算一帧语音信号的能量电平；

获取装置，用于获得前一帧语音信号的背景噪声估算；和

生成装置，用于根据所述能量电平和前一帧语音信号的所述背景噪声估算生成当前背景噪声估算。

7. 如权利要求 6 所述的语音处理器，其特征在于，所述生成装置包括限制装置（312，318），用于限制当前背景噪声估算相对于前一帧语音信号之背景噪声估算的增量，使其低于一预定值。

8. 如权利要求 7 所述的语音处理器，其特征在于，所述限制装置（312，318）包括将当前背景噪声估算限制为小于或等于所述能量电平的值的装置（318）。

9. 如权利要求 7 所述的语音处理器，其特征在于，所述增量由一增量百分数构成。

10. 如权利要求 7 所述的语音处理器，其特征在于，所述增量由一绝对增量构成。

11. 如权利要求 6 所述的语音处理器，其特征在于，所述计算装置（202）包括第一电路（202），用于计算一帧语音信号的能量电平；并且

所述获得装置和所述生成装置还包括：

第二电路，用于计算前一帧语音信号的背景噪声估算，并且对前一帧语音信号的背景噪声估算增加一预定量，以生成一增大的估算值；和

第一多路转换器（300），它与第一和第二电路耦连，用于接收所述增大的估算值以及所述能量电平，并且选择增大的估算值或能量电平，作为语音信号当前帧的背景噪声估算。

12. 如权利要求 11 所述的语音处理器，其特征在于，还包括第一限制器（314），它与第二电路耦连，用于将所述增大的估算值限制为低于一预定值的值。

13. 如权利要求 12 所述的语音处理器，其特征在于，还包括第二限制器（318），它与第一多路转换器耦连，用于将语音信号当前帧的背景噪声估算限制为小于或等于所述能量电平的值。

14. 如权利要求 12 所述的语音处理器，其特征在于，第二电路包括：

第一加法器（304），用于将一标称常数与前一帧语音信号的背景噪声估算相加，生成一绝对增大的估算值；

第二乘法器（306），用于将前一帧语音信号的背景噪声估算乘以一个略大于 1 的常数，生成一百分数增大的估算值；

第二多路转换器，它与第一加法器（304）和所述第二乘法器（306）耦连，用于接收绝对增大的估算值和百分数增大的估算值；

第三电路，它与第一加法器（304）、第二乘法器（306）和第二多路转换器（308）耦连，用于控制第二多路转换器（308），选择绝对增大的估算值和百分数增大的估算值中的较大者，作为增大的估算值。

## 可变速率声码器

本申请是申请日为 1992 年 6 月 11 日、申请号为 92104618.9、与本申请相同发明名称的发明专利申请的分案申请。

### 发明领域

本发明涉及语音处理。具体讲，本发明涉及一种用于压缩语音的新颖且已改进的方法和系统，其中，压缩量动态地变化，但对再生语音的质量影响最小。此外，由于压缩后的语音数据要通过一个有可能引入错误的信道传送，故而本发明的方法和系统还会将信道传输错误对声音的影响减至最小。

### 背景技术

利用数字技术传送声音已经得到广泛的应用，在远距离和数字式无线电话设备中尤其如此。这转而使人们有兴趣确定在保持再生语音听觉质量的信道上能够传送的最低信息量。如果语音是在简单取样并数字化后传送的，那么，为了达到普通模拟式电话机的语音质量，就需要 64 千比特/每秒(Kbps)数量级的数据速率。但是，通过采用语音分析，再进行适当的编码、传送，在接收器中进行再合成，就能显著地降低数据速率。

采用人类语言信号产生模型所提取的参数来完成语声压缩技术的装置通常称为声码器。这类装置由一个分析传入的语音以提取相关参数的编码器和一个利用在传输信道中接收到的参数重新合成功音的译码器组成。为了精确些，模型必须不时地变化。这样，语音被划分成时间段或分析帧，每一帧内进行参数的计算。然后对每一新的帧更新参数。

在不同类别的语音编码器中，编码激励线性预测编码(CELP)、随机编码或矢量激励语音编码是属于一类的。在汤姆斯 E. 特莱门等人所著，发表于 1988 年移动卫星研讨会论文集上的论文“一种 4.8 Kbps 编码激励线性预测编码器”中描述了这一特定类别中的一个编码算法例子。

声码器的功能是消除语音中固有的自然冗余，从而将数字化语音信号压缩成低比特率信号。通常，语音中含有主要是因声道的滤波动作而产生的短期冗余，和

因发声区域声带的激励而产生的长期冗余。在 CELP 编码器中，这些发声区域的动作用两个滤波器来模拟，一个是短期共振峰滤波器，另一个是长期音调滤波器(pitch filter)。一旦除去这些冗余，得到的剩余信号就能模拟成白色高斯噪声，这也必须编码。这种技术的基础是计算一个称为 LPC 滤波器的参数，该滤波器利用人的声道模型来进行语音波形的短期预测。另外，与语音的音调有关的长期作用则通过计算一个主要模拟人的声带的音调滤波器的参数而得到模拟。最后，这些滤波器还必须被激励，这是通过从码书的大量随机激励波形中确定一个在激励上述两个滤波器的时候产生最接近原始语音的效果的波形而完成的。这样，被传送的参数涉及到三项：(1)LPC 滤波器，(2)音调滤波器，(3)码书激励。

声码技术的应用在尝试减少信道中传送的信息量而仍旧保持再生语音的质量这一点上离目标更进了一步，虽然如此，但还需要采用其它技术，来实现进一步的缩减。以前用来缩减信息传送量的一种技术是声音动作选通。在这种技术中，语音停顿时不传送信息。这种技术实现了预期的数据缩减效果，但存在着几个缺陷。

在许多情况下，由于削去了字语的初始部分，语音质量降低了。在不动作时切断信道的另一问题是系统用户感觉到缺少了通常伴随着语音的背景噪声，因此认为信道的质量低于普通电话。动作选通的再一个问题是，在没有语音时背景中偶然的突发噪声会引发发送器，在接收器中产生恼人的噪声群。

改进声音动作选通系统中合成语音的质量的一个尝试是在译码过程中加入合成的令人舒适的噪声。虽然通过加入舒适的噪声达到了质量上的某些提高，但是，由于舒适的噪声并不模拟编码器中的实际背景噪声，所以它并不能显著地改进总体质量。

进行数据压缩以实现需要传送的信息的缩减的一种更好的技术是进行可变速率声音编码。因为语音本身含有静默期间，例如停顿，所以，代表这些期间所需的数据量可以缩减。可变速率声音编码通过降低这些静默期间的数据速率而最有效地利用了这一事实。在静默期间数据速率的降低，与数据传送完全停止相反，克服了与声音动作选通相关的难题，同时实现了传送信息的缩减。

因而，本发明的一个目的是提供一种利用可变速率声音编码技术来压缩语音的新颖和改进的方法和系统。

## 发明内容

本发明采用了上述类别语音编码器的声音编码算法，即编码激励线性预测编

码(CELP)、随机编码或矢量激励语音编码。CELP 技术本身就能使表示语音所必需的数据量显著减少，并在重新合成时给出高质量的语音。如前所述，声码器参数就每一帧进行更新。本发明的声码器藉助于改变模型参数的频率和精度，形成可变的输出数据速率。

本发明极显著地不同于基本的 CELP 技术，在语音动作的基础上产生可变的输出数据速率。其结构被定义后，使得参数在语音停顿过程中较少或以较低精度进行更新。该技术使得要传送的信息量更加减少。被利用来降低数据速率的关键是声音动作因子，它是某一指定说话者在交谈过程中实际说话的平均时间百分比，就通常的双向电话交谈而言，该平均数据速率可降到 1/2 或更多。在语音停顿过程中，声码器只对背景噪声进行编码。在这些时候，与人的声道模型有关的某些参数勿需传送。

如前所述，先前在静默过程中限制信息传送量的途径称为声音动作选通，这是一种在静默的期间不传送信息的技术。在接收侧，该期间中加入合成的“舒适噪声”。与此相反，可变速率声码器持续不断地进行数据传送，在较佳实施例中是以大约介于 8 Kbps 和 1 Kbps 之间的速率进行的。进行连续数据传送的声码器消除了对“舒适噪声”的需求，而以背景噪声的编码给重新合成的语音提供更真实的质量。因而，本发明在语音和背景音之间进行平滑的转变，藉此，在重新合成的语音质量上，比声音动作选通有了更显著的改善。

本发明还结合了一种掩盖发生的错误的新颖技术。因为数据要通过一个可能有噪音的信道传送，如在无线电线路中传送，所以，它必须在数据中能容错。利用信道编码来减少所遇到的出错总数的先有技术在减少错误方面能够有某些成功之处。但是，仅有信道编码还不能完全达到保证高质量再生语言所必需的错误警戒水平。在声音编码持续进行的可变速率声码器中，一个错误可能会破坏涉及到某个重要的语音内容的数据，例如与一个单词或一个音节开头部分有关的数据。基于线性预测编码的声码器所带来的一个典型难题是，与声道模型有关的参数中的错误会产生类似人声的声音，还会将原始单词的发音改变到足以使听者迷惑的程度。在本发明中，错误被掩盖住，以降低其可觉察程度。这样，本发明所采用的错误掩盖技术使得错误对语音清晰度的影响急剧减少。

因为任何参数能够改变的最大数量在低速率下被限制在较小范围内，所以，在这些速率下所传送的参数中的错误会较少地影响语音质量。由于在不同速率下的错误对语音质量有不同的可觉察的影响，所以，传送系统可以最优化，以对较高速

率的数据提供更多保护。因而，作为一个附加特征，本发明容忍信道错误能力较强(具有鲁棒性)。

本发明通过采用可变速率输出形式的 CELP 算法，使得语音压缩率根据声音大小不同在 8：1 到 64：1 范围内动态地变动。刚才的压缩因子是对  $\mu$  律输入而言的，对于线性输入，该压缩因子将增大一倍。速率的确定是对各个帧分别进行的，以便充分地得益于声音动作因子。即使由于语音停顿而产生较少的数据，重新合成的背景噪声让人察觉到的质量降低也是最小的。利用本发明的技术，在最大数据速率 8 Kbps 下可以实现质量近似于长途电话的语音，在普通对话中的平均数据速率处于 3.5 Kbps 的量级。

本发明能够检测出语音中的短暂停顿，因而实现了有效声音动作因子的降低。速率确定是逐帧进行的，没有遗漏，因而，在短到一个帧周期的语音停顿间隙内数据速率可放慢，对较佳实施例，帧周期的典型值为 20 msec。因而，可以捕捉诸如在音节之间的那些停顿。因为不仅词组之间的长间歇停顿可以编码成较低速率，而且短暂停顿也可以，所以，这一技术降低了声音动作因子，在程度上超过了通常所考虑到。

因为速率确定是逐帧进行的，所以不会象声音动作选通系统那样，削去单词的初始部分。在声音动作选通系统中发生的这种性质的削去归因于语音检测和数据传送再开始之间的延迟。在每一帧基础上进行的速率确定使得语音中所有的变化都有自然的声调。

在声码器正常传送的情况下，在接收端一直能听到讲话者周围的背景噪声，因而在语音停顿时有更自然的声音。这样，本发明实现了向背景噪声的平稳转换。听者在语音中听到的背景音不会象声音动作选通系统中那样，在停顿时突然转变成一种合成的舒适噪声。

因为背景噪声持续地被编码后进行传送，所以，背景中的重要事件可以完整清晰地得到传送。在某些情况下，重要的背景噪声甚至可以以最高速率进行编码。例如，当背景中有一个人大声说话，或者有一辆救护车从站在街角处的一个用户身旁开过的时候，可能进行最高速率编码。但是，一成不变或者缓慢变化的背景噪声会以低速率进行编码。

利用可变速率声音编码能够确保基于码分多路访问(CDMA)的数字蜂窝式电话系统的容量增加一倍以上。因为在 CDMA 中，信道间的干扰随着任一信道中数据传送速率的降低而自动地减少，所以，CDMA 和可变速率声音编码是完全匹配

的。与此相反，看一下指定传送时隙的系统，如 TDMA 或 FDMA 等。为了使这种系统得益于数据传送速率的降低，需要进行外部干预，以协调将不用的时隙转让给其它用户。这种系统中固有的延迟意味着只有在长时的语音停顿过程中才可能转让信道。因而，不能充分地得益于声音动作因子。但是，出于上述其它原因，在外部协调下，可变速率声音编码在 CDMA 之外的系统中有用。

在 CDMA 系统中，当需要附加的系统容量时可以略微降低语音质量。简而言之，该声码器可以看作是多个声码器以不同的速率工作，产生不同的语音质量。因而，语音质量可以混合起来，以进一步降低平均数据传送速率。初步的实验显示出，将全速和半速声音编码语音混合起来，亦即最大允许数据速率在 8 Kbps 与 4 Kbps 之间逐帧变化，藉此，得到的语音比半速可变，即最大值为 4 Kbps 情况下质量要好，而差于全速可变，即 8 Kbps 最大值的情况。

众所周知，在绝大多数电话交谈中，每次只有一个人说话。全双工电话线路可以增加一个速率联锁的附加功能。如果线路的一个方向上以最高传送速率进行传送，那么，另一方向就被迫以最低速率传送。链路两个方向上的联锁保证对链路每一方向的平均使用不超过 50%。但是，当信道切断时，例如在动作选通中速率联锁情况下，收听者在对话中无法打断对方的谈话，成为说话者。本发明容易地提供了藉助于设定声音编码速率的控制信号进行速率联锁的能力。

最后应该注意到，利用可变速率声音编码系统，通信信息可以和对语音质量影响极小的语音数据分享信道。例如，一个高速率帧可以分割成两半，一半用于传送较低速率声音数据，另一半用于通信数据。在较佳实施例的声码器中实现了声音传送在全速率和半速率声音编码语音之间仅有轻微的语音质量降低。因而，为了与其它数据分享传送而以较低速率进行语音的声音编码在语音质量上只给用户带来几乎不能察觉的差异。

依照本发明的一个方面，提供了一种为多帧语音信号中一帧语音的背景噪声生成当前估算的方法。所述方法包括以下步骤：

计算一帧语音信号的能量电平；

获得前一帧语音信号的背景噪声估算；和

根据所述能量电平和前一帧语音信号的所述背景噪声估算，生成当前背景噪声估算。

在本发明的方法中，所述生成步骤可以包括限制当前背景噪声估算相对于前一帧语音信号之背景噪声估算的增量，使其低于一预定值。而所述限制步骤可以包

括将当前背景噪声估算限制为小于或等于所述能量电平的值。所述增量可以由一增量百分数构成，或者由一绝对增量构成。

依照本发明的另一方面，提供了一种语音处理器，用于处理包含多个帧的语音信号。所述语音处理器包括：

计算装置，用于计算一帧语音信号的能量电平；

获取装置，用于获得前一帧语音信号的背景噪声估算；和

生成装置，用于根据所述能量电平和前一帧语音信号的所述背景噪声估算生成当前背景噪声估算。

在本发明的语音处理器中，所述生成装置可以包括限制装置，用于限制当前背景噪声估算相对于前一帧语音信号之背景噪声估算的增量，使其低于一预定值。而所述限制装置可以包括将当前背景噪声估算限制为小于或等于所述能量电平的值的装置。所述增量可以由一增量百分数构成，或者由一绝对增量构成。

在本发明的语音处理器中，所述计算装置可以包括第一电路，用于计算一帧语音信号的能量电平。并且，所述获得装置和所述生成装置还可以包括：第二电路，用于计算前一帧语音信号的背景噪声估算，并且对前一帧语音信号的背景噪声估算增加一预定量，以生成一增大的估算值；和第一多路转换器，它与第一和第二电路耦连，用于接收所述增大的估算值以及所述能量电平，并且选择增大的估算值或能量电平，作为语音信号当前帧的背景噪声估算。

本发明的语音处理器还可以包括第一限制器，它与第二电路耦连，用于将所述增大的估算值限制为低于一预定值的值。还可以包括第二限制器，它与第一多路转换器耦连，用于将语音信号当前帧的背景噪声估算限制为小于或等于所述能量电平的值。

在本发明的语音处理器中，第二电路可以包括：第一加法器，用于将一标称常数与前一帧语音信号的背景噪声估算相加，生成一绝对增大的估算值；第二乘法器，用于将前一帧语音信号的背景噪声估算乘以一个略大于1的常数，生成一百分数增大的估算值；第二多路转换器，它与第一加法器和所述第二乘法器耦连，用于接收绝对增大的估算值和百分数增大的估算值；第三电路，它与第一加法器、第二乘法器和第二多路转换器耦连，用于控制第二多路转换器，选择绝对增大的估算值和百分数增大的估算值中的较大者，作为增大的估算值。

## 附图说明

从下文中与附图相结合的详细说明中，本发明的特征、目的和优点将更为显而易见，在所有附图中，类似的标号标志相应的对象，其中：

图 1a-1e 以图线形式示出声码器在各种速率下的分析帧和子帧；

图 2a-2d 是一系列表示声码器在不同速率下输出位分布的图表；

图 3 是一例示编码器的大致框图；

图 4 是编码器的流程图；

图 5 是例示译码器的大致框图；

图 6 是译码器流程图；

图 7 是编码器更详细的功能性框图；

图 8 是一例示汉明窗口和自相关子系统的框图；

图 9 是一例示速率确定子系统的框图；

图 10 是一例示 LPC 分析子系统的框图；

图 11 是一例示 LPC 至 LSP 转换子系统的框图；

图 12 是一例示 LPC 量化子系统的框图；

图 13 是例示 LSP 插值及 LSP 至 LPC 转换子系统的框图；

图 14 是用于音调检索的自适应码书的框图；

图 15 是编码器的译码器的框图；

图 16 是音调检索子系统的框图；

图 17 是码书检索子系统的框图；

图 18 是数据压缩子系统的框图；

图 19 是译码器更详尽的功能性框图；

图 20a-20d 是显示译码器在不同速率下接收到的参数和子帧译码数据的图表；

图 21a-21d 是进一步显示译码器在特定场合下接收到的参数及子帧译码数据的图表；

图 22 是 LSP 反量化子系统的框图；

图 23 是带有后置滤波及自动增益控制的译码器更详细的框图；

图 24 是自适应亮度滤波器特性的图线。

## 具体实施方式

根据本发明，利用众所周知的技术对诸如语音和/或背景噪声之类声音进行取

样并数字化。例如，模拟信号可以用标准的 8 位/ $\mu$  律格式转换成数字格式，然后进行  $\mu$  律/标准编码转换。在另一方式下，模拟信号可以直接转换成标准脉冲编码调制(PCM)格式的数字形式。这样，在该较佳实施例中，每一取样由一 16 位的数据字表示。众多取样被组织成输入数据的帧，每一帧由预定量的取样组成。在现在揭示的实施例中，就 8 KHz 的取样频率进行考虑。对于 8 KHz 取样频率，每帧由 160 个取样或 20 msec 的语音组成。应该理解，其它取样频率和帧尺寸也可以采用。

声音编码领域包括许多不同的语音编码技术，其中之一是 CELP 编码技术。在前文提到论文“一种 4.8 Kbps 的编码激励线性预测编码器”中描述了这种 CELP 编码技术的概要。本发明采用一种形式的 CELP 编码技术，以提供编码语音数据的可变速率，其中，LPC 分析是对一定量的取样进行的，而音调和码书检索则是根据传送速率对变量的取样进行的。在原理上，参照图 3 和 5 讨论应用于本发明中的 CELP 编码技术。

在本发明的该较佳实施例中，语音分析帧为 20 ms 长，意味着所提取的参数以每秒 50 次的脉冲进行传送。此外，数据传送速率大致从 8 Kbps 变化至 4 Kbps、2 Kbps 和 1 Kbps。在全速率(亦称为速率 1)下，数据传送处于 8.55 Kbps 的速率下，为每一帧而编码的参数用了 171 位，其中包括一个 11 位的内部 CRC(循环冗余检验)。除去 CRC 位，速率将是 8 Kbps。在半速率(亦称为速率 1/2)下，数据传送以 4 Kbps 的速率进行，每一帧的编码参数用了 80 位。在四分之一速率(亦称为速率 1/4)下，数据传送为 2 Kbps 的速率，每帧的编码数据用了 40 位。在八分之一速率(亦称为速率 1/8)下，数据传送速率略低于 1 Kbps，各帧的编码参数使用 16 位。

图 1 例示出语音数据的一个分析帧 10，以及它与在 LPC 分析中采用的汉明窗口 12 的关系。不同速率下的 LPC 分析帧、音调和码书子帧以图表形式在图 2a-2d 中示出。应该理解，LPC 分析帧在所有速率下是大小相同的。

现在参见附图，具体讲是图 1a，LPC 分析是采用帧 10 的 160 个语音数据取样完成的，它们由汉明窗口 12 窗口化。如图 1a 所示，各个帧中的取样  $s(n)$  记数为 0-159。汉明窗口 12 是这样定位的，偏离帧 10 中的 60 个取样。该汉明窗口 12 从当前的数据帧 10 的第 60 个取样  $s(59)$  开始，延续进入下一数据帧 14，包括其第 59 个取样  $s(58)$ 。因而，为当前帧帧 10 产生的加权数据也含有以下一帧帧 14 的数据为依据的数据。

根据数据传送速率，对于图 1b-1e 中示出的数据帧 10 的不同子帧进行多次检索，以计算出音调滤波器和码书激励参数。应该理解成，在本实施例中，给帧 10

只选择一个速率，因而音调和码书检索在不同大小的子帧中对应于下面说明的选择速率进行。然而为了示意，在图 1b-1e 中示出在本较佳实施例的各种允许速率下对帧 10 的音调和码书检索的子帧结构。

如图 1a 所示，在所有速率下，对每一帧 10 都有一次 LPC 计算。如图 1b 所示，在全速率下，对应每一音调子帧 16 有两个码书子帧 18。在全速率下有四次音调更新，四个音调子帧 16 各一次，每次长为 40 个取样(5 ms)。此外，在全速率下，有八次码书更新，八个码书子帧 18 各一次，每次长为 20 个取样(2.5 ms)。

在半速率下，如图 1c 所示，对应每一音调子帧 20，有两个码书子帧 22。音调更新两次，两个音调子帧 20 各一次，而码书更新四次，四个码书子帧 22 各一次。在四分之一速率下，如图 1d 所示，对单个音调子帧 20 有两个码书子帧 26。对音调子帧 24 进行一次音调更新，而码书则为两次，两个码书子帧 26 各一次。如图 1e 所示，在八分之一速率下，不确定音调，在对应于帧 10 的帧 28 中码书只更新一次。

另外，虽然 LPC 系数每帧只计算一次，但是，在线频谱对(LSP)表示式中，利用从前一帧得到的 LSP 频率对它们线性内插达四次，以逼近汉明窗口位于每一帧中心时的 LPC 分析的结果。其例外情况是在全速率下对码书子帧 LPC 系数不被内插。LSP 频率计算的细节在此后说明。

在较低速率下音调和码书检索较少地进行，此外，给 LPC 系数的传送也分配少量位数。在不同速率下配置的位的数量示出在图 2a-2d 中。图 2a-2d 中的每一个表示出分配给语音的每一 160 取样帧的声码器编码数据位的数量。在图 2a-2d 中，各个 LPC 块 30a -30d 中的数量是在相应速率下用于短期 LPC 系数编码的位数。在本较佳实施例中，用于 LPC 系数编码的位数在全、半、四分之一和八分之一速率下分别是 40、20、10 和 10。

为了采用可变速率编码，LPC 首先变换成线频谱对(LSP)，得到的 LSP 频率用 DPCM 编码器单独进行编码。LPC 的阶数为 10，因而有 10 个 LSP 频率和 10 个独立的 DPCM 编码器。对 DPCM 编码器的位分配按照表 I 进行。

表 I

	DPCM 编码器号									
	1	2	3	4	5	6	7	8	9	10
速率 1	4	4	4	4	4	4	4	4	4	4
速率 1/2	2	2	2	2	2	2	2	2	2	2
速率 1/4	1	1	1	1	1	1	1	1	1	1
速率 1/8	1	1	1	1	1	1	1	1	1	1

在编码器和译码器中, LSP 频率都在用于音调和码书检索之前转换回 LPC 滤波器系数。

关于音调检索, 图 2a 示出在全速率下, 音调更新计算四次, 对语音帧的每四分之一长度各一次。对于全速率下的每次音调更新, 新的音调参数编码用了 10 位。如图 2b-2d 所示, 在其它速率下, 音调更新的次数不相同。当速率降低时, 更新的次数也减少。图 2b 示出在半速率下的音调更新, 它计算两次, 对语音帧的每一半各进行一次。与此类似, 图 2c 示出在四分之一速率下的音调更新, 对每个语音帧计算一次。同全速率一样, 每次半速率和四分之一速率音调更新也用了 10 位来将新的音调参数编码。但在八分之一速率下, 如图 2d 所示, 不计算音调更新, 因为在没有或很少语音以及音调冗余不存在的时候, 就采用该速率来对各个帧编码。

在每 10 位音调更新值中, 7 位表示音调滞后, 而 3 位表示音调增益。音调滞后限制在 17 和 143 之间。音调增益线性是量化至 0 和 2 之间, 以便用 3 位的值来表示。

关于码书检索, 图 2a 示出, 在全速率下码书更新计算八次, 语音帧的每八分之一各一次。全速率下的每次码书更新用 10 位来进行新的码书参数编码。图 2b-2d 中示出, 在其它速率下进行不同次数的码书更新。然而, 随着速率降低, 码书更新次数也减少。图 2b 示出半速率下的码书更新, 它计算四次, 对语音帧的每四分之一各一次。图 2c 示出四分之一速率下的码书更新, 计算两次, 对语音帧的每一半各一次。同全速率下一样, 每个半速率和四分之一速率音调更新用 10 位来将新码书参数编码。最后, 图 2d 示出八分之一速率下的码书更新, 对每一语音帧进行一次计算。要注意到, 在八分之一速率下, 传送了 6 位, 其中两位代表码书增益, 其余四位是随机位。后文将进一步讨论码书更新的位分配。

为码书更新而分配的位数表示将音调预测余量矢量量化所需的数据位。在全、半和四分之一速率下, 每次码书更新由 7 位码书索引加 3 位码书增益总共 10 位数

据组成。码书增益是用工作在对数区域的差分脉码调制(DPCM)编码器进行编码的。对于八分之一速率，虽然也可以采用类似的位分布，但最好用一不同的方案。在八分之一速率下，码书增益用两位表示，而四个随机产生的位则与接收到的数据一起使用，作为替代码书的伪随机数发生器的种子。

参见图 3 所示的编码器框图，LPC 分析是以开环方式进行的。从输入语音取样  $s(n)$  的各个帧中计算出 LPC 系数( $\alpha_1-\alpha_{10}$ )，如后文中说明，由 LPC 分析/量化器 50 来完成，已供共振峰合成滤波器 60 使用。

然而，音调检索的计算是以闭环方式进行的，通常这称作分析-合成方法。但是，在实际应用中，采用一种新颖的闭环/开环混合技术来进行音调检索。在音调检索中，通过选取使输入语音与合成语音的均方之差为最小的参数而进行编码。为了简化这部分的讨论，速率的问题不作考虑。但在下文中将更详细地讨论选出的速率对音调和码书检索的影响。

在图 3 所示的初步实施例中，感知加权滤波器 52 由下列等式特征化：

$$W(z) = \frac{A(z)}{A(z/\mu)} \quad (1)$$

其中

$$A(z) = 1 - \sum_{i=1}^{10} \alpha_i z^{-i} \quad (2)$$

它是共振峰预测滤波器， $\mu$  是感知加权参数，在本实施例中， $\mu = 0.8$ 。音调合成滤波器 58 由下列等式特征化：

$$\frac{1}{P(z)} = \frac{1}{1 - bz^{-L}} \quad (3)$$

下文讨论共振峰合成滤波器 60，它是加权滤波器，由下列等式特征化：

$$H(z) = \left(\frac{1}{A(z)}\right) W(z) = \frac{1}{A(z/\mu)} \quad (4)$$

输入语音取样  $S(n)$  由感知加权滤波器 52 加权，所产生的加权语音取样  $X(n)$  作为和输入提供给加法器 62。感知加权用于给低信号功率频率中的错误加权。在这些低信号功率频率中，噪声更易察觉。合成的语音取样  $X'(n)$  从共振峰合成滤波器 60 输出作为差输入加到加法器 62 与  $X(n)$  作减法运算。加法器 62 输出的取样值之差输入至均方差(MSE)单元 64，它们被平方后累加。WSE 单元 64 的结果提供给最小化单元 66，它产生音调滞后 L、音调增益 b、码书索引 I 和码书增益 G 的数值。

在最小化单元 66 中  $P(z)$  中的音调滞后参数 L 的所有可能值与来自乘法器 56

的数值  $c(n)$ 一起输入音调合成滤波器 58。在音调检索过程中，码书没有任何贡献，即  $c(n)=0$ 。由最小化单元 66 选出使输入语音与合成语音的加权差最小的  $L$  和  $b$  的数值。音调合成滤波器 58 产生并输出  $P(n)$  值给共振峰合成滤波器 60，一旦找到音调滤波器的音调滞后  $L$  和音调增益  $b$ ，就以类似方式进行码书检索。

应该理解，图 3 是本发明所采取的分析-合成方法的原理表示。在本发明的示例应用中，这些滤波器不应用在典型的闭环反馈结构中。在本发明中，在检索过程中反馈联系被切断，而代之以开环共振峰余量，其细节在下文中提供。

最小化单元 66 然后产生码书索引  $I$  和码书增益  $G$  的值。码书 54 的输出值是按照码书索引  $I$  从许多随机高斯矢量值中选出的，在乘法器 56 中乘以码书增益  $G$ ，产生音调合成滤波器 58 中所用的  $c(n)$  值序列。选取使均方差为最小的码书索引  $I$  和码书增益  $G$  用于传送。

应注意，感知加权  $W(z)$  既通过感知加权滤波器 52 加在输入语音上，也通过共振峰合成滤波器 60 中包含的加权功能加在合成语音上。因而，共振峰合成滤波器 60 实际上是加权后的共振峰合成滤波器，它把等式 1 的加权功能与典型的共振峰预测滤波器特性  $\frac{1}{A(z)}$  组合起来，产生等式 3 的加权共振峰合成功能。

应该理解的是，换种方式，感知加权滤波器 52 可以放在加法器 62 和 MSE 单元 64 之间。在这种情况下，共振峰合成滤波器 60 具有通常的滤波器特性  $\frac{1}{A(z)}$ 。

图 4 示出用图 3 的编码器对语音编码时所需步骤的流程。为了说明，涉及速率判定的步骤包含在图 4 的流程中。在步骤 80，从取样电路获取数字化语音取样，在步骤 82 中计算出 LPC 系数。在 LPC 系数计算中用到了汉明窗口和自相关技术。在步骤 84，在本实施例中依据帧能量对重要的帧作出初步的速率确定。

为了有效地以少量位数将 LPC 系数编码，在步骤 86，LPC 系数转成线频谱对(LSP)频率，然后在步骤 88 量化，以便传送。一种选择方案是，在步骤 90 可以进行附加的速率确定，在步骤 92，如果认为对初始速率作的 LSP 量化不充分，则可增加速率。

在步骤 94，对于处在分析中的语音帧的第一个音调子帧，LSP 频率经内插后转换成 LPC 系数，用于进行音调检索。在音调检索中，码书激励设置为零。在步骤 96 和 98 的作为由先前讨论的合成方法所进行的一种分析的音调检索中，就每一可能的音调滞后  $L$ ，将合成语音与原始语音比较。对于每一  $L$  值，确定一个取整数值的最佳音调增益  $b$ 。在  $L$  和  $b$  值的组合中，最佳的  $L$  和  $b$  值组合给出合成语

音与原始语音之间的最小感知加权均方误差。在所确定的音调子帧的 L、b 最佳值中，在步骤 100 将 b 值量化，以便与对应的 L 值一起传送。在音调检索的另一种实现方法中，b 值可以作为音调检索一部分进行量化，这些量化值用于进行音调检索。因而，在这种实现方式中消除了在音调检索之后选中的 b 值在步骤 100 的量化要求。

对于处在分析中的语音帧的第 1 码书子帧，在步骤 102，LSP 频率经内插后转换成 LPC 系数，以用于进行码书检索。然而，在本实施例中，在全速率下，LSP 频率仅仅内插至音调子帧水平。由于在各个频率下音调和码书子帧的大小有差异，在音调检索之外，对码书检索也进行这种内插和转换步骤。只有速率 1/8 是例外，因为不计算音调数据，所以这一问题悬而未决。在步骤 104 和 106 的码书检索中，在音调合成滤波器中用到音调滞后 L 和音调增益 b 的最佳值，使得就每一个可能的码书索引 I，将会成语音与原始语音作比较。对于每个 I 值，确定一个取整数值的最佳码书增益 G。I 和 G 值组合中，最佳的 I 和 G 值组合给出成语音与原始语音之间最小的误差。在给码书子帧确定的最佳值 I 和 G 中，G 值在步骤 108 被量化，以便与对应 I 值一起传送。在码书检索的另一种变换应用中，G 值可以量化成码书检索的一部分，这些量化值用于进行码书检索。在这种变换应用中，省除了码书检索之后在步骤 108 将选中的 G 值量化的要求。

在码书检索之后，编码器中的一个译码器在 I、G、L 和 b 的最佳值下运行。编码中的译码器的运行使得编码滤波存储器重组，以用于以后的子帧。

然后在步骤 110 进行验证，以确定刚完成分析的该码书子帧是否对应于已进行音调检索的音调子帧的码书子帧组中的最后一个码书子帧。换句话说，确认是否还有其他对应于该音调子帧的码书子帧。在本实施例中，对应每一音调子帧有两个码书子帧。如果确认有另一个对应于该音调帧的码书子帧，则对该码书子帧重复步骤 102-108。

如果不再有对应于该音调帧的码书子帧，则在步骤 112 进行验证，以确定处于分析中的语音帧中是否有其他任何音调子帧存在。如果在当前处于分析中的语音帧中有另一个音调子帧存在，则对每个音调子帧和对应的码书子帧重复步骤 94-110。当目前处于分析中的语音帧的计算都已完成之后，在步骤 114，将代表语音帧的 LPC 系数的数值，即各音调子帧的音调滞后 L 和增益 b 和各码书子帧的码书索引 I 和增益 G 进行压缩，以便传送。

参见图 5，其中示出一个译码器框图，按收到的 LPC 系数( $\alpha_i$ )，音调滞后和增

益(L&b)、码书索引和增益(I&G)用来合成语音。同图 3 一样，为了简化讨论，在图 5 中也不考虑速率信息。数据速率信息可以作为辅助信息传送，在某些情况下，可以在信道解调级导出。

译码器由码书 130 组成，得到所接收到的码书索引，或者在八分之一速率下得到随机种子。码书 130 的输出提供给乘法器 132 的一个输入端，而乘法器 132 的另一输出端接收码书增益 G。乘法器 132 的输出与音调滞后 L 和增益 b 一起供给音调合成滤波器 134。音调合成滤波器 134 的输出与 LPC 系数  $\alpha_i$  一起供给共振峰合成滤波器 136。共振峰合成滤波器 136 的输出供给自适应后置滤波器 138，其中滤波和输出的是再生语音。后文中将讨论，有一种形式译码器是在编码器中实现的。编码器的译码器不包括自适应后置滤波器 138，但是包括一个感知加权滤波器。

图 6 是与图 5 译码器的动作相对应的流程。在步骤 150 译码器中，根据接收到的参数重建语音。具体来讲，在步骤 152，接收到的码书索引值输入进码书，产生码矢量或码书输出值。在步骤 154，乘法器接收该码矢量和码书增益 G，并将这些值相乘，得出的信号提供给音调滤波器。应该注意到，码书增益 G 是通过将收到的 DPCM 参数译码和反量化而得以重建的。在步骤 156，音调合成滤波器得到所接收到的音调滞后 L 和增益 b，以及乘法器输出信号，以对乘法器输出进行滤波。

音调合成滤波器对码书矢量滤波后得到的数值输入共振峰合成滤波器。在步骤 158，LPC 系数  $\alpha_i$  与供给共振峰合成滤波器，用于对音调合成滤波器的输出信号滤波。在译码器中，将收到的参数译码成量化 LSP 频率，将 LSP 频率反量化并转换成 LPC 系数  $\alpha_i$ ，藉此重建 LPC 系数，以进行内插。在步骤 160，共振峰合成滤波器的输出供给自适应后置滤波器，量化噪声在其中被掩盖，重建后的语音得到增益控制。在步骤 162，重建后的语音输出，以转换成模拟形式。

现在参见图 7a 和 7b 的框图，说明本发明的语音编码技术的进一步细节。在图 7a 中，各个数字化语音取样的帧提供给一个汉明窗口系统 200，在该子系统中，输入语音在自相关子系统 202 进行自相关系数计算之前被窗口化。

图 8 的例示应用中示出了汉明窗口子系统 200 和自相关子系统 202。汉明窗口子系统 200 由通常是  $80 \times 16$  位的 ROM 的查算表 250 和乘法器 252 组成。在每一速率下，语音的窗口中心处于 160 个取样长的分析帧的第 139 和 140 个取样之间。这样，用于计算自相关系数的窗口偏移分析帧 60 个取样。

因为汉明窗口是中心对称的，所以窗口化是利用包含有 160 个  $W_H(n)$  值中的

80 个值的 ROM 表进行的。汉明窗口的偏移是通过将 ROM 的地址指针相对于分析帧的第一个取样偏移 60 个位置而实现的。这些值由乘法器以单精度与对应的输入语音取样相乘。在分析窗口中，将  $s(n)$  作为输入语音信号，因而窗口化的语音信号  $S_w(n)$  被定义为：

$$S_w(n) = s(n+60)W_H(n) \quad 0 \leq n \leq 79 \quad (5)$$

和

$$S_w(n) = s(n+60)W_H(159 - n) \quad 80 \leq n \leq 159 \quad (6)$$

查算表 250 内容的 16 进制例示值在表 II 中给出。这些数值解释成具有 14 个小数位的二进制补码，表格要从左到右、从上到下阅读。

表 II

0x051f	0x0525	0x0536	0x0554	0x057d	0x05b1	0x05f2	0x063d
0x0694	0x06f6	0x0764	0x07dc	0x085e	0x08ec	0x0983	0x0a24
0x0ad0	0x0b84	0x0c42	0x0d09	0x0dd9	0x0eb0	0x0f90	0x1077
0x1166	0x125b	0x1357	0x1459	0x1560	0x166d	0x177f	0x1895
0x19af	0x1acd	0x1bee	0x1d11	0x1e37	0x1f5e	0x2087	0x21b0
0x22da	0x2403	0x252d	0x2655	0x277b	0x28a0	0x29c2	0xae1
0x2bfd	0x2d15	0x2e29	0x2f39	0x3043	0x3148	0x3247	0x333f
0x3431	0x351c	0x3600	0x36db	0x37af	0x387a	0x393d	0x39f6
0x3aa6	0x3b4c	0x3be9	0xec7b	0x3d03	0x3d80	0xdf3	0x3e5b
0x3eb7	0x3f09	0x3ff4	0x3f89	0x3fb8	0x3fdb	0x3ff3	0x3ff

自相关子系统 202 由寄存器 254、多路转换器 256、移位寄存器 258、乘法器 260、加法器 262、循环移位寄存器 264 和缓存器 266 组成。窗口化的语音取样  $S_w(n)$  每隔 20 ms 运算一次并锁存在寄存器 254 中。在 LPC 分析帧的第一个取样  $S_w(0)$  上，移位寄存器 258 和 264 中复位到 0。在每一个新的取样  $S_w(n)$  上，多路转换器 256 接收一个新的取样选择信号。该信号使取样能够从寄存器 254 进入多路转换器。新的取样  $S_w(n)$  也提供给乘法器 260。与取样  $S_w(n-10)$  相乘，取样  $S_w(n-10)$  处在移位寄存器 258 的末位 SR10 上。得到的数值在加法器 262 中与循环移位寄存器 264 的末位 CSR11 上的数值相加。

移位寄存器 258 和 260 每收到一次时钟脉冲，就在移位寄存器 258 的首位 SR1 中用  $S_w(n)$  替换  $S_w(n-1)$ ，并替换先前在位置 CSR10 中的数值。在移位寄存器 258 有时钟脉冲时，新脉冲选择信号从多路转换器 256 的输入端中去除，使得当前处在移位寄存器 260 位 SR10 中的取样  $S_w(n-9)$  被允许进入多路转换器 256。在循

环移位寄存器 264 中，先前在位 CSR11 上的数值移入首位 CSR1 中。新取样选择信号从多路转换器中去除后，移位寄存器 258 置位，以和循环移位寄存器 264 一样，进行数据的循环移位。

对每个取样，移位寄存器 258 和 264 总共都得到 11 次时钟脉冲，因而执行 11 次乘积/累加动作。在对 160 个取样进行时钟输入后包含在循环移位寄存器 264 中的自相关结果作为 R(0)-R(10) 值时钟输入缓存器 266。所有移位寄存器复位到零，对窗口化语音取样的下一个帧重复此过程。

回头再参见图 7a，对语音帧计算好自相关系数后，速率确定子系统 204 和 LPC 分析子系统 206 就利用该数据来分别计算帧数据速率和 LPC 系数。由于这些运算是彼此独立的，所以，它们可以以任何顺序甚至同时计算。这里为了说明，而首先讨论速率确定。

速率确定子系统 204 有两个功能；(1)确定当前帧的速率，(2)重新估算背景噪声水平。当前分析帧的速率最初是依据当前帧的能量、背景噪声水平的先前估值、先前速率和来自一控制微机的速率指令确定的。而新的背景噪声水平是利用背景噪声水平的先前估值和当前帧能量估算出的。

本发明将自适应阈值技术用于速率确定。当背景噪声变化时，用于选择速率的阈值也变化。在本实施例中，计算三个阈值，以确定初始的速率选择  $RT_p$ 。这些阈值是背景噪声先前估值的二次函数，示出如下：

$$T1(B) = -5.544613(10^{-6})B^2 + 4.047152B + 363.1293 \quad (7)$$

$$T2(B) = -1.529733(10^{-5})B^2 + 8.750045B + 1136.214 \quad (8)$$

和

$$T3(B) = -3.957050(10^{-5})B^2 + 18.89962B + 3346.789 \quad (9)$$

其中，B 是背景噪声的先前估值。

帧能量与三个阈值 T1(B)、T2(B) 和 T3(B) 作比较。如果帧能量低于所有三个阈值，则选择最低的传送速率(1 Kbps)，即  $RT_p=4$  速率 1/8。如果帧能量低于两个阈值，则选择第二个传送速率(2Kbps)，即  $RT_p=3$ ，速率 1/4。如果帧能量仅低于一个阈值，就选择第三个传送速率(4Kbps)，即  $RT_p=2$ ，速率 1/2。如果帧能量高于所有阈值，则选取最高传送速率(8Kbps)，即  $RT_p=1$ ，速率 1。

此后，可以依据前一帧的最终速率  $RT_r$  来修改初始速率  $RT_p$ 。如果初始速率  $RT_p$  低于前一帧的最终速率减 1，即( $RT_r-1$ )，则设定一个中间速率  $RT_m=(RT_r-1)$ 。在发生高能量信号向低能量信号的转换时，这种修改过程使得速率缓慢地降低。然而，

如果初始速率选择等于或大于前一速率减 1，即( $RT_r - 1$ )，则中间速率  $RT_m$  设置成与初始速率  $RT_p$  相同，即  $RT_m = RT_p$ 。在这种场合，当发生低能量信号向高能量信号的转换时，速率会立即提高。

最后，中间速率  $RT_m$  由来自微机的速率约束指令作进一步修改。如果速率  $RT_m$  大于微机所允许的最高速率，则设定初始速率  $RT_i$  为最高允许值。与此类似，如果中间速率  $RT_m$  小于微机允许的最低速率，则初始速率  $RT_i$  设为最低允许值。

在某些情况下，可能会要求以微机决定的速率对所有语音编码。速率约束指令通过将最高和最低允许速率设为期望速率，而将帧速率设置成期望速率。速率约束指令可用于特殊的速率控制场合，如用于速率联锁和模糊及突发传送，这些都在后文说明。

图 9 给出速率确定算法的例示执行过程。要起动运算，寄存器 270 要预先装入提供给加法器 272 的数值 1。循环移位寄存器 274、276 和 278 分别装载二次阈值等式(7)-(9)的第 1、2 和 3 系数。例如，循环移位寄存器 274 的末位、中位和首位分别装载计算  $T_1$ 、 $T_2$  和  $T_3$  的等式的第 1 系数。与此类似，循环移位寄存器 276 的末位、中位和首位分别装入计算  $T_1$ 、 $T_2$  和  $T_3$  的等式的第 2 系数。最后，循环移位寄存器 278 的末位、中位和首位分别装入计算  $T_1$ 、 $T_2$  和  $T_3$  的等式的常数项。在各个循环移位寄存器 274、276 和 278 中，数值从末位输出。

在计算第 1 阈值  $T_1$  时，先前帧的背景噪声估值  $B$  在乘法器 280 中自乘。得到的  $B^2$  值乘以第 1 系数  $-5.544613(10^{-6})$ ，该系数从循环移位寄存器 274 的末位输出。得到的结果在加法器 286 中加上来自乘法器 284 的背景噪声  $B$  与循环移位寄存器 276 末位输出的第 2 系数  $4.047152$  的乘积。然后，加法器 286 的输出值在加法器 288 中加上循环移位寄存器 278 的末位输出的常数项  $363.1293$ 。加法器 288 的输出即为  $T_1$  的计算值。

加法器 288 输出的  $T_1$  计算值在加法器 290 中减去帧能量值  $E_f$ ，在本实施例中， $E_f$  是线性区域中的数值  $R(0)$ ，由自相关子系统提供。

换种方式，帧能量  $E_f$  也可以以分贝数在对数域中表示，它近似为由有效窗口长度归一化后的第一自相关系数  $R(0)$  的对数值：

$$E_f = 10 \log_{10} \frac{R(0)}{L_A/2} \quad (10)$$

其中， $L_A$  是自相关窗口长度。应该理解到，声音动作也可以从包括音调预测增益或共振峰预测增益  $G_a$  在内的其它各种参数中测得：

$$G_a = 10 \log_{10} \frac{E^{(10)}}{E^{(0)}} \quad (11)$$

其中,  $E^{(10)}$ 是第 10 次叠代后的预测残留能量, 而  $E^{(0)}$ 是初始 LPC 预测残留能量, 后文中说明, 在 LPC 分析中与  $R(0)$ 相等。

利用比较器或限幅器 292, 从加法器 290 的输出中提取二进制补码差值符号位的补码, 并提供给加法器 272, 与寄存器 270 的输出相加。这样, 如果  $R(0)$ 与  $T_1$  之差为正, 则寄存器 270 增量 1。如果差值为负, 寄存器 270 保持原样。

然后, 循环寄存器 274、276 和 278 进行循环, 因而,  $T_2$  的等式(8)的系数出现在其输出端。同讨论阈值  $T_1$  的过程一样, 重复计算阈值  $T_2$  和将它与帧能量比较的过程。然后, 循环寄存器 264、276 和 278 再次循环, 使  $T_3$  的等式(9)的系数出现在其输出端。阈值  $T_3$  的计算和与帧能量的比较与上文中说明的相同。在完成所有三个阈值的计算和比较后, 寄存器 270 包含有初始速率估算值  $RT_i$ 。初始速率估算值  $RT_p$  提供给速率缓降逻辑 294。存储在寄存器 298 中来自 LSP 频率量化子系统的先前帧最终速率  $RT_r$  也供给逻辑 294。逻辑 296 计算数值( $RT_r - 1$ ), 并提供初始速率估算值  $RT_p$  和数值( $RT_r - 1$ )中的较大者作为输出。数值  $RT_m$  给到速率限制逻辑 296。

前面说到, 微机提供速率限制指令给声码器, 具体讲是给逻辑 296。在用到数字信号处理器时, 该指令在编码过程的 LPC 分析部分结束之前送入逻辑 296 中。逻辑 296 确认该速率不超过速率限制值, 并在  $RT_m$  值超出限制值时修改它。如果  $RT_m$  值在允许速率范围内, 它就作为初始速率值  $RT_i$  从逻辑 296 中输出。初始速率值  $RT_i$  从逻辑 296 输出到图 7a 的 LSP 量化子系统 210 中。

前面提到的背景噪声估算值用于计算自适应速率阈值。先前帧的背景噪声估算值  $B$  用于建立当前帧的速率阈值。然而, 对于每一个帧, 背景噪声估算值被更新后用于为下一个帧确定速率阈值。新的背景噪声估算值  $B'$  是根据先前帧的背景噪声估算值  $B$  和当前帧能量  $E_f$  在当前帧中确定的。

在确定用于下一帧的新的背景噪声估算值  $B'$ (作为先前帧背景噪声值  $B$ )时, 要计算两个数值。第 1 个值  $V_1$  仅仅是当前帧能量  $E_f$ 。第 2 个值  $V_2$  是  $B+1$  和  $KB$  中的较大者,  $K=1.00547$ 。为了防止第 2 个值变得太大, 将其限制在一个很大的常数  $M=160,000$  之下。选择两个值  $V_1$  和  $V_2$  中小的一方作为新的背景噪声估算值  $B'$ 。

数学表示为:

$$V_1=R(0) \quad (12)$$

$$V_2 = \min(160000, \max(KB, B+1)) \quad (13)$$

新的背景噪声估算值  $B'$  是：

$$B' = \min(V_1, V_2) \quad (14)$$

其中， $\min(x, y)$  是  $x, y$  中的最小值，而  $\max(x, y)$  是  $x, y$  中的最大值。

图 9 进一步示出了背景噪声估算算法的例示执行过程。第一个值  $V_1$  仅仅是直接提供给多路转换器 300 一个输入端的当前帧能量  $E_f$ 。

第二个值  $V_2$  从首先算出的数值  $KB$  和  $B+1$  中计算得到。在计算数值  $KB$  和  $B+1$  时，存储在寄存器 302 中的先前帧背景噪声估算值  $B$  输出到加法器 304 和乘法器 306 中。应该注意到，存储在寄存器 302 中的用于当前帧的先前帧背景噪声估算值  $B$  与在先前帧中计算出的新的背景噪声估算值  $B'$  相等。加法器 304 还得到一个输入值 1 与数值  $B$  相加，从而产生  $B+1$ 。乘法器 304 还得到一个输入值  $K$  与数值  $B$  相乘，从而产生  $KB$ 。 $B+1$  和  $KB$  分别从加法器 304 和乘法器 306 输出到多路转换器 308 和加法器 310 的各个输入端。

加法器 310 和比较器或限幅器 312 用于选择  $B+1$  和  $KB$  中较大的一方。加法器 310 从  $KB$  中减去  $B+1$  并将结果提供给比较器或限幅器 312。限幅器 312 提供一个控制信号给多路转换器 308，选择  $B+1$  和  $KB$  较大的一方作为输出。选出的  $B+1$  或  $KB$ (选中项)从多路转换器 308 输出到限幅器 314，限幅器 314 是饱和型限幅器，当选中项低于常数值  $M$  时，输出选中项或者在选中项超出数值  $M$  时输出数值  $M$ 。限幅器 314 的输出作为第二输入供给多路转换器 300 并作为加法器 316 的一个输入。

加法器 316 在另一个输入端还接收到帧能量值  $E_f$ 。加法器 316 和比较器或限幅器 318 用于选择数值  $E_f$  和限幅器 314 的输出中较小的一方。加法器 316 从限幅器 314 的输出值中减去帧能量值，并将结果提供给比较器或限幅器 318。限幅器 318 向多路转换器 300 提供一个控制信号。用于选择  $E_f$  值和限幅器 314 的输出值中较小的一方。多路转换器 300 输出的选中数值作为新的背景噪声估算值  $B'$  提供给寄存器 302，存储后作为当前帧背景噪声估算值  $B$  用在下一帧中。

再参见图 7，各自相关系数  $R(0)-R(10)$  从自相关子系统 202 输出到 LPC 分析子系统 206 中。在 LPC 分析子系统 206 中计算出的 LPC 系数用在感知加权滤波器 52 和共振峰合成滤波器 60 中。

LPC 系数可以通过采用德宾(Durbin)递推式的自相关方法获得，该方法在“语音信号数字化处理”(赖宾纳和谢弗，普蓝蒂斯-豪公司 1978 年出版)一书中讨论过。

这种技术是用于获得 LPC 系数的高效计算方法。该算法可以用下列等式表示：

$$E^{(0)} = R(0), i=1; \quad (15)$$

$$k_i = \{R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)\}/E^{i-1}; \quad (16)$$

$$\alpha_i^{(i)} = K_i; \quad (17)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \quad 1 < j < i-1; \quad (18)$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \quad (19)$$

如果  $i < 10$ , 则进到等式(16), 并取  $i=i+1$  (20)

十个 LPC 系数标为  $\alpha_j^{(10)}$ ,  $1 \leq j \leq 10$ 。

在将 LPC 系数编码之前必须确保滤波器的稳定性。滤波器的稳定性是这样实现的, 将滤波器的极点沿径向向内微量移动, 这一变化减小峰值频率响应的幅值而扩展峰值的带宽。这种技术作为带宽扩展是众所周知的, 在拖苦拉等人所著的“PARCOR 语音分析-合成中的频谱平滑”一文(ASSP 学报)中有进一步说明。在现情况下通过按比例改变各 LPC 系数, 可以迅速地完成带宽扩展。因而如表III中所得到的 LPC 都乘以对应的十六进制数值以产生 LPC 分析子系统 206 的最终输出 LPC 系数  $\alpha_1-\alpha_{10}$ 。应该注意到, 表III中列出的数值是十六进制形式的。用二进制补码标记时有十五位。在这种形式下数值  $0 \times 8000$  表示-1.0, 向数值  $0 \times 7333$ (或 29491)表示 0.899994 等于 29491/32768。

表 III

$$\alpha_1=\alpha_1^{(10)} \cdot 0x7333$$

$$\alpha_2=\alpha_2^{(10)} \cdot 0x67ae$$

$$\alpha_3=\alpha_3^{(10)} \cdot 0x5d4f$$

$$\alpha_4=\alpha_4^{(10)} \cdot 0x53fb$$

$$\alpha_5=\alpha_5^{(10)} \cdot 0x4b95$$

$$\alpha_6=\alpha_6^{(10)} \cdot 0x4406$$

$$\alpha_7=\alpha_7^{(10)} \cdot 0x3d38$$

$$\alpha_8=\alpha_8^{(10)} \cdot 0x3719$$

$$\alpha_9=\alpha_9^{(10)} \cdot 0x3196$$

$$\alpha_{10}=\alpha_{10}^{(10)} \cdot 0x2ca1$$

这些运算最好以双精度进行, 即进行 32 位的除法、乘法和加法。最好有双精度的准确度, 以保持自相关函数和滤波器系数的动态范围。

在图 10 中示出了 LPC 子系统 206 的一个例示实施例的方框图, 它用到上述等式(15)-(20)。LPC 子系统 206 由三个电路部分组成, 一个计算电路 330 和两个

用于更新主计算电路 330 的寄存器的缓存更新电路 332 和 334。计算首先是从数值 R(1)到 R(10)装入缓存器 340 中开始的。为了启动计算寄存器 346，通过多路转换器 344 预先装入数值 R(1)。寄存器由多路转换器 354 用 R(0)初始化，缓存器 352(它保存十个 $\alpha_j^{(i-1)}$ 数值)由多路转换器 354 初始化为全零，缓存器 356(它保持十个 $\alpha_j^{(i)}$ 数值)由多路转换器 358 初始化为全零，对于计算循环 i 设定为 1。为了明了起见，计数器 i 和 j 以及其它运算循环控制在图中未画出，但是这种类型的逻辑电路的设计和综合是处在数字逻辑设计领域中普通熟练者的能力范围之内的。

$\alpha_j^{(i-1)}$  数值从缓存器 356 输出，以计算等式(14)中的  $K_i E^{(i-1)}$  项。各个数值  $R(i-j)$  从缓存器 340 输出，在乘法器 360 中与  $\alpha_j^{(i-1)}$  相乘。得到的数值与来自寄存器 346 的数值在加法器 362 中作减法，每次相减的结果存储在寄存器 346 中以供作下一项减法。在第 i 个循环中，如等式(14)的求和项中所示，有  $i-1$  次乘法和累加运算。在该循环的最后，寄存器 346 中的数值在除法器 364 中除以来自寄存器 348 的数值  $E^{(i-1)}$ ，产生数值  $k_i$ 。

然后，该数值  $k_i$  用在缓存更新电路 332 中，计算等式(19)中的  $E^{(i)}$ ，数值  $E^{(i)}$  在  $k_i$  的下一个计算循环中用作为数值  $E^{(i-1)}$ 。当前循环的数值  $k_i$  在乘法器 366 中自乘以获得数值  $k_i^2$ 。然后，数值  $k_i^2$  在加法器 368 中被减去 1。这一加法器运算的结果在乘法器 370 中乘以来自寄存器 348 的数值  $E^{(i)}$ 。其结果  $E^{(i)}$  通过多路转换器 350 输入寄存器 348 存储起来作为下一周期的数值  $E^{(i-1)}$ 。

然后，数值  $k_i$  用来以等式(15)计算数值  $\alpha_i^{(i)}$ 。在这种情况下，数值  $k_i$  通过多路转换器 358 输出缓存器 356。数值  $k_i$  还用在缓存更新电路 334 中，以等式(18)从数值  $\alpha_i^{(i-1)}$  计算出数值  $\alpha_i^{(i)}$ 。缓存器 352 中现存的数值用于计算  $\alpha_j^{(i)}$ 。如等式(18)所示，在第 i 个循环中有  $i-1$  次运算。在  $i=1$  的迭代中，不需要这些运算。对于第 i 次循环的各个 j 值，计算出一个  $\alpha_j^{(i)}$  值。在计算各  $\alpha_j^{(i)}$  值时，各  $\alpha_{i,j}^{(i-1)}$  值在乘法器 372 中乘以数值  $k_i$ ，以输出到加法器 374。在加法器 374 中，从另外输入加法器 374 的数值  $\alpha_j^{(i-1)}$  中减去数值  $k_i \alpha_{i,j}^{(i-1)}$ 。各个乘法和加法的结果作为  $\alpha_j^{(i)}$  值通过多路转换器 358 供给缓存器 356。

对当前循环计算出  $\alpha_i^{(i)}$  和  $\alpha_j^{(i)}$  值后，就将刚算出并存储于缓存器 356 中的这些数值通过多路转换器 354 输出给缓存器 352。存在缓存器 356 中的数值存储于缓存器 352 的相应位置中。这样，缓存器 352 得到更新，以计算  $i+1$  循环的数值  $k_i$ 。

在前一循环的最后产生的数据  $\alpha_j^{(i-1)}$  在当前循环中用于为下一循环产生更新数据  $\alpha_j^{(i)}$ ，注意到这一点是很重要的。必须保留该前一循环的数据，以便为下一循环

完整地产生更新数据。这样，两个缓存器 356 和 352 用于将前一循环的数据保存到更新数据完全产生为止。

上述说明是针对在完成更新数值的计算时数据从缓存器 356 向缓存器 352 并行转送作出的。这种执行方式确保旧的数据在计算新数据的整个过程中得以保留，而不会象在设置单个缓存器情况下，旧的数据在完全利用之前就丢失了。上述方式是实现相同结果的现有几个方式之一。例如，缓存器 352 和 356 可以经过多路转换，以使得在当前循环中依据存储在一个第一缓存器中的数值计算数值  $k_i$  时，更新数据存在第二缓存器中，用在下一计算循环中。在下一循环中，数值  $k_i$  依据存在第二缓存器中的数值算出。第二缓存器中的数值和数值  $k_i$  用于为下一循环产生更新数据，这些更新数据存在第一缓存器中。缓存器的这种替换使得前一计算循环的数据得以保存，以此产生更新数据，并在不重写需要用来产生更新数据的先前数据的情况下存储更新数据。使用这种技术能够将与下一循环  $k_i$  值的计算相关的延迟减至最小。因而，在计算  $k_i$  时乘积/累加的更新可以在计算下一  $\alpha_j^{(i-1)}$  值的同时进行。

在完成最后一个计算循环( $i=10$ )后存在缓存器 356 中的 10 个 LPC 系数  $\alpha_j^{(10)}$  换算到相应的最后 LPC 系数  $\alpha_j$ 。向多路转换器 344、376 和 378 提供一个换算选择信号，因而，存在系数查算表 342 中的换算数值，即表III的十六进制数值，被选中，通过多路转换器 344 输出。存在表 342 中的数值按顺序在时钟脉冲控制下输出，并输入乘法器 360。乘法器 360 还通过多路转换器 370 接收从寄存器 356 顺序输出的数值  $\alpha_j^{(10)}$ 。换算后的数值通过多路转换器 378 从乘法器 360 输出到 LPC 至 LSP 转换子系统 208(图 7)。

为了迅速有效地以少量位数将 10 个换算后的 LPC 系数编码，各系数转换成线频谱对频率，在宋和江发表于 ICASSP'84 中的文章“线频谱对(LSP)和语音数据压缩”中对此有说明。在下列等式(21)和(22)及表IV中示出了 LSP 参数的计算。

LSP 频率是处在 0 和  $\pi$  之间的下列等式的十个根：

$$P(\omega)=\cos 5\omega+P_1 \cos 4\omega+\cdots+P_4 \cos \omega+P_5/2 \quad (21)$$

$$Q(\omega)=\cos 5\omega+q_1 \cos 4\omega+\cdots+q_4 \cos \omega+q_5/2 \quad (22)$$

其中， $P_n$  和  $q_n$  对  $n=1, 2, 3, 4$  分别取值，并在表IV中递归定义

表 IV

$$\begin{array}{ll}
 P_1 = -(\alpha_1 + \alpha_{10}) - 1 & q_1 = -(\alpha_1 - \alpha_{10}) + 1 \\
 P_2 = -(\alpha_2 + \alpha_9) - P_1 & q_2 = -(\alpha_2 - \alpha_9) + q_1 \\
 P_3 = -(\alpha_3 + \alpha_8) - P_2 & q_3 = -(\alpha_3 - \alpha_8) + q_2 \\
 P_4 = -(\alpha_4 + \alpha_7) - P_3 & q_4 = -(\alpha_4 - \alpha_7) + q_3 \\
 P_5 = -(\alpha_5 + \alpha_6) - P_4 & q_5 = -(\alpha_5 - \alpha_6) + q_4
 \end{array}$$

表IV中,  $\alpha_1, \dots, \alpha_{10}$  是从 LPC 分析得到的换算系数。为了简化, 等式(21)和(22)的 10 个根换算成 0 到 0.5 之间的值。LSP 频率的特性是, 当 LPC 滤波器稳定时, 两个等式的根变动; 即, 最低的根  $\omega_1$  是  $P(\omega)$  的最低的根, 次最低的根  $\omega_2$  是  $Q(\omega)$  的最低的根, 以此类推。在 10 个频率中, 奇频率是  $P(\omega)$  的根, 而偶频率是  $Q(\omega)$  的根。

根的查找方法如下。首先, 如上面所示那样将 LPC 系数加入, 从而以双精度计算出  $P$  和  $q$  系数。然后, 每隔  $\pi/256$  弧度计算一次  $P(\omega)$ , 再观察这些值的符号变化, 辨别在该子区域中的一个根。如果找到一个根, 则在该区域两端之间进行一次线性内插, 以逼近根的真实位置。由于频率的定序特性, 在每对  $P$  根之间保证存在一个  $Q$  根(第 5 个  $Q$  根处于第 5 个  $P$  根与  $\pi$  之间)。在每对  $P$  根之间进行对分查找, 确定  $Q$  根的位置。为了简便, 每个  $P$  根近似取最近的  $\pi/256$  数值, 在这些近似值之间进行对分查找。如果未找到根, 则使用先前找到根的最近的帧的 LSP 频率的非量化值。

现在参见图 11, 其中示出一个用于产生 LSP 频率的例示电路。上述操作需要总共 257 个在 0 和  $\pi$  之间的可能的余弦值, 它们在双精度下存储在余弦表 400 中, 该表格由模 256 计数器 402 寻址。对于输入表 400 的各个  $j$  值, 给出  $\cos \omega, \cos 2\omega, \cos 3\omega, \cos 4\omega$  和  $\cos 5\omega$  输出, 其中:

$$\omega = j \pi / 256 \quad (23)$$

$j$  是计数值。

余弦表 400 输出的数值  $\cos \omega, \cos 2\omega, \cos 3\omega$  和  $\cos 4\omega$  输入到相应的各个乘法器 404、406、408 和 410, 而数值  $\cos 5\omega$  则直接输入加法器 412 中。这些值在各乘法器 404、406、408 和 410 中分别乘以通过多路转换器 414、416、418 和 420 输入其中的数值  $P_4, P_3, P_2$  和  $P_1$ 。

相乘结果也输入累加器 412。此外, 数值  $P_5$  通过多路转换器 422 送到乘法器 424, 同时, 常数 0.5, 即  $1/2$  也送到乘法器 424。乘法器 424 的输出结果也作为另

一个输入给加法器 412。为了利用同一电路计算  $P(\omega)$  和  $Q(\omega)$  值，根据  $P/q$  系数选通信号，多路转换器 414-422 在  $P1-P5$  或  $q1-q5$  之间作出选择。图中未示出产生  $P1-P5$  或  $q1-q5$  的电路，但是，利用一系列将 LPC 系数和  $P1-P5$  或  $q1-q5$  进行加减运算的加法器，以及存储  $P1-P5$  或  $q1-q5$  值的寄存器，很容易实现该电路。

加法器 412 将输入值相加，依情况而定，输出  $P(\omega)$  或  $Q(\omega)$ 。为了简便，在以后的讨论中将考虑  $P(\omega)$  值的情况， $Q(\omega)$  值以类似方式利用  $q1-q5$  数值算出。加法器 412 输出  $P(\omega)$  的当前值，存储在寄存器 426 中。先前存在寄存器 426 中的前一  $P(\omega)$  值移位至寄位器 428 中。当前和先前的  $P(\omega)$  值的符号位在异或门 430 中异或，以传给线性内插器 434 的起动信号的形式显示零交叉或符号变换。 $P(\omega)$  的当前和先前数值也从寄存器 426 和 428 输出到线性内插器 434，该内插器响应起动信号，在  $P(\omega)$  发生零交叉的两个值之间进行内插。该线性内插小数值结果，即与数值  $j-1$  的距离，与来自计数器 256 的数值  $j$  一起提供给缓存器 436。门 430 还将起动信号提供给缓存器 436，缓存器 436 允许数值  $j$  和对应的小数值  $FV_j$  存入。

在加法器 438 中，数值  $j$  中减去该小数值，作为缓存器 436 的输出，或者换种方式，减去后输入缓存器 436。在替换方式下，在输入缓存器 436 的  $j$  线中用一个寄存器，使得数值  $j-1$  与小数值输入一起输入缓存器 436。小数值可以在存入寄存器 436 之前或从中输出时加到数值  $j-1$  上。在任何情况下，混合值  $j+FV_j$  或  $(j-1)+FV_j$  输出到除法器 440，除以输入的常数 512。只需改变二进制字中的二进制小数点位置，就可以简单地完成除法操作。这一除法操作为达到处于 0 和 0.5 之间的 LSP 频率而提供必需的换算。

$P(\omega)$  或  $Q(\omega)$  的函数计算需要 5 次余弦值查表，4 次双精度乘法运算和 4 次加法运算。算出的根通常只精确到大约 13 位，并在单精度下存放。LSP 频率提供给 LSP 量化子系统 210(图 7)，用于量化。

LSP 频率算出之后，就必须量化，以便传送。十个 LSP 频率中的每一个都大致以一偏置数为中心。应该注意到，当输入语音具有平坦的 LSP 频谱特性曲线并且不能做任何短期预测时 LSP 频率接近偏置数。偏置数从编码器中取出，并且用到一个简单的 DPCM 量化器。在译码器中，偏置数又加回去。由 LPC 至 LSP 转换子系统所提供的各 LSP 频率  $\omega_1-\omega_{10}$  的偏置数的非，以十六进制形式给出在表 V 中。表 V 中给出的数值也是二进制补码，有 15 个小数位。十六进制数值 0x8000(或 -32768) 表示 -1.0。这样，表 V 的第一个值 0xfa2f(或 -1489) 表示  $-0.045441=-1489/32768$ 。

表 V

LSP 频率	负偏置数
$\omega_1$	0xfa2f
$\omega_2$	0xf45e
$\omega_3$	0xee8c
$\omega_4$	0xe8bb
$\omega_5$	0xe2e9
$\omega_6$	0xdd18
$\omega_7$	0xd746
$\omega_8$	0xd175
$\omega_9$	0xcb3
$\omega_{10}$	0xc5d2

子系统中用的预测器等于子系统一个缓存器中存储的来自先前帧的量化 LSP 频率的 0.9 倍。加入衰减常数 0.9，因而信道错误最终将消除掉。

用到的量化器是线性的，但动态范围和步长随速率而变。在高速率帧中，每一 LSP 频率有更多位数要传送，因而，量化级数依赖于速率。在表 VI 中，示出了每速率下对各频率的量化的位分配和动态范围。例如，在速率 1， $\omega_1$  统一量化成 4 位(即 16 级)，最高量化级为 0.025，而最低的是-0.025。

表 VI

速率	全	半	四分之一	八分之一
$\omega_1$	4: ±.025	2: ±.015	1: ±.01	1: ±.01
$\omega_2$	4: ±.04	2: ±.015	1: ±.01	1: ±.015
$\omega_3$	4: ±.07	2: ±.03	1: ±.01	1: ±.015
$\omega_4$	4: ±.07	2: ±.03	1: ±.01	1: ±.015
$\omega_5$	4: ±.06	2: ±.03	1: ±.01	1: ±.015
$\omega_6$	4: ±.06	2: ±.02	1: ±.01	1: ±.015
$\omega_7$	4: ±.05	2: ±.02	1: ±.01	1: ±.01
$\omega_8$	4: ±.05	2: ±.02	1: ±.01	1: ±.01
$\omega_9$	4: ±.04	2: ±.02	1: ±.01	1: ±.01

$\omega_{10}$	4: ±.04	2: ±.02	1: ±.01	1: ±.01
总计	40 位	20 位	10 位	10 位

如果由速率确定算法选中的速率的量化范围不够大，或者发生斜率溢出，则速率提高到下一个较高的速率上。速率持续提升，一直到动态范围得到调节或达到全速率为止。在图 12 中，给出了运用选择速率提高技术的装置的一个例示框图。

图 12 以框图形式示出了 LSP 量化子系统 210 的一个例示设备，它包括速率提升电路。图 12 中，当前帧 LSP 频率从除法器 440(图 11)输出到寄存器 442，存储起来，以在下一帧中进行速率提升限时输出。先前帧 LSP 频率和当前帧 LSP 频率分别从寄存器 442 和除法器 440 输出到速率提升逻辑 444，进行当前帧的速率提升限定。速率提升逻辑 442 还接收到初始速率决定，以及来自速率限定子系统 204 的速率限制指令。在确定是否必须增大速率时，逻辑 444 根据当前和先前帧 LSP 频率差值的平方之和，将先前帧 LSP 频率与当前帧 LSP 频率比较。其结果再与一个阈值比较，阈值如被超出，则显示有必要提升速率来确保高质量的语音编码。在超出阈值时，逻辑 444 将初始速率增加一个速率级别，从而给出整个编码器通用的最终速率。

图 12 中，各 LSP 频率值  $\omega_1 - \omega_{10}$  逐个与对应的偏置数一起输入加法器 450。从输入的 LSP 值中减去偏置数，其结果输出给加法器 452。加法器 452 还收到一个输入-预测值，即乘以衰减常数后的先前帧对应的 LSP 值。加法器 452 从加法器 450 的输出中减去预测值。加法器 452 的输出作为量化器 454 的一个输入。

量化器 454 由限幅器 456、最小动态范围查算表 458、逆步长查算表 460、加法器 462、乘法器 464 和位屏蔽 466 组成。在量化器 454 中进行量化，首先确定输入值是否处在量化器 454 的动态范围内。输入值提供给限幅器 456，如果该输入超出了表 458 提供的限定值，限幅器 456 就将它限制在动态范围的上界和下界内。查算表 458 按照表 VI，响应输入其中的速率和 LSP 频率索引 i，向限幅器 456 提供存储的限定值。限幅器 456 的输出值输入加法器 462，从中减去表 458 提供的动态范围的最小值。按照表 VI 中所列的最小动态范围值，先不管其符号，依据速率和 LSP 频率索引 i 再次确定表格 458 的输出值。例如，对于(全速率， $\omega_1$ )，查算表 458 的值为 0.025。

加法器 462 的输出在乘法器 464 中乘以一个选自查算表 460 的数值。查算表 460 包含有与表 VI 中所列数值一致的、各个速率下的各 LSP 值的步长之逆相对应

的数值。依据速率和 LSP 频率索引  $i$  来选择查算表 460 的输出值。对于各个速率和 LSP 频率索引  $i$ , 存在查表 460 中的数值是 $((2^n-1)/\text{动态范围})$ , 其中,  $n$  是代表量化值的位数。例如, 对于(速率 1,  $\omega_1$ ), 查算表 460 的数值是 $(15/0.05)$ 或 300。

乘法器 464 的输出是介于 0 和  $2^n-1$  之间的一个数值, 它供给位屏蔽 466。位屏蔽 466 响应速率和 LSP 频率索引, 根据表 VI, 从输入值中提取适当的位数。提出的是输入值中的  $n$  个整数位, 从而形成位限制输出  $\Delta\omega_i$ 。数值  $\Delta\omega_i$  是量化未偏置差分编码 LSP 频率, 从代表 LPC 系数的信道中传送。

数值  $\Delta\omega_i$  也通过一个由逆量化器 468、加法器 470、缓存器 472 和乘法器 474 组成的预测器反馈。逆量化器 468 由步长查算表 476、最小动态范围查算表 478、乘法器 480 和加法器 482 组成。

数值  $\Delta\omega_i$  与选自查算表 476 的一个数值一起输入乘法器 480。查算表 476 含有在各个速率下依据于表 VI 中所列数值的各 LSP 数值的步长所对应的数值。依据速率和 LSP 频率索引  $i$ , 从查算表 476 中选择输出值。对于各个速率和 LSP 频率索引  $i$ , 存在查算表 460 中的数值是 $(\text{动态范围}/2^n-1)$ , 其中,  $n$  是代表量化值的位数。乘法器 480 将输入值相乘, 提供输出给加法器 482。

加法器 482 从另一输入接收表 478 的一个数值。表 478 的输出数值根据表 VI 中所列最小动态范围数值, 无视其符号, 由速率和 LSP 频率索引  $i$  来确定。加法器 482 将表 478 提供的最小动态范围值与乘法器 480 的输出值相加, 结果输出给加法器 470。

加法器 470 在另一输入端接收到乘法器 474 的预测值输出。这些值在加法器 470 中相加, 并存储在十个字容量的缓存器 472 中。在当前帧中缓冲器 472 的每个先前帧输出值在乘法器 474 中乘以常数 0.9。乘法器 474 输出的预测值如前面讨论过的那样, 供给加法器 452 和 470。

在当前帧中, 缓存器 472 中存储的数值是先前帧重建 LSP 值减去偏置值。相类似, 在当前帧中, 加法器 470 的输出值是不带偏置的当前帧重建的 LSP 值。在当前帧中, 缓存器 472 和加法器 470 的输出分别提供给加法器 484 和 486, 加上偏置数。加法器 484 和 486 的输出值分别是先前帧重建 LSP 频率和当前帧重建 LSP 频率值。依照下列等式在更低速率下进行 LSP 平滑化:

$$\text{平滑后 LSP} = a(\text{当前 LSP}) + (1-a)(\text{先前 LSP}) \quad (24)$$

其中:

a=0 全速率下

a=0.1 半速度下

a=0.5 四分之一速率下

a=0.85 八分之一速率下

先前帧(f-1)重建 LSP 频率值  $\omega'_{i, f-1}$  和当前帧(f)重建频率值  $\omega'_{i, f}$  从量化子系统 210 输出到音调子帧 LSP 内插子系统 216 和码书子帧 LSP 内插子系统 226。量化后的 LSP 频率值  $\Delta\omega_j$  从 LSP 量化子系统 210 输出到数据汇编子系统 236, 用于传送。

以后说明的加权滤波器和共振峰合成滤波器所用到的 LPC 系数对于正在编码的音调子帧是合适的。对各个音调子帧, LPC 系数的内插对每个音调子帧进行一次, 如表VII:

表VII

速率 1:

$$\omega_i = 0.75\omega'_{i, f-1} + 0.25\omega'_{i, f} \quad \text{音调子帧 1}$$

$$\omega_i = 0.5\omega'_{i, f-1} + 0.5\omega'_{i, f} \quad \text{音调子帧 2}$$

$$\omega_i = 0.25\omega'_{i, f-1} + 0.75\omega'_{i, f} \quad \text{音调子帧 3}$$

$$\omega_i = \omega'_{i, f} \quad \text{音调子帧 4}$$

速率 1/2:

$$\omega_i = 0.625\omega'_{i, f-1} + 0.375\omega'_{i, f} \quad \text{音调子帧 1}$$

$$\omega_i = 0.125\omega'_{i, f-1} + 0.875\omega'_{i, f} \quad \text{音调子帧 2}$$

速率 1/4:

$$\omega_i = 0.625\omega'_{i, f-1} + 0.375\omega'_{i, f} \quad \text{音调子帧 1}$$

速率 1/8:

不进行音调检索

音调子帧计数器 224 用于跟踪计算音调参数的音调子帧, 计数器的输出提供给音调子帧 LSP 内插子系统 216, 用于音调子帧 LSP 内插。音调子帧计数器 224 还提供一个输出给数据压缩子系统 236, 指示在选中速率下音调子帧的结束。

图 13 示出音调子帧 LSP 内插子系统 216 的一个例示装置, 它用于对相关音调子帧的 LSP 频率进行内插。在图 13 中, 先前和当前 LSP 频率  $\omega'_{i, f-1}$  和  $\omega'_{i, f}$  分别从 LSP 量化子系统输出到乘法器 500 和 502, 再分别乘以存储器 504 提供的常

数。存储器 504 存储一组常数，并且，根据来自音调子帧计数器的音调子帧数输入(这在后文讨论)，提供一个表VII中所列的常数输出，与先前和当前帧 LSP 数值相乘。乘法器 500 和 502 的输出按照表VII中的等式在加法器 506 中相加，形成该音调子帧的 LSP 频率值。对于各音调子帧，LSP 频率的内插一旦完成，就执行 LSP 至 LPC 的反向转换，以获得感知加权滤波器和 A(z)的当前系数值。这样，内插后的 LSP 频率提供给图 7 的 LSP 至 LPC 转换子系统 218。

LSP 至 LPC 转换子系统 218 将内插 LSP 频率逆转换成 LPC 系数，用于语音的再合成。先前的参考文章，宋和江所著的“线频谱对(LSP)和语音数据压缩”中对本发明在转换过程中用的算法有全面的讨论和推导。计算特征是这样的，P(z) 和 Q(z)可以利用 LSP 频率由下列等式表示：

$$P_{(z)} = (1 + Z^{-1}) \prod_{i=1}^5 (1 - 2\cos(\omega_{2i-1})Z^{-1} + Z^{-2}) \quad (25)$$

其中， $\omega_i$  是 P'多项式的根(奇频率)，

$$Q_{(z)} = (1 - Z^{-1}) \prod_{i=1}^5 (1 - 2\cos(\omega_{2i})Z^{-1} + Z^{-2}) \quad (26)$$

其中， $\omega_i$  是 Q'多项式的根(偶频率)，

$$A_{(z)} = \frac{Pz + Qz}{2} \quad (27)$$

这一运算首先是对所有奇频率 i 算出数值  $2\cos(\omega_i)$ 。利用零附近的余弦的 5 阶单精度泰勒级数展开式完成该计算。在余弦表中最接近点处的泰勒展开式可能更准确，但是，零附近的展开式有足够的精度，并且，不会带来多余的计算。

接下去，计数 P 多项式的系数。多项式之积的系数是单个多项式的系数序列的卷积。上述等式(25)中 6 个 z 多项式系数序列的卷积 {1, -2cos( $\omega_1$ ), 1}, {1, -2cos( $\omega_3$ ), 1}……{1, -2cos( $\omega_9$ ), 1} 和 {1, 1} 被算出。

P 多项式一经开出，就对 Q 多项式重复相同的程序，上述等式(26)中的 6 个 z 多项式系数序列 {1, -2cos( $\omega_2$ ), 1}, {1, -2cos( $\omega_4$ ), 1}……{1, -2cos( $\omega_{10}$ ), 1} 和 {1, 1} 以及适当的系数被加起来，并除以 2，即移动 1 位，以形成 LPC 系数。

图 13 还详细地示出 LSP 至 LPC 转换子系统的例示执行过程。电路部分 508 依据输入值  $\omega_i$  计算出数值  $-2\cos(\omega_i)$ 。电路部分 508 由缓存器 509、加法器 510 和 515、乘法器 511、512、514、517 和 518、寄存器 513 和 516 组成。在计算数值  $-2\cos(\omega_i)$  时，寄存器 513 和 516 被初始化为零。因为该电路是计算  $\sin(\omega_i)$  的，所以，在加法器 510 中首先先以输入常数值  $\pi/2$  减  $\omega_i$ 。该数值由乘法器 511 取平方，然后，用乘法器 512 和寄存器 513 依次计算数值  $(\pi/2 - \omega_i)^2$ 、 $(\pi/2 - \omega_i)^4$ 、 $(\pi/2 - \omega_i)^6$  和

$(\pi/2 - \omega_i)^8$ 。

泰勒级数展开式系数  $C[1]-C[4]$  依次和乘法器 512 的输出值一起输入乘法器 514。乘法器 514 的输出值输入加法器 515，与寄存器 516 的输出相加，这些值累加形成输出  $C[1](\pi/2 - \omega_i)^2 + C[2](\pi/2 - \omega_i)^4 + C[3](\pi/2 - \omega_i)^6 + C[4](\pi/2 - \omega_i)^8$ ，输给乘法器 517。来自寄存器 516 的乘法器的输入与来自加法器 510 的输出  $(\pi/2 - \omega_i)$  在乘法器 517 中相乘。来自乘法器 517 的数值  $\cos(\omega_i)$  在乘法器 518 中乘以常数-2，从而形成输出  $-2\cos(\omega_i)$ ，提供给电路部分 520。

电路部分 520 用于计算 P 多项式的系数，它由存储器 512、乘法器 522 和加法器 523 组成。存储器单元  $P(1) \cdots P(11)$  的阵列除  $P(1)$  置为 1 之外其余初始化为 0。已标到的  $-2\cos(\omega_i)$  数值送入乘法器 524，执行卷积  $(1, -2\cos(\omega_i), 1)$ ，其中， $1 \leq i \leq 5, 1 \leq j \leq 2i+1, j < 1$  时  $P(j)=0$ 。电路部分 520 是双份的(未示出)，以便计算 Q 多项式的系数。得到的  $P(1)-P(11)$  和  $Q(1)-Q(11)$  新的最终值送给电路部分 524。

电路 524 用来完成  $i=1$  至  $i=10$  的十个音调子帧 LPC 系数  $\alpha_i$  的计算。它由缓存器 525 和 526、加法器 527、528 和 529、除法器或位移位器 530 组成。最终值  $P(i)$  和  $Q(i)$  存储在缓存器 525 和 526 中。 $P(i)$  和  $P(i+1)$  数值在加法器 527 中相加，而相应的  $Q(i)$  和  $Q(i+1)$  值在加法器 528 中相减，这里  $1 \leq i \leq 10$ 。加法器 527 和 528 的输出，分别为  $P(z)$  和  $Q(z)$ ，输入加法器 529，相加并输出数值  $(P(z)+Q(z))$ 。将各个位移动一位，从而将加法器的输出除以 2。各移位后的数值  $(P(z)+Q(z))/2$  是 LPC 系数  $\alpha_i$  输出。音调子帧 LPC 系数送给图 7 的音调检索子系统 220。

除了全速率之外，对选中速率所确定的各个码书子帧，LSP 频率也经过内插。内插的计算方式与音调子帧 LSP 内插的相同。码书子帧 LSP 内插在码书子帧 LSP 内插子系统 226 中运算，并提供给 LSP 至 LPC 转换子系统 228，转换是以类似于 LSP 至 LPC 转换子系统 218 的方式进行的。

在参见图 3 时讨论过，音调检索是一种利用合成技术的分析，通过选择使输入语音和合成语音的误差最小的参数来进行编码。在音调检索中，语音是用等式(2) 表示频率特性的音调合成滤波器合成的。每隔 20ms，语音帧被划分成一定量的音调子帧，如前所述，它依赖于为该帧所选择的数据速率。对于每一个音调子帧分别计算一次参数  $b$  和  $L$ ，即音调增益和滞后。在这儿例示的执行过程中，音调滞后  $L$  介于 17 和 143 之间的，在  $b=0$  的情况下，为了传送原因  $L=16$ 。

语音编码器用到了在等式(1)中所列形式的感知噪声加权滤波器。如前所述，感知加权滤波器的目的是给处在低能量的频率中的错误加权，从而减少错误引起的

噪声的影响。感知加权滤波器是从以前建立的短期预测滤波器中导出的。在该加权滤波器以及后文将说明的共振峰合成滤波器中用到的 LPC 系数是为正在编码的子帧适当地内插过的数值。

在进行合成-分析操作时，在编码器中用到语音译码器/合成器的备份。在语音编码器中所用的合成滤波器的形式由等式(3)和(4)给出。等式(3)和(4)对应于一个其后有感知加权滤波器的译码语音合成滤波器，因而称其为加权合成滤波器。

在假设当前帧时码书基值为零，即  $G=0$  情况下，进行音调检索。对于每个可能的音调滞后  $L$  进行语音合成，并将它与原始语音比较。输入语音和合成语音之间的误差在计算其均方误差(MSE)之前由感知加权滤波器加权。其目的是从所有可能的  $L$  和  $b$  值中选取使感知加权语音和感知加权合成语音之间的误差最小的  $L$  和  $b$  数值。误差的最小化可以用下列等式表示：

$$MSE = \frac{1}{LP} \sum_{n=0}^{L_p-1} (x(n) - x'(n))^2 \quad (28)$$

其中， $L_p$  是音调子帧中取样的数量，在本实施例中，对全速率音调子帧为 40。计算出使 MSE 最小的音调增益  $b$ 。对所有允许的  $L$  数值重复这些运算，并给音调滤波器选择产生最小 MSE 的  $L$  和  $b$ 。

最佳音调滞后的计算涉及在  $n=-L_{\max}$  与  $n=(L_p-L_{\min})-1$  之间所有时间下的共振峰余量(图 3 中的  $P(n)$ )，其中  $L_{\max}$  是最大音调滞后值， $L_{\min}$  是最小音调滞后值，而  $L_p$  是选中速率下的音调子帧长度， $n=0$  是音调子帧的起点。在本实施例中， $L_{\max}=143$ ，而  $L_{\min}=17$ 。采用图 14 中给出的编号方案，对于速率 1/4， $n=-143$  到  $n=142$ ，而对于速率 1/2， $n=-143$  到  $n=62$ ，对于速率 1， $n=-143$  到  $n=22$ 。当  $n < 0$  时，共振峰余量就是来自先前音调子帧的音调滤波器输出，它保存在音调滤波器存储器中，称为闭环共振峰余量。在  $n \geq 0$  的时候，共振峰余量是共振峰分析滤波器的输出，该滤波器具有滤波特性  $A(z)$ ，其输入是当前分析帧语音取样。在  $n \geq 0$  时，共振峰余量称为开环共振峰余量，并且当音调滤波器和码书在该子帧中作了极好的预测时，正好为  $P(n)$ 。对于从相关的共振峰余量值计算出最佳音调滞后的说明将参照图 14-17 给出。

音调检索是对 143 个重建的闭环共振峰余量取样  $P(n)(n<0)$  以及  $L_p-L_{\min}$  非量化开环共振峰余量取样  $P_o(n)(n \geq 0)$  作出的。该检索有效地从一种基本开环检索，其中  $L$  是小的，并且所用到的大部分余量取样为  $n > 0$ ，逐渐转变成一种基本闭环检索，其中  $L$  是大的，因而用到的所有余量取样是  $n < 0$ 。例如，在全速率下利

用图 14 中的编号方案，其中，音调子帧由 40 个语音取样组成，则语音检索从采用号码  $n=-17$  到  $n=22$  的共振峰余量取样组开始进行。在这一方案中，从  $n=-17$  到  $n=-1$ ，取样为闭环共振峰余量取样，而从  $n=0$  到  $n=22$ ，取样为开环共振峰余量取样。用于确定最佳音调滞后的下一组共振峰余量取样是编号由  $n=-18$  到  $n=21$  的取样。而从  $n=-18$  到  $n=-1$ ，取样是闭环共振峰余量取样，从  $n=0$  到  $n=21$ ，则是开环共振峰余量取样。对所有取样组进行该过程，直到给最后的共振峰余量取样组  $n=-143$  至  $n=-104$  计算出音调滞后为止。

前面就等式(28)讨论过，目的是在随机码书给出零基值( $G=0$ )的情况下，对于所有可能的  $L$  和  $b$  值，使  $x(n)$  与  $x'(n)$  之间的误差最小，这里  $x(n)$  是感知加权语音减去加权共振峰滤波器的零输入特性曲线(ZIR)， $x'(n)$  是滤波器中无存储时的感知加权合成语音。等式(28)可以重写成关于  $b$  的等式：

$$MSE = \frac{1}{L_p} \sum_{n=0}^{L_p-1} (x(n) - by(n))^2 \quad (29)$$

其中：

$$y(n) = h(n) * P(n-L) \quad 0 \leq n \leq L_p-1 \quad (30)$$

其中， $y(n)$  当  $b=1$  时是带有音调滞后  $L$  的加权合成语音，而  $h(n)$  是具有等式(3)的滤波特性的加权共振峰合成滤波器的冲激响应。

该最小化过程等同于使  $E_L$  值最大化，其中：

$$E_L = \frac{(E_{xy})^2}{E_{yy}} \quad (31)$$

其中：

$$E_{xy} = \sum_{n=0}^{L_p-1} x(n)y(n) \quad (32)$$

及

$$E_{yy} = \sum_{n=0}^{L_p-1} y(n)y(n) \quad (33)$$

对于给定的  $L$ ，最佳值  $b$  为：

$$b_L = \frac{E_{xy}}{E_{yy}} \quad (34)$$

这一检索对所有允许的  $L$  值重复进行。最佳值  $b$  限制为正，因而，对于在检索中导致任何负的  $E_{xy}$  值的  $L$  值不作考虑。最后，选出使  $E_L$  值最大的音调滞后  $L$  和音调增益  $b$  用于传送。

如前所述， $x(n)$  实际是输入语音与加权共振峰滤波器的 ZIR 之间的感知加权差，这是因为，对于后面的等式(35)-(38)中设定的递归卷积，已假设滤波器  $A(z)$

总是从滤波器存储器中的零起始。然而，滤波器从滤波器存储器中的零起始并非实际情况。在合成时，滤波器处在从先前子帧保留下来的状态下。在执行过程中，在起始点，从感知加权语音中除去初始状态的影响。在这种方式下，对每一 L 值，只需计算稳态滤波器  $A(z)$  的特性曲线直到  $p(n)$ ，所有存储器最初为 0，并且可以用到递归卷积。 $x(n)$  的值只需计算一次，但是，共振峰滤波器对音调滤波器的输出的零状态响应  $y(n)$  要就每一滞后 L 进行计算。各  $y(n)$  的计算涉及许多次不必就每一滞后都进行的冗余乘法运算。下文中说明的递归卷积方法用于将所需计算减至最少。

关于递归卷积，数值  $y_L(n)$  用  $y(n)$  值定义成：

$$y_L(n) = h(n)^* P(n-L) \quad 17 \leq L \leq 143 \quad (35)$$

或

$$y_L(n) = h(i)p(n-L-i) \quad 17 \leq L \leq 143 \quad (36)$$

从等式(32)和(33)可看出：

$$y_L(0) = P(-L)h(0) \quad (37)$$

$$y_L(n) = y_{L-1}(n-1) + p(-L)h(n) \quad 1 \leq n \leq L_p, \quad 17 \leq L \leq 143 \quad (38)$$

这样，一旦完成  $y_{17}(n)$  的初始卷积，余下的卷积就可以递归地进行，大大减少了所需的计算次数。对于速率 1 的上述特定例子，利用编号为  $n=-17$  到  $n=22$  的共振峰余量取样组藉助等式(36)计算出数值  $y_{17}(n)$ 。

参见图 15，编码器包括图 5 译码器的备份，即图 7 的译码器子系统 235，但不带自适应后置滤波器。在图 15 中，音调合成滤波器 550 的输入是码书值  $c_r(n)$  与码书增益 G 的乘积。输出的共振峰余量取样  $p(n)$  输入共振峰合成滤波器 552，经过滤波并作为重建语音取样  $s'(n)$  输出。在加法器 554 中，从对应的输入语音取样  $s(n)$  中减去重建语音取样  $s'(n)$ 。取样  $s'(n)$  与  $s(n)$  之差输入感知加权滤波器 556。关于音调合成滤波器 550、共振峰合成滤波器 552 和感知加权滤波器 556，每一滤波器含有一个滤波器状态记忆，其中： $M_p$  是音调合成滤波器 550 中的记忆； $M_a$  是共振峰合成滤波器 552 中的记忆，而  $M_w$  是感知加权滤波器 556 中的记忆。

来自译码器子系统共振峰合成滤波器 552 的滤波器状态  $M_a$  提供给图 7 的音调检索子系统 220。在图 16 中，滤波器状态  $M_a$  用来计算滤波器 560 的零输入响应(ZIR)，滤波器 560 计算共振峰合成滤波器 552 的 ZIR。在加法器 562 中从输入语音取样  $s(n)$  中减去计算出的 ZIR 值，其结果由感知加权滤波器 564 加权。感知加权滤波器 564 的输出  $x_p(n)$  用在等式(28)-(34)中作为加权输入语音，其中，

$x(n)=x_p(n)$ 。

再参见图 14 和 15, 图 14 中示出的音调合成滤波器 552 输出到自适应码书 568, 后者基本上是一个存储器, 用于存储如上文讨论的那样计算出的闭环和开环共振峰余量取样。闭环共振峰余量存在存储器部分 570 中, 而开环共振峰余量存在存储器部分 572 中。这些取样按照前文讨论过的编号方案存储。如上文中讨论的那样, 按照各音调滞后  $L$  检索中的使用情况来组织闭环共振峰余量。开环共振峰余量用共振峰分析滤波器 574 从各音调子帧的输入语音取样中算出, 滤波器 574 在计算  $p_o(n)$  值时用到译码器子系统共振峰滤波器 552 的记忆  $Ma$ 。用于各当前音调子帧的  $P_o(n)$  数值移位通过一系列延迟元件 576, 提供给自适应码书 568 的存储器部分 572。开环共振峰余量被存储起来, 所产生的第一个余量取样编号为 0, 而最后一个编号为 142。

现参见图 16, 共振峰滤波器的冲激响应  $h(n)$  在滤波器 566 中算出, 并输出给移位寄存器 580。在滤波器中为每一音调子帧计算这些数值, 这在上文中谈到共振峰滤波器  $h(n)$  的冲激响应时, 即谈到等式(29)-(30)和(35)-(38)时讨论过。为了进一步减少音调滤波器子系统的计算需求, 共振峰滤波器  $h(n)$  的冲激响应截断为 20 个取样。

移位寄存器 580 及乘法器 582、加法器 584 和移位寄存器 586 组合起来, 如上文中讨论的那样, 进行来自移位寄存器 580 的数值  $h(n)$  与来自自适应码书 568 的数值  $c(m)$  之间的递归卷积。执行这一卷积操作是为了在假定音调增益设为 1 的情况下寻找共振峰滤波器对来自音调滤波器存储器的输入的零状态响应(ZSR)。在卷积电路的操作中, 对每一  $m$  值,  $n$  从  $L_p$  循环至 1, 而  $m$  从  $(L_p-17)-1$  循环至 -143。在寄存器 586 中, 当  $n=1$  时数据不移动, 而当  $n=L_p$  时不锁存数据。当  $m \leq -17$  时, 数据提供为卷积电路的输出。

继卷积电路之后的是相关和比较电路, 执行检索, 以寻找最佳音调滞后  $L$  和音调增益  $b$ 。相关电路亦称为均方误差(MSE)电路, 计算 ZSR 与共振峰滤波器的 ZIR 和输入语音即  $x(n)$  之间的感知加权差的自相关性和互相关性。利用这些数值, 相关电路就每一音调滞后数值计算出最佳音调增益  $b$  的数值。相关电路由移位寄存器 588、乘法器 590 和 592、加法器 594 和 596、寄存器 598 和 600、除法器 602 组成。在相关电路中, 计算是这样进行的, 即  $n$  从  $L_p$  循环至 1, 而  $m$  从  $(L_p-17)-1$  循环至 -143。

相关电路之后是比较电路, 它执行数据比较和存储, 以确定音调滞后  $L$  和增

益 b 的最佳值。比较电路由乘法器 604、比较器 606、寄存器 608、610 和 612、量化器 614 组成。比较电路为每一音调子帧输出使合成语音和输入语音之间的误差最小的 L 和 b 值。b 的值由量化器 614 量化到 8 级，用一 3 位的数值表示，当 L=16 时导出一个附加级，b=0 级。这些 L 和 b 值提供给码书检索子系统 230 和数据缓存器 222。它们通过数据压缩子系统 238 或数据缓存器 222 提供给译码器 234，用于音调检索。

类似于音调检索，码书检索是一种由合成编码系统进行的分析，在系统中，编码是通过选取使输入语音与用这些参数合成的语音之间的误差最小的参数完成的。对于速率 1/8，音调增益 b 设为零。

如前文中讨论的那样，每 20ms 分成多个码书子帧，如前所述，它们依赖于为该帧所选取的数据速率。对每一码书子帧分别计算一次参数 G 和 I，即码书增益和索引。在计算这些参数时，除了全速率之外，在码书子帧 LSP 内插子系统 226 中以类似于对音调子帧 LSP 内插子系统 216 说明过的方式对该子帧进行 LSP 频率内插。对于每个码书子帧，码书子帧内插 LSP 频率也由 LSP 至 LPC 转换子系统 228 转换成 LPC 系数。码书子帧计数器 232 用于跟踪计算码书参数的码书子帧，计数器输出提供给码书子帧 LSP 内插子系统 226，用于码书子帧 LSP 内插。码书子帧计数器 232 也提供一个输出给音调子帧计数器 224，显示在选中速率下的码书子帧的结束。

激励码书由  $2^M$  个码矢量组成，它们是依据一种单元变量白色高斯随机序列而建立的。 $M=7$  时，码书中 128 个入口。码书以某种递归方式组织，以使各个码矢量与相邻的码矢量相差一个取样，即一个码矢量中的取样移动一位，在一端移入一个新的取样，而在另一端抛掉一个取样。因而，一个递归码书可以作为一个长  $2^M + (L_c - 1)$  的线性阵列存储，其中， $L_c$  是码书子帧长度。然而，为了简化执行过程，和保存存储器空间，采用  $2^M$  个取样长的循环码书(128 个取样)。

为了减少计算量，码书中的高斯数值的中心部分值被消除。这些数值最初选自方差为 1 的白色高斯过程。然后，将小于 1.2 的数值设为零。这有效地将大约 75% 的数值设置成零，产生一个脉冲码书。码书的中心削除将在码书检索中执行递归卷积所需的计算量降低到 1/4，因为对零无须进行计算。用在当前执行过程中的码书在下面的表VII中给出。

表 VII

0x0000	0x0000	0x0000	0x0000	0x2afe	0x0000	0x0000	0x0000
0x41da	0x0000						
0x0000	0x0000	0x0000	0x0000	0x0000	0x3bb3	0x0000	0x363e
0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x417d	0x0000
0x0000							
0x0000							
0x0000	0x0000	0x0000	0x0000	0x0000	0x9dfe	0x0000	0x0000
0x0000							
0x0000	0xc58a	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
0x0000	0xc8db	0xd365	0x0000	0x0000	0xd6a8	0x0000	0x0000
0x0000	0x3e53	0x0000	0x0000	0xd5ed	0x0000	0x0000	0x0000
0xd08b	0x0000	0x0000	0x0000	0x0000	0x3d14	0x396a	0x0000
0x0000	0x0000	0x0000	0x0000	0x0000	0x4ee7	0xd7ca	0x0000
0x0000	0x438c	0x0000	0x0000	0xad49	0x30b1	0x0000	0x0000
0x0000	0x0000	0x0000	0x0000	0x0000	0xd187	0x2e16	0xd09b
0xcd8d	0x0000	0x3fed	0x0000	0x0000	0x0000	0x0000	0x32ff

又语音编码器采用包括公式(3)给出形式的加权合成滤波器在内的公式(1)给出形式的感知噪声加权滤波器。对于各个码书指数 I, 合成语音并与原来的语音比较。在计算出其 MSE 前误差由感知加权滤波器加权。

如前所述, 目的是使  $X(n)$  和  $X'(n)$  之间的误差在 I 和 G 的所有可能值上为极小。该误差的极小化可以由下面公式表示:

$$MSE = \frac{1}{L_c} \sum_{n=0}^{L_c-1} (X(n) - X'(n))^2 \quad (39)$$

这里  $L_c$  是码书子帧中的样本数。这里对于 G 公式(38)可改写:

$$MSE = \frac{1}{L_c} \sum_{n=0}^{L_c-1} (X(n) - GY(n))^2 \quad (40)$$

这里 Y 是通过用第 1 码矢量并假定  $G=1$  卷积共振峰主频成分滤波器的冲激响应而得到的。使该 MSE 极小转而等效于使大

$$E_I = \frac{(E_{xy})^2}{E_{yy}} \quad (41)$$

为极大, 这里

$$E_{xy} = \sum_{n=0}^{L_c-1} X(n) Y(n) \quad (42)$$

和

$$E_{yy} = \sum_{n=0}^{L_c-1} y(n) Y(n) \quad (43)$$

根据下面的公式得到对于所给的 I 最优的 G,

$$G_i = \frac{E_{xy}}{E_{yy}}$$

对于 I 的所有允许值重复这检索。与音调检索相反，该最优增益 G 允许为正值或负值。最后使  $E_i$  极大的指数 I, 和码书增益 G 被选择用于传送。

还应注意加权音调和共振峰主频成分的滤波器的 ZIR 跟输入语音之间的感知加权差值  $X(n)$  仅需计算一次。而对于各个码矢量的音调和共振峰主频成分滤波器的零状态响应需对于各个指数 I 来计算。由于采用循环码书，用于音调检索的所述递归卷积方法可用来使所需的计算量为最小。

再参见图 15, 编码器包括与图 5 中的译码器一样的即图 7 的译码器子系统在此计算出滤波器状态。其中  $M_p$  是在音调合成滤波器 550 中的记忆， $M_a$  是在共振峰合成滤波器 552 中的记忆； $M_w$  是在感知加权滤波器中的记忆。

分别来自译码器子系统的音调合成和共振峰滤波器的滤波器状态  $M_p$  和  $M_a$  送给图 7 的码书检索子系统 230。在图 17 中滤波器状态  $M_p$  和  $M_a$  送给零冲激响应(ZIR) 滤波器 620 以计算音调和共振峰合成滤波器 550 和 552 的 ZIR。在加法器 622 中从所输入的输入语音样值  $S(n)$  减去计算出的音调和共振峰滤波器的 ZIR, 该结果由感知加权滤波器加权。感知加权滤波器的输出  $X_c(n)$  用作上面的 MSE 公式(39)-(44) 中的加权了的输入语音则  $X(n)=X_c(n)$ 。

图 17, 在滤波器 626 中计算共振峰滤波器的冲激响应  $h(n)$  并输出至移位寄存器 628。对于各个码书子帧计算共振峰滤波器的冲激响应  $h(n)$ 。为进一步减少所需的计算量，该共振峰滤波器的冲激响应  $h(n)$  截取 20 个样本。

移位寄存器 628 随同乘法器 630、加法器 632 和移位寄存器 634 一起进行来自移位寄存器 628 的值  $h(n)$  与来自于如上所述含有码书矢量的码书 636 的值  $c(m)$  间的递归卷积。假定码书增量为 1, 进行这卷积运算以对各个码矢量寻找共振峰滤波器的零状态响应。该卷积电路的运行, 对于每个  $m$ ,  $n$  从  $L_c$  到 1 循环, 而  $m$  则从 1 到 256 循环。在寄存器 586 中当  $n=1$  数据不送出而当  $n=L_c$  数据不锁存。当  $m \leq 1$  数据才作为该卷积电路的输出给出。应该注意的是在开始紧接在卷积电路后的相关与比较电路前必须起动卷积电路以通过循环  $m$  子帧大小次数进行递归卷积运算。

该相关与比较电路进行实际的码书检索产生码书指数 I 值和码书增益 G 值。该相关电路除了被作为均方差(MSE) 电路借用以外, 还由音调与共振峰滤波器的

ZIR 跟输入语音  $X'(n)$  之间的感知加权差值来计算 ZSR 的自相关和互相关。换句话说该相关电路对码书指数 I 的各个值计算码书增益 G。该相关电路包括移位寄存器 638，乘法器 640 和 642，加法器 644 和 646，寄存器 648 和 650，和除法器 652。在相关电路中计算是 n 从  $L_c$  到 1 循环而 m 从 1 到 256 循环的。

接在相关电路后的比较电路为确定码书指数 I 和增益 G 的最优值进行数据的比较和存储。该比较电路包括乘法器 654，比较器 656，寄存器 658、660 和 662，和量化器 664。该比较电路为各个码书子帧提供使合成语音与输入语音之间的误差为极小的 I 和 G 的值。该码书增益 G 在量化器 614 中量化，DPCM 在量化期间以类似于参考图 12 所述的偏置消除的 LSP 频率量化和编码对该值编码。这些 I 和 G 值然后送给数据缓冲器 222。

在量化和该码书的 DPCM 编码中增益 G 是按照下面的公式计算的。

$$\text{量化后的 } G_i = 20\log G_i - 0.45(20\log G_{i-1} + 20\log G_{i-2}) \quad (45)$$

这里  $20\log G_{i-1}$  和  $20\log G_{i-2}$  分别是对前一帧(i-1)和再前一帧(i-2)计算出的值。

该 LSP、I、G、L 和 b 值随同速率送到数据压缩子系统 236 以安排这数据用于传输。在一项实施中该 LSP、I、G、L 和 b 值随同速率可经数据压缩子系统 236 送给译码器 234。在另一项实施中这些数据可经数据缓冲器 222 送给译码器 234 以用于音调检索。而在较佳实施例中在数据压缩子系统 236 中所采用的对码书符号位的保护会影响码书指数。所以 I 和 G 数据直接从数据缓冲器 222 送出时必须考虑到该保护。

在数据压缩子系统 236 中数据可根据各种传输格式压缩。图 18 示出数据压缩子系统 236 的功能单元的说明性实施例。数据压缩子系统 236 包括伪随机数发生器(PN)670，循环冗余校验(CRC)，计算单元 672，数据保护逻辑 674 和数据组合器 676。PN 发生器 670 接收速率并对八分之一速率产生一 4 位的随机数送给数据组合器 676。CRC 单元 672 接收码书增益和 LPS 值以及速率，并对全速率产生一 11 位内部 CRC 码送到数据组合器 676。

数据组合器 676 接收该随机数，CRC 码，以及来自数据缓冲器 222(图 7b)的速率和 LSP、I、G、L 和 b 值将输出送到传输信道数据处理子系统 234。在该项实施中数据以最小量直接从数据缓冲器 222 送到译码器 234 则 PN 发生器 4 位数则从 PN 发生器 670 经数据组合器 676 送到译码器 234，在全速率时包括 CRC 位在内随同帧数据作为从数据组合器 674 的输出，而在八分之一速率该码书指数丢下并由随机 4 位数取代。

在该例示实施例中最好向码书增益符号位提供保护。对该位的保护会使声码器的译码器对在该位的单个位的误码不敏感。如果因一个未检出的误码而使该符号位改变，该码书指数就会指向与最优值不相关的矢量。在误码情况下没有保护，就会选取该最优矢量的相反量，实质上为所要采用的可能矢量的最差矢量。这里所采用的保护方案确保增益符号位中的单个位误码不会使该最优矢量的相反量在该误码情况下被选用。数据保护逻辑 674 接收该码书指数和增益并检查该增益值的符号位。若确定出增益值符号位是相反的对相关联的码书指数加该值 89，取模 128。该码书指数不论是否修改都从数据保护逻辑 674 输出到数据组合器 676。

在该例示实施例中最好在全速时例如由一内部 CRC(循环冗余校验)保护所压缩的话音包数据在感觉上最敏感的位。11 特殊位被用来进行可校正所保护的任何单个误码的这种误码校验功能。所保护的块包括 10 个 LSP 频率的最重要位和 8 个码书增益值的最重要位。若在该块中出现无法校正的误码，该包则被废弃并标明下面述及的消除记录。否则该音调增益设为 0 而余下的参数仍用收到的值。在例示实施例中让一生成多项式：

$$g(X)=1+X^3+X^5+X^6+X^8+X^9+X^{10}$$

产生一(31, 21)循环码来选出循环码。但应注意的是也可用其他生成多项式。附加一总的奇偶校验位使它为-(32, 21)码。既然仅有 18 个信息位，则该码字中的前 3 位置零并且未被传输。该项技术提供附加的保护，这样若校正子表明在这些位置有误码，这就意味着有无法校正的误码。系统地对循环码的编码涉及当  $X^{10} \mu(X)$  模  $g(X)$  奇偶校验位的计算则  $\mu(X)$  是信息多项式。

在译码结束时，该校正子是用  $g(X)$  除所接收到的矢量计算得出的余数。若该校正子未显示有误码，则不论总的奇偶校验位的状态均接收该包。若该校正子显示有单个误码，若未对总的奇偶校验位的状态检查则该误码被校正若该校正子显示有多于一个的误码，则该包被废弃。关于这些误码防护方案的更详细内容可在林和科斯特罗的“误码控制编码：基础及应用”4.5 节中校正子计算的详细内容中找到。

在 CDMA 蜂窝式电话系统实施中为在 20 秒数据传送帧内传送将数据从数据组合器 674 送到传输信道数据处理子系统 238 用于数据压缩。在其中声码器设定为全速率的传输帧中，由于有效位速率为 9.6Kbps 故传输 192 位。本例中的传输帧包括 1 位用以指示混合帧形式(0=仅为话音，1=话音和数据/信号码)的混合方式位；160 位声码器数据随同 11 位内部 CRC 位；12 位外部或帧 CRC 位；和 8 位尾部位或嵌入位。在半速率时，由于 4.8Kbps 的有效位速率故传送 80 位声码器数

据位以及 8 位帧 CRC 位和 8 位尾部位。在四分之一速率时，由于 2.4Kbps 的有效位速率故传送 40 位声码器数据位以及 8 位尾部位。最后在八分之一速率时，由于 1.2Kbps 的有效位速率故传送 16 位声码器数据位以及 8 位尾部位。

要应用本发明声码器的 CDMA 系统中所采用的调制的更详细细节在 1990 年 6 月 25 日申请的申请号为 07/543, 496, 转让给本发明的受让人的题为“用于产生 CDMA 蜂窝式电话系统中信号波形的系统和方法”的共同待批专利申请中揭示。在这种系统中在不是全速率的速率时采用了数据位组织成位组而位组伪随机地置于 20ms 的数据传送帧中的方案。应该理解除了那些用于这里对声码器和 CDMA 系统实施作说明的还可现成地采用其他帧速率和位表示，这样别的实施例可用于声码器和其他系统的应用。

在 CDMA 系统与也适用的其他子系统中，以一帧接一帧为基础的处理子系统 238 可中断声码器数据的传输以传输诸如信号码数据或其他非语音数据之类的其他数据。传输情况的这种特殊形式是指“间隔和突发”。处理器子系统 238 必须用在该帧所要传输的数据取代该声码器数据。

会出现另一种情况人们希望在同一数据传输帧期间传输声码器数据和其他数据。传输情况的这种特殊形式是指“迟缓和突发”。在“迟缓和突发”传输中，该声码器没有速率约束指令以设定声码器最终速率在所需要的速率上，例如半速率。该半速率编码的声码器数据送给处理器子系统 238 而为该数据传输帧插入随同声码数据的附加数据。

提供给全双工电话线路的另一个附加功能是速率联锁。若线路的一个方向以最高的传输速率传输，那么上线路的另一个方向则被迫以最低的速率传输。甚至以最低速率，对于主动谈话者有足够的理解以意识到他正被打断和要停止谈话是可行的，藉此允许该线路别的方向估计主动谈话者一方。而且若主动谈话者在企图中断时继续谈话，因为他自己的语音干扰了感觉质量的能力也可能会未感觉出质量的降级。该声码器还能通过采用速率限制指令设定成以低于通常速率的速率对话音进行声音编码。

应该理解当在 CDMA 系统中需要额外的容量时该速率限制指令可设定该声码器最大速率低于全速率。在其中公共频谱用于传输的 CDMA 系统中某用户信号会对该系统中的别的用户有干扰。系统用户容量因而由系统用户引起的总干扰所限制。随着干扰水平的上升，通常是由于在该系统中的用户的增长，由于干扰的增加用户体验到质量的降级。

各个用户对 CDMA 系统的干扰的贡献是用户传输数据速率的函数。通过设定声码器以低于通常的速率对语音编码，该编码数据随后以相应降低的数据速率传输，以降低由该用户所引起的干扰的水平。所以通过以较低速率对语音进行声音编码系统容量会得到实质性的提高。随着系统的需求上升，可由系统管理机或蜂窝基地站指令用户声码器降低编码速率。本发明的声码器具有一种良好的质量，语音以全速和半速编码时虽然两者可感觉有些差异但量很小。因而语音以例如半速率的较低速率进行声音编码对系统用户间的通信质量的后果不及因系统用户数量的上升所引起的干扰水平的上升所带来的后果严重。

因而可采用各种方案为低于通常的声音编码的速率来设定各自的声音编码器速率界限。例如可指令在蜂窝中的所有用户以半速率对语音编码。这种办法实质上以很小的用户间通信质量的损失降低系统的干扰，而为额外客户在容量上提供实质的增长。直到由额外用户引起的系统的总干扰量上升到降级水平故而不会对用户间的通信质量形成冲击。

如前所述，为实现对语音样本帧编码的分析一合成技术该编码器包括一个与译码器相同的复本如图 7 所示，译码器 234 接收经过数据压缩子系统 238 或数据缓冲器 222 的 L, b, I 和 G 值以重建合成的语音来与输入语音比较。译码器的输出如前所述为  $M_p$ ,  $M_a$  和  $M_w$  值。用于编码器和在传输信道另一端重建合成语音的译码器 234 的更详细细节可参见图 19-24 说明。

图 19 是本发明译码器例示性实施例的流程图。在编码器中和在接收机中所实施的译码器具有共同的结构，因而这些实施例在一起说明。对图 19 的说明主要针对在传输信道一端的译码器，因为在那接收到的数据必须在译码器中预处理正象在编码器的译码器中直接从数据压缩子系统 238 或数据缓冲器 222 收到合适数据(速率, I, G, L 和 b)。不过对于实施编码器和译码器来说译码器的基本功能是相同的。

如前面参见图 5 所讨论的那样，对各个码书子帧由该码书指数 I 所规定的码书矢量是从存储的码书中检索出的。该矢量以该码书增益 G 随后对每音调子帧由音调滤波器滤波产生共振峰余量。该共振峰余量由共振峰滤波器滤波随后通过一自适应共振峰后置滤波器和一清晰度后置滤波器，以及自动增益控制(AGC)以产生输出语音信号。

尽管码书和音调子帧的长度是变化的，为方便实施在 40 样本块中进行译码。所收到的压缩数据首先被分解成码书增益，码书指数，音调增益，音调滞后和 LSP

频率。该 LSP 频率如参见图 22 所说明的那样必须经他们各自的逆向量化器和 DPCM 译码器处理。类似地码书增益值除了没有偏置必须以与 LSP 频率相似的方式处理。音调增益值也被逆向量化。这些参数随后送给各个译码子帧。在各个译码子帧中, 2 套码书参数(G 和 I), 1 套音调参数(b 和 I)和一套 LPC 系数需产生 40 个输出样本。图 20 和 21 对于各种速度和其他帧状况示出例示性子帧译码参数。

对于全速率帧, 有 8 套收到的码书参数和 4 套收到的音调参数。LSP 频率被内插 4 次产生 4 套 LSP 频率。所收到的参数和相应的子帧信息列于图 20a 中。

对于半速率帧, 4 套收到码书参数的各套重复一次, 2 套收到的音调参数的各套重复一次。LSP 频率被内插 3 次产生 4 套 LSP 频率。所收到的参数和相应的子帧信息列于图 20b 中。

对于四分之一速率帧, 2 套收到的码书参数的各套重复 4 次, 该套音调参数也被重复 4 次。LSP 频率被内插一次产生 2 套 LSP 频率。所收到的参数和相应的子帧信息列于图 20c 中。

对于八分之一速率帧, 该套收到的码书参数用于整个帧。音调参数对于八分之一速率未给出且该音调增益简单地设为零。该 LSP 频率内插 1 次产生 1 套 LSP 频率。所收到的参数和相应的子帧信息列于图 20d 中。

有时, 话音包可按序空出让 CDMA 单元或移动台传输信号码信息。当声码器收到一空白帧, 它继续对前一帧参数作较小的改动。码书增益设为零。前一帧的音调滞后和增益用于当前帧的音调滞后和增益, 除非该增益被限定成某值或更小。就像没有内插而用前一帧的 LSP 频率。注意编码端和译码端一直被同步且声码器能很快从一空白帧复原。所收到的参数和相应的子帧信息列于图 21a。

在因信道差错而丢失一帧的情况下, 该声码器试图通过保持前一帧的能量部分且平滑地过渡到背景噪声来屏蔽该差错。在该例中音调增益设成零; 通过用前一帧的码书指数加 89 选取一随机码书; 码书增益则是前一子帧的码书增益的 0.7 倍。应该注意的是数 89 没有什么特别的东西, 这只是选取伪随机码书矢量的通常方法。该前一帧的 LSP 频率被迫向他们偏置值衰落为:

$$\omega_i = 0.9(\text{前面的 } \omega_i - \omega_i \text{ 的偏置值}) + \omega_i \text{ 的偏置值} \quad (47)$$

该 LSP 频率偏置值在表 5 中示出。所收到的数据和相应的子帧信息列于图 21b 中。

若在接收机中未能确定出速率, 该包被废弃并标上消除记录。然而若接收机确定有很大可能该帧是以全速率传输的, 尽管带有差错仍进行下面的过程。如前所

述在全速率时该压缩的话音包数据的感觉上最敏感的位由内部 CRC 保护。在译码端正如接收到的矢量除以来自公式(46)的  $g(X)$  而得出余式那样计算出校正子。若校正子显示无误码，则不论总的奇偶校验位而接收该包。若该校正子显示单个误码，若总的奇偶校验位不检查该误码被校正。若该校正子显示多于一个误码。该包则被废弃。若在该块中出现无法校正的误码，该包被废弃且标明消除记录。否则该音调增益被置零而其余的参数如图 21c 所示随着校码被接收而被采用。

本实施例中所采用的后置滤波器首先在 J.H. 陈等人的“用自适应后置滤波以 4800BPS 实时向量 APC 语言编码”(Proc.ICASSP, 1987) 中揭示。既然语音共振峰主频成分比起频谱凹谷在感觉上更为重要，该后置滤波器促使该共振峰主频成分稍微提升以改善该编码语音的感性质量。通过相对于原来的径向地对共振峰主频成分的极点定标来进行。然而全极点后置滤波器通常引入频谱倾斜而导致滤波后的语音消音。通过增加具有与极点相同相位角而较小直径的零点来减少该全极点后置滤波器的频率的倾斜，从而后置滤波器具有形式：

$$H(z) = \frac{A(z/\rho)}{A(z/\sigma)} \quad 0 < \rho < \sigma < 1 \quad (48)$$

这里  $A(z)$  是共振峰主频成分预测滤波器，且  $\rho$  和  $\sigma$  值为后置滤波器定标因数。这里  $\rho$  设为 0.5， $\sigma$  设为 0.8。

加上一自适应清晰度滤波器就能进一步补偿由该共振峰主频成分后置滤波器引进的频谱倾斜。该清晰度滤波器具有形式：

$$B(z) = \frac{1 - Kz^{-1}}{1 - Kz^{-1}} \quad (49)$$

这里  $K$  值(分接滤波器的系数)由与  $A(z)$  的频谱倾斜中的变化相近似的 LSP 频率平均值确定。

为避免因后置滤波器引起的任何大的增益漂移，采用 AGC 环路对语音输出定标使它具有与未后置滤波的语音近乎相同的能量。通过用 40 个滤波器输出样本的平方和除 40 个滤波器输入样本的平方和来实现增益控制以获得反向滤波器增益。随后使该增益因数的平方根平滑：

平滑后的  $p = 0.2$  当前  $\beta + 0.98$  前一个

然后滤波器输出乘以该平滑后反向增益以产生输出语音。

在图 19 中来自信道的数据以及随同数据传输或由其他装置得到的速率被送给数据压缩子系统 700。在一个 CDMA 系统的实施例中当以各种不同速率进行译码时就可从所收到的数据的误码率来得到速度判定。在数据分解子系统 700 中，在

全速率时对差错进行 CRC 检码因而该检码送到子帧数据分解子系统 702。子系统 700 将诸如空白帧，消除帧或具有可用数据的差错帧之类异常帧状况的标志送到子系统 702。子系统 700 将对于该帧的速率以及 I, G, L, 和 b 参数送到子系统 702。在提供码书指数 I 和增益 G 值时，该增益值的符号位在子系统 702 中被检码。若符号位相反，从关联的码书指数减去值 89，取模(mod)128。而且在子系统中该码书增益被反向量化且 DPCM 编码，同时音调增益被反向量化。

子系统 700 还将速率与 LSP 频率送到 LSP 反向量化/内插子系统 704。子系统 700 还将空白帧、消除帧或有可用数据的差错帧的标志送到子系统 704。编码子帧计数器 706 将子帧计数值 i 和 j 的标志送到子系统 702 和 704。

在子系统 704 中该 LSP 频率被反向量化和内插。图 22 示出子系统 704 的反向量化部分的实施例，而该内插部分与参见图 12 说明的那种在实质上是相同的。在图 22 中，子系统 704 的反向量化部分包括反向量化器 750，它与图 12 中的反向量化器 468 结构相同且以相似方式工作。反向量化器 750 的输出作为输入送到加法器 752。加法器 752 的其他输入则由乘法器 754 的输出提供。加法器 752 的输出送给寄存器 756 存储并输出在乘法器 754 与常数 0.9 相乘。加法器 752 的输出回被送到加法器 758 将偏置值加回给 LSP 频率。由逻辑 760 确保 LSP 频率的序列使 LSP 频率具有最小间隔。除非在传输中出现差错一般不会出现强迫分隔的需要。该 LSP 频率如参见图 13 和参见图 20a-20d 和 21a-21c 说明的那样是内插值。

再参见图 19，存储器 706 与子系统 704 连接存储前一帧  $LSP_s$ ,  $\omega_{i, f_1}$ ，还可用来存储偏置值  $bw_i$ 。这些前一帧值对于所有速率都用于内插。在空白帧，消除帧或有可用数据的差错帧的情况下，按照图 21a-21c 中的表应用前一帧的  $LSP_s$ ,  $\omega_i$ ,  $\omega_{i, f_1}$ 。响应来自子系统 700 的空白帧标志，子系统 704 检取出存于存储器 708 的前一帧 LSP 频率用于当前帧。响应消除帧标志，子系统 704 随同偏置值再从存储器 708 检取出前一帧 LSP 频率以如上所述计算当前帧的 LSP 频率。在进行该计算时，在一加法器中从前一帧 LSP 频率减去所存储的偏置值，该结果在一乘法器中乘以常数值 0.9，其结果在一加法器中加到所存储的偏置值。响应具有可用数据的差错帧标志，对于全速率若 CRC 通过该 LSP 频率就被内插。

该  $LSP_s$  送到 LSP 至 LPC 转换子系统 710 将 LSP 频率变换回 LPC 值。子系统 710 实质上与图 7 的 LSP 至 LPC 转换子系统 218 和 228 相同并如参见图 13 说明的那样。该 LPC 系数  $a_i$  随后送给共振峰主频成分滤波器 714 和共振峰主频成分后置滤波器 716。该 LSP 频率在 LSP 平均值子系统还在子帧上取平均并作为

值 K 送给自适应清晰度滤波器 718。

子系统 702 随同速率或异常帧状态标志从子系统 700 接收该帧的参数 I、G、L 和 b。子系统 702 还从子帧计数器收到对于各个译码子帧 1-4 中的每个 i 计数的 j 计数。子系统 702 还连接存储器 720 以存储用于异常帧状态的 G, I, L 和 b 的前一帧值。子系统 702 处于正常帧状态，除去八分之一速率，要将码书指数值  $I_j$  送到码书 722；将码书增益  $G_j$  送到乘法器 724，音调滞后 L 和增益 b 值按照 20a-20d 送到音调滤波器 726。对于八分之一速率，由于所送的码书指数没有值，对于八分之一速率，由于所送的码书指数没有值，对于八分之一速率是 16 位参数值的包种籽随同度标志送到码书 722。在异常帧状态子系统 702 按照图 21a-21c 提供这些数据。而且对于八分之一速率，如参见图 23 所说明的那样向码书 722 提供标志。

响应来自子系统 700 的空白帧标志，子系统 702 检取出存在存储器中的前一帧音调滞后 L 值和增益 b 值除了该增益限制为某值或小些，以用于当前帧的译码子帧。而且未提供码书指数 I 且码书增益 G 被设为零。响应消除帧标志，子系统 702 再从存储器 720 检取出前一帧子帧码书指数并在加法器加 89 的值。该前一帧子帧码书增益在乘法器乘以常数 0.7 以产生各自子帧的 G 值。未提供音调滞后值并将音调增益置零。响应具有可用数据的差错帧标志，当在全速率帧中假使 CRC 通过就运用码书指数和增益，而不提供音调滞后值且音调增益置零。

如参考分析-合成技术中的编码器的译码器所说明的那样，该码书指数 I 用作该码书值的起动地址以向乘法器 724 输出。该码书增益值在乘法器 724 乘以来自码书 722 的输出值该结果送到音调滤波器 726。音调滤波器用输入的音调滞后 L 值和增益 b 值以产生输出给共振峰主频成分滤波器 714 的共振峰主频成分余量。在共振峰主频成分滤波器 714 LPC 系数用来对共振峰主频成分余量滤波以重建语音。在接收端的译码器重建语音进一步由共振峰主频成分后置滤波器 716 和自适应清晰度滤波器 718 滤波。在共振峰主频成分滤波器 714 和共振峰主频成分后置滤波器 716 的输出采用 AGC 回路，其输出在乘法器 730 中乘以自适应清晰度滤波器 718 的输出。乘法器 730 输出的重建语音利用熟知技术变换为模拟形式并送给听众。在编码器的译码器中，为更新它的记忆，在输出端设置感知加权滤波器。

参见图 22，说明实施译码器本身的更详细节。在图 22 中码书 722 包括与参见图 17 所说明的那种相类似的存储器 750。然而为便于说明，对于存储器 750。说明稍有差异的方法并在图 22 中示出其编址。码书 722 还包括开关 752，多路转换器 753 和伪随机数发生器(PN)754。开关 752 如参见图 17 说明的那样响应该码书

指数以指向存储器 750 的指数地址位置。存储器 750 是有开关 752 指向起始记忆位置的循环存储器，数值移过存储器而输出。该码书数值经开关 752 从存储器 750 输出作为多路转换器 753 的输入。多路转换器 753 响应全速率，半速率和四分之一速率以将经开关 752 提供的数值送到码书增益放大器，乘法器 724。多路转换器 753 还响应八分之一速率标志以选取 PN 发生器 754 的输出使码书 722 的输出送到乘法器 724。

为维持在 CELP 编码的高话音质量，编码器和译码器必须有相同数值存储在他们的内部滤波器存储器中。这可通过传输码书指数进行，使译码器和编码器的滤波器由数值的相同序列激励。而对高话音质量这些序列包括大多数零，并有一些尖峰分布其间。这种激励方式对于对背景噪声编码不是最优的。

在对背景噪声的编码中，以最低速率进行，可实现伪随机序列激励滤波器。为确保该滤波器存储器在编码器和译码器中的相同，这两个伪随机序列必须相同。不管怎样必须传输种籽到接收机译码器。既然没有额外的位可用以送该种籽，所传输包位可用作种籽，似乎它们构成一个数。该技术可进行是因为是以低速率用完全相同 CELP 分析合成结构来确定码书增益和指数的。差异是该码书指数被扔掉而编码器的滤波器存储器用一伪随机序列来更新。因而分析完后可确定出用于激励的种籽。为确保包本身不在一套位模式周期地循环，在该八分之一速率的包中插入 4 位随机位来取代该码书指数值。因此该包种籽是如图 2 所见到的 16 位数据。

NP 发生器 754 用熟知技术构筑并可由各种算法实施。在例示性实施例中所采用的算法其本质在保罗·门恩的文章“DSP 片采用已证明的算法可产生随机数”(EDN, 1991, 1, 21)中说明。所传输的位包用来产生序列的种籽(来自图 18 中的子系统 700)。在一实施例中种籽乘以数值 521 且加上数值 259。从这合成数值的最不重要的位用来作为有符号的 16 位数。该数值随后用作产生下一码书数值的种籽。由 PN 发生器产生的序列被归一化而具有 1 的方差。

从码书 722 输出的各个数值在乘法器 724 中乘以如在译码子帧期间所提供的码书增益 G。该数值作为输入送给音调滤波器 726 的加法器 756。音调滤波器 726 还包括乘法器 758 和存储器 760。该音调滞后 L 确定存储器 760 分接位置被输出给乘法器 758。该存储器 760 的输出在乘法器 758 中乘以音调增益值 b，结果输出给加法器 756。加法器 756 的输出送到存储器 760 的输入端，它是诸如移位寄存器的一系列的延时单元。该数值经存储器 760 移出(沿箭头所指方向)且如 L 值所确定那样在所选取的分接输出端给出。既然该值经存储器 760 移出，比 143 移出值更

老的值被废弃。加法器 756 的输出还作为输入送到共振峰主频成分滤波器 714。

加法器 756 的输出送到共振峰主频成分滤波器 714 的加法器 762 的一个输入端。共振峰主频成分滤波器还包括乘法器 764a-764j 的存储单位和存储器 766。该加法器 762 的输出作为输入送到存储器 766，它也构建成一系列分接的延时单元如移位寄存器。该数值经存储器 766 移出(沿箭头所指方向)且最后清出每个单元有分接口以将存在那里的数据作为输入送给乘法器 764a-764j 的相应的分接口。每个乘法器 764a-764j 还接收 LPC 系数  $\alpha_1-\alpha_{10}$  的各个值来乘以来自存储器 766 的输出。来自加法器 762 的输出送给作为共振峰主频成分滤波器的输出。

共振峰主频成分滤波器 714 的输出作为输入送到共振峰主频成分后置滤波器 716 和 AGC 子系统 728。共振峰主频成分滤波器 716 包括加法器 768 和 770 以及存储器 772 和乘法器 774a-774j, 776a-776j, 和 782a-782j。随着该数值经存储器 772 移出它们在相应的分接口输出用来乘以定标的 LPC 系数值而在加法器 768 和 770 累加。共振峰主要频率成分后置滤波器 716 的输出作为输入送给自适应清晰度滤波器 718。

自适应清晰度滤波器 718 包括加法器 784 和 786, 寄存器 788 和 790, 和乘法器 792 和 794。图 24 是说明自适应清晰度滤波器特点的图表。共振峰主频成分后置滤波器 716 作为输入送给加法器 784 而另一输入由乘法器 792 的输出提供。加法器 784 的输出送给寄存器 788 并存储一循环而在下一循环随同图 19 的 LSP 平均器 712 提供的 K 值输出给乘法器 792 和 794。乘法器 792 和 794 的输出被送给加法器 784 和 786。加法器 786 的输出送给 AGC 子系统 728 和移位寄存器 790。寄存器 790 作为延时线以确保从共振峰主频成分后置滤波器 716 输出给 AGC 子系统 728 的数据与经共振峰主频成分后置滤波器 716 提供给自适应清晰度滤波器 718 的数据间的协调。

AGC 子系统 728 接收来自共振峰主频成分滤波器和自适应清晰度滤波器的数据以对语音输出能量定标至大约输入到共振峰主频成分后置滤波器 716 和自适应清晰度滤波器 718 的语音的能量。AGC 子系统 728 包括乘法器 798、800、802 和 804, 加法器 806、808 和 810, 寄存器 812、814 和 816, 除法器 818, 和平方根单元 820。从共振峰主频成分滤波器输出的 40 个样本在乘法器 798 平方并在包括加法器 806 和寄存器 812 的累加器中累加以产生数值 “X”。相似地从自适应清晰度滤波器 718 输出的 40 样本, 优先于存储器 790, 在乘法器 800 中平方并在包括加法器 808 和寄存器 814 的累加器中累加以产生数值 “Y”。该数值 “Y” 在除

法器 816 中除以数值 “X” 以产生该滤波器的反向增益。在单元 818 得到该反向增益因数的平方根其结果被平滑。通过当前增益 G 值在乘法器 802 乘以常数 0.02，该结果在加法器 810 加上如寄存器 820 和乘法器 804 所计算出的 0.98 倍的前增益的结果来实现平滑工作。滤波器 718 的输出随后在乘法器 730 乘以平滑后的反向增益以提供输出的重建语音。该输出语音用各种已知变换技术转换成模拟形式输出给用户。

应该理解这里所揭示的本发明的实施例只是例示性的实施例，对该实施例作出的改变可认为在功能上的等同物。本发明可以在合适的程序控制下在数字信息处理机中实施，提供如这里所公开的对语音样本编码并对该编码的语音译码的功能操作。在本发明的另一个实施例中可以在实用熟知的超大规模集成(VLSZ)技术的专用集成电路(ASIC)来实现。

所提供的对较佳实施例的前面的说明能使本领域技术人员运用本发明。对这些实施例的各种变动对于本领域技术人员将是现成明显的，这里确定的总原则可不需要创造性劳动就可应用于其他实施例中。因而本发明希望不是限定在这所示出的实施例而是给予与这里所说的原则和新颖特征相一致的最大保护范围。

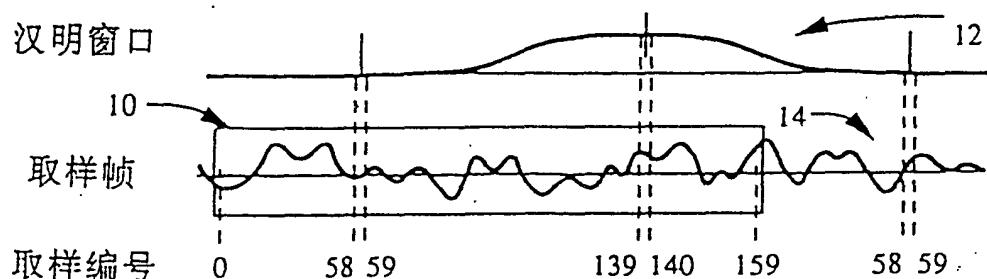


图 1a

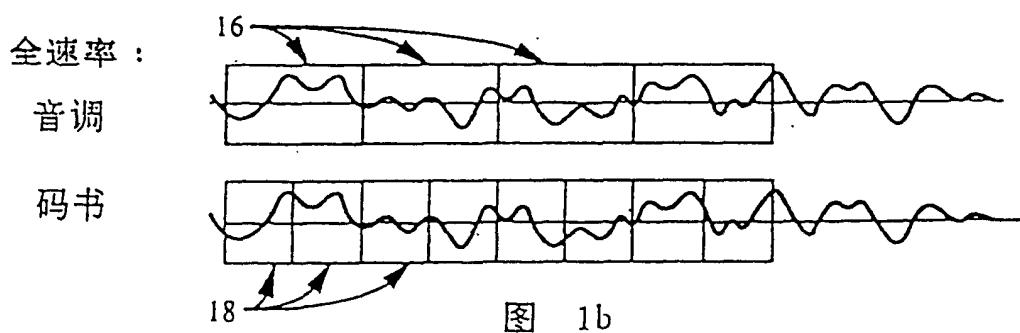


图 1b

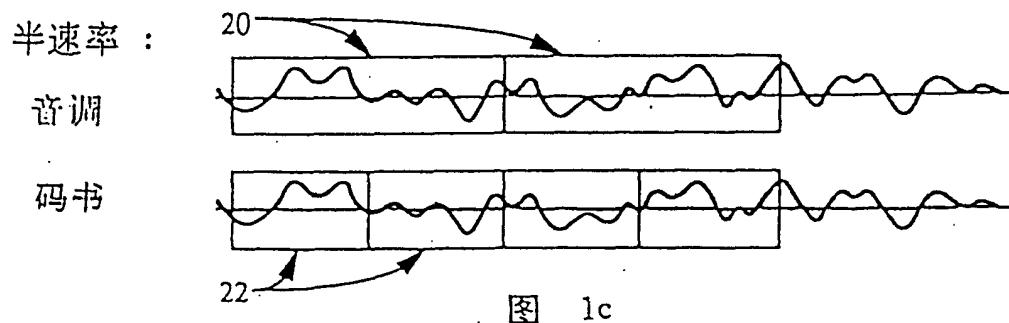


图 1c

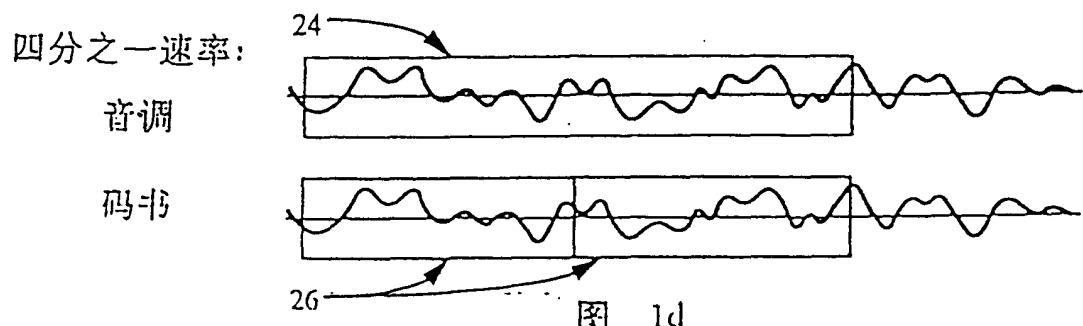


图 1d

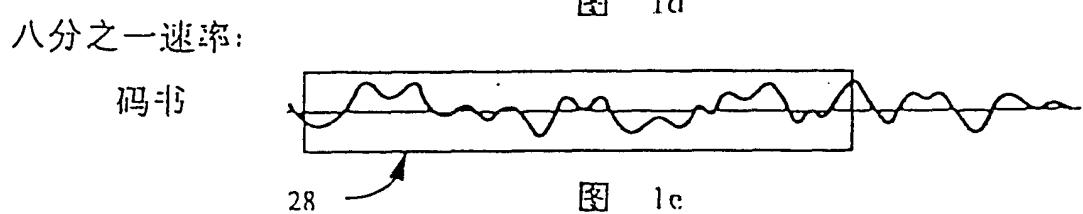


图 1e

## 全速率

LPC	40							
音调	10		10		10		10	
码书	10	10	10	10	10	10	10	10

总计 = 160  
(加 11 个 CRC)

图 2a

## 半速率

LPC	20			
音调	10		10	
码书	10	10	10	10

总计 = 80

图 2b

## 四分之一速率

LPC	10	
音调	10	
码书	10	10

总计 = 40

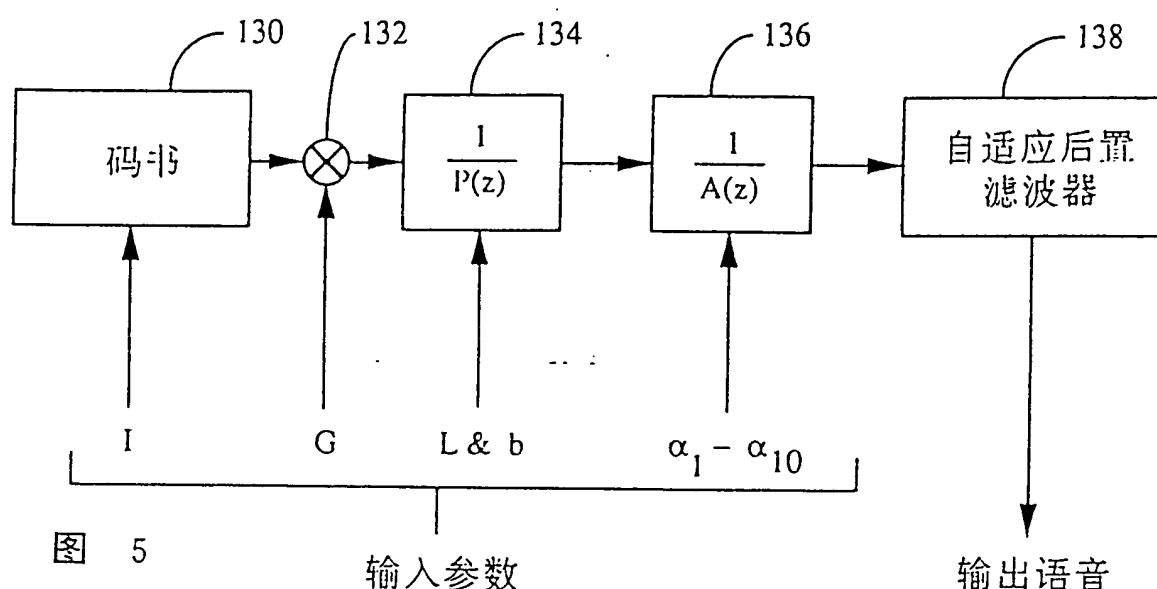
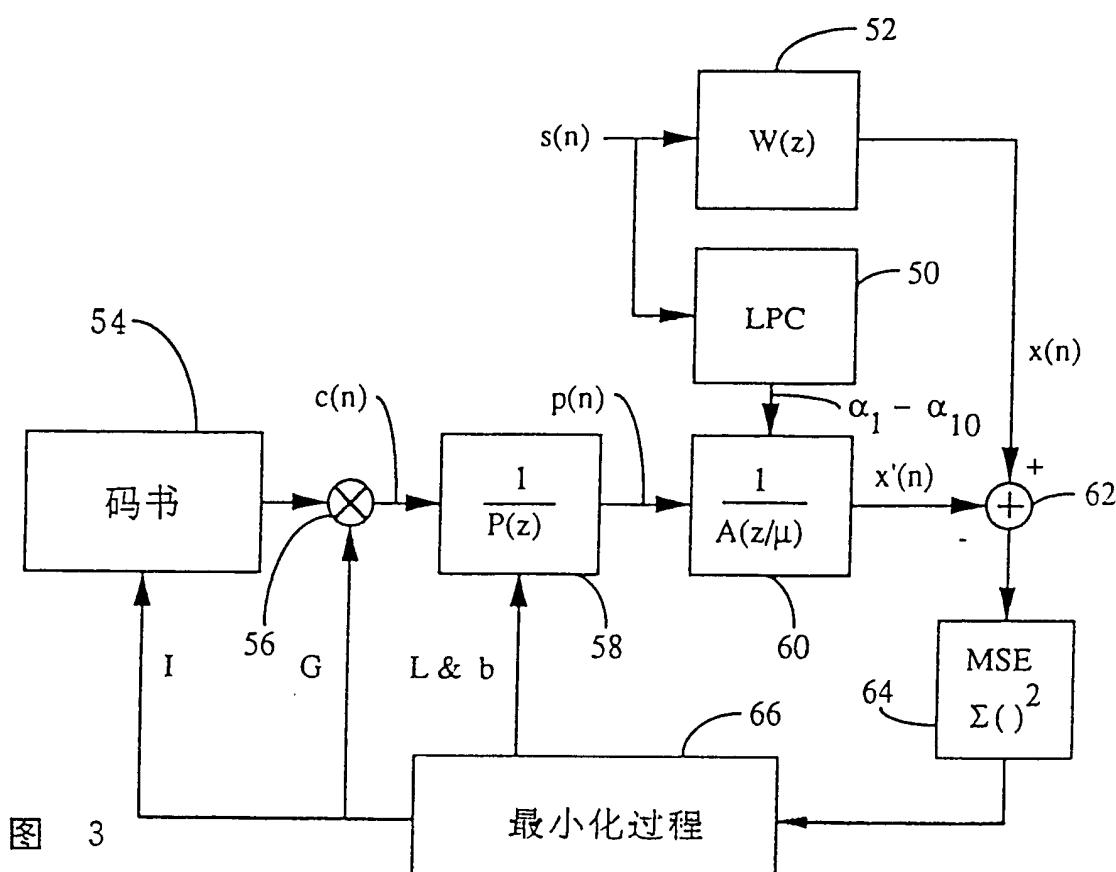
图 2c

## 八分之一速率

LPC	10	
音调	0	
码书	6	

总计 = 16

图 2d



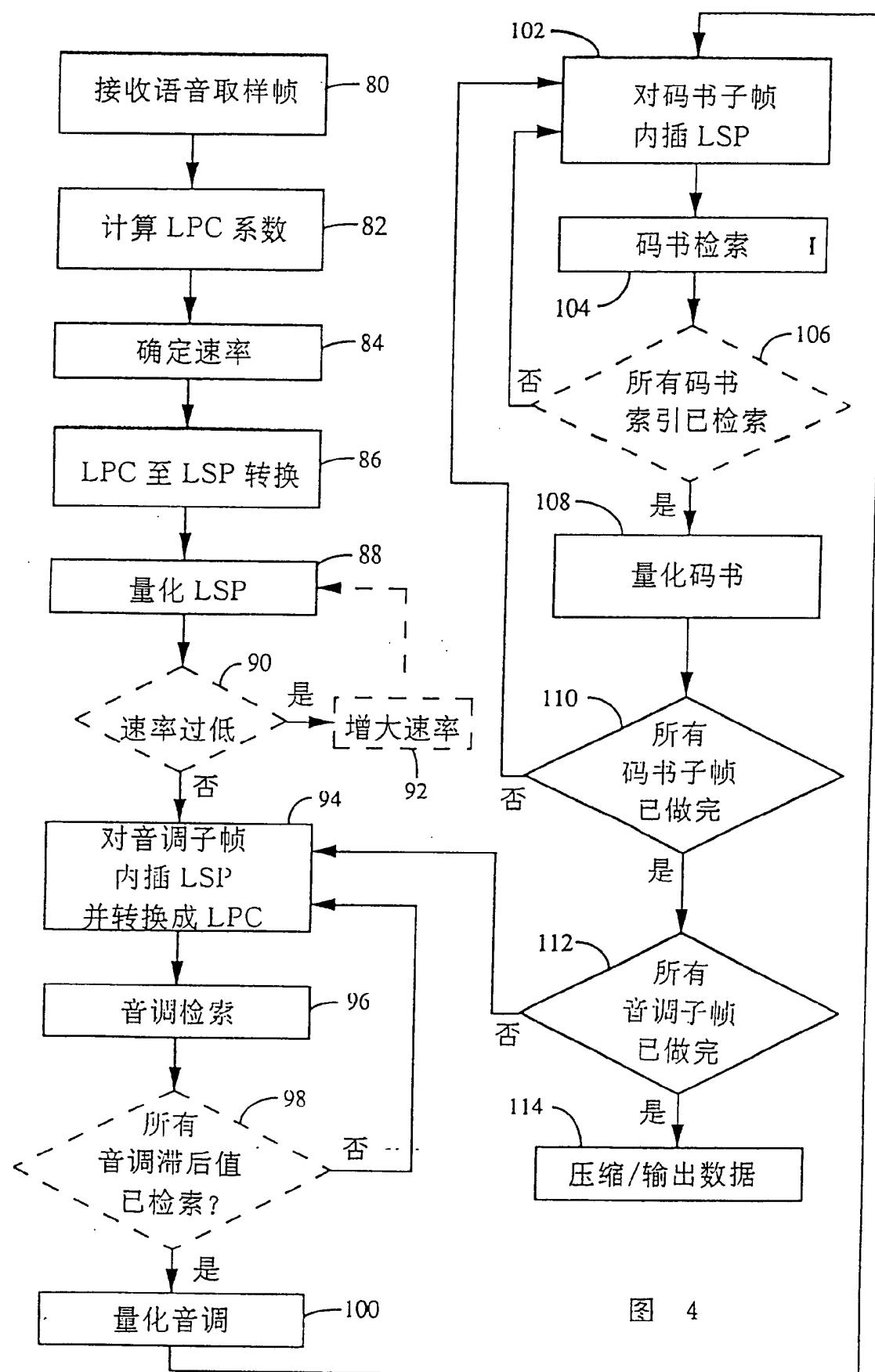


图 4

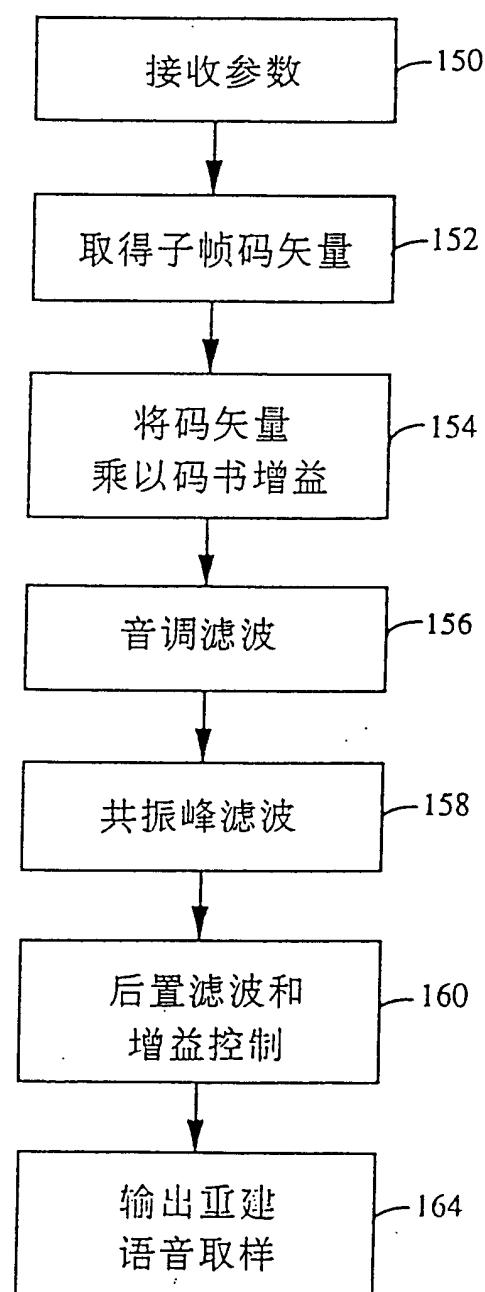


图 6

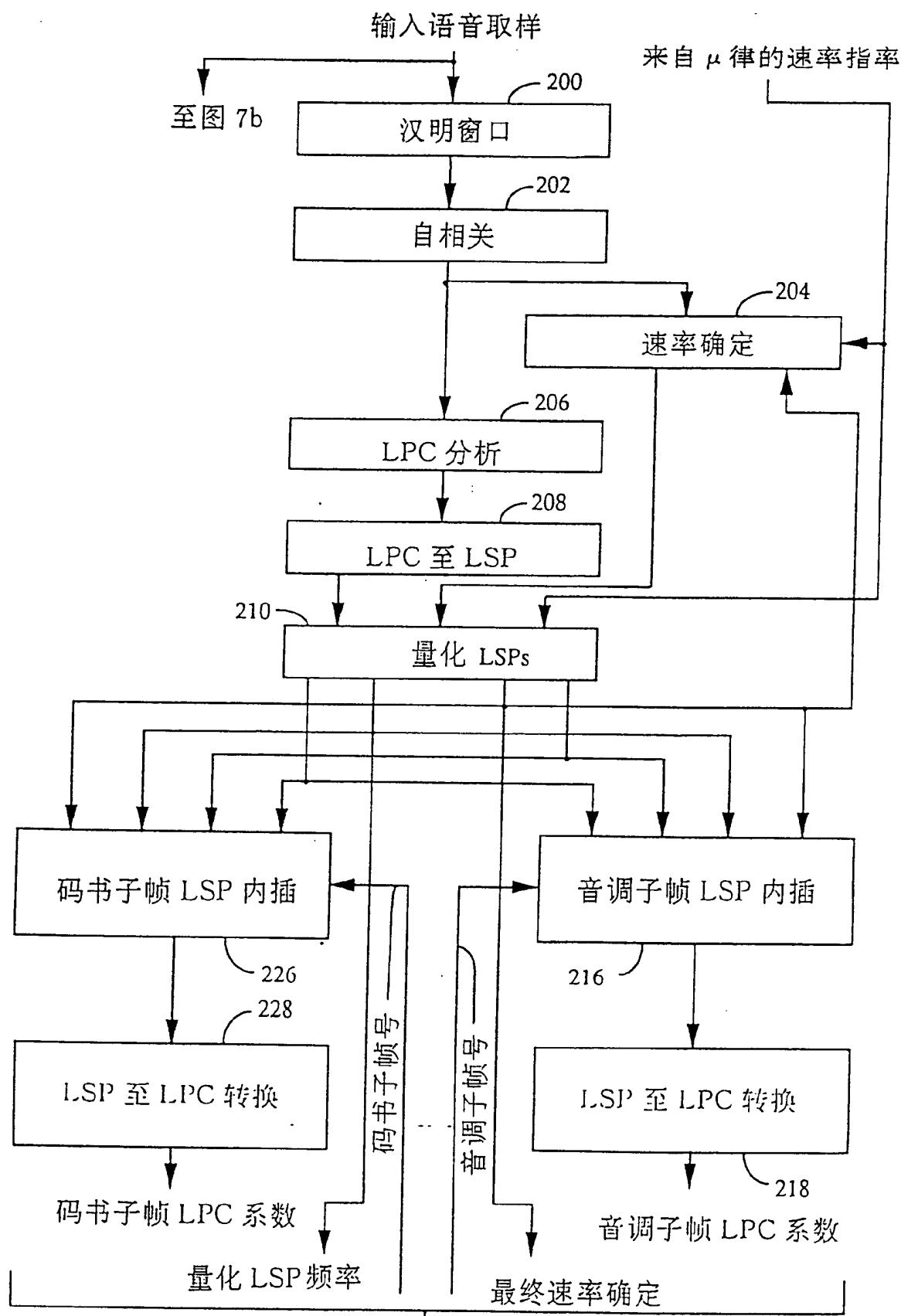
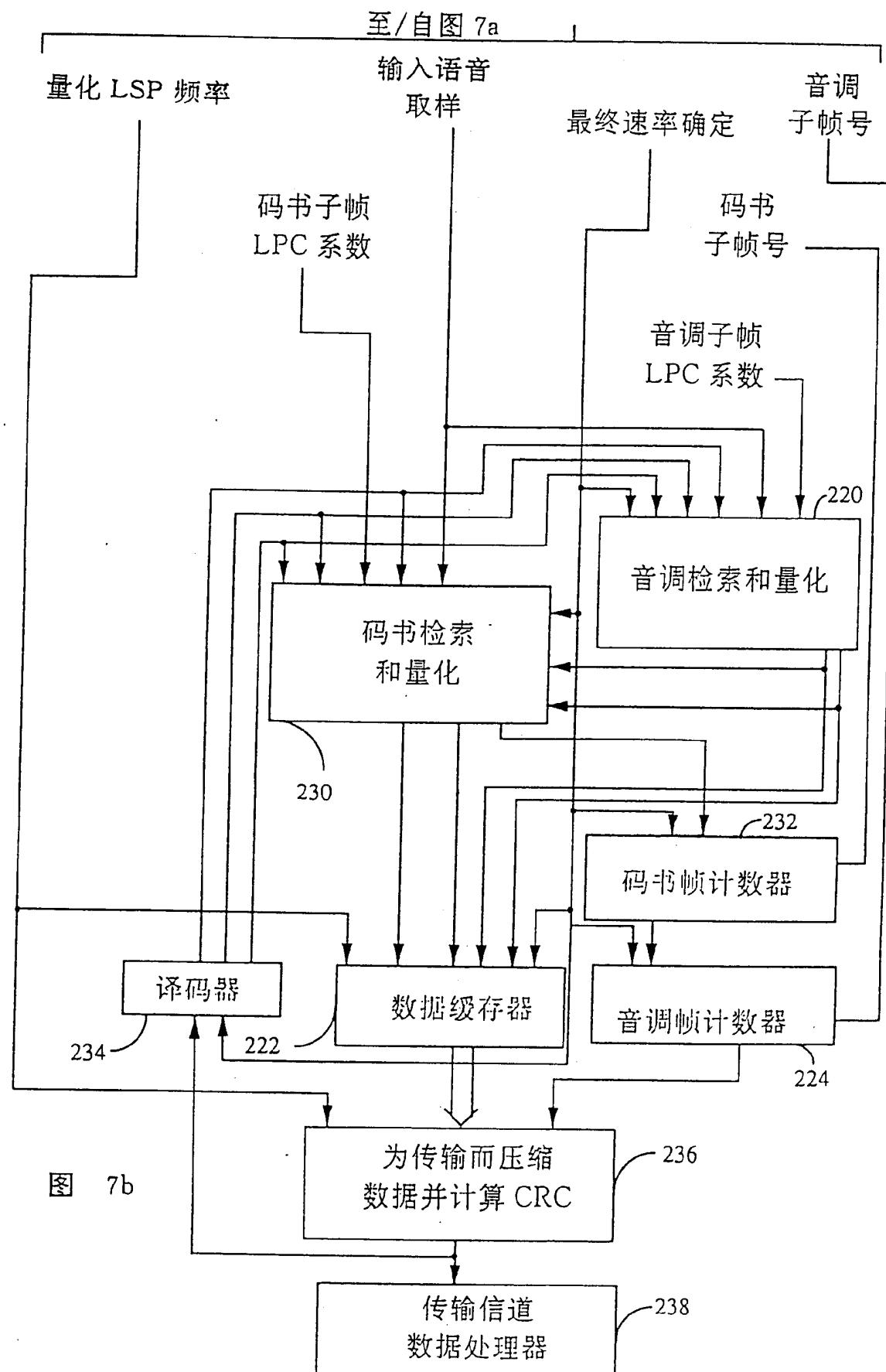


图 7a

至/自图 7b



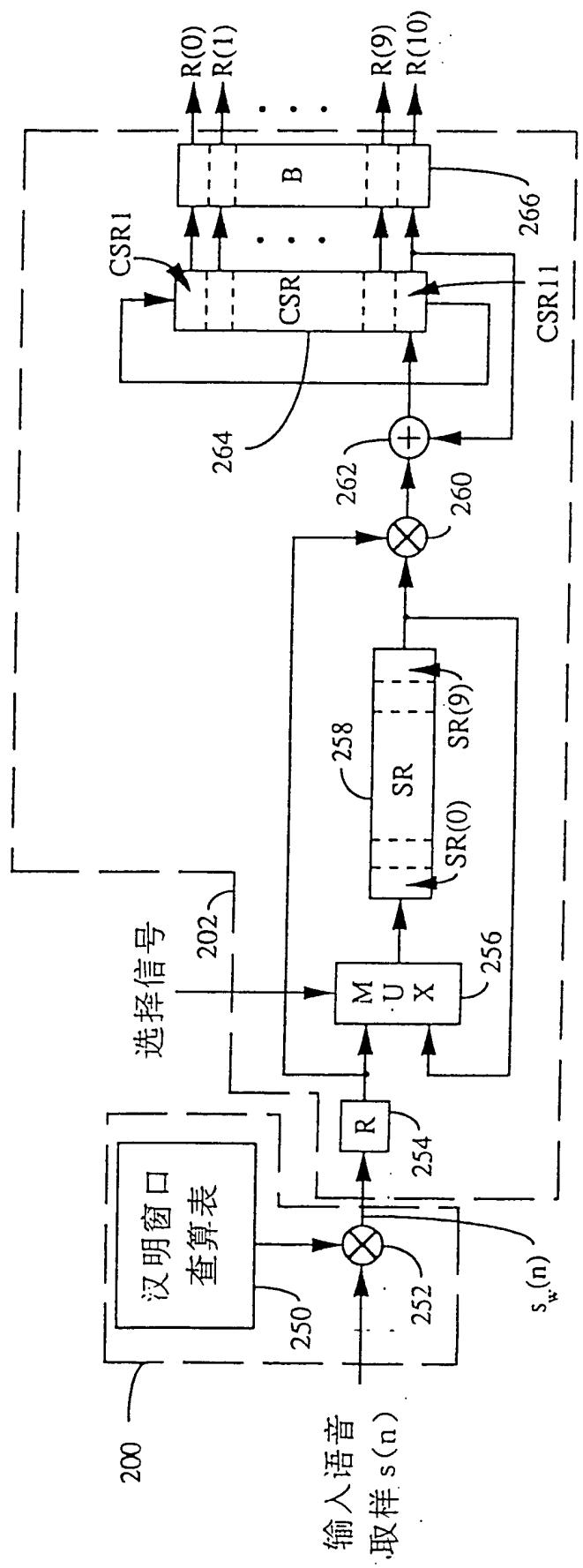
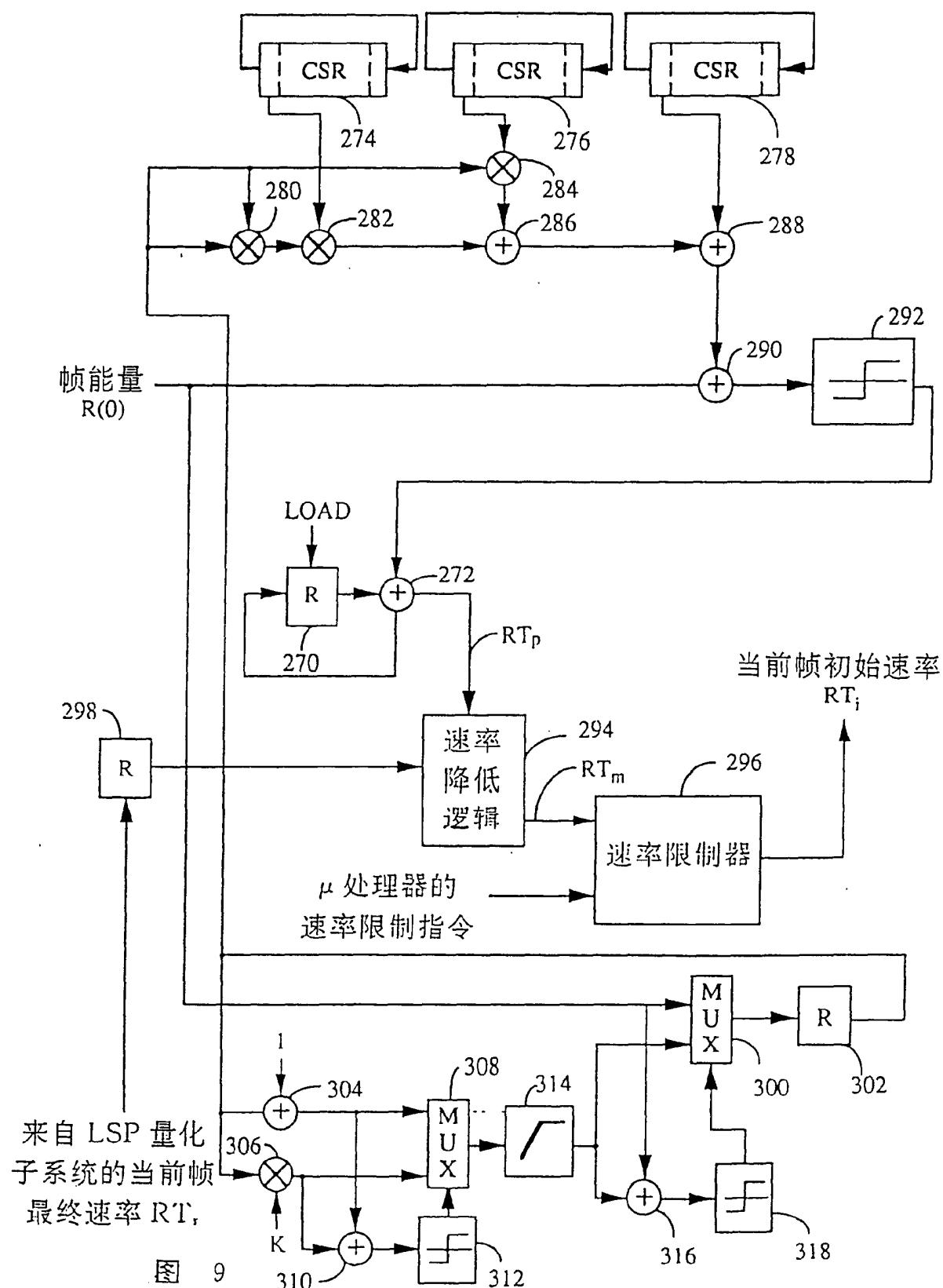


图 8



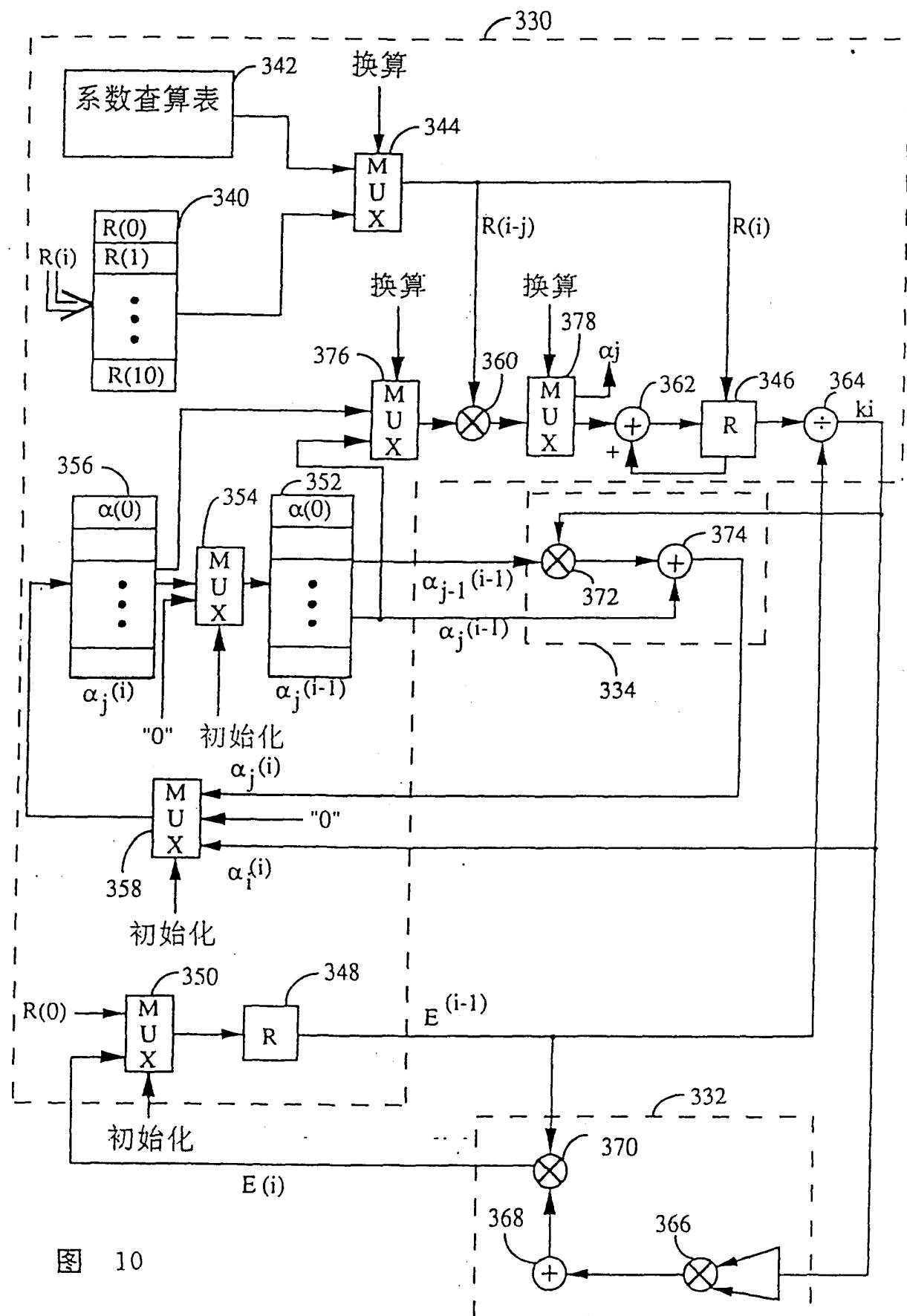


图 10

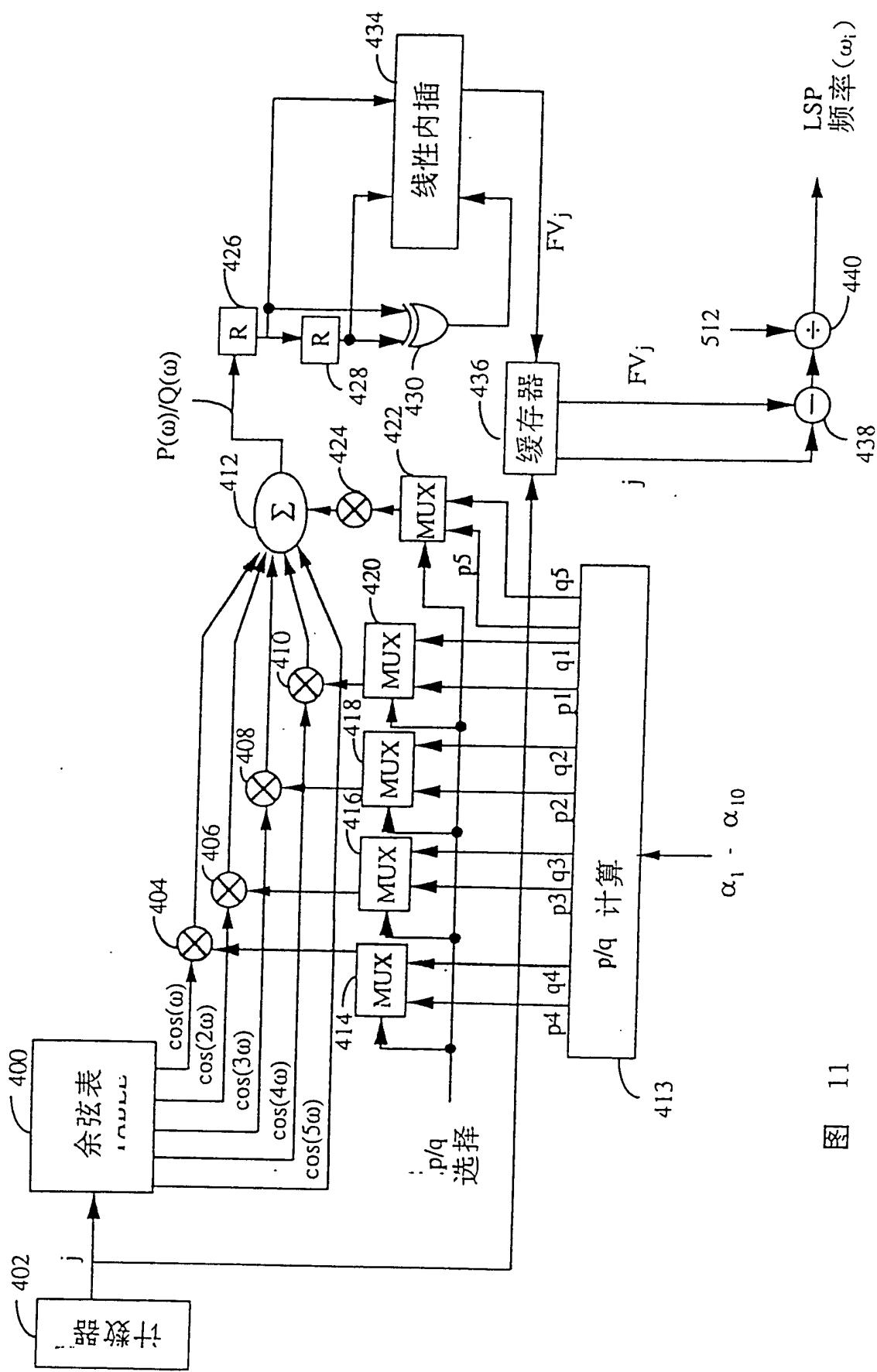


图 11

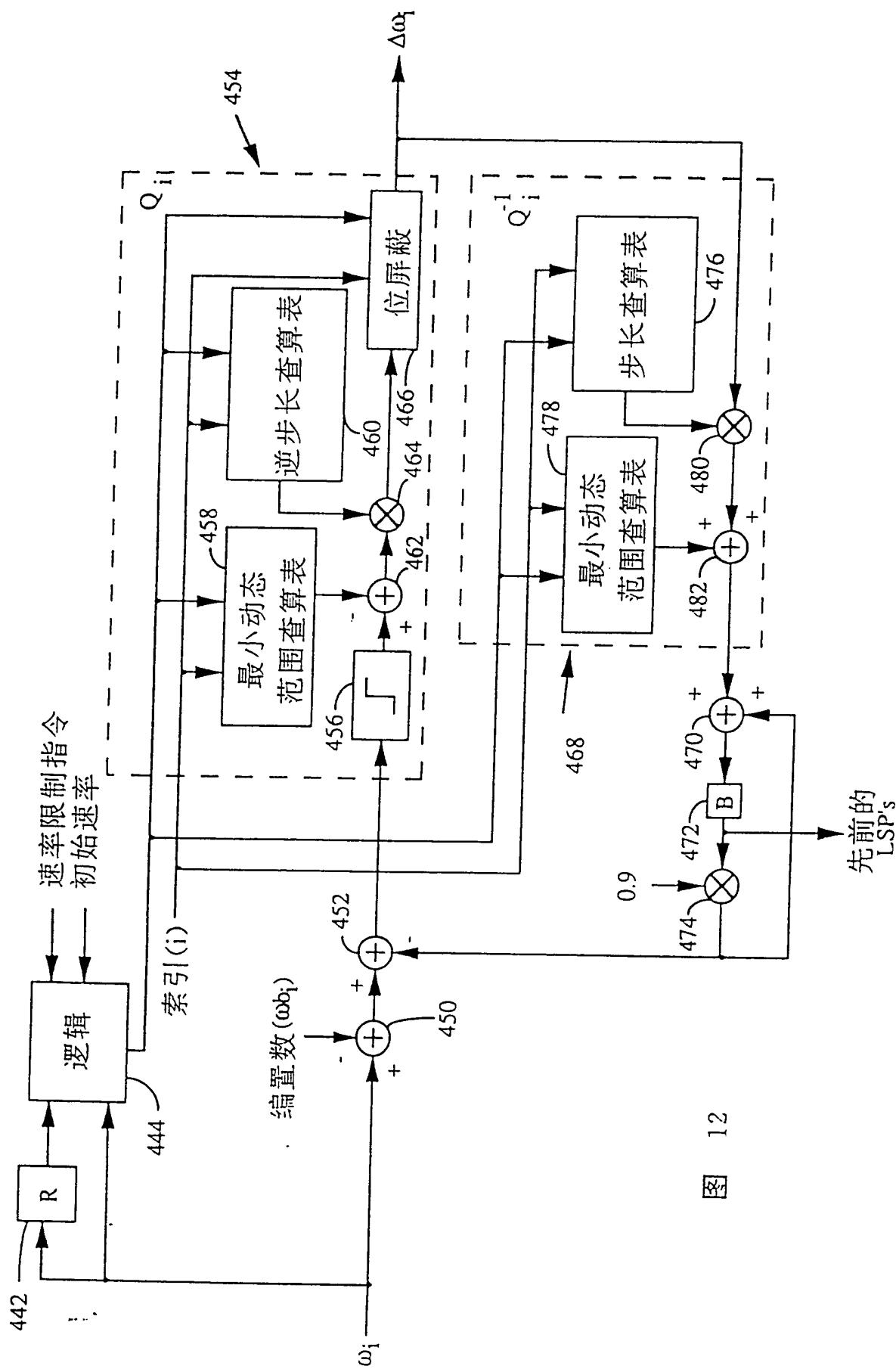


图 12

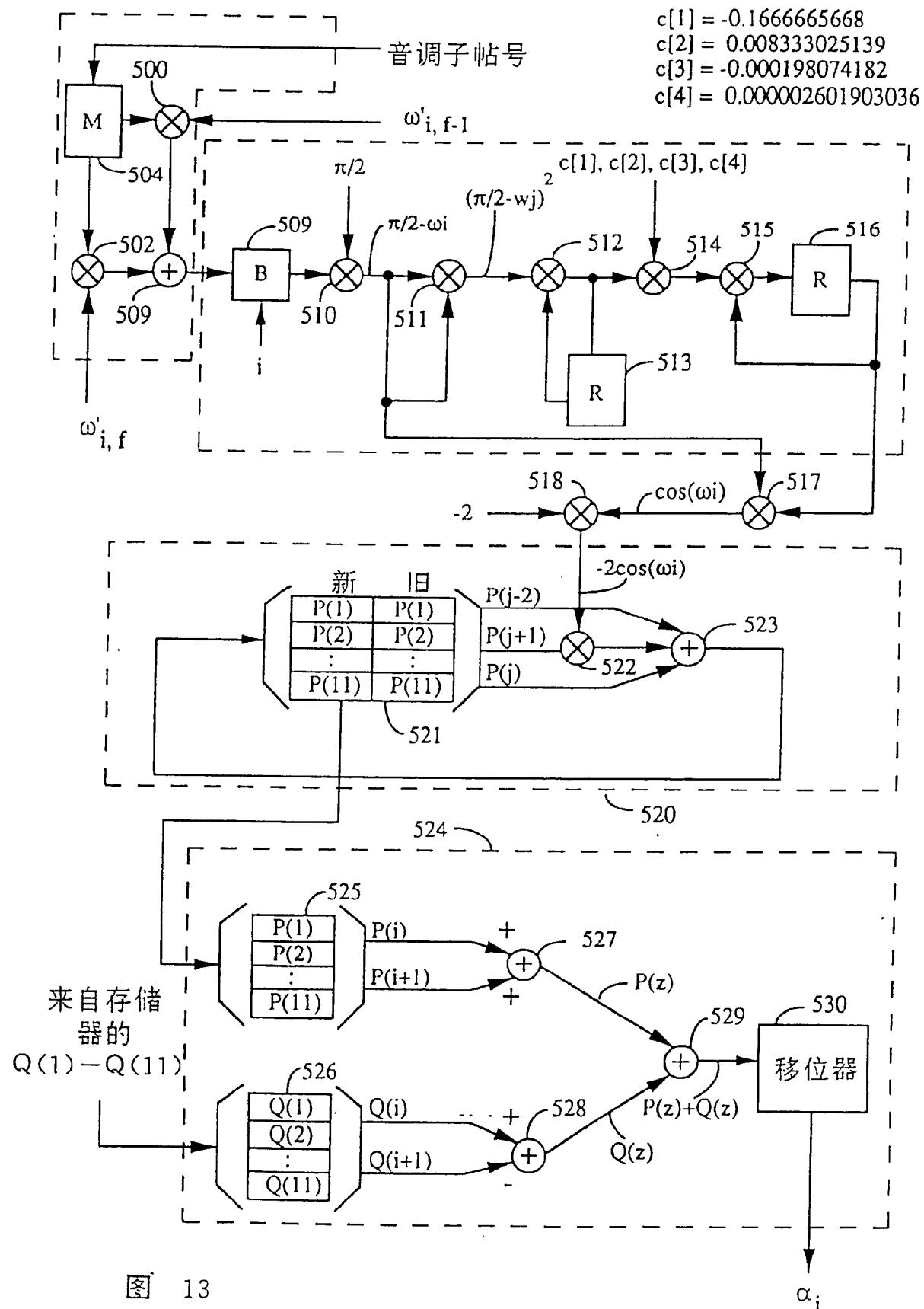


图 13

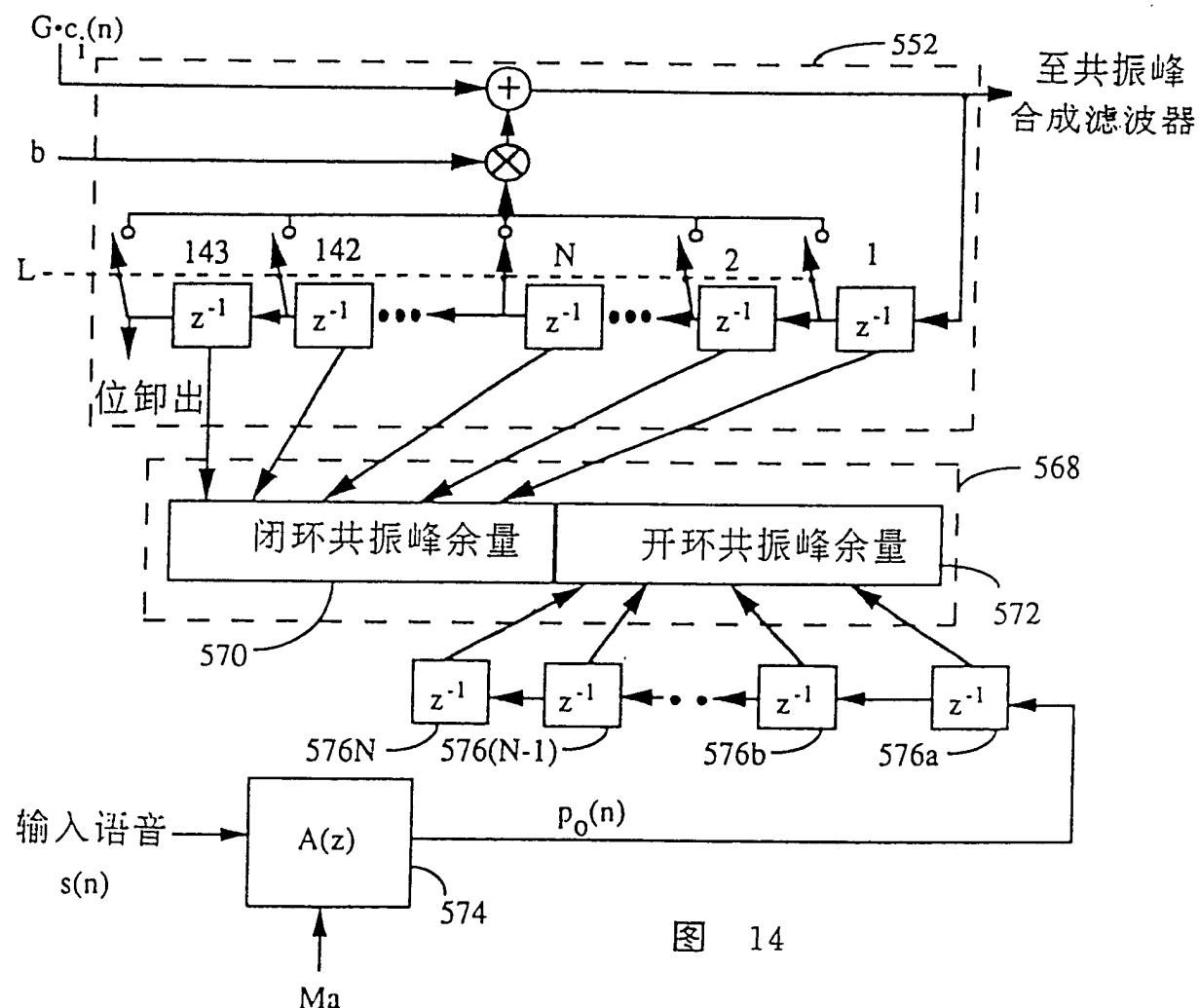


图 14

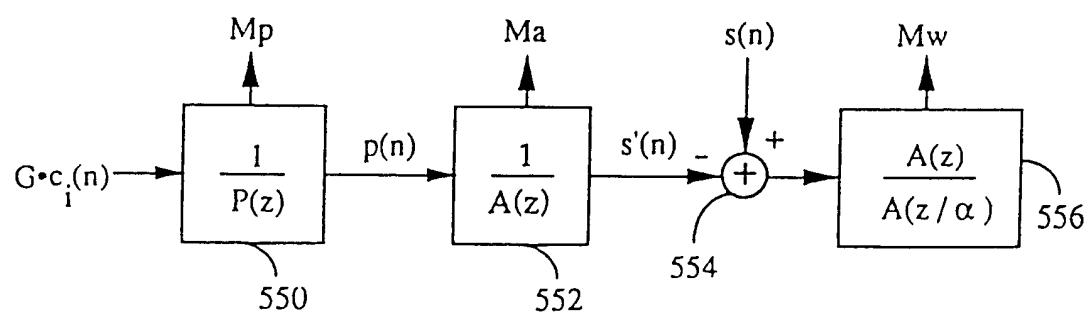
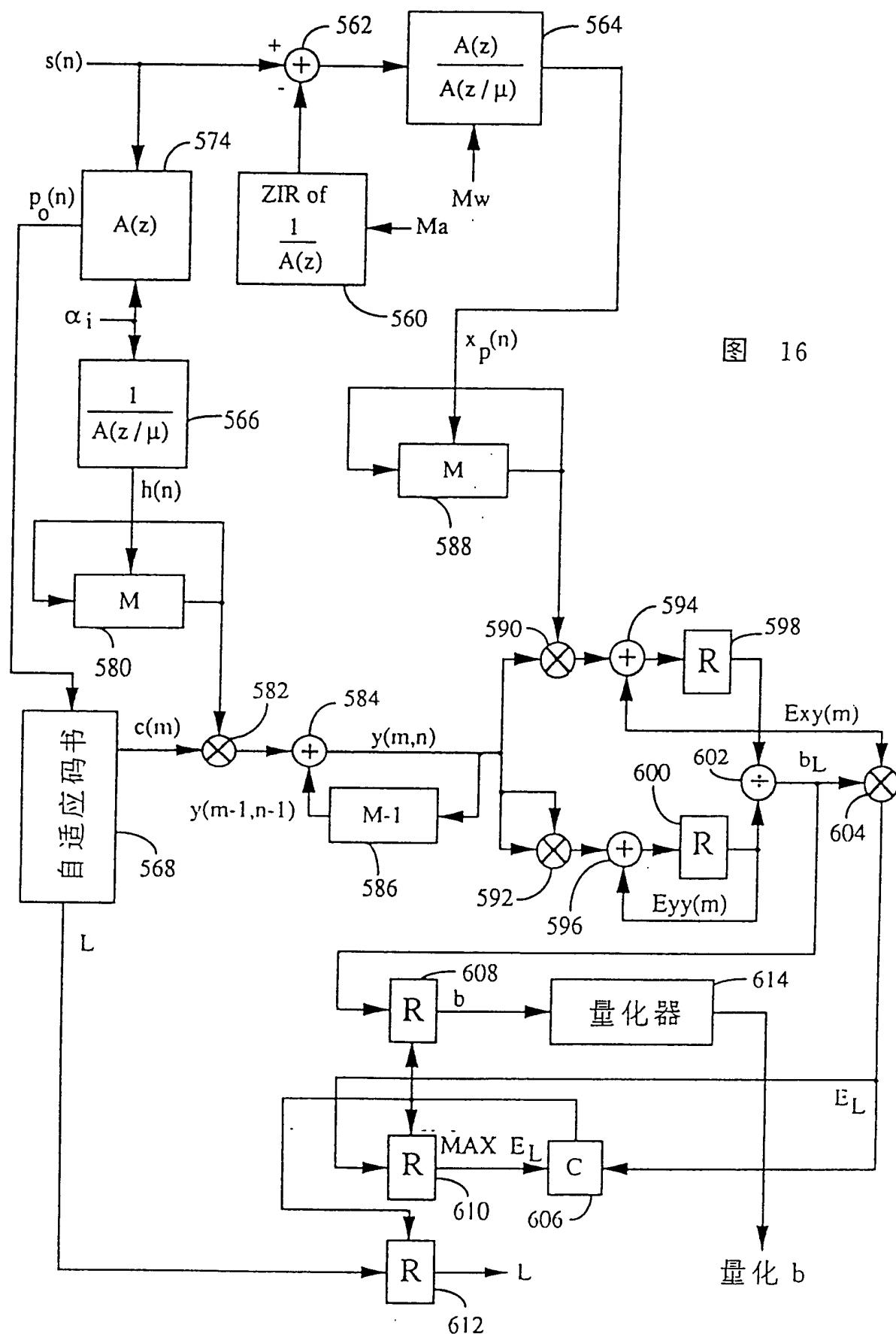
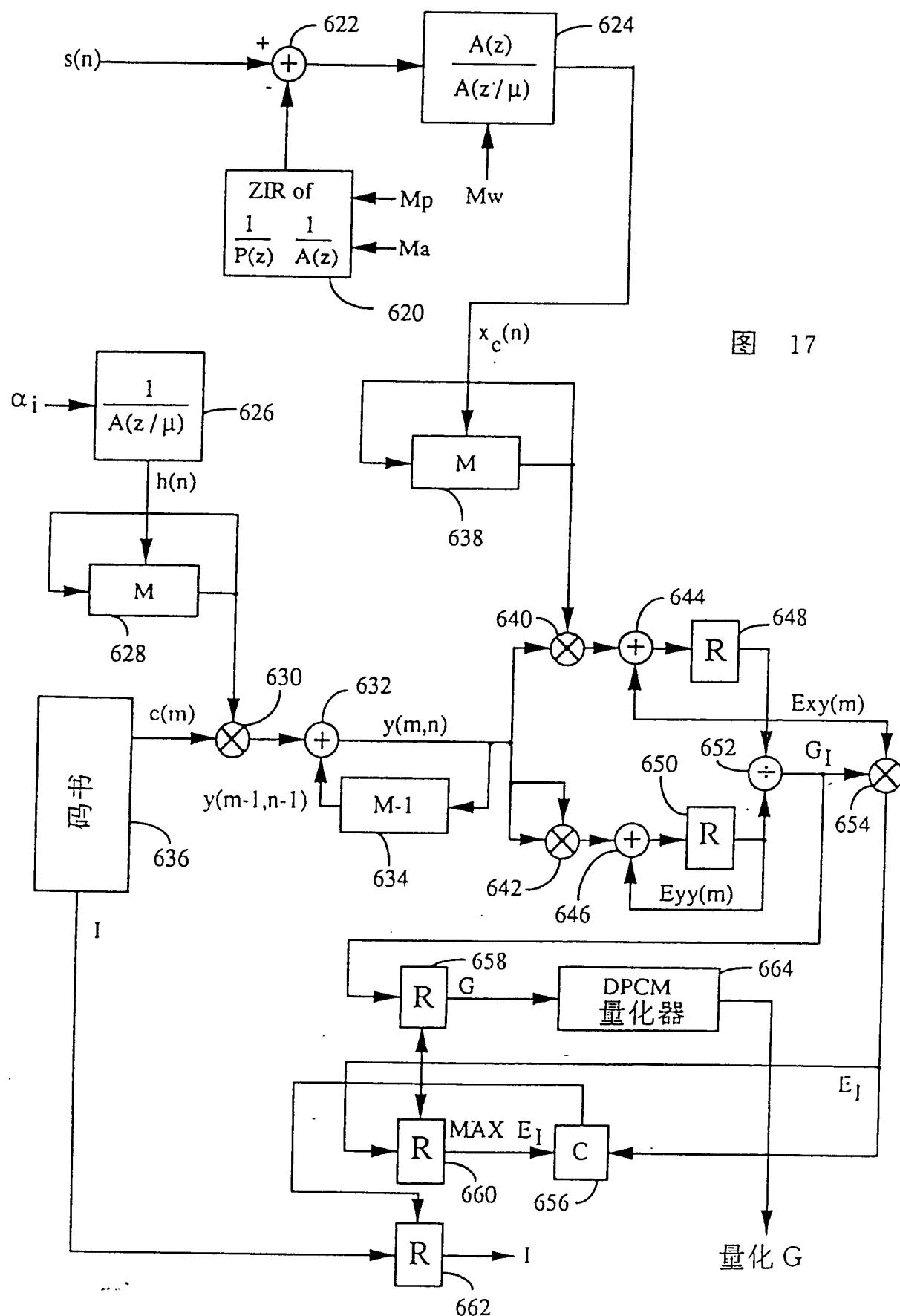


图 15





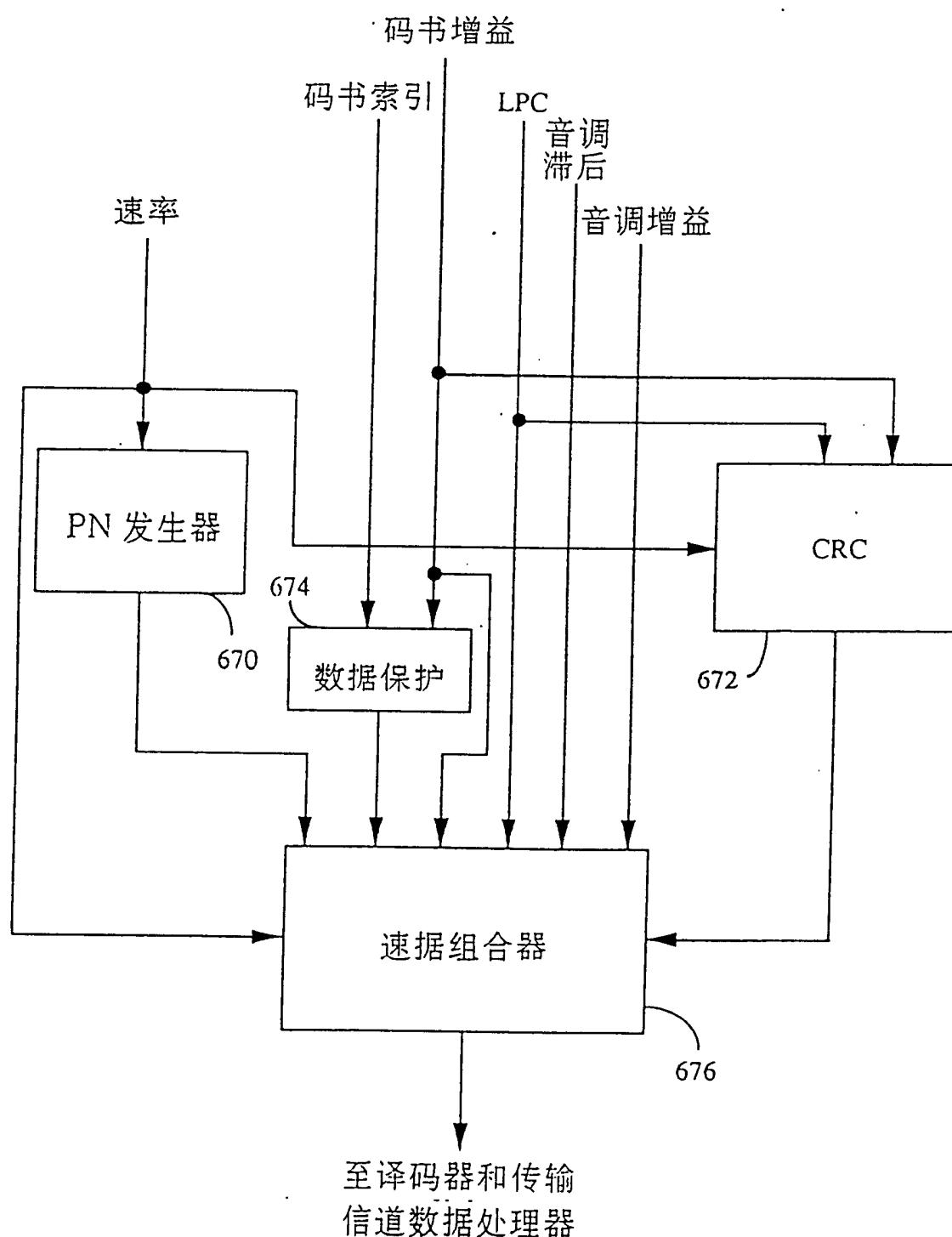


图 18

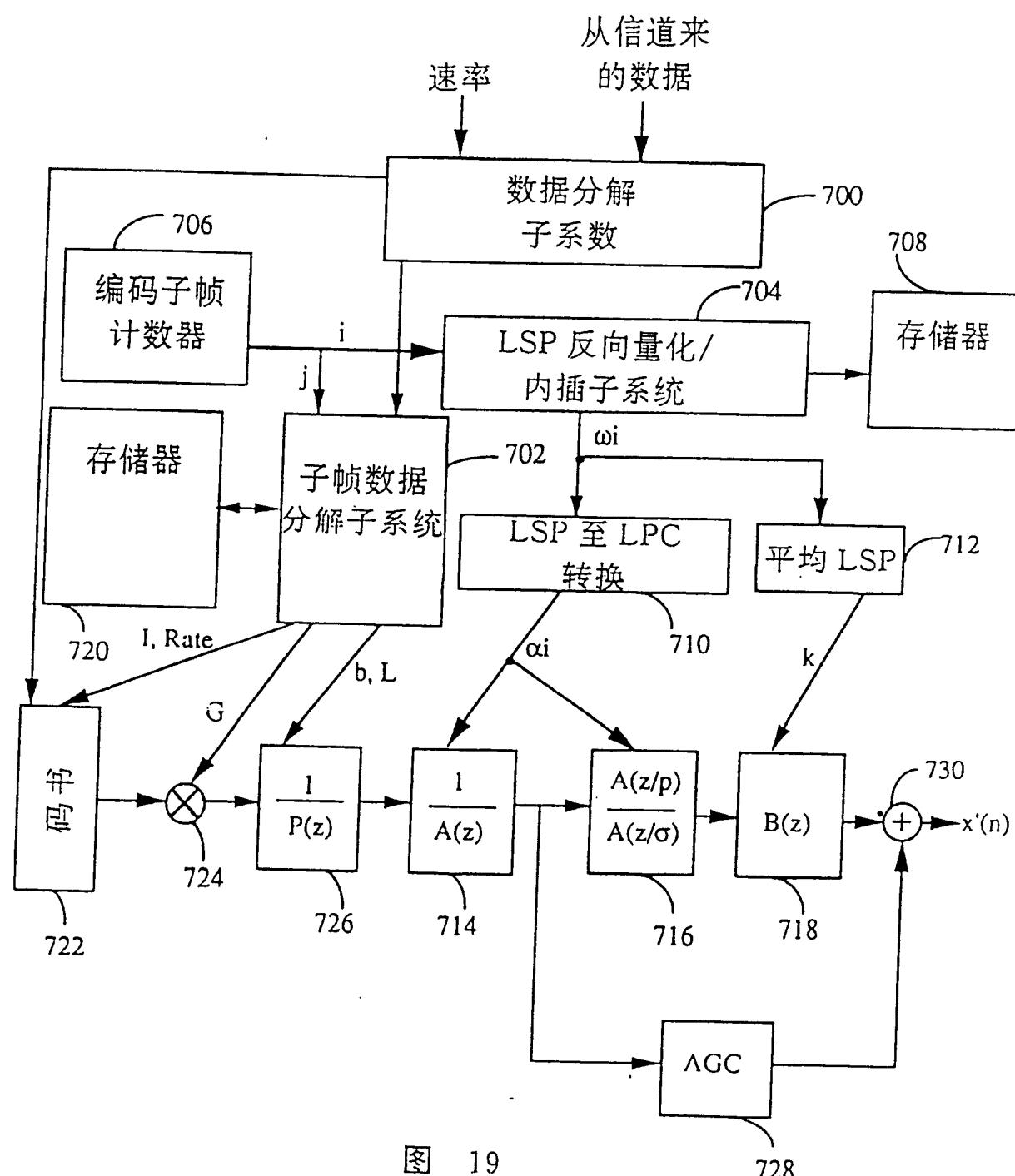


图 19

全速率帧中接收到的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子帧 4
I <sub>1</sub> , ..., I <sub>8</sub>	I <sub>1</sub> , I <sub>2</sub>	I <sub>3</sub> , I <sub>4</sub>	I <sub>5</sub> , I <sub>6</sub>	I <sub>7</sub> , I <sub>8</sub>
G <sub>1</sub> , ..., G <sub>8</sub>	G <sub>1</sub> , G <sub>2</sub>	G <sub>3</sub> , G <sub>4</sub>	G <sub>5</sub> , G <sub>6</sub>	G <sub>7</sub> , G <sub>8</sub>
L <sub>1</sub> , ..., L <sub>4</sub>	L <sub>1</sub>	L <sub>2</sub>	L <sub>3</sub>	L <sub>4</sub>
b <sub>1</sub> , ..., b <sub>4</sub>	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>
ω <sub>1,f</sub> , ..., ω <sub>10,f</sub>	ω <sub>i</sub> =0.75ω <sub>i,f-1</sub> + 0.25ω <sub>i,f</sub>	ω <sub>i</sub> =0.5ω <sub>i,f-1</sub> + 0.5ω <sub>i,f</sub>	ω <sub>i</sub> =0.25ω <sub>i,f-1</sub> + 0.75ω <sub>i</sub>	ω <sub>i</sub> =ω <sub>i,f</sub>

图 20a

半速率帧中接收到的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子帧 4
I <sub>1</sub> , ..., I <sub>4</sub>	I <sub>1</sub> , I <sub>1</sub> -20	I <sub>2</sub> , I <sub>2</sub> -20	I <sub>3</sub> , I <sub>3</sub> -20	I <sub>4</sub> , I <sub>4</sub> -20
G <sub>1</sub> , ..., G <sub>4</sub>	G <sub>1</sub> , G <sub>1</sub>	G <sub>2</sub> , G <sub>2</sub>	G <sub>3</sub> , G <sub>3</sub>	G <sub>4</sub> , G <sub>4</sub>
L <sub>1</sub> , L <sub>2</sub>	L <sub>1</sub>	L <sub>1</sub>	L <sub>2</sub>	L <sub>2</sub>
b <sub>1</sub> , b <sub>2</sub>	b <sub>1</sub>	b <sub>1</sub>	b <sub>2</sub>	b <sub>2</sub>
ω <sub>1,f</sub> , ..., ω <sub>10,f</sub>	ω <sub>i</sub> =0.75ω <sub>i,f-1</sub> + 0.25ω <sub>i,f</sub>	ω <sub>i</sub> =0.5ω <sub>i,f-1</sub> + 0.5ω <sub>i,f</sub>	ω <sub>i</sub> =0.25ω <sub>i,f-1</sub> + 0.75ω <sub>i,f</sub>	ω <sub>i</sub> =ω <sub>i,f</sub>

图 20b

八分之一速率帧中接收到的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子帧 4
I <sub>1</sub> , I <sub>2</sub>	I <sub>1</sub> , I <sub>1</sub> -20	I <sub>1</sub> -40, I <sub>1</sub> -60	I <sub>2</sub> , I <sub>2</sub> -20	I <sub>2</sub> -40, I <sub>2</sub> -60
G <sub>1</sub> , G <sub>2</sub>	G <sub>1</sub> , G <sub>1</sub>	G <sub>1</sub> , G <sub>1</sub>	G <sub>2</sub> , G <sub>2</sub>	G <sub>2</sub> , G <sub>2</sub>
L <sub>1</sub>	L <sub>1</sub>	L <sub>1</sub>	L <sub>1</sub>	L <sub>1</sub>
b <sub>1</sub>	b <sub>1</sub>	b <sub>1</sub>	b <sub>1</sub>	b <sub>1</sub>
ω <sub>1,f</sub> , ..., ω <sub>8,f</sub>	ω <sub>i</sub> =0.625ω <sub>i,f-1</sub> + 0.375ω <sub>i,f</sub>	ω <sub>i</sub> =0.625ω <sub>i,f-1</sub> + 0.375ω <sub>i,f</sub>	ω <sub>i</sub> =0.125ω <sub>i,f-1</sub> + 0.875ω <sub>i,f</sub>	ω <sub>i</sub> =0.125ω <sub>i,f-1</sub> + 0.875ω <sub>i,f</sub>

图 20c

四分之一速率帧中接收到的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子帧 4
I <sub>1</sub> (无)	PN 种子	PN 种子	PN 种子	PN 种子
G <sub>1</sub>	G <sub>1</sub> , G <sub>1</sub>	G <sub>1</sub> , G <sub>1</sub>	G <sub>1</sub> , G <sub>1</sub>	G <sub>1</sub> , G <sub>1</sub>
L (无)	无	无	无	无
b (无)	0	0	0	0
ω <sub>1,f</sub> , ..., ω <sub>5,f</sub>	ω <sub>i</sub> =0.625ω <sub>i,f-1</sub> + 0.375ω <sub>i,f</sub>			

图 20d

接收到的 空白帧的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子 帧 4
I( 无 )	无	无	无	无
G( 无 )	0	0	0	0
L( 无 )	$L_p$	$L_p$	$L_p$	$L_p$
b( 无 )	$b_p$	$b_p$	$b_p$	$b_p$
$\omega_{i,f}$ ( 无 )	$\omega_i = \omega_{i,f-1}$	$\omega_i = \omega_{i,f-1}$	$\omega_i = \omega_{i,f-1}$	$\omega_i = \omega_{i,f-1}$

NOTE:  $L_p$  AND  $b_p$  ARE SET TO L AND b VALUES OF PREVIOUS DECODING SUBFRAME WHERE  $b_p$  IS SET TO 1 IF  $b_p > 1$ .

图 21a

接收到的 消除帧的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子 帧 4
I( 无 )	$I_p + 89, I_p + 109$	$I_p + 129, I_p + 149$	$I_p + 169, I_p + 189$	$I_p + 209, I_p + 169$
G( 无 )	$0.7G_p, 0.7G_p$	$0.7G_p, 0.7G_p$	$0.7G_p, 0.7G_p$	$0.7G_p, 0.7G_p$
L( 无 )	无	无	无	无
b( 无 )	0	0	0	0
$\omega_{i,f}$ ( 无 )	$\omega_i = 0.9(\omega_{i,f-1} - \omega b_i) + \omega b_i$	$\omega_i = 0.9(\omega_{i,f-1} - \omega b_i) + \omega b_i$	$\omega_i = 0.9(\omega_{i,f-1} - \omega b_i) + \omega b_i$	$\omega_i = 0.9(\omega_{i,f-1} - \omega b_i) + \omega b_i$

NOTE:  $I_p$  AND  $G_p$  ARE THE TO I AND G VALUES OF PREVIOUS DECODING SUBFRAME AND  $\omega b_i$  IS THE BIAS VALUE OF  $\omega_i$ .

图 21b

接收到的 错误帧的参数	用到的译码子帧参数			
	子帧 1	子帧 2	子帧 3	子 帧 4
$I_1, I_8$	$I_1, I_2$	$I_3, I_4$	$I_5, I_6$	$I_7, I_8$
$G_1, G_8$	$G_1, G_2$	$G_3, G_4$	$G_5, G_6$	$G_7, G_8$
$L_1, L_4$	无	无	无	无
$b_1, b_4$	0	0	0	0
$\omega_{1,f}, \dots, \omega_{8,f}$	$\omega_i = 0.75\omega_{i,f-1} + 0.25\omega_{i,f}$	$\omega_i = 0.5\omega_{i,f-1} + 0.5\omega_{i,f}$	$\omega_i = 0.25\omega_{i,f-1} + 0.75\omega_{i,f}$	$\omega_i = \omega_{i,f}$

图 21c

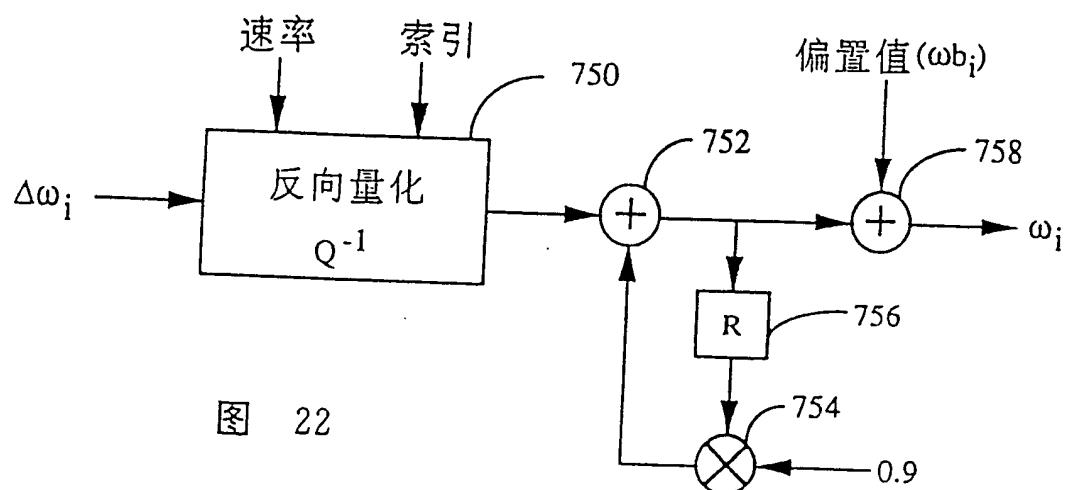
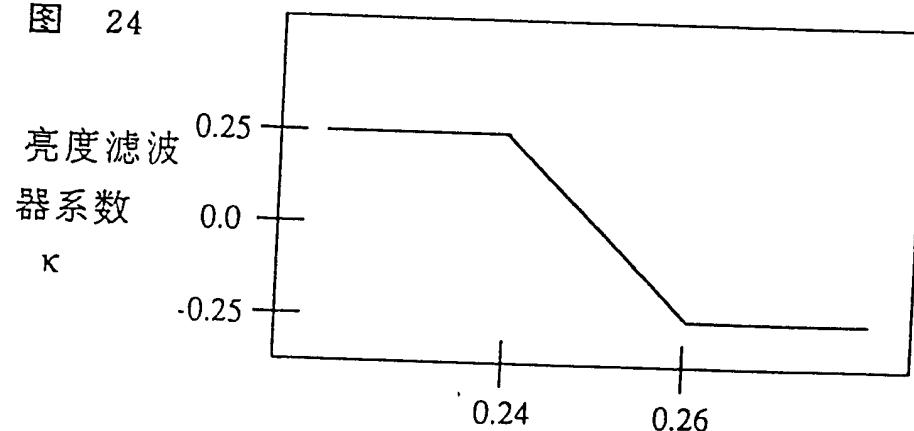


图 22



$$\text{LSP 平均值} \quad \frac{1}{10} \sum_{i=1}^{i=10} \omega_i$$

图 23

